On Path-Dependent Option Pricing for the Heston Model

by

Huiting Hu

A thesis submitted in partial fulfillment of the requirements for the degree of

Masters of Science

in

Statistics

Department of Mathematical and Statistical Sciences

University of Alberta

Abstract

In this thesis, we are focusing on developing an efficient simulation algorithm to price the path-dependent options, which remains a challenging problem in derivatives finance. The Heston model, a widely used stochastic volatility model, will first be introduced. Then, we will discuss and evaluate several methods used in simulating the Heston model, including the Explicit and Weighted Heston simulation algorithm. The research will be extended to the path-dependent option pricing with the simulation results of the Heston model. The least squares Monte Carlo approach and its favorable alternative method, stochastic approximation, will be explained and compared. Finally, we will introduce the branching algorithm to improve the pricing scheme. Numerical results for pricing different kinds of path-dependent options will show the performance of the branching stochastic approximation algorithm is orders of magnitude better in pricing options than the traditional method.

Acknowledgment

First of all, I would like to extend my sincere gratitude to my supervisor Dr. Michael Kouritzin. He provided me with a lot of valuable guidance on understanding the difficult theory and programming the model. I am deeply grateful of his help in the completion of this thesis.

Also, I am thankful to my classmates Chi and Xingpu. They helped me a lot on theoretical questions and specific coding issues.

I would finally like to express my gratitude to my beloved parents who have always been helping out of difficulties and supporting me without a word of complaint

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In section 1.1, we will review the history and development of the stochastic volatility models and their simulation method. In section 1.2, a brief introduction to the path-dependent option valuation, especially the dynamic programming scheme will be given. In section 1.3, we will introduce the development of particle filters and review the theory involved within solving the problems in this thesis. Finally in section 1.4, we will describe the motivation and the main contributions of this thesis. Also, the outline of this thesis will be presented.

1.1 Stochastic Volatility and the Heston Model

The behavior of financial markets can be modeled by stochastic differential equation(SDE), which describe the motion of the underlying assets such as stock prices. The geometric Brownian motion(GBM), utilized in the Black-Sholes model, is the most widely used model for describing stock prices behavior. A stochastic process S_t is regarded to follow a geometric Brownian motion if it satisfies

$$dS_t = \mu S_t \, dt + \kappa S_t \, dB_t, \tag{1.1}$$

where B is a standard normal Brownian motion and μ, κ denotes the constant drift and volatility. Despite having an analytic solution, the GBM model has overly simple assumptions such as constant volatility and interest rate. Since the volatility always changes over time, a number of works have been done to relax the constant volatility assumption by making the volatility stochastic (see [1],[2], and [3]).

By far, the most popular model within a class of stochastic volatility model was introduced by Heston[4]. It has the advantages of giving the closed-form European call option. Let B and β be two independent standard Brownian motions. The Heston model can be formulated as

$$d\begin{pmatrix} S_t \\ V_t \end{pmatrix} = \begin{pmatrix} \mu S_t \\ \nu - \varrho V_t \end{pmatrix} dt + \begin{pmatrix} \sqrt{1-\rho^2} S_t V_t^{\frac{1}{2}} & \rho S_t V_t^{\frac{1}{2}} \\ 0 & \kappa V_t^{\frac{1}{2}} \end{pmatrix} \begin{pmatrix} dB_t \\ d\beta_t \end{pmatrix}, \quad (1.2)$$

with $\mu \in \mathbb{R}$, $\nu, \varrho, \kappa > 0$ and $\rho \in [-1, 1]$, which represents the instantaneous correlation between the stock process and the variance process. As the volatility component in (1.2) is the Cox-Ingersoll-Ross model, if the model parameters do not satisfy $\nu \geq \frac{\kappa^2}{2}$ (known as Feller condition), then V_t process can hit 0 and result in the failure of taking square root of a negative number. As suggested in [5], the Heston model performs superior than other well-known stochastic volatility model on real data.

Despite the fact that the Heston model was introduced more than 20 years, the simulation for its process dynamic using the discretization method still remains inefficient. Also, there are few researches on providing any form of the explicit solution directly to the Heston model. These limit the use of Heston model especially when one considers pricing and hedging a path dependent derivatives such as American and Asian option. However, several important breakthroughs have been proposed recently.

First of all, instead of the traditional Euler-Maruyama and Milstein approximation scheme, Broadie and Kaya [6] developed a bias-free exact method for the Heston by simulating from its exact distribution. Although theoretically attractive, its use is limited by the complicated implementation. For example, the algorithm involves a Fourier inversion of the conditional characteristic function of the integrated volatility distribution. The complex calculation also becomes slow as time increases.

Then, Kouritzin[7] considered an explicit weak solution to the Heston model with some restriction on the model parameters and furthered the result to the general cases by re-weighting the outcomes of the so called Closest Explicit Heston. The simulation based on the explicit weak solution and weighted Heston solution is relatively easy and will be widely used in pricing heavily-path-dependent options.

1.2 Valuing Path-dependent Options via Dynamic Programming and Simulation

In this section, we will focus on a more practical problem of valuing the options. Unlike pricing the European option, which only requires the terminal value (except for the cases of stochastic volatility with jump model), pricing path-dependent options, such as American option, require the entire simulation path, leading the problem to be more complicated. Traditionally, it can be achieved by using the finite difference method suggested by Schwartz [8] and Howison[9]. Also, the binomial tree pricing model originally proposed by Cox, Stephen and Ross[10] provides an executable way in pricing the options that can be exercised at any time prior to the expiration. However, these methods will be computationally expensive or fail for the dimensionality issue, when being applied to the models with multiple factors or jumps. An alternative is to use the Monte Carlo method and dynamic programming to find the optimal exercise time and calculate the option price. The most successful and popular least squares Monte Carlo(LSM) method is introduced by Longstaff and Schwartz[11] and further analyzed by Clément et. al. [12]. We will also explain this classical algorithm in detail in section 3.1.

Suppose there is a complete filtered risk-neutral probability space $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t=0}^T, P)$ supporting a Markov chain $\{(S_t, V_t)\}_{t=0}^T$ with state space $D = D_S \times D_V$, representing the price and volatility process. We denote the discounted payoff process as $\{Z_t,\ t \geq 0\}$ where Z_t represents the discount payoff received for executing the option at time t. For example, $Z_t = e^{-\mu t}(K - S_t) \vee 0$ for an American put. Then, the goal of option pricing is to compute option value $\sup_{\tau \in \mathcal{T}_{0,T}} \widehat{E}[Z_\tau]$, where $\mathcal{T}_{t,T}$ denotes the collection of stopping times with values in $\{t, t+1, ..., T\}$. In order to get the optimal stopping time $\tau_0 \in \mathcal{T}_{0,T}$, we can use the dynamic programming in this case. Clément E.[12] found that the best stopping time can be found by working backwards:

$$\begin{cases} \tau_T &= T \\ \tau_t &= t 1_{\{Z_t \geq E[Z_{\tau_{t+1}}|\mathcal{F}_t]\} \cap \{Z_t > 0\}} + \tau_{t+1} 1_{\{Z_t < E[Z_{\tau_{t+1}}|\mathcal{F}_t]\} \cup \{Z_t = 0\}} \ \forall\ t < T \end{cases}. \tag{1.3}$$

However, how to calculate these conditional expectations to obtain the stopping time remains unknown. Therefore, Longstaff and Schwartz[11] suggested employing the projection and Monte Carlo techniques. By projecting the conditional expectation onto the closed linear span, we can rewrite the equation (1.3) in terms of the number J of the projection function used:

$$\begin{cases} \tau^J_T &= T \\ \tau^J_t &= t 1_{\{Z_t \geq P^J_t[Z_{\tau^J_{t+1}}]\} \cap \{Z_t > 0\}} + \tau^J_{t+1} 1_{\{Z_t < P^J_t[Z_{\tau^J_{t+1}}]\} \cup \{Z_t = 0\}} \ \forall\ t < T \end{cases}. \tag{1.4}$$

where $P^J_t[Z_{\tau^J_{t+1}}]$ is computed by

$$P^J_t[Z_{\tau^J_{t+1}}] = \alpha^J_t \cdot e^J(S_t, V_t) \tag{1.5}$$

To ensure the projection and Monte Carlo work, Clément E.[12] proposed the following assumptions:

Total there are measurable $\mathbb{R}$-valued functions $(e_k)_{k=1}^{\infty}$ on D such that $\{e_k(S_t, V_t)\}_{k=1}^{\infty}$ is total on $L^2(\sigma(S_t, V_t), 1_{\{Z_t>0\}} d\widehat{P})$ for all $t = 1, ..., T-1$.

Non-singular $\widehat{E}[e^J(S_t, V_t)(e^J(S_t, V_t))' 1_{\{Z_t>0\}}]$ is positive definite for all $J \in \mathbb{N}$, where $e^J = (e_1, ..., e_J)'$.

The key idea of the LSM algorithm is to estimate the coefficients α^J_t

by using cross-section of Monte Carlo and least-squared regression. However, Kouritzin[7] showed a favorable alternative, applying the stochastic algorithm(SA) with the coefficient estimations, which avoids the messy matrix inversion step. The SA algorithm will also be analyzed in section 3.2.

1.3 Development and Theory of Particle Filter Method

As mentioned in the end of section 1.1, Kouritzin [7] introduced the weighted Heston solution to the general Heston model. Although it will be carefully explained in section 2.2, we want to state here that the simulation for the option price using weighted Heston can be regarded as the analog of a weighted particle filter in sequential Monte Carlo.

The weighted particle filter can be used as an approximation of the unnormalized filters, a solution to the non-linear filtering problems. The non-linear filtering problem refers to estimating the current state of a non-observable random signal X given the history of a distorted, corrupted partial observation process Y living on the same probability space $(\Omega, \mathcal{F}, P)$ as X. The main object of the non-linear filtering is to calculate the conditional probabilities:

$$\pi_k(A) = P\left(X_k \in A \,\middle|\, \mathcal{F}_k^Y\right), k = 1, 2, ..., \tag{1.6}$$

for all Borel sets A where $\mathcal{F}_k^Y \doteq \sigma\{Y_l, l = 1, ..., k\}$ contains all the information from the previous observations. This can be solved by the unnormalized filter approach. Under the notion of the unnormalized filter, the conditional probability stated in equation 1.6 can be expressed by:

$$\pi_k(f) = \frac{\sigma_k(f)}{\sigma_k(1)}, k = 1, 2..., \tag{1.7}$$

where $\sigma_k(f)$ is the unnormalized filter defined by:

$$\sigma_k(f) = E^Q \left(L_k f\left(X_k\right) \middle| \mathcal{F}_k^Y \right), k = 1, 2... \tag{1.8}$$

Finally, to get the filters implemented on computers, we can numerically approximate the unnormalized filter by sequential Monte Carlo method such as the weighted particle filter. A particle filter estimates the unnormalized filter $\sigma_k(f)$ by

$$\sigma_k^N(f) = \sum_{i=1}^{N} L_k^i f(x_k^i), k = 1, 2, ..., \tag{1.9}$$

where N is the number of particles and initial particles $\{x_0^i\}_{i=1}^N$ are drawn independently from the initial distribution of X_n.

Since the conditional probability $\pi_k(f)$ in equation (1.9) is recovered form the unnormalized, we can find particle filter estimates for $\pi_k(f)$ by

$$\pi_k^N(f) = \frac{\sigma_k^N(f)}{\sigma_k^N(1)} = \frac{\sum\limits_{i=1}^{N} L_k^i f(x_k^i)}{\sum\limits_{i=1}^{N} L_k^i}, \tag{1.10}$$

where N is the number of particles.

Although the weighted particle filter can be applied to approximate the unnormalized filter, it often suffers from the particle spread. Gorden et al.[13] discovered the problem can be solved by simply redistributing particles independently to the same location according to their relative likelihood. They named the algorithm as the bootstrap filter and was later improved by the residual resampling by Liu and Chen [14], stratified resampling by Kitagawa G. [15], combined residual-stratified resampling discussed in [16] and systematic resampling proposed in [17]. Then a different approach was taken in [18], which has been shown to outperform to the resampled method mentioned above.

1.4 Contributions and Outline

Pricing path-dependence options by simulation, especially for the Heston model, remains a difficult problem for the following reasons: For the simulation of Heston model, the existing methods such as Euler and Milstein are highly time consuming that they will not be the proper choices for pricing especially when T is large. Secondly, the popular tool for valuing the path-dependent option, the LSM method, will usually fail when seeking higher accuracy or pricing with a multi-factor model due to the ill-conditioned matrix inversion. Therefore, we consider solving these two problems simultaneously and providing a more effective as well as accurate algorithm for pricing options.

The main contribution of this thesis are that with the support of my supervisor, Dr. Michael Kouritzin, we develop a new approach to simulate the Heston model and empirically show that the Explicit and Weighted Heston simulation algorithm is far superior to the traditional discretization method in computation efficiency. Besides, we prove the failure of LSM algorithm based on the experiment and demonstrate the advantages of using the stochastic approximation scheme for pricing options. Finally, we first realize the sequential Monte Carlo in pricing. In particular, we introduce a class of branching method into price evaluation. By simulation results, we show the pricing performance for Branching-SA is orders of magnitude better than the traditional methods.

In Chapter 2, we will first review the traditional simulation method to the Heston model. Then we will put emphasis on the newly proposed explicit and weighted solution to the Heston model and provide the algorithm. By comparing the performance and execution time, the explicit and weighted Heston simulation is so much better than the existing method. In Chapter 3, focusing on the algorithm of pricing the path-dependent options, we will recall both the well known LSM method and its favorable alternative, stochastic approximation. Then we will extend and further our works by employing the branching particle system. In Chapter 4, numerical comparison for the performance and execution time of the method discussed in Chapter 3 will be provided. A parameter for comparing the efficiency of the algorithm, the "Performance Factor" will be defined and used to show that branching with stochastic approximation algorithm performs much faster than the tradition pricing method. For the convenience of readers, we put all the proofs and calculations in Chapter 5, the Appendix.

Chapter 2

Simulation of the Heston Model

In section 2.1, we will review the Euler and Milstein scheme for the Heston model and discuss its limitation. Then in section 2.2, we will state and explain the results given in [7] for Explicit and Weighted solution to the Heston model. The algorithm and the implementation method will also be provided in this section. Finally in section 2.3, a numerical comparison will be presented to show the significant time advantage gained by using the Explicit simulation algorithm.

2.1 Discretization method of Heston Model

2.1.1 Euler Scheme

The most popular and simplest way to approximate the paths of both the stock price and the stochastic variance process is by applying the first order Euler scheme. Partition time T into M equal length intervals, so $\Delta t = \frac{1}{M}$. The Euler discretization of the stochastic variance process is

$$V_t = V_{t-1} + (\nu - \varrho V_{t-1})\Delta t + \kappa\sqrt{V_{t-1}}\Delta\beta_t. \tag{2.1}$$

The discretization for the stock price process is

$$S_t = S_{t-1} + \mu S_t \Delta t + \sqrt{1-\rho^2} S_{t-1} V_{t-1}^{\frac{1}{2}} \Delta B_t + \rho S_{t-1} V_{t-1}^{\frac{1}{2}} \Delta \beta_t. \tag{2.2}$$

Using the independent and Gaussian increments properties for the Brownian motion, we can simulate ΔB_t and $\Delta \beta_t$ by generating a standard normal random variable N and replacing the increment by $\sqrt{\Delta t} N$. To be implemented on computer, there are other efficient algorithm for generating the standard normal random variables such as applying the Box-Muller transformation method. If U_1 and U_2 are two independent uniform random variables on $(0,1)$, then Z_1 and Z_2 are independently distributed according to the standard normal distribution if

$$Z_1 = \sqrt{-2 \ln U_1} cos 2\pi U_2$$
$$Z_2 = \sqrt{-2 \ln U_1} sin 2\pi U_2$$

2.1.2 Milstein Scheme

An alternative way to simulate the Heston model is using the 2-dimension Milstein Scheme. The basic 2-dimension Milstein Scheme for Heston is

$$V_t = V_{t-1} + (\nu - \varrho V_{t-1}) \Delta t + \kappa \sqrt{V_{t-1}} \Delta \beta_t + \frac{1}{4} \kappa^2 (\Delta \beta_t^2 - \Delta t) \tag{2.3}$$

$$\begin{aligned} S_t = S_{t-1} &+ \mu S_t \Delta t + \sqrt{1-\rho^2} S_{t-1} V_{t-1}^{\frac{1}{2}} \Delta B_t + \rho S_{t-1} V_{t-1}^{\frac{1}{2}} \Delta \beta_t \\ &+ \frac{\kappa}{2} (\sqrt{1-\rho^2} \tilde{I}_{(2,1)} + \frac{\rho}{2} (\Delta \beta_t^2 - \Delta t)) \end{aligned} \tag{2.4}$$

where

$$\tilde{I}_{(2,1)} = \int_{t-1}^{t} \int_{t-1}^{t} d\beta_t dB_t \tag{2.5}$$

To simulate the double Wiener integral, we have to introduce several additional random numbers which will bring more noise to the method. Kahl and Jackel [19] suggested another integration scheme based on interpolation of the drift and diffusion term, as well as incorporated the decorrelation of the diffusion term and included the Milstein scheme. The so called IJK scheme for stock price process reads

$$\begin{aligned} lnS_t = lnS_{t-1} + \mu\Delta t - \frac{1}{4}(V_{t-1} + V_t)\Delta t + \rho V_{t-1}^{\frac{1}{2}}\Delta\beta_t \\ + \frac{1}{2}(V_{t-1}^{\frac{1}{2}} + V_t^{\frac{1}{2}})(\Delta B_t - \rho\Delta\beta_t) + \frac{1}{4}\rho(\Delta\beta_t - \Delta t) \end{aligned} \tag{2.6}$$

Kahl has shown in their paper that the IJK method is superior in convergence behavior than the Euler method and does not require additional random numbers or aids for convergence acceleration.

2.1.3 Failure of the discretization method

From (2.2) and (2.3), both the Euler and Milstein methods need to handle the negative volatility values or we can not take square root in both price and variance processes. The fix under wide use is called the Full Truncation scheme. By setting the negative volatility to zero, the algorithm can work but more bias is introduced. Observed by Bordie and Kaya, when $\frac{2\nu}{\kappa} \leq 1$, the bias of the Euler discretization will stay large even with a very fine time step size. Therefore, it is essential to study how often the Euler and Milstein schemes will produce a negative volatility under certain parameter settings.

We will call a simulation where a *negative* volatility is produced a *failure* and define the *break time* as the first time this occurs. Suppose we have the Heston model with the parameters as the following: $\mu = 0.0319, \rho = -0.7, \varrho = 6.21, \kappa = 0.61$ and $\nu = \frac{\kappa^2}{4}$ and the initial state is $S_0 = 100, V_0 = 0.010201$. As $\nu < \frac{\kappa^2}{2}$, the Feller condition does not hold and the volatility can hit zero but not drop below zero.

The relative breaking frequency of Euler and Milstein simulations in this case are shown in Tables 1 and 2 below.

Scheme	Euler		Milstein	
N	$10,000$	$40,000$	$10,000$	$40,000$
Steps	100	200	100	200
T$= 1$	0.972386	0.972184	0.932158	0.914071
T$= 2$	0.026434	0.025734	0.062245	0.077341
T$= 3$	0.001134	0.001033	0.005166	0.007731
T$= 4$	0.000045	0.0000465	0.000394	0.000777
T$= 5$	0.000001	0.0000025	0.000037	0.0000713
T$= 50$	0	0	0	0

Table 1: Relative breaking frequency for $\nu = \frac{\kappa^2}{4}, \kappa = 0.61, \varrho = 6.21$

As shown in the result, none of the particles survived till the end of $T = 50$ for both Euler and Milstein. In other words, simulating the Heston model directly using the traditional discretization method is unfeasible when $\nu < \frac{\kappa}{2}$.

One might think that this only happens when the volatility is supposed to hit zero. However, increasing ν to $\frac{\kappa^2}{2}$, which is the critical or first case that the volatility should not hit 0, we still encounter the same problem, especially for Euler scheme.

Scheme	Euler		Milstein	
N	$10,000$	$40,000$	$10,000$	$40,000$
Steps	100	200	100	200
T$= 1$	0.802964	0.767827	0.000492	0
T$= 2$	0.147584	0.165	0.000488	0
T$= 3$	0.037084	0.0.047847	0.000506	0
T$= 4$	0.009277	0.013768	0.000524	0
T$= 5$	0.002313	0.003941	0.000484	0
T$= 50$	0	0	0.976822	1

Table 2: Relative breaking frequency for $\nu = \frac{\kappa^2}{2}, \kappa = 0.61, \varrho = 6.21$

For $\nu = \frac{\kappa^2}{2}$, we see that Milstein scheme with 200 steps works well while the Euler will suffer from hitting zero in simulation.

We further increased the ν to κ^2 and found that with Euler scheme, there is still large number of particles going negative in this case.

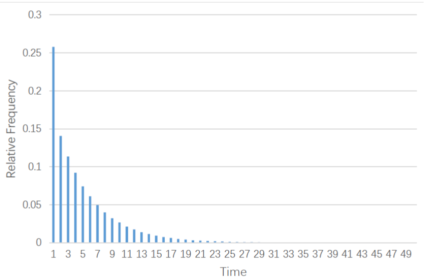

Figure 1: Euler with N=10,000, Step Size=100

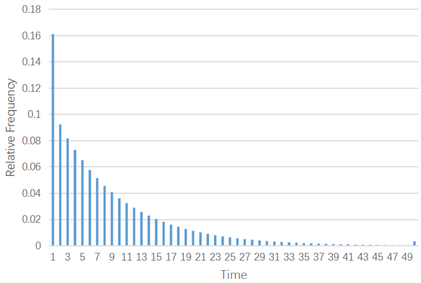

Figure 2: Euler with N=40,000, Step Size=200

This figure shows how often the Euler will actually hit zero and algorithm will fail as time evolves. With 100 steps, we found that none of the 10,000 particles survive until the end.

2.2 Explicit and Weighted Solution of Heston

Whereas the failure and the slow convergence rate of the Euler and Milstein, [7] showed that the Heston SDEs were actually explicitly solvable in a way that is very convenient for simulation, which will boost the efficiency of path-dependent option pricing.

2.2.1 Explicit Solution

Discovered in the companion paper of Kouritzin and Remillard [20] that a necessary and sufficient condition for the Itô SDE

$$dX_t = b(X_t)dt + \sigma(X_t)dW_t, \tag{2.7}$$

to have a strong solution with such an explicit representation locally (for some drift coefficient b) is the diffusion coefficient columns σ_j satisfy the Lie bracket condition:

$$(\nabla\sigma_i)\sigma_j = (\nabla\sigma_j)\sigma_i \quad \forall i, j. \tag{2.8}$$

Unfortunately, the Heston model does not satisfy (2.8) since

$$(\nabla\sigma_1)\sigma_2 = \begin{pmatrix} sv\rho\sqrt{1-\rho^2} + \frac{s\kappa\sqrt{1-\rho^2}}{2} \\ 0 \end{pmatrix} \neq \begin{pmatrix} sv\rho\sqrt{1-\rho^2} \\ 0 \end{pmatrix} = (\nabla\sigma_2)\sigma_1$$

when $\sigma = (\sigma_1\,\sigma_2) = \begin{pmatrix} \sqrt{1-\rho^2}sv^{\frac{1}{2}} & \rho sv^{\frac{1}{2}} \\ 0 & \kappa v^{\frac{1}{2}} \end{pmatrix}$, where s and v represent the state variables for price and variance. Hence, weak solutions have to be considered to get an explicit representation for the Heston SDEs.

However, it is proved by Kouritzin [21] that scalar SDEs only have explicit solutions for specific drift coefficients. Therefore the explicit solution has to be restricted under certain condition. As stated in Kouritzin's paper [7], when the following Condition (C) holds, there exists a weak solution for the Heston model.

C $\nu = \frac{n\kappa^2}{4}$ for some $n = 1, 2, 3, ...$

Theorem 2.1. *Suppose $n \in \{1, 2, 3, 4, ...\}$, Condition (C) holds and $W^1, ..., W^n, B$ are independent standard Brownian motions. Then, the Heston (price and volatility) model (1.2) has explicit weak solution:*

$$\begin{aligned} S_t = S_0 \exp\bigg(&\sqrt{1-\rho^2}\int_0^t V_s^{\frac{1}{2}} dB_s + \left[\mu - \frac{\nu\rho}{\kappa}\right] t \\ &+ \left[\frac{\rho\varrho}{\kappa} - \frac{1}{2}\right]\int_0^t V_s ds + \frac{\rho}{\kappa}(V_t - V_0)\bigg), \end{aligned} \tag{2.9}$$

$$V_t = \sum_{i=1}^n (Y_t^i)^2, \tag{2.10}$$

where $\{Y_t^i = \frac{\kappa}{2}\int_0^t e^{-\frac{\varrho}{2}(t-u)} dW_u^i + e^{-\frac{\varrho}{2}t} Y_0^i\}_{i=1}^n$ are Ornstein-Uhlenbeck processes and

$$\beta_t = \sum_{i=1}^n \int_0^t \frac{Y_u^i}{\sqrt{\sum_{j=1}^n (Y_u^j)^2}} dW_u^i$$

is the other Brownian motion appearing in (1.2).

The above theorem establishes a weak solution to the Heston model. Also, instead of applying the discretization method, it provides an explicitly computable way to simulate the Heston. The advantage for simulating through the explicit solution is quite obvious. Comparing with the traditional discretization methods, it eliminates the discrete bias and significantly improves the efficiency. Rather than sampling for the stochastic integrals, the explicit solution only requires the users to compute two deterministic integrals. The stochastic integral in (2.9) is conditionally Guassian given V and since V and B are independent, the simulation is only a normal random variable with mean 0 an variance $\sqrt{1-\rho^2}\int_0^t V_s^{\frac{1}{2}}ds$. Besides, we can sample the Ornstein-Uhlenbeck Process easily since

$$Y_t = \frac{\kappa}{2}\int_0^t e^{-\frac{\varrho}{2}(t-u)}dW_u + e^{-\frac{\varrho}{2}t}Y_0 \sim \mathcal{N}(e^{-\frac{\varrho}{2}t}Y_0, \frac{\kappa^2(1-e^{-\varrho})}{4\varrho}) \tag{2.11}$$

2.2.2 Weighted Solution

Though the explicit solution is an efficient way to simulate the Heston model, its merit is restricted by the specific parameters required by Condition (C). To handle the general Heston model, Kouritzin [7] proposed a solution by re-weighting the outcomes of the so-called *Closest Explicit Heston*, which is defined as:

$$d\begin{pmatrix} \widehat{S}_t \\ \widehat{V}_t \end{pmatrix} = \begin{pmatrix} \mu_\kappa \widehat{S}_t \\ \nu_\kappa - \varrho\widehat{V}_t \end{pmatrix} dt + \begin{pmatrix} \sqrt{1-\rho^2}\widehat{S}_t\widehat{V}_t^{\frac{1}{2}} & \rho\widehat{S}_t\widehat{V}_t^{\frac{1}{2}} \\ 0 & \kappa\widehat{V}_t^{\frac{1}{2}} \end{pmatrix} \begin{pmatrix} dB_t \\ d\widehat{\beta}_t \end{pmatrix}, \tag{2.12}$$

where $n = \lfloor \frac{4\nu}{\kappa^2} + \frac{1}{2} \rfloor \vee 1, \nu_\kappa = \frac{n\kappa^2}{4}, \mu_\kappa = \mu + \frac{\varrho}{\kappa}(\nu_\kappa - \nu)$. In this case, the Condition (C) holds and the model can be explicitly solved.

Theorem 2.2. *Let $\varepsilon \in (0,1)$, $T > 0$, $(\Omega, \mathcal{F}, \{\mathcal{F}\}_{t\in[0,T]}, P)$ be a filtered probability space, V_0, S_0 be given random variables with $V_0 > \varepsilon$, $\{W^1, ..., W^n, B\}$ be* *independent standard Brownian motions with respect to* $(\Omega, \mathcal{F}, \{\mathcal{F}\}_{t\in[0,T]}, P)$,

$$\begin{aligned}\widehat{S}_t =& S_0 \exp\bigg(\sqrt{1-\rho^2}\int_0^t \widehat{V}_s^{\frac{1}{2}} dB_s + \Big[\mu - \frac{\nu\rho}{\kappa}\Big]t + \Big[\frac{\rho\varrho}{\kappa} - \frac{1}{2}\Big]\int_0^t \widehat{V}_s ds \\ & + \frac{\rho}{\kappa}(\widehat{V}_t - \widehat{V}_0)\bigg)\end{aligned} \tag{2.13}$$

$$\widehat{V}_t = \sum_{i=1}^{n} (Y_t^i)^2, \quad \eta_\varepsilon = \inf\Big\{t : \widehat{V}_t \leq \varepsilon\Big\} \quad \textit{and} \tag{2.14}$$

$$\widehat{L}_t = \exp\left\{\frac{\nu - \nu_\kappa}{\kappa^2}\left[\ln(\widehat{V}_t) - \ln(\widehat{V}_0) + \int_0^t \frac{\kappa^2 - \nu_\kappa - \nu}{2\widehat{V}_s} + \varrho\, ds\right]\right\}, \tag{2.15}$$

where $Y_t^i = \frac{\kappa}{2}\int_0^t e^{-\frac{\varrho}{2}(t-u)} dW_u^i + e^{-\frac{\varrho}{2}t} Y_0^i$ *for* $i = 1, 2, ..., n$. *Define*

$$\beta_t = \sum_{i=1}^{n} \int_0^t \frac{Y_u^i}{\sqrt{\sum_{j=1}^n (Y_u^j)^2}} dW_u^i + \int_0^{t\wedge\eta_\varepsilon} \frac{\nu - \nu_\kappa}{\kappa \widehat{V}_s^{\frac{1}{2}}} ds, \quad \textit{and} \tag{2.16}$$

$$\widehat{P}(A) = E[1_A \widehat{L}_{T\wedge\eta_\varepsilon}] \ \ \forall A \in \mathcal{F}_T. \tag{2.17}$$

Then, η_ε *is a stopping time and* $\widehat{L}_{t\wedge\eta_\varepsilon}$ *is a* L^r*-martingale with respect to* P *for any* $r > 0$*. Moreover,* $(B,\, \beta)$ *are independent standard Brownian motions and*

$$d\begin{pmatrix} \widehat{S}_t \\ \widehat{V}_t \end{pmatrix} = \begin{cases} \begin{pmatrix} \mu \widehat{S}_t \\ \nu - \varrho \widehat{V}_t \end{pmatrix} dt + \begin{pmatrix} \sqrt{1-\rho^2}\widehat{S}_t \widehat{V}_t^{\frac{1}{2}} & \rho \widehat{S}_t \widehat{V}_t^{\frac{1}{2}} \\ 0 & \kappa \widehat{V}_t^{\frac{1}{2}} \end{pmatrix} \begin{pmatrix} dB_t \\ d\beta_t \end{pmatrix}, & t \leq \eta_\varepsilon \\ \begin{pmatrix} \mu_\kappa \widehat{S}_t \\ \nu_\kappa - \varrho \widehat{V}_t \end{pmatrix} dt + \begin{pmatrix} \sqrt{1-\rho^2}\widehat{S}_t \widehat{V}_t^{\frac{1}{2}} & \rho \widehat{S}_t \widehat{V}_t^{\frac{1}{2}} \\ 0 & \kappa \widehat{V}_t^{\frac{1}{2}} \end{pmatrix} \begin{pmatrix} dB_t \\ d\beta_t \end{pmatrix}, & t > \eta_\varepsilon \end{cases} \tag{2.18}$$

on $[0, T]$ *with respect to* $\widehat{P}$.

This theorem provides a way to produce weighted particles of the desired Heston model until the volatility drops below ε and then falls back to the closest explicit model after hitting the stopping rule.

Notation: We are using $\widehat{S}, \widehat{V}$ for solutions to the *closest explicit* Heston model under P, reserving S, V for the solution to (2.18). Henceforth, we will use $\widehat{\beta}_t = \sum\limits_{i=1}^{n} \int_0^t \frac{Y_u^i}{\sqrt{\sum_{j=1}^n (Y_u^j)^2}} dW_u^i$ and $\beta_t = \widehat{\beta}_t + \int_0^{t\wedge\eta_\varepsilon} \frac{\nu - \nu_\kappa}{\kappa \widehat{V}_s^{\frac{1}{2}}} ds$.

It is important to stop (at η_ε) before the volatility gets too small. To illustrate this, we consider the situation where the volatility $V_t^{\frac{1}{2}} = 0$. Then, the (closest explicit and general) Heston volatility equations become deterministic

$$d\widehat{V}_t = \nu_\kappa dt, \qquad dV_t = \nu dt$$

and it is obvious which solution one has. This makes model distributions singular to each other when $\nu_k \neq \nu$. Actually, due to the randomness within the model, the simulation outcomes from the closest explicit Heston model can actually match the desired Heston. However, when the volatility drops too low or even hits 0, there is not enough randomness that simulations of the closest explicit Heston would look like the correct model.

In terms of path-dependent option pricing, we first simulate many Heston particles $\{(S_t^i, V_t^i),\ t \geq 0\}_{i=1}^N$ from the closest explicit Heston model and use the relative likelihood $L_t^i = \frac{\widehat{P}^i}{P}\Big|_{\mathcal{F}_t}$ as the particle weight to denote the probability that the results are from the right Heston model. These likelihoods can be used in lieu of the observation likelihoods (that convert from a canonical noise process to the actual observations) in sequential Monte Carlo. In this case, the closest explicit Heston model plays the role of the signal and the simulation for option pricing can be thought of as the analog, $\sigma_{[0,t]}^N = \dfrac{1}{N}\displaystyle\sum_{i=1}^N L_t^i \delta_{(S_{[0,t]}^i, V_{[0,t]}^i)}$, of a weighted particle filter. Here, $\delta_{(S_{[0,t]}^i, V_{[0,t]}^i)}$ denotes Dirac measure and $(S_{[0,t]}^i, V_{[0,t]}^i)$ denotes the path of the i^{th}-particle's price and volatility over $[0,T]$ but held constant from t on.

2.2.3 Explicit and Weighted Heston Simulation Algorithm

Before presenting the algorithm, we first define some constants to keep things simple

$$a = \sqrt{1-\rho^2},\ b = \mu - \frac{\nu\rho}{\kappa},\ c = \frac{\rho\varrho}{\kappa} - \frac{1}{2},\ d = \frac{\rho}{\kappa},\ e = \frac{\nu - \nu_\kappa}{\kappa^2},\ f = e\frac{\kappa^2 - \nu - \nu_\kappa}{2},$$

To compute the two deterministic integrals, we can apply the following numerical approximation method

$$\text{Trapezoidal} \int_{t-1}^{t} \widehat{V}_s ds \approx \frac{1}{2M}\left\{\widehat{V}_{t-1} + \widehat{V}_t + 2\sum_{l=1}^{M-1} \widehat{V}_{t-\frac{l}{M}}\right\}$$

$$\text{Simpson's } \tfrac{1}{3} \int_{t-1}^{t} \widehat{V}_s ds \approx \frac{1}{3M}\left\{\widehat{V}_{t-1} + \widehat{V}_t + 2\sum_{l=1}^{\frac{M}{2}-1} \widehat{V}_{t-\frac{2l}{M}} + 4\sum_{l=1}^{\frac{M}{2}} \widehat{V}_{t-\frac{2l-1}{M}}\right\}$$

$$\text{Simpson's } \tfrac{3}{8} \int_{t-1}^{t} \widehat{V}_s ds \approx \frac{3}{8M}\left\{\widehat{V}_{t-1} + \widehat{V}_t + 2\sum_{l=1}^{\frac{M}{3}-1} \widehat{V}_{t-\frac{3l}{M}} + 3\sum_{l=1}^{\frac{M}{3}} \widehat{V}_{t-\frac{3l-2}{M}} + 3\sum_{l=1}^{\frac{M}{3}} \widehat{V}_{t-\frac{3l-1}{M}}\right\}.$$

Theoretically, all the above methods will converge to the integral as $M \to \infty$. The experiment results provided in Section 2.3 also confirm the claim but we can find that Trapezoidal performs a bit better than the other two approximation schemes and it allows a more flexible way for choosing the number of subinterval M.

On the purpose of saving computing time, as shown in (2.11), the mean (excepted for the initial value Y_0) and variance of the Ornstien-Uhlenbeck Process are fixed once model parameters are determined. Therefore we can calculate these terms ahead of the loop to reduce the computational time. Define

$$\sigma = \kappa\sqrt{\frac{1-e^{-\frac{\varrho}{M}}}{4\varrho}}, \alpha = e^{-\frac{\varrho}{2M}}$$

Finally, it will be more convenient for us to state the algorithm by restricting n to the even case since we mean to use the Box-Muller Scheme. Denote $n_2 = \frac{n}{2}$. In other cases, we can modify the algorithm slightly by drawing one more normal random variable and calculating Y_t^j after the $n_2 = \left\lfloor \frac{n}{2} \right\rfloor$ loop.

The algorithm for the Weighted Heston simulation is provided together with the Explicit simulation since it just has a minor differences between the two schemes. In the situation of $\nu = \frac{n\kappa^2}{4}$, we can skip the code for generating the weight and eliminate all statements about the stopping time η_ε^j and weight L^j.

Algorithm 1 Explicit and Weighted Heston Simulation

```
Initialize
for $j = 1$ to $N$ do
    $S_0^j = S_0$, $V_0^j = V_0$,$(L_0^j = 1, \eta_\varepsilon^j = T just for weighted Heston)$
    for $i = 1$ to $n$ do
        $Y_0^{j,i} = \sqrt{\frac{V_0}{n}}$
    end for
end for
Evolve
for $t = 1$ to $T$ do
    for $j = 1$ to $N$ do
        for $k = 1$ to $M$ do
            $V_{t-1+\frac{k}{M}} = 0$
            for $i = 1$ to $n_2$ do
                draw $U_1, U_2$ $U[0,1]$
                $Y_{t-1+\frac{k}{M}}^{j,2i-1} = \alpha Y_{t-1+\frac{k-1}{M}}^{j,2i-1} + \sigma\sqrt{-2logU_1}cos(2\pi U_2)$
                $Y_{t-1+\frac{k}{M}}^{j,2i} = \alpha Y_{t-1+\frac{k-1}{M}}^{j,2i} + \sigma\sqrt{-2logU_1}sin(2\pi U_2)$
                $V_{t-1+\frac{k}{M}} = V_{t-1+\frac{k}{M}} + (Y_{t-1+\frac{k}{M}}^{j,2i-1})^2 + (Y_{t-1+\frac{k}{M}}^{j,2i})^2$
            end for
        end for
        $Int(V^j) = 0$, $k = 0$
        while $k < M$ do
            $Int(V^j) = Int(V^j) + V_{t-1+\frac{k}{M}} + V_{t-1+\frac{k+1}{M}}$
            $Int_{\frac{1}{V^j}} = Int_{\frac{1}{V^j}} + \frac{1}{V_{t-1+\frac{k}{M}}} + \frac{1}{V_{t-1+\frac{k+1}{M}}}$
            $k = k + 1$
        end while
        $Int(V^j) = \frac{1}{2M} Int(V^j)$
        $N^j = N(0, a\sqrt{Int(V^j)})$
        $S_t^j = S_{t-1}^j exp(N^j + b + cInt(V^j) + d(V_t^j - V_{t-1}^j))$
        if $t \leq \eta_\varepsilon^j$ then
            if $\min_{k \in \{0,1,\ldots,M-1\}} V_{t-\frac{k}{2}}^j > \varepsilon$ then
                $L_t^j = L_{t-1}^j \exp\left\{ e\left(\ln\left(\frac{V_t^j}{V_{t-1}^j}\right) + \varrho\right) + fInt_{\frac{1}{V^j}} \right\}$
            else
                $\eta_\varepsilon^j = t - 1$
            end if
        end if
    end for
end for
```

2.3 Numerical Comparison of Explicit Solution

In this section, we will focus on comparing the explicit Heston Simulation to the traditional discretization methods. In this approach, we create a *ground truth* to judge performance by fixing Brownian paths B, β and running the Milstein method once with the extremely fine time step $\Delta t = 1/2,000$. Then we used these fixed B, β paths to calculate the error and evaluate the performance of the simulations discussed in this subsection.

The following collection of parameters will be used in this example: $\nu = \nu_\kappa = \frac{\kappa^2}{4}, \mu = 0.0319, \rho = -0.7, \varrho = 6.21, \kappa = 0.61$ and $T = 10$. We also take the (non-ground-truth) Euler and Milstein time steps to be $\Delta t = 1/M$, where the number of steps are $M = 200,\ 400,\ 1,000$. Since Condition (C) holds, the experiment is conducted for Explicit Heston algorithm. Table 3 gives the result of Euler and Milstein schemes with different M. And Table 4 below shows the performance and execution time of our Explicit Heston algorithm with the Trapezoidal, Simpson's $\frac{1}{3}$ as well as Simpson's $\frac{3}{8}$ rule.

The performance is evaluated in terms of RMS error. The RMS error is defined as followed:

$$E^C = \sqrt{\frac{1}{N}\sum_{t=1}^{T}\sum_{i=1}^{N}\left[(S_t^{C,i} - S_t^i)^2 + (V_t^{C,i} - V_t^i)^2\right]},$$

with S^C, V^C being the price and volatility for comparison and S, V being the groundtruth price and volatility.

	Euler Scheme			**Milstein Scheme**		
Step Size	200	400	1,000	200	400	1,000
RMS	18.8256	14.1382	9.79565	10.5435	7.08773	4.2306
Time	0.81	1.672	4.026	0.936	1.733	4.731

Table 3: Comparison between Traditional Discretization

	Explicit Solution			
	Trapezoidal		**Simpson's $\frac{1}{3}$**	**Simpson's $\frac{3}{8}$**
M	1	6	6	6
RMS	3.62901	2.89821	2.91712	3.08562
Time	0.0054	0.012	0.01	0.014

Table 4: Accuracy and Execution Time for Explicit Solution Simulation

It is clear that Explicit Heston is much more accurate and faster than the other methods. However, to provide a more intuitive and convenient way to calibrate the improvement, we combine performance and time factors to define an Explicit Gain as:

$$\text{Explicit Gain} = \frac{\tau_{\text{Other}}}{\tau_{\text{Explcit}}}, \tag{2.19}$$

where τ_{Explcit} and τ_{Other} are the execution times for our Explicit Heston algorithm and some other methods for a fixed performance. However, it is very hard to get the Milstein method, which has better convergence than the Euler one, to perform the same as the worst we can obtain using the explicit weak solution method. So we conduct the comparison by plotting the existing Milstein points and extending a smooth curve to get the time estimation with the roughly equal RMS error of the explicit solution. In this way, the estimated time for Milstein getting the same RMS error is 5.9. We follow a similar procedure to get the time estimation for Euler scheme and state the gains in Table 5.

Method	Euler	Milstein
Gain	2630	1093

Table 5: Explicit Gain over Euler and Milstein

It is obvious that there is significant gain in using the Explicit simulations compared to the traditional discretization method.

For other error levels and durations, the explicit also has the similar gains that exceed 1000.

Chapter 3

Valuation Algorithm for Path-Dependent Options

After improving the simulation of Heston Model, we turn our attention to pricing path-dependent options for the Heston. As mentioned in section 1.2, a popular tool for valuing these options is by applying the method introduced by Longstaff and Schwartz's [11]. A further research on improving the LSM method is done by Kouritzin [7] through replacing the regression with stochastic approximation method.

In this section, we will first review these two algorithms. Then we will consider enhancing the performance of these two methods by employing the resampling and branching into them.

3.1 The Least Squares Monte Carlo Algorithm

Longstaff and Schwartz [11] introduced in 2001 a method for finding the estimate of coefficients in equation (1.5) by solving a minimization problem

$$\min_{\alpha^J} E[|Z_{\tau^J_{t+1}} - \alpha^J \cdot e^J(S_t, V_t)|^2 1_{\{Z_t>0\}}], \tag{3.1}$$

and this can be implemented using Ordinary Least Squares regression. To get a better understanding of the backward process and LSM algorithm, we consider the following simple numerical sample.

Now, we wish to price an American Puts on a stock whose current price is 10, and will be exercisable at a strike price $K = 10$ at time 1 and 2, where time 2 is the expiration day. Suppose that the annual risk-neutral free interest rate is 3.19% and the period between each possible exercise one day. Then we will apply the LSM algorithm to a collection of simulated price paths in Table 3.6(a) so as to determine the optimal stopping time. Table 3.6(b) presents the cash flow of the holder condition on the option not being exercised before time 2. The cash flow is calculated following the optimal exercise strategy which equals to $(K - S_t) \vee 0$.

Then, if we are in time 1, a decision has to be made to whether to exercise the option now when the option is in-the-money. From the simulated price matrix, we see that paths $2, 3, 7, 9, 10$ are in the money at time 1. Denote $S_{in}(1)$ as the vector of the in the money stock price at time 1 and $Y_{in}(1)$ the corresponding (discounted) payoff at time 2. Conduct regression on the discounted payoff $Y_in(1)$ against the state variable $S_{in}(1)$. As shown in Table 3.7(a), we regress $Y_{in}(1)$ on a constant and $S_{in}(1)$. The result of the conditional expectation function is $E[Y|S(1) = s(1)] = 10.409 - 1.044S(1)$. While the conditional expectation of holding the option is less than the payoff received from immediate exercise, we will regard time 1 as the optimal stopping time (temporary) for path j. The recursion proceeds by rolling back to $t = 1$. A modified cash flow matrix according to the optimal exercise time has been shown in Table 3.7(b).

The value of the option can be obtained by taking average of the discounted payoff over all paths as:

$$\frac{\sum_{paths} cashflow_i \times e^{0.0319t_i}}{\text{Number of paths}}$$

Table 6: Numerical Example Matrix

(a) Simulated Sample Paths

Path	$t = 0$	$t = 1$	$t = 2$
1	10	11.32	11.15
2	10	7.20	6.34
3	10	8.41	9.64
4	10	10.97	11.14
5	10	11.62	14.49
6	10	11.89	10.22
7	10	9.52	9.25
8	10	11.0253	10.9624
9	10	9.85	10.83
10	10	9.88	9.42

(b) Cash Flow Matrix at Time 2

Path	$t = 0$	$t = 1$	$t = 2$
1	–	–	0
2	–	–	3.66
3	–	–	0.36
4	–	–	0
5	–	–	0
6	–	–	0
7	–	–	0.75
8	–	–	0
9	–	–	0
10	–	–	0.58

Table 7: Numerical Example Matrix (Continue)

(a) Regression at Time 1

Path	$t = 1$	$t = 2$
1	–	–
2	7.20	$3.66e^{-0.0319}$
3	8.41	$0.36e^{-0.0319}$
4	–	–
5	–	–
6	–	–
7	9.52	$0.75e^{-0.0319}$
8	–	–
9	9.85	0
10	9.88	$0.58e^{-0.0319}$

(b) Option Cash Flow Matrix

Path	$t = 1$	$t = 2$
1	–	–
2	–	3.66
3	–	0.36
4	–	–
5	–	–
6	–	–
7	0.477	–
8	–	–
9	0.14578	–
10	0.1215	–

As shown by this example, the LSM is easy to conduct and the general algorithm for American Puts will be provided below in Algorithm 2. But first, we will specify some possible choices of basis function e_k that can be used for linearly approximating the conditional expectation. As discussed in Longstaff and Schwartz paper [11], all of the experiments in this thesis will use the following weighted Laguerre polynomials as the basis function.

$$
\begin{gathered}
L_0(x) = exp(-x/2) \\
L_1(x) = exp(-x/2)(1-x) \\
L_2(x) = exp(-x/2)(1-2x+\frac{x^2}{2}) \\
L_n(x) = exp(-x/2)\frac{e^x}{n!}\frac{d^n}{dx^n}(x^n e^{-x})
\end{gathered}
$$

There are also other types of basis function include Hermite, Chebyshev and Genenbauer polynomials.

Except for selecting an appropriate type of basis functions, we also need to decide a suitable number of the basis. For the simplest cases, when only discussing one state variable, we might be able to obtain a desired level of convergence by increasing the number of basis without encountering the problem of inverting matrix. However, for higher-dimensional problems, the set of basis functions should not only include terms in each factor, but also their cross-products. Therefore, the number of basis function will increase exponentially and easily result in computational problem in computer. In Longstaff and Schwartz paper, they only use at most 22 basis function. However, consider a three factor model, such as the Asian Calls, 22 basis function is not enough to get the satisfying accuracy.

Remark The algorithm below is set up for Euler, Milstein and Explicit Heston simulation. For Weighted Heston, we have to include L for re-weighting (S, V, Z) to the correct joint process distribution with respect to a new probability measure $\widehat{P}$ when they do not under P. To be specific, we have to add a step in Initialize with setting $\zeta = \lambda = 0$ and use $\zeta = \zeta + L^j_{\tau^{J,j}} Z^j_{\tau^{J,j}}$,$\lambda =$

Algorithm 2 LSM Algorithm

```
Initialize
Pick bases function $e_k$,set all $\alpha_t^J = 0$ and all $\tau^{J,j} = T$.
Simulate Create N paths for the all the state variables. i.e. $\{S^j, V^j\}_{j=1}^N$ of
  $(S, V)$; set $Z_t = e^{-\mu t}(K - S_t) \vee 0$.
Backward Induction
for $t = T - 1$ to 0 do
    Set $k = 0$
    for $j = 1$ to $N$ do
        if $Z_t^j > 0$ then
            $S_t^{j_l} = S_t^j, V_t^{j_l} = V_t^j, k = k + 1$
        end if
    end for
    $\mathrm{A}_t^J = \frac{1}{k}\sum_{i=1}^k e^J(S_t^{j_l}, V_t^{j_l})e^J(S_t^{j_l}, V_t^{j_l})^{\mathrm{T}}$
    $\mathrm{B}_t^J = \frac{1}{k}\sum_{i=1}^k e^J(S_t^{j_l}, V_t^{j_l})Z^j_{\tau^{J,j}}$
    $\alpha_t^J = \mathrm{A}_t^{-1}\mathrm{B}_t$
    Adjust Stopping Time
    for $j = 1$ to $k$ do
        if $Z_t^j > 0$ and $Z_t^j > \alpha_t^J \cdot e^J(S_t^{j_l}, V_t^{j_l})$ then
            $\tau^{J,j} = t$
            for $tt = t$ to $T$ do
                $Z_{tt}^j = 0$
            end for
        else
            $Z_t^j = 0$
        end if
    end for
end for
Value the American Puts
for $i = 1$ to $N$ do
    for $t = 1$ to $T$ do
        $\zeta = \zeta + Z_t^j$
    end for
end for
Value:$O = \frac{\zeta}{N}$
```

$\lambda + L^{j}_{\tau^{J,j}}$instead of $\zeta = \zeta + Z^{j}_{t}$ in the valuation. The final option value will be given by $O = \frac{\zeta}{\lambda}$.

3.2 The Stochastic Approximation Algorithm

As explained before section 3.1, the LSM method will become highly ill-conditioned as the number of factors and basis functions increases, Kouritzin[7] proposed a stochastic approximation alternative to the problem which avoids the main difficulty of matrix inversion.

First introduced by Robbins and Monro [22], stochastic approximations (SA) algorithm has been used to solve stochastic optimization problems as the one we state in (3.1). Following the works of Kouritzin[23] as well as Kouritzin and Sadeghi [20] in 2015, Kouritzin[7] explained the convergence of the stochastic approximation to the same solution as the least square method given for equation 3.1.

Here we will give an intuitive explanation of how the stochastic approximation works. Consider the solution to the following problem. Suppose θ_n is a sequence satisfying:

$$\theta_{n+1} = \theta_n + \epsilon(b - A\theta_n)$$

where ϵ is an arbitrary small constant. While A and b are known, it can be proved that $\theta_n \to A^{-1}b$. Taking the simplest case when $b = 0$ and A is positively defined, by the recursion, we will find $\theta_{n+1} = (I - \epsilon A)\theta_n = (I - \epsilon A)^{n+1}\theta_0$, and it converges to $0 = A^{-1}b$ as n increased. However, assuming that A and b are unknown, instead, we have a random sequence $\{b_n\}$ and $\{A_n\}$ such that

$$\frac{1}{N}\sum_{n=1}^{N} b_n \to b, \qquad \frac{1}{N}\sum_{n=1}^{N} A_n \to A$$

as $n \to \infty$ i.e. satisfy a strong law of large numbers. To obtain the convergent result in this case, we have to replace ϵ with $\frac{\gamma}{n}$ to asymptotically kill the noise.

Then, θ_n is generated by the recursion:

$$\theta_{n+1} = \theta_n + \frac{\gamma}{n}(b_n - A_n\theta_n), \qquad n = 1, 2, 3, 4 \ldots \tag{3.2}$$

To show that $\theta_n \to A^{-1}b$, we still take $b_n = 0$ and $A_n = A$, then

$$\theta_{n+1} = \prod_{k=1}^{n+1}(I - \frac{\gamma A}{k})\theta_0 \approx \theta_0 exp(-\sum_{k=1}^{n+1}\frac{\gamma A}{k}) \to 0$$

as $n \to \infty$ and thus $\theta_n \to A^{-1}b = 0$. To solve the the problem in (3.1), we can substitute $b_j = Z^j_{\tau^{J,j}} e^J(S^j_t, V^j_t)$ and $A_j = e^J(S^j_t, V^j_t)^{\mathrm{T}} e^J(S^j_t, V^j_t)$ in equation (3.2) .

The SA pricing algorithm can be simply obtained by replacing several steps in the backward induction in algorithm 2 as follows:

Algorithm 3 Stochastic Approximation (backward induction)

```
1:  for $t = T - 1$ to 0 do
2:      Set $k = 0$
3:      for $j = 1$ to $N$ do
4:          if $Z_t^j > 0$ then
5:              $S_t^{j\iota} = S_t^j, V_t^{j\iota} = V_t^j, k = k + 1$
6:          else
7:              $Z_t^j = 0$
8:          end if
9:          $\alpha_t^J = \alpha_t^J + \frac{\gamma}{k}(Z_{\tau^{J,j}}^j - e^J(S_t^j, V_t^j)^{\mathrm{T}}\alpha_t^J)e^J(S_t^j, V_t^j)$
10:     end for
11:     for $j = 1$ to $k$ do
12:         if $Z_t^j > 0$ and $Z_t^j > \alpha_t^J \cdot e^J(S_t^{j\iota}, V_t^{j\iota})$ then
13:             $\tau^{J,j} = t$
14:             for $tt = t$ to $T$ do
15:                 $Z_{tt}^j = 0$
16:             end for
17:         else
18:             $Z_t^j = 0$
19:         end if
20:     end for
21: end for
```

Similar to the cases in LSM, when pricing under the Weighted Heston simulation, we need to involve the particle weight to calibrate the odds that the particles $\{S^j, V^j\}_{j=1}^N$ generated from the closet Heston represent the real (S, V) of underlying model. We will then change the equation in line 9 as:

$$\alpha_t^J = \alpha_t^J + \frac{\gamma L_t^j}{k}(Z_{\tau^{J,j}}^j - e^J(S_t^j, V_t^j)^{\mathrm{T}}\alpha_t^J)e^J(S_t^j, V_t^j) \tag{3.3}$$

and do the final valuation steps as Remark 1.

It is relatively important to choose the right γ carefully in the algorithm since it controls the step size towards the solution and affects the rate of convergence of SA. Taking the γ too large might leave too much randomness in the approximation procedure and result in a slow convergence or even a bad result. On the contrary, if γ has been chosen too small, we might not be able to reach the convergence result especially when the number of particles is quite small. An alternative way is to conduct a two step stochastic approximation as:

$$\widehat{\alpha}_t^J = \widehat{\alpha}_t^J + \frac{\gamma L_t^j}{\sqrt{k}}(Z^j_{\tau^{J,j}} - e^J(S_t^j, V_t^j)^{\mathrm{T}}\widehat{\alpha}_t^J)e^J(S_t^j, V_t^j)$$
$$\alpha_t^J = \alpha_t^J + \frac{1}{k}(\widehat{\alpha}_t^J - \alpha_t^J)$$

3.3 The Branching Heston Algorithm

Having discussed the two pricing methods, we turn our attentions back to the Theorem 2.2 in Chapter 2. As elaborated before, the idea of finding the general Heston solutions is by re-weighting the results generated from the closest explicit. However, the weights $\{\widehat{L}^j_{T\wedge\eta_\varepsilon}\}_{j=1}^N$ will usually be uneven, resulting in smaller number of effective particles within the whole simulation. The problem will finally affect the efficiency and accuracy of pricing. To increase the effective particles, we will introduce the branching algorithm suggested by Kouritzin [18] in 2015.

3.3.1 General Branching Algorithm

In the general branching algorithm, to branch or to kill a particle is only based on the ratio of that individual particle's weight to the average weight A_t of all the particles. First, we have to choose our resample parameter $r_t \in [1, \infty]$, which is used to decide whether the particle will be resampled or not. If $r_t = 1$, all the particle will be resampled while $r_t = \infty$ means the algorithm works equally as the weighted algorithm. To illustrate the method more clearly, we provide a graphical view of the process in Figure 3.

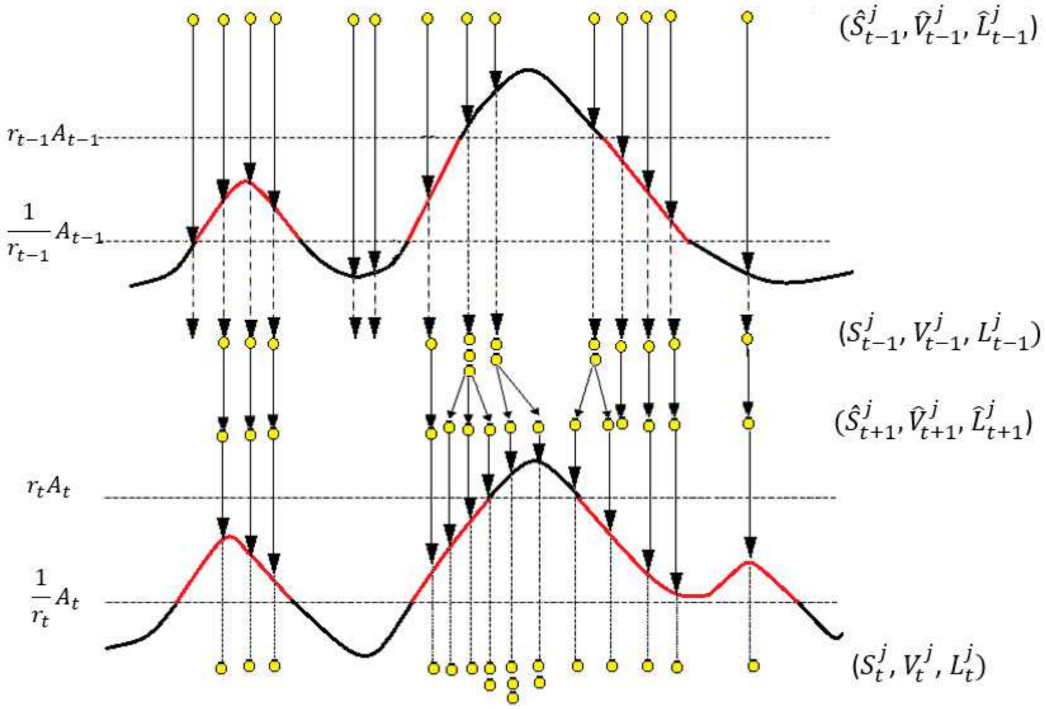
$(\hat{S}^j_{t-1}, \hat{V}^j_{t-1}, \hat{L}^j_{t-1})$
$r_{t-1}A_{t-1}$
$\frac{1}{r_{t-1}}A_{t-1}$
$(S^j_{t-1}, V^j_{t-1}, L^j_{t-1})$
$(\hat{S}^j_{t+1}, \hat{V}^j_{t+1}, \hat{L}^j_{t+1})$
r_tA_t
$\frac{1}{r_t}A_t$
(S^j_t, V^j_t, L^j_t)

Figure 3: General Branching Algorithm

As shown in Figure 3, the branching particle filter starts at time $t-1$ with $(\widehat{S}^j_{t-1}, \widehat{V}^j_{t-1}, \widehat{L}^j_{t-1})$. For each particle, we compute the importance weights using the information at $t-1$. The curve denotes the importance weight and the imaginary lines are decided by using the resample parameter and the average weight. While the particle has its weight fall on the red part of the curve between the imaginary lines, the particle will be preserved and its weight will be inherited. Otherwise, we do the residual-style branching, leading the particle to die out or duplicate with the average weight. Only the survived particle will evolve again using Theorem 2.2.

The algorithm for the general branching is given in Algorithm 4.

Algorithm 4 General Branching

```
Initialize
Set $N_0 = N$, $N_t = 0$ for all $t \in \mathbb{N}$
for $j = 1$ to $N$ do
    $S_0^j = S_0$, $V_0^j = V_0$, $L_0^j = 1$
end for
Branching
for $t = 1$ to $T$ do
    for $j = 1$ to $N_{t-1}$ do
        Use Theorem 2.2 and $(S_{t-1}^j, V_{t-1}^j, L_{t-1}^j)$ to create $(\widehat{S}_{(t-1,t]}^j, \widehat{V}_{(t-1,t]}^j, \widehat{L}_t^j)$
    end for
    Average Weight: $A_t = \frac{1}{N}\sum_{j=1}^{N_{t-1}} \widehat{L}_t^j$
    for $k = 1$ to $N_{t-1}$ do
        if $\widehat{L}_t^k \notin \left(\frac{1}{r_t}A_t, r_t A_t\right)$ then
            Offspring Number: $N_t^k = \left\lfloor \frac{\widehat{L}_t^k}{A_t} \right\rfloor + \rho_t^k$, with $\rho_t^k$ a $\left(\frac{\widehat{L}_t^k}{A_t} - \left\lfloor \frac{\widehat{L}_t^k}{A_t} \right\rfloor\right)$-Bernoulli
            for $j = 1$ to $N_t^k$ do
                Resample: $L_t^{N_t+j} = A_t, (S_t^{N_t+j}, V_t^{N_t+j}) = (\widehat{S}_t^k, \widehat{V}_t^k)$
            end for
            Add Offspring Number: $N_t = N_t + N_t^k$
        else
            $N_t = N_t + 1$, $L_t^{N_t} = \widehat{L}_t^k$, $(S_t^{N_t}, V_t^{N_t}) = (\widehat{S}_t^k, \widehat{V}_t^k)$.
        end if
    end for
end for
```

To employ the algorithm into the pricing methods, we have to create the historical particles of a child. This can be done by appending the child's path to its parent and grandparent, etc. As we only focus on the historical path of the particles survived till time T, to save computational resource, we will only preserve the index of its parents and create the ancestor ties once we obtain$(\widehat{S}_T^j, \widehat{V}_T^j, \widehat{L}_T^j)$.

The primary advantage of the branching method is the control of the number of particles. As the average A_t is normalized by the initial number of particles instead of the current, the expected number of future particles given the current state is to be the initial N. The variation mitigation will enhance the performance of the method. In the following sections, we will list some branching options and discuss their performance based on how successful they achieved in controlling particle numbers.

3.3.2 Residual Branching

As mentioned in the general branching algorithm, the offspring number N_t^k of the branched particle is calculated by

$$N_t^k = \left\lfloor \frac{\widehat{L}_t^k}{A_t} \right\rfloor + \rho_t^k \tag{3.4}$$

where ρ_t^k is a $\left(\frac{\widehat{L}_t^k}{A_t} - \left\lfloor \frac{\widehat{L}_t^k}{A_t} \right\rfloor\right)$-Bernoulli random variable and hence N_t^k is unbiased. Actually, there are different choices of $\{\rho_t^k\}_{k=1}^{N_{t-1}}$ for unbiased branching which will influence the performance and computational efficiency. A simple possibility is

i) Let $\{U_t^k\}_{k=1}^{N_{t-1}}$ be independent $[0,1]$-Uniform RVs.

ii) Set $\rho_t^k = 1_{U_t^k \leq \left(\frac{\widehat{L}_t^k}{A_t} - \left\lfloor \frac{\widehat{L}_t^k}{A_t} \right\rfloor\right)}$.

In this case, the $\{\rho_t^k\}_{k=1}^{N_{t-1}}$ are independent of each other and everything else. The algorithm of residual branching is as follows:

Algorithm 5 Residual Branching

```
Initialize
Set $N_0 = N$, $N_t = 0$ for all $t \in \mathbb{N}$
for $j = 1$ to $N$ do
    $S_0^j = S_0$, $V_0^j = V_0$, $L_0^j = 1$
end for
Branching
for $t = 1$ to $T$ do
    for $j = 1$ to $N_{t-1}$ do
        Use Theorem 2.2 and $(S_{t-1}^j, V_{t-1}^j, L_{t-1}^j)$ to create $(\widehat{S}_{(t-1,t]}^j, \widehat{V}_{(t-1,t]}^j, \widehat{L}_t^j)$
    end for
    Average Weight: $A_t = \frac{1}{N} \sum_{j=1}^{N_{t-1}} \widehat{L}_t^j$
    for $k = 1$ to $N_{t-1}$ do
        if $\widehat{L}_t^k \notin \left(\frac{1}{r_t} A_t, r_t A_t\right)$ then
            Let $U_t^k$ be independent $[0, 1]$-Uniform RVs.
            Set $\rho_t^k = 1_{U_t^k \le \left(\frac{\widehat{L}_t^k}{A_t} - \left\lfloor \frac{\widehat{L}_t^k}{A_t} \right\rfloor\right)}$.
            Offspring Number: $N_t^k = \left\lfloor \frac{\widehat{L}_t^k}{A_t} \right\rfloor + \rho_t^k$
            for $j = 1$ to $N_t^k$ do
                Resample: $L_t^{N_t+j} = A_t, (S_t^{N_t+j}, V_t^{N_t+j}) = (\widehat{S}_t^k, \widehat{V}_t^k)$
            end for
            Add Offspring Number: $N_t = N_t + N_t^k$
        else
            $N_t = N_t + 1$, $L_t^{N_t} = \widehat{L}_t^k$, $(S_t^{N_t}, V_t^{N_t}) = (\widehat{S}_t^k, \widehat{V}_t^k)$.
        end if
    end for
end for
```

When implemented in the computer, we first create as many particles as we can and allocate the remaining offspring using residuals $\left(\frac{\widehat{L}_t^k}{A_t} - \left\lfloor \frac{\widehat{L}_t^k}{A_t} \right\rfloor\right)$ and independent uniform random variables.

3.3.3 Combined Branching

We can also decrease the noise and stabilize the number of particles following the same idea of stratified resampling. The key idea is to generate the uniform random variables in a smaller interval $\left[\frac{k-1}{N_{t-1}-l}, \frac{k}{N_{t-1}-l}\right]$ instead of drawing directly from the $[0,1]$ interval. Here l denotes the non-resample count. Beside, we will still employ the residual techniques for ρ_t^k in this case and so $\rho_t^k = 1_{U_t^k \leq \left(\frac{\widehat{L}_t^k}{A_t} - \left\lfloor \frac{\widehat{L}_t^k}{A_t} \right\rfloor\right)}$.

Combining the stratified and residual methods makes it less possible to get mostly large or small uniform random numbers so there is less variation in the ρ_t^k and thus the number of particles. The combined branching algorithm is stated in detail as followed:

Algorithm 6 Combined Branching

```
Initialize
Set $N_0 = N$, $N_t = 0$ for all $t \in \mathbb{N}$
for $j = 1$ to $N$ do
    $S_0^j = S_0$, $V_0^j = V_0$, $L_0^j = 1$
end for
Branching
for $t = 1$ to $T$ do
    for $j = 1$ to $N_{t-1}$ do
        Use Theorem 2.2 and $(S_{t-1}^j, V_{t-1}^j, L_{t-1}^j)$ to create $(\widehat{S}_{(t-1,t]}^j, \widehat{V}_{(t-1,t]}^j, \widehat{L}_t^j)$
    end for
    Average Weight: $A_t = \frac{1}{N}\sum_{j=1}^{N_{t-1}} \widehat{L}_t^j$
    Non-resample count: $l = 0$
    for $k = 1$ to $N_{t-1}$ do
        if $\widehat{L}_t^k \notin \left(\frac{1}{r_t}A_t, r_t A_t\right)$ then
            $\widehat{L}_t^{k-l} = \widehat{L}_t^k, (\widehat{S}_t^{k-l}, \widehat{V}_t^{k-l}) = (\widehat{S}_t^k, \widehat{V}_t^k)$
        else
            $l = l + 1$, $L_t^l = \widehat{L}_t^k$, $X_t^l = \widehat{X}_t^k$
        end if
    end for
    $N_t = l$
```

Algorithm 6 Combined Branching (continued)

```
    for $k = l+1$ to $N_{t-1}$ do
        $W_t^k$ be independent with $W_t^k \sim \left[\frac{k-1}{N_{t-1}-l}, \frac{k}{N_{t-1}-l}\right]$-Uniform
        $U_t^k = W_t^{p(k)}$, $p$ is a random permutation of $\{l+1, l+2, ..., N_t - 1\}$
        $N_t^k = \left\lfloor \frac{\widehat{L}_t^{k-l}}{A_t} \right\rfloor + 1_{U_t^k \leq \left(\frac{\widehat{L}_t^{k-l}}{A_t} - \left\lfloor \frac{\widehat{L}_t^{k-l}}{A_t} \right\rfloor\right)}$
        for $j = 1$ to $N_t^k$ do
            $L_t^{N_t+j} = A_t, (S_t^{N_t+j}, V_t^{N_t+j}) = (\widehat{S}_t^{k-l}, \widehat{V}_t^{k-l})$
        end for
        $N_t = N_t + N_t^k$.
    end for
end for
```

In both Residual and Combined branching, we only focus on enhancing the performance through optimizing the choice of ρ_t^k. However, the resample parameter r_t is taken to be constant. We may also consider adjusting the resample parameter along the time steps t.

3.3.4 Dynamic Branching

One example of this varying parameter is *dynamic branching* suggested by Kouritzin [18], where

$$r_t = \exp\left(c \left[\frac{1}{N_{t-1}} \sum_{k=1}^{N_{t-1}} (\ln \widehat{L}_t^k)^2 - \left(\frac{1}{N_{t-1}} \sum_{k=1}^{N_{t-1}} \ln \widehat{L}_t^k \right)^2 \right]^{\frac{q}{2}} \right) \tag{3.5}$$

with $c, q > 0$. To be specific, maintaining the same average amount of branching, we can use larger $q > 1$ and adjust c to do more resample when the system entropy is low. In the other cases, a smaller $q < 1$ will be chosen. We can still run the other steps of Combine and Residual branching by replacing the fix parameter to the time dependent r_t.

3.3.5 Effective Particle Branching

Another way to deal with uneven weights that would arise from direct use of Theorem 2.2 is through an *effective* number of particles estimate, N^{eff}. We set the effective and non-effective particle estimates as:

$$N_{t-1}^{eff} = \frac{N^2 A_t^2}{\sum\limits_{k=1}^{N_{t-1}} \left(\widehat{L}_t^k\right)^2} = \frac{\left(\sum\limits_{k=1}^{N_{t-1}} \widehat{L}_t^k\right)^2}{\sum\limits_{k=1}^{N_{t-1}} \left(\widehat{L}_t^k\right)^2}, \quad N_{t-1}^{noneff} = N_{t-1} - N_{t-1}^{eff}.$$

By the equalities above, there are two extreme cases. When all $\widehat{L}_t^k$ are the same and all particles are equally effective, we will have $N_{t-1}^{eff} = N_{t-1}$. In contrast, if all but one of the $\widehat{L}_t^k$ were 0(or arbitrarily close to 0) and there is only one effective particle, then $N_{t-1}^{eff} = 1$. Otherwise, it gives us a number somewhere in between that can be interpreted as the effective number of particles. It is reasonable to expect better results when branching either more or fewer particles in the situation there are few effective ones. Intuitively, we might branch more in order to obtain more effective particles immediately. However, if those few particles with high weights happen to be wrong, then we will be likely to move the majority of particles to the bad states. Hence, in *effective particle branching*, we set

$$r_t = \frac{c^{eff} N_{t-1}^{eff} + c^{noneff} N_{t-1}^{noneff}}{N_{t-1}} = c^{noneff} + (c^{eff} - c^{noneff}) \frac{N_{t-1}^{eff}}{N_{t-1}} \tag{3.6}$$

and let the data experimentally determine the constants c^{eff}, $c^{noneff} > 0$.

Chapter 4

Numerical Result

In Chapter 4, we will provide the empirical results of the pricing method discussed above. This section will be organized as followed: first, we will compare the weighted Heston simulation to the traditional discretization method on pricing American Puts and Asian Straddles with LSM algorithm. Then, a comparison between the LSM algorithm and its alternative, the stochastic approximation approach will be presented. Finally, we will conduct an experiment to see how the performance and efficiency will be improved by employing the branching algorithm and reach our final conclusion for the best pricing algorithm.

4.1 Comparison between Weighted Heston and Traditional Discretization Method

In this section, we will focus on comparing the weighted Heston algorithm to the Euler and Milstein schemes. To help convey the high efficiency of the weighted Heston algorithm, we will stick to the LSM method in pricing procedure but substitute the simulation of stock price and volatility by the weighted Heston algorithm as well as other methods.

4.1.1 Pricing the American Puts

We will conduct the comparison on American put option first. The experiment will use the following model parameters: $\nu = \frac{8.1\kappa^2}{4}, \mu = 0.0319, \rho = -0.7, \varrho = 6.21, \kappa = 0.2, S_0 = 100, V_0 = 0.502, T = 50$ and the strike price $K = 100$. Here $n = 8.1 \notin \mathbb{N}$ and Condition (C) does not hold. Hence, we will use the full Weighted Heston algorithm with $\nu_\kappa = 2\kappa^2$ in the closest explicit Heston model. Besides, we will use the weighted Laguerre polynomials with $J = 3^2$ for the LSM pricing process.

To provide a baseline for comparing the accuracy, we run the groundtruth with Milstein using an extraordinary fine time step $M = \frac{1}{1000}$ and enormous number of particle as shown in Table 8. To evaluate the performance of different method, we will fix the error for the three method and compare their execution time. The error is defined as:

$$error = \frac{1}{100}\sum_{i=1}^{100} \mid P_i^E - P \mid \tag{4.1}$$

with P_i^E being the option price with Euler scheme and P being the groundtruth option price. The error will be calculated using 100 independent experiment results. The other error are defined similarly. The results are provided in Table 9 and 10

Ground Truth	
N	$1,000,000$
M	$1,000$
Option Price	12.269

Table 8: Ground Truth of the American Puts for Weighted Comparison

	Euler	**Milstein**	**Weighted Heston**
N	$10,000$	$7,225$	2.500
M	100	85	15
Price	12.3116	12.2254	12.2258
Error	0.0426	0.0436	0.0432
Time	17.4178	13.156	1.387
Time Factor	1	1.324	12.562

Table 9: Execution Time for Euler, Milstein and Weighted Heston with lower Accuracy on American Puts

	Euler	**Milstein**	**Weighted Heston**
N	$40,000$	$30,625$	$3,500$
M	200	175	17
Price	12.3013	12.2367	12.2366
Error	0.0323	0.0323	0.0324
Time	143.356	84.6254	2.20966
Time Factor	1	1.694	64.877

Table 10: Execution Time for Euler, Milstein and Weighted Heston with Higher Accuracy on American Puts

In Table 9 and 10, we defined a Time Factor that resembled to the Explicit Gain in section 2.3. It describes how many times faster can Milstein and Weighted Heston simulation achieve compared to the basic Euler scheme with the same error. As presented above, the Weighted Heston algorithm shows a remarkable improvement over the traditional discretization methods. The speed advantage will be more significant when we require a higher accuracy.

4.1.2 Pricing the Asian Straddles

Then we will perform another comparison among the Euler, Milstein and Weighted Heston algorithm on the Asian Straddles. The payoff process for an

Asian straddle is $Z_t = |R_t - K|$, where R is the running average of the Heston price. It can be calculated as the following:

$$R_t = \frac{t-1}{t} R_{t-1} + \frac{1}{t} S_t \tag{4.2}$$

As the Asian Straddles option pricing model is a three factor model(stock price, average price and volatility), we only use $J = 2$ for each factor to simplify the problem. Other parameters remain the same as the American put option. The ground truth price used for measuring the error is given in Table 11 and the comparisons are shown in Table 12 and 13

Ground Truth	
N	$1,000,000$
M	$1,000$
Option Price	136.174

Table 11: Ground Truth of the Asian Straddles for Weighted Comparison

	Euler	**Milstein**	**Weighted Heston**
N	$10,000$	$4,900$	$3,510$
M	100	70	12
Price	135.956	135.952	136.019
Error	0.218	0.214	0.222
Time	18.8237	11.2313	1.8943
Time Factor	1	1.676	9.937

Table 12: Execution Time for Euler, Milstein and Weighted Heston with Lower Accuracy on Asian Straddles

	Euler	Milstein	Weighted Heston
N	$40,000$	$25,600$	$4,800$
M	200	160	13
Price	136.043	136.046	136.303
Error	0.131	0.128	0.124
Time	145.864	73.958	2.861
Time Factor	1	1.972	50.984

Table 13: Execution Time for Euler, Milstein and Weighted Heston with Higher Accuracy on Asian Straddles

With a less demand of accuracy, the Weighted Heston performs at approximate 10 times better than the widely-used Euler scheme. And we only need $\frac{1}{50}$ computation time to get a more precise estimation for the option price by applying Weighted Heston algorithm.

The experiments conducted in this section empirically proved the Weighted Heston algorithm is a better choice for simulating the Heston model in option pricing problems for with highly efficiency.

4.2 Comparison of Stochastic Approximation and LSM Scheme

In this section, we will decide which valuating method should be applied when we price path-dependent options. The experiments will be performed for American Puts first and further the discussion to a more complicated three factor problem.

4.2.1 Valuation of American Puts

To value the American Puts, we will use the following parameters: $\mu = 0.0319, \rho = -0.7, \varrho = 6.21, \kappa = 0.61, K = 100, S_0 = 100, V_0 = 0.0102, T = 50$ and $\nu = \frac{1}{2}\kappa^2$ so the Explicit algorithm applies in this case. As in section 4.1, we will use the weighted Laguerre polynomials as the basis functions and all the prices are calculated by taking the average of 100 independent experiment.

First we investigate how many basis functions we can add before the LSM and SA algorithms fail to give a correct estimation of the option price. Tables 14, 15 show this along with performance.

	SA Price	**SA Time**	**LSM Price**	**LSM Time**
J$=2^2$	8.44858	0.11298	8.40775	0.124679
J$=4^2$	8.49936	0.14411	8.38028	0.258755
J$=8^2$	8.41892	0.2566856	5.58625	2.13897

Table 14: SA and LSM with $N = 10,000$ on American Puts

	SA Price	**SA Time**	**LSM Price**	**LSM Time**
J$=2^2$	8.4213	1.24712	8.39404	1.51143
J$=4^2$	8.50788	1.79924	8.51376	2.7524
J$=8^2$	8.51644	2.64996	7.18587	20.1488

Table 15: SA and LSM with $N = 100,000$ on American Puts

We can draw several conclusions from Tables 14 and 15. First, the execution time for the SA algorithm is much less than the popular LSM algorithm, especially as J increases and matrix inversion becomes difficult. For small number of the basis functions, SA is about 10% faster than LSM. However, when the number of basis function increases, the SA time performance becomes even more preferable.

Next, given enough particles (eg. $N = 100,000$ here), prices and pricing accuracy should both increase as J increase because we will obtain a better estimate of the optimal stopping time since more information has been exhausted. Table 15 demonstrates that as J increase from 2^2 to 8^2 the SA option prices increase and the SA algorithm does not fail. Indeed, it should never fail as it avoids the numeric problems of inverting a huge matrix. The LSM algorithm does fail as prices drop dramatically and time explodes for large J in both Table 14 and Table 15 result from the complicated matrix inversion in the least squares estimate.

Also we notice that prices fall in Table 14 for the SA algorithm. However, the reason for this slight drop down (compared to change in LSM price) is different. When N is small the projection parameter estimates are often bad since the SA might not be able to reach the stable and convergent results. The situation will be worst when there are lots of parameters to estimate so the optimal stopping is easily missed, even when J is large. Therefore, with a limited number of particles, increasing J might not be able to provide a better result.

To provide more credible evidence of this expected price improvement in J given large enough N , we also run the Stochastic Approximation method with $N = 1,000,000$ and $J = 12$, as shown in Table 16, the American put option price rises to 8.58712. As the number of particles and basis functions are rather high in this situation, we will take the price as the ground truth too.

Ground Truth	
N	$1,000,000$
J	12^2
SA Option Price	8.58712

Table 16: Ground Truth of the American Put Price

The SA prices in Tables 14 and 15 were running in right direction. The SA algorithm behaves better than the LSM, especially as the desired accuracy increases.

4.2.2 Valuation of Asian Calls

We continue our comparison of SA and LSM algorithms but now on an Asian Call option and in a situation where the Weighted Heston has to be used.

In this section, we will use model parameters: $\nu = \frac{8.1\kappa^2}{4}, \mu = 0.0319, \rho = -0.7, \varrho = 6.21, \kappa = 0.2$ and $T = 50$ so $n = 8.1$ and $\nu_\kappa = 2\kappa^2$ is used in the closest explicit Heston. The ground truth for this experiment is:

Ground Truth	
M	12
N	$1,000,000$
J	12^3
SA Option Price	31.3455

Table 17: Ground Truth of the Asian Call Price

We choose to run the ground truth using SA since it has been proved in the previous case that LSM will fail when we need to inverse a 64×64 matrix. Hence, for the Asian Calls, which is a three factors model, it is impossible to get the accurate results with the LSM. Also, it will take a rather long time to run the experiment with Euler and Milstein for this value of N and a high enough number of steps M. Limited by the time and facilities, we apply the Weighted Heston simulation in finding the ground truth.

Following the same procedure of pricing the American Put option, we first consider performance with different numbers of bases functions and show this in Table 18:

	SA Price	**SA Time**	**LSM Price**	**LSM Time**
J=2^3	31.3411	11.2404	25.2365	12.511
J=4^3	31.3411	36.2066	20.3398	92.432

Table 18: SA and LSM with $N = 100,000$,$M = 12$

We can clearly see that the LSM fails already when $J = 2^3$. The main reason still lies in the matrix inversion part. Although the size of the matrix is 8×8, which is smaller than the feasible case in Table 15 as 16×16, we have both price and average price in the Asian Calls. This might result in a greater chance of the matrix having nearly linearly dependent rows and hence being highly ill-conditioned to inversion even matrix size is not that huge.

The SA algorithm does not fail even for large number of basis function. The price remains the same for $J = 2^3$ and 4^3 might on account of its averaging feature. Since we are pricing options on average spot price, which varies less and less as time goes on, the volatility inherent in the option is reduced so increase the number of basis functions will not help much with mining addition information. Indeed, a comparison between Tables 17 and 18 shows that the SA algorithm with $J = 2, 4$ and $N = 100,000$ already provides a rather close result to the ground truth.

4.3 Comparison of the Branching-SA, Weighted-SA, and Euler-LSM Algorithm on American Puts

In this section, we will conduct the experiment mainly to prove the advantages for employing the branching method into option pricing. However, as the final part of the numerical results, we are trying to establish the best algorithm of simulating and pricing path-dependent options for Heston model. Therefore, we will compare the traditional simulation pricing schemes as Euler with LSM to the combination of Weighted Heston and SA algorithm. Furthermore, the combined branching algorithm will be applied on it.

The model parameters used in this section are: $\nu = \frac{8.1\kappa^2}{4}, \mu = 0.0319, \rho = -0.7, \varrho = 6.21, \kappa = 0.2$ and $T = 50$ so $n = 8.1 \notin \mathbb{N}$ and Condition (C) does not hold. Hence, we will use the full Weighted Heston algorithm with $\nu_\kappa = 2\kappa^2$ in the closest explicit Heston model. The initial state $S_0 = 100, V_0 = 0.102$, and the strike price $K = 100$. To measure the improvement, instead of fixed performance as we did in the previous experiments, we mean to conduct the accuracy comparison given a suitable execution time in this section. The alteration is made as in real market, a precise option price is placed as the first priority to the investors. But we are not going to make our fixed time to be large since the results will be useless under the circumstances of fast trading nowadays.

As demonstrated by the previous experiments, Weighted Heston shows time advantages compared to Euler and Milstein while the SA algorithm outperforms the LSM in pricing accuracy and saving computation time. Therefore, we will combine these two methods for simulating and pricing options to get a better result. The ground truth is obtained by using 12^2 basis functions and a million particle for the Weighted Heston with SA method.

Ground Truth	
M	5
N	$1,000,000$
J	12^2
SA Option Price	7.9426

Table 19: Optimal American Put Price

For the actual experiments, we fixed our computation time at around 19.5 second and calculating the pricing error for different methods. The error is computed in the same way as in section 4.1.

	Euler-LSM	**Weighted-SA**
M	100	5
N	$10,000$	$65,000$
J	4^2	8^2
Price	7.371	7.932
Error	0.572	0.0103
Time	19.662	19.433
Performance Factor	1	55.534

Table 20: Comparison of Euler-LSM and Weighted-SA on American Puts

We define the Performance Factor in table 20 analogous to the Time Factor in the previous sections so as to measure the accuracy improvement using the Weighted Heston with SA method. Actually this factor can be arbitrarily large since we have proved in section 4.2 that the LSM will finally fail when we increase the number of basis functions to 8^2. However, we still want the LSM works in this case so that the comparison will be more meaningful. Therefore, we choose 4^2 here, which has been shown in Table 14 that LSM algorithm works. From the experiment results, we see that given a time, the accuracy will be increased more than 50 times by replacing the simulation and pricing method to Weighted Heston with SA.

Then we will extend our results by introducing branching algorithm into pricing options. As suggested by Kouritizin [18], the number of particles in combined branching is more stable than in residual branching, which leads to reducing execution time and enhancing the performance of the branching algorithm. Therefore, combined branching will be used here to draw the comparison with the Weighted-SA method in valuing the American put option. We will still fix the computation time around 19.5 and treat the error of Weighted-SA without branching as the base line. The resample parameter for the combined branching is set to be 3.45.

	Weighted -SA	**Branching-SA**
M	5	30
N	$65,000$	$18,000$
J	8^2	6^2
Price	7.932	7.9392
Error	0.0103	0.0034
Time	19.433	19.455
Performance Factor	1	3.03

Table 21: Comparison between Weighted-SA and Branching-SA on American Puts

The performance factor for the branching algorithm is around 3, does not show as impressive as the one in Table 21. However, that might due to the fact that the error of Weighted-SA is already being really small and it requires much more effort to increase the accuracy in this case. Actually, the price estimation for Branching-SA is extremely close to the ground truth.

To get the best pricing algorithm for the path-dependent option pricing problem, we can combine the results from Table 20 and Table 21. To show the results in a more intuitive way, we sum up the performance factor based on the traditional Euler-LSM method.

	Euler-LSM	**Weighted-SA**	**Branching-SA**
Performance Factor	1	55.534	168.27

Table 22: Performance Factor

Apparently, the performance of Branching-SA is orders of magnitude better than the traditional Euler-LSM, which is the most widely-used method today. The pricing accuracy has been improved by around 170 times given the same time using the new Branching-SA algorithm.

Chapter 5

Appendix

5.1 Proof of Theorem 2.1

The prove is mainly taken from Kouritzin**??**. The SDE can be interpreted and solved explicitly either in the strong or weak way. Weak interpretations are often sufficient in applications like mathematical finance and filtering and allow solutions to a greater number of equations than strong solutions. However, there is also the possibility of finding new explicit strong solutions through the guise of weak solutions, which should not be surprising given the result of [24]. Moreover, weak solutions can often be converted to (marginals of) strong solutions of a higher dimension sde, which is the first way that we will use weak interpretations. Our approach will be to show everything explicitly in the case $n = 2$ and then explain the necessary changes for $n \in \{1, 3, 4, ...\}$. However, we first simplify the task by observing the "independently driven" part of the price can be split off.

5.1.1 Price Splitting

Suppose that

$$d\begin{pmatrix} S_t^c \\ V_t \end{pmatrix} = \begin{pmatrix} \mu S_t^c \\ \nu - \varrho V_t \end{pmatrix} dt + \begin{pmatrix} \rho S_t^c V_t^{\frac{1}{2}} \\ \kappa V_t^{\frac{1}{2}} \end{pmatrix} d\widehat{\beta}_t, \tag{5.1}$$

$$S_t^i = \exp\left(\sqrt{1-\rho^2}\int_0^t V_s^{\frac{1}{2}} dB_s - \frac{1-\rho^2}{2}\int_0^t V_s ds\right) \tag{5.2}$$

with respect to independent Brownian motions $\widehat{\beta}, B$. Then, it follows by Itô's formula and the independence of $\widehat{\beta}, B$ that $S_t = S_t^c S_t^i$ and V_t satisfy (1.2) with $\beta = \hat{\beta}$. Moreover, S^i is conditionally (given V) log-normal and hence trivial to simulate. Hence, we only have to solve (5.1), which we do using weak interpretations to create a higher dimension sde that does satisfy (2.8) and hence have an *explicit* strong solution.

5.1.2 Volatility in Case $n = 2$

To ease the notation, we will use Y and Z in place of Y^1, Y^2 in Theorem 2.1. We consider solutions to a Cox-Ingersoll-Ross (CIR) type Ito equation

$$dV_t = (\nu - \varrho V_t)\, dt + \kappa\sqrt{V_t}\, d\widehat{\beta}_t, \tag{5.3}$$

for some Brownian motion $\widehat{\beta}$. Let W^1, W^2 be independent Brownian motions so

$$Y_t = \frac{\kappa}{2}\int_0^t e^{-\frac{\varrho}{2}(t-u)} dW_u^1 + e^{-\frac{\varrho}{2}t} Y_0,\ Z_t = \frac{\kappa}{2}\int_0^t e^{-\frac{\varrho}{2}(t-u)} dW_u^2 + e^{-\frac{\varrho}{2}t} Z_0 \tag{5.4}$$

are independent Ornstein-Uhlenbech processes. It follows by Itô's formula that, *if* Condition (C) is true (with $n = 2$), then $V = Y^2 + Z^2$ satisfies (5.3) with

$$\widehat{\beta}_t = \int_0^t \frac{Y_u}{\sqrt{Y_u^2 + Z_u^2}} dW_u^1 + \int_0^t \frac{Z_u}{\sqrt{Y_u^2 + Z_u^2}} dW_u^2. \tag{5.5}$$

(Note that $(\widehat{\beta}, W)$ is a standard two dimensional Brownian motion, where $W_t = \int_0^t \frac{Z_u}{\sqrt{Y_u^2+Z_u^2}} dW_u^1 - \int_0^t \frac{Y_u}{\sqrt{Y_u^2+Z_u^2}} dW_u^2$, by Levy's characterization.) We call $(V, \widehat{\beta})$ a weak solution since the definition of $\widehat{\beta}$ was part of the solution. V will also be a strong solution if V_t is measurable with respect to $\mathcal{F}_t^{\widehat{\beta}} \doteq \sigma\{\widehat{\beta}_u,\, u \leq t\}$. A strong solution does not immediately follow from the Yamada-Watanabe theorem since the conditions for pathwise uniqueness in e.g. Theorem IX.3.5 of [25] can not immediately be validated. Moreover, explicit form in terms of only $\widehat{\beta}$ is unknown. (Example 3.4 of Kouritzin [21] shows that it unrepresentable in terms of a single Ornstein-Uhlenbeck processs.) Regardless, it is unimportant to us if V is a strong solution or not. [1]

5.1.3 Extended Price Formulation in Case $n = 2$

Recall W^1, W^2 are independent standard Brownian motions, set

$$\sigma(y, z, s) = \begin{bmatrix} \frac{\kappa}{2} & 0 \\ 0 & \frac{\kappa}{2} \\ \rho\, sy & \rho\, sz \end{bmatrix} \tag{5.6}$$

and define a new sde of the form:

$$d\begin{bmatrix} Y_t \\ Z_t \\ S_t^c \end{bmatrix} = \begin{bmatrix} -\frac{\varrho}{2} Y_t \\ -\frac{\varrho}{2} Z_t \\ \mu S_t^c \end{bmatrix} dt + \sigma(Y_t, Z_t, S_t^c) \begin{bmatrix} dW_t^1 \\ dW_t^2 \end{bmatrix}. \tag{5.7}$$

This equation has a unique strong solution. Indeed, the first two rows immediately give strong uniqueness for Y, Z and then S^c is uniquely solved as a

[1] There is a famous example of H. Tanaka of a simple SDE with weak but not strong solutions.

stochastic exponential. This solution can be rewritten as:

$$d\begin{bmatrix} Y_t \\ Z_t \\ S_t^c \end{bmatrix} = \begin{bmatrix} -\frac{\varrho Y_t}{2} \\ -\frac{\varrho Z_t}{2} \\ \mu S_t^c \end{bmatrix} dt + \begin{bmatrix} \frac{\frac{\kappa}{2}\ Z_t}{\sqrt{Y_t^2+Z_t^2}} & \frac{\frac{\kappa}{2}\ Y_t}{\sqrt{Y_t^2+Z_t^2}} \\ \frac{-\frac{\kappa}{2}\ Y_t}{\sqrt{Y_t^2+Z_t^2}} & \frac{\frac{\kappa}{2}\ Z_t}{\sqrt{Y_t^2+Z_t^2}} \\ 0 & \rho S_t^c V_t^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} dW_t \\ d\widehat{\beta}_t \end{bmatrix}, \tag{5.8}$$

where

$$\begin{bmatrix} dW_t \\ d\widehat{\beta}_t \end{bmatrix} = \begin{bmatrix} \frac{Z_t}{\sqrt{Y_t^2+Z_t^2}} & \frac{-Y_t}{\sqrt{Y_t^2+Z_t^2}} \\ \frac{Y_t}{\sqrt{Y_t^2+Z_t^2}} & \frac{Z_t}{\sqrt{Y_t^2+Z_t^2}} \end{bmatrix} \begin{bmatrix} dW_t^1 \\ dW_t^2 \end{bmatrix}. \tag{5.9}$$

Now, the last row of (5.8) together with (5.1,5.2,5.3,5.4,5.5) show that ($S = S^i S^c, V = Y^2 + Z^2$) is the Heston model with $\nu = \frac{\kappa^2}{2}$. Moreover, (5.6) does satisfy (2.8) since

$$(\nabla\sigma_1)\sigma_2 = \begin{pmatrix} 0 \\ 0 \\ \rho^2\, s\, y\, z \end{pmatrix} = (\nabla\sigma_2)\sigma_1 \tag{5.10}$$

so we will be able to look for simple explicit solutions. Our extended Heston system (5.7) can also be written as a Stratonovich equation:

$$d\begin{bmatrix} Y_t \\ Z_t \\ S_t^c \end{bmatrix} = \begin{bmatrix} -\frac{\varrho}{2} Y_t \\ -\frac{\varrho}{2} Z_t \\ \mu S_t^c - \frac{\kappa\rho S_t^c}{2} - S_t^c \rho^2 \frac{Y_t^2+Z_t^2}{2} \end{bmatrix} dt + \begin{bmatrix} \frac{\kappa}{2} & 0 \\ 0 & \frac{\kappa}{2} \\ \rho\, S_t^c Y_t & \rho\, S_t^c Z_t \end{bmatrix} \bullet \begin{bmatrix} dW_t^1 \\ dW_t^2 \end{bmatrix} \tag{5.11}$$

where the stochastic integral implied by the $\bullet$ is now interpretted in the Fisk-Stratonovich sense. We define the full Fisk-Stratonovich drift coefficient to be:

$$h(y,z,s,v) = \begin{bmatrix} -\frac{\varrho}{2} y \\ -\frac{\varrho}{2} z \\ \mu s - \frac{\kappa\rho s}{2} - s\rho^2 \frac{y^2+z^2}{2} \end{bmatrix}. \tag{5.12}$$

Reformulating the Heston equations into a higher dimensional equation so that commutator conditions like (5.10) are true and explicit solutions exist is one of our main contributions. It is believed that similar techniques can be used on some other interesting financial models.

5.1.4 Explicit Solutions for Extended Heston in case $n = 2$

We can solve for the possible strong solutions to (5.8). The first step is to transform the equation to a simpler one using Theorem 2 of [20], restated here in the case $p = 3$ and $d = r = 2$ for convenience:

Theorem 5.1. *Let $D \subset \mathbb{R}^3$ be a bounded convex domain, X_0 be a random variable living in D, W be an $\mathbb{R}^2$-valued standard Brownian motion and $h : D \to \mathbb{R}^3$, $\sigma : D \to \mathbb{R}^{3\times 2}$ be twice continuously differentiable functions with $\sigma(X_0)$ having full rank and satisfying (2.8). Then, the Stratonovich SDE $dX_t = h(X_t)dt + \sigma(X_t) \bullet dW_t$ has a solution $X_t = \Lambda^{-1}\begin{pmatrix} \overline{X}_t \\ \widehat{X}_t \end{pmatrix}$ on $[0, \tau]$ for some stopping time $\tau > 0$, in terms of a simpler SDE*

$$\begin{bmatrix} \overline{X}_t \\ \widehat{X}_t \end{bmatrix} = \int_0^t \widehat{h}\begin{pmatrix} \overline{X}_s \\ \widehat{X}_s \end{pmatrix} ds + \begin{pmatrix} W_t \\ 0 \end{pmatrix} + \Lambda(X_0), \quad \text{with } \widehat{h}(x) = (\nabla\Lambda h) \circ \Lambda^{-1}(x),$$

and a local diffeomorphism Λ if and only if the simpler SDE has a solution up to a stopping time at least as large as τ. Without loss of generality, the local diffeomorphism can have the form $\Lambda = \Lambda_2 \circ \Lambda_1$ for any local diffeomorphisms $\Lambda_1 : D \to \mathbb{R}^3$ satisfying $\nabla\Lambda_1\sigma_1 \circ \Lambda_1^{-1}(x) = e_1$ and $\Lambda_2 : \Lambda_1(D) \to \mathbb{R}^3$ satisfying $\{\nabla\Lambda_2\nabla\Lambda_1\sigma_2\} \circ (\Lambda_1^{-1} \circ \Lambda_2^{-1}(x)) = e_2$, where $(e_1\, e_2\, e_3) = I_3$ is the identity matrix.

There are three things to note:

1. The diffusion coefficient is just $\begin{pmatrix} I_2 \\ 0 \end{pmatrix}$ for the simpler SDE.

2. We can check this local solution to see if it is actually a global solution. We will do this below and determine that it is a global solution in our case.

3. We can check $\widehat{h}$ to see if these equations are solvable. We will do this below and actually solve the simplified SDE and the diffeomorphism in the extended Heston case.

4. It is shown in [20] that (2.8) is also necessary if want to have such local solutions for all initial random variables X_0.

In our Heston case $\overline{X} = \left(\overline{Y}, \overline{Z}\right)'$ and $\widehat{X} = \widehat{S}^c$ and we can use Theorem 5.1 to obtain:

Theorem 5.2. *Suppose* $(W^1, W^2)'$ *is a standard* $\mathbb{R}^2$*-valued Brownian motion and* $\left(\overline{Y}_t, \overline{Z}_t, \widehat{S}^c_t\right)'$ *is the strong solution to:*

$$d\begin{bmatrix} \overline{Y}_t \\ \overline{Z}_t \end{bmatrix} = \begin{bmatrix} -\frac{\varrho}{2}\overline{Y}_t \\ -\frac{\varrho}{2}\overline{Z}_t \end{bmatrix} dt + d\begin{bmatrix} W^1_t \\ W^2_t \end{bmatrix},$$

$$d\widehat{S}^c_t = \widehat{S}^c_t\left[\mu - \frac{\kappa\rho}{2} + \left[\frac{\kappa\rho\varrho}{4} - \frac{\kappa^2\rho^2}{8}\right]\left\{\overline{Y}^2_t + \overline{Z}^2_t\right\}\right] dt.$$

Then, $\begin{bmatrix} Y_t \\ Z_t \\ S^c_t \end{bmatrix} = \Lambda^{-1}\begin{pmatrix} \overline{Y}_t \\ \overline{Z}_t \\ \widehat{S}^c_t \end{pmatrix}$ *with* $\begin{pmatrix} W^1_t \\ W^2_t \end{pmatrix}$ *satisfies (5.8,5.9), where*

$$\Lambda(x) = \begin{bmatrix} \frac{2}{\kappa}x_1 \\ \frac{2}{\kappa}x_2 \\ x_3\exp\left(-\frac{\rho}{\kappa}(x_1^2 + x_2^2)\right) \end{bmatrix},\ \Lambda^{-1}(x) = \begin{bmatrix} \frac{\kappa}{2}x_1 \\ \frac{\kappa}{2}x_2 \\ x_3\exp\left(\rho\frac{\kappa}{4}(x_1^2 + x_2^2)\right) \end{bmatrix}, \tag{5.13}$$

is a $\mathcal{C}^2$*-diffeomorphism on* $\mathbb{R} \times \mathbb{R} \times (0, \infty)$.

We do not need Condition (C) for this theorem nor even for the solution of price S in terms of V below. We only need this condition to express the volatility in terms of the sums of squares of independent Ornstein-Uhlenbeck processes. We only really care that we have a solution for the last rows of (5.8,5.9) but we have to solve for all rows and then later throw away the unnecessary ones. $\overline{Y}$ and $\overline{Z}$ are independent Ornstein-Uhlenbeck processes while $\widehat{S}^c$ just solves a linear ordinary differential equation (with coefficients depending upon the random processes $\overline{Y}, \overline{Z}$). Hence, simulation and calculation is made easy by the explicit form of the diffeomorphism and its inverse. Notice that $\widehat{S}^c$ has finite variation while S^c does not. The explanation for this is that the diffeomorphism Λ^{-1} brings $\overline{Y}$ and $\overline{Z}$ into the solution for S^c and thereby handles the quadratic variation.

Proof. The idea is to find the diffeomorphisms Λ_1, Λ_2 in Theorem 5.1. Solving $\frac{d}{dt}\theta(t;x) = \sigma_1(\theta(t;x))$ leads to

$$\frac{d}{dt}\theta(t;x) = \begin{bmatrix} \frac{\kappa}{2} \\ 0 \\ \rho\,\theta_1(t;x)\theta_3(t;x) \end{bmatrix} \text{ subject to } \theta(0;x) = \begin{bmatrix} 0 \\ x_2 \\ x_3 \end{bmatrix}, \tag{5.14}$$

and we find that $\theta_1(t;x) = \frac{\kappa}{2}t;\ \theta_2(t;x) = x_2;\ \theta_3(t;x) = x_3 \exp\left(\frac{\rho\kappa}{4}\, t^2\right)$. Substituting $t = x_1$ in, we have that

$$\Lambda_1^{-1}(x) = \begin{bmatrix} \frac{\kappa}{2}\, x_1 \\ x_2 \\ x_3 \exp\left(\frac{\rho\kappa}{4}\, x_1^2\right) \end{bmatrix}, \tag{5.15}$$

which has inverse

$$\Lambda_1(y) = \begin{bmatrix} \frac{2}{\kappa}\, y_1 \\ y_2 \\ y_3 \exp\left(-\frac{\rho}{\kappa}\, y_1^2\right) \end{bmatrix}. \tag{5.16}$$

Next, it follows that

$$\nabla\Lambda_1(y) = \begin{bmatrix} \frac{2}{\kappa} & 0 & 0 \\ 0 & 1 & 0 \\ -2\frac{\rho}{\kappa}\, y_1 y_3 \exp\left(-\frac{\rho}{\kappa}\, y_1^2\right) & 0 & \exp\left(-\frac{\rho}{\kappa}\, y_1^2\right) \end{bmatrix} \tag{5.17}$$

so $\widehat{\sigma}_1(x) = \{\nabla\Lambda_1\sigma_1\}(\Lambda_1^{-1}x) = e_1$ and we have found our first diffeomorphism in Theorem 5.1. To find the second diffeomorphism, we set

$$\alpha_2(x) = \{\nabla\Lambda_1\sigma_2\}(\Lambda_1^{-1}x) = \begin{bmatrix} 0 \\ \frac{\kappa}{2} \\ \rho\; x_2 x_3 \end{bmatrix}. \tag{5.18}$$

Then, solving $\frac{d}{dt}\theta(t;x) = \alpha_2(\theta(t;x))$ leads to

$$\frac{d}{dt}\theta(t;x) = \begin{bmatrix} 0 \\ \frac{\kappa}{2} \\ \rho\,\theta_2(t;x)\,\theta_3(t;x) \end{bmatrix} \text{ s.t. } \theta(0;x) = \begin{bmatrix} x_1 \\ 0 \\ x_3 \end{bmatrix}, \tag{5.19}$$

and we find that $\theta_1(t;x) = x_1;\;\; \theta_2(t;x) = \frac{\kappa}{2}t;\;\; \theta_3(t;x) = x_3 \exp\left(\frac{\rho\kappa}{4}\, t^2\right)$. Substituting $t = x_2$ in and taking the inverse, we have that

$$\Lambda_2^{-1}(x) = \begin{bmatrix} x_1 \\ \frac{\kappa}{2}x_2 \\ x_3 \exp\left(\frac{\rho\kappa}{4}\, x_2^2\right) \end{bmatrix},\;\; \Lambda_2(y) = \begin{bmatrix} y_1 \\ \frac{2}{\kappa}\, y_2 \\ y_3 \exp\left(-\frac{\rho}{\kappa}\, y_2^2\right) \end{bmatrix}. \tag{5.20}$$

Next, it follows that

$$\nabla\Lambda_2(y) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{2}{\kappa} & 0 \\ 0 & -2\frac{\rho}{\kappa}\, y_2\, y_3 \exp\left(-\frac{\rho}{\kappa}\, y_2^2\right) & \exp\left(-\frac{\rho}{\kappa}\, y_2^2\right) \end{bmatrix} \tag{5.21}$$

so $\widehat{\sigma}_2(x) = \{\nabla\Lambda_2\alpha_2\}(\Lambda_2^{-1}x) = e_2$ and we indeed have our second homeomorphism in Theorem 5.1. Now, we find $\Lambda = \Lambda_2 \circ \Lambda_1$ gives the diffeomorphism in

(5.13) and

$$\nabla\Lambda(y) = \begin{bmatrix} \frac{2}{\kappa} & 0 & 0 \\ 0 & \frac{2}{\kappa} & 0 \\ \frac{-2\frac{\rho}{\kappa}y_1y_3}{\exp\left(\frac{\rho}{\kappa}(y_1^2+y_2^2)\right)} & \frac{-2\frac{\rho}{\kappa}y_2y_3}{\exp\left(\frac{\rho}{\kappa}(y_1^2+y_2^2)\right)} & \frac{1}{\exp\left(\frac{\rho}{\kappa}(y_1^2+y_2^2)\right)} \end{bmatrix} \tag{5.22}$$

so $\widehat{h}(x) \doteq (\nabla\Lambda)h \circ \Lambda^{-1}(x)$ in Theorem 5.1 satisfies

$$\widehat{h}(x) = \begin{bmatrix} -\frac{\varrho}{2}x_1 \\ -\frac{\varrho}{2}x_2 \\ x_3\Big[\mu - \frac{\kappa\rho}{2} + \Big[\frac{\kappa\rho\varrho}{4} - \frac{\kappa^2\rho^2}{8}\Big]\{x_1^2+x_2^2\}\Big] \end{bmatrix}. \tag{5.23}$$

□

5.1.5 Finishing Proof of Theorem 2.1 by Solving Equations in case $n = 2$

The solution for $\left(\overline{Y}_t, \overline{Z}_t, \widehat{S}_t^c\right)'$ in Theorem 5.2 is: $\overline{Y}_t = \int_0^t e^{-\frac{\varrho}{2}(t-u)}dW_u^1 + e^{-\frac{\varrho}{2}t}\overline{Y}_0$, $\overline{Z}_t = \int_0^t e^{-\frac{\varrho}{2}(t-u)}dW_u^2 + e^{-\frac{\varrho}{2}t}\overline{Z}_0$ (with $\overline{Y}_0^2 + \overline{Z}_0^2 = \frac{\kappa^2}{4}V_0$ to be consistent with (5.3,5.4)), and

$$\widehat{S}_t^c = \widehat{S}_0^c \exp\left(\left[\mu - \frac{\kappa\rho}{2}\right]t + \left[\frac{\kappa\rho\varrho}{4} - \frac{\kappa^2\rho^2}{8}\right]\int_0^t \left\{\overline{Y}_s^2 + \overline{Z}_s^2\right\}ds\right). \tag{5.24}$$

Moreover, it follows by (5.13) and (5.4) that

$$S_t^c = \widehat{S}_t^c \exp\left(\frac{\rho\kappa}{4}(\overline{Y}_t^2 + \overline{Z}_t^2)\right) = \widehat{S}_t^c \exp\left(\frac{\rho}{\kappa}(Y_t^2 + Z_t^2)\right) = \widehat{S}_t^c \exp\left(\frac{\rho}{\kappa}V_t\right)$$

and it follows by (5.24), Theorem 5.2, (5.13) and substitution that

$$\begin{aligned} S_t^c =& S_0^c \exp\Bigg(\left[\mu - \frac{\kappa\rho}{2}\right] t + \left[\frac{\kappa\rho\varrho}{4} - \frac{\kappa^2\rho^2}{8}\right] \int_0^t \left\{\overline{Y}_s^2 + \overline{Z}_s^2\right\} ds \\ &+ \frac{\rho}{\kappa}(V_t - V_0)\Bigg) \qquad (5.25) \\ =& S_0^c \exp\Bigg(\left[\mu - \frac{\kappa\rho}{2}\right] t + \left[\frac{\rho\varrho}{\kappa} - \frac{\rho^2}{2}\right] \int_0^t V_s ds + \frac{\rho}{\kappa}(V_t - V_0)\Bigg). \end{aligned}$$

We also get a solution for the simplified Heston (2.12) by computing

$$S_t^i = \exp\left(\sqrt{1-\rho^2} \int_0^t V_s^{\frac{1}{2}} dB_s - \frac{1-\rho^2}{2} \int_0^t V_s ds\right) \qquad (5.26)$$

and then multiplying $S_t = S_t^c S_t^i$ to get (2.9) of Theorem 2.1 in the case $n = 2$. □

5.1.5.1 Case $n \neq 2$

Insomuch as the guess and check proof of Theorem 2.1 is as simple as Itô's formula, our real goal here is to motivate how this solution was actually arrived at and how weak solutions for other models might be found. With this easy Ito lemma test, a formal proof along these lines is less important. Hence, we have given all the steps just in the case $n = 2$ and we will just explain the differences required for the case $n \neq 2$ instead of going through the formal proof with these methods.

The price splitting was already done in general. There is no change there.

For the volatility in the case $n \in \{1, 3, 4, ...\}$, we start with n independent standard Brownian motions $W^1, ..., W^n$ and follow Subsection 5.1.2. The differences are: We replace Y, Z with $\{Y_t^i = \frac{\kappa}{2}\int_0^t e^{-\frac{\varrho}{2}(t-u)} dW_u^i + e^{-\frac{\varrho}{2}t} Y_0^i\}_{i=1}^n$ and set

$$\widehat{\beta}_t = \sum_{i=1}^n \int_0^t \frac{Y_u^i}{\sqrt{\sum_{j=1}^n (Y_u^j)^2}} dW_u^i \qquad (5.27)$$

to find that $V = \sum\limits_{i=1}^n (Y^i)^2$ satisfies (5.3) when $\nu = \frac{n\kappa^2}{4}$ (and $V_0 = \sum\limits_{i=1}^n (Y_0^i)^2$).

For the extended price formulation when $n \in \{1, 3, 4, ...\}$, we set

$$\sigma(y_1, ..., y_n, s) = \begin{bmatrix} \frac{\kappa}{2} & 0 & 0 & \cdots & 0 \\ 0 & \frac{\kappa}{2} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \frac{\kappa}{2} & 0 \\ 0 & 0 & \cdots & 0 & \frac{\kappa}{2} \\ s\rho\, y_1 & s\rho\, y_2 & \cdots & s\rho\, y_{n-1} & s\rho\, y_n \end{bmatrix} \tag{5.28}$$

and find $\nabla\sigma_i\sigma_j = (0, ..., 0, s\rho^2 y_i y_j)'$ for $i \neq j$ so (2.8) clearly holds. (For clarity, $\sigma = (\frac{\kappa}{2}, s\rho y_1)'$ when $n = 1$.) Now, define a new sde of the form:

$$d\begin{bmatrix} Y_t^1 \\ \vdots \\ Y_t^n \\ S_t^c \end{bmatrix} = \begin{bmatrix} -\frac{\varrho}{2} Y_t^1 \\ \vdots \\ -\frac{\varrho}{2} Y_t^n \\ \mu S_t^c \end{bmatrix} dt + \sigma(Y_t^1, ..., Y_t^n, S_t^c) \begin{bmatrix} dW_t^1 \\ \vdots \\ dW_t^n \end{bmatrix}. \tag{5.29}$$

This equation has a unique strong solution and it can be rewritten by post-multiplying σ by OO^{-1}, where

$$O = \begin{bmatrix} \frac{Y_t^n}{\sqrt{V_t}} & 0 & \cdots & 0 & \frac{Y_t^1}{\sqrt{V_t}} \\ 0 & \frac{Y_t^n}{\sqrt{V_t}} & \cdots & 0 & \frac{Y_t^2}{\sqrt{V_t}} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \frac{Y_t^n}{\sqrt{V_t}} & \frac{Y_t^{n-1}}{\sqrt{V_t}} \\ -\frac{Y_t^1}{\sqrt{V_t}} & -\frac{Y_t^2}{\sqrt{V_t}} & \cdots & -\frac{Y_t^{n-1}}{\sqrt{V_t}} & \frac{Y_t^n}{\sqrt{V_t}} \end{bmatrix}, \tag{5.30}$$

and (abusing notation by letting $Y_i = Y_t^i$)

$$O^{-1} = \begin{bmatrix} \frac{Y_2^2+\cdots+Y_n^2}{Y_n\sqrt{V_t}} & -\frac{Y_1Y_2}{Y_n\sqrt{V_t}} & -\frac{Y_1Y_3}{Y_n\sqrt{V_t}} & \cdots & -\frac{Y_1Y_{n-1}}{Y_n\sqrt{V_t}} & -\frac{Y_1}{\sqrt{V_t}} \\ -\frac{Y_1Y_2}{Y_n\sqrt{V_t}} & \frac{Y_1^2+Y_3^2+\cdots+Y_n^2}{Y_n\sqrt{V_t}} & -\frac{Y_2Y_3}{Y_n\sqrt{V_t}} & \cdots & -\frac{Y_2Y_{n-1}}{Y_n\sqrt{V_t}} & -\frac{Y_2}{\sqrt{V_t}} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ -\frac{Y_1Y_{n-1}}{Y_n\sqrt{V_t}} & -\frac{Y_2Y_{n-1}}{Y_n\sqrt{V_t}} & -\frac{Y_3Y_{n-1}}{Y_n\sqrt{V_t}} & \cdots & \frac{Y_1^2+\cdots+Y_{n-2}^2+Y_n^2}{Y_n\sqrt{V_t}} & -\frac{Y_{n-1}}{\sqrt{V_t}} \\ \frac{Y_1}{\sqrt{V_t}} & \frac{Y_2}{\sqrt{V_t}} & \frac{Y_3}{\sqrt{V_t}} & \cdots & \frac{Y_{n-1}}{\sqrt{V_t}} & \frac{Y_n}{\sqrt{V_t}} \end{bmatrix} \tag{5.31}$$

as:

$$d\begin{bmatrix} Y_t^1 \\ \vdots \\ Y_t^n \\ S_t^c \end{bmatrix} = \begin{bmatrix} -\frac{\varrho Y_t^1}{2} \\ \vdots \\ -\frac{\varrho Y_t^n}{2} \\ \mu S_t^c \end{bmatrix} dt + \begin{bmatrix} \frac{\kappa}{2}\frac{Y_t^n}{\sqrt{V_t}} & 0 & \cdots & 0 & \frac{\kappa}{2}\frac{Y_t^1}{\sqrt{V_t}} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \frac{\kappa}{2}\frac{Y_t^n}{\sqrt{V_t}} & \frac{\kappa}{2}\frac{Y_t^{n-1}}{\sqrt{V_t}} \\ 0 & 0 & \cdots & 0 & \rho S_t^c V_t^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} dA_t^1 \\ \vdots \\ dA_t^{n-1} \\ d\widehat{\beta}_t \end{bmatrix}, \tag{5.32}$$

where $(A^1, ..., A^{n-1}, \widehat{\beta})' = O^{-1}(W^1, ..., W^n)'$ so $\widehat{\beta}$ does satisfy (5.27). This extended Heston solution (5.29) can also be written in Fisk-Stratonovich form as

$$d\begin{bmatrix} Y_t^1 \\ \vdots \\ Y_t^n \\ S_t^c \end{bmatrix} = \begin{bmatrix} -\frac{\varrho}{2}Y_t^1 \\ \vdots \\ -\frac{\varrho}{2}Y_t^n \\ \left(\mu - \frac{n\kappa\rho}{4}\right)S_t^c - S_t^c\rho^2\frac{(Y_t^1)^2+\cdots+(Y_t^n)^2}{2} \end{bmatrix} dt + \sigma(Y_t^1, ..., Y_t^n, S_t^c) \bullet \begin{bmatrix} dW_t^1 \\ \vdots \\ dW_t^n \end{bmatrix}, \tag{5.33}$$

from which we can apply Proposition 2 of [20] (knowing (2.8) holds) in the case $p = n+1$ and $d = r = n$ to find (5.33) has a strong solution up to some stopping time $\tau > 0$ if and only if

$$d\begin{bmatrix} \overline{Y}_t^1 \\ \vdots \\ \overline{Y}_t^n \end{bmatrix} = \begin{bmatrix} -\frac{\varrho}{2}\overline{Y}_t^1 \\ \vdots \\ -\frac{\varrho}{2}\overline{Y}_t^n \end{bmatrix} dt + d\begin{bmatrix} W_t^1 \\ \vdots \\ W_t^n \end{bmatrix}, \tag{5.34}$$

$$\begin{aligned} d\widehat{S}^c_t =&\widehat{S}^c_t\Big[\mu - \frac{n\kappa\rho}{4} + \Big[\frac{\kappa\rho\varrho}{4} - \frac{\kappa^2\rho^2}{8}\Big] \\ &\times \{\Big(\overline{Y}^1_t\Big)^2 + \cdots + \big(\overline{Y}^n_t\big)^2 dt \end{aligned} \tag{5.35}$$

does. Moreover, the solutions to (5.33) and (5.34,5.35) satisfy $\begin{bmatrix} Y^1_t \\ \vdots \\ Y^n_t \\ S^c_t \end{bmatrix} = \Lambda^{-1}\begin{pmatrix} \overline{Y}^1_t \\ \vdots \\ \overline{Y}^n_t \\ \widehat{S}^c_t \end{pmatrix}$, where $\mathcal{C}^2$-diffeomorphism Λ is given by

$$\Lambda(x) = \begin{bmatrix} \frac{2}{\kappa}\,x_1 \\ \vdots \\ \frac{2}{\kappa}\,x_n \\ x_{n+1}\exp\left(-\frac{\rho}{\kappa}(x_1^2 + \cdots + x_n^2)\right) \end{bmatrix},\ \Lambda^{-1}(x) = \begin{bmatrix} \frac{\kappa}{2}\,x_1 \\ \vdots \\ \frac{\kappa}{2}\,x_n \\ x_{n+1}\exp\left(\rho\frac{\kappa}{4}(x_1^2 + \cdots + x_n^2)\right) \end{bmatrix}. \tag{5.36}$$

The solution to (5.34,5.35) is then

$$\overline{Y}^i_t = \int_0^t e^{-\frac{\varrho}{2}(t-u)}dW^i_u + e^{-\frac{\varrho}{2}t}\overline{Y}^i_0,\ i = 1, ..., n \text{ and} \tag{5.37}$$

$$\widehat{S}^c_t = \widehat{S}^c_0 \exp\Bigg(\Big[\mu - \frac{n\kappa\rho}{4}\Big]t + \Big[\frac{\kappa\rho\varrho}{4} - \frac{\kappa^2\rho^2}{8}\Big]\int_0^t \Big\{\Big(\overline{Y}^1_s\Big)^2 + \cdots + \big(\overline{Y}^n_s\big)^2\Big\}\,ds\Bigg) \tag{5.38}$$

from which it follows using (5.36) that

$$S^c_t = S^c_0 \exp\Bigg(\Big[\mu - \frac{n\kappa\rho}{4}\Big]t + \Big[\frac{\rho\varrho}{\kappa} - \frac{\rho^2}{2}\Big]\int_0^t V_s ds + \frac{\rho}{\kappa}(V_t - V_0)\Bigg) \tag{5.39}$$

with $V_t = \frac{\kappa^2}{4}\Big\{\Big(\overline{Y}^1_t\Big)^2 + \cdots + \big(\overline{Y}^n_t\big)^2\Big\}$. The result follows by multiplying $S_t = S^i_t S^c_t$ and Itô's formula. □

5.2 Proof of Theorem 2.2

By Theorem 2.1, $(\widehat{S},\widehat{V})$, defined in (2.13,2.14) satisfies the Heston model with parameters ν_κ,μ_κ defined in (2.12). Hence

$$\begin{aligned} M_t(f) \;=\; & f(\widehat{S}_t,\widehat{V}_t)-\int_0^t \mu_\kappa\widehat{S}_u\,\partial_s f(\widehat{S}_u,\widehat{V}_u)+(\nu_\kappa-\varrho\widehat{V}_u)\partial_v f(\widehat{S}_u,\widehat{V}_u) \\ &+\;\frac{1}{2}\widehat{S}_u^2\widehat{V}_u\,\partial_s^2 f(\widehat{S}_u,\widehat{V}_u)+\rho\kappa\widehat{S}_u\widehat{V}_u\,\partial_s\partial_v f(\widehat{S}_u,\widehat{V}_u)+\frac{1}{2}\kappa^2\widehat{V}_u\,\partial_v^2 f(\widehat{S}_u,\widehat{V}_u)du \end{aligned} \tag{5.40}$$

(for $f\in\mathcal{S}(\mathbb{R}^2)$, the rapidly decreasing functions) has the following P-martingale representation

$$\begin{aligned} M_t(f)=&\int_0^t[\kappa\partial_v f(\widehat{S}_u,\widehat{V}_u)+\rho\widehat{S}_u\partial_s f(\widehat{S}_u,\widehat{V}_u)]\widehat{V}_u^{\frac{1}{2}}\,d\widehat{\beta}_u \\ &+\int_0^t\sqrt{1-\rho^2}\widehat{S}_u\partial_s f(\widehat{S}_u,\widehat{V}_u)\widehat{V}_u^{\frac{1}{2}}\,dB_u\quad\text{with } \widehat{\beta}_t=\sum_{i=1}^n\int_0^t\frac{Y_u^i}{\sqrt{\sum_{j=1}^n(Y_u^j)^2}}dW_u^i. \end{aligned} \tag{5.41}$$

Separately, it follows by Itô's formula and (2.12) that

$$\ln(\widehat{V}_t)-\ln(\widehat{V}_0)=\int_0^t\frac{\nu_\kappa-\varrho\widehat{V}_s}{\widehat{V}_s}ds+\int_0^t\frac{\kappa}{\widehat{V}_s^{\frac{1}{2}}}d\widehat{\beta}_s-\frac{1}{2}\int_0^t\frac{\kappa^2}{\widehat{V}_s}ds \tag{5.42}$$

so, using (**??**), (2.15) is equivalent to

$$\widehat{L}_t=\exp\left\{\int_0^t\frac{\nu-\nu_\kappa}{\kappa\widehat{V}_s^{\frac{1}{2}}}d\widehat{\beta}_s-\frac{1}{2}\int_0^t\frac{|\nu-\nu_\kappa|^2}{\kappa^2\widehat{V}_s}ds\right\}. \tag{5.43}$$

It follows from (5.43) and the Novikov condition that $t\to\widehat{L}_t^{\eta_\varepsilon}\doteq\widehat{L}_{\eta_\varepsilon\wedge t}$ is an L^r-martingale for any $r>0$. This fact will be used in the development below and to conclude $m_t(f)$ is a martingale versus just a local martingale. Next, it follows by (5.41), Itô's formula, (2.12) and the fact $d\widehat{L}_t=\widehat{L}_t\frac{\nu-\nu_\kappa}{\kappa}\widehat{V}_t^{-\frac{1}{2}}d\widehat{\beta}_t$ (by

(5.43)) that the quadratic covariance satisfies

$$\begin{aligned}[\widehat{L}^{\eta_\varepsilon}, f(\widehat{S},\widehat{V})]_t &= \int_0^{t\wedge\eta_\varepsilon} \widehat{L}_u^{\eta_\varepsilon} \frac{\nu-\nu_\kappa}{\kappa} \widehat{V}_u^{-\frac{1}{2}} \Big[\kappa \partial_v f(\widehat{S}_u,\widehat{V}_u) + \rho \widehat{S}_u \partial_s f(\widehat{S}_u,\widehat{V}_u)\Big] \widehat{V}_u^{\frac{1}{2}} du \\ &= \int_0^{t\wedge\eta_\varepsilon} \widehat{L}_u^{\eta_\varepsilon} \Big[(\nu-\nu_\kappa)\partial_v f(\widehat{S}_u,\widehat{V}_u) + (\mu-\mu_\kappa)\widehat{S}_u \partial_s f(\widehat{S}_u,\widehat{V}_u)\Big]\, du.\end{aligned} \tag{5.44}$$

Now, it follows by (5.40,5.44) and integration by parts that

$$\begin{aligned} m_t(f) = \widehat{L}_t^{\eta_\varepsilon} f(\widehat{S}_t,\widehat{V}_t) &- \int_0^{t\wedge\eta_\varepsilon} \widehat{L}_u^{\eta_\varepsilon} \big[\mu \widehat{S}_u\, \partial_s f(\widehat{S}_u,\widehat{V}_u) + (\nu - \varrho \widehat{V}_u)\partial_v f(\widehat{S}_u,\widehat{V}_u)\big] du \\ &- \int_{t\wedge\eta_\varepsilon}^{t} \widehat{L}_u^{\eta_\varepsilon} \big[\mu_\kappa \widehat{S}_u\, \partial_s f(\widehat{S}_u,\widehat{V}_u) + (\nu_\kappa - \varrho \widehat{V}_u)\partial_v f(\widehat{S}_u,\widehat{V}_u)\big] du \\ &- \int_0^t \widehat{L}_u^{\eta_\varepsilon} \left[\frac{1}{2}\widehat{S}_u^2 \widehat{V}_u\, \partial_s^2 f(\widehat{S}_u,\widehat{V}_u) + \rho\kappa \widehat{S}_u \widehat{V}_u\, \partial_s\partial_v f(\widehat{S}_u,\widehat{V}_u) + \frac{1}{2}\kappa^2 \widehat{V}_u\, \partial_v^2 f(\widehat{S}_u,\widehat{V}_u)\right] du \end{aligned} \tag{5.45}$$

is a local martingale, which by (5.41) has form

$$\begin{aligned} m_t(f) = &\int_0^t \widehat{L}_u^{\eta_\varepsilon} [\kappa \partial_v f(\widehat{S}_u,\widehat{V}_u) + \rho \widehat{S}_u \partial_s f(\widehat{S}_u,\widehat{V}_u) + \frac{\nu-\nu_\kappa}{\kappa \widehat{V}_u} f(\widehat{S}_u,\widehat{V}_u)] \widehat{V}_u^{\frac{1}{2}} d\widehat{\beta}_u \\ &+ \int_0^t \widehat{L}_u^{\eta_\varepsilon} \sqrt{1-\rho^2}\, \widehat{S}_u \partial_s f(\widehat{S}_u,\widehat{V}_u) \widehat{V}_u^{\frac{1}{2}} dB_u. \end{aligned} \tag{5.46}$$

(Since we have used other randomness to create the $\{Y^i\}_{i=1}^n$ we can not conclude that $m_t(f)$ is adapted to the filtration generated by β, B but it is adapted to the filtration created by $B, W^1, ..., W^n$.)

Now, $\widehat{L}_t^{\eta_\varepsilon}$ and $m_t^{\eta_\varepsilon}(f) \doteq m_{t\wedge\eta_\varepsilon}(f)$ are martingales so one has by (5.45) and Fubini's theorem that

$$\begin{aligned} &\widehat{E}\left[\left(f(\widehat{S}_{t_{n+1}},\widehat{V}_{t_{n+1}}) - f(\widehat{S}_{t_n},\widehat{V}_{t_n}) - \int_{t_n}^{t_{n+1}} A_u f(\widehat{S}_u,\widehat{V}_u) du\right) \prod_{k=1}^n h_k(\widehat{S}_{t_k},\widehat{V}_{t_k})\right] \\ &= E\left[\widehat{L}_T^{\eta_\varepsilon}\left(f(\widehat{S}_{t_{n+1}},\widehat{V}_{t_{n+1}}) - f(\widehat{S}_{t_n},\widehat{V}_{t_n}) - \int_{t_n}^{t_{n+1}} A_u f(\widehat{S}_u,\widehat{V}_u) du\right) \prod_{k=1}^n h_k(\widehat{S}_{t_k},\widehat{V}_{t_k})\right] \\ &= E\left[\left(m_{t_{n+1}}(f) - m_{t_n}(f)\right) \prod_{k=1}^n h_k(\widehat{S}_{t_k},\widehat{V}_{t_k})\right] = 0, \end{aligned} \tag{5.47}$$

for all $0 \leq t_1 < t_2 < \cdots < t_n < t_{n+1}$, $f \in \mathcal{S}(\mathbb{R}^2)$ and $h_1, ..., h_n \in B(\mathbb{R}^2)$ (the bounded, measurables), where

$$\begin{aligned} A_u f(s,v) &= [\mu s \partial_s f(s,v) + (\nu - \varrho v)\partial_v f(s,v)] 1_{[0,\eta_\varepsilon]}(u) \\ &+ [\mu_\kappa s \partial_s f(s,v) + (\nu_\kappa - \varrho v)\partial_v f(s,v)] 1_{[\eta_\varepsilon,T]}(u) \\ &+ \frac{1}{2} s^2 v \partial_s^2 f(s,v) + \rho\kappa \partial_v \partial_s f(s,v) + \frac{\kappa^2}{2} \partial_v^2 f(s,v). \end{aligned} \tag{5.48}$$

Now, it follows by the argument on page 174 of [26] that (S, V) satisfies the A_u-martingale problem with respect to $\widehat{P}$ $\Box$.

Bibliography

[1] Hull J. and White A. The pricing of options on assets with stochastic volatilities. *Journal of Finance*, 42(2):281–300, 1987.

[2] Stein Elias M. and Stein Jeremy C. Stock price distributions with stochastic volatility: An analytic approach. *Review of Financial Studies*, 42(2):727–752, 1991.

[3] Schbel R. and Zhu J. Stochastic volatility with an ornsteinuhlenbeck process: An extension. *European Finance Review*, 3(1):23–46, 1999.

[4] Heston and Steven L. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Operations Research*, 54(2):217–231, 2006.

[5] Kouritzin M. Microstructure models with short-term inertia and stochastic volatility. *Mathematical Problems in Engineering*, 2015, 2015.

[6] Broadie M. and Kaya Ö. Exact simulation of stochastic volatility and other affine jump diffusion processes. *Oper. Res.*, 54(2), March 2006.

[7] Kouritzin M. Path-dependent option pricing with explicit solutions, stochastic approximation and heston example. https://arxiv.org/abs/1608.02028, 2016.

[8] Schwartz and Eduardo S. The valuation of warrants: Implementing a new approach. *Journal of Financial Economics*, 4(1):79–93, 1977.

[9] Howison S. Wilmott P. and Dewynne J. *The Mathematics of Financial Derivatives: A Student Introduction*. Cambridge University Press, 1995.

[10] Ross Stephen A. Cox John C. and Rubinstein Mark. Option pricing: A simplified approach. *Journal of Financial Economics*, 7(3):229–263, 1979.

[11] Francis A. Longstaff and Eduardo S. Schwartz. Valuing american options by simulation: A simple least-squares approach. 14(1):113–147, 2001.

[12] Lamberton D. Clément E. and Protter P. An analysis of a least squares regression method for american option pricing. *Finance and Stochastics*, 6(4):449–471, 2002.

[13] Gordon N.J., Salmond D.J., and Smith A.F.M. Novel approach to nonlinear/non-gaussian bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, 140(2):107–113, Apr 1993.

[14] Liu J.S. and Chen R. Novel approach to nonlinear/non-gaussian bayesian state estimation. *Sequential Monte-Carlo methods for dynamic systems*, 93:1032–1044, 1998.

[15] Kitagawa G. Monte-carlo filter and smoother for non-gaussian nonlinear state space models. *J. Comput. Graph. Statist.*, 1:1–25, 1996.

[16] Douc R. and Cappe O. Comparison of resampling schemes for particle filtering. In *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on*, pages 64–69, 2005.

[17] Clifford P. Carpenter J. and Fearnhead P. An improved particle filter for nonlinear problems. *IEE Proc. Radar Sonar Navigation*, 146:2–7, 1999.

[18] Kouritzin M. Residual and stratified branching particle filters. https://era.library.ualberta.ca/files/sj1394586.VrzwzuYW30E, 2015.

[19] Jackel P. Kahl C. Fast strong approximation monte-carlo schemes for stochastic volatility models. *Quantitative Finance*, 6(6):513–536, 2006.

[20] Kouritzin M. Sadeghi S. Convergence rates and decoupling in linear stochastic approximation algorithms. *SIAM J. Control Optim*, 53(3):1484–1508, 2015.

[21] Kouritzin M. Exact infinite dimensional filters and explicit solutions. *Stochastic Models Eds. Luis G. Gorostiza and B. Gail Ivanoff*, 26:265–282, 2000.

[22] Robbins H. and Monro S. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.

[23] Kourizin M. On the convergence of linear stochastic approximation procedures. *IEEE Trans. Inform. Theory*, 42(4):1305–1309, 1996.

[24] Heunis A. On the prevalence of stochastic differential equations with unique strong solutions. *Ann. Probab*, 14(2):653–662, 1993.

[25] Revuz D. and Yor M. Continuous martingales and brownian motion. *Springer Berlin*, 1991(474), 1991.

[26] Stewart N. Ethier and Thomas G. Kurtz. Markov processes. Characterization and convergence. Wiley Series in Probability and Mathematical Statistics. New York etc.: John Wiley & Sons. X, 534 p. 49.10 (1986)., 1986.