MÄLARDALEN UNIVERSITY
SWEDEN

Department of Mathematics and Physics

BACHELOR THESIS IN MATHEMATICS/ APPLIED MATHEMATICS

Title: Monte Carlo studies of generalized barrier options.

by

Takura Witness Muusha

Kandidatarbete i matematik/tillämpad matematik

Code: MDH.IMA.MAT.0087 (2007) 10p – AF

MÄLARDALEN UNIVERSITY
SWEDEN

Department of Mathematics and Physics

Bachelor thesis in mathematics/applied mathematics

Date:
2007-06-13

Project name:
Monte Carlo studies of generalized barrier contracts.

Author:
Takura Witness Muusha

Supervisor:
Robin Lundgren

Examiner:
Professor Dimitrii Silverstrov

Comprising: 10 points

Abstract

This paper examines the pricing of barrier options using Monte Carlo Simulations. MATLAB based software is developed to estimate the price of the option using Monte Carlo simulation. We consider a generalized barrier option of knock out type, but we let the domain take the shape of a rectangular box. We investigate the price of this kind of barrier options. We investigate how the box is placed and what effect it will have on the price of the option. We compare the number of trajectories that are needed in order to achieve the same accuracy between this box barrier option and an ordinary option.

Acknowledgement

I would like to thank my supervisor Robin Lundgren for his guidance and assistance during this project.

Table of Contents

1. Introduction

In this paper we are going to use the numerical method of Monte Carlo to value the price of generalized barrier option. Barrier options have been traded since the 1960`s, but the first analytical formulas were proposed by [Merton 1973]. The use of Monte Carlo simulation in finance was pioneered by [Boyle, 1977] and today it's a popular and cornerstone method of pricing financial options and derivatives thanks to the powerful mathematical software. The simulations have many advantages, including the ease of implementation and applicability to multi-dimensional problems commonly encountered in finance.

This rest of the paper is organized as follows section 2 Theoretical overview 3 simulation procedure, section 4 Monte Carlo methods, section 5 Numerical experiments and section 6 concludes the paper. Section 7 contains the referees and lastly in section 8 we have the appendix which contains the full MATLAB codes and tables used in the experiments analysis in this paper.

2. Theoretical overview

An option is a contract to buy or sell an underlying instrument; the contract is very precise as it establishes a specific strike price, and expiration date at which the contract may be exercised. Options come in two kinds' calls and puts. A call option gives its holder the right to buy the underlying instrument at the strike price, anytime prior to the options expiration date. The writer of the option has the obligation to sell the instrument. A put option gives its holder the right to sell the underlying instrument at the strike price, anytime prior to the options expiration date. The writer of the option has the obligation to buy the instrument.

Barrier options are options whose main characteristic is that the payoff is initiated when the underlying asset price reaches a predetermined level during a certain period of time. Barrier options belong to the class of path-dependent options, this mean that it's not only the value of the underlying asset price at maturity that is important, but also the path the underlying price has taken up to maturity

In recent years a lot of interest in barrier options has developed both amongst scholars and investors as evidenced by huge number of literature published and an increase in trading transactions. This can be attributed to the fact that these contracts are relatively cheaper as compared to the traditional options. Also these options can be tailor made to suit the preferences of an investor when determining the value of the barrier.

Barrier options can have either one or two boundaries, and both single and double barrier options can be classified in two categories, knock in and knock out. A knock in option cease to exist when the underlying asset price reaches a certain boundary and a knock out option comes into existence only when the underlying asset price reaches the barrier. Under these two categories we have four classes which a boundary option can take;

2.1 Knock-in

1. Up and in the option is only active if the barrier is hit from below.

$$Max(S_T - K, 0)\chi(S_t \geq B)$$

2. Down-and in, then the option is only active if the barrier is hit from above.

$$Max(K - S_T, 0)\chi(S_t \leq B)$$

2.2 Knock-out

3. Up and out, then the option is worthless if the barrier is hit from below.

$$Max(S_T - K, 0)\chi(S_t \leq B)$$

4. Down and out, then the option is worthless if the barrier is hit from above.

$$Max(K - S_T, 0)\chi(S_t \geq B)$$

Where; S_T *is the underlying asset at expiry.*
K *is the strike price.*
S_t *is the underlying asset price.*
B *is the barrier.*
$\chi(S_t \geq B)$ *is the indicator which mean that its true otherwise its zero.*

Since barrier options are path dependent their underlying asset price of a can either be checked continuously or discretely. In the case where we have continuous monitoring the option is instantly knocked in or out once the barrier is reached while with discrete it can be a once day, week or month. In this paper we are going to look at knock out barrier options under discrete time. The payoff of this option depends on the breaching behavior of the underlying stock price process with respect to the barriers. We define the structure knock out domain as H and $H = \{H_1, H_2, \dots, H_n\}$ if the underlying process enters the domain then the contract becomes worthless [R.Lundgren, 2007].

$$H_n = \{x: h(n,x) \leq 1\} \subseteq \mathbb{R}^+$$

$$for\ h(n,x) = \chi(c \leq n \leq d) \left|\frac{zx-a-b}{a-b}\right|$$

$$where\ c,d \in \mathbb{Z}^+\ \ and\ c < d\ \ \ Also\ \ a,b \in \mathbb{R}^+\ and\ a > b$$

$a - is\ the\ upper\ boundary$
$b - is\ the\ lower\ boundary$
$c - is\ starting\ point\ of\ our\ boundary\ interval$
$d - is\ the\ end\ point\ of\ our\ boundary\ interval$

In this paper we consider a complete discrete world, so the theory for pricing barrier contracts with Monte Carlo simulation presented in [Baldi et, 1999] will not be needed.

3. Simulation Procedure

3.1 Geometrical Brownian motion and Pricing of an Option

In this paper we assume that the underlying asset price S_t at time t evolves randomly as geometric Brownian motion in the time interval $[0, T]$. To be able to simulate the asset price over the interval $[0, T]$ we have to discretize the interval with time steps δt such $T = N * \delta t$ then we indicate the discrete time intervals by $t = 0,1,2, \dots, T$. There is a unique solution at each time step δt which is simulated by its value at t. The model under risk neutral probability measure is as shown below the drift is set equal to the risk free interest rate [Brandimarte,2002].

$$S_t = S_0 exp\left(\left(r - \frac{\sigma^2}{2}\right)\delta t + \sigma\sqrt{\delta t}\,\varepsilon\right) \quad (1)$$

Where S_0 the initial stock price, r Denotes the risk free interest rate, σ is the volatility and $\varepsilon \sim N(0,1)$ a standard normal random variable, we use this to generate sample prices S_t at some future time T, given the initial price S_0, the payoff function of the option is;

$$g(S_t) = Max(S_T - K, 0)\ \chi\ (S_t \notin H)\ \forall: t \quad (2)$$

Where K denotes the strike price. This gives us the option pricing formula as;

$$C_0 = e^{-rT}\mathbb{E}[g(S_t)] \quad (3)$$

$g(S_t)$ is the payoff at the maturity date T and $\mathbb{E}^c[g(S_t)]$ is the expectation with respect to risk neutral measure [Glasserman,2004].

4. Monte Carlo Methods

Monte Carlo is a method of estimating a value by generating a set of random normally distributed numbers, the simulations are a replication of the real-world procedure over time. Monte Carlo Simulations are used to determine solutions of problems that are difficult to solve using analytical formulas.

The classical result for the strong law of large numbers gives us the following crucial theorem for Monte Carlo simulation.

Theorem 1: (Strong law of large numbers)

It states that for a family of independent and identically distributed random variables;
$g(S^{(1)}), g(S^{(2)}), \ldots$, suppose that the mean $\mu = E[g(S^{(1)})]$ exists then

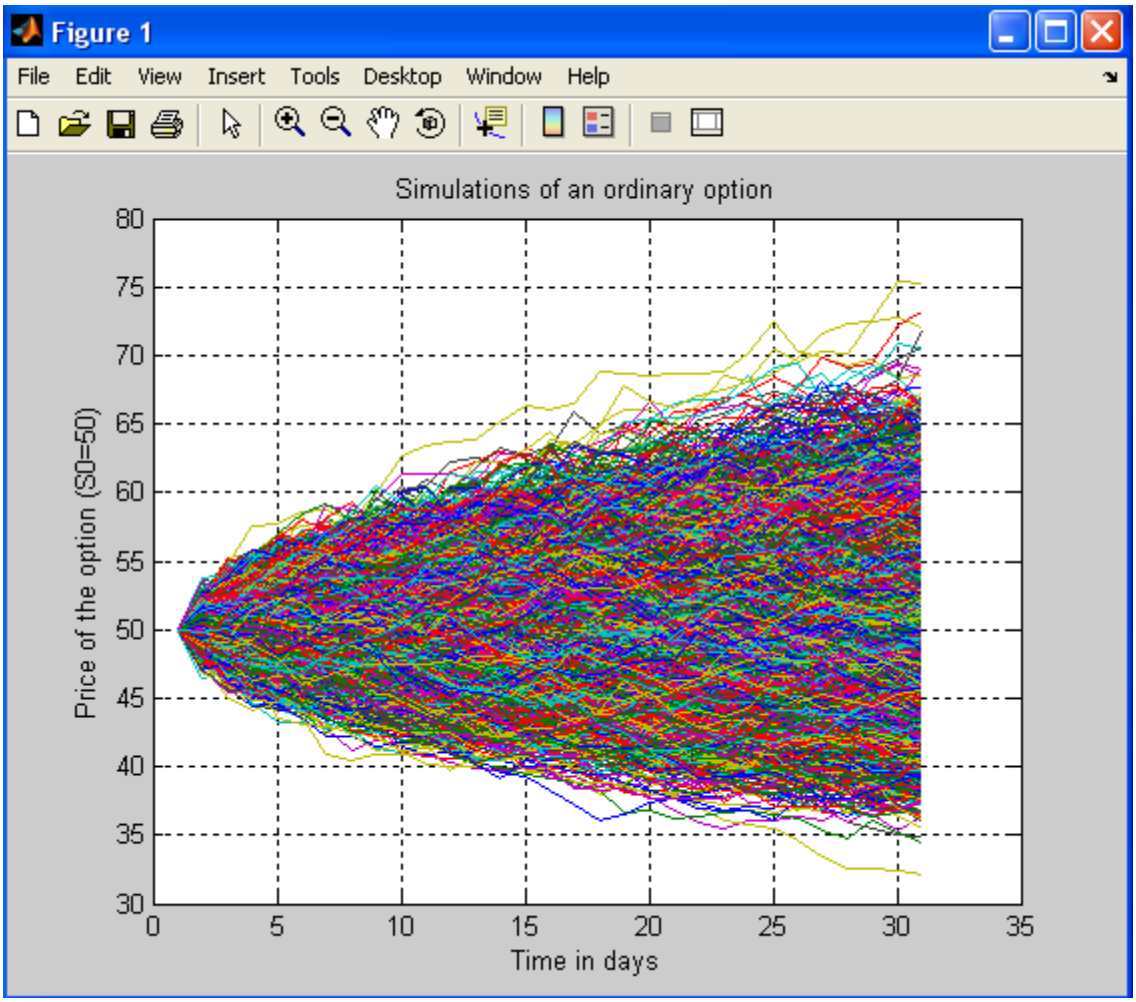
$$\lim_{n\to\infty} \frac{g(S^{(1)}) + g(S^{(2)}) + \cdots + g(S^{(n)})}{n} = E[g(S^{(1)})]$$

with probability one, this law ensures that the sample mean converges to the population mean as $n \to \infty$.

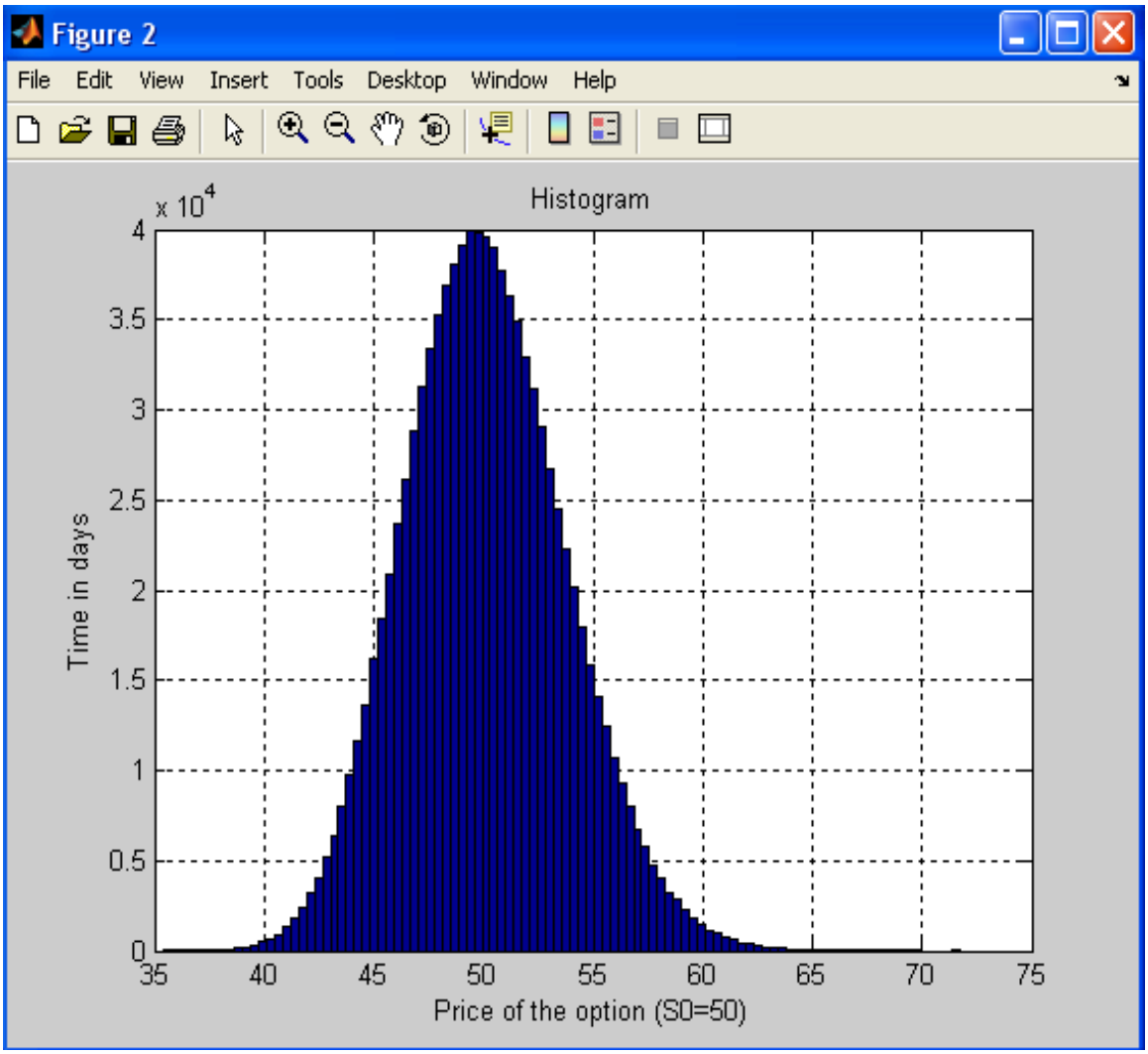
Our aim is to calculate the option prices $C(S_0)$ in this case we need to we need to estimate equation (3) the expected value of a discounted payoff of the option. In order to estimate the expected value we need to have a large integer value of N then approximate the future expected value by taking the mean of the discounted payoff as shown below.

$$C(S_0) \approx \frac{e^{-rT}}{N} \sum_{i=1}^{N} \left[g\left(S_t^{(i)}\right) \right] \qquad (4)$$

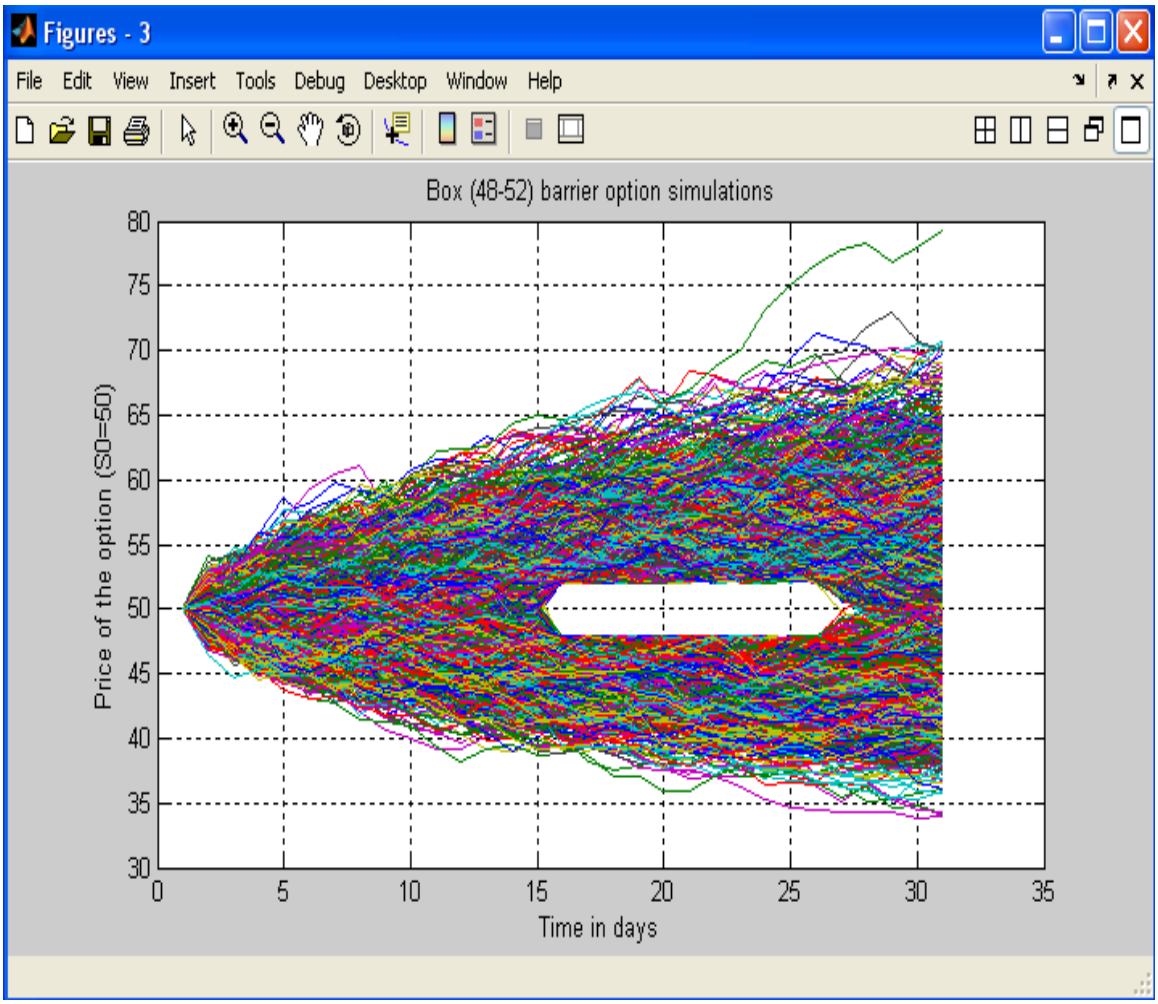
Figure 1 below shows the simulations of stock prices in general as explained by the mathematical procedure above.



To determine the position of our box we use the properties of a histogram, this is constructed using the estimated price of the ordinary option. As seen from the figure below the shape of this histogram is log normally distributed. The histogram at maturity tells us where it is most probable that the underlying process will be.



Given the above facts above we decide to place our box where many of the trajectories are concentrated as this will give us interesting statistical analysis for this paper. The area of interest to us is given by a rectangle (box) with diagonal co-ordinates with y_1, x_1 being the lower left co-ordinates and x_2, y_2 being the upper right co-ordinates. The rest of the area can easily be derived using the known co-ordinates. Since x_1 and x_2 are predetermined.



The blank place shown at the centre of the diagram above clearly marks our rectangular box. It is in this area of this box that we are interested in examining what the price of our option will be. The position of the box can be moved to various places but the number of days under examination will remain the same.

5. Numerical Experiments

The central limit theorem states that the sum of many independent and identically-distributed random variables will be approximately normally distributed. The characteristics of normal distribution are that the curve is bell shaped, there is a single peak point on the curve, the mean lies in the center of the distribution and therefore the curve is symmetric around the mean.

The outlined in simulation procedure is now applied in this section to compute the option prices in five experiments. In the first experiment we are going to calculate the prices of our default option, this will have the following parameters;

1. $S0\ is\ the\ initial\ stock\ price.$
2. $r\ is\ the\ risk\ free\ interest\ rate.$
3. $sigma\ is\ the\ volatility.$
4. $T\ is\ time\ horizon\ (year).$
5. $NStep\ is\ the\ number\ of\ trading\ days\ in\ a\ year.$
6. $NRepl\ is\ the\ number\ of\ simulations.$

$S0 = 50, r = 0.03, sigma = 0.3, T = 1\ , Nstep(Days) = 30, Number$ Simulations $\{from\ 10^4 to (20 * 10^6)\}$

For us to carry out further investigations on the knock out barrier option we are going to add more parameters and these are

1. $y_2\ being\ the\ upper\ boundary.$
2. $y_1\ being\ the\ lower\ boundary.$
3. $x_1\ being\ the\ starting\ point\ of\ our\ interval.$
4. $x_2\ being\ the\ ending\ point\ of\ our\ interval.$

$x_1 = 15\ to\ x_2 = 25\ and\ y_1 = 48\ to\ y_2 = 52$

In order compare the results of our experiments we are going to use some statistical measure that will enable us to give detailed analysis of our experiments.

5.1 Mean

The mean is one of several indices of central tendency that statisticians use to indicate the point on the scale of measures where the population is centered. In our paper the mean provides with an estimation of the option price for the sample data. In this paper the mean will also act as the expected value because the calculation of expected value is given by price multiply by probability of the price occurring.

5.2 Standard deviation

The standard deviation of random variables is a measure of the spread of its values in our paper it will measure the accuracy. We will use the standard deviation to compare how many simulations that needed as we look at different options and the position of our rectangular box.

5.3 Error

The Standard Error, or Standard Error of the Mean, is an estimate of the standard deviation of the sampling distribution of means, based on the data from one or more random samples. The Central Limit Theorem insures that the standard error of the estimate tends to zero; this convergence rate is based on the assumption that the random variables are generated with the use of pseudo-random numbers

5.4 Confidence interval

A confidence interval gives an estimated range of values which is likely to include an unknown population parameter, the estimated range being calculated from a given set of sample data. This interval shows how accurate the estimate really is and if more time and effort are needed for additional precision. In our paper we use the 95% confidence interval and we need to check whether our estimated expected price of the option fall in this range.

5.5 Results

For each one of our five numerical experiments we estimated the prices by Monte Carlo simulations, this done was twenty times from ten thousand to twenty million. In all the five experiments there is enough evidence that the higher the number of simulations the better the results are. The estimate of our expected price is represented by the mean and at two decimal places is stable with more simulations. The same trend is noticeably with the standard deviation, the error, and the confidence interval, they all show higher level of accuracy as simulations increase.

Our first analysis is going to be the comparison of three statistical results of table one, two and three. Table one results represent the ordinary option, the price is stable and reasonable at 1.97 after five hundred thousand simulations. Its accuracy as determined by the standard deviation is 0.00889; the number of simulations that are needed in order to achieve the same accuracy if we have a small box are the almost same. The standard deviation for the small box is 0.00827; after five hundred thousand simulations as well. Despite the fact that difference between the two options is little throughout the simulations of the two options, we can still see that more simulations are needed for the ordinary option if we are to get the exact figure obtained by the small box.

The figures used for the parameters are as follows;

$S0 = 50, r = 0.03, sigma = 0.3, T = 1\ , Nstep(Days) = 30, Number\ of\ simulations\ \{from\ 10^4 to (20 * 10^6)\}$

Table1:

Statistical results of default Simulations								
No: Sim	**10^4**	**10^5**	**$5*10^5$**	**10^6**	**$5*10^6$**	**10^7**	**$15*10^6$**	**$20*10^6$**
Mean	1.974515	1.974215	1.972445	1.97199	1.97225	1.971835	1.97194	1.972025
Std dev	0.035617	0.008887	0.003080	0.003042	0.00142601	0.00105295	0.000844	0.000668
Error	0.007964	0.001987	0.000689	0.000680	0.00031887	0.00023545	0.000189	0.000149
Conf Int	0.015610	0.003895	0.001350	0.001333	0.00062496	0.00054615	0.000370	0.000293
Conf range	1.95891; 1.99013	1.97032; 1.97811	1.97110; 1.97380	1.97066; 1.973323	1.960623; 1.972569	1.971374; 1.972297	1.97157; 1.97231	1.97173; 1.97232

$S0 = 50, r = 0.03, sigma = 0.3, T = 1\,, Nstep(Days) = 30, Number\ of\ simulations$
$\{from\ 10^4 to (20 * 10^6)\}\ x_1 = 15\ to\ x_2 = 25\ and\ y_1 = 48\ to\ y_2 = 52$

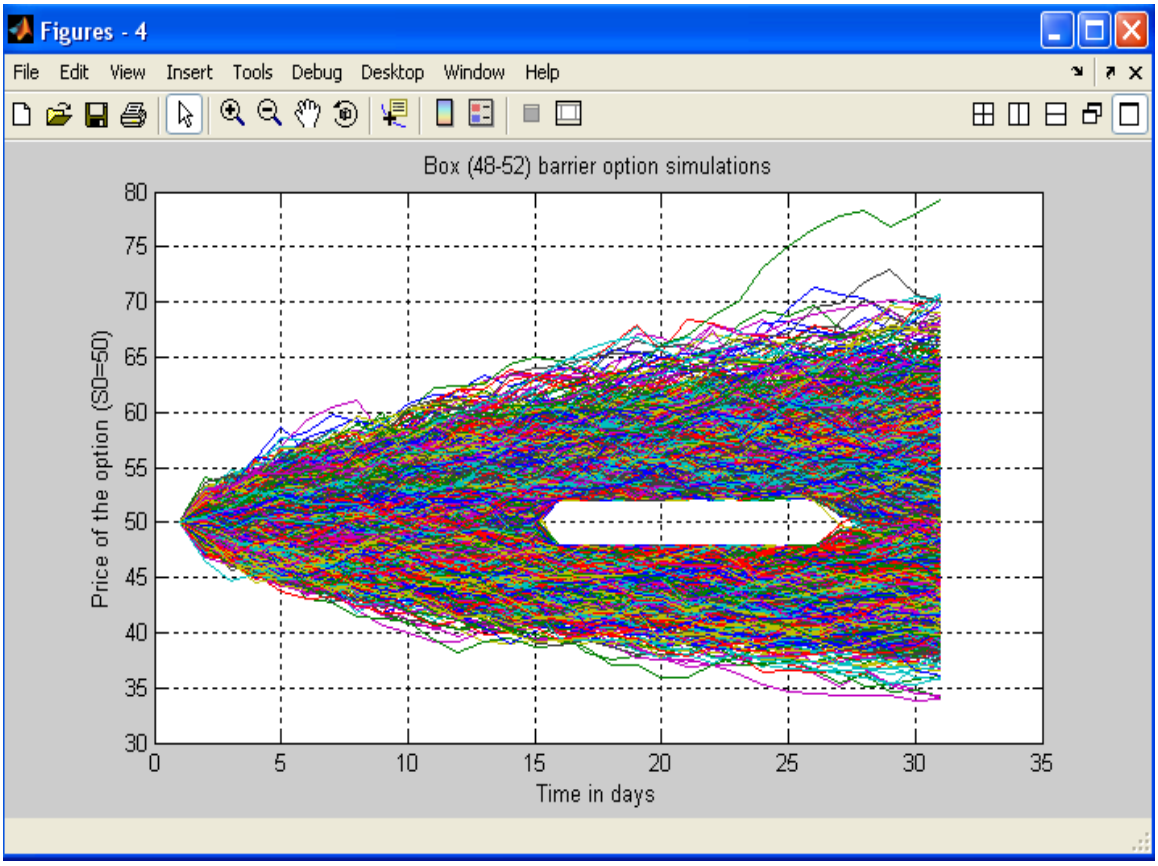


Table2:

Statistical results box simulations 48-52 (Small box)								
No: Sim	**10^4**	**10^5**	**$5*10^5$**	**10^6**	**$5*10^6$**	**10^7**	**$15*10^6$**	**$20*10^6$**
Mean	1.024165	1.019345	1.01823	1.018175	1.017655	1.01749	1.01749	1.01747
Std dev	0.028336	0.008269	0.003864	0.003363	0.001322	0.001021	0.000621	0.0005974
Error	0.006336	0.001849	0.000752	0.000864	0.000296	0.000228	0.000139	0.0001334
Conf Int	0.012419	0.003624	0.001474	0.001694	0.000579	0.000447	0.000272	0.0002618
Conf range	1.011745; 1.036585	1.015721; 1.022969	1.016756; 1.019704	1.016481; 1.019869	1.017076; 1.018232	1.017043; 1.017937	1.017228; 1.017772	1.017208; 1.017732

When we change the size of our box we notice a huge difference in our results. It is now clearer that less number of simulations is needed for the box to achieve the same accuracy. Using three decimal places our standard deviation for the ordinary option is 0.003 after one million simulations and it`s exactly half that when we are considering a bigger box (45-55). As a matter of fact the number of simulations needed for the box is always fewer than when we are considering an ordinary option.

The results of box (50-55) and (60-65) confirm this finding and provide proof that regardless of the position of the box the number of simulations is less at all times. The size of the box is inversely linked to the number of simulations required to achieve the same accuracy. The bigger the box the less the number of simulations required and vice versa.

$S0 = 50, r = 0.03, sigma = 0.3, T = 1\,, Nstep(Days) = 30, Number\ of\ simulations\ \{from\ 10^4 to (20 * 10^6)\}\ x_1 = 15\ to\ x_2 = 25\ and\ y_1 = 45\ to\ y_2 = 55$

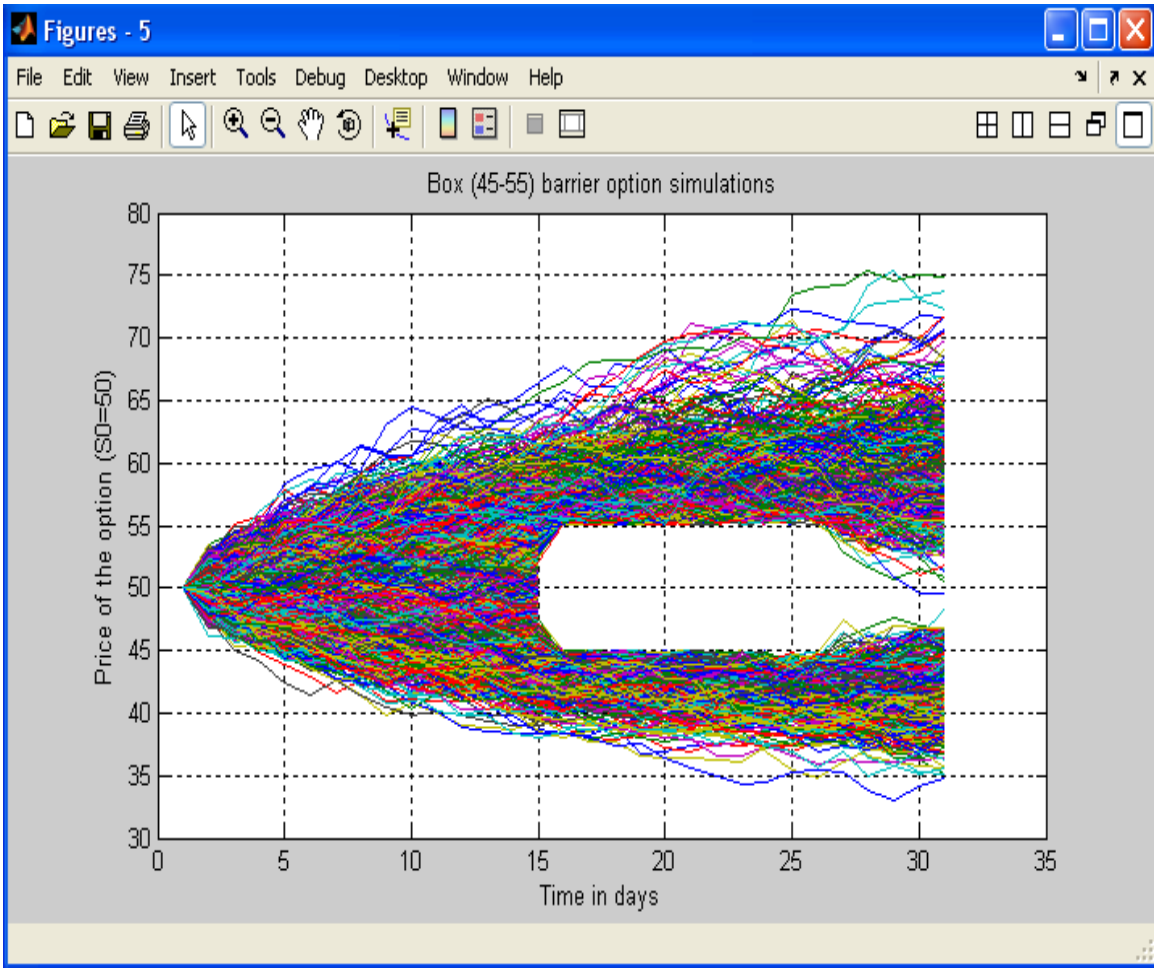


Table 3

Statistical results box simulation 45-55 (Big box)								
No: Sim	**10^4**	**10^5**	**$5*10^5$**	**10^6**	**$5*10^6$**	**10^7**	**$15*10^6$**	**$20*10^6$**
Mean	0.3189	0.31517	0.31525	0.31524	0.31506	0.3149	0.3152	0.3151
Std dev	0.01570	0.00448	0.00252	0.00141	0.00065	0.00047	0.00039	0.00035
Error	0.00351	0.00100	0.00056	0.00032	0.00015	0.00010	0.000086	0.000075
Conf Int	0.00688	0.00196	0.00110	0.00062	0.00029	0.00021	0.00017	0.00015
Conf range	0.3130; 0.3258	0.3132; 0.3172	0.3141; 0.3163	0.3146; 0.3159	0.3148; 0.3154	0.3147; 0.3151	0.3150; 0.3154	0.3150; 0.3153

We now consider boxes which are equal in size but placed at different positions, our aim is to try and find out whether the position matter. The criteria used to choose the positions of the two boxes is derived from the concept of our histogram which is discussed in section three. The first box is placed where a huge number of trajectories are concentrated (50-55). The second box is placed where fewer trajectories are (60-65). It would be obvious that the prices of the barrier options are going to be different but perhaps the interesting part is which position is has a higher price.
There is a difference of 30% in the price of the two options with the higher price in the box (60-65). The price for this box is stable at 1.97 after five hundred thousand simulations. The accuracy of the prices is fluctuating between the boxes with insignificant differences therefore the number of simulations required are almost the same.

$S0 = 50, r = 0.03, sigma = 0.3, T = 1\,, Nstep(Days) = 30, Number\ of\ simulations$
$\{from\ 10^4 to (20 * 10^6)\}\ x_1 = 15\ to\ x_2 = 25\ and\ y_1 = 50\ \ to\ y_2 = 55$

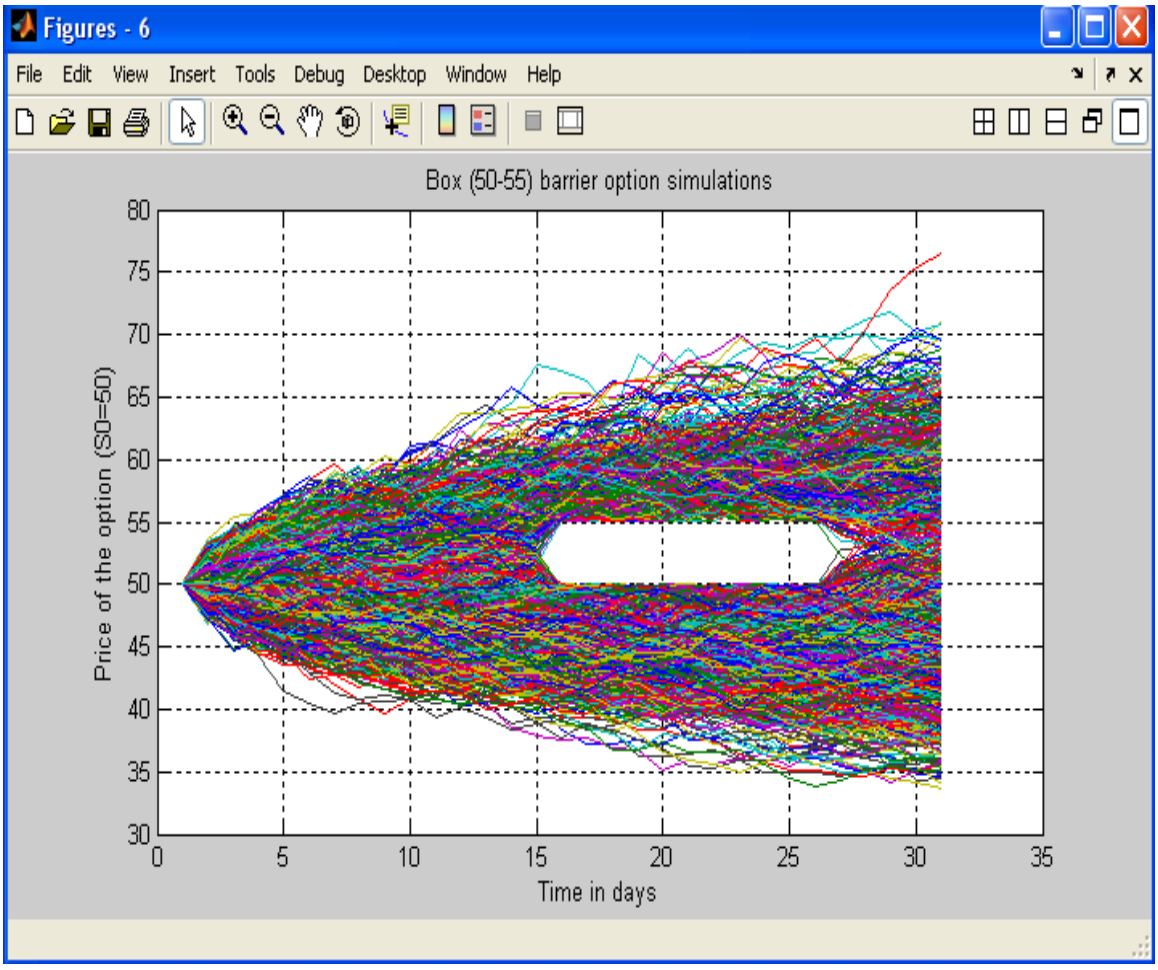


Table 4

Statistical results box simulation 50-55								
No: Sim	**10^4**	**10^5**	**$5*10^5$**	**10^6**	**$5*10^6$**	**10^7**	**$15*10^6$**	**$20*10^6$**
Mean	1.5222	1.5116	1.5122	1.5123	1.5117	1.5115	1.5116	1.5115
Std dev	0.02110	0.00770	0.006229	0.002589	0.00144	0.00109	0.00066	0.00061
Error	0.00472	0.00172	0.00139	0.000580	0.00032	0.00024	0.00015	0.00013
Conf Int	0.00925	0.00338	0.00273	0.001135	0.00063	0.00048	0.00029	0.00027
Conf range	1.5213; 1.5232	1.5083; 1.5151	1.5096; 1.5150	1.5112; 1.5134	1.5111; 1.5124	1.5109; 1.5119	1.5113; 1.5119	1.5112; 1.5117

$S0 = 50, r = 0.03, sigma = 0.3, T = 1\,, Nstep(Days) = 30, Number\ of\ simulations\ \{from\ 10^4 to (20 * 10^6)\}\ x_1 = 15\ to\ x_2 = 25\ and\ y_1 = 60\ to\ y_2 = 65$

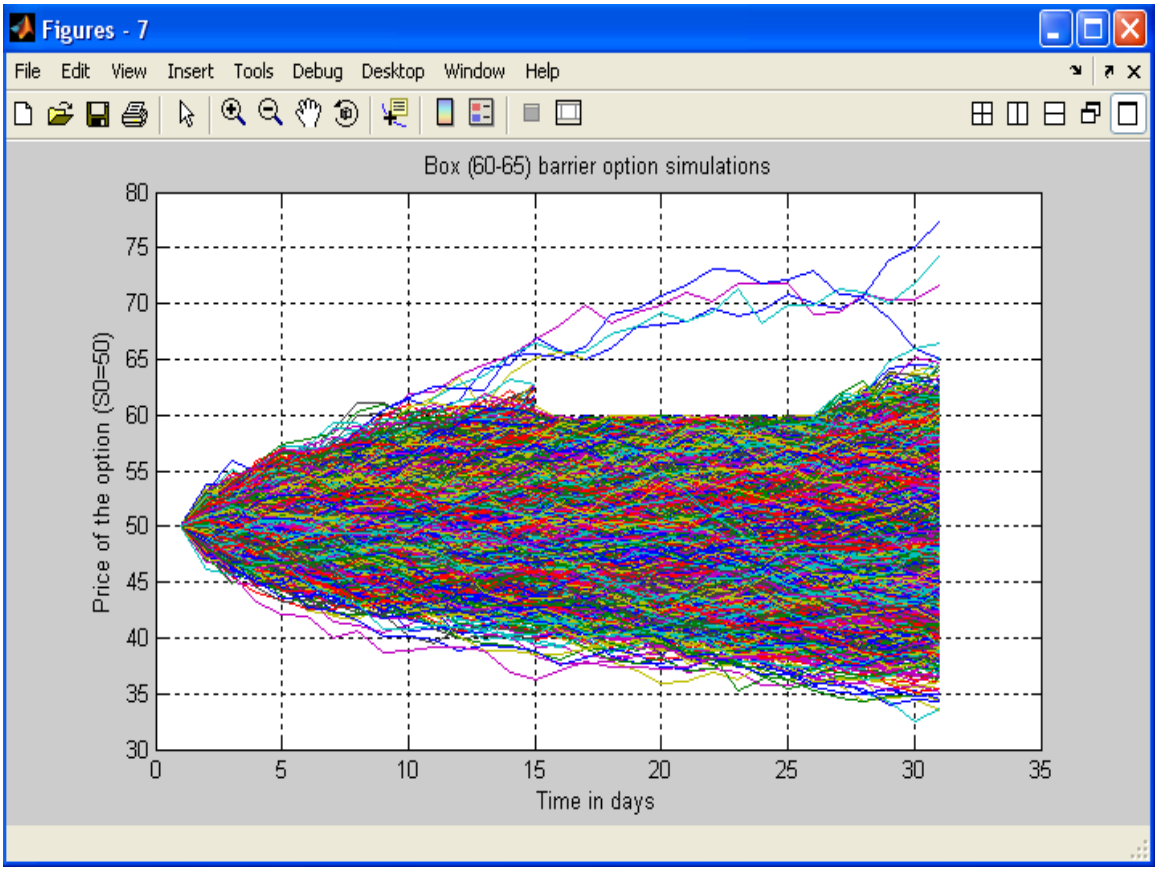


Table 5

Statistical results box simulation 60-65								
No: Sim	**10^4**	**10^5**	**$5*10^5$**	**10^6**	**$5*10^6$**	**10^7**	**$15*10^6$**	**$20*10^6$**
Mean	1.9606	1.9707	1.9718	1.9721	1.9720	1.9718	1.9720	1.9718
Std dev	0.02677	0.01197	0.00489	0.00243	0.00159	0.00072	0.00071	0.00052
Error	0.00600	0.00268	0.00110	0.000543	0.00036	0.00016	0.00016	0.00012
Conf Int	0.01173	0.00525	0.00214	0.00106	0.00070	0.00032	0.00035	0.00023
Conf range	1.9489; 1.9724	1.9649; 1.9739	1.9697; 1.9739	1.9711 1.9731	1.9713; 1.9727	1.9686; 1.9721	1.9717; 1.9723	1.9716; 1.9720

6. Conclusion

Our goal in this paper was to price knock out barrier options in a rectangular box form and compares the number of simulations needed to achieve the same accuracy it's with an ordinary option. This objective was accomplished by the use MATLAB code for pricing options using Monte Carlo simulation. It was remarkably easy to generate estimated option prices using the code and simulations although time consuming. Extending the basic technique from ordinary options to path dependent option helped us compare the accuracy of the two different options. The major finding in the paper was the fact that less number simulations are required when we have box.

Since we had our strike price at 50 that meant all prices below that 50 were zero, in both the small and big box this might have contributed to less number of simulations than the ordinary option.

The additional parameters might also have also led to the difference in number of simulations. The ability of Monte Carlo method to compute price of an option for a multiple parameters in a single simulation might have triggered price sensitivity, to the input parameters thereby reducing the number of simulations.

The time period could also be a factor although we are moving the box around we are still having the time fixed from 15^{th} to 25^{th}, thus our prices observed only during this time. On the other hand for the default option we have the whole period from the 1^{st} to the 30^{th}.

When we compared two boxes of the same size we noted that there was a difference in prices of 30 %. The (50-55) box has a price of 1.51 against (60-65) box which has a price of 1.97. What we have observed that the price of barrier option depends on the relative position of the box with respect to the underlying asset price. This was proved by the fact that despite having fewer trajectories the box (60-65) had a higher price compared to box (50-55).

7. References

[Baldi et al.,1999]P Baldi, L.Caramellino, and M.G.Iovina. Pricing general barrier options. A numerical approach using sharp large deviations. *Mathematical Finance*,9(4):293-322, 1999.
[Boyle,1977]P.P Boyle. Options: A monte carlo approach. *Journal of Financial Economics*,4:323-338,1977.
[Brandimarte,2002]P.Brandimarte *Numerical methods in finance, a MATLAB based introduction*.Wiley 2002.
[Glasserman,2004]P.Glasserman. *Monte Carlo Methods in Financial Engineering.* Springer, 2004
[Lundgren]R.Lundgren Monte Carlo studies of optimal domains knock out options.
[Merton, 1973]R.Merton *"Theory of Rational Option Pricing"*, Bell Journal of Economics & Management June 1973.

8. Appendix

MATLAB functions.

1. $randn - this\ \ generate\ random\ numbers\ and\ arrays.$
2. $cumsum - returns\ the\ cumulative\ sum\ of\ \ colums.$
3. $hist - \ \ will\ trace\ the\ simulations\ paths\ in\ constructing\ the\ histogram$

PayoffD1:

```
function PayoffD1=Default1(SO,r,sigma,T,NSteps,NRepl)
```

PayoffD1: This function gives us the matrix of asset paths where replications are row and columns which corresponds to the time changes for an ordinary option. Use this code to plot simulation paths and histograms, can easily plot the simulations up to 100 000 MATLAB version R2007a

```
    %SO is the intial price.
    SO=50;
    %sigma represents the volatility.
    sigma=0.3;
    %T represents the time horizons.
    T=1;
    %Riskfree interest rate.
    r=0.03;
    %The number of trading days in a year.
    NSteps=252;
    %NRepl is the number of replications simultaneously.
    NRepl=10000;
    %number of days (length of trajectories)
    Steps=30;
    %Strikeprice
    %K=50;
    %payoff function
    g=inline('max(50-x,0)','x');
    %dt represents the changes in time with respect to the trading
days.
    dt=T/NSteps;
    %nudt represents the drift.
    nudt=(r-0.5*sigma^2)*dt;
    sidt=sigma*sqrt(dt);
    %randn gives normally distributed random numbers and arrays.
    Increments=nudt+sidt*randn(Steps,NRepl);
    %Cumsum returns the cumulative sum of columns.
    Paths=cumsum(Increments);
    %Individual paths
    SPaths=SO*(exp(Paths));
    %Plot the individual paths starting at the initial stock price.
    H=[SO*ones(1,NRepl); SPaths];
    plot(H);
    grid
    %Ploting the histogram
    %hist(SPaths(15,:),100)
    %grid
    %payoff at expiery
    PayoffD1=exp(-r*dt*Steps)*mean(g(SPaths(end,:)));
%end
```

Inbox:

Inbox this defines our barrier option prices are found here by the MATLAB function find. If any of the conditions mentioned below are not met then an error message is obtained. Therefore this code is added to the payoffD1 and together they produce payoff function.

```
function I=Inbox(y1,y2,X)
if nargin==3
    if y1<y2
        I=find(X>=y1&X<=y2);
    else
        disp('y1 should be less than y2');
    end
else
    disp('There has to be 3 input parameters');
end
```

Payoff:

```
function payoff=GBMsim(SO,r,sigma,T,NSteps,NRepl)
```

Payoff: This function gives us the matrix of asset paths where replications are row and columns which corresponds to the time changes for a barrier (box rectangular) option. It works exactly in the same way as payoffD1 but for barrier options. It requires eleven parameters the first six being the default parameters and the last five defining our box. Use this code to plot simulation paths and histograms, can easily plot the simulations up to 100 000 MATLAB version R2007a

```
%if (nargin==6|nargin==11)
    %SO is the intial price.
    SO=50;
    %sigma represents the volatility.
    sigma=0.3;
    %T represents the time horizons in this case one year.
    T=1;
    %Riskfree interest rate.
    r=0.03;
    %The number of trading days in a year.
    NSteps=252;
    %NRepl is the number of replications simultaneously.
    NRepl=10000;
    %number of days(length of trajectory)
    Steps=30;
    %Strikeprice
    %K=50;
    %payoff function
    g=inline('max(50-x,0)','x');
    %Box Definition
    x1=15; y1=60; x2=25; y2=65;
    %dt represents the changes in time with respect to the trading
days.
    dt=T/NSteps;
    %nudt represents the drift.
    nudt=(r-0.5*sigma^2)*dt;
    sidt=sigma*sqrt(dt);
    %randn gives normally distributed random numbers and arrays.
    Increments=nudt+sidt*randn(Steps,NRepl);
    %Cumsum returns the cumulative sum of columns.
    Paths=cumsum(Increments);
    %Individual paths
    SPaths=SO*(exp(Paths));
    %if(nargin==11)
        for(i=x1:x2)
            I=Inbox(y1,y2,SPaths(i,:));
            SPaths(i:end,I)=NaN;
        end
    %end
    %Plot the individual paths
    H=[SO*ones(1,NRepl); SPaths];
    plot(H);
    grid
    %payoff at expiery
    payoff=exp(-r*dt*Steps)*mean(g(SPaths(end,:)));
%end
```

PayoffD2:

PayoffD2: This function was created in order to solve the memory problem faced by the computer the only thing that it adds is the loop otherwise it saves exactly the same purpose as PayoffD1 for all simulations above one million.

```
function PayoffD2=Default2(SO,r,sigma,T,NSteps,NRepl)
```
PayoffD2: This function gives us the matrix of asset paths where replications are row and columns which correspond to the time changes for an ordinary option over one million simulations.

```
    %SO is the initial price.
    SO=50;
    %sigma represents the volatility.
    sigma=0.3;
    %T represents the time horizons.
    T=1;
    %Riskfree interest rate.
    r=0.03;
    %The number of trading days in a year.
    NSteps=252;
    %NRepl is the number of replications simultaneously.
    NRepl=20000000;
    %Number of days(length of trajectory)
    Steps=30;
    %Strikeprice
    %K=50;
    % To save memory space we add this loop
    if NRepl>(5*10^5);
        ms=NRepl/(5*10^5);
        NRepl=(5*10^5);
        for v=1:ms
    %payoff function
    g=inline('max(50-x,0)','x');
    %dt represents the changes in time with respect to the trading
    days
    dt=T/NSteps;
    %nudt represents the drift.
    nudt=(r-0.5*sigma^2)*dt;
    sidt=sigma*sqrt(dt);
    %randn gives normally distributed random numbers and arrays.
    Increments=nudt+sidt*randn(Steps,NRepl);
    %Cumsum returns the cumulative sum of columns.
    Paths=cumsum(Increments);
    %Individual paths
    SPaths=SO*(exp(Paths)); %#ok<NASGU>
    %memory saving function shows the loop.
    memosav(v)=mean(g(SPaths(end,:)));  %#ok<AGROW>
        end
    end
    %payoff at expiery
    PayoffD2=exp(-r*dt*Steps)*mean(memosav);
%end
```

Payoff1:

```
function Payoff1=GBMsim1(SO,r,sigma,T,NSteps,NRepl)
```

Payoff1: This function was created in order to solve the memory problem faced by the computer the only thing that it adds is the loop otherwise it saves exactly the same purpose as Payoff for all simulations above one million. To move the box change the numbers under box definition.

```
   %if (nargin==6|nargin==11)
    %SO is the intial price.
    SO=50;
    %sigma represents the volatility.
    sigma=0.3;
    %T represents the time horizons in this case one year.
    T=1;
    %Riskfree interest rate.
    r=0.03;
    %The number of trading days in a year.
    NSteps=252;
    %NRepl is the number of replications simultaneously.
    NRepl=20000000;
    %number of days(length of trajectory)
    Steps=30;
    %Strikeprice
    %K=50;
    %payoff function
    % To save memory space we add this loop
    if NRepl>(5*10^5);
        ms=NRepl/(5*10^5);
        NRepl=(5*10^5);
        for v=1:ms
    g=inline('max(50-x,0)','x');
    %Box Definition
    x1=15; y1=48; x2=25; y2=52;
    %dt represents the changes in time with respect to the trading
    days
    dt=T/NSteps;
    %nudt represents the drift.
    nudt=(r-0.5*sigma^2)*dt;
    sidt=sigma*sqrt(dt);
    %randn gives normally distributed random numbers and arrays.
    Increments=nudt+sidt*randn(Steps,NRepl);
    %Cumsum returns the cumulative sum of columns.
    Paths=cumsum(Increments);
    %Individual paths
    SPaths=SO*(exp(Paths));
    %if(nargin==11)
        for(i=x1:x2)
            I=Inbox(y1,y2,SPaths(i,:));
            SPaths(i:end,I)=NaN;
    %The loop that allows to handle simulations of over 5*10^2
    memosav(v)=mean(g(SPaths(end,:)));
        end
        end
    end
    %payoff at expiery
    Payoff1=exp(-r*dt*Steps)*mean(memosav);
    %end
```

Table 6

Estimated price simulations of ordinary option (Default) & statistical results.								
Sample	**10^4**	**10^5**	**$5*10^5$**	**10^6**	**$5*10^6$**	**10^7**	**$15*10^6$**	**$20*10^6$**
1	1.9496	1.9671	1.9725	1.9758	1.9727	1.9718	1.9731	1.9714
2	1.9531	1.9633	1.9781	1.9674	1.9718	1.9703	1.9712	1.9719
3	1.9741	1.9525	1.9735	1.9698	1.9724	1.9728	1.9725	1.9728
4	1.9715	1.9671	1.9696	1.9723	1.9713	1.9709	1.9716	1.9717
5	2.0090	1.9802	1.9672	1.9733	1.9751	1.9729	1.9715	1.9719
6	1.9540	1.9756	1.9728	1.9778	1.9714	1.9738	1.9715	1.9718
7	2.0017	1.9763	1.9762	1.9710	1.9711	1.9721	1.9708	1.9730
8	2.0015	1.9697	1.9720	1.9784	1.9709	1.9717	1.9727	1.9719
9	2.0209	1.9712	1.9695	1.9737	1.9739	1.9723	1.9718	1.9727
10	2.0050	1.9746	1.9752	1.9701	1.9710	1.9733	1.9721	1.9716
11	1.9503	1.9754	1.9718	1.9745	1.9729	1.9706	1.9723	1.9725
12	1.9964	1.9819	1.9737	1.9707	1.9711	1.9705	1.9720	1.9735
13	1.9609	1.9858	1.9707	1.9719	1.9735	1.9718	1.9731	1.9714
14	1.9368	1.9876	1.9714	1.9711	1.9727	1.9711	1.9718	1.9723
15	1.9558	1.9819	1.9646	1.9674	1.9728	1.9733	1.9718	1.9717
16	2.0213	1.9660	1.9745	1.9684	1.9738	1.9709	1.9720	1.9726
17	1.9493	1.9833	1.9758	1.9742	1.9720	1.9705	1.9738	1.9713
18	1.8959	1.9765	1.9730	1.9729	1.9739	1.9716	1.9701	1.9707
19	1.9391	1.9634	1.9718	1.9692	1.9685	1.9718	1.9719	1.9717
20	2.0441	1.9858	1.9750	1.9699	1.9720	1.9727	1.9712	1.9721
Mean	1.97452	1.97422	1.97245	1.97199	1.97225	1.97184	1.97194	1.97203
Std dev	0.03562	0.00889	0.00308	0.00304	0.00143	0.00105	0.00084	0.00067
Error	0.00796	0.00199	0.00069	0.00068	0.00032	0.00024	0.00019	0.00015
Conf Int	0.01561	0.00390	0.00135	0.00133	0.00063	0.00055	0.00037	0.00030
Conf range	1.9589; 1.9901	1.9703; 1.9781	1.9711; 1.9738	1.9707; 1.9733	1.9606; 1.9726	1.9714; 1.9723	1.9716; 1.9723	1.9717; 1.9723

Table 7

Price simulation of box 48-52 and its statistical results.								
Sample	**10^4**	**10^5**	**$5*10^5$**	**10^6**	**$5*10^6$**	**10^7**	**$15*10^6$**	**$20*10^6$**
1	1.0201	1.0199	1.0122	1.0141	1.0165	1.0169	1.0182	1.0191
2	1.0541	1.0272	1.0132	1.0214	1.0161	1.0195	1.0164	1.0177
3	1.0767	1.0214	1.0164	1.0153	1.0192	1.0172	1.0171	1.0173
4	1.0685	1.0224	1.0217	1.0220	1.0185	1.0157	1.0175	1.0176
5	1.0332	1.0100	1.0222	1.0146	1.0163	1.0172	1.0168	1.0168
6	1.0001	1.0223	1.0155	1.0213	1.0178	1.0164	1.0177	1.0170
7	0.9956	1.0106	1.0173	1.0156	1.0165	1.0154	1.0182	1.0176
8	1.0209	1.0193	1.0236	1.0196	1.0171	1.0175	1.0175	1.0162
9	1.0255	1.0262	1.0221	1.0210	1.0197	1.0196	1.0173	1.0174
10	1.0337	1.0087	1.0208	1.0197	1.0170	1.0163	1.0182	1.0172
11	1.0327	1.0305	1.0161	1.0128	1.0172	1.0186	1.0172	1.0181
12	1.0669	1.0124	1.0223	1.0230	1.0176	1.0169	1.0165	1.0176
13	0.9833	1.0090	1.0198	1.0171	1.0198	1.0173	1.0177	1.0170
14	1.0278	1.0352	1.0143	1.0219	1.0172	1.0176	1.0165	1.0171
15	1.0015	1.0189	1.0205	1.0190	1.0173	1.0183	1.0177	1.0182
16	0.9786	1.0149	1.0098	1.0160	1.0161	1.0168	1.0174	1.0176
17	0.9917	1.0091	1.0160	1.0205	1.0189	1.0176	1.0167	1.0178
18	1.0007	1.0300	1.0185	1.0130	1.0187	1.0172	1.0178	1.0173
19	1.0360	1.0114	1.0200	1.0153	1.0158	1.0166	1.0173	1.0177
20	1.0359	1.0265	1.0212	1.0214	1.0198	1.0182	1.0185	1.0171
Mean	1.02417	1.01935	1.01823	1.01818	1.01766	1.01749	1.0175	1.01747
Std dev	0.02834	0.00827	0.00386	0.00336	0.00132	0.00102	0.00062	0.00059
Error	0.00634	0.00185	0.00075	0.00086	0.00030	0.00023	0.00014	0.00013
Conf Int	0.01242	0.00362	0.00147	0.00169	0.00058	0.00045	0.00027	0.00026
Conf Range	1.0118; 1.0366	1.0157; 1.0230	1.0168; 1.0197	1.0165; 1.0199	1.0171; 1.0182	1.0170; 1.0179	1.0172; 1.0178	1.0172; 1.0177

Table 8

Price simulation of box 45-55 and its statistical results.								
Sample	**10^4**	**10^5**	**$5*10^5$**	**10^6**	**$5*10^6$**	**10^7**	**$15*10^6$**	**$20*10^6$**
1	0.3096	0.313	0.3137	0.3181	0.3137	0.3153	0.3155	0.315
2	0.3107	0.3063	0.3176	0.3139	0.3144	0.3151	0.3152	0.3153
3	0.3387	0.3142	0.3173	0.3163	0.3159	0.315	0.315	0.3156
4	0.3314	0.3169	0.3125	0.3172	0.3152	0.3141	0.3148	0.3154
5	0.3034	0.3213	0.3168	0.3151	0.3154	0.3147	0.3154	0.315
6	0.3247	0.3203	0.3117	0.3146	0.3147	0.3145	0.3149	0.3148
7	0.3327	0.3104	0.3164	0.3141	0.3158	0.3149	0.316	0.3156
8	0.3346	0.3109	0.3182	0.3153	0.3162	0.3144	0.3153	0.3152
9	0.3007	0.3159	0.3171	0.3151	0.3161	0.3142	0.3151	0.3154
10	0.2999	0.3168	0.3134	0.3129	0.3149	0.3153	0.3152	0.3151
11	0.3167	0.3108	0.3187	0.3165	0.3142	0.3154	0.3154	0.3148
12	0.3026	0.3097	0.3151	0.3148	0.3152	0.3149	0.3157	0.315
13	0.3286	0.3118	0.3121	0.3135	0.3148	0.3153	0.3154	0.3153
14	0.3127	0.3227	0.3148	0.315	0.3145	0.3149	0.3152	0.3148
15	0.3296	0.3157	0.3181	0.3135	0.3154	0.3142	0.315	0.3151
16	0.3036	0.3165	0.3147	0.3163	0.3151	0.3147	0.3155	0.3141
17	0.3074	0.3139	0.3172	0.3176	0.3154	0.316	0.3146	0.3149
18	0.3023	0.3191	0.317	0.3145	0.3145	0.3153	0.3143	0.3147
19	0.3521	0.3222	0.3111	0.3157	0.3146	0.315	0.3152	0.3151
20	0.3366	0.315	0.3115	0.3147	0.3152	0.3148	0.3148	0.3152
Mean	0.3189	0.31517	0.31525	0.31524	0.31506	0.3149	0.3152	0.3151
Std dev	0.01570	0.00448	0.00252	0.00141	0.00065	0.00047	0.00039	0.00035
Error	0.00351	0.00100	0.00056	0.00032	0.00015	0.00010	0.000086	0.000075
Conf Int	0.00688	0.00196	0.00110	0.00062	0.00029	0.00021	0.00017	0.00015
Conf Range	0.3130; 0.3258	0.3132; 0.3172	0.3141; 0.3163	0.3146; 0.3159	0.3148; 0.3154	0.3147; 0.3151	0.3150; 0.3154	0.3150; 0.3153

Table 9

Price simulation of box 50-55 and its statistical results.								
Sample	**10^4**	**10^5**	**$5*10^5$**	**10^6**	**$5*10^6$**	**10^7**	**$15*10^6$**	**$20*10^6$**
1	1.5086	1.5155	1.5109	1.5173	1.5119	1.5109	1.5123	1.5113
2	1.5377	1.5078	1.5107	1.5098	1.512	1.5106	1.5115	1.5111
3	1.496	1.5155	1.5038	1.5104	1.5124	1.5113	1.5119	1.5123
4	1.4631	1.5154	1.5059	1.5105	1.5109	1.5126	1.5105	1.5115
5	1.5299	1.5124	1.5044	1.5159	1.535	1.5125	1.5118	1.5122
6	1.513	1.5165	1.5123	1.5145	1.5109	1.5126	1.511	1.511
7	1.5379	1.507	1.5062	1.5143	1.5116	1.5123	1.5119	1.5112
8	1.532	1.5049	1.5121	1.5105	1.5099	1.5114	1.5116	1.512
9	1.5201	1.5094	1.5022	1.5145	1.5138	1.5125	1.5123	1.511
10	1.5492	1.5158	1.5161	1.5084	1.5105	1.5131	1.5107	1.5125
11	1.5189	1.4963	1.5135	1.5071	1.5125	1.5116	1.5126	1.5121
12	1.5459	1.5332	1.5141	1.5109	1.51	1.5097	1.5124	1.5117
13	1.5398	1.5062	1.5112	1.5119	1.5127	1.5106	1.5113	1.5129
14	1.5249	1.5066	1.5109	1.5127	1.512	1.5124	1.5112	1.5113
15	1.5154	1.5114	1.53	1.5111	1.5122	1.51	1.5118	1.5111
16	1.5341	1.5231	1.5163	1.5139	1.5128	1.5102	1.5116	1.5117
17	1.5506	1.5042	1.5178	1.5146	1.5141	1.5124	1.5125	1.5116
18	1.512	1.508	1.5169	1.513	1.5082	1.5112	1.5105	1.51
19	1.5185	1.5117	1.5158	1.514	1.5118	1.5105	1.5108	1.5109
20	1.497	1.5119	1.5139	1.5108	1.5108	1.5097	1.5119	1.5108
Mean	1.5222	1.5116	1.5122	1.5123	1.5117	1.5115	1.5116	1.5115
Std dev	0.02110	0.00770	0.006229	0.002589	0.00144	0.00109	0.00066	0.00061
Error	0.00472	0.00172	0.00139	0.000580	0.00032	0.00024	0.00015	0.00013
Conf int	0.00925	0.00338	0.00273	0.001135	0.00063	0.00048	0.00029	0.00027
Conf Range	1.5213; 1.5232	1.5083; 1.5151	1.5096; 1.5150	1.5112; 1.5134	1.5111; 1.5124	1.5109; 1.5119	1.5113; 1.5119	1.5112; 1.5117

Table 10

Price simulation of box 60-65 and its statistical results.								
Sample	**10^4**	**10^5**	**$5*10^5$**	**10^6**	**$5*10^6$**	**10^7**	**$15*10^6$**	**$20*10^6$**
1	1.9641	1.9729	1.9701	1.9722	1.9746	1.9712	1.9713	1.9725
2	1.9791	1.952	1.9787	1.9713	1.9727	1.9726	1.9717	1.9716
3	1.9705	1.984	1.9734	1.9724	1.9721	1.9716	1.9716	1.9711
4	1.9709	1.9848	1.9709	1.9764	1.9693	1.971	1.9714	1.9714
5	1.971	1.976	1.9646	1.9713	1.9727	1.9718	1.9717	1.972
6	1.9271	1.9758	1.9746	1.9755	1.9741	1.9728	1.9729	1.971
7	1.9758	1.9591	1.9658	1.9739	1.9728	1.9722	1.9715	1.9726
8	1.9708	1.9715	1.9739	1.9731	1.9713	1.9715	1.9727	1.9716
9	1.9761	1.9899	1.9704	1.9725	1.9696	1.9702	1.9722	1.9719
10	1.9743	1.9675	1.9731	1.9726	1.9707	1.9711	1.9712	1.9711
11	1.9414	1.9693	1.9683	1.9702	1.9713	1.972	1.9713	1.9712
12	1.9083	1.9572	1.9661	1.9716	1.9724	1.9713	1.9738	1.9724
13	1.9537	1.963	1.9726	1.9707	1.972	1.9705	1.9716	1.9719
14	1.9209	1.9868	1.9862	1.9695	1.9692	1.9722	1.9718	1.9721
15	2.0146	1.9593	1.9754	1.9772	1.9714	1.9721	1.9731	1.9719
16	1.9171	1.9522	1.9701	1.9701	1.9708	1.9728	1.9715	1.9718
17	1.9976	1.9756	1.9699	1.9715	1.9724	1.9722	1.9731	1.9726
18	1.9643	1.9696	1.9738	1.9664	1.9742	1.9717	1.972	1.9711
19	1.9655	1.9597	1.9701	1.9715	1.9739	1.9725	1.9731	1.9721
20	1.9496	1.9881	1.9676	1.9725	1.9731	1.972	1.9712	1.9719
Mean	1.9606	1.9707	1.9718	1.9721	1.9720	1.9718	1.9720	1.9718
Std dev	0.02677	0.01197	0.00489	0.00243	0.00159	0.00072	0.00071	0.00052
Error	0.00600	0.00268	0.00110	0.000543	0.00036	0.00016	0.00016	0.00012
Conf Int	0.01173	0.00525	0.00214	0.00106	0.00070	0.00032	0.00035	0.00023
Conf Range	1.9489; 1.9724	1.9649; 1.9739	1.9697; 1.9739	1.9711 1.9731	1.9713; 1.9727	1.9686; 1.9721	1.9717; 1.9723	1.9716; 1.9720