A1.3)

iii)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N | 50 | 500 | 5000 | 10000 |
| Binomial Tree – Lookback Floating Strike European Call,t0 cfl\_bm(0.02,0.01,5,0.35,0.5,N) | 0.8515 | 0.8968 | 0.9119 | 0.9139 |
| Closed Form – Lookback Floating Strike European Call,t0 cfl(0.02,0.01,5,5,0.35,0.5) | 0.9189 | 0.9189 | 0.9189 | 0.9189 |

Comment

As the number of iterations increases the value obtained from the binomial method converges towards the value produced by the closed formula in i). However, the convergence rate decreases as the number of iterations increases.

iv)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N | 50 | 500 | 5000 | 10000 |
| Binomial Tree – Lookback Floating Strike European Call,tn cfl\_bm3(0.02,0.01,5,4.75,0.35,0.5,N) | 0.8838 | 0.9189 | 0.9311 | 0.9328 |
| Closed Form – Lookback Floating Strike European Call,tn cfl(0.02,0.01,5,4.75,0.35,0.5) | 0.9369 | 0.9369 | 0.9369 | 0.9369 |

Comment

The option price obtained from the binomial method is quite close to the price obtained from the closed formula. The price could possibly be closer if a quadratic interpolation is used instead. Also, note the slowing rate of increment as the number of iterations increases.

v)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N | 50 | 500 | 5000 | 10000 |
| Binomial Tree – Lookback Floating Strike American Call,tn cfl\_bm\_am(0.02,0.01,5,4.75,0.35,0.5,N) | 0.883789 | 0.918903 | 0.931083 | 0.932772 |
| Binomial Tree – Lookback Floating Strike European Call,tn cfl\_bm3(0.02,0.01,5,4.75,0.35,0.5,N) | 0.883787 | 0.918901 | 0.931081 | 0.932769 |

Comments

The values of the American option are very close to the European ones. This could be because that there is not much additional value to exercise the option early as the payoff is: max(S-Smin,0).

A 1.4)

i)

|  |  |  |  |
| --- | --- | --- | --- |
| N | 5 | 10 | 15 |
| Binomial Tree – Floating Strike Asian Arithmetic-average European Put aafsput(0.03,0.02,7.5,0.45,0.25,N) | 0.3776 | 0.3781 | 0.3790 |
| Running Time (s) | 0.001077 | 0.000706 | 0.004672 |

ii)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| N | 5 | 10 | 15 | 100 | 200 |
| Linear Interpolation (Running Time, s) fsg\_aafsput\_linear(0.03,0.02,7.5,0.45,0.25,N,0.5) | 0.3793 (0.002497) | 0.3828 (0.001574) | 0.3839 (0.003130) | 0.3840 (0.255384) | 0.3834 (1.710569) |
| Quadratic Interpolation (Running Time, s) fsg\_aafsput\_quad(0.03,0.02,7.5,0.45,0.25,N,0.5) | 0.3705 (0.000677) | 0.3749 (0.001961) | 0.3769 (0.004169) | 0.3806 (0.347508) | 0.3810 (2.223355) |
| Nearest Point (Running Time,s) fsg\_aafsput\_near(0.03,0.02,7.5,0.45,0.25,N,0.5) | 0.3634 (0.000912) | 0.4077 (0.001330) | 0.4428 (0.002863) | 0.6130 (0.286186) | 0.6368 (2.063820) |

Comment

The Forward Shooting grid method gives a much more accurate result as compared to the standard binomial tree. Further, although the computation time for 15 iterations and below is ionger as compared to the binomial tree, the computation time for 100 iterations and above is much shorter than that of the binomial tree.

Comparing the 3 different interpolation methods, the nearest point interpolation method gives the most inaccurate results. Between the Linear and Quadratic Interpolation, the Quadratic method seems to me more accurate, from the fact that the values are increasing across the different number of N’s , whereas there is a turn from N=100 to N=200 for the Linear interpolation. Of course, the quadratic interpolation running time suffers due to the increased complexity.