2019

Analysis of CLMR trees for European and Asian option pricing under regime-switching jump-diffusion models

Yaode Sui
suix3850@mylaurier.ca

Analysis of CLMR trees for European and Asian option pricing under regime-switching jump-diffusion models

by

Yaode Sui

Bachelor of Science, Southwestern University of Finance and Economics, 2018

Thesis

Submitted to the Department of Mathematics
Faculty of Science
in partial fulfilment of the requirements for the
Master of Science in Mathematics
for Science and Finance
Wilfrid Laurier University
2019

Abstract

In this paper, we study the convergence rates of the multinomial trees constructed by [*Costabile, Leccadito, Massabó and Russo, Journal of Computational and Applied Mathematics, 256 (2014), 152 - 167*] for European option pricing under the regime-switching jump-diffusion model, which is named as CLMR tree. We also extend the CLMR tree to the pricing of Asian options under the models. Numerical examples are carried out to confirm the theoretical results and the accuracy of computation.

Acknowledgments

First, I would like to express my gratitude towards Dr. Yongzeng Lai for his guidance and support throughout this project. His encouragement fueled my passion to explore and explore with new ideas, giving me a much broader perspective of the field.

Secondly, I thank for Dr. Mark Reezor and Dr.Giuseppe Campolieti. On their financial math class, I learned how to simulate different kinds of stochastic process. Their inspirational and conscientious teaching have provided me with a firm basis for the composing of this paper and will always be of great value to my future academic research.

Finally, I would like to thank my parents and my girlfriend for their loving considerations and great confidence in me through all these years. They motivated me to be a better person every day.

To mom.

Table of Contents

Abstract . . . . **ii**

Acknowledgments . . . . **iii**

Dedication . . . . **iv**

Table of Contents . . . . **v**

List of Tables . . . . **vii**

List of Figures . . . . **viii**

1 Introduction **1**

1.1 Montivation 1

1.2 Literature Review 4

1.3 Contributions of the Thesis 6

1.4 Basic Concepts about Option Pricing 6

2 CLMR trees for European options **14**

3 CLMR trees for Asian options **28**

4 Numerical Results **40**

5 Conclusion and Future Work **47**

5.1 Conclusion 47

5.2 Future Work 48

Notes **49**

Bibliography **50**

Appendix **62**

List of Tables

4.1 The convergence rates of the CLMR tree for Example 4.0.1, the exact option value is 1.8542 for regime 0 and 1.3737 for regime 1. 41

4.2 The convergence rates of the CLMR tree for Example 4.0.2 . . . 41

4.3 Comparisons of CLMR trees and FDMs for pricing European call option under the regime-switching jump-diffusion model (1.1) for Example 4.0.3. 44

4.4 The convergence rates of the CLMR trees for out-the-money Asian call option with $S_0 = 60$. MC values are 2.6418 for regime 0 and 0.9377 for regime 1. 45

4.5 The convergence rates of the CLMR trees for at-the-money Asian call option with $S_0 = 100$. MC values are 19.5855 for regime 0 and 14.8243 for regime 1. 45

4.6 The convergence rates of the CLMR trees for in-the-money Asian call option with $S_0 = 140$. MC values are 51.5671 for regime 0 and 48.8914 for regime 1. 46

4.7 The convergence rates of the CLMR trees for fixed strike Asian call option for Example 4.0.4. 46

List of Figures

1.1 In this figure, the circle represents the time when there is a jump and the crossing represents the time when the regime switches. Regime 1 represents the case of high volatility and regime 2 represents the case of low volatility. 2
1.2 Illustration of a binomial tree 9
1.3 Illustration of a trinomial tree 10

2.1 The approximation of the process (1.1) at the first time step in regime 0 16

Chapter 1

Introduction

1.1 Montivation

The classical Black-Scholes formula for option pricing uses a Geometric Brownian Motion model to capture price dynamics of the underlying asset. However, it is well known that the stochastic variability in the market parameters is not reflected in the BS model. In recent years, considerable attention has been drawn to regime-switching models which can solve this problem. Furthermore, the necessity of taking into account large market movements, as well as a great amount of information arriving suddenly(i.e. a jump) has led to add the jump into the regime-switching model. And that is called regime-switching jump diffusion model. And the following figure gives the simulated path of the stock price under regime-switching jump-diffusion model.

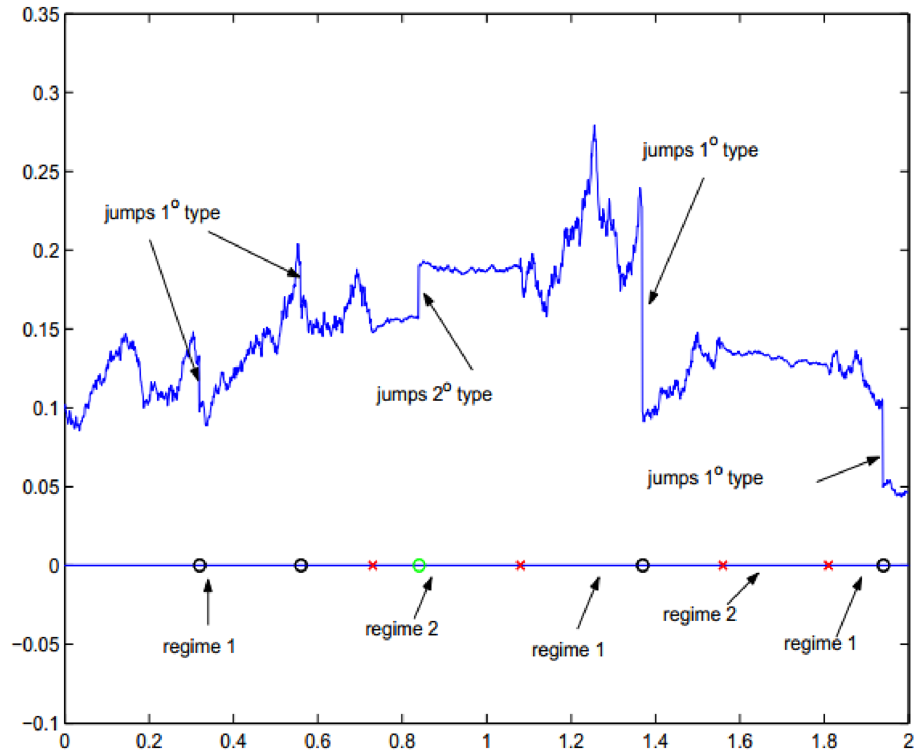

Figure 1.1: In this figure, the circle represents the time when there is a jump and the crossing represents the time when the regime switches. Regime 1 represents the case of high volatility and regime 2 represents the case of low volatility.

Therefore we introduce the mathmatical defination of regime-switching jump-diffusion model.

Let $\alpha(t)$ be a continuous-time, homogeneous and stationary Markov chain on the state space $\mathbb{D} = \{0, 1, \ldots, \mathcal{M}\}$ with a generator A, a $(\mathcal{M}+1)\times(\mathcal{M}+1)$-size real matrix, whose elements are constants satisfying $a_{il} \geq 0$ for $i \neq l$ and $\sum_{l=0}^{\mathcal{M}} a_{il} = 0$ for $i \in \mathbb{D}$. On a filtered probability space $(\Omega, \mathcal{F}, \mathcal{P})$, the asset price follows the following process

$$S(t) = e^{X(t)},$$

where $X(t)$ is specified as the following regime-switching jump-diffusion model,

$$\begin{cases} dX(t) = [\varepsilon_l - \lambda_l m_l]dt + \sigma_l dW(t) + dJ(t), \\ X(0) = \ln S(0), \end{cases} \tag{1.1}$$

where for each regime $l \in \mathbb{D}$, $\varepsilon_l = r_l - \sigma_l^2/2$, $m_l := E[e^{Z_1^l} - 1]$, σ_l is the volatility and $W(t)$ is a standard Brownian motion under the pricing measure. Furthermore, $J(t)$ represents the cumulative jumps by time t,

$$J(t) = \sum_{k=1}^{N(t)} Z_k^{\alpha(\tau_k)},$$

where τ_k denotes the k-th jump time of the Poisson process $N(\cdot)$. The Poisson process $\{N(t),\, t \geq 0\}$ has regime-dependent intensity $\lambda_{\alpha(t)}$, i.e., if the current regime is $\alpha(t) = l$, then the time until the next jump is given by an exponential random variable with mean $1/\lambda_l$. For each $l \in \mathbb{D}$, let Z_k^l, $k \geq 1$ be a sequence of independent identically distributed (i.i.d.) random variables with the known distribution function $F_l(z)$, not necessarily of the normal type, that specifies the jump sizes when the regime is l. Note that here we consider a very general model setup allowing different jump distributions $F_l(\cdot)$ for different regimes l.

1.2 Literature Review

In financial literature, regime-switching models have been prevalently applied in order to allow Lévy processes to switch in a finite state space. Among the contributions in the field of option pricing, it is worth mentioning M. Konikov (2002), who introduce an extension of the variance-gamma model in which the parameters switch, according to a two-state Markov chain, between two fixed sets of values at infinitesimal time intervals. Furthermore, they evidence that more than two states should be considered for the Lévy processes, but the mathematical approach they use cannot easily accommodate more than two states for option valuations. To overcome this limit, Elliott and Osakwe (2006) extend their work to more than two states introducing a multi-state Markov switching model where the underlying process is a jump process with parameters that may switch among drift/compensator pairs. C. Albanese and Rubisov (2003) develop a model similar to the one of M. Konikov (2002) except that switches occur only at finite time intervals, deriving as well closed form formulae for European options. A different approach, which uses a Markov chain with a progressively denser state space to approximate a continuous time stochastic volatility model with jumps, has been proposed by Chourdakis (2004) who in this way obtains option prices in semi-closed form.

Among the contributions in option pricing that consider a regime-switching model with jumps, it is worth mentioning Yuen and Yang (2009) who, after generalizing the Naik (1993)'s model to more than two regimes, provide a trinomial lattice to price options under a jump diffusion Markov regime-switching model. Indeed, as in Naik (1993), the underlying asset process presents jumps only during the switches among states with the jump size depending upon the state before and after the switching and the current asset price. J.X. Jiang and Nguyen (2016) develops a recombining tree method for European option pricing with state-dependent switching rates. J. Ma and Zhu (2018) prove the convergence rates of the tree method in J.X. Jiang and Nguyen (2016). I. Florescu and Sewell (2013) study the system of PIDEs for option pricing in regime-switching jump diffusion model. Furthermore, a more general tree method is proposed by M. Costabile and Russo (2014), who construct a multinomial approach that is flexible enough to accommodate an arbitrary distribution for the jump-components and an arbitrary number of regimes both for the diffusion and the jump component. The method is easily to be applied to price European and American options. As for the pricing of Asian options, Yuen and Yang (2012) construct a trinomial tree method to price the Asian option and equity-indexed annuities with regime-switching models (no jump-diffusion). D. Dang and Sewell (2016) use the system of PIDEs to price the Asian options in regime-switching jump-diffusion models. The trinomial tree method has not been studied in the literature for the Asian option pricing with regime-switching jump-diffusion models.

1.3 Contributions of the Thesis

In this paper, we prove the convergence rates of the CLMR tree proposed by M. Costabile and Russo (2014) by establishing the high-order equivalence with the finite difference methods. Moreover, we apply the CLMR trees to the Asian option pricing. To the best of our knowledge, this paper is the first attempt on the tree methods for Asian option pricing under regime-switching jump-diffusion models.

1.4 Basic Concepts about Option Pricing

Option

This thesis will focus on only one specific type of derivatives - options. These are contracts that give the holder the right to buy or sell an underlying asset at a certain point in time for a certain price, both specified when purchasing the option. This is in contrast with other derivatives - forwards and futures, where the holder is obligated to buy or sell the underlying asset.

We identify two types of options. A call option gives the holder the right to buy the underlying asset by a certain date for a certain price. A put option gives the holder the right to sell the underlying asset by a certain date for a certain price. The price in the contract is called strike price and the expiration date is called maturity.

European Option

A European Call Option is a financial contract that gives its holder the right to buy an asset for a prescribed price at a prescribed future date. On the contrary, a European Put Option gives the holder the right to sell the asset for a prescribed price at a prescribed future date. When an option is being traded, it involves two parties, namely the writer and the holder. The writer of a call option must sell the asset if the holder chooses to exercise the option. Similarly, the writer of the put option is obliged to buy the asset if the holder of the put chooses to exercise the right to sell the asset.

If the prescribed time for the European call option is T, the strike price is K and the price of asset is S_T, then the holder will buy the asset if $S_T > K$, otherwise they do not exercise the option. The holder realises a profit of $S_T - K$ by buying the asset for K, and selling it on the market for S_T. Therefore the holders profit or payoff is

$$max(S_T - K, 0).$$

On the other hand if $K > S_T$, the holder of the put will buy the asset and exercise the the right of selling the asset. The payoff in that case is

$$max(K - S_T, 0).$$

Asian Option

Unlike the case of European Options where the payoff depends only on the price of the underlying at the last day of holding the option, an Asian

Option is an option for which the payoff depends on the average of the price of the underlying asset.

Now there are different ways of taking the average, resulting in different kinds of payoff structures and hence different types of options. We can have an Asian option written on a stock with price S_t at time t t which can be exercised at time T with strike price K by taking the arithmetic average for the period $[t_0, T]$. In such a case we can define a fixed strike Asian call option payoff as

$$(\frac{1}{T - t_0} \int_{t_0}^{T} S_u du - K)^+.$$

The fixed strike put payoff is therefore

$$(K - \frac{1}{T - t_0} \int_{t_0}^{T} S_u du)^+.$$

Numerical Methods for Option Pricing

Numerical methods are mathematical tools designed to give approximate but accurate solutions to various numerical problems. Most often used methods are iterative methods that converge to a satisfactory result with a certain assumed level of approximation by iterating in a finite number of steps from the initial conditions. In this project, we deal with interpolation and solutions to continuous-time stochastic differential equations describing financial phenomena like interest rate. As we can only represent problems with finite amount of data, we need to discretize these problems by finding values in a finite number of points in a problem domain. We use numerical methods to solve problems that do not have an analytical solution, e.g.

some differential equations cannot be solved exactly. While using numerical methods, it is important to address the numerical stability of the used algorithm to estimate and control round-off errors arising from the use of floating point arithmetic.

Binomial Tree

A most basic numerical method for pricing options involves the construction of trees or lattices. A binomial tree (see (1.2)) is a numerical method, which allows to graphically represent the possible values that an option may take at different nodes or time periods. The value of the option depends on the underlying stock or bond, and computing values on the nodes is based on the probability that the price of the underlying asset will decrease or increase. The main advantage of binomial trees is that they are analytically tractable. This means that price valuation for a derivative can be performed on each node for every time step. This gives more flexibility and allows pricing of path-dependent derivatives, such as exotic options, having more complex cashflows.

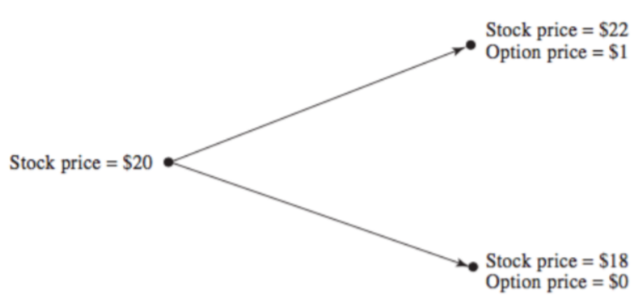

Figure 1.2: Illustration of a binomial tree

Trinomial Tree

The trinomial option pricing model (an example is shown on fig. (1.3)) is an alternative numerical method for constructing trees, with the difference that it consolidates another possible value per single time period. This makes the trinomial model even more relevant to real life situations, as it ensures the possibility that the value of the underlying asset may not change over a time period (taking the mid path on the tree). Calculations for a trinomial tree are analogous to those for a binomial tree. On the other hand, this constraint is less of concern in current computing environment with abundant and efficient compute capabilities.

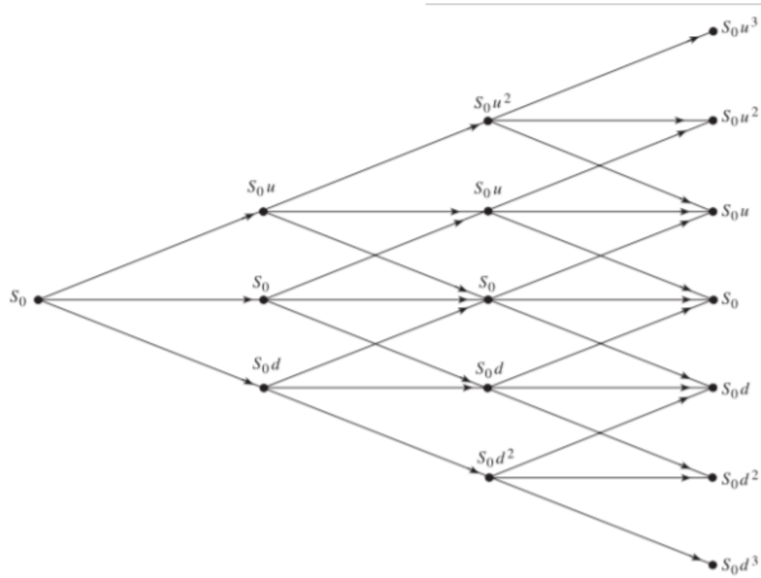

Figure 1.3: Illustration of a trinomial tree

Monte Carlo simulations

Monte Carlo method is an important numerical methods. It is mainly used to calculated the integral using random experiment method, i.e. transforming the integral to the expectation of a random variable $f(x)$, which has distribution $F(x)$.

Suppose we want to calcute

$$A = \int_0^1 f(x)dx.$$

It's easy to rewrite the equation into

$$A = E[f(X)]$$

where X follows uniform distribution. Then we can generate N uniform random number X_i and use the estimator

$$\hat{A} = \frac{1}{N}\sum_1^N f(X_i)$$

to calculate this integral. If $N \to \infty$, then by law of large numbers, $\hat{A}$ is close to A.

In financial mathematics, Monte Carlo method is often used to sample different paths to obtain the expected payoff of the asset in a risk-neutral world and are then discounted at this risk-free rate. Monte Carlo Methods are particularly useful in the valuation of options with multiple sources of uncertainty (multiple dimensions) They are suitable for pricing instruments with complicated features, when the payoff depends on the path followed by the underlying variables, as this makes them difficult to value through a straightforward Black Scholes analytical model or tree-based computation.

The advantage of MC method is that it tends to be simple and flexible. Since the algorithm is straightforward, coding becomes easy. What's more, it can be used to solve the high dimension problem while other numerical methods don't work. Finally, Monte Carlo algorithms are parallelizable, in particular when various parts can be run independently. This allows the parts to be run on different processors, therefore significantly reducing the computation time. However, MC method has also received some criticism.One common criticism is that he method is computationally intensive and might be too slow to be competitive over an analytical solution or other numerical techniques like trees. Another disadvantage is it may not incorporate the fat tailed nature of return distributions, as well as things like autocorrelation.

In this work, we use Monte Carlo method to verify the Asian Option prices calculated by the CLMR trees.

Finite Difference Method

Finite difference Mehthods value a derivative by solving the differential equation that the derivative satisfies. The differential equations are converted into sets of difference equations and are solved iteratively. This is similar to the trinomial trees method, since the computations also work back from the end of the derivative maturity to the beginning. In fact, tree based methods, if suitably parameterized, are a special case of the explicit finite difference method. These type of methods can solve derivative pricing problems that have, in general, the same level of complexity as those problems solved by tree approaches, but, given their relative complexity, are usually employed only when other approaches are inappropriate. Furthermore, like tree-based methods, they are limited in terms of the number of underlying variables, i.e. multiple dimensions, they can handle.

The remaining parts are arranged as follow. In Chapter 2, we present the CLMR trees for the European options under the regime-switching jump-diffusion models, prove the high-order equivalence with the perturbed FDMs, and prove the convergence rates of CLMR trees and perturbed FDMs. In addition, we make a remark on the relation between CLMR trees and the multinomial trees of J.X. Jiang and Nguyen (2016). In Chapter 3, the CLMR trees are developed to solve the Asian option pricing under the regime-switching jump-diffusion models. In Chapter 4, numerical results are given to verify the convergence rates of the CLMR trees and perturbed FDMs, make a comparison of the CLMR trees with the multinomial trees of J.X. Jiang and Nguyen (2016). Conclusions are given in the final Chapter.

Chapter 2

CLMR trees for European options

In this section, we will prove the equivalence between the CLMR trees of M. Costabile and Russo (2014) and the explicit perturbed FDMs for the option pricing under regime-switching jump-diffusion model (1.1), and the convergence rates of the perturbed FDMs and the CLMR trees.

Here we introduce the multinomial recombining tree of M. Costabile and Russo (2014) for the European option with the price of the underlying asset following the regime-switching jump-diffusion model (1.1). For ease of exposition, we only study the case of two regimes, i.e., $\mathbb{D} = \{0, 1\}$, where regime 0 is the high volatility state and regime 1 is the low volatility one. The algorithm is easily applicable to the case of more than two regimes.

Denote the length of the time steps as $\Delta t = T/n$, with T being maturity date, and $\beta_l = \varepsilon_l - \lambda_l m_l$ for all $l \in \mathbb{D}$. Firstly we start our process at node

$(0,0)$ and the asset has logarithm value $X(0,0)=0$. Then for every regime, the logarithm values of the asset price return at each node j at the i-th time are all the same, i.e., $X(i,j)=i\beta_0\Delta t+j\Delta y$, $\Delta y:=\bar{\sigma}\sqrt{\Delta t}$ and we choose $\bar{\sigma}=\sqrt{3/2}\sigma_0$ where σ_0 is the asset volatility in regime 0. The successor points at time $(i+1)\Delta t$ are $X(i+1,j+x)=(i+1)\beta_0\Delta t+(j+x)\Delta y$ with $x=-d,\ldots,u$, where $d=\max\{d_l: l\in\mathbb{D}\}$ and $u=\max\{u_l: l\in\mathbb{D}\}$, and d_l, u_l are the smallest integer numbers satisfying

$$\begin{cases} \mathcal{F}_l(\beta_0\Delta t+(-d_l+0.5)\Delta y)<\epsilon, \\ 1-\mathcal{F}_l(\beta_0\Delta t+(u_l-0.5)\Delta y)<\epsilon, \end{cases} \tag{2.1}$$

where $\mathcal{F}_l(x)=\int_{-\infty}^{x} f_l(z)dz$ is the cumulative distribution function of Z_i^l, ϵ is a tolerance level by which the number of nodes at each time step is bounded. Note that u and d are independent of $X(i,j)$.

Moreover, the probabilities corresponding to the prices' movement are

$$q_l(x)=\begin{cases} d\mathcal{F}_l(1)\lambda_l\Delta t+\pi_l^u(1-\lambda_l\Delta t), & \text{if } x=1, \\ d\mathcal{F}_l(0)\lambda_l\Delta t+\pi_l^m(1-\lambda_l\Delta t), & \text{if } x=0, \\ d\mathcal{F}_l(-1)\lambda_l\Delta t+\pi_l^d(1-\lambda_l\Delta t), & \text{if } x=-1, \\ d\mathcal{F}_l(x)\lambda_l\Delta t, & \text{otherwise,} \end{cases} \tag{2.2}$$

where

$$\begin{aligned} d\mathcal{F}_l(x) &:= P\{Z_i^l=x\Delta y\} \qquad (2.3)\\ &= \mathcal{F}_l(\beta_0\Delta t+(x+0.5)\Delta y)-\mathcal{F}_l(\beta_0\Delta t+(x-0.5)\Delta y), \quad x=0,\pm1,\pm2,\ldots,\pm M, \end{aligned}$$

where M is a sufficiently large positive integer such that the probability $d\mathcal{F}_l(x)=P\{Z_i^l=x\Delta y\}$ is extremely small.

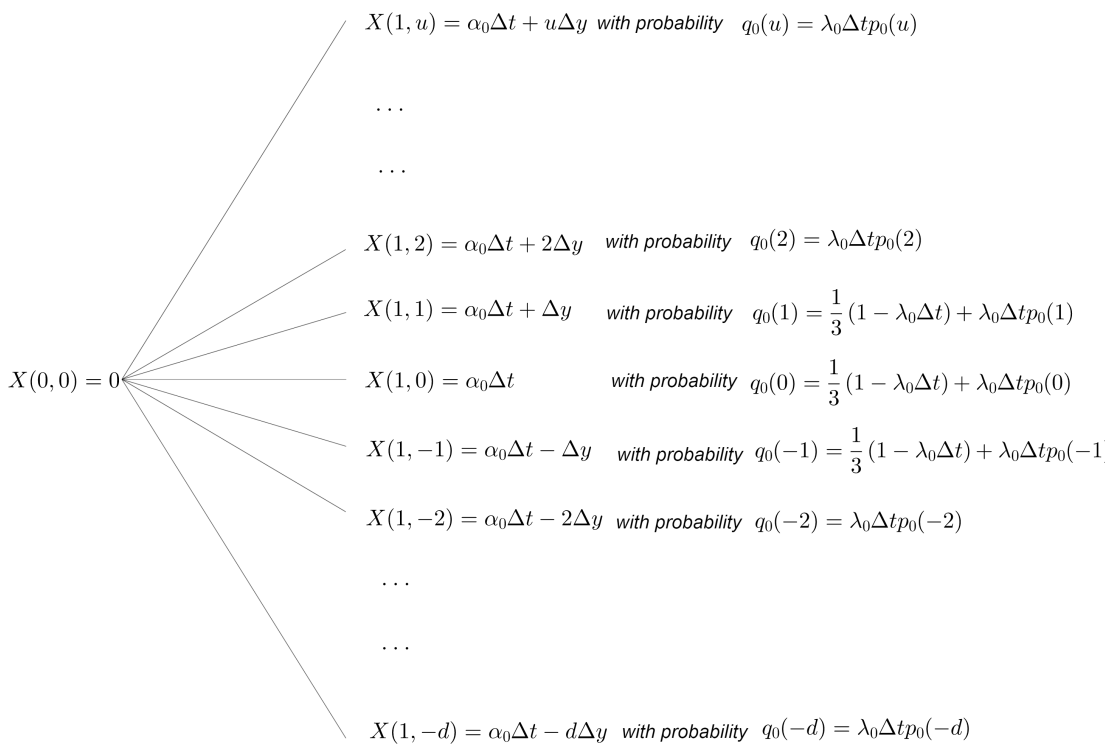

Figure 2.1: The approximation of the process (1.1) at the first time step in regime 0

Furthermore, for regime 0, we assign

$$\pi_0^u = \pi_0^m = \pi_0^d = 1/3, \tag{2.4}$$

which guarantees that the discrete approximating process has the same local mean and the same local variance of the diffusion component in the continuous process.

As for regime 1, the probabilities are

$$\pi_1^u = \frac{\sigma_1^2 \Delta t + (\beta_1 - \beta_0)^2 (\Delta t)^2 + (\beta_1 - \beta_0)\Delta t \Delta y}{2(\Delta y)^2}, \tag{2.5}$$

$$\pi_1^m = 1 - \frac{\sigma_1^2 \Delta t + (\beta_1 - \beta_0)^2 (\Delta t)^2}{(\Delta y)^2}, \tag{2.6}$$

$$\pi_1^d = \frac{\sigma_1^2 \Delta t + (\beta_1 - \beta_0)^2 (\Delta t)^2 - (\beta_1 - \beta_0)\Delta t \Delta y}{2(\Delta y)^2}. \tag{2.7}$$

The multinomial value of European options with maturity T for regime 0 and regime 1 can be recursively calculated by, for $i = 0, 1, \ldots, n-1$,

$$\begin{aligned} \widehat{V}_l(i,j) \quad = \quad & e^{-r_l \Delta t} \sum_{c=0}^{1} p_{lc} \Big[[\pi_c^u (1 - \lambda_c \Delta t) + \lambda_c \Delta t \, d\mathcal{F}_c(j+1)] \widehat{V}_c(i+1, j+1) \\ & + [\pi_c^m (1 - \lambda_c \Delta t) + \lambda_c \Delta t \, d\mathcal{F}_c(j)] \widehat{V}_c(i+1, j) \\ & + [\pi_c^d (1 - \lambda_c \Delta t) + \lambda_c \Delta t \, d\mathcal{F}_c(j-1)] \widehat{V}_c(i+1, j-1) \\ & + \sum_{k \neq 1, 0, -1} \lambda_c \Delta t \, d\mathcal{F}_c(k) \widehat{V}_c(i+1, j+k) \Big] \end{aligned} \tag{2.8}$$

with the European option's payoff

$$\widehat{V}_l(n,j) = \max(S_0 \exp(X(n,j)) - K, 0) \tag{2.9}$$

for $j = -nd, \ldots, 0, \ldots, nu$, in which p_{lc} is the transition probability from regime state l to state c for the time interval with length Δt. It is given by Taylor's expansion

$$(p_{lc})_{l,c \in \mathbb{D}} = e^{\mathrm{A} \Delta t} = \mathrm{I} + \sum_{c=1}^{\infty} (\Delta t)^c \mathrm{A}^c / c!, \tag{2.10}$$

where I is the identity matrix and A is the generator matrix of the Markov chain process. Thus, we have

$$p_{ll} = 1 + a_{ll} \Delta t + O\left((\Delta t)^2\right), \tag{2.11}$$

$$p_{lc} = a_{lc} \Delta t + O\left((\Delta t)^2\right), \quad l \neq c. \tag{2.12}$$

where $a_{lc} \in A$.

Remark 0.1. Jiang et al. in J.X. Jiang and Nguyen (2016) also establish a multinomial tree method (MTM) for pricing options under regime-switching jump-diffusion models. In this section, we provide a comparison between CLMR trees and the MTM of J.X. Jiang and Nguyen (2016):

$$\begin{aligned} V_l(i,j) &= e^{-r_l\Delta t}\sum_{c=1}^{m} p_{lc}\Big[\sum_{\tau=u,m,d}(\pi_l^{\tau}(1-\lambda_l\Delta t)+\lambda_l\Delta t d\mathcal{F}_l(k_\tau))V_l(i+1,j+k_\tau) \\ &\quad + \sum_{k\neq k_u,k_m,k_d}\lambda_l\Delta t d\mathcal{F}_k V_c(i+1,j+k)\Big]. \end{aligned} \tag{2.13}$$

Both the two trees are needed to capture the diffusion and the jump component. In CLMR trees the movements of the underlying asset price include both the local drift and the local volatility. However, in Liu's tree, we don't distinguish whether the change comes from a local change (due to the continuous component) or from a jump (due to the occurring of a jump).

The ways of dealing with the regime switching of the two trees are different. In Liu's tree, as a result of efficiently capturing the possible mean-reverting feature of the model, we should choose $-\infty \leq x^{l,1} \leq x^{l,2} \leq +\infty$ as the lower bound and upper bound for the state variable X when the regime is l. Depending on the value x taken by X_i, one of the three cases is chosen for X_{i+1}: $x^{l,1} \leq x \leq x^{l,2}$, $x \leq x^{l,1}$ and $x^{l,2} \leq x$. The probabilities π_l^u, π_l^m and π_l^d are different in these three cases. Hence the tree has at most $m(2Ki+1)$ nodes at the i-th time step, where $K := \max_{1\leq l\leq m} k_i$. CLMR trees approximate the diffusion part by a trinomial tree and add branches to capture jumps, too. However, CLMR trees are mainly based on the lattice for the highest volatility regime, and adjust the probability measure to describe the underlying assets in the other regime. For the other regimes, instead of generating new lattices, it simply adjusts branching probabilities as suggested by Yuen and Yang (2010). That means instead of changing the volatility if the regime state changes, it changes the risk-neutral probability, so the multinomial tree is still a combined one. That means when using CLMR trees to deal with the regime switching, we face less nodes than Jiang, Liu and Nguyen's trees. Hence the computational time of CLMR tree will be shorter than that of the Jiang, Liu and Nguyen's trees.

What's more, as for the number of nodes at each time step, CLMR trees require a tolerance level ϵ so that we can obtain the number of up and down movements. However, Jiang, Liu and Nguyen's tree only needs us directly to give the number of moves k_i, which may cause the artificial errors. Thus CLMR trees are more precise than the Jiang, Liu and Nguyen's trees in respect of accuracy.

Yuen and Yang (2009) also study the European option under the regime-switching jump-diffusion models. But the method investigated by Yuen and Yang (2009) is the trinomial tree not the multinomial tree.

Since we are supposed to prove the equivalence between CLMR trees and perturbed FDMs, we denote $x_j \equiv j\Delta y$. Then (2.8) can be re-written as

$$\begin{aligned}
\widehat{V}_l(i,x_j) &= e^{-r_l\Delta t}\sum_{c=0}^{1} p_{lc}\Big[[\pi_c^u(1-\lambda_c\Delta t)+\lambda_c\Delta t\, d\mathcal{F}_c(j+1)]\widehat{V}_c(i+1,x_{j+1}) \\
&\quad +[\pi_c^m(1-\lambda_c\Delta t)+\lambda_c\Delta t\, d\mathcal{F}_c(j)]\widehat{V}_c(i+1,x_j) \\
&\quad +[\pi_c^d(1-\lambda_c\Delta t)+\lambda_c\Delta t\, d\mathcal{F}_c(j-1)]\widehat{V}_c(i+1,x_{j-1}) \\
&\quad +\sum_{k\neq 1,0,-1}\lambda_c\Delta t\, d\mathcal{F}_c(k)\widehat{V}_c(i+1,x_{j+k})\Big] \\
&= e^{-r_l\Delta t}\sum_{c=0}^{1} p_{lc}\Big[(1-\lambda_c\Delta t) \\
&\quad \cdot\Big(\pi_c^u\widehat{V}_l(i+1,x_{j+1})+\pi_c^m\widehat{V}_l(i+1,x_j)+\pi_c^d\widehat{V}_l(i+1,x_{j+1})\Big) \\
&\quad +\sum_{k=-M}^{M}\lambda_c\Delta t\, d\mathcal{F}_c(k)\widehat{V}_l(i+1,x_k)\Big] \qquad (2.14)
\end{aligned}$$

with $\widehat{V}_l(n,x_j)=f(e^{x_j})$ (payoff function) for $j=-d,-d+1,\dots,u-1,u$; $i\in\mathbb{D}$ and M is a sufficiently large positive integer.

Following I. Florescu and Sewell (2013), the value of European option $\widehat{V}(x,t,l)$ with maturity date T, payoff $f(e^x)$ and regime state l satisfies the following partial integro-differential equations (PIDEs)

$$\begin{aligned}
&\frac{\partial\widehat{V}(x,t,l)}{\partial t}+\frac{\sigma_l^2}{2}\frac{\partial^2\widehat{V}(x,t,l)}{\partial x^2}+(b_l-\lambda_l\kappa_l)\frac{\partial\widehat{V}(x,t,l)}{\partial x}-(r_l+\lambda_l)\widehat{V}(x,t,l) \\
&\quad +\sum_{c=0}^{1}a_{lc}\widehat{V}(x,t,c)+\lambda_l\int_{-\infty}^{\infty}\widehat{V}(x+y,t,l)\,d\mathcal{F}_l(y)=0, \qquad (2.15)
\end{aligned}$$

with terminal condition $\widehat{V}(x,T,i)=f(e^x),\ i\in\mathbb{D}$.

Denote $\Delta y = x_{j+1}-x_j$ and $\widehat{V}_j^i(l)\approx\widehat{V}(x_j,t_i,l)$. Then the explicit finite difference method for solving the PIDEs (2.15) is given by, for $i=$

$0, 1, \ldots, n-1$,

$$\frac{\widehat{V}_j^{i+1}(l) - \widehat{V}_j^i(l)}{\Delta t} + \frac{\sigma_l^2}{2} \frac{\widehat{V}_{j+1}^{i+1}(l) - 2\widehat{V}_j^{i+1}(l) + \widehat{V}_{j-1}^{i+1}(l)}{(\Delta y)^2} \tag{2.16}$$
$$+ \left(\frac{((\varepsilon_l - \lambda_l m_l) - (\varepsilon_0 - \lambda_0 m_0))^2 \Delta t}{2\bar{\sigma}^2} \right) \frac{\widehat{V}_{j+1}^{i+1}(l) - 2\widehat{V}_j^{i+1}(l) + \widehat{V}_{j-1}^{i+1}(l)}{(\Delta y)^2}$$
$$+ ((\varepsilon_l - \lambda_l m_l) - (\varepsilon_0 - \lambda_0 m_0)) \frac{\widehat{V}_{j+1}^{i+1}(l) - \widehat{V}_{j-1}^{i+1}(l)}{2\Delta y} - (r_l + \lambda_l)\widehat{V}_j^i(l)$$
$$+ \sum_{c=0}^{1} a_{lc} \widehat{V}_j^{i+1}(c) + \lambda_l \sum_{k=-M}^{M} V_{j+k}^{i+1}(l) \, d\mathcal{F}_l(k) = 0,$$

with $\widehat{V}_j^n(l) = f(e^{x_j})$ for $j = -d, -d+1, \ldots, u-1, u$; $l \in \mathbb{D}$. Since (2.16) contains a perturbation term, it's not a standard FDM. In this paper, we call it perturbed FDM. And in J. Ma and Zhu (2018) it has been shown that the convergence rate of perturbed FDM is the same as the standard FDM in the following. Note that we have used the composite mid-point quadrature rules to discretize the integrals in PIDEs (2.15) based on the mesh nodes $y = x_k$, $k = 0, \pm 1, \pm 2, \ldots, \pm M$.

Theorem 0.1. For the European option pricing with regime-switching jump-diffusion model (1.1), the explicit perturbed FDM (2.16) is equivalent to the CLMR tree by neglecting high-order term $O((\Delta t)^2)$ under condition

$$((\beta_1 - \beta_0)\Delta t \Delta y \leq \sigma_1^2 \Delta t + (\beta_1 - \beta_0)^2 (\Delta t)^2 \leq (\Delta y)^2. \tag{2.17}$$

Proof. With the set-up $\Delta y = \bar{\sigma}\sqrt{\Delta t}$, we rewrite the perturbed FDM (2.16) into the following form

$$\frac{\widehat{V}_j^{i+1}(l) - \widehat{V}_j^i(l)}{\Delta t} + \frac{\sigma_l^2}{2}\frac{\widehat{V}_{j+1}^{i+1}(l) - 2\widehat{V}_j^{i+1}(l) + \widehat{V}_{j-1}^{i+1}(l)}{(\bar{\sigma}\sqrt{\Delta t})^2} \tag{2.18}$$

$$+\left(\frac{((\varepsilon_l - \lambda_l m_l) - (\varepsilon_0 - \lambda_0 m_0))^2 \Delta t}{2\bar{\sigma}^2}\right)\frac{\widehat{V}_{j+1}^{i+1}(l) - 2\widehat{V}_j^{i+1}(l) + \widehat{V}_{j-1}^{i+1}(l)}{(\bar{\sigma}\sqrt{\Delta t})^2}$$

$$+((\varepsilon_l - \lambda_l m_l) - (\varepsilon_0 - \lambda_0 m_0))\frac{\widehat{V}_{j+1}^{i+1}(l) - \widehat{V}_{j-1}^{i+1}(l)}{2\bar{\sigma}\sqrt{\Delta t}} - (r_l + \lambda_l)\widehat{V}_j^i(l)$$

$$+\sum_{c=0}^{1} a_{lc}\widehat{V}_j^{i+1}(c) + \lambda_l \sum_{k=-M}^{M} V_{j+k}^{i+1}(l)\, d\mathcal{F}_l(k) = 0,$$

Then we have the recursively form:

$$\begin{aligned}
\widehat{V}_j^i(l) &= \frac{1}{1+(r_l+\lambda_l)\Delta t}\Bigg\{\Bigg[\frac{\sigma_l^2}{2(\bar{\sigma})^2} + \frac{\sqrt{\Delta t}}{2\bar{\sigma}}\Big((\varepsilon_l - \lambda_l m_l) - (\varepsilon_0 - \lambda_0 m_0)\Big) \\
&+ \frac{((\varepsilon_l - \lambda_l m_l) - (\varepsilon_0 - \lambda_0 m_0))^2 \Delta t}{2\bar{\sigma}^2}\Bigg]\widehat{V}_{j+1}^{i+1}(l) \\
&+ \Bigg[1 - \frac{\sigma_l^2}{(\bar{\sigma})^2} - \frac{((\varepsilon_l - \lambda_l m_l) - (\varepsilon_0 - \lambda_0 m_0))^2 \Delta t}{(\bar{\sigma})^2}\Bigg]\widehat{V}_j^{k+1}(l) \\
&+ \Bigg[\frac{\sigma_l^2}{2(\bar{\sigma})^2} - \frac{\sqrt{\Delta t}}{2\bar{\sigma}}\Big((\varepsilon_l - \lambda_l m_l) - (\varepsilon_0 - \lambda_0 m_0)\Big) \\
&\quad + \frac{((\varepsilon_l - \lambda_l m_l) - (\varepsilon_0 - \lambda_0 m_0))^2 \Delta t}{2(\bar{\sigma})^2}\Bigg]\widehat{V}_{j-1}^{i+1}(l) \\
&+ \Delta t \sum_{c=0}^{1} a_{lc}\widehat{V}_j^{i+1}(c) + \lambda_l \Delta t \sum_{k=-M}^{M} \widehat{V}_{j+k}^{i+1}(l)\, d\mathcal{F}_l(k)\Bigg\}, \quad l \in \mathbb{D},
\end{aligned} \tag{2.19}$$

with $\widehat{V}_j^n(i) = f(e^{x_j})$ for $j = -d, -d+1, \ldots, u-1, u$; $i \in \mathbb{D}$.

Using (2.11), (2.12) and the relation $\sum_{c=0}^{1} a_{lc} = 0$ for $l \in \mathbb{D}$, the CLMR trees (2.14) is rewritten as

$$\begin{aligned}\widehat{V}_l(i,x_j) &= e^{-r_l\Delta t}(1-\lambda_l\Delta t)\Bigg[\pi_l^u\widehat{V}_l(i+1,x_{j+1})+\pi_l^m\widehat{V}_l(i+1,x_j)+\pi_l^d\widehat{V}_l(i+1,x_{j-1})\\ &\quad +\Delta t\sum_{c=0}^{1}a_{lc}\left[\pi_l^u\widehat{V}_c(i+1,x_{j+1})+\pi_l^m\widehat{V}_c(i+1,x_j)+\pi_l^d\widehat{V}_c(i+1,x_{j-1})\right]\Bigg]\\ &\quad +e^{-r_l\Delta t}\lambda_l\Delta t\sum_{k=-M}^{M}\widehat{V}_l(i+1,x_{j+k})\,d\mathcal{F}_l(k)+O((\Delta t)^2). \qquad (2.20)\end{aligned}$$

The Taylor's expansion gives that

$$e^{-r_l\Delta t}=1-r_l\Delta t+O((\Delta t)^2).$$

Therefore we have

$$\begin{aligned}e^{-r_l\Delta t}(1-\lambda_l\Delta t)\left[1+(r_l+\lambda_l)\Delta t\right] &= 1-r_l^2(\Delta t)^2-\lambda_l(r_l+\lambda_l)(1-r_l\Delta t)(\Delta t)^2+O((\Delta t)^2)\\ &= 1+O((\Delta t)^2).\end{aligned}$$

So

$$e^{-r_l\Delta t}(1-\lambda_l\Delta t)=\frac{1}{1+(r_l+\lambda_l)\Delta t}+O((\Delta t)^2). \qquad (2.21)$$

For regime 0,on one hand, we insert (2.4) and (2.21) into (2.20), we have

$$\begin{aligned}\widehat{V}_0(i,x_j) &= \frac{1}{1+(r_0+\lambda_0)\Delta t}\Bigg\{\frac{1}{3}\widehat{V}_0(i+1,x_{j+1})+\frac{1}{3}\widehat{V}_0(i+1,x_j)+\frac{1}{3}\widehat{V}_0(i+1,x_{j-1})\\ &+ \Delta t\sum_{c=0}^{1}a_{0c}\widehat{V}_c(i+1,x_j)+\lambda_0\Delta t\sum_{k=-M}^{M}\widehat{V}_0(i+1,x_{j+k})\,d\mathcal{F}_0(k)\Bigg\}\\ &+ O((\Delta t)^2). \qquad (2.22)\end{aligned}$$

On the other hand, we can rewrite (2.19) with $\bar{\sigma} = \sqrt{3/2}\sigma_0$:

$$\begin{aligned}\widehat{V}_j^i(0) &= \frac{1}{1+(r_0+\lambda_0)\Delta t}\Bigg\{\frac{1}{3}\widehat{V}_{j+1}^{i+1}(0)+\frac{1}{3}\widehat{V}_j^{i+1}(0)+\frac{1}{3}\widehat{V}_{j-1}^{i+1}(0)\\ &+ \Delta t\sum_{c=0}^{1} a_{0c}\widehat{V}_j^{i+1}(c)+\lambda_0\Delta t\sum_{k=-M}^{M}\widehat{V}_{j+k}^{i+1}(0)\,d\mathcal{F}_0(k)\Bigg\}.\end{aligned}$$

As for regime 1, we update (2.5), (2.6) and (2.7) as

$$\pi_1^u = \frac{\sigma_1^2\Delta t + ((\varepsilon_1-\lambda_1 m_1)-(\varepsilon_0-\lambda_0 m_0))^2(\Delta t)^2 + ((\varepsilon_1-\lambda_1 m_1)-(\varepsilon_0-\lambda_0 m_0))\Delta t\Delta y}{2(\Delta y)^2},$$

$$\pi_1^m = 1-\frac{\sigma_1^2\Delta t + ((\varepsilon_1-\lambda_1 m_1)-(\varepsilon_0-\lambda_0 m_0))^2(\Delta t)^2}{(\Delta y)^2},$$

$$\pi_1^d = \frac{\sigma_1^2\Delta t + ((\varepsilon_1-\lambda_1 m_1)-(\varepsilon_0-\lambda_0 m_0))^2(\Delta t)^2 - ((\varepsilon_1-\lambda_1 m_1)-(\varepsilon_0-\lambda_0 m_0))\Delta t\Delta y}{2(\Delta y)^2}.$$

What's more, substituting $\Delta y = \bar{\sigma}\Delta t$ and the above formulas into (2.20), we get

$$\begin{aligned}\widehat{V}_1(i,x_j) &= \frac{1}{1+(r_1+\lambda_1)\Delta t}\Bigg\{\Big[\frac{\sigma_1^2}{2(\bar{\sigma})^2}+\frac{\sqrt{\Delta t}}{2\bar{\sigma}}\Big((\varepsilon_1-\lambda_1 m_1)-(\varepsilon_0-\lambda_0 m_0)\Big)\\ &+ \frac{((\varepsilon_1-\lambda_1 m_1)-(\varepsilon_0-\lambda_0 m_0))^2\Delta t}{2\bar{\sigma}^2}\Big]\widehat{V}_1(i+1,x_{j+1})\\ &+ \Big[1-\frac{\sigma_1^2}{\bar{\sigma}^2}-\frac{((\varepsilon_1-\lambda_1 m_1)-(\varepsilon_0-\lambda_0 m_0))^2\Delta t}{\bar{\sigma}^2}\Big]\widehat{V}_1(i+1,j)\\ &+ \Big[\frac{\sigma_1^2}{2\bar{\sigma}^2}-\frac{\sqrt{\Delta t}}{2\bar{\sigma}}\Big((\varepsilon_1-\lambda_1 m_1)-(\varepsilon_0-\lambda_0 m_0)\Big)\\ &\quad+\frac{((\varepsilon_1-\lambda_1 m_1)-(\varepsilon_0-\lambda_0 m_0))^2\Delta t}{2\bar{\sigma}^2}\Big]\widehat{V}_1(i+1,x_{j-1}) \qquad (2.23)\\ &+ \Delta t\sum_{c=0}^{1} a_{1c}\widehat{V}_c(i+1,x_j)+\lambda_1\Delta t\sum_{k=-M}^{M}\widehat{V}_1(i+1,x_{j+k})\,d\mathcal{F}_1(k)\Bigg\}+O((\Delta t)^2),\end{aligned}$$

with $\widehat{V}_l(n,x_j) = f(e^{x_j})$ $j=-d,-d+1,\ldots,u-1,u$; To ensure probability (2.5)-(2.7) are nonnegative, it requires that (2.17) holds true. Comparing (2.23) with (2.19) completes the proof. □

Theorem 0.2. Set $\Delta y = \bar{\sigma}\sqrt{\Delta t}$. Then under condition (2.17), the CMLR tree is stable.

Proof. From the theory for FDM (see Jiang (2005)), we know that both

$$\frac{\bar{\sigma}^2 \Delta t}{\Delta y^2} \leq 1, \tag{2.24}$$

and condition (2.17) are required to guarantee the stability of the FDM (2.19) and thus the stability of the CLMR trees as they are equivalent (see Theorem 0.1). In fact condition (2.24) is always satisfied as we have already set $\Delta y = \bar{\sigma}\sqrt{\Delta t}$. □

Now we present the convergence rates of perturbed FDMs and CLMR trees for pricing the European option under state-dependent switching rate model (1.1).

Theorem 0.3. Assume the payoff function f is continuous. Let

$$\gamma_j^i(l) := \widehat{V}(x_j, t_i, l) - \widehat{V}_j^i(l), \quad \gamma^i(l) := \left(\gamma_{-d}^i(l), \ldots, \gamma_u^i(l)\right)'$$
$$\eta_j^i(l) := \widehat{V}(x_j, t_i, l) - \widehat{V}_l(i, x_j), \quad \eta^i(l) := \left(\eta_{-d}^i(l), \ldots, \eta_u^i(l)\right)'.$$

Then under the set-up $\Delta y = \bar{\sigma}\sqrt{\Delta t}$ and (2.17), the convergence rates of the perturbed FDM (2.18) at time t_i for pricing European option under regime-switching jump-diffusion model (1.1) are estimated by

$$\|\gamma^i(l)\|_\infty := \max_j |\gamma_j^i(l)| = |O(\Delta t)|, \quad i = 0, 1, \ldots, n-1;\ l \in \mathbb{D}, \tag{2.25}$$

and the convergence rates of the CLMR are given by

$$\|\eta^i(l)\|_\infty := \max_j |\eta_j^i(l)| = |O(\Delta t)|, \quad i = 0, 1, \ldots, n-1;\ l \in \mathbb{D}. \tag{2.26}$$

Proof. The proof of the perturbed FDM's convergence rates (2.25) is similar to the Theorem 3.3 in Ma and Zhu (2015).

Now we prove (2.26). To this end, we re-write the error of the CLMR into the following form

$$\begin{aligned} \eta_j^i(l) &= [\widehat{V}(x_j, t_i, l) - \widehat{V}_j^i(l)] + [\widehat{V}_j^i(l) - \widehat{V}_l(i, x_j)] \\ &= \gamma_j^i(l) + \chi_j^i(l), \end{aligned} \tag{2.27}$$

where $\chi_j^i(l) := \widehat{V}_j^i(l) - \widehat{V}_l(i, x_j)$. Easily, we derive that

$$\begin{aligned} \chi_j^i(l) = \quad & \frac{1}{1+(r_l+\lambda_l)\Delta t}\Bigg\{\Big[\frac{\sigma_l^2}{2\bar{\sigma}^2} + \frac{\sqrt{\Delta t}}{2\bar{\sigma}}\Big(\varepsilon_l - \lambda_l m_l\Big) + \frac{(\varepsilon_l - \lambda_l m_l)^2\Delta t}{2\bar{\sigma}^2}\Big]\chi_{j+1}^{i+1}(l) \\ & + \Big[1 - \frac{\sigma_l^2}{\bar{\sigma}^2} - \frac{(\varepsilon_l - \lambda_l m_l)^2\Delta t}{\bar{\sigma}^2}\Big]\chi_j^{i+1}(l) \\ & + \Big[\frac{\sigma_l^2}{2\bar{\sigma}^2} - \frac{\sqrt{\Delta t}}{2\bar{\sigma}}\Big(\varepsilon_l - \lambda_l m_l\Big) + \frac{(\varepsilon_l - \lambda_l m_l)^2\Delta t}{2\bar{\sigma}^2}\Big]\chi_{j-1}^{i+1}(l) \\ & + \Delta t\sum_{c=0}^{1} a_{lc}\chi_j^{i+1}(c) + \lambda_l\Delta t\sum_{k=-M}^{M}\chi_{j+k}^{i+1}(l)\, d\mathcal{F}_l(k)\Bigg\} + |O((\Delta t)^2)|, \quad l \in \mathbb{D}. \end{aligned} \tag{2.28}$$

Let $\chi^i(l) := \big(\chi_{-d}^i(l), \ldots, \chi_u^i(l)\big)'$. Since $a_{lc} > 0$ for $l \neq c$ and $a_{ll} < 0$, we have

$$\begin{aligned} \|\chi^i(l)\|_\infty \;\leq\; & \frac{1}{1+(r_l+\lambda_l)\Delta t}\Big[(1 - a_{ll}\Delta t + \lambda_l\Delta t)\|\chi^{i+1}(l)\|_\infty + \sum_{c=0, c\neq l}^{1} a_{lc}\Delta t\|\chi^{i+1}(c)\|_\infty\Big] \\ & + |O((\Delta t)^2)|, \quad l \in \mathbb{D}. \end{aligned} \tag{2.29}$$

Denote vector $\mathbf{\Psi}_i = (\|\chi^i(0)\|_\infty, \|\chi^i(1)\|_\infty)'$. Then (2.29) can be re-written into a vector form

$$\mathbf{\Psi}_i \leq \mathbf{D}\mathbf{\Psi}_{i+1} + |O((\Delta t)^2)|\mathbf{1}, \tag{2.30}$$

where $\mathbf{1} = (1, 1)'$ is a 2-dimension vector and $\mathbf{D}$ is a 2×2 matrix

$$\mathbf{D} = \begin{bmatrix} 1 - (a_{00} - \lambda_0)\Delta t & a_{01}\Delta t \\ a_{10}\Delta t & 1 - (a_{11} - \lambda_1)\Delta t \end{bmatrix}.$$

Since each element of matrix $\mathbf{D}$ is nonnegative, iterating of inequality (2.30) gives that

$$\mathbf{\Psi}_i \leq \mathbf{D}^{n-i}\mathbf{\Psi}_n + \left[\mathbf{I} + \sum_{m=1}^{n-i-1} \mathbf{D}^m\right] |O((\Delta t)^2)|\mathbf{1}, \tag{2.31}$$

where $\mathbf{I}$ is a 2×2 identity matrix and

$$\begin{aligned} &\mathbf{I} + \sum_{m=1}^{n-i-1} \mathbf{D}^m \\ &= \begin{bmatrix} (n-i) - \frac{(n-i)(n-i-1)}{2}(a_{00} - \lambda_0)\Delta t & \frac{(n-i)(n-i-1)}{2}a_{01}\Delta t \\ \frac{(n-i)(n-i-1)}{2}a_{10}\Delta t & (n-i) - \frac{(n-i)(n-i-1)}{2}(a_{11} - \lambda_1)\Delta t \end{bmatrix} \\ &\quad + \begin{bmatrix} |O((\Delta t)^2)| & |O((\Delta t)^2)| \\ |O((\Delta t)^2)| & |O((\Delta t)^2)| \end{bmatrix}. \end{aligned}$$

Since at the terminal time $t_n \equiv T$ the CLMR value equals the true option value, we know that $\mathbf{\Psi}_n$ is a zero vector. Therefore from (2.31) using $\Delta t = T/n$, we obtain that

$$\|\chi^i(l)\|_\infty = |O(\Delta t)|, \quad i = 0, 1, \ldots, n-1; \ l \in \mathbb{D}. \tag{2.32}$$

Therefore, from (2.27), using triangle inequality and estimations (2.25) and (2.32), we obtain that

$$\|\eta^i(l)\|_\infty \leq \|\gamma^i(l)\|_\infty + \|\chi^i(l)\|_\infty = |O(\Delta t)|.$$

Thus the proof of this theorem is complete. $\square$

Chapter 3

CLMR trees for Asian options

In this section, we apply the CLMR trees to price Asian option when the price of underlying asset follows a very general state-dependent regime-switching jump-diffusion model (1.1) for which the tree methods have never been studied in the literature. As we want to price the Asian option, the average stock price plays a role that makes the pricing of the Asian option more complex. The average security price depends on the path and can take many different values that cannot be reflected by the tree nodes directly. This causes the pricing of this strong path-dependent option to be a difficult problem. When pricing the Asian options, we are going to use the idea of Hull and White (1993) involving representative sets of values. The price of the Asian option with the average stock price that equals to representative sets of values is calculated. When pricing level is not the representative sets of values, the linear approximation is used to obtain the option price.

We establish the multinomial recombing tree based on N time steps of length $\Delta t = T/N$ where T is the maturity date. And we study the case of $\mathcal{M} + 1$ regimes i.e., $\mathbb{D} = \{0, 1, \ldots, \mathcal{M}\}$. And just like in section 2, we consider the case of 2 regimes i.e., $\mathbb{D} = \{0, 1\}$. The algorithm is easily applicable to the case of more than two regimes. At time step t_k, there are $(u + d)k + 1$ nodes in the lattice, where u and d are numbers of up-move and down-move which can be calculated by (2.1), the nodes are counted from the lowest stock price level. $S_{k,n}$ denotes the stock price of the n-th node at time step t_k and in our tree. We let $S_0 = S_{0,0}$ be the initial stock price. Denote jump extent $a = \exp(\bar{\sigma}\sqrt{\Delta t})$ and $b = a^{-1}$. From CLMR tree, we can write the stock price at k time step in the form of $S_{k,i} = S_0 \mathrm{e}^{k\beta_0 \Delta t} a^i$, $i = -dk, -dk + 1, \ldots, uk$. Then we can calculate the representative sets of values by the following method.

Instead of calculating the nodal value directly, we use a recursive method in Yuen and Yang (2012) to find the highest average and lowest average of the stock prices on each node of the CLMR trees. We consider an arbitrage node (k, n), which represents the n-th node counted from the lowest stock price at time step t_k. At this node, one principle to find the highest average stock price is that on condition that reaching the aimed node, the stock price should increase with the largest extent a^u as early as possible and as much as possible. For example, suppose $u = d = 2$, and if we want to find the highest average stock price at node $(3, 12)$, we should go up with the extent a^2 at time step t_1 and t_2. Then go up with the extent a at time step t_3. And similarly, when the average stock price achieves its lowest average stock price, the stock price should decrease with the largest extent b^d as early as possible and as much as possible.

We all know if n equals 0 or $(u+d)k$, that means the stock will increase with extent a^u or decrease with extent b^d all the time, which implies the highest average stock price will be the same as the lowest average stock at this node. Let $AS_{high}(k,n)$ and $AS_{low}(k,n)$ be the highest and lowest average stock price at node (k,n). And we assign the weight of the first node and the last node a half of others. From Yuen and Yang (2012) we can know that, compared with simple average, using this kind of adjusted average can improve the consistency of approximating. Denote $w = \mathrm{e}^{\beta_0 \Delta t}$. Then since if n equals 0 or $(u+d)k$, in all time the maximum average and minimum average stock price level will be the same, i.e.,

$$AS_{high}(k,0) = AS_{low}(k,0) = \frac{S_0}{k+1}\frac{1-(wb^d)^{k+1}}{1-wb^d}, \tag{3.1}$$

$$AS_{high}(k,(u+d)k) = AS_{low}(k,(u+d)k) = \frac{S_0}{k+1}\frac{(wa^u)^{k+1}-1}{wa^u-1} \tag{3.2}$$

If n does not equal 0 and $(u+d)k$, the path of the stock price resulting in the lowest average comes from $(k-1, n-u-d)$ just before reaching (k,n), and the highest average comes from $(k-1,n)$. Then we can recursively calculate the highest average and lowest average of an arbitrage node. However, there still is a problem of this method: at node (k,n), if $n < u+d$ or $n > (u+d)(k-1)$, then at time step t_{k-1}, there does not exist nodes $(k-1, n-u-d)$ and $(k-1,n)$. For example, if $n=1$, obviously $1-u-d<0$ and there does not exist node $(k-1, 1-u-d)$ in our CLMR trees; and if $n=(u+d)k-1$, similarly when time step is t_{k-1}, there does not exist node $(k-1,(u+d)k-1)$ in our trees, either. Thus we should deal with this phenomenon specially, and we have:

$$AS_{low}(k,p) = AS_{low}(k,0) + \frac{S_0}{k+1} b^{dk} w^k (a^p - 1), \tag{3.3}$$

$$AS_{high}(k,p) = b^{-p} AS_{low}(k,p) + \frac{S_0}{k+1}\big((1-b^{-p})(w^k b^{dk-p} + 1)\big), \tag{3.4}$$

$$\begin{aligned} AS_{high}(k,(u+d)k-q) &= AS_{high}(k,(u+d)k) \\ &\quad + \frac{S_0}{k+1} a^{uk} w^k (b^q - 1), \end{aligned} \tag{3.5}$$

$$\begin{aligned} AS_{low}(k,(u+d)k-q) &= a^{-q} AS_{high}(k,(u+d)k-q) \\ &\quad + \frac{S_0}{k+1}\big((1-a^{-q})(w^k a^{uk-q} + 1)\big), \end{aligned} \tag{3.6}$$

where $0 < p < u + d$ and $0 < q < u + d$.

After solving the above problem, we can have the recursive formula for the arbitrary node (k, n) when $u + d \le n \le (u + d)(k - 1)$:

$$AS_{low}(k,n) = \frac{1}{k+1}\Big[kAS_{low}(k-1,n-u-d) + S_0 w^{k-1}(a^{n-u-dk} + wa^{n-dk})\Big], \tag{3.7}$$

$$AS_{high}(k,n) = \frac{1}{k+1}\Big[kAS_{high}(k-1,n) + S_0 w^{k-1}(a^{n-d(k-1)} + wa^{n-dk})\Big]. \tag{3.8}$$

Then we can calculate the number of the representative values of each node based on the idea of representative value in Hull and White (1993). The representative levels are taken as the form $S_0 e^{mh}$, where m is an integer. We first find the largest and smallest representative values at each node,

$$M_{\min}(k,n) = \lfloor \ln(AS_{\text{low}}(k,n)/S_0)/h \rfloor, \tag{3.9}$$

$$M_{\max}(k,n) = \lceil \ln(AS_{\text{high}}(k,n)/S_0)/h \rceil. \tag{3.10}$$

We can see that $M_{\min}(k,n)$ and $M_{\max}(k,n)$ are the minimum and the maximum value of m, and $\lfloor\cdot\rfloor$ and $\lceil\cdot\rceil$ are the floor and the ceiling integer functions, respectively. Since we know $M_{\min}(k,n)$ and $M_{\max}(k,n)$, we know the range of m at each node (k,n). Then we have the average stock price at each node which equals to all the representative values $S_0 e^{mh}$.

In our recombing trees, instead of changing the volatility if the regime state changes, we change the risk-neutral probability, so the multinomial tree is still a combined one and all the regimes share the same lattice. We assume there are 2 regimes, then at each node there are 2 different derivative values. Asian option value depends on the path of the stock price and we use representative value to summarize it. Let $V^k_{n,m,i}$ be the value of Asian options at the n-th node of time step t_k with the representative level m when the regime is i, and we let $V^k(n,s,i)$ be the value of the Asian options with average stock price s. Then by definition, $V^k_{n,m,i} = V^k(n, S_0 e^{mh}, i)$.

Therefore, on the delivery day, the call option price with strike price K is

$$V^N_{n,m,i} = (S_0 e^{mh} - K)^+. \tag{3.11}$$

The transition probability p_{ij} is given in Section 2.

Since the real average price might not be at the representative level, we can use a linear approximation:

Let $\bar{A}^k_{n,m,i}$ be the average stock price value of the n-th node in regime i of time t_k, and $m = \lfloor \ln(\bar{A}^k_{n,m,i})/h \rfloor$ which represents the representative level.

Then we have the interpolation option value

$$\begin{aligned}\overline{V}^k(n, \bar{A}^k_{n,m,i}, i) &= \frac{\bar{A}^k_{n,m,i} - S_0 e^{mh}}{S_0 e^{mh}(e^h - 1)} V^k(n, S_0 e^{(m+1)h}, i) \\ &\quad + \frac{S_0 e^{(m+1)h} - \bar{A}^k_{n,m,i}}{S_0 e^{mh}(e^h - 1)} V^k(n, S_0 e^{mh}, i) \\ &= \frac{\bar{A}^k_{n,m,i} - S_0 e^{mh}}{S_0 e^{mh}(e^h - 1)} V^k_{n,m+1,i} + \frac{S_0 e^{(m+1)h} - \bar{A}^k_{n,m,i}}{S_0 e^{mh}(e^h - 1)} V^k_{n,m,i}.\end{aligned} \tag{3.12}$$

With the derivative price at delivery day we can use the following CLMR trees formulas to calculate the Asian option price recursively:

$$\begin{aligned}V^k_{n,m,i} &= V^k(n, S_0 e^{mh}, i) \\ &= e^{-r_i \Delta t} \sum_{j=0}^{\mathcal{M}} p_{ij} \Big[\big[\pi^u_j (1 - \lambda_j \Delta t) + \lambda_j \Delta t \, d\mathcal{F}_j(1) \big] \\ &\quad \cdot \overline{V}^{k+1}\Big(n + d + 1, \frac{k S_0 e^{mh} + S_{k,n} w a}{k+1}, j\Big) \\ &\quad + \big[\pi^m_j (1 - \lambda_j \Delta t) + \lambda_j \Delta t \, d\mathcal{F}_j(0) \big] \overline{V}^{k+1}\Big(n + d, \frac{k S_0 e^{mh} + S_{k,n} w}{k+1}, j\Big) \\ &\quad + \big[\pi^d_c (1 - \lambda_j \Delta t) + \lambda_j \Delta t \, d\mathcal{F}_j(-1) \big] \overline{V}^{k+1}\Big(n + d - 1, \frac{k S_0 e^{mh} + S_{k,n} w b}{k+1}, j\Big) \\ &\quad + \sum_{\gamma \neq 1,0,-1} \lambda_j \Delta t \, d\mathcal{F}_j(\gamma) \overline{V}^{k+1}\Big(n + d + \gamma, \frac{k S_0 e^{mh} + S_{k,n} w a^\gamma}{k+1}, j\Big) \Big].\end{aligned} \tag{3.13}$$

That is the application of CMLR trees to the Asian (fixed-strike) option pricing.

Remark 0.1. Yuen and Yang (2012) construct a trinomial tree to price the Asian option, and here are the comparison between the Yuen-Yang's tree and the CLMR tree. Both the trees are based on the idea of representative value raised by Hull and White (1993). In order to find the representative value, CLMR trees use the recursive method of Yuen and Yang (2012).

However, the Yuen-Yang's tree aims at pricing Asian option under the regime-switching model without jump diffusion while the CLMR tree approximates the diffusion part by a trinomial tree and adds branches to capture the jumps. Thus CLMR tree can price the option under regime-switching jump-diffusion model. Therefore, the numbers of the nodes at every time step t_k are different: in the CLMR tree, there are $(u + d)k + 1$ nodes at the k-th time step while in the Yuen-Yang's tree, there are $2k + 1$ nodes at the k-th time step. Accordingly, when calculating the highest average and lowest average of the stock prices on an arbitrage node of the CLMR tree, the CLMR tree should deal with more special nodes than the Yuen-Yang's tree.

Theorem 0.1. Under condition (2.17), the convergence rate of Asian option value computed by the CLMR tree model through (3.13) is $|O(\Delta t)|$.

Proof. Let $V^k(n, \bar{A}^k_{n,m,i}, i)$ be the be the value of the Asian options with average stock price $\bar{A}^k_{n,m,i}$, let $\overline{m} = \ln(\bar{A}^k_{n,m,i})/h$, and we can denote $V^k_{n,\overline{m},i} \equiv V^k(n, \bar{A}^k_{n,m,i}, i)$. First we define the local interpolation error:

$$\xi^k_{n,\overline{m},i} := |V^k(n, \bar{A}^k_{n,m,i}, i) - \overline{V}^k(n, \bar{A}^k_{n,m,i}, i)|, \quad k = 0, 1, \ldots, N-1;\ i \in \mathbb{D}, \tag{3.14}$$

The Taylors expansion gives that

$$V^k_{n,m+1,i} = V^k_{n,\overline{m},i} + \frac{\partial V^k_{n,\overline{m},i}(\tau)}{\partial(S_0 e^{\tau h})}(S_0 e^{(m+1)h} - \bar{A}^k_{n,m,i}) + \frac{1}{2}\frac{\partial^2 V^k_{n,\overline{m},i}(\tau)}{\partial(S_0 e^{\tau h})^2}(S_0 e^{(m+1)h} - \bar{A}^k_{n,m,i})^2$$

$$V^k_{n,m,i} = V^k_{n,\overline{m},i} + \frac{\partial V^k_{n,\overline{m},i}(\tau)}{\partial(S_0 e^{\tau h})}(S_0 e^{mh} - \bar{A}^k_{n,m,i}) + \frac{1}{2}\frac{\partial^2 V^k_{n,\overline{m},i}(\tau)}{\partial(S_0 e^{\tau h})^2}(S_0 e^{mh} - \bar{A}^k_{n,m,i})^2.$$

where $\tau \in [m, m+1]$.

Insert the above equation and (3.12) into (3.14), we have:

$$\xi^k_{n,\overline{m},i} = |\frac{(S_0e^{(m+1)h} - \bar{A}^k_{n,m,i})(\bar{A}^k_{n,m,i} - S_0e^{mh})}{2}\frac{\partial^2 V^k_{n,\overline{m},i}(\tau)}{\partial(S_0e^{\tau h})^2}| \tag{3.15}$$

From P. A. Forsyth and Zvan (2002) we can assume that for any i, k, n, there exists a number M, which is large enough and independent of Δt, such that $|\frac{\partial^2 V^k_{n,\overline{m},i}(\tau)}{\partial(S_0e^{xh})^2}| \leq M$; and there exists A^* such that the effect of interpolation errors induced at $S_0e^{\tau h} > A^*$ is negligible at t_0. Then (3.15) can be bounded as

$$\begin{aligned} \xi^k_{n,\overline{m},i} &\leq |\frac{(S_0e^{(m+1)h} - \bar{A}^k_{n,m,i})(\bar{A}^k_{n,m,i} - S_0e^{mh})}{2}M|, \\ &\leq |\frac{(S_0e^{(m+1)h} - S_0e^{mh})^2}{2}M|, \\ &\leq M(A^*)^2(1 - e^{-h})^2 \end{aligned} \tag{3.16}$$

At the k-th step, let the real value of Asian option of regime i ,node n and representative value m be $R^k_{n,m,i}$, then we have

$$
\begin{aligned}
R^k_{n,m,i} &= e^{-r_i\Delta t}\sum_{j=0}^{\mathcal{M}} p_{ij}\Big[\big[\pi^u_j(1-\lambda_j\Delta t)+\lambda_j\Delta t\, d\mathcal{F}_j(1)\big] \\
&\quad\cdot\overline{R}^{k+1}\Big(n+d+1,\frac{kS_0e^{mh}+S_{k,n}wa}{k+1},j\Big) \qquad (3.17)\\
&\quad+\big[\pi^m_j(1-\lambda_j\Delta t)+\lambda_j\Delta t\, d\mathcal{F}_j(0)\big]\overline{R}^{k+1}\Big(n+d,\frac{kS_0e^{mh}+S_{k,n}w}{k+1},j\Big)\\
&\quad+\big[\pi^d_c(1-\lambda_j\Delta t)+\lambda_j\Delta t\, d\mathcal{F}_j(-1)\big]\overline{R}^{k+1}\Big(n+d-1,\frac{kS_0e^{mh}+S_{k,n}wb}{k+1},j\Big)\\
&\quad+\sum_{\gamma\neq 1,0,-1}\lambda_j\Delta t\, d\mathcal{F}_j(\gamma)\overline{R}^{k+1}\Big(n+d+\gamma,\frac{kS_0e^{mh}+S_{k,n}wa^{\gamma}}{k+1},j\Big)\Big]\\
&\quad+\mathit{interpolation\ error}+\mathit{truncation\ error}.
\end{aligned}
$$

Let $E^k_{n,m,i}=R^k_{n,m,i}-V^k_{n,m,i}$, we have

$$
\begin{aligned}
E^k_{n,m,i} &= e^{-r_i\Delta t}\sum_{j=0}^{\mathcal{M}} p_{ij}\Big[\big[\pi^u_j(1-\lambda_j\Delta t)+\lambda_j\Delta t\, d\mathcal{F}_j(1)\big] \\
&\quad\cdot\overline{E}^{k+1}\Big(n+d+1,\frac{kS_0e^{mh}+S_{k,n}wa}{k+1},j\Big) \qquad (3.18)\\
&\quad+\big[\pi^m_j(1-\lambda_j\Delta t)+\lambda_j\Delta t\, d\mathcal{F}_j(0)\big]\overline{E}^{k+1}\Big(n+d,\frac{kS_0e^{mh}+S_{k,n}w}{k+1},j\Big)\\
&\quad+\big[\pi^d_c(1-\lambda_j\Delta t)+\lambda_j\Delta t\, d\mathcal{F}_j(-1)\big]\overline{E}^{k+1}\Big(n+d-1,\frac{kS_0e^{mh}+S_{k,n}wb}{k+1},j\Big)\\
&\quad+\sum_{\gamma\neq 1,0,-1}\lambda_j\Delta t\, d\mathcal{F}_j(\gamma)\overline{E}^{k+1}\Big(n+d+\gamma,\frac{kS_0e^{mh}+S_{k,n}wa^{\gamma}}{k+1},j\Big)\Big]\\
&\quad+\mathit{interpolation\ error}+\mathit{truncation\ error}.
\end{aligned}
$$

And from (3.12) we have:

$$
\overline{E}^k(n,\bar{A}^k_{n,m,i},i)=\frac{\bar{A}^k_{n,m,i}-S_0e^{mh}}{S_0e^{mh}(e^h-1)}E^k_{n,m+1,i}+\frac{S_0e^{(m+1)h}-\bar{A}^k_{n,m,i}}{S_0e^{mh}(e^h-1)}E^k_{n,m,i} \qquad (3.19)
$$

Define

$$
\|E^k(i)\|_\infty := \max_{n,m}|E_{n,m,i}|, \quad k=0,1,\dots,N-1;\ i\in\mathbb{D}, \qquad (3.20)
$$

Then from (3.18), we have

$$\begin{aligned}
\|E^k(i)\|_\infty &= e^{-r_i\Delta t}\sum_{j=0}^{\mathcal{M}} p_{ij}\Big[\big[\pi_j^u(1-\lambda_j\Delta t)+\lambda_j\Delta t\, d\mathcal{F}_j(1)\big]\|E^{k+1}(j)\|_\infty \qquad (3.21)\\
&\quad +\big[\pi_j^m(1-\lambda_j\Delta t)+\lambda_j\Delta t\, d\mathcal{F}_j(0)\big]\|E^{k+1}(j)\|_\infty\\
&\quad +\big[\pi_c^d(1-\lambda_j\Delta t)+\lambda_j\Delta t\, d\mathcal{F}_j(-1)\big]\|E^{k+1}(j)\|_\infty\\
&\quad +\sum_{\gamma\neq 1,0,-1}\lambda_j\Delta t\, d\mathcal{F}_j(\gamma)\|E^{k+1}(j)\|_\infty\Big]\\
&\quad +\textit{interpolation error}+\textit{truncation error}\\
&\quad \leq \sum_{j=0}^{\mathcal{M}} p_{ij}\|E^{k+1}(j)\|+\textit{interpolation error}+\textit{truncation error}\\
&= \big[(1+a_{ii}\Delta t)\|E^{k+1}(i)\|+a_{ij}\Delta t\|E^{k+1}(j)\|\big] \qquad (3.22)\\
&\quad +\textit{interpolation error}+\textit{truncation error}.
\end{aligned}$$

Denote vector $\mathbf{\Psi}_i = \big(\|E^k(0)\|_\infty, \|E^k(1)\|_\infty\big)'$. Then (3.21) can be re-written into a vector form

$$\mathbf{\Psi}_i \leq \mathbf{D}\mathbf{\Psi}_{i+1} + |O((\Delta t)^2)|\mathbf{1} + \textit{interpolation error} + \textit{truncation error}, \qquad (3.23)$$

where $\mathbf{1}=(1,1)'$ is a 2-dimension vector and $\mathbf{D}$ is a 2×2 matrix

$$\mathbf{D} = \begin{bmatrix} 1+a_{00}\Delta t & a_{01}\Delta t \\ a_{10}\Delta t & 1+a_{11}\Delta t \end{bmatrix}.$$

Since each element of matrix $\mathbf{D}$ is nonnegative, iterating of inequality (3.23) gives that

$$\begin{aligned}
\mathbf{\Psi}_0 &\leq \mathbf{D}^n\mathbf{\Psi}_n + \left[\mathbf{I}+\sum_{m=1}^{n}\mathbf{D}^m\right]|O((\Delta t)^2)|\mathbf{1}\\
&\quad +\textit{interpolation error}+\textit{truncation error}, \qquad (3.24)
\end{aligned}$$

where $\mathbf{I}$ is a 2×2 identity matrix and

$$\mathbf{I} + \sum_{m=1}^{n} \mathbf{D}^m$$

$$= \begin{bmatrix} (n-i) + \frac{n(n-1)}{2} a_{00} \Delta t & \frac{n(n-1)}{2} a_{01} \Delta t \\ \frac{n(n-1)}{2} a_{10} \Delta t & (n-i) + \frac{n(n-1)}{2} a_{11} \Delta t \end{bmatrix}$$

$$+ \begin{bmatrix} |O((\Delta t)^2)| & |O((\Delta t)^2)| \\ |O((\Delta t)^2)| & |O((\Delta t)^2)| \end{bmatrix}.$$

Since at the terminal time $t_n \equiv T$ the CLMR value equals the true option value, we know that $\mathbf{\Psi}_n$ is a zero vector. Therefore from (2.31) using $\Delta t = T/n$, we obtain that

$$\|E^0\|_\infty = |O(\Delta t)| + \|E^0\|^I + \|E^0\|^T. \tag{3.25}$$

where $\|E^0\|^I$ is global interpolation error and $\|E^0\|^T$ is global truncation error. From (3.16), we know

$$\|E^0\|^I = \sum_{k=0}^{N-1} \|\xi^k_{n,m,i}\|_\infty \le NM(A^*)^2(1-e^{-h})^2 = |O(\frac{h^2}{\Delta t})| \tag{3.26}$$

So if we let $h = C\Delta t$, C is a constant, we have

$$\|E^0\|^I = |O(\Delta t)|, \tag{3.27}$$

And from Theorem 2.3, we know

$$\|E^0\|^T = |O(\Delta t)|, \tag{3.28}$$

Finally, rewrite (3.25) as

$$\|E^0\|_\infty = |O(\Delta t)| + |O(\Delta t)| + |O(\Delta t)| = |O(\Delta t)|. \tag{3.29}$$

Then we finish the proof.

□

Chapter 4

Numerical Results

Example 4.0.1. At first, We use this example to verify the convergence rates of the CLMR tree. We take the regime-switching jump-diffusion of Merton type in Costabile et al. M. Costabile and Russo (2014). In table 1, we report the prices of European call options with maturity $T = 1$ year in a two-regime economy for different numbers of time steps N. The risk-free rate is $r = 0.1$ in both regimes, while the high-volatility regime is characterized by $\sigma_0 = 0.6$ and the low-volatility one by $\sigma_1 = 0.2$. Other parameters are taken as: $S_0 = 10$, $K = 10$, $a_{0,1} = a_{1,0} = 1$, $\lambda_0 = \lambda_1 = 7$, $Z_k^l \sim N(\eta_l, \delta_l)$ for $l = 0, 1$. The parameters are given as $\eta_0 = -0.02$, $\eta_1 = -0.01125$, $\delta_0 = 0.2$, $\delta_1 = 0.15$. This example has explicit solution as in M. Costabile and Russo (2014). The convergence rates are calculated by the commonly used formula (see e.g., Ma and Zhu (2015)):

$$\text{Rate} = \log \left| \frac{\text{Error with number of time step } N_1}{\text{Error with number of time step } N_2} \right| \Big/ \log \left(\frac{N_2}{N_1} \right).$$

Table 4.1: The convergence rates of the CLMR tree for Example 4.0.1, the exact option value is 1.8542 for regime 0 and 1.3737 for regime 1.

N	Regime 0			Regime 1		
	Value	Abs Error	Rate	Value	Abs Error	Rate
40	1.73608	0.11811	1.1	1.28852	0.08517	1.0
80	1.79871	0.05548	1.0	1.33050	0.04319	1.0
160	1.82734	0.02685	1.1	1.35159	0.02210	1.0
320	1.84126	0.01293	1.4	1.36296	0.01073	1.0
640	1.84927	0.00492	0.9	1.36827	0.00542	0.9
1280	1.85677	0.00256	–	1.37089	0.00280	–

Example 4.0.2. In this example, we consider the at-the-money American put option under regime-switching jump-diffusion model in Costabile et al. M. Costabile and Russo (2014). The parameters are taken as: $r_0 = r_1 = 0.08$,$\sigma_0 = 0.3$,$\sigma_1 = 0.1$, $T = 1$, $S_0 = 40$, $K = 40$, $a_{0,1} = a_{1,0} = 0.5$, $\lambda_0 = \lambda_1 = 5$, $Z_k^l \sim N(\eta_l, \delta_l)$ for $l = 0, 1$. The parameters are given as $\eta_0 = \eta_1 = -0.025$, $\delta_0 = \delta_1 = \sqrt{0.05}$. We choose the numerical example in M. Costabile and Russo (2014) as our benchmark. The exact option value is 2.9577 for regime 0 and 2.4703 for regime 1. The convergence rates are calculated by the same formula as Example 4.1.

Table 4.2: The convergence rates of the CLMR tree for Example 4.0.2

N	Regime 0			Regime 1		
	Value	Abs Error	Rate	Value	Abs Error	Rate
40	2.97130	0.01360	1.4	2.52493	0.05463	1.0
80	2.96283	0.00513	1.0	2.49846	0.02816	1.0
160	2.96032	0.00262	0.9	2.48474	0.01444	0.9
320	2.95914	0.00144	0.9	2.47815	0.00785	0.8
640	2.95845	0.00075	0.9	2.47470	0.00440	0.9
1280	2.95720	0.00041	–	2.47253	0.00223	–

Example 4.0.3. In this example we verify the convergence rates of the CLMR trees and finite difference methods (FDMs). We consider the European option under the regime-switching jump-diffusion models (1.1) with two regime economy $\mathbb{D} = \{0, 1\}$. The interest rates at state 0 and 1 are $r_0 = r_1 = 0.05$, while volatilities are $\sigma_0 = 0.25$ and $\sigma_1 = 0.15$ respectively. The maturity date of the European call option is taken as $T = 1$, the strike price $K = 100$ the initial values of underlying asset $S_0 = 92$. Furthermore, the generator matrix is taken as

$$A = \begin{pmatrix} -0.5 & 0.5 \\ 0.5 & -0.5 \end{pmatrix},$$

or

$$B = \begin{pmatrix} -2/3 & 2/3 \\ 1/3 & -1/3 \end{pmatrix}.$$

The jump diffusion is a double exponential type of Kou Kou (2002) with regime dependency. The density function $f_l(z)$, $l = 0, 1$, for the double exponential distribution for the jump sizes is given by:

$$f_l(z) = \begin{cases} p_l \eta_{1,l} e^{-\eta_{1,l} z}, & \text{if } z \geq 0, \\ (1 - p_l)\eta_{0,l} e^{\eta_{0,l} z}, & \text{if } z < 0, \end{cases} \tag{4.1}$$

where $\eta_{1,l} > 1$, $\eta_{0,l} > 0$, and $0 < p_l < 1$. In this case, we have

$$m_l = \frac{(1 - p_l)\eta_{0,l}}{\eta_{0,l} + 1} + \frac{p_l \eta_{1,l}}{\eta_{1,l} - 1} - 1.$$

The parameters for the jump density functions are taken as: $\eta_{0,0} = \eta_{0,1} = 3.0775$, $\eta_{1,0} = \eta_{1,1} = 3.0365$, $p_0 = p_1 = 0.3445$. In addition we choose the intensities $\lambda_0 = 5, \lambda_1 = 2$.

From the Table 4.3, we obtain that the convergence rates for both CLMR trees and perturbed FDMs are about 1. What's more, the absolute difference between the option values computed by CLMR trees and perturbed FDMs, which is denoted by "Diff" in the tables, is decreasing in N with rate 1. All these results are consistent with the theoretical findings.

Example 4.0.4. In this example we verify the accuracy of the CLMR tree for pricing arithmetic Asian option. We consider a two-regime economy and the parameters are the same as Example (4.0.1). In Table 4 - Table 6 we present the values of in-the money, at-the-money and out-the-money Asian call option with strike price $K = 100$.

We also carried out Monte Carlo simulations with 200000 sample paths and 1000 time step. In Table 7, we report, for selected values of S_0, the mean(the 'MC' row) and the 95% confidence interval, obtained by taking average of 10 simulation.

In the following tables, the 'Diff' columns represent the difference between the value from CLMR tree and the MC results and the 'Rate' columns are calculated by the formula in Example (4.0.1). It is clear that the results in Table are in good agreement with the Monte Carlo result.

Remark 0.1. In order to asses the performances of the European and Asian pricing formulas. We can use the data of real world to verify the model. To estimate the parameters of the model is important. In Andersen and Andreasen (2000), they proposed a method about fitting the jump parameters using S&P 500 index. And for regime-switching parameters,

Table 4.3: Comparisons of CLMR trees and FDMs for pricing European call option under the regime-switching jump-diffusion model (1.1) for Example 4.0.3.

Regime 0 (generator A)								
N	CLMR			FDM			Diff	Rate
	Value	Abs Error	Rate	Value	Abs Error	Rate		
40	36.75208	0.087849	0.8	36.48706	0.223257	1.0	0.265022	1.0
80	36.83993	0.051961	1.0	36.71031	0.111961	0.8	0.129614	0.9
160	36.89189	0.026296	1.4	36.82227	0.065056	1.4	0.069614	1.2
320	36.91818	0.009677	0.8	36.88733	0.025297	1.0	0.030854	1.0
640	36.92786	0.005482	1.3	36.91263	0.012865	1.0	0.015234	1.0
1280	36.93334	0.002168	–	36.92549	0.006407	–	0.007851	1.1
2560	36.93551	–	–	36.93190	–	–	0.003612	–
Regime 1 (generator A)								
N	CLMR			FDM			Diff	Rate
	Value	Abs Error	Rate	Value	Abs Error	Rate		
40	35.33844	0.067728	1.2	35.21259	0.135188	1.2	0.125855	1.0
80	35.40617	0.030275	0.8	35.34778	0.060515	0.9	0.058395	1.0
160	35.43645	0.016850	0.8	35.40829	0.031354	0.9	0.028155	1.0
320	35.45330	0.009977	1.1	35.43964	0.016951	1.1	0.013651	1.0
640	35.46327	0.004760	0.9	35.45660	0.008161	0.9	0.006677	1.0
1280	35.46803	0.002592	–	35.46476	0.004255	–	0.003276	1.0
2560	35.47062	–	–	35.46901	–	–	0.001613	–
Regime 0 (generator B)								
N	CLMR			FDM			Diff	Rate
	Value	Abs Error	Rate	Value	Abs Error	Rate		
40	36.872847	0.0870587	0.9	36.83927	0.103663	0.9	0.033572	1.0
80	36.959905	0.0463317	0.9	36.94293	0.054781	1.0	0.016967	1.0
160	37.006237	0.0240329	1.0	36.99762	0.028314	1.0	0.008507	1.0
320	37.030270	0.0116572	0.9	37.02603	0.013799	0.9	0.004235	1.0
640	37.041927	0.0065824	0.9	37.03983	0.007236	0.9	0.002113	1.0
1280	37.048159	0.0033689	–	37.04707	0.003915	–	0.001089	1.0
2560	37.051528	–	–	37.05098	–	–	0.000542	–
Regime 1 (generator B)								
N	CLMR			FDM			Diff	Rate
	Value	Abs Error	Rate	Value	Abs Error	Rate		
40	35.45690	0.119247	1.0	35.44267	0.126363	1.0	0.014230	1.0
80	35.57616	0.059672	1.0	35.56904	0.063196	1.0	0.007114	1.0
160	35.63582	0.030125	1.0	35.63223	0.031949	1.0	0.003590	1.0
320	35.66595	0.014865	1.0	35.66418	0.015759	1.0	0.001766	1.0
640	35.68081	0.007623	1.0	35.67994	0.008060	1.0	0.000872	1.0
1280	35.68844	0.003789	–	35.68800	0.004001	–	0.000435	1.0
2560	35.69223	–	–	35.69200	–	–	0.000223	–

Ramponi (2012) first used real data to estimate the parameter.

For our model, we can fit the regime-switching jump diffusion model on a set of observed call prices on the S&P 500 index to get realistic values for the parameters. The average of the bid and ask Treasury bill discounts can be used and converted to annualized risk-free rates. As for the dividend rate q, following Ramponi (2012), it can be estimated by a non linear least squares algorithm. For the jump parameters and local volatility, we can follow Andersen and Andreasen (2000)'s method.

Table 4.4: The convergence rates of the CLMR trees for out-the-money Asian call option with $S_0 = 60$. MC values are 2.6418 for regime 0 and 0.9377 for regime 1.

N	Regime 0			Regime 1		
	Value	Diff	Rate	Value	Diff	Rate
20	2.1501	0.4917	1.0	0.6386	0.2991	1.1
40	2.3873	0.2545	1.0	0.7975	0.1402	1.1
80	2.5101	0.1317	1.0	0.8713	0.0664	1.0
160	2.5798	0.062	1.1	0.9034	0.0343	1.0
320	2.6133	0.0285	–	0.9211	0.0166	–

Table 4.5: The convergence rates of the CLMR trees for at-the-money Asian call option with $S_0 = 100$. MC values are 19.5855 for regime 0 and 14.8243 for regime 1.

N	Regime 0			Regime 1		
	Value	Diff	Rate	Value	Diff	Rate
20	18.7257	0.8598	0.9	14.0132	0.8111	1.1
40	19.1366	0.4489	1.0	14.4065	0.4178	1.0
80	19.3579	0.2276	1.1	14.6242	0.2001	0.9
160	19.4831	0.1024	1.0	14.7181	0.1062	1.1
320	19.5334	0.0521	–	14.7768	0.0475	–

Table 4.6: The convergence rates of the CLMR trees for in-the-money Asian call option with $S_0 = 140$. MC values are 51.5671 for regime 0 and 48.8914 for regime 1.

N	Regime 0			Regime 1		
	Value	Diff	Rate	Value	Diff	Rate
20	50.6716	0.8955	1.0	47.8573	1.0341	1.2
40	51.1104	0.4567	1.0	48.4326	0.4588	1.1
80	51.3302	0.2369	1.1	48.6739	0.2001	0.9
160	51.4481	0.119	1.2	48.7867	0.1047	1.1
320	51.5139	0.0532	–	48.8423	0.0491	–

Table 4.7: The convergence rates of the CLMR trees for fixed strike Asian call option for Example 4.0.4.

	Regime 0			Regime 1		
S_0	60	100	140	60	100	140
MC	2.6418	19.5855	51.5671	0.9377	14.8243	48.8914
95%CI	[2.58,2.70]	[19.41,19.76]	[51.27,51.86]	[0.91,0.97]	[14.71,14.94]	[48.69,49.10]
Stand Error	0.445	1.266	2.143	0.199	0.824	1.453

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this paper we introduce the regime-switching jump diffusion model and the multinomial tree method established by M. Costabile and Russo (2014). We prove the the convergence rates of CLMR tree by establishing the equivalence of convergence between the multinomial tree model and the finite difference method for European option pricing. In addition, we compare the multinomial tree of CLMR M. Costabile and Russo (2014) with the tree methods proposed by J.X. Jiang and Nguyen (2016). Moreover, we have extended the CLMR tree to the price of fixed strike Asian option under the regime-switching jump-diffusion model using the representative value method in Hull and White (1993). We also prove the convergence rates. Numerical examples are provided to verify the convergence rates. The option prices calculated by the CLMR tree model are close to the prices obtained through Monte Carlo Method.

5.2 Future Work

In future work, we may focus on how to use the CLMR to hedge the Greeks like Delta and Gamma. As we all know, for binomial tree, the numerical Greeks are easy to calculated. For numerical Delta, we have

$$\Delta = \frac{f_u - f_d}{S_u - S_d},$$

where f is the value of derivative and S is the stock price. For numerical Gamma, we have

$$\Gamma = \frac{\frac{f_{uu}-f_{dd}}{S_{uu}-S_{ud}} - \frac{f_{ud}-f_{dd}}{S_{ud}-S_{dd}}}{(S_{uu} - S_{dd})/2}$$

However, under the regime-switching jump diffusion model implies that the stock index and the money market account do not together form a complete market, i.e. due to the regime switching risk, the market is incomplete.

Following H.Yang (2010), assuming there are K regimes, then each of the derivatives has k price information for the k regimes. Thus if we want to hedge, we need add $K-1$ derivatives to complete the market. Then we can construct a unique risk neutral transition probability for pricing. However, after adding $K-1$ derivatives, it's not easy for us to construct the portfolio to hedge the option due to the complexity of the CLMR tree. So in future, we may focus on the delta hedging and gamma hedging using this tree and studying their convergence rate.

Notes

Bibliography

Andersen, L. and J. Andreasen. 2000. Jump-diffusion processes: Volatility smile fitting and numerical methods for option pricing. *Rev. Derivatives Res.* 4: 231–262.

C. Albanese, S. Jaimungal and D. Rubisov. 2003. A two state jump model. *Quantitative Finance* 3: 145154.

Chourdakis, K. 2004. Non-affine option pricing. *Journal of Derivatives* 3: 1025.

D. Dang, D. Nguyen and G. Sewell. 2016. Numerical schemes for pricing asian options under state-dependent regime-switching jump-diffusion models. *Computers and Mathematics with Applications* 71: 443–458.

Elliott, R.J. and C.J.U. Osakwe. 2006. Option pricing for pure jump processes with markov switching compensators. *Finance and Stochastics* 10: 250275.

Hull, J.C. and A. White. 1993. Efficient procedures for valuing european and american path-dependent options. *Journal of Derivatives* 1: 21–31.

H.Yang. 2010. Option pricing with regime switching by trinomial tree method. *Journal of Computational and Applied Mathematics* 233: 18211833.

I. Florescu, M.C. Mariani, R.H. Liu and G. Sewell. 2013. Numerical schemes for option pricing in regime-switching jump diffusion models. *International Journal of Theoretical and Applied Finance* 16: 1–25.

J. Ma, H. Tang and S.P. Zhu. 2018. Connection between trinomial trees and finite difference methods for option pricing with state-dependent switching rates. *International Journal of Computer Mathematics* 95: 341–360.

Jiang, L. 2005. Mathematical modeling and methods for option pricing. *World Scientific* .

J.X. Jiang, R.H. Liu and D. Nguyen. 2016. A recombining tree method for option pricing with state-dependent switching rates. *International Journal of Theoretical and Applied Finance* 19: 21–29.

Kou, S. 2002. A jump-distribution model for option pricing. *Management Science* 48: 1086–1101.

M. Costabile, I. Massabó, A. Leccadito and E. Russo. 2014. Option pricing under regime-switching jump-diffusion models. *Journal of Computational and Applied Mathematics* 38: 152–157.

M. Konikov, D.B. Madan. 2002. Option pricing using variance gamma markov chains. *Review of Derivatives Research* 31: 81–115.

Ma, J. and T. Zhu. 2015. Convergence rate of trinomial tree methods for option pricing under regime-switching models. *Applied Mathematics Letters* 39: 13–18.

Naik, V. 1993. Option valuation and hedging strategies with jumps in the volatility of asset returns. *Journal of Finance* 5: 19691984.

P. A. Forsyth, K. R. Vetzal and R. Zvan. 2002. Convergence of numerical methods for valuing pathdependent options using interpolation. *Review of Derivatives Research* 5: 273–314.

Ramponi, A. 2012. Fourier transform methods for regime-switching jump-diffusions and the pricing of forward starting options. *International Journal of Theoretical and Applied Finance* 15: 31–45.

Yuen, F. and H. Yang. 2009. Option pricing in a jump-diffusion model with regime-switching. *ASTIN Bulletin* 39: 515–539.

Yuen, F. and H. Yang. 2010. Option pricing with regime switching by trinomial tree method. *Journal of Computational and Applied Mathematics* 233: 1821–1833.

Yuen, F. and H. Yang. 2012. Pricing asian options and equity-indexed annuities with regime switching by the trinomial tree method. *North American Actuarial Journal* 14: 256–272.

Appendix

```
clear
l=5;
lamda1=7;
lamda2=7;
yita1=-0.02;
yita2=-0.01125;
sigma1=0.2;
sigma2=0.15;
sg1=0.6;
sg2=0.2;
sg=sqrt(1.5)*sg1;
t=1;
q11=-1;
q12=1;
q21=1;
q22=-1 ;
n=40;
```

h=1/n;

r1=0.1;

r2=0.1;

b1=r1-(sg1)^2/2;

$b2 = r2 - (sg2)^2/2;$

$l1 = 1;$

$l2 = 1;$

$k1 = exp(yita1 + sigma1^2/2) - 1;$

$k2 = exp(yita2 + sigma2^2/2) - 1;$

$a1 = b1 - lamda1 * k1;$

$a2 = b2 - lamda2 * k2;$

$p2u = ((sg2)^2 + (a2 - a1) * (l1 * sg) * (h)^0.5 + (a2 - a1)^2 * h)/(2 * (l1 * sg)^2);$

$p2d = ((sg2)^2 - (a2 - a1) * (l1 * sg) * (h)^0.5 + (a2 - a1)^2 * h)/(2 * (l1 * sg)^2);$

$p2m = 1 - (p2u + p2d);$

$p1u = 1/3;$

$p1d = 1/3;$

$p1m = 1/3;$

$u = exp(sg * h^0.5);$

$d = 1/u;$

$p11 = exp(q11 * h);$

$p12 = (1 - p11);$

$p21 = (1 - exp(q22 * h)) * (q21/(-q22));$

$p22 = 1 - p21;$

$s0 = 100;$

$k = 100;$

$for i = 1 : 1 : 2 * l + 1;$

$pl1(i, i) = (N1(yita1, sigma1, a1 * h + (i - l - 1 + 0.5) * sg * h^0.5) - N1(yita1, sigma1, a1 * h + (i - l - 1 - 0.5) * sg * h^0.5)) * lamda1 * h;$

end

$for i = 1 : 1 : 2 * l + 1;$

$pl2(i, i) = (N2(yita2, sigma2, a1 * h + (i - l - 1 + 0.5) * sg * h^0.5) - N2(yita2, sigma2, a1 * h + (i - l - 1 - 0.5) * sg * h^0.5)) * lamda2 * h;$

end

$x0 = 0;$

$for j = 1 : n + 1;$

$for i = (n - j + 1) * l + 1 : (n + j - 1) * l + 1;$

$x(i, j) = (j - 1) * a1 * h + (i - (n * l + 1)) * sg * h^0.5;$

$S(i, j) = s0 * exp(x(i, j));$

end

end

$for j = 1 : n + 1;$

$for i = (n - j + 1) * l + 1 : (n + j - 1) * l + 1;$

$s1vals(i, j) = max(s0 * exp(x(i, j)) - k, 0);$

$s2vals(i, j) = max(s0 * exp(x(i, j)) - k, 0);$

end

end

$for j = n : -1 : 1;$

$for i = (n - j + 1) * l + 1 : (n + j - 1) * l + 1;$

$s1vals(i,j) = exp(-r1*h)*p11*[svalsparts(s1vals, pl1, i, j, l)+p1u*(1-lamda1*h)*s1vals(i+1, j+1)+p1m*(1-lamda1*h)*s1vals(i, j+1)+p1d*(1-lamda1*h)*s1vals(i-1, j+1)]+exp(-r1*h)*p12*[svalsparts(s2vals, pl1, i, j, l)+p1u*(1-lamda1*h)*s2vals(i+1, j+1)+p1m*(1-lamda1*h)*s2vals(i, j+1)+p1d*(1-lamda1*h)*s2vals(i-1, j+1)];$

$s2vals(i,j) = exp(-r2*h)*p21*[svalsparts(s1vals, pl2, i, j, l)+p2u*(1-lamda2*h)*s1vals(i+1, j+1)+p2m*(1-lamda2*h)*s1vals(i, j+1)+p2d*(1-lamda2*h)*s1vals(i-1, j+1)]+exp(-r2*h)*p22*[svalsparts(s2vals, pl2, i, j, l)+p2u*(1-lamda2*h)*s2vals(i+1, j+1)+p2m*(1-lamda2*h)*s2vals(i, j+1)+p2d*(1-lamda2*h)*s2vals(i-1, j+1)];$

end

end

$s1vals(n*l+1, 1)$

$s2vals(n*l+1, 1)$

$clear;$

$T = 1;$

$n = 40;$

$S0 = 60;$

$K = 100;$

$h = 0.1;$

$u = 10;$

$d = 10;$

$l1 = 1;$

$l2 = 1;$

$lamda1 = 7;$

$lamda2 = 7;$

$r1 = 0.1;$

$r2 = 0.1;$

$r = r1;$

$sg1 = 0.6;$

$sg2 = 0.2;$

$sg = sqrt(1.5) * sg1;$

$b1 = r1 - (sg1)^2/2;$

$b2 = r2 - (sg2)^2/2;$

$yita1 = -0.02; yita2 = -0.01125;$

$sigma1 = 0.2;$

$sigma2 = 0.15;$

$dt = T/n;$

$m1 = exp(yita1 + sigma1^2/2) - 1;$

$m2 = exp(yita2 + sigma2^2/2) - 1;$

$beta = b1 - lamda1 * m1;$

$beta2 = b2 - lamda2 * m2;$

$p2u = 0.2;$

$p2m = 0.6;$

$p2d = 0.2;$

$p1u = 1/3;$

$p1d = 1/3;$

$p1m = 1/3;$

$fori = 1 : 1 : u + d + 1;$

$pl1(i, i) = (N1(yita1, sigma1, beta * dt + (i - d - 1 + 0.5) * sg * dt^0.5) - N1(yita1, sigma1, beta * dt + (i - d - 1 - 0.5) * sg * dt^0.5)) * lamda1 * dt;$

end

$fori = 1 : 1 : u + d + 1;$

$pl2(i, i) = (N2(yita2, sigma2, beta * dt + (i - d - 1 + 0.5) * sg * dt^0.5) - N2(yita2, sigma2, beta * dt + (i - d - 1 - 0.5) * sg * dt^0.5)) * lamda2 * dt;$

end

$pl1(d, d) = pl1(d, d) + p1u * (1 - lamda1 * dt);$

$pl1(d + 1, d + 1) = pl1(d + 1, d + 1) + p1m * (1 - lamda1 * dt);$

$pl1(d + 2, d + 2) = pl1(d + 2, d + 2) + p1d * (1 - lamda1 * dt);$

$pl2(d, d) = pl2(d, d) + p2u * (1 - lamda2 * dt);$

$pl2(d + 1, d + 1) = pl2(d + 1, d + 1) + p2m * (1 - lamda2 * dt);$

$pl2(d + 2, d + 2) = pl2(d + 2, d + 2) + p2d * (1 - lamda2 * dt);$

$q11 = -1;$

$q12 = 1;$

$q21 = 1;$

$q22 = -1;$

$p11 = exp(q11 * dt);$

$p12 = (1 - p11);$

$p21 = (1 - exp(q22 * dt)) * (q21/(-q22));$

$p22 = 1 - p21;$

$a = exp(sg * sqrt(dt));$

$b = 1/a;$

$w = exp(beta * dt);$

$trace(1) = 1;$

$for i = 2 : n + 1$

$trace(i) = trace(i - 1) + (u + d) * (i - 1) + 1;$

end

$S(1) = S0;$

$for i = 1 : n$

$for j = 1 : (u + d) * i + 1$

$S(trace(i) + j) = S0 * w^i * (a^(j - d * i - 1));$

end

end

$MM(1, 1) = S0;$

$MM(2, 1) = S0;$

$F = ;$

$F1 = [S0; 0];$

$for i = 2 : n + 1$

$MM(1, i) = ((i - 1) * Fi - 1(1, 1) + S(trace(i - 1) + 1))/(i); MM(2, i) =$
$((i - 1) * Fi - 1(1, end) + S(trace(i - 1) + (u + d) * (i - 1) + 1))/(i);$

$k = 0;$

$semh = [S0];$

$m = [0];$

$while 1$

$k = k + 1;$

$m = [mk];$

$semh = [semh S0 * exp(k * h)];$

$if semh(end) > MM(2, i);$

$break$

end

end

$k = 0;$

$while 1$

$k = k - 1;$

$m = [km];$

$semh = [S0 * exp(k * h) semh];$

$if semh(1) < MM(1, i)$

$break;$

end

end

$Fi = [semh; m];$

end

$V1 = ;$

$V2 = ;$

$for i = 1 : length(Fn + 1(1, :))$

$V1n + 1, 1(i) = max(Fn + 1(1, i) - K, 0);$

$V2n + 1, 1(i) = V1n + 1, 1(i);$

end

$fori = 2 : (u + d) * (n) + 1$

$V1n + 1, i = V1n + 1, 1;$

$V2n + 1, i = V2n + 1, 1;$

end

$fori = n : -1 : 1$

$forj = 1 : (u + d) * (i - 1) + 1$

$V1i, j = ;$

$V2i, j = ;$

$FF1 = Fi(1, :);$

$FF2 = Fi + 1(1, :);$

$ValF = [];$

$FF1Next = zeros(u + d + 1, length(FF1));$

$CalF1 = zeros(u + d + 1, length(FF1));$

$CalF2 = zeros(u + d + 1, length(FF1));$

$CalF12 = zeros(u + d + 1, length(FF1));$

$CalF21 = zeros(u + d + 1, length(FF1));$

$forx = 1 : u + d + 1$

$FF1Next(x, :) = (i * FF1 + S(trace(i) + j + x - 1))/(i + 1);$

$UF = [];$

$DF = [];$

$forww = 1 : length(FF1Next(x, :))$

$kk = FF1Next(x, ww) * ones(1, length(FF2));$

$ind = sum(kk > FF2);$

$ifind == 0$

$ind = 1;$

end

$UF = [UF(V1i + 1, j + x - 1(ind{+}1){-}V1i + 1, j + x - 1(ind)){*}(FF1Next(x, ww){-}$

$FF2(ind))/(FF2(ind + 1) - FF2(ind)) + V1i + 1, j + x - 1(ind)];$

$DF = [DF(V2i + 1, j + x - 1(ind{+}1){-}V2i + 1, j + x - 1(ind)){*}(FF1Next(x, ww){-}$

$FF2(ind))/(FF2(ind + 1) - FF2(ind)) + V2i + 1, j + x - 1(ind)];$

end

$CalF1(x, :) = exp(-dt * r1) * pl1(x, x) * UF;$

$CalF12(x, :) = exp(-dt * r1) * pl2(x, x) * UF;$

$CalF2(x, :) = exp(-dt * r2) * pl2(x, x) * DF;$

$CalF21(x, :) = exp(-dt * r2) * pl1(x, x) * DF;$

end

$ValF1 = sum(CalF1);$

$ValF12 = sum(CalF12);$

$ValF2 = sum(CalF2);$

$ValF21 = sum(CalF21);$

$V1i, j = p11 * ValF1 + p12 * ValF21;$

$V2i, j = p21 * ValF12 + p22 * ValF2;$

end

end

$V11, 1$

$V21, 1$