E4732 CaseStudy 1 name: Yunfei Yan UNI: yy2516

(1) Description of coding environment.

I use Mobaxterm to code this case study 1 @clic-lab.cs.columbia.edu in linux environment, my compiler is git and my editor is vim.

I haven't used any external library to do the math calculation.

(2) My answers to the question.

Notice: all the results with bold and underline font are the ones same as BS formula.

FFT:

eta = 0.25	N = 2 ^ 8	N = 2 ^ 10	N = 2 ^ 12	N = 2 ^ 14
Alpha = -2	K = 900; European put option price = **0.044996**			
Alpha = -5	K = 1100; European put option price = **1.148194**			
Alpha = -10	K = 1300; European put option price = **9.588557**			
Alpha = -20	K = 1500; European put option price = **40.256689**			

FRFFT:

K = 1500	N = 2 ^ 7	N = 2 ^ 8	N = 2 ^ 9	N = 2 ^ 10
Alpha = -2	40.256137	**40.256689**	**40.256689**	**40.256689**
Alpha = -5	40.257111	**40.256689**	**40.256689**	**40.256689**
Alpha = -10	40.256274	**40.256689**	**40.256689**	**40.256689**
Alpha = -20	40.291469	**40.256689**	**40.256689**	**40.256689**

K = 1300	N = 2 ^ 7	N = 2 ^ 8	N = 2 ^ 9	N = 2 ^ 10
Alpha = -2	9.588917	**9.588557**	**9.588557**	**9.588557**
Alpha = -5	9.588330	**9.588557**	**9.588557**	**9.588557**
Alpha = -10	9.588567	**9.588557**	**9.588557**	**9.588557**
Alpha = -20	9.587092	**9.588557**	**9.588557**	**9.588557**

K = 1100	N = 2 ^ 7	N = 2 ^ 8	N = 2 ^ 9	N = 2 ^ 10
Alpha = -2	1.147974,	**1.148194**	**1.148194**	**1.148194**
Alpha = -5	1.148286	**1.148194**	**1.148194**	**1.148194**
Alpha = -10	1.148196	**1.148194**	**1.148194**	**1.148194**
Alpha = -20	1.148242	**1.148194**	**1.148194**	**1.148194**

K = 900	N = 2 ^ 7	N = 2 ^ 8	N = 2 ^ 9	N = 2 ^ 10
Alpha = -2	0.045145	**0.044996**	**0.044996**	**0.044996**
Alpha = -5	0.044967	**0.044996**	**0.044996**	**0.044996**
Alpha = -10	0.044998	**0.044996**	**0.044996**	**0.044996**
Alpha = -20	0.044994	**0.044996**	**0.044996**	**0.044996**

COS:

N = 2 ^ 9	[a, b] = [-2, 2]	[a, b] = [-5, 5]	[a, b] = [-10, 10]	[a, b] = [-20, 20]
K = 900	**0.044996**	**0.044996**	**0.044996**	**0.044996**
K = 1100	**1.148194**	**1.148194**	**1.148194**	**1.148194**
K = 1300	**9.588557**	**9.588557**	**9.588557**	**9.588557**
K = 1500	**40.256689**	**40.256689**	**40.256689**	**40.256689**

N = 2 ^ 8	*[a, b] = [-2, 2]*	*[a, b] = [-5, 5]*	*[a, b] = [-10, 10]*	*[a, b] = [-20, 20]*
K = 900	***0.044996***	***0.044996***	***0.044996***	*0.045021*
K = 1100	***1.148194***	***1.148194***	***1.148194***	*1.148103*
K = 1300	***9.588557***	***9.588557***	***9.588557***	*9.588671*
K = 1500	***40.256689***	***40.256689***	***40.256689***	*40.256553*

N = 2 ^ 7	*[a, b] = [-2, 2]*	*[a, b] = [-5, 5]*	*[a, b] = [-10, 10]*	*[a, b] = [-20, 20]*
K = 900	***0.044996***	***0.044996***	*0.045026*	*0.224234*
K = 1100	***1.148194***	***1.148194***	*1.148095*	*1.318946*
K = 1300	***9.588557***	***9.588557***	*9.588680*	*8.995516*
K = 1500	***40.256689***	***40.256689***	*40.256542*	*40.215988*

N = 2 ^ 6	*[a, b] = [-2, 2]*	*[a, b] = [-5, 5]*	*[a, b] = [-10, 10]*	*[a, b] = [-20, 20]*
K = 900	***0.044996***	*0.045038*	*0.246895*	*-4.998904*
K = 1100	***1.148194***	*1.148078*	*1.315780*	*-8.704494*
K = 1300	***9.588557***	*9.588698*	*8.961801*	*7.075620*
K = 1500	***40.256689***	*40.256519*	*40.218453*	*51.965224*

*According to the formula: c1 = (r – q) T; c2 = sigma ^ 2 * T; c4 = 0; L = 10, then we have the best choice for [a, b] is [-8.9 * 0.001 – 2.12, -8.9 * 0.001 + 2.12] approximately equal to [-2, 2].*

(3) Compare and conclusion.

a. Theoretically, the speed about the 3 methods should be ranked like this: COS >FRFFT>FFT. Due to the machine I used is the one in Columbia Clic-lab, which is very powerful, I didn't feel the big speed difference between these methods.

b. COS estimation: [a, b], N
FFT estimation: alpha, eta, N
FRFFT estimation: alpha, eta, N, lambda

c. They can only be used in the path-independent vanilla options, and when the option is highly out of the money, they are useless.

d. Both FFT and FRFFT could give us N option prices at N different strikes. For example:

fft K
0.000050, 653.861889
0.000502, 721.311383
0.004166, 795.718668
0.028460, 877.801479
0.160517, 968.351588
0.750762, 1068.242445
2.928012, 1178.437601
9.588557, 1300.000000
26.598111, 1434.102237
63.186041, 1582.037867
130.285553, 1745.233881
236.923841, 1925.264472
386.921106, 2123.866220
578.497978, 2342.954845
806.965025, 2584.643681

The pity is that not all the price can be used (eg: deep out of the money). The COS method give us only one price each time.

e. These three methods are model free, all they need is a simple characteristic function. Also, the COS separate the model from its payoff, which make it more useful.

f. According to the results, the FFT method is not sensitive to the alpha and N, which means it is comparative stable.

g. The FRFFT are a little sensitive to the alpha and N, it seems that no matter what alpha we use, N = 2 ^ 7 is always not a good choice for us. I think we should use N >= 2 ^ 8 in practical to price the option.

h. The COS is very sensitive to the coefficient we selected. We can see that the interval and N are somewhat complementary. With the smaller N, the interval should be narrower to fit for the N's decrease.

(4) Description of the logic present in the written source code

I am a new coder and have no experience before. So I used the simplest method to calculate the result in the main function step by step. There is no other function except main in all my 3 codes. Every time I change the coefficients manually to get the result.

I download my code from Mobaxterm directly with .cc type. You can open it using notepad (I have already tried, and it works.) If you cannot open it, please contact me. Thank you!