Math 623 - Homework Assignment 1

Luying Qiu 167006765

February 4, 2016

1 Programming Results:

```
Call option price using closed-form formula = 9.05705
Put option price using closed-form formula = 15.6724

Call option price using single-step exact SDE solution = 9.16408
Put option price using single-step exact SDE solution = 15.6828

Call option price using Euler numerical solution of SDE for spot = 9.14494
Put option price using Euler numerical solution of SDE for spot = 15.5842

Call option price using Euler numerical solution of SDE for log spot = 9.2777
Put option price using Euler numerical solution of SDE for log spot = 15.6502

Call option price using Milstein numerical solution of SDE for spot = 9.00096
Put option price using Milstein numerical solution of SDE for spot = 15.7338

Program ended with exit code: 0
```

2 Explanation of the algorithms

(a) Closed-form formula

We use the Black Scholes-Merton function with dividend here:
$C(S_t, t) = S_t e^{-d(T-t)} * N(d_1) - Ke^{-r(T-t)} * N(d_2)$
$P(S_t, t) = -S_t e^{-d(T-t)} * N(-d_1) + Ke^{-r(T-t)} * N(-d_2)$
where
$d_1 = \frac{log(\frac{S_t}{K})+(r+\frac{\sigma^2}{2})(T-t)}{\sigma\sqrt{T-t}}$
$d_2 = d_1 - \sigma\sqrt{T-t}$
and $N(d) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{d} e^{-\frac{x^2}{2}}dx$
Set $t = 0$ to implement the functions, we can get the option prices.

(b) Analytical solution and Monte Carlo simulation

For the model: $dS(t) = S(t)((r-d)dt + \sigma dW(t))$,
the Black-Scholes pricing theory tells us that the price of a vanilla option, with expiry T and payoff f, is equal to $e^{-rT}\mathbb{E}(f(S_T))$.
Now we solve the model by passing to the log and using Ito's lemma; we compute

$$dlogS_t = (r - d - \frac{1}{2}\sigma^2)dt + \sigma dW_t.$$

Combine with the fact that W_t is a Brownian motion, W_T is distributed as a Gaussian with mean zero and variance T, the solution is:

$$S_T = S_0 e^{(r-d-\frac{1}{2}\sigma^2)T+\sigma\sqrt{T}N(0,1)}.$$

So the price of the option is: $e^{-rT}\mathbb{E}(f(S_0 e^{(r-d-\frac{1}{2}\sigma^2)T+\sigma\sqrt{T}N(0,1)}))$.
To implement the Monte Carlo simulation, we draw a random variable, x , from an $N(0,1)$ distribution and compute $f(S_0 e^{(r-d-\frac{1}{2}\sigma^2)T+\sigma\sqrt{T}x})$,
where $f(S)=(S-K)_+$ for call option and $f(S)=(K-S)_+$ for put option. Using the fact that $\frac{1}{N}\sum_{i=1}^N f(S_{iT}) \to \mathbb{E}(f(S_T))$, we do this many times and take the average, then multiply this average by e^{-rT} to get the option price.

(c) Euler numerical solution of SDE for spot

For the model: $dS(t)=S(t)((r-d)dt+\sigma dW(t))$,
the Euler numerical solution is:

$$S_{i+1}=S_i+S_i(r-d)\Delta t+S_i\sigma\Delta W_t.$$

Note that ΔW_t is distributed as a Gaussian with mean zero and variance Δt,
We have

$$S_{i+1}=S_i+S_i(r-d)\Delta t+S_i\sigma\sqrt{\Delta t}N(0,1).$$

Set the number of steps to 252, we can calculate S_{252} step by step from S_0, using the fact that $\frac{1}{N}\sum_{i=1}^N f(S_{iT}) \to \mathbb{E}(f(S_T))$ again, We do this many times and take the average, then multiply this average by e^{-rT} to get the option price.

(d) Euler numerical solution of SDE for log spot

For the model: $dlogS(t)=(r-d-\frac{\sigma^2}{2})dt+\sigma dW(t)$,
the Euler numerical solution is:

$$logS_{i+1}=logS_i+(r-d-\frac{\sigma^2}{2})\Delta t+\sigma\Delta W_t.$$

Note that ΔW_t is distributed as a Gaussian with mean zero and variance Δt,
We have

$$logS_{i+1}=logS_i+(r-d-\frac{\sigma^2}{2})\Delta t+\sigma\sqrt{\Delta t}N(0,1).$$

Set the number of steps to 252, we can calculate $logS_{252}$ step by step from $logS_0$, then calculate S_{252} by $S_{252}=e^{logS_{252}}$. Using the fact that $\frac{1}{N}\sum_{i=1}^N f(S_{iT}) \to \mathbb{E}(f(S_T))$ again, We do this many times and take the average, then multiply this average by e^{-rT} to get the option price.

(e) Milstein numerical solution of SDE for spot

For the model: $dS(t)=S(t)((r-d)dt+\sigma dW(t))$,
the Milstein numerical solution is:

$$S_{i+1}=S_i+S_i(r-d)\Delta t+S_i\sigma\Delta W_t+\tfrac{1}{2}\sigma^2 S_i(\Delta W_t^2-\Delta t).$$

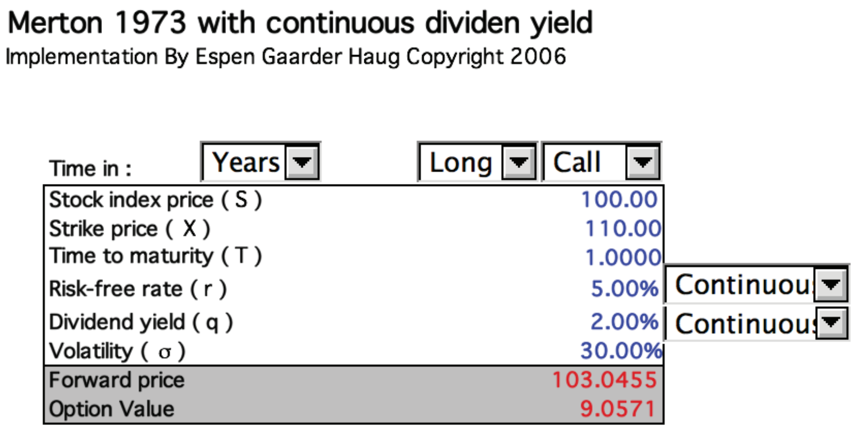
Note that ΔW_t is distributed as a Gaussian with mean zero and variance Δt,
We have

$$S_{i+1}=S_i(1+(r-d-\tfrac{1}{2}\sigma^2)\Delta t+\sigma\sqrt{\Delta t}Z+\tfrac{1}{2}\sigma^2\Delta t Z^2),$$

where $Z\sim N(0,1)$.
Set the number of steps to 252, we can calculate S_{252} step by step from S_0, using the fact that $\frac{1}{N}\sum_{i=1}^N f(S_{iT}) \to \mathbb{E}(f(S_T))$ again, We do this many times and take the average, then multiply this average by e^{-rT} to get the option price.

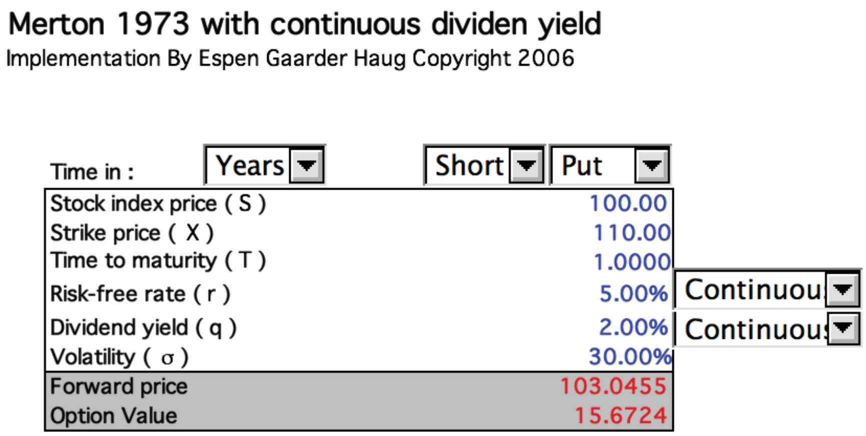
(f) Benchmarking

Using the Excel-VBA spreadsheet of Haug(Merton 1973 with continuous dividend yield) to benchmark, we get the following results:



Merton 1973 with continuous dividen yield
Implementation By Espen Gaarder Haug Copyright 2006

Time in :	Years	Long	Call
Stock index price (S)	100.00		
Strike price (X)	110.00		
Time to maturity (T)	1.0000		
Risk-free rate (r)	5.00%	Continuou	
Dividend yield (q)	2.00%	Continuou	
Volatility (σ)	30.00%		
Forward price	103.0455		
Option Value	9.0571		



Merton 1973 with continuous dividen yield
Implementation By Espen Gaarder Haug Copyright 2006

Time in :	Years	Short	Put
Stock index price (S)	100.00		
Strike price (X)	110.00		
Time to maturity (T)	1.0000		
Risk-free rate (r)	5.00%	Continuou	
Dividend yield (q)	2.00%	Continuou	
Volatility (σ)	30.00%		
Forward price	103.0455		
Option Value	15.6724		

The results shows that option prices given by closed-form formula is closet to the benchmark and the prices given by the other four solutions are very close.

(g) Is it necessary to simulate entire paths?

The option is path-independent, it is unnecessary to simulate entire paths, since the payoff function only depends on the terminal stock price, and increasing the number of steps doesn't improve the precision of the price much. However, for the European-style Asian option, it is path-dependent, so it is necessary to simulate entire paths since the terminal stock price cannot give us the payoff.

3 Error Analysis

We use call option prices as example to analyze the errors, the error analysis for put option prices is similar. Fixing the number of steps to 252, we change the number of paths from 2500 to 88818, increasing exponentially, with each path, we run 100 times and get 100 different prices, then calculate the standard deviation. Finally, plot the logarithm of the standard deviations with respect to the logarithm of their corresponding number of paths, the results are as follows:

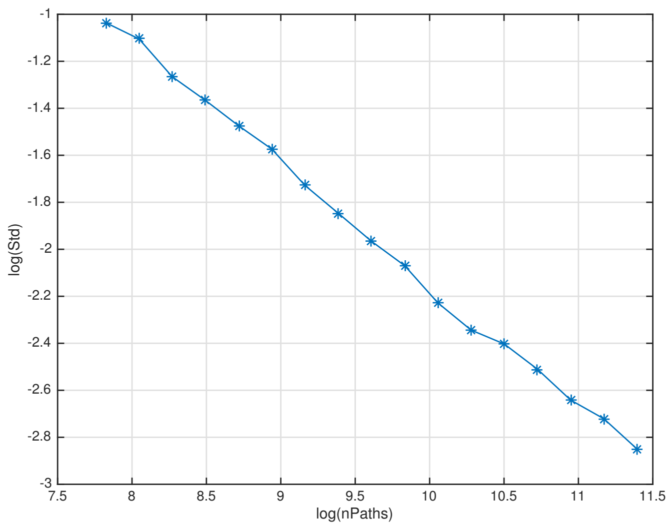


Figure 1: Errors-Paths for analytical solution

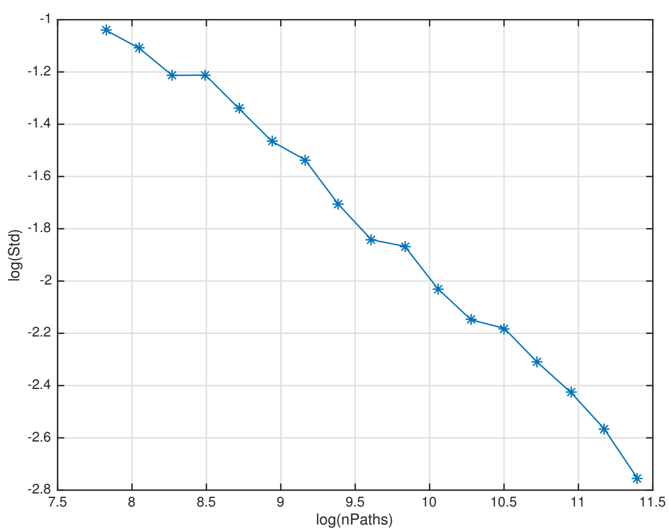


Figure 2: Errors-Paths for Euler numerical solution of SDE for spot

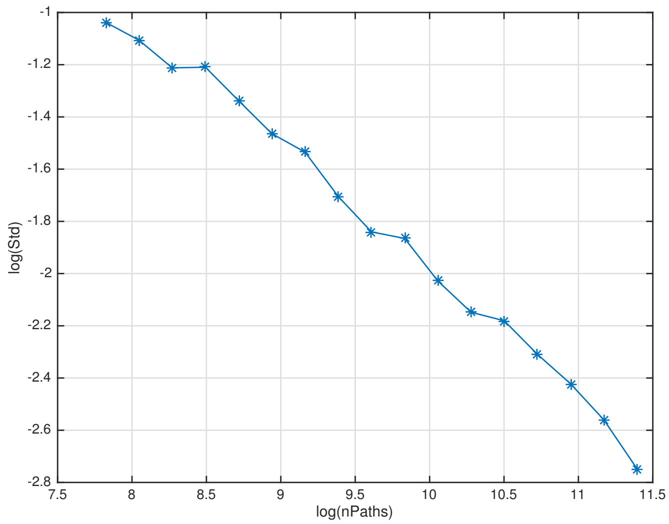


Figure 3: Errors-Paths for Euler numerical solution of SDE for log spot

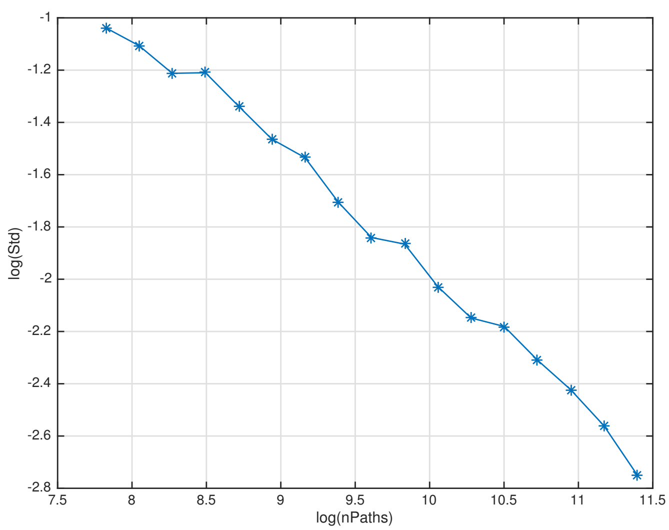


Figure 4: Errors-Paths for Milstein numerical solution of SDE for spot

The slops are close to $\frac{1}{2}$ in each picture, and the figures are quite similar to each other, so we can conclude that in each solution, the estimation of price converges as the number of paths increases.

Now we turn to analyze the convergence with the change of number of steps. Fixing the number of paths to 10000, we change the number of steps from 161.28 to 616, increasing exponentially, with each path, we run 100 times and get 100 different prices, then calculate the standard deviation. Finally, plot the logarithm of the standard deviations with respect to the logarithm of their corresponding number of steps, the results are as follows:

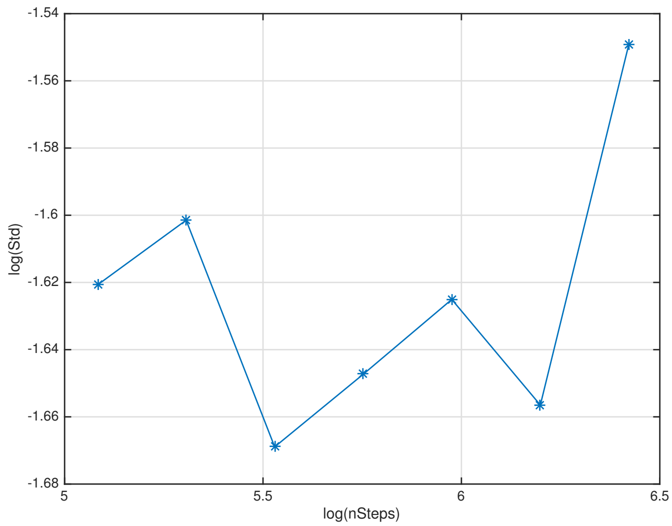


Figure 5: Errors-Steps for Euler numerical solution of SDE for spot

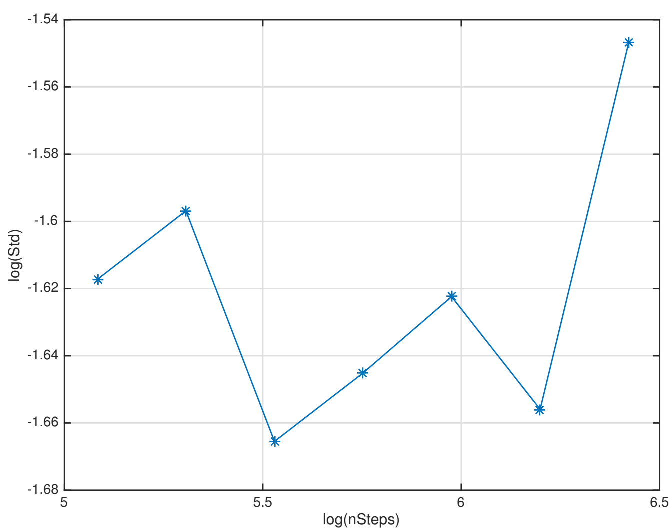


Figure 6: Errors-Steps for Euler numerical solution of SDE for log spot

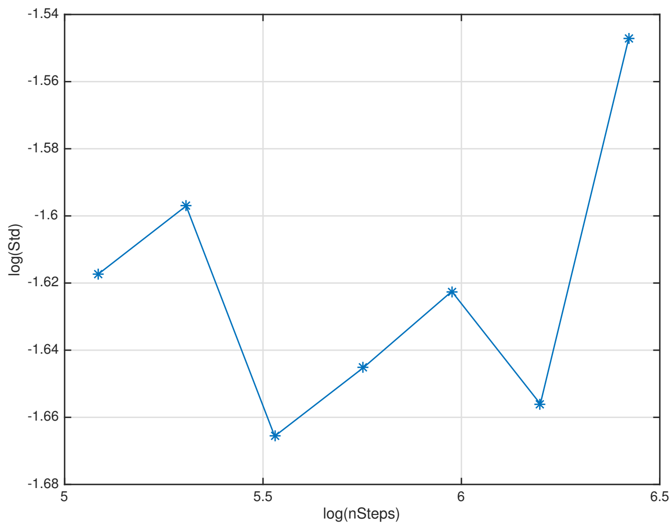


Figure 7: Errors-Steps for Milstein numerical solution of SDE for spot

The figures are quite similar to each other also, however, they don't indicate any trend of convergence this time. So we can conclude that the number of steps has little influence on the estimation of the option price, which agrees with the answer to question (g) above.