EE2703 : Applied Programming Lab Assignment 4

P N Neelesh Sumedh
ee19b047

March 10, 2021

1 Abstract

We will fit two functions, e^x and $cos(cos(x))$ over the interval $[0, 2\pi)$ using the fourier series

2 Introduction

The Fourier Series of a function $f(x)$ with period $2 * \pi$ is computed as:

$$a_0 + \sum_{n=1}^{\infty} a_n cos(nx) + b_n sin(nx) \tag{1}$$

where

$$a_0 = \frac{1}{2\pi}\int_0^{2\pi} f(x)dx$$
$$a_n = \frac{1}{\pi}\int_0^{2\pi} f(x) * cos(x)dx$$
$$b_n = \frac{1}{\pi}\int_0^{2\pi} f(x) * sin(x)dx$$

The function e^x is not periodic. So we take only $[0, 2\pi]$ part of the function and extend it to make it 2π periodic. The function $\cos(\cos(x))$ is 2π periodic and no need of any extension.

3 Assignment Questions

3.1 Writing functions of e^x and $\cos(\cos(x))$

The functions are defined as follows:

```
def exp(In):
    flr = np.floor(In/(2*np.pi))
    In = In - (2*np.pi)*flr
    out = np.exp(In)
    return out


def coscos(In):
    temp = np.cos(In)
```

```
out = np.cos(temp)
return out
```

We know that Fourier series always generates function which are periodic. But the function e^x is not periodic. Therefore the plot of e^x is different from the actual plot as seen in the Figure 1. The plot of $\cos(\cos(x))$ coincides with the expected plot as the function is periodic. This can be seen in Figure 2.

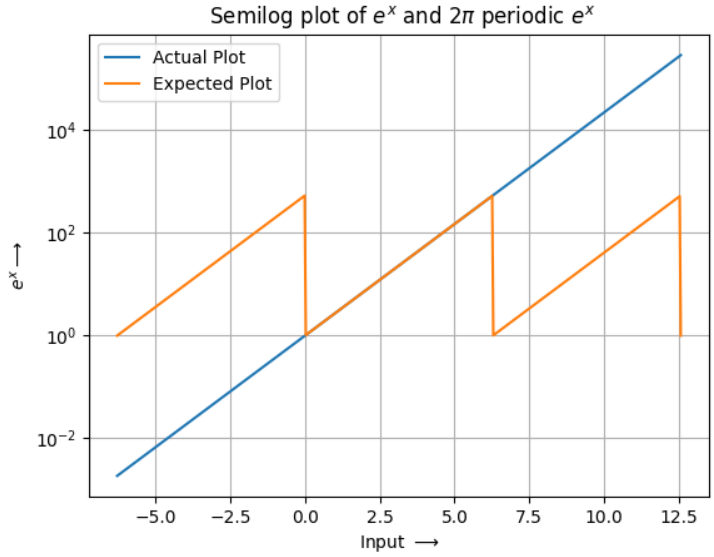

Figure 1: Actual and Expected Plots of e^x

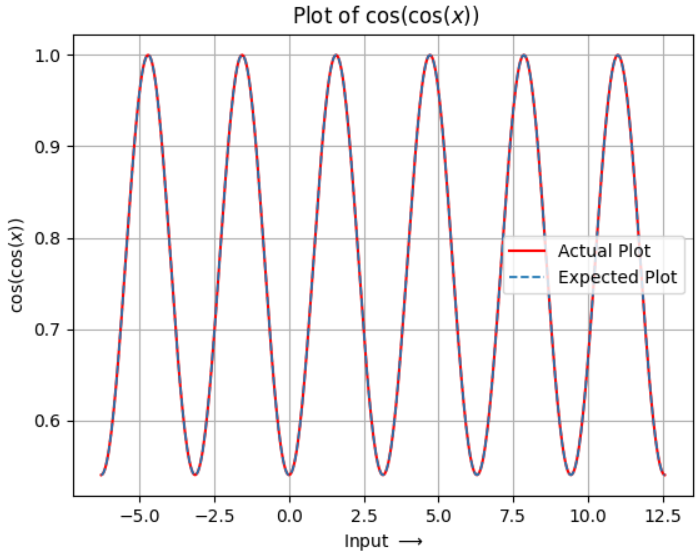

Figure 2: Actual and Expected Plots of $\cos(\cos(x))$

3.2 Generating coefficients

To generate the first 51 coefficients, we define a function *coeff()* which takes the function as input whose coefficients are to be found out. The 51 coefficients are:

$$\begin{pmatrix} a_0 \\ a_1 \\ b_1 \\ a_2 \\ b_2 \\ \dots \\ a_{25} \\ b_{25} \end{pmatrix}$$

The code for the function:

```
def coeff(func):
    def uv(func):
        u = lambda x,k: func(x)*np.cos(k*x)/np.pi
        v = lambda x,k: func(x)*np.sin(k*x)/np.pi
        return u, v
```

```
u, v = uv(func)
a = [integrate.quad(u,0,2*np.pi,args=(k))[0] for k in range(26)]
b = [integrate.quad(v,0,2*np.pi,args=(k))[0] for k in range(26)]
a[0] = a[0]/2
out = np.delete((np.array([[i,j] for i, j in zip(a,b)]).ravel()),1)
return out
```

The function returns the above vector of numpy array type.

3.3 Plotting Fourier Coefficients

Q3 (a):

The function $\cos(\cos(x))$ is an even function. Thus for the estimated function to be even the *sin()* terms should not be present in the equation. Since the *sin()* is an odd function, *sin()* terms in the estimated function would not make the function even. Thus *sin()* terms should vanish. Therefore, the terms b_n must be zero.

Q3 (b):

The function e^x has many frequency components whereas the function $\cos(\cos(x))$ has frequency $\frac{1}{\pi}$. Thus the function $\cos(\cos(x))$ doesnot have higher frequencies and thus the higher coefficients decay to zero more rapidly than that of the function e^x.

Q3 (c):

The Fourier Coefficients of e^x vary as:

$$a_n \propto \frac{1}{n^2+1} \tag{2}$$

$$b_n \propto \frac{n}{n^2+1} \tag{3}$$

Now as n increases, the terms a_n and b_n become:

$$a_n \propto \frac{1}{n^2+1} \simeq \frac{1}{n^2} \tag{4}$$

$$b_n \propto \frac{n}{n^2+1} \simeq \frac{1}{n} \tag{5}$$

Now for log-log plot, we should take logarithm of a_n and b_n

$$\log(a_n) \simeq \log(\frac{1}{n^2}) \simeq -2\log(n) \tag{6}$$

$$\log(b_n) \simeq \log(\frac{1}{n}) \simeq -\log(n) \tag{7}$$

Thus the coefficients of e^x in log-log graph are linear as we obtained in equations (6) and (7) and can also be seen in Figure 4. Whereas the coefficients of $\cos(\cos(x))$ vary exponentially with n. Therefore, this implies semilog plot of this will be linear as seen in Figure 5. The coefficients are plotted on semilog scale using the following code:

```
plt.figure(num=3)
n = np.array(list(range(51)))
plt.semilogy(n, np.abs(exp_vector))
plt.xlabel('n   $\longrightarrow$')
plt.ylabel('Coefficient  $\longrightarrow$')
plt.title(' Semilog plot of fourier coefficients of $e^x$')
plt.legend()
plt.show()

plt.figure(num=5)
plt.semilogy(n, np.abs(coscos_vector))
plt.xlabel('n   $\longrightarrow$')
plt.ylabel('Coefficient  $\longrightarrow$')
plt.title(' Semilog plot of fourier coefficients of $\cos(\cos(x))$')
plt.legend()
plt.show()
```

The coefficients are plotted on loglog scale using the following code:

```
plt.figure(num=4)
plt.loglog(n, np.abs(exp_vector))
plt.xlabel('n   $\longrightarrow$')
plt.ylabel('Coefficient  $\longrightarrow$')
plt.title(' log-log plot of fourier coefficients of $e^x$')
plt.legend()
plt.show()

plt.figure(num=5)
plt.loglog(n, np.abs(coscos_vector))
plt.xlabel('n   $\longrightarrow$')
```

```
plt.ylabel('Coefficient  $\longrightarrow$')
plt.title(' Log-Log plot of fourier coefficients of $\cos(\cos(x))$')
plt.legend()
plt.show()
```

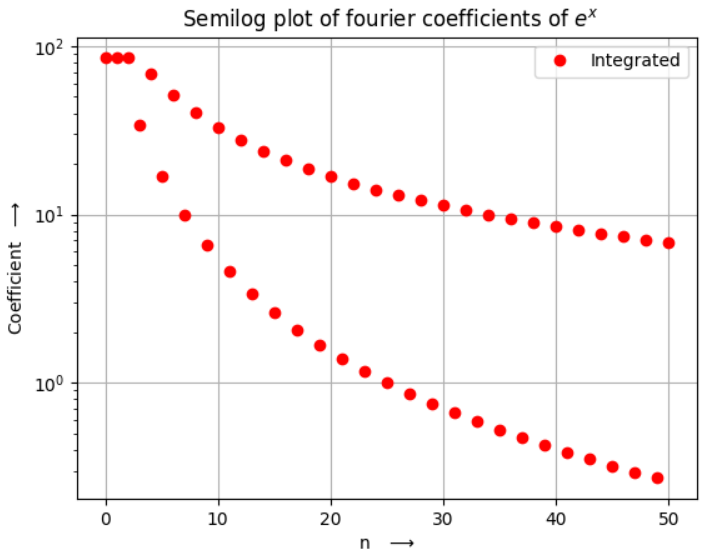

Figure 3: Semilog plot of Fourier Coefficients of e^x

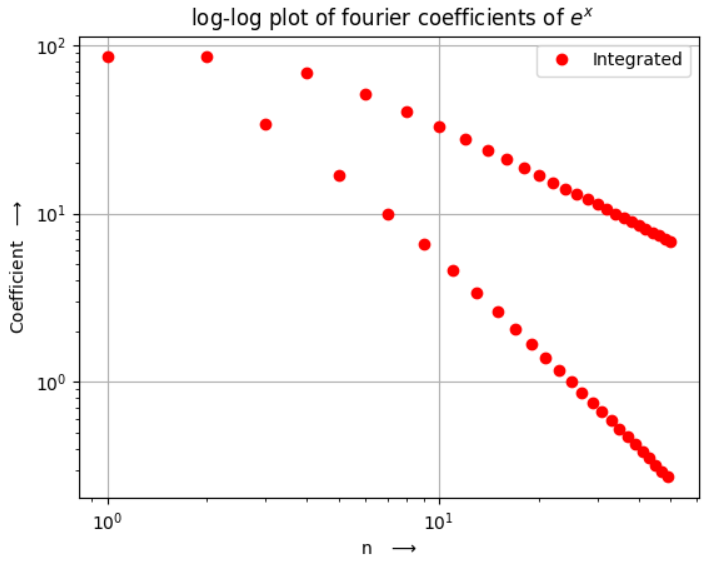

Figure 4: Log-log plot of Fourier Coefficients of e^x

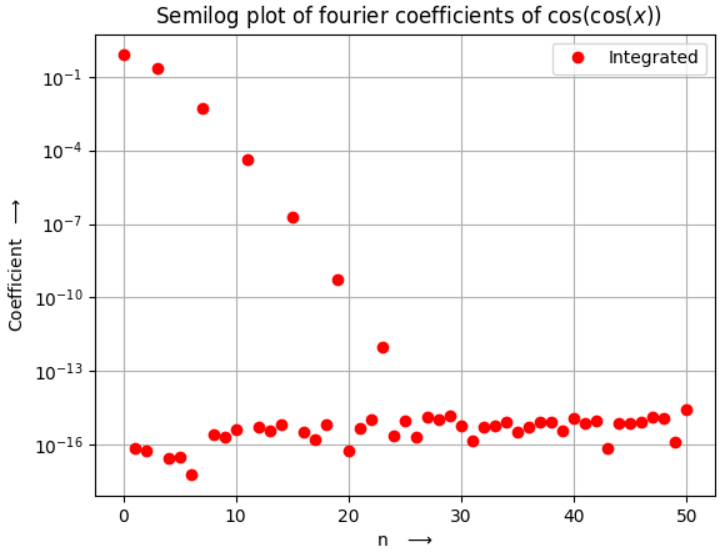

Figure 5: Semilog plot of Fourier Coefficients of $\cos(\cos(x))$

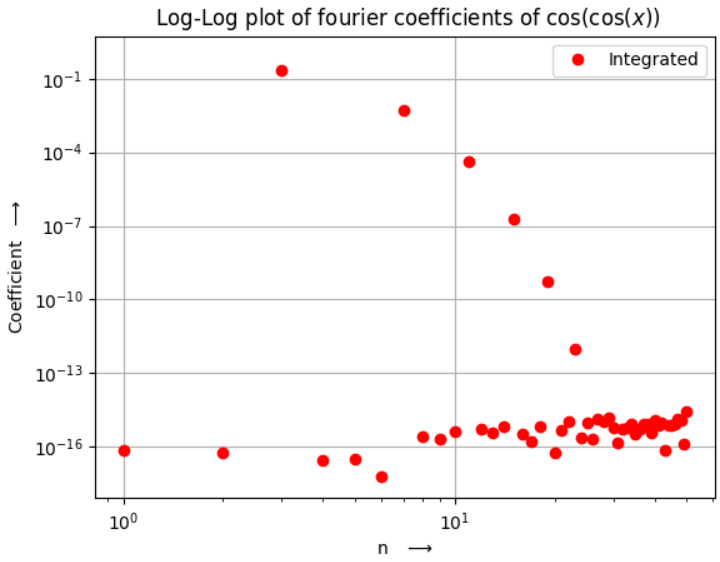

Figure 6: Log-log plot of Fourier Coefficients of $\cos(\cos(x))$

3.4 Coefficients using Least Squares Method

By using linear regression, the fourier coefficients can be found out using the following matrix problem $A.c = b$:

$$\begin{pmatrix} 1 & cosx_1 & sinx_1 & ... & cos25x_1 & sin25x_1 \\ 1 & cosx_2 & sinx_2 & ... & cos25x_2 & sin25x_2 \\ ... & ... & ... & ... & ... & ... \\ 1 & cosx_{400} & sinx_{400} & ... & cos25x_{400} & sin25x_{400} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ b_1 \\ a_2 \\ b_2 \\ ... \\ a_{25} \\ b_{25} \end{pmatrix} = \begin{pmatrix} f(x_1) \\ f(x_2) \\ ... \\ f(x_{400}) \end{pmatrix}$$

To generate the Matrix A and matrix b:

```
x = np.linspace(0, 2*np.pi,401)
x = x[:-1]
def f_exp(x):
    out = np.exp(x)
    return out
def f_coscos(x):
    out = np.cos(np.cos(x))
```

```
    return out
b_lstsq_exp = f_exp(x)
b_lstsq_coscos = f_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)
```

The vector *b_exp_lstsq* is the vector of values of e^x for the input values between $[0, 2\pi)$ divided into 400 points. Similarly, the vector *b_coscos_lstsq* is the vector of values of $\cos(\cos(x))$ for the same input values.

3.5 Plotting output of Least Squares Method

The Coefficient vector is found out using the the function lstsq from the scipy.linalg library. The coefficients found out using this method for function e^x is stored in c_exp_lstsq and for function $\cos(\cos(x))$ is stored in c_coscos_lstsq. The code to find these vectors is:

```
c_exp_lstsq = lstsq(A,b_lstsq_exp)[0]
c_coscos_lstsq = lstsq(A,b_lstsq_coscos)[0]
```

The coefficients obtained are plotted in the semilog plot and loglog plot for the two functions along with the coefficients obtained from integration method. The coefficients from integration are marked in red and those from least square method are marked in green in the figure.

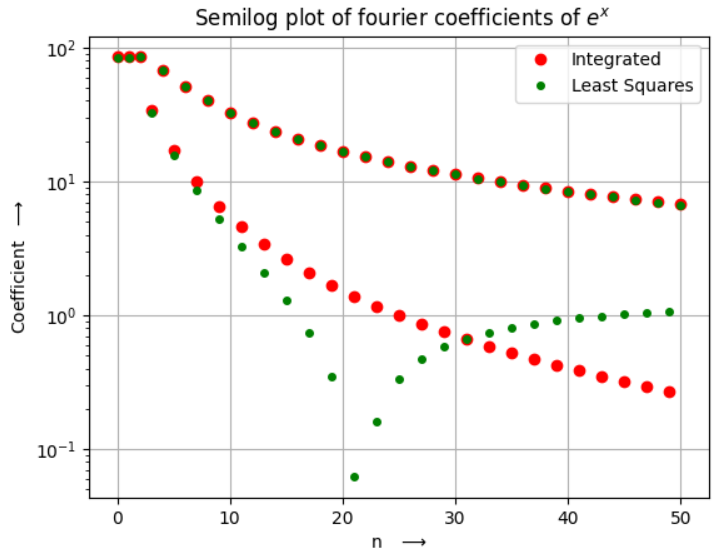

Figure 7: Semilog plot of Fourier Coefficients of e^x

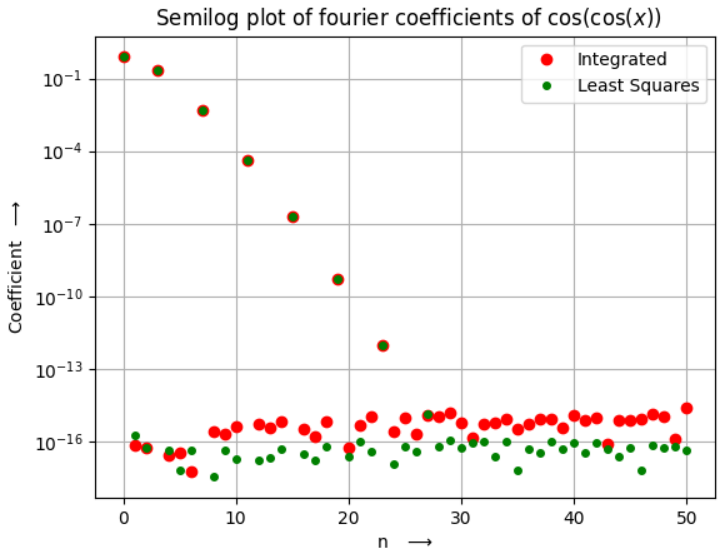

Figure 8: Semilog plot of Fourier Coefficients of $\cos(\cos(x))$

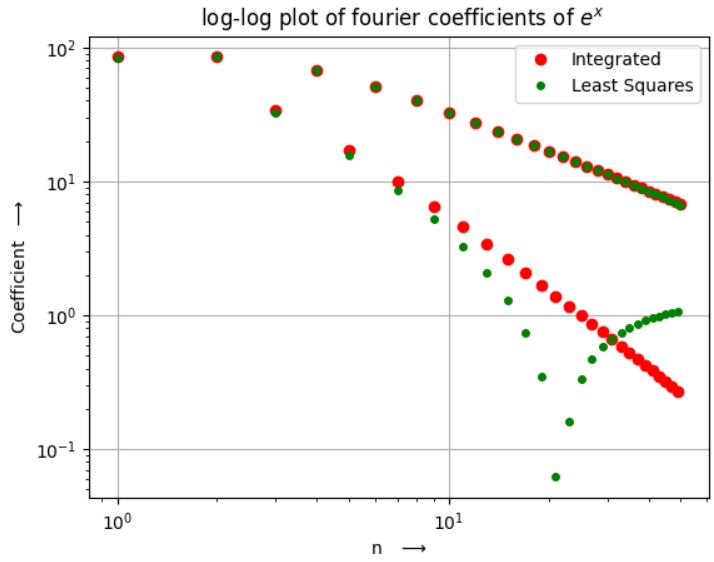

Figure 9: Log-log plot of Fourier Coefficients of e^x

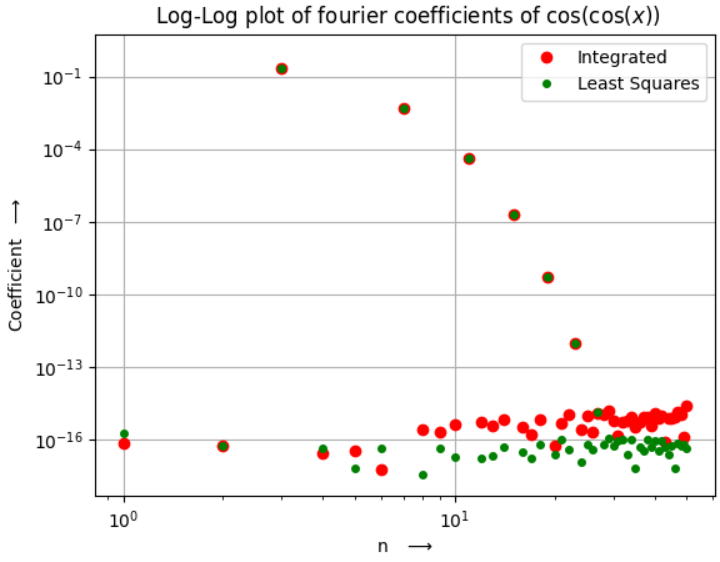

Figure 10: Log-log plot of Fourier Coefficients of $\cos(\cos(x))$

3.6 Comparing Predictions

The Fourier Coefficients obtained from Least Squares method and Integration method are plotted in Figure 7 and Figure 8 as shown below. As we can see the plots partly agree with each other. The largest deviation of Least Squares from the Coefficients from Integration method are:

```
Largest Error for the fucntion exp(x): 1.3327308703354106
Largest Error for the fucntion cos(cos(x)): 2.650068350102312e-15
```

The both coefficients should agree with each other. In this case they dont agree with each other as the input sample size is small. If we take a large input size, then these values will match.

3.7 Plotting Results

The function values are found out using the estimated coefficients and the matrix equation $A.c = b$. These are plotted in a graph along with the original function to see the deviation of the estimated function from the original function. The plots are shown in the pictures given. The red color graph indicates the original function and the green points indicate the estimated graph for e^x. The yellow graph indicates the original function of $\cos(\cos(x))$ and the green points are the estimated function data points.
Question: Why is there so much deviation in Figure 11 but nearly perfect agreement in Figure 12?
The graph of approximate calculated function $\cos(\cos(x))$ almost agrees with the actual function. This is because the frequency components are less and all those are satisfied in the first 51 coefficients. Whereas the graph of the approximate calculated function e^x doesnot agree with the actual function. This is because the function has many high frequency components and these components are missed as only 51 coefficients are taken. Since the high frequency terms are missing, the graphs do not agree with eachother,

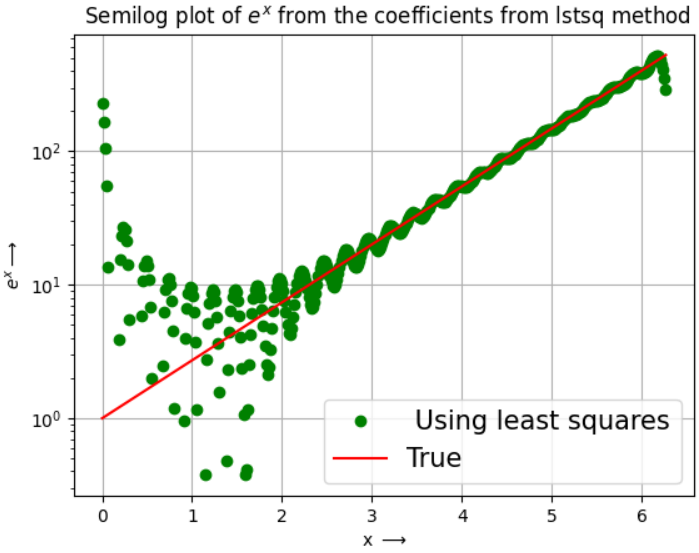

Figure 11: Plot of e^x

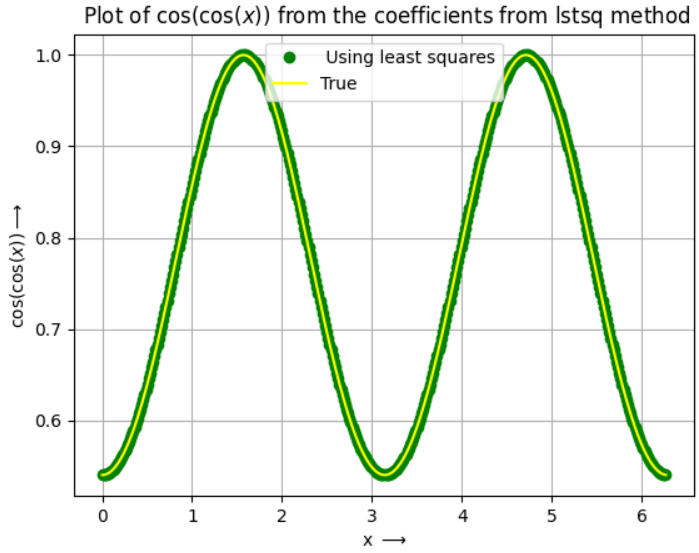

Figure 12: Plot of $\cos(\cos(x))$

4 Conclusions

In this assignment, we solved for fourier series coefficients using two methods. One using the integration method and other using the least squares method. We notice that the functions match in the case of $\cos(\cos(x))$. The error in the coefficients obtained in of the order $10^{-}15$. In the case of e^x the functions obtained from integration do not match with that of least squares. The error is also large with value around 1.33. Thus for calculation efficiency we could use Least Squares method for $\cos(\cos(x))$ while this method is not feasible for e^x.