Assignment 4

Sagnik Ghosh, EE19B132

10^{th} March 2021

1 Abstract

The objective of this assignment is to fit two functions, $exp(x)$ and $cos(cos(x))$ using their **Fourier Series Coefficients**, using two different methods, using the exact formula and using Least Squares.

2 Introduction

The Fourier Series representation of a function $f(x)$ is as follows:

$$f(x) = a_0 + \sum_{n=1}^{\infty}[a_n cos(nx) + b_n sin(nx)]$$

where,

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

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

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

3 Assignment

3.1 Creating and Plotting the Functions

We create functions for $exp(x)$ and $cos(cos(x))$ using built in Numpy functions which can be used on matrices and vectors. They are created as follows:

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

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

Both the functions were plotted over the interval $[-2\pi, 4\pi)$. Note that $exp(x)$ is plotted on a semilog scale for convenience.

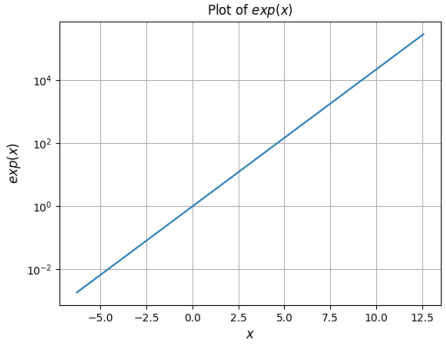

Figure 1: Plot of $exp(x)$

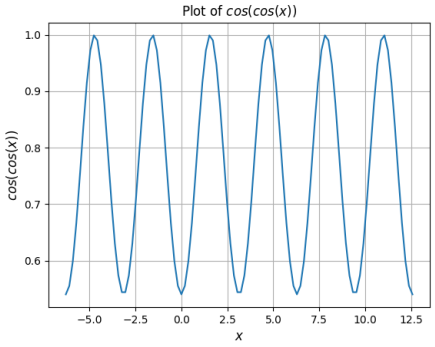

Figure 2: Plot of $cos(cos(x))$

3.2 Obtaining Fourier Series Coefficients

The Fourier Series Coefficients were obtained by calculating the definite integral defined in the Introduction. Then they were stored in the following manner:

$$\begin{Bmatrix} a_0 \\ a_1 \\ b_1 \\ ... \\ a_{25} \\ b_{25} \end{Bmatrix}$$

```
exp_coeff = []
exp_coeff.append((1/(2*np.pi))*quad(lambda x: np.exp(x), 0, 2*np.pi)[0])
for i in range(1,26):
    exp_coeff.append((1/np.pi)*quad(lambda x: np.exp(x)*np.cos(i*x), 0, 2*np.pi)[0])
    exp_coeff.append((1/np.pi)*quad(lambda x: np.exp(x)*np.sin(i*x), 0, 2*np.pi)[0])
```

We notice that the values of b_n for $cos(cos(x))$ are very close to zero. This happens because $cos(cos(x))$ is an even function and the coefficients of the sin terms should be zero.

For the case of $exp(x)$, as it increases exponentially, it has many frequency components and hence the Fourier Coefficients do not die out easily for higher frequencies. Whereas, for the case of $cos(cos(x))$, it has a low frequency of $\frac{1}{\pi}$ and does not have higher frequency components. Hence, the coefficients decay quickly for the second case.

The above coefficients are then plotted as shown.

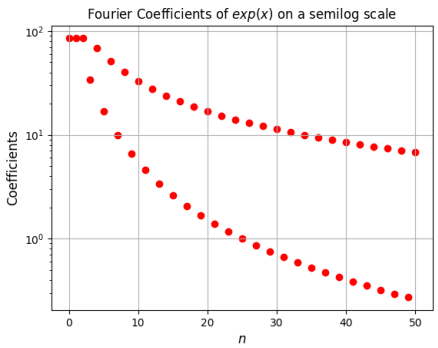

Figure 3: Plot of Fourier Coefficients of exp(x) on a semilog scale

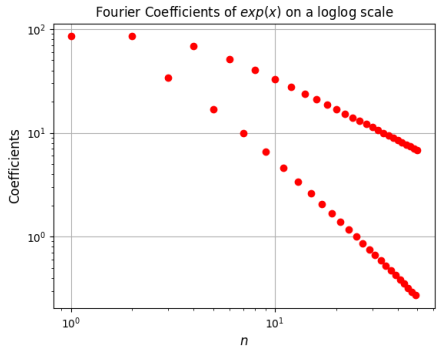

Figure 4: Plot of Fourier Coefficients of exp(x) on a loglog scale

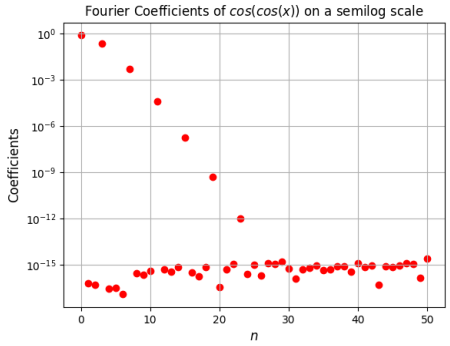

Figure 5: Plot of Fourier Coefficients of cos(cos(x)) on a semilog scale

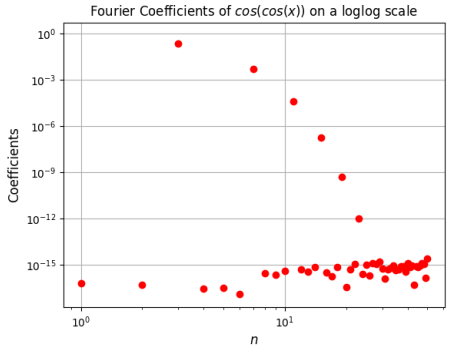

Figure 6: Plot of Fourier Coefficients of cos(cos(x)) on a loglog scale

3.3 Least Squares Approach

In this section, we try to estimate the Fourier Coefficients using the **Least Squares Approach**. We solve the following matrix equation:

$$Ac = b$$

where,

$$A = \begin{pmatrix} 1 & \cos(x_1) & \sin(x_1) & ... & \cos(25x_1) & \sin(25x_1) \\ 1 & \cos(x_2) & \sin(x_2) & ... & \cos(25x_2) & \sin(25x_2) \\ ... & ... & ... & ... & ... & ... \\ 1 & \cos(x_{400}) & \sin(x_{400}) & ... & \cos(25x_{400}) & \sin(25x_{400}) \end{pmatrix}$$

$$b = \begin{pmatrix} f(x_1) \\ f(x_2) \\ ... \\ f(x_{400}) \end{pmatrix}$$

The above matrices was generated as follows:

```
x = np.linspace(0, 2*np.pi, 401)
x = x[:-1]

def generate_matrix():
    A = np.zeros((400, 51))
    A[:,0] = 1

    for i in range(1, 51):
        if i%2 == 0:
            A[:,i] = np.sin(x*((i+1)//2))
        else:
            A[:,i] = np.cos(x*((i+1)//2))

    return A

A = generate_matrix()
```

3.4 Plotting and Comparing both Approaches

The first plot was plotted as follows:

```
b_exp = exp_func(x)

c_exp = lstsq(A, b_exp)[0]

plt.scatter(np.arange(0, len(exp_coeff)), np.abs(exp_coeff), color = 'red')
plt.scatter(np.arange(0, len(c_exp)), np.abs(c_exp), color = 'green')
plt.legend(['True Coefficients', 'Predicted Coefficients'])
plt.yscale('log')
plt.xlabel(r'$n$', size = 12)
plt.ylabel(r'Coefficients', size = 12)
plt.title(r'Fourier Coefficients of $exp(x)$ on a semilog scale using Least Squares')
plt.grid(True)
plt.show()
```

We can notice that there is some deviation of the Fourier Coefficients as estimated using Least Squares and by direct integration. We can treat the coefficients estimated using direct integration as the true values because we use the exact formulae whereas the Least Squares coefficients are just estimates.

There is more deviation in the case of the exponential function as compared to $cos(cos(x))$ because the latter is a periodic function and we only take a periodic extension of the former.

The magnitude of coefficients of $exp(x)$ varies as $\frac{1}{n^2+1}$ and hence the logarithm of coefficients vary as $log(\frac{1}{n^2+1})$ which can be approximated as $-2log(n)$ for large n. This explains why the loglog plot of $exp(x)$ is linear for large n.

The magnitude of coefficients of $cos(cos(x))$ vary exponentially with n. Hence the semilog plot for this function is linear.

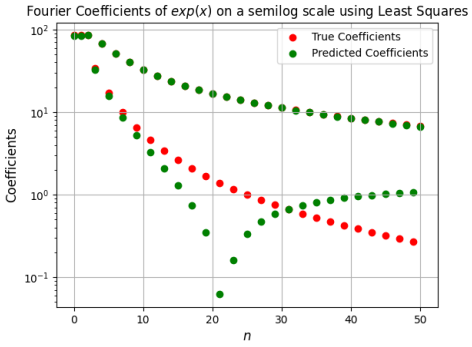

Figure 7: Fourier Coefficients of $exp(x)$ on a semilog scale using Least Squares

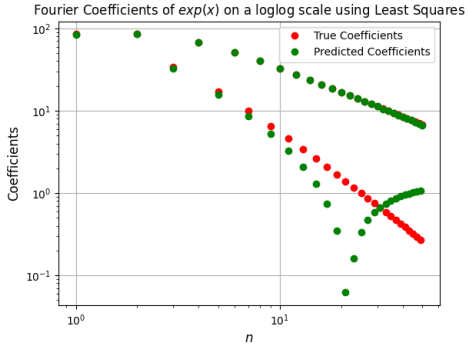

Figure 8: Fourier Coefficients of $exp(x)$ on a loglog scale using Least Squares

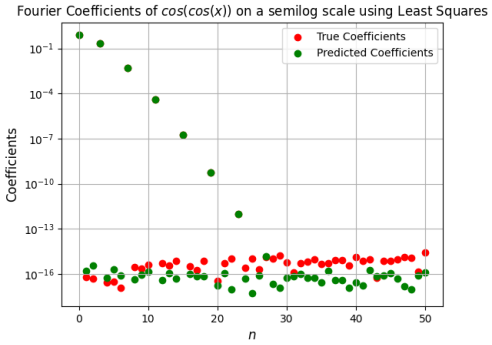

Figure 9: Fourier Coefficients of $cos(cos(x))$ on a semilog scale using Least Squares

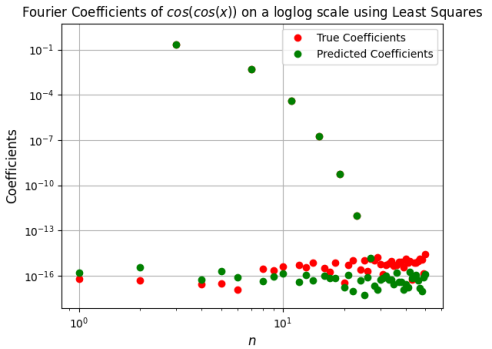

Figure 10: Fourier Coefficients of $cos(cos(x))$ on a loglog scale using Least Squares

3.5 Finding Maximum Deviation between the Coefficients

The maximum deviation between the coefficients obtained using the two approaches can be calculated as follows:

```
exp_dev = np.abs(exp_coeff - c_exp)
coscos_dev = np.abs(coscos_coeff - c_coscos)
max_exp_dev = np.max(exp_dev)
max_coscos_dev = np.max(coscos_dev)
```

Using the above code to calculate the maximum deviations, we get the following values:

$$\text{Maximum Deviation for } exp(x) = 1.3327308703353395$$
$$\text{Maximum Deviation for } cos(cos(x)) = 2.7194358321802792e-15$$

As we can notice, the maximum deviation of $exp(x)$ is much larger than $cos(cos(x))$. The reason for this was provided in the previous section.

3.6 Reconstructing the Functions

The matrix product Ac gives an estimate for the original functions using the coefficients estimated using Least Squares.

It can be noticed that there is considerable deviation for the case of $exp(x)$ whereas there is negligible deviation for the case of $cos(cos(x))$.

This happens because we approximate $exp(x)$ using low frequency components. To approximate it with greater accuracy, we need to consider components with higher frequencies.

This is not an issue for $cos(cos(x))$ because it is a periodic function and it can be accurately represented using low frequency components.

Evaluating this for both the functions and plotting them gives the following plots:

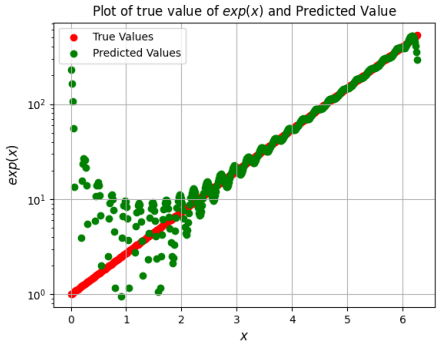

Figure 11: Reconstructed Function $exp(x)$ and the Original Function

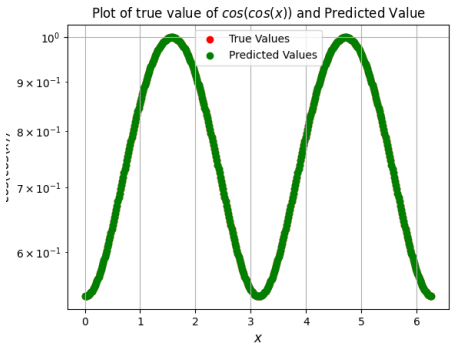

Figure 12: Reconstructed Function $cos(cos(x))$ and the Original Function

3.7 Conclusion

In this assignment we approximated the functions $exp(x)$ and $cos(cos(x))$ using their respective Fourier Coefficients. We estimated the Fourier Coefficients using two different approaches, exact integration using the formulae, and using Least Squares.

We can conclude that the method of Least Squares gives us a faster and computationally efficient way of approximating functions even though it is less accurate than direct integration.

We also noticed that there is a larger deviation of the coefficients for the two approaches for the case of the $exp(x)$ than $cos(cos(x))$.