Essays in Computational Finance

Niels Rom-Poulsen

Ph.D. Dissertation
Department of Finance
Copenhagen Business School

Thesis advisor: Carsten Sørensen

April 2006
Contents

Preface 4

1 English Summary 6

2 Bias Reduction in European Option Pricing 9
  2.1 Introduction 11
  2.2 A Short Description of the Least-Squares Monte Carlo Approach 14
    2.2.1 The Bias Reduction Method 16
  2.3 Model Setup 17
    2.3.1 Bias in Derivative Prices 18
    2.3.2 The Bias Reduction Simulation Algorithm 26
  2.4 Test Cases and the Interest Rate Model 29
    2.4.1 The Hull-White Interest Rate Model 29
  2.5 Numerical Results 30
    2.5.1 Zero Coupon Bond as Underlying 31
    2.5.2 MBS as Underlying 36
    2.5.3 Convergence 36
    2.5.4 Error Analysis 37
    2.5.5 Efficiency Issues 41
    2.5.6 Result - Antithetic Sampling 44
    2.5.7 Basis Function Sensitivity 47
  2.6 Improvements 48
    2.6.1 Inner Versus Outer Paths 48
    2.6.2 An Infinite Dimensional Set of Basis Functions 51
    2.6.3 Importance Sampling 52
  2.7 Conclusion 54
During the writing of this thesis I have been employed in Quantitative Research in Danske Bank. Quantitative Research is a unit within Danske Markets responsible for developing and implementing financial models used for pricing and risk management purposes, primarily in the fixed income market. A major issue in Quantitative Research is efficiency of the implemented models. Due to the amount of computations and the need for intraday market calibration of the models it is crucial that the pricing routines are fast and efficient. Thus, a lot of attention is devoted to find ways to speed up traditional solution methods such as Monte Carlo simulation and finite difference.

It is from this environment that the thesis has grown. It lies within the field of computational finance and much of the motivation for each project can be attributed to the need for having fast pricing routines. The projects in the thesis are results of concrete projects in Quantitative Research, and some of them are currently an integrated part of Danske Bank’s financial pricing systems.

Although fixed-rate mortgage backed securities (MBS) play a role in all three papers, this is not a thesis on MBS. However, the complexity of MBS makes them ideal for testing new numerical routines built to handle very complicated pricing problems. This is the case for the first two papers in which MBS are used to test the proposed methods. The last paper, however, deals directly with the pricing of MBS.

The paper corresponding to Chapter 4 has been accepted for publication in The Journal of Real Estate Finance and Economics. The papers corresponding to Chapter 2 and Chapter 3 have been submitted to international journals and are currently in the review process.

Preface
Acknowledgements

My work on this thesis has been influenced by many people. First of all I am grateful to my advisers Carsten Sørensen for guidance throughout the past 3 years. I would also like to thank faculty members at the Department of Finance at Copenhagen Business School for making my stay pleasant and very fruitful. Thanks to Peter Raahauge for helping me with MatLab and $\LaTeX$.

Danske Bank deserves a special thank for their financial support during the past 3 years. Without its support this thesis would not have been written. I would also like to thank my colleagues at Quantitative Research department Morten Bjerregaard, Brian Fuglsbjerg, Esben Hedegaard, Brian Huge, Henrik Lauridsen, Lars Peter Lilleøre, Kenneth Møller, Jesper Hyldgaard Nielsen, Mikkel Olsen and Nicki Rasmussen for a very inspiring environment and a good social climate. A special thank goes to my co-author Brian Huge.

Also, I am thankful to my daughter Maria for putting up with a father who has been occupied with thesis writing the past years. Finally many thanks to my companion in life, my wife Anne Mai Linh for her support and always positive attitude.

Niels Rom-Poulsen

Copenhagen, April 2006
Chapter 1

English Summary

Essay I: Bias Reduction in European Option Pricing

In this paper a new method for reducing bias in European option pricing is presented. The bias arises from computing option pay-offs using noisy price estimates of the underlying security, which due to a Jensen-inequality effect creates an upward bias in the option price. Such problems typically arise in option pricing problems where the price of the underlying security is found by crude Monte Carlo simulations. We show that if an unbiased Monte Carlo estimate of the price of the underlying security exists at option expiration, the bias can be controlled by increasing the computational effort put into computing such Monte Carlo estimates. This strategy, however, may lead to very slow pricing routines. To increase the speed we proceed by assuming that the true price of the underlying security at option expiration belongs to a space spanned by a set of basis functions. We then propose a new estimator for the price of the underlying security found by regressing the crude Monte Carlo estimates onto a set of basis functions. This new estimator is less volatile than the crude Monte Carlo estimates and thus the option price bias is reduced. If our spanning assumption is fulfilled, we prove that the resulting option price estimator is consistent. We demonstrate that the bias reduction technique can be viewed as a way to trade off the number of paths used to generate prices of the underlying with the number of crude Monte Carlo estimates used in the regression. In the limit only one path is necessary if the number of crude Monte Carlo estimates used in the regression is sufficiently high. We present two examples, one in which the spanning assumption is fulfilled and one in which it is not. In both examples the bias reduction routine effectively reduces the option price bias.
Essay II: An Algorithm for Simulating Bermudan Option Prices on Simulated Asset Prices

This paper presents an algorithm for pricing Bermudan style options written on securities so complex that they must be priced by Monte Carlo. The algorithm is of the (F. Longstaff & Schwartz, 2001) type, extended with the bias reduction technique developed in (Huge & Rom-Poulsen, 2004). As shown in their paper, using noisy price estimates of the underlying security to compute option pay-offs creates an upward bias in the option price and thus bias reduction is needed. We prove consistency of the option price estimator. A particular simple algorithm is constructed utilizing that only one path is necessary to compute the price of the underlying at any exercise date. Using the Hull-White interest rate model five test cases are presented. In the first three test cases we compute the price of a Bermudan option with 2 exercise dates written on a bullet. In "Testcase1", we use the simulation algorithm to compute the Bermudan option price and we utilize that the price of the underlying security is known in closed form at each exercise date. In "Testcase2", the closed form solution for the price of the underlying is replaced by its simulated value, but no bias reduction is performed. In "Testcase3", bias reduction is performed on the set-up from Testcase2. We demonstrate that bias reduction is needed and when used, the bias reduction technique efficiently reduces the option price bias. In "Testcase4" the price of a Bermudan option with 104 exercise dates is computed. The underlying is a bullet whose simulated price is used to compute option-payoffs. Compared to a price computed by finite difference, it is shown that the algorithm has no problem in computing the Bermudan option price. Finally, in "Testcase5", the price of a Bermudan option with 104 exercise dates written on a callable mortgage backed security is computed. The additional complexity increases the option price uncertainty, but as the number of simulations increase, convergence is achieved.

Essay III: Semi-Analytic MBS Pricing

This paper presents a multi-factor valuation model for callable mortgage backed securities (MBS). The model yields semi-analytic solutions for the value of MBS in the sense that the MBS value is found by solving a system of ordinary differential equations. Instead of modelling the conditional prepayment rate (CPR), as is customary, the pool size is the primary modelling object. It is shown that the value of a single MBS payment due at
time $t_n$ can be found by computing two expectations of the pool size at time $t_{n-1}$ and $t_n$ respectively. This is a general result independent of any interest rate model. However, if the pool size is specified in a way that makes the expectations solvable using transform methods, semi-analytic pricing formulas are achieved. The affine and quadratic pricing frameworks are combined to get flexible and sophisticated prepayment functions. We show that the model has no problem of generating negative convexity as the spot rate falls, and still be close to a similar non-callable bond when the spot rate rises.
Chapter 2

Bias Reduction in European Option Pricing

Co-authored with Brian Huge, Danske Bank
Abstract

Pricing European options using noisy price estimates of the underlying security creates a bias in the option price. We present a method to reduce this bias based on ideas from the (F. Longstaff & Schwartz, 2001) algorithm. Assuming that the true price is spanned by a set of basis functions, we prove that (i) the option price bias can be controlled by increasing the computational burden, (ii) the proposed estimator for the price of the underlying security is less volatile than the crude Monte Carlo estimate, and (iii) the resulting option price estimator is consistent.
2.1 Introduction

In this paper we propose a new technique aimed at reducing bias in European option pricing. The bias comes from using price estimates of the underlying security containing noise when computing option pay-offs. Pricing problems to which the bias reduction technique applies are typically options written on securities that are priced by simulations, i.e. problems where option pay-offs are computed using crude Monte Carlo estimates for the price of the underlying security.

The traditional approach to pricing European options is to solve the fundamental partial differential equation (PDE) common to all derivative securities with boundary conditions defining the security at hand. In contrast, the modern approach is primarily based on probability theory and states that asset prices relative to a numeraire are martingales. In this framework, prices are found as expectations of discounted terminal pay-offs, where the expectation is computed under a probability measure associated with the numeraire. The modern formulation is well suited for Monte Carlo simulations, especially for pricing complex securities where the PDE approach cannot be applied. However, the slow convergence rate of \( O(\sqrt{M}) \) in crude Monte Carlo, \( M \) being the number of simulations, has triggered an enormous amount of research trying to speed up the method. These techniques are known as variance reduction methods.

In finance, the variance reduction methods used so far are: antithetic sampling, control variates, importance sampling, stratification and low discrepancy sequences. Antithetic sampling has been used by (Boyle, 1977) to price a European call option on a dividend paying stock. His paper was the first in the finance literature to apply simulations. Other studies that have used antithetic sampling have been carried out, among others, (Hull & White, 1987), who applied simulations to price a European call option on a stock that exhibits stochastic volatility, and (Clewlow & Carverhill, 1994), who computed the price on a discrete foreign exchange look-back call option under stochastic volatility. Whereas antithetic sampling is completely independent of the derivative security to be priced, the control variate technique is developed to a particular pricing problem. The method has been used by (Boyle, 1977), (Kemna & Vorst, 1990) to price an arithmetic Asian option using the geometric Asian option as
the control variate, by (Broadie & Glasserman, 1996) to compute price derivatives in a simulation framework, and by (Clewlow & Carverhill, 1994) and (Carverhill & Pang, 1995) to price options on coupon bonds. An application of stratified sampling can be found in (Curran, 1994), an application of importance sampling can be found in (Andersen, 1996), while an example of using low discrepancy sequences can be found in (Brotherton-Ratcliffe, 1994).

In the above mentioned literature, the price of the underlying security at option expiration can relatively easily be computed from the simulated variables. However, when the price of the underlying security at option expiry is difficult to obtain, a (sub)simulation initiated at expiry may be applied to compute the price. In this case an upward bias in the option price is introduced. Intuitively, the variance of the underlying security price increases because the simulated price is only an estimate of the true price, and as such contains a stochastic error term with an expected value equal to zero and strictly positive variance. Reducing the bias can only be done by lowering the variance on the error, either by using variance reduction methods or by increasing the number of simulations.

In this paper we propose a new variance reduction technique especially designed to reduce the bias resulting from using simulated prices of the underlying security when computing options pay-offs. The idea is to use all information available from the simulated prices at option expiry. Using regression, variations in the simulated prices are divided into a systematic component and its residual, which is primarily noise. In this way we are able to filter away noise, i.e. variations not stemming from the model. Under fairly restrictive assumptions we can prove that (i) the option price bias can be controlled by increasing the computational burden, (ii) our alternative estimator for the price of the underlying security is less volatile than the crude Monte Carlo estimator, and finally (iii) consistency of the option price estimator that is constructed by using the price estimate from the regression in the option pay-offs. We demonstrate that the method is applicable to any quality of the crude Monte Carlo estimates for the price of the underlying security. This means that for a fixed computational budget there is a trade-off between improving the crude Monte Carlo price estimates of the underlying security and the number of simulations between today and option expiry. The latter will improve the option
price estimate but in general will not reduce the option price bias. However, for the method we propose, the option price bias will also be reduced and this makes our method particularly efficient when the cost of improving the crude Monte Carlo estimates of the underlying security is high compared to the cost of simulating between today and option expiry. The method can be combined with other variance reduction techniques and is very easy to implement.

The bias reduction is a result of replacing the crude Monte Carlo estimate of the price of the underlying security with a least squares Monte Carlo estimate found using all the simulated paths. The crude Monte Carlo estimate of the price of the underlying security at option expiry is an estimate of a conditional expectation. In our approach we approximate this expectation with a linear function of some basis functions. From this perspective our proposed method is a direct application of the method for estimating the continuation value in the (F. Longstaff & Schwartz, 2001) algorithm for pricing American options with Monte Carlo. We simply regress future simulated prices of the underlying security onto a set of basis functions and use this expression to calculate option pay-offs instead of the raw Monte Carlo simulated price estimates. Both in (F. Longstaff & Schwartz, 2001) and in our model, least squares is used to find the estimate of an iterated expectation. However, there is a difference in the way the estimate is used. Our use of the least squares estimate has a much higher impact on the option price because we use it directly in the option pay-off. This is in contrast to (F. Longstaff & Schwartz, 2001), who use the least squares estimate to determine the optimal exercise boundary. The assumption underlying the (F. Longstaff & Schwartz, 2001) algorithm is therefore much more critical to our model than to theirs.

Numerical investigations of the proposed method are done using two test cases in the Hull-White interest rate model. In the first case, we price a European call option on a zero coupon bond. For this case, all assumptions of the model are fulfilled and we can therefore compare our proposed technique with closed form solutions. In the second case, we price a European option on a callable mortgage backed bond. Here the assumptions are not fulfilled, but nevertheless we show that the bias reduction technique is very effective in reducing the option price bias.

In Section 2.2 we give a very short introduction to regression based algorithms,
which are used to value Bermudan style options by Monte Carlo simulations. In Section 2.3 the model is set up and the main results are presented. The proofs are given in Appendix 2.8. Also, the simulation algorithms are presented. In Section 2.4 the test cases are described and, for the sake of completeness, well-known results for the Hull-White model are given. In Section 2.5 numerical results are presented. For the zero coupon bond case, only the bias reduction technique is employed but for the MBS case, an improvement using antithetic sampling between today and option expiry is also considered. Further improvements of the method are discussed in Section 2.6 but only using the zero coupon bond example. Section 2.7 presents our conclusions. In Appendix 2.9 a short description of MBS Monte Carlo valuation is given along with an intuitive example showing why the bias reduction technique works in this case. Finally, Appendix 2.10 displays the details of our computational test cases.

2.2 A Short Description of the Least-Squares Monte Carlo Approach

The intention of this section is to give a very short and informal introduction to the Least-Squares Monte Carlo approach for simulating Bermudan option values. Since the idea in our Bias Reduction Method is inspired by these types of algorithms, the Least-Squares Monte Carlo algorithm is presented at this stage. For an in-depth description please see (F. Longstaff & Schwartz, 2001), Chapter 8 in (Glasserman, 2004), (Tsitsiklis & Van Roy, 2001) and, for convergence results of the Longstaff-Schwartz algorithm, (Clemente, Lamberton, & Protter, 2002). The presentation in this section is taken from (Glasserman, 2004).

The difficulty in valuing a Bermudan style option by simulation lies in the fact, that Monte Carlo simulation works forward in the time dimension whereas the dynamic programming principle, which must be used to find the optimal exercise strategy, works backward in the time dimension. This difficulty is basically what is overcome in the Least-Squares Monte Carlo algorithms. First the state space is simulated, and then the dynamic programming principle is applied to the simulated paths. When performing the dynamic programming principle, the value of immedi-
ate exercise must repeatedly be compared with the value of postponing exercise. The latter is known as the continuation value, and it is this value that is approximated by linear functions of the state variables.

In the following, we only consider problems that can be formulated through an \( \mathbb{R}^d \)-valued Markov state process \( \{X(t), 0 \leq t \leq T\} \), which records all necessary information about the relevant financial variables. We only consider Bermudan options, and thus it is only necessary to know the value of the state process at the exercise dates \( t_0 < t_1 < \ldots < t_n \). The discrete time process \( X_0 = X(0), X_1, \ldots, X_n \) is then a Markov chain on \( \mathbb{R}^d \). We will use the notation \( X_i = X(t_i) \) and assume that \( X_i \) can be simulated without discretization errors at the exercise dates. The pay-off to the option holder from exercise at time \( t_i \) given \( X_i = x \) is denoted \( h_i(x) \) and is measured in time \( t_0 \)-dollars. The same goes for \( V_i(x) \), which denotes the value of the option at \( t_i \) given \( X_i = x \). We want to determine \( V_0(X_0) \) recursively by the dynamic programming principle.

The continuation value at date \( t_i \) in state \( X_i = x \) is equal to the expected value today of tomorrow’s option value conditioned on the current state. I.e.

\[
C_i(x) = \mathbb{E}^Q [V_{i+1}(X_{i+1}) | X_i = x], \quad i = 1, \ldots, n - 1
\]

Clearly, the continuation value at the last exercise date is 0 and thus, the dynamic programming principle can be stated as

\[
C_n \equiv 0
\]

\[
C_i(x) = \mathbb{E}^Q \left[ \max (h_{i+1}(X_{i+1}), C_{i+1}(X_{i+1})) | X_i = x \right]
\]

\[i = 0, \ldots, n - 1\]

and the Bermudan option value is given by \( C_0(X_0) \). The value function of the Bermudan option is given by

\[
V_i(x) = \max (h_i(x), C_i(x))
\]

where it is implicitly assumed that \( h_0(x) = 0 \).

Equation (2.1) is the regression of the option value \( V_{i+1}(X_{i+1}) \) on the current state \( x \). This suggests a valuation procedure: approximate the continuation value in Equation (2.1) by a linear combination of functions of the current state \( x \). These
functions are known as basis functions, and their coefficients are typically estimated by least squares regression.

The main assumption in Least-Squares Monte Carlo is:

$$\mathbb{E}^Q[V_{i+1}(X_{i+1}) | X_i = x] = \sum_{b=1}^{B} \beta_b \psi_b(x)$$

for some basis functions $\psi_b : \mathbb{R}^d \rightarrow \mathbb{R}$ and constants $\beta_b$, $b = 1, \ldots, B$. We now have

$$C_i(x) = \beta_i^T \psi(x)$$

where

$$\beta_i^T = (\beta_{i1}, \ldots, \beta_{iB}), \quad \psi(x) = (\psi_1(x), \ldots, \psi_B(x))^T$$

From (Glasserman, 2004) we have reproduced the complete algorithm in Algorithm 1.

**Algorithm 1 Regression-Based Pricing Algorithm**

1: Simulate $M$ independent paths $\{X_{1j}, \ldots, X_{nj}\}$, $j = 1, \ldots, M$ of the Markov chain
2: At terminal nodes, set $\hat{V}_{nj} = h_n(X_{nj})$, $j = 1, \ldots, M$
3: Apply backward induction:
4: for $(i = n - 1$ to 1) do
5: given estimated values $\hat{V}_{i+1,j}$, $j = 1, \ldots, M$, use regression to calculate $\hat{\beta}_i$ (the estimate of $\beta_i$);
6: set $\hat{V}_{ij} = \max \left( h_i(X_{ij}), \hat{C}_i(X_{ij}) \right)$, $j = 1, \ldots, M$
7: with $\hat{C}_i(x) = \hat{\beta}_i \psi(x)$
8: end for
9: Set $\hat{V}_0 = (\hat{V}_{11} + \ldots + \hat{V}_{1M}) / M$

Two approximations are made in the Least-Squares approach described in Algorithm 1. The first approximation consists of approximating the continuation value in Equation (2.1) by a finite number of basis functions, i.e. having $B < \infty$. The second approximation consists of using a finite number of simulations of the Markov chain. It is shown in (Tsitsiklis & Van Roy, 2001) that if Equation (2.1) holds at all $i = 1, \ldots, n - 1$, then the estimate $\hat{V}_0$ converges to the true value $V_0$ as $M \rightarrow \infty$.

**2.2.1 The Bias Reduction Method**

In the Bias Reduction Method presented in this paper, the price of the underlying security at option expiration $T$ is approximated, just like the continuation value in
Equation (2.1), by a finite set of basis functions. The basic idea is to construct a functional relation between the price of the underlying security at option expiration and the state at that time. This is, in most cases, clearly an approximation but for the cases we consider in this paper, it is a good approximation. The functional relation is found by specifying a set of basis functions and then use least squares to estimate the coefficients to the basis functions using the crude Monte Carlo estimates as dependent variables. Once the functional relation between the price of the underlying security and the current state has been estimated, the estimated function is used to compute option pay-offs. This is different from the (F. Longstaff & Schwartz, 2001) approach, in which the estimated continuation value is only used to determine the exercise region and not used directly to compute pay-offs. In this respect, we are much more in line with (Tsitsiklis & Van Roy, 2001), who use the estimated function of the continuation value as the Bermudan option value, as can be seen in line 6 of Algorithm 1.

If the price of the underlying security happens to belong to the space spanned by the basis functions we are able to prove some analytical results about the convergence of the resulting option price estimator. Basically, we have that if the price of the underlying security is spanned by the basis functions, the noise contained in the crude Monte Carlo estimates can effectively be removed. This is the same spanning assumption used to prove convergence in (Tsitsiklis & Van Roy, 2001).

We now turn to the bias reduction model which is the primary objective of this paper.

### 2.3 Model Setup

A completed filtered probability space \((\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t=0}^U, \mathbb{Q})\) is taken as given, and we let the filtration be generated by the relevant state processes in the economy. The state variables are given by an \(\mathbb{R}^D\)-valued Markov process \(\{X(t), 0 \leq t \leq U\}\) recording all relevant financial information in the economy. Sometimes the Markov property can be achieved by augmenting the state vector to include supplementary variables. An equivalent martingale measure, \(\mathbb{Q}\), is assumed to exist under which all pricing are done. We do not assume that the martingale measure is unique, the
particular martingale measure used is found by calibrating to market prices. Under the equivalent martingale measure, $Q$, prices are computed by

$$V_t = \mathbb{E}_t^Q \left[ h_T \exp \left( - \int_t^T r_s \, ds \right) \right]$$

(2.3)

where $V_t$ is the time $t$ value of the time $T$ pay-off $h_T$, $t \leq T \leq U$, and $r_t$ is the spot interest rate. In Equation (2.3) we have implicitly used the shorthand notation $V_t = V(X_t)$, $h_T = h(X_T)$ and $r(X_s) = r_s$, which will be used in the rest of the paper.

In some cases the relevant expectations in (2.3) can be calculated analytically if the joint distribution of $\exp \left( - \int_t^T r_s \, ds \right)$ and $h_T$ is known. However, in many cases numerical routines, such as simulations, must be used in evaluating $V_t$. In a general setup, the pay-off, $h_T$, may only be available through a numerical routine, as is the case when the price of the underlying security can only be found numerically. The specific problem considered in this paper is the case where both $V_t$ and $h_T$ must be evaluated by simulations.

### 2.3.1 Bias in Derivative Prices

**Option Bias**

When $h_T$ is computed numerically by simulations, it induces a systematic error in the evaluation of $V_t$ as made precise in the following proposition, where $h_T = f(P_T, K)$ is the pay-off from a European (call/put) option with strike $K$. We consider the case where the option pay-off, $h_T$, is a function of the value of an underlying contract with price $P_T$, where $P_T$ is computed by simulations. The important feature is that $h_T$ is convex as a function of the underlying contract price $P_T$. Proposition 2.1 states that the derived price, $V_t$, in this case will be systematically upward biased (see also (Glasserman, 2004) p. 15), but it also provides an upper bound for this bias.

**Proposition 2.1** Assume $\hat{P}_T$ is an unbiased estimator of the true price of the underlying security $P_T$, i.e.

$$\hat{P}_T = P_T + \hat{\epsilon}$$

(2.4)

$$\mathbb{E}_t^Q [\hat{\epsilon}] = 0$$

(2.5)

$$\text{Var}_t^Q [\hat{\epsilon}] = \sigma_\epsilon^2$$

(2.6)
Let \( c_t \) be the true price of a European call/put option with strike \( K \) and convex pay-off function \( f(P, K) \). Let \( \hat{c}_t \) be the estimated price of \( c_t \) where the option pay-off is computed using \( \hat{P}_T \), i.e.

\[
\hat{c}_t = \mathbb{E}_t^Q \left[ e^{-\int_t^T r_s \, ds} f(\hat{P}_T, K) \right]
\]

Then

\[
c_t \leq \hat{c}_t \leq c_t + \sigma \mathbb{E}_t^Q \left[ e^{-\int_t^T r_s \, ds} \right]
\]

**Proof:** See Appendix 2.8.1

Note that Assumption (2.5) yields \( \mathbb{E}_T^Q [\hat{P}_T] = P_T \), and Assumptions (2.5) and (2.6) yield \( \mathbb{E}_T^Q [\bar{\epsilon}] = 0 \), \( \text{Var}_T^Q [\bar{\epsilon}] = \sigma^2 \). The stochastic nature of \( \bar{\epsilon} \) is not used anywhere in the proof of Proposition 2.1, but as the number of simulations used to determine \( \hat{P}_T \) increases, \( \bar{\epsilon} \) will be approximately normally distributed with mean zero and variance \( \sigma^2 \). It is important, however, that an unbiased estimate exists, i.e. \( \bar{\epsilon} \) has an expected value of zero and that \( \bar{\epsilon} \) has finite variance.

Proposition 2.1 shows that the noise in the price of the underlying security results in an upward bias in the option price. The intuitive explanation for this bias is that the noise corresponds to a higher volatility on the underlying security. This means that the prices used for computing option pay-offs are too volatile, which leads to a higher option price. However, the option price bias can be reduced by lowering the variance on the price estimates of the underlying security either by employing variance reduction methods, or by increasing the number of simulations used to price the underlying security at option expiry. This, however, can be very time consuming and in the rest of this section we present an alternative method to reduce the option price bias.

**Crude Monte Carlo Simulations**

Before we proceed we will describe the simulation algorithm we call crude Monte Carlo, and introduce the two central concepts outer simulations and inner simulations.
Crude Monte Carlo simulations can be described as follows. Suppose that a sequence of independent identically distributed (i.i.d.) price estimates, \( \{ \hat{V}_i^t, i = 1, \ldots, M \} \) each with mean \( V_t \) and variance \( \sigma^2 \) has been calculated where \( M \) is the total number of replications. Usually \( \sigma^2 \) is also unknown and must be estimated from the sample requiring the pay-off function to be square integrable. From the strong law of large numbers we know that if \( \hat{V}_i^t \) are unbiased estimates of \( V_t \), the sample mean

\[
\hat{V}_t = \frac{1}{M} \sum_{i=1}^{M} \hat{V}_i^t
\]

(2.7)

converges to the true mean \( V \) (the expectation in (2.3)) as \( M \to \infty \). Furthermore, the central limit theorem\(^2\) states that \( \hat{V}_t \) will be normally distributed with mean \( V \) and variance \( \sigma^2/M \). A probabilistic error bound is given by (\( \hat{V}_t - sz_{\alpha/2}/\sqrt{M}, \hat{V}_t + sz_{\alpha/2}/\sqrt{M} \)), the \( 1-\alpha \) confidence interval, where \( z_{\alpha/2} \) is the \( 1-\alpha/2 \) quantile of the standard normal distribution and \( s \) is the estimated standard deviation of \( \hat{V}_i^t \). This illustrates one of the weaknesses of Monte Carlo, namely that the result is only an estimate of the true price. However, we can make the interval arbitrarily small by increasing \( M \) or by lowering \( \sigma \). If it is costly to compute new paths, as is usually the case, decreasing \( \sigma \) will generally be the fastest way to generate better estimates. This is emphasized by the fact that decreasing \( \sigma \) by a factor of 10 gives the same variance reduction as a 100 fold increase in \( M \), other things being equal.

When we want to simulate the value of a European option, the crude Monte Carlo method specializes in the following way. We describe a situation where the underlying security also must be valued by Monte Carlo simulations, which is the situation displayed in Figure 2.1. First \( M \) paths are simulated between today and option expiration. We will refer to this number as the number of outer simulations. At option expiration, we need to compute the state dependent option pay-off conditioned on each single outer simulation. Thus we need to know the value of the underlying security, which will be determined by initiating a sub-simulation conditioned on the given state (outer simulation). The number of simulations in that sub-simulation, \( N \), we will refer to as the number of inner simulations. The option

\(^2\)We are using these statistical theorems in spite of the non-randomness induced by the use of a computer to generate the random numbers. Random numbers generated on a computer are labelled pseudo-random numbers.
value is now determined by averaging the discounted option pay-offs. A version of the algorithm, for the Hull-White interest rate model, is displayed in Section 2.3.2, Algorithm 2.

As demonstrated in Proposition 2.1, the crude Monte Carlo algorithm leads to an upward bias in the option price. In the next section we thus introduce a method to handle this upward bias.

**Bias Reduction using Least Squares**

In this section we set up the model which is used to reduce the option price bias arising from using price estimates of the underlying security that contain a noisy element. We will suggest a method based on the Least-Squares Monte Carlo ideas in (F. Longstaff & Schwartz, 2001) and compare it to a crude Monte Carlo method. In the following we will assume that the true price of the underlying security is given as a finite linear combination of some basis functions, as formalized in Assumption 2.1 below. In this setup we show that using least squares estimates for computing
option pay-offs results in a lower bias than when crude Monte Carlo estimates are used. Also, we show that the technique can be viewed as a way to substitute inner simulations (simulations of the underlying security price at option expiration) with outer simulations (simulations of the option price) making the pricing algorithm very fast and accurate compared to crude Monte Carlo simulations. Our method is particularly efficient for problems where outer simulations are cheap relative to inner simulations, which is typically the case for short-termed options on long-lived assets. In Section 2.5 we examine the effect on estimated option prices in a more realistic situation where the linear combination of the basis functions only approximates the true price of the underlying security. Using a numerical example, we demonstrate that for relatively few basis functions a pricing algorithm based on option pay-offs computed with least squares estimates of the price of the underlying is generally more efficient than an algorithm where option pay-offs are computed with crude Monte Carlo price estimates of the underlying security.

**Assumption 2.1** Assume that the true time $T$ price of the underlying security can be written as a linear combination of $B$ basis functions ($B < \infty$),

$$P_T = L(X_T)^\top b$$

(2.8)

where $X_T$ is a time $T$-measurable $D \times 1$ vector of state variables, $L$ and $b$ are $B \times 1$ vectors. $L$ is a vector of basis functions taking as input the vector $X_T$, and $b$ is the vector of coefficients to the basis functions. $\top$ denotes the matrix transpose operator.

In the following we will sometimes suppress the dependence of the basis functions on the state variables in order to lighten the notation, i.e. $L(X_T)$ is written as $L$.

Assume that for all simulated outer paths, $i = 1 \ldots, M$, there exists an unbiased crude Monte Carlo estimate of the underlying security, $\hat{P}_T^{MC,i}(N)$, obtained by $N$ inner simulations. Furthermore, assume that the variance of these estimates are equal and that the noise terms are uncorrelated, i.e.
Assumption 2.2

\[ \hat{P}_T^{MC,i}(N) = P^i_T + \tilde{\epsilon}^{MC,i}(N) \] (2.9)

\[ \mathbb{E}_T^Q[\tilde{\epsilon}^{MC,i}(N)] = 0 \] (2.10)

\[ \text{Var}_T^Q[\tilde{\epsilon}^{MC,i}(N)] = \sigma^2, \forall i \] (2.11)

\[ \text{Cov}_T^Q[\tilde{\epsilon}^{MC,i}(N), \tilde{\epsilon}^{MC,j}(N)] = 0 \] (2.12)

In general the variance in (2.11) may depend on the specific path \( i \). However, the number of simulations at the end of a given path \( i \) can always be chosen so that (2.11) is fulfilled for all the simulated paths. In that case, \( N \) will not necessarily be equal across the simulated paths. The assumption about constant variance of the error term is used to prove the theoretical results below, but in practice it does not seem to be important as demonstrated in Section 2.5 where \( N \) is constant across all the simulated paths. Assumption (2.12) means that the error terms are independent of each other, i.e. that the error term along the \( i \)th outer path is independent of the error term along the \( j \)th outer path. This will be fulfilled whenever we can generate independent sample paths of the state vector \( X_t \), which we assume can be done. In the following, the superscripts ”MC,i” will generally refer to the crude Monte Carlo estimate at the end of the \( i \)th outer path.

We want to estimate Model (2.8) by regression, using the crude Monte Carlo estimates in (2.9) as dependent variables. Combining (2.8) and (2.9) yields

\[ \hat{P}_T^{MC,i}(N) = \mathbf{L}(X_i^T)b + \tilde{\epsilon}^{MC,i}(N) \] (2.13)

Stacking our \( M \) Monte Carlo estimates into the vector \( \hat{P}_T^{MC} \) we get

\[ \hat{P}_T^{MC} = \mathbf{L}b + \tilde{\epsilon}^{MC} \] (2.14)

where \( \hat{P}_T^{MC} \) is a \( M \times 1 \) vector of Monte Carlo simulated prices of the underlying security, \( \mathbf{b} \) is the \( B \times 1 \) vector of coefficients to the basis functions, \( \tilde{\epsilon}^{MC} \) is a \( M \times 1 \) vector of error terms, and \( \mathbf{L} \) is a \( M \times B \) matrix\(^3 \). The \( i \)th row of \( \mathbf{L} \) is the \( 1 \times B \)
vector $L(X_i^T)^T$, which represents the value of the vector of basis functions along the $i$th path. In the following we will denote this $i$th row of $L$ by $L^i (L^i = L(X_i^T)^T)$.

**Option Pricing Using Least Squares Estimates of the Underlying**

Before option pay-offs can be computed the Model in (2.14) must be estimated. This is done with ordinary least squares, which yields the unbiased estimator, see (Greene, 2002)

$$\hat{b} = (L^T L)^{-1} L^T \hat{P}^MC_T$$

$$\mathbb{E}_T^Q \left[ \hat{b} \right] = b$$
$$\text{Var}_T^Q \left[ \hat{b} \right] = \sigma^2 \left( L^T L \right)^{-1}$$

(2.15)

When calculating option pay-offs we use least squares Monte Carlo estimates instead of using the crude Monte Carlo estimates $\hat{P}^MC_T$. The least squares estimate, $\hat{P}^{LS}_T$, of the price of the underlying security along path $i$ is given as

$$\hat{P}^{LS}_T = L^i \hat{b}$$

(2.16)

Also, define $\hat{P}^{LS}_T = L \hat{b}$ as the $M \times 1$ vector of least squares estimates of the vector of true prices, $P_T$, of the underlying security at option expiration with the $i$th row equal to (2.16). It follows from Assumption 2.1 and Assumption 2.2 that $\hat{P}^{LS}_T$ is an unbiased estimator of the true price $P_T$. Furthermore, the variance of $\hat{P}^{LS,i}_T$ is lower than the variance of $\hat{P}^{MC,i}_T$, thereby reducing the option price bias as shown in Proposition 2.2 below. These properties mean that $\hat{P}^{LS}_T$ is a more efficient estimator of the true prices $P_T$ than $\hat{P}^{MC}_T$.

**Proposition 2.2** Assume that the least squares price estimates have been estimated using crude Monte Carlo simulated prices of the underlying security. Then

$$\mathbb{E}_T^Q \left[ \hat{P}^{LS}_T \right] = P_T$$
$$\text{Var}_T^Q \left( \hat{P}^{LS}_T \right) \leq \text{Var}_T^Q \left( \hat{P}^{MC}_T \right)$$

Proof: See Appendix 2.8.2
Proposition 2.2 indicates that using least squares estimates when calculating option pay-offs could reduce the option price bias induced by the noisy estimates of the prices of the underlying security.

By regressing crude Monte Carlo estimates onto a set of basis functions, the variance is decomposed into a component stemming from variations in the basis functions and a residual component contained in a space orthogonal to the space spanned by the basis functions. When the true price belongs to the space spanned by the basis functions, the residual space only contains noise. The option price bias is therefore effectively removed when using least squares estimates from the space of true prices instead of crude Monte Carlo estimates for computing option pay-offs.

Consistency of the Least Squares Computed Option Price

Given Assumption 2.1, the option price computed using least squares estimates of the option pay-offs is a consistent estimator of the true option price. The intuition is that given (2.8), the only error on $\hat{P}_{LS}^T$ comes from $\hat{b}$ not being close to $b$ and this error disappears for the number of observations approaching infinity. Thus, for any given number of inner simulations, one can increase the number of outer simulations until convergence has been achieved.

**Proposition 2.3** Let $M$ and $N$ be the number of outer and inner simulations respectively. Define

$$Q_M = \frac{1}{M}L^\top L$$

and

$$\hat{c}_t^{LS} = \mathbb{E}_t^Q \left[ e^{-\int_t^T r_s \, ds} f(\hat{P}_{LS}^T, K) \right]$$

Given Assumption 2.1, Assumption 2.2 and that

$$\lim_{M \to \infty} Q_M = Q$$

is a positive definite matrix, so that $Q_M^{-1}$ exists from a certain step, then for any choice of $N$, the least squares computed option price fulfils

$$\hat{c}_t^{LS} \to c_t \text{ as } M \to \infty$$

□
Assumption (2.19) states that the sample second order matrix, \( \frac{1}{n}L^TL \), approaches the population second order matrix. This can be seen by noting that \( \frac{1}{n}L^TL \) is an average of \( L^TL \) and by the law of large numbers, the average will approach the true mean as the number of observations increases. We do not check for this in our numerical test cases below.

Besides proving consistency of the least squares option price estimator, \( \hat{c}_{LS}^t \), Proposition 2.3 also suggests that it is possible to substitute inner simulations with outer simulations. Since the proposition is valid for any \( N \), we can use few inner simulations and compensate by increasing the number of outer simulations. This is especially valuable when the computational burden of generating outer simulations is low compared to the computational burden of generating inner simulations.

### 2.3.2 The Bias Reduction Simulation Algorithm

In this section the bias reduction algorithm is described. We are primarily interested in valuing European options on path dependent securities, however, we restrict ourselves to cases in which the option pay-off depends on the value of the state vector at a fixed set of dates \( t = t_0 < t_1 < \cdots < t_n = T \). When we later value a European option on a mortgage backed security, the set of dates that influences future option pay-offs are the payment dates of the underlying bond.\(^4\)

The algorithm has been tailor made to the Hull-White interest rate model. The reason is that we can use results resting on the Gaussian structure of the Hull-White model to speed up the algorithm. Especially we can use a result from (Gandhi & Hunt, 1997), saying that the zero coupon price for the period \([t_{i-1}, t_i]\) can be computed exactly when the spot rates at the end points are known. Thus

\[
P_{zcb}(t_{i-1}, t_i) = E^Q \left[ \exp \left( - \int_{t_{i-1}}^{t_i} r_s \, ds \right) | r_{t_{i-1}}, r_{t_i} \right]
\]

is known in closed form. In this particular case, the state vector consist of the spot rate and the variables that influence the option pay-offs.

\(^4\)For a mortgage backed security these variables are the pool factor and tranche weights, see Appendix 2.9.
A simple valuation scheme would be to define the pathwise option price estimator by
\[ c_i(t) = \exp \left( -\sum_{j=1}^{n} r^i_{t_j}(t_j - t_{j-1}) \right) \max \left( P^i_T - K, 0 \right) \]
However, with the result from (Gandhi & Hunt, 1997) in mind, we instead define the pathwise option price estimator as
\[ c_i(t) \equiv \prod_{j=1}^{n} P^{zcb}_i(t_{j-1}, t_j) \max \left( P^i_T - K, 0 \right) \]
\[ = P^{zcb}_i(t, T) \max \left( P^i_T - K, 0 \right) \]  
(2.21)
The advantage is that if the value of the underlying security at option expiration only depends on the dates \( t_1, \ldots, t_n \), no discretization errors are made.

For securities that are priced by simulations, we replace \( P^i_T \) with \( \hat{P}^i_T \) and define the path estimator as
\[ \hat{c}_i(t) \equiv P^{zcb}_i(t, T) \max \left( \hat{P}^i_T - K, 0 \right) \]
Then we form the following simulation estimator of the option price \( \hat{c}_i \) by
\[ \hat{c}_i = \frac{1}{M} \sum_{i=1}^{M} \hat{c}_i \]
\[ = \frac{1}{M} \sum_{i=1}^{M} P^{zcb}_i(t, T) \max \left( \hat{P}^i_T - K, 0 \right) \]

In Algorithm 2 below, the procedure described above for crude Monte Carlo simulation is shown in pseudo code. The algorithm prices a European option written on a path dependent security whose price at option expiration must be computed by simulations. Comments are put in \{\}.

As an alternative, our proposed algorithm is shown in pseudo code in Algorithm 3. When using the least squares approach the calculations must be performed in a slightly different order. Variables along each path that must be stored and used in the regression are dependent on the type of the underlying security. In the algorithm below they are labelled \( f^i_t = f(x^i_{t_1}, \ldots, x^i_{t_n}) \) where \( f^i_t \) is a vector function.

The two algorithms are almost identical, the difference being that the crude Monte Carlo estimates in Algorithm 2 are used directly to compute option pay-offs whereas in Algorithm 3 they are used to estimate the Model (2.8). Option pay-offs are then computed using prices of the underlying security found from the estimated Model (2.16). For the least squares algorithm it is important to use independent
Algorithm 2 Crude Monte Carlo

1: for $i = 1$ to $M$ do
2:     for $j = 1$ to $n$ do
3:         Simulate $r_{t_j}$
4:         Update relevant state variables at $t_j$, which influence option pay-offs at time $T$
5:         Compute $P^i_{zc}(t_{j-1}, t_j)$ using (Gandhi & Hunt, 1997)
6:         Compute $P^i_{zc}(t_0, t_j) = P^i_{zc}(t_0, t_{j-1})P^i_{zc}(t_{j-1}, t_j)$
7:     end for
8:     Simulate $\hat{P}^{MC,i}_T$ {The underlying simulated price conditioned on path $i$}
9:     $\hat{c}_T = \max(P^i_T - K, 0)$ {The option pay-off conditioned on path $i$}
10:    $\hat{c}_t = P^i_{zc}(t, T)c_T$ {Discounted option pay-off along path $i$}
11:   end for
12: $\hat{c}_t = \frac{1}{M} \sum_{i=1}^M \hat{c}_t$ {Average as price estimator}
13: $\hat{\sigma}_{\hat{c}_t} = \sqrt{\frac{1}{M-1} \sum_{i=1}^M (\hat{c}_t - \hat{c}_t)^2}$ {Standard deviation of sample estimator}

Algorithm 3 Least squares Monte Carlo

1: for $i = 1$ to $M$ do
2:     for $j = 1$ to $n$ do
3:         Simulate $r_{t_j}$
4:         Update relevant state variables at $t_j$, which influence option pay-offs at time $T$
5:         Compute $P^i_{zc}(t_{j-1}, t_j)$ using (Gandhi & Hunt, 1997)
6:         Compute $P^i_{zc}(t_0, t_j) = P^i_{zc}(t_0, t_{j-1})P^i_{zc}(t_{j-1}, t_j)$
7:     end for
8:     Simulate $\hat{P}^{MC,i}_T$ {The underlying price estimate along on path $i$}
9:     Store $f^i_T$ {Store state variables along path $i$}
10:    Store $P^i_{zc}(t, T)$ {Store the discount factor along path $i$}
11:   end for
12: $\hat{b} = (L^T L)^{-1} L^T \hat{P}^MC$ {Regressing simulated prices onto the basis functions}
13: for $i = 1$ to $M$ do
14:     Compute $\hat{P}^{LS,i}_T = L^i \hat{b}$ {The least squares price of the underlying security along path $i$}
15: $\hat{c}_T = \max(\hat{P}^{LS,i}_T - K, 0)$ {The option pay-off conditioned on path $i$}
16: $\hat{c}_t = P^i_{zc}(t, T)c_T$ {Discounted option pay-off conditioned on path $i$}
17: end for
18: $\hat{c}_t = \frac{1}{M} \sum_{i=1}^M \hat{c}_t$ {Average as price estimator}
19: $\hat{\sigma}_{\hat{c}_t} = \sqrt{\frac{1}{M-1} \sum_{i=1}^M (\hat{c}_t - \hat{c}_t)^2}$ {Standard deviation of sample estimator}
paths in the inner simulations. If the same paths are used, the systematic variations in
the simulated prices will not be present. For example, if using a specific inner path conditioned on a given starting point (outer path) results in a much too low price, using the same inner path from a starting point near by the first (another outer path) is also likely to generate a price much too low. Using independent inner paths we ensure that the errors are independent as stated in Assumption 2.2. Note, that the regression is performed after all the crude Monte Carlo prices have been computed; hence they must be stored in the memory.

2.4 Test Cases and the Interest Rate Model

In this section we describe our test case setup; numerical results are postponed until Section 2.5. We present two test cases. In the first test case, we price a European call option on a zero coupon bond. The reason for using this simple example is that the option price can be computed analytically in the interest rate model we use, and we can therefore make very precise conclusions about the performance of our proposed Algorithm 3. In the second test case, we price a European call option on a Danish mortgage backed security (MBS). Pricing a MBS is a multi-dimensional path dependent problem, and simulation is therefore employed. It is precisely this setting that Algorithm 3 is designed to handle. Because no closed form solution exists for the option price, we compare the performance of Algorithm 3 to the crude Monte Carlo method as described in Algorithm 2. In both test cases the Hull-White model is the underlying interest rate model, and for the sake of completeness a very short description of the interest rate model is given below in Section 2.4.1. In Appendix 2.9 the valuation procedure for mortgage backed securities is given.

2.4.1 The Hull-White Interest Rate Model

In the Hull-White model the $\mathbb{Q}$-dynamics of the spot rate is given by

$$dr_t = (\Theta(t) - \kappa r_t)\,dt + \sigma(t)dW_t$$

(2.22)

where $r_t$ is the spot rate, $\kappa$ is the mean-reversion rate, which is assumed to be constant, $\frac{\Theta(t)}{\kappa}$ is the interest rate level that the spot rate will be pushed towards,
\( \sigma(t) \) is the spot rate volatility, and \( W_t \) is a Brownian motion. In our first test case \( \sigma(t) \) is equal to a constant \( \sigma \) and in our second test case the volatility function is a step function with four levels. In practice, specifying the volatility as a step function allows for some flexibility in fitting and calibrating the model to existing interest rate derivatives. Both of these specifications of the volatility function yield an affine term structure model.

\( \Theta(t) \) is found so that the initial term structure of market interest rates are matched and is given by

\[
\Theta(t) = \frac{\partial f^M(0,t)}{\partial t} + \kappa f^M(0,t) + \int_0^T \sigma^2(s) e^{-2\kappa(T-s)} ds
\]

where \( f^M(0,t) \) is the observed market term structure of forward rates. Zero coupon bond prices are given by

\[
P_{zcb}(t,T) = e^{\alpha(t,T)+\beta(t,T)r_t}
\]

where \( \alpha(t,T) \) and \( \beta(t,T) \) can be found by

\[
\beta(t,T) = \frac{1}{\kappa} \left( e^{-\kappa(T-t)} - 1 \right)
\]

\[
\alpha(t,T) = \int_t^T \left( \frac{1}{2} \sigma^2(s) \beta^2(s,T) + \Theta(s) \beta(s,T) \right) ds
\]

Options on zero coupon bonds can be computed analytically in this model, and for \( \sigma(t) = \sigma \), the time \( t \) price of a call option expiring at time \( T_0 \) on a zero coupon bond maturing at time \( T_1 \), \( t < T_0 < T_1 \), and strike \( K \) is given by

\[
c(t,T_0,T_1,K) = P_{zcb}(t,T_1) N(d) - KP_{zcb}(t,T_0) N(d - \nu(t,T_0,T_1))
\]

\[
d = \frac{\ln \left( \frac{P_{zcb}(t,T_1)}{P_{zcb}(t,T_0)} \right) + \frac{1}{2} \nu^2(t,T_0,T_1)}{\nu(t,T_0,T_1)}
\]

\[
\nu(t,T_0,T_1) = \frac{\sigma^2}{2\kappa^3} \left( 1 - e^{-\kappa(T_1-T_0)} \right)^2 \left( 1 - e^{-2\kappa(T_0-t)} \right)
\]

where \( N(\cdot) \) is the cumulative standard normal distribution (see (Jamshidian, 1989)).

Options on complicated bonds, such as e.g. mortgage backed securities, can in general not be evaluated analytically.

### 2.5 Numerical Results

In this section we present numerical results for our two test cases. In Section 2.5.1 below the underlying security is a zero coupon bond, which can be priced analytically.
In the following Section 2.5.2 the underlying security is a mortgage backed bond, which must be priced by simulations.

2.5.1 Zero Coupon Bond as Underlying

We will price a call option on a zero coupon bond in a Hull-White model using Monte Carlo simulation. The price of a zero coupon bond in this model is given in (2.23). Now the call price \( c_t \) can be calculated as

\[
c_t = \mathbb{E}_t^Q \left[ e^{-\int_{T_0}^{T_1} r_s \, ds} \max(P_{zcb}(T_0, T_1) - K, 0) \right]
\]

where \( K \) is the strike value, \( T_0 \) is option expiry and \( T_1 \) is bond maturity with \( T_0 < T_1 \). This value can be computed analytically in the Hull-White model, and the result is given in (2.24).

Define a set of basis functions by

\[
L(x) = \begin{bmatrix} L_1(x) \\ L_2(x) \\ L_3(x) \\ L_4(x) \end{bmatrix} = \begin{bmatrix} e^{\beta(T_0,T_1)x} \\ 1 \\ x \\ x^2 \end{bmatrix}.
\]

For this choice of basis functions Assumption 2.1 is fulfilled because

\[
P_{zcb}(T_0, T_1) = \mathbf{L}^\top(r_{T_0}) \mathbf{b} = e^{\alpha(T_0,T_1)+\beta(T_0,T_1)r_{T_0}}
\]

where \( \mathbf{b}^\top = [e^{\alpha(T_0,T_1)}, 0, 0, 0] \).

We will simulate the call premium by Monte Carlo simulation. Also, we will simulate the zero coupon bond price at time \( T_0 \) given the simulated value of the interest rate. Define the Monte Carlo estimate as

\[
\hat{P}_{zcb}^{MC}(T_0, T_1) = P_{zcb}(T_0, T_1) + \tilde{\epsilon}_{T_0}
\]

For this example we have used 5 paths for each Monte Carlo estimate \( \hat{P}_{zcb}^{MC}(T_0, T_1) \).

We have simulated 10 paths for a 1-year option on a 10-year bond with strike \( K = 0.55 \). The results are shown in Table 2.1.

As in (2.16), the least squares estimate of the zero coupon bond price is given by

\[
\hat{P}_{zcb}^{LS}(T_0, T_1) = \hat{\mathbf{b}} \mathbf{L}(r_{T_0})
\]
Table 2.1
Pathwise simulation

<table>
<thead>
<tr>
<th></th>
<th>( P_{zcb}^{MC}(T_0, T_1) )</th>
<th>( L_1(r_{T_0}^1) )</th>
<th>( L_2(r_{T_0}^2) )</th>
<th>( L_3(r_{T_0}^3) )</th>
<th>( L_4(r_{T_0}^4) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5311</td>
<td>0.6708</td>
<td>1</td>
<td>0.0507</td>
<td>0.0026</td>
<td></td>
</tr>
<tr>
<td>0.6258</td>
<td>0.6689</td>
<td>1</td>
<td>0.0511</td>
<td>0.0026</td>
<td></td>
</tr>
<tr>
<td>0.5916</td>
<td>0.6675</td>
<td>1</td>
<td>0.0514</td>
<td>0.0026</td>
<td></td>
</tr>
<tr>
<td>0.5731</td>
<td>0.6667</td>
<td>1</td>
<td>0.0515</td>
<td>0.0027</td>
<td></td>
</tr>
<tr>
<td>0.5232</td>
<td>0.6513</td>
<td>1</td>
<td>0.0545</td>
<td>0.0030</td>
<td></td>
</tr>
<tr>
<td>0.5780</td>
<td>0.6503</td>
<td>1</td>
<td>0.0547</td>
<td>0.0030</td>
<td></td>
</tr>
<tr>
<td>0.5255</td>
<td>0.6199</td>
<td>1</td>
<td>0.0608</td>
<td>0.0037</td>
<td></td>
</tr>
<tr>
<td>0.5489</td>
<td>0.6071</td>
<td>1</td>
<td>0.0634</td>
<td>0.0040</td>
<td></td>
</tr>
<tr>
<td>0.4776</td>
<td>0.5729</td>
<td>1</td>
<td>0.0708</td>
<td>0.0050</td>
<td></td>
</tr>
<tr>
<td>0.4973</td>
<td>0.5585</td>
<td>1</td>
<td>0.0740</td>
<td>0.0055</td>
<td></td>
</tr>
</tbody>
</table>

Notes: Results of pathwise simulation of basis functions and zero coupon bond prices in the Hull-White model.

where \( \hat{b} \) is the least squares estimate of \( b \).

Using the 10 paths from Table 2.1 we compute the least squares estimates of the coefficients. The result is shown in Table 2.2. Note that since only 10 observations are used for estimating 4 parameters there is a large difference between the true coefficient vector and the estimated coefficient vector. The resulting option prices are shown in Table 2.3 along each of the 10 paths for each of the 3 option price estimators.

Table 2.2
Basis function coefficients

<table>
<thead>
<tr>
<th></th>
<th>( b )</th>
<th>( \hat{b} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( b_1 )</td>
<td>0.8750</td>
<td>1.8889</td>
</tr>
<tr>
<td>( b_2 )</td>
<td>0.0000</td>
<td>-0.9443</td>
</tr>
<tr>
<td>( b_3 )</td>
<td>0.0000</td>
<td>4.9426</td>
</tr>
<tr>
<td>( b_4 )</td>
<td>0.0000</td>
<td>2.0173</td>
</tr>
</tbody>
</table>

Notes: True and estimated basis function coefficients. The coefficients are estimated by ordinary least squares.

In Table 2.3 the price estimators are defined by the expression for the price estimator of the underlying security at option expiry. \( c_{i}^{MC} \) means that crude Monte Carlo has been used, \( c_{i}^{LS} \) means that least squares has been used and \( c_{i}^{CF} \) means that
Table 2.3
Pathwise option price estimators

<table>
<thead>
<tr>
<th>( r^i_{T_0} )</th>
<th>( \hat{c}^{MC,i}_t )</th>
<th>( \hat{c}^{LS,i}_t )</th>
<th>( c^{CF,i}_t )</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.07%</td>
<td>0.0000</td>
<td>0.0273</td>
<td>0.0352</td>
</tr>
<tr>
<td>5.11%</td>
<td>0.0721</td>
<td>0.0257</td>
<td>0.0336</td>
</tr>
<tr>
<td>5.14%</td>
<td>0.0396</td>
<td>0.0245</td>
<td>0.0324</td>
</tr>
<tr>
<td>5.15%</td>
<td>0.0219</td>
<td>0.0238</td>
<td>0.0318</td>
</tr>
<tr>
<td>5.45%</td>
<td>0.0000</td>
<td>0.0107</td>
<td>0.0189</td>
</tr>
<tr>
<td>5.47%</td>
<td>0.0266</td>
<td>0.0098</td>
<td>0.0181</td>
</tr>
<tr>
<td>6.08%</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>6.34%</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>7.08%</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>7.40%</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

Notes: Pathwise option price estimators and the spot rate at option expiry.

the price of the underlying security has been found by the closed form solution. If we let \( Y = \{MC, LS, CF\} \), then for each of the 3 estimators in Table 2.3 Equation (2.21) yields

\[
\hat{c}_t^{Y,i} = P_{zcb}^{Y,i}(t, T_0) \max \left( \hat{P}_{zcb}^{Y,i}(T_0, T_1) - K, 0 \right)
\]

The importance of using least squares estimates can be seen by looking at Figure 2.2 below. In the figure crude Monte Carlo simulated prices, least squares prices and closed form prices are plotted against the spot rate at option expiration. As can be seen, the least squares estimated prices are much closer to the closed form expressions and have much less variance than crude Monte Carlo computed prices. This is as expected since the example is constructed so that both Assumptions 2.1 and 2.2 are fulfilled and Proposition 2.2 can be applied. Therefore by replacing the crude Monte Carlo computed prices with least squares computed prices in the option pay-offs, the bias is reduced.

Next, we want to examine the effect of increasing the number of outer simulations \( M \) while keeping the number of inner simulations, \( N \), constant. As shown in Proposition 2.3 this should effectively reduce the option price bias and result in much more accurate option price estimates. The intuition is that more outer simulations will produce a more precise regression and hereby reduce the bias. The bias for \( N \)
inner simulations and $M$ outer simulations is defined as

$$\eta_t^{LS}(M, N) = \frac{E_t^Q [c_t^{LS}(M, N)] - c_t}{c_t}$$

$\eta_t^{LS}(M, N)$ has been estimated from a sample of 50000 independent observations. In Figure 2.3 we show the estimates $\hat{\eta}_t^{LS}(M, 5)$ for different $M$'s between 20 and 50000. We have also indicated the interval $\pm 2$ standard deviations of $\hat{\eta}_t^{LS}(M, 5)$. As we can see the bias is reduced as the number of outer paths is increased. Another attractive feature of the bias reduction technique is that as the number of outer simulations is increased in order to reduce option price bias, the standard deviation of the option price estimate is reduced. This is also easily seen from Figure 2.3.

In Figure 2.4 we have plotted the average of $\hat{b}_1$ and $\pm 2$ standard deviations of $\hat{b}_1$ computed from a sample of 50000 independent observations. Ideally, $E_t^Q[\hat{b}_1]$ should be equal to $e^{c(T_0, T)}$ since Assumption 2.1 is fulfilled in this setting. Figure 2.4 shows that the average of $\hat{b}_1$ is indistinguishable from $b_1$ but for few outer simulations the uncertainty on $\hat{b}_1$ is high. However, as the number of outer simulations increases, the uncertainty reduces. For $\hat{b}_2$, $\hat{b}_3$, and $\hat{b}_4$ the picture is similar.
Figure 2.3: Convergence of option premium
The number of inner simulations is kept constant equal to 5. The lines above and below the dots are 2 times the standard deviation of the bias estimate.

Figure 2.4: Convergence of the regression coefficient of the exponential function.
The number of inner simulations is kept constant equal to 5. The lines above and below the dots are 2 times the standard deviation of the bias estimate.
2.5.2 MBS as Underlying

Before presenting our results from simulating call option prices on mortgage backed securities, a short description of our choice of basis functions will be presented. The basis functions used in least squares Monte Carlo are constructed from Chebyshev polynomials taking as input variables that determine the price of the underlying security. In the case of a mortgage backed bond we use the time $T$ values of the spot rate($r$), the pool factor($p$) and the tranche weights($\omega_1$ and $\omega_2$). More precisely, we only use up to squared values of the variables and cross products as input. In this example the inputs are given by

$$r, p, \omega_1, \omega_2, r^2, p^2, \omega_1^2, \omega_2^2, rp, r\omega_1, r\omega_2, p\omega_1, p\omega_2, \omega_1\omega_2$$

Each of the input variables is now assigned a basis function in the following way. The input variable $x$ is assigned the basis function $T_1(x) = x$, $x^2$ is assigned $T_2(x) = 2x^2 - 1$, and $xy$ is assigned $T_1(x)T_1(y) = xy$. Since we also include a constant in the regression we end up with a total of 15 basis functions.

On January 9, 2003, we price a European call option with strike 1.02 and expiration on July 1, 2004. The underlying callable mortgage backed bond is an annuity with a coupon rate of 6% and with 4 payments per year. The details for the test case can be found in Appendix 2.10. The price, which we will use as the true price is a limit price found from crude Monte Carlo using no variance reduction methods. The limit price has been computed using Algorithm 2 with 64500 outer and 10000 inner simulations and is equal to

$$P_{\text{limit}}^\text{MBS} (64500, 10000) = 0.016480$$

2.5.3 Convergence

In Figure 2.5 the convergence of the simulated option price in the crude Monte Carlo case is shown. Convergence is achieved for about 3000 outer simulations. However, the uncertainty on the price of the underlying security results in an upward bias on the option price, its size depending on the number of inner simulations used. In the model with only 5 inner simulations, the option price converges to a price
much higher than the limit price, but as the number of inner simulations increase the option price bias is reduced.

Figure 2.6 shows the convergence of simulated option prices when least squares prices have been used to compute option pay-offs. Convergence is achieved for approximately the same number of outer simulations as for crude Monte Carlo but there is no upward option price bias. Even for the model with only 1 single inner simulation, the least squares Monte Carlo method is able to come up with an option price close to the limit price. With more than 3000 outer simulations used, the number of inner simulations do not seem to have any major effect on the option price. In that case all the models in Figure 2.6 produce option prices close to the limit price. The exception is the model with only 1 inner simulation, which seems to undervalue the true option price.

Besides reducing the option price bias, the least squares Monte Carlo approach also gives lower variance on the option price estimate than crude Monte Carlo for a low number of inner simulations. In Figure 2.7 the numerical difference between standard deviations from crude Monte Carlo and least squares Monte Carlo are plotted. For a small number of inner simulations the least squares Monte Carlo computed option prices have lower variability than the crude Monte Carlo computed option prices. As the number of inner simulations increases this difference diminishes.

2.5.4 Error Analysis

Each computed price shown in Figure 2.5 and 2.6 is the result of a single call to a pricing routine. In order to study the error from the crude Monte Carlo and the least squares Monte Carlo methods we have computed a sample of 50 independent prices for each of the models. From these samples we compute the distribution of the relative pricing error. Figures 2.8 and 2.9 show the result for the model with 400 outer simulations and 5 inner simulations - (400,5), and for the (12800,5) model. The figures clearly demonstrate that the conclusions drawn from Figure 2.5 and 2.6 were not due to the paths used for those particular calculations. In general, not surprisingly, the existence of many outer paths reduces the variance of the option estimate for both algorithms. However, the option price computed with crude Monte Carlo will converge to a wrong option price much higher than the limit price. For
Figure 2.5: Convergence of crude Monte Carlo simulated option prices
Convergence of Monte Carlo simulated option prices in different models. Model "MC,ix" means that x inner simulations has been used.

Figure 2.6: Convergence of least squares Monte Carlo option prices
Convergence of least squares Monte Carlo option prices in different models. Model "LSMC,ix" means that x inner simulations have been used.
400 outer simulations, about 12% of the least squares Monte Carlo computed option prices have a relative pricing error of 0 and as many as 48% have a relative pricing error below 5%. Option prices computed with crude Monte Carlo are all biased high and none of the computed prices are within a 5% range of the true price. For 12800 outer simulations, approximately 54% have a relative pricing error of 0% using the least squares method and 98% fall within a 5% range of the true price. In the case of crude Monte Carlo, 12800 outer simulations result in convergence to a wrong price much higher than the true price. None of the crude Monte Carlo simulated prices are within a 5% range of the true price.

From the distributions of the relative pricing errors in different models, we have calculated the probability of getting a relative pricing error below 5%. The results are shown in Table 2.4. From the table a couple of interesting points can be made. First, the result of Proposition 2.3 is very clear. Price estimates computed with least squares Monte Carlo become closer to the limit price as the number of outer simulations increases - the error distribution becomes more centered around 0. Also, the least squares Monte Carlo prices do not depend so much on the number of inner simulations in the sense that the number of outer simulations can always be increased.
Figure 2.8: Histogram for the relative pricing error.

Histogram for the relative pricing error in the Crude Monte Carlo and least squares Monte Carlo case using 400 outer and 5 inner paths. No other variance reduction methods are used. The probabilities are found using a batch of size 50.

Figure 2.9: Histogram for the relative pricing error.

Histogram for the relative pricing error in the Crude Monte Carlo and least squares Monte Carlo case using 12800 outer and 5 inner paths. No other variance reduction methods are used. The probabilities are found using a batch size of 50.
in order for the price estimates to become more valid, i.e. into close range of the limit price. In fact, for the calculations in Table 2.4, when more than 6400 outer simulations are used no more that 25 inner simulations are needed in order to get very precise option prices. Moreover, when using crude Monte Carlo it is crucial to reduce the option price bias in order to get precise option prices. For very few inner simulations crude Monte Carlo simulations cannot be used to produce a price within an error margin of 5% of the limit price. In Appendix 2.9.4 the empirical distribution function for the error term is shown for different models.

2.5.5 Efficiency Issues

As described in Section 2.3.2 least squares Monte Carlo is a crude Monte Carlo plus a regression. The cost of performing the regression must therefore be lower than the cost of the inner simulations saved, if least squares Monte Carlo is to be more efficient than crude Monte Carlo. In Figure 2.10 we have graphed combinations of outer and inner simulations that with 95% probability result in an absolute relative pricing error below 5%. For a given number of inner simulations we have increased the number of outer simulations until the probability of getting the wanted accuracy has been achieved.

In the least squares Monte Carlo method the number of inner simulations are 1, 5, 25, 50, 100, 200, 400 and 800. As suggested by Proposition 2.3 inner simulations can be substituted by outer simulations. Few inner simulations require many outer simulations to compensate for the high uncertainty in the price of the underlying security. However, from an efficiency viewpoint, since many outer simulations also incur higher computation time in the regression it might not be optimal just to use few inner simulations. Looking at the least squares Monte Carlo computation time in Figure 2.11, we can see that it is optimal to have the number of inner simulations below 25, which is also the minimum time used for achieving the desired accuracy. With 25 inner simulations 4800 outer simulations are required to get the desired accuracy with 95% probability. For this combination it takes 115 seconds\(^5\) to compute the option price.

In the crude Monte Carlo method only inner simulations from 50 to 800 have

---

\(^5\)The calculations have been performed on a Pentium 4, 2.4 GHz, 512 MB RAM PC.
Table 2.4
Probability of $|\epsilon| \leq 5$

<table>
<thead>
<tr>
<th>Model</th>
<th>#Outer</th>
<th>#Inner</th>
<th>LSMC</th>
<th>MC</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>1</td>
<td>6%</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td>800</td>
<td>1</td>
<td>14%</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td>1600</td>
<td>1</td>
<td>8%</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td>3200</td>
<td>1</td>
<td>40%</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td>6400</td>
<td>1</td>
<td>78%</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td>12800</td>
<td>1</td>
<td>96%</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>5</td>
<td>48%</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td>800</td>
<td>5</td>
<td>76%</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td>1600</td>
<td>5</td>
<td>70%</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td>3200</td>
<td>5</td>
<td>92%</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td>6400</td>
<td>5</td>
<td>94%</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td>12800</td>
<td>5</td>
<td>98%</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>25</td>
<td>52%</td>
<td>52%</td>
<td></td>
</tr>
<tr>
<td>800</td>
<td>25</td>
<td>60%</td>
<td>54%</td>
<td></td>
</tr>
<tr>
<td>1600</td>
<td>25</td>
<td>88%</td>
<td>56%</td>
<td></td>
</tr>
<tr>
<td>3200</td>
<td>25</td>
<td>90%</td>
<td>56%</td>
<td></td>
</tr>
<tr>
<td>6400</td>
<td>25</td>
<td>100%</td>
<td>42%</td>
<td></td>
</tr>
<tr>
<td>12800</td>
<td>25</td>
<td>100%</td>
<td>50%</td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>50</td>
<td>56%</td>
<td>62%</td>
<td></td>
</tr>
<tr>
<td>800</td>
<td>50</td>
<td>82%</td>
<td>68%</td>
<td></td>
</tr>
<tr>
<td>1600</td>
<td>50</td>
<td>90%</td>
<td>84%</td>
<td></td>
</tr>
<tr>
<td>3200</td>
<td>50</td>
<td>94%</td>
<td>90%</td>
<td></td>
</tr>
<tr>
<td>6400</td>
<td>50</td>
<td>98%</td>
<td>90%</td>
<td></td>
</tr>
<tr>
<td>12800</td>
<td>50</td>
<td>100%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>100</td>
<td>52%</td>
<td>68%</td>
<td></td>
</tr>
<tr>
<td>800</td>
<td>100</td>
<td>66%</td>
<td>78%</td>
<td></td>
</tr>
<tr>
<td>1600</td>
<td>100</td>
<td>82%</td>
<td>92%</td>
<td></td>
</tr>
<tr>
<td>3200</td>
<td>100</td>
<td>88%</td>
<td>98%</td>
<td></td>
</tr>
<tr>
<td>6400</td>
<td>100</td>
<td>98%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>12800</td>
<td>100</td>
<td>100%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>800</td>
<td>56%</td>
<td>62%</td>
<td></td>
</tr>
<tr>
<td>800</td>
<td>800</td>
<td>78%</td>
<td>84%</td>
<td></td>
</tr>
<tr>
<td>1600</td>
<td>800</td>
<td>84%</td>
<td>96%</td>
<td></td>
</tr>
<tr>
<td>3200</td>
<td>800</td>
<td>90%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>6400</td>
<td>800</td>
<td>100%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>12800</td>
<td>800</td>
<td>100%</td>
<td>100%</td>
<td></td>
</tr>
</tbody>
</table>

Notes: The distributions are found on a 50 sample basis. #Outer and #Inner are the number of outer and inner paths respectively. No variance reduction has been used.
been computed. For fewer inner simulations it is not possible to compute an option price within 5% of the limit price. For 50 inner simulations 9600 outer simulations are needed but this number reduces to 2400 when 100 inner simulations are used. This is also the most efficient combination of outer and inner simulations in the crude Monte Carlo case. This combination of inner and outer simulations takes 133 seconds for computing an option price.

![Figure 2.10: Outer/Inner trade-off.](image)

Combinations of outer and inner simulations that with a probability of 95% results in an relative pricing error below 5%. For a fixed number of inner simulations, the number of outer simulations is found using interpolation between the number of simulations resulting in a probability just below and just above 95%. The sample size is equal to 50.

A measure that takes into account both the option price bias and the variance of an estimator is the RMSE, which is defined as $\sqrt{\mathbb{E}[(\hat{c} - c)^2]}$. RMSE can therefore be used to summarize, in a single number, the points drawn from Figure 2.7 and Table 2.4. Using relative pricing errors as in (Broadie & Detemple, 1996), RMSE can be estimated by the following formula, where $J$ is the sample size.

$$\sqrt{\frac{1}{J} \sum_{j=1}^{J} \left( \frac{\hat{c}_j - c_j}{c_j} \right)^2}$$

The traditional approach to compute RMSE is based on prices of contracts with different specifications, i.e. different strikes, maturities etc. We, however, compute it from the prices in the batches computed for estimating the distribution of the
relative pricing error. The results are shown in Figure 2.12. The preferred region is the upper-left corner, which combines high precision with high speed. As can be seen the most efficient method to use in our test case is least squares Monte Carlo with 5 inner simulations.

2.5.6 Result - Antithetic Sampling

To see the impact on efficiency using antithetic sampling between today and option expiry only, we have computed the same numbers presented in Table 2.4. The numbers are shown in Table 2.5 below together with the number from Table 2.4. Using antithetic sampling has a major positive effect on the least squares Monte Carlo method. With only 5 inner simulations the number of outer simulations does not have to be higher than 1600 in order to be almost certain to have an error within a 5% range of the true price. Note that using antithetic sampling between today and option expiry will not reduce the option price bias for crude Monte Carlo estimates. It that case only the variance in the option price will be reduced but the option price bias will not.

---

6Using antithetic sampling between today and option expiration is a violation of the conditions in Equation (2.11) and Equation (2.12), but since the results are so promising we include them. The reason for the violation is that antithetic paths are not independent.
<table>
<thead>
<tr>
<th>Model</th>
<th>#Outer</th>
<th>#Inner</th>
<th>LSMC</th>
<th>Antithetic</th>
<th>LSMC</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>1</td>
<td></td>
<td>16%</td>
<td>6%</td>
<td></td>
</tr>
<tr>
<td>800</td>
<td>1</td>
<td></td>
<td>6%</td>
<td>14%</td>
<td></td>
</tr>
<tr>
<td>1600</td>
<td>1</td>
<td></td>
<td>50%</td>
<td>8%</td>
<td></td>
</tr>
<tr>
<td>3200</td>
<td>1</td>
<td></td>
<td>76%</td>
<td>40%</td>
<td></td>
</tr>
<tr>
<td>6400</td>
<td>1</td>
<td></td>
<td>94%</td>
<td>78%</td>
<td></td>
</tr>
<tr>
<td>12800</td>
<td>1</td>
<td></td>
<td>100%</td>
<td>96%</td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>5</td>
<td></td>
<td>88%</td>
<td>48%</td>
<td></td>
</tr>
<tr>
<td>800</td>
<td>5</td>
<td></td>
<td>80%</td>
<td>76%</td>
<td></td>
</tr>
<tr>
<td>1600</td>
<td>5</td>
<td></td>
<td>98%</td>
<td>70%</td>
<td></td>
</tr>
<tr>
<td>3200</td>
<td>5</td>
<td></td>
<td>100%</td>
<td>92%</td>
<td></td>
</tr>
<tr>
<td>6400</td>
<td>5</td>
<td></td>
<td>100%</td>
<td>94%</td>
<td></td>
</tr>
<tr>
<td>12800</td>
<td>5</td>
<td></td>
<td>100%</td>
<td>98%</td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>25</td>
<td></td>
<td>80%</td>
<td>52%</td>
<td></td>
</tr>
<tr>
<td>800</td>
<td>25</td>
<td></td>
<td>96%</td>
<td>60%</td>
<td></td>
</tr>
<tr>
<td>1600</td>
<td>25</td>
<td></td>
<td>98%</td>
<td>88%</td>
<td></td>
</tr>
<tr>
<td>3200</td>
<td>25</td>
<td></td>
<td>100%</td>
<td>90%</td>
<td></td>
</tr>
<tr>
<td>6400</td>
<td>25</td>
<td></td>
<td>100%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>12800</td>
<td>25</td>
<td></td>
<td>100%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>50</td>
<td></td>
<td>82%</td>
<td>56%</td>
<td></td>
</tr>
<tr>
<td>800</td>
<td>50</td>
<td></td>
<td>98%</td>
<td>82%</td>
<td></td>
</tr>
<tr>
<td>1600</td>
<td>50</td>
<td></td>
<td>98%</td>
<td>90%</td>
<td></td>
</tr>
<tr>
<td>3200</td>
<td>50</td>
<td></td>
<td>100%</td>
<td>94%</td>
<td></td>
</tr>
<tr>
<td>6400</td>
<td>50</td>
<td></td>
<td>100%</td>
<td>98%</td>
<td></td>
</tr>
<tr>
<td>12800</td>
<td>50</td>
<td></td>
<td>100%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>400</td>
<td></td>
<td>94%</td>
<td>56%</td>
<td></td>
</tr>
<tr>
<td>800</td>
<td>400</td>
<td></td>
<td>97%</td>
<td>56%</td>
<td></td>
</tr>
<tr>
<td>1600</td>
<td>400</td>
<td></td>
<td>98%</td>
<td>80%</td>
<td></td>
</tr>
<tr>
<td>3200</td>
<td>400</td>
<td></td>
<td>100%</td>
<td>84%</td>
<td></td>
</tr>
<tr>
<td>6400</td>
<td>400</td>
<td></td>
<td>100%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>12800</td>
<td>400</td>
<td></td>
<td>100%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>800</td>
<td></td>
<td>92%</td>
<td>56%</td>
<td></td>
</tr>
<tr>
<td>800</td>
<td>800</td>
<td></td>
<td>100%</td>
<td>78%</td>
<td></td>
</tr>
<tr>
<td>1600</td>
<td>800</td>
<td></td>
<td>98%</td>
<td>84%</td>
<td></td>
</tr>
<tr>
<td>3200</td>
<td>800</td>
<td></td>
<td>100%</td>
<td>90%</td>
<td></td>
</tr>
<tr>
<td>6400</td>
<td>800</td>
<td></td>
<td>100%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>12800</td>
<td>800</td>
<td></td>
<td>100%</td>
<td>100%</td>
<td></td>
</tr>
</tbody>
</table>

Notes: The distributions are found on a 50 sample basis. #Outer and #Inner are the number of outer and inner paths respectively. Antithetic sampling has been used between today and option expiry.
Figure 2.12: Speed-accuracy trade-off for different models.

RMS relative error is defined by \( \sqrt{\frac{1}{J} \sum_{j=1}^{J} (\hat{c}_j - c_j)^2} \), where \( c_j \) is the true option value computed by crude Monte Carlo with 64500 outer and 10000 inner paths and \( \hat{c}_j \) is either the crude Monte Carlo or the least squares Monte Carlo estimate. Speed is measured as \( 1/\text{(computation time in sec.)} \). The preferred region is the upper-left corner. The sample size, \( J \), is equal to 50.

Of course, using antithetic sampling cannot be done without incurring additional costs in the form of higher computation time. The efficiency of using least squares Monte Carlo with antithetic sampling until option expiry is compared in Figure 2.13 with the least squares Monte Carlo without any variance reduction method. The figure shows that using antithetic sampling considerably reduces the number of outer simulations necessary for getting an error below 5% with a probability of 95%. For more than 5 inner simulations the number of outer simulations are reduced by a factor of two or more, indicating that antithetic sampling will indeed speed up the method. This is also evident when looking at the computation times for the two methods. The computation time for least squares Monte Carlo with antithetic sampling is always below the computation time for the method without the variance reduction technique. Least squares Monte Carlo with antithetic sampling is most efficient for 25 inner simulations, in which case 775 outer simulations are necessary to obtain an error below 5% with 95% certainty. This only takes 35 seconds to compute, whereas the most efficient set-up takes 115 seconds when no variance reduction techniques are used, yielding a speed-up factor of more than 3.
2.5.7 Basis Function Sensitivity

In order to investigate the proposed algorithm’s dependence on the choice of basis functions we have compared relative pricing errors for different choices of basis functions in Table 2.6. The basis functions vary in their correlation with the price of the underlying security at option expiration. For example, in Model 1, the only basis function is a constant, which is zero correlated with the underlying security price.

As can be seen from the table it is crucial that the basis functions are highly correlated with the price they are trying to approximate. A way to achieve this is to use the simulated state variables as basis functions (in this example the spot rate is the only state variable). Since we must assume that the simulated state variables have a large influence on the price of the underlying security (otherwise it would be unnecessary to simulate them) it would be advisable to use them as basis functions. The resulting basis functions will per construction be highly correlated with the price of the underlying security. Another point that can be drawn from the table is that information collected during time zero and option expiration does not have a major impact.

**Figure 2.13:** Outer/Inner trade-off.

Combinations of outer and inner simulations that with a probability of 95% result in a relative pricing error below 5%. For a fixed number of inner simulations, the number of outer simulations is found using interpolation between the number of simulations resulting in a probability just below and just above 95%. The sample size is equal to 50. LSMC anti means that antithetic sampling has been used between today and option expiry.
Table 2.6  
Basis Function Sensitivity

<table>
<thead>
<tr>
<th>Model</th>
<th>Basis Fct.</th>
<th>#Basis Fct.</th>
<th>Error (400/50)</th>
<th>Error (6400/50)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-74.6%</td>
<td>-100.0%</td>
</tr>
<tr>
<td>2</td>
<td>1, r, r^2</td>
<td>3</td>
<td>3.2%</td>
<td>-0.4%</td>
</tr>
<tr>
<td>3</td>
<td>1, r^2, p, p^2, r p</td>
<td>6</td>
<td>-0.2%</td>
<td>-1.2%</td>
</tr>
<tr>
<td>4</td>
<td>1, r, r^2, \omega_1, \omega_2</td>
<td>10</td>
<td>-1.4%</td>
<td>-1.8%</td>
</tr>
<tr>
<td>5</td>
<td>1, p, p^2</td>
<td>3</td>
<td>-21.9%</td>
<td>-23.8%</td>
</tr>
<tr>
<td>6</td>
<td>1, p, p^2, \omega_1, \omega_2</td>
<td>10</td>
<td>-6.4%</td>
<td>-8.0%</td>
</tr>
<tr>
<td>7</td>
<td>1, \omega_1, \omega_2, \omega_1 \omega_2</td>
<td>6</td>
<td>-16.2%</td>
<td>-17.3%</td>
</tr>
</tbody>
</table>

Notes: Errors relative to the price obtained by using 14 basis functions for the two models (400/50) and (6400/50).

impact on the pricing. In other words, at least for this example, the simulated state variable alone is useful in explaining the price of the underlying security at option expiration, meaning that the path dependent nature of the mortgage bond for a large part is captured by observing the spot rate at the beginning and at option expiration.

2.6 Improvements

In this paper we have reduced the option price bias by regressing the Monte Carlo computed prices of the underlying security onto a set of basis functions. The most important question here is how to create a precise regression. In this section we will discuss a few ways to accomplish this and some complications that can arise when trying.

2.6.1 Inner Versus Outer Paths.

The first and probably most obvious way to get a more precise regression is to increase the number of outer paths which will result in more observations to use in the regression. Another just as obvious choice is to increase the number of inner paths which will create more precise observations to use in the regression.

It is not obvious which combination of \((M, N)\) we should use. Is there an optimal
combination of outer and inner simulations? Of course such a combination will depend on the problem at hand and the implementation. We will illustrate this issue by continuing the example from Section 2.5.1.

\((M_1, N_1)\) and \((M_2, N_2)\) will be called time equivalent if \(\hat{c}^{LS}(M_1, N_1)\) and \(\hat{c}^{LS}(M_2, N_2)\) have the same calculation time. In Figure 2.14 we have shown a set of time equivalent pairs. Now, we are interested in finding the best combination among these pairs.

![Figure 2.14: Iso time line. Combinations of outer and inner simulations, \((M,N)\), yielding identical computation time.](image)

There are different ways of comparing these combinations. In Figure 2.15 we have shown the average and indicated \(\pm 2\) standard deviations for the estimated option prices for each combination.\(^7\) As we can see the standard deviation is much larger for a high number of inner simulations, indicating that it is inefficient to use too many inner simulations. It is not clear if there is any difference in the bias produced by each combination. In order to take both bias and variation into account we have estimated the probability that the error is less than both 1% and 5%. These numbers are shown in Figure 2.16. Here, we can see that a combination of 5 inner and 5125 outer simulations seems to be optimal both for 1% and 5%.

In our MBS example from Appendix 2.9 the calculation of an inner path was

\(^7\)We have used a sample of 50000 independent simulations to estimate averages and standard deviations.
Figure 2.15: Bias estimation for time equivalent pairs of inner and outer simulations.

Figure 2.16: $P(\text{Error} \leq a)$ for time equivalent pairs of inner and outer simulations.
far more complex than an outer path. This suggests that we should use fewer inner
paths for that type of problem. In that example the variance of the MBS estimates
is highly state dependent. Hence, we can improve the regression simply by using
more inner paths in the high variance regions. For the callable mortgage bond this
is around the time when people start to call the mortgage bond.

2.6.2 An Infinite Dimensional Set of Basis Functions.

In the previous sections we have assumed that the price of the underlying asset is
spanned by a finite set of basis functions. This is usually not the case. In Appendix
2.9 we valued an option on a callable mortgage bond which was not spanned by the
set of basis functions used in the regression.

To illustrate this type of problem we will again turn to our example from Section
2.5.1. In this example we know the price at the exercise time and can therefore
create a set of basis functions that will span the solution. How will this method
work if we do not know the true value? In Figure 2.17 we show for $N = 5$ and
$M = 5000$ the bias and ±2 standard deviations of the bias estimate. The series
labelled ”Polynomial” does not include the exponential function as a basis function
whereas the series labelled ”Incl. exp” does. Recall that if the exponential function
is included in the set of basis functions the true price will be spanned by the basis
functions. As can be seen, in the ”Polynomial” case, nothing is gained by having
more than 3 basis functions, so just increasing the number of basis functions to
beyond 3 will not produce better bias reduction. Note, that the bias created when the
exponential function is not included in the basis functions is greater than when it is
included. At first it might seem peculiar that we do not gain any more accuracy from
increasing the number of basis functions. The reason is that more basis functions
give more volatile coefficients for the same number of observations. Hence, in order
to get the same precision a higher number of observations is required. On the other
hand, too few basis functions will produce a functional form which is far from the
true functional form. (Glasserman & Yu, 2003) show that the number of outer
simulations needed grows exponentially with the number of basis functions. Hence,
more is not always better. For this reason it is very important to choose the best set
of basis functions for which a low-dimensional set of basis functions is acceptable.
2.6.3 Importance Sampling.

For a call option we are only interested in an accurate estimate above the strike. For this reason we want to concentrate our regression on this region. This is important both to get a more precise regression but also few basis functions will be better at approximating a minor region. The simplest way of concentrating our regression on the critical region is to exclude some observations from the regression. In Figure 2.18 we have excluded all observations of the zero coupon bond price from Section 2.5.1 where the true price is less than 0.5. As we can see the regression is shifted to the right.

This reduces the bias significantly as can be seen from comparing Figure 2.19 with Figure 2.17 where the only difference is the sample used in the regression.

This is a very extreme way of focusing the regression. Another way is to use importance sampling. This is a well-documented variance reduction technique which is used to create more samples of the state variables in the most critical regions. This technique can be used to increase the number of samples in the most volatile areas instead of increasing the number of inner paths in the same areas.
Figure 2.18: In order to get a more precise regression above the strike we have excluded bond prices less than 0.5 from the regression.

Figure 2.19: Bias in the option price when bond prices less than 0.5 have been excluded from the regression.
2.7 Conclusion

In this paper we have developed a method to reduce the option price bias in the European option price, resulting from using simulated prices of the underlying security when computing option pay-offs. We have shown that in an ideal world where the true price is spanned by a set of basis functions the option price bias can be completely removed by increasing the number of outer simulations independent of the number of inner simulations. Under the same assumptions, we have also shown that the least squares estimate of the price of the underlying security has lower variance than the crude Monte Carlo estimate leading to lower option price bias.

Even when the true price is approximated by a finite linear combination of the basis functions, i.e. the true price does not belong to the space spanned by the basis functions, we have seen in a numerical example that the least squares computed option price converges to the limit price as the number of outer simulations increases. For a single inner simulation we were able to compute an option price very close to the limit price using many outer simulations. The method can also be viewed as a way to substitute inner simulations with outer simulations. This is particularly efficient when the cost of simulating inner paths is high compared to the cost of simulating outer paths as in the case of pricing short-term options on long-lived securities. In our computational test case the least squares Monte Carlo method proved to be faster than crude Monte Carlo when no other variance reduction methods were used.

We also demonstrated that the bias reduction method can be combined with other variance reduction techniques. In Section 2.6 it was shown that the regression could be considerably improved by focusing on the relevant region as is done in important sampling. In Section 2.5.6 antithetic sampling between today and option expiry was employed to improve the regression. Compared to the case with no variance reduction techniques we obtained a speed-up factor of three by using the least squares Monte Carlo method.
2.8 Appendix A: Proofs

2.8.1 Proof of Proposition 2.1

We first show that option prices are upwardly biased when option pay-offs are computed using the prices of the underlying security that contain a noisy element.

\[ c_t = \mathbb{E}_t^Q \left[ e^{-\int_t^T r_s \, ds} f(P_T, K) \right] \]

\[ = \mathbb{E}_t^Q \left[ e^{-\int_t^T r_s \, ds} f \left( \mathbb{E}_T^Q \left[ \hat{P}_T \right], K \right) \right] \]

\[ \leq \mathbb{E}_t^Q \left[ e^{-\int_t^T r_s \, ds} \mathbb{E}_T^Q \left[ f \left( \hat{P}_T, K \right) \right] \right] \]

\[ = \mathbb{E}_t^Q \left[ e^{-\int_t^T r_s \, ds} f \left( \hat{P}_T, K \right) \right] \]

\[ = \hat{c}_t \]

The first equality comes from the general pricing formula for pricing under the risk neutral probability measure. In the second equality \( \hat{P}_T \) is an unbiased estimator of the true price of the underlying security, \( P_T \). The third line follows from Jensen’s inequality and the convexity of the pay-off function \( f(P, K) \). In the fourth line we use the law of iterated expectations and the fifth equality is given by definition.

Next we show that the option price bias can be controlled by determining an upper bound on the deviation from the true price of the underlying security.

\[ \hat{c}_t = \mathbb{E}_t^Q \left[ e^{-\int_t^T r_s \, ds} f \left( P_T + \hat{\epsilon}, K \right) \right] \]

\[ \leq \mathbb{E}_t^Q \left[ e^{-\int_t^T r_s \, ds} \left( f \left( P_T, K \right) + |\hat{\epsilon}| \right) \right] \]

\[ = \mathbb{E}_t^Q \left[ e^{-\int_t^T r_s \, ds} f \left( P_T, K \right) \right] + \mathbb{E}_t^Q \left[ e^{-\int_t^T r_s \, ds} |\hat{\epsilon}| \right] \]

\[ = c_t + \mathbb{E}_t^Q \left[ e^{-\int_t^T r_s \, ds} \mathbb{E}_T^Q \left[ \sqrt{\epsilon^2} \right] \right] \]

\[ \leq c_t + \sigma \mathbb{E}_t^Q \left[ e^{-\int_t^T r_s \, ds} \sqrt{\text{Var}_T^Q [\hat{\epsilon}]} \right] \]

\[ = c_t + \sigma \mathbb{E}_t^Q \left[ e^{-\int_t^T r_s \, ds} \right] \]

The first equality follows from Equation (2.4). The second line follows from moving \( \epsilon \) out of the pay-off function, and using the anatomy of the pay-off function. In the third line we use the linearity of the expectation operator. The fourth equality follows from the definition of the true option price and the law of iterated expectations. Then we use Equation (2.5) and Jensen’s inequality to interchange the \( \mathbb{E} \) operator and the \( \sqrt{\cdot} \) function. Finally we use Equation (2.6).
2.8.2 Proof of Proposition 2.2

The first statement of the proposition follows by

$$
E_Q^T \left[ \hat{P}_{LS}^T \right] = E_Q^T \left[ T \hat{b} \right] = L E_Q^T \left[ b \right] = Lb = P_T
$$

where the first equality follows from (2.16), the second equality follows from $L$ being $T$-measurable, the third equality follows from the unbiasedness of the estimator $\hat{b}$ as stated in (2.15) and the last equality follows from Assumption 2.1. The second statement is proved by

$$
\text{Var}_Q^T \left[ \hat{P}_{LS,i}^T \right] = \text{Var}_Q^T \left[ L_i \hat{b} \right]
= L_i \text{Var}_Q^T \left[ b \right] (L_i^i)^T
= \sigma_e^2 L_i (L_i^T L_i)^{-1} (L_i^i)^T
$$

The first equality is given by (2.16), the second follows from moving the vector $L_i$ out of the variance operator, and in the third equality we use Equation (2.15).

If $L_i (L_i^T L_i)^{-1} (L_i^i)^T \leq 1$ the result is proved. To confirm that this is the case we use a corollary of the extended Cauchy-Schwartz inequality (see (Johnson & Wichern, 1998) page 82), which states

$$
\max_{x \neq 0} \frac{(x^T d)^2}{x^T B x} = d^T B^{-1} d
$$

(2.25)

where $B$ is a positive definite matrix and $d$ and $x$ are vectors.

We will use this result with $d = (L_i^i)^T$ and $B = L_i^T L_i$. Note, that both the nominator and denominator are scalars and can be evaluated separately. The denominator is equal to

$$
x^T (L_i^T L_i)x = (L_i x)^T (L_i x)
= \sum_{k=1,k \neq i}^M (L_k x)^2 + (L_i x)^2
$$

(2.26)

The nominator is equal to

$$
(x^T (L_i^i))^2 = (L_i x)^2
$$

(2.27)
hence, by inserting (2.26) and (2.27) into (2.25) we can conclude that

\[ L^i (L^iu)^{-1}(L^iT) \leq 1 \quad \forall \, i \]

from which the result follows. ■

### 2.8.3 Proof of Proposition 2.3

From Proposition 2.1 we know that \( \hat{c}_t^{LS} \leq c_t + \mathbb{E}_t^Q \left[ e^{-\int_t^T r_s \, ds} \right] \sqrt{\text{Var}_t^Q[\hat{\epsilon}^{LS}]} \). If we can show that the last term approaches 0 as the number of outer simulations \( M \) increases we have shown (2.20). In the proof of Proposition 2.2 we saw that \( \text{Var}_t^Q[\hat{P}_T^{LS,i}] = \sigma^2 L^i (L^iT)^{-1} (L^iT) \). Now,

\[
\text{Var}_t^Q[\hat{\epsilon}^{LS,i}] = \mathbb{E}_t^Q \left[ \text{Var}_t^Q[\hat{\epsilon}^{LS,i}] \right] + \mathbb{E}_t^Q \left[ \text{Var}_t^Q[\hat{P}_T^{LS,i}] \right] \\
= \mathbb{E}_t^Q \left[ \text{Var}_t^Q[\hat{P}_T^{LS,i}] \right] \\
= \mathbb{E}_t^Q \left[ \sigma^2 L^i (L^iT)^{-1} (L^iT) \right] \\
= \frac{\sigma^2}{M} \mathbb{E}_t^Q \left[ L^i Q^{-1}_M (L^iT) \right] \\
\rightarrow 0 \text{ as } M \rightarrow \infty \quad \forall \, i
\]

The first equality can be found in e.g. (Greene, 2002), page 866, the second and third lines follow from Assumption 2.2 and \( \hat{P}_T^{LS,i} = P_T + \hat{\epsilon}_T^{LS,i} \). The result follows now from Assumption (2.19).

■
2.9 Appendix B: MBS Valuation

In this section we will briefly describe pricing of mortgage backed bonds by simulations. We also give an example showing the intuition in using least squares Monte Carlo estimates for computing option pay-offs when few inner simulations are employed.

A callable mortgage bond gives the borrower the right to prepay the remaining loan balance at any point in time. However, debt holders do not follow the optimal exercise strategy which excludes the use of traditional option theory in the valuation of the borrower side. Instead, a statistical model is used, which over time explains the majority of the variations in prepayments across different mortgage pools. The prepayment function must take into account heterogeneity among borrowers as a result of a long issue period\(^8\) and behavioural differences among debt holders for utilizing the prepayment option. Many models use the after tax and costs\(^9\) gain from refinancing as one of the explanatory variables. The value of this variable captures prepayments arising from future market situations. However, as the mortgage pool characteristics change because the most inclined prepayers leave the mortgage pool, a given level of refinancing gains tends to result in lower prepayments than when the level was hit the first time. To capture this and dampen prepayments in series that have already experienced prepayments, (Jakobsen, 1994) suggests dividing the mortgage pool into tranches, each with its own prepayment function. This will capture some divergences in prepayments among debtor groups over time, because when the pool changes from large to small loans, prepayments will fall. However, we find that a combination of this method and a more direct modelling of the burnout gives a better explanation of variations in prepayments. The model we use thus utilizes the series pool factor as a way to capture additional burnout among debt holders. See (Schwartz & Torous, 1989) and (Jakobsen, 1992) for an in-depth description of prepayments and valuation of MBS\(^{10}\).

---

\(^8\)In Denmark callable mortgage bonds are open for issue for up to three years, which means that the same underlying pool of mortgage bonds can be issued to borrowers under very different market conditions.

\(^9\)The costs of refinancing are: DKK 2500 in fixed costs, 2.5% in differential rate, 1% in creation fee and 0.15% in brokerage fee.

\(^{10}\)(Jakobsen, 1992) is the main reference for valuation of Danish callable mortgage bonds.
2.9.1 The Prepayment Model

Our model for pricing callable Danish mortgage backed bonds results in a path dependent pricing problem in at least four dimensions. One state variable comes from the interest rate model and the other three come from the way that prepayments are modelled. We divide the pool of mortgage loans into tranches, each with its own prepayment function. Each prepayment function takes the series pool factor as input. For a given path the price of the underlying mortgage bond at option expiration will therefore depend on the value of the pool factor and the relative weights between the tranches along that path. Because the series pool factor is used as input to the prepayment function, prepayments in a given tranche will depend on prepayments in the other tranches, which means that the tranches cannot be priced separately.

In fact, it is the conditional prepayment rate (CPR) that is estimated by the prepayment function. The CPR is the share of the remaining debtors that choose to prepay their loans. Once the CPR has been found, the total prepayment rate can be calculated as

\[ u_n = A_n + (1 - A_n)\lambda_n, \]

where \( \lambda_n \) is the CPR, \( A_n \) is the scheduled prepayment rate and \( n \) is the payment date. Our prepayment model consists of three tranches, tranche one is loans with a debt balance below DKK 500,000, tranche two consists of loans with a remaining debt balance between DKK 500,000 and 3,000,000 and tranche three includes loans above DKK 3,000,000. Each tranche has its own prepayment function of the following form

\[ \lambda = N(\beta_1 \cdot GAIN \cdot f(POOLFACTOR) + \beta_2 + \beta_3 \cdot TTM) \]  
\[ f(POOLFACTOR) = N\left(\frac{\ln(POOLFACTOR) - \beta_3}{\beta_4}\right) \]

where \( GAIN \) is the refinancing gain, \( TTM \) is the time to maturity and \( N \) is the cumulative standard normal distribution.

As mentioned above, the variable \( POOLFACTOR \) that enters Equation (2.28) is the series pool factor. Due to the noise in the statistical data on Danish mortgage bonds, it is not possible to get accurate estimates of tranche pool factors. This has implications for the choice of method used to calculate prices. At any point in

\footnote{Due to the statistical material released from the mortgage credit institutions we are able to break down the total pool of mortgage loans into different tranches, each with its own prepayment function.}
the future, the series pool factor must be calculated as a weighted average of the tranche pool factors (see Equation (2.31) below). However, this requires knowledge of all the tranche weights simultaneously. Additional state variables come from the interest rate model. We therefore use Monte Carlo simulations when pricing Danish mortgage backed bonds. If a PDE solution were to be used, we would have to augment the state space with the number of tranches, and with a total of more than four state variables, the PDE approach would be unmanageable. In (Paskov, 1996), a similar pricing problem is presented although he prices different tranches of a CMO on a pool of mortgage loans instead of the mortgage pool itself.

2.9.2 Updating Rules

Along a given path, variables influencing the option pay-off must be updated. In the case of options on MBS, variables that determine the bond price must be updated between today and option expiration. The variables are the pool factor and the relative tranche weights, which summarizes changes in debtor distribution.

The pool factor for tranche \( k \) at date \( n \) measures how much debt is left in the tranche at time \( n \). Knowing the tranche size relative to the series size at time 0 and prepayments in the tranche between 0 and \( n \), the pool factor of the tranche at time \( n \) can be calculated as

\[
p_n^k = p_0 \omega_0^k \prod_{j=1}^{n} (1 - \lambda_j^k)
\]

(2.30)

where \( p_0 \) is the series pool factor at time 0, \( \omega_0^k \) is the tranche weight for tranche \( k \) at time 0 and \( \lambda_j^k \) is the cpr rate for tranche \( k \) at time \( j \). The series pool factor at date \( n \) can be found by

\[
p_n = \sum_{k=1}^{K} p_n^k
\]

(2.31)

where \( K \) is the number of tranches.

The tranche weights are defined as the tranche size relative to the whole series and are given by

\[
\omega_n^k = \frac{p_0 \omega_0^k \prod_{j=1}^{n} (1 - \lambda_j^k)}{\sum_k p_0 \omega_0^k \prod_{j=1}^{n} (1 - \lambda_j^k)} = \frac{p_n^k}{p_n}
\]

(2.32)
Notice that the tranche weights add to unity.

\[ \sum_{k=1}^{K} \omega_k = 1 \]

### 2.9.3 Monte Carlo Formulation of MBS Valuation Problem

Let \( b^k_n \) be the payment from tranche \( k \) at payment date \( n \). It can be computed as

\[
b^k_n = \prod_{j=1}^{n-1} (1 - \lambda^k_j) \left( \lambda^k_n H^k_n + (1 - \lambda^k_n) A^k_n + I^k_n \right)
\]  

(2.33)

where \( A^k_n, I^k_n \) are the scheduled prepayment and interest payment from tranche \( k \) at payment date \( n \) respectively and \( H^k_n \) is the scheduled remaining debt balance at payment date \( n \) before a redemption payment is made. Note that when the prepayment function depends on previously computed \( \lambda \)'s, \( b^k_n \) will be path dependent.\(^{12}\)

Using (2.3) we find the value of a single payment as

\[
\hat{V}_t(b^k_n) = \mathbb{E}_t^{Q} \left[ b^k_n e^{-\int_t^{T_n} r_s ds} \right]
\]

where \( T_n \) is the payment date. Let \( \hat{V}_t^m(k) \) be an estimate of the MBS value along path \( m \). With \( N \) payment dates the price of tranche \( k \) at time \( t \) along path \( m \) is given by the sum of the prices of the individual payments along that path

\[
\hat{V}_t^m(k) = \sum_{n=1}^{N} \hat{V}_t^m(b^k_n)
\]

The crude Monte Carlo estimate of the value of tranche \( k \) is therefore given by

\[
\hat{V}_t(k) = \frac{1}{M} \sum_{m=1}^{M} \hat{V}_t^m(k)
\]

and the price of the total pool of mortgage loans is computed as a weighted sum of the prices of each of the tranches, times the tranche weights.

\[
\hat{V}_t(MBS) = \sum_{k=1}^{K} \omega_k \hat{V}_t(k) \]  

(2.34)

\(^{12}\)It is precisely these \( \lambda \)'s that depend on the series pool factor, and the value of the tranche weights must thus be known simultaneously.
2.9.4 An Example

To illustrate how least squares can be used to reduce the bias, we have plotted 95% confidence intervals for future MBS prices calculated with and without least squares. The prices are computed using 400 outer paths and 5, 50 and 500 inner paths respectively. No other variance reduction methods are used. The future trading date is July, 1, 2004 using data from January, 9, 2003. The results are shown in Figures 2.20, 2.21 and 2.22. The x-axis shows the future spot rates and the y-axis shows the simulated prices given the future state for a principal of DKK 1. The light gray lines are prices calculated without using the least squares approach and the black lines are calculated using least squares. Some of the variation in the figures can be explained by differences in variables like the pool factor and the debtor distribution, which is why the regressed expression is not a smooth function. In Figure 2.23 the corresponding spot rate distribution at option expiration is shown.

With only five inner paths the MC prices are, not surprisingly, extremely volatile. The maximum difference between the upper and lower bound in the confidence interval is as large as DKK 0.25 meaning that another simulation could result in a price DKK 0.25 away from the calculated price given the same state. Running a least squares regression, however, has a remarkable effect on the confidence interval. The maximum distance between the upper bound and the lower bound narrows to DKK 0.07 and if we only look at spot rates between 1% and 5%, the maximum difference declines to DKK 0.04. Results from increasing the number of inner paths to 50 are shown in Figure 2.21. As expected the MC price estimates becomes much less volatile compared to the prices in Figure 2.20. The maximum distance between the upper and lower bound in the confidence interval without using least squares is DKK 0.06 and only DKK 0.03 when a regression is applied. Finally, increasing the number of inner paths to 500 yields almost identical confidence intervals for the two methods. The maximum distance now falls to DKK 0.02 in the crude Monte Carlo case and to DKK 0.007 in the case where least squares is used. It is worth noting that even in the case of 500 inner paths the crude Monte Carlo confidence band is still not as good as the band obtained using least squares. In fact, in this example, it seems that the maximum distance between the upper and lower bounds declines with the same factor in both cases when the number of inner paths increases suggesting,
that it is always worthwhile to run a regression. For very low rates, however, there is large difference between the crude Monte Carlo prices and the prices obtained from least squares. The reason is that there are only few observations in this area making the regression bad.

In the regressions used above, all the simulated prices have been used. This approach can be refined. To improve the regression in areas which are important, for example in areas where the option is in the money, the regression could be performed using mostly in the money data.
Figure 2.20: 95% confidence intervals for MBS prices at option expiration
Crude Monte Carlo and least squares Monte Carlo prices for the underlying security at option expiration using 400 outer and 5 inner paths. No other variance reduction methods are used. The standard deviation of the least squares Monte Carlo estimate is found using 20 batches.

Figure 2.21: 95% confidence intervals for MBS prices at option expiration
Crude Monte Carlo and least squares Monte Carlo prices for the underlying security at option expiration using 400 outer and 50 inner paths. No other variance reduction methods are used. The standard deviation of the least squares Monte Carlo estimate is found using 20 batches.
Figure 2.22: 95% confidence intervals for MBS prices at option expiration
Crude Monte Carlo and least squares Monte Carlo prices for the underlying security at option expiration using 400 outer and 500 inner paths. No other variance reduction methods are used. The standard deviation of the least squares Monte Carlo estimate is found using 20 batches.

Figure 2.23: Spot rate distribution at option expiration
The distribution is found from the simulated spot rates associated with the least squares Monte Carlo simulation in the figures above, i.e. from 400 simulations.
Figure 2.24: Empirical cumulative distribution functions

Cumulative distribution functions for the relative pricing error in the models (Outer, Inner). The distributions are found on a 50 sample basis.
2.10 Appendix C: MBS Computational Test Case

Table 2.7
Basic bond data

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Isin code</td>
<td>DKJ0009265076</td>
</tr>
<tr>
<td>Coupon rate</td>
<td>6%</td>
</tr>
<tr>
<td>Maturity</td>
<td>01/10/2032</td>
</tr>
<tr>
<td>Payments per year</td>
<td>4</td>
</tr>
<tr>
<td>Amortization</td>
<td>Annuity</td>
</tr>
<tr>
<td>Initial pool factor</td>
<td>0.97</td>
</tr>
<tr>
<td>Tranche weight 1</td>
<td>0.15</td>
</tr>
<tr>
<td>Tranche weight 2</td>
<td>0.75</td>
</tr>
<tr>
<td>Tranche weight 3</td>
<td>0.10</td>
</tr>
</tbody>
</table>

The parameter values for the prepayment model used in this paper are given in Table 2.8 below.

Table 2.8
Basic bond data

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Tranche 1</th>
<th>Tranche 2</th>
<th>Tranche 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\beta_1$</td>
<td>0.15</td>
<td>0.20</td>
<td>0.25</td>
</tr>
<tr>
<td>$\beta_2$</td>
<td>-1.60</td>
<td>-1.00</td>
<td>-0.60</td>
</tr>
<tr>
<td>$\beta_3$</td>
<td>-4.20</td>
<td>-3.40</td>
<td>-2.50</td>
</tr>
<tr>
<td>$\beta_4$</td>
<td>1.80</td>
<td>1.20</td>
<td>0.80</td>
</tr>
<tr>
<td>$\beta_5$</td>
<td>-0.05</td>
<td>-0.10</td>
<td>-0.15</td>
</tr>
</tbody>
</table>

Parameters in the interest rate model are calibrated to be as shown in Table 2.9.

Table 2.9
Basic bond data

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\kappa$</td>
<td>0.03</td>
</tr>
<tr>
<td>$\sigma_0^2$</td>
<td>0.01</td>
</tr>
<tr>
<td>$\sigma_{0.25}^2$</td>
<td>0.009</td>
</tr>
<tr>
<td>$\sigma_{25-5}^2$</td>
<td>0.008</td>
</tr>
<tr>
<td>$\sigma_{2-10}^2$</td>
<td>0.007</td>
</tr>
</tbody>
</table>
Chapter 3

An Algorithm for Simulating Bermudan Option Prices on Simulated Asset Prices

Co-authored with Brian Huge, Danske Bank
Abstract

In this paper we present an algorithm that combines the (F. Longstaff & Schwartz, 2001) simulation algorithm and the bias reduction technique developed in (Huge & Rom-Poulsen, 2004) to simulate Bermudan option prices on securities so complex that their price must be found by Monte Carlo simulations. Bias reduction is needed because using simulated prices of the underlying security to compute option pay-offs causes an upward bias in the option price. We prove consistency of the option price computed by the algorithm.
3.1 Introduction

In this paper we use the (F. Longstaff & Schwartz, 2001) simulation algorithm to price Bermudan options written on securities so complex that their price must be computed by Monte Carlo simulations. In recent years, approximation methods for pricing Bermudan style options by Monte Carlo simulations have been developed. The first attempts to price Bermudan style options by Monte Carlo simulations were done in (Tilley, 1993) and (Carriere, 1996). Recently, several other methods have been proposed. These include (F. Longstaff & Schwartz, 2001) and (Tsitsiklis & Van Roy, 2001), the random tree method in (Broadie & Glasserman, 1997) and the stochastic mesh method of (Broadie & Glasserman, 2004), state space partitioning as in (Barraquand & Martineau, 1995) and the dual method in (Rogers, 2002). A survey of American option pricing using Monte Carlo simulations can be found in (Fu, Laprise, Madan, Su, & Wu, 2001) and chapter 8 in (Glasserman, 2004).

Not much attention has been given to the problem of pricing Bermudan options on securities that themselves must be priced by simulations. Although the theoretical problem is already covered by the methods above, the additional problem of simulating the price of the underlying security increases the computational burden considerably. In this paper we show that this does not have to be the case. To the best of our knowledge, this paper is the first to address this particular problem.

All the above mentioned methods use dynamic programming to solve the inherited maximization problem that must be solved in order to compute the price of a Bermudan option. Since the maximum is taken over all stopping times, dynamic programming is essentially used to find the optimal exercise policy for the option. The procedure is as follows; first, the state space is simulated using a (in time) forward running routine. Secondly, the dynamic programming is solved using a backward running routine. The backward running routine is a recursive procedure that is initialized at option expiration by setting the option value equal to the option pay-off. It then recursively works back to time 0 by, at each exercise date, setting the option value equal to the maximum of the immediate exercise value and the present value of the expected future option value, also known as the continuation value. I.e. at each exercise date a decision is made whether to exercise the option.
or to hold it for another period.

To use the dynamic programming approach, it is therefore necessary to be able to compute option pay-offs at each exercise date. However, if the price of the underlying security must be computed by simulations, the problem of computing option pay-offs poses a huge computational burden. For example, consider a Bermudan option with \( L \) exercise dates. Using \( U \) paths to simulate the state space and \( N \) paths to simulate prices of the underlying security at each exercise date results in a total of \( L \times U \times N \) path generations of the underlying security. However, by using the bias reduction technique in (Huge & Rom-Poulsen, 2004), the number of simulations used to compute the price of the underlying security, \( N \), can be reduced considerably. In fact, it is shown that using one single simulation, \( N = 1 \), can be sufficient to find prices of the underlying security and still get acceptable option pay-offs. However, the number of state space simulations, \( U \), must increase if \( N \) is to be low and a good functional relationship between the price of the underlying security and the state variables has to be established. We utilize this to develop a simulation algorithm that prices Bermudan options on complex securities.

Section 3.2 contains the model setup. The dynamic programming principle is presented for the original Bermudan pricing problem and for the Least-Squares Monte Carlo approximation. In addition we introduce our proposed extension, constructed to handle cases in which the value of the underlying security must be found by Monte Carlo simulations. In Section 3.3, our numerical test cases are presented along with a short description of the Hull-White interest rate model. An explicit expression for a class of basis functions in the Hull-White model is also developed for a bullet loan. Section 3.4 contains the results of the test cases. Finally, conclusions are made in Section 3.5.

3.2 The Model

Define a time line \( S = \{s_0, s_1, \ldots, s_M\} \) and let \( X_{s_0}, X_{s_1}, \ldots, X_{s_M} \) be a \( \mathbb{R}^d \)-valued Markov chain with \( X_{s_0} \) fixed. \( X_{s_i}, i = 1, \ldots, M \), represents the state variables which must be simulated and upon which the underlying security depends. \( S \) contains both
exercise dates and dates influencing the future price of the underlying security\(^1\). It is necessary to include non-exercise dates in \(S\) because, for path dependent securities, variables that are important for the underlying security, may need to be updated between exercise dates for future option pay-off computations to be possible. We also define \(T = \{t_0, t_1, \ldots, t_L\} \subseteq S\), \(t_0 = s_0\) and \(t_L < s_M\), as the set of stopping times. \(t_L\) is the expiration date of the option. We take as given a complete probability space \((\Omega, \mathcal{F}, \{\mathcal{F}_s\}_{i=0}^{M}, \mathbb{Q})\), where the discrete time filtration is generated by the relevant processes in the economy. We assume that an equivalent martingale measure \(\mathbb{Q}\) exists under which all expectations are computed. Let \(h_{t_i}(X_{t_i})\), \(t_i \in T\) be the \(t_0\) value of the pay-off at time \(t_i\). Since the pay-off function only influences the option value on exercise dates, we put \(h_{s_i}(X_{s_i}) = 0\) for all non-exercise dates \(s_i \in S\). This formulation of including the discounting into the pay-off function is discussed in (Glasserman, 2004) section 8.1. To lighten notation, we let the subscript \(i\) indicate time index \(t_i\) or \(s_i\).

### 3.2.1 Bermudan Option Pricing and Optimal Stopping

In this section we present the original Bermudan option pricing problem in which no approximations are made. The Bermudan option price is given as the value of the option if optimal exercise behaviour is followed by the holder of the option (see (Duffie, 1996)). The optimal exercise strategy amounts to choosing the time of exercise (stopping time) that maximizes the value of the option today. Let \(V^*_t(X_t)\) denote the value of the option at exercise date \(t_i \in T\) in state \(X_t\). The superscript \(^*\) indicates that this is the option price in the problem where no approximations are made. The value of the option is found by choosing, from the set of stopping times \(T_i = \{i, i + 1, \ldots, L\}, \tau \in T_i\) that maximizes \(V^*_t(X_t)\). I.e.

\[
V^*_t(X_t) = \max_{\tau \in T_i} \mathbb{E}_\tau^\mathbb{Q}[h_\tau(X_\tau)]
\]

\(V^*_t(X_t)\) is the value of the option given that the option was not exercised before date \(t_i\). We are interested in computing \(V^*_0(X_0)\), the value of the option today. The

---

\(^1\)Such dates could for example be payment dates of a callable mortgage backed bond, which will influence future values of the bond, because future payments from the bond might be dependent on historical prepayments.
optimal stopping time is found using dynamic programming, and $V_0^*(X_0)$ is found recursively as follows:

$$V_L^*(X_L) = h_L(X_L)$$  \hspace{1cm} (3.2)

$$V_i^*(X_i) = \max (h_i(X_i), E^Q_i[V_{i+1}^*(X_{i+1})]), t_i \in T, i < L$$  \hspace{1cm} (3.3)

At option expiration the value of the option is equal to its exercise value given by the function $h_L(X_L)$. An any exercise date $i$ before expiration, the option holder has to decide whether to hold the option for another period, which has the value $E^Q_i[V_{i+1}^*(X_{i+1})]$, or to exercise immediately yielding a pay-off of $h_i(X_i)$. Working backwards iteratively, this procedure gives the solution to the pricing problem in Equation (3.1).

The Bermudan option valuation problem can also be formulated using continuation values. Define

$$C_i^*(X_i) = E^Q_i[V_{i+1}^*(X_{i+1})]$$  \hspace{1cm} (3.4)

and the recursion procedure in (3.2) and (3.3) now becomes

$$C_L^*(X_L) \equiv 0$$  \hspace{1cm} (3.5)

$$C_i^*(X_i) = E^Q_i[V_{i+1}^*(X_{i+1})], t_i \in T, i < L$$  \hspace{1cm} (3.6)

$$V_i^*(X_i) = \max (h_i(X_i), C_i^*(X_i)), t_i \in T$$  \hspace{1cm} (3.7)

Obviously, the continuation value is 0 at option expiration. At any date before option expiration, the continuation value is equal to the present value of the expected option value one period ahead. The option value is then given as the maximum of immediate exercise and the continuation value.

As can be seen from (3.2) and (3.3) or (3.5), (3.6) and (3.7), the dynamic programming routine works backward in the time dimension, and solving it using a forward working Monte Carlo simulation therefore seem impossible. However, in recent years, methods that combine Monte Carlo simulations and the dynamic programming problem have been proposed.

### 3.2.2 Least Squares Monte Carlo Simulations (LSMC)

In order to distinguish approximated functions from functions from the original problem presented in Section 3.2.1, we will use the notational convention that ap-
proximated functions do not have the * superscript as opposed to functions in the original problem.

Simulation routines for pricing Bermudan type options approximates the continuation value in (3.4). This is the case in both the (Tsitsiklis & Van Roy, 2001) and the (F. Longstaff & Schwartz, 2001) approaches. In both algorithms the continuation value in (3.4) is approximated by a linear function of the form

\[ C_i^*(X_i) \approx \sum_{r=1}^{R} \zeta_{ir} \psi_{ir}(X_i) \equiv C_i(X_i) \quad (3.8) \]

where \( \psi_i(X_i) = (\psi_{i0}(X_i), \ldots, \psi_{iR}(X_i))^\top \) is the value of the basis functions at time \( t_i \) in state \( X_i \) and \( \zeta_i = (\zeta_{i0}, \ldots, \zeta_{iR}) \) are the coefficients to the basis functions. \( ^\top \) denotes the matrix transpose operator. \( \zeta_i \) is found by projecting the basis functions onto the continuation values. I.e.

\[ \zeta_i = (\mathbb{E}_t^Q [\psi_i(X_i)\psi_i(X_i)^\top])^{-1}\mathbb{E}_t^Q [\psi_i(X_i)V_{i+1}(X_{i+1})] \quad (3.9) \]

The (Tsitsiklis & Van Roy, 2001) and the (F. Longstaff & Schwartz, 2001) algorithms differ in the way the value in (3.7) is approximated. In (Tsitsiklis & Van Roy, 2001) the value is approximated by

\[ V_i(X_i) = \max(h_i(X_i), C_i(X_i)) \quad (3.10) \]

and in (F. Longstaff & Schwartz, 2001) it is approximated by

\[ V_i(X_i) = \begin{cases} h_i(X_i) & h_i(X_i) \geq C_i(X_i) \\ V_{i+1}(X_{i+1}) & h_i(X_i) < C_i(X_i) \end{cases} \quad (3.11) \]

In the (Tsitsiklis & Van Roy, 2001) algorithm, the option pay-off is directly determined by the approximated continuation value. In the (F. Longstaff & Schwartz, 2001) algorithm, the approximated continuation value is only used to determine the exercise boundary and does not enter directly into the option pay-off.

Using the approximation in (3.8), the dynamic programming problem can now be written as

\[ C_L(X_L) \equiv 0 \quad (3.12) \]
\[ C_i(X_i) = \sum_{r=1}^{R} \zeta_{ir} \psi_{ir}(X_i) , t_i \in T , i < L \quad (3.13) \]
and $V_i(X_i)$ equal to either (3.10) or (3.11).

In the rest of the paper we will focus on the (F. Longstaff & Schwartz, 2001) algorithm only.

### 3.2.3 Extension of the LSMC Method

As dictated by the dynamic programming routine we must be able to compute the immediate exercise value $h_i(X_i)$. When the price of the underlying security itself must be computed by simulations, an additional approximation is made. In that case, the option pay-off $h_i(X_i)$ is replaced by the approximation

$$\hat{h}_i(X_i) = f(\hat{P}_i(X_i))$$

where $\hat{P}_i(X_i)$ is the price estimate of the underlying security in state $X_i$ and $f(\hat{P}_i(X_i))$ is the pay-off function for a call/put taking as input the state dependent price of the underlying security at exercise date $t_i$. $\hat{P}_i(X_i)$ could be a crude Monte Carlo estimate, but as shown in (Huge & Rom-Poulsen, 2004) this creates a bias in the option price. To reduce this bias much computational effort has to be allocated to find $\hat{P}(X_i)$. (Huge & Rom-Poulsen, 2004) propose an alternative price estimator for the price of the underlying security. This estimator uses crude Monte Carlo estimates to construct a new estimator that is less dependent of the computational effort put into generating the crude Monte Carlo estimates. The study in (Huge & Rom-Poulsen, 2004) only considers European options but in this section their arguments are extended so that the bias reduction method can be applied to price Bermudan options as well.

We will make the following assumptions:

**Assumption 3.1** Assume that the true price of the underlying security, $P_i$, $t_i \in \mathcal{T}$, is given as a finite linear combination of $V$ basis functions. I.e.

$$P_i(X_i) = \sum_{v=1}^{V} v_{iw}\phi_v(X_i)$$

**Assumption 3.2** For each exercise date $t_i \in \mathcal{T}$, along any two paths $m,n$, an unbiased crude Monte Carlo estimate, with independent error terms, for the price
of the underlying security can be computed. I.e.

\[ \hat{P}_{i}^{MC,n}(X_i) = P_i(X_i) + \hat{\epsilon}_{i}^{MC,n} \]  
(3.16)

\[ \mathbb{E}_{i}^{Q} \left[ \hat{\epsilon}_{i}^{MC,n} \right] = 0 \]  
(3.17)

\[ \text{Var}_{i}^{Q} \left[ \hat{\epsilon}_{i}^{MC,n} \right] = \sigma_{\epsilon}^{2} \]  
(3.18)

\[ \text{Cov}_{i}^{Q} \left[ \hat{\epsilon}_{i}^{MC,m}, \hat{\epsilon}_{i}^{MC,n} \right] = 0, \forall m,n \]  
(3.19)

If Assumptions 3.1 and 3.2 are fulfilled, a better estimate for the price of the underlying security is given by

\[ \hat{P}_{i}^{LS}(X_i) = \sum_{v=1}^{V} \hat{u}_{v} \phi_{iv}(X_i) \]  
(3.20)

where \( \hat{u}_{i} = (\hat{u}_{i1}, \ldots, \hat{u}_{iV})^{T} \) is computed by

\[ \hat{u}_{i} = \left( \mathbb{E}_{i}^{Q} \left[ \phi_{i}(X_i) \phi_{i}(X_i)^{T} \right] \right)^{-1} \mathbb{E}_{i}^{Q} \left[ \phi_{i}(X_i) \hat{P}_{i}^{MC}(X_i) \right] \]  
(3.21)

I.e. given Assumption 3.1 and Assumption 3.2, noise is removed by projecting the crude Monte Carlo estimates of the price of the underlying security onto a set of basis functions.

We denote functions that use the approximations (3.14) and (3.20) by superscript \( ^{LS} \). The dynamic programming problem can now be stated as

\[ C_{L}^{LS}(X_L) \equiv 0 \]  
(3.22)

\[ C_{i}^{LS}(X_i) = \sum_{r=1}^{R} c_{ir}^{LS} \psi_{ir}(X_i), t_i \in T, i < L \]  
(3.23)

\[ V_{i}^{LS}(X_i) = \begin{cases} \hat{h}_{i}^{LS}(X_i) & \hat{h}_{i}^{LS}(X_i) \geq C_{i}^{LS}(X_i), t_i \in T, i < L \\ V_{i+1}^{LS}(X_{i+1}) & \hat{h}_{i}^{LS}(X_i) < C_{i}^{LS}(X_i), t_i \in T \end{cases} \]  
(3.24)

with \( \hat{h}_{i}^{LS}(X_i) = f(\hat{P}_{i}^{LS}(X_i)) \) and \( \hat{P}_{i}^{LS}(X_i) \) given by (3.20).

### 3.2.4 Solving with Simulations

In a practical implementation of the LSMC method from Section 3.2.2, the expectations in (3.9) are replaced by their sample counterparts thereby introducing one more approximation. Assume that \( U \) independent replications of the Markov chain have
been sampled and denote them by \( X_{ij}, \ldots, X_{Mj}, j = 1, \ldots, U \). Then the coefficients in (3.9) are estimated by

\[
\hat{\zeta}_i = \left( \frac{1}{U} \sum_{j=1}^{U} \psi_i(X_{ij})\psi_i(X_{ij})^T \right)^{-1} \left( \frac{1}{U} \sum_{j=1}^{U} \psi_i(X_{ij})V_{i+1}(X_{i+1,j}) \right)
\]  

The dynamic programming problem can now be written for \( j = 1, \ldots, U \)

\[
\hat{C}_{Lj}(X_{Lj}) \equiv 0 \quad (3.26)
\]

\[
\hat{C}_{ij}(X_{ij}) = \sum_{r=1}^{R} \hat{\zeta}_{ir}\psi_i(X_{ij}), \, t_i \in T, i < L \quad (3.27)
\]

where (3.11) is given by

\[
\hat{V}_{ij}(X_{ij}) = \begin{cases} 
\hat{h}_{ij}(X_{ij}) & \hat{h}_{ij}(X_{ij}) \geq \hat{C}_{ij}(X_{ij}) \\
\hat{V}_{i+1,j}(X_{i+1,j}) & \hat{h}_{ij}(X_{ij}) < \hat{C}_{ij}(X_{ij})
\end{cases} 
\]  

(3.28)

Because \( X_0 \) is fixed, the price at time 0 is given by

\[
\hat{V}_0(X_0) = \frac{1}{U} \sum_{j=1}^{U} \hat{V}_{1j}(X_{1j}) 
\]  

(Clemente et al., 2002) shows that as the number of replications, \( U \), increases and for \( R < \infty \) fixed, the option price estimator in Equation (3.29) approaches the value of the approximative option price defined by relation (3.8). They also show that as \( R \to \infty \) the stopping problem defined by the approximation (3.8) approaches the original optimal stopping problem.

In the extended LSMC approach from Section 3.2.3 we further have to approximate the price of the underlying security. The expectations in (3.21) are also found using the simulated paths and the coefficients can thus be computed by

\[
\hat{\upsilon}_i = \left( \frac{1}{U} \sum_{j=1}^{U} \phi_i(X_{ij})\phi_i(X_{ij})^T \right)^{-1} \left( \frac{1}{U} \sum_{j=1}^{U} \phi_i(X_{ij})\hat{P}_{ij}^{MC}(X_{ij};N) \right)
\]  

(3.30)

where \( N \) is the number of simulations used to generate the crude Monte Carlo prices.

The option price estimator is now defined as

\[
\hat{V}_{0LS} = \frac{1}{U} \sum_{j=1}^{U} \hat{V}_{1j}^{LS}(X_{1j}) 
\]  

(3.31)

We will now show that using least squares price estimates, \( \hat{P}_t^{LS} \), when computing option pay-offs, yields an option price estimator that converges to the option price...
computed using least squares Monte Carlo as the number of paths $U$ increases. The result does not depend on the number of simulations, $N$, used to generate the crude Monte Carlo estimates, $\hat{P}_i^{MC}(X_i)$, and is therefore valid for any such number.

**Proposition 3.1** Let $U$ be the number of crude Monte Carlo estimates used to estimate the coefficients in (3.30) and let $N$ be any positive integer. Assume that $\forall i$

\[
\psi(X_i)V_{i+1}^{LS}(X_{i+1}; U) \overset{d}{\to} \psi(X_i)V_{i+1}(X_{i+1}) \text{ as } U \to \infty \tag{3.32}
\]

\[
\sup_{U \geq 1} \mathbb{E}_i^\mathbb{Q}[|\psi(X_i)V_{i+1}^{LS}(X_{i+1}; U)|^\delta] < \infty \tag{3.33}
\]

for some $\delta > 1$. Given Assumptions 3.1 and 3.2, the option price computed using (3.22), (3.23) and (3.24) converge to the least squares Monte Carlo computed option price from Section 3.2.2 as $U$ approaches infinity, i.e.

\[
V_0^{LS}(X_0) \to V_0(X_0) \text{ as } U \to \infty
\]

**Proof:** See Appendix 3.6

An implication of proposition 3.1 is that the estimator (3.31) will approach the estimator (3.29) as $U \to \infty$.

### 3.2.5 Biases in the Least Squares Monte Carlo Algorithm

Two sources of bias arises when simulating Bermudan option prices. A high bias resulting from using information about the future in making decisions whether to exercise or not, and a low bias coming from applying a suboptimal exercise strategy. The high bias is a result of applying the dynamic programming principle to simulated paths. At each exercise date, the dynamic programming recursion states that the state dependent option value is equal to the maximum of immediate exercise and the estimated continuation value. But the estimated continuation value is found using a backward recursion on the future simulated paths and future information thus influences the current exercise decision. This is for example the case in the (Tsitsiklis & Van Roy, 2001) algorithm in which the state dependent option value will be equal to the continuation value if the continuation value is greater than immediate exercise.
To remove the high bias, it is necessary to separate the exercise decision from the continuation value. One way is to simulate a new continuation value in case of no exercise. An example of this can be found in (Broadie & Glasserman, 1997) resulting in an additional option price estimator that is biased low. In the original (F. Longstaff & Schwartz, 2001) algorithm the two sources of bias are mixed. The state dependent option value is equal to the maximum of the intrinsic value and the discounted value of the one period ahead option value. This is a special case of simulating a new continuation value in which the existing simulated paths are used to evaluate the new continuation value. In practice, the (F. Longstaff & Schwartz, 2001) algorithm thus produces low-biased option price estimates (see (Glasserman, 2004) chapter 8). Our algorithm contains a third source of bias, which we know is a high bias, but since the original (F. Longstaff & Schwartz, 2001) algorithm mixes high and low biases, the bias in the resulting option price cannot be determined.

3.2.6 The Simulation Algorithm

As shown in (Huge & Rom-Poulsen, 2004) it is sufficient to have \( N = 1 \) in (3.30) in order to get a good approximation for the price of the underlying security if the number of crude Monte Carlo price estimates \( (U) \) is sufficiently high. This can be used to construct a particular simple algorithm for pricing Bermudan options on securities that are priced by simulations. We start by sampling the underlying state space and store payments made by the underlying security (coupons, prepayments etc.). For each generated path, at an exercise date \( t_i \), the price of the underlying security is given as the time \( t_i \) value of future payments along that path. Thus, the crude Monte Carlo price of the underlying security is found using one single path. Before computing option pay-offs we improve the price estimate of the underlying security by regressing the crude Monte Carlo computed prices onto a set of state dependent basis functions. The estimated expression is then used to find the state dependent price of the underlying security, and this price is used to compute option pay-offs. Also, we run a regression in order to find the exercise boundary at time \( t_i \), just as in the (F. Longstaff & Schwartz, 2001) algorithm. The Bermudan option value at time \( t_i \) is then given as the maximum of the continuation value and the option’s approximated intrinsic value at time \( t_i \). The recursion is carried out for all
exercise dates from option expiration until today. In case of a Bermudan option on a bond, the state space is sampled from today until the maturity of the bond, not only until option expiration. The algorithm is described in details in Algorithm 4 below.

In the algorithm, \( \tilde{b}_ij^{MC} \) denotes the crude Monte Carlo simulated payment from the underlying security at time \( i \) along path \( j \). Recall that the set \( S \) is constructed so that both payment dates and the exercise dates belongs to \( S \), and the algorithm thus iterates over the dates in \( S \). In general the subscript \( ij \) denotes the value of some security at time \( i \) along path \( j \) whereas the subscript \( i \) alone denotes a \( U \times 1 \) vector with \( j \)’th row equal to the value along the \( j \)’th path \( (ij) \). Values of the underlying security are denoted by \( \tilde{F}_ij^{MC} \) and along the \( j \)’th path, prices of zero coupon bonds for the period \((t_i, t_k)\) are denoted by \( P_{ij}^{zcb}(k) \).

### 3.3 Test Cases

In this section we describe our computational test cases. Since our underlying interest rate model is the Hull-White model, we first present well known results for this model.

#### 3.3.1 The Hull-White Model

In the Hull-White model the \( Q \)-dynamics of the spot rate is given by

\[
    dr_s = (\Theta(s) - \kappa r_s) ds + \sigma dW_s
\]

where \( r_s \) is the spot rate, \( \kappa \) is the mean-reversion rate, \( \sigma \) is the spot rate volatility, and \( W_s \) is a Brownian motion. \( \Theta(s) \) is found so that the initial term structure of market interest rates are matched and is given by

\[
    \Theta(s) = \frac{\partial f^M(0,s)}{\partial s} + \kappa f^M(0,s) + \frac{\sigma^2}{2\kappa} \left(1 - e^{-2\kappa s}\right)
\]

where \( f^M(0,s) \) is the observed market term structure of forward rates.

Applying Ito’s lemma on \( e^{\kappa s} r_s \) yields the following solution to the stochastic
Algorithm 4 Extended Least Squares Monte Carlo

1: for \((j = 1\) to \(U\)) do
2:   for \((i = 1\) to \(M\)) do
3:      Simulate and store \(X_{ij}\)
4:      Compute and store \(\hat{b}_{ij}^{MC}(X_{ij})\)
5:   end for
6: end for
7: for \((i = M - 1\) to \(1\)) do
8:   for \((j = 1\) to \(U\)) do
9:      if \((i = M - 1)\) then
10:         \(\hat{P}_{i+1,j}^{MC} = \hat{b}_{ij}^{MC}\)
11:      end if
12:      \(\hat{P}_{ij}^{MC}(X_{ij}) = P_{ij}^{zc}(i + 1)\hat{P}_{i+1,j}^{MC} + \hat{b}_{ij}^{MC}\)
13: end for
14: if \((t_i \in T)\) then
15:   \(\hat{\upsilon}_i = \left(\frac{1}{U} \sum_{j=1}^{U} \phi(X_{ij})\phi(X_{ij})^T\right)^{-1}\left(\frac{1}{U} \sum_{j=1}^{U} \phi(X_{ij})\hat{P}_{ij}^{MC}(X_{ij})\right)\)
16:   for \((j = 1\) to \(U\)) do
17:      \(\hat{P}_{ij}^{LS}(X_{ij}) = \phi(X_{ij})^T\hat{\upsilon}_i\)
18:   end for
19: if \((i = L)\) then
20:   for \((j = 1\) to \(U\)) do
21:      \(\hat{C}_{i+1,j}^{LS}(X_{ij}) = \max(\hat{P}_{ij}^{LS}(X_{ij}) - K, 0)\)
22:   end for
23: else
24:   \(\hat{\upsilon}_i = \left(\frac{1}{U} \sum_{j=1}^{U} \psi(X_{ij})\psi(X_{ij})^T\right)^{-1}\left(\frac{1}{U} \sum_{j=1}^{U} \psi(X_{ij})P_{ij}^{zc}(i + 1)\hat{C}_{i+1,j}^{LS}(X_{i+1,j})\right)\)
25:   for \(j = 1\) to \(U\) do
26:      \(\hat{h}_{ij}^{LS}(X_{ij}) = \max(\hat{P}_{ij}^{LS}(X_{ij}) - K, 0)\)
27:      if \((\hat{h}_{ij}^{LS}(X_{ij}) \geq \psi(X_{ij})^T\hat{\upsilon}_i)\) then
28:         \(\hat{C}_{ij}^{LS}(X_{ij}) = \hat{h}_{ij}^{LS}(X_{ij})\)
29:      else
30:         \(\hat{C}_{ij}^{LS}(X_{ij}) = P_{ij}^{zc}(i + 1)\hat{C}_{i+1,j}^{LS}(X_{i+1,j})\)
31:      end if
32:   end for
33: end if
34: end if
35: end for
36: \(\hat{C}_{0}^{LS}(X_0) = \frac{1}{U} \sum_{j=1}^{U} P_{0j}^{zc}(1)\hat{C}_{1j}^{LS}(X_{1j})\)
differential equation in (3.34)

\[ r_s = e^{-\kappa(S-s)}r_s + \gamma(S) - e^{-\kappa(S-s)}\gamma(s) + \sigma \int_s^S e^{-\kappa(S-u)}dW_u \]

\[ \gamma(s) = f^M(0, s) + \frac{\sigma^2}{2\kappa^2} \left(1 - e^{-\kappa s}\right)^2 \]  

(3.36)

Since the stochastic integral \( \sigma \int_s^S e^{-\kappa(S-u)}dW_u \) is normally distributed with mean zero and variance \( \sigma^2 \int_s^S e^{-2\kappa(S-u)}du \) (see (Björk, 1998) Lemma 3.15), the conditional expected spot rate is given by

\[ \mathbb{E}_s^Q \left[ r_s \right] = e^{-\kappa(S-s)} + \gamma(S) - e^{-\kappa(S-s)}\gamma(s) \]

\[ \gamma(s) = f^M(0, s) + \frac{\sigma^2}{2\kappa^2} \left(1 - e^{-\kappa s}\right)^2 \]  

(3.37)

and has conditional variance

\[ \text{Var}_s^Q = \frac{\sigma^2}{2\kappa} \left(1 - e^{-2\kappa(S-s)}\right) \]  

(3.38)

Prices of zero coupon bonds are given by

\[ P_{zc}^b(s, S) = e^{\alpha(s, S) + \beta(s, S)r_s} \]  

(3.39)

where \( \alpha(s, S) \) and \( \beta(s, S) \) are given by

\[ \beta(s, S) = \frac{1}{\kappa} \left(e^{-\kappa(S-s)} - 1\right) \]

\[ \alpha(s, S) = -f^M(0, s)\beta(s, S) + \ln \left( \frac{P_{zc}^b(0, S)}{P_{zc}^b(0, s)} \right) + \frac{\sigma^2}{4\kappa} \beta^2(s, S)(e^{-2\kappa s} - 1) \]

Basis Functions in the Hull-White Model

Choosing a set of basis functions in (3.15) can be a difficult task. In this section we will choose a set of basis function that can be computed explicitly. To do this we let the underlying security be a bullet.

Start by writing the exponential function as the infinite sum

\[ e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} \]  

(3.40)

This leads to the following expression for a zero coupon bond in the Hull-White model

\[ P_{zc}(t, T) = e^{\alpha(t, T) + \beta(t, T)r_t} \]

\[ = e^{\alpha(t, T)} \sum_{n=0}^{\infty} \frac{(\beta(t, T)r_t)^n}{n!} \]  

(3.41)
The value at time $t \leq t_1$ for a security with non-stochastic payments $b_1, \ldots, b_N$ at the dates $t_1, \ldots, t_N$ where $b_i = b$ for $i = 1, \ldots, N - 1$ and $b_N = 1 + b$ can be written as

$$P(t) = P^{zcb}(t, t_N) + b \sum_{i=1}^{N} P^{zcb}(t, t_i)$$

$$= \sum_{i=1}^{N} b_i P^{zcb}(t, t_i)$$

$$= \sum_{i=1}^{N} b_i e^{\alpha(t, t_i)} \sum_{n=0}^{\infty} \frac{(\beta(t, t_i) r_t)^n}{n!}$$

$$= \sum_{n=0}^{\infty} \left( \sum_{i=1}^{N} b_i e^{\alpha(t, t_i)} \frac{\beta^n(t, t_i)}{n!} \right) r_t^n$$

$$= \sum_{n=0}^{\infty} v_n r_t^n$$  (3.42)

Line 1, 2 and 6 in Equation (3.42) all define a set of basis functions. Choosing the basis functions $P^{zcb}(t, t_N), \sum_{i=1}^{N} P^{zcb}(t, t_i)$ will lead to coefficients 1 and $b$. Choosing the basis functions $P^{zcb}(t, t_1), \ldots, P^{zcb}(t, t_N)$ will lead to coefficients $b_1, \ldots, b_N$. Both sets fulfill Assumption 3.1 but these choices would require information about the pricing function $P(t)$. Thus, these choices do not seem realistic so instead we have chosen powers of the spot rate as basis functions. In this case Assumption 3.1 is violated. However, we can choose an arbitrary number of basis functions to get sufficient close to the true price of the underlying security. This choice will also give us an indication of how serious it would be to violate Assumption 3.1.

By this choice of basis functions, the coefficients to the basis functions are given by

$$v_n = \sum_{i=1}^{N} b_i e^{\alpha(t, t_i)} \frac{\beta^n(t, t_i)}{n!}$$  (3.43)

3.3.2 Test Cases

In Table 3.1 we have listed the test cases that will be investigated in Section 3.4 below. Testcase1, Testcase2 and Testcase3 computes the Bermudan put option price on a bullet. To keep the computational burden to a minimum, we only have 2 exercise
dates. Testcase1 covers the case where the price of the underlying security, at each exercise date, is given as a known function of the state variables (spot rate), meaning that the true state dependent price of the underlying easily can be computed from the simulated spot rate (state variable). This corresponds to the algorithm in (F. Longstaff & Schwartz, 2001) and is basically Algorithm 4 without bias reduction (without line 15 to 18). In Testcase2 we apply Algorithm 4 without bias reduction to illustrate that this creates a high bias in the Bermudan put option price. The bias arises because highly uncertain crude Monte Carlo simulated price estimates of the underlying security are used to compute the value of immediate exercise in the backward recursion of the dynamic programming algorithm. Testcase3 is Testcase2 plus bias reduction. Testcase3 should thus be compared with Testcase1 in order to evaluate the effectiveness of our proposed algorithm. In Testcase4 we compute the price of a 2-year Bermudan option with weekly exercise giving a total of 104 exercise dates. The underlying bond is the same as for Testcase1, Testcase2 and Testcase3. This test case is included to study how effective the algorithm is in approximating American option prices. For Testcase1 to Testcase4, we compare the option price computed with Algorithm 4 with a PDE\(^2\) computed option price. Testcase5 is similar to Testcase4 except that the underlying security is a callable mortgage backed security instead of a bullet. This test case is included to study the effectiveness of the algorithm when the underlying security is complex, and Testcase5 should thus be viewed as the prototype pricing problem that our algorithm is tailored to solve.

The parameters in the HW-model \(\kappa, \sigma\) are chosen to be equal to 0.05 and 0.01 respectively and the model is calibrated to a flat yield curve of 5\% by computing \(\Theta(s)\) according to Equation (3.35). The trading date is April 6, 2004 and for Testcase1, Testcase2 and Testcase3 the exercise dates are April 6, 2005 and April 6, 2006 respectively. For Testcase4 and Testcase5 we have weekly exercise starting at April 13, 2004 and ending at April 4, 2006. The infinite sum in (3.42) is approximated\(^2\) with solving the fundamental Partial Differential Equation. We have used a Crank-Nicholson scheme to solve the partial differential equation.
Table 3.1  
**Test cases**

<table>
<thead>
<tr>
<th>Id</th>
<th>Underlying security</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Testcase1</td>
<td>4% bullet, maturity:01/06/2034, yearly payments</td>
<td>Put option, 2 exercise dates, true price for the underlying security</td>
</tr>
<tr>
<td>Testcase2</td>
<td>4% bullet, maturity:01/06/2034, yearly payments</td>
<td>Put option, 2 exercise dates, simulated price for the underlying security, no bias reduction</td>
</tr>
<tr>
<td>Testcase3</td>
<td>4% bullet, maturity:01/06/2034, yearly payments</td>
<td>Put option, 2 exercise dates, simulated price for the underlying security + bias reduction</td>
</tr>
<tr>
<td>Testcase4</td>
<td>4% bullet, maturity:01/06/2034, yearly payments</td>
<td>Put option, 104 exercise dates, simulated price for the underlying security + bias reduction</td>
</tr>
<tr>
<td>Testcase5</td>
<td>Mortgage backed bond</td>
<td>Put option, 104 exercise dates, simulated price for the underlying security + bias reduction</td>
</tr>
</tbody>
</table>

using the first nine terms yielding the following approximation for the bullet

\[ P(t) \approx \sum_{n=0}^{8} v_n r_i^n \]  

(3.44)

Note, that using only a finite number of terms in Equation (3.44) is a violation of assumption 3.1 and thus one more approximation is introduced. Since the underlying security is a bullet the payments received each year at April 1, are given by

\[ b_i = \begin{cases} 
0.04 & t_i < 01/06/2034 \\
1.04 & t_i = 01/06/2034 
\end{cases} \]  

(3.45)

3.4 Results

In this section we present the results of the test cases described in Table 3.1. Figure 3.1 shows the convergence of the least squares Monte Carlo computed option price as the number of simulations increases and when true prices of the underlying security are used to compute option pay-offs. This case corresponds to the traditional least squares Monte Carlo algorithm as described in Section 3.2.2. In the figure, $\pm 2$
standard deviations of the option price estimate are also shown. The dashed lines indicate ±2.5% deviations from the PDE computed option price. As can be seen, the least squares Monte Carlo computed option price converges to the PDE computed option price as the number of simulations increases. Also, not surprisingly, as the number of simulations increases, the uncertainty of the price estimate decreases. It is also worth noting that all the price estimates are almost identically to the PDE computed option price, even for a small number of simulations. This would probably not be the case if we were to price an option on a more complex underlying than the option on a bullet we investigate in Figure 3.1.

In Figure 3.2, the convergence of the least squares Monte Carlo computed option price for Testcase2 is shown. In this test case, true prices of the underlying security have been replaced with crude Monte Carlo simulated prices when option pay-offs are computed. No bias reduction are performed. Recall that when option pay-offs are computed, Algorithm 4 only uses the current path to compute the state dependent price of the underlying security. Of course this price estimate will be highly uncertain, since only one single path has been used to simulate the price. This leads to a high bias in the option pay-off, and we would expect the resulting least squares Monte Carlo computed option price to be biased high as well. Also,
the bias cannot be removed no matter how many simulations we put into computing the option price. The only way to reduce the bias is to lower the uncertainty of the crude Monte Carlo estimates of the underlying, which only can be achieved by increasing the work that is put into generating these prices. In many cases this is very computationally burdensome and Algorithm 4 is constructed in a way that makes it impossible to use more than one path in the simulation of the underlying.

When we add bias reduction to Testcase2 we get Testcase3. The convergence result for this test case is shown in Figure 3.3. For few simulations, the simulated option price is biased high. But as the number of simulations increase, the bias is reduced and the option price becomes almost equal to the PDE option price. Recall that we are only using 9 basis functions in the polynomial approximating the underlying bond price, but this does not seem to be of any problem. Another attractive feature of the algorithm is that both bias reduction and accuracy are improved by increasing the number of simulations. This is in contrast to Testcase2 in which only accuracy, but no bias reduction, is improved by increasing the number of simulations. Comparing with Testcase1, in which the true price of the underlying security was used to compute option pay-offs, the option prices computed in Testcase3 are very close to the ones computed in Testcase1. Only for 100, 200 and 400 simulations

Figure 3.2: Testcase2 - Convergence of least squares Monte Carlo. Diamonds represent the average relative pricing error. The upper and lower bar show ± 2 standard deviation of the pricing error respectively. The dotted lines indicate ±2.5% deviations from the PDE computed option price. The sample size is 1000.
there is a small difference between the option prices in the two test cases. However, the uncertainty induced by using simulated prices of the underlying security when computing option pay-offs also turns up in higher uncertainty on the simulated option price and this higher uncertainty persists even for 204800 simulations.

In Figure 3.4 the price of a Bermudan option with 104 exercise dates has been simulated. The effect of increasing the number of exercise dates from 2 to 104 is higher bias and higher standard errors, especially for a low number of simulations. But as the number of simulations increases, both the bias and the standard errors are reduced and for approximately 12800 simulations they are at the same low level as in Testcase3.

Finally, for Testcase5 we have simulated the price of a Bermudan option with 104 exercise dates on a MBS. The results are shown in Figure 3.5. We have not computed a PDE price for Testcase5, so the values shown are the simulated option prices. The conclusions are the same as for the previous test cases - both bias and standard errors are reduced when the number of simulations increase.

\textbf{Figure 3.3:} Testcase3 - Convergence of least squares Monte Carlo. Diamonds represent the average relative pricing error. The upper and lower bar show $\pm 2$ standard deviation of the pricing error respectively. The dotted lines indicate $\pm 2.5\%$ deviations from the PDE computed option price. The sample size is 1000.
3.4.1 Estimated Price Relation

In this section we will investigate the estimated price relations of the underlying security at the exercise dates for Testcase3. These estimated relations are used to compute option pay-offs and it is therefore crucial that they are good approximations of the true relation between the price of the underlying security and the state variables.

In Figure 3.6 the true relation between the price of the coupon bond and the spot rate is shown at the first exercise date April 6, 2005. The true price relation can easily be found in the Hull-White model since, in each future state, the entire yield curve is just an exponential affine function of the spot rate. Thus, the true price relation fulfills equation 3.42. Figure 3.6 also shows three estimated relations between the price of the underlying security and the spot rate. These relations are found by projecting the crude Monte Carlo simulated price estimates of the underlying security onto powers of the spot rate. I.e. they are approximations of the expression given by Equation (3.42). The approximations are found using 100, 6400 and 204800 simulations respectively. As can be seen, the approximation becomes more accurate as the number of simulations increase. For only 100 simulations,
the estimated price for high spot rates are completely wrong, whereas it is close to the true price relation for low and middle ranged spot rates. The reason for this divergence is that when we only use 100 simulations, there is little or no price information in the high spot rate region of the state space. The probability of getting a spot rate, at exercise date 1, above 8% is only 0.11% meaning that on average only 0.11 out of 100 simulated spot rates will be above 8%. Thus, the probability of getting a crude Monte Carlo estimate of the underlying security in this part of the state space is very low, making the estimated price relation very difficult to infer. The bias reduction method takes advantages of a diversification effect in the sense that if one has enough observations, the error term embedded in each observation can be filtered away. But since the elapsed time from today until the first exercise date is relatively short, the simulated spot rates are not very dispersed and will be concentrated around it’s conditional mean at exercise date 1. This can also be seen in Figure 3.6, namely in regions of the state space where there are many observations, the estimated price relations are very accurate. Looking at Figure 3.3 we see that the inaccurate estimated price relation of the underlying results in an inaccurate and biased option price. The reason why the computed option prices are not that inaccurate in spite of the divergence in the estimated price relation of the

Figure 3.5: Testcase5 - Convergence of least squares Monte Carlo.
Diamonds represent the average option price. The upper and lower bar shows ±2 standard deviation of the pricing respectively. The sample size is 1000.
underlying is, that the weight (probability) which these pay-offs get, is relatively low. Increasing the number of simulations to 6400 and 204800 removes the divergence of the estimated price relation and with 204800 simulations the estimated price relation is relatively close to the true price relation for all levels of the spot rate. This can also be seen in the option prices as a more accurate option price estimate (see Figure 3.3).

Now turn to exercise date 2. In Figure 3.7 the relation between the price of the underlying and the spot rate is shown for the same models as in Figure 3.6. For 6400 and 204800 simulations the conclusions are the same as for exercise date 1 - that the estimated price relations are good at approximating the true price relation over a large range of the state space. For 100 simulations, however, we have a much better approximation of the true price relation of the underlying. The reason is the longer time period between today and exercise date 2, which results in more dispersed spot rates over the state space than for exercise date 1. At exercise date 2, the probability of getting a spot rate above 8% is now 1.34%, more than 10 times higher than for exercise date 1.
3.4.2 Estimated Price Coefficients

In Section 3.3.1 we derived an expression for the coefficients to the basis functions in the Hull-White model for a bullet loan. In this section we investigate the first five estimated coefficients at exercise date 1 and 2. The true coefficients are computed by Equation (3.43) and they are summarized in Table 3.2.

In Figure 3.8, the bars show the relative errors on the estimated constant coefficient, $\nu_0$, as the number of simulations increases. In addition, the standard deviations of the relative errors are displayed as lines with values on the right hand axis. Relative errors and standard deviations of the relative errors are shown for

![Figure 3.7](image-url)  
**Figure 3.7:** Testcase3 - Model price-spot rate relation and estimated price-spot rate relation of the underlying coupon bond at exercise date 2. Initial spot rate is 5%
both exercise date 1 and exercise date 2. As can be seen, both the relative error and the standard deviation of the relative error decreases and approaches zero as the number of simulations increases. It can also be seen that the relative errors are lower for exercise date 2 than for exercise date 1, and that the standard deviation of the relative errors are lower on exercise date 2 than on exercise date 1. The only major exception for this pattern is the case with 100 simulations, where the relative error of $\nu_0$ is lower on exercise date 1 than on exercise date 2. It is also worth noting that, as the number of simulations increases, the differences in terms of relative errors and standard deviations of the relative errors between exercise date 1 and exercise date 2, vanishes. The reason for this phenomenon is that for a small number of simulations, the spot rates are more dispersed throughout the state space on exercise date 2 than they are on exercise date 1. This is illustrated in Figure 3.9 where the solution to the spot rate process in Equation (3.36) has been simulated on exercise date 1 and exercise date 2 respectively. Note also, that the standard deviations of the relative errors are very high for a small number of simulations.

The conclusions made for $\nu_0$ also apply for the estimated coefficients $\nu_1$, $\nu_2$, $\nu_3$ and $\nu_4$. This is illustrated in Figure 3.10, Figure 3.11, Figure 3.12 and Figure 3.13. The differences is that the level for both the relative error and the standard deviation of the relative error are much higher, the higher the power of the spot rate is. This suggests that a large part of variations on prices of the underlying coupon bond at exercise date 1 and exercise date 2 can be explained by the first basis functions in the approximation of the price given by (3.44) for this simple underlying security. This would probably not be the case if the underlying security were more complex, as would be the case for a mortgage backed bond. However, as the number of simulations increases, the estimates become more and more accurate, because the spot rates become more dispersed throughout the state space. Our results are consistent with the result in (Glasserman & Yu, 2003), saying that the number of simulations needed grows exponentially in the number of basis functions.
3.5 Conclusion

In this paper we develop an algorithm aimed at simulating Bermudan option prices written on securities that must also be priced by simulations. The method extends the algorithm developed in (F. Longstaff & Schwartz, 2001) by applying the bias reduction technique from (Huge & Rom-Poulsen, 2004) to reduce the bias in the option price induced by using noisy price estimates of the underlying security in the dynamic programming recursion. We show that if the price of the underlying security belongs to a space spanned by some basis functions, the result of the algorithm approaches the solution to the programming problem in (F. Longstaff & Schwartz, 2001). (Clemente et al., 2002) prove that this solution approaches the true option price as the number of basis functions and paths approaches infinity. However, generally the spanning assumption is not fulfilled, and the bias reduction thus introduces an additional approximation besides those introduced using the algorithm in (F. Longstaff & Schwartz, 2001).

We demonstrate, using simple test cases, that the algorithm is capable of computing Bermuda option prices that are very close to the option prices computed using a PDE approach.
The estimated coefficients to the basis functions seem to approach their true values, especially for the coefficients multiplying low powers of the spot rate. However, for a small number of simulations, the standard deviation of the estimated coefficient is extremely high. Many simulations are needed in order to get accurate coefficient estimates. This is in line with the findings in (Glasserman & Yu, 2003).

Using many simulations yields relatively accurate coefficient estimates and this, in turn, results in relatively good approximations of the relations between the price of the underlying and the spot rate. However, for a small number of simulations and short term exercise dates, the relation between the price of the underlying and the spot rate is only accurate in non-extreme regions of the state space. Thus, one must be sure to use a sufficient number of paths when simulating Bermudan option prices with a strike corresponding to extreme values of the state variables.
Figure 3.10: Testcase3 - Estimated $\nu_1$ coefficient.
Estimated $\nu_1$ at exercise date 1 and 2. The bars show the average relative error and are related to the left axis. The lines show the standard deviation of the estimate and are related to the right axis. The sample size is 1000.

Figure 3.11: Testcase3 - Estimated $\nu_2$ coefficient.
Estimated $\nu_2$ at exercise date 1 and 2. The bars show the average relative error and are related to the left axis. The lines show the standard deviation of the estimate and are related to the right axis. The sample size is 1000.
Figure 3.12: Testcase3 - Estimated $\nu_3$ coefficient.
Estimated $\nu_3$ at exercise date 1 and 2. The bars show the average relative error and are related to the left axis. The lines show the standard deviation of the estimate and are related to the right axis. The sample size is 1000.

Figure 3.13: Testcase3 - Estimated $\nu_4$ coefficient.
Estimated $\nu_4$ at exercise date 1 and 2. The bars show the average relative error and are related to the left axis. The lines show the standard deviation of the estimate and are related to the right axis. The sample size is 1000.
3.6 Appendix A: Proofs

3.6.1 Proof of Proposition 3.1

We want to show that the solution to the dynamic programming problem in Section 3.2.3 approaches the solution to the problem in Section 3.2.2 when the approximation in Equation (3.30) is used. In the following, we emphasise the dependence of the number of paths \( U \) used in the simulation of the state space and thus used in Equation (3.30) to estimate the coefficients in the least squares price estimate. By Assumptions 3.1 and 3.2, the price of the underlying security fulfills
\[
P_{LS}^i(X_i; U) \to P_i(X_i) \quad \forall \ i
\]
as \( U \) increases. Thus we have
\[
\hat{h}_{LS}^i(X_i; U) \to h_i(X_i) \quad \text{as} \quad U \to \infty
\]
By definition
\[
C_{LS}^i(X_i; U) = C_L(X_L) = 0 \quad \text{and} \quad V_{LS}^i(X_i; U) = \hat{h}_{LS}^i(X_i; U) \to h_L(X_L) = V_L(X_L) \quad \text{as} \quad U \to \infty.
\]
The proof now goes by induction. Assume that
\[
V_{LS}^i(X_i; U) \to V_i(X_i) \quad \text{as} \quad U \to \infty
\]
for \( j = i + 1, \ldots, L \). If we can prove that
\[
C_{LS}^i(X_i; U) \to C_i(X_i)
\]
as \( U \to \infty \), then we have
\[
V_{LS}^i(X_i; U) \to V_i(X_i) \quad \text{as} \quad U \to \infty
\]
and we are done. Now from (3.23)
\[
C_{LS}^i(X_i; U) = \sum_{r=1}^{R} \zeta_{ir}^L(U) \psi_{ir}(X_i)
\]
where
\[
\zeta_i^L(U) = (\mathbb{E}_{i}^{Q}[\psi_i(X_i) \psi_i(X_i)^T])^{-1} \mathbb{E}_{i}^{Q}[\psi_i(X_i) V_{LS}^{i+1}(X_{i+1}; U)]
\]
If
\[
\mathbb{E}_{i}^{Q}[\psi_i(X_i) V_{LS}^{i+1}(X_{i+1}; U)] \to \mathbb{E}_{i}^{Q}[\psi_i(X_i) V_{i+1}(X_{i+1})] \quad \text{as} \quad U \to \infty \quad (3.46)
\]
then \( \zeta_i^L(U) \to \zeta_i \) as \( U \to \infty \), which implies that
\[
C_{LS}^i(X_i; U) \to C_i(X_i) \quad \text{as} \quad U \to \infty
\]
Since \( \psi_i(X_i) V_{LS}^{i+1}(X_{i+1}; U) \) defines a sequence of stochastic variables as \( U \) increases, a sufficient condition for (3.46) is (3.32) and (3.33). \[\blacksquare\]
Chapter 4

Semi-Analytic MBS Pricing\textsuperscript{1}

\textsuperscript{1}This Essay is forthcoming in The Journal of Real Estate Finance and Economics. I am grateful to Lars Peter Lillevoe, Allan Mortensen and Carsten Sorensen for useful discussions and comments. All errors are of course my own.
Abstract

This paper presents a multi-factor valuation model for callable mortgage backed securities (MBS). The model gives semi-analytic solutions for the value of MBS. By modelling the size of the remaining debt, also known as the pool size, we show that the value of a single MBS payment due at time $t_n$ can be found by computing two expectations of the pool size at $t_{n-1}$ and $t_n$, respectively. To obtain semi-analytic solutions, we specify the pool size in such a way that these expectations can be computed by transform methods. The expectations are taken of two simple pay-offs, but instead of using the spot rate for discounting, a prepayment-adjusted discounting rate must be used. By combining the affine and quadratic pricing frameworks, flexible and sophisticated prepayment functions can be specified. The result is a MBS pricing model that is able to generate negative convexity as interest rates fall and still be close to the non-callable counterpart as interest rates rise. The valuation model is potentially useful in managing large MBS portfolios and in pricing options on MBS.
4.1 Introduction

In this paper a semi-analytic model for pricing fixed-rate callable mortgage backed securities (MBS) is presented. The model is based on ideas from credit risk modelling and belongs to the class of intensity-based models. At almost no computational cost the model can handle multiple state variables. Usually, MBS are priced with lattice methods, finite difference or Monte Carlo simulations, which is the only alternative if the dimension of the state space is high. The presented model thus offers an attractive alternative if computational speed is an important factor.

A callable mortgage backed security is a pass-through giving borrowers, at any point in time, the possibility to cancel their loans by paying back the outstanding loan balance at par. From the investor’s perspective, a MBS is a fixed rate bond plus a sold American call option on the remaining scheduled cash flow. In principle, prices of MBS can be computed by valuing the embedded American style option using standard option pricing theory. In its purest form, this procedure implicitly assumes that all debt holders prepay the first time it is economically profitable, a decision that is made endogenously within the model. However, such a model will produce MBS prices that only exceed par with the cost of prepaying, usually a few percent of the debt size. This is not consistent with observations in the market where some mortgage holders prepay as soon as it is profitable and others do not prepay at all, leading to MBS prices that far exceed par. To handle this heterogeneity among borrowers, different types of models have been proposed. One class of models uses an econometric model to explain prepayments by some exogenous variables that are assumed to determine the mortgagor’s prepayment decision. The most important explanatory variable in these models is the gain from refinancing. The model is then estimated by fitting it to observed prepayment data, and the heterogeneity among borrowers becomes a function of observed variables. Another line of research holds on to the rational hypothesis in option pricing theory and let differences in prepayment costs drive the heterogeneity among borrowers. Typically, such models assume that mortgage holders minimize their lifetime mortgage costs, which include both economical and non-economical costs. Examples of the first class of models are (Schwartz & Torous, 1989) and (Jakobsen, 1992) and examples of the second class
of models are (Stanton, 1995), (Kau, C., Muller III, & Epperson, 1995), (Yongheng, Quigley, & Van Order, 2000). (F. A. Longstaff, 2002) also belongs to the second class of models, and he extends the concept of costs to include borrower credit worthiness (see (Kau & Keenan, 1995) for an overview of the option approach for MBS pricing). In recent years, models that use elements from credit risk modelling to price MBS have emerged. This is for example the case in (Collin-Dufresne & Harding, 1999), (Kariya & Kobayashi, 2000), (Nakamura, 2001) and (Kau, Keenan, & Smurov, 2004).

In this paper an intensity-based model for pricing callable MBS is constructed. By modelling the pool size\(^2\) instead of the conditional prepayment rate\(^3\) (CPR) as is usual, we are able to derive a semi-analytic formula for the price of a callable MBS. The trick is to specify the pool size in such a way that it has a form that looks similar to the pay-offs that can be valued semi-analytically in the pricing frameworks of (Duffie, Pan, & Singleton, 2000) and (Leippold & Wu, 2002). From the pool sizes, an expression for the instantaneous CPR is derived and it is eventually this function that must be estimated. In this respect our model belongs to the first class of models described above. Using this specification, the value of a single MBS payment can be computed by valuing two simple pay-offs that only depend on the state variables at payment time. However, instead of discounting the pay-offs with the spot rate, they must be discounted by a prepayment-adjusted rate defined by the instantaneous CPR function. The advantage of this approach is that the values of these pay-offs are known in semi-analytic form. In the (Duffie et al., 2000) pricing framework, they are given as exponential affine functions of the state variables, and in the (Leippold & Wu, 2002) pricing framework they are given as quadratic exponential functions of the state variables. The coefficients on the state variables (and squared state variables) are in both cases deterministic functions of time, which in both the affine and quadratic pricing framework are found by solving a set of coupled ordinary differential equations (ODE). This, in turn, yields semi-analytical pricing formulas for the MBS price. Moreover, this ODE system only has to be solved once in order to value all payments from a particular MBS. The presented model can handle multiple

---

\(^2\)The pool size measures how many loans are left in the pool of mortgage loans.

\(^3\)The conditional prepayment rate gives the proportion of the pool size that prepays during a given period.
state variables allowing for realistic yield curve dynamics and a flexible specification of the prepayment model. We do not, however, model any burn out effects. A simple way to introduce burn out is to divide the MBS pool into sub-pools, each with its own prepayment model.

In this paper, a two factor affine term structure model (ATSM) is employed as the underlying interest rate model. More precisely, the two-factor shifted Gaussian model from (Brigo & Mercurio, 2001) is used. (Litterman & Scheinkman, 1991) show that yield curve dynamics are best described by using two or three state variables. Another reason for using multiple state variables is that correlation between rates of different maturities are decoupled allowing the prepayment model to depend on both short and long rates. The model fits into the (Duffie et al., 2000) pricing framework and provides closed form solutions for a wide range of securities. However, the pay-offs that can be priced must be affine functions of the state variables.

A good model for pricing MBS is a model in which CPR speeds up as interest rates fall. In (Schwartz & Torous, 1989) this is done by including cubic terms of the refinancing incentive. In our model, however, quadratic terms are used to accelerate prepayments for sufficiently low interest rates. The reason for using quadratic terms is that the quadratic pricing framework can then be applied. Quadratic terms structure models (QTSM) like ATSM, yield closed form solutions for many interest rate products. The difference between an ATSM and a QTSM is that in the former, the spot rate is given as an affine function of the state variables whereas in the latter, the spot rate is a quadratic function of the state variables. From (Dai & Singleton, 2000) we know that in ATSM, Gaussian and square root processes can be mixed. In QTSM, however, only Gaussian processes are allowed. The main reason that we have chosen the two-factor shifted Gaussian term structure model is that it also fits into the quadratic pricing framework. Since spot rates are affine in the state variables, we are now allowed to build prepayment functions that depend on quadratic terms of the spot rate and still remain within the quadratic pricing framework. Put another way, we mix the two pricing frameworks by working in the area where ATSM and QTSM coincide, to get a more sophisticated description of prepayments.4

4See (Duffie & Kan, 1996), (Dai & Singleton, 2000) and (Duffie et al., 2000) for papers on ATSM and
The paper by (Collin-Dufresne & Harding, 1999) is the one closest to the study in this article. Using an affine term structure model\(^5\), they derive semi-analytic pricing formulas for MBS. However, they are restricted to one-factor models. The specification of the instantaneous CPR is, like in our model, restricted to a specific form in order for their argument to work. We extend the work done in (Collin-Dufresne & Harding, 1999) by allowing for multiple state variables, which allows for a more sophisticated specification of the prepayment model. Since this model is so closely related to our work we will take a closer look at this model and show how it can be cast into the more general pricing framework of (Duffie et al., 2000), which also allows for jumps in the state variables. (Nakamura, 2001) also obtains semi-analytic MBS pricing formulas by modelling the individual loan’s exit time from the pool of mortgage loans.

We only consider MBS that are backed by the issuing institution and thus we treat them as default-free. Some papers in the literature also model prepayments resulting from mortgagors’ defaulting, which rationally happens when the value of the remaining mortgage payments exceeds the value of the property. In Denmark, however, some MBS pools are backed by the other mortgage holders in the pool and other MBS pools are backed by the issuing institution. This makes the need for modelling the default option less important for Danish MBS and we do not consider it in this study. Papers that include the default option are (Kau et al., 1995), (Kau & Keenan, 1995), (Yongheng et al., 2000), (Nakamura, 2001) and (Kau et al., 2004).

Besides providing fast computation of MBS prices and price sensitivities, the model can be used to construct efficient pricing routines for pricing options on callable MBS and prepayment dependent pay-offs. The price of a European option on a MBS for example, can be computed by simulating the state variables from today until option expiration, where the price of the underlying MBS is known in semi-analytic form, providing fast computation of the option pay-off.\(^6\) Alternatively, (Jamshidian, 1996), (Ahn, Dittmar, & Gallant, 2002), (Leippold & Wu, 2002), (Cheng & Scaillet, 2004) and (Chen, Filipovic, & Poor, 2004) for papers on quadratic term structures.

\(^5\)Their reported results are based on a Vasicek one-factor model, but their result carries over to all affine one-factor models.

\(^6\)Also, since Gaussian state variables are used, the joint distribution of the state variables is known, allowing for single-step simulation when no pool factor is present.
a new pricing routine would have to be initiated to compute the state dependent MBS price at option expiration as done in (Huge & Rom-Poulsen, 2004). A prepayment cap is a cap on prepayments made at a particular payment date. Since the pool size is a known function of the state variables, the value of a prepayment cap can easily be computed by a Fourier inversion of the discounted cap pay-off. The inversion must typically be done numerically, but since it implies solving a one-dimensional integral, efficient numerical routines exist.

The paper is organized as follows; in Section 4.2 the pool size is defined and we show how it is related to CPR. Furthermore, we show how to value a single mortgage payment by decomposing the mortgage into two terms that must be valued separately. This result is quite general and does not depend on any term structure model. In Section 4.3 both the affine and the quadratic pricing frameworks are introduced. Section 4.4 shows how the (Collin-Dufresne & Harding, 1999) model can be cast into the (Duffie et al., 2000) pricing framework. In Section 4.5, the specific two-factor intensity-based model for pricing MBS is introduced. First we present the shifted Gaussian interest rate model and then we present the prepayment model. Section 4.6 contains results from the model. Finally, Section 4.7 concludes on the paper.

4.2 Pool Size, CPR and Mortgage Payments

In each period, \([t_{i-1}, t_i]\), a percentage of the mortgage holders in the pool decide to prepay their loans. This percentage is called CPR\((t_{i-1}, t_i)\) and it depends on the state of the economy through the vector of state variables \(X_i\).\(^7\) The remaining amount of loans left in the pool after \(n\) periods is called the pool size and is denoted by \(Q_n\). The relation between CPR and pool size is

\[
Q_n = \prod_{j=1}^{n} (1 - \text{CPR}(t_{j-1}, t_j)) \tag{4.1}
\]

and in terms of the pool size, the CPR for the period \((t_{i-1}, t_i]\) is given by

\[
\text{CPR}(t_{i-1}, t_i) = \frac{Q_{i-1} - Q_i}{Q_{i-1}} \tag{4.2}
\]

\(^7\)For readability the following notation will be used \(X_i = X_{t_i}\).
Relation (4.2) states that the CPR is the amount of terminated loans in a period relative to the remaining loans left in the pool at the beginning of the period.

The pricing models to be used are formulated in continuous time and thus we define the instantaneous CPR as

\[
f(X_t) = \lim_{\Delta \to 0} \left( \frac{\text{CPR}(t, t+\Delta)}{\Delta} \right) = \frac{1}{Q_t} \lim_{\Delta \to 0} \frac{Q_t - Q_{t+\Delta}}{\Delta} = -\frac{dQ_t}{dt} = -\frac{d}{dt} \ln Q_t
\]

which yields the following relation between the instantaneous CPR and the pool size

\[
Q_t = e^{-\int_0^t f(X_u) \, du}
\]  
(4.3)

Ito’s lemma gives

\[
dQ_t = -f(X_t)Q_t \, dt , \quad Q_s = q_s
\]  
(4.4)

which says that over time, the pool size depreciates continuously at the rate \( f(X_t) \).

Also, by Equation (4.2), \( \text{CPR}(t_{i-1}, t_i) = 1 - Q_i/Q_{i-1} \), which yields

\[
f(X_i) \approx -\frac{\ln (1 - \text{CPR}(t_{i-1}, t_i))}{t_i - t_{i-1}}
\]  
(4.5)

over small intervals.

In this paper, instead of modelling \( \text{CPR}(t_{j-1}, t_j) \), the primary object to be modelled is \( Q_j \), which basically means that we are modelling the instantaneous CPR, \( f(X_t) \). However, \( f(X_t) \) cannot be chosen arbitrary since the following properties must be fulfilled

- \( Q_0 = 1 \)
- \( Q_t \in [0, 1] \ \forall \ t \)
- \( Q_t \) must be a non-increasing function of time

Furthermore, to obtain realistic prepayment functions, one would usually require that

- \( f(X_t) \) must be negatively correlated with interest rates
The first property states that the pool size at the time of MBS origination is equal to 1 and it is satisfied by construction as stated in relation (4.3). The second property states that the pool size must belong to the interval \([0, 1]\). The third property states that the pool size cannot rise, which basically is the same as requiring that \(f(X_t)\) is non-negative. The last property implies that CPR is negatively correlated with interest rates, which is a requirement because we want prepayments to rise when interest rates fall, and prepayments to fall when interest rates rise. The property is easily achieved, since the factors explaining prepayments are the same as the factors explaining interest rates. Alternatively, if the instantaneous CPR, \(f(X_t)\), was modelled by a separate state variable process, this process must be negatively correlated with the state variables driving interest rates.\(^8\)

The general expression for a single MBS payment due at time \(t_n\) is

\[
h_n = \prod_{j=1}^{n-1} (1 - \text{CPR}(t_{j-1}, t_j)) (\text{CPR}(t_{n-1}, t_n)(H_n + R_n) + (1 - \text{CPR}(t_{n-1}, t_n))(O_n + R_n))
\]

(4.6)

where \(H_n\) is the remaining debt as scheduled if no prepayments occur before any redemption payment is made, \(R_n\) is the scheduled interest payment, and \(O_n\) is the scheduled redemption payment. The product term scales the pool size according to prepayments up to time \(t_{n-1}\). At \(t_n\) a fraction of the mortgagors prepay their loans, this fraction is given by \(\text{CPR}(t_{n-1}, t_n)\). Borrowers who prepay must return the remaining debt balance given by \(H_n\) and interest for the period given by \(R_n\). Borrowers who do not prepay make a scheduled redemption payment, \(O_n\), and must also pay interest for the period.\(^9\)

Insert into Equation (4.6) the expression for the pool size Equation (4.1) and the expression for the CPR Equation (4.2) to get

\[
h_n = Q_{n-1} \left( \left( \frac{Q_{n-1} - Q_n}{Q_{n-1}} \right) (H_n + R_n) + \frac{Q_n}{Q_{n-1}} (O_n + R_n) \right)
\]

\[
= Q_{n-1} (H_n + R_n) - Q_n (H_n - O_n)
\]

(4.7)

Note that \(O_n\), \(R_n\), and \(H_n\) are known when the MBS is issued.

\(^8\)This is another reason to use a QTSM rather than a ATSM when pricing MBS. Negative correlation between state variables and positiveness of the state variables is not possible in ATSM, see (Dai & Singleton, 2000) page 1956.

\(^9\)Setting \(\text{CPR}(t_{j-1}, t_j) = 0 \ \forall j\) yields the cash flow from a non-callable bond.
Thus, by formulating the MBS payment in terms of the pool size as in Equation (4.7) instead of the CPR as in Equation (4.6), the nasty term $\prod_{j=1}^{n}(1 - \text{CPR}(t_j-1, t_j))$ disappears. Having the MBS payment due at time $t_n$ expressed in terms of functions of the pool size at time $t_{n-1}$ and $t_n$ greatly simplifies the pricing of MBS, as will become clear in the following sections.

We are now ready to compute the value of a single MBS payment. The time $t$ value of a mortgage payment due at time $t_n$ is equal to

$$V(t, h_n) = \mathbb{E}^Q_t \left[ e^{-\int_t^{t_n} r(X_u) du} h_n \right]$$

$$= (H_n + R_n)\mathbb{E}^Q_t \left[ e^{-\int_t^{t_n} r(X_u) du} Q_{n-1} \right]$$

$$- (H_n - O_n)\mathbb{E}^Q_t \left[ e^{-\int_t^{t_n} r(X_u) du} Q_n \right]$$

$$= (H_n + R_n)\mathbb{E}^Q_t \left[ e^{-\int_t^{t_{n-1}} r(X_u) du} Q_{n-1} \mathbb{E}^Q_{t_{n-1}} \left[ e^{-\int_{t_{n-1}}^{t_n} r(X_u) du} \right] \right]$$

$$- (H_n - O_n)\mathbb{E}^Q_t \left[ e^{-\int_t^{t_n} r(X_u) du} Q_n \right]$$

$$= (H_n + R_n)\mathbb{E}^Q_t \left[ e^{-\int_t^{t_{n-1}} r(X_u) du} Q_{n-1} P(t_{n-1}, t_n) \right]$$

$$- (H_n - O_n)\mathbb{E}^Q_t \left[ e^{-\int_t^{t_n} r(X_u) du} Q_n \right]$$

(4.8)

where the expectation is taken under the risk-neutral probability measure and where $P(t_{n-1}, t_n)$ is the time $t_{n-1}$ price of a zero coupon bond with maturity at time $t_n$. Inserting the expression for $Q_n$ yields

$$V(t, h_n) = (H_n + R_n)Q_t\mathbb{E}^Q_t \left[ e^{-\int_t^{t_{n-1}} (r(X_u)+f(X_u)) du} P(t_{n-1}, t_n) \right]$$

$$- (H_n - O_n)Q_t\mathbb{E}^Q_t \left[ e^{-\int_t^{t_n} (r(X_u)+f(X_u)) du} \right]$$

(4.9)

The quantity $Q_t = e^{-\int_0^t f(X_u) du}$ is the time $t$ pool factor, i.e. the remaining pool size at time $t$. If we want the prepayment function to be dependent on the pool size, this is the object that should enter into the prepayment function. Typically the current pool factor is approximated by the total current debt size relative to the initial debt size.

From the expression for the value of a single MBS payment, the MBS value is found by adding the values of all its payments. Note that the price computed by Equation (4.9) is the price for a pool size of $Q_t = e^{-\int_0^t f(X_u) du}$. Market prices,
however, are typically quoted for a pool size of 100. Thus, when normalised to a pool size of 1 the term $e^{-\int_0^t f(X_u) \, du}$ in (4.9) disappears.

The complicated task of finding the value of the expression in Equation (4.6) has now been reduced to that of computing the time $t$ value of the pay-offs $P(t_{n-1}, t_n)$ and 1. However, instead of using the spot rate, $r(X_t)$, for discounting, we must use a prepayment-adjusted rate equal to $r(X_t) + f(X_t)$. In other words, payments from callable MBS are discounted at a higher rate than payments from non-callable securities. In this respect, the formulation given here is similar to an intensity-based model used for pricing securities that are subject to credit risk. In intensity-based models, payments from bonds with credit risk are discounted by a spot rate plus a stochastic intensity representing the default risk of the bond in question. See e.g. (Lando, 2004) for a description of intensity-based credit risk models.

4.3 Computational Framework

In this section a short description of the affine and the quadratic pricing frameworks is given. In ATSM the spot rate is an affine function of the state variables and in QTSM the spot rate is a quadratic function of the state variables. Both frameworks give closed form pricing solutions for a wide range of pay-offs. The restriction is that the pay-offs can be written as an exponential affine or an exponential quadratic form of the state variables. Both models allow for jumps in the state processes, but this is not used in this paper.

A completed filtered probability space $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t=0}^T, \mathbb{Q})$ is taken as given. The filtration is generated by the relevant price processes in the economy. We assume the existence of a risk-neutral martingale measure, $\mathbb{Q}$, under which all expectations are taken. Assume that the uncertainty of the economy is described by a vector of state variables\(^{10}\)

$$dX_t = \mu(t, X_t)dt + \sigma(t, X_t)dW_t \quad (4.10)$$

where $\mu(t, X)$ is a $n \times 1$ vector defining the drift, $\sigma(t, X)$ is a $n \times n$ matrix defining the diffusion of the process, and $W_t$ is a $n \times 1$ vector of independent Brownian motions\(^{10}\) To include jumps in ATSM see (Duffie et al., 2000). To include jumps in QTSM see (Cheng & Scaillet, 2004).
under $\mathbb{Q}$.

**4.3.1 The Duffie, Pan, and Singleton (2000) Framework**

The affine set-up puts an affine structure on $\mu, \sigma \sigma^T$ and the spot rate, $r(X_t)$, as stated in the following

$$
\mu(t, X_t) = K_0 + K_1 X_t, \quad \text{for } K_0 \in \mathbb{R}^n, K_1 \in \mathbb{R}^{n \times n}
$$

$$
(\sigma(t, X_t) \sigma(t, X_t)^T)_{ij} = (H_0)_{ij} + (H_1)_{ij} \cdot X_t \quad \text{for } H_0 \in \mathbb{R}^{n \times n}, H_1 \in \mathbb{R}^{n \times n \times n}
$$

$$
r(X_t) = \omega_0 + \omega_1 \cdot X_t \quad \text{for } \omega_0 \in \mathbb{R}, \omega_1 \in \mathbb{R}^n
$$

The transform defined by

$$
\psi(u, X_t, t, T) = \mathbb{E}_t^Q \left[ \exp \left( - \int_t^T r(X_s) \, ds \right) e^{u X_T} \right] \quad (4.11)
$$

is known in closed form with solution

$$
\psi(u, X_t, t, T) = e^{\alpha(t, T) + \beta(t, T) \cdot X_t} \quad (4.12)
$$

where $\alpha(t, T)$ and $\beta(t, T)$ are solutions to the following ODEs

$$
\dot{\beta}(t) = \omega_1 - K_1^T \beta(t) - \frac{1}{2} \beta(t)^T H_1 \beta(t), \quad \beta(T) = u
$$

$$
\dot{\alpha}(t) = \omega_0 - K_0 \cdot \beta(t) - \frac{1}{2} \beta(t)^T H_0 \beta(t), \quad \alpha(T) = 0 \quad (4.13)
$$

The transform in (4.11) is in fact the price at time $t$ of a pay-off $e^{u X_T}$ at time $T$ and many pay-offs can be cast into this form allowing for semi-analytic pricing. For example, the price of a zero coupon bond is obtained by solving the ODEs in (4.13) with $u = 0$. If we allow for complex numbers, and price a security with time $T$ pay-off equal to $e^{i u X_t}$, where $i = \sqrt{-1}$, the transform in (4.11) actually gives us the discounted characteristic function for the stochastic variable $X_T$, which is the foundation for using Fourier transforms in option pricing. See also (Bakshi & Madan, 2000) for properties of this transform.

**4.3.2 Quadratic Interest Rate Modelling**

In quadratic term structure models the spot rate is a quadratic function of the state variables. This framework is a little more restrictive when specifying the state


variables in that only Gaussian processes are allowed.

\[ \mu(t, X_t) = K_0 + K_1 X_t, \quad \text{for } K_0 \in \mathbb{R}^n, K_1 \in \mathbb{R}^{n \times n} \]

\[ (\sigma(t, X_t)\sigma(t, X_t)^T)_{ij} = \Sigma \Sigma^T \quad \text{for } \Sigma \in \mathbb{R}^{n \times n} \]

\[ r(X_t) = c_r + b_r^T X_t + X_t^T A_r X_t \quad \text{for } c_r \in \mathbb{R}, b_r \in \mathbb{R}^n, A_r \in \mathbb{R}^{n \times n} \]

Positive spot interest rates are guaranteed by assuming \( A_r \) to be positive semidefinite and having \( c_r - \frac{1}{4} b_r^T A_r^{-1} b_r \geq 0 \).\(^{11}\) Note that in quadratic term structure models, the volatility is not allowed to depend on the state process vector.

The transform defined by

\[
\phi(u, X_t, t, T) = \mathbb{E}_t^Q \left[ \exp \left( - \int_t^T r(X_s) \, ds \right) e^{u^T X_T + X_T^T v} \right] \tag{4.14}
\]

with \( u \in \mathbb{R}^n \) and \( v \in \mathbb{R}^{n \times n} \) is known in closed form with solution

\[
\psi(u, X_t, t, T) = e^{\alpha(t, T) X_t + X_t^T \beta(t, T) X_t} \tag{4.15}
\]

where \( \alpha(t, T), \beta(t, T) \) and \( \gamma(t, T) \) are solutions to the following ODEs

\[
\begin{align*}
\dot{\gamma}(t) & = A_r - 2K_1^T \gamma(t) - 2\gamma(t)^T \Sigma \Sigma^T \gamma(t), & \gamma(T) = v \\
\dot{\beta}(t) & = b_r - K_1^T \beta(t) - 2\gamma(t)^T K_0 - 2\gamma(t)^T \Sigma \Sigma^T \beta(t), & \beta(T) = u \\
\dot{\alpha}(t) & = c_r - K_0^T \beta(t) - \frac{1}{2} \beta(t)^T \Sigma \Sigma^T \beta(t) - \text{tr} \left( \Sigma \Sigma^T \gamma(t) \right), & \alpha(T) = 0
\end{align*}
\tag{4.16}
\]

In Appendix 4.8, the ODEs are derived for the quadratic case. The ODEs in the affine case can be derived in the same way by setting the quadratic term equal to zero.

### 4.4 The Collin-Dufresne and Harding (1999) Model

Since the (Collin-Dufresne & Harding, 1999) model is similar to our model, this section takes a closer look at it and shows how to solve the model with the techniques used in this paper. In the original (Collin-Dufresne & Harding, 1999) paper only one-factor ATSM are allowed\(^{12}\), but this section demonstrates that their model in fact can cope with multiple factors by casting it into the (Duffie et al., 2000) affine pricing

\(^{11}\)To see this, differentiate \( r(X_t) \) with respect to \( X_t \) to find the \( X_t \) at which \( r(X_t) \) attains its minimum. Then insert this \( X_t \) into \( r(X_t) \) to find the restriction.

\(^{12}\)As stated in (Collin-Dufresne & Harding, 1999) page 134; “we are limited to a single state variable, the interest rate.”

111
framework. In the (Collin-Dufresne & Harding, 1999) model the instantaneous CPR is defined as

\[
f(X_t) = a_0 + a_1 \ln \left( \frac{P(0,T)}{P(t,T)} \right)
\]

\[
= a_0 + a_1 (\alpha(0,T) + \beta(0,T)r_0) - a_1 (\alpha(t,T) + \beta(t,T)r_t)
\]

where \( P(t,T) \) is the time \( t \) price of a zero coupon bond maturing at time \( T \) and \( T \) is the MBS maturity date. The mortgage bond is issued at time 0 and is priced at time \( t \). The fraction \( \ln(P(0,T)/P(t,T)) \) is a measure for the refinancing incentive, thus \( a_1 \leq 0 \). The instantaneous CPR is linear in the spot rate and there is nothing to accelerate prepayments when interest rates becomes sufficiently low. This is a drawback of the model. Note also that it is possible for \( f(X_t) \) to become negative for some choices of \( a_0 \) and \( a_1 \).

In this model, no bond specific variables enter the CPR function. This means that two different mortgage pools will have the same model price if they were issued at the same date and have the same maturity. Thus, the model implicitly assumes homogeneous mortgage holders over different pools. It is easy, however, to extend the model to take into account, for example, the coupon rate and time-to-maturity. For an example of how this could be incorporated into the model see Section 4.5.2.

Using a Vasicek interest rate model, the price of a mortgage bond with a continuous payout is computed by solving ODEs and numerical integration. The dynamics of the spot rate is given by

\[
dr_t = \kappa (\theta - r_t) dt + \sigma dW_t
\]

where \( W_t \) is a one-dimensional standard Brownian motion. The Vasicek model belongs to the class of affine term structure models and thus the framework from (Duffie et al., 2000) can be applied.

Prices of zero coupon bonds are obtained by taking \( u = 0 \) in (4.11), which yields
the well-know solutions to the ODEs in the Vasicek model\footnote{For later use we note that \( \int_{t}^{t_n} \beta(u,T) \, du = \frac{1}{\kappa} (\beta(t_n,T) - \beta(t,T) - (t_n - t)) \) and \( \int_{t}^{t_n} \beta(u,T)^2 \, du = \frac{1}{2\kappa} (\beta(t,T) - \beta(t_n,T) + (t_n - t)) - \frac{\sigma^2}{2\kappa \theta} (\beta(t,T)^2 - \beta(t_n,T)^2) \)}

\[
\begin{align*}
P(t,T) &= e^{\alpha(t,T)+\beta(t,T)r_t} \\
\beta(t,T) &= \frac{1}{\kappa} \left( e^{-\kappa(T-t)} - 1 \right) \quad (4.19) \\
\alpha(t,T) &= \left( \frac{\sigma^2}{2\kappa^2} - \theta \right) (\beta(t,T) + (T-t)) - \frac{\sigma^2}{2\kappa} \beta(t,T)^2
\end{align*}
\]

It is straightforward to extend this set-up to cope with deterministic coefficients in \( r(X_t) \) instead of constant coefficients. This is necessary since the discounting term in Equation (4.20) below has deterministic \( \omega_1(t,T) = (1 - a_1 \beta(t,T)) \). The extension is carried out simply by replacing the constant \( \omega_1 \) in Equation (4.13) with its deterministic counterpart.

Using the form for prices of zero coupon bonds from (4.19) and using the equations (4.17) and (4.9) the value of a mortgage payment can be written

\[
V(t,b_n) = (H_n + R_n)e^{\alpha(t_{n-1},t_n)}C(t,t_{n-1})
\]

\[
\mathbb{E}_t^Q \left[ e^{-\int_t^{t_{n-1}} (1-a_1 \beta(u,T)) r_u \, du} e^{\beta(t_{n-1},t_n) r_{n-1}} \right] - (H_n - O_n)C(t,t_n)\mathbb{E}_t^Q \left[ e^{-\int_t^{t_n} (1-a_1 \beta(u,T)) r_u \, du} \right] \quad (4.20)
\]

where

\[
C(t,t_n) = \exp \left( a_1 \int_t^{t_n} \alpha(u,T) \, du - (t_n - t)(a_0 + a_1 (\alpha(0,T) + \beta(0,T)r_0)) \right) \quad (4.21)
\]

Using footnote 13 we immediately have

\[
\int_t^{t_n} \alpha(u,T) \, du = \left( \frac{\sigma^2}{2\kappa^2} - \theta \right) \left[ \left( T - \frac{1}{2} t_n \right) t_n - \left( T - \frac{1}{2} t \right) t \right] + \left( \frac{3\sigma^2}{4\kappa^3} - \frac{\theta}{\kappa} \right) [\beta(t_n,t) - \beta(t,T) - (t_n - t)] + \frac{\sigma^2}{8\kappa^2} (\beta(t,T)^2 - \beta(t_n,T)^2) \quad (4.22)
\]

Each of the two expectations in (4.20) can be computed using the (Duffie et al., 2000) framework from Section 4.3.1. Their solutions have the form given in (4.12) where \( \alpha(t,T) \) and \( \beta(t,T) \) solves the ODEs (4.55) in Appendix 4.8 with boundary conditions \( \beta(t_{n-1},t_n) \) and 0 respectively and \( \omega_1 = (1 - a_1 \beta(t,T)) \).
4.5 An Intensity-Based Model for MBS Pricing

4.5.1 The Stochastic Interest Rate Model

In this section the intensity-based MBS pricing model is put forward. The general result from Section 4.2 is used to construct a specific model for valuing MBS semi-analytically. First, however, the interest rate model is presented, followed by the prepayment model. Since we want prepayments to depend on interest rates in a quadratic way, the term structure model is restricted to have a Gaussian specification. Thus, the relatively flexible G2++ model from (Brigo & Mercurio, 2001) is applied. The G2++ model is a two factor short rate model where the spot rate is given by

\[ r(X_t) = \varphi(t) + b^r_t X_t, \quad b^r_t = [1 \ 1] \]  

(4.23)

with \( r(X_0) = r_0 \) and where the state variables have the following \( \mathbb{Q} \)-dynamics

\[
dX_t = -\begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix} X_t dt + \begin{bmatrix} \sigma & 0 \\ \eta \rho \eta \sqrt{1-\rho^2} \end{bmatrix} dW_t, \quad X_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}
\]  

(4.24)

where \( dW_t \) is a two dimensional vector of independent Brownian motions. Correlation between the two factors is modelled by having a non-zero entry in \( \Sigma_{21} = \eta \rho \). The reason for this is that the formulations in both (Duffie et al., 2000) and (Leippold & Wu, 2002) assume independent Brownian motions. In contrast, (Brigo & Mercurio, 2001) work with correlated Brownian motion, but of course, the two formulations lead to identical pricing formulas. \( r_0, a, b, \sigma, \eta \) are positive constants and \(-1 \leq \rho \leq 1\) is the correlation. \( \varphi(t) \) is a deterministic function with \( \varphi_0 = r_0 \) used to match the initial yield curve.

\[
\varphi(T) = f^M(0,T) + \frac{\sigma^2}{2a^2} (1 - e^{-aT})^2 \\
+ \frac{\eta^2}{2b^2} (1 - e^{-bT})^2 + \rho \frac{\sigma \eta}{ab} (1 - e^{-aT}) (1 - e^{-bT})
\]  

(4.25)

\( f^M(0,T) \) is the market instantaneous forward rate at time 0 for maturity \( T \) defined by

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

(4.26)
and $P^M(0, T)$ is the price of a zero coupon bond with maturity $T$ currently observed in the market.

The model yields closed form solutions for a wide range of interest rate products. Especially, there exist closed form solutions for zero coupon bonds, European options on zero coupon bonds, interest rate caplets and European swaptions.

Defining the functions

$$
\beta_1(t, T) = \frac{1}{a} \left( e^{-a(T-t)} - 1 \right)
$$

$$
\beta_2(t, T) = \frac{1}{b} \left( e^{-b(T-t)} - 1 \right)
$$

allows us to write the yield on a zero coupon bond as

$$
y(X_t, T) = \frac{1}{T-t} \left( \int_t^T \varphi(u) du - \beta_1(t, T) X_{1,t} - \beta_2(t, T) X_{2,t} - \frac{1}{2} V(t, T) \right) \tag{4.28}
$$

where

$$
V(t, T) = \frac{\sigma^2}{a^2} \left[ T - t + \frac{2}{a} e^{-a(T-t)} - \frac{1}{2a} e^{-2a(T-t)} - \frac{3}{2a} \right] + \frac{\eta^2}{b^2} \left[ T - t + \frac{2}{b} e^{-b(T-t)} - \frac{1}{2b} e^{-2b(T-t)} - \frac{3}{2b} \right] + 2\rho \frac{\sigma \eta}{ab} \left[ T - t + \beta_1(t, T) + \beta_2(t, T) - \frac{1}{a+b} (e^{-(a+b)(T-t)} - 1) \right] \tag{4.29}
$$

and

$$
\exp \left( - \int_t^T \varphi(u) du \right) = \frac{P^M(0, T)}{P^M(0, t)} \exp \left( - \frac{1}{2} [V(0, T) - V(0, t)] \right) \tag{4.30}
$$

The two processes $X_{1,t}$ and $X_{2,t}$ drive the spot rate and they are correlated through the diffusion term for $X_{2,t}$. In contrast to one-factor models, yields of different maturities are not perfectly correlated in multi-factor term structure models, why it makes sense to build a prepayment function that depends on both the short and long rate. This is an attractive feature since a mortgagor (at least in Denmark) faces both refinancing alternatives in the short end of the yield curve in the form of adjustable-rate mortgage loans, and refinancing alternatives in the long end of the yield curve in the form of fixed-rate callable loans. In the quadratic pricing framework the pay-offs that can be priced are allowed to be quadratic in the state variables. This means that if we let the spot rate be an affine function of the state variables, the prepayment function can be quadratic in the spot rate. However,
Table 4.1
PARAMETERS IN THE SHIFTED GAUSSIAN TWO-FACTOR MODEL

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$</td>
<td>0.1467</td>
</tr>
<tr>
<td>$b$</td>
<td>0.3037</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>0.02742</td>
</tr>
<tr>
<td>$\eta$</td>
<td>0.03457</td>
</tr>
<tr>
<td>$\rho$</td>
<td>-0.8330</td>
</tr>
</tbody>
</table>

Notes: Parameters in the G2++ model taken from (Andersen & Andreasen, 2001).

since only Gaussian processes are allowed, the positiveness of the spot rate cannot be guaranteed.

We only consider prepayments that are functions of $X_1$ and $X_2$. However, it would be straightforward to introduce a third source of uncertainty by introducing a state variable representing non-interest driven prepayments. This could be relevant since prepayments can change dramatically when new refinancing opportunities comes to the market. This also suggests that including jumps into such a state process could be a good idea.\(^\text{14}\)

For our numerical illustrations we have taken the parameter values of the G2++ term structure model from (Andersen & Andreasen, 2001). The values are shown in Table 4.1. $\varphi(t)$ is then chosen to match the Danish swap curve as of January 3, 2005. The yield curve is shown in Figure 4.1.

4.5.2 The Prepayment Model

In the quadratic pricing framework the CPR function is allowed to have the following form

$$f(X_t) = c + b^T X_t + X_t^T A X_t$$  (4.31)

where $A$ is a $n \times n$ matrix, $b$ is a vector of size $n$ and $c$ is a scalar. It is possible to ensure positivity of $f(X_t)$ by $A$ being positive semidefinite and having $c - \frac{1}{4} b^T A^{-1} b \geq 0$. Such a restriction, however, should not be imposed uncritically since it might put

\(^{14}\)For example, adding to the system $X_3$ where $dX_3 = (-\kappa_3 X_{1,t} - \kappa_2 X_{2,t} - \kappa_3 X_{3,t}) dt + dW_3$ would introduce a third source of prepayment uncertainty not captured by interest rates but which could depend on the interest rate path through the feedback in the drift term. If the default option was modelled, $X_3$ could represent the house price dynamics or it could describe a stochastic pool factor.
too many restrictions on the parameters in the prepayment model. Worse case, it might prevent the prepayment model from becoming negatively correlated with interest rates. In this paper the restriction is hence not imposed.

Generally, $A$, $b$ and $c$ are allowed to be deterministic functions of time and this is used in the following to build a prepayment model that depends on both the short and long term interest rates.

**A Bond Specific Prepayment Model**

In this section we incorporate bond specific information into the prepayment function. Ideally we would like to use a prepayment model of the following kind

$$f(X_t) = a_0 + a_1 TT M + a_2 (COUPON - r(X_t))$$

$$+ a_3 (COUPON - r(X_t))^2 + a_4 (y(X_t, T) - r(X_t))$$  \hspace{1cm} (4.32)

where $COUPON$ and $TTM$ denote the coupon rate and term-to-maturity of the bond respectively. $r(X_t)$ is the spot rate and $y(X_t, T)$ is the time $t$ yield on a zero coupon bond with maturity $T$. The difference $(COUPON - r(X_t))$ measures the refinancing incentive and $(y(X_t, T) - r(X_t))$ measures the yield curve slope’s influence.
on prepayments. The quadratic term \((COUPON - r(X_t))^2\) is used to accelerate prepayment when the spot rate becomes sufficiently low.

The specification in Equation (4.32), however, involves products of \(\varphi(t)\) and \(X_i\), \(i = 1, 2\), terms which cannot be taken out of the expectations. The problem with this is that since \(\varphi(t)\) depends on the initial yield curve these terms make the ODE system non-homogeneous in the sense that two different MBS payments follow two different ODE systems. The ODE system becomes dependent on time \(t\) itself and not only on \(T - t\). For a MBS with 120 payment dates this means 120 ODEs of different lengths must be solved, which makes the pricing routines computationally inefficient. Thus instead of using the prepayment model in Equation (4.32) we propose the following variant of Equation (4.32) as prepayment model

\[
f(X_t) = a_0 + a_1 TTM + a_2 (D - (X_{1,t} + X_{2,t})) + a_3 (D - (X_{1,t} + X_{2,t}))^2 \\
+ a_4 \left( \frac{1}{T-t} \left( -\beta_1(t,T)X_{1,t} - \beta_2(t,T)X_{2,t} - \frac{1}{2}V(t,T) \right) - (X_{1,t} + X_{2,t}) \right)
\]

(4.33)

Instead of using \(r(X_t)\) and \(Y(X_t,T)\) directly we use the state variables and the \(V(t,T)\) function, which only depends on \(T - t\). In this way changes in the spot rate and the long rate caused by changes in the state variables are reflected in \(f(X_t)\). Also, we have introduced a new bond specific variable \(D\) instead of the \(COUPON\) of the MBS.

\(a_2, a_3\) and \(a_4\) measure the response in CPR to changes in refinancing incentives caused by changes in the spot rate and the long rate. \(a_3\) measures the response to the square of the refinancing incentive caused by changes in the spot rate. It is this term that is intended to speed up prepayments at sufficiently low interest rates. \(T\) is the MBS maturity date.

We cannot guarantee that the specification in Equation (4.33) always stays positive and this is a major drawback of this CPR specification. Also, due to the quadratic term, it is not necessarily negatively related to the spot rate \(r(t)\) for high rates. The implication is that for high interest rates the term \((D - (X_{1,t} + X_{2,t}))\) and for inverted yield curves the term \((-\beta_1(t,T)X_{1,t} - \beta_2(t,T)X_{2,t} - \frac{1}{2}V(t,T)) - (X_{1,t} + X_{2,t})\) may become negative. This can lead to negative CPR. Whether this poses a significant problem remains to be solved.

Having a quadratic term of the spot rate in the instantaneous CPR function is in
line with (Schwartz & Torous, 1989) who include cubic terms in their prepayment function specification. It is a major advantage of the model that such terms are allowed.

With this specification of the instantaneous CPR, the prepayment-adjusted spot rate \( r(X_t) + f(X_t) \) has a quadratic form given by

\[
f(X_t) + r(X_t) = c_t + b_t^* X_t + X_t^T A X_t
\]

(4.34)

with

\[
c_t = \varphi(t) + a_0 + a_1 TTM + a_2 D + a_3 D^2 - \frac{a_4}{2(T-t)} V(t, T)
\]

(4.35)

and for \( i = 1, 2 \)

\[
b_{it} = (1 - a_2 - 2a_3 D - a_4) - \frac{a_4 \beta_i(t, t + T)}{(T-t)}
\]

(4.36)

and

\[
A = a_3 \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}
\]

(4.37)

The value of a mortgage payment is given by Equation (4.9) and specializes to

\[
V(t, h_n) = (H_n + R_n) e^{\alpha(t_n, t_n)} \mathbb{E}_t^Q \left[ e^{- \int_t^{t_n-1} r(X_u) du} e^{\beta(t_n, t_n)^T X_n} \right] - (H_n - O_n) \mathbb{E}_t^Q \left[ e^{- \int_t^{t_n} r(X_u) + f(X_u) du} \right]
\]

(4.38)

where \( \alpha(t, T) \) and \( \beta(t, T) \) solve Equation (4.13) with \( \omega_1 = [1 \ 1]^T \), \( \omega_0(t) = \varphi(t) \), \( v = 0 \) and \( u = 0 \).

**Computing the Expectations**

From Section 4.3.2 we know that expectations of the form as in Equation (4.38) have solutions given by (4.15). Thus

\[
\mathbb{E}_t^Q \left[ e^{- \int_t^{t_n-1} r(X_u) du} e^{\beta(t_n, t_n)^T X_n} \right] = e^{\tilde{a}(t, t_n-1) + \tilde{\beta}(t_n, t_n) X_n + X_t^T \tilde{\gamma}(t, t_n-1) X_t}
\]

\[
\mathbb{E}_t^Q \left[ e^{- \int_t^{t_n} r(X_u) du} \right] = e^{\tilde{a}(t, t_n) + \tilde{\beta}(t_n, t_n) X_n + X_t^T \tilde{\gamma}(t, t_n) X_t}
\]

(4.39)

where \( \tilde{a}(t, t_n-1) \), \( \tilde{\beta}(t_n, t_n) \) and \( \tilde{\gamma}(t, t_n-1) \) solve Equation (4.16) with \( c_t \), \( b_t \) and \( A \) defined as in Equation (4.35), (4.36) and (4.37), respectively, with \( v = 0 \) and \( u = \beta(t_n-1, t) \). \( \tilde{a}(t, t_n) \), \( \tilde{\beta}(t_n, t_n) \) and \( \tilde{\gamma}(t, t_n) \) also solve Equation (4.16) with identical \( c_t \), \( b_t \) and \( A \), but where \( v = 0 \) and \( u = 0 \). Both expectations are solved using the prepayment-adjusted spot rate function \( \tilde{r}(X_t) = r(X_t) + f(X_t) \).
4.5.3 Prepayments Below Par

In Denmark it is possible for a mortgagor to cancel his loan by delivering the nominal amount of the remaining loan balance back to the issuing institution. Thus, if the market value of the underlying MBS is below par no prepayments will occur. To prevent prepayments below par, we would need to include an indicator function in the pool size as shown in the following

\[ Q_n = e^{-\int_0^{t_n} f(X_u) \mathbb{1}_{(P(X_u)>1)} \, du} \]  

(4.40)

where \( P(X_u) \) is the price of the MBS at time \( t_u \) conditioned on the state variables at that time. In (4.40) the pool size only decreases if the price of the mortgage bond is above par. Discounted expectations of such expressions are not possible to solve within the affine or quadratic pricing frameworks. Also, the recursive self-referencing nature induced by having the MBS price appearing in the indicator function, poses a problem. The MBS price depends on future prepayments which in turn depends on the pool size. Similar kind of expressions can be found in (Duffie & Huang, 1996) and (Huge, 2001) in connection with pricing swaps with two-sided default risk. In both papers the expectations are solved numerically.

4.6 Results

In this section we investigate whether the model is capable of generating some of the distinctive features of MBS. We are especially interested in studying whether the model is able to generate negative convexity when interest rates drop and generating MBS model prices close to a similar non-callable bond when interest rates rise. In order to do this we compare prices of MBS and the similar non-callable bond when the yield curve changes. In the following examples we have used a 30-year 6% annuity. The callable bond is labelled MBS and the non-callable counterpart is labelled BND. We generate several yield curves by varying \( X_{1,0} \) and \( X_{2,0} \). Following this procedure, all sorts of yield curve shifts occur. Thus, to ensure that our conclusions are not the result of a particular yield curve shift, we always compare MBS prices with BND prices. In this way we know that the conclusions are a consequence of the model.
Table 4.2
PARAMETERS IN THE PREPAYMENT MODEL

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_0$</td>
<td>0.0</td>
</tr>
<tr>
<td>$a_1$</td>
<td>0.0</td>
</tr>
<tr>
<td>$a_2$</td>
<td>1.5</td>
</tr>
<tr>
<td>$a_3$</td>
<td>25.0</td>
</tr>
<tr>
<td>$a_4$</td>
<td>1.0</td>
</tr>
<tr>
<td>$D$</td>
<td>0.08</td>
</tr>
</tbody>
</table>

Notes: Parameters used in the prepayment model.

The only purpose of this section is to demonstrate that the MBS valuation model is capable of generating some of the most important MBS price movements and to analyse how the prepayment parameters influence MBS model prices. Thus the parameter values used should not necessarily be taken as literal estimates of true values coming from an estimation using real world data. The parameters are chosen in order to demonstrate some of the strengths and weaknesses of the MBS valuation model.

The analysis is split into two parts. The first part computes, for a given set of prepayment parameters MBS and BND prices for changing yield curves. This part is primarily intended to demonstrate that the computed MBS model prices exhibit negative convexity for low rates and at the same time are close to BND prices for high rates. The second part is a partial analysis intended to show how the prepayment parameters influence MBS model prices. Like in Section 4.6.1, the prepayment parameters used have not been verified on real world data.

4.6.1 General Analysis

The analysis in this section uses the prepayment model parameters given in Table 4.2.

In Figure 4.2 and Figure 4.3, BND model prices and MBS model prices are shown as functions of the two initial state variables $X_{1,0}$ and $X_{2,0}$. Since the spot rate is equal to $r_t = \varphi_t + X_{1,t} + X_{2,t}$, the East corner corresponds to falling spot rates and the West corner corresponds to high spot rates (see also Figure 4.5). As Figure 4.2 shows, non-callable bond prices exhibit the well-known form of positive convexity
and, as Figure 4.3 shows, MBS prices exhibit negative convexity as the spot rate declines.

In Figure 4.4 and Figure 4.5, the CPR surface and the spot rate surface is shown. The model prices from Figure 4.2 and Figure 4.3 are based on these CPR and spot rates. As can be seen from Figure 4.4, CPR is quadratic in the state variables and this has the undesired effect that CPR can rise when interest rates are high. However, this only occurs for very high spot rates and we are able to guarantee, because of the quadratic form, positive CPR for all levels of the spot rate. Also, the quadratic form helps speeding up prepayments when the spot rate falls.

The black lines in Figure 4.2 to Figure 4.5 shows combinations of $X_{1,0}$ and $X_{2,0}$ where $X_{1,0} - X_{2,0} = 0.01$. In the following figures, Figure 4.6 to Figure 4.8, we will go into more detail with these lines on the surfaces. In Figure 4.6, MBS model prices and BND model prices are shown as functions of the spot rate conditioned on $X_{1,0} - X_{2,0} = 0.01$. The value of the non-callable bond exhibits positive convexity whereas MBS model prices bend off, as they should, when interest rates fall. The reason being that some borrowers terminate their loans by prepaying at par when it gets more attractive to refinance at lower interest rates. The other important
Figure 4.3: MBS prices as a function of X1 and X2 for a 30Y 6% annuity MBS.

Figure 4.4: CPR as a function of X1 and X2 for a 30Y 6% annuity MBS.
observation is that for high spot rates MBS model prices become close to prices of the non-callable bond.

In Figure 4.7, the MBS model prices along the black line from Figure 4.3 are shown, but now as a function of the spot rate instead of the state variables. It is seen that the MBS valuation model is able to generate both negative and positive convexity over the interest rate interval in question. For high interest rates, the MBS model price exhibits positive convexity and as interest rates fall the MBS model price starts to bend off resulting in negative convexity. This is a very important feature of a MBS valuation model.

In Figure 4.8, a two-dimensional plot of the black line in Figure 4.4 is shown. It shows CPR as a function of the spot rate and it is easy to see the quadratic nature of the CPR. The quadratic form speeds up prepayments for low spot rates, but has the unwanted effect of rising CPR for high spot rates. However, because of the quadratic form, CPR are positive for all levels of the spot rate.

Finally, we consider the distribution over time of today’s value of each single payment for both the MBS and the non-callable bond. As borrowers prepay their loans, payments are effectively moved from the far future until the near future. This also happens in the MBS pricing model as is evident from Figure 4.9 where today’s
**Figure 4.6:** Price sensitivity as a function of the spot rate for a 30Y 6% annuity non-callable and callable bond.
Spot rate found by varying initial X1 and X2. Initial spot rate is 2.39%.

**Figure 4.7:** Price sensitivity as a function of the spot rate for a 30Y 6% annuity callable MBS.
Spot rate found by varying initial X1 and X2. Initial spot rate is 2.39%.
value of each single payment for a MBS and a non-callable bond is shown. As can be seen, the MBS receives its payments in the beginning whereas the non-callable bond is not prepaid before maturity. In fact, by looking at Figure 4.9, it seems that the mortgage bond has been almost fully prepaid after 60 payment dates corresponding to 15 years.

4.6.2 Partial Analysis

We now turn to analysing the partial effect of changing prepayment model parameters. Generally we have that if CPR is positive, payments are moved from the far future to the near future and the prepayment adjusted discounting rate is higher than the spot rate. This causes MBS prices to be lower than BND prices for low interest rates, but it also means prepayments below par, putting an upward pressure on MBS prices relative to BND prices for high interest rates. In some cases the capital gain from receiving prepayments below par more than offsets the higher discounting rate and MBS prices become higher than BND prices for high spot rates. If CPR is negative, three effects influences MBS prices, two which push MBS prices above BND prices, and one which pushes MBS prices below BND prices. Negative
CPR causes MBS payments to be moved from the near future to the far future\textsuperscript{15} and this puts a downward pressure on MBS prices relative to BND prices. On the other hand, negative CPR lowers the discounting effect and, what is worse, negative CPR can press the pool size above unity (see Equation (4.3)). These two effects put an upward pressure on MBS prices relative to BND prices. Some of these effects will be illustrated in the following examples, where figures similar to Figure 4.6 are given for 3 different values of the particular prepayment model parameter we analyse.

In Figure 4.10, MBS prices have been computed for $a_0 = 0.05$, $a_0 = 0.10$ and $a_0 = 0.15$ and in Figure 4.11\textsuperscript{16} MBS prices have been computed for $a_1 = 2$, $a_1 = 4$ and $a_1 = 6$. We see that $a_0$ and $a_1$ have almost a similar effect on MBS model prices. The difference between $a_0$ and $a_1$ is that $a_0$ is independent of the time-to-maturity whereas $a_1$ is weighted by the time-to-maturity. Thus the difference between $a_0$ and $a_1$ is not so evident from this particular example but would be more apparent if two MBS with different maturities were compared. A positive parameter value, for either $a_0$ or $a_1$, ensures that CPR are always positive and we get lower MBS

\textsuperscript{15}Negative CPR can even lead to negative MBS payments for some payment dates.

\textsuperscript{16}When computing the MBS prices in Figure 4.11, the time-to-maturity has been normalised by a factor 100.
Figure 4.10: MBS price sensitivity.
MBS price as a function of the spot rate for prepayment parameter vector $(a_0, a_1, a_2, a_3, a_4, D) = (a_0, 0, 0, 0, 0).

Figure 4.11: MBS price sensitivity.
MBS price as a function of the spot rate for prepayment parameter vector $(a_0, a_1, a_2, a_3, a_4, D) = (0, a_1, 0, 0, 0, 0)$.  

128
prices for low spot rates relative to non-callable bond prices. However, when spot rate rises, MBS prices become higher than non-callable bond prices. The reason why MBS prices become higher than non-callable bond prices for high spot rates, is that prepayments at par have positive value for the investor when MBS prices are below par. The model allows for prepayments below par and this makes MBS more valuable than the non-callable counterpart even though MBS payments are discounted at a higher rate. When \( a_0 \) and \( a_1 \) are the only positive parameters in the prepayment model different from 0, \( f(X_t) \) is constant and this will eventually push the MBS price above BND price. The curvature does not seem to be affected for these values of \( a_0 \) and \( a_1 \).

In Figure 4.12 and Figure 4.13 the MBS price is shown as a function of the spot rate when \( a_2 \) and \( a_4 \) are varied respectively. \( a_2 \) is the response in prepayments when the spot rate changes and \( a_4 \) is the response in prepayments as a result of a changing long rate. To avoid starting out with negative CPR we have put \( D = 0.1 \) in Figure 4.12. As can be seen from Figure 4.12, the \( a_2 \) variable has the desired effect on MBS prices. A positive value of \( a_2 \) lowers the MBS price and we get negative convexity for low spot rates but also for high spot rates. Even though we get negative CPR.
when the spot rates hits approximately 22.5%, it does not seem to pose a major problem for the considered range of the spot rate. MBS prices are lower than BND prices for high spot rates, but not very much. However, as can be seen, the problem will become worse for spot rates above 30%. The $a_4$ variable should have almost the same effect as the $a_2$ variable, since the economic reasoning for including this term is the same as for the spot rate term - namely to include the gain from refinancing. However, for the considered yield curve shifts, since the range of the long rate is much smaller than for the spot rate, the effect on MBS prices is not so apparent as in the spot rate case. In fact, for the specific prepayment parameter values used in Figure 4.13, the term is not enough to yield negative convexity, and only MBS prices for high spot rates are affected. The MBS prices become lower than the BND prices because CPR is positive but small, and thus the discounting effect is dominating the gain from having prepayments below par. Also, quadratic terms involving the long end of the yield curve could be included, but this is not done here.

Figure 4.14 shows how the MBS price curve is influenced by the quadratic term in the prepayment function. In this example $D = 0$, since the quadratic term ensures positive CPR for all spot rates. Generally, the curvature of the function between
MBS prices and the spot rate changes by including this term. Both positive and negative convexity is generated with this term, but the fall in the spot rate is too small to generate negative convexity for this particular example. For low spot rates the MBS price is lower than the non-callable price, but for high spot rates the MBS price rises above the non-callable price. The reason is the quadratic relationship between instantaneous CPR’s and spot rates, which yields high CPR when the difference between $D(=0)$ and $(X_{1,t} + X_{2,t})$ is high. CPR is always positive and the higher MBS prices relative to the BND prices for high spot rates are due to prepayments below par that offset the discounting effect.

Finally, in Figure 4.15 and 4.16, MBS prices as functions of spot rates are shown for various values of $D$. Since $D$ only matters when $a_2$ and $a_3$ are different from zero, $a_2$ and $a_3$ have been set to 1 and 20 respectively. The parameter $D$ is used instead of the coupon of the MBS, and it can be used to control the level at which the CPR stemming from the spot rate term becomes negative. As can be seen from Figure 4.15, a higher value of $D$ twists the MBS price curve. From Figure 4.16 it seems that $D$ has more impact when $a_3$ is different from zero than when $a_2$ is the only other parameter different from zero. Different values of $D$ can in fact change
Figure 4.15: MBS price sensitivity.
MBS price as a function of the spot rate for prepayment parameter vector 
\((a_0, a_1, a_2, a_3, a_4, D) = (0, 0, 1, 0, 0, D).\)

Figure 4.16: MBS price sensitivity.
MBS price as a function of the spot rate for prepayment parameter vector 
\((a_0, a_1, a_2, a_3, a_4, D) = (0, 0, 1, 0, 0, D).\)
the curvature of the MBS price curve. In the examples from Section 4.6.1, \( D \) has been used to fine-tune the prepayment model.

4.7 Conclusion

In this paper, a model for semi-analytic MBS valuation has been presented. By switching the focus from the CPR to the pool size, we were able to express a single MBS payment due at time \( t_n \) as a function of two terms. The first term depends on the state variables at time \( t_{n-1} \) and the second is dependent on the state variables at time \( t_n \). By defining the pool size as a quadratic function of the state variables, we were able to compute the value of each single MBS payment in the quadratic pricing framework by solving a system of coupled ODEs. Solving ODEs can be done almost instantly even in high dimensions, yielding very fast MBS pricing routines. However, it is crucial that the prepayment function is specified in a way that makes the ODE system homogeneous so that the same ODE system can be used for all MBS payments.

We were able to cast the (Collin-Dufresne & Harding, 1999) model into the (Duffie et al., 2000) affine pricing framework, and thus we have extended their model to cope with multiple state variables. This also shows that their model is in fact an intensity based model. The model we propose, however, combines the affine and the quadratic pricing framework, the reason being that we want to speed up prepayments as interest rates fall.

Since the quadratic pricing framework is used, we were restricted to use Gaussian processes for the state variables. Thus we chose the two-factor shifted Gaussian model from (Brigo & Mercurio, 2001). A multi-factor model was chosen since correlation between yields of different maturities in these models are non-perfect, and we wanted prepayments to depend on both the short and long end of the yield curve. It is straightforward to include jumps in the state variables, which gives the possibility to build very sophisticated prepayment models.

We showed that the model was able to capture many of the features of MBS. In particular, it can generate both negative and positive convexity as the spot rate changes, and it can generate MBS prices very close to the non-callable counterpart.
for high spot rates. However, the model cannot, when solved semi-analytically, prevent prepayments below par, which is an issue in Denmark but not in the U.S. A more severe problem is the possibility of negative CPR.

Future work has to be put into finding the best prepayment model specification and to estimate it on real world data. This we leave to a future study.
4.8 Appendix A: ODE Derivation

In this section we go through the basic arguments leading to the solutions of the transforms in Equation (4.15) and to the ODE system in Equation (4.16). The arguments are valid both for the affine and the quadratic case and thus only the quadratic case will be considered. The affine case can be dealt with by setting the quadratic term equal to zero. The intention of this section is to give some intuition and feeling with the arguments used in transform pricing. First, however, the fundamental partial differential equation (PDE) is derived.

Let $X_t$ be a $n$-dimensional process with dynamics given by

$$dX(t) = \mu(t, X_t)dt + \Sigma(t, X_t)dW(t) \tag{4.41}$$

where $\mu_t$ is a $n$-dimensional vector, $\Sigma_t$ is a $n \times n$ matrix and $W_t$ is a $n$-dimensional vector of independent Brownian motions. As in Section 4.3.2, the spot rate is given as a quadratic function of the state variables

$$r(X_t) = c_r + b_r^T X_t + X_t^T A_r X_t \tag{4.42}$$

where $c_r$ is a constant, $b_r$ is a $n$-dimensional vector of constants, and $A_r$ is a $n \times n$ matrix of constants. $c_r$, $b_r$, and $A_r$ can be functions of time but for tractability we will here treat them as constants.

Using a Markov assumption allows us to write the time $t$ value of an asset as a function of the state variables at that time only and time itself, i.e. the value of any asset can be written as $V(t, X_t)$. Use Ito’s Lemma on $V(t, X_t)$ to get the dynamics in $V$.

$$dV = \left( \frac{\partial V}{\partial t} + \left[ \frac{\partial V}{\partial X} \right]^T \mu + \frac{1}{2} \text{tr} \left( \Sigma \Sigma^T \frac{\partial^2 V}{\partial X \partial X^T} \right) \right) dt + \left[ \frac{\partial V}{\partial X} \right]^T \Sigma dW_t \tag{4.43}$$

where tr denotes the trace of a matrix.

Under the risk-neutral pricing measure the bank account is used as numeraire. Now define

$$A(t) = e^{-\int_0^t r(X_u) du} \tag{4.44}$$

where $r(X_u)$ is the spot rate. The dynamics of $A(t)$ is

$$dA(t) = -r(X_t)A(t)dt \tag{4.45}$$
Since deflated prices are martingales under the risk-neutral measure, we can establish a PDE fulfilled by all assets by finding the dynamics in deflated prices and setting the drift term equal to zero. The dynamics in deflated prices is found by Ito’s lemma and is given by

\[ dA(t)V(t, X_t) = A(t) dV(t, X_t) + V(t, X_t) dA(t) \]  

which yields

\[
\frac{dA(t)V(t, X_t)}{A(t)} = \left( \frac{\partial V(t, X_t)}{\partial t} + A(t, X_t) - r(X_t)V(t, X_t) \right) dt 
+ \left[ \frac{\partial V(t, X_t)}{\partial X} \right]^T \Sigma(t, X_t) dW_t
\]

where

\[ A(t, X_t) = \left[ \frac{\partial V(t, X_t)}{\partial X} \right]^T \mu(t, X_t) + \frac{1}{2} \text{tr} \left( \Sigma(t, X_t) \Sigma(t, X_t)^T \frac{\partial^2 V(t, X_t)}{\partial X \partial X^T} \right) \]

The martingale property of deflated prices and the martingale representation theorem means that the dt-term is equal to zero, which holds for all t and for all X_t. Thus, any asset fulfills a PDE of the following kind

\[
\frac{\partial V(t, X_t)}{\partial t} + \left[ \frac{\partial V(t, X_t)}{\partial X} \right]^T \mu(t, X_t) + \frac{1}{2} \text{tr} \left( \Sigma(t, X_t) \Sigma(t, X_t)^T \frac{\partial^2 V(t, X_t)}{\partial X \partial X^T} \right) 
- r(X_t)V(t, X_t) = 0
\]

with boundary condition specifying the particular asset at hand. In the quadratic pricing framework such terminal pay-offs has the form \( V(T, X_T) = e^{c+b^T X_T + X_T^T \gamma X_T} \).

In order to find the ODEs that must be fulfilled for an asset to solve the PDE we do as follows; Assume that the solution has the form as stated in Equation (4.15). Take the relevant partial differentials and put them into the fundamental PDE in Equation (4.48). Thus we assume that \( V(t, X_t) \) has the following form

\[ V(t, X_t) = e^{\alpha(t) + \beta(t)^T X_t + X_t^T \gamma(t) X_t} \]

where \( \alpha(t) \) is a number, \( \beta(t) \) is a n-dimensional vector and \( \gamma(t) \) is a \( n \times n \) matrix. All are deterministic functions of time. We can now identify the boundary conditions from the terminal pay-off to be \( \alpha(T) = c, \beta(T) = b, \) and \( \gamma(T) = A \). Taking partial
derivatives gives
\[
\frac{\partial V}{\partial t} = \left( \frac{\partial \alpha(t)}{\partial t} + \left[ \frac{\partial \beta(t)}{\partial t} \right]^T X + X^T \left[ \frac{\partial \gamma(t)}{\partial t} \right] X \right) V(t, X)
\]
\[
\frac{\partial V}{\partial X} = (2\gamma(t)X + \beta(t)) V(t, X)
\]
\[
\frac{\partial^2 V}{\partial X^2} = (2\gamma(t) + (2\gamma(t)X + \beta(t)) (2\gamma(t)X + \beta(t))^T) V(t, X)
\]
(4.50)

Recall from Section 4.3.2 that \(\mu(t, X_t) = (K_0 + K_1X_t)\). Using this, Itô’s lemma, and the derivatives from (4.50) we get
\[
\left( \frac{\partial \alpha(t)}{\partial t} + \left[ \frac{\partial \beta(t)}{\partial t} \right]^T X + X^T \left[ \frac{\partial \gamma(t)}{\partial t} \right] X \right)
\]
\[
+ (K_0 + K_1X_t)^T (2\gamma(t)X + \beta(t))
\]
\[
+ \frac{1}{2} \text{tr} \left[ \Sigma \Sigma^T (2\gamma(t) + (2\gamma(t)X + \beta(t)) (2\gamma(t)X + \beta(t))^T) \right] - r(t) = 0
\]
(4.51)

It is now clear why only Gaussian processes with affine drift terms can be used in QTSM. If the drift term was not affine, the product of \((K_0 + K_1X)^T (2\gamma(t)X + \beta(t))\) would bring us out of the quadratic class. Likewise, if the diffusion term depended on \(X\), the product in the last line in Equation (4.51) would also bring us out of the quadratic class.

Using that for any matrix \(C\), \(\text{tr}(CXX^T) = X^TCX\), the last term in Equation (4.51) can be written
\[
\text{tr} (\Sigma \Sigma^\gamma(t)) + \frac{1}{2} (2\gamma(t)X + \beta(t))^T \Sigma \Sigma^T (2\gamma(t)X + \beta(t))
\]
(4.52)

and the PDE in Equation (4.51) now becomes
\[
\frac{\partial \alpha(t)}{\partial t} + K_0^\gamma(t) \beta(t) + \text{tr} (\Sigma \Sigma^\gamma(t)) + \frac{1}{2} \beta(t)^T \Sigma \Sigma^T \beta(t) - c_t
\]
\[
+ \left[ \frac{\partial \beta(t)}{\partial t} \right]^T X + 2K_0^\gamma(t)X + \beta(t)^T K_1X + 2\beta(t)^T \Sigma \Sigma^\gamma(t)X - b_t^\gamma X
\]
\[
+ X^T \left[ \frac{\partial \gamma(t)}{\partial t} \right] X + 2X^T K_1^\gamma(t)X + 2X^T \gamma(t)^T \Sigma \Sigma X - X^TA_t X = 0
\]
(4.53)

Since this PDE must hold for all \(t\) and all \(X_t\) each line in Equation (4.53) must be equal to zero, which yields the ODE system in Equation (4.16).

What remains to be shown is that Equation (4.49) is in fact a solution to
\[
E_t^Q \left[ e^{-\int_0^t r(X_u) du} e^{c+b^TX_t+X_t^\gamma(t)X_t} \right]
\]
Following Duffie et al., 2000 Proposition 1, we will show that the function defined by
\[
\Psi(t) = \exp \left( - \int_0^t r(X_u) du \right) e^{\alpha(t)+\beta(t)^TX_t+X_t^\gamma(t)X_t}
\]
is a martingale, for then $\Psi(t) = E_Q^t[\Psi(T)]$ and $\Psi(t)$ can be multiplied by $\exp(\int_0^t r(X_u) \, du)$ to get the result. From Equation (4.44) and Equation (4.49) we immediate see that $\Psi(t) = A_t V(t, X_t)$ and this is indeed a martingale since $\alpha(t), \beta(t)$ and $\gamma(t)$ solve the ODE system in Equation (4.16).

4.9 Appendix B: ODEs in the CDH-Model

For the MBS payment due at time $t_n$ we must solve two expectations. From (4.11) we have

$$E_Q^t \left[ e^{-\int_{t_n}^{t_{n-1}} (1-a_1 \beta(u,T)) r_u \, du} e^\beta(t_{n-1}, t_n) r_{n-1} \right] = e^{\bar{\alpha}(t,t_{n-1})+\bar{\beta}(t,t_{n-1}) r_t} \quad (4.54)$$

where $\bar{\alpha}$ and $\bar{\beta}$ solves

$$\frac{\partial}{\partial t} \bar{\beta}(t, t_{n-1}) = 1 + \frac{a_1}{\kappa} \left( 1 - e^{-\kappa(T-t)} \right) + \kappa \bar{\beta}(t, t_{n-1})$$

$$\bar{\beta}(t_{n-1}, t_{n-1}) = \beta(t_{n-1}, t_n)$$

$$\frac{\partial}{\partial t} \bar{\alpha}(t, t_{n-1}) = -\kappa \bar{\beta}(t, t_{n-1}) - \frac{1}{2} \kappa^2 \bar{\beta}(t, t_{n-1})^2$$

$$\bar{\alpha}(t_{n-1}, t_{n-1}) = 0 \quad (4.55)$$

We also have

$$E_Q^t \left[ e^{-\int_{t_n}^{t_{n-1}} (1-a_1 \beta(u,T)) r_u \, du} \right] = e^{\bar{\alpha}(t,t_n)+\bar{\beta}(t,t_n) r_t} \quad (4.56)$$

where $\bar{\alpha}$ and $\bar{\beta}$ solves

$$\frac{\partial}{\partial t} \bar{\beta}(t, t_n) = 1 + \frac{a_1}{\kappa} \left( 1 - e^{-\kappa(T-t)} \right) + \kappa \bar{\beta}(t, t_n)$$

$$\bar{\beta}(t_n, t_n) = 0$$

$$\frac{\partial}{\partial t} \bar{\alpha}(t, t_n) = -\kappa \bar{\beta}(t, t_n) - \frac{1}{2} \sigma^2 \bar{\beta}(t, t_n)^2$$

$$\bar{\alpha}(t_n, t_n) = 0 \quad (4.57)$$

This yields

$$\bar{\alpha}(t, t_{n-1}) = \kappa \theta \int_t^{t_{n-1}} \bar{\beta}(u, t_{n-1}) \, du + \frac{\sigma^2}{2} \int_t^{t_{n-1}} \bar{\beta}(t, t_{n-1})^2 \, du$$

$$\bar{\alpha}(t, t_n) = \kappa \theta \int_t^{t_n} \bar{\beta}(u, t_n) \, du + \frac{\sigma^2}{2} \int_t^{t_n} \bar{\beta}(t, t_n)^2 \, du$$
Chapter 5

Danish Summary


Det er på baggrund af disse udfordringer at emnerne i min afhandling er fremkommet. Afhandlingen er emnemæssigt placeret indenfor området ”numeriske metoder i finansiering” og motivationen for hver af afhandlingens artikler er behovet for hurtige prisfastsættelsesprogrammer. De enkelte artikler er resultatet af konkrete projekter i Kvantitativ Analyse, og flere af dem er på nuværende tidspunkt integreret i Danske Banks finansielle prisfastsættelsessystemer.

Selvom fastforrentede konverterbare realkreditobligationer spiller en rolle i alle tre artikler er dette ikke en afhandling om dette aktiv. Det er derimod den komplekse prisfastsættelse der gør fastforrentede konverterbare realkreditobligationer velegnet til at afprøve de udviklede numeriske metoder. Undtagelse fra dette er den sidste artikel i hvilken der udvikles en alternativ metode til prisfastsættelse af fastforrentede konverterbare realkreditobligationer.
Essay I: Bias Reduction in European Option Pricing

Denne artikel introducerer en ny metode til at reducere bias i prisen på europæiske optioner. Denne bias opstår hvis udregningen af options pay-off foretages på baggrund af usikre prisestimater på det underliggende aktiv. Da options pay-off funktion er konveks medfører Jensens ulighed at støjen i priserne på det underliggende aktiv forplanter sig til en højere pris på optionen. Intuitivt betyder støjen at prisvariansen på optionens udløbstidspunkt øges, hvilket resulterer i en højere options pris. Sådanne problemer opstår typisk i prisfastsættelsesproblemer hvor det underliggende aktiv bliver prisfastsat ved Monte Carlo simulering. Usikkerheden i prisestimaterne på det underliggende aktiv kan reduceres ved at øge den beregningsmæssige indsats, men dette kan i mange tilfælde føre til særdeles beregningstunge prisfastsættelsesrutiner. Alternativt kan man, ved at antage at prisen på det underliggende aktiv ligger i et underrum udspændt af en mængde af basisfunktioner, konstruere en estimator for prisen på det underliggende aktiv ved at projicere rene Monte Carlo estimater ned i underrummet udspændt af basisfunktionerne. Eftersom den sande pris ligger i dette underrum reduceres støjen i Monte Carlo prisestimaterne, og benyttes disse nye priseestimater til at beregne pay-off på optionen, i stedet for de rene Monte Carlo prisestimater, reduceres optionens prisbias. Artiklen demontrerer at metoden kan opfattes som en metode til at substituere antallet af stier brugt til at bestemme Monte Carlo priserne med selve antallet af Monte Carlo estimater. Vi viser, at hvis antallet af Monte Carlo estimater vokser uendeligt, vil metoden konvergere mod den sande options pris uafhængig af antallet af stier brugt til at finde Monte Carlo estimaterne af det underliggende aktiv. Således er det i grænsen kun nødvendigt at bruge én sti til at beregne Monte Carlo estimatet af prisen på det underliggende aktiv på optionens udløbstidspunkt, forudsat antallet af sådanne estimater vokser mod uendeligt. To eksempler illustrerer effektiviteten af metoden. I det første eksempel ligger prisen på det underliggende aktiv i et underrum udspændt af basisfunktionerne, i det andet eksempel er antagelsen ikke opfyldt. For begge eksempler er metoden i stand til effektivt at reducere bias i optionsprisen.
Essay II: An Algorithm for Simulating Bermudan Option Prices on Simulated Asset Prices

Essay III: Semi-Analytic MBS Pricing

I denne artikel præsenteres en multifaktor model til prisfastsættelse af konverterbare realkreditobligationer (MBS). Modellen giver semi-analytiske løsninger for værdien af MBS. Ved at modellere størrelsen af restgælden, også kendt som pool-størrelsen, vises det i artiklen at værdien af en enkelt MBS-betaling, til udbetaling på tidspunkt $t_n$, kan findes ved at udregne to forventninger af pool-størrelsen på tidspunkterne $t_{n-1}$ og $t_n$. For at få semi-analytiske løsninger er det nødvendigt at specificere pool-størrelsen på en måde så forventningerne kan løses med ”transform”-metoder. Forventningerne er givet som værdien af to simple betalinger, men i stedet for at diskontere disse betalinger med spotrenten skal de diskonteres med en diskonteringsrente der er justeret for førtidige indfrielser. For at få en så fleksibel og sofistikeret model for førtidige indfrielser som overhovedet muligt kombineres de affine og kvadratiske prisfastsættelsesmodeller. Resultatet er en prisfastsættelsesmodel der både kan generere positiv og negativ konveksitet på MBS. Prisfastsættelsesmodellen er ekstrem hurtig til at beregne MBS priser, og den kan derfor vise sig nyttig til styring af store MBS porteføljer og til prisfastsættelse af optioner på MBS.
References


Glasserman, P., & Yu, B. (2003). *Number of paths versus number of basis functions in...*


