C++ Level 9 Group C

bondxue

December 2, 2017

Problem b. Testing for Batch 1 and 2:
• Testing for Batch 1:
$T = 0.25$, $K = 65$, $sig = 0.30$, $r = 0.08$, $S = 60$ (then $C = 2.13337$, $P = 5.84628$).
First, I fix $NSIM = 100000$ to see the accuracy of call price approximation C_{approx} with increasing NT. The simulation results are shown as follows.

Table 1: Approximation of Call price for Batch 1

NT	NSIM	C_{approx}	error
25	100000	2.09816	0.0352141
50	100000	2.11001	0.0233649
100	100000	2.13295	0.000422561
200	100000	2.16282	0.0294475
300	100000	2.16161	0.0282362
400	100000	2.15928	0.0259094

We could observe that it is not really the larger NT we choose, the more accurate approximate call price we will obtain. When NT increases up to a certain number (in this case $NT_{threshold} = 100$), C_{approx} approaches to the exact value C. When NT keeps increasing, the approximate result will get worse. The similar result will be obtained for put price approximation, although the threshold point is different in this case ($NT_{threshold} = 300$).

Table 2: Approximation of Put price for Batch 1

NT	NSIM	P_{approx}	error
25	100000	5.87348	0.0272024
50	100000	5.85897	0.0126931
100	100000	5.8726	0.0263177
200	100000	5.86236	0.0160752
300	100000	5.8416	0.000181169
400	100000	5.83399	0.0122882
500	100000	5.83729	0.00898622

Therefore, with a fixed $NSIM$, the increasing NT to a certain number will be expected a more accurate price approximation. Once NT keeps increasing over the certain point, the

approximate result will get worse. The underlying reason for this observation is that larger number of intervals means within one simulation, the result is more volatile than that with a smaller number. The stock price is assumed to follow a geometric brownian motion in our simulation between two mesh points, it has randomness in each interval. Thus with more intervals, the randomness can reasonably be expected to increase, i.e. final results will be more volatile. Thus, with each simulation's result more volatile, we need more simulations to expect results converge to an acceptable level.
Then I fix $NT = 100$ to the accuracy of call price approximation C_{approx} with increasing $NSIM$. The simulation results are shown as follows.

Table 3: Approximation of Call price for Batch 1

NT	NSIM	C_{approx}	error
100	100000	2.11001	0.0233649
100	500000	2.14306	0.00969012
100	800000	2.13572	0.00235277
100	1000000	2.13288	0.000487942
100	2000000	2.12921	0.00416193
100	3000000	2.13067	0.00269967

We could observe that with fixed NT, the increasing $NSIM$ will lead to a more accurate approximation of call price. The similar result is obtained for the put price approximation:

Table 4: Approximation of Put price for Batch 1

NT	NSIM	P_{approx}	error
100	100000	5.8726	0.0263177
100	500000	5.84404	0.00223901
100	800000	5.84778	0.00149959
100	1000000	5.85106	0.00477992
100	2000000	5.8512	0.0049238

Therefore, we can conclude that the larger $NSIM$, the more accurate the approximate price will be. This observation is guaranteed by the law of large numbers. As $NSIM$ approaches to infinity, the average (approximate price) will approach to the real value (actual price).
Therefore, in order to get enough accuracy to approximate option price ($error < 0.01$), I would set $NT = 100$ and $NSIM > 1000000$.

• Test for Batch 2:

$T = 1.0$, $K = 100$, $sig = 0.2$, $r = 0.0$, $S = 100$ (then $C = 7.96557$, $P = 7.96557$)
In this test, I increase NT with a decent speed and increase $NSIM$ with a dramatic speed for each simulation to see the accuracy of price change. The call price simulation results are shown as follows.

Table 5: Approximation of Call price for Batch 2

NT	NSIM	C_{approx}	error
25	100000	7.88769	0.0778828
50	500000	7.99036	0.0247896
75	1000000	7.96585	0.000278551
100	2000000	7.9577	0.00787339

Then the put price simulation results are shown as follows.

Table 6: Approximation of Put price for Batch 2

NT	NSIM	P_{approx}	error
25	100000	8.02621	0.060643
50	500000	7.96599	0.000418892
75	1000000	7.96744	0.0018714
100	2000000	7.97716	0.0115946

Therefore, in order to get enough accuracy to approximate option price ($error < 0.01$), I would set $NT = 75$ and $NSIM = 1000000$.

Problem c. Now, when we do the stress-testing for Batch 4 data.
$T = 30.0$, $K = 100.0$, $sig = 0.30$, $r = 0.08$, $S = 100.0$ (then $C = 92.17570$, $P = 1.24750$).

Table 7: Approximation of Call price for Batch 4

NT	NSIM	C_{approx}	error
50	100000	86.3363	5.83935
100	500000	90.2964	1.8793
500	2000000	92.0458	0.129899
500	3000000	91.9119	0.263751
500	5000000	91.8413	0.334375
500	6000000	91.8047	0.370984
1500	2000000	92.1071	0.0686346
1500	3000000	91.9407	0.235041

The general performance is really similar to what we observe in Problem (b). However, since for Batch 4, the maturity date T_4 is much larger than Batch 1 and Batch 2 options, in order to obtain higher accuracy of approximation of option prices, NT_4 should be set reasonably higher than NT_1 and NT_2. Since larger NT will cause more randomness of the stock price as explained in Problem (b), so we also need much larger $NSIM_4$ compared with $NSIM_1$ and $NSIM_2$ for Batch 1 and Batch 2 to see the convergence of the approximate option price. As we can see from Table 7, even I tried $NT = 1500$ and $NSIM = 3000000$, the approximate call price still have $error = 0.235041$. Best performance is for $NT = 1500$ and $NSIM = 2000000$ I obtain one decimal place accuracy. So I conclude that when $NT = 1500$, much larger $NSIM$ should be set ($NSIM \geq 10000000$) to obtain the two decimal places accuracy.

Table 8: Approximation of Put price for Batch 4

NT	NSIM	P_{approx}	error
500	500000	1.25896	0.0114639
500	1000000	1.25418	0.00667774
500	2000000	1.25306	0.00555888

However, for Put option, the approximation is much easier to obtain the two decimal places accuracy. We could set $NT = 500$ and $NSIM = 1000000$. Of course, the larger $NSIM$ we could set, the better approximation we could obtain. However, NT should be set in a reasonable range as we discussed before, the too large NT may cause the high variance of the simulation results.