EE2703 : Applied Programming Lab
Assignment-4

Nihal Gajjala
EE19B044

March 2021

Functions

Defining python functions for e^x and $\cos(\cos(x))$ and plotting these functions over the interval [-2π,4π). These functions can be defined using built-in functions in numpy.

```python
def exp(x):
    return np.exp(x)

def coscos(x):
    return np.cos(np.cos(x))
```

Figure 1: Functions

The above functions are plotted over the interval [-2π,4π) using 300 points sampled uniformly over this interval. The function e^x is plotted on a linear x axis and semilog y axis. The function $\cos(\cos(x))$ is plotted on a linear x and y axis.

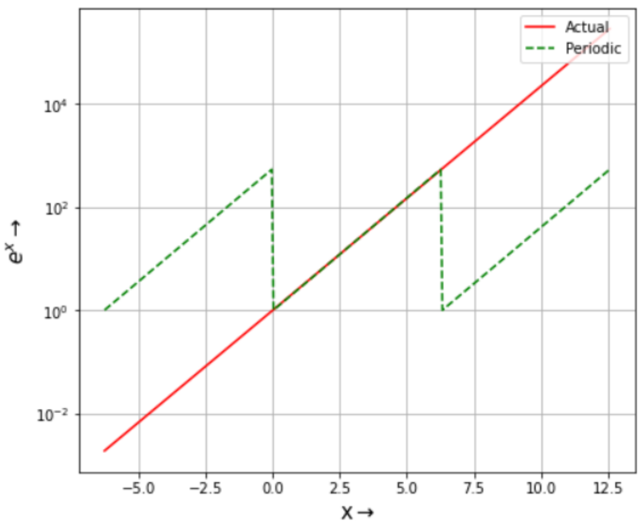

Figure 2: Plot Of e^x

The function e^x is a non-periodic increasing function. To calculate the Fourier series coefficients this function is made 2π periodic. The Fourier series coefficients of a non-periodic function is calculated by looking at the function pattern in the period [0,2π) and repeating it as shown in Figure 2.

The function $\cos(x)$ is a periodic function with fundamental period 2π. The function $\cos(\cos(x))$ is a periodic function with fundamental period π.

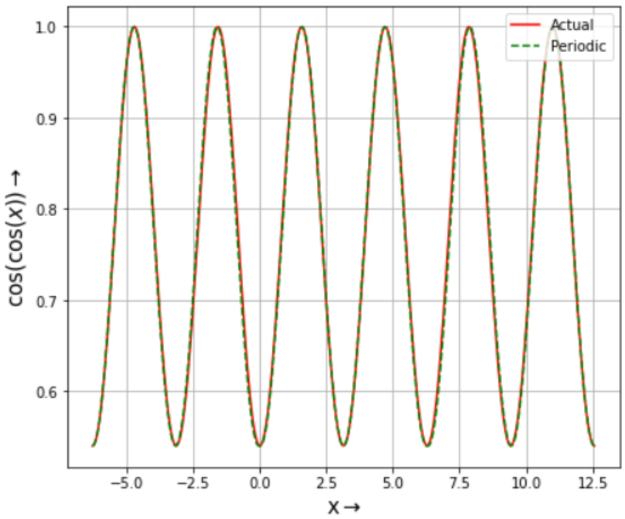

Figure 3: Plot Of $\cos(\cos(x))$

Fourier series coefficients

A Fourier series is an expansion of a periodic function f(x) in terms of an infinite sum of sines and cosines.

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

The coefficients a_n and b_n are given by

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

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

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

The below code is used to find the first 'n' Fourier coefficients.

```
functions={'exp(x)':exp,'cos(cos(x))':coscos}
# Function To Find Fourier Coefficient
def findCoefficient(n,function):
    coefficient=np.zeros(n)                                          # Constructing A Array Filled With Zeros
    fourier=functions[function]
    a=lambda x,k:fourier(x)*np.cos(k*x)                             # Cosine Function
    b=lambda x,k:fourier(x)*np.sin(k*x)                             # Sine Function
    coefficient[0]=quad(fourier,0,2*np.pi)[0]/(2*np.pi)              # DC Coefficient
    for i in range(1,n,2):
        coefficient[i]=quad(a,0,2*np.pi,args=((i+1)/2))[0]/np.pi    # Sine Coefficient
    for i in range(2,n,2):
        coefficient[i]=quad(b,0,2*np.pi,args=(i/2))[0]/np.pi        # Cosine Coefficient
    return coefficient
```

Figure 4: Code To Find Fourier Coefficients

Plotting first 51 Fourier coefficients of e^x on LogLog and SemiLog.

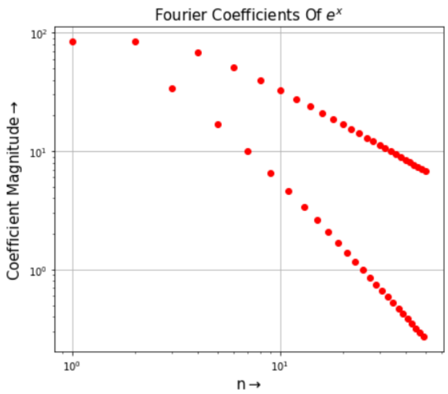

Figure 5: LogLog Plot Of e^x

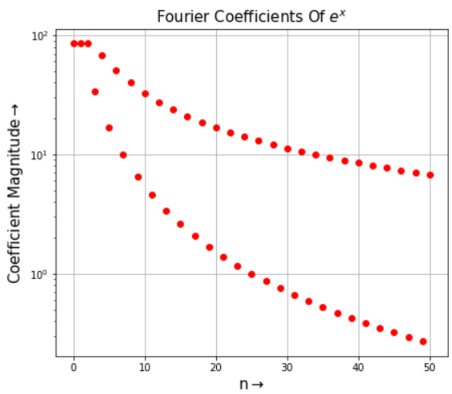

Figure 6: SemiLog Plot Of e^x

Plotting first 51 Fourier coefficients of $\cos(\cos(x))$ on LogLog and SemiLog.

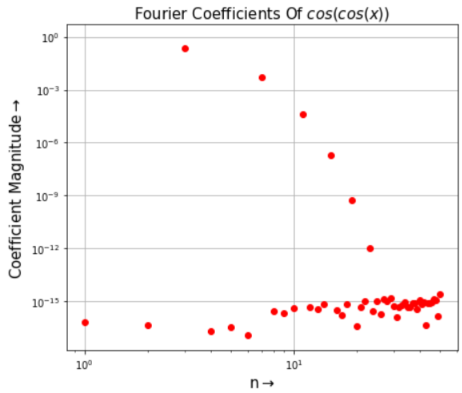

Figure 7: LogLog Plot Of $\cos(\cos(x))$

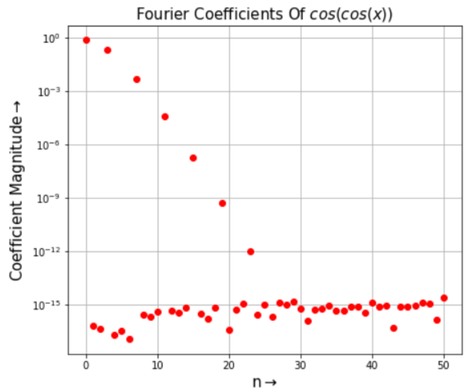

Figure 8: SemiLog Plot Of $\cos(\cos(x))$

We expect the b_n coefficients for the function $\cos(\cos(x))$ to be zero as $\cos(\cos(x))$ is an even function but we notice small non-zero values because of the limitation in the numerically accuracy upto which the value of π can be stored in the memory.

Function e^x has an exponentially increasing slope which results in a wide range of frequencies in its Fourier series. The magnitude of the Fourier series coefficients of e^x is inversely proportional to n^2. For large values of n, $\log(a_n)$ and $\log(b_n)$ are approximately equal to -2log(n) which results in a linear graph in the LogLog plot.

The function $\cos(\cos(x))$ has a relatively low frequency due to which contributions by higher sinusoids is less, leading to a quick decay of the magnitude of the Fourier series coefficients which results in a linear graph in the SemiLog plot.

Least Squares

The Fourier series coefficients can be approximated using Least Squares.

$$a_0 + \sum_{n=1}^{25} a_n \cos(nx_i) + b_n \sin(nx_i) = f(x_i) \tag{5}$$

$$A = \begin{pmatrix} 1 & \cos(x_1) & \sin(x_1) & \dots & \sin(25x_1) \\ 1 & \cos(x_2) & \sin(x_2) & \dots & \sin(25x_2) \\ \dots & \dots & \dots & \dots & \dots \\ 1 & \cos(x_{400}) & \sin(x_{400}) & \dots & \sin(25x_{400}) \end{pmatrix} B = \begin{pmatrix} f(x_1) \\ f(x_2) \\ \dots \\ f(x_{400}) \end{pmatrix} \tag{6}$$

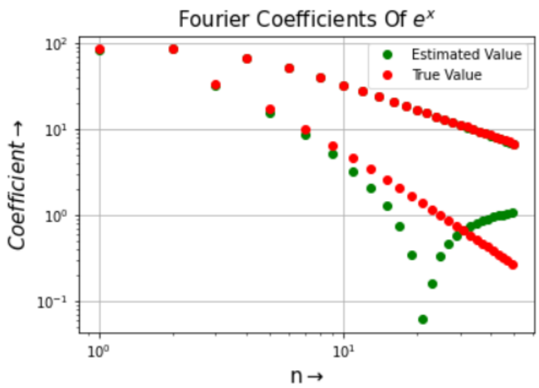

Figure 9: LogLog LS Plot Of e^x

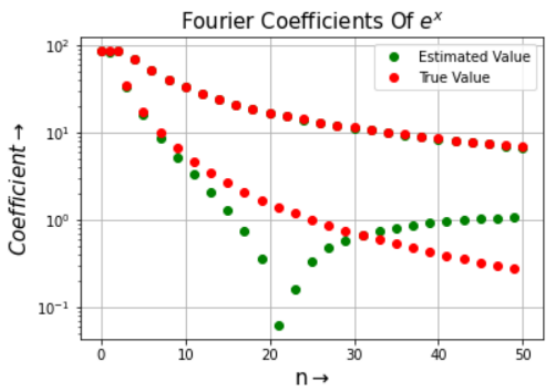

Figure 10: SemiLog LS Plot Of e^x

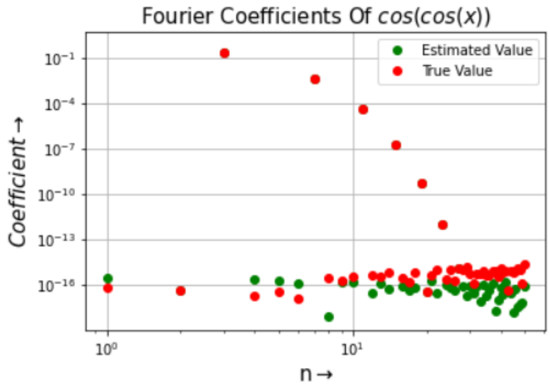

Figure 11: LogLog LS Plot Of $\cos(\cos(x))$

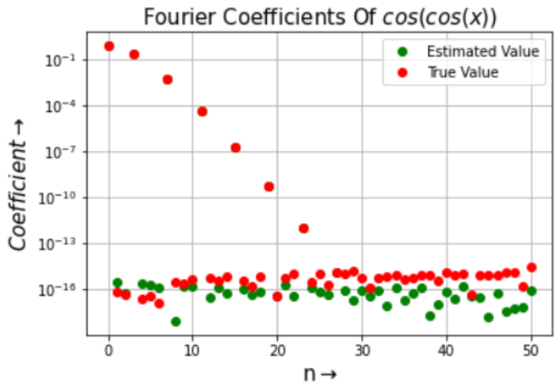

Figure 12: SemiLog LS Plot Of $\cos(\cos(x))$

In case of $\cos(\cos(x))$ the Least Square coefficients are close to the coefficients obtained from integration whereas in case of e^x the Least Square coefficient differ drastically from those obtained from integration. This is because $\cos(\cos(x))$ is a periodic signal whereas e^x is a non-periodic signal with an exponentially growing slope. The deviation of coefficients found using Least Squares from the true value (Fourier coefficient found using integration) can be found using the below code.

```
deviationExp=abs(coefficientExp-cExp)
deviationCosCos=abs(coefficientCosCos-cCosCos)
maxDeviationExp=np.max(deviationExp)
maxDeviationCosCos=np.max(deviationCosCos)
print(maxDeviationExp)
print(maxDeviationCosCos)
```

```
1.3327308703353395
2.691038055079395e-15
```

Figure 13: Deviation

The deviation is less in case of $\cos(\cos(x))$ when compared to the deviation in case of e^x.

Approximation

Plot of e^x and its Fourier approximation

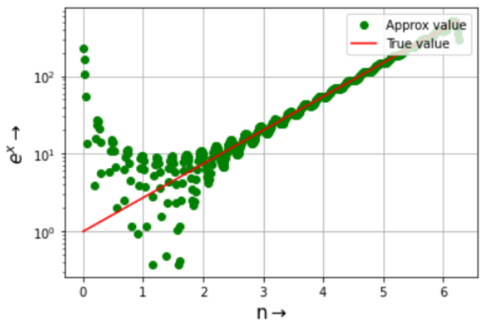

Figure 14: Fourier Approximation Of e^x

Plot of $\cos(\cos(x))$ and its Fourier approximation

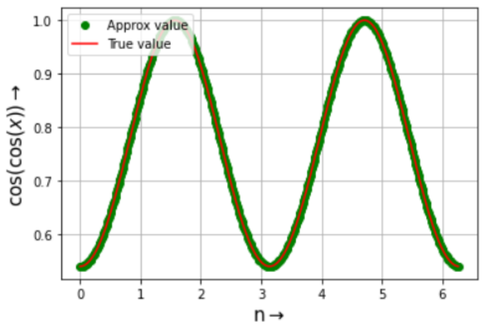

Figure 15: Fourier Approximation Of $\cos(\cos(x))$

The Fourier Approximation of $\cos(\cos(x))$ is very similar to the actual function. This is due to the fact that the function $\cos(\cos(x))$ is periodic resulting in lesser deviation from its true value. In case of e^x, the function has a exponentially increasing slope and the higher frequencies of the Fourier series contribute significantly to the true value because of which the Fourier Approximation of e^x deviates from its true value. For a better estimate of e^x we would need to consider the higher sinusoidal frequencies.

Conclusion

From the above report it can be observed that the Fourier series coefficients for any functions can be found out in two ways i.e direct integration and least squares. The least square approach is easier execute as compared to direct integration. Least square approach takes lesser time and is computationally inexpensive.

We can even observe that the least square approach deviates more for non-periodic functions with exponentially increasing slope. For periodic function the least square approach gives similar results to that obtained from direct integration.

Least Squares approach can be used to calculate the Fourier series coefficients as it takes lesser time and the values are very close to the ones obtained from direct integration. For exact values of Fourier coefficients direct integration is preferred.