Assignment 4

K.R.Srinivas - EE18B136

February 27, 2020

Abstract

This week's Python assignment focuses on the following topics.

- To plot functions $cos(cosx))$ and e^x
- To plot the same functions using Fourier Series expansion .
- Compute and compare the Fourier Series coefficient using Integration as well as LstSq method.
- Reconstruct the function using the coefficients computed .

1 Introduction

The goal of this assignment is to find Fourier Approximations of two function e^x and $\cos(\cos(x))$ from its integral definition and using Least Squares method. We will fit two functions, e^x and $\cos(\cos(x))$ over the interval $[0,2\pi)$ using the fourier series expansion.

$$a_0 + \sum_{n=1}^{\infty} a_n \cos(nx_i) + b_n \sin(nx_i) \approx f(x_i) \tag{1}$$

The equations used here to find the Fourier coefficients are as follows:

$$a_0 = \frac{1}{2\pi} \int_0^{2\pi} f(x)dx \tag{2}$$

$$a_n = \frac{1}{\pi} \int_0^{2\pi} f(x)\cos(nx)dx \tag{3}$$

$$b_n = \frac{1}{\pi} \int_0^{2\pi} f(x)\sin(nx)dx \tag{4}$$

2 Assignment Tasks

2.1 Plotting of Actual Functions

We define Python functions for the two functions e^x and $\cos(\cos(x))$ which return a vector (or scalar) value and plot it in the interval [2π,4π).

```
def func_0(x):
        return np.cos(np.cos(x))
def func_1(x):
        return exp(x)
```

We see that while $\cos(\cos(x))$ is 2π periodic, e^x is not. e^x is monotously increasing hence not periodic. Fourier series expansion are valid only for periodic functions.Below, we plot the periodic version of e^x which is also the expected output from fourier series approximation.

```
semilogy(x, func_0(x), 'k', label="Original Function")
semilogy(x, func_0(x % period), '--',label="Expected
        Function from fourier series")
semilogy(x, func_1(x), 'k', label="Original Function")
semilogy(x, func_1(x % period), '--',label="Expected
        Function from fourier series")
```

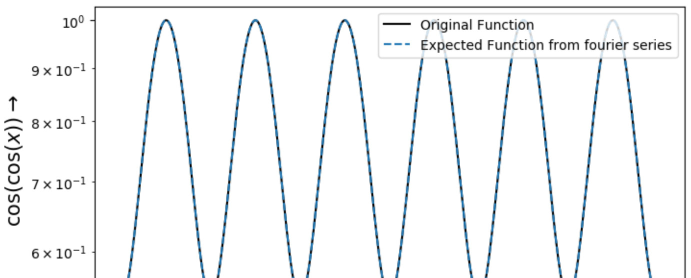

Figure 1: Plotting $\cos(\cos(x))$

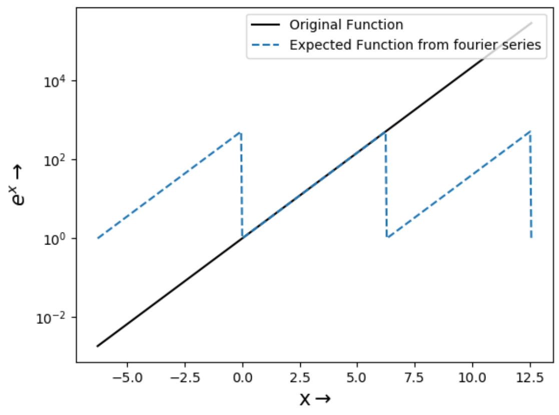

Figure 2: Plottinge^x

2.2 Computing Fourier Coefficients

The first 51 fourier coefficients of the functions are computed by the equations 2, 3, 4 in python using the `quad()` function available in `scipy.integrate` module.

```
 def integrand_a(x, k, f):
     return f(x)*cos(k*x)

def integrand_b(x, k, f):
     return f(x)*sin(k*x)

def coef_vec(f):

    coeff = []
    coeff.append((quad(f, 0, 2*pi)[0])/(2*pi))
    for i in range(1, 26):
        coeff.append((quad(integrand_a , 0, 2*pi ,
        args=(i , f))[0])/pi)
        coeff.append((quad(integrand_b , 0, 2*pi ,
        args=(i , f))[0])/pi)

return coeff
```

The vector `coeff[]` contains the coefficients a_n and b_n calculated as discussed before and "f" is the function of interest that is passed as argument.The coefficients here are computed by integration method.The corresponding plots in Semilog and Loglog scale for the respective functions are given below.

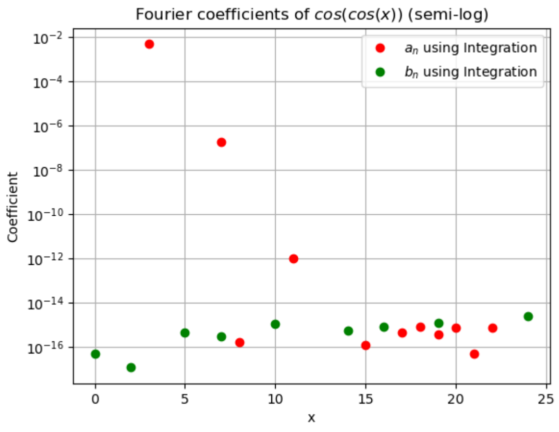

Figure 3:

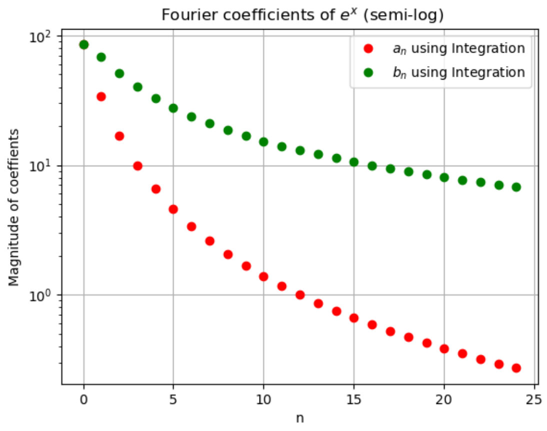

Figure 4:

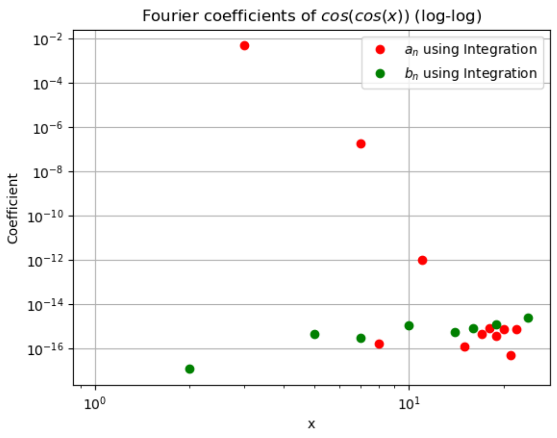

Figure 5:

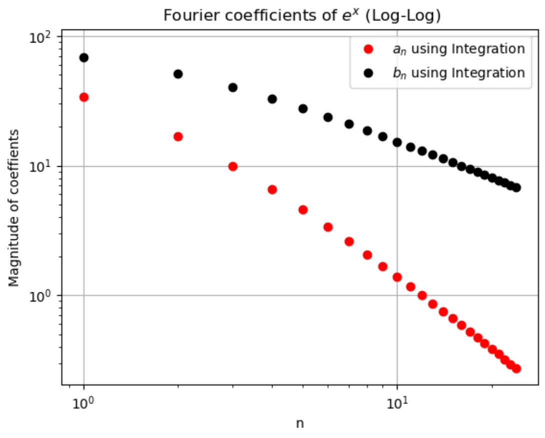

Figure 6:

The functions are reconstructed from the fourier coefficients computed by direct integration method are given below.

$$\begin{pmatrix} 1 & \cos(x_1) & \sin(x_1) & .... & \cos(25x_1) & \sin(25x_1) \\ 1 & \cos(x_2) & \sin(x_2) & .... & \cos(25x_2) & \sin(25x_2) \\ ... & ... & ... & .... & ... & ... \\ 1 & \cos(x_{400}) & \sin(x_{400}) & .... & \cos(25x_{400}) & \sin(25x_{400}) \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ b_1 \\ ... \\ a_{25} \\ b_{25} \end{pmatrix} = \begin{pmatrix} f(x_1) \\ f(x_2) \\ ... \\ f(x_{400}) \end{pmatrix}$$

We create the matrix on the left side and call it A . We want to solve $Ac = b$ where c are the fourier coefficients, following the rule given by equation 1.

```
def argument_matrix(nrow, ncol, x):
        A = zeros((nrow, ncol))
        A[:, 0] = 1
        for k in range(1, int((ncol+1)/2)):
            A[:, 2*k-1] = cos(k*x)
            A[:, 2*k] = sin(k*x)
        return A

x_ = linspace(-2*pi, 4*pi, 400)

def compute_fn(coeff):
        A = argument_matrix(400, 51, x_)
        f_fourier = np.dot(A, coeff)
        return f_fourier
```

where function `argument_matrix()` creates the A matrix and function `compute_fn()` computes Fourier Series approximation by taking the inner product of coefficent matrix and A. The plots of the computed functions are given below.

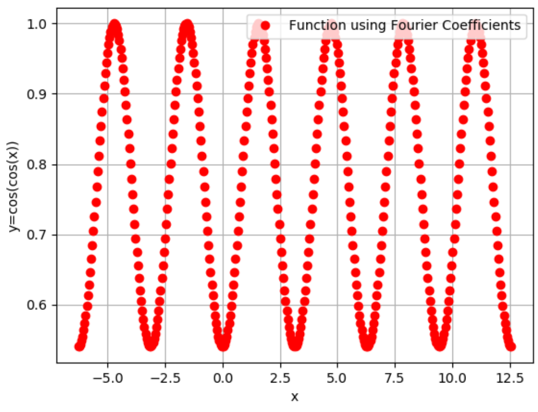

Figure 7: Fourier Series Approximation ofcos(cos(x))

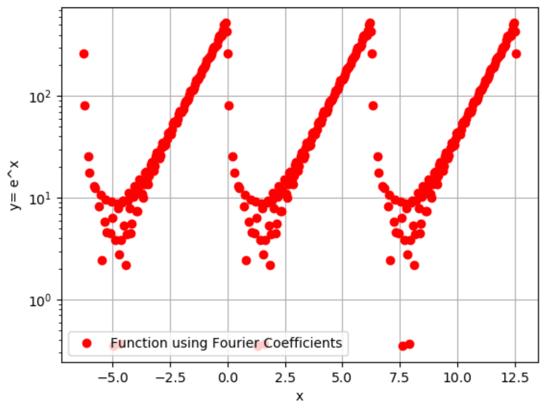

Figure 8: Fourier Series Approximation of e^x

2.3 Computing Fourier Coefficients by Lstsq Method

Having calculated the fourier coefficients by integration method, we now try computing them by ”LstSq” method.We define a vector x going from 0 to 2π in 400 steps (linspace). Evaluate the function f(x) at those x values and and solve $Ac = b$ like before, but using `lstsq()` function.

```
x_ = linspace(-2*pi, 4*pi, 400)
b = [func(i) for i in x_]
A_ = argument_matrix(400, 51, x_)
c = lstsq(A_, b, rcond=None)[0]
```

The coefficients computed are plotted below.

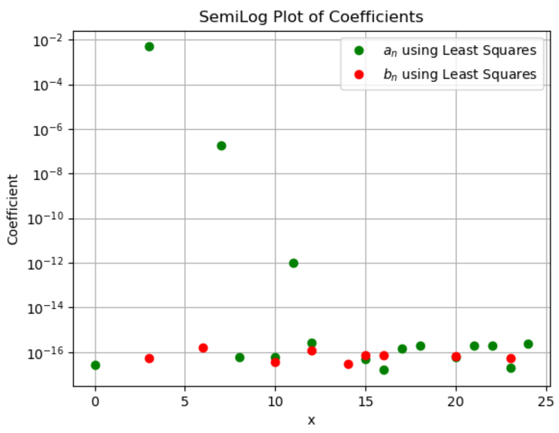

Figure 9:

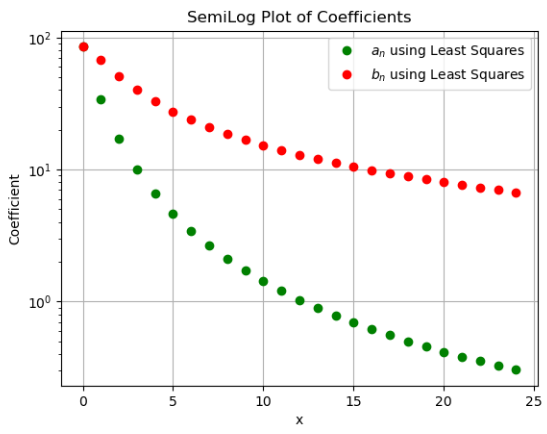

Figure 10:

We now, aim to reconstruct the function from the coeffecients evaluated from LstSq method as performed before.The corresponding plots are given below.

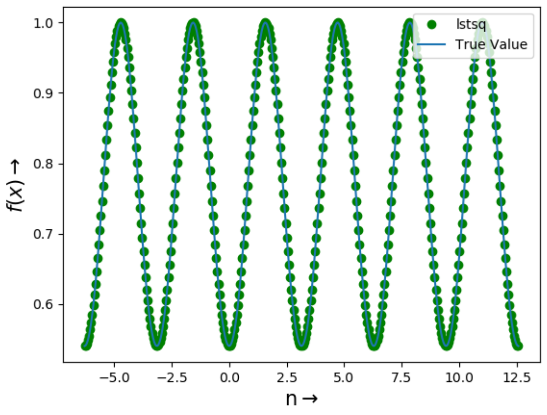

Figure 11: Function $\cos(\cos(x))$ constructed by LstSq Method

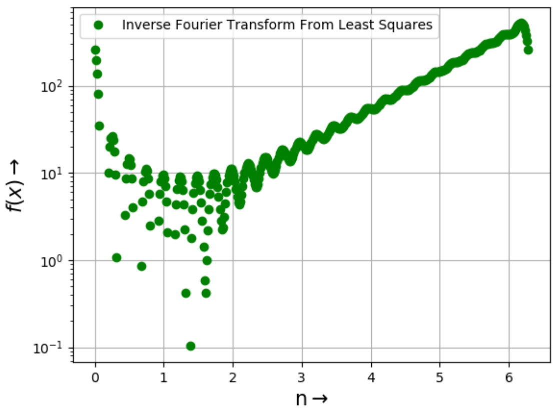

Figure 12: Function e^x constructed by LstSq Method

2.4 Deviation of LstSq from Integration Method.

Having calculated the coefficients by both Integration and LstSq methods, it is necessary to understand how much deviation is there.

- The maximum deviation in Fourier series coefficients for $(\cos(\cos(x)))$ computed by LstSq method from Integration method is $2.6211317962716135e-15$
- The maximum deviation in Fourier series coefficients for (e^x) computed by LstSq method from Integration method is 0.08812169778767576

Our Predictions for e^x are very poor compared to that of cos(cos(x)). This can be fixed by sampling at a larger number of points, however at the cost of higher computations.

3 Conclusion

Thus using the tools of Python we have approximated the functions $(\cos(\cos(x)))$ and (e^x) using their Fourier Series by computing the coefficients directly by integration as well as LstSq method.We notice close matching of the two methods in case of $\cos(\cos(x))$ while, there is a larger discrepancy in $\exp(x)$ owing to the fact that the latter is non-periodic and hence we have considered the variation of e^x with period 2π . It was also verified that the odd sinusoidal components of cos(cos(x)) are nearly zero.