Assignment 4: Fourier Approximations

Chollangi Dheeraj Sai EE20B029

Febraury 26, 2022

Overview

This assignment mainly focuses upon

- To find the fourier coefficients of e^x and $cos(cos(x))$ using python
- To get the best estimate of the fourier coefficients using least squares method
- Using python modules to plot and study the the variation of these coefficients.

1 The functions e^x and $cos(cos(x))$

We are going to plot these functions with x ranging from -2π to 4π. Since $cos(cos(x))$ is periodic with period 2π it will have the same pattern over the entire x-axis, but e^x will have a periodic repetition when plotted for fourier coefficients. It is more convenient to plot e^x in semilogy.

```
def e(x):
    return exp(x)


def coscos(x):
    return cos(cos(x))

figure()
semilogy(x, e(x), "r", label="True Value")
semilogy(x, e(period), "b", label="Periodic Extension")
xlabel("x \u2192", fontsize=12)
ylabel("e^(x) \u2192", fontsize=12)
legend(loc="upper right")
grid()
```

```
figure()
plot(x, coscos(x), "b")
xlabel("x \u2192", fontsize=12)
ylabel("cos(cos(x)) \u2192", fontsize=12)
grid()
```

The plots corresponding to above code:

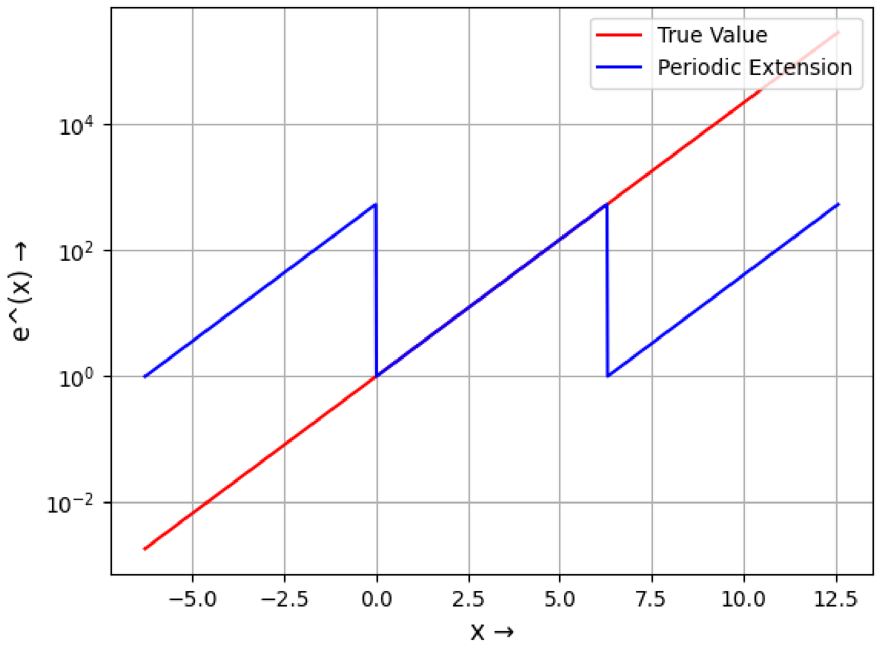

Figure 1: Plot of e^x

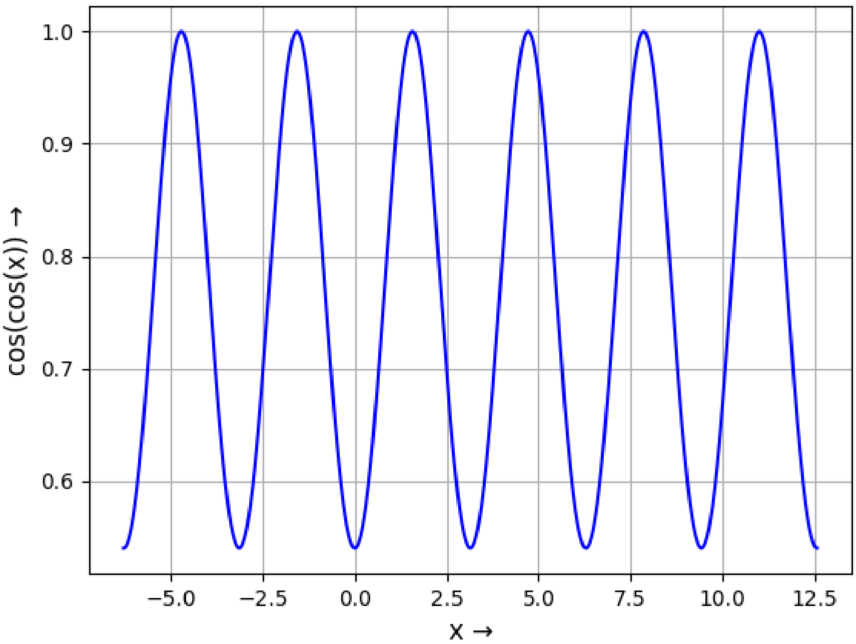

Figure 2: Plot of $cos(cos(x))$

2 The fourier series and it's coefficients

The fourier series of a function with period 2π is

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

It's coefficients can be calculated using

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

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

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

To find the fourier coefficients using python, first we will define the functions to integrate and use the *quad()* method in *scipy.integrate* module.

```
def u1(x, k):
    return exp(x) * cos(k * x)


def u2(x, k):
    return cos(cos(x)) * cos(k * x)


def v1(x, k):
    return exp(x) * sin(k * x)


def v2(x, k):
    return cos(cos(x)) * sin(k * x)

coeff_exp = np.zeros(51)
coeff_coscos = np.zeros(51)
coeff_exp[0] = quad(e, 0, 2 * pi)[0] / (2 * pi)
coeff_coscos[0] = quad(coscos, 0, 2 * pi)[0] / (2 * pi)
for k in range(1, 51, 2):
    coeff_exp[k] = quad(u1, 0, 2 * pi, args=((k + 1) / 2))[0] / pi
    coeff_coscos[k] = quad(u2, 0, 2 * pi, args=((k + 1) / 2))[0] / pi
for k in range(2, 51, 2):
    coeff_exp[k] = quad(v1, 0, 2 * pi, args=(k / 2))[0] / pi
    coeff_coscos[k] = quad(v2, 0, 2 * pi, args=(k / 2))[0] / pi
```

We can now construct the plots of these coefficients in semilog scale and logarithmic scale using the below code.

```
figure()
title("Semilog plot of fourier coefficients of e^(x)", fontsize=16)
semilogy(range(51), abs(coeff_exp), "ro")
xlabel("n \u2192", fontsize=12)
ylabel("Magnitude of coefficients \u2192", fontsize=12)
grid()

figure()
title("Loglog plot of fourier coefficients of e^(x)", fontsize=16)
loglog(range(51), abs(coeff_exp), "ro")
xlabel("n \u2192", fontsize=12)
ylabel("Magnitude of coefficients \u2192", fontsize=12)
grid()

figure()
title("Semilog plot of fourier coefficients of cos(cos(x))", fontsize=16)
```

```
semilogy(range(51), abs(coeff_coscos), "ro")
xlabel("n \u2192", fontsize=12)
ylabel("Magnitude of coefficients \u2192", fontsize=12)
grid()

figure()
title("Loglog plot of fourier coefficients of e^(x)", fontsize=16)
loglog(range(51), abs(coeff_coscos), "ro")
xlabel("n \u2192", fontsize=12)
ylabel("Magnitude of coefficients \u2192", fontsize=12)
grid()
```

The corresponding plots in logarithmic scale and semilog scale of the coefficients are as follows

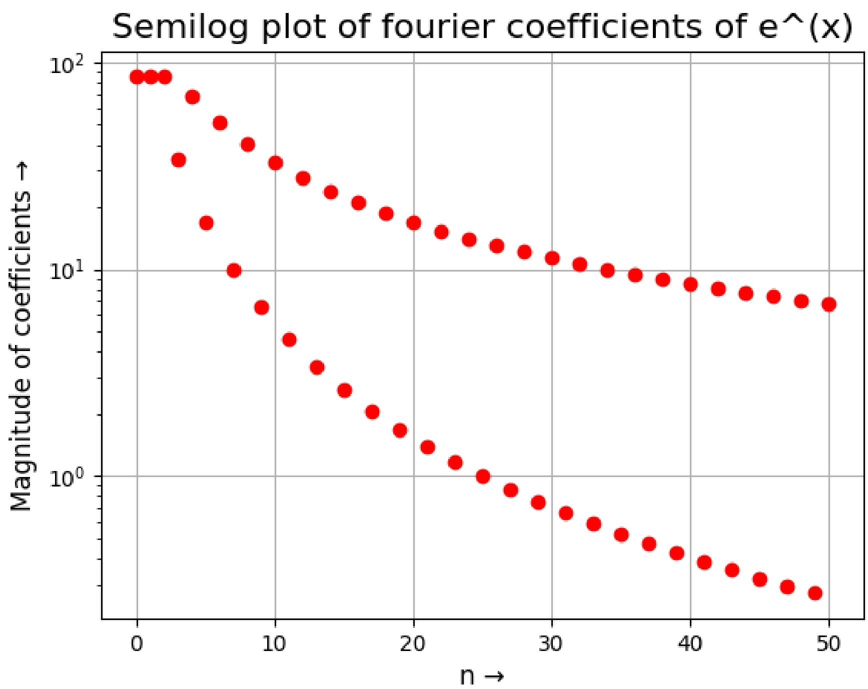

Figure 3: Semilog plot of fourier coefficients of e^x

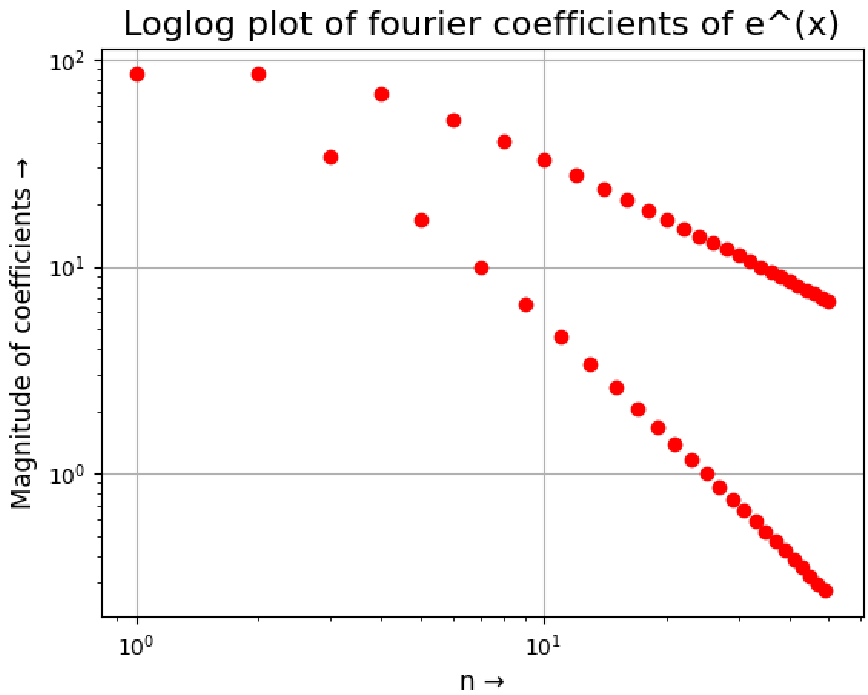

Figure 4: Loglog plot of fourier coefficients of e^x

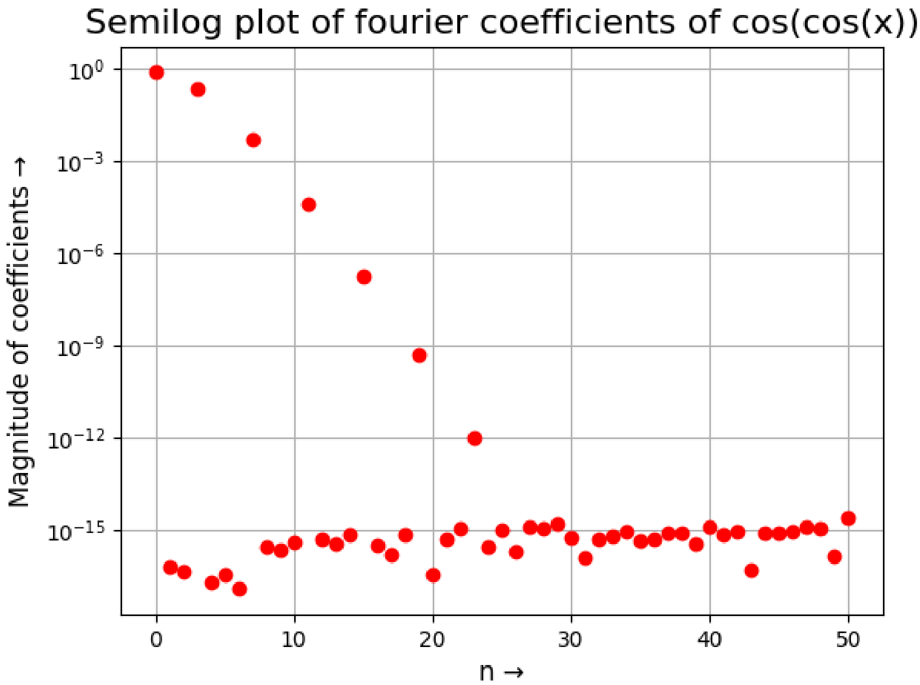

Figure 5: Semilog plot of fourier coefficients of $cos(cos(x))$

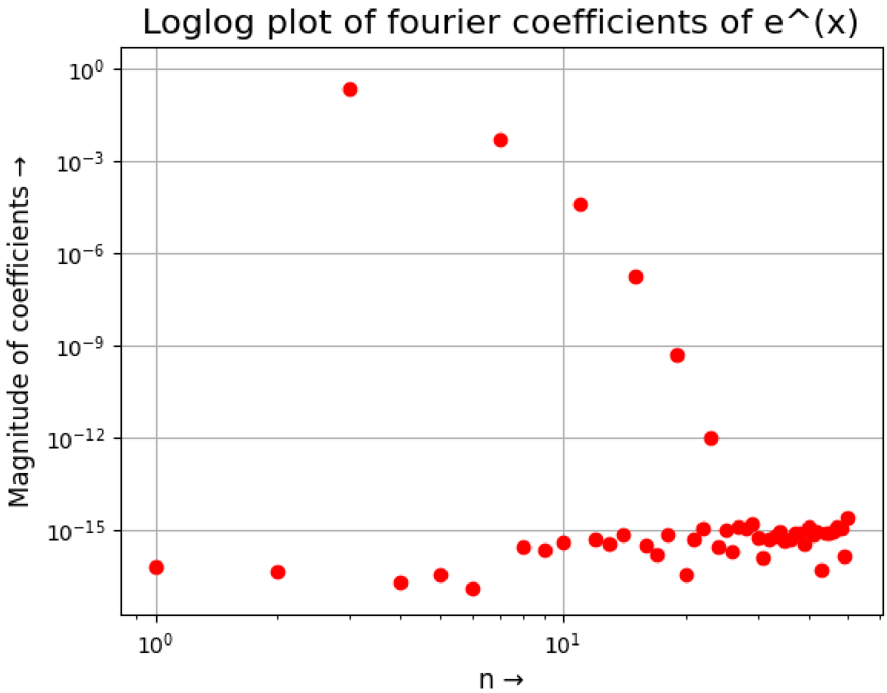

Figure 6: Loglog plot of fourier coefficients of $cos(cos(x))$

3 Using the least Squares aproach

We can get the best estimate of the fourier coefficients by studying the function value over a range of x. This can be done using *lstsq()* function to get the solution matrix of the below matrix equation.

$$\begin{pmatrix} 1 & \cos(\text{x1}) & \sin(\text{x1}) & .... & \cos(25\text{x1}) & \sin(25\text{x1}) \\ 1 & \cos(\text{x2}) & \sin(\text{x2}) & .... & \cos(25\text{x2}) & \sin(25\text{x2}) \\ .... & .... & .... & .... & .... & .... \\ 1 & \cos(\text{x400}) & \sin(\text{x400}) & .... & \cos(25\text{x400}) & \sin(25\text{x400}) \end{pmatrix} \begin{pmatrix} a0 \\ a1 \\ b1 \\ .... \\ a25 \\ b25 \end{pmatrix} = \begin{pmatrix} \text{f(x1)} \\ \text{f(x2)} \\ .... \\ \text{f(x400)} \end{pmatrix}$$

It can be done using the following python code

```
x1 = linspace(0, 2 * pi, 401)
x1 = x1[:-1]
b1 = e(x1)
b2 = coscos(x1)
A = zeros((400, 51))
A[:, 0] = 1
```

```
for k in range(1, 26):
    A[:, 2 * k - 1] = cos(k * x1)
    A[:, 2 * k] = sin(k * x1)


c_exp = lstsq(A, b1)[0]
c_coscos = lstsq(A, b2)[0]
```

The below plots show the variation of these actual fourier coefficients with the best estimate obtained in least Squares method.

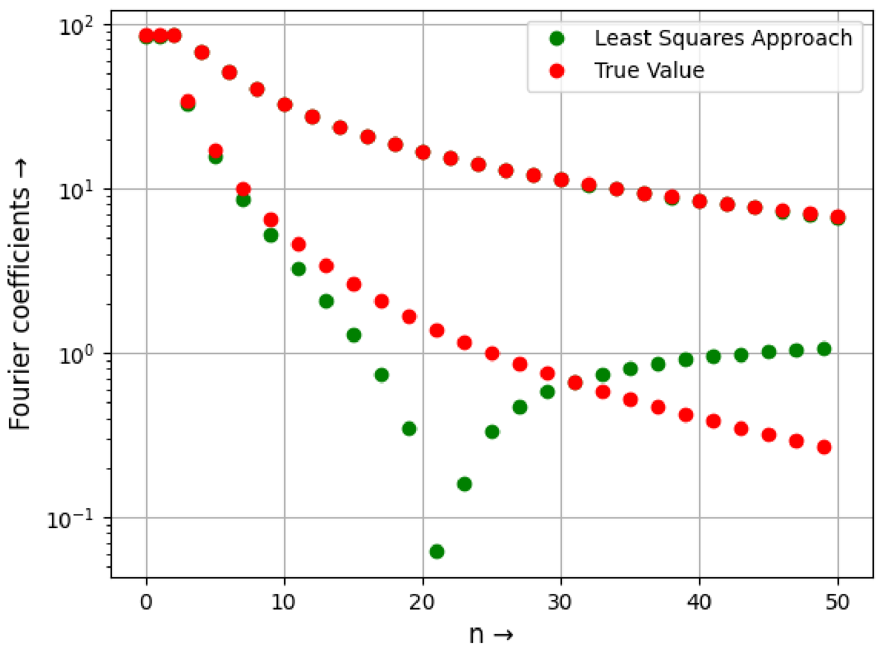

Figure 7: Semilog plot of fourier coefficients of e^x

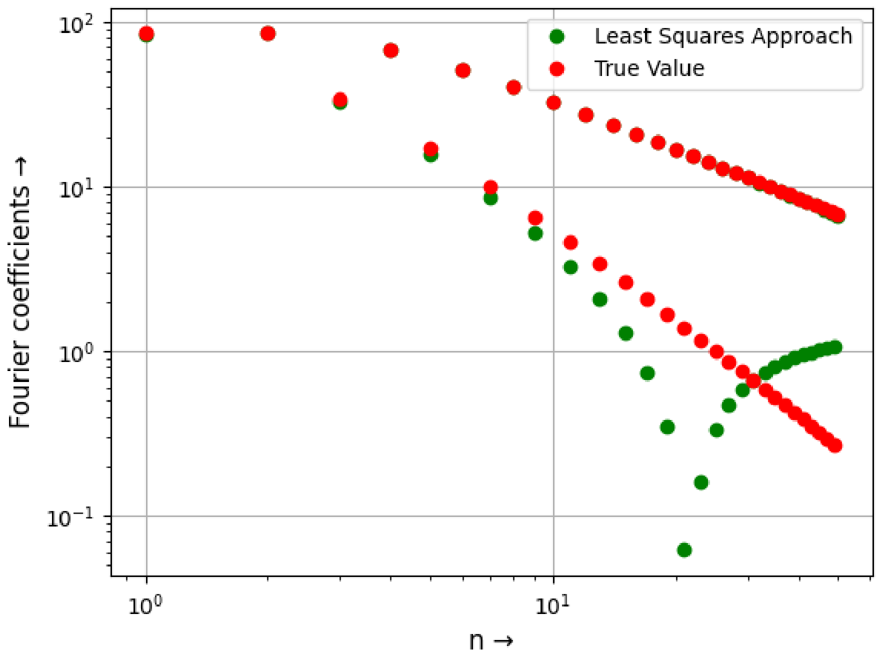

Figure 8: Loglog plot of fourier coefficients of e^x

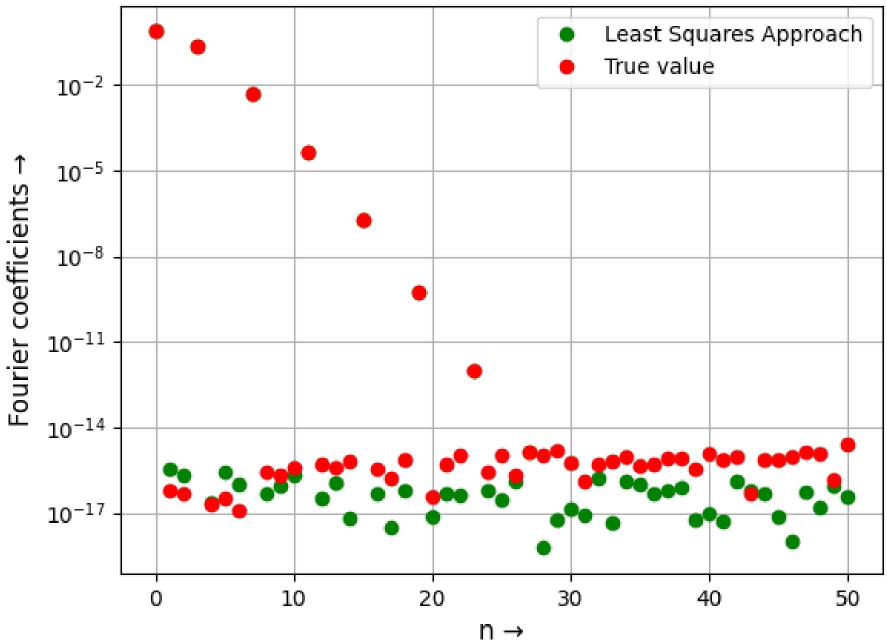

Figure 9: Semilog plot of fourier coefficients of $cos(cos(x))$

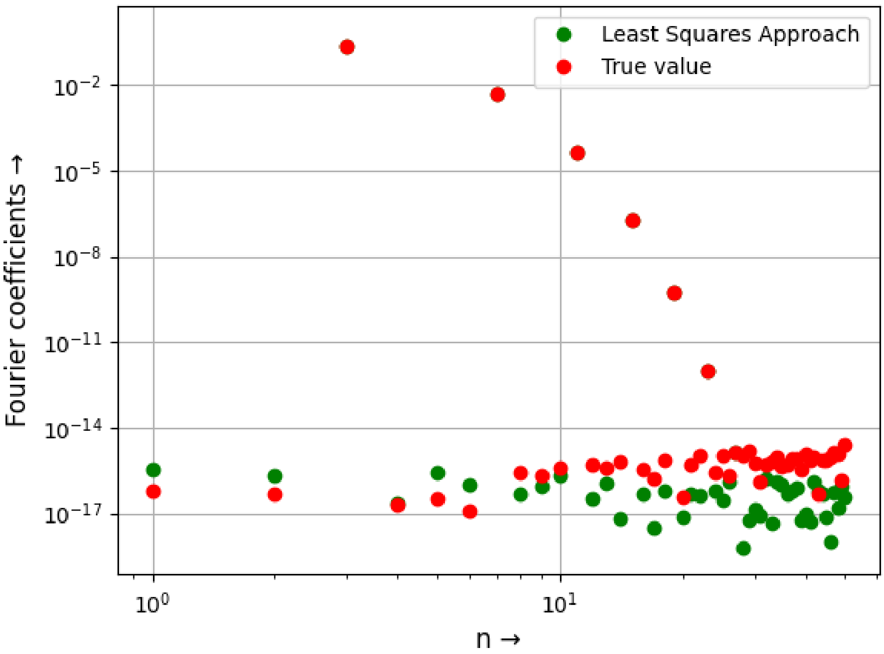

Figure 10: Loglog plot of fourier coefficients of $cos(cos(x))$

It can be seen that fourier coefficients for $cos(cos(x))$ are much closer in agreement when compared to the fourier coefficients of e^x. This is because $cos(cos(x))$ is periodic while e^x is not periodic. Since the least Squares method gives only the best estimate of these coefficients, the deviation is evident. The maximum deviation corresponding to these functions can be found using the below code.

```
coeffdiff_exp = abs(coeff_exp - c_exp)
coeffdiff_coscos = abs(coeff_coscos - c_coscos)

maxdev_exp = max(coeffdiff_exp)
maxdev_coscos = max(coeffdiff_coscos)
```

The maximum deviation for e^x is 1.3327308703354106.
The maximum deviation for $cos(cos(x))$ is 2.6442516119476574e-15

4 Function approximation

Using the best estimates of the fourier coefficients, we can calculate the functional values for both e^x and $cos(cos(x))$.

It can be done using the below matrix multiplication

```
val_exp = A @ c_exp
val_coscos = A @ c_coscos
```

The plots showing both the actual and predicted function values are as shown below:

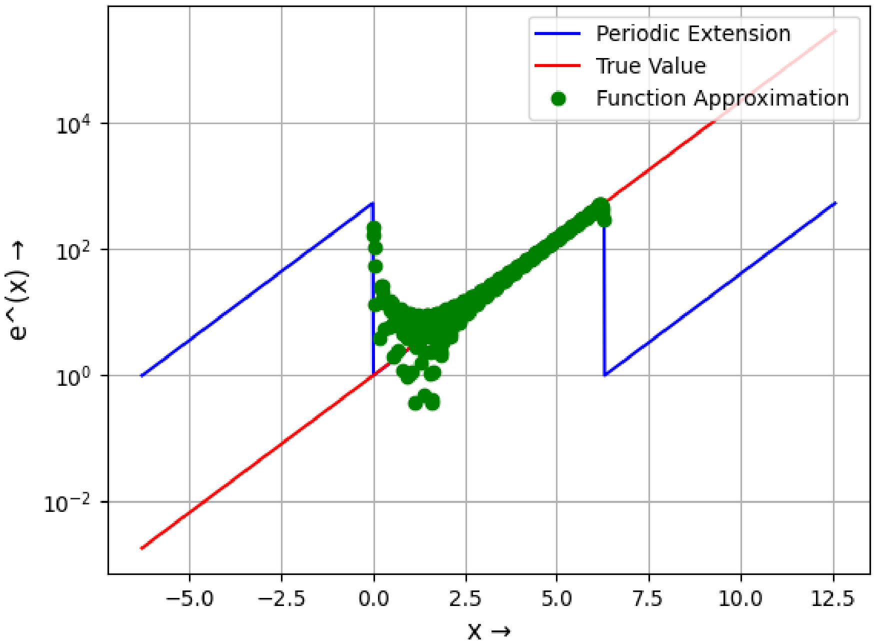

Figure 11: Function approximation for e^x

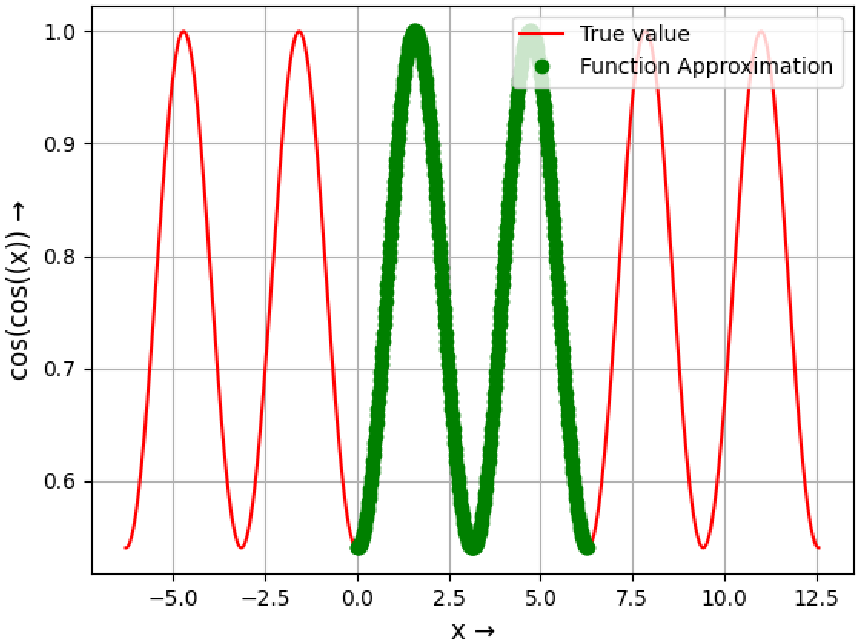

Figure 12: Function approximation for $cos(cos(x))$

5 Conclusions

- The least Squares approach deviated more for e^x than $cos(cos(x))$.
- We were able to find the fourier coefficients using least Squares approach which gave us the best estimate of the coefficients.
- We got the approximate function values using the least Squares approach.