

## Assignment 4: Fourier Approximations

ROHIT KUMAR [EE20B111]

March 11, 2022

### Abstract

The main goals of this assignment are :-

- To fit two functions  $e^x$  and  $\cos(\cos(x))$  using their computed Fourier coefficients through direct integration.
- To use least squares fitting method to compute the Fourier coefficients for the two functions  $e^x$  and  $\cos(\cos(x))$ .
- To understand and compare the variation in these two different methods by plotting graphs.

### 1 Question 1

Define Python functions for the two functions above, that take a vector (or scalar) input, and return a vector (or scalar) value. Plot the functions over the interval  $[-2\pi, 4\pi]$  in Figure 1 and 2 respectively. Determine whether the functions are periodic. What function do you expect to be generated by the Fourier series? Compute and plot those functions as well in the respective figures.

The following python code snippet is used to declare the functions  $e^x$  and  $\cos(\cos(x))$ . Also,  $x \in [-2\pi \text{ to } 4\pi]$ .

```
def Exp(x):  
    return exp(x)  
  
def coscos(x):  
    return cos(cos(x))  
  
x = linspace(-2*pi, 4 * pi, 1200)
```

The below python code snippet is used to plot the graphs of  $e^x$  and  $\cos(\cos(x))$  function wrt to x:-

```

figure(1)
title("Semilog plot of  $e^x$  function")
xlabel(r'$x\rightarrow$', size = 15)
ylabel(r'$e^x\rightarrow$', size = 15)
semilogy(x, Exp(x), 'r', label = 'True Value')
semilogy(x, Exp(t), '-b', label = 'Periodic Extension')
semilogy(xt, Est_C_Exp, 'go', label = 'Estimated Value')
grid(True)
legend()
show()

```

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

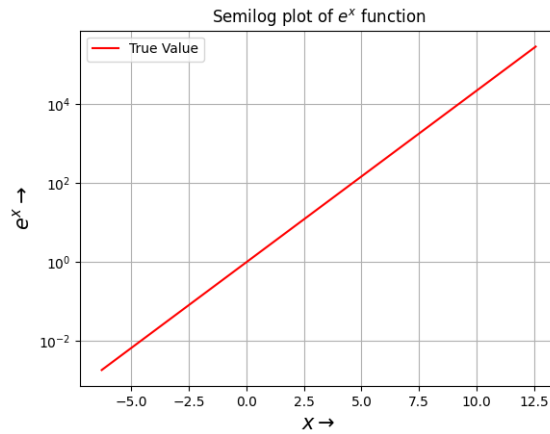


Figure 1: Semilog plot of  $e^x$

```

figure(2)
title("plot of  $\cos(\cos(x))$  function")
xlabel(r'$x\rightarrow$', size = 15)
ylabel(r'$\cos(\cos(x))\rightarrow$', size = 15)
plot(x, coscos(x), 'r', label = 'True Value')
plot(x, coscos(t), '-b', label = 'Periodic Extension')
plot(xt, Est_C_coscos, 'go', label = 'Estimated Value')
grid(True)
legend()
show()

```

From the plots, It is evident that  $e^x$  is aperiodic whereas  $\cos(\cos(x))$  is periodic. The function  $e^x$  is ever increasing with  $x$  and is non-periodic. The

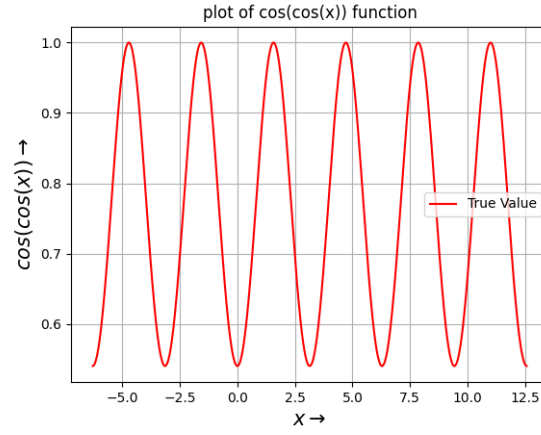


Figure 2: Plot of  $\cos(\cos(x))$

functions generated by the fourier series should be a periodic extension of the actual functions. However for the evaluation of fourier series, the function is made  $2\pi$  periodic. The plots showing the periodic extensions of  $e^x$  and  $\cos(\cos(x))$  are as shown below along with actual functions :-

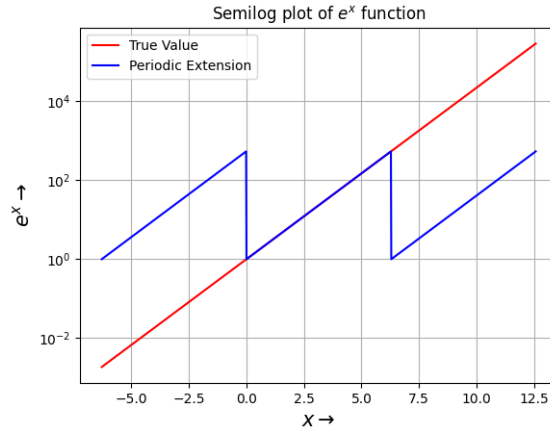


Figure 3: semilog plot for  $e^x$

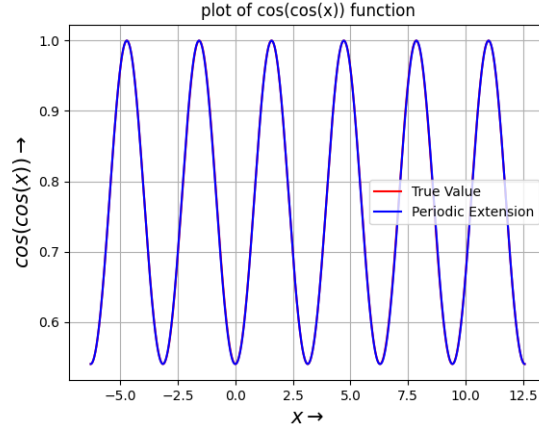


Figure 4: plot of  $\cos(\cos(x))$

## 2 Question 2 and 3

*Obtain the first 51 coefficients for the two functions above by using quad function by applying the limits. Create a matrix of size 51\*1 with the coefficients. For each of the two functions, make two different plots using “semilogy” and “loglog” and plot the magnitude of the coefficients vs n.*

To define the python functions the actual function and equations are as follows:-

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

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

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

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

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

First we are defining functions which are the integrands that are present in formula to calculate fourier coefficients. Then the fourier coefficients are

calculated using the quad function. We will use the *quad()* function to perform an integration operation. Then we are assigning the values to a matrix with the 51 coefficients for both functions. The python code for finding the functions, finding the fourier coefficients and finding matrix is shown below.

```
def u_Exp(x, k):
    return Exp(x) * cos(k*x)

def v_Exp(x, k):
    return Exp(x) * sin(k*x)

def u_cosc(x, k):
    return cosc(x) * cos(k*x)

def v_cosc(x, k):
    return cosc(x) * sin(k*x)

xt = linspace(0, 2*pi, 400)
t = tile(xt, 3)
x = linspace(-2*pi, 4 * pi, 1200)

C_Exp = np.zeros((51, 1))
C_Exp[0][0] = (1/(2*pi)) * (integrate.quad(Exp, 0, 2 * pi))[0]
for i in range(1, 26):
    C_Exp[(2*i)-1][0] = (1/pi) * (integrate.quad(u_Exp, 0, 2 * pi, args =
(i)))[0]
    C_Exp[2*i][0] = (1/pi) * (integrate.quad(v_Exp, 0, 2*pi, args = (i)))[0]

C_cosc = np.zeros((51,1))
C_cosc[0][0] = (1/(2*pi)) * (integrate.quad(cosc, 0, 2 * pi))[0]
for i in range(1,26):
    C_cosc[(2*i)-1][0] = (1/pi) * (integrate.quad(u_cosc, 0, 2*pi, args =
(i)))[0]
    C_cosc[2*i][0] = (1/pi) * (integrate.quad(v_cosc, 0, 2 * pi, args =
(i)))[0]
```

### Plots of Fourier coefficients for both functions

The semilog plot for the Magnitude of Fourier coefficients of  $e^x$  and the loglog plot for the Magnitude of fourier coefficients of  $\cos(\cos(x))$  are shwn below:-

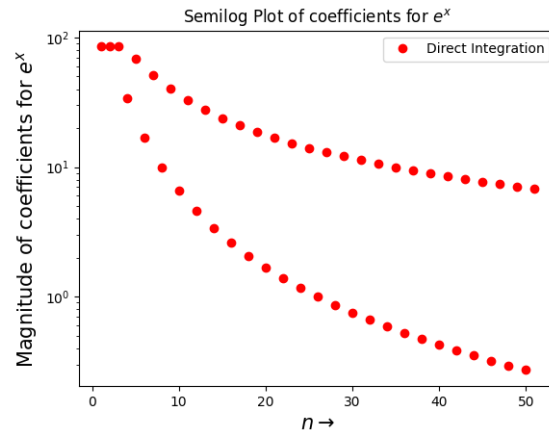


Figure 5: Semilog plot of the fourier coefficients of  $e^x$

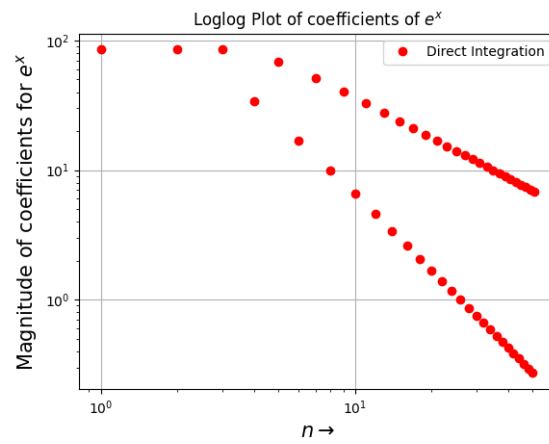


Figure 6: Loglog plot of the fourier coefficients of  $e^x$

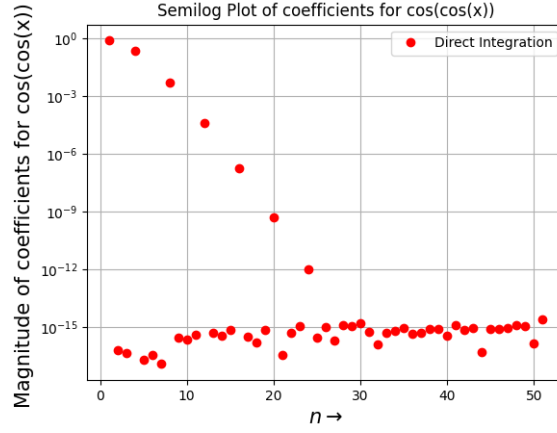


Figure 7: Semilog plot of the fourier coefficients of  $\cos(\cos(x))$

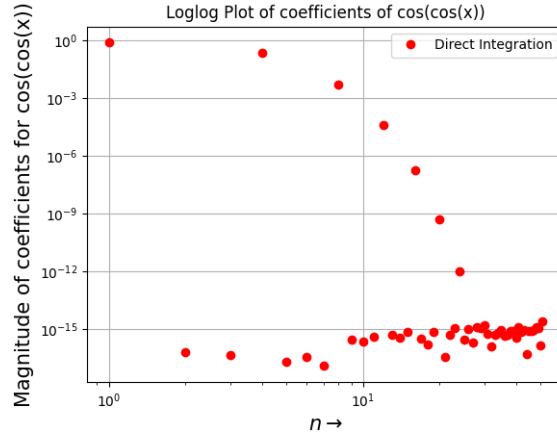


Figure 8: Loglog plot of the fourier coefficients of  $\cos(\cos(x))$

1. From the plots, it is evident that  $b_n$  coefficients are nearly zero for  $\cos(\cos(x))$ . Since  $\cos(\cos(x))$  is an even function, it does not contain odd harmonics(the integrand becomes odd and hence whole integral is zero). Hence in the fourier series expansion, all the  $b_n$  terms should be zero for the series to be an even function.

2. The magnitude of the coefficients would represent how much of certain frequencies happen to be in the output.  $\cos(\cos(t))$  does not have very many frequencies of harmonics, so it dies out quickly. However, since the periodic extension of  $e^t$  is discontinuous. To represent this discontinuity as a sum of continuous sinusoids, we would need high frequency components, hence coefficients do not decay as quickly.

3. The loglog plot looks linear for  $e^t$  since Fourier coefficients of  $e^t$  decay as a power of  $n$ , in particular,  $a_n$  decays as  $1/n$  and  $b_n$  decays as  $1/n^2$ . The magnitude of the complex fourier coefficients of  $e^x$  therefore decays as  $1/n$ . This is because of the discontinuities in the periodic extension of  $e^x$ . The semilog plot seems linear in the  $\cos(\cos(t))$  case as its fourier coefficients decay exponentially with  $n$ .

### 3 Question 4,5 and 6

***Using least squares approach, Find the fouriers coefficients for both the functions by creating the matrix A,  $A*c = b$ . Plot the curves. Compare the answers got by least squares and by the direct integration.***

The fourier coefficients are estimated using least square estimation other than integration. 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 python code to create the matrices and to get the least squared value of the coefficients is as follows:

```
A = np.zeros((400, 51))
A[:, 0] = 1
vector_x = linspace(0, 2*pi, 401)
vector_x = vector_x[:-1]

for i in range(1, 26):
    A[:, 2*i-1] = cos(i*vector_x)
    A[:, 2*i] = sin(i*vector_x)
C_Expvector_x = Exp(vector_x)
C_coscvector_x = coscos(vector_x)

lstsq_C_Exp = lstsq(A, C_Expvector_x, rcond=None)[0]
lstsq_C_coscvector_x = lstsq(A, C_coscvector_x, rcond=None)[0]
```

The plots in order to show the differences between the actual and predicted values of the fourier coefficients are shown below:-



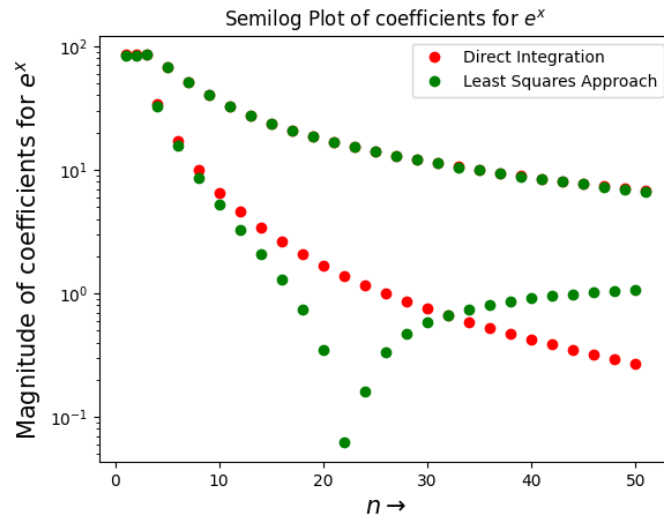


Figure 9: Semilog plots for  $e^x$

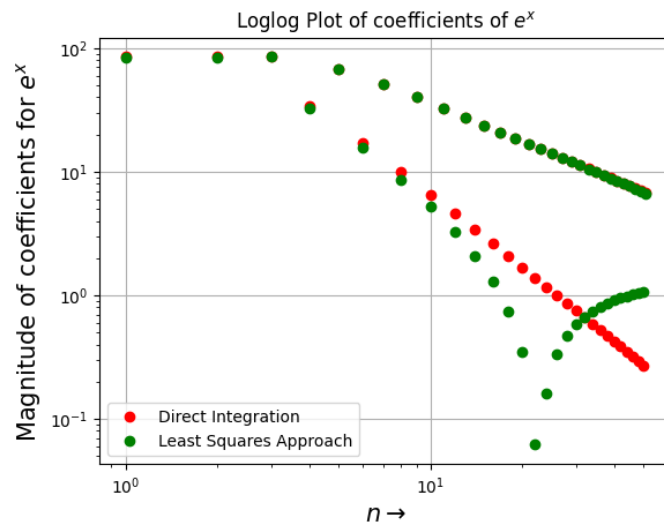


Figure 10: Loglog plots for  $e^x$

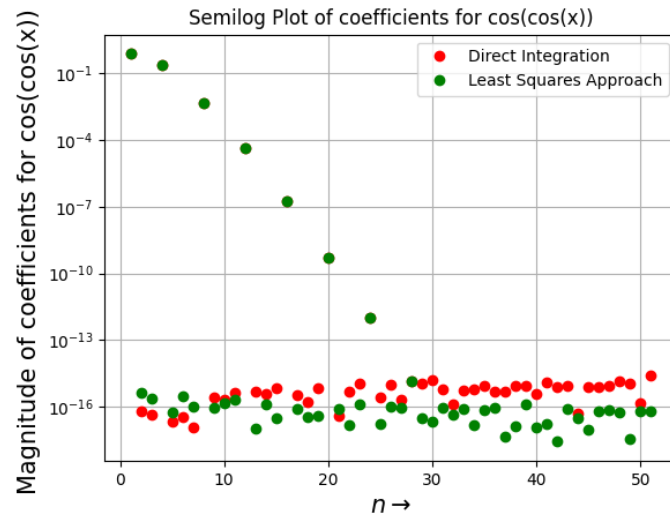


Figure 11: Semilog plots for  $\cos(\cos(x))$

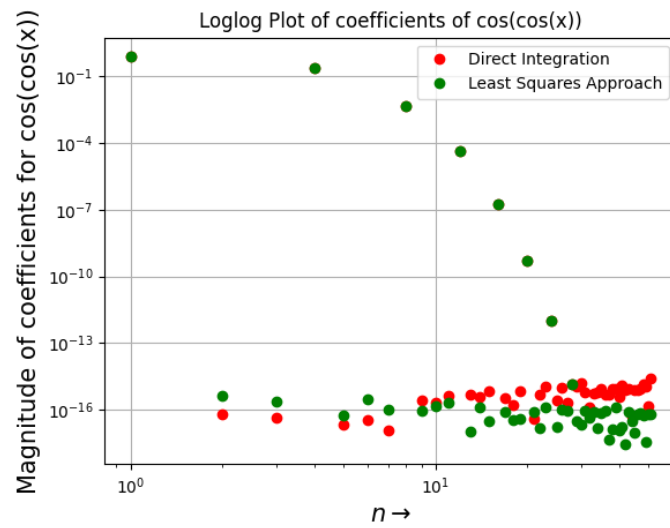


Figure 12: Loglog plots for  $\cos(\cos(x))$

## 4 Question 6

**Compare the answers got by least squares and by the direct integration. Do they agree? Should they? How much deviation is there (find the absolute difference between the two sets of coefficients and find the largest deviation. How will you do this using vectors?)**

Since the least squares approach is only approximate method, slight deviation is expected from actual value.

The following python code used to calculate the absolute difference and largest deviation:-

```
Abs_Diff_Exp = abs(lstsq_C_Exp - Ct_Exp)
Abs_Diff_cosc = abs(lstsq_C_cosc - Ct_cosc)

Lar_Dev_Exp = max(Abs_Diff_Exp[0])
Lar_Dev_cosc = max(Abs_Diff_cosc[0])
```

The largest deviation between coefficients for  $e^x$  is 1.332730870335368

The largest deviation between coefficients for  $\cos(\cos(x))$  is 2.669164544992301e-15

From the plots, the coefficients of  $\cos(\cos(x))$  are more agreeable but not much agreeable for coefficients of  $e^x$ . This is because a significant deviation in  $e^x$  case. The reason for this is  $e^x$  function is not periodic and  $\cos(\cos(x))$  is periodic. The periodic extension of  $e^x$  function is discontinuous and has increasing gradient. So, it requires more samples i.e.  $>> 400$  to reduce the deviation and get accurate fourier coefficients. Even increasing the samples won't give exact result but helps to reduce the deviation mostly. The deviation is mostly seen at discontinuous points.

## 5 Question 7

**Compute  $A_c$  from the estimated values of  $c$ . These should be the function values at  $x_i$ . Plot them (with green circles) in Figures 1 and 2 respectively for the two functions. Why is there so much deviation in Figure 1 but nearly perfect agreement in Figure 2?**

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

The matrix product  $A * c = b$ . By multiplication 400 functional values are obtained for both functions.

The python code for this matrix generation is as shown :-

```

Est_C_Exp = dot(A, lstsq_C_Exp)
Est_C_coscoss = dot(A, lstsq_C_coscoss)

```

The plots showing both the actual and estimated functional values are as shown below:

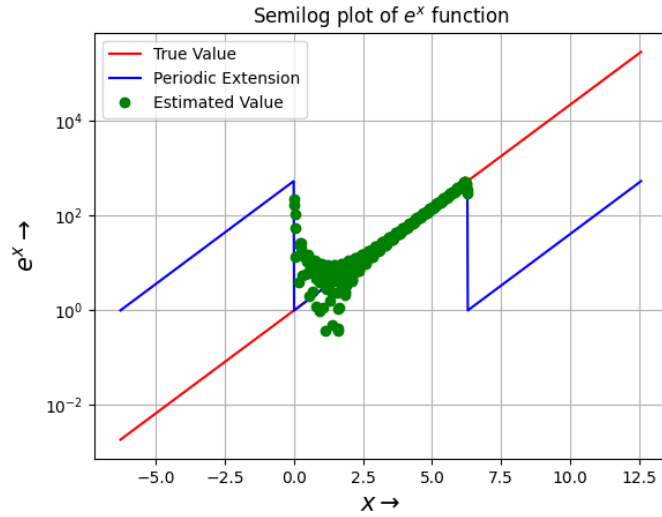


Figure 13: Actual and estimated values for  $e^x$

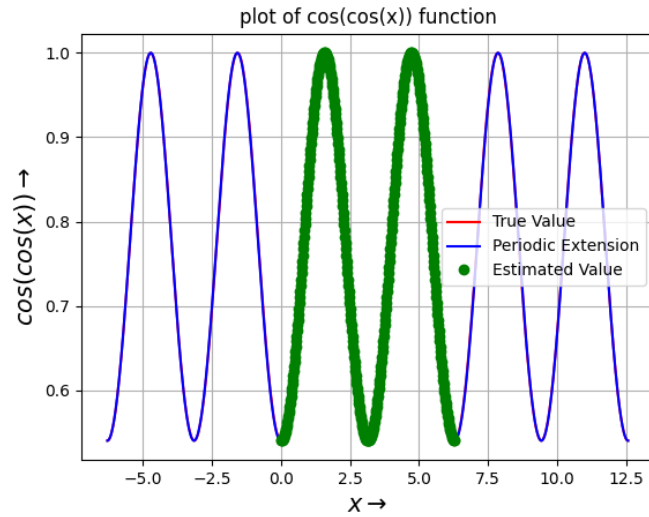


Figure 14: Actual and estimated values for  $\cos(\cos(x))$

There is much deviation in Figure 1 but nearly perfect agreement in Figure 2 because it should be noted that Fourier series are for periodic functions. Without consideration the more samples in least square method,  $\cos(\cos(x))$  plot is almost perfectly same. The reason why the  $e^x$  plot is more deviated is the partial sums of the Fourier series will have large oscillations near the discontinuity of the function where  $n$  increases, these oscillations wont die out but reaches a finite limit. This is known Gibbs phenomenon.

## Conclusions

- The odd harmonics are zero for even functions in fourier coefficients calculation. The fourier series can be calculated by direct integration and also through least squares method.
- The fourier series can be calculated by direct integration and also through least squares method.
- More samples are to be taken to reduce the deviation in least squares. Mainly for discontinuous functions at discontinuous points.
- For computation of fourier coefficients, least squares method is not effective than direct integration in this case of fourier series(non-periodic).
- The Gibbs phenomenon is observed at discontinuity points.