Assignemnt 4:Fourier Approximations

Gudivada Harshad Kumar
EE20B038

February 26, 2022

1 Abstract

The main aim of the assignment Fourier approximations is

- Study the fourier series expansion of function
- How coefficients are effected due to discontinuous function
- Using least squares method to find best fit
- Compare both the results and observe deviation

2 Defining and plotting e^x and $cos(cos(x))$

We define the functions e^x and $cos(cos(x))$ using the following python code snippet and make the function e^x periodic

```
def e(x):
    return exp(x)
def cos_cos(x):
    return cos(cos(x))
def periodic(fun):
    def newfun(x):
        return fun(np.remainder(x,2*np.pi))
    return newfun
```

Using plot function we plot the graphs of the functions in interval [-2π,4π)
The function e^x is plotted on semilog plot due to it's large magnitudes
The function $cos(cos(x))$ is plotted normally

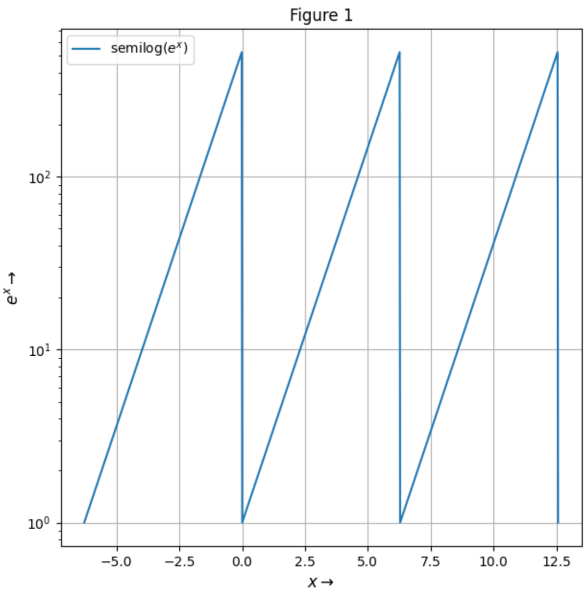

Figure 1: Plot of e^x

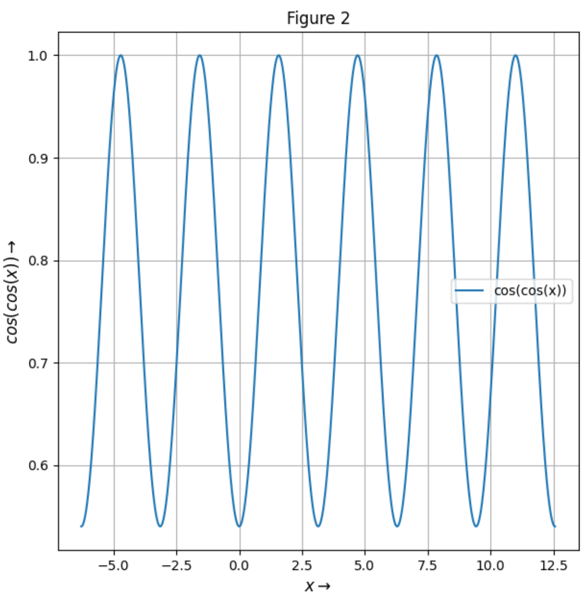

Figure 2: Plot of $cos(cos(x))$

3 Fourier approximation

The function is approximated by Fourier series as given below

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

Fourier coefficients are found using:

$$a_0 = \frac{1}{2\pi} \int\limits_0^{2\pi} f(x)dx \tag{2}$$

$$a_n = \frac{1}{\pi} \int\limits_0^{2\pi} f(x)\cos(nx)dx \tag{3}$$

$$b_n = \frac{1}{\pi} \int\limits_0^{2\pi} f(x)\sin(nx)dx \tag{4}$$

We now define the equations that will be used in computation of fourier coefficients

```
def a_f(x,k,f):
    return f(x)*np.cos(k*x)/np.pi
def b_f(x,k,f):
    return f(x)*np.sin(k*x)/np.pi
```

We now calculate the fourier coeffcients for $cos(cos(x))$ using the following python code snippet

```
a_cos_cos = np.zeros(26)
b_cos_cos = np.zeros(25)
a_cos_cos[0] = integrate.quad(cos_cos,0,2*np.pi)[0]/(2*np.pi)
for i in range(1,26):
    a_cos_cos[i] = integrate.quad(a_f,0,2*np.pi,args=(i,cos_cos))[0]
for j in range(25):
    b_cos_cos[j] = integrate.quad(b_f,0,2*np.pi,args=(j+1,cos_cos))[0]
```

Similarly for the function e^x

```
a_exp = np.zeros(26)
b_exp = np.zeros(25)
a_exp[0] = integrate.quad(e,0,2*np.pi)[0]/(2*np.pi)
for i in range(1,26):
    a_exp[i] = integrate.quad(a_f,0,2*np.pi,args=(i,e))[0]
for j in range(25):
    b_exp[j] = integrate.quad(b_f,0,2*np.pi,args=(j+1,e))[0]
```

Now the coefficients are grouped together to plot using the following code:

```
Coeff_cos_cos = np.zeros(51)
Coeff_cos_cos[0] = a_cos_cos[0]
for  i in range(1,26):
    Coeff_cos_cos[2*i-1] += a_cos_cos[i]
    Coeff_cos_cos[2*i] += b_cos_cos[i-1]
Coeff_exp = np.zeros(51)
Coeff_exp[0] = a_exp[0]
for  i in range(1,26):
    Coeff_exp[2*i-1] += a_exp[i]
    Coeff_exp[2*i] += b_exp[i-1]
```

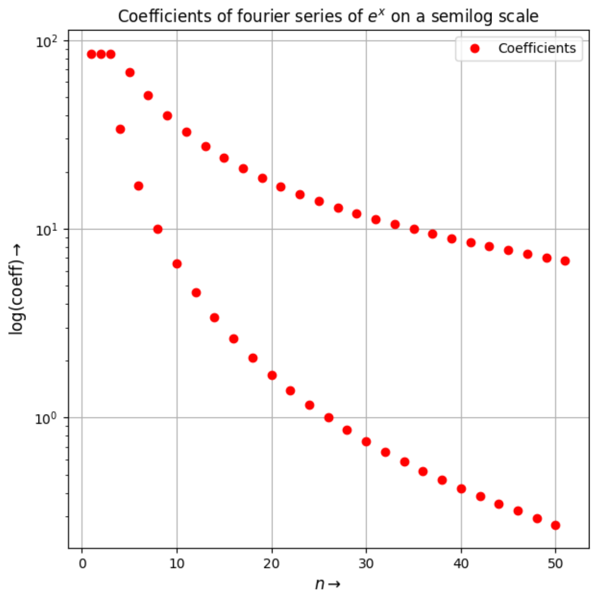

Figure 3: Semilog plot of Fourier coefficients of e^x

4 Plots of Fourier coefficients

Now we plot the coefficients of both the functions e^x and $cos(cos(x))$ one both semilog and loglog scale

b_n coefficients of $cos(cos(x))$ are nearly zero because it is an even function

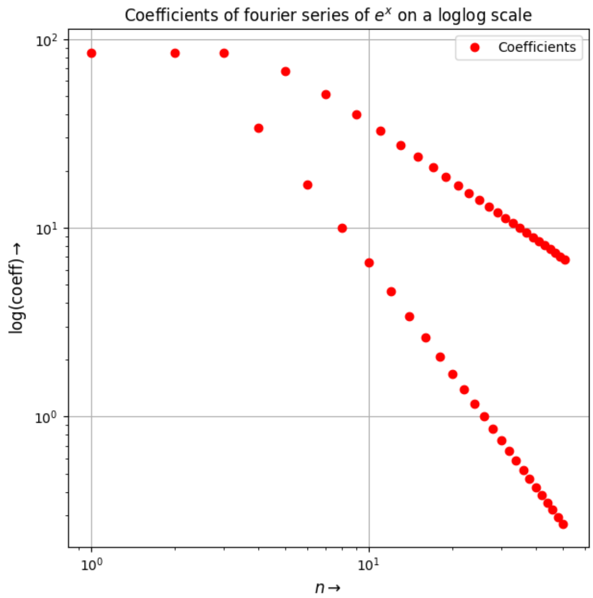

Figure 4: loglog plot of Fourier coefficients of e^x

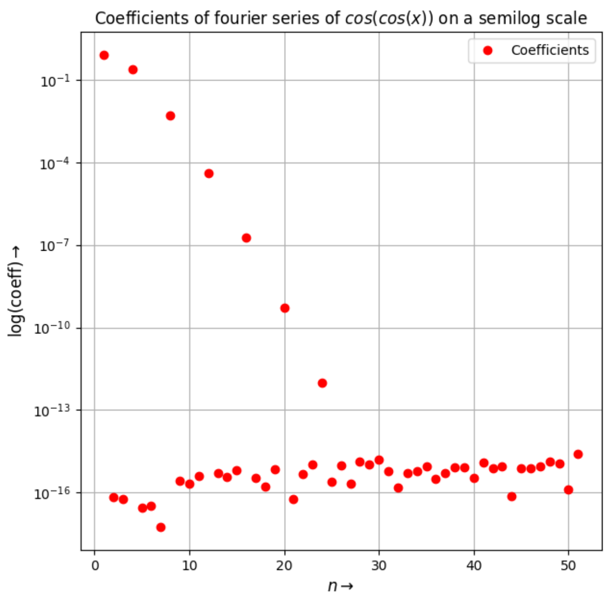

Figure 5: Semilog plot of Fourier coefficients of $cos(cos(x))$

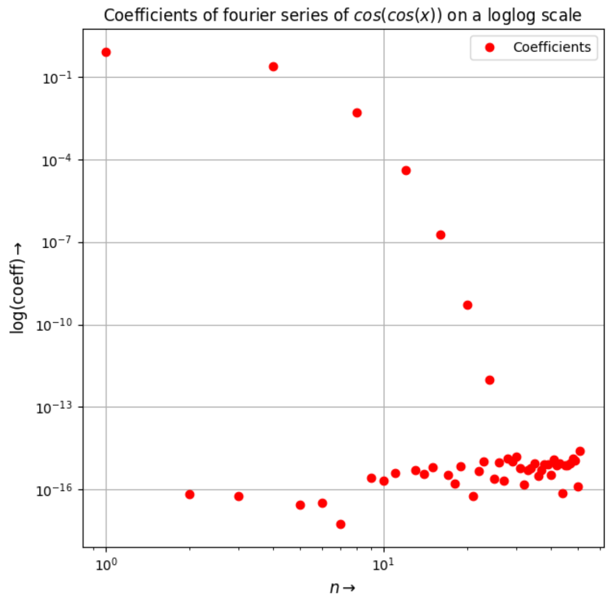

Figure 6: loglog plot of Fourier coefficients of $cos(cos(x))$

$cos(cos(x))$ being finite valued periodic function and extremes of small magnitude the coefficients for larger n approach zero whereas for e^x it is an increasing function and hence do not decay quickly

loglog plot for an exponential function is linear in general hence loglog plot in Figure4 is linear and semilog plot for Figure is linear

5 Least squares approach

Create matrices A and B and use the function *lstsq()* to obtain the least squares solutions of the matrix equation. The python code snippet to create matrices and find least squares solution is

```
Coeff_cos_cos = np.zeros(51)
def matrixAb(x,f):
    b = f(x)
    A = np.zeros((x.shape[0],51))
    A[:,0] = 1
    for i in range(1,26):
        A[:,2*i-1]=np.cos(i*x)
        A[:,2*i]=np.sin(i*x)
    return A,b
```

```
x = np.linspace(0,2*np.pi,num=400,endpoint = True)
A_coscos,b_coscos = matrixAb(x,cos_cos)
A_ex,b_ex = matrixAb(x,e)
c_coscos = scipy.linalg.lstsq(A_coscos,b_coscos)[0]
c_exp = scipy.linalg.lstsq(A_ex,b_ex)[0]
```

The above found solutions are plotted along with the solutions to observe the difference between them.

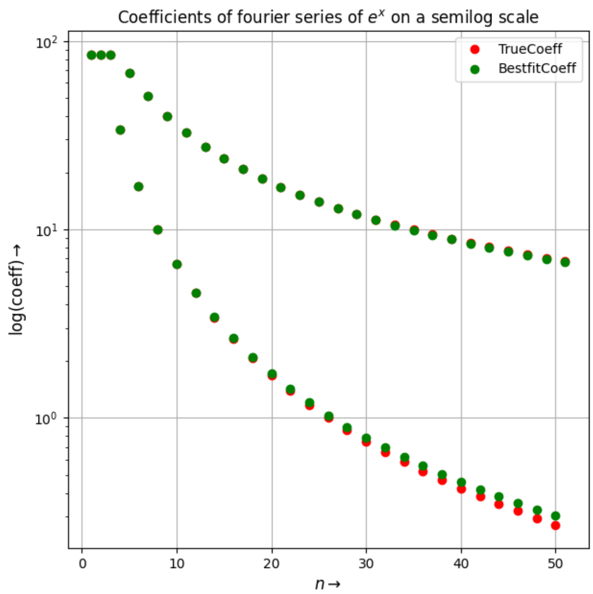

Figure 7: Semilog plot of Fourier coefficients of e^x

6 Deviation from actual Values

As least squares method to find solutions is also an approximate method we get slight deviations from the actual values, the maximum deviation is found by the following code snippet

```
def error (u,v):
    p = abs(u-v)
    p1 = abs(p.max())
    return p1
print("The error in Coefficients of e^x =",error(Coeff_exp,c_exp))
print("The error in Coefficients of cos(cos(x)) =",error(Coeff_cos_cos,c_coscos))
```

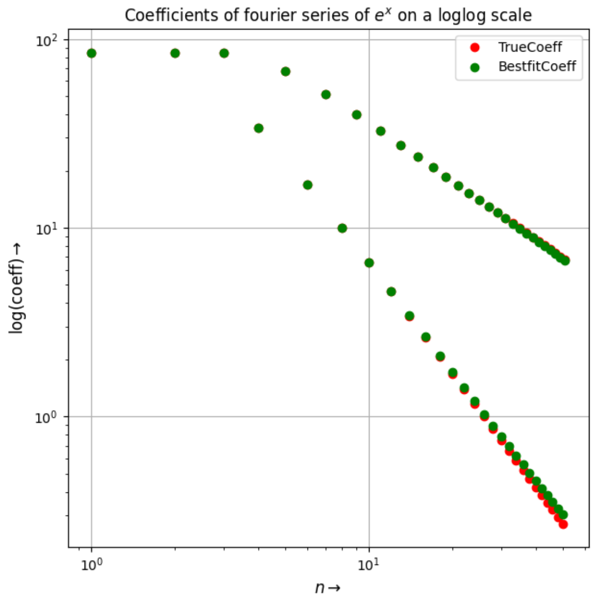

Figure 8: loglog plot of Fourier coefficients of e^x

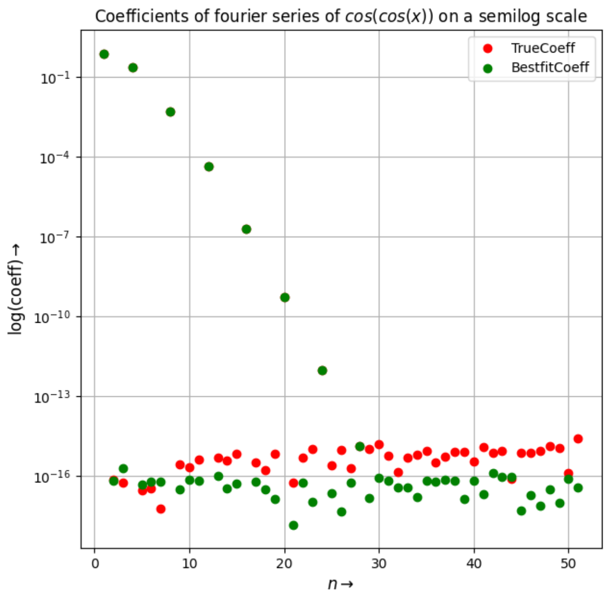

Figure 9: Semilog plot of Fourier coefficients of $cos(cos(x))$

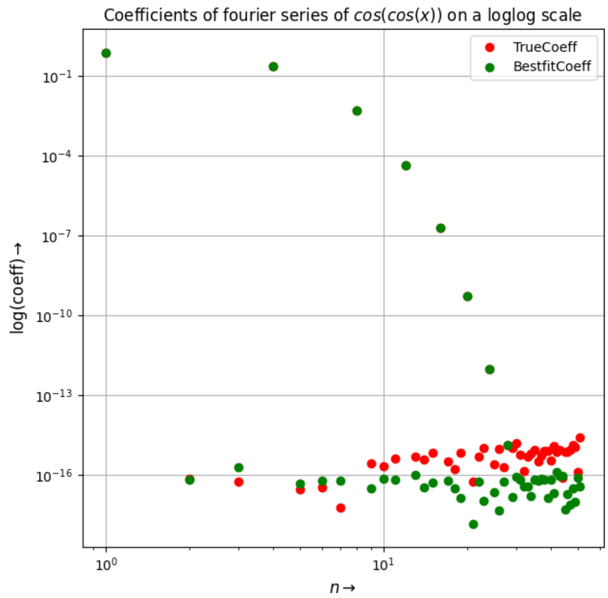

Figure 10: loglog plot of Fourier coefficients of $cos(cos(x))$

The absolute error in case of e^x is 0.08812169778765266
The absolute error in case of $cos(cos(x))$ is 2.571903420801254e-15
From the errors it is very much clear that $cos(cos(x))$ has a very good approximation but it is not the case with e^x , it has a significant amount of difference. This is because periodic extension of e^x has discontinuties which disturb the approximation. This can be countered by increasing the number of samples, the lack of samples is felt more near the discontinuity.

7 Estimated Functions

Using the values of coefficients we got from the least squares method we plot e^x and $cos(cos(x))$.
The plots are given below: The $cos(cos(x))$ vs x graph agrees almost perfectly with great precision to the actual function, whereas the approximation of e^x varies from the original function due to discontinuities. The partial sum of Fourier series has lot of oscillations near discontinuity of the function. This oscillations reach a particular value but don't completely die out with increase in value of n. This phenomenon is also known as Gibbs phenomenon.

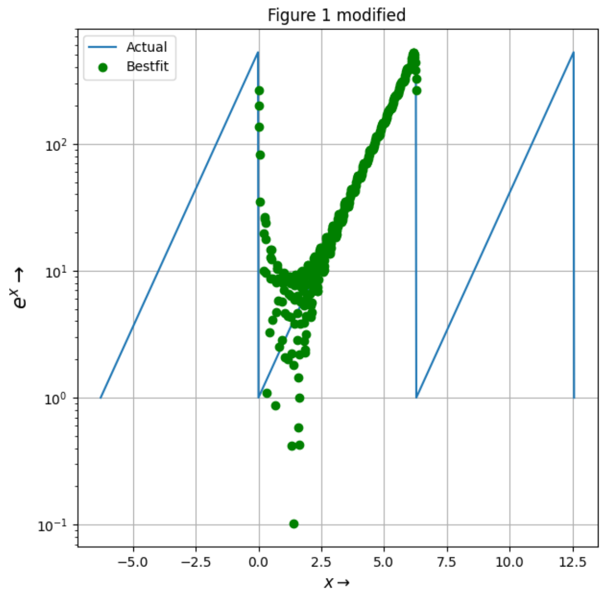

Figure 11: Actual and approximated plot of e^x

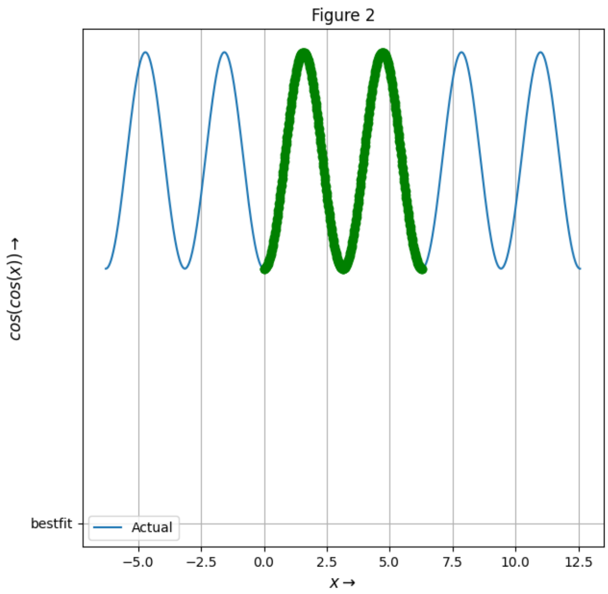

Figure 12: Actual and approximated plot of $cos(cos(x))$

Conclusions

We have examined the case of approximating functions using their fourier coefficients upto a threshold number. Whilst doing so, we perform the same for two cases, one a continuous function, and the other a function with finite discontinuities. We also observed the Gibbs phenomena or the non uniform convergence of the discontinuous functions.