**Answers of Group C**

Hanlin WANG

**a) Compile and run the program**

My program runs well in this figure:

文本

描述已自动生成

I divided the time into 100 subintervals with the same size from t = 0 to t = T (expired date). I can get the situation of the specific time t from the situation one step before.

During every single simulation, I have a sample path of the evolution of the stock price which is an Ito process. At time T, I get the payoff under this simulation. And I choose to simulate it for 50,000 times. Therefore, I get 50,000 prices and calculate their mean number. At last, I discount the mean of those prices to get the option price now. When the program is running, I can know the progress by printing out the times of simulations every 10,000 times. Also, I can check how many times the stock price hit 0.

**b) Batch 1 and 2 experiment**

Here is my experiment of batch 1 and 2. The line marked in red is the simulation with the least difference with the exact solution.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **NT** | **NSIM** | **Price** | **Difference** |
| **batch 1: call** | 100 | 50000 | 2.11675 | 0.01662 |
| 200 | 50000 | 2.14956 | 0.01619 |
| 300 | 50000 | 2.16432 | 0.03095 |
| 400 | 50000 | 2.15482 | 0.02145 |
| 500 | 50000 | 2.16105 | 0.02768 |
| **batch 1: put** | 100 | 10000 | 5.90807 | 0.06179 |
| 100 | 20000 | 5.90728 | 0.06100 |
| 100 | 30000 | 5.90512 | 0.05884 |
| 100 | 40000 | 5.87041 | 0.02413 |
| 100 | 50000 | 5.87749 | 0.03121 |
| **batch 2: call** | 100 | 10000 | 7.94097 | 0.02460 |
| 100 | 20000 | 7.78593 | 0.17964 |
| 100 | 30000 | 7.81810 | 0.14747 |
| 200 | 30000 | 7.95113 | 0.01444 |
| 300 | 30000 | 7.97197 | 0.00640 |
| **batch 2: put** | 100 | 10000 | 8.06336 | 0.09779 |
| 200 | 10000 | 8.13569 | 0.17012 |
| 300 | 10000 | 8.19128 | 0.22571 |
| 300 | 20000 | 8.14435 | 0.17878 |
| 300 | 30000 | 8.10626 | 0.14069 |

Above all, though the most accurate price was not always derived from the simulation with the largest NT and NSIM due to the roud-off error and the simulation’s particularity, the basic law as following of the simulation does not change. With NSIM fixed, when NT get larger, the numerical price will be more accurate because more finely divided discrete time result in a more continuous time series. While with the NT fixed, when NSIM get larger, the numerical price will be more accurate according to the Law of Large Number.

**c) Stress-testing of MC method**

First of all, I want to declare that finding the NT and NSIM with an accuracy to two placecs behind the decimal point is really a difficult work! I’ll show my effort in the following table.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **NT** | **NSIM** | **Price** | **Difference** |  | **NT** | **NSIM** | **Price** | **Difference** |
| **batch 4: call** | 100 | 10000 | 88.04120 | 4.13450 | **batch 4: put** | 100 | 10000 | 1.32196 | 0.07446 |
| 200 | 10000 | 90.48100 | 1.69470 | 200 | 10000 | 1.30484 | 0.05734 |
| 300 | 10000 | 94.33650 | 2.16080 | 300 | 10000 | 1.29393 | 0.04643 |
| 400 | 10000 | 91.33240 | 0.84330 | 400 | 10000 | 1.28359 | 0.03609 |
| 500 | 10000 | 95.40140 | 3.22570 | 500 | 10000 | 1.30399 | 0.05649 |
| 600 | 10000 | 95.99900 | 3.82330 | 600 | 10000 | 1.28915 | 0.04165 |
| 700 | 10000 | 95.41050 | 3.23480 | 700 | 10000 | 1.29886 | 0.05136 |
| 400 | 20000 | 90.69010 | 1.48560 | 400 | 20000 | 1.29779 | 0.05029 |
| 400 | 30000 | 91.57150 | 0.60420 | 400 | 30000 | 1.28278 | 0.03528 |
| 400 | 40000 | 92.20450 | 0.02880 | 400 | 40000 | 1.27078 | 0.02328 |
| 400 | 50000 | 92.00190 | 0.17380 | 400 | 50000 | 1.26830 | 0.02080 |
| 400 | 60000 | 92.46160 | 0.28590 | 400 | 60000 | 1.26473 | 0.01723 |
| 400 | 70000 | 92.90500 | 0.72930 | 400 | 70000 | 1.26309 | 0.01559 |
| 400 | 80000 | 93.42240 | 1.24670 | 400 | 80000 | 1.26734 | 0.01984 |
| 1000 | 100000 | 92.70010 | 0.52440 | 400 | 100000 | 1.26066 | 0.01316 |
| 5000 | 100000 | 91.90420 | 0.27150 | 400 | 200000 | 1.26254 | 0.01504 |

In terms of the call pricing, I started with 100 NT and 10000 NSIM, then I increased the number of NT and got better accuracy of the price until NT was bigger than 400. So I fixed NT on 400 and tried to unleash the power of the Law of Large Numbers. By increasing the number of NSIM, I did get more accurate price at the beginning. However, when NSIM was larger than 40,000, the accuracy got worse and worse. I was confused and tried another two groups NT and NSIM, but the best accuracy didn’t come back even NT and NSIM came really large. Therefore, my best NT and NSIM for pricing call option of batch 4 is 400 and 40,000.

In terms of the put option, things seemed better. Like the situation of call option, accuracy get worse when NT was larger than 400. Then I fixed NT on 400 and increased NSIM and get the best accuracy when NSIM was 100,0000. So my best NT and NSIM for pricing put option of batch 4 is 400 and 100,000.

Unfortunately, even I did some more attempts, I still didn’t get the NT and NSIM with an accuracy to two placecs behind the decimal point but only one place. I’m sorry about that and hope my description has shown my understanding of the MC method.