EE2703 : Applied Programming Lab
Assignment 3

Andapally Snehitha
EE20B008

March 11, 2022

Function Definition and Visualization

We need to ensure that the function can output correct values even for an N-dimensional vector as an input. The built-in functions in numpy can be used to accomplish the same :

```
import numpy as np
    def expo(x):
        return np.exp(x)
    def coscos(x):
        return np.cos(np.cos(x))
```

The above functions are plotted over the interval $[2\pi, 4\pi)$ using 300 points sampled uniformly over this interval. The following plots are obtained :

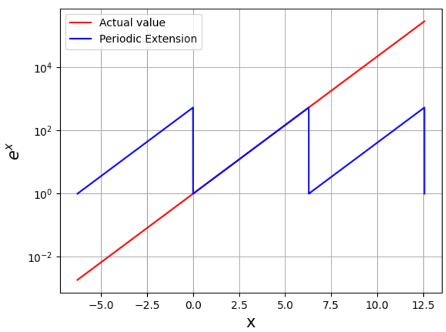

Figure 1: e^x in semi-log scale along with expected Fourier e^x

The function e^x is ever increasing with x and is non-periodic. However for the evaluation of the Fourier series, the function is made 2π periodic. cos(x) is periodic and thus cos(cos(x)) is also periodic. Since cos(x) $=$ cos(x), the period of cos(cos(x)) is π .

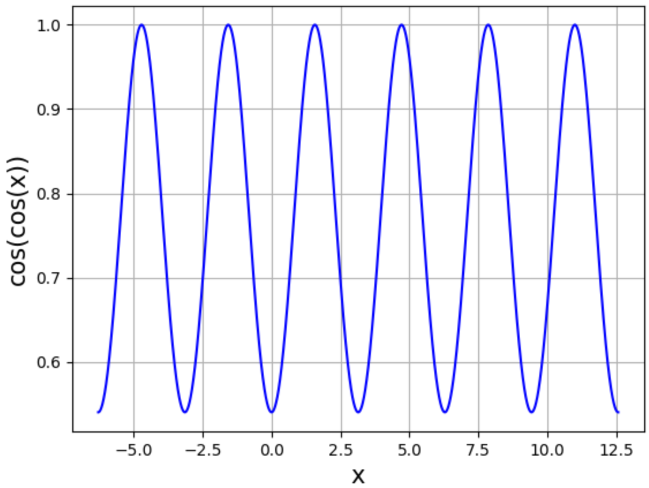

Figure 2: Plot of cos(cos(x)) in linear scale

The Fourier series gives us the expression for the function in terms of 2π periodic sinusoids .For an aperiodic function, it is calculated by looking at the function in the interval $[0,\ 2^*\pi)$ and repeating the function pattern so that we obtain a 2π periodic function for which the Fourier series can be evaluated. e^x is a function with a swift increase and this can be represented only by high frequency sinusoids. The Fourier series would only yield a reasonable approximation even for sufficiently large number of coefficients. Since cos(cos(x)) has a period of π , we can expect the Fourier series expansion to yield an almost exact approximation for the function.

The Fourier Series Coefficients

The Fourier series coefficients are obtained using the quad function in the scipy library. The following code snippet evaluates the first n coefficients of the Fourier series expansion of e^x and cos(cos(x)):

```
def u(x,k):
    return expo(x)*np.cos(k*x)
def v(x,k):
    return expo(x)*np.sin(k*x)
def p(x,k):
    return coscos(x)*np.cos(k*x)
def q(x,k):
    return coscos(x)*np.sin(k*x)

```

```
expo_a0= (integrate.quad(u,0,2*np.pi,args=(0,))[0])/(2*np.pi)
expo_a = np.array([integrate.quad(u,0,2*np.pi,args=(i,))[0]
\newline
for i in range(1,26)])/(np.pi)
expo_b = np.array([integrate.quad(v,0,2*np.pi,args=(i,))[0]
for i in range(1,26)])/(np.pi)

coscos_a0= (integrate.quad(p,0,2*np.pi,args=(0,))[0])/(2*np.
    pi)
coscos_a = np.array([integrate.quad(p,0,2*np.pi,args=(i,))[0]
     for i in range(1,26)])/(np.pi)
coscos_b = np.array([integrate.quad(q,0,2*np.pi,args=(i,))[0]
     for i in range(1,26)])/(np.pi)
```

We thus obtain the first 51 coefficient of the two functions. A plot of the coefficients of e^x in a semilog and loglog scale is :

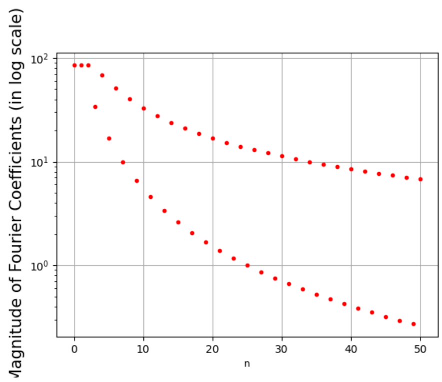

Figure 3: Fourier Coefficients of e^x by direct integration in semilog scale

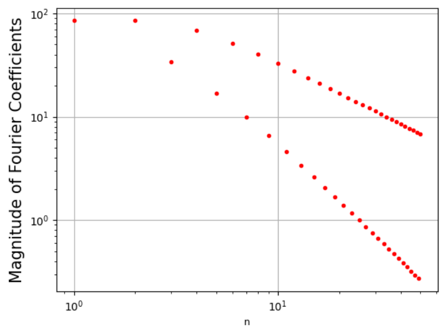

Figure 4: Fourier Coefficients of e^x by direct integration in loglog scale

Similarly, the plots for cos(cos(x)) are as follows :

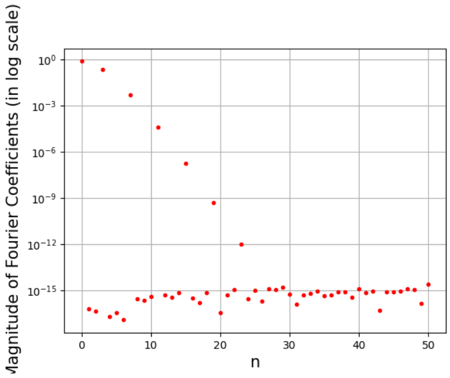

Figure 5: Fourier Coefficients of cos(cos(x)) by direct integration in semilog scale

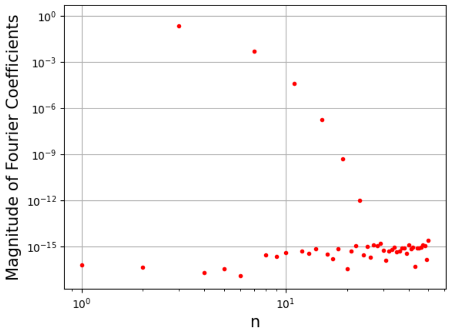

Figure 6: Fourier Coefficients of cos(cos(x)) by direct integration in loglog scale

We notice that the bn coefficients in the second case are expected to be zero as the function cos(cos(x)) is even and thus the odd sinusoidal components are zero. The values obtained are non-zero because of the limitation in the numerical accuracy upto which π can be stored in memory. In the first case, the function, having an exponentially increasing gradient, contains a wide range of frequencies in its fourier series. In the second case, cos(cos(x)) has a relatively low frequency of $1/\pi$ and thus contribution by the higher sinusoids is less, manifesting in the quick decay of the coefficients with n.

The Least Squares Approach

The original functions are tried to be reconstructed using the Fourier expansion upto first 51 terms. The Fourier coefficients are calculated using np.linalg.lstsq function.

```
X = np.linspace(0,2*np.pi,401)
X=X[:-1]
b1 = expo(X)
b2 = coscos(X)
A = np.zeros((400,51))
A[:,0] = 1

for k in range(1,26):
    A[:,2*k-1] = np.cos(k*X)
    A[:,2*k] = np.sin(k*X)

c1 = np.linalg.lstsq(A,b1,rcond = -1)[0]
c2 = np.linalg.lstsq(A,b2,rcond = -1)[0]
```

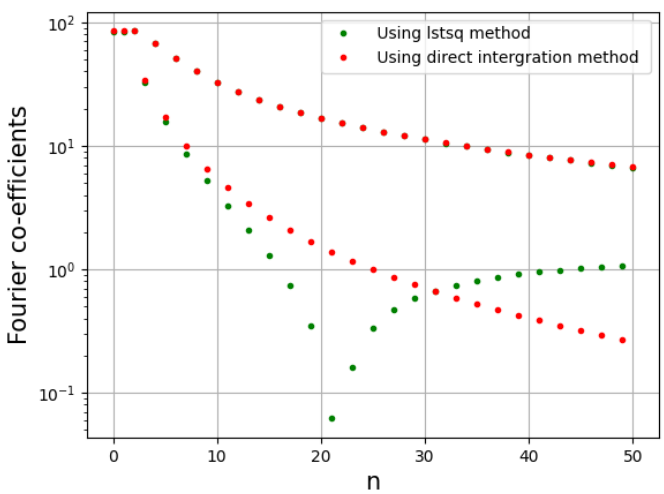

Figure 7: Fourier co-efficients of e^x in semilog scale

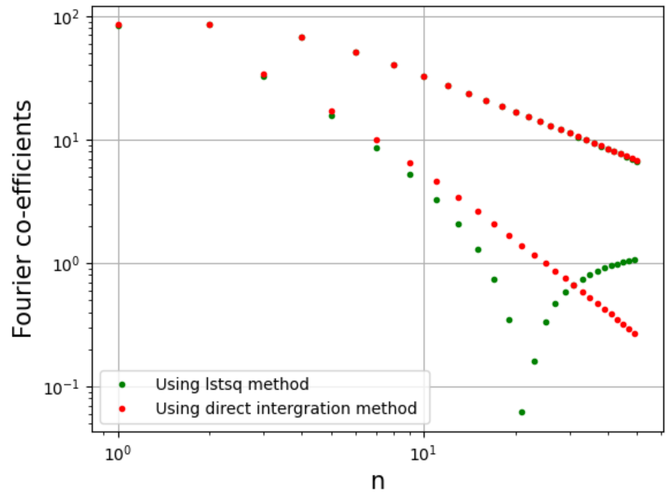

Figure 8: Fourier co-efficients of e^x in loglog scale

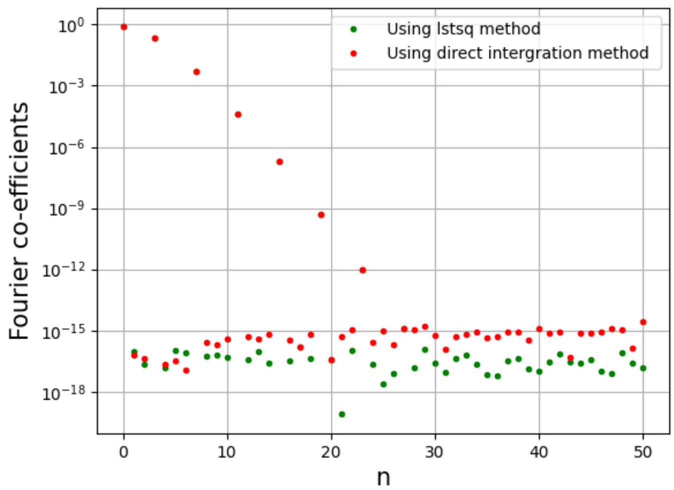

Figure 9: Fourier co-efficients of cos(cos(x)) in semilog scale

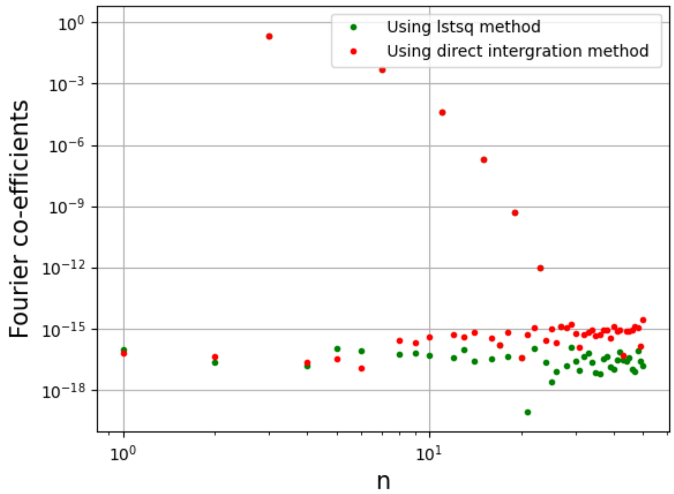

Figure 10: Fourier co-efficients of cos(cos(x)) in loglog scale

Plots of Reconstructed Functions:

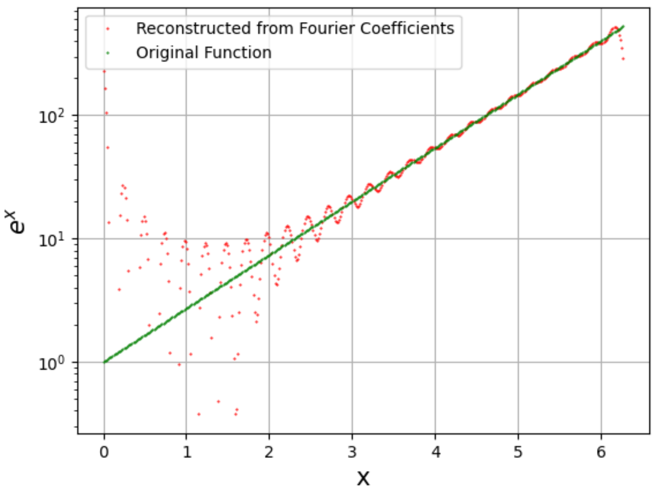

Figure 11: Original function e^x and the Reconstructed function from Fourier Coefficients.

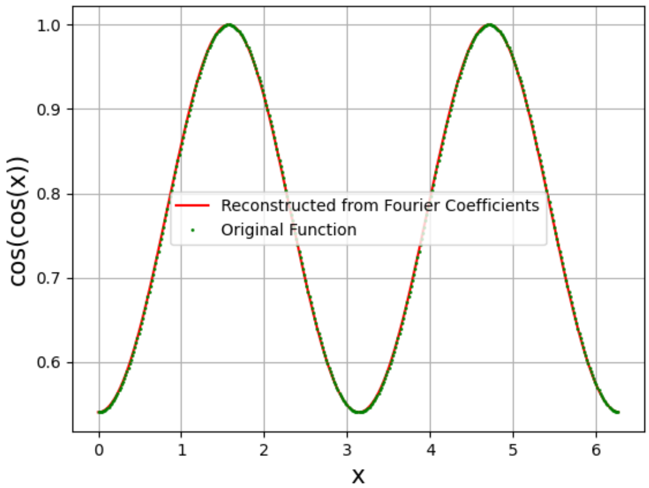

Figure 12: Original function cos(cos(x)) and the Reconstructed function from Fourier Coefficients.

Error Analysis

The error is calculated as follows:

```
print("Absolute difference between Fourier Coefficients in expo(x) = exp(x) is {}".format(np.sum(abs(c1 - F))))
print("Largest Deviation between Fourier Coefficients in expo(x) = exp(x) is {}".format(np.amax(abs(c1 - F))))
print("Mean Error in expo(x) = exp(x) is {}".format(np.mean(abs(c1 - F))))

print("Absolute difference between Fourier Coefficients in coscos(x) = cos(cos(x)) is {}".format(np.sum(abs(c2 - G))))
print("Largest Deviation between Fourier Coefficients in coscos(x) = cos(cos(x)) is {}".format(np.amax(abs(c2 - G))))
print("Mean Error in coscos(x) = cos(cos(x)) is {}".format(np.mean(abs(c2 - G))))
```

Maximum Deviation:

- The Maximum deviation in f(x) = e^x in between the coefficients estimated through LSTSQ method and Direct Integration Method is 1.3327308703354106 , which occurs at a25
- The Maximum deviation in g(x) = cos(cos(x)) in between the coefficients estimated through LSTSQ method and Direct Integration Method is 2.656286271711572e-15 , which occurs at a25

Absolute Error:

- The absolute error in f(x) = e^x in between the coefficients estimated through LSTSQ method and Direct Integration Method is 35.12283611763861
- The absolute error in g(x) = cos(cos(x)) in between the coefficients estimated through LSTSQ method and Direct Integration Method is 2.8114348359621343e-14

Mean Error:

- The Mean Error in f(x) = e^x in between the coefficients estimated through LSTSQ method and Direct Integration Method is 0.6886830611301689
- The Mean Error in g(x) = cos(cos(x)) in between the coefficients estimated through LSTSQ method and Direct Integration Method is 5.51261732541595e-16

Result:

- The function cos(cos(x)) is periodic while e^x is not periodic but for e^x we get a periodic extension using Fourier analysis.
- Coefficients bn of cos(cos(x)) are expected to be 0 as the function is even and bn are coefficients of odd sinusoidal components but the small values of bn are due to error in integration.
- The coefficients of e^x decay slowly compared to cos(cos(x)) as the function is not periodic in the given interval of [0,2π).

Conclusion:

We calculated the Fourier Expansion of two functions f(x) = e^x and g(x) = cos(cos(x)) and calculated the deviation the reconstructed function from the actual function.