Group C

1. Study the source code in the file TestMC.cpp and relate it to the theory that we have just discussed. The project should contain the following source files and you need to set project settings in VS to point to the correct header files: Compile and run the program as is and make sure there are no errors.

Ans:

The project, TestMC.cpp, aims to determine the option value through Monte Carlo simulation. The Stochastic Differential Equation (SDE) function simulates the stock pattern from time = 0 to time = T with a time step (NT) defined by the user. The main function incorporates a for loop to simulate this process NSIM times, a parameter also defined by the user. During each simulation, the stock price at time T is obtained, the payoff at time T is calculated, summed, and averaged. The simulation relies on a random number generator, specifically the "lagged\_fibonacci607" generator, with the random variable following a normal distribution.

The average is then discounted to time t=0 to yield the final result. However, due to the reliance on random numbers, each simulation may yield different results.

1. Run the MC program again with data from Batches 1 and 2. Experiment with different value of NT (time steps) and NSIM (simulations or draws). In particular, how many time steps and draws do you need in order to get the same accuracy as the exact solution? How is the accuracy affected by different values for NT/NSIM?

Ans:

Batches 1

Call: The exact solution for this call is 2.13337. When I tried (NT = 500, NSIM = 1,000,000), I got call = 2.13071 (difference is .00266). When I tried (NT = 500, NSIM = 5,000,000), I got call = 2.13394 (difference is .00057). When I tried (NT = 1000, NSIM = 1,000,000), I got call = 2.13456(difference is .00119).

Put: The exact solution for this put is 5.84628. When I tried (NT = 500, NSIM = 1,000,000), I got put = 5.84125 (difference is .00503). When I tried (NT = 500, NSIM = 5,000,000), I got put = 5.84145 (difference is .00483). When I tried (NT = 1000, NSIM = 1,000,000), I got put = 5.83585(difference is .01043).

Batches 2

Call: The exact solution for this call is 7.96557. When I tried (NT = 500, NSIM = 1,000,000), I got call = 7.96142 (difference is .00415). When I tried (NT = 500, NSIM = 5,000,000), I got call = 7.97014 (difference is .00457). When I tried (NT = 1000, NSIM = 1,000,000), I got call = 7.97529(difference is .00972).

Put: The exact solution for this call is 7.96557. When I tried (NT = 500, NSIM = 1,000,000), I got put = 7.95663 (difference is .00894). When I tried (NT = 500, NSIM = 5,000,000), I got put = 7.95826 (difference is .00731). When I tried (NT = 1000, NSIM = 1,000,000), I got put = 7.94982(difference is .01575).

In general, the higher the values of NT and NSIM, the greater the accuracy achieved. For instance, in Batches 1's call, I utilized NT = 500 and NSIM = 5 million to attain accuracy up to three decimal places. To match the accuracy of the exact solution, it seems necessary for NSIM to exceed 5 million, or alternatively, the value of NT should be increased. However, efficiency concerns arise as the numbers escalate. The simulation with NT = 500 and NSIM = 5 million took over 5 minutes to complete. It is a trade-off between precision and computational efficiency.

1. Now we do some stress-testing of the MC method. Take Batch 4. What values do we need to assign to NT and NSIM in order to get an accuracy to two places behind the decimal point? How is the accuracy affected by different values for NT/NSIM?

Ans:

Call:

Exact solution is 92.1757

|  |  |  |  |
| --- | --- | --- | --- |
| NT | NSIM | Result | Difference |
| 100 | 10,000,000 | 89.299 | 2.8767 |
| 500 | 10,000 | 95.4014 | 3.2257 |
| 500 | 50,000 | 94.3209 | 2.1452 |
| 500 | 100,000 | 93.4016 | 1.2259 |
| 500 | 500,000 | 91.858 | 0.3177 |
| 500 | 1,000,000 | 91.845 | 0.3307 |
| 500 | 5,000,000 | 91.7312 | 0.4445 |
| 1,000 | 10,000 | 89.7103 | 2.4654 |
| 1,000 | 50,000 | 93.5268 | 1.3511 |
| 1,000 | 100,000 | 92.7001 | 0.5244 |
| 1,000 | 500,000 | 91.9841 | 0.1916 |
| 1,000 | 1,000,000 | 91.5646 | 0.6111 |

Put:

Exact solution is 1.2475

|  |  |  |  |
| --- | --- | --- | --- |
| NT | NSIM | Result | Difference |
| 500 | 10,000 | 1.30399 | 0.05649 |
| 500 | 50,000 | 1.25449 | 0.00699 |
| 500 | 100,000 | 1.25376 | 0.00626 |
| 500 | 500,000 | 1.25868 | 0.01118 |
| 500 | 1,000,000 | 1.25458 | 0.00708 |
| 500 | 5,000,000 | 1.25478 | 0.00728 |
| 1,000 | 10,000 | 1.2856 | 0.0381 |
| 1,000 | 50,000 | 1.24091 | 0.00659 |
| 1,000 | 100,000 | 1.2599 | 0.0124 |
| 1,000 | 500,000 | 1.25077 | 0.00327 |
| 1,000 | 1,000,000 | 1.24861 | 0.00111 |

For a call option, the simulation time needs to exceed 5 million in order to achieve accuracy up to two decimal places. Alternatively, the number of time steps (NT) should be increased to more than 1000.

As for a put option, an accuracy meeting our goal is attained when the simulation time reaches 50,000 with NT set at 1000. Similarly, reaching a simulation time of 10,000 with NT set at 500 also achieves our accuracy goal.

In general, increasing both NT and NSIM results in greater accuracy. Keeping NT at the same level, a higher NSIM tends to outperform a lower NSIM. In other words, increasing the number of simulations enhances the performance, contributing to more robust and reliable results. Similar trends are observed with NT. However, it's important to note that as NT increases, the time required to obtain the stock price at time T extends. This implies that, at the same NSIM level, higher NT values take longer to yield results. However, the trade-off is that the results are generally more precise compared to simulations with lower NT values.