Stockholms universitet

Studies of Rate of Convergence for Approximations of American Type Options

Yanxiong Li

Masteruppsats i matematisk statistik
Master Thesis in Mathematical Statistics

Masteruppsats 2012:1
Matematisk statistik
Februari 2012

www.math.su.se

Matematisk statistik
Matematiska institutionen
Stockholms universitet
106 91 Stockholm

Mathematical Statistics
Stockholm University
Master Thesis **2012:1**
http://www.math.su.se

Studies of Rate of Convergence for Approximations of American Type Options

Yanxiong Li*

Februari 2012

Abstract

This master thesis presents three different approximation lattice methods, namely binomial sum model, trinomial sum model and skeleton model, for pricing of American options. The underlying stock price process is assumed to be a geometric Gaussian random walk. First we study reward functions for American options and impose certain conditions on simulating price processes to insure the existence of corresponding reward functions. The conditions un- der which the reward functions for approximating price process converge to the corresponding limiting rewards are also presented. Also backward recur- rence algorithm, for reward functions is discussed. The convergence conditions of these three approximation models are tested. Then numerical tests based on the methods are implemented in MATLAB and some comparisons are made.

*Postal address: Mathematical Statistics, Stockholm University, SE-106 91, Sweden. E-mail:liyx2se@gmail.com. Supervisor: Dmitrii Silvestrov.

Acknowledgements

Here I would like to express my deep, sincere gratitude to my supervisor, Professor Dmitrii Silvestrov at Stockholm University, Department of Mathematics. Thanks for his inspirational advice, patience and wide knowledge! I would like to thank associate professor Anatoliy Malyarenko at Mälardalen University for his consultation concerned MATLAB.

Yanxiong Li
December 2011, Stockholm

Contents

List of Figures

List of Tables

1 Introduction

1.1 Background

This paper is devoted to present three different approximation lattice methods for pricing American options. The corresponding convergence conditions are tested, and some numerical tests based on those three methods are implemented in MATLAB and then comparisons are made.

An European option gives the holder the right to buy or sell the underlying asset by a certain date for a certain price. It is well-known that for European options, there is an explicit formula for its value given by Black and Scholes (1973) and Merton (1973). Among others, one way to derive this formula is by solving the so called Black-Scholes fundamental partial differential equation (PDE):

$$\frac{\partial C}{\partial t} + \frac{1}{2}\sigma^2 S^2 + \frac{\partial^2 C}{\partial t^2} + rS\frac{\partial C}{\partial S} = rC \tag{1}$$

with

$$C(0,t) = 0, \quad C(S,t) \sim S \quad \text{as} \quad S \to \infty, \tag{2}$$

for a European call option with value $C(S,t)$ on a non-dividend paying stock, and the boundary condition

$$C(S_T,T) = max(S_T - K, 0). \tag{3}$$

The solution is the celebrated Black-Scholes pricing formulas, shown below, for this option price at time zero.

$$c = S_0 N(d_1) - Ke^{-rT} N(d_2) \tag{4}$$

where

$$\begin{cases} d_1 = \frac{ln(S_0/K)+(r+\sigma^2/2)T}{\sigma\sqrt{T}} \\ d_2 = \frac{ln(S_0/K)+(r-\sigma^2/2)T}{\sigma\sqrt{T}} = d_1 - \sigma\sqrt{T} \end{cases} \tag{5}$$

and $N(\cdot)$ is the cumulative probability distribution function for a standardized normal distribution.

But situations will be more complicated with the American options. The American options can be exercised at any time up to their maturity. But the European options can only be exercised at their maturity time.

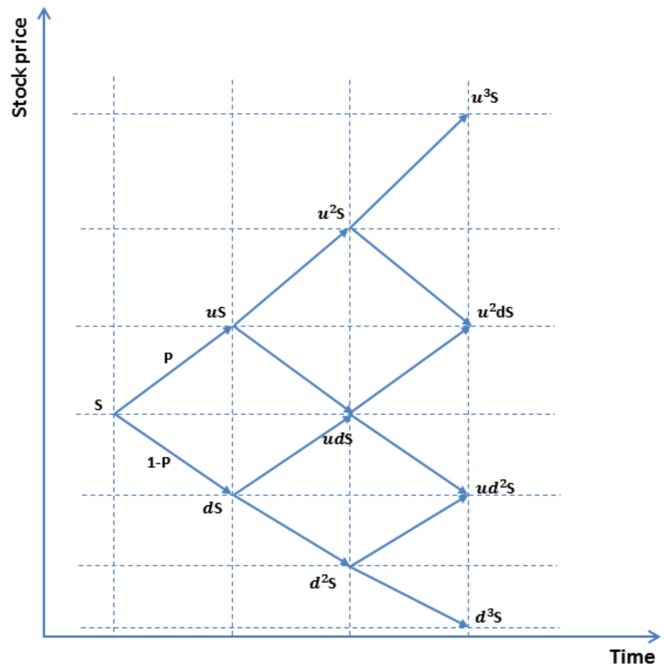

Figure 1: Standard binomial tree

To American options, at each time up to it's maturity, we have to determine not only the option value, but also, for each value of S, whether or not it should be exercised. So in contrast with the boundary condition for European options, formula (3), for American options, we get free boundary conditions. We don't know which boundary conditions to apply, and equally importantly, neither where to apply boundary conditions. Up until now, there is no simple closed-form solution to the American type options. Some recent detailed survey in this field are Broadie and Detemple(2004), Pauly (2004) and Ahn *et al*. (2011).

Stochastical simulation methods are one of those among which the researchers and practitioners can apply to compute the prices of American Options. Further more, these methods can mainly be subdivided into the categories of Monte-Carlo-Simulation (MCS) and approximation lattice methods, such as the binomial and the trinomial model.

The standard binomial medal, see Figure 1, was introduced by Cox, Ross, and Rubinstein (1979) and has won widespread acceptance by its simplicity and efficiency. In pricing the European options, when the stock price parameters are set to $u = e^{\sigma\sqrt{\Delta t}}$ for up movement, $d = 1/u$ down-movement, where $\Delta t = T/n$, and n is the number of time steps between time 0 and the maturity time T, the probability of an upmove is set to $p = (e^{r\Delta t} - d)/(u - d)$. Under such settings, Cox, Ross, and Rubinstein (1979) have proved that the Black-Scholes option pricing formula, formula (4), is the limiting case of binomial model as $n \to \infty$.

The trinomial tree has also been developed by Boyel (1986, 1988), as shown in Figure 2.

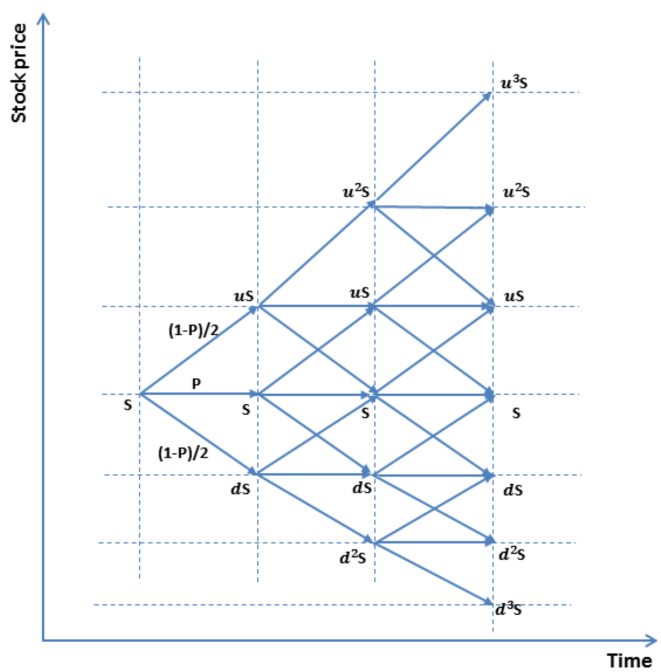

Figure 2: Standard trinomial tree

Based on the binomial method, new improved methods are developed and their rate of convergence are tested, for examples by Broadie and Detemple(1996), Heston and Zhou (2000).

In a perfect market, every kinds of prices, from stocks, real estates to different commodities, are unpredictable. It means that based on the whole information we have gathered until now, we cannot with certainty to know what those prices will be in next month, next week, next day, even in the next minute! In order to handle this kind of questions, in the field of financial mathematics, the price process $S(t)$, $t \geq 0$, is usually assumed to be a geometric Brownian motion,

$$S(t) = S(0)e^{\mu t+\sigma B(t)}, t \geq 0, \tag{6}$$

where $\mu \in \mathbb{R}, \sigma > 0$, are real numbers, $B(t)$ is standard Brownian motion and S(0) is a positive constant.

It is often that we take the logarithm of S(t) and consider so-called log-price price process

$$Y(t) = \ln S(t) = \ln S(0) + \mu t + \sigma B(t), t \geq 0. \tag{7}$$

This treatment will simplify our calculation in the later sections. The main benefit of this treatment is that the multiplicative increments in the case of S(t) changes to additive increments in Y(t). At the same time, because S(t) is just a continuous and well defined function of $Y(t)$, so any conclusion we got for $Y(t)$ can be translated on $S(t)$.

Its own value has also the discrete time analogue of the geometrical Gaussian random walk, which is a discrete time price process given by the following relation,

$$S_{n+1} = S_n e^{\mu_n+\sigma_n B_n}, n = 0, 1, \dots, \tag{8}$$

and the corresponding log-price process,

$$Y_{n+1} = \ln S_{n+1} = Y_n + \mu_n + \sigma_n B_n, n = 0, 1, \dots, \tag{9}$$

where $\mu_n \in \mathbb{R}$ and $\sigma_n > 0, \quad n = 1, 2, \dots$ are real numbers; $B_n, n = 1, 2, \dots$ is a sequence of independent and identically distributed (i.i.d.) random variables with standard normal distribution, $Y(0)$ and $S(0)$ are, respectively, real-value and positive constants connected by the formula $Y(0) = \ln S(0)$.

1.2 Outline

In the next section of this paper, we first present the reward functions for American options in discrete time, and their convergence conditions. After that, we describe three models, namely Binomial-Sum, Trinomial-Sum and Skeleton model, which will be tested to calculate the reward price. In section 3, numerical tests and comparisons with related models are given. Section 4 contains some notations we get from our MATLAB programming. Concluding remarks are given in Section 5. MATLAB codes are collected in Appendix B.

2 Approximation and Convergence for American Type Options

2.1 Models of price processes represented by random walks

In this paper only discrete time setting is studied, so the price and the log-price processes can be written, respectively, as

$$S_{n+1} = S_n e^{W_{n+1}},\ n = 0, 1, \dots, \tag{10}$$

and

$$Y_{n+1} = \ln S_{n+1} = Y_n + W_{n+1}, n = 0, 1, \dots, \tag{11}$$

where: **(a)** $W_n, n = 1, 2, \dots$ is a sequence of real-valued independent and identically distributed (i.i.d.) random variables, and **(b)** Y_0 and S_0 are, respectively, real-value and positive random variables connected by the formula $Y_0 = \ln S_0$ and independent of random variables $W_n, n = 1, 2, \dots$.

The log-price process can also be written in the following integral form,

$$Y_n = Y_0 + \sum_{k=1}^{n} W_k,\ n = 0, 1, \dots. \tag{12}$$

The log-price process Y_n defined above is a random walk, i.e. a discrete time Markov price process with independent increments.

As was mentioned in the introduction, the standard variant of this model is where: **(c)** random variables $W_n = \mu_n + \sigma_n B_n, n = 1, \dots,$ where $\mu_n \in \mathbb{R}$ and $\sigma_n > 0, \quad n = 1, 2, \dots$ are real numbers and **(d)** $B_n, n = 1, 2, \dots$ is a sequence of independent and identically distributed (i.i.d.) random variables with standard normal distribution.

2.2 American type options

Let $\mathcal{F}_n = \sigma[Y_0, \dots, Y_n],\ n = 0, 1, \dots$ be a natural filtration generated by the log-price process Y_n.

It is worth to mention that the price process S_n and the log-price process Y_n will generate the same nature filtration $\mathcal{F}_n$, because these two process are connected by formula (11) .

A nonnegative random variable τ is called an (optional) stopping time with respect to the filtration $\mathcal{F}_n$ if it satisfies the following relation

$$\{\tau > n\} \in \mathcal{F}_n, \text{ for every } n = 0, 1, \ldots. \tag{13}$$

Let also denote by $\mathcal{M}_{n,N}$ the class of all stopping times τ_n such that $n \leq \tau_n \leq N$.

Let us also introduce a payoff function $g(n, y)$, which is a real-valued *continuous* function in y, defined for $(n, y) \in \mathbb{N} \times \mathbb{R}$, where $\mathbb{N} = \{0, 1, \ldots\}$.

The American type option contract in which an option holder has the right, but not the obligation, to execute the contract at any stopping time $\tau \in \mathcal{M}_{n,N}$ and to get in this case the payoff $g(\tau, Y_\tau)$. The parameter N is called a maturity of the option.

One of the goals for an option holder is to find so called reward functions $\phi_n(y),\ y \in \mathbb{R}$ for the option contract defined by the following relation, for $n = 0, \ldots, N$,

$$\phi_n(y) = \sup_{\tau_n \in \mathcal{M}_{n,N}} \mathsf{E}_{y,n} g(\tau_n, Y_{\tau_n}). \tag{14}$$

Here and henceforth, $\mathsf{P}_{y,n}$ and $\mathsf{E}_{y,n}$ denote, respectively, conditional probability and expectation under condition $Y_n = y$.

Below, we shall impose some conditions on the log-price process Y_n and the payoff function $g(n, y)$, which would guarantee that

$$\mathsf{E}_{y,n} \max_{n \leq k \leq N} |g(k, Y_k)| < \infty, \tag{15}$$

where $y \in \mathbb{R},\ n = 0, \ldots, N$ and, thus, the optimal expected reward $|\phi(y)| < \infty,\ y \in \mathbb{R},\ n = 0, \ldots, N$.

A standard examples of payoff functions related to so-called call and put option contracts are, respectively,

$$g(n, y) = e^{-rn} \max(0, e^y - K) = e^{-rn}[e^y - K]_+, \tag{16}$$

and

$$g(n, y) = e^{-rn} \max(0, K - e^y) = e^{-rn}[K - e^y]_+. \tag{17}$$

Here, $K, r > 0$ are positive constants, which are a strike price and a risk-free interest rate respectively.

2.3 Reward functions for American options

We impose the following condition on the the log-price process Y_n, which is assumed to hold for some $\beta \geq 0$:

$\mathbf{A}[\beta]:\ \max_{0\leq n\leq N} \mathsf{E} e^{\pm\beta W_n} < K$, where $1 < K < \infty$.

We also impose the following condition on the payoff function $g(n, y)$, which is assumed to hold for some $\gamma \geq 0$:

$\mathbf{B}[\gamma]:\ \max_{0\leq n\leq N} \sup_{y\in\mathbb{R}} \frac{|g(n,y)|}{1+L''e^{\gamma|y|}} < L'$, where $0 < L' < \infty$ and $0 \leq L'' < \infty$.

This condition will make sure that we only study those payoff functions which have not more than polynomial rate of growth in argument of $e^{|y|}$. For example, in the case of the call option contract, defined by formula (16), by setting $\gamma = 1$, condition $\mathbf{B}[\gamma]$ will be fulfilled. In the case of the put option contract with formula (17), γ can be set to zero to fulfill this condition.

The following theorem is given in Lundgren and Silvestrov (2010).

Theorem 1. *Let conditions $\mathbf{A}[\beta]$ and $\mathbf{B}[\gamma]$ holds for parameters $0 \leq \gamma \leq \beta < \infty$. Then there exist constants $0 \leq M', M'' < \infty$ such that the following inequalities hold for any $y \in \mathbb{R}, n = 0, \ldots, N$,*

$$\begin{aligned} |\phi_n(y)| &= \sup_{\tau_n\in\mathcal{M}_{n,N}} \mathsf{E}_{y,n}|g(\tau_n, Y_{\tau_n})| \\ &\leq \mathsf{E}_{y,n} \max_{n\leq r\leq N} |g(r, Y_r)| \leq M' + M''e^{\gamma|y|}. \end{aligned} \tag{18}$$

2.4 A backward recurrence algorithm for reward functions

Let assume now that the following condition holds:

C: $W_n,\ n = 1, \ldots, N$ are independent discrete random variables such that W_n takes values of $l\delta,\ l = -r_n, -r_n + 1, \ldots, r_n$, for every $n = 1, \ldots, N$, where δ and $r_n,\ n = 1, \ldots, N$ are, respectively a positive real and positive integer numbers.

In this case, the conditional distribution of the random variable Y_m, under the condition $Y_n = y$, is symmetrical and concentrated in points $y + l\delta,\ l = -r_{n,m}, -r_{n,m} + 1, \ldots, r_{n,m}$, for every $0 \leq n \leq m \leq N$, where $r_{n,m} = \sum_{k=n+1}^{m} r_k,\ 0 \leq n \leq m \leq N$.

Note that we assume that δ is not changing with $n,\ n = 1, \ldots, N$, which leads to that the so-called recombining condition for Y_n is fulfilled. We can see that, for example, assuming the change of Y_n at moment $n+1$ is a upmove with $l = 1$ and downmove with $l = -1$ at $n+2$, or contrary, $l = -1$ at $n+1$ and $l = 1$ at $n+2$, the value of Y_{n+2} will be the same in those two cases! By such assumption, at time moment $m,\ n < m$, we will only have a number of $r_{n,m} = \sum_{k=n+1}^{m} r_k,\ 0 \leq n \leq m \leq N$ values of the reward function to calculate! But if the recombining condition is violated, in the extrem case, we will have $r_{n,m} = \prod_{k=n+1}^{m} r_k$ values of the reward function to handle with. The calculation will be gigantic if we want to get a acceptable simulation result without recombining condition!

Under condition **C**, we get

$$\mathsf{E} e^{\pm\beta W_n} = \sum_{l=-r_n}^{r_n} e^{\pm\beta l\delta} P\left(W_n = l\delta\right) \tag{19}$$

Obviously condition **C** implies that condition $\mathbf{A}[\beta]$ holds for any $\beta \geq 0$.

The following theorem is a variant of the corresponding results from Chow, Robbins and Siegmund (1971) and Shiryaev (1976).

Theorem 2. *Let conditions* **C** *holds. Then the reward functions satisfy the following recurrence backward relations for every $y \in \mathbb{R}$ and $0 \leq n \leq N$,*

$$\begin{cases} \phi_N(y + l\delta) = g(N, y + l\delta),\ l = -r_{n,N}, \ldots, r_{n,N}, \\ \phi_m(y + l\delta) = \max\Big(g(m, y + l\delta), \\ \qquad \sum_{k=-r_{m+1}}^{r_{m+1}} \phi_{m+1}(y + l\delta + k\delta))\mathsf{P}(W_{m+1} = k\delta)\Big), \\ l = -r_{n,m}, \ldots, r_{n,m},\ m = N-1, \ldots, n. \end{cases} \tag{20}$$

2.5 Convergence of reward functions

Let consider now the family of log-price processes, which depend on some perturbation parameter $\varepsilon \geq 0$ and are defined for every ε by the following relation similar with (11),

$$Y_{\varepsilon,n+1} = Y_{\varepsilon,n} + W_{\varepsilon,n+1},\ n = 0, 1, \ldots, \tag{21}$$

where: **(a)** $W_{\varepsilon,n},\ n = 1, 2, \ldots$ is a sequence of real-valued independent and identically distributed (i.i.d.) random variables, and **(b)** $Y_{\varepsilon,0}$ is a real-valued constant.

Let $\mathcal{F}_{\varepsilon,n} = \sigma[Y_{\varepsilon,0}, \ldots, Y_{\varepsilon,n}],\ n = 0, 1, \ldots$ be a natural filtration generated by the log-price process $Y_{\varepsilon,n}$.

Let also denote by $\mathcal{M}_{\varepsilon,n,N}$ the class of all stopping times $\tau_{\varepsilon,n}$ for the process $Y_{\varepsilon,n}$ such that $n \leq \tau_{\varepsilon,n} \leq N$.

The following condition will replace condition $\mathbf{A}[\beta]$ in this case:

$\mathbf{A}'[\beta]$: $\overline{\lim}_{\varepsilon\to 0} \max_{0\leq n\leq N} \mathsf{E}e^{\pm\beta W_{\varepsilon,n}} < K'$, where $1 < K' < \infty$.

Condition $\mathbf{A}'[\beta]$ obviously implies that there exists $\varepsilon_0 > 0$ such that condition $\mathbf{A}[\beta]$ holds for every $\varepsilon \in [0, \varepsilon_0]$. In such case, according Theorem 1, there exists, the reward functions $\phi_{\varepsilon,n}(y), y \in \mathbb{R}$ for the option contract defined by the following relation, for $\varepsilon \in [0, \varepsilon_0]$ and $n = 0, \ldots, N$,

$$\phi_{\varepsilon,n}(y) = \sup_{\tau_{\varepsilon,n}\in\mathcal{M}_{\varepsilon,n,N}} \mathsf{E}_{y,n} g(\tau_{\varepsilon,n}, Y_{\varepsilon,\tau_{\varepsilon,n}}). \tag{22}$$

We also impose the following condition of convergence in distribution for jumps of log-price processes:

D: $W_{\varepsilon,n} \xrightarrow{d} W_{0,n}$ as $\varepsilon \to 0$, for $n = 1, \ldots, N$.

The following theorem is a direct corollary of results given in Lundgren and Silvestrov (2010).

Theorem 3. *Let conditions* $\mathbf{A}'[\beta]$ *and* $\mathbf{B}[\gamma]$ *holds for parameters* $0 < \gamma < \beta < \infty$ *or* $\gamma = \beta = 0$*, and also condition* **D** *holds. Then the following relation holds for any* $y \in \mathbb{R},\ n = 0, \ldots, N$*,*

$$\phi_{\varepsilon,n}(y) \to \phi_{0,n}(y) \text{ as } \varepsilon \to 0. \tag{23}$$

2.6 Approximation algorithms for reward functions

Let assume now that the following conditions holds:

C$'$: $W_{\varepsilon,n}, n = 1, \ldots, N$ are, for every $\varepsilon \in (0, \varepsilon_0]$, independent discrete random variables such that the random variable $W_{\varepsilon,n}$ takes values $l\delta_\varepsilon,\ l = -r_{\varepsilon,n}, -r_{\varepsilon,n} + 1, \ldots, r_{\varepsilon,n}$, for $n = 1, \ldots, N$, where δ_ε and $r_{\varepsilon,n}, n = 1, , \ldots, N$ are, respectively a positive real and positive integer numbers.

As was mentioned above, the conditional distribution of the random variable $Y_{\varepsilon,m}$ under the condition $Y_{\varepsilon,n} = y$ is symmetrical and concentrated in points $y + l\delta_\varepsilon,\ l = -r_{\varepsilon,n,m}, -r_{\varepsilon,n,m}+1, \ldots, r_{\varepsilon,n,m}$, for every $0 \leq n \leq m \leq N$, where $r_{\varepsilon,n,m} = \sum_{k=n+1}^{m} r_{\varepsilon,k},\ 0 \leq n \leq m \leq N$.

The following approximation algorithm for computing the reward functions $\phi_{0,n}(y)$ takes place. One should sequentially apply the following two theorems, which are corollaries of Theorems 2 and 3 respectively.

Theorem 4. *Let the the log-price processes $Y_{\varepsilon,n}$ satisfy condition $\mathbf{C}'$. Then the following recurrence backward relations hold for every $y \in \mathbb{R}, 0 \leq n \leq N$ and $\varepsilon \in (0, \varepsilon_0]$,*

$$\begin{cases} \phi_{\varepsilon,N}(y + l\delta_\varepsilon) = g(N, y + l\delta_\varepsilon),\ l = -r_{\varepsilon,n,N}, \ldots, r_{\varepsilon,n,N}, \\ \phi_{\varepsilon,m}(y + l\delta_\varepsilon) = \max\Big(g(m, y + l\delta_\varepsilon), \\ \qquad \sum_{k=-r_{\varepsilon,m+1}}^{r_{\varepsilon,m+1}} \phi_{\varepsilon,m+1}(y + l\delta_\varepsilon + k\delta_\varepsilon))\mathsf{P}(W_{\varepsilon,m+1} = k\delta_\varepsilon)\Big), \\ l = -r_{\varepsilon,n,m}, \ldots, r_{\varepsilon,n,m},\ \text{where } m = N-1, \ldots, n. \end{cases} \tag{24}$$

Theorem 5. *Let the the log-price processes $Y_{\varepsilon,n}$ satisfy condition $\mathbf{A}'[\beta]$, $\mathbf{B}[\gamma]$ and $\mathbf{C}'$, with parameters $0 < \gamma < \beta < \infty$ or $\gamma = \beta = 0$, and also condition $\mathbf{D}$ holds. Then the following relation holds for any $y \in \mathbb{R}, n = 0, \ldots, N$,*

$$\phi_{\varepsilon,n}(y) \to \phi_{0,n}(y) \text{ as } \varepsilon \to 0. \tag{25}$$

2.7 American type options for log-price processes represented by Gaussian random walks

Let assume that the following condition holds:

E: **(a)** $W_{0,n} = \mu_n + \sigma_n B_n,\ n = 1, \ldots,\ n = 1, \ldots, N$, where $\mu_n \in \mathbb{R}$ and $\sigma_n > 0,\ n = 1, 2, \ldots$ are real numbers and $B_n,\ n = 1, 2, \ldots$ is a sequence of independent and identically distributed (i.i.d.) random variables with standard normal distribution; **(b)** $Y_0 = y_0 = const \in \mathbb{R}$ with probability 1.

In this case the process $Y_{0,n}$ can be represented in the following form,

$$Y_{0,n} = y_0 + \alpha_n + \tilde{Y}_{0,n},\ n = 0, 1, \ldots \tag{26}$$

where (a) $\alpha_n = \sum_{k=1}^{n} \mu_k, \ n = 0, 1, \ldots$ is a non-random function; (b) $\tilde{Y}_{0,n} = \sum_{k=1}^{n} \sigma_k B_k, \ n = 0, 1, \ldots$ is a centered Gaussian random walk with the initial value $\tilde{Y}_{0,0} = 0$.

It is obvious the both price processes $Y_{0,n}$ and $\tilde{Y}_{0,n}$ generate the same natural filtration $F_{0,n} = \sigma[Y_{0,n}, \ldots, Y_{0,N}] = \sigma[\tilde{Y}_{0,n}, \ldots, \tilde{Y}_{0,N}], \ n = 0, 1, \ldots$ and, therefore, have the same classes of stopping times $\mathcal{M}_{0,n,N}, 0 \leq n \leq N$.

In this case, one can use the following transformation formula for the reward functions

$$\begin{aligned} \phi_n(y) &= \sup_{\tau_n \in \mathcal{M}_{n,N}} \mathsf{E}_{y,n} g(\tau_n, Y_{0,\tau_n}) \\ &= \sup_{\tau_n \in \mathcal{M}_{n,N}} \mathsf{E}_{y,n} g(\tau_n, y_0 + \alpha_{\tau_n} + \tilde{Y}_{0,\tau_n}) \\ &= \sup_{\tau_n \in \mathcal{M}_{n,N}} \mathsf{E}_{y,n} \tilde{g}(\tau_n, \tilde{Y}_{0,\tau_n}). \end{aligned} \tag{27}$$

where $\tilde{g}(n, y)$ is a new payoff function defined for $y \in \mathbb{R}, n = 0, \ldots, N$, by the following formula,

$$\tilde{g}(n, y) = g(n, y_0 + \alpha_n + y). \tag{28}$$

Relation (27) shows that one can always reduce the approximation option problem for log-price processes represented by Gaussian random walk with non-zero initial value and non-zero trend to the the approximation option problem for log-price processes represented by Gaussian random walk with zero initial value and zero trend. This can be done using the appropriate transformations of the log-price process and the payoff function.

Thus, without loss of generality, we can assume, if necessary, that the following condition holds:

F: **(a)** $\mu_n = 0, n = 1, 2, \ldots$; **(b)** $Y_0 = 0$ with probability 1.

Under condition **F**, we can rewriten

$$W_{0,n} = \tilde{W}_{0,n} = \sigma_n B_n \tag{29}$$

We are going to investigate three alternative approximation models for log-price processes represented by Gaussian random walks.

2.8 A binomial sum approximation model

In this model, random variables $W_{\varepsilon,n},\ n = 1,\dots$ are defined for every $\varepsilon > 0$ by the following relation

$$W_{\varepsilon,n} = X_{\varepsilon,n,1} + \cdots + X_{\varepsilon,n,r_{\varepsilon,n}} \tag{30}$$

where (a) $r_{\varepsilon,n},\ n = 1,\dots$ are positive integers, and (b) $X_{\varepsilon,n,k}, k,\ n = 1,2,\dots,r_{\varepsilon,n}$ are independent binary random variables with parameters $\delta_\varepsilon > 0$ and $0 \le p_{\varepsilon,n} \le 1$, i.e.,

$$X_{\varepsilon,n,k} = \begin{cases} +\delta_\varepsilon & \text{with probability } p_{\varepsilon,n} \\ -\delta_\varepsilon & \text{with probability } 1 - p_{\varepsilon,n}. \end{cases} \tag{31}$$

Here we only assume that the jump step δ_ε is a function of ε, not depending of n. This implies that the recombination condition holds.

In order to fit parameters, we should provide the asymptotic fitting of the moments for random variables $W_{\varepsilon,n}$ and $W_{0,n}$, for every $n = 1,\dots,N$. According the remarks made in Section 2.7, we can restrict consideration by the case where condition **F** holds. In this case, fitting is provided by the following relations, which should hold for every $n = 1,\dots,N$,

$$\begin{cases} \mathsf{E}\tilde{W}_{\varepsilon,n} = r_{\varepsilon,n}(\delta_\varepsilon p_{\varepsilon,n} - \delta_\varepsilon(1 - p_{\varepsilon,n})) \to 0, \text{ as } \varepsilon \to 0, \\ Var\tilde{W}_{\varepsilon,n} = r_{\varepsilon,n}(\delta_\varepsilon^2 p_{\varepsilon,n} + \delta_\varepsilon^2(1 - p_{\varepsilon,n}) = r_{\varepsilon,n}\delta_\varepsilon^2 \to \sigma_n^2, \text{ as } \varepsilon \to 0. \end{cases} \tag{32}$$

It is readily seen that the asymptotic fitting relations (32) holds if we chose parameters δ_ε and $r_{\varepsilon,n}, n = 1,\dots,N$ in the following forms,

$$r_{\varepsilon,n} = [r_\varepsilon \sigma_n^2], \quad p_{\varepsilon,n} = \frac{1}{2}, \quad \delta_\varepsilon = \frac{1}{\sqrt{r_\varepsilon}}, \tag{33}$$

where r_ε is a positive integer, we can assume that $r_\varepsilon = \frac{1}{\varepsilon}$, such that $r_\varepsilon \to \infty$ as $\varepsilon \to 0$, and $r_{\varepsilon,n}$ is the rounded value $[r_\varepsilon \sigma_n^2]$. We can check that:

$$Var\tilde{W}_{\varepsilon,n} = r_{\varepsilon,n}(\delta_\varepsilon^2 p_{\varepsilon,n} + \delta_\varepsilon^2(1 - p_{\varepsilon,n}) = [r_\varepsilon \sigma_n^2]\frac{1}{r_\varepsilon} \to \sigma_n^2, \text{ as } r_\varepsilon \to \infty. \tag{34}$$

where $-0.5 \le \theta \le 0.5$. So now, we can simulate $\tilde{W}_{\varepsilon,n}$ as $Bin(r_{\varepsilon,n}, p_{\varepsilon,n})$, with jump step $\delta_\varepsilon = \frac{1}{\sqrt{r_\varepsilon}}$, up or down.

Let us now prove that, under conditions **E** and **F**, the choice of the above parameters according relation (33) implies that conditions $\mathbf{A}'[\beta]$ (with any parameter $\beta \geq 0$) and **C** hold.

The random variable $W_{0,n}$ has the normal distribution with parameters 0 and σ_n^2, for every $n = 1, \ldots, N$. Thus, for any $n = 1, \ldots, N$ and $\beta \geq 0$,

$$\mathsf{E}e^{\pm\beta W_{0,n}} = e^{\frac{\beta^2\sigma_n^2}{2}} < \infty. \tag{35}$$

Also, random variable $W_{\varepsilon,n}$ has the binomial distribution with parameters given in relation (33), for every $\varepsilon > 0$ and $n = 1, \ldots, N$. Thus, for any $n = 1, \ldots, N$ and $\beta \geq 0$, using Taylor expansion for expected function we get,

$$\begin{aligned}
\mathsf{E}e^{\pm\beta W_{\varepsilon,n}} &= \Big(\mathsf{E}e^{\pm\beta X_{\varepsilon,n,1}}\Big)^{[r_\varepsilon\sigma_n^2]} \\
&= \Big(e^{\pm\beta\frac{1}{\sqrt{r_\varepsilon}}}\frac{1}{2} + e^{\mp\beta\frac{1}{\sqrt{r_\varepsilon}}}\frac{1}{2}\Big)^{[r_\varepsilon\sigma_n^2]} \\
&= \Big(1 + \frac{\beta^2}{2r_\varepsilon} + o(\frac{1}{r_\varepsilon})\Big)^{[r_\varepsilon\sigma_n^2]} \to e^{\frac{\beta^2\sigma_n^2}{2}} < \infty, \text{as } \varepsilon \to 0.
\end{aligned} \tag{36}$$

Relation (36) implies condition $\mathbf{A}'[\beta]$ (with any parameter $\beta \geq 0$) and $W_{\varepsilon,n}$ $W_{0,n}$ have same moment generating function which lead to that condition **D** holds.

Condition $\mathbf{C}'$ also holds, so by Theorem 4, we get the the following recurrence backward relations, for every $y \in \mathbb{R}, 0 \leq n \leq N$ and $\varepsilon \in (0, \varepsilon_0]$,

$$\begin{cases}
\phi_{\varepsilon,N}(y + \delta_\varepsilon l) = g(N, y + \delta_\varepsilon l), \ l = -r_{\varepsilon,n,N}, \ldots, r_{\varepsilon,n,N}, \\
\phi_{\varepsilon,m}(y + \delta_\varepsilon l) = \max\Big(g(m, y + \delta_\varepsilon l), \\
\qquad \sum_{k=0}^{r_{\varepsilon,m+1}} \phi_{\varepsilon,m+1}(y + \delta_\varepsilon l + \delta_\varepsilon\,(2k - r_{\varepsilon,m+1}))f(k; r_{\varepsilon,m+1}, p_{\varepsilon,m+1})\Big), \\
l = -r_{\varepsilon,n,m}, \ldots, r_{\varepsilon,n,m}, m = N-1, \ldots, n.
\end{cases} \tag{37}$$

where $f(k; r_{\varepsilon,m+1}, p_{\varepsilon,m+1}) = \binom{r_{\varepsilon,m+1}}{k} p_{\varepsilon,m+1}^k \,(1 - p_{\varepsilon,m+1})^{r_{\varepsilon,m+1}-k}$.

We can also apply Theorem 3 that yields the following theorem.

Theorem 6. *Let the log-price processes $Y_{\varepsilon,n}$ be constructed using the binomial approximation scheme satisfying conditions* **E** *and* **F** *with parameters given by relation* (33). *Also let condition* $\mathbf{B}[\gamma]$ *hold with some parameter*

$\gamma \geq 0$ and $r_\varepsilon \to \infty$ as $\varepsilon \to 0$. Then the following relation holds for any $y \in \mathbb{R}, n = 0, \dots, N$,

$$\phi_{\varepsilon,n}(y) \to \phi_{0,n}(y) \text{ as } \varepsilon \to 0. \tag{38}$$

2.9 A trinomial sum approximation model

In this model, random variables $W_{\varepsilon,n}, n = 1, \dots$ are defined for every $\varepsilon > 0$ by the following relation

$$W_{\varepsilon,n} = X_{\varepsilon,n,1} + \cdots + X_{\varepsilon,n,r_{\varepsilon,n}} \tag{39}$$

where (a) $r_{\varepsilon,n}, n = 1, \dots$ are positive integers, and (b) $X_{\varepsilon,n,k}, k, n = 1, 2, \dots$ are independent trinomial random variables with parameters $\delta_\varepsilon > 0$ and $0 \leq p_{\varepsilon,n} \leq 1$, i.e.,

$$X_{\varepsilon,n,k} = \begin{cases} +\delta_\varepsilon & \text{with probability } \frac{1-p_{\varepsilon,n}}{2} \\ 0 & \text{with probability } p_{\varepsilon,n} \\ -\delta_\varepsilon & \text{with probability } \frac{1-p_{\varepsilon,n}}{2}. \end{cases} \tag{40}$$

So $W_{\varepsilon,n} \in [-r_{\varepsilon,n}\delta_\varepsilon, r_{\varepsilon,n}\delta_\varepsilon]$.

According the remarks made in Section 2.7, we can restrict consideration by the case where condition **F** holds. In order to fit parameters, we should provide the asymptotic fitting of the moments for random variables $\tilde{W}_{\varepsilon,n}$, some we did with binomial sum model, for every $n = 1, \dots, N$. This fitting is provided by the following relations, which should hold for every $n = 1, \dots, N$,

$$\begin{cases} \mathsf{E}\tilde{W}_{\varepsilon,n} = r_{\varepsilon,n}(\delta_\varepsilon \frac{1-p_{\varepsilon,n}}{2} + 0 - \delta_\varepsilon \frac{1-p_{\varepsilon,n}}{2}) = 0 \text{ for every } \varepsilon \\ Var\tilde{W}_{\varepsilon,n} = r_{\varepsilon,n}\delta_\varepsilon^2(1 - p_{\varepsilon,n}) \to \sigma_n^2 \text{ as } \varepsilon \to 0. \end{cases} \tag{41}$$

Let us take $\delta_\varepsilon = \frac{1}{\sqrt{r_\varepsilon}}$, where r_ε are positive integers such that $r_\varepsilon \to \infty$ as $\varepsilon \to 0$, same as in Binomial model. So parameters can take values as

$$r_{\varepsilon,n} = [\frac{r_\varepsilon \sigma_n^2}{1 - p_{\varepsilon,n}}], \quad 0 < p_{\varepsilon,n} < 1, \quad \delta_\varepsilon = \frac{1}{\sqrt{r_\varepsilon}} \tag{42}$$

Let us now prove that, under conditions **E** and **F**, the choice of the above parameters according relation (42) implies that conditions $\mathbf{A}'[\beta]$ (with any parameter $\beta \geq 0$) and **D** hold.

Random variable $\tilde{W}_{\varepsilon,n}$ here has the trinomial-sum distribution with parameters given in relation (42), for every $\varepsilon > 0$ and $n = 1, \ldots, N$. Thus, for any $n = 1, \ldots, N$ and $\beta \geq 0$,

$$\begin{aligned}
\mathsf{E} e^{\pm\beta\tilde{W}_{\varepsilon,n}} &= \left(\mathsf{E} e^{\pm\beta X_{\varepsilon,n,1}}\right)^{[\frac{r_\varepsilon \sigma_n^2}{1-p_{\varepsilon,n}}]} \\
&= \left(e^{\pm\beta\frac{1}{\sqrt{r_\varepsilon}}}\frac{1-p_{\varepsilon,n}}{2} + p_{\varepsilon,n} + e^{\mp\beta\frac{1}{\sqrt{r_\varepsilon}}}\frac{1-p_{\varepsilon,n}}{2}\right)^{[\frac{r_\varepsilon \sigma_n^2}{1-p_{\varepsilon,n}}]} \\
&= \left(1 + \frac{(1-p_{\varepsilon,n})\beta^2}{2r_\varepsilon} + o\left(\frac{1}{r_\varepsilon}\right)\right)^{[\frac{r_\varepsilon \sigma_n^2}{1-p_{\varepsilon,n}}]} \\
&\to e^{\frac{\beta^2\sigma_n^2}{2}} < \infty \text{ as } \varepsilon \to \infty. \qquad (43)
\end{aligned}$$

Relation (43) implies both conditions $\mathbf{A}'[\beta]$ (with any parameter $\beta \geq 0$) and $\mathbf{D}$ to hold. Condition $\mathbf{C}'$ also holds.

The recurrence backward relations in this case, by Theorem 4, for every $y \in \mathbb{R}, 0 \leq n \leq N$ and $\varepsilon \in (0, \varepsilon_0]$ is following:

$$\begin{cases}
\phi_{\varepsilon,N}(y + \delta_\varepsilon l) = g(N, y + \delta_\varepsilon l), \ l = -r_{\varepsilon,n,N}, \ldots, r_{\varepsilon,n,N}, \\
\phi_{\varepsilon,m}(y + \delta_\varepsilon l) = \max\Big(g(m, y + \delta_\varepsilon l), \\
\qquad \sum_{k=-r_{\varepsilon,m+1}}^{r_{\varepsilon,m+1}} \phi_{\varepsilon,m+1}(y + \delta_\varepsilon l + \delta_\varepsilon k)\mathsf{P}\{W_{\varepsilon,m+1} = \delta_\varepsilon k\}\Big), \\
l = -r_{\varepsilon,n,m}, \ldots, r_{\varepsilon,n,m}, m = N-1, \ldots, n.
\end{cases} \qquad (44)$$

where

$$P\{W_{\varepsilon,m+1} = \delta_\varepsilon k\} = \sum_{k_+,k_-} \frac{r_{\varepsilon,m+1}!}{k_+!k_-!\,(r_{\varepsilon,m+1} - k_+ - k_-)!} p_+^{k_+} p_-^{k_-} p_0^{r_{\varepsilon,m+1}-k_+-k_-} \qquad (45)$$

and k_+, k_- fulfill conditions

$$\begin{cases}
k_+, k_- \geq 0 \\
k_+ + k_- \leq r_{\varepsilon,m+1} \\
k_+ + k_- = k
\end{cases} \qquad (46)$$

at the same time $p_+ = p_- = \frac{1-p_{\varepsilon,n}}{2}, \quad p_0 = p_{\varepsilon,n}$.

We can also apply Theorem 3 that yields the following theorem.

Theorem 7. *Let the log-price processes $Y_{\varepsilon,n}$ be constructed using the trinomial approximation scheme satisfying conditions* **E** *and* **F** *with parameters given by relation (42). Also let condition* $\mathbf{B}[\gamma]$ *hold with some parameter* $\gamma \geq 0$ *and* $r_\varepsilon \to \infty$ *as* $\varepsilon \to 0$. *Then the following relation holds for any* $y \in \mathbb{R}, n = 0, \dots, N$,

$$\phi_{\varepsilon,n}(y) \to \phi_{0,n}(y) \text{ as } \varepsilon \to 0. \tag{47}$$

2.10 A skeleton approximation model

In this model, random variables $W_{\varepsilon,n}, n = 1, \dots$ are defined for every $\varepsilon > 0$ by the following relation

$$W_{\varepsilon,n} = \begin{cases} +r_{\varepsilon,n}\delta_\varepsilon & \text{with probability } 1 - \Phi(\frac{(r_{\varepsilon,n}-\frac{1}{2})\delta_\varepsilon}{\sigma_n}) \\ l\delta_\varepsilon & \text{with probability } \Phi(\frac{(l+\frac{1}{2})\delta_\varepsilon}{\sigma_n}) - \Phi(\frac{(l-\frac{1}{2})\delta_\varepsilon}{\sigma_n}) \\ -r_{\varepsilon,n}\delta_\varepsilon & \text{with probability } \Phi(\frac{(-r_{\varepsilon,n}+\frac{1}{2})\delta_\varepsilon}{\sigma_n}). \end{cases} \tag{48}$$

where (a) $r_{\varepsilon,n}, n = 1, \dots$ are positive integers, (b) $-r_{\varepsilon,n} < l < r_{\varepsilon,n}$, and (c) $\Phi(x)$ is the standard normal culmaltive distribution function. We can set

$$r_{\varepsilon,n} = [r_\varepsilon \sigma_n^2], \quad \delta_\varepsilon = \frac{1}{\sqrt{r_\varepsilon}}, \tag{49}$$

where r_ε are positive integers such that $r_\varepsilon \to \infty$ as $\varepsilon \to 0$. And we also have $r_{\varepsilon,n}\delta_\varepsilon \sim \sigma_n^2\sqrt{r_\varepsilon} \to \infty$, as $\varepsilon \to 0$.

Let us now prove that, under conditions **E** and **F**, the choice of the above parameters according relation (49) implies that conditions $\mathbf{A}'[\beta]$ (with any parameter $\beta \geq 0$) and **D** hold.

The method of proof below is learned from Silvestrov (2012).

As the formula (48), we can define, for ε and $n = 0, 1, \dots,$ functions as

$$\tilde{h}_{\varepsilon,n}(y) = \begin{cases} +r_{\varepsilon,n}\delta_\varepsilon & \text{if } y \geq (r_{\varepsilon,n} - \frac{1}{2})\delta_\varepsilon \\ l\delta_\varepsilon & \text{if } (-r_{\varepsilon,n} + \frac{1}{2})\delta_\varepsilon \leq y \leq (r_{\varepsilon,n} - \frac{1}{2})\delta_\varepsilon \\ -r_{\varepsilon,n}\delta_\varepsilon & \text{if } y \leq (-r_{\varepsilon,n} + \frac{1}{2})\delta_\varepsilon \end{cases} \tag{50}$$

where (a) $r_{\varepsilon,n}, n = 1, \dots$ are positive integers, (b) $-r_{\varepsilon,n} < l < r_{\varepsilon,n}$. So the relation between $W_{\varepsilon,n}$ and $W_{0,n}$ can be written as:

$$W_{\varepsilon,n} = \tilde{h}_{\varepsilon,n}(W_{0,n}) \tag{51}$$

We can also get the following inequality for every ε and $y \in \mathbb{R}$,

$$\left|\tilde{h}_{\varepsilon,n}(y)\right| \leq |y|\mathbf{I}(y \notin A) + (|y| + \delta_\varepsilon)\mathbf{I}(y \in A) \leq |y| + \delta_\varepsilon \tag{52}$$

where interval $A = [(-r_{\varepsilon,n} + \frac{1}{2})\delta_\varepsilon,\ (r_{\varepsilon,n} - \frac{1}{2})\delta_\varepsilon]$.

Under condition $\mathbf{F}$, $W_{0,n} \sim N(0, \sigma_n^2)$ for every $n = 1, \dots, N$, thus for $\beta \leq 0$,

$$\mathsf{E}e^{\pm\beta\tilde{W}_{\varepsilon,n}} = \mathsf{E}e^{\pm\beta\tilde{h}_{\varepsilon,n}(W_{0,n})} \leq e^{\beta\delta_\varepsilon}\mathsf{E}e^{\pm\beta W_{0,n}} = e^{\beta\delta_\varepsilon}e^{\frac{\beta^2\sigma_n^2}{2}} < \infty \tag{53}$$

So condition $\mathbf{A}'[\beta]$ (with any parameter $\beta \geq 0$) is fulfilled. For condition $\mathbf{D}$, we have

$$\begin{aligned}|\tilde{h}_{\varepsilon,n}(y) - y| &\leq ((-r_{\varepsilon,n} + \frac{1}{2})\delta_\varepsilon - y)\mathbf{I}(y \leq (-r_{\varepsilon,n} + \frac{1}{2})\delta_\varepsilon) \\ &+ \delta_\varepsilon\mathbf{I}(y \in A) + (y - (r_{\varepsilon,n} + \frac{1}{2})\delta_\varepsilon)\mathbf{I}((-r_{\varepsilon,n} + \frac{1}{2})\delta_\varepsilon \leq y)\end{aligned} \tag{54}$$

By setting (49), we have $r_{\varepsilon,n}\delta_\varepsilon \sim \sigma_n^2\sqrt{r_\varepsilon} \to \infty$ and $\delta_\varepsilon \to 0$, as $\varepsilon \to 0$, hence relation (54) implies, for every $y \in \mathbb{R}$, and $n = 0, 1, 2 \dots N$,

$$|\tilde{h}_{\varepsilon,n}(y) - y| \to 0, \text{as } \varepsilon \to 0 \tag{55}$$

So we get the almost sure convergence:

$$\tilde{h}_{\varepsilon,n}(y) \overset{a.s.}{\to} y,\ \text{as } \varepsilon \to 0 \Longrightarrow \tilde{W}_{\varepsilon,n} = \tilde{h}_{\varepsilon,n}(W_{0,n}) \overset{a.s.}{\to} W_{0,n},\ \text{as } \varepsilon \to 0 \tag{56}$$

Relation (56) implies that condition of convergence in distribution $\mathbf{D}$ for this model holds.

Condition $\mathbf{C}'$ also holds. The recurrence backward relations in this case, by Theorem 4, for every $y \in \mathbb{R}, 0 \leq n \leq N$ and $\varepsilon \in (0, \varepsilon_0]$ is following:

$$\begin{cases}\phi_{\varepsilon,N}(y + \delta_\varepsilon l) = g(N, y + \delta_\varepsilon l),\ l = -r_{\varepsilon,n,N}, \dots, r_{\varepsilon,n,N}, \\ \phi_{\varepsilon,m}(y + \delta_\varepsilon l) = \max\Big(g(m, y + \delta_\varepsilon l), \\ \qquad \sum_{k=-r_{\varepsilon,m+1}}^{r_{\varepsilon,m+1}} \phi_{\varepsilon,m+1}(y + \delta_\varepsilon l + \delta_\varepsilon k)\mathsf{P}\{W_{\varepsilon,m+1} = \delta_\varepsilon k\}\Big), \\ l = -r_{\varepsilon,n,m}, \dots, r_{\varepsilon,n,m}, m = N-1, \dots, n.\end{cases} \tag{57}$$

where $P\{W_{\varepsilon,m+1} = \delta_\varepsilon k\}$ is discribted by relations (48).

We can also apply Theorem 3 that yields the following theorem.

Theorem 8. *Let the log-price processes $Y_{\varepsilon,n}$ be constructed by using the skeleton approximation scheme satisfying conditions* **E** *and* **F** *with parameters given by relation* (49). *Also let condition* $\mathbf{B}[\gamma]$ *hold with some parameter* $\gamma \geq 0$ *and* $r_\varepsilon \to \infty$ *as* $\varepsilon \to 0$. *Then the following relation holds for any* $y \in \mathbb{R}, n = 0, \dots, N$,

$$\phi_{\varepsilon,n}(y) \to \phi_{0,n}(y) \text{ as } \varepsilon \to 0. \tag{58}$$

3 Rate of Convergence for Approximations of American Type Options

Based on the standard binomial tree method, we can define a changing unit for log-price $Y_n, 0 \leq n \leq N$, as shown in Figure 3.

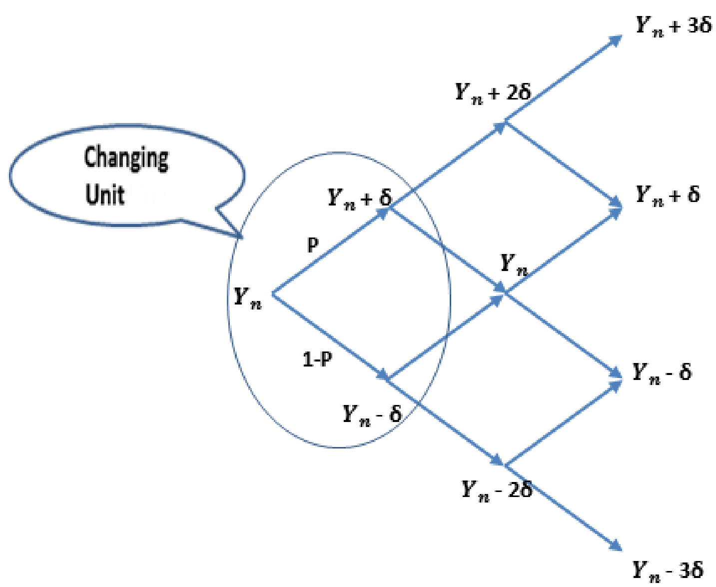

Figure 3: Log price developments of std. binomial tree model

The changing unit shows every possible results of $Y_n, 0 \leq n \leq N$, after just one time step Δt, $\Delta t = T/N$. For example, in the case of the standard binomial tree model, we can see that Y_n will change either to $Y_n + \delta$ with probability p or to $Y_n - \delta$ with probability $1 - p$.

In the present paper, the changing unit is assumed to be according to the binomial-sum, trinomial-sum and skeleton model respectively. More possible changing results after one time interval (Δt) are concerned, not only two as in the case of standard binomial model. The number of nodes, $r_{\varepsilon,n}$, is a function of r_ε. If needed, $r_{\varepsilon,n}$ can be very large. So the normal distribution assumption for $W_{0,n}$ is catched better in our models.

In this section, we will first test the rate of convergence of a call option in these three models with the following parameters, which are denoted as standard conditions. We assume, a risk-free interest rate $r = 5\%$, a initial stock price $S_0 = 10$, and a stock price process, defined by formula (6), having a drift $\mu = 0.25\%$, and a yearly volatility $\sigma = 30\%$. In our experimental studies, we assume:

$$\mu = \frac{r - \sigma^2/2}{2}$$

instead of $\mu = r - \sigma^2/2$ in risk neurual situation.

We assume also that the strike price $K = 10$ and the days to maturity are 90 days, or $T = 90/365 \approx 0.24657$. At the end, we set $N = 12$ in the standard conditions, so one time interval is almost equal to one week, then we set $N = 100$, $N = 1000$ which makes the time interval become almost one day, and one hour respectively.

Recall that in formulas (33), (42), and (49) for these three models, under the standard conditions, r_ε is the only changing parameter, and we will investigate the development of the value of the reward function $\phi_{\varepsilon,n}(y)$, or in other words the option price, as a function of $r_\varepsilon = \frac{1}{\varepsilon}$, and $r_\varepsilon \to \infty$ as $\varepsilon \to 0$.

All the simulations in this thesis are performed in double precision in MATLAB R2007B, on a laptop with 1.3 GHZ Intel Mobile Core 2 Duo SU7300 CPU and 4 GB of internal memory, and the operation system is Windows 7 Home Premium 64 bit.

We will also emphasize the importance of programming in MATLAB. For example, have the code utilized the advantages which the MATLAB has in terms of vectors and matrix calculations? Such factors have significant effect on execution speed.

Higham (2002) presents nine ways to implement the standard binomial method for pricing European option in MATLAB. He showed that, by working on vectors instead of 'for' loops, the execution time was dramatically improved. It was reduced more than 50 times compared to the initial version which is based on 'for' loops. And further more, by little extra rewriting of the binomial probability function, he showed that the execution time was reduced another ten times.

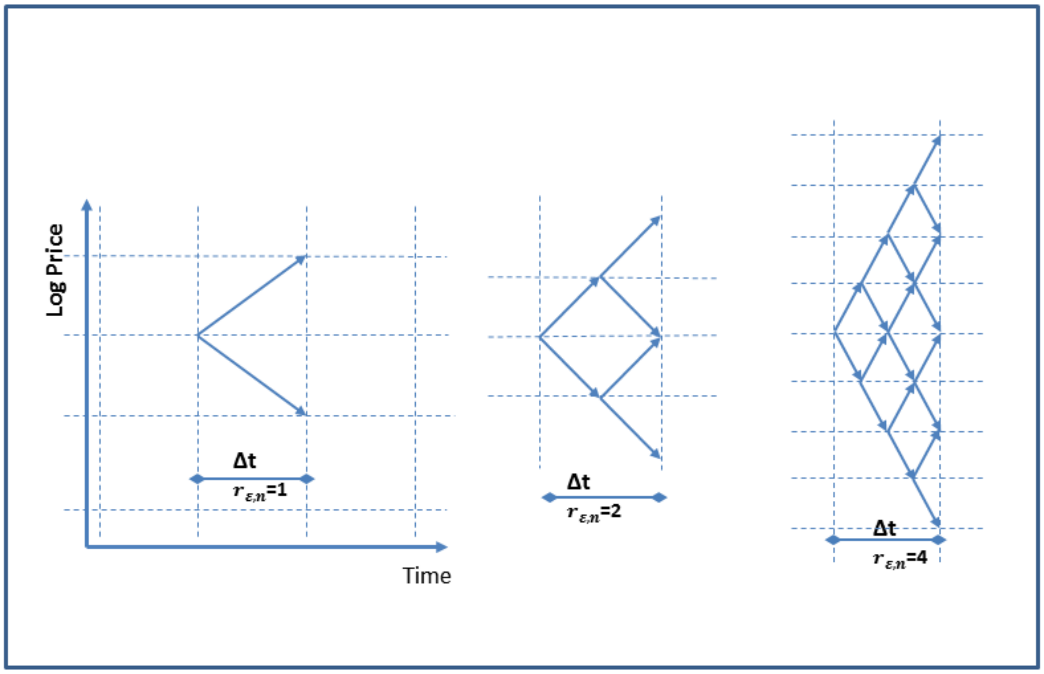

Figure 4: Different changing units in binomial sum model

3.1 Rate of convergence for the binomial sum approximation model

Recall that in this model, the changing unit at time n^{th} is a binomial distribution $Bin(r_{\varepsilon,n}, p_{\varepsilon,n})$, and its parameters are assume as:

$$r_{\varepsilon,n} = [r_\varepsilon \sigma_n^2], \; p_{\varepsilon,n} = \frac{1}{2}, \; \delta_\varepsilon = \frac{1}{\sqrt{r_\varepsilon}}, \tag{59}$$

Some examples for this model are shown in Figure 4, with different number of possible jumps, $r_{\varepsilon,n}$, and different jump magnitude, δ_ε, under one time interval. Both $r_{\varepsilon,n}$ and δ_ε are depended on parameter r_ε and volatility σ_n, as shown in formula (59).

Note that in the standard binomial model, the options can be exercised only at these time moment points, but between the time moments, no operations can be made. The same dicipline is applied here too. So the points shown between two time moments, are just used to illustrate how we choose out the possible results of log price and how to calculate the corresponding probabilities after one time interval, Δt.

We can see that, when the recombination condition is fulfilled, after one time step, we say at time point $n+1$ for $Y_n = y$, there will be $r_{\varepsilon,n}+1$ possible values for Y_{n+1}. The total number of price points, at maturity, N, will be $1 + \sum_{i=1}^{N} r_{\varepsilon,i}$.

Be aware of that, if by choosing some special value on r_ε we can get $r_{\varepsilon,n} = 1$, for $n = 1, 2, \ldots, N$. In such a situation, the price development tree structure in this model will be the same as in standard Binomial model, which is shown in Figure 3. Structures are same, but pay attention that they have different parameter vaules, namely $\delta_\varepsilon = \frac{1}{\sqrt{r_\varepsilon}}$ here, but $u = 1/d = \delta = e^{\sigma\sqrt{\delta t}}$ in standard binomial model.

In order to compare the rate of convergence, the limit value is needed. As we showed above, when $r_\varepsilon \to \infty$, the option price will converge to a certain value. In practice, we only need a value, a benchmark value, which is closed to this limit value. How close it should be is defined according to the need in reality.

The method, we use, to find this benchmark value is following. We know that larger values of r_ε leads us to this value. So we can set large values on r_ε, with big increament and study the their difference in percentage. If the value of percentage fulfill our requirment, we will take the last option value, which is calculated from the largest r_ε we used, as our benchmark.

Note that for some cases, such as N=1000, to find the benchmark can be very time-consuming. But we don't need to worry about that, because such values can be prepared in advance.

Option Value	r_ε	$r_{\varepsilon,n}$	Diff. (% of option value)
0.649606557847695	210000	388	0
0.649647576161140	230000	425	0.0063
0.649681970751150	250000	462	0.0053

Table 1: Benchmark, N=12 in binomial sum model.

In Table 1, it shows a example of how benchmark can be choosed. We can see that the change scope of the option price becomes smaller and smaller with larger r_ε. For the last value of r_ε, the difference is less than 0.006%. In the later, we will only consider 5% and 1% precision. So the accuracy on the benchmark value is enough.

We will use the same method to find the benchmark values for cases of $N = 100$ and $N = 1000$.

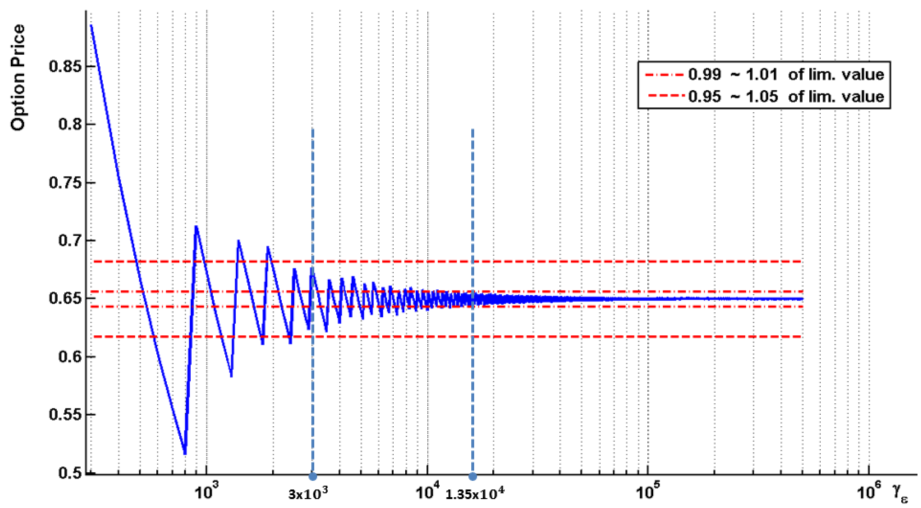

Figure 5: Convergence of the call-option price in binomial sum model

Figure 5 shows that how the option price changes with a increasing r_ε. Y-axle shows the call-option price, and on X-axle the value of r_ε is shown in log-scale. We can see that when r_ε is bigger than 3×10^3 roughly, our simulated option price changes only between $\pm 5\%$ of the limit value, when r_ε passes 1.6×10^4 roughly, the changes is within $\pm 1\%$, and finally when r_ε moves toward 10^6, the option price is almost converging to a certain value, which is also proved by Theorem 6 for this model.

So some very interesting questions needed to be answered here are, when the option value will fall into the $\pm 5\%$ precision interval of its "true value", how about $\pm 1\%$ precision. In such cases, and if we will have a $\pm 1\%$ precision of option value, what is the corresponding calculation time of it?

In order to answer those questions, Table 2 and Table 3 are build under the standard conditions.

	N= 12	N= 100	N = 1000
Lim. Value	0.649682	0.649951	0,65002
r_ε	3000	2.6×10^4	2.2×10^5
$r_{\varepsilon,n} = [r_\varepsilon \sigma_n^2]$	6	6	5
Times (Sec.)	0.078	0.125	1.08

Table 2: Rate of convergence in binomial sum model, 5% precision

	N= 12	N= 100	N = 1000
Lim. Value	0.649682	0.649951	0,65002
r_ε	1.35×10^4	1.2×10^5	1.16×10^6
$r_{\varepsilon,n} = [r_\varepsilon \sigma_n^2]$	25	27	26
Times (Sec.)	0.094	0.203	7.05

Table 3: Rate of convergence in binomial sum model, 1% precision

For example, the Table 2 focuses on $\pm 5\%$ precision for different N. We can read, from the first column, that the limit value of a call option, with $N = 12$, is 0.649681970751150, same as shown above. One limit value for r_ε is 3000, which means when r_ε is bigger than this value, the deviation of option value from the "true value" is less than $\pm 5\%$. This development is also illustrated in Figure 5.

To provide a more detailed picture of the simulation of log price, related to this r_ε, i.e. $r_\varepsilon = 3000$, we also figure out how many number of nodes in every changing union, as we defined in Figure 4. The answer is 5.

So the whole picture for this simulation will be a triangle matrix with 11 columns. The first column consists of only one point, which is related to Y_0 certainly. The second column is composed of six points, which are:

$$Y_0 + 5\delta_\varepsilon,\ Y_0 + 3\delta_\varepsilon,\ Y_0 + 1\delta_\varepsilon, \ldots,\ Y_0 - 5\delta_\varepsilon$$

In the same manner, the last column contains of $11 \times 5 + 1 = 56$ points, which are

$$Y_0 + 55\delta_\varepsilon,\ Y_0 + 53\delta_\varepsilon, \ldots,\ Y_0 - 55\delta_\varepsilon.$$

Now going back to Table 2 and Table 3, the last row of them present how many seconds the MATLAB programme needs to calculate the corresponding option price. We are not very interested in the exact calculation time but just to get an idea of times' quantity.

From these two tables, we can see that the convergency processes for different values of N are similar, because $r_{\varepsilon,n}$ has almost same value for differnt N, in both 5% and 1% percent cases. It will be very useful to find the right r_ε when we try to reach the wished precision with different N.

3.2 Rate of convergence for the trinomial sum approximation model

Recall in this model, the changing unit for n^{th} becomes a trinomial-sum distribution $Tri(r_{\varepsilon,n}, p_{\varepsilon,n})$, with parameters:

$$r_{\varepsilon,n} = [\frac{r_\varepsilon \sigma_n^2}{1-p_{\varepsilon,n}}], \quad 0 \leq p_{\varepsilon,n} < 1, \quad \delta_\varepsilon = \frac{1}{\sqrt{r_\varepsilon}} \tag{60}$$

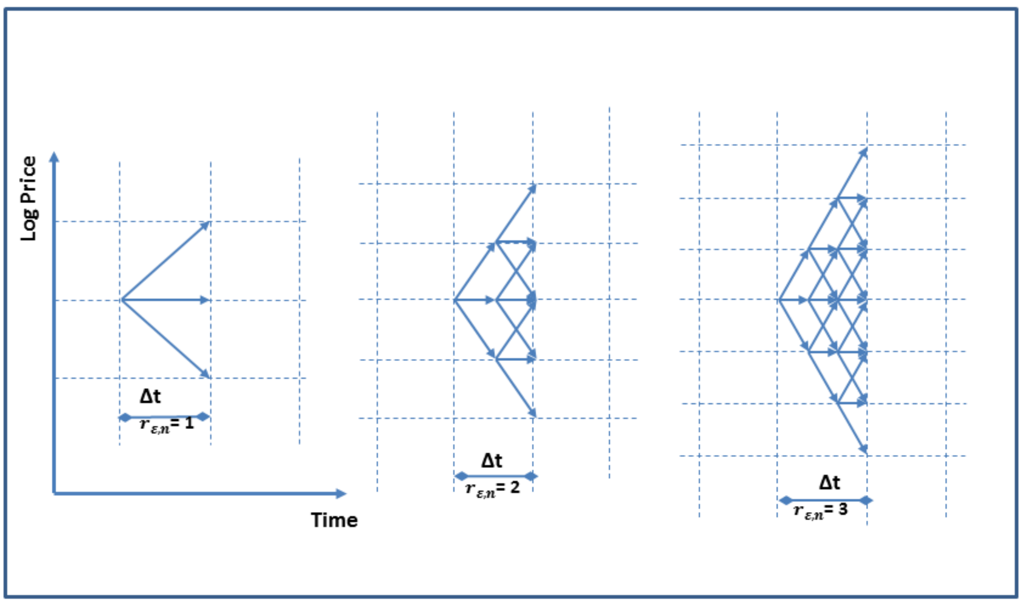

Figure 6: Changing units in trinomial sum model

Some examples of changing units are shown in Figure 6. Here, when the recombination condition is fulfilled, after one time step, say at time point $n+1$, there will be $2r_{\varepsilon,n}+1$ possible values for $Y_n = y$. The total numbers at maturity N is $1 + \sum_{i=1}^{N} 2 \times r_{\varepsilon,i}$. So it is clear that, in this model more values should be calculated, which leads to more calculation time.

In formula (60), it shows that we get some free hand to choose $p_{\varepsilon,n}$, defined as the probability of no changing. So the question is, which value shall we choose?

$p_{\varepsilon,n}$	1/10	1/3	2/3	4/5	9/10	99/100
r_ε	12500	9600	4500	2800	2000	580
$r_{\varepsilon,n} = [\frac{r_\varepsilon \sigma_n^2}{1-p_{\varepsilon,n}}]$	25	27	25	26	32	107

Table 4: Rate of convergence in trinomial sum model, 5% precision.

In Table 4, we set different values for $p_{\varepsilon,n}$, and show at which values of r_ε and corresponding number of nodes $r_{\varepsilon,n}$, the option price will reach $\pm 1\%$ precision. It seems that up to $p_{\varepsilon,n} = 0.8$, there are no any noticeable differences for this model. Just when $p_{\varepsilon,n}$ increases to 0.9 and beyond, the efficiency of this model decreases (the more the number of nodes, the more calculation time is needed). As there is still no a clear choice for $p_{\varepsilon,n}$, we choose $p_{\varepsilon,n} = 2/3$.

The calculations of $\mathsf{P}(W_{\varepsilon,m+1} = k\delta_\varepsilon)$ in this model is shown in formula (45), and it can be rewrited as a sum of products of two binomial probability functions. One example from the combination in formula (45) can be

$$
\begin{aligned}
&Prob(X_+ = k_1, X_- = k_2, X_0 = n - k_1 - k_2) = \\
&\quad = \frac{n!}{k_1!k_2!\,(n-k_1-k_2)!} a^{k_1} b^{k_2} c^{n-k_1-k_2} \\
&\quad = \frac{n!}{k_1!\,(n-k_1)!} a^{k_1} b^{n-k_1} \cdot b^{k_2-n+k_1} \frac{(n-k_1)!}{k_2!\,(n-k_1-k_2)!} c^{n-k_1-k_2} \\
&\quad = \binom{n}{k_1} a^{k_1} b^{n-k_1} \cdot \frac{(n-k_1)!}{k_2!\,(n-k_1-k_2)!} b^{k_2} c^{n-k_1-k_2} b^{-n+k_1} \\
&\quad = \binom{n}{k_1} a^{k_1} b^{n-k_1} \cdot \binom{n-k_1}{k_2} b^{k_2} c^{n-k_1-k_2} \cdot b^{-n+k_1} \qquad (61)
\end{aligned}
$$

where

$$
k_1 - k_2 = k, \quad X_i = \begin{cases} 1, \text{with probability a} \\ -1, \text{with probability b} \\ 0, \text{with probability c} \end{cases} \quad i = 1, 2 \ldots n, \text{ and } \quad a+b+c = 1
$$

and X_+ is the number of $X_i = 1$, X_- the number of $X_i = -1$, X_0 the number of $X_i = 0$. Last $X_+ + X_- + X_0 = n$.

Notice that this is just one of many combinations in formula (45). So it is quite sure that the trinomial sum model should have higher calculation-time requirement, if we actually program this model in this way, i.e. to calculate the sum of products of two binomial probability functions.

Here is our solution of how to simplify the calculation. Let's first go back to Figure 6, and take a look the picture of $r_{\varepsilon,n} = 2$. We can see that there are three points in the middle of Δt. With probabilities $(P_+ \ P_0 \ P_-)'$, log price Y_n will reach those three different points. Every one of them in their turn have the same opportunity to go up, stay or to go down. We will utilize this feature to calculate the probabilities of which price the stock price will be after Δt. Our solution for $r_{\varepsilon,n} = 2$ is:

$$\begin{pmatrix} P_{++} \\ P_{+0} \\ P_{00} \\ P_{-0} \\ P_{--} \end{pmatrix} = P_+ \cdot \begin{pmatrix} P_+ \\ P_0 \\ P_- \\ 0 \\ 0 \end{pmatrix} + P_0 \cdot \begin{pmatrix} 0 \\ P_+ \\ P_0 \\ P_- \\ 0 \end{pmatrix} + P_- \cdot \begin{pmatrix} 0 \\ 0 \\ P_+ \\ P_0 \\ P_- \end{pmatrix} \qquad (62)$$

Here, the left side of formula (62) is the respective probabilities to reach those 5 different points after Δt in the picture of $r_{\varepsilon,n} = 2$. It shows that based on the information one step before, namely $(P_+ \ P_0 \ P_-)'$, we can figure out the followed probabilities very simple. We just need to build three vectors with suitable length, times related probability and add them up. If we continue in this way, the probabilities for $r_{\varepsilon,n} = 3, 4, \ldots$ can also be calculated easily and quickly. It is a general solution and working for every value of $r_{\varepsilon,n}$. We can call such algorithm as a forward calculation and such vector-calculation match MATLAB just great and it is very efficient. The related MATLAB code is shown in Appendix B, ProbMatrixTrioFuc.

In Figure 7, we can see how the option price changes as a function of r_ε. Careful readers will wonder why the option-price follows different patterns as r_ε increasing beyond 10^3. The explanation is different increments of choosen r_ε values. In order to illustrate the convergence, we use different increment steps for r_ε. So when r_ε is less the one thousand, the changing step is just one hundred, beyond that we just increase it to one thousand.

In this model, under the standard conditions described above, by the same principle of choosing the limit-value of option price in binomial sum model, the limit value we find here is $0,649931361$, no noticeable difference

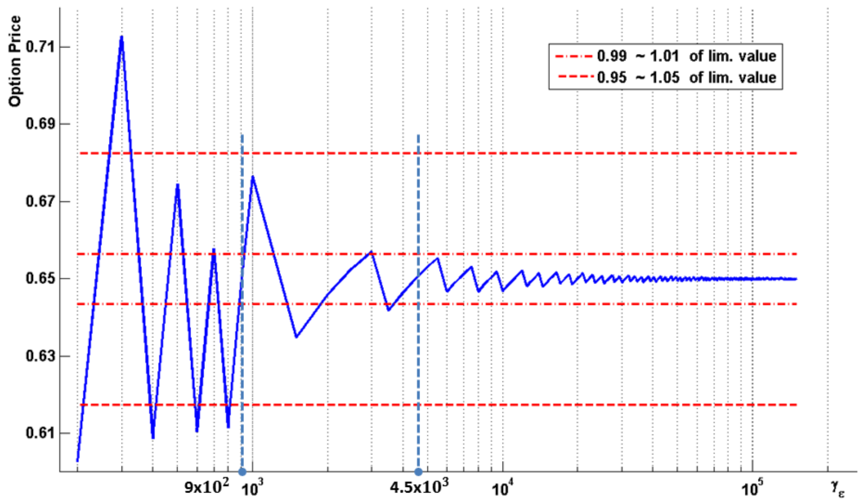

Figure 7: Convergence of the call-option price in trinomial sum model

from the value in binomial sum model.

Two similar tables, Table 5 and Table 6, are made up as we did with the binomial sum model.

	N= 12	N= 100	N = 1000
Lim. Value	0.649950	0.649943676	0.6500834
r_ε	900	7×10^3	7×10^4
$r_{\varepsilon,n} = [\frac{r_\varepsilon \sigma_n^2}{1-p_{\varepsilon,n}}]$	5	5	5
Time Sec.	0.062	0.078	1.778

Table 5: Rate of convergence in trinomial sum model, 5% precision

	N= 12	N= 100	N = 1000
Lim. Value	0.649950	0.649943676	0.6500834
r_ε	4.5×10^3	4.2×10^4	3×10^5
$r_{\varepsilon,n} = [\frac{r_\varepsilon \sigma_n^2}{1-p_{\varepsilon,n}}]$	26	28	20
Time Sec.	0.0624	0.374	13.822

Table 6: Rate of convergence in trinomial sum model, 1% precision

3.3 Rate of convergence for the skeleton approximation model

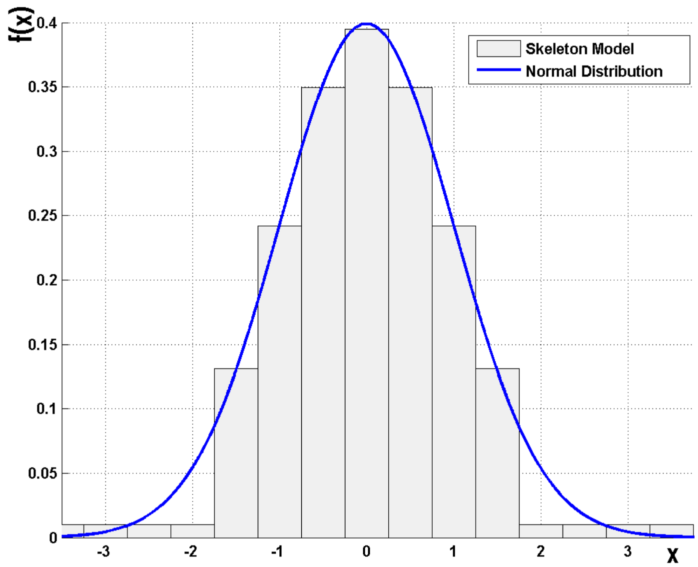

Figure 8: Skeleton Simulation

In this model, parameters are assumed as

$$r_{\varepsilon,n} = [r_\varepsilon \sigma_n^2], \quad \delta_\varepsilon = \frac{1}{\sqrt{r_\varepsilon}}, \tag{63}$$

for $n = 1, 2, 3 \ldots,$ where r_ε are positive, such that $r_\varepsilon \to \infty$ as $\varepsilon \to 0$.

In Figure 8, a case with $r_\varepsilon = 4$ is shown to illustrate how the skeleton model simulates the standard normal $N(0,1)$. In this case, we have:

$$r_{\varepsilon,n} = [r_\varepsilon \sigma_n^2] = [4 \times 1^2] = 4, \text{and } \delta_\varepsilon = \frac{1}{\sqrt{r_\varepsilon}} = \frac{1}{2}. \tag{64}$$

$W_{\varepsilon,n+1}$ can then take values of $\{-2, -1.5, -1, -0.5, 0, 0.5, 1, 1.5, 2\}$. By Skeleton model, the x-axle is divided, into intervals as $(-\infty, -1.75]$ $(-1.75 - 1.25]$,..., $(1.75, \infty)$, as suggested by the formula (50) . For example, if $W_{\varepsilon,n}$ is equal to -2, and the probability of it is

$$P\{W_{\varepsilon,n+1} = -2\} = \Phi(-1.75) = 1 - \Phi(1.75) = 0.0401$$

Figure 9 shows that how the option price changes as a function of r_ε. It is clear that the convergence of option price in this model is very smooth and quick in comparing with the other models.

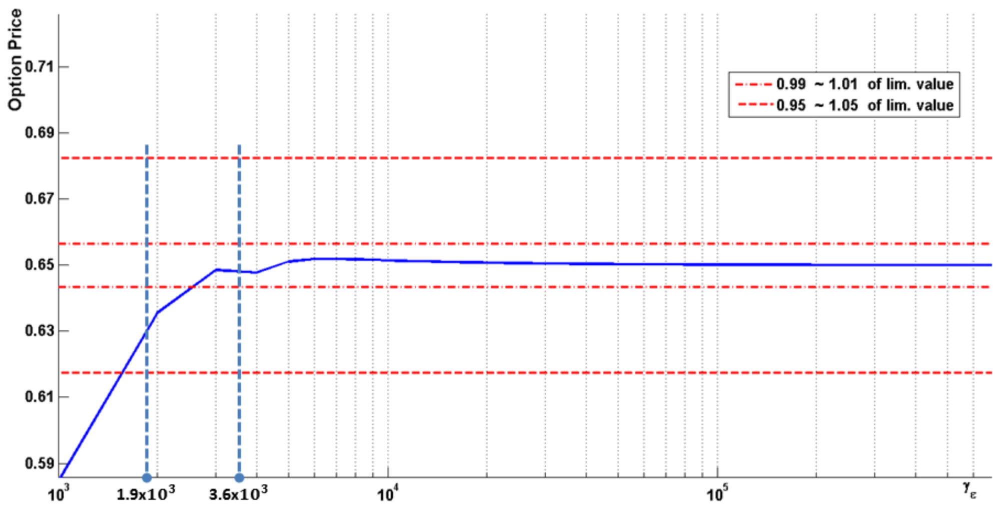

Figure 9: Convergence of the call-option price in skeleton model

To compare the calculation time, and the convergence with different values of N in this model, the same style tables are also built, namely Table 7 and Table 8.

	N= 12	N= 100	N = 1000
Lim. Value	0.64997178	0.64996949	0.6500674
r_ε	1900	1.6×10^4	1.7×10^5
$r_{\varepsilon,n} = [r_\varepsilon \sigma_n^2]$	4	4	4
Time Sec.	0.0468	0.1404	1.92

Table 7: Rate of convergence in skeleton model, 5% precision

	N= 12	N= 100	N = 1000
Lim. Value	0.64997178	0.64996949	0.6500674
r_ε	3600	3×10^4	3×10^5
$r_{\varepsilon,n} = [r_\varepsilon \sigma_n^2]$	7	7	7
Time Sec.	0.0624	0.2028	3.87

Table 8: Rate of convergence in skeleton model, 1% precision

3.4 Comparison of binomial sum, trinomial sum and skeleton approximation model

For the three models, the limit values we get are almost same, and it should be so! Another thing we notice is that the limit value is not affected so much by N!

Remark that the trinomial sum model with $p_{\varepsilon,n} = 0$ is actually equal to binomial sum model. So the comparison between those two model becomes easy, if we can use this feature. We can just use this specific case in trinomial sum model to compete with binomial sum model under standard conditions and play around little with N to get a more general picture!

So the test environment is set like this, under the standard conditions, r_ε takes values from (50000 : 500 : 100000), 11 different values. The calculation time will be recorded. N will take values as before, namely $12, 100, 1000$. In order to reduce error, every case will run 5 times, the calculation time the mean value of these 5 runnings. The results are shown in Table 9.

	N= 12	N= 100	N = 1000
Bino. Model	0.559	1.154	6.471
Bino. Model[1]	0.52104	0.77064	2,9328
Trino. Model	1.248	1.622	5.014

Table 9: Execution times in bino. and trino. sum models

In Table 9, we use two different methods to calculate probabilities for different jumps in binomial sum model. We begin by using the MATLAB function *binopdf*( ) to calculate probability of each jump. In second method, we use the same algorithm as in the trinomial sum model, so the comparison

[1]Same model but instead of using existing MATLAB func. binopdf(), we use the same algorithm as we did with trinomial sum model, see formula (62).

between those two models becomes more fair. The related MATLAB code is shown in Appendix B, ProbMatrixBinoFuc.

In Table 9, it is clearly shown that the binomial sum model is the winner between them, in terms of speed.

By comparing Table 2, Table 3 in binomial sum model with Table 5, Table 6 in Trinomial model, we can see that both of the two models reach the 5% and 1% precision with almost same $r_{\varepsilon,n}$. We can say that in terms of precision, the two models have the same efficency.

So in total, the binomial sum model will be the winner among them.

By comparing the convergence pictures above and in the Appendix A, of those three models, we can see that the skeleton model has less oscillation for high values of r_ε. The skeleton model is also the most efficient model among those three models. Table 7 and Table 8 show that this model needs the lowest $r_{\varepsilon,n}$ to reach the required precision. The less $r_{\varepsilon,n}$ is, the faster the calculation becomes. Clearly the skeleton model is the best of the three in terms of efficiency and speed.

In Appendix A, convergence of option prices are shown for all the three models based on very high volatility $\sigma = 300\%$ with, $N = 12$ and $N = 100$.

4 MATLAB Implementation

4.1 MATLAB application program

Based on these three models, an application programme in MATLAB has been written. The program has two main functions, the first is to calculate option price depended on the input parameters, such as initial stock price, strike price and so on, and the model one chooses. The second function is to show how the option value is changing as a function of r_ε, the parameter 'Gamma' in the Figure 10, program interface.

Here, we can input those related parameters, such as initial stock price, strike price and so on. Besides them, we can choose the value of Gamma, i.e. r_ε, and which model we will use. There is a checking box, "To Show the Development", if we want to calculate just one option price related to a specific Gamma, we can left the checking box blank!

By clicking the "Calculate" button, the result will be shown in the left corner of this grafic interface, namely the "Result" box. The corresponding number of nodes and the option value based on Black-Scholes formula are

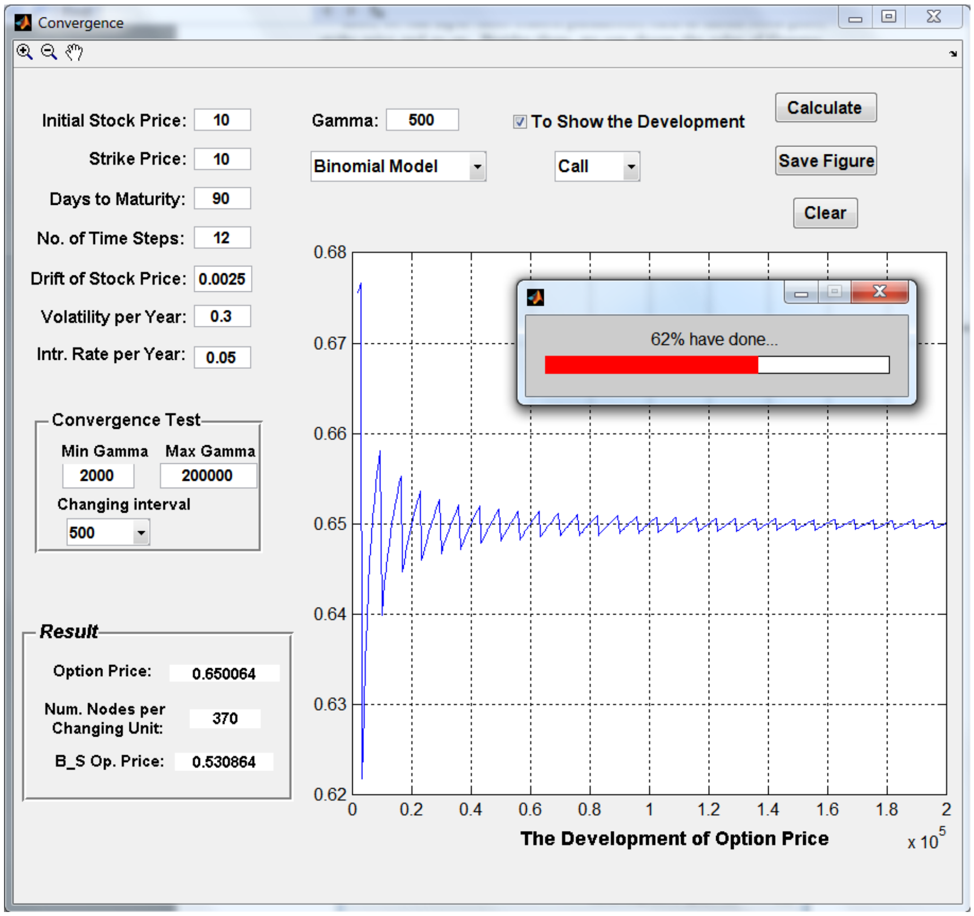

Figure 10: Application program based on these three models

given too.

If we want to show the development option price with an increasing Gamma, we just need to cross the checking box, "To Show the Development", and input the minimal and maximal values of Gamma and the increasing step of it in the "Convergence Test". By clicking the "Calculate" button, the option price trend is shown in the plotting area. We can save the picture by push the "Save Figure" button. The "Clear" button erease all input information and result, so we can start again easily.

4.2 Rounding errors

MATLAB stores numeric data as a double by default. Otherwise single-precision can be used too. By input following commandos respectively in Command Window in MATLAB[2],

```
-realmax   -realmin    realmin    realmax
-realmax('single') -realmin('single')
 realmin('single')  realmax('single')
```

we can get the ranges for both double- and single-precision, and they are, for double:

$$-1.79769e^{308} \sim -2.22507e^{-308} \text{ and } 2.22507e^{-308} \sim 1.79769e^{308}$$

The range for single is:

$$-3.40282e^{038} \sim -1.17549e^{-038} \text{ and } 1.17549e^{-038} \sim 3.40282e^{038}$$

An other difference between them is, in single-precision we can get only a 8-digit floating value but in double-precision we can a 16-digit floating vaule. For example,

```
>> single(2/3)
ans =
   0.6666667
```

and

```
>> double(2/3)
ans =
   0.666666666666667
```

When we calculate the probaility in formula (57), for example, when the number of nodes $r_{\varepsilon,n}$ becomes large, we want to have high precision in the calculation, because the value of the probabilities are very small especially for tail values. Based on this concern, we will test, if the rounding error has a significent effect on our calculation result!

In MATLAB, it supplies a good envirment to apply a such test. We have seen the difference between single- and double-precision. We will ask, can

[2]http://www.mathworks.se/help/techdoc/matlab_prog/f2-12135.html

Bino. Model	N= 12	N= 100	N = 1000
r_ε	10^4	10^5	10^6
Op. V. (Sing.)	0,6398935	0,6467319	0,645829
Op. V. (Doub.)	0,6398811347236	0,6468344630263	0,6468593861018
Diff.	$< 10^{-4}$	1.6×10^{-4}	1.6×10^{-3}
Time for Single	0.078	0.296	8.58
Time for Double	0.0936	0.2184	5.7096

Table 10: Option prices in single- and double-precision

we find any significent rounding error if we change from double- to single-precision? If it is not, then we can say that this kind of error is negligible when we use double precision.

So we choose three examples under the standard condition with different r_ε and N, in Binomial model. The result is shown in Table 10.

Here, we can see that the difference[3] of option values between single- and double-precisions is less than 0.02% up to $N = 100$, far below our precision concern for simulation results, which is just 1%. Even for $N = 1000$, the difference is less than 0.2%. So we can say that the rounding error in our case is negligible. We did not find the speed advantage with the single-precision neither. The reason, maybe, is that the double-precision is the default setting in MATLAB.

5 Concluding Remarks

5.1 Conclusion

Because of the free boundary problem for American options, there is no simple closed-form solution to them, except for some special cases. In the present paper, we tried to apply lattice methods to compute the prices of American options.

Three different models, the binomial sum model, trinomial sum model and skeleton model are tested for pricing American options. The convergence conditions for them are tested, and some numerical tests based on those three methods are implemented in MATLAB.

Our conclusion is:

[3]It is the absolut value of rate of difference.

- All of these three models are, theoretically and practically, suitable to price American options.

- Skeleton Model is the most efficient model among them.

- The rounding error problem is tested, and it does not have significificient effect in our calcultion of option price.

5.2 Discussion and further development

In this paper, the stock price process we simulate is a standard geometric Brownian Motion, with constant drift and volatility. But there are other kinds of processes which are more suitable in the economical reality, for example, the mean-reversion model. The constraint of constant volatility can also be released, and instead be treated as a stochastic process. Those kinds of price process models can be suitable for further studies by using the models studied here.

Another direction could to be focus on different reward functions, for exampel barrier option, Asian option or other so-called exotic options. There the exercise conditions are more flexible and also path-dependent, so lattice method will be suitable for them too.

Appendix A: Convergence of Option Prices, $\sigma = 300\%$.

Here, the pictures in the three models under the standard condition, except the high standard volatility are shown below. In some pictures, we changed the value on step for large r_ε.

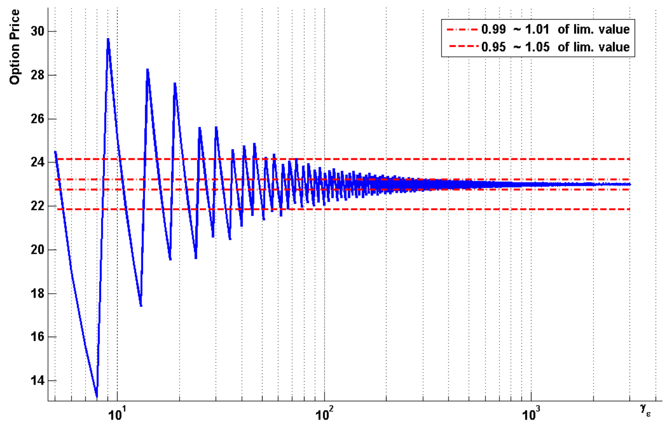

Figure 11: Call-opt. converg. in binomial sum model, N=12, $\sigma = 300\%$.

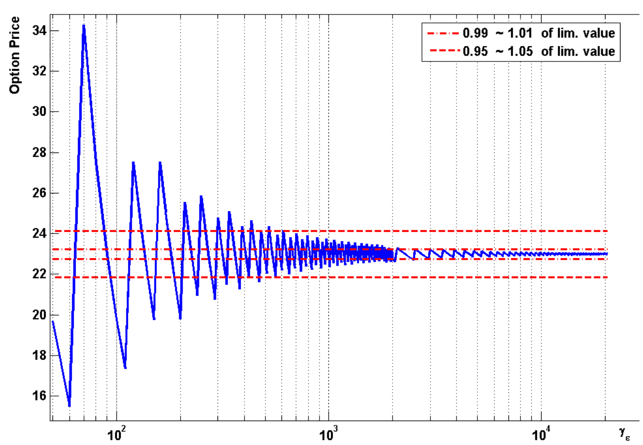

Figure 12: Call-opt. converg. in binomial sum model, N=100, $\sigma = 300\%$.

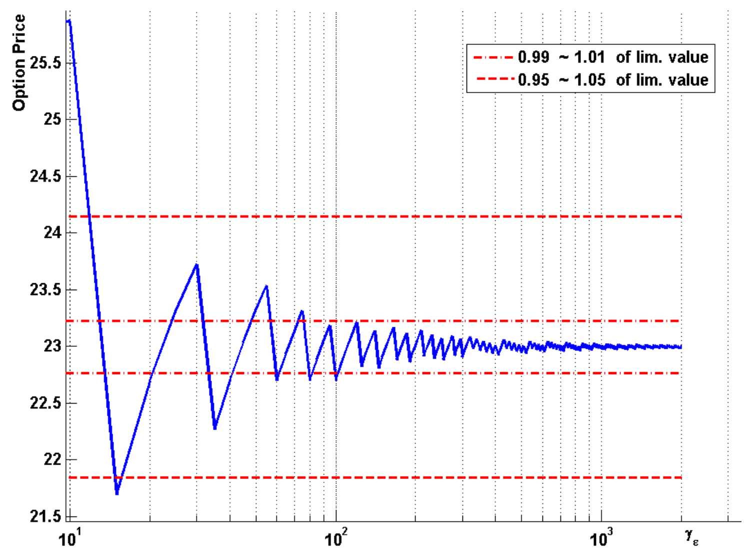

Figure 13: Call-opt. converg. in triomial sum model, N=12, $\sigma = 300\%$.

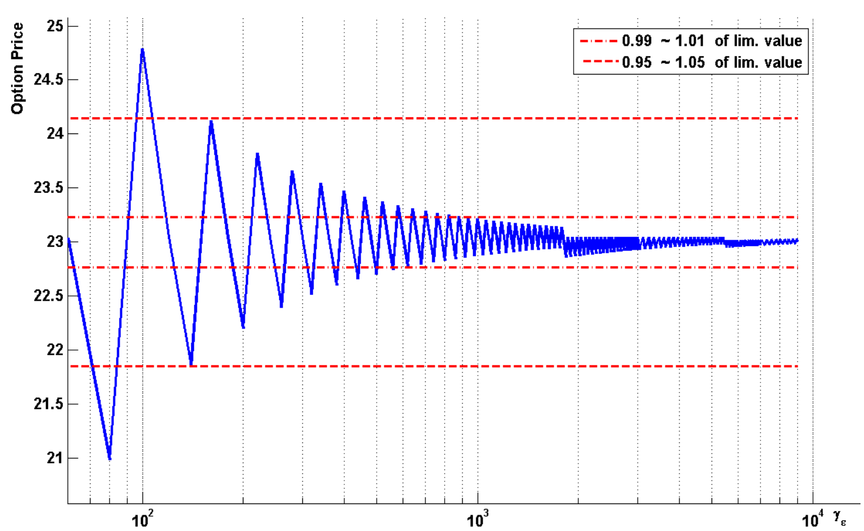

Figure 14: Call-opt. converg. in trinomial sum model, N=100, $\sigma = 300\%$.

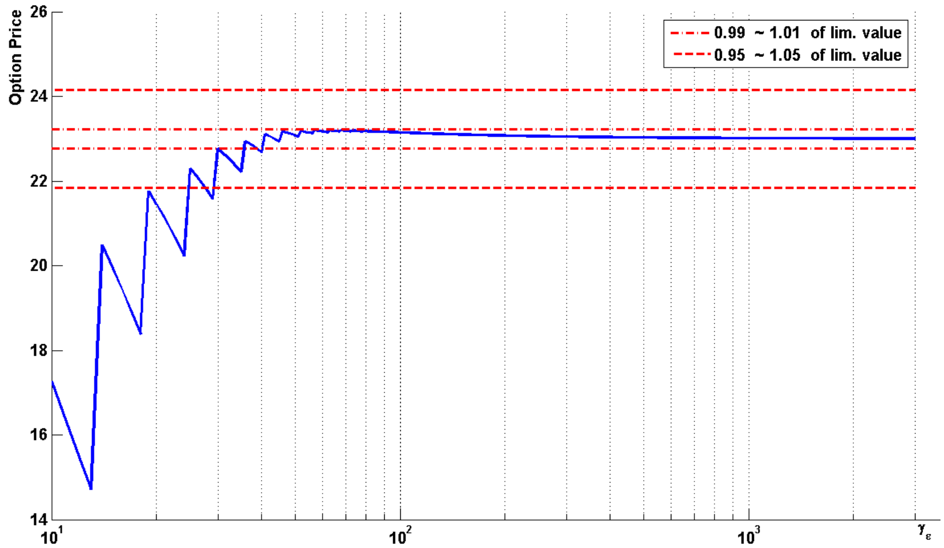

Figure 15: Call-opt. converg. in skeleton model, N=12, $\sigma = 300\%$.

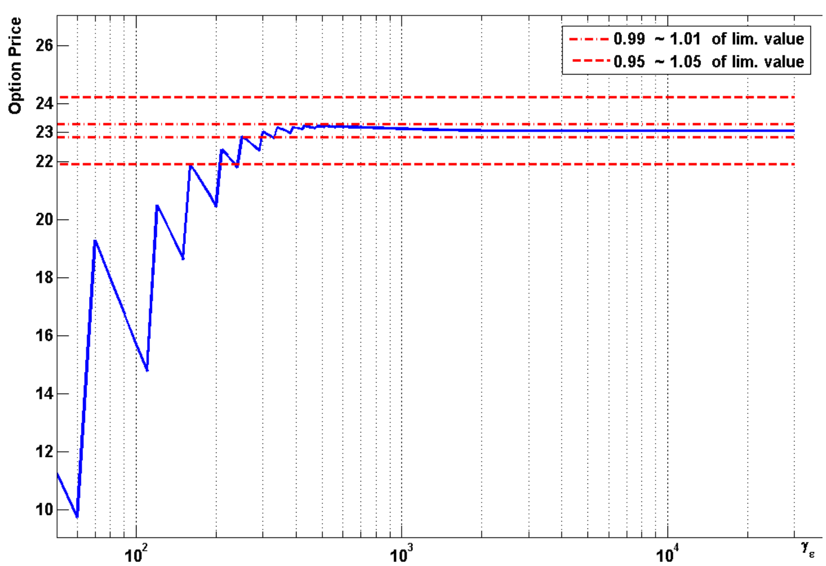

Figure 16: Call-opt. converg. in skeleton model, N=100, $\sigma = 300\%$.

Appendix B: MATLAB Codes

We write our program in the standard GUI Layout Editor (GUIDE) in MATLAB, and the interface of our program is shown in Figure 10. Here we will just present the core functional programs of our MATLAB program for curious readers. These functional programs are:

- ProbMatrixBinoFuc
- ProbMatrixTrioFuc
- BinoSumCallFuc
- TriSumCallFuc
- SkeletonCallFuc

• ProbMatrixBinoFuc:[4]

```
function PMatrix = ProbMatrixBinoFuc(P,N)
    % P=1/2, N is the time steps
    % \delta_t= T/N;

P_Plus = P;
P_Minus = P;

PMatrix{1,1} = 1;            % To build a Probability Cell Array

for i = 2:(N+1)

         tempMatrix = PMatrix{i-1,1};
     tempMatrixPlus = [tempMatrix 0]*P_Plus;
    tempMatrixMinus = [0 tempMatrix]*P_Minus;

    PMatrix{i,1} = tempMatrixPlus + tempMatrixMinus;

end
```

[4] A detailed decribetion of similar algorithm is referred to formula (62).

```
PMatrix(1,:) = [];            % To get rid of the first row of the
                              % Pmatrix cell arry, which constains
                              % just numberal 1.
```

• ProbMatrixTrioFuc:[5]

```
function PMatrix = ProbMatrixTrioFuc(P,N)
    % P_0=2/3,  the probability of no changing;
    % N is the time steps, \delta_t= T/N.

    P_0 = P;                   % Probability for no-changing.
P_Minus = (1-P_0)/2;           % Probability for "down" movement.
 P_Plus = (1-P_0)/2;           % Probability for "up" movement.

PMatrix{1,1} = 1;              % To build a Probability Cell Array

for i = 2:(N+1)

    tempMatrix = PMatrix{i-1,1};

     tempMatrixPlus = [tempMatrix 0 0]*P_Plus;
        tempMatrix0 = [0 tempMatrix 0]*P_0;
    tempMatrixMinus = [0 0 tempMatrix]*P_Minus;

    PMatrix{i,1} = tempMatrixPlus + tempMatrix0 + tempMatrixMinus;
end

PMatrix(1,:) = [];             % To get rid of the first row of the
                               % Pmatrix cell arry, which constains
                               % just numberal 1.
```

• BinoSumCallFuc:

```
function PutPrice = BinoSumCallFuc(gamma,N,Stock_0,strikeP,...
                                   mu,sigma,alfa,PMatrix)
```

[5] A detailed decribetion of this algorithm is referred to formula (62).

```
s0 = Stock_0;                % The Initial Stock Price.
K = strikeP;                 % Strike Price.

delta = 1/sqrt(gamma);       % Jump magnitude.

muM = (mu)*ones(N,1);        % Trend for Stock Price Process.
sigmaM = (sigma)*ones(N,1);  % Volatility of Stock Price.
alfaM = (alfa)*ones(N,1);    % Risk Free Interest Rate.

%% Changing Probilities for every time interval.
gamma_n = round(sigmaM.*sigmaM.*gamma);   % Num. of Nodes in
                                          % one Changing Unit.

for i = 1:N
  SubProbability{i,1} = PMatrix{gamma_n(i),1};
end

%% Total Possible Jumping Numbers.
M = sum(gamma_n);
dpowers = -delta*((0:M)');
upowers = delta*((M:-1:0)');

%%Option Price at maturity, 'N'.
priceN = max(s0*exp(dpowers+upowers+sum(muM))-K,0);  %Call Option

%% Re-trace to get option value at time zero
for i = N:-1:1

    expPrice = 0;

    prob = SubProbability{i,1};

    % to calculation exp. value of Option Price at moment 'i-1'.
    for j = 1:(gamma_n(i)+1)
        expPrice = expPrice+prob(j)*priceN(j:(M-gamma_n(i)+j));
    end

    % to calculation Stock-price at moment 'i-1'.
```

```
    M = sum(gamma_n(1:(i-1)));
    dpowers = -delta*((0:M)');
    upowers = delta*((M:-1:0)');
    Si = s0*exp(dpowers+upowers+(i-1)*muM(i));

    % max(excercised option price at moment (i-1), present value
    % of expected option value at moment (i-1)). So this is
    % option price at moment 'i-1'.

    priceN = max(max(Si-K,0),expPrice*exp(-alfaM(i)));
end

PutPrice = priceN;
```

- **TriSumCallFuc:**

```
function PutPrice = TriSumCallFuc(gamma,N,Stock_0,StrikeP,mu,...
                                  sigma,alfa,PMatrix,P_2)

s0 = Stock_0;                 % The Initial Stock Price.
K = StrikeP;                  % Strike price.
PP = PMatrix;

delta = sqrt(1/gamma);        % Jump magnitude.

muM = (mu)*ones(N,1);         % Trend for Stock Price Process.
sigmaM = (sigma)*ones(N,1);   % Volatility of Stock Price.
alfaM = (alfa)*ones(N,1);     % Risk Free Interest Rate.

%% Changing Probilities for every time interval.
gamma_n = round(sigmaM.*sigmaM.*gamma/(1-P_2));

for i = 1:N
  SubProbability{i,1} = PP{gamma_n(i),1};
end

%% Total Possible Jumping Numbers.
```

```
M = sum(gamma_n);
powers = delta*((M:-1:-M)');


%%Option Price at maturity, 'N'.
priceN = max(s0*exp(powers+sum(muM))-K,0);

%% Re-trace to get option value at time zero
for i = N:-1:1

    expPrice = 0;

    prob = SubProbability{i,1}';

    % to calculation exp. value of Option Price at moment 'i-1'.
    for j = 1:(2*gamma_n(i)+1)
        expPrice = expPrice+prob(j)*priceN(j:(2*M-2*gamma_n(i)+j));
    end

    % to calculation Stock-price at moment 'i-1'.
    M = sum(gamma_n(1:(i-1)));
    powers = delta*((M:-1:-M)');
    Si = s0*exp(powers+sum(muM(1:(i-1))));

    % max(excercised option price at moment (i-1), present value
    % of expected option value at moment (i-1)). So this is
    % option price at moment 'i-1'.

    priceN = max(max(Si-K,0),expPrice*exp(-alfaM(i)));
end

PutPrice = priceN;
```

- **SkeletonCallFuc:**

```
function PutPrice = SkeletonCallFuc(gamma,N,Stock_0,strikeP,...
                                    mu,sigma,alfa)
```

```
s0 = Stock_0;                  % The Initial Stock Price.
K = strikeP;                   % Strike price

endPlus = inf;
endMinus = -inf;

delta = 1/sqrt(gamma);         % Jump magnitude.

muM = (mu)*ones(N,1);          % Trend for Stock Price Process.
sigmaM = (sigma)*ones(N,1);    % Volatility of Stock Price.
alfaM = (alfa)*ones(N,1);      % Risk Free Interest Rate.

%% Changing Probilities for every time interval.
gamma_n = round(sigmaM.*sigmaM.*gamma);
for i = 1:N

    A = (gamma_n(i)-0.5):-1:(-gamma_n(i)+0.5);
    D = [endPlus A endMinus];

    for j = 1:(2*gamma_n(i)+1)
        temp1 = D(j)*delta/sigmaM(i);
        temp2 = D(j+1)*delta/sigmaM(i);
        temp3 = normcdf([temp1 temp2]);
        Prob(j) = temp3(1)-temp3(2);
    end

SubProbability{i,1} = Prob;

end

%% Total Possible Jumping Numbers.
M = sum(gamma_n);
powers = delta*((M:-1:-M)');

%%Option Price at maturity, 'N'.
priceN = max(s0*exp(powers+sum(muM))-K,0);
%% Re-trace to get option value at time zero
```

```
for i = N:-1:1

    expPrice = 0;

    prob = SubProbability{i,1}';

    % to calculation exp. value of Option Price at moment 'i-1'.
    for j = 1:(2*gamma_n(i)+1)
        expPrice = expPrice+prob(j)*priceN(j:(2*M-2*gamma_n(i)+j));
    end

    % to calculation Stock-price at moment 'i-1'.
    M = sum(gamma_n(1:(i-1)));
    powers = delta*((M:-1:-M)');
    Si = s0*exp(powers+sum(muM(1:(i-1))));

    % max(excercised option price at moment (i-1), present value
    % of expected option value at moment (i-1)). So this is
    % option price at moment 'i-1'.

    priceN = max(max(Si-K,0),expPrice*exp(-alfaM(i)));

end

PutPrice = priceN;
```

References

- Ahn, S.R., Bae, H., Koo, H.K., Lee, K. A survey on American options: old approaches and new trends, *Bull. Korean Math. Soc.* **48**, no. 4, (2011), 791-812.

- Black F., Scholes, M. The pricing of options and corporate liabilities, *J. Polit. Econ.*, **81**, (1973), 637-659.

- Boyle, P. Option valuation using a three jump process, *Inter. Option. J.*, **3**, (1986), 7-12.

- Boyle, P. A lattice framework for option pricing with two state variables, *Finan. Quant. Anal.*, **23**, (1988), 1-12.

- Broadie, M., Detemple, J.B. American option valuation: new bounds, approximations, and a comparison of existing methods, *Rev. Finan. Stud.*, **9**, no. 4 (1996), 1211-1250.

- Broadie, M., Detemple, J.B. Option pricing: valuation models and applications, *Manag. Sci.*, **50**, no. 9 (2004), 1145-1177.

- Chow, Y.S., Robbins, H., Siegmund, D. The Theory of Optimal Stopping, *Houghton Mifflin Comp.* (1971).

- Cox, J.C., Ross, S.A., Rubinstein, M. Option pricing: a simplified approach, *J. Finan. Econ.*, **7**, (1979), 229-264.

- Heston, S., Zhou, G. On the rate of convergence of discrete-time contingent claims, *Mathe. Finan.*, **10**, no. 1 (2000), 53-75.

- Higham, D.J. Nine ways to implement the binomial method for option valuation in MATLAB, *SIAM Rev.*, **44**, no. 4, (2002), 661-677.

- Lundgren, R., Silvestrov, D. Optimal stopping and reselling of European options, In: Rykov, V., Balakrishnan, N., Nikulin, M. (Eds.) *Mathematical and Statistical Models and Methods in Reliability*, Birkhäuser, (2010), 371-390.

- Lundgren, R., Silvestrov, D. Convergence of option rewards for multivariate price processes. *Research Report 2009:10*, Department of Mathematics, Stockholm University, 53 pages.

- Merton, R.C. Theory of rational option pricing, *Bell J. Econom. Manag. Sci.*, **4**, (1973), 141-183.

- Pauly, O. Numerical simulation of American options, Ulm University, Diplomarbeit in Wirtschaftsmathematik, (2004).

- Shiryaev, A.N. *Sequential Analysis. Optimal Stopping Rules,* Nauka, Moscow, (1976) (English edition: Optimal Stopping Rules, Springer, New York, 1978).

- Silvestrov, D. *Optimal Stopping and Convergence for American Type Options*, (2012), (a book in preparation).