Delft University of Technology
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft Institute of Applied Mathematics

Pricing American options using the Stochastic Grid Method with Bundling

A thesis submitted to the
Delft Institute of Applied Mathematics
in partial fulfillment of the requirements

for the degree

MASTER OF SCIENCE
in
APPLIED MATHEMATICS

by

Suzanne de Jong

Delft, Nederland
July 2012

MSc thesis APPLIED MATHEMATICS

"Pricing American options using the Stochastic Grid Method with Bundling"

SUZANNE DE JONG

Delft University of Technology

Daily supervisor	**Responsible professor**
Prof.dr.ir. C.W. Oosterlee	Prof.dr.ir. C.W. Oosterlee

Other thesis committee members

Dr.ir. R.J. Fokkink

Dr. J.A.M. van der Weide

S. Jain MSc

July, 2012 Delft

Acknowledgements

Hereby I would like to thank my daily supervisor Prof.dr.ir. C.W. Oosterlee for his guidance and advice, and also the other members of my thesis committee Dr.ir. R.J. Fokkink and Dr. J.A.M. v.d. Weide. A special thanks goes out to Shashi Jain MSc, for taking the time to explain his method and help me with the Matlab challenges that arose whilst working on the thesis. Second to last I would like to thank Dr. L. Grzelak for his help with the analytic solution to the Black-Scholes Hull-White model. Finally I want to thank my family and friends for their support during my study.

Contents

Chapter 1

Preface

The research field of computational finance searches among other things for new and efficient methods to price financial products. In this thesis we will discuss the refinement of a method for pricing (American) options using Monte Carlo simulations. The first chapter will give a short introduction to the pricing of American options, and why we need Monte Carlo simulations.

The method we will be refining in this thesis is known as the stochastic grid method and is described by Jain, S. and Oosterlee, C.W. in [9]. The stochastic grid method is a method that combines Monte Carlo simulations and least squares regression to obtain prices for American options in particular. By combining Monte Carlo simulations and least squares regression the computational effort compared to ordinary Monte Carlo simulations for American options can be significantly improved. However the current version of the stochastic grid method is still computationally rather expensive as we will see in chapter 3. The first research question we will address in this thesis is:

"How can we improve the stochastic grid method in order to decrease computational effort?"

After improving the stochastic grid method so that computational effort is reduced, we will focus on the second research question:

"How can we adapt the improved stochastic grid method for different models for the asset price?"

The current version of the stochastic grid method is so far only applicable if the Black-Scholes dynamics are used to model the asset price. For practical applications however it is a useful addition when we can also use models in which the interest rate and/or the volatility is stochastic. In this thesis we will focus on the Black-Scholes Hull-White model with stochastic interest rate in chapter 6, the Heston model with stochastic volatility in chapter 7 and the Heston-Hull-White model with stochastic interest rate and volatility in chapter 8.

In the conclusion at the end of this thesis we will give a concise answer to the research questions.

Chapter 2

Introduction to American option pricing

In this chapter a short introduction will be given to American option pricing. In order to do this thoroughly we will take a short look at what an option actually is, then we will see how we can price these options, and give a short explanation what the difficulties are if we want to price American options.

2.1 European Options

An option is a contract between two parties for a future transaction on a prespecified time, for a certain price. To make this formal definition a little more tangible we will look at the more practical definitions of a European call option and a European put option, the two most simple forms of options. Note that in option context we use the term 'asset' to describe any financial object whose value is currently known, but can change in the future.

European call option: A European call option is a contract between the option holder and the option writer, where the option holder has the right but not the obligation to purchase a certain asset from the option writer for a prescribed price at a prescribed time.

European put option: A European put option is a contract between the option holder and the option writer, where the option holder has the right but not the obligation to sell a certain asset to the option writer for a prescribed price at a prescribed time.

The prescribed asset in the option contract is also known as the underlying. The time at which the transaction may take place in the future is known as the expiration date or simply expiration and the prescribed price in the contract is known as the strike price or strike. So, mathematically we can say that at expiration time, T, we can write the value of an option with strike X as a function of the asset price at time T, S_T:

$$h(T, S_T) = (\Psi\,(S_T - X))^+ \qquad \Psi \in \{-1, 1\}$$

Where $\Psi = 1$ for a call option and $\Psi = -1$ for a put option, and the notation $(x)^+$ stands for $\max(x, 0)$.

Now in case of a put option as well as a call option the option writer will receive a certain amount of money, the option price, from the option holder when he buys the contract. In order to see why this make sense consider the following example:

Suppose that today, person A writes a European call option that gives person B the right to buy 100 shares of Royal Dutch Shell for €2500 a year from now, so after a year we could have one of two scenario's:

- The actual value of 100 shares of Royal Dutch Shell is more than €2500, person B will exercise his option and buy the shares from person A. If he would wish to do so he could then sell his shares immediately and make a profit.

- The actual value of 100 shares of Royal Dutch Shell is less than €2500, person B will let the option expire without exercising, hence making a loss nor a profit at expiration.

Clearly at expiration the option holder, person B, has no risk of making a loss. But the option writer, person A, on the other hand will certainly not make a profit at expiration but might make a potentially unlimited loss. Clearly the option writer will need some sort of compensation for this imbalance at expiration. So the person B will need to pay person A a certain amount now, known as the option price or value of the option. The difficulty lies in determining this option price.

2.1.1 Determining the European option price

As described above, the challenge is to find a good way to determine the option price. There are several methods for pricing (European) options. Which method we can use depends on the model we use for the price of the underlying. The simpler the model for the asset price, the easier it is to determine the option price, but at the same time usually the more inaccurate the results. Because if we take a very simple model for the asset price, like the Black-Scholes model, where the asset price follows a geometric Brownian motion with constant drift and variance (in finance governed by the volatility parameter), we can for European options determine the option price analytically. In practice however the interest rate and also the variance aren't constant. Because we know so much about this model it is usually the model that we start from, to later expand the model to stochastic interest rates and/or volatility.

For a little more involved models analytic solutions usually aren't available so we will need to resort to some form of approximation for the option price. Examples of these type of methods are the COS-method, described by Fang and Oosterlee ([6]), or Monte Carlo methods, as described by Glasserman in [8]. The COS-method approximates the option price using Fourier-cosine expansions. Monte Carlo methods are based on path simulations, in order to approximate the option value. In this thesis we will be focussing on a special type of Monte Carlo simulations.

2.2 American options

Now that we have seen a global overview of the simplest form of options, the European put and call option, it is time to introduce the little more involved type of option that will be studied in this thesis, the American option. An American put or call option is the same as a European put or call option, but in addition the holder may choose to exercise the option at any time prior to expiration. As can be imagined this makes pricing these options a lot more involved than pricing European options, because now we will also need some way of determining when the option will be exercised in order to obtain an accurate option price. Here we always assume a worst case scenario for the option writer, hence we assume that the option holder will exercise optimally.

2.2.1 Determining the American option price

As described above it will be harder to price an American option than it was to price a European option. This will actually only hold for an American put option. Because it can be shown that under the assumption of an efficient market, and the no arbitrage principle (you can never make a riskless profit bigger than the interest rate on a bank account) that in fact you will never exercise an American call option before expiration, and hence the price of an American call option will be the same as the price of a European call option, providing that no dividends are paid during the lifetime of the option. Then we can simply determine the price of the European call option in order to find the price of the American call option. So we will devote our attention in this thesis to pricing American put options.

For American put options it often is advantageous to exercise before expiration. There will no longer be an analytical solution for the option price, even if we assume that the asset price can be modeled by a very simple model like the Black-Scholes model. So now we always need to use some form of an approximation method. The COS-method, mentioned for European options, can also be used for pricing American options, but in this report we will devote our attention to a Monte Carlo method. Clearly it will be much harder to price an American option using Monte Carlo methods than it is to price European options using Monte Carlo. That is because at each timestep we have to determine whether it is optimal to exercise or hold. We have to go backwards in time, using the fact that at expiration we have that the value of the American put is the same as the value of the European put option hence at time T we have that:

$$V_T = (X - S_T)^+$$

But now at each timestep $0 \leq t \leq T$ we need to determine the option value as:

$$V_t = \max \{X - S_t, Q_t(S_t)\}$$

Where $Q_t(S_t)$ is the so called continuation value of the option at time t with asset price S_t. This is the value of the option if we don't exercise at time t. The difficulty in pricing American options using Monte Carlo methods lies in efficient computation of the continuation value, in a way that doesn't require the use of too many asset paths and computation time.

Chapter 3

Stochastic Grid Method

In this chapter we will describe the Monte Carlo method we will be studying in this report. The stochastic grid method uses Monte Carlo situations in a clever way, combined with least squares regressions in order to approximate option prices of (high-dimensional) American options efficiently. After giving a thorough introduction of the stochastic grid method in its most simple form we will also do some simulation experiments using this method, and make a comparison to the Longstaff-Schwartz method (another Monte Carlo method suitable for American options) in order to gain some insight in the performance of the stochastic grid method. Finally then we will take a look at a first opportunity to make the method more efficient in the form of a bundling strategy, where we bundle asset paths at each time step to gain computation speed.

The stochastic grid method (SGM) as described in [9] provides us a framework for performing Monte Carlo simulations on (high-dimensional) American options. The SGM computes the option price recursively moving backwards in time. By regression a functional approximation of the option price is obtained which is then used to compute the option price at the previous time step. First the method computes a direct estimator of the option price and determines the optimal exercise policy, and then a lower bound value of the option price is computed by discounting the payoff that is obtained when this exercise policy is followed. This method is also viable for high-dimensional options since the regression takes place along the payoff, hence the dimensionality of the problem is reduced to one in every recursion step. We will now give a description of the method after first defining the problem setting.

3.1 Problem

We assume we have the complete probability space $(\Omega, \mathcal{F}, \mathcal{P})$ and finite time interval $[0, T]$ where T is the expiration time of the option and we want to determine the value of the option at time $t = 0$. Hence we have that Ω is the set of all possible realizations of our stochastic economy in $[0, T]$. $\mathcal{F}_t$, $0 \leq t \leq T$ is the sigma algebra of distinguishable events at time t and $\mathcal{P}$ is the risk-neutral probability measure on $\mathcal{F}$. It is assumed that the filtration $\mathcal{F}_t$ is generated by a d-dimensional Brownian motion, and the state of our stochastic economy is represented as an $\mathcal{F}_t$-adapted Markov process $S_t = (S_t^1, ..., S_t^d) \in \mathbb{R}^d$. We discretize our interval, so that $t \in [t_0 = 0, t_1, ..., t_k = T]$. We define $h(t, S_t)$ to be the payoff of the option at time t, that is non-negative and adapted to our filtration. We can define our problem as the computation of:

$$V_0 = \max_{\tau} \mathbb{E}\left[D_\tau h(\tau, S_\tau)\right]$$

Here $D_s = \exp(-\int_0^s r_u du)$ is the discount factor from time $t = 0$ to time $t = s$ and τ is a stopping time taking values in $[t_0 = 0, t_1, ..., t_k = T]$.

Note: For functions of the form $f(t)$ or $f(t, .)$ the dependence of t will be regularly be noted in a subscript i.e. as f_t or $f_t(.)$ for notational convenience. Both notations are used interchangeably.

At time T we know that $V(T, x) = h(T, x)$, that is the value of the option equals the products payoff. Now for all $t_i < T$ we can define the continuation value, $Q_t(S_t)$, i.e. the expected future payoff as:

$$Q(t_i, S_{t_i} = x) = \frac{D_{t_{i+1}}}{D_{t_i}} \mathbb{E}\left[V(t_{i+1}, S_{t_{i+1}}) | S_{t_i} = x\right]$$

Then the option value at time t_i is given by:

$$V(t_i, S_{t_i}) = \max\{h(t_i, S_{t_i}), Q(t_i, S_{t_i})\}$$

3.2 Algorithm

In this section the algorithm followed in the SGM is presented. To simplify the form of the solution we rewrite our payoff function in the following form:

$$h(t, S_t) = \max\{g(S_t) + X, 0\}$$

where we have that $g : [0, T] \times \mathbb{R}^d \to \mathbb{R}$, hence g maps the high-dimensional S_t-space to the one-dimensional $g(S_t)$ space. The stochastic grid method follows the following scheme:

1. Starting from $S_{t_0} = S_0$, generate n sample paths $(S_{t_0}, ..., S_{t_k})$ using a discretization scheme of choice, for instance the Euler scheme.

2. Compute the option value for each grid point at expiration from:

$$V(T, S_T) = h(T, S_T) = \max\{g(S_T) + X, 0\}$$

3. Compute the approximate functional form, $Z(T, S_T)$, by regressing the option value at the gridpoints over M polynomial basis functions $\{\psi_m\}_{m=1}^{M}$ of $g(S_T)$, hence by determining $a = (a_1, ..., a_M)$ in order to minimize: $r = \min_a |\hat{Z}(T, g(S_T|S_0)) - V(T, S_T)|^2$, where

$$\hat{Z}(T, g(S_T|S_0)) = \mathbb{E}\left[\hat{V}(T, S_T)|g(S_T)\right] = \sum_{m=0}^{M} a_m \psi_m(g(S_T))$$

is the approximate functional form.

4. These steps have to be carried out for each time t_i moving backwards in time starting from t_{k-1} in order to obtain the direct estimator for $V(t_0, S_0)$.

   (a) Compute the continuation value for each of the grid points at time t_i using the functional approximation $\hat{Z}(t_{i+1}, g(S_{t_{i+1}}))$ as computed in the previous timestep:

   $$\hat{Q}(t_i, S_{t_i}) = \frac{D_{t_{i+1}}}{D_{t_i}} \mathbb{E}\left[\hat{Z}(t_{i+1}, g(S_{t_{i+1}}))|S_{t_i}\right]$$

   In order to perform this computation we need to determine the conditional riskneutral probability density function $\mathbb{P}(g(S_{t_{i+1}})|S_{t_i} = x)$. A schematic image of how this will be done is given in figure 3.1. There are three options:

   i. The probability density function is known, then we can just compute the continuation value.

   ii. The probability density function is unknown, but we know the first four non central moments of the distribution $(\mu_1, \mu_2, \mu_3, \mu_4)$. In this case we can use the Gram Charlier series as described in [12] to approximate the conditional riskneutral density function:

   $$\hat{f}(x|y) = \frac{1}{\sqrt{2\pi\mu_2}} \exp\left[-\frac{(x-\mu_1)^2}{2\mu_2}\right] \left[1 + \frac{\mu_3}{3!\mu_2^{3/2}} H_3\left(\frac{x-\mu_1}{\sqrt{\mu_2}}\right) + \frac{\kappa_4}{4!\mu_2^2} H_4\left(\frac{x-\mu_1}{\sqrt{\mu_2}}\right)\right]$$

   where: $H_3(x) = x^3 - 3x$ and $H_4(x) = x^4 - 6x^2 + 3$ are Hermite polynomials and $\kappa_4 = \mu_4 - 3\mu_2^2$. Hence, we have that $(\mu_1, \mu_2, \mu_3, \kappa_4)$ are the first four cumulants.

        iii. The probability density function and the moments of the distribution are unknown. Now we can compute the first four moments by subsimulation, and then the Gram Charlier series can again be used to approximate the conditional density function.

   (b) Compute the option value for the gridpoints at time t_i as:

$$V(t_i, S_{t_i}) = \max\{g(S_{t_i}) + X, \hat{Q}(t_i, S_{t_i})\}$$

   (c) Compute the functional approximation for the conditional expectation as described in step 3 above by regressing again over a set of M polynomial basis functions of $g(S_{t_i})$.

   (d) Now go to the previous timestep and repeat steps 4(a) to 4(c). When the computations are completed for t_0 we have obtained the direct estimator for the option value.

5. From the computations of the direct estimator of the option value we have also obtained the optimal exercise policy, so for each path we can determine the earliest time to exercise i.e. the earliest time where the continuation value is lower than the value of exercising, so define for each asset path $S_t(k)$, $k = 1, ..., n$:

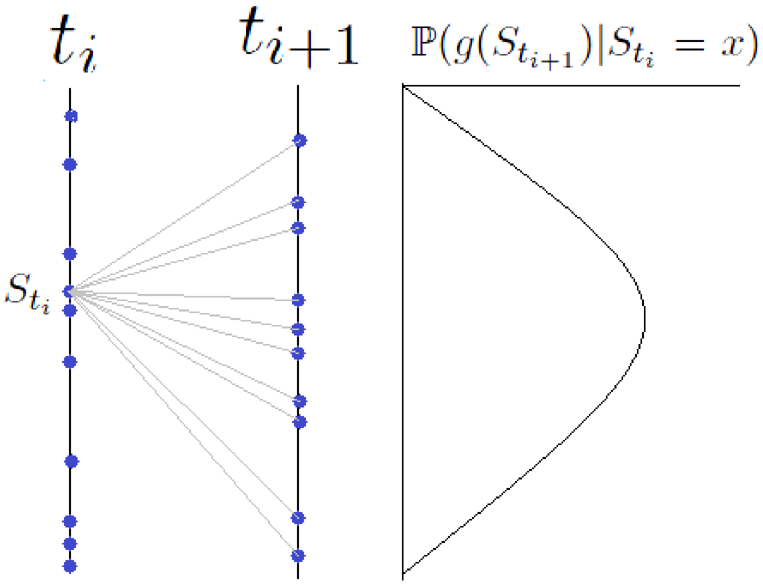
$$\tilde{\tau}(k) = \min\{t \in [0, T] : Q_t(\hat{S_t}(k)) \leq h_t(S_t(k))\}$$

   Now a lower bound on the option value can be determined by computing:

$$\mathbb{E}_0 \left[D_{\tilde{\tau}} h_{\tilde{\tau}}\right] \tag{3.1}$$

   We can compute the expectation in equation (3.1) in two different ways, one is to use the same set of asset paths used for computing the direct estimator for the option price the other is by generating a new set of paths and use the exercise policy for this set of paths. The second method will give a better estimate for a lower bound value, however this will increase computational effort significantly. As we will be most interested in the direct estimator for the option price and we will mostly use the lower bound estimator to see if the method produces consistent results we will in this thesis use the same set of paths for constructing a lower bound on the option value.

Figure 3.1: Schematic image of approximating the transition density



This completes the description of the SGM algorithm in its most simple form. Better results can be obtained using local regression in each timestep instead of global regression as described above and by bundling points at t_i and then using only the gridpoints in step t_{i+1} that are associated with the bundle. The development and implementation of such a technique will be the basis for answering the first research question of this thesis.

3.3 Comparison to Longstaff-Schwartz method

An alternative way of pricing American options using the Monte Carlo approach is the algorithm as described by Longstaff and Schwartz [13]. The Longstaff-Schwartz method (LSM) first computes an optimal exercise policy for a set of simulated paths and then finds the expected value of the discounted payoff if this strategy is followed using a new set of paths to avoid a strong bias in the outcome. The option price that is obtained using LSM is a lower bound for the actual option price as the determined exercise policy is either inferior or equal to the optimal exercise policy. As with the SGM, the LSM method uses regression on a set of basis functions to approximate a functional form of the payoff at each timestep. The big difference is that in the LSM the dimensionality isn't reduced before the regression is performed.

So although the LSM is computationally fast in low-dimensional cases and is easy to implement (much easier than the SGM), the computation time grows fast as the dimensionality of the problem at hand increases. The LSM requires a large number of paths to require accurate results and the number of basis functions required for the regression grows almost exponentially with the dimensionality. The SGM on the other hand requires far fewer paths to obtain a good exercise policy and the number of basis functions required for the regression is independent of the dimensionality of the problem. The only problem is that the subsimulation used to acquire the moments of the transition density function can make the algorithm computationally expensive, this problem only occurs in the SGM and not in the LSM because the LSM doesn't reduce the dimensionality of the problem in the regression step.

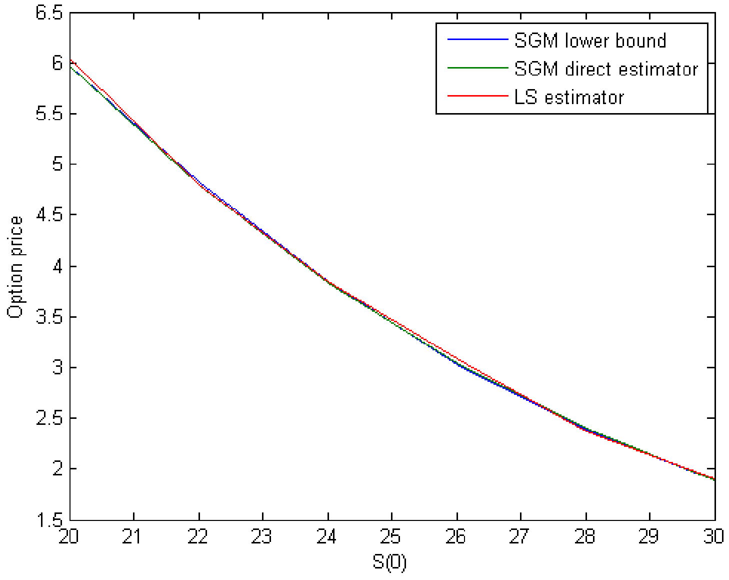
3.4 A simulation experiment with different initial stock prices

In order to get some more intuition for the methods described above we perform a simulation experiment. We will compute the option value of a one-dimensional American put option for different values of the stock price at time $t = 0$ with both the SGM method (which gives us both a lower bound and a direct estimator) and with the LSM method to be able to compare both the results and the computation time required. The Black-Scholes dynamics are used to model the asset price. We will use the following parameters:

- Initial stock prices $S_0 = [20, 22, 24, 26, 28, 30]$
- Strike $X = 25$
- Expiration $T = 1$
- Interest rate $r = 0.05$
- Volatility $\sigma = 0.4$
- Number of time steps $N = 50$
- Number of replications $n = 10^4$
- The option may be exercised at each timestep.

Figure 3.2 shows the computed option prices.

Figure 3.2: Option prices of American put



As we can see in the figure 3.2, the option prices computed using the different methods are nearly identical for all starting values S_0. We can see this similarity even better when we look at the results in table 3.1.

Table 3.1: Option prices of American put

$\mathbf{S_0}$	**SGM lower bound**	**SGM direct estimator**	**LSM**
20	6.0275	5.9780	5.9603
22	4.7971	4.7993	4.7913
24	3.8305	3.8290	3.8407
26	3.0105	3.0397	3.0611
28	2.3973	2.4032	2.4081
30	1.8869	1.8945	1.8855

In the table it clearly shows that the option prices computed using the different methods never differ more than a few cents. Hence at least in this simple case the the SGM method and the LSM method give very similar results. However to get a thorough understanding of the accuracy of both methods we will also need to take a look at the standard deviation of the results for each method. In order to do this another simulation experiment will be performed in section 3.5.

Now we take a look at the running times of the simulations using the two methods:

Figure 3.3: Computation time of option prices of American put

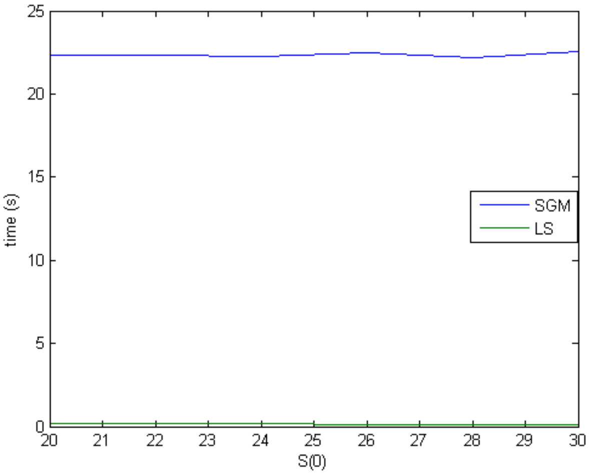


Figure 3.3, with the computation time of the different methods, shows a signigicant difference between the two methods. The LSM method is computationally much faster than the SGM method in such a low-dimensional case, when we take the same number of paths and timesteps for both methods. Where each option price computation with SGM takes over 20 seconds, an option price computation with LSM takes less than 1 second. So the advantages in computation time of the SGM might show in higher-dimensional cases, or perhaps with a more involved model for the stock price. Further more the SGM might provide us with significantly more accurate results, so that for a given level of accuracy we might need fewer paths in SGM than we need in LSM, leading to a decrease in computation time of SGM compared to LSM.

3.5 Simulation experiment to determine the accuracy

We will now perform a simulation experiment to compare the accuracy of the SGM and the LSM by means of the standard deviation of the produced results. In order to do this we will perform a series of 30 simulations using $S_0 = 20$ and with all other parameters equal to those chosen in section 3.4. From the results of these simulations we will look at the average time per simulation, the mean of the results of the simulations and the standard deviation of the results. The computation time per simulation gives roughly the same results as in the first simulation experiment. The SGM simulations take 21.6411 seconds computation time per simulation on average and the LSM simulations 0.2039 seconds. The mean and standard deviations from simulation results are given in table 3.2.

Table 3.2: Simulation results American put

	Mean	**St. Dev.**
SGM lower bound	5.9792	0.0237
SGM direct estimator	5.9779	5.3933e-5
LSM	5.9842	0.0218

In the table we can see that the mean of the results of both methods are very close together, the mean of the lower bound and the direct estimator of SGM are very close together and the mean of the LSM estimator is slightly higher but not significantly. When we take a look at the standard deviations of the results we see that the standard deviation of the lower bound estimator of SGM and the LSM estimator are very close, the standard deviation of the LSM estimator is even slightly smaller than the one of the lower bound estimate of the SGM. But the standard deviation of the direct estimator of the SGM is much smaller than that of the other two, the difference is an astonishing factor 400. Since we are in general more interested in the direct estimator than in the lower bound estimator we are most interested in the size of the standard deviation of the direct estimator in comparison with the size of the standard deviation from the LSM estimator. In order to get the same accuracy with LSM we would need to use many more paths per simulation, which would in turn increase the computation time per simulation significantly.For instance one simulation with 10^6 replications with the LSM takes over 30 seconds already, and this would probably not even be sufficient to make the standard deviation as small as the standard deviation of the direct estimator of the SGM. So now we can conclude that when we wish to know the direct estimator of the option price that the SGM provides us with more accurate results than the LSM.

3.6 Bundling algorithm

In order to make the computations more efficient and less time consuming we will be looking into the possibility of somehow bundling the simulations at each recursion step and then to perform certain computations such as determining the probability density function only once for all paths in this bundle. In order to do this we first take a look at the bundling algorithm as described by Tilley in [16]. The general idea of the algorithm is that at each timestep the paths are ordered by stock price, then partitioned into bundles. Using these bundles the option's intrinsic value and continuation value for each of the paths are computed. Then there is a choice to either determine a sharp exercise boundary or let a transition zone exist. A sharp exercise boundary means that we determine at each timestep a certain stock value so that for all paths where S_t is below this value the option is exercised. Because it has been shown in [16] that we get much better results when a sharp exercise boundary is determined we will only describe the version of the algorithm with the sharp exercise boundary. In addition the algorithm without a sharp exercise boundary is identical, except some steps are left out. After the determination of the exercise boundary we can see what the option value is for each of the paths. Now we will give a more detailed description of the algorithm. At the final timestep we can compute the option value as we are used to at expiration, then at all previous timesteps going backwards in time we do the following:

1. Reorder the stock paths by stock price at time t, from highest to lowest price for a put option and the other way around for a call option. Also re-index the paths according to the reordering.

2. For each of the paths we compute the intrinsic value $I(k, t)$, where $k = 1, ..., n$ is the index of the path.

3. Partition the set of n paths into Q bundles each containing P paths, hence the first bundle contains the first P paths, the second contains path $P + 1$ to $2P$ and so on. Here we choose P and Q to be integer factors of n.

4. For each path k we compute the continuation value of the option as the expectation of the option value taken over all paths in the bundle j which contains path k, so we take:

   $$C(k,t) = d(k,t)\frac{1}{P}\sum_{i \in j} V(i, t+1)$$

   Where $d(k, t)$ is the discount factor for path k at from time t tot time $t + 1$.

5. For each of the paths, determine whether exercising or continuing is more advantageous by defining:

   $$x(k,t) = \begin{cases} 1 & \text{if} \quad I(k,t) > C(k,t) \\ 0 & \text{if} \quad I(k,t) \leq C(k,t) \end{cases}$$

6. Now we are going to determine the sharp exercise boundary by examining the sequence $\{x(k, t); k = 1, ..., n\}$ and then finding the first

string of ones, of which the length exceeds the length of every subsequent string of zeros. The start of this string of ones is the sharp exercise boundary, define its index to be $k^*(t)$.

7. Define the new exercise or hold variable that will be used in the computations of the option value to be:

$$y(k,t) = \begin{cases} 1 & \text{for} \quad k \geq k^*(t) \\ 0 & \text{for} \quad k < k^*(t) \end{cases}$$

8. For each of the paths we can now determine the option value at time t as:

$$V(k,t) = \begin{cases} I(k,t) & \text{if} \quad y(k,t) = 1 \\ C(k,t) & \text{if} \quad y(k,t) = 0 \end{cases}$$

When we have completed this steps for all timesteps we can determine the indicator variable $z(k,t)$ which equals 1 if and only if the option is exercised at that path at that time, and zero otherwise so we have:

$$z(k,t) = \begin{cases} 1 & \text{if} \quad y(k,t) = 1 \text{and} y(k,s) = 0 \forall s < t \\ 0 & \quad \text{otherwise} \end{cases}$$

So now finally the option value can be estimated as:

$$V(0) = \frac{1}{n} \sum_{k=1}^{n} \left(\sum_{t} z(k,t) D(k,t) I(k,t) \right)$$

Here $D(k,t)$ is the discount factor for path k, from time 0 to time t this means that:

$$D(k,t) = \sum_{j=0}^{t} d(k,j)$$

Chapter 4

Subsimulation

In this chapter we will consider the possibility that the analytic moments of the transition density are unknown. So this means that we will need to find some clever way of approximating the transition density. It will turn out that this in fact comes down to finding an approximation of the first four moments of the distribution. So here we propose using so-called subsimulations to approximate this moments. This will entail simulating additional paths from each datapoint to the next timestep in order to approximate the moments using these generated subpaths. After describing this procedure some simulation experiments will be performed to assess computation time and accuracy if this method is used, then we propose using (part of) the bundling algorithm above to speed up computations an finally we will test this possibility with some simulation experiments.

If we now assume that we don't know the moments of our riskneutral transition density i.e. $\mathbb{P}(g(S_{t_{i+1}})|S_{t_i})$, we need the first four cumulants of this stochastic variable to approximate the probability density function using the Gram-Charlier series as we know from section 3.2. Often we don't know these cumulants either, and hence we need a way to approximate the first four cumulants of the distribution. We can do this by means of subsimulations. When we simulate paths starting at S_{t_i} only determining stock values at t_{i+1}, and then using these simulated values of $S_{t_{i+1}}$ we can approximate the first four cumulants. Because if we take $Z = \log S_{t_{i+1}}$ we have that:

$$\mu_1 = \mathbb{E}[Z]$$
$$\mu_2 = \mathbb{E}[Z^2] - \mathbb{E}[Z]^2$$
$$\mu_3 = \mathbb{E}[Z^3] - 3\mathbb{E}[Z^2]\mathbb{E}[Z] + 2\mathbb{E}[Z]^3$$
$$k_4 = \mathbb{E}[Z^4] - 4\mathbb{E}[Z^3]\mathbb{E}[Z] - 3\mathbb{E}[Z^2]^2 + 12\mathbb{E}[Z]^2\mathbb{E}[Z^2] - 6\mathbb{E}[Z]^4$$

Note that if we have simulated values of Z that then these values can indeed be very easily approximated.

So at any given time t_i, $0 \leq i \leq N-1$, where N denotes the number of timesteps, we estimate the moments for each of the n asset paths, by simulating subpaths until the next timestep and from these values determining the first four cumulants μ_1, μ_2, μ_3 and k_4 using the formulas given above. Then we can further follow the regular procedure of SGM using these cumulants and repeating the subsimulations at all places where we need different cumulants.

4.1 Simulation experiment for running time

As we need to perform many subsimulations if we don't know the moments analytically it seems obvious that it will be a very time consuming process. In order to determine how much time it will cost for different numbers of replications per subsimulation we will perform a simulation experiment. To make a good comparison with the situation in which we do not perform subsimulations we have here implemented the subsimulations for a one-dimensional American put with the Black-Scholes dynamics for the stock price. We will choose the following parameters:

- $S_0 = 10$
- $X = 12$
- $r = 0.05$
- $\sigma = 0.03$
- $T = 1$
- $N = 50$
- $n = 1000$

We will perform runs with 50 up to 1000 replications per subsimulation, with steps of 50. We will also perform the same simulation using the analytical moments and compare the running time and make statements about the different outcomes of the simulations. All different scenario's are calculated 30 times to be able to determine both mean and standard deviations of the results. The results are given in table 4.1, where the first line is the reference value where the analytical moments are used.

Table 4.1: Simulation results SGM with and without subsimulation 1d American put

Repl.	Running time	Lower B.	Std. Dev.	Dir. Est.	Std. Dev.
0	3.2527	2.2603	0.0373	2.2600	0.0053
50	25.6507	2.2665	0.0290	2.2878	0.0049
100	28.5074	2.2700	0.0268	2.2831	0.0050
150	31.6747	2.2639	0.0285	2.2811	0.0028
200	34.6397	2.2607	0.0255	2.2780	0.0026
250	37.9028	2.2652	0.0234	2.2755	0.0037
300	40.9763	2.2626	0.0304	2.2738	0.0037
350	44.2485	2.2602	0.0290	2.2726	0.0036
400	47.3709	2.2663	0.0235	2.2716	0.0039
450	50.5670	2.2665	0.0283	2.2697	0.0049
500	53.7458	2.2564	0.0259	2.2689	0.0057
550	56.8564	2.2645	0.0247	2.2642	0.0106
600	60.0996	2.2604	0.0251	2.2664	0.0042
650	63.2360	2.2585	0.0281	2.2658	0.0069
700	66.3818	2.2729	0.0284	2.2660	0.0062
750	69.6409	2.2646	0.0295	2.2659	0.0087
800	72.7427	2.2615	0.0328	2.2653	0.0049
850	76.1779	2.2667	0.0257	2.2631	0.0064
900	79.0965	2.2594	0.0305	2.2636	0.0063
950	82.5519	2.2706	0.0276	2.2636	0.0037
1000	85.5739	2.2696	0.0270	2.2627	0.0041

We see that as expected the subsimulations have a significant effect on the computation time. Just 50 replications per subsimulation means that the computation time grows by a factor 8. Further the computation time seems to grow almost linearly with the number of replications. As we approach a thousand replications per subsimulation the computation time per simulation even exceeds 80 seconds. What stands out is that from approximately 200 replications it seems that the results for the lower boundary do not really improve if we increase the number of replications. The direct estimator on the other hand seems to converge rather nicely to our reference value as we increase the number of replications per subsimulation. However from 600 replications per subsimulation onwards it seems that there is no more significant improvement in the estimated value of the direct estimator. Hence for our level of accuracy of SGM it seems suitable to use 600 replications per subsimulation. However this makes computations expensive, so sometimes it might be advantageous to lose some accuracy in favor of computational benefits by performing fewer replications per subsimulation.

It is very clear from this experiment that in more complicated cases where we have to resort to subsimulations it is very important to make sure we have to perform as few subsimulations as possible. There are of course two aspects to minimizing the time used for subsimulations, namely reducing the number of times we have to perform subsimulations and reducing the number of replications per subsimulation. In order to reduce the number of times a subsimulation has to be performed the bundling algorithm (see section 3.6) may provide a solution. In order to see how much the accuracy of the results of the simulation is affected by the number of replications per subsimulation and hence to be able to make an analysis of the number of replications per subsimulation we need another simulation experiment to be performed.

4.2 Reducing the number of subsimulations

A way to reduce the computational effort required for SGM with subsimulations is reducing the number of times that subsimulation is required. In order to do this bundling will be used. Hence we will no longer perform a subsimulation to determine the first four cumulants for each asset path at each timestep. Instead we first sort our simulated paths by the stock price at the current time step. Then we bundle the stock prices into P bundles each containing Q paths (so that $PQ = n$). For each of these bundles we take:

$$S_\mu = \frac{1}{Q}\sum_{i=1}^{Q} S_t(i)$$

where the $S_t(i)$ are the stock prices at the current time step sorted by size. Using S_μ we perform a subsimulation to approximate the first four cumulants of the transition density function of all paths contained in the bundle, thereby reducing the number of times a subsimulation has to be performed substantially by making a fine choice for P and Q.

4.2.1 Matlab implementation

Bundling is in fact a rather simple adjustment to the program used for SGM with subsimulations. At each timestep we sort the stock paths by stock price at that time, then we divide the paths into the P bundles and perform one subsimulation for each bundle in order to make the vectors containing the first four cumulants for all stock paths. This last part is repeated at the other places where we need the cumulants and hence where subsimulations have to be performed. At time $t = 0$, we only have to perform subsimulation once, for S_0 so there there is no need for bundling.

4.2.2 Simulation experiment

A simulation experiment is performed using the same parameters as in section 4.1. We will perform 600 replications per subsimulation, because this seemed to give very accurate results in previous experiments. Since we have 1000 stock paths, we choose to perform subsimulations with 10, 20, 25, 40, 50 and 100 bundles to see what the results are and also what the computation time is. As a reference the values and computation time of simulation without subsimulations and with subsimulations but without bundling are added to table 4.2.

Table 4.2: Simulation results SGM with and without bundling with subsimulations for American put

	Running time	**Lower B.**	**Std. Dev.**	**Dir. Est.**	**Std. Dev.**
No subsim.	3.2527	2.2603	0.0373	2.2600	0.0053
No bundling	60.0996	2.2604	0.0251	2.2664	0.0042
10 bundles	5.9799	2.2879	0.0349	2.2560	0.0230
20 bundles	5.9876	2.4791	0.0497	2.2907	0.0525
25 bundles	6.2213	2.5042	0.0576	2.3097	0.0555
40 bundles	7.0741	2.5597	0.0484	2.3134	0.0510
50 bundles	7.6892	2.5616	0.0453	2.2946	0.0523
100 bundles	10.7341	2.5604	0.0422	2.2983	0.0278

We see in table 4.2 that reducing the number of times we have to perform subsimulations has the a significant effect on the computation time. Where the SGM with subsimulations without bundling takes 20 times longer than SGM with analytic moments, the SGM with bundled subsimulations only takes about 2 to 3 times longer. There however seems to arise a problem with the accuracy, as the estimations for the lower bound are all much higher than when we use analytical moments and than with the ordinary subsimulations. The direct estimator also turns out rather high (with the exception of the first result with 10 bundles).

Chapter 5

Improved bundling algorithm

In this chapter we will refine the bundling strategies used so far in order to come to a more sophisticated bundling algorithm which gives us better results, and also gives us more computational advantages then the bundling strategy used so far. In addition this new algorithm will provide us with a new framework so that we can avoid subsimulations in most cases, which will greatly improve the computational performance of our method. In addition we will provide a mathematical proof that using the bundling algorithm we construct here the results will in fact converge asymptotically to the true option value. We will then conclude this chapter with simulation experiments of the new improved algorithm.

As we have seen the bundling strategy followed in section 4.2 doesn't give us results as accurate as we would like. Hence we will develop a new bundling strategy in order to attain more accurate results. To do this consider the set of generated asset paths $S_1(k), S_2(k), ..., S_N(k)$, for $k = 1, ..., n$, where we take N timesteps i.e. $S_N = S_T$ for all asset paths. Now at time i, with $1 \leq i \leq N$ we will divide the n asset paths into 2^b bundles via a bisection strategy. We first divide the results into two subgroups where the first subgroup contains all asset paths with $S_i(k) < \frac{1}{n}\sum_{j=1}^{n} S_i(j)$ and the second subgroup contains the rest of the asset paths. For each of this subgroups we repeat this strategy a total of b times, this way we will attain 2^b bundles where, by nature of the distribution the bundles in the middle will contain more points than the ones at the boundaries. We will give a simple example to illustrate the procedure described above.

Suppose at a certain timestep we have the following vector of asset prices:

$$S = [20, 25, 16, 23, 22, 19, 17, 26]$$

We compute the mean of S:

$$\frac{1}{8}\sum_{k=1}^{8} S_k = \frac{1}{8}(20 + 25 + 16 + 23 + 22 + 19 + 17 + 26) = 21$$

Let $b = 2$, so in total we want to divide S into $2^2 = 4$ bundles. In the first step the vector S will be split into:

$$S1 = [20, 16, 19, 17] \quad S2 = [25, 23, 22, 26]$$

We repeat the procedure for both $S1$ and $S2$. The vector $S1$ has mean 18, so this will be split into:

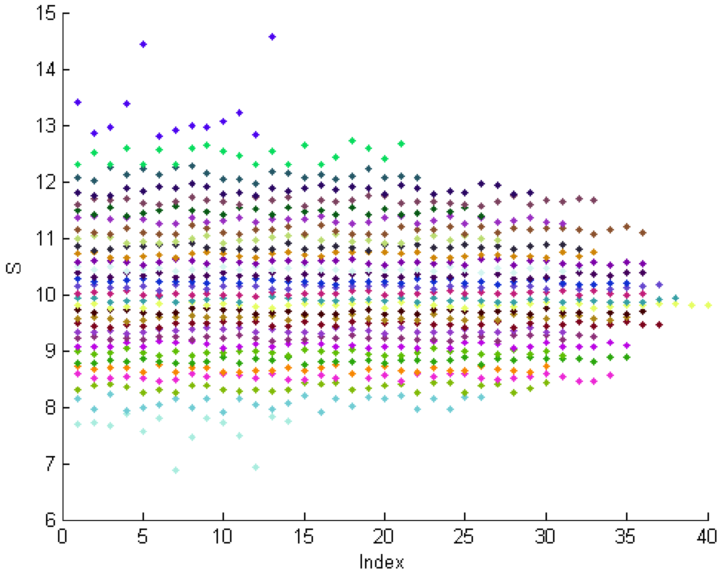
$$S11 = [16, 17] \quad S12 = [20, 19]$$

The mean of $S2$ equals 24 so finally we will get:

$$S21 = [22, 23] \quad S22 = [25, 26]$$

Figure 5.1 gives an example of the bundled asset prices, each color represents one bundle for $b = 5$ so that we have a total of $2^5 = 32$ bundles and $n = 1000$. We see in the figure that indeed the bundles in the center of the distribution contain more paths than the bundles at the boundaries.

Figure 5.1: Asset price bundles, $b = 5$ and $n = 1000$



The way the bundles are defined is not the only change we make in the bundling algorithm described in section 3.6, we will further reduce the computational effort of the algorithm by using only the paths within a bundle in the regression step. So first we used the bundles only to determine the moments of the transition density function and then the method proceeds the same as in the case with the analytical moments, hence by performing local regression on predefined intervals independent of the bundles. But now we will use the bundles further in the regression step of SGM to make the algorithm more efficient.

5.1 Approximating the continuation value

In order to speed up computations Jin, Tan and Sun in [11] suggest to approximate continuation values per bundle. Hence instead of defining the continuation value for each path separately they only approximate the continuation value once per bundle, taking the center of the bundle as the point of origin from which they approximate the continuation value. If $a(n)$ represents the number of bundles and n the total number of paths this means that if $c_{n,i}$ is the center of bundle i, $1 \leq i \leq a(n)$, for all paths in bundle i they approximate the continuation value as:

$$\hat{Q}_t(c_{n,i}) = \frac{D_t}{D_{t+1}} \mathbb{E}\left[\hat{V}_{t+1}(S_{t+1}) | \text{the paths that lie in bundle } i\right]$$

We will deviate from this strategy, since when using that strategy the quality of the results depends on how many bundles are used, and we would in general need many bundles and also simulated asset paths to obtain good results. To avoid this we will not use this strategy for approximating the continuation value, but instead we will use a technique based on iterated conditioning. Consider therefore some set of basis-functions $\Psi_1, ..., \Psi_m$, for instance we will use $\Psi_i = S_t^i$ for $i = 1, ..., 4$. We can rewrite the continuation value at a certain timestep $t-1$, for path k, $1 \leq k \leq n$ as:

$$\begin{aligned} Q_{t-1}(S_{t-1}(k)) &= d(t-1,k)\mathbb{E}\left[V_t(S_t)|S_{t-1}(k)\right] \\ &= d(t-1,k)\mathbb{E}\left[\mathbb{E}\left[V_t(S_t)|\Psi_1(S_t), ..., \Psi_m(S_t), S_{t-1}(k)\right]|S_{t-1}(k)\right] \\ &= d(t-1,k)\mathbb{E}\left[Z(S_t, S_{t-1})|S_{t-1}(k)\right] \qquad (5.1) \end{aligned}$$

We can in fact use the law of iterated conditioning here since it clearly holds that $\sigma(S_{t-1}(k)) \subseteq \sigma(\Psi_1(S_t), ..., \Psi_m(S_t), S_{t-1}(k))$. We first have to compute the inner expectation, followed by the outer expectation. As described by Jain and Oosterlee in [10] we can simply approximate the inner expectation by a least squares regression on the basis functions, hence for each path j in bundle i, we can define:

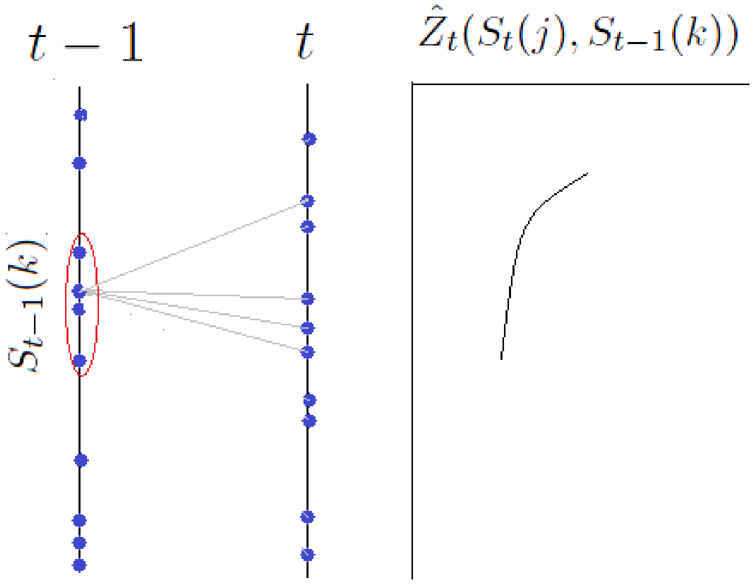
$$\hat{Z}_t(S_t(j), S_{t-1}(k)) = \sum_{l=1}^{m} q_l \Psi_l(S_t(j)),$$

where we choose the parameters q_l, $l = 1, ..., m$ such that we minimize the sum of the squared residuals:

$$r = \min_{q_l \in \mathbb{R}} \sum_{j \in \text{bundle } i} \left| \hat{Z}_t(S_t(j), S_{t-1}(k)) - V_t(S_t(j)) \right|^2$$

A schematic image of the resulting approximation of the inner expectation is found in figure 5.2.

Figure 5.2: Schematic image of determining the inner expectation



Then we can approximate our continuation value as:

$$\begin{aligned}\hat{Q}(S_{t-1}(k)) &= d(t-1,k)\mathbb{E}\left[Z(S_t, S_{t-1})||S_{t-1}(k)\right] \\ &= d(t-1,k)\sum_{l=1}^{m} q_l \mathbb{E}\left[\Psi_l(S_t)|S_{t-1}(k)\right] \end{aligned} \tag{5.2}$$

This has in fact an added benefit, since usually the expectation in equation (5.2) is known in closed form which means we can avoid subsimulations altogether.

5.2 Approximating moments using bundles

As pointed out above, by using the least squares regression to approximate the inner expectation in equation (5.1) we can in almost all cases avoid subsimulations altogether. In the few cases where we can't we might still need some form of subsimulation and hence it will be useful to find a good protocol for using the bundles within the subsimulations.

The most simple way would be at time t_i to take all values in the same bundle at time t_{i+1} and use these as datapoints in the next step from which we can approximate the moments of the distribution as we did with the subsimulations earlier (see chapter 4). This way we will compute the moments only one time for each bundle and we assume that the moments of the datapoints within one bundle are more or less equal and can be approximated by one single computation. In this case it could very well be that the number of replications and the number of bundles are of great influence on the accuracy of our results, and we might need very many paths to get accurate results.

Another possibility will be to adopt a strategy more like the one described by Jin et al. ([11]) where we make one central approximation for the bundles by performing subsimulations for the center of each bundle only. This means that we will need to perform far fewer subsimulations compared to the situation without bundles, and this might save us a considerable amount of time. The only problem is that we then also will have the difficulty that Jin et al. encountered, namely choosing the number of bundles and paths required for optimal results.

5.3 Theoretical proof of the algorithm

The first thing we would like to know is whether the new algorithm (at least asymptotically) will give us correct results. In order to do this we will follow the reasoning given by Jin, Tan and Sun in [11], as far as it will be applicable. In addition we will need to do some additional steps typical for our situation. Assume we discretize the interval $[0, T]$ into $t_0, ..., t_N$ for some $N > 0$, and that we generate n paths. We want to prove that as n tends to infinity, and also the number of bundles $a(n)$ tends to infinity, we will eventually have that $\lim_{n\to\infty} \mathbb{E}[|\hat{V}_{n,0} - V_0(S_0)|] = 0$. In order to do this we will first need some auxiliary results, and we will also have to make some assumptions.

For every timestep t_i, let for al x in the state space of the stock price $\mathcal{S}_i$, $c_{n,i}(x)$ denote the center of the bundle to which x belongs. In addition, for all bundles define $\tilde{S}_{t_i}(j)$, $j = 1, ..., a(n)$, to be the mean stock value in bundle j. We will need the following assumptions:

ASSUMPTION 1: For $i = 1, ..., N - 1$, $\{\tilde{S}_{t_i}, j = 1, ..., a(n)\}$ is a dense set in the state space $\mathcal{S}_i$.

ASSUMPTION 2: For $i = 1, ..., N-1$, given $y \in \mathbb{R}$, the function $F_i(y|x)$ is continuous with respect to $x \in \mathbb{R}^d$, where $F_i(y|x) = \mathbb{P}\left[V_{i+1}(S_{t_{i+1}}) \leq y | S_{t_i} = x\right]$, $i = 0, ..., N-1$.

ASSUMPTION 3: Given $i \in \{1, ..., N\}$, $\mathbb{E}\left[h^2(t_i, S_{t_i})\right] < \infty$ and $\mathbb{E}\left[S_{t_i}^2\right] < \infty$.

ASSUMPTION 4: For all $i = 1, ..., N-1$, $\mathbb{P}\left(Q_i(S_{t_i}) = h(t_i, S_{t_i})\right) = 0$

ASSUMPTION 5: We assume that as the number of bundles and the number of asset paths tend to infinity, we will also have that the minimum number of paths in each bundle will tend to infinity:

$$\lim_{n\to\infty} b(n) = \infty,$$

and also that the minimum number of paths in a bundle will tend to infinity strictly slower than the number of bundles:

$$\lim_{n\to\infty} \frac{a(n)b(n)}{n} = 0$$

We can wonder whether this is an assumption we can make, but by the nature of the distributions of asset paths we will see that we will have fewer paths in the bundles near the tails of the distribution and hence in practical situations this assumption will make sense.

Now we have the assumptions we need to start proving some auxiliary results. First we start by showing that for all $x \in \mathbb{R}_{\geq 0}$ the center of the bundle that x belongs to will converge to x. From this we can then conclude that the approximation we use for the continuation value is asymptotically the same as approximating the continuation value from the center of the bundle for the entire bundle at once.

LEMMA 1: As n and $a(n)$ tend to infinity we have for all $x \in \mathcal{S}_i$, $i = 1, ..., N-1$ that:

$$\lim_{n\to\infty} c_{n,i}(x) = x$$

Proof: Let $\epsilon > 0$, arbitrary. Denote $B_x(\epsilon)$ the sphere centered at x with radius ϵ. We will prove the Lemma 1 by showing that with probability 1 there exists a random $N_x \in \mathbb{N}$ so that for every $n \geq N_x$ we have $c_{n,i}(x) \in B_x(\epsilon)$. We choose N_x^1, and $a(n)$ sufficiently large so that $\tilde{S}_{t_i}(j) \in B_x(\epsilon/4)$, for some $j < a(n)$ and $n \geq N_x^1$. This is possible because of ASS. 1. It follows by the large number theorem that:

$$\lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} \mathbb{1}\{S_{t_i}(k) \in B_x(\epsilon/4)\} = \mathbb{P}(S_{t_i} \in B_x(\epsilon/4))$$

Hence, with probability 1 it will hold for some $\alpha > 0$, that there is a value $N_x^2 \in \mathbb{N}$, so that for all $n \geq N_x^2$:

$$\sum_{k=1}^{n} \mathbb{1}\{S_{t_i}(k) \in B_x(\epsilon/4)\} \geq \alpha n$$

This means that there are at least $\lfloor \alpha n \rfloor$ points in $\{S_{t_i}(j)|j = 1, ..., n\} \cap B_x(\epsilon/4)$. Also by ASS. 5 we have for n sufficiently large, say $n \geq N^3$, that $a(n)b(n) < \lfloor \alpha n \rfloor$. Now for $n \geq N_x = \max\{N_x^1, N_x^2, N^3\}$ we have for each point $z \in \{S_{t_i}(j)|j = 1, ..., n\} \cap B_x(\epsilon/4)$, that there exists some $y \in \{\tilde{S}_{t_i}(j)|j = 1, ..., a(n)\} \cap B_x(\epsilon)$ so that we have that $z \in B_y(\epsilon/2)$. This in turn tells us that we must have that $c_{n,i}(x) \in B_x(\epsilon)$ when $n > N_x$, which concludes the proof.

Indeed it follows that we can continue our proof of the method as if we compute the continuation value of the option at each time for the center of the bundle only. If we do that, we want to show next that the approximated continuation value will asymptotically converge to the actual continuation value. In order to do this we will not only need the result of LEMMA 1, but we will also need an additional result showing us that the diameter of each bundle will converge to 0 as n tends to infinity.

LEMMA 2:Assume that the stock prices are bounded almost surely, hence $\mathbb{P}(|S_{t_i}| \leq M) = 1$ for some $M > 0$, and for all $i \in \{1, ..., N\}$. Then the diameter of each bundle, $\mathcal{A}_{n,i}$, will converge to 0 as $n \to \infty$.

Proof: We will prove this lemma by contradiction. Assume that the diameter of $\mathcal{A}_{n,i}(x)$ does not converge to 0 with probability 1. Then there exists a subset $\Omega_1 \subset \Omega$ with $\mathbb{P}(\Omega_1) > 0$ so that for every $\omega \in \Omega_1$ there exists a subsequence $\{S_{t_i}(k_n)|n = 1, 2, ...\} \subset \mathcal{A}_{n,i}(x)$ which does not converge to x. Now because the stock prices are bounded almost surely we know that given $\omega_0 \in \Omega_1$ there exists a convergent subsequence in $\{S_{t_i}(k_n)|n = 1, 2, ...\}$. Without loss of generality we can assume the convergent subsequence is $\{S_{t_i}(k_n)|n = 1, 2, ...\}$. However by assumption we must have that this sequence converges to some $x' \neq x$. If we define $\epsilon_0 = \parallel x - x' \parallel_2 /4$ then there exists a number $N_1 \in \mathbb{N}$ such that for all $n \geq N_1$ we have that $c_{n,i}(x) \in B_x(\epsilon_0)$ and $S_{t_i}(k_n) \in B_{x'}(\epsilon_0)$. But by LEMMA 1, there is a $N_2 \in \mathbb{N}$ such that for all $n \geq N_2$ it holds that $c_{n,i}(x') \in B_{x'}(\epsilon_0)$. Combining these facts gives us that for $n \geq N_3 = \max\{N_1, N_2\}$ we have that $c_{n,i}(x) \in B_x(\epsilon_0)$ and $c_{n,i}(x') \in B_{x'}(\epsilon_0)$. This would imply that $c_{n,i}(x')$ would be closer to $S_{t_i}(k_n)$ than $c_{n,i}(x)$ for $n \geq N_3$, but by the definition of the bundles this would mean that $S_{t_i}(k_n)$ is not in $\mathcal{A}_{n,i}$ which is a contradiction and hence the proof is complete.

We have now seen in LEMMA 1 and LEMMA 2 that for each $x \in \mathbb{R}_{\geq 0}$ the center of the bundle containing x will converge to x and also that the diameter of each bundle will converge to 0 as n tends to infinity. Whereas Jin et. al in [11] need to perform a lot of work to prove their approximated continuation value will indeed converge to the actual continuation value we won't have to do that. In order to see that, note that we approximate the inner expectation in (5.1) using a least squares regression in the improved bundling algorithm, in order to have an analytic expression for the outer expectation of the continuation value. In cases where we don't have the analytic expression for the last expectation we approximate the expectation via a Riemann sum, in which case convergence is also apparent. Now from LEMMA 1 and LEMMA 2 it then follows immediately that for all $x \in \mathbb{R}_{\geq 0}$, at all timesteps t, $0 \leq t < T$, it holds that:

$$\lim_{n\to\infty} \hat{Q}_{n,t}(x) = Q_t(x)$$

So that we have that result we only need to show that the approximated stopping time will converge to the actual optimal stopping time and then we can finally show that asymptotically the method will converge to the actual option value.

LEMMA 3: The stopping time of the bundling algorithm asymptotically converges to the actual optimal stopping time:

$$\lim_{n\to\infty} \mathbb{P}(\hat{\tau}_n \neq \tau) = 0$$

Proof: We will construct an upper bound on the probability that the approximation of the stopping time will not converge asymptotically to the optimal stopping time and show that this upper bound tends to 0 as $n \to \infty$.

$$\begin{aligned}
\mathbb{P}(\hat{\tau}_n \neq \tau) &= \mathbb{P}(\hat{\tau}_n < \tau) + \mathbb{P}(\hat{\tau}_n > \tau) \\
&= \mathbb{P}\left(\hat{Q}_{n,t_i}(S_{t_i}) \leq h_{t_i}(S_{t_i}) < Q_{t_i}(S_{t_i})\right) + \mathbb{P}\left(\hat{Q}_{n,t_i}(S_{t_i}) > h_{t_i}(S_{t_i}) \geq Q_{t_i}(S_{t_i})\right) \\
&\leq \sum_{i=0}^{N-1} \mathbb{P}\left(\hat{Q}_{n,t_i}(S_{t_i}) \leq h_{t_i}(S_{t_i}) < Q_{t_i}(S_{t_i})\right) \\
&\qquad + \sum_{i=0}^{N-1} \mathbb{P}\left(\hat{Q}_{n,t_i}(S_{t_i}) > h_{t_i}(S_{t_i}) \geq Q_{t_i}(S_{t_i})\right)
\end{aligned}$$

Now we can rewrite ASS. 4 as:

$$\lim_{\gamma\to 0} \mathbb{P}(|h(t,x) - Q_t(x)| \leq \gamma) = 0$$

This means that for every $\epsilon > 0$ there exists a $\gamma > 0$ so that:

$$\mathbb{P}(|h(t,x) - Q_t(x)| \leq \gamma) \leq \epsilon$$

Combining these results yields the following expression:

$$\mathbb{P}(\hat{\tau}_n \neq \tau) \leq N\epsilon + \sum_{i=0}^{N-1} \mathbb{P}\left(|\hat{Q}_{n,i}(S_{t_i}) - Q(S_{t_i})| \geq \gamma\right)$$

Now Broadie and Glasserman have proven in [3] for this last statement that:

$$\limsup_{n\to\infty} \mathbb{P}(\hat{\tau}_n \neq \tau) \leq \epsilon,$$

which then gives us the concluding result that:

$$\lim_{n\to\infty} \mathbb{P}(\hat{\tau}_n \neq \tau) = 0$$

We now have all we need to prove our initial assertion, that we have that $\lim_{n\to\infty} \mathbb{E}[|\hat{V}_{n,0} - V_0(S_0)|] = 0$. Note that we can write:

$$\begin{aligned}\lim_{n\to\infty} \mathbb{E}[|\hat{V}_{n,0} - V_0(S_0)|] &= \lim_{n\to\infty} \mathbb{E}\left[\left|\hat{V}_{n,0} - \frac{1}{n}\sum_{k=1}^{n} e^{-r\tau} h(\tau, S_\tau(k)) + \frac{1}{n}\sum_{k=1}^{n} e^{-r\tau} h(\tau, S_\tau(k)) - V_0(S_0)\right|\right] \\ &\leq \lim_{n\to\infty} \mathbb{E}\left[\left|\hat{V}_{n,0} - \frac{1}{n}\sum_{k=1}^{n} e^{-r\tau} h(\tau, S_\tau(k))\right|\right] \\ &\quad + \lim_{n\to\infty} \mathbb{E}\left[\left|\frac{1}{n}\sum_{k=1}^{n} e^{-r\tau} h(\tau, S_\tau(k)) - V_0(S_0)\right|\right]\end{aligned} \tag{5.3}$$

It is trivial to see that under ASS. 3, it follows from the large number theorem and dominated convergence theorem that for the first expectation in (5.3) we have that:

$$\lim_{n\to\infty} \mathbb{E}\left[\left|\frac{1}{n}\sum_{k=1}^{n} e^{-r\tau} h(\tau, S_\tau(k)) - V_0(S_0)\right|\right] = 0$$

It remains to show that for the second expectation in (5.3) it holds that:

$$\lim_{n\to\infty} \mathbb{E}\left[\left|\hat{V}_{n,0} - \frac{1}{n}\sum_{k=1}^{n} e^{-r\tau} h(\tau, S_\tau(k))\right|\right] = 0$$

From the definition of our approximation for the option value we can write $\hat{V}_{n,0}$ as:

$$\hat{V}_{n,0} = \frac{1}{n}\sum_{k=1} n e^{-r\hat{\tau}_n} h(\hat{\tau}_n, S_{\hat{\tau}_n}(k)),$$

and it follows that:

$$\begin{aligned}
\mathbb{E}\left[\left|\hat{V}_{n,0} - \right.\right. & \left.\left. \frac{1}{n}\sum_{k=1}^{n} e^{-r\tau} h(\tau, S_\tau(k))\right|\right] \\
&= \mathbb{E}\left[\left|\frac{1}{n}\sum_{k=1} n\left(e^{-r\hat{\tau}_n} h(\hat{\tau}_n, S_{\hat{\tau}_n}(k)) - e^{-r\tau} h(\tau, S_\tau(k))\right) \mathbb{1}\{\hat{\tau}_n \neq \tau\}\right|\right] \\
&\leq \mathbb{E}\left[\left|\sum_{k=1} n\left(e^{-r\hat{\tau}_n} h(\hat{\tau}_n, S_{\hat{\tau}_n}(k)) - e^{-r\tau} h(\tau, S_\tau(k))\right) \mathbb{1}\{\hat{\tau}_n \neq \tau\}\right|\right] \\
&\leq \left(\mathbb{E}\left[\left|\sum_{k=1} n\left(e^{-r\hat{\tau}_n} h(\hat{\tau}_n, S_{\hat{\tau}_n}(k)) - e^{-r\tau} h(\tau, S_\tau(k))\right)\right|^2\right]\right)^{1/2} (\mathbb{P}(\hat{\tau}_n \neq \tau))^{1/2}
\end{aligned}$$

Where the last inequality follows from the Cauchy-Schwarz inequality. From ASS. 3 it follows that we have that:

$$\begin{aligned}
&\max_n \mathbb{E}\left[h^2(\hat{\tau}_n, S_{\hat{\tau}_n})\right] \leq \infty \\
&\mathbb{E}\left[h^2(\tau, S_\tau)\right] \leq \infty
\end{aligned}$$

It follows directly from LEMMA 3, that we have that:

$$\lim_{n\to\infty} \mathbb{E}\left[\left|\hat{V}_{n,0} - \frac{1}{n}\sum_{k=1}^{n} e^{-r\tau} h(\tau, S_\tau(k))\right|\right] = 0,$$

which concludes the proof of the (asymptotic) correctness of the bundling algorithm.

We are ready to start implementing this new algorithm. In order to check the correctness of our new algorithm we will first do a test where we use the Black-Scholes model and the analytical moments.

5.4 Simulation experiment for new bundling algorithm

As we have seen in section 5.3 the algorithm we have now developed will converge asymptotically to the true option value. Now it is time to see if this convergence is fast enough as to give us accurate results in reasonable computation time. We will compare the results to the results of SGM without bundling for accuracy and computation time, and use a variety of combinations of the number of bundles and the number of asset paths. This way we get an idea of what a good combination is of the number of bundles and the number of asset paths at the same time. The parameters that will be equal in all of the simulations are:

- $S_0 = 10$
- $X = 12$
- $r = 0.05$
- $T = 1$
- $\sigma = 0.3$
- $N = 50$, so we take 50 timesteps and at each timestep the option can be exercised.

First we will create our reference results by performing two normal SGM simulations, one with $n = 10^3$, and one with $n = 10^4$ asset paths. For each of the scenario's we do 30 runs so that we can base our further conclusions on the mean and standard deviations of the results. In addition we will determine the computation time per run for each of the scenario's as to be able to make a comparison between the computation time of the SGM without bundling and the SGM with our new bundling algorithm. The results of the simulations to obtain reference values are found in table 5.1.

Table 5.1: Simulation results SGM without bundling for American put

		Mean	**Std. dev.**	**Time/run (s)**
$n = 10^3$	**lower bound**	2.2620	0.0259	3.5672
	direct estimator	2.2608	0.0032	
$n = 10^4$	**lower bound**	2.2653	0.0074	21.5850
	direct estimator	2.2657	0.0002	

Now that we have our reference results we can take a look at the SGM including the new bundling algorithm. As in the case with without bundling we will look at $n = 10^3$ and $n = 10^4$ asset paths. For the number of bundles we will take $b = 2$, $b = 3$, $b = 4$, $b = 5$ for $n = 10^3$ and $n = 10^4$, in addition for $n = 10^4$ we will also take $b = 6$. This means that the number of bundles will range from $2^2 = 4$ to $2^5 = 32$ for $n = 10^3$, and form $2^2 = 4$ to $2^6 = 64$ for $n = 10^4$.

The results for $n = 10^3$ asset paths are found in table 5.2, the results for $n = 10^4$ asset paths can be found in table 5.3

Table 5.2: Simulation results SGM with bundling, $n = 10^3$, for American put

		Mean	**Std. dev.**	**Time/run (s)**
$b = 2$	**lower bound**	2.2605	0.0276	0.2421
	direct estimator	2.2667	0.0013	
$b = 3$	**lower bound**	2.2700	0.0219	0.3623
	direct estimator	2.2661	0.0007	
$b = 4$	**lower bound**	2.2693	0.0256	0.5900
	direct estimator	2.2658	0.0009	
$b = 5$	**lower bound**	1.9068	0.0225	1.0331
	direct estimator	$7.0545 \cdot 10^9$	$3.6221 \cdot 10^{10}$	

Table 5.3: Simulation results SGM with bundling, $n = 10^4$, for American put

		Mean	**Std. dev.**	**Time/run (s)**
$b = 2$	**lower bound**	2.2666	0.0079	1.2165
	direct estimator	2.2659	0.0002	
$b = 3$	**lower bound**	2.2684	0.0072	1.5498
	direct estimator	2.2658	0.0001	
$b = 4$	**lower bound**	2.2679	0.0084	2.0793
	direct estimator	2.2658	0.0001	
$b = 5$	**lower bound**	2.2663	0.0095	3.0464
	direct estimator	2.2658	0.0001	
$b = 6$	**lower bound**	2.2654	0.0074	4.7790
	direct estimator	2.2658	0.0001	

If we first compare the results of table 5.2 to the results in table 5.1 the first thing that stands out is the gain in computation time using the new method, combined with the gain in accuracy. The new SGM with bundling has decreased our computation time by a factor 10. With $n = 10^3$ asset paths in the SGM with bundling we can obtain much more accurate results, with a much smaller standard deviation then we did with the old SGM without the bundling algorithm. A striking result in table 5.2 is the result for $b = 5$, here we clearly see that it is important not to have too many bundles, since this makes the results very unaccurate. This happens because then we get bundles containing very little asset paths, which means that we will perform the regression step based on very little points which can lead to very big deviations in the results. Another striking thing in the result is the relatively small influence of increasing the number of bundles from 4 to 16. The results are a little more accurate, especially the mean value of the direct estimator, but the standard deviation and lower bound estimators don't differ much.

Comparing the results in table 5.3 to the results in table 5.1 gives us more or less the same picture. Again we see that the computation time decreases by a factor 10 compared to the SGM algorithm without bundling. The standard deviation also decreased and just as in the case where $n = 10^3$ we see that increasing the amount of bundles hardly improves the results. Here we don't see any actual improvement in the direct estimator results for $b \geq 3$, the mean estimate of the lower bound seems to be best for $b = 4$.

As we have seen from the results the new method gives us substantially better results than the SGM algorithm without bundling. The convergence to the true option value is also very fast, since we already get very good results using a very limited number of bundles and asset paths. This means that using this new SGM algorithm with bundling we get very accurate results in very limited computation time.

Chapter 6

Black-Scholes Hull-White model

In this chapter we will implement the Black-Scholes Hull-White model in the SGM with bundling algorithm. First we will derive an analytic solution for a European option price under the Black-Scholes Hull-White model, so that we can use this result as a comparison for our numerical results. We will conclude this chapter with simulation experiments for the Black-Scholes Hull-White model.

The Black-Scholes Hull-White model will be implemented within the SGM, so that we can determine the prices of American options using a more realistic model for the underlying. The Black-Scholes Hull-White model is such a model, since not only the stock price has a stochastic component, also the interest rates are stochastic. Hence, we define the state vector $X_t = (S_t, r_t)^T$ with probability space $(\Omega, \mathcal{F}, \mathbb{P})$, so that X_t is adapted to the filtration $\mathcal{F}_t$. Instead of one stochastic differential equation, we deal with the system of stochastic differential equations (6.1).

$$\begin{cases} dS_t = r_t S_t dt + \sigma S_t dW_t^S \\ dr_t = \lambda(\overline{r}_t - r_t)dt + \eta dW_t^r \\ dW_t^S dW_t^r = \rho dt \end{cases} \tag{6.1}$$

As can be seen the stock follows geometric Brownian motion, the interest rate on the other hand follows a mean reverting process, as can be seen in the drift term. Here $\overline{r}_t$ represents the expected value of r_t, λ is the parameter that determines how strong the mean reversion is and η describes the volatility of the interest rate. The third equation determines the dependency between S and r, where ρ is the correlation between W^S and W^r, hence $-1 \leq \rho \leq 1$. If $\rho = 1$ then the Brownian motion generating the stock price and the Brownian motion generating the interest rate are perfectly correlated, and if $\rho = -1$ they are perfectly negatively correlated.

If we use the Black-Scholes Hull-White model an important difference is that our discount factor is more involved than it is for the ordinary Black-Scholes dynamics. We have that the discount factor from time $t = 0$ to time $t = s$ is given by:

$$D(s) = e^{-\int_0^s r(u)du}$$

Hence we have to approximate this by a summation in our implementation.

6.1 Analytic solution

By a transformation technique, described by Brigo and Mercurio in [2], it is possible to acquire an analytic solution for a European option under the Black-Scholes Hull-White dynamics. The key of this technique is to make a measure transformation to the so-called forward measure in order to make the stochastic components under the new measure independent. To simplify the analysis we will only treat the case where $\bar{r}_t = \bar{r}$, since this is the form we will use in our implementation and this will make our computations less involved. Starting from the Black-Scholes Hull-White model as given in the previous section, we can rewrite the dynamics given in equation (6.1) using the Cholesky decomposition to get:

$$\begin{cases} dr_t = \lambda(\bar{r} - r_t)dt + \eta d\widetilde{W}_t \\ dS_t = S_t\left(r_t dt + \sigma\rho d\widetilde{W}_t + \sigma\sqrt{1-\rho^2}d\widetilde{Z}_t\right) \end{cases} \tag{6.2}$$

This means that in our original system we had:

$$\begin{cases} dW_t^r = d\widetilde{W}_t \\ dW_t^S = \rho d\widetilde{W}_t + \sqrt{1-\rho^2}d\widetilde{Z}_t \end{cases}$$

We will rewrite the differential equation for r_t in (6.2), using the fact that we can write $r_t = x_t + \phi_t$ with $r_0 = r(0)$, where the process x_t satisfies the following differential equation:

$$dx_t = -\lambda x_t dt + \eta d\widetilde{W}_t, \quad x_0 = 0.$$

Hence, we can write for $0 \leq t < u \leq T$:

$$x_u = x_t e^{-\lambda(u-t)} + \eta \int_t^u e^{-\lambda(u-s)} d\widetilde{W}_s.$$

The function ϕ_t is a well-defined deterministic function on $[0, T]$ with $\phi_0 = r(0)$. If we assume that the term structure of discount factors is given by the sufficiently smooth function $P(0, t)$ for all $t \in [0, T]$ such that:

$$P(u, v) = \mathbb{E}[\exp\{-\int_u^v r_s ds\}|\mathcal{F}_u] \qquad \text{for} \quad 0 \leq u < v \leq T,$$

then it follows that the instantaneous forward rate implied by the term structure must be given by:

$$f(0,t) = -\frac{\partial(\ln P(0,t))}{\partial t}$$

It follows, in order to fit the observed term structure, that we have for all $t \in [0,T]$:

$$\phi_t = f(0,t) + \frac{\eta^2}{2\lambda^2}\left(1 - e^{-\lambda t}\right)^2$$

To define the new measure we need to know $\int_t^T x_u du$. We will determine this integral; note that at time t, the value x_t is known. By stochastic integration by parts we get:

$$\begin{aligned}
\int_t^T x_u du &= Tx_T - tx_t - \int_t^T u dx_u \\
&= \int_t^T (T-u) dx_u + (T-t)x_t \\
&= -\lambda \int_t^T (T-u)x_u du + \eta \int_t^T (T-u) d\widetilde{W}_u + (T-t)x_t. \qquad (6.3)
\end{aligned}$$

Here, we used in the last step the definition of dx_t. We will work out the first integral in equation (6.3). Using the definition of x_t we can write:

$$-\lambda \int_t^T (T-u)x_u du = -\lambda\left(x_t \int_t^T (T-u)e^{-\lambda(u-t)}du + \eta \int_t^T (T-u)\int_t^u e^{-\lambda(u-s)}d\widetilde{W}_s du\right). \qquad (6.4)$$

The first integral in equation (6.4) can easily be computed as:

$$-\lambda x_t \int_t^T (T-u)e^{-\lambda(u-t)}du = \frac{x_t}{\lambda}\left(1 - e^{-\lambda(T-t)}\right) - x_t(T-t). \qquad (6.5)$$

The computation of the second integral in equation (6.4) is somewhat more involved. We need to again apply integration by parts to get:

$$\begin{aligned}
-\lambda\eta \int_t^T (T-u)\int_t^u e^{-\lambda(u-s)}d\widetilde{W}_s du &= -\lambda\eta \int_t^T \left(\int_t^u e^{\lambda s} d\widetilde{W}_s\right) d\left(\int_t^u (T-v)e^{-\lambda v}dv\right) \\
&= -\lambda\eta \left\{\left(\int_t^T e^{\lambda u}d\widetilde{W}_u\right)\left(\int_t^T (T-v)e^{-\lambda v}dv\right) - \int_t^T \left(\int_t^u (T-v)e^{-\lambda v}dv\right)e^{\lambda u}d\widetilde{W}_u\right\} \\
&= -\lambda\eta \int_t^T \left(\int_u^T (T-v)e^{-\lambda v}dv\right)e^{\lambda u}d\widetilde{W}_u \\
&= -\lambda\eta \int_t^T \left(\frac{(T-u)e^{-\lambda u}}{\lambda} + \frac{e^{-\lambda t} - e^{-\lambda u}}{\lambda^2}\right)e^{\lambda u}d\widetilde{W}_u \\
&= -\eta \int_t^T T - u + \frac{e^{-\lambda(T-u)} - 1}{\lambda}d\widetilde{W}_u \qquad (6.6)
\end{aligned}$$

Now adding the two expressions in equations (6.5) and (6.6) yields:

$$-\lambda\int_t^T (T-u)x_u du = \frac{x_t}{\lambda}\left(1-e^{-\lambda(T-t)}\right) - x_t(T-t) - \eta\int_t^T T-u+\frac{e^{-\lambda(T-u)}-1}{\lambda}d\widetilde{W}_u \tag{6.7}$$

It finally follows from equations (6.3) and (6.7) that:

$$\begin{aligned}\int_t^T x_u du &= -\lambda\int_t^T (T-u)x_u du + \eta\int_t^T (T-u)d\widetilde{W}_u + (T-t)x_t \\ &= \frac{x_t}{\lambda}\left(1-e^{-\lambda(T-t)}\right) + \frac{\eta}{\lambda}\int_t^T \left(1-e^{-\lambda(T-u)}\right)d\widetilde{W}_u. \end{aligned}\tag{6.8}$$

6.1.1 The T-forward measure

We define the new measure $\mathbb{Q}^T$ to be the T-forward (risk-adjusted) measure as (note that $\mathbb{Q}$ denotes the current risk-neutral measure):

$$\begin{aligned}\frac{d\mathbb{Q}^T}{d\mathbb{Q}} &= \frac{\exp\left\{-\int_0^T r_u du\right\}}{P(0,T)} \\ &= \frac{\exp\left\{-\int_0^T x_u du - \int_0^T \phi_u du\right\}}{P(0,T)} \\ &= \frac{\exp\left\{-\frac{\eta}{\lambda}\int_0^T 1-e^{-\lambda(T-u)}d\widetilde{W}_u - \int_0^T f(0,u)du - \int_0^T \frac{\eta^2}{2\lambda^2}(1-e^{-\lambda u})^2 du\right\}}{P(0,T)} \\ &= \exp\left\{-\frac{\eta}{\lambda}\int_0^T 1-e^{-\lambda(T-u)}d\widetilde{W}_u - \int_0^T \frac{\eta^2}{2\lambda^2}(1-e^{-\lambda u})^2 du\right\}.\end{aligned}$$

Where we have used equation (6.8). We have that the two processes defined by:

$$\begin{aligned} d\widetilde{W}_t^T &= d\widetilde{W}_t + \frac{\eta}{\lambda}\left(1-e^{-\lambda(T-t)}\right)dt, \\ d\widetilde{Z}_t^T &= d\widetilde{Z}_t, \end{aligned}$$

are independent Brownian motions under the T-forward measure $\mathbb{Q}^T$. We can redefine the dynamics of the interest rate and asset price under the T-forward measure as:

$$\begin{cases} dr_t &= \left[\lambda\bar{r} - \frac{\eta^2}{\lambda}\left(1-e^{-\lambda(T-t)}\right) - \lambda r_t\right]dt + \eta d\widetilde{W}_t^T \\ dS_t &= S_t\left[r_t - \frac{\rho\sigma\eta}{\lambda}\left(1-e^{-\lambda(T-t)}\right)\right]dt + S_t\left[\sigma\rho d\widetilde{W}_t^T + \sigma\sqrt{1-\rho^2}\widetilde{Z}_t^T\right] \end{cases} \tag{6.9}$$

The next step is to determine the expectation and variance of the asset price. First we integrate the dynamics in (6.9) to get for the interest rate ($0 \leq s < t \leq T$):

$$\begin{aligned} r_t &= r_s e^{-\lambda(t-s)} + \int_s^t \lambda \bar{r} e^{-\lambda(t-u)} du - \frac{\eta^2}{\lambda} \int_s^t e^{-\lambda(t-u)} \left[1 - e^{-\lambda(T-u)}\right] du + \eta \int_s^t e^{-\lambda(t-u)} d\widetilde{W}_u^T \\ &= x_s e^{-\lambda(t-s)} - \frac{\eta^2}{\lambda} \int_s^t e^{-\lambda(t-u)} \left[1 - e^{-\lambda(T-u)}\right] du + \eta \int_s^t e^{-\lambda(t-u)} d\widetilde{W}_u^T + \phi_t. \end{aligned}$$

For the asset price we can perform a similar integration to get, for $t < T$:

$$\begin{aligned} S_T =& S_t \exp\left\{ \int_t^T r_u du - \frac{\rho\sigma\eta}{\lambda} \int_t^T \left(1 - e^{-\lambda(T-u)}\right) du - \frac{1}{2}\sigma^2(T-t) \right. \\ & \left. + \sigma\rho\left(\widetilde{W}_T^T - \widetilde{W}_t^T\right) + \sigma\sqrt{1-\rho^2}\left(\widetilde{Z}_T^T - \widetilde{Z}_t^T\right) \right\} \\ =& S_t \exp\left\{ \frac{1 - e^{-\lambda(T-t)}}{\lambda} x_t + \frac{\eta}{\lambda} \int_t^T \left[1 - e^{-\lambda(T-u)}\right] d\widetilde{W}_u^T \right. \\ & - \frac{\eta^2}{\lambda} \int_t^T \int_t^u e^{-\lambda(u-s)} \left[1 - e^{-\lambda(T-s)}\right] ds du + \int_t^T f(0,u) du \\ & + \frac{\eta^2}{2\lambda^2} \int_t^T (1 - e^{-\lambda u})^2 du - \frac{\rho\sigma\eta}{\lambda} \int_t^T \left(1 - e^{-\lambda(T-u)}\right) du - \frac{1}{2}\sigma^2(T-t) \\ & \left. + \sigma\rho\left(\widetilde{W}_T^T - \widetilde{W}_t^T\right) + \sigma\sqrt{1-\rho^2}\left(\widetilde{Z}_T^T - \widetilde{Z}_t^T\right) \right\}. \end{aligned} \tag{6.10}$$

Note that S_T is lognormally distributed, so $\ln(S_T)$ is normally distributed. We will compute here $\mathbb{E}\left[\ln(S_T)|\mathcal{F}_t\right]$ and consecutively $\text{Var}(\ln(S_T))$. Using the expression for S_T in (6.10), it follows from simple calculations that we have:

$$\begin{aligned} \mathbb{E}^T\left[\ln(S_T)|\mathcal{F}_t\right] &= \ln(S_t) + \mathbb{E}\left[\ln\left(\frac{S_T}{S_t}\right)|\mathcal{F}_t\right] \\ &= \ln(S_t) + \frac{1 - e^{-\lambda(T-t)}}{\lambda} x_t - \frac{\sigma^2}{\lambda^2}\left\{T - t + \frac{2}{\lambda}e^{-\lambda(T-t)} - \frac{1}{2\lambda}e^{-2\lambda(T-t)} - \frac{3}{2\lambda}\right\} \\ &\quad + \ln\left(\frac{P(0,t)}{P(0,T)}\right) + \frac{\sigma^2}{2\lambda^2}\left\{T - t + \frac{2}{\lambda}\left(e^{-\lambda T} - e^{-\lambda t}\right) - \frac{1}{2\lambda}\left(e^{-2\lambda T} - e^{-2\lambda t}\right)\right\} \\ &\quad - \frac{\rho\sigma\eta}{\lambda}\left\{T - t - \frac{1}{\lambda}\left(1 - e^{-\lambda(T-t)}\right)\right\} - \frac{\sigma^2}{2}(T-t). \end{aligned}$$

In order to rewrite the expression, we define:

$$v(t,T) := \frac{\eta^2}{\lambda^2}\left\{T - t + \frac{2}{\lambda}e^{-\lambda(T-t)} - \frac{1}{2\lambda}e^{-2\lambda(T-t)} - \frac{3}{2\lambda}\right\},$$

so that we can rewrite the expression as:

$$\begin{aligned}\mathbb{E}^T\left[\ln(S_T)|\mathcal{F}_t\right] &= \ln(S_t) - \ln(P(t,T)) + \frac{1-e^{-\lambda(T-t)}}{\lambda}x_t - v(t,T) + \frac{1}{2}\left\{v(0,T)-v(0,t)\right\} \\ &\quad - \frac{\rho\sigma\eta}{\lambda}\left\{T-t-\frac{1}{\lambda}\left(1-e^{-\lambda(T-t)}\right)\right\} - \frac{\sigma^2}{2}(T-t) \\ &= \ln\left(\frac{S_t}{P(t,T)}\right) - \frac{\rho\sigma\eta}{\lambda}\left\{T-t-\frac{1}{\lambda}\left(1-e^{-\lambda(T-t)}\right)\right\} - \frac{\sigma^2}{2}(T-t) - \frac{1}{2}v(t,T).\end{aligned} \tag{6.11}$$

By similar computations we can determine $\mathrm{Var}^T(\ln(S_T))$:

$$\mathrm{Var}^T(\ln(S_T)) = V(t,T) + \sigma^2(T-t) + \frac{2\rho\sigma\eta}{\lambda}\left\{T-t-\frac{1}{\lambda}\left(1-e^{-\lambda(T-t)}\right)\right\}. \tag{6.12}$$

6.1.2 An explicit expression for the option value

We have all information to analytically determine the value of a European option under the Black-Scholes Hull-White model. To simplify notation we will define:

- $y := \ln(S_T)$
- $V^2 := Var(y)$
- $M := \mathbb{E}[y|\mathcal{F}_t]$

Note that we already know V^2 and M from equations (6.11) and (6.12).

The value of a European option at time T equals $(\psi(S_T - X))^+$, where $\psi \in \{1,-1\}$ depending on whether we have a call or a put option with strike X. We will determine the price of a European option at time t with expiration $T \geq t$ and strike X as:

$$
\begin{aligned}
V(t,T,X) &= P(t,T)\mathbb{E}^T\left[(\psi(S_T - X))^+ |\mathcal{F}_t\right] \\
&= P(t,T)\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}V}\left(\psi(e^y - X)\right)^+ e^{-\frac{1}{2}\frac{(y-M)^2}{V^2}}dy \\
&= P(t,T)\int_{\ln(X)}^{+\infty\cdot\psi} \frac{1}{\sqrt{2\pi}V}(e^y - X)e^{-\frac{1}{2}\frac{(y-M)^2}{V^2}}dy \\
&= P(t,T)\int_{\frac{\ln(X)-M}{V}}^{+\infty\cdot\psi} \frac{1}{\sqrt{2\pi}}(e^{M+Vz} - X)e^{-\frac{1}{2}z^2}dz \\
&= P(t,T)\left\{e^{M+\frac{1}{2}V^2}\int_{\frac{\ln(X)-M}{V}}^{+\infty\cdot\psi}\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}(z-V)^2}dz - X\int_{\frac{\ln(X)-M}{V}}^{+\infty\cdot\psi}\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}z^2}dz\right\} \\
&= P(t,T)\left\{e^{M+\frac{1}{2}V^2}\left[\Phi(+\infty\cdot\psi) - \Phi\left(\frac{\ln(X)-M-V^2}{V}\right)\right]\right. \\
&\quad \left.-X\left[\Phi(+\infty\cdot\psi) - \Phi\left(\frac{\ln(X)-M}{V}\right)\right]\right\} \\
&= P(t,T)\left\{\psi e^{M+\frac{1}{2}V^2}\Phi\left(-\psi\frac{\ln(X)-M-V^2}{V}\right) - \psi X\Phi\left(-\psi\frac{\ln(X)-M}{V}\right)\right\} \\
&= \psi S_t\Phi\left(\psi\frac{\ln\left(\frac{S_t}{XP(t,T)}\right) + \frac{1}{2}\left[V(t,T) + \sigma^2(T-t) + \frac{2\rho\sigma\eta}{\lambda}\left\{T - t - \frac{1}{\lambda}\left(1 - e^{-\lambda(T-t)}\right)\right\}\right]}{\sqrt{V(t,T) + \sigma^2(T-t) + \frac{2\rho\sigma\eta}{\lambda}\left\{T - t - \frac{1}{\lambda}\left(1 - e^{-\lambda(T-t)}\right)\right\}}}\right) \\
&\quad - \psi XP(t,T)\Phi\left(\psi\frac{\ln\left(\frac{S_t}{XP(t,T)}\right) - \frac{1}{2}\left[V(t,T) + \sigma^2(T-t) + \frac{2\rho\sigma\eta}{\lambda}\left\{T - t - \frac{1}{\lambda}\left(1 - e^{-\lambda(T-t)}\right)\right\}\right]}{\sqrt{V(t,T) + \sigma^2(T-t) + \frac{2\rho\sigma\eta}{\lambda}\left\{T - t - \frac{1}{\lambda}\left(1 - e^{-\lambda(T-t)}\right)\right\}}}\right).
\end{aligned}
$$

6.1.3 An explicit expression for the bond price

The last step that remains in the derivation of the analytic solution for a European option under the Black-Scholes Hull-White model is determining an explicit expression for the discount rate, or zero coupon bond $P(t,T)$. In order to do this, recall that the dynamics of the interest rate under the T-forward measure are given by:

$$dr_t = \left[\lambda\bar{r} - \frac{\eta^2}{\lambda}\left(1 - e^{-\lambda(T-t)}\right) - \lambda r_t\right]dt + \eta d\widetilde{W}_t^T.$$

If we define $\theta = \lambda\bar{r} - \frac{\eta^2}{\lambda}\left(1 - e^{-\lambda(T-t)}\right)$ then we see $P(t,T)$ as a function $Y(t,r_t)$, where the function $Y(t,r)$ must solve the boundary value problem:

$$\frac{\partial Y}{\partial t} + (\theta - \lambda r)\frac{\partial Y}{\partial r} + \frac{1}{2}\eta^2\frac{\partial^2 Y}{\partial r^2} = rY,$$

subject to: $Y(T,r) = 1 \quad \forall r$.
We can guess a solution $Y(t,r,T)$ of the form $Y(t,r,T) = A(t,T)e^{-B(t,T)r}$, if we see $A(t,T)$ and $B(t,T)$ as functions of t then they must solve the following system of ordinary differential equations:

$$\begin{aligned}
&\frac{dA}{dt} - \theta AB + \frac{1}{2}\eta^2 AB^2 = 0,\\
&\frac{dB}{dt} - \lambda B + 1 = 0,
\end{aligned}$$

subject to: $A(T,T) = 1$ and $B(T,T) = 0$. This leads us to an expression for $B(t,T)$:

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

Subsequently it can be derived that for $A(t,T)$ it holds that:

$$A(t,T) = \exp\left\{\left(\frac{\theta}{\lambda} - \frac{\eta^2}{2\lambda^2}\right)(B(t,T) - T + t) - \frac{\eta^2}{4\lambda}B^2(t,T)\right\}.$$

From this line of reasoning it follows finally that we have for our discount factor, or zero coupon bond we have the following explicit expression:

$$P(t,T) = \exp\left\{\left(\frac{\theta}{\lambda} - \frac{\eta^2}{2\lambda^2}\right)(B(t,T) - T + t) - \frac{\eta^2}{4\lambda}B^2(t,T) - r_t B(t,T)\right\}.$$

6.1.4 Reference value

Now that we have found an explicit expression for the value of a European put option under the Black-Scholes Hull-White model we can compute a reference value using the parameters we will use for our simulation further along. In section 6.3 we will compare the value of a European put from our simulations to this analytic value in order to see how accurate our results are. We will choose the following parameters:

- $S_0 = 10$
- $X = 12$
- $r_0 = 0.05$
- $\bar{r} = 0.06$
- $\eta = 0.02$
- $\lambda = 2$
- $\rho = 0.1$

- $T = 1$
- $\sigma = 0.3$

When we substitute these values into the formulas derived above we first get for time $t = 0$:

$$P(0, T) = 0.945909335495978,$$

and using this value we can then conclude that the option value at time $t = 0$ must be:

$$V(0, T, X) = 2.061635287422108$$

6.2 Matlab implementation

Now that we know the details of the Black-Scholes Hull-White model it is time to implement the model in the SGM. We will use the Hull-White equation with $\overline{r}_t$ constant to simplify the implementation, this special case is known as the Vasicek model (see [17]). This means that we have a mean-reverting interest rate, where the mean will equal $\bar{r}$ as $t \to \infty$. In order to implement this we can use large parts of the already existing program for the SGM with bundling and only adjust the function file that generates the paths to also generate the interest rate paths and adapt all lines including the discounting factor or interest rate to use the generated values r_t.

6.2.1 Generating paths using Black-Scholes Hull-White model

As mentioned we will have to adapt the way we generate paths to accommodate the stochastic interest rates. In order to do so we now construct one large matrix containing both the simulated asset paths and the simulated interest rate paths. If we take N timesteps this means that in this large matrix the first $N + 1$ columns constitute the matrix of stock paths and the last $N + 1$ columns constitute the matrix of interest rate paths. We will need to store the interest rate paths as well as the asset price paths because we will need these paths in our computations.

Next, we need to decide how to simulate the interest rate paths. Since it is unlikely but not impossible that the interest rate becomes negative, we can use an Euler discretization to simulate the interest rate paths, just as we do for the asset price. This means that we will use the following scheme to generate the asset and interest rate paths:

$$\begin{cases} S_{t+1} = S_t + S_t(r_t dt + \rho\sigma W_t + \sqrt{1-\rho^2}\sigma Z_t) \\ r_{t+1} = r_t + \lambda(\bar{r} - r_t)dt + \eta W_t \end{cases}$$

Here we have that W_t and Z_t are independent standard Gaussian variables.

6.2.2 Adjusting the SGM for Black-Scholes Hull-White model

Using the latest version of the stochastic grid method, including the new bundling algorithm we will be able to use the analytic moments of the distribution. We will however need to adjust these moments to accommodate for the stochastic interest rates. The first moment will now be described by:

$$\mu_t = \ln(S_t) + (r_t - 0.5\sigma^2)dt.$$

Also in the discounting formulas needed to compute the continuation value we must use r_t instead of r. At the end we must create a vector containing the discounting values for the optimal exercise step, hence if in path i it is optimal to exercise at time j we must have that in the discounting vector the i-th element equals $\exp(-\sum_{k=0}^{j-1} r_{i,k} dt)$, so that we can use this vector to compute our final estimators.

6.3 Simulation experiments Black-Scholes Hull-White model

Three simulation experiments will be performed to assess the quality of the estimations made using the SGM for the Black-Scholes Hull-White model. In the first experiment we will determine the price of a standard European put option using ordinary Monte Carlo simulation and using the SGM. The results of this experiment we can compare to the analytic solution for the European option under the Black-Scholes Hull-White model given in section 6.1.4. In the second experiment we will show that if we choose the parameters correctly the price of a European option under the Black-Scholes Hull-White model will converge to the price under the Black-Scholes model. We will thus show that by choosing the parameters for Black-Scholes Hull-White in such a way that the results converge to the Black-Scholes results, the results of SGM for the Black-Scholes Hull-White model in fact converge to the results for the Black-Scholes model. Then finally we will perform some computations for American options using the SGM and the Black-Scholes Hull-White model.

In order to be able to compare the results to the analytic solution given in section 6.1.4 we will use the same parameters in all our simulation experiments. In addition we will generally use the following discretization parameters:

- $N = 50$, i.e. we will take 50 timesteps
- $n = 1000$, hence we will generate 1000 asset and interest rate paths per simulation

- $b = 4$, we will make 2^4 bundles at each timestep

Note that choosing η, the volatility of the interest rate, as small as we have done reduces the likelihood of getting negative values for the interest rate, especially since we take relatively large timesteps.

6.3.1 Ordinary MC and SGM for European put

In this experiment we compare the results of a standard Monte Carlo simulation and the SGM for a European put using the Black-Scholes Hull-White model. The parameters are as defined in sections 6.1.4 and 6.3 and 30 runs of each of the methods are performed to determine the mean values and the standard deviations of the results of both methods. The computation time per simulation is also given for both SGM and the ordinary Monte Carlo simulation. This gives us the results in table 6.1:

Table 6.1: Simulation results with BSHW for European put

	Mean	**St. Dev.**	**Time/run**
SGM lower bound	2.0617	0.0219	
SGM direct estimator	2.0608	0.0051	0.7500
Ordinary MC	2.0559	0.0245	0.1325

What we can see here is that the results in terms of the mean of the simulations are very close for the two methods. More importantly though, the results of both methods are very close to the results obtained in section 6.1.4. This tells us that the method provides us very accurate results. Not only do we get very accurate results on average, but we also find a very small standard deviation in our results from SGM compared to the results from the ordinary Monte Carlo simulations. The standard deviation of the direct estimator of the SGM is almost five times smaller than the standard deviation of the ordinary Monte Carlo estimator. In order to gain the same amount of accuracy using ordinary Monte Carlo we would need many more paths and/or much smaller time steps, which will lead to longer computation time. As we can see the computation time of the SGM currently is longer than that of the ordinary Monte Carlo simulation, but here it is important to keep in mind that SGM is mostly meant to price options for which ordinary Monte Carlo is too involved. Also SGM might be computationally somewhat more expensive than ordinary Monte Carlo simulation, but all computations can still be done in less than a second.

6.3.2 Convergence of SGM to Black-Scholes model

In the previous experiment, comparing ordinary Monte Carlo results for a European put to results of SGM for a European put and the analytic solution for a European put we have seen that our results here are very accurate. In this section we will exploit some of the mathematical features of the Black-Scholes Hull-White model, and in particular its relation to the Black-Scholes model to test the accuracy of the method even further. We have already seen in section 5.4 that our latest version of SGM, using the bundles gives us very accurate results under the Black-Scholes model for an American put option. Here we will show that by choosing the parameters for the Black-Scholes Hull-White model and the Black-Scholes model in a clever way, the results of the two methods will converge.

Choosing the parameters

We know that SGM gives us accurate results for the Black Scholes model so if we choose the parameters appropriately then we can make a comparison between the results for SGM with Black Scholes and with Black-Scholes Hull-White to gain more confidence in the results of our method. The next step is to determine the appropriate parameters. We know that the differential equation for the interest rate in the BSHW model is given by (with constant expectation for r_t):

$$dr_t = \lambda(\overline{r} - r_t)dt + \eta dW_t^r.$$

If we start at r_0 we can approximate r_T with:

$$r_T = r_0 + \sum_{i=1}^{T/dt} \left(\lambda(\overline{r} - r_i)dt + \eta dW_t^r\right).$$

Hence it follows:

$$\mathbb{E}[r_T] = r_0 + \sum_{i=1}^{T/dt} \left(\lambda(\overline{r} - r_i)dt\right).$$

If let λ tend to zero and let $r_0 = \overline{r}$, this will converge to the standard Black-Scholes model with parameter $r = r_0$. In order to speed up the convergence it is sensible to also choose $\rho = 0$ and η to be small to reduce the variance of r_t. We will use in our computations $r_0 = \overline{r} = 0.05$, $\rho = 0$, and subsequently choose the following sets of parameters:

- $\lambda = 0.2$, $\eta = 0.2$
- $\lambda = 0.1$, $\eta = 0.1$
- $\lambda = 0.05$, $\eta = 0.05$
- $\lambda = 0.00001$, $\eta = 0.00001$

All other parameters will be the same as in section 6.1.4. Since in this case we can test the results for an American put option we will use that at every timestep made the option can be exercised.

Results

We will perform 30 runs with each set of parameters and compare the results to the results obtained by 30 runs of the SGM for the Black-Scholes model under the same parameters (so $r = r_0 = 0.05$). The SGM for the Black-Scholes model gives us the following results:

Direct Estimator: 2.2661, with a standard deviation of 0.0007

Lower Bound: 2.2572, with a standard deviation of 0.0289

First we will take a look at the direct estimator results under the parameters we have chosen for the Black-Scholes Hull-White model, given in table 6.2.

Table 6.2: Direct estimator results SGM with BSHW and BS for American put

	Mean	**St. Dev.**
$\lambda = 0.2$, $\eta = 0.2$	2.2777	0.0091
$\lambda = 0.1$, $\eta = 0.1$	2.2671	0.0045
$\lambda = 0.05$, $\eta = 0.05$	2.2648	0.0035
$\lambda = 0.00001$, $\eta = 0.00001$	2.2659	0.0010

As can be seen in table 6.2 the results converge very nicely to the direct estimator value from the Black-Scholes Hull-White model. In addition we also see that the standard deviation converges to the value we found in the Black-Scholes case. Now it is time to take a look at the results for the lower bound estimator, given in table 6.3.

Table 6.3: Lower bound results SGM with BSHW and BS for American put

	Mean	**St. Dev.**
$\lambda = 0.2$, $\eta = 0.2$	2.4360	0.0298
$\lambda = 0.1$, $\eta = 0.1$	2.3078	0.0247
$\lambda = 0.05$, $\eta = 0.05$	2.2795	0.0344
$\lambda = 0.00001$, $\eta = 0.00001$	2.2606	0.0249

Just as with the direct estimator, the lower bound estimator converges nicely to the value for the Black-Scholes model. We see that the convergence seems to be somewhat slower for the lower bound than for the direct estimator, but this picture could be influenced by the fact that the standard deviation of the lower bound is larger than the standard deviation of the direct estimator. Nevertheless the convergence is clearly visible and again the standard deviation is very close to the one found in the Black-Scholes case.

The results for the convergence of the Black-Scholes Hull-White model to the Black-Scholes model for the proper choice of parameters shows us that we can not only obtain good results for European options using the implementation of the Black-Scholes Hull-White model in SGM but also for American style options.

6.3.3 Computation time and accuracy

Finally we will also perform a simulation experiment to determine computation time and accuracy for some different combinations of the number of paths and the number of bundles. Clearly the fewer bundles and paths we use, the shorter the computation time will be. But if we choose to use too few paths or bundles the results will be less accurate. This experiment will make an assessment of the tradeoff that is made between computation time and accuracy if we change the number of bundles and the number of paths. As we have seen in section 5.4 there is no point in taking too many bundles for a certain number of paths. We also saw there that increasing the number of paths and/or bundles doesn't give a significant improvement in the results. To see if this is the case here, we will now take $n = 10^3$ and $n = 10^4$ asset paths, and look at the results for $b = 2$, $b = 3$ and $b = 4$ if $n = 10^3$, and for $b = 3$, $b = 4$ and $b = 5$ if we have $n = 10^4$ asset paths. As with the experiment in section 6.3.2 we will take $N = 50$ timesteps, where at each timestep the option may be exercised. All other parameters will be equal to those stated in section 6.1.4. As in previous experiments we will perform 30 runs of each scenario to get a good estimate for the mean and standard deviation of the results. The results for $n = 10^3$ asset paths are given in table 6.4, the results for $n = 10^4$ asset paths can be found in table 6.5.

Table 6.4: Simulation results SGM with bundling for BSHW, $n = 10^3$, for American put

		Mean	**Std. dev.**	**Time/run (s)**
$b = 2$	**lower bound**	2.2461	0.0350	0.2985
	direct estimator	2.2499	0.0010	
$b = 3$	**lower bound**	2.2428	0.0311	0.4239
	direct estimator	2.2496	0.0007	
$b = 4$	**lower bound**	2.2488	0.0281	0.6727
	direct estimator	2.2493	0.0014	

Table 6.5: Simulation results SGM with bundling for BSHW, $n = 10^4$, for American put

		Mean	**Std. dev.**	**Time/run (s)**
$b = 3$	**lower bound**	2.2474	0.0104	2.2910
	direct estimator	2.2494	0.0002	
$b = 4$	**lower bound**	2.2486	0.0075	2.9461
	direct estimator	2.2494	0.0001	
$b = 5$	**lower bound**	2.2501	0.0074	4.1621
	direct estimator	2.2493	0.0002	

If we look at the results in table 6.4 and table 6.5 we see very similar results to the results in table 5.2 and table 5.3. The result don't improve much as we increase the number of bundles, and in this case it seems that we get the best results for $n = 10^3$ asset paths if we take $2^3 = 8$ bundles and for $n = 10^4$ asset paths if we take $2^4 = 16$ bundles. The computation time is somewhat longer in the Black-Scholes Hull-White case than in the Black-Scholes case, but this is reasonable since we need to simulate interest rate paths as well as asset paths, and need to do a little more computations for each timestep because we have to use the simulated interest rate paths in stead of the deterministic interest rate. Because there is more uncertainty in the model we see that in general the standard deviation of the results is somewhat bigger than in the Black-Scholes case, but that they are still very small especially for the direct estimator.

Chapter 7

Heston Model

In this chapter we will introduce the Heston model, where the volatility is stochastic. In order to be able to use this model under all parameter values we will need a new discretization scheme. First we will describe this new scheme. Then we will implement the Heston model in our stochastic grid method and we will end the chapter with simulation experiments for the Heston model.

The Heston model helps to make asset paths more realistic, as it allows for the volatility smile or skew that is often observed in implied volatility curves. The volatility will be modeled as a mean reverting process, such that we will have the stochastic differential equations given in equation (7.1)

$$\begin{cases} dS_t = rS_t dt + \sqrt{\nu_t} S_t dW_t^S \\ d\nu_t = \kappa(\theta - \nu_t)dt + \xi\sqrt{\nu_t} dW_t^\nu \\ dW_t^S dW_t^\nu = \rho dt \end{cases} \tag{7.1}$$

In the dynamics in (7.1) we have that ν_t is the variance of the asset price, and hence describes the volatility of the asset price, ξ is the volatility of the variance, θ is the mean reversion parameter for the variance of the asset price, κ gives us the speed of mean reversion for the variance of the asset price and ρ is the correlation between the volatility and the asset price.

For notational convenience we rewrite the dynamics for the asset price. By taking $X_t = \ln(S_t)$, a simple application of Itô's lemma gives us that the system in (7.1) is equivalent to:

$$\begin{cases} dX_t = (r - \frac{1}{2}\nu_t)dt + \sqrt{\nu_t} dW_t^S \\ d\nu_t = \kappa(\theta - \nu_t)dt + \xi\sqrt{\nu_t} dW_t^\nu \\ dW_t^S dW_t^\nu = \rho dt \end{cases} \tag{7.2}$$

7.1 QE-Scheme

It is clear from the stochastic differential equations in (7.2) that these will be somewhat more involved to handle numerically, since we can't allow the volatility process to become negative because we would take a square root from ν_t in the asset price process and in the volatility process. If ν_t would become negative at some point we would get imaginary asset prices and volatilities and the whole pricing process would fail. The Feller condition tells us that as long as we have that $2\kappa\theta \geq \xi^2$ we should never get non-negative values for ν_t in the continuous solution of the dynamics, and we avoid this whole issue. Numerically however, an Euler discretization may still provide negative values for the variance ν_t. If the Feller condition is satisfied and we choose our timesteps sufficiently small the probability of producing negative values under the Euler discretization is very small.

There is another problem with this condition, because in practical situations it usually holds that $2\kappa\theta \ll \xi^2$, so that if we choose the parameters as to obey the Feller condition we are modeling rather unrealistic values for the volatility. We must find a smarter way to deal with this problem, and the answer is to use a different discretization scheme, namely the QE-scheme or quadratic-exponential scheme. We will now derive the QE-scheme step by step, where we will follow the line of reasoning as described in [1]. First we do some observations on the distribution of the volatility, then we will define an approximation method and approximate density for the approximated volatility, and then we will define an approximation for ν_t.

7.1.1 Distribution of the volatility

The volatility as described by the dynamics of the Heston model resemble a non-central chi-squared distribution, as Broadie and Kaya explain in [4] in detail. In order to see this, define:

$$\begin{aligned} d &= \frac{4\kappa\theta}{\xi^2} \\ n(t,T) &= \frac{4\kappa e^{-\kappa(T-t)}}{\xi^2\left(1-e^{-\kappa(T-t)}\right)}, \quad T > t \end{aligned}$$

For all $T > t \geq 0$ we have that conditional on ν_t, ν_T follows the distribution of $e^{-\kappa(T-t)}/n(t,T)$ times a non-central chi-square variable with d degrees of freedom and non-centrality parameter $\nu_t n(t,T)$. Hence we have that:

$$\mathbb{P}(\nu_T < x|\nu_t) = F_\chi\left(\frac{xn(t,T)}{e^{-\kappa(T-t)}}; d, \nu_t n(t,T)\right),$$

where:

$$F_\chi(z; v, \lambda) = e^{-\lambda/2}\sum_{j=0}^{\infty}\frac{(\lambda/2)^j}{j!2^{v/2+j}\Gamma(v/2+j)}\int_0^z z^{v/2+j-1}e^{-x/2}dx$$

is the distribution function of the non-central chi-squared distribution with v degrees of freedom and non-centrality parameter λ. We can use this knowledge of the non-central chi-square distribution to determine the first two moments of ν_T. Conditional on ν_t, we have for all $T > t \geq 0$ that:

$$\begin{aligned}
\mathbb{E}\left(\nu_T|\nu_t\right) &= \theta + \left(\nu_t - \theta\right) e^{-\kappa(T-t)}, \\
\mathrm{Var}\left(\nu_T|\nu_t\right) &= \frac{\nu_t \xi^2 e^{-\kappa(T-t)}}{\kappa}\left(1 - e^{-\kappa(T-t)}\right) + \frac{\theta\xi^2}{2\kappa}\left(1 - e^{-\kappa(T-t)}\right)^2.
\end{aligned}$$

7.1.2 Approximation method

Now that we have some more knowledge of the distribution of the volatility, we can start defining our approximation method. For $\hat{\nu}_t$, the approximation of ν_t, sufficiently large (we will define 'sufficiently large' more precisely later on), we will write:

$$\hat{\nu}(t + \Delta t) = a(b + W^{\nu})^2 \tag{7.3}$$

Where a and b are constants depending on the timestep Δt, the current volatility $\hat{\nu}_t$ and the parameters of the stochastic differential equation for the volatility. The values of a and b will be determined by moment matching. This in fact is the reason this approximation can only be used for sufficiently large values of $\hat{\nu}_t$, since otherwise the moment matching will fail. For small values of $\hat{\nu}_t$ we will need a different scheme. As ν_t becomes small we have that the non-centrality parameter of the non-central chi-squared distribution approaches zero and hence the distribution of $\nu(t+\Delta t)$ becomes proportional to that of an ordinary chi-squared distribution with d degrees of freedom, recall that the density of this distribution is given by:

$$f_{\chi^2}(x; d) = \frac{1}{2^{\nu/2}\Gamma(\nu/2)} e^{-x/2} x^{\nu/2-1}.$$

We will use an approximated density for $\hat{\nu}(t + \Delta t)$ of the form:

$$\mathbb{P}\left(\hat{\nu}_t \in [x, x + dx]\right) \approx \left(p\delta(0) + \beta(1-p)e^{-\beta x}\right) dx, \qquad x \geq 0 \tag{7.4}$$

Here δ denotes the Dirac delta-function, and p and β are non-negative constants that will be determined shortly. Note that the parameter p specifies the mass concentrated at the origin, this mass at the origin is then supplemented by an exponential tail similar to the ordinary chi-squared density. It can be shown that if $p \in [0, 1]$ and $\beta \geq 0$ equation (7.4) indeed gives us a valid density function. We need to define a way to sample according to the approximate density. This will turn out to be straightforward. If we integrate the density to determine the distribution function we find that:

$$\Psi(x) = \mathbb{P}\left(\hat{\nu}(t + \Delta_t) \leq x\right) = p + (1-p)\left(1 - e^{-\beta x}\right), \qquad x \geq 0.$$

Clearly we can compute the inverse of $\Psi(x)$ to be:

$$\Psi^{-1}(u;p,\beta) = \begin{cases} 0, & 0 \leq u \leq p \\ \beta^{-1} \ln\left(\frac{1-p}{1-u}\right), & p < u \leq 1 \end{cases}$$

From this we obtain the simple sampling scheme:

$$\hat{\nu}(t+\Delta t) = \Psi^{-1}(U_\nu; p, \beta) \tag{7.5}$$

Here we draw U_ν from a uniform (0,1)-distribution. Now we have the discretiziation scheme for both small and large values of ν_t, so what remains is to determine the constants a and b for the scheme given in equation (7.3), p and β for the scheme given in (7.5) and a way to determine when to use which discretization scheme.

Determining a en b

We start by determining a and b for the scheme in (7.3). For notational convenience define:

$$m = \theta + (\hat{\nu}_t - \theta)\, e^{-\kappa\Delta t} = \mathbb{E}[\hat{\nu}(t+\Delta t)], \tag{7.6}$$

$$s^2 = \frac{\hat{\nu}_t \xi^2 e^{-\kappa\Delta t}}{\kappa}\left(1 - e^{\kappa\Delta t}\right) + \frac{\theta\xi^2}{2\kappa}\left(1 - e^{-\kappa\Delta t}\right)^2 = \mathrm{Var}(\hat{\nu}(t+\Delta t)), \tag{7.7}$$

$$\Upsilon = \frac{s^2}{m^2}. \tag{7.8}$$

Because $\hat{\nu}_t \in [0,\infty)$ it follows by inserting these values in equation (7.8) that $\Upsilon \in (0, \xi^2/(2\kappa\theta)]$. In discretization equation:

$$\hat{\nu}(t+\Delta t) = a(b + W^\nu)^2,$$

we actually describe the distribution of $\hat{\nu}(t+\Delta t)$ as a times a non-central chi-squared distribution with one degree of freedom and non-centrality parameter b^2, so it follows that:

$$\mathbb{E}[\hat{\nu}(t+\Delta t)] = a(1+b^2), \tag{7.9}$$

$$\mathrm{Var}(\hat{\nu}(t+\Delta t)) = 2a^2(1+2b^2). \tag{7.10}$$

Setting these expressions equal to m and s^2 as given in equations (7.6) and (7.7) yields the following system of equations:

$$a(1+b^2) = m, \tag{7.11}$$

$$2a^2(1+2b^2) = s^2. \tag{7.12}$$

So if we set $x = b^2$ then elimination of a from (7.11) and (7.12) yields the following expression:

$$x^2 + 2x(1 - 2\Upsilon^{-1}) + 1 - 2\Upsilon^{-1} = 0.$$

By examining the discriminant of this expression, we find that a solution is only possible if $\Upsilon \leq 2$, which is the case for large values of $\hat{\nu}_t$. If we assume this to be true we find that:

$$\begin{aligned} b^2 &= 2\Upsilon^{-1} - 1 + \sqrt{2\Upsilon^{-1}}\sqrt{2\Upsilon^{-1} - 1} \geq 0, \\ a &= \frac{m}{1 + b^2} \end{aligned}$$

Determining p and β

For the determination of p and β in the discretization scheme defined in equation (7.5), let m, s and Υ be defined as in equations (7.6), (7.7) and (7.8). If we directly integrate the density:

$$\mathbb{P}\left(\hat{\nu}_t \in [x, x + dx]\right) \approx \left(p\delta(0) + \beta(1 - p)e^{-\beta x}\right) dx \qquad \text{for} x \geq 0,$$

we see that:

$$\begin{aligned} \mathbb{E}[\hat{\nu}(t + \Delta t)] &= \frac{1 - p}{\beta}, \\ \mathrm{Var}(\hat{\nu}(t + \Delta t)) &= \frac{1 - p^2}{\beta^2}. \end{aligned}$$

Again by moment-matching it follows that:

$$\frac{1 - p}{\beta} = m, \tag{7.13}$$

$$\frac{1 - p^2}{\beta^2} = s^2. \tag{7.14}$$

Elimination of β from the equations (7.13) and (7.14) then yields the following equation:

$$(1 + \Upsilon)p^2 - 2\Upsilon p + \Upsilon - 1 = 0.$$

Since we need a solution $p \leq 1$, we will always have one solution, namely:

$$p = \frac{\Upsilon - 1}{\Upsilon + 1}.$$

For the solution to make sense we must have that $p \geq 0$, which tells us that $\Upsilon \geq 1$. It then follows that for β, we have:

$$\beta = \frac{2}{m(\Upsilon + 1)}.$$

Switching schemes

The thing that remains is to determine when to switch from the discretization scheme in (7.3) to the discretization scheme given in (7.5). We have seen that the quadratic sampling scheme, as given in (7.3) (for high values of $\hat{\nu}_t$) is applicable if $\Upsilon \leq 2$, while the exponential sampling scheme, given in (7.5) (for low values of $\hat{\nu}_t$) is applicable for $\Upsilon \geq 1$. Clearly, it follows that under all circumstances at least one of the two schemes can be used, since these domains overlap. A simple way to decide when to use which sample scheme is to choose some critical level $\Upsilon_c \in [1, 2]$ and use the quadratic sampling scheme if $\Upsilon \leq \Upsilon_c$, and to use the exponential sampling scheme otherwise. It turns out that the exact choice of this critical value has almost no effect on the overall results, hence a critical value of $\Upsilon_c = 1.5$ will be used.

7.1.3 A discretization scheme for the asset price

In order to make computations more accurate when using the Heston model to simulate asset prices we will need a more sophisticated discretization scheme for the asset price. To derive this scheme we will first derive an analytic expression for the logarithm of the asset price, as was done in 7.1. Recall the system of stochastic differential equations for $X_t = ln(S_t)$:

$$\begin{aligned} dX_t &= (r - \frac{1}{2}\nu_t)dt + \sqrt{\nu_t} dW_t^S, \\ d\nu_t &= \kappa(\theta - \nu_t)dt + \xi\sqrt{\nu_t} dW_t^\nu, \\ dW_t^S dW_t^\nu &= \rho dt. \end{aligned}$$

If we integrate the stochastic differential equation for the volatility we get that:

$$\nu(t + \Delta t) = \nu_t + \int_t^{t+\Delta t} \kappa(\theta - \nu_u)du + \xi \int_t^{t+\Delta t} \sqrt{\nu_u} dW_u^\nu,$$

which can be rewritten as:

$$\int_t^{t+\Delta t} \sqrt{\nu_u} dW_u^\nu = \frac{1}{\xi}\left(\nu(t + \Delta t) - \nu_t - \kappa\theta\Delta t + \kappa \int_t^{t+\Delta t} \nu_u du\right) \tag{7.15}$$

By a Cholesky decomposition of the differential equations we can write:

$$dX_t = (r - \frac{1}{2}\nu_t)dt + \rho\sqrt{\nu_t} dW_t^\nu + \sqrt{1 - \rho^2}\sqrt{\nu_t} dW_t \tag{7.16}$$

Where W_t is a Brownian motion independent of W_t^ν. Integrating the stochastic differential equation for the logarithm of the asset price given in equation (7.16) then yields, using the equation for $\int_t^{t+\Delta t} \sqrt{\nu_u} dW_u^\nu$ derived in equation (7.15):

$$\begin{aligned}
X(t+\Delta t) &= X_t + \int_t^{t+\Delta t} \left(r - \frac{1}{2}\nu_u \right) du + \rho \int_t^{t+\Delta t} \sqrt{\nu_u} dW_u^\nu + \sqrt{1-\rho^2} \int_t^{t+\Delta t} \sqrt{\nu_u} dW_u \\
&= X_t + r\Delta t - \frac{1}{2} \int_t^{t+\Delta t} \nu_u du + \frac{\rho}{\xi} \left(\nu(t+\Delta t) - \nu_t - \kappa\theta\Delta t + \kappa \int_t^{t+\Delta t} \nu_u du \right) \\
&\quad + \sqrt{1-\rho^2} \int_t^{t+\Delta t} \sqrt{\nu_u} dW_u \\
&= X_t + r\Delta t + \frac{\rho}{\xi} \left(\nu(t+\Delta t) - \nu_t - \kappa\theta\Delta t \right) + \left(\frac{\kappa\rho}{\xi} - \frac{1}{2} \right) \int_t^{t+\Delta t} \nu_u du \\
&\quad + \sqrt{1-\rho^2} \int_t^{t+\Delta t} \sqrt{\nu_u} dW_u. \qquad (7.17)
\end{aligned}$$

In order to be able to use the expression in (7.17) for a discretization of the asset price we need to find a way to discretize the time-integral of the volatility. As suggested in [1] we will approximate the time-integral by:

$$\int_t^{t+\Delta t} \nu_u du \approx \Delta t \left(\gamma_1 \nu_t + \gamma_2 \nu(t+\Delta t) \right), \qquad (7.18)$$

with constants γ_1 and γ_2. Choosing $\gamma_1 = 1, \gamma_2 = 0$ will give an "Euler-like" setting, a central discretization, on the other hand would take $\gamma_1 = \gamma_2 = \frac{1}{2}$. It is also possible to use a more sophisticated approach to choose the values of γ_1 and γ_2 through moment matching. In order to do this the analytic moments of the time-integral of the volatility are required, these moments can be found in [5], we will not explore this possibility here. Since the Brownian motion in the exact solution for X_t is independent of ν_t, the integral;

$$\int_t^{t+\Delta t} \sqrt{\nu_u} dW_u,$$

has a Gaussian distribution with mean zero and variance $\int_t^{t+\Delta t} \nu_u$ du. Combining this knowledge with the approximation of the time-integral in (7.18) and the analytic expression for X_t in (7.17) leads to the following discretization scheme:

$$\begin{aligned}
\hat{X}(t+\Delta t) &= \hat{X}_t + r\Delta t + \frac{\rho}{\xi} \left(\hat{\nu}(t+\Delta t) - \hat{\nu}_t - \kappa\theta\Delta t \right) + \Delta t \left(\frac{\kappa\rho}{\xi} - \frac{1}{2} \right) \left(\gamma_1 \hat{\nu}_t + \gamma_2 \hat{\nu}(t+\Delta t) \right) \\
&\quad + \sqrt{\Delta t} \sqrt{1-\rho^2} \sqrt{\gamma_1 \hat{\nu}_t + \gamma_2 \hat{\nu}(t+\Delta t)} \cdot Z \\
&= \hat{X}_t + K_0 + K_1 \hat{\nu}_t + K_2 \hat{\nu}(t+\Delta t) + \sqrt{K_3 \hat{\nu}_t + K_4 \hat{\nu}(t+\Delta t)} \cdot Z, \qquad (7.19)
\end{aligned}$$

Where Z is a standard normal random variable, independent of $\hat{\nu}$ and the constants K_0, K_1, K_2, K_3 and K_4 are given by:

$$\begin{aligned}
K_0 &= \left(r - \frac{\rho\kappa\theta}{\xi}\right)\Delta t,\\
K_1 &= \gamma_1\left(\frac{\kappa\rho}{\xi} - \frac{1}{2}\right)\Delta t - \frac{\rho}{\xi},\\
K_2 &= \gamma_2\left(\frac{\kappa\rho}{\xi} - \frac{1}{2}\right)\Delta t + \frac{\rho}{\xi},\\
K_3 &= \gamma_1\left(1-\rho^2\right)\Delta t,\\
K_4 &= \gamma_2\left(1-\rho^2\right)\Delta t.
\end{aligned}$$

7.2 SGM for the Heston model

We adapt the SGM for the Heston model. Clearly, we have to make some changes to the method to accommodate the stochastic volatility. First of all we need to make a new version of the code that generates the asset prices, in which we use the QE-scheme, with the adapted discretization scheme for the asset price, as given in equation (7.19). Also we need to store all simulated volatilities to be able to adjust the moments of the density in SGM accordingly. This means that, instead of using a constant volatility we will need to use the simulated volatilities in all computation steps.

7.3 Simulation experiments

In order to test our method we will start by setting the parameters so that the Feller condition is satisfied, and hence we can compare results from the QE-scheme for a European put option with the results of an ordinary Monte Carlo simulation using an Euler discretization and sufficiently small timesteps. By taking sufficiently small timesteps we reduce the probability of getting negative values for the volatility from the Euler scheme. Thereafter we will also do an experiment to show what happens if the Feller condition is not satisfied and we hence have a high probability of simulating a negative value for the volatility if we use the Euler scheme. To test the results of SGM for the Heston model we will first make a comparison with the results of the European put by approximating the price of the European put using the SGM, then we will make a comparison for an American put using reference values from literature. The last experiment for the Heston model will be pricing an American put using SGM and the QE-scheme using different numbers of bundles and asset paths, to compare results and gain insight in the computation time.

7.3.1 QE-scheme and Euler discretization for European put

In order to test the QE-scheme we start by performing two ordinary Monte Carlo simulations for a European put, one using an Euler discretization and one using the QE-scheme when the Feller condition is satisfied and we have high probability that if we take sufficiently small timesteps the volatility will remain positive even when using the Euler scheme. In order to show the advantages of the QE-scheme we will show thereafter what happens if the Feller condition is not satisfied, and hence we have a high probability of simulating negative values for the volatility with the Euler discretization.

Feller condition satisfied

Because we want to be able to use an Euler discretization as a comparison and test for the implementation of the QE-scheme, we will need to choose our parameters so that $2\kappa\theta \geq \xi^2$. We will take:

- $S_0 = 10$
- $\nu_0 = 0.2$
- $X = 12$
- $T = 1$
- $r = 0.05$
- $\kappa = 0.4$
- $\theta = 0.3$
- $\xi = 0.2$
- $\rho = -0.1$
- $\Psi_c = 1.5$
- $\gamma_1 = \gamma_2 = 0.5$

Note that it holds that $2\kappa\theta = 2 \cdot 0.4 \cdot 0.3 = 0.24 \geq 0.04 = \xi^2$. We will do 30 runs of n=1000 paths, with N=50 timesteps per path. This gives us results found in table 7.1.

Table 7.1: Simulation results MC for Euler and QE scheme, European put

	Mean	**Std. dev.**
Euler	2.7777	0.0755
QE-scheme	2.7587	0.0873

As we can see in table 7.1 the results of both discretization schemes are very similar. The difference between the results is so small (about one fourth of the standard deviation) that these differences can be attributed to random errors. Hence we can assume that the QE-scheme is implemented correctly, we will come back further to the correctness of our methods in section 7.3.3.

Feller condition not satisfied

If we change the value of ξ and take for instance $\xi = 1$ we have that $2\kappa\theta = 0.24 < 1 = \xi^2$ and we no longer have that with the Euler discretization the simulated values of the volatility are positive with high probability. Using the QE-scheme however we should have no trouble finding the option price under these conditions. All other parameters are the same as those used in the experiment in section 7.3.1. The results are found in table 7.2.

Table 7.2: Simulation results MC for Euler and QE scheme, $2\kappa\theta < \xi^2$, European put

	Mean	**Std. dev.**
Euler	$1.2955 - 0.0041i$	0.3017
QE-scheme	2.4510	0.0217

The advantages of the QE-scheme are visible in the results, where the Euler discretization fails to give us a realistic estimation of the option price the QE-scheme still provides us accurate results. This means that using the QE-scheme we can price options under any choice of parameters. There are suggestions for using an adapted version of the Euler scheme, but all of these adaptations include a bias as described in [14].

7.3.2 SGM and ordinary Monte Carlo for a European put

Since we can assume that the QE-scheme is implemented correctly and gives us the desired results we test the SGM for the Heston model. Because we can test the method by comparing results to the results of the ordinary Monte Carlo simulation for the European put we will price a European put using SGM. The parameters we will choose equal to those in section 7.3.1, in addition we choose to use $b = 3$, so that we create $2^3 = 8$ bundles. The reference value for the European put price, i.e. the price from the Monte Carlo simulation, can be found in table 7.1. The results from the SGM for the European put under the Heston model can be found in table 7.3.

Table 7.3: Simulation results SGM with bundling for Heston, European put

	Mean	**Std. dev.**
lower bound	2.7371	0.0785
direct estimator	2.7402	0.0068

At first glance it seems that the results found with the SGM are a little low compared to those found using ordinary Monte Carlo simulations. However the standard deviation of the ordinary Monte Carlo simulations is rather high, around 0.08, which means that we can expect that the results can differ significantly if we would redo this experiment. To check this hypothesis we've redone the experiment with the ordinary Monte Carlo simulation using the QE-scheme and found an estimated option price of 2.7172, which is lower than our SGM estimates. This supports the conclusion that the results for SGM are in fact correct and that the differences from the ordinary Monte Carlo results in table 7.1 may be caused by random errors. In order to gain more certainty on the accuracy of the results we will perform a simulation to compare the value from SGM with reference value from literature in section 7.3.3.

7.3.3 SGM for American put, comparison to literature

In the paper by Fang and Oosterlee, [7], we find some reference values for American put options. We will use the following parameters:

- $S_0 = 8$
- $\nu_0 = 0.0625$
- $X = 10$
- $T = 0.25$
- $r = 0.1$
- $\kappa = 5$
- $\theta = 0.16$
- $\xi = 0.9$
- $\rho = 0.1$

The reference value for the American put option than is 2.0000. We perform 30 runs for the SGM with bundling with $n = 1000$, $N = 50$, $b = 3$ and we assume that we may exercise the option at any timestep. This gives the results in table 7.4.

Table 7.4: Simulation results SGM with bundling for Heston, American put

	Mean	**Std. dev.**
lower bound	1.9962	0.0071
direct estimator	1.9969	0.0045

We see in table 7.4 that the SGM method with the improved bundling algorithm gives us very accurate results compared to the reference value. This tells us that using a limited amount of paths we can already obtain accurate results.

7.3.4 SGM for American put, computation time and accuracy

Since we've confirmed in section 7.3.3 that SGM is implemented correctly for the Heston model using the QE-scheme we can perform an experiment for an American put using SGM, in order to examine accuracy and computation time for different numbers of asset paths and bundles. We will perform an experiment similar to that in section 6.3.3, where we will first take $n = 10^3$ asset paths with subsequently $b = 2$, $b = 3$ and $b = 4$ and then take $n = 10^4$ asset paths with subsequently $b = 2$, $b = 3$, $b = 4$ and $b = 5$. Hence the number of bundles will range from $2^2 = 4$ to $2^4 = 16$ for $n = 10^3$ asset paths and from $2^2 = 4$ to $2^5 = 32$ for $n = 10^4$ asset paths. All other parameters will be equal to the parameters in section 7.3.1, and we assume that the option can be exercised at each timestep. We will also take a look at the computation time per run for each of the scenario's. For each of the scenario's 30 runs will be performed to get accurate estimates for the mean and standard deviation of the results. The results for $n = 10^3$ asset paths are found in table 7.5, the results for $n = 10^4$ asset paths are found in table 7.6.

Table 7.5: Simulation results SGM with bundling for Heston, $n = 10^3$, for American put

		Mean	**Std. dev.**	**Time/run (s)**
$b = 2$	**lower bound**	2.8523	0.0653	3.4975
	direct estimator	2.8523	0.0086	
$b = 3$	**lower bound**	2.8298	0.0738	3.6707
	direct estimator	2.8508	0.0095	
$b = 4$	**lower bound**	2.8328	0.0891	3.9233
	direct estimator	2.8748	0.1461	

Table 7.6: Simulation results SGM with bundling for BSHW, $n = 10^4$, for American put

		Mean	**Std. dev.**	**Time/run (s)**
$b = 2$	**lower bound**	2.8498	0.0197	33.9872
	direct estimator	2.8480	0.0025	
$b = 3$	**lower bound**	2.8409	0.0231	36.8896
	direct estimator	2.8484	0.0018	
$b = 4$	**lower bound**	2.8523	0.0283	35.4220
	direct estimator	2.8472	0.0020	
$b = 5$	**lower bound**	2.8378	0.0186	36.7184
	direct estimator	2.8468	0.0022	

The first thing that stands out is that the computation time is significantly higher than the computation time for the Black-Scholes Hull-White model, as visible in table 6.4 and 6.5. This difference in computation time occurs because of the use of the QE-scheme. The path generation with the QE-scheme is somewhat less efficient than the Euler discretization, which gives us an increase in computation time. An advantage of the QE-scheme could be that you can take fewer timesteps and still get accurate results in order to reduce computational effort. However in our case we are not able to use this aspect of the QE-scheme since decreasing the amount of timesteps is not a viable option when we are pricing American options. Since this is the case SGM has the advantage that we get very accurate results using a relatively small amount of asset paths. We see that in the Heston case the standard deviation is larger than in the Black-Scholes and Black-Scholes Hull-White case (see tables 5.2, 5.3, 6.4 and 6.5) but is still small for such a limited amount of asset paths. The larger standard deviation can be explained from the Heston model, since we introduce a higher degree of uncertainty to the model, resulting in a distribution with fatter tails, making convergence to the actual value somewhat slower. Also it is clear from the results that it is even more important in the Heston case not to choose b to be too big, since this will have a negative influence on the results. If we take $n = 10^3$ asset paths we see that we obtain the best results for $b = 2$, and the results for $b = 3$ are similar, while for $b = 4$ we already get less accurate results. This effect is not visible for $n = 10^4$ asset paths, but we do see that for $n = 10^4$ asset paths the computation time becomes very long compared to the gain in accuracy in the results. In practical situations it thus will be more viable to choose $n = 10^3$ asset paths, since it is very expensive to generate more paths which gives very little gain in accuracy.

Chapter 8

Heston-Hull-White model

In this chapter we will discuss the Heston-Hull-White model. This is a combination of the Black-Scholes Hull-White model as described in chapter 6 and the Heston model described in chapter 7, which leads to a model with stochastic interest rate as well as stochastic volatility. First a mathematical description of the model will be presented, followed by a description of the discretization scheme that we can use for this model and the chapter will be concluded with some simulation experiments for the Heston-Hull-White model.

8.1 Model description

The Heston-Hull-White model is a model where both the interest rate and the volatility are stochastic as described in [15]. This results in a system of three stochastic differential equations, one for the interest rate, one for the volatility and one for the asset price. The dynamics of the interest rate are described by an Hull-White process as described in chapter 6. The dynamics of the volatility are given by a Heston process as described in chapter 7. In order to make computations a little less involved we will here only consider the special case where the correlation between the volatility and the interest rate is equal to zero. Intuitively this is a logical assumption since the probability of a volatile market under both high and low interest rates seems to be equal. Combining the Heston model with the Hull-White process for the interest rate then yields us the system of equations given in (8.1).

$$\begin{cases} dS_t = r_t S_t dt + \sqrt{\nu_t} S_t dW_t^S \\ dr_t = \lambda(\overline{r}_t - r_t)dt + \eta dW_t^r \\ d\nu_t = \kappa(\theta - \nu_t)dt + \xi\sqrt{\nu_t} dW_t^\nu \\ dW_t^S dW_t^\nu = \rho_1 dt \\ dW_t^S dW_t^r = \rho_2 dt \end{cases} \tag{8.1}$$

The parameters are given as described for the Black-Scholes Hull-White model in 6 and the Heston model in 7.
Now for notational convenience we rewrite the dynamics for the asset price. By taking $X_t = \ln(S_t)$, a simple application of Itô's lemma gives us that the system above is equivalent to:

$$\begin{cases} dX_t = (r_t - \frac{1}{2}\nu_t)dt + \sqrt{\nu_t}dW_t^S \\ dr_t = \lambda(\overline{r}_t - r_t)dt + \eta dW_t^r \\ d\nu_t = \kappa(\theta - \nu_t)dt + \xi\sqrt{\nu_t}dW_t^\nu \\ dW_t^S dW_t^\nu = \rho_1 dt \\ dW_t^S dW_t^r = \rho_2 dt \end{cases} \tag{8.2}$$

8.2 Discretization scheme

For the discretization of the dynamics of the Heston-Hull-White model we use a combination of the QE-scheme described in section 7.1 and an Euler discretization. The interest rate process will be discretized using an Euler discretization, the volatility process will be discretized using the QE-scheme described in section 7.1.2. The asset price process will be discretized the same way as with the QE-scheme in section 7.1.3, but here some additional work will have to be done to accomodate the stochastic interest rate. In order to do this we rewrite the dynamics of the Heston-Hull-White model as given in (8.2) using a Cholesky decomposition. This yields the system of equations given in (8.3).

$$\begin{cases} d\nu_t = \kappa(\theta - \nu_t)dt + \xi\sqrt{\nu_t}dW_t^\nu \\ dr_t = \lambda(\overline{r}_t - r_t)dt + \eta dW_t^r \\ dX_t = (r_t - \frac{1}{2}\nu_t)dt + \rho_1\sqrt{\nu_t}dW_t^\nu + \rho_2\sqrt{\nu_t}dW_t^r + \chi\sqrt{\nu_t}dZ_t \end{cases} \tag{8.3}$$

where W_t^ν, W_t^r and Z_t are independent Brownian motions and the constant χ is given by:

$$\chi = \sqrt{1 - \rho_1^2 - \frac{\rho_2^2}{1 - \rho_1^2}}$$

Following the same line of reasoning as in section 7.1.3 we can derive the discretization scheme for the asset price given in equation (8.4).

$$\begin{aligned} \hat{X}(t + \Delta t) = \hat{X}_t &+ r_t\Delta t + \frac{\rho_1}{\xi}\left(\hat{\nu}(t + \Delta t) - \hat{\nu}_t - \kappa\theta\Delta t\right) \\ &+ \Delta t\left(\frac{\kappa\rho_1}{\xi} - \frac{1}{2}\right)\left(\gamma_1\hat{\nu}_t + \gamma_2\hat{\nu}(t + \Delta t)\right) \\ &+ \chi\sqrt{\Delta t}\sqrt{\gamma_1\hat{\nu}_t + \gamma_2\hat{\nu}(t + \Delta t)} \cdot Z_t \\ &+ \rho_2\sqrt{\Delta t}\sqrt{\gamma_1\hat{\nu}_t + \gamma_2\hat{\nu}(t + \Delta t)} \cdot W_t^r \\ = \hat{X}_t &+ K_0 + K_1\hat{\nu}_t + K_2\hat{\nu}(t + \Delta t) + \sqrt{K_3\hat{\nu}_t + K_4\hat{\nu}(t + \Delta t)} \cdot Z_t \\ &+ \rho_2\sqrt{\Delta t}\sqrt{\gamma_1\hat{\nu}_t + \gamma_2\hat{\nu}(t + \Delta t)} \cdot W_t^r, \end{aligned} \tag{8.4}$$

where the constants K_0, K_1, K_2, K_3 and K_4 are given in equation (8.5).

$$\begin{aligned} K_0 &= \left(r_t - \frac{\rho_1 \kappa \theta}{\xi}\right) \Delta t \\ K_1 &= \gamma_1 \left(\frac{\kappa \rho_1}{\xi} - \frac{1}{2}\right) \Delta t - \frac{\rho_1}{\xi} \\ K_2 &= \gamma_2 \left(\frac{\kappa \rho_1}{\xi} - \frac{1}{2}\right) \Delta t + \frac{\rho_1}{\xi} \\ K_3 &= \gamma_1 \left(1 - \rho_1^2\right) \Delta t \\ K_4 &= \gamma_2 \left(1 - \rho_1^2\right) \Delta t \end{aligned} \tag{8.5}$$

8.3 Matlab implementation

The Matlab implementation of the Heston-Hull-White model comes down to combining the implementations of the Heston model described in section 7.2 and the Black-Scholes Hull-White model described in 6.2.2. Using equation (8.5) to discretize the asset price, in combination with the discretization of the QE-scheme for the volatility process and an Euler discretization for the interest rate process we are able to generate paths for the Heston-Hull-White model. The adaptation of the regression entails the merger of the regression step in the Black-Scholes Hull-White case and the regression step in the Heston-case, so that we use both the generated interest rate paths and the generated volatility paths.

8.4 Simulation experiments

We will perform two simulation experiments, first we will check the implementation of the Heston-Hull-White model in SGM by computing the price of a European put using the SGM and using ordinary Monte Carlo to compare the results. Then we will perform a simulation experiment for an American put option using SGM under the Heston-Hull-White model. For both simulation experiments we will use the following parameters:

$$\begin{aligned} &S_0 = 10, X = 12, T = 1, \\ &\nu_0 = 0.2, \kappa = 0.4, \theta = 0.3, \xi = 0.2, \\ &r_0 = 0.05, \lambda = 2, \bar{r} = 0.06, \eta = 0.02, \\ &\rho_1 = -0.1, \rho_2 = 0.1, \\ &\Psi_c = 1.5, \gamma_1 = \gamma_2 = 0.5. \end{aligned}$$

For all SGM simulations we will in addition use $n = 1000$, $N = 50$ and $b = 3$.

8.4.1 SGM and ordinary Monte Carlo for European put

In order to assess the accuracy of the results of SGM we perform a simulation experiment to determine the price of a European put under the Heston-Hull-White model. First we determine a reference value by means of an ordinary Monte Carlo simulation for a European put under the Heston-Hull-White model. We will perform 30 runs of the simulation to determine the mean and standard deviation of the estimated option value. Then we can determine the mean value and standard deviation of the option price obtained using SGM by performing 30 runs of the simulation. The results are found in table 8.1.

Table 8.1: Simulation results MC and SGM for Heston-Hull-White, European put

	Mean	**Std. dev.**
Monte Carlo	2.7294	0.0628
SGM lower bound	2.7208	0.0889
SGM direct estimator	2.7058	0.0134

The results show that we get very accurate results for the European put price using the SGM compared to ordinary Monte Carlo simulations. The estimated value of the option is somewhat lower for SGM than it is with Monte Carlo but the difference is not significant if we take the magnitude of the standard deviation into account. The standard deviation is as expected significantly smaller for the direct estimator of SGM than it is for ordinary Monte Carlo simulations.

8.4.2 SGM for American put

As we have seen in section 7.3.4 the computation time rises significantly if we take $n = 10^4$ asset paths. Since we use the same discretization scheme for the volatility process this will also be the case here, and since so far the gain in accuracy has been very limited if we take $n = 10^4$ asset paths instead of $n = 10^3$ asset paths we will here only consider the case where $n = 10^3$. For this number of asset paths we will compare results on accuracy and computation time if we choose $b = 2$, $b = 3$ and $b = 4$. The results are given in table 8.2.

Table 8.2: Simulation results SGM with bundling for Heston-Hull-White for American put

		Mean	**Std. dev.**	**Time/run (s)**
$b = 2$	**lower bound**	2.8528	0.0627	3.6268
	direct estimator	2.8315	0.0115	
$b = 3$	**lower bound**	2.8529	0.0720	3.8029
	direct estimator	2.8298	0.0136	
$b = 4$	**lower bound**	2.8384	0.0763	4.0898
	direct estimator	3.0524	1.2040	

As in the Heston case we see that if we take $b = 4$ the number of bundles is too large to get accurate results. Also we see a slight increase in computation time compared to the Heston case (see table 7.5), which is to be expected. For $b = 2$ and $b = 3$ we get very accurate results where the standard deviation of the direct estimator in particular is very small, which means that if we use the Heston-Hull-White model the SGM provides us an accurate method to price American options. We see also that the results are as we could expect from the results for the Black-Scholes model in section 5.4, the Black-Scholes Hull-White model in section 6.3.3 and the Heston model in section 7.3.4. There we saw that the price of the option under the Black-Scholes Hull-White model was very similar to the price under the Black-Scholes model, while the price under the Heston model was significantly higher. So it is to be expected that the price under the Heston-Hull-White model is similar to the price under the Heston model, and significantly higher than the price under the Black-Scholes and the Black-Scholes Hull-White model, which is exactly what we see in the results in table 8.2.

Chapter 9

Conclusion

In this final chapter we will return to the research questions stated in the preface of the report. Recall that the first research question was:

"How can we improve the stochastic grid method in order to decrease computational effort?"

By using a bundling strategy, described in chapter 5, where we divide the asset paths into bundles for each timestep and then perform the regression locally for each of the bundles we were able to reduce computation time significantly. Using the original version of the stochastic grid method it took 21.5850 seconds to do a simulation for an American put option using $n = 10^4$ asset paths, under the Black-Scholes model. Using the stochastic grid method with bundling a similar simulation with $n = 10^4$ asset paths only took 1.2165 seconds for $b = 2$, i.e. $2^2 = 4$ bundles, and using $b = 6$, i.e. $2^6 = 64$ bundles, it only took 4.7790 seconds. Meanwhile the standard deviation of the lower bound estimate remained equal, while the standard deviation of the direct estimator even decreased somewhat. Another advantage of the stochastic grid method with bundling is that the estimates for $n = 10^3$ paths already have very small standard deviations, the standard deviation of the direct estimator is significantly smaller than the one obtained using the stochastic grid method without bundling with $n = 10^3$ asset paths.

Besides the very promising simulation results we also gave a formal proof that the results from the stochastic grid method, with our new bundling technique, will asymptotically converge to the true option value.

The second research question posed was:

"How can we adapt the improved stochastic grid method for different models for the asset price?"

After the successful implementation of the new bundling algorithm into the stochastic grid method, the next step was to adapt the algorithm so that asset prices could be driven by dynamics other than Black-Scholes. First we implemented the Black-Scholes Hull-White model where the interest rate process is stochastic, as was described in chapter 6. For this model we could use the analytic solution for a European option as a comparison for our simulation results. The simulation results were very promising. The computation time did increase slightly compared to the Black-Scholes case but not significantly.

Next we worked with the Heston model, described in chapter 7, where the volatility is stochastic. Because under this model it is vital that the volatility remains positive at all times we needed a different discretization scheme known as the QE-scheme, see section 7.1. By using the QE-scheme the method will give accurate results, regardless of the parameters chosen for the Heston model. The use of this new discretization scheme did entail an increase in computation time, since we need the volatility from the previous timestep in the simulation which makes computations less efficient. Fortunately we can obtain very accurate results using only $n = 10^3$ asset paths which means that the computation time can be limited to just over 3 seconds for a simulation.

Combining the resulting methods from the Black-Scholes Hull-White model and the Heston model enabled the implementation of the Heston-Hull-White model, described in chapter 8, where both the interest rate and the volatility are stochastic. For computational convenience we have only considered the case where the correlation between the volatility and the interest rate is equal to 0. Since we desire to be able to use any choice of parameters for the dynamics of the volatility process we used the QE-scheme for discretization of the volatility process, combined with an Euler discretization for the interest rate process. The discretization for the asset price is similar to the one used in the Heston case, adapted to accommodate the stochastic interest rate. For this model we thus saw again a somewhat longer computation time, but not significantly longer than in the Heston case, however using $n = 10^3$ asset paths we can still get accurate results in less than 4 seconds per simulation.

Concluding we were able to improve the stochastic grid method by implementing a bundling technique and performing local regressions rather than global regressions. This way we were able to decrease computation time by 75%, maintaining the same accuracy level. This new stochastic grid method with bundling also proved to be very useful in pricing options using dynamics other than the Black-Scholes dynamics for the asset price. Under the Black-Scholes Hull-White dynamics, the computation time and accuracy remained almost unchanged. Under the Heston and Heston-Hull-White dynamics we see an increase in computation time, because of the different discretization scheme, and a slight increase in the standard deviation of the results due to the fatter tails of the distribution of the asset prices.

Bibliography

[1] Andersen, L., Efficient Simulation of the Heston Stochastic Volatility Model, SSRN eLibrary, 2007

[2] Brigo, D., Mercurio, F., *Interest rate models, theory and practice*, Springer, 2006, Heidelberg

[3] Broadie, M., Glasserman, P, Pricing American-style securities using simulation, *Journal of economic dynamics and control*, Vol. 21, 1997, pp. 1323-1352

[4] Broadie, M., Kaya, Ö, Exact simulation of stochastic volatility and other affine jump diffusion processes, *Operations Research*, Vol. 54, No. 2, 2006, pp. 217-231

[5] Dufresne, D, *The integrated square-root process"*, University of Montreal, 2001

[6] Fang, F, Oosterlee, C.W., A novel pricing method for European options based on Fourier-cosine series expansions, *SIAM Journal on Scientific Computing*, Vol. 31, No.2, 2008, pp. 826-848

[7] Fang, F., Oosterlee, C.W., A Fourier-based Valuation Method for Bermudan and Barrier Options under Heston's Model, *Journal of Financial Mathematics*, Vol 2., 2011, pp. 439-463

[8] Glasserman, P., *Monte Carlo methods in financial engineering*, Springer Verlag, 2003, New York

[9] Jain, S., Oosterlee, C.W., Pricing High-Dimensional American Options Using The Stochastic Grid Method, *Int. J. Comp. Mathematics*, Vol. 89, No. 9, 2012, pp. 1186-1211

[10] Jain, S., Oosterlee, C.W., Stochastic Grid Method with Bundling: Pricing of Multidimensional Bermudan Options, working paper, 2012

[11] Jin, X., Tan, H.H., Sun, J., A State-space Partitioning Method for Pricing High-dimensional American-style Options, *Mathematical Finance*, Vol. 17, No.3, 2007, pp. 399-426

[12] Kendall, M.G., Stuart, A. *The advanced theory of statistics*, Hafner Publishing Company, Vol. 1, 1969

[13] Longstaff, F.A., Schwartz, E.S., Valuing American Options by Simulation: A Simple Least-Squares Approach *The Review of Financial Studies*, Vol. 14, No. 1., 2001, pp. 113-147

[14] Lord, R., Koekkoek, R., van Dijk, D., A comparison of biased simulation schemes for stochastic volatility models, *Quantatitive Finance*, Vol. 10, No. 2, 2010, pp. 177-194

[15] Seydel, R.U., *Tools for computational finance*, Springer, 2010, London

[16] Tilley, J.A., Valuing American Options in a Path Simulation Model, *Trans. Soc. Actuaries*, Vol. 45, 1993, pp. 83-104

[17] Vasicek, O, An equilibrium characterization of the term structure, *Journal of Financial Economics*, Vol. 5, No. 2, 1977, pp. 177188