FU FOUNDATION SCHOOL OF ENGINEERING AND APPLIED SCIENCE
DEPARTMENT OF ELECTRICAL ENGINEERING
Master of Science in Electrical Engineering

Homework 1

Computational Methods in Finance

IEOR 4732

Author:

Alban Dietrich (ad4017)

19 February 2023

Contents

1 Case Study 1 **1**

1.1 Fast Fourier transform (FFT) 1

1.2 Fractional fast Fourier transform (FrFFT) 2

1.3 Fourier-cosine (COS) method 3

1.4 Conclusion 4

2 Appendix **4**

1 Case Study 1

In this problem we will study the Black-Scholes framework. We consider the following parameters:

- Spot price $S_0 = 1900$
- Maturity $T = 0.25$
- Volatility $\sigma = 0.36$
- Risk-free interest rate $r = 2.00\%$
- Continuous dividend rate $q = 1.87\%$
- Strike range $K = [2000, 2100, 2200]$

The goal is to price European call options via the following transform techniques:

- Fast Fourier transform (FFT): $\eta = \Delta\nu = 0.25$, $\alpha = [0.4, 1.0, 1.4, 3.0]$, $N = 2^n$ for $n = [9, 11, 13, 15]$ and $\beta = \ln K - \frac{\lambda N}{2}$.
- Fractional fast Fourier transform (FrFFT): $\eta = \Delta\nu = 0.25$, $\alpha = [0.4, 1.0, 1.4, 3.0]$, $N = 2^n$ for $n = [6, 7, 8, 9]$, $\lambda = \Delta k = 0.1$ and $\beta = \ln K - \frac{\lambda N}{2}$.
- Fourier-cosine (COS) method, for the following [a, b] intervals: [-1, 1], [-4, 4], [-8, 8] and [-12, 12].

First, by using the Black-Scholes formula, we calculate the option premiums to have an idea later if our estimations are good or not (see code in appendix). The results obtained are the following:

- For $K = 2000$: 95.2467
- For $K = 2100$: 64.8346
- For $K = 2200$: 42.9472

1.1 Fast Fourier transform (FFT)

A code for the FTT is provided at the end of this report. The code is based on the one provided during the course and lecture 1.

Here below are the results for the different input parameters.

<u>K = 2000</u>

N \ α	0.4	1.0	1.4	3.0
2^9	95.3281	95.2467	95.2467	95.2467
2^{11}	95.3281	95.2467	95.2467	95.2467
2^{13}	95.3281	95.2467	95.2467	95.2467
2^{15}	95.3281	95.2467	95.2467	95.2467

K = 2100

N \ α	0.4	1.0	1.4	3.0
2^9	64.9160	64.8346	64.8346	64.8346
2^{11}	64.9160	64.8346	64.8346	64.8346
2^{13}	64.9160	64.8346	64.8346	64.8346
2^{15}	64.9160	64.8346	64.8346	64.8346

K = 2200

N \ α	0.4	1.0	1.4	3.0
2^9	43.0286	42.9472	42.9472	42.9472
2^{11}	43.0286	42.9472	42.9472	42.9472
2^{13}	43.0286	42.9472	42.9472	42.9472
2^{15}	43.0286	42.9472	42.9472	42.9472

We can notice that for all the K values, the results are pretty stable (for $1.0 \leq \alpha \leq 3.0$, the outputs are the same). The prices of the call options are thus (as seen at the beginning of the report): 95.2467 for K = 2000, 64.8346 for K = 2100 and 42.9472 for K = 2200.

Concerning the time execution, it oscillates between roughly 0.002 and 0.1 seconds (when n varies between 9 and 15). For K = 2000 and $n = 9$, it takes approximately 0.002 seconds.

1.2 Fractional fast Fourier transform (FrFFT)

Now let us consider the Fractional fast Fourier transform method. The code was also based on lecture 1 and the provided codes.

Here below are the results obtained for this method:

K = 2000

N \ α	0.4	1.0	1.4	3.0
2^6	120.7730	49.1717	47.6224	47.1893
2^7	53.3165	47.6260	47.6225	47.6199
2^8	315.1885	249.7166	215.2645	127.0006
2^9	429.4228	265.5165	197.5578	81.2383

K = 2100

N \ α	0.4	1.0	1.4	3.0
2^6	105.7646	34.2007	32.6630	32.2015
2^7	38.1122	32.4225	32.4192	32.4159
2^8	274.6038	215.3367	184.1533	104.2635
2^9	400.7706	242.6354	177.0704	64.8482

K = 2200

N \ α	0.4	1.0	1.4	3.0
2^6	94.8983	23.3896	21.8780	21.4615
2^7	7.1678	21.4791	21.4765	21.4746
2^8	238.6922	185.5293	157.5617	85.9106
2^9	376.3807	224.0165	160.8454	52.7204

In contrary with the previous case (FFT), we do not have stable results. We can see fluctuations and it is quite difficult to guess the pricing of the call options.

These outputs can be due to bad input parameters. A possible solution would be to find other parameters which allow to obtain good results.

Concerning the time execution, it oscillates between roughly 0.0004 and 0.002 seconds (when n varies between 6 and 9). For K = 2000 and $n = 9$, it takes approximately 0.0025 seconds. FFT and FrFFT have comparable execution times.

1.3 Fourier-cosine (COS) method

Finally, let us study the Fourier-cosine method. The code provided at the end of this document is based on lecture 2 and the following article "A Novel Pricing Method for European Options Based on Fourier-Cosine Series Expansions"[1].

Here below are the results obtained:

K = 2000

N \ [a, b]	[-1, 1]	[-4, 4]	[-8, 8]	[-12, 12]
2^6	95.2467	95.2475	94.9038	-22626.0999
2^7	95.2467	95.2467	95.2699	310.1706
2^8	95.2467	95.2467	95.2467	95.2469
2^9	95.2467	95.2467	95.2467	95.2467

K = 2100

N \ [a, b]	[-1, 1]	[-4, 4]	[-8, 8]	[-12, 12]
2^6	64.8346	64.8358	224.1476	13516.6644
2^7	64.8346	64.8346	64.8646	761.9153
2^8	64.8346	64.8346	64.8346	64.8346
2^9	64.8346	64.8346	64.8346	64.8346

[1] Source: `https://mpra.ub.uni-muenchen.de/8914/`.

K = 2200

N \ [a, b]	[-1, 1]	[-4, 4]	[-8, 8]	[-12, 12]
2^6	42.9472	42.9472	312.9719	49139.3478
2^7	42.9472	42.9472	42.9452	817.2623
2^8	42.9472	42.9472	42.9472	42.9469
2^9	42.9472	42.9472	42.9472	42.9472

These results are stable for small values of a, b and big values of N, and correspond to the ones found with FFT which confirms our pricing expectations for the call options.

When we take big value of |a| and |b| (e.g. a = -12 and b = 12), we notice with the results that this method has some sensitivity regarding the a and b parameters (see table above). Indeed for big values (see example of code in appendix) of these parameters (in absolute value), we can get very big results which do not make sense. The payoff of a call option grows exponentially with the logarithm of the stock price[1] and introduces significant cancellation error for large values of b - a. Therefore we should be careful when choosing those parameters and limit the value of b - a.

Concerning the time execution, it oscillates around 0.8 seconds. This is probably due to the use of the integrate function of the Scipy library which is time consuming. An idea to speed up this algorithm would be to use directly the integration results provided in the article mentioned before[1] to avoid to do the integration.

This has be done in the code in the appendix and we can notice that it reduces the execution time drastically. Indeed it takes on average around 0.003 seconds to calculate the results. For K = 2000 and $n = 9$, it takes approximately 0.004 seconds.

1.4 Conclusion

In the end, we can conclude that the estimated prices for the call options are as follows: 95.2467 for K = 2000, 64.8346 for K = 2100 and 42.9472 for K = 2200.

The best methods are the FFT and Fourier-cosine methods (but we have to be careful to choose small enough values for a and b, and a large enough value for N to avoid bad results). The results of the FrFTT method are unstable and do not predict a correct price. A possible solution to correct this would be to find better input parameters.

Finally, concerning the execution time, for the code used in this report (see appendix), the FFT code is faster than the Fourier-cosine method. Therefore, in this situation, the FFT is the best method to use.

2 Appendix

On the next page is the code used for this case study.

case-study-1

February 19, 2023

Alban Dietrich, UNI: ad4017. 19 February 2023

1 Case Study 1

Import libraries.

[1]:
```
import warnings
warnings.filterwarnings("ignore")

import numpy as np
import matplotlib.pyplot as plt
import time
import scipy
import scipy.integrate as integrate
from scipy.stats import norm
import math
```

We define the needed parameters.

[2]:
```
# Fixed Parameters
S0 = 1900
K_list = [2000, 2100, 2200]
r = 0.02
q = 0.0187
sig = 0.36
T = 0.25
```

[3]:
```
params = []
params.append(sig)
```

Let us first calculate the option premiums to have an idea later if our estimations are good or no. Here we use the Black-Scholes formula.

[4]:
```
# Black-Merton-Scholes calculator
def BS_d1(S, K, r, q, sigma, tau):
    ''' Computes d1 for the Black-Merton-Scholes formula '''
    d1 = 1.0*(np.log(1.0 * S/K) + (r - q + sigma**2/2) * tau) / (sigma * np.
 ↪sqrt(tau))
```

```
    return d1

def BS_d2(S, K, r, q, sigma, tau):
    ''' Computes d2 for the Black-Merton-Scholes formula '''
    d2 = 1.0*(np.log(1.0 * S/K) + (r - q - sigma**2/2) * tau) / (sigma * np.
 ↪sqrt(tau))
    return d2

def BS_price(type_option, S, K, r, q, sigma, T, t=0):
    ''' Computes the Black-Merton-Scholes price for a 'call' or 'put' option '''
    tau = T - t
    d1 = BS_d1(S, K, r, q, sigma, tau)
    d2 = BS_d2(S, K, r, q, sigma, tau)
    if type_option == 'call':
        price = S * np.exp(-q * tau) * norm.cdf(d1) - K * np.exp(-r * tau) *␣
 ↪norm.cdf(d2)
    elif type_option == 'put':
        price = K * np.exp(-r * tau) * norm.cdf(-d2) - S * np.exp(-q * tau) *␣
 ↪norm.cdf(-d1)
    return price
```

[5]:
```
print("Option Premiums")
print("For K = 2000: ", BS_price("call", S0, 2000, r, q, sig, T))
print("For K = 2100: ", BS_price("call", S0, 2100, r, q, sig, T))
print("For K = 2200: ", BS_price("call", S0, 2200, r, q, sig, T))
```

```
Option Premiums
For K = 2000:  95.2466924265716
For K = 2100:  64.83462030513067
For K = 2200:  42.94717532152765
```

Now let us define the characteristic function of the model. Here we consider the Black-Scholes model.

[6]:
```
# Model under consideration
model = 'BS'
```

[7]:
```
def generic_CF(u, params, S0, r, q, T, model):

    if(model=='BS'):
        sig = params[0]

        phi = np.exp(1j*(np.log(S0)+(r-q-sig**2/2)*T)*u-1/2*sig**2*u**2*T)

    else:
        # Other models can be added later, for the moment just the 'BS' model␣
 ↪is considered
        print("No model called ",model)
```

```
    return phi
```

1.1 Fast Fourier Transform (FFT)

This code is based on the one provided during the course.

We define the FTT parameters.

[9]:
```
# Parameters for FFT

n_FFT_list = [9,11,13,15]

# Step-size
eta = 0.25

# Damping factor
alpha_list = [0.4,1.0,1.4,3.0]
```

We define the FTT function.

[10]:
```
def genericFFT(params, S0, K, r, q, T, alpha, eta, n, model, lda, beta):

    N = 2**n

    # forming vector x and strikes km for m=1,...,N
    km = np.zeros((N))
    xX = np.zeros((N))

    # discount factor
    df = math.exp(-r*T)

    nuJ = np.arange(N)*eta
    psi_nuJ = generic_CF(nuJ-(alpha+1)*1j, params, S0, r, q, T, model)/((alpha␣
↪+ 1j*nuJ)*(alpha+1+1j*nuJ))

    for j in range(N):
        km[j] = beta+j*lda
        if j == 0:
            wJ = (eta/2)
        else:
            wJ = eta

        xX[j] = np.exp(-1j*beta*nuJ[j])*df*psi_nuJ[j]*wJ

    yY = np.fft.fft(xX)
    cT_km = np.zeros((N))
    for i in range(N):
```

```
        multiplier = np.exp(-alpha*km[i])/math.pi
        cT_km[i] = multiplier*np.real(yY[i])

    return km, cT_km
```

Print the results.

[11]:
```
print(' ')
print('====================')
print('Model is %s' % model)
print('--------------------')

# FFT

print('||| FTT |||')

for K in K_list:

    k = math.log(K)

    for n in n_FFT_list:

        N_FFT = 2**n

        # Step-size in log strike space
        lda_FFT = (2*math.pi/N_FFT)/eta


        # Choice of beta
        beta_FTT = np.log(K)-N_FFT*lda_FFT/2

        for alpha in alpha_list:

            print("k = ", K, " AND n = ",n, " AND alpha = ",alpha)
            start_time = time.time()
            km, cT_km = genericFFT(params, S0, K, r, q, T, alpha, eta, n,␣
↪model, lda_FFT, beta_FTT)

            # Interpolation
            cT_k = np.interp(k, km, cT_km)

            elapsed_time = time.time() - start_time

            print("Option via FFT: for strike %s the option premium is %6.4f" %␣
↪(K, cT_k))

            print('FFT execution time was %0.7f' % elapsed_time)
```

```
==================
Model is BS
------------------
||| FTT |||
k =  2000  AND n =  9  AND alpha =  0.4
Option via FFT: for strike 2000 the option premium is 95.3281
FFT execution time was 0.0022249
k =  2000  AND n =  9  AND alpha =  1.0
Option via FFT: for strike 2000 the option premium is 95.2467
FFT execution time was 0.0017390
k =  2000  AND n =  9  AND alpha =  1.4
Option via FFT: for strike 2000 the option premium is 95.2467
FFT execution time was 0.0019188
k =  2000  AND n =  9  AND alpha =  3.0
Option via FFT: for strike 2000 the option premium is 95.2467
FFT execution time was 0.0019000
k =  2000  AND n =  11  AND alpha =  0.4
Option via FFT: for strike 2000 the option premium is 95.3281
FFT execution time was 0.0066299
k =  2000  AND n =  11  AND alpha =  1.0
Option via FFT: for strike 2000 the option premium is 95.2467
FFT execution time was 0.0066750
k =  2000  AND n =  11  AND alpha =  1.4
Option via FFT: for strike 2000 the option premium is 95.2467
FFT execution time was 0.0065658
k =  2000  AND n =  11  AND alpha =  3.0
Option via FFT: for strike 2000 the option premium is 95.2467
FFT execution time was 0.0063610
k =  2000  AND n =  13  AND alpha =  0.4
Option via FFT: for strike 2000 the option premium is 95.3281
FFT execution time was 0.0246201
k =  2000  AND n =  13  AND alpha =  1.0
Option via FFT: for strike 2000 the option premium is 95.2467
FFT execution time was 0.0247972
k =  2000  AND n =  13  AND alpha =  1.4
Option via FFT: for strike 2000 the option premium is 95.2467
FFT execution time was 0.0251968
k =  2000  AND n =  13  AND alpha =  3.0
Option via FFT: for strike 2000 the option premium is 95.2467
FFT execution time was 0.0259080
k =  2000  AND n =  15  AND alpha =  0.4
Option via FFT: for strike 2000 the option premium is 95.3281
FFT execution time was 0.1015353
k =  2000  AND n =  15  AND alpha =  1.0
Option via FFT: for strike 2000 the option premium is 95.2467
FFT execution time was 0.1005521
k =  2000  AND n =  15  AND alpha =  1.4
```

```
Option via FFT: for strike 2000 the option premium is 95.2467
FFT execution time was 0.0971539
k =  2000  AND n =  15  AND alpha =  3.0
Option via FFT: for strike 2000 the option premium is 95.2467
FFT execution time was 0.0979509
k =  2100  AND n =  9  AND alpha =  0.4
Option via FFT: for strike 2100 the option premium is 64.9160
FFT execution time was 0.0018110
k =  2100  AND n =  9  AND alpha =  1.0
Option via FFT: for strike 2100 the option premium is 64.8346
FFT execution time was 0.0015850
k =  2100  AND n =  9  AND alpha =  1.4
Option via FFT: for strike 2100 the option premium is 64.8346
FFT execution time was 0.0015130
k =  2100  AND n =  9  AND alpha =  3.0
Option via FFT: for strike 2100 the option premium is 64.8346
FFT execution time was 0.0015030
k =  2100  AND n =  11  AND alpha =  0.4
Option via FFT: for strike 2100 the option premium is 64.9160
FFT execution time was 0.0061779
k =  2100  AND n =  11  AND alpha =  1.0
Option via FFT: for strike 2100 the option premium is 64.8346
FFT execution time was 0.0064709
k =  2100  AND n =  11  AND alpha =  1.4
Option via FFT: for strike 2100 the option premium is 64.8346
FFT execution time was 0.0068221
k =  2100  AND n =  11  AND alpha =  3.0
Option via FFT: for strike 2100 the option premium is 64.8346
FFT execution time was 0.0072007
k =  2100  AND n =  13  AND alpha =  0.4
Option via FFT: for strike 2100 the option premium is 64.9160
FFT execution time was 0.0257320
k =  2100  AND n =  13  AND alpha =  1.0
Option via FFT: for strike 2100 the option premium is 64.8346
FFT execution time was 0.0248330
k =  2100  AND n =  13  AND alpha =  1.4
Option via FFT: for strike 2100 the option premium is 64.8346
FFT execution time was 0.0249550
k =  2100  AND n =  13  AND alpha =  3.0
Option via FFT: for strike 2100 the option premium is 64.8346
FFT execution time was 0.0246341
k =  2100  AND n =  15  AND alpha =  0.4
Option via FFT: for strike 2100 the option premium is 64.9160
FFT execution time was 0.0991189
k =  2100  AND n =  15  AND alpha =  1.0
Option via FFT: for strike 2100 the option premium is 64.8346
FFT execution time was 0.0995119
k =  2100  AND n =  15  AND alpha =  1.4
```

```
Option via FFT: for strike 2100 the option premium is 64.8346
FFT execution time was 0.0970678
k =  2100  AND n =  15  AND alpha =  3.0
Option via FFT: for strike 2100 the option premium is 64.8346
FFT execution time was 0.0990419
k =  2200  AND n =  9  AND alpha =  0.4
Option via FFT: for strike 2200 the option premium is 43.0286
FFT execution time was 0.0015738
k =  2200  AND n =  9  AND alpha =  1.0
Option via FFT: for strike 2200 the option premium is 42.9472
FFT execution time was 0.0015092
k =  2200  AND n =  9  AND alpha =  1.4
Option via FFT: for strike 2200 the option premium is 42.9472
FFT execution time was 0.0015171
k =  2200  AND n =  9  AND alpha =  3.0
Option via FFT: for strike 2200 the option premium is 42.9472
FFT execution time was 0.0017149
k =  2200  AND n =  11  AND alpha =  0.4
Option via FFT: for strike 2200 the option premium is 43.0286
FFT execution time was 0.0066102
k =  2200  AND n =  11  AND alpha =  1.0
Option via FFT: for strike 2200 the option premium is 42.9472
FFT execution time was 0.0064390
k =  2200  AND n =  11  AND alpha =  1.4
Option via FFT: for strike 2200 the option premium is 42.9472
FFT execution time was 0.0064130
k =  2200  AND n =  11  AND alpha =  3.0
Option via FFT: for strike 2200 the option premium is 42.9472
FFT execution time was 0.0067010
k =  2200  AND n =  13  AND alpha =  0.4
Option via FFT: for strike 2200 the option premium is 43.0286
FFT execution time was 0.0247490
k =  2200  AND n =  13  AND alpha =  1.0
Option via FFT: for strike 2200 the option premium is 42.9472
FFT execution time was 0.0245178
k =  2200  AND n =  13  AND alpha =  1.4
Option via FFT: for strike 2200 the option premium is 42.9472
FFT execution time was 0.0245588
k =  2200  AND n =  13  AND alpha =  3.0
Option via FFT: for strike 2200 the option premium is 42.9472
FFT execution time was 0.0245039
k =  2200  AND n =  15  AND alpha =  0.4
Option via FFT: for strike 2200 the option premium is 43.0286
FFT execution time was 0.0974362
k =  2200  AND n =  15  AND alpha =  1.0
Option via FFT: for strike 2200 the option premium is 42.9472
FFT execution time was 0.0982070
k =  2200  AND n =  15  AND alpha =  1.4
```

```
Option via FFT: for strike 2200 the option premium is 42.9472
FFT execution time was 0.0975811
k =  2200  AND n =  15  AND alpha =  3.0
Option via FFT: for strike 2200 the option premium is 42.9472
FFT execution time was 0.0978580
```

1.2 Fractional fast Fourier Transform (FrFFT)

This code is based on the one provided during the course.

We define the FrFTT parameters.

[12]:
```
# Parameters for FrFTT

n_FrFFT_list = [6,7,8,9]

# Step-size
eta = 0.25

# Damping factor
alpha_list = [0.4,1.0,1.4,3.0]

# Lda
lda_FrFTT = 0.1
```

We define the FrFTT function.

[13]:
```
def genericFrFFT(params, S0, K, r, q, T, alpha, eta, n, model, lda, beta):

    N = 2**n
    gamma = eta*lda/(2*math.pi)

    # initialize x, y, z, and cT_km
    km = np.zeros((N))
    x = np.zeros((N))
    y = np.zeros((2*N), dtype=np.complex)
    z = np.zeros((2*N), dtype=np.complex)
    cT_km = np.zeros((N))

    # discount factor
    df = math.exp(-r*T)

    # compute x
    nuJ = np.arange(N)*eta
    psi_nuJ = generic_CF(nuJ-(alpha+1)*1j, params, S0, r, q, T, model)/((alpha
 ↪+ 1j*nuJ)*(alpha+1+1j*nuJ))

    for j in range(N):
        km[j] = beta+j*lda
```

```
        if j == 0:
            wJ = (eta/2)
        else:
            wJ = eta
        x[j] = np.exp(-1j*beta*nuJ[j])*df*psi_nuJ[j]*wJ

    # set up y
    for i in range(N):
        y[i] = np.exp(-1j*math.pi*gamma*i**2)*x[i]
    y[N:] = 0

    # set up z
    for i in range(N):
        z[i] = np.exp(1j*math.pi*gamma*i**2)
    z[N:] = z[:N][::-1]

    # compute xi_hat
    xi_hat = np.fft.ifft(np.fft.fft(y) * np.fft.fft(z))

    # compute call prices
    for i in range(N):
        cT_km[i] = np.exp(-alpha*(beta + i*lda))/math.pi * (np.exp(-1j*math.
↪pi*gamma*i**2)*xi_hat[i]).real

    return km, cT_km
```

We print the results.

[14]:
```
print(' ')
print('====================')
print('Model is %s' % model)
print('--------------------')

# FrFFT

print('||| FrFTT |||')

for K in K_list:

    k = math.log(K)

    for n in n_FrFFT_list:

        N_FrFFT = 2**n


        # Choice of beta
```

```
        beta_FrFTT = np.log(K)-N_FrFFT*lda_FrFTT/2

        for alpha in alpha_list:

            print("k = ", K, " AND n = ",n, " AND alpha = ",alpha)
            start_time = time.time()
            km, cT_km = genericFrFFT(params, S0, K, r, q, T, alpha, eta, n,␣
↪model, lda_FrFTT, beta_FrFTT)

            # Interpolation
            cT_k = np.interp(k, km, cT_km)

            elapsed_time = time.time() - start_time

            print("Option via FrFFT: for strike %s the option premium is %6.4f"␣
↪% (K, cT_k))

            print('FrFFT execution time was %0.7f' % elapsed_time)
```

```
====================
Model is BS
--------------------
||| FrFTT |||
k =  2000  AND n =  6  AND alpha =  0.4
Option via FrFFT: for strike 2000 the option premium is 120.7730
FrFFT execution time was 0.0007439
k =  2000  AND n =  6  AND alpha =  1.0
Option via FrFFT: for strike 2000 the option premium is 49.1717
FrFFT execution time was 0.0004468
k =  2000  AND n =  6  AND alpha =  1.4
Option via FrFFT: for strike 2000 the option premium is 47.6224
FrFFT execution time was 0.0004201
k =  2000  AND n =  6  AND alpha =  3.0
Option via FrFFT: for strike 2000 the option premium is 47.1893
FrFFT execution time was 0.0005460
k =  2000  AND n =  7  AND alpha =  0.4
Option via FrFFT: for strike 2000 the option premium is 53.3165
FrFFT execution time was 0.0007198
k =  2000  AND n =  7  AND alpha =  1.0
Option via FrFFT: for strike 2000 the option premium is 47.6260
FrFFT execution time was 0.0007071
k =  2000  AND n =  7  AND alpha =  1.4
Option via FrFFT: for strike 2000 the option premium is 47.6225
FrFFT execution time was 0.0008759
k =  2000  AND n =  7  AND alpha =  3.0
Option via FrFFT: for strike 2000 the option premium is 47.6199
```

```
FrFFT execution time was 0.0008838
k =  2000  AND n =  8  AND alpha =  0.4
Option via FrFFT: for strike 2000 the option premium is 315.1885
FrFFT execution time was 0.0013659
k =  2000  AND n =  8  AND alpha =  1.0
Option via FrFFT: for strike 2000 the option premium is 249.7166
FrFFT execution time was 0.0012867
k =  2000  AND n =  8  AND alpha =  1.4
Option via FrFFT: for strike 2000 the option premium is 215.2645
FrFFT execution time was 0.0012820
k =  2000  AND n =  8  AND alpha =  3.0
Option via FrFFT: for strike 2000 the option premium is 127.0006
FrFFT execution time was 0.0014031
k =  2000  AND n =  9  AND alpha =  0.4
Option via FrFFT: for strike 2000 the option premium is 429.4228
FrFFT execution time was 0.0028720
k =  2000  AND n =  9  AND alpha =  1.0
Option via FrFFT: for strike 2000 the option premium is 265.5165
FrFFT execution time was 0.0026932
k =  2000  AND n =  9  AND alpha =  1.4
Option via FrFFT: for strike 2000 the option premium is 197.5578
FrFFT execution time was 0.0025630
k =  2000  AND n =  9  AND alpha =  3.0
Option via FrFFT: for strike 2000 the option premium is 81.2383
FrFFT execution time was 0.0026109
k =  2100  AND n =  6  AND alpha =  0.4
Option via FrFFT: for strike 2100 the option premium is 105.7646
FrFFT execution time was 0.0003510
k =  2100  AND n =  6  AND alpha =  1.0
Option via FrFFT: for strike 2100 the option premium is 34.2007
FrFFT execution time was 0.0003459
k =  2100  AND n =  6  AND alpha =  1.4
Option via FrFFT: for strike 2100 the option premium is 32.6630
FrFFT execution time was 0.0003331
k =  2100  AND n =  6  AND alpha =  3.0
Option via FrFFT: for strike 2100 the option premium is 32.2015
FrFFT execution time was 0.0003340
k =  2100  AND n =  7  AND alpha =  0.4
Option via FrFFT: for strike 2100 the option premium is 38.1122
FrFFT execution time was 0.0006452
k =  2100  AND n =  7  AND alpha =  1.0
Option via FrFFT: for strike 2100 the option premium is 32.4225
FrFFT execution time was 0.0006511
k =  2100  AND n =  7  AND alpha =  1.4
Option via FrFFT: for strike 2100 the option premium is 32.4192
FrFFT execution time was 0.0006361
k =  2100  AND n =  7  AND alpha =  3.0
Option via FrFFT: for strike 2100 the option premium is 32.4159
```

```
FrFFT execution time was 0.0006621
k =  2100  AND n =  8  AND alpha =  0.4
Option via FrFFT: for strike 2100 the option premium is 274.6038
FrFFT execution time was 0.0012689
k =  2100  AND n =  8  AND alpha =  1.0
Option via FrFFT: for strike 2100 the option premium is 215.3367
FrFFT execution time was 0.0012939
k =  2100  AND n =  8  AND alpha =  1.4
Option via FrFFT: for strike 2100 the option premium is 184.1533
FrFFT execution time was 0.0012562
k =  2100  AND n =  8  AND alpha =  3.0
Option via FrFFT: for strike 2100 the option premium is 104.2635
FrFFT execution time was 0.0012569
k =  2100  AND n =  9  AND alpha =  0.4
Option via FrFFT: for strike 2100 the option premium is 400.7706
FrFFT execution time was 0.0025127
k =  2100  AND n =  9  AND alpha =  1.0
Option via FrFFT: for strike 2100 the option premium is 242.6354
FrFFT execution time was 0.0025351
k =  2100  AND n =  9  AND alpha =  1.4
Option via FrFFT: for strike 2100 the option premium is 177.0704
FrFFT execution time was 0.0025349
k =  2100  AND n =  9  AND alpha =  3.0
Option via FrFFT: for strike 2100 the option premium is 64.8482
FrFFT execution time was 0.0025661
k =  2200  AND n =  6  AND alpha =  0.4
Option via FrFFT: for strike 2200 the option premium is 94.8983
FrFFT execution time was 0.0003417
k =  2200  AND n =  6  AND alpha =  1.0
Option via FrFFT: for strike 2200 the option premium is 23.3896
FrFFT execution time was 0.0003412
k =  2200  AND n =  6  AND alpha =  1.4
Option via FrFFT: for strike 2200 the option premium is 21.8780
FrFFT execution time was 0.0003371
k =  2200  AND n =  6  AND alpha =  3.0
Option via FrFFT: for strike 2200 the option premium is 21.4615
FrFFT execution time was 0.0003400
k =  2200  AND n =  7  AND alpha =  0.4
Option via FrFFT: for strike 2200 the option premium is 27.1678
FrFFT execution time was 0.0006468
k =  2200  AND n =  7  AND alpha =  1.0
Option via FrFFT: for strike 2200 the option premium is 21.4791
FrFFT execution time was 0.0006571
k =  2200  AND n =  7  AND alpha =  1.4
Option via FrFFT: for strike 2200 the option premium is 21.4765
FrFFT execution time was 0.0006528
k =  2200  AND n =  7  AND alpha =  3.0
Option via FrFFT: for strike 2200 the option premium is 21.4746
```

```
FrFFT execution time was 0.0006542
k =  2200  AND n =  8  AND alpha =  0.4
Option via FrFFT: for strike 2200 the option premium is 238.6922
FrFFT execution time was 0.0012562
k =  2200  AND n =  8  AND alpha =  1.0
Option via FrFFT: for strike 2200 the option premium is 185.5293
FrFFT execution time was 0.0012741
k =  2200  AND n =  8  AND alpha =  1.4
Option via FrFFT: for strike 2200 the option premium is 157.5617
FrFFT execution time was 0.0012660
k =  2200  AND n =  8  AND alpha =  3.0
Option via FrFFT: for strike 2200 the option premium is 85.9106
FrFFT execution time was 0.0012629
k =  2200  AND n =  9  AND alpha =  0.4
Option via FrFFT: for strike 2200 the option premium is 376.3807
FrFFT execution time was 0.0025561
k =  2200  AND n =  9  AND alpha =  1.0
Option via FrFFT: for strike 2200 the option premium is 224.0165
FrFFT execution time was 0.0025053
k =  2200  AND n =  9  AND alpha =  1.4
Option via FrFFT: for strike 2200 the option premium is 160.8454
FrFFT execution time was 0.0024810
k =  2200  AND n =  9  AND alpha =  3.0
Option via FrFFT: for strike 2200 the option premium is 52.7204
FrFFT execution time was 0.0026062
```

1.3 Fourier-cosine (COS) method

This code is based on the lecture 2 and the following article: https://mpra.ub.uni-muenchen.de/8914/.

We define the COS parameters.

[15]:
```
ab_list = [(-1, 1), (-4, 4), (-8, 8), (-12, 12)]

# As nothing has been asked in the guidelines. We choose n = 9.

N = 2**n

sig = params[0]

n_cos_list = [6,7,8,9]
```

We define the COS function.

[16]:
```
def chi(k, a, b, c, d):
    return integrate.quad(lambda y: np.exp(y)*np.cos(k*math.pi*(y-a)/(b-a)), c,␣
 ↪d)[0]
def phi(k, a, b, c, d):
```

```
    return integrate.quad(lambda y: np.cos(k*math.pi*(y-a)/(b-a)),c,d)[0]
```

[17]:
```
def COS_Fourier(a,b,S0,r,q,sigma,T,K,n):
    N = 2**n

    A = np.zeros(N)
    V = np.zeros(N)

    for i in range(N):
        BS_CF = generic_CF(i*math.pi/(b-a), params, S0/K, r, q, T, model)
        A[i] = 2/(b-a)*np.real(BS_CF*np.exp(-1j*i*a*math.pi/(b-a)))

        V[i] = 2/(b-a)*K*(chi(i,a,b,0,b)-phi(i,a,b,0,b))

    V[0] = 0.5*V[0]

    # Here we choose C = e^(-r*T)
    return 0.5*(b-a)*sum(A*V)*math.exp(-r*T)
```

We calculate the results and print them.

[18]:
```
print(' ')
print('====================')
print('Model is %s' % model)
print('--------------------')

# COS-Fourier

print("||| COS-Fourier |||")

for K in K_list:

    for n in n_cos_list:

        for a, b in ab_list:

            start_time = time.time()
            cT_k = COS_Fourier(a,b,S0,r,q,params,T,K,n)
            elapsed_time = time.time() - start_time

            print('COS-Fourier execution time was %0.7f' % elapsed_time)
            print("K = ",K, " AND n = ",n, " AND interval [a, b] =␣
↪[",a,",",b,'].')

            print("Option via COS-Fourier: for strike %s the option premium is␣
↪%6.4f" % (K, cT_k))
```

```
==================
Model is BS
------------------
||| COS-Fourier |||
COS-Fourier execution time was 0.0295050
K =  2000  AND n =  6  AND interval [a, b] = [ -1 , 1 ].
Option via COS-Fourier: for strike 2000 the option premium is 95.2467
COS-Fourier execution time was 0.0192578
K =  2000  AND n =  6  AND interval [a, b] = [ -4 , 4 ].
Option via COS-Fourier: for strike 2000 the option premium is 95.2475
COS-Fourier execution time was 0.0165839
K =  2000  AND n =  6  AND interval [a, b] = [ -8 , 8 ].
Option via COS-Fourier: for strike 2000 the option premium is 94.9038
COS-Fourier execution time was 0.0144000
K =  2000  AND n =  6  AND interval [a, b] = [ -12 , 12 ].
Option via COS-Fourier: for strike 2000 the option premium is -22626.0999
COS-Fourier execution time was 0.0576713
K =  2000  AND n =  7  AND interval [a, b] = [ -1 , 1 ].
Option via COS-Fourier: for strike 2000 the option premium is 95.2467
COS-Fourier execution time was 0.0604022
K =  2000  AND n =  7  AND interval [a, b] = [ -4 , 4 ].
Option via COS-Fourier: for strike 2000 the option premium is 95.2467
COS-Fourier execution time was 0.0612612
K =  2000  AND n =  7  AND interval [a, b] = [ -8 , 8 ].
Option via COS-Fourier: for strike 2000 the option premium is 95.2699
COS-Fourier execution time was 0.0587881
K =  2000  AND n =  7  AND interval [a, b] = [ -12 , 12 ].
Option via COS-Fourier: for strike 2000 the option premium is 310.1706
COS-Fourier execution time was 0.2299390
K =  2000  AND n =  8  AND interval [a, b] = [ -1 , 1 ].
Option via COS-Fourier: for strike 2000 the option premium is 95.2467
COS-Fourier execution time was 0.2495701
K =  2000  AND n =  8  AND interval [a, b] = [ -4 , 4 ].
Option via COS-Fourier: for strike 2000 the option premium is 95.2467
COS-Fourier execution time was 0.2555289
K =  2000  AND n =  8  AND interval [a, b] = [ -8 , 8 ].
Option via COS-Fourier: for strike 2000 the option premium is 95.2467
COS-Fourier execution time was 0.2420471
K =  2000  AND n =  8  AND interval [a, b] = [ -12 , 12 ].
Option via COS-Fourier: for strike 2000 the option premium is 95.2469
COS-Fourier execution time was 0.8008699
K =  2000  AND n =  9  AND interval [a, b] = [ -1 , 1 ].
Option via COS-Fourier: for strike 2000 the option premium is 95.2467
COS-Fourier execution time was 0.8349721
K =  2000  AND n =  9  AND interval [a, b] = [ -4 , 4 ].
Option via COS-Fourier: for strike 2000 the option premium is 95.2467
COS-Fourier execution time was 0.8476579
K =  2000  AND n =  9  AND interval [a, b] = [ -8 , 8 ].
```

```
Option via COS-Fourier: for strike 2000 the option premium is 95.2467
COS-Fourier execution time was 0.8357751
K =  2000  AND n =  9  AND interval [a, b] = [ -12 , 12 ].
Option via COS-Fourier: for strike 2000 the option premium is 95.2467
COS-Fourier execution time was 0.0149550
K =  2100  AND n =  6  AND interval [a, b] = [ -1 , 1 ].
Option via COS-Fourier: for strike 2100 the option premium is 64.8346
COS-Fourier execution time was 0.0154748
K =  2100  AND n =  6  AND interval [a, b] = [ -4 , 4 ].
Option via COS-Fourier: for strike 2100 the option premium is 64.8358
COS-Fourier execution time was 0.0154741
K =  2100  AND n =  6  AND interval [a, b] = [ -8 , 8 ].
Option via COS-Fourier: for strike 2100 the option premium is 224.1476
COS-Fourier execution time was 0.0146322
K =  2100  AND n =  6  AND interval [a, b] = [ -12 , 12 ].
Option via COS-Fourier: for strike 2100 the option premium is 13516.6644
COS-Fourier execution time was 0.0578229
K =  2100  AND n =  7  AND interval [a, b] = [ -1 , 1 ].
Option via COS-Fourier: for strike 2100 the option premium is 64.8346
COS-Fourier execution time was 0.0615201
K =  2100  AND n =  7  AND interval [a, b] = [ -4 , 4 ].
Option via COS-Fourier: for strike 2100 the option premium is 64.8346
COS-Fourier execution time was 0.0626099
K =  2100  AND n =  7  AND interval [a, b] = [ -8 , 8 ].
Option via COS-Fourier: for strike 2100 the option premium is 64.8646
COS-Fourier execution time was 0.0616028
K =  2100  AND n =  7  AND interval [a, b] = [ -12 , 12 ].
Option via COS-Fourier: for strike 2100 the option premium is 761.9153
COS-Fourier execution time was 0.2332520
K =  2100  AND n =  8  AND interval [a, b] = [ -1 , 1 ].
Option via COS-Fourier: for strike 2100 the option premium is 64.8346
COS-Fourier execution time was 0.2491992
K =  2100  AND n =  8  AND interval [a, b] = [ -4 , 4 ].
Option via COS-Fourier: for strike 2100 the option premium is 64.8346
COS-Fourier execution time was 0.2640679
K =  2100  AND n =  8  AND interval [a, b] = [ -8 , 8 ].
Option via COS-Fourier: for strike 2100 the option premium is 64.8346
COS-Fourier execution time was 0.2489412
K =  2100  AND n =  8  AND interval [a, b] = [ -12 , 12 ].
Option via COS-Fourier: for strike 2100 the option premium is 64.8346
COS-Fourier execution time was 0.8098330
K =  2100  AND n =  9  AND interval [a, b] = [ -1 , 1 ].
Option via COS-Fourier: for strike 2100 the option premium is 64.8346
COS-Fourier execution time was 0.8505971
K =  2100  AND n =  9  AND interval [a, b] = [ -4 , 4 ].
Option via COS-Fourier: for strike 2100 the option premium is 64.8346
COS-Fourier execution time was 0.8631577
K =  2100  AND n =  9  AND interval [a, b] = [ -8 , 8 ].
```

```
Option via COS-Fourier: for strike 2100 the option premium is 64.8346
COS-Fourier execution time was 0.8320651
K =  2100  AND n =  9  AND interval [a, b] = [ -12 , 12 ].
Option via COS-Fourier: for strike 2100 the option premium is 64.8346
COS-Fourier execution time was 0.0144668
K =  2200  AND n =  6  AND interval [a, b] = [ -1 , 1 ].
Option via COS-Fourier: for strike 2200 the option premium is 42.9472
COS-Fourier execution time was 0.0150611
K =  2200  AND n =  6  AND interval [a, b] = [ -4 , 4 ].
Option via COS-Fourier: for strike 2200 the option premium is 42.9472
COS-Fourier execution time was 0.0147750
K =  2200  AND n =  6  AND interval [a, b] = [ -8 , 8 ].
Option via COS-Fourier: for strike 2200 the option premium is 312.9719
COS-Fourier execution time was 0.0142000
K =  2200  AND n =  6  AND interval [a, b] = [ -12 , 12 ].
Option via COS-Fourier: for strike 2200 the option premium is 49139.3478
COS-Fourier execution time was 0.0572298
K =  2200  AND n =  7  AND interval [a, b] = [ -1 , 1 ].
Option via COS-Fourier: for strike 2200 the option premium is 42.9472
COS-Fourier execution time was 0.0603919
K =  2200  AND n =  7  AND interval [a, b] = [ -4 , 4 ].
Option via COS-Fourier: for strike 2200 the option premium is 42.9472
COS-Fourier execution time was 0.0612371
K =  2200  AND n =  7  AND interval [a, b] = [ -8 , 8 ].
Option via COS-Fourier: for strike 2200 the option premium is 42.9452
COS-Fourier execution time was 0.0612922
K =  2200  AND n =  7  AND interval [a, b] = [ -12 , 12 ].
Option via COS-Fourier: for strike 2200 the option premium is 817.2623
COS-Fourier execution time was 0.2307141
K =  2200  AND n =  8  AND interval [a, b] = [ -1 , 1 ].
Option via COS-Fourier: for strike 2200 the option premium is 42.9472
COS-Fourier execution time was 0.2438228
K =  2200  AND n =  8  AND interval [a, b] = [ -4 , 4 ].
Option via COS-Fourier: for strike 2200 the option premium is 42.9472
COS-Fourier execution time was 0.2570188
K =  2200  AND n =  8  AND interval [a, b] = [ -8 , 8 ].
Option via COS-Fourier: for strike 2200 the option premium is 42.9472
COS-Fourier execution time was 0.2397220
K =  2200  AND n =  8  AND interval [a, b] = [ -12 , 12 ].
Option via COS-Fourier: for strike 2200 the option premium is 42.9469
COS-Fourier execution time was 0.7927642
K =  2200  AND n =  9  AND interval [a, b] = [ -1 , 1 ].
Option via COS-Fourier: for strike 2200 the option premium is 42.9472
COS-Fourier execution time was 0.8261859
K =  2200  AND n =  9  AND interval [a, b] = [ -4 , 4 ].
Option via COS-Fourier: for strike 2200 the option premium is 42.9472
COS-Fourier execution time was 0.8441210
K =  2200  AND n =  9  AND interval [a, b] = [ -8 , 8 ].
```

```
Option via COS-Fourier: for strike 2200 the option premium is 42.9472
COS-Fourier execution time was 0.8262930
K =  2200  AND n =  9  AND interval [a, b] = [ -12 , 12 ].
Option via COS-Fourier: for strike 2200 the option premium is 42.9472
```

What happens if we take bigger value of a and b (in absolute value)?

First let us create a new list.

[19]:
```
ab_list_sensitivity = [(-16, 16), (-22, 22), (-28, 28), (-34, 34)]
```

Now let us calculate the outputs.

[20]:
```
for K in K_list:

    for n in n_cos_list:

        for a, b in ab_list_sensitivity:

                start_time = time.time()
                cT_k = COS_Fourier(a,b,S0,r,q,params,T,K,n)
                elapsed_time = time.time() - start_time

                print('COS-Fourier execution time was %0.7f' % elapsed_time)
                print("K = ",K, " AND n = ",n, " AND interval [a, b] =␣
↪[",a,",",b,'].')

                print("Option via COS-Fourier: for strike %s the option premium␣
↪is %6.4f" % (K, cT_k))
```

```
COS-Fourier execution time was 0.0144742
K =  2000  AND n =  6  AND interval [a, b] = [ -16 , 16 ].
Option via COS-Fourier: for strike 2000 the option premium is -3972177.5854
COS-Fourier execution time was 0.0135472
K =  2000  AND n =  6  AND interval [a, b] = [ -22 , 22 ].
Option via COS-Fourier: for strike 2000 the option premium is -3651513946.3390
COS-Fourier execution time was 0.0133030
K =  2000  AND n =  6  AND interval [a, b] = [ -28 , 28 ].
Option via COS-Fourier: for strike 2000 the option premium is
-2304922580373.5674
COS-Fourier execution time was 0.0123420
K =  2000  AND n =  6  AND interval [a, b] = [ -34 , 34 ].
Option via COS-Fourier: for strike 2000 the option premium is
-1240503042752099.0000
COS-Fourier execution time was 0.0561810
K =  2000  AND n =  7  AND interval [a, b] = [ -16 , 16 ].
Option via COS-Fourier: for strike 2000 the option premium is 10958.3670
COS-Fourier execution time was 0.0522969
K =  2000  AND n =  7  AND interval [a, b] = [ -22 , 22 ].
```

```
Option via COS-Fourier: for strike 2000 the option premium is -135968756.3525
COS-Fourier execution time was 0.0491509
K =  2000  AND n =  7  AND interval [a, b] = [ -28 , 28 ].
Option via COS-Fourier: for strike 2000 the option premium is -190437571417.1243
COS-Fourier execution time was 0.0475211
K =  2000  AND n =  7  AND interval [a, b] = [ -34 , 34 ].
Option via COS-Fourier: for strike 2000 the option premium is
-150511609673646.1875
COS-Fourier execution time was 0.2306118
K =  2000  AND n =  8  AND interval [a, b] = [ -16 , 16 ].
Option via COS-Fourier: for strike 2000 the option premium is 130.1292
COS-Fourier execution time was 0.2136872
K =  2000  AND n =  8  AND interval [a, b] = [ -22 , 22 ].
Option via COS-Fourier: for strike 2000 the option premium is 1258429.3791
COS-Fourier execution time was 0.1947579
K =  2000  AND n =  8  AND interval [a, b] = [ -28 , 28 ].
Option via COS-Fourier: for strike 2000 the option premium is 2013034633.1885
COS-Fourier execution time was 0.1752303
K =  2000  AND n =  8  AND interval [a, b] = [ -34 , 34 ].
Option via COS-Fourier: for strike 2000 the option premium is -93810367089.7833
COS-Fourier execution time was 0.8048408
K =  2000  AND n =  9  AND interval [a, b] = [ -16 , 16 ].
Option via COS-Fourier: for strike 2000 the option premium is 95.2467
COS-Fourier execution time was 0.7778709
K =  2000  AND n =  9  AND interval [a, b] = [ -22 , 22 ].
Option via COS-Fourier: for strike 2000 the option premium is 95.3202
COS-Fourier execution time was 0.7320251
K =  2000  AND n =  9  AND interval [a, b] = [ -28 , 28 ].
Option via COS-Fourier: for strike 2000 the option premium is 127040.1647
COS-Fourier execution time was 0.6554110
K =  2000  AND n =  9  AND interval [a, b] = [ -34 , 34 ].
Option via COS-Fourier: for strike 2000 the option premium is 3576992244.1149
COS-Fourier execution time was 0.0140831
K =  2100  AND n =  6  AND interval [a, b] = [ -16 , 16 ].
Option via COS-Fourier: for strike 2100 the option premium is -951947.2412
COS-Fourier execution time was 0.0133348
K =  2100  AND n =  6  AND interval [a, b] = [ -22 , 22 ].
Option via COS-Fourier: for strike 2100 the option premium is -2200159656.8207
COS-Fourier execution time was 0.0129550
K =  2100  AND n =  6  AND interval [a, b] = [ -28 , 28 ].
Option via COS-Fourier: for strike 2100 the option premium is
-1719317188132.7969
COS-Fourier execution time was 0.0124171
K =  2100  AND n =  6  AND interval [a, b] = [ -34 , 34 ].
Option via COS-Fourier: for strike 2100 the option premium is
-1019061930496634.7500
COS-Fourier execution time was 0.0557230
K =  2100  AND n =  7  AND interval [a, b] = [ -16 , 16 ].
```

```
Option via COS-Fourier: for strike 2100 the option premium is 243619.5104
COS-Fourier execution time was 0.0522242
K =  2100  AND n =  7  AND interval [a, b] = [ -22 , 22 ].
Option via COS-Fourier: for strike 2100 the option premium is 186184456.7783
COS-Fourier execution time was 0.0492611
K =  2100  AND n =  7  AND interval [a, b] = [ -28 , 28 ].
Option via COS-Fourier: for strike 2100 the option premium is 19044118135.7706
COS-Fourier execution time was 0.0458009
K =  2100  AND n =  7  AND interval [a, b] = [ -34 , 34 ].
Option via COS-Fourier: for strike 2100 the option premium is
-46049205189711.2109
COS-Fourier execution time was 0.2296541
K =  2100  AND n =  8  AND interval [a, b] = [ -16 , 16 ].
Option via COS-Fourier: for strike 2100 the option premium is 107.0809
COS-Fourier execution time was 0.2148600
K =  2100  AND n =  8  AND interval [a, b] = [ -22 , 22 ].
Option via COS-Fourier: for strike 2100 the option premium is 3156311.1601
COS-Fourier execution time was 0.1959560
K =  2100  AND n =  8  AND interval [a, b] = [ -28 , 28 ].
Option via COS-Fourier: for strike 2100 the option premium is 10008432780.2792
COS-Fourier execution time was 0.1738093
K =  2100  AND n =  8  AND interval [a, b] = [ -34 , 34 ].
Option via COS-Fourier: for strike 2100 the option premium is
10184912355600.4883
COS-Fourier execution time was 0.8046620
K =  2100  AND n =  9  AND interval [a, b] = [ -16 , 16 ].
Option via COS-Fourier: for strike 2100 the option premium is 64.8346
COS-Fourier execution time was 0.7775621
K =  2100  AND n =  9  AND interval [a, b] = [ -22 , 22 ].
Option via COS-Fourier: for strike 2100 the option premium is 64.8205
COS-Fourier execution time was 0.7322230
K =  2100  AND n =  9  AND interval [a, b] = [ -28 , 28 ].
Option via COS-Fourier: for strike 2100 the option premium is 87153.0695
COS-Fourier execution time was 0.6655149
K =  2100  AND n =  9  AND interval [a, b] = [ -34 , 34 ].
Option via COS-Fourier: for strike 2100 the option premium is 4972440677.3012
COS-Fourier execution time was 0.0140889
K =  2200  AND n =  6  AND interval [a, b] = [ -16 , 16 ].
Option via COS-Fourier: for strike 2200 the option premium is 2292868.5817
COS-Fourier execution time was 0.0133829
K =  2200  AND n =  6  AND interval [a, b] = [ -22 , 22 ].
Option via COS-Fourier: for strike 2200 the option premium is -572567415.2473
COS-Fourier execution time was 0.0129220
K =  2200  AND n =  6  AND interval [a, b] = [ -28 , 28 ].
Option via COS-Fourier: for strike 2200 the option premium is
-1052140240676.7181
COS-Fourier execution time was 0.0123992
K =  2200  AND n =  6  AND interval [a, b] = [ -34 , 34 ].
```

```
Option via COS-Fourier: for strike 2200 the option premium is
-764645991289541.0000
COS-Fourier execution time was 0.0558150
K =  2200  AND n =  7  AND interval [a, b] = [ -16 , 16 ].
Option via COS-Fourier: for strike 2200 the option premium is 401235.2585
COS-Fourier execution time was 0.0524039
K =  2200  AND n =  7  AND interval [a, b] = [ -22 , 22 ].
Option via COS-Fourier: for strike 2200 the option premium is 488237462.7271
COS-Fourier execution time was 0.0492470
K =  2200  AND n =  7  AND interval [a, b] = [ -28 , 28 ].
Option via COS-Fourier: for strike 2200 the option premium is 236293458034.8820
COS-Fourier execution time was 0.0463579
K =  2200  AND n =  7  AND interval [a, b] = [ -34 , 34 ].
Option via COS-Fourier: for strike 2200 the option premium is
67201329746769.2188
COS-Fourier execution time was 0.2309320
K =  2200  AND n =  8  AND interval [a, b] = [ -16 , 16 ].
Option via COS-Fourier: for strike 2200 the option premium is 37.6215
COS-Fourier execution time was 0.2161758
K =  2200  AND n =  8  AND interval [a, b] = [ -22 , 22 ].
Option via COS-Fourier: for strike 2200 the option premium is 2855745.7521
COS-Fourier execution time was 0.1958311
K =  2200  AND n =  8  AND interval [a, b] = [ -28 , 28 ].
Option via COS-Fourier: for strike 2200 the option premium is 13805422196.9942
COS-Fourier execution time was 0.1752481
K =  2200  AND n =  8  AND interval [a, b] = [ -34 , 34 ].
Option via COS-Fourier: for strike 2200 the option premium is
17757036589908.0625
COS-Fourier execution time was 0.8054512
K =  2200  AND n =  9  AND interval [a, b] = [ -16 , 16 ].
Option via COS-Fourier: for strike 2200 the option premium is 42.9472
COS-Fourier execution time was 0.7780240
K =  2200  AND n =  9  AND interval [a, b] = [ -22 , 22 ].
Option via COS-Fourier: for strike 2200 the option premium is 42.8827
COS-Fourier execution time was 0.7337608
K =  2200  AND n =  9  AND interval [a, b] = [ -28 , 28 ].
Option via COS-Fourier: for strike 2200 the option premium is -100897.8202
COS-Fourier execution time was 0.6558900
K =  2200  AND n =  9  AND interval [a, b] = [ -34 , 34 ].
Option via COS-Fourier: for strike 2200 the option premium is 594313563.9337
```

Let us now consider the integrated expression of chi and phi to speed up the calculation. This is based on the article mentioned above.

[21]:
```
def chi_integrated(k, a, b, c, d):
    u  = k * np.pi/(b-a)
    chi = 1/(1+u**2)*(np.cos(u * (d-a)) * np.exp(d) - np.cos(u * (c-a)) * np.
 ↪exp(c) + u*np.sin(u * (d-a)) * np.exp(d)- u*np.sin(u * (c-a)) * np.exp(c))
```

```
    return chi


def phi_integrated(k, a, b, c, d):
    if k==0:
        psi = d-c
    else:
        u     = k * np.pi/(b-a)
        psi    = 1/u * ( np.sin(u * (d-a)) - np.sin(u * (c-a)) )

    return psi
```

[22]:
```
def COS_Fourier_integrated(a,b,S0,r,q,sigma,T,K,n):
    N = 2**n

    A = np.zeros(N)
    V = np.zeros(N)

    for i in range(N):
        BS_CF = generic_CF(i*math.pi/(b-a), params, S0/K, r, q, T, model)
        A[i] = 2/(b-a)*np.real(BS_CF*np.exp(-1j*i*a*math.pi/(b-a)))

        V[i] = 2/(b-a)*K*(chi_integrated(i,a,b,0,b)-phi_integrated(i,a,b,0,b))

    V[0] = 0.5*V[0]

    # Here we choose C = e^(-r*T)
    return 0.5*(b-a)*sum(A*V)*math.exp(-r*T)
```

[23]:
```
print(' ')
print('====================')
print('Model is %s' % model)
print('--------------------')

# COS-Fourier

print("||| COS-Fourier-Integrated |||")

for K in K_list:

    for n in n_cos_list:

        for a, b in ab_list:

            start_time = time.time()
            cT_k = COS_Fourier_integrated(a,b,S0,r,q,params,T,K,n)
            elapsed_time = time.time() - start_time
```

```
            print('COS-Fourier execution time was %0.7f' % elapsed_time)
            print("K = ",K, " AND n = ",n, " AND interval [a, b] =␣
↪[",a,",",b,'].')

            print("Option via COS-Fourier: for strike %s the option premium is␣
↪%6.4f" % (K, cT_k))
```

```
====================
Model is BS
--------------------
||| COS-Fourier-Integrated |||
COS-Fourier execution time was 0.0005379
K =  2000  AND n =  6  AND interval [a, b] = [ -1 , 1 ].
Option via COS-Fourier: for strike 2000 the option premium is 95.2467
COS-Fourier execution time was 0.0005519
K =  2000  AND n =  6  AND interval [a, b] = [ -4 , 4 ].
Option via COS-Fourier: for strike 2000 the option premium is 95.2475
COS-Fourier execution time was 0.0005040
K =  2000  AND n =  6  AND interval [a, b] = [ -8 , 8 ].
Option via COS-Fourier: for strike 2000 the option premium is 94.9038
COS-Fourier execution time was 0.0005240
K =  2000  AND n =  6  AND interval [a, b] = [ -12 , 12 ].
Option via COS-Fourier: for strike 2000 the option premium is -22626.0999
COS-Fourier execution time was 0.0010159
K =  2000  AND n =  7  AND interval [a, b] = [ -1 , 1 ].
Option via COS-Fourier: for strike 2000 the option premium is 95.2467
COS-Fourier execution time was 0.0010121
K =  2000  AND n =  7  AND interval [a, b] = [ -4 , 4 ].
Option via COS-Fourier: for strike 2000 the option premium is 95.2467
COS-Fourier execution time was 0.0011191
K =  2000  AND n =  7  AND interval [a, b] = [ -8 , 8 ].
Option via COS-Fourier: for strike 2000 the option premium is 95.2699
COS-Fourier execution time was 0.0010421
K =  2000  AND n =  7  AND interval [a, b] = [ -12 , 12 ].
Option via COS-Fourier: for strike 2000 the option premium is 310.1706
COS-Fourier execution time was 0.0020382
K =  2000  AND n =  8  AND interval [a, b] = [ -1 , 1 ].
Option via COS-Fourier: for strike 2000 the option premium is 95.2467
COS-Fourier execution time was 0.0018749
K =  2000  AND n =  8  AND interval [a, b] = [ -4 , 4 ].
Option via COS-Fourier: for strike 2000 the option premium is 95.2467
COS-Fourier execution time was 0.0018859
K =  2000  AND n =  8  AND interval [a, b] = [ -8 , 8 ].
Option via COS-Fourier: for strike 2000 the option premium is 95.2467
COS-Fourier execution time was 0.0018921
```

```
K =  2000  AND n =  8  AND interval [a, b] = [ -12 , 12 ].
Option via COS-Fourier: for strike 2000 the option premium is 95.2469
COS-Fourier execution time was 0.0038242
K =  2000  AND n =  9  AND interval [a, b] = [ -1 , 1 ].
Option via COS-Fourier: for strike 2000 the option premium is 95.2467
COS-Fourier execution time was 0.0037420
K =  2000  AND n =  9  AND interval [a, b] = [ -4 , 4 ].
Option via COS-Fourier: for strike 2000 the option premium is 95.2467
COS-Fourier execution time was 0.0038991
K =  2000  AND n =  9  AND interval [a, b] = [ -8 , 8 ].
Option via COS-Fourier: for strike 2000 the option premium is 95.2467
COS-Fourier execution time was 0.0040450
K =  2000  AND n =  9  AND interval [a, b] = [ -12 , 12 ].
Option via COS-Fourier: for strike 2000 the option premium is 95.2467
COS-Fourier execution time was 0.0004773
K =  2100  AND n =  6  AND interval [a, b] = [ -1 , 1 ].
Option via COS-Fourier: for strike 2100 the option premium is 64.8346
COS-Fourier execution time was 0.0004711
K =  2100  AND n =  6  AND interval [a, b] = [ -4 , 4 ].
Option via COS-Fourier: for strike 2100 the option premium is 64.8358
COS-Fourier execution time was 0.0004711
K =  2100  AND n =  6  AND interval [a, b] = [ -8 , 8 ].
Option via COS-Fourier: for strike 2100 the option premium is 224.1476
COS-Fourier execution time was 0.0004721
K =  2100  AND n =  6  AND interval [a, b] = [ -12 , 12 ].
Option via COS-Fourier: for strike 2100 the option premium is 13516.6644
COS-Fourier execution time was 0.0009301
K =  2100  AND n =  7  AND interval [a, b] = [ -1 , 1 ].
Option via COS-Fourier: for strike 2100 the option premium is 64.8346
COS-Fourier execution time was 0.0009511
K =  2100  AND n =  7  AND interval [a, b] = [ -4 , 4 ].
Option via COS-Fourier: for strike 2100 the option premium is 64.8346
COS-Fourier execution time was 0.0009241
K =  2100  AND n =  7  AND interval [a, b] = [ -8 , 8 ].
Option via COS-Fourier: for strike 2100 the option premium is 64.8646
COS-Fourier execution time was 0.0009308
K =  2100  AND n =  7  AND interval [a, b] = [ -12 , 12 ].
Option via COS-Fourier: for strike 2100 the option premium is 761.9153
COS-Fourier execution time was 0.0018530
K =  2100  AND n =  8  AND interval [a, b] = [ -1 , 1 ].
Option via COS-Fourier: for strike 2100 the option premium is 64.8346
COS-Fourier execution time was 0.0018339
K =  2100  AND n =  8  AND interval [a, b] = [ -4 , 4 ].
Option via COS-Fourier: for strike 2100 the option premium is 64.8346
COS-Fourier execution time was 0.0018549
K =  2100  AND n =  8  AND interval [a, b] = [ -8 , 8 ].
Option via COS-Fourier: for strike 2100 the option premium is 64.8346
COS-Fourier execution time was 0.0018592
```

```
K =  2100  AND n =  8  AND interval [a, b] = [ -12 , 12 ].
Option via COS-Fourier: for strike 2100 the option premium is 64.8346
COS-Fourier execution time was 0.0038211
K =  2100  AND n =  9  AND interval [a, b] = [ -1 , 1 ].
Option via COS-Fourier: for strike 2100 the option premium is 64.8346
COS-Fourier execution time was 0.0038018
K =  2100  AND n =  9  AND interval [a, b] = [ -4 , 4 ].
Option via COS-Fourier: for strike 2100 the option premium is 64.8346
COS-Fourier execution time was 0.0037322
K =  2100  AND n =  9  AND interval [a, b] = [ -8 , 8 ].
Option via COS-Fourier: for strike 2100 the option premium is 64.8346
COS-Fourier execution time was 0.0037031
K =  2100  AND n =  9  AND interval [a, b] = [ -12 , 12 ].
Option via COS-Fourier: for strike 2100 the option premium is 64.8346
COS-Fourier execution time was 0.0004652
K =  2200  AND n =  6  AND interval [a, b] = [ -1 , 1 ].
Option via COS-Fourier: for strike 2200 the option premium is 42.9472
COS-Fourier execution time was 0.0004630
K =  2200  AND n =  6  AND interval [a, b] = [ -4 , 4 ].
Option via COS-Fourier: for strike 2200 the option premium is 42.9472
COS-Fourier execution time was 0.0004678
K =  2200  AND n =  6  AND interval [a, b] = [ -8 , 8 ].
Option via COS-Fourier: for strike 2200 the option premium is 312.9719
COS-Fourier execution time was 0.0004652
K =  2200  AND n =  6  AND interval [a, b] = [ -12 , 12 ].
Option via COS-Fourier: for strike 2200 the option premium is 49139.3478
COS-Fourier execution time was 0.0009279
K =  2200  AND n =  7  AND interval [a, b] = [ -1 , 1 ].
Option via COS-Fourier: for strike 2200 the option premium is 42.9472
COS-Fourier execution time was 0.0009279
K =  2200  AND n =  7  AND interval [a, b] = [ -4 , 4 ].
Option via COS-Fourier: for strike 2200 the option premium is 42.9472
COS-Fourier execution time was 0.0009243
K =  2200  AND n =  7  AND interval [a, b] = [ -8 , 8 ].
Option via COS-Fourier: for strike 2200 the option premium is 42.9452
COS-Fourier execution time was 0.0009620
K =  2200  AND n =  7  AND interval [a, b] = [ -12 , 12 ].
Option via COS-Fourier: for strike 2200 the option premium is 817.2623
COS-Fourier execution time was 0.0018580
K =  2200  AND n =  8  AND interval [a, b] = [ -1 , 1 ].
Option via COS-Fourier: for strike 2200 the option premium is 42.9472
COS-Fourier execution time was 0.0018470
K =  2200  AND n =  8  AND interval [a, b] = [ -4 , 4 ].
Option via COS-Fourier: for strike 2200 the option premium is 42.9472
COS-Fourier execution time was 0.0018449
K =  2200  AND n =  8  AND interval [a, b] = [ -8 , 8 ].
Option via COS-Fourier: for strike 2200 the option premium is 42.9472
COS-Fourier execution time was 0.0018427
```

```
K =  2200  AND n =  8  AND interval [a, b] = [ -12 , 12 ].
Option via COS-Fourier: for strike 2200 the option premium is 42.9469
COS-Fourier execution time was 0.0038009
K =  2200  AND n =  9  AND interval [a, b] = [ -1 , 1 ].
Option via COS-Fourier: for strike 2200 the option premium is 42.9472
COS-Fourier execution time was 0.0036910
K =  2200  AND n =  9  AND interval [a, b] = [ -4 , 4 ].
Option via COS-Fourier: for strike 2200 the option premium is 42.9472
COS-Fourier execution time was 0.0036912
K =  2200  AND n =  9  AND interval [a, b] = [ -8 , 8 ].
Option via COS-Fourier: for strike 2200 the option premium is 42.9472
COS-Fourier execution time was 0.0037019
K =  2200  AND n =  9  AND interval [a, b] = [ -12 , 12 ].
Option via COS-Fourier: for strike 2200 the option premium is 42.9472
```