

## Report for HW1

### Part 1 Implementation of algorithm:

For the a) & b) question, I directly applied the Black-Scholes formula function from the Joshi textbook. a) has nothing to say, they just used the Black-Scholes formula. What they did in b) is that they applied  $S(t) = S(0) \cdot \exp((r \cdot t - 1/2 \cdot \sigma^2 \cdot t))$  and I repeat it for 10000 times to get the mean.

For c), I write a method called

```
"double MilsteinScheme1(double,double,double,int,double,double,double,double  
(*)(double,double,double),double (*)(double,double),double (*)(double)) ;".
```

This function basically pass through all the parameters I need from the main function and give reference to the different version of payoff function for spot and log spot,  $a(X(t),t)$  and  $b(X(t),t)$  in the textbook. However, in our question, the parameter passed through are not just those two variables. So, my functions are not using the exact  $X(t)$  and  $t$ .

```
double Functiona(double x,double r,double d) & Functionb(double x,double sigma)
```

are for spot one and

```
double Functiona_log(double/*simply fit the signiture of EulerScheme1*/  
sigma,double r,double d) & double Functionb_log(double null,double sigma)
```

are for the log spot one. The payoff function are adjusted accordingly as well. The Boolean variable at the end of the EulerScheme1 function is used to flag which set of functions are used.

The Milstein scheme is applied in similar way. The only new thing is the derivative function- $b'(X(t),t)$  in the slides. I built a new function for that.

The main function repeat the loop of 10000 iterations and get outputs aligned out.

I want to first compare my results. It seems that the Euler simulation of spot is giving a closer result than others. The Euler log spot is giving a higher value than Euler sport algorithm. The Milstein method is adding another term to it. The sign of this term depends on  $(W(t(i))-W(t(i-1)))^2 - (t(i)-t(i-1)))$ . It seems that over 10000 repetitions, it consistently shows higher value than Euler of the same underlying model.

### Part 2 Benchmark:

	My Code	Haug's VBA Code
Exact Solution	9.05705	<b>9.0571</b>
One step MC simulation	9.25303	<b>9.0470</b>
Euler simulation	9.01504	NA

of spot		
Milstein simulation of log spot	9.36136	NA
Euler simulation of log spot	9.1209	NA

The comparison shows that the MC simulation methods are close but not exactly. I tried to make number of iteration even larger. The number tends to be more accurate result than this version. It is really a tradeoff between time and accuracy.

### Part 3 Path Dependence :

Given that the option value depends only on the final result indicates that the only thing matters the ending value. So, we don't need to generate the whole path at all. In fact, one step Monte Carlo simulation is producing a more accurate result here than Milstein scheme though Milstein promises a better order of convergence than Euler scheme.

The assignment doesn't ask me to program a Asian option case. So, I simply apply the Excel code of Broadie-Glasserman excel code for American option from Haug. The similar situation given 2 opportunities to exercise the price is 7.78264, while given 3 opportunities to exercise, the price changes to 8.17998. So, it is possible that given more chances to monitor the process, higher values can appear. If one average them just like in the Asian option case, the value may changes. As the number of chances allow to monitor increases, the value of option should increase because between any two adjacent monitor points, a higher value may exists. So, for an Asian option, the path is important and therefore need to be generated completely.