Assignment 4: Fourier Approximations

Tadigoppala Yeswanth [EE19B122]

March 10, 2021

Abstract

The goal of this assignment is the following.

- To fit two functions e^x and $cos(cos(x))$ using the Fourier series.
- To use least squares fitting to simplify the process of calculating Fourier series.
- plotting graphs of the fourier coefficients and functions so that we can understand better

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

The python code snippet used to declare the functions e^x and $cos(cos(x))$ and to give x values from -2π to 4π.

```
def f(x):
    return exp(x)
def g(x):
    return cos(cos(x))

x = linspace(-2*pi,4*pi,1200)
y_exp = f(x)
y_cos = g(x)
```

The python code snippet used to plot the graphs of e^x is:

```
figure(1)
semilogy(x,y_exp,label='Actual Value')
xlabel(r'$x\rightarrow$',size=20)
ylabel(r'$e^x\rightarrow$',size=20)
grid(True)
```

The python code snippet used to plot $cos(cos(x))$ is:

```
plot(x,y_cos,label='Actual Value')
xlabel(r'$x\rightarrow$',size=20)
ylabel(r'$cos(cos(x))\rightarrow$',size=20)
grid(True)
```

The plots of e^x and $cos(cos(x))$ are as shown below:

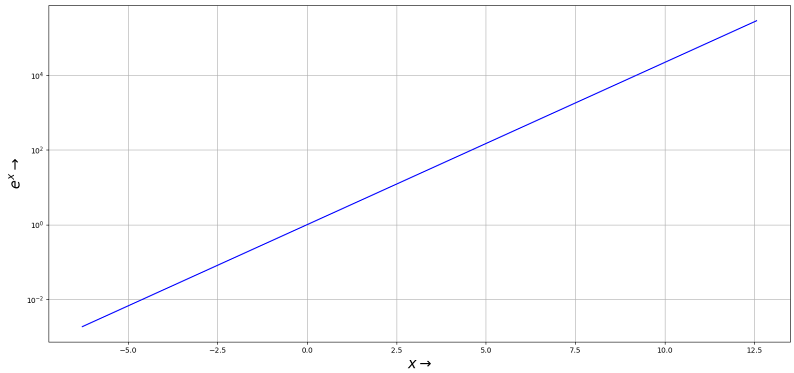

Figure 1: Plot of e^x

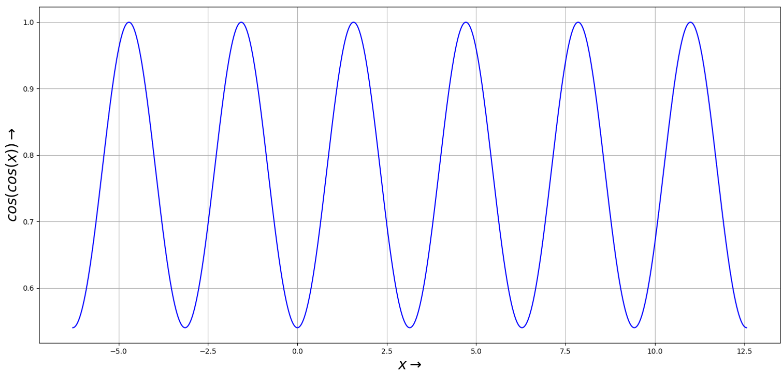

Figure 2: plot of $cos(cos(x))$

From the plots we can see that e^x is aperiodic and $cos(cos(x))$ is periodic. But fourier series is a periodic. So we have to plot periodic extension of above figures and the plots of periodic extensions of e^x and $cos(cos(x))$ are shown below:

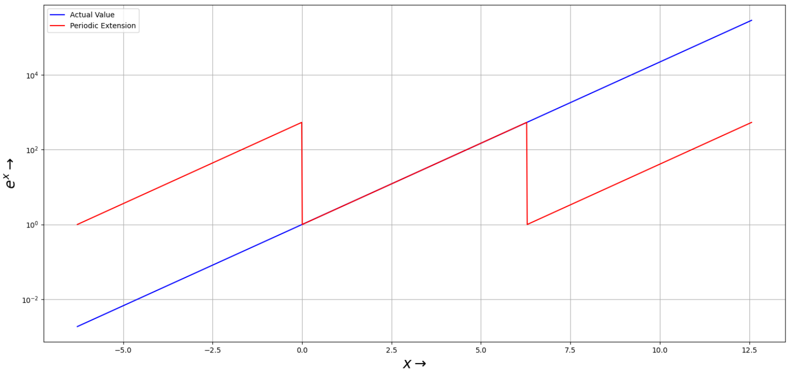

Figure 3: Data plot

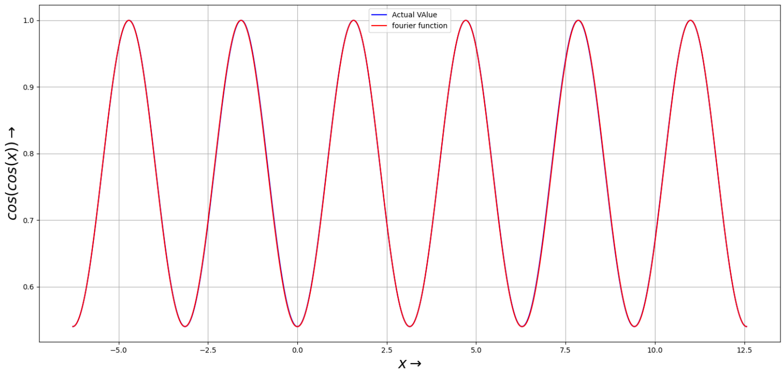

Figure 4: Data plot

The Fourier Series Coefficients

The fourier series used to approximate a function is as follows:

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

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

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

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

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

In python we will use the *quad()* function to perform an integration. First we'll have to create functions which contains the variable k and x. The python code snippet for declaring the functions with an additional variable k and to find fourier coefficients is as follows:

```
C_exp = zeros((51,1))
C_cos = zeros((51,1))
C_exp[0][0] = (1/(2*pi))*(integrate.quad(f,0,2*pi))[0]
C_cos[0][0] = (1/(2*pi))*(integrate.quad(g,0,2*pi))[0]
for k in range(1,26):
def u_exp(x,k):
return f(x)*cos(k*x)
def v_exp(x,k):
return f(x)*sin(k*x)
def u_cos(x,k):
return g(x)*cos(k*x)
def v_cos(x,k):
return g(x)*sin(k*x)
result_exp_a = integrate.quad(u_exp,0,2*pi,args=(k))
result_exp_b = integrate.quad(v_exp,0,2*pi,args=(k))
result_cos_a = integrate.quad(u_cos,0,2*pi,args=(k))
result_cos_b = integrate.quad(v_cos,0,2*pi,args=(k))
C_exp[2*k-1][0] = (1/pi)*result_exp_a[0]
C_exp[2*k][0] = (1/pi)*result_exp_b[0]
C_cos[2*k-1][0] = (1/pi)*result_cos_a[0]
C_cos[2*k][0] = (1/pi)*result_cos_b[0]
```

The plots of Fourier coefficients of e^x and $cos(cosx)$:

The semilog and log plots of the Fourier coefficients of e^x are as as shown:

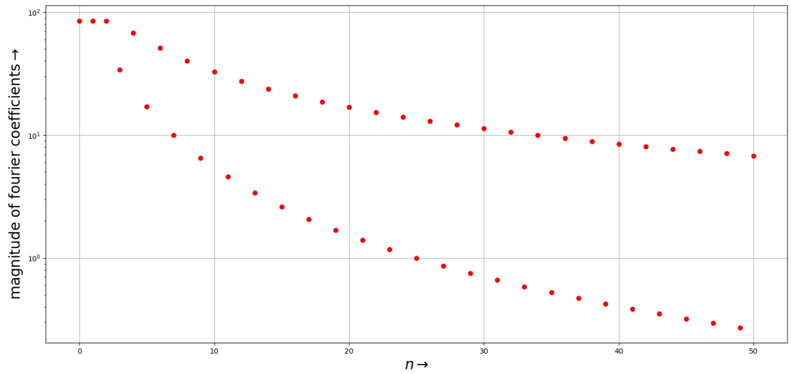

Figure 5: Semilog plot of the fourier coefficients of e^x

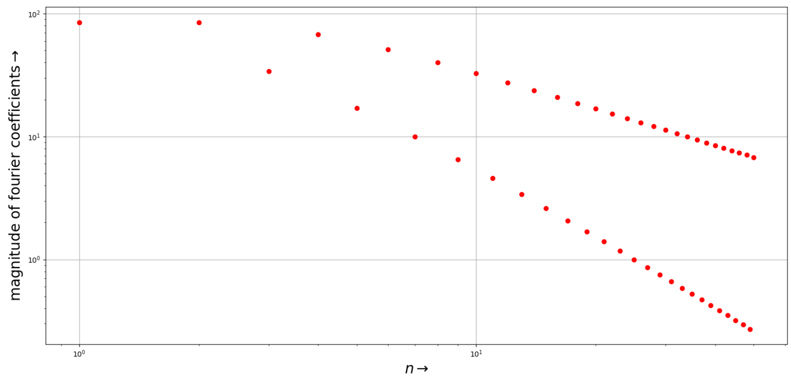

Figure 6: loglog plot of the fourier coefficients of e^x

The semilog and loglog plots of fourier coefficients of $cos(cosx)$ are as shown:

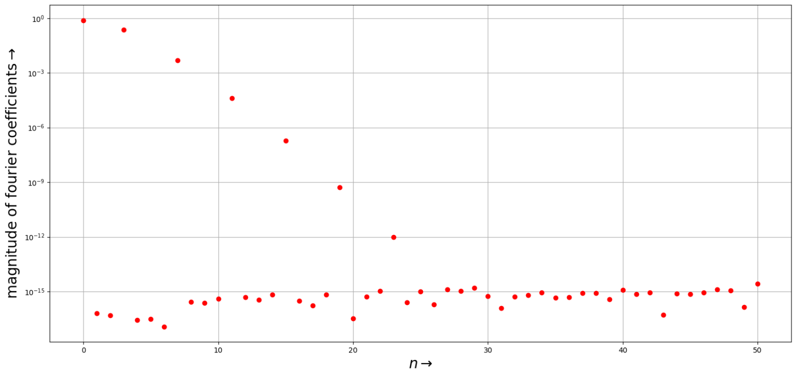

Figure 7: Semilog plot of the fourier coefficients of $cos(cos(x))$

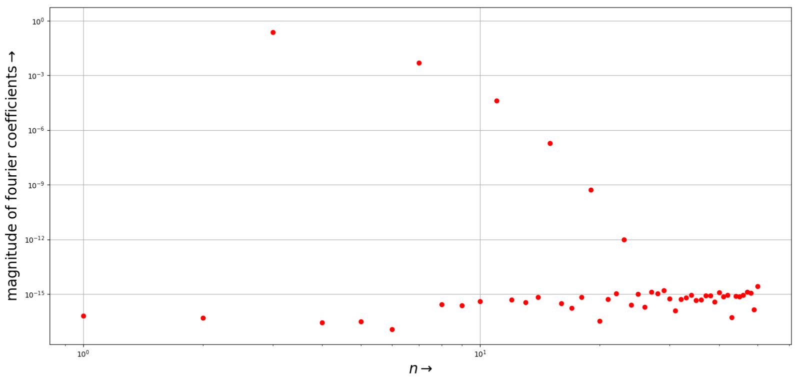

Figure 8: loglog plot of the fourier coefficients of $cos(cos(x))$

a. From the plots we can clearly see that b_n is nearly zero for $cos(cos(x))$. This is because $cos(cos(x))$ is an even function, Therefore in the fourier series expansion, all the b_n terms should be zero for the series to be an even function.The values obtained are non-zero because of the limitation in the numerical accuracy upto which π can be stored in memory.

b.In the first case, the function, having an exponentially increasing gradient, contains a wide range of frequencies in its fourier series. In the second case, cos(cos(x)) has a relatively low frequency of $\frac{1}{\pi}$ and thus contribution by the higher sinusoids is less, manifesting in the quick decay of the coefficients with n.

c. The loglog plot is linear for e^t since Fourier coefficients of e^t decay with $1/n$ or $1/n^2$. The semilog plot seems linear in the $cos(cos(t))$ case as its fourier coefficients decay exponentially with n.

The Least Squares Approach

For the least squares approach, we'll have to create matrices and then use *lstsq()* function inorder to get the most approximate values of the fourier coefficients. The matrix equation is

$$Ac = b \tag{5}$$

The python code snippet to create the matrices and to get the least squared value of the coefficients is as follows:

```
x2 = linspace(0,2*pi,401)
x2 = x2[:-1] # drop last term to have a proper periodic integral
b_exp=f(x2) # f has been written to take a vector
b_cos=g(x2)
A = zeros((400,51)) # allocate space for A
A[:,0]=1
for k in range(1,26):
A[:,2*k-1]=cos(k*x2) # cos(kx) column
A[:,2*k]=sin(k*x2) # sin(kx) column
c1=lstsq(A,b_exp,rcond=None)[0]
c2=lstsq(A,b_cos,rcond=None)[0]
B_exp = dot(A,c1)
B_cos = dot(A,c2)
```

The plots of fourier coefficients (both estimated and calculated) showing both loglog and semilog of e^x are:

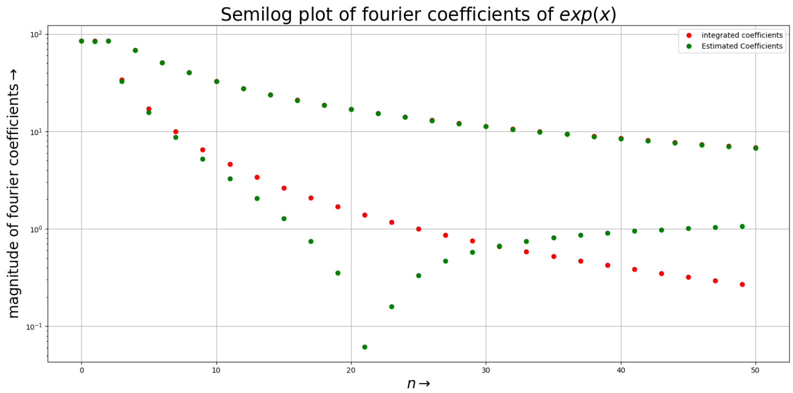

Figure 9: Semilog plots for e^x

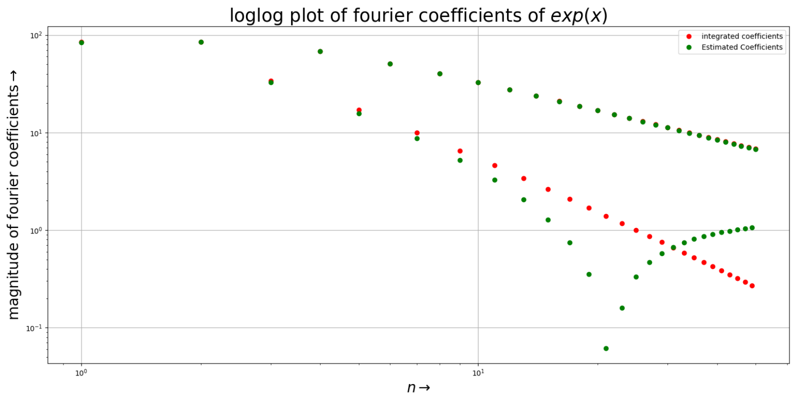

Figure 10: loglog plots for e^x

The plots of fourier coefficients (both estimated and calculated) showing both loglog and semilog of $cos(cosx)$ are:

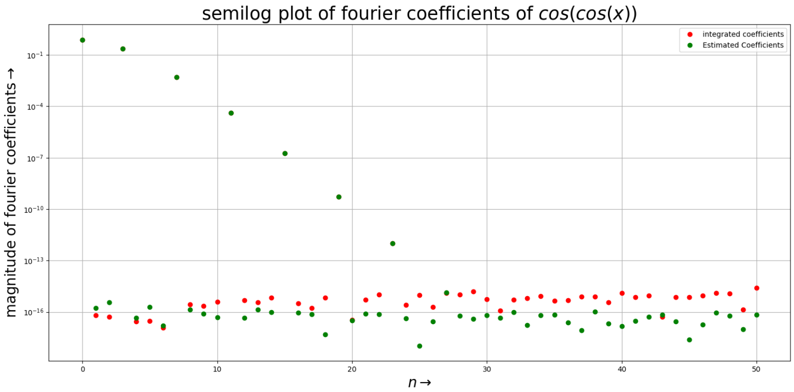

Figure 11: Semilog plots for $cos(cos(x))$

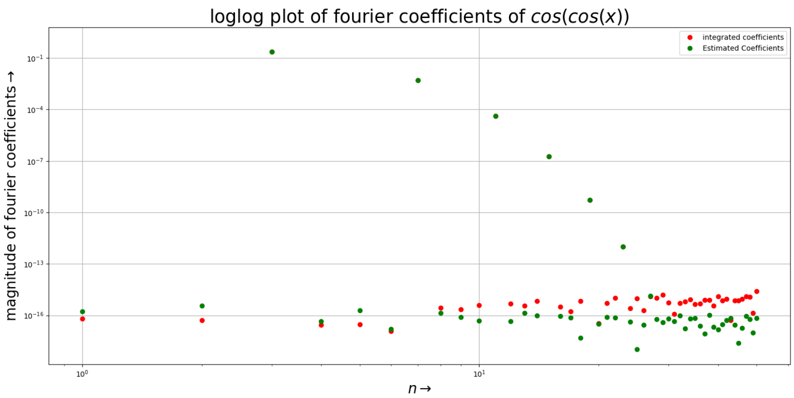

Figure 12: loglog plots for $cos(cos(x))$

Deviation of Fourier Coefficients from Actual Values

The least squares approach is still an approximate method and will definitely have a slight deviation from the actual value.
The following python code snippet is used to calculate the deviation:

```
Abs_exp = abs((c1)-transpose(C_exp))
Abs_cos = abs((c2)-transpose(C_cos))
```

```
largedev_exp = Abs_exp.max()
largedev_cos = Abs_cos.max()
```

The deviation in the case of e^x is1.33273087033
The deviation in the case of $cos(cos(x))$ is $2.66550 * 10^{-}15$

There is no noticable difference in values in the case of $cos(cos(x))$ but a significant amount of difference in the case of e^t. The reason for this is that the periodic extension of the exponential function is discontinuous, and hence would require a lot more samples to accurately determine its Fourier coefficients. If we increased the number of samples to 10^6, the maximum deviation would reduce, but not vanish. The effect of this lack of samples is felt more near the discontinuity of the signal.

Plotting Estimated Functions and Actual Functions

Using the predicted values of the fourier coefficients, we can calculate the functional values for both e^x and $cos(cos(x))$.
The plots showing both the actual and predicted functional values are as shown below:

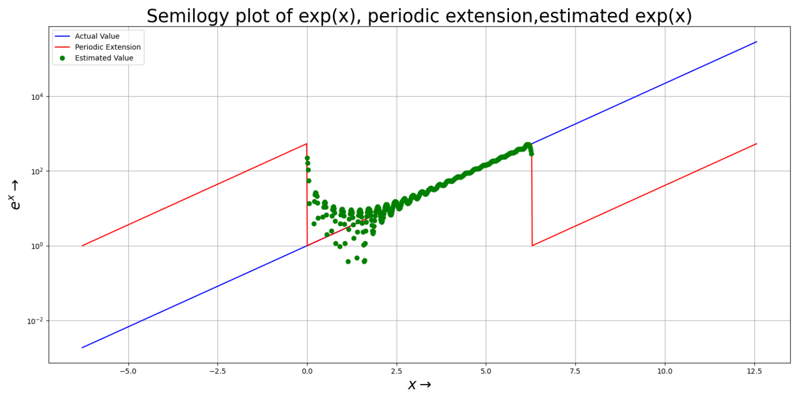

Figure 13: Actual and predicted values for e^x

The $cos(cos(x))$ vs x graph, agrees almost perfectly, beyond the scope of the precision of the least squares fitter. The Fourier approximation of e^x does not agree very well to the ideal case near the discontinuity. The cause for this is the Gibbs Phenomenon, which can be described as below. The partial sums of the Fourier series will have large oscillations near the discontinuity of the function. These oscillations do not die out as n increases, but approaches a finite limit. This is one of the causes of ringing artifacts in

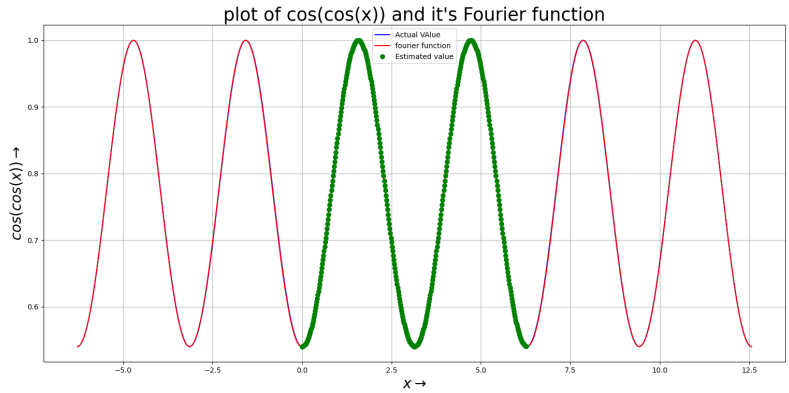

Figure 14: Actual and predicted values for $cos(cos(x))$

signal processing, and is very undesirable. Plotting the output on a linear graph would make this ringing much more apparent.

Conclusions

- We saw two different ways to calculate the Fourier series of a periodic signal.
- We saw how least squares fitting can be used to simplify the process of calculating the Fourier Series.
- We observed Gibbs phenomenon at the discontinuity in the Fourier approximation of e^t.