Part3\_Question1

LujiaYangMMF1914H

Proof:

, and

Note that

* ; since
* since

Therefore, for to be satisfied, then

Part3\_Question2 [RELATED FILE: Part3Q2.m, P3Q2\_CRROptionPricer.m]

LujiaYangMMF1914H

For Euro Barrier Call and Put, tested if the adjusted function works:

**Euro Barrier Call**

MATLAB output

Price Delta Gamma Theta

\_\_\_\_\_\_ \_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_

0.5721 0.069393 -0.033803 0.0006257 % initial test

1.7762 0.45124 0.0087114 -0.001184 % change total steps and barrier

3.7015 0.13682 -0.16951 0.0038461 % change total steps and strike

0 0 0 0 % test when barrier < strike [Error-condition violate]

0 0 0 0 % test when barrier = strike [Error-condition violate]

**Euro Barrier Put**

MATLAB output

Price Delta Gamma Theta

\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_

1.6129 -0.36632 0.068424 -0.0016261. % initial test

1.6661 -0.36349 0.063074 -0.0013864. % change total steps and barrier

0.32398 -0.1031 0.029659 -0.00099372. % change total steps and strike

0 0 0 0 % test when barrier > strike [Error-condition violate]

0 0 0 0 % test when barrier = strike [Error-condition violate]

Part3\_Question3

LujiaYangMMF1914H

For path-dependent option, Monte Carlo method is appropriate. It allows pricing of options with very complicated features and required when the options have path dependent features.

It has much slower to converge than other method. Monte Carlo methods are computational algorithms that rely on repeated random sampling to obtain numerical results. Other than the Knock – Out option, this can also be used for Asian options, “Real” options, etc..

Part3\_Question4

LujiaYangMMF1914H

[RELATED FILE: Part3Q4.m, P3Q4\_CRROptionPricer.m, P3Q4\_CRROptionPricerForLoops.m]

Plot the Price and difference against steps, for both vector version and loop version:

* Steps and price difference:

Based on the MATLAB output and the “difference vs. steps” plot, we find that for 100 steps, the price difference is 0.0158, and for 120 steps, the price difference is 0.0058. Therefore, we roughly conclude that 120 steps are required before the calculated price converges to within $0.01.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Steps | 20 | 40 | 60 | 80 | 100 | **120** | 140 | … |
| Vector\_  price\_difference | 0 | -0.0752 | -0.0393 | 0.0107 | 0.0158 | **0.0058** | 0.0010 | … |
| Loop\_  price\_difference | 0 | -0.0752 | -0.0393 | 0.0107 | 0.0158 | **0.0058** | 0.0010 | … |

* The “price vs steps” and “difference vs, steps” for both versions are similar. However, the elapsed time for vector version and loop version are significantly different.

Graphical user interface, chart

Description automatically generated

Based on the plot, we observed that the elapsed time for loops version increase significantly when the lattice steps increase. Conversely, the elapsed time for vector version only experienced small increase as the lattice steps increase. This is because of the nested for loop in the “CRROptionPricerForLoops” function.

Chart, line chart

Description automatically generated