EE2703: Assignment 4

Sarthak Vora (EE19B140)

March 11, 2021

1 Abstract

In this report, we will look at approximating some functions using the Fourier series. We employ two methods to find the Fourier approximation of e^x and $cos(cos(x))$:

- Direct Integration Method
- Least Squares Method

We shall see how to compare the Fourier coefficients in these two methods.

2 Introduction

Any periodic function can be expressed as a linear combination of various sinusoids. We can also approximate any function in the domain [0,2π) by extending the function periodically on either side of x-axis, thus creating a periodic function.

For a function $f(x)$, the Fourier series representation is:

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

and,

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

$$\begin{aligned} a_k &= \frac{1}{\pi}\int_0^{2\pi} f(x)cos(kx)dx \\ b_k &= \frac{1}{\pi}\int_0^{2\pi} f(x)sin(kx)dx \end{aligned} \tag{3}$$

The above equations describe the *Direct Integration* method of finding the Fourier approximation of a function.

3 Procedure

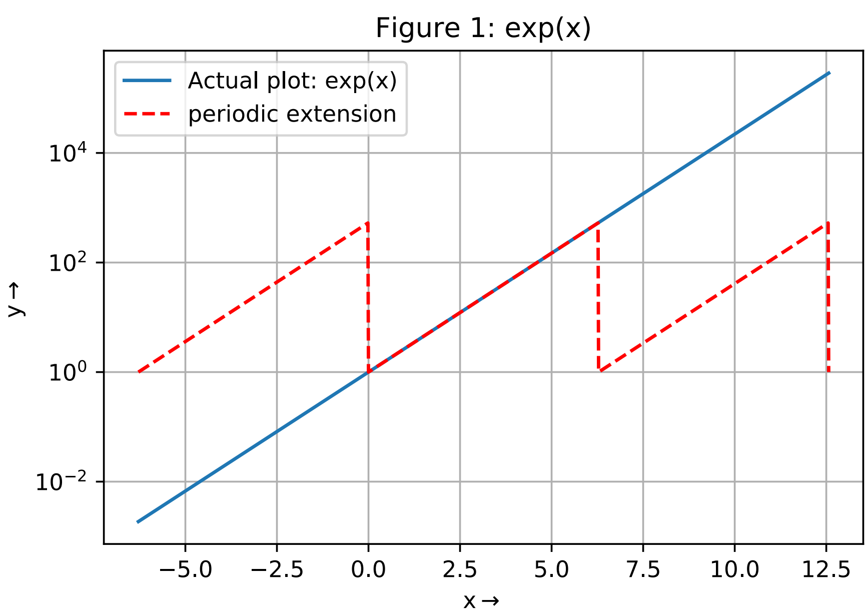
3.1 Task 1

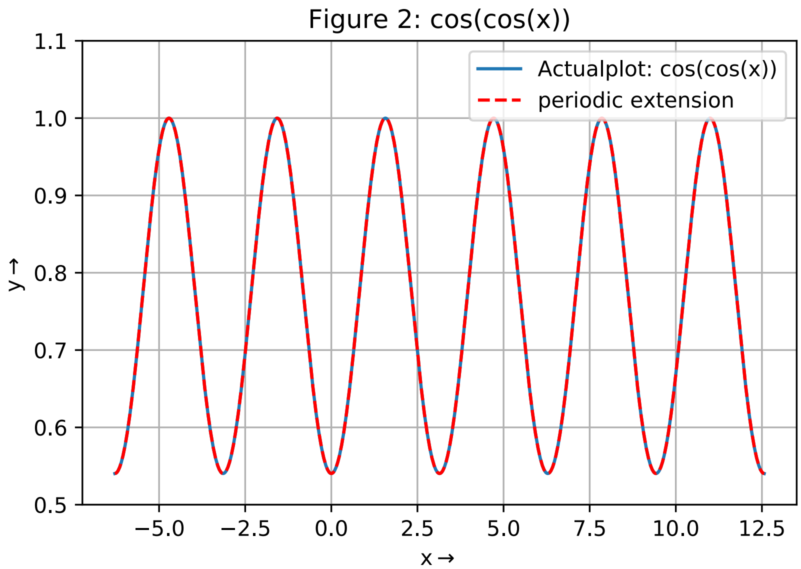
- Define python functions for e^x and $cos(cos(x))$ that can take an input vector (or scalar) and return a corresponding vector(or scalar) value.
- Plot both the functions in the range $[-2\pi, 4\pi)$.
- To determine whether the functions are periodic.

3.1.1 Code

```
def exponential(x):
    out = exp(x)
    return out

def coscos(x):
    out_i = cos(x)
    out_f = cos(out_i)
    return out_f
```

3.1.2 Plots





3.1.3 Observation

- $\cos(\cos(x))$ is a periodic function with a period 2π whereas e^x is a monotonically increasing, non-periodic function.
- Due to non periodicity of exponential function, it is periodically extended from the range $[0,2\pi]$ to the entire real axis range.

3.2 Task 2

- Obtain the first 51 coefficients for the functions e^x and $\cos(\cos(x))$.
- Use the built-in Python integrator function, ***quad***.

3.2.1 Code

```
def u(x,k,f):
    out_1 = f(x)*cos(k*x)
    return out_1

def v(x,k,f):
    out_2 = f(x)*sin(k*x)
    return out_2

def get51coeffs(f):
    acoeffs = list()
    bcoeffs = list()
    acoeffs.append(quad(u,0,2*pi,args=(0,f))[0]/(2*pi))
    bcoeffs.append(0)
    for i in linspace(1,25,25):
        acoeffs.append(quad(u,0,2*pi,args=(i, f))[0]/pi)
        bcoeffs.append(quad(v,0,2*pi,args=(i, f))[0]/pi)

    coeffs = [0 for i in linspace(0,50,51)]
    coeffs[0] = acoeffs[0]
    coeffs[1::2] = acoeffs[1:]
    coeffs[2::2] = bcoeffs[1:]
```

```
    return coeffs

exp_coeff = get51coeffs(exponential)
coscos_coeff = get51coeffs(coscos)
```

3.2.2 Observation

- The first 51 coefficients are generated using the **quad** function available in *scipy.integrate* library
- The result is stored in two variables **expcoeff** and **coscoscoeff**, as shown in the above code.

3.3 Task 3

- Plot the magnitude of each of the two functions, e^x and $\cos(\cos(x))$ individually in a **semilog** plot and **loglog** plot.
- Plot the magnitude of **coefficients** versus **n**.

3.3.1 Code

```
# Semilogy for Fourier series coeffs of exp(x)
figure('Figure3: semilog plot: $e^x$')
semilogy(0,exp_coeff[0],'ko',label='$a_{0}$')
semilogy(linspace(1,25,25),exp_coeff[1::2],'ro',label='$a_{n}$')
semilogy(linspace(1,25,25),exp_coeff[2::2],'bo',label='$b_{n}$')
#annotate('Exact location',(0,aexp[0]))
title('Figure3: Semilogy plot for $e^x$')
xlabel(r'n$\rightarrow$')
ylabel(r'semilog(y)$\rightarrow$')
grid()
legend()

# Semilogy for Fourier series coeffs of coscos(x)
figure('Figure5: semilog plot: $cos(cos(x))$')
semilogy(0,coscos_coeff[0],'ko',label='$a_{0}$')
semilogy(linspace(1,25,25),coscos_coeff[1::2],'ro',label='$a_{n}$')
semilogy(linspace(1,25,25),coscos_coeff[2::2],'bo',label='$b_{n}$')
title('Figure5: Semilogy plot for $cos(cos(x))$')
xlabel(r'n$\rightarrow$')
ylabel(r'semilog(y)$\rightarrow$')
grid()
legend()
```

3.3.2 Plots

Figure 3: Semilogy plot for e^x

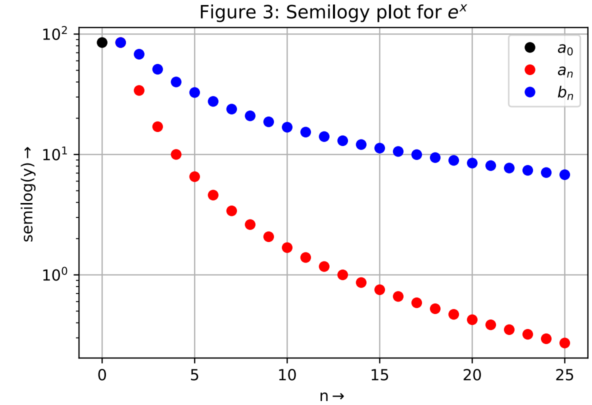


Figure 4: loglog plot for e^x

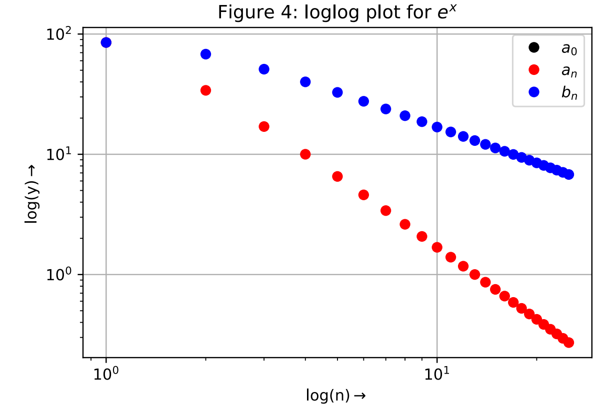


Figure 5: Semilogy plot for $cos(cos(x))$

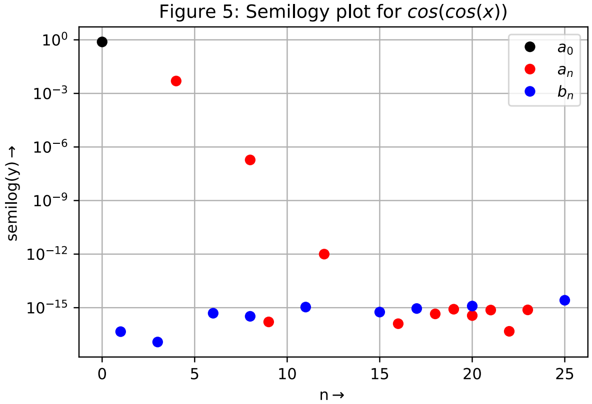
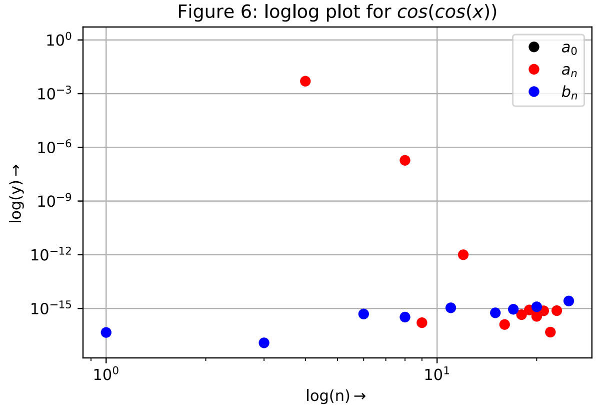


Figure 6: loglog plot for $cos(cos(x))$



3.3.3 Observation

The **Fourier Series Coefficients** are obtained by calculating the definite integral defined above. The coefficients are stored in the following manner:

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

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 sine terms should be zero.

For the case of e^x, as it increases exponentially, it has many frequency components and hence the **Fourier Series Coefficients** do no die out easily for higher frequencies. Whereas, for the case of $\cos(\cos(x))$, it as a low frequency of $\frac{1}{\pi}$ and does not have higher frequency components. Hence, the coefficients decay rapidly for the second case.

The coefficients are then plotted as shown above.

3.4 Task 4

- Using **Least Squares Approach** to find Fourier coefficients of functions using lstsq.
- Creating the matrix A and coefficient matrix B.

Least Squares Approach

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

$$\begin{pmatrix} 1 & \cos(x_1) & \sin(x_1) & ... & \cos(25x_1) & \sin(25x_1) \\ 1 & \cos(x_2) & \sin(x_2) & ... & \cos(25x_2) & \sin(25x_2) \\ ... & ... & ... & ... & ... & ... \\ 1 & \cos(x_{400}) & \sin(x_{400}) & ... & \cos(25x_{400}) & \sin(25x_{400}) \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ b_1 \\ ... \\ a_{25} \\ b_{25} \end{pmatrix} = \begin{pmatrix} f(x_1) \\ f(x_2) \\ ... \\ f(x_{400}) \end{pmatrix} \tag{4}$$

where,

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

3.4.1 Code

```
x = linspace(0,2*pi,401) # Vector with 401 steps
x = x[:-1] # dropping the last term for periodic integral

b1 = exponential(x) # vector entries for exponential(x)
b2 = coscos(x) # vector entries for cos(cos(x))

A = zeros((400,51))
A[:,0] = 1 # 1st col is all 1

range_col = [int(el) for el in linspace(1,25,25)]

for k in range_col :
    A[:,2*k-1] = cos(k*x) #cos(kx) column
    A[:,2*k] = sin(k*x) #sin(kx) column

c1 = lstsq(A,b1)[0] # best fit coeffeicients for exp(x)
c2 = lstsq(A,b2)[0] # best fit coefficients for cos(cos(x))
```

3.5 Task 5 and 6

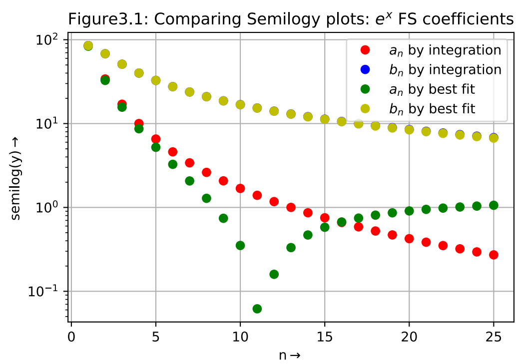
- Compare the coefficients obtained through the *least squares method* and *direct integration* method.
- Finding the maximum deviation between the coeffecients obtianed in two methods.

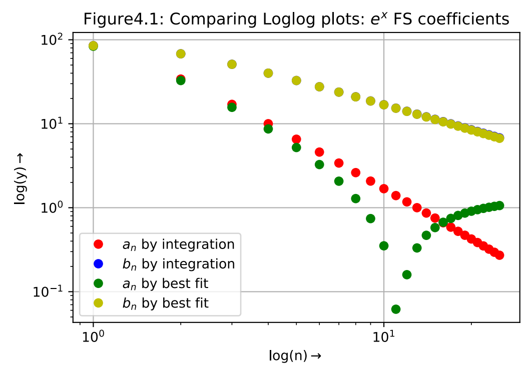
3.5.1 Code

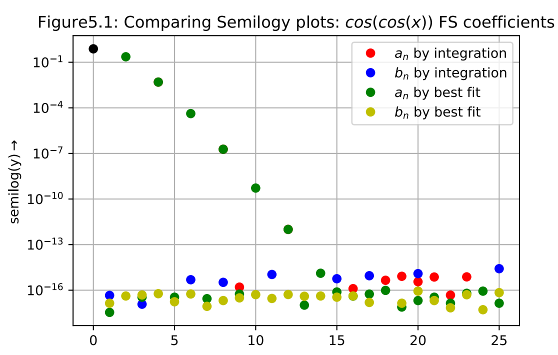
```
abserr_coscos = max([abs(coscos_coeff[int(i)] - c2[int(i)]) for i in linspace(0,50,51)])
abserr_exp = max([abs(exp_coeff[int(i)] - c1[int(i)]) for i in linspace(0,50,51)])
print("The max deviation for cos(cos(x)) FS coefficients is: " + str(abserr_coscos))
print("The max deviation for exp(x) FS coefficients is: " + str(abserr_exp))
```

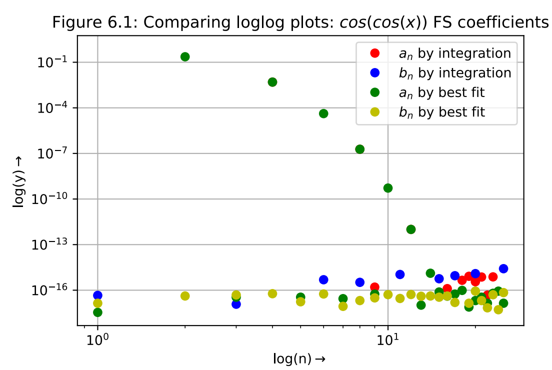
The max deviation for **cos(cos(x))** FS coefficients is: **2.674677035413032e-15.**
The max deviation for **exp(x)** FS coefficients is: **1.3327308703353395**.

3.5.2 Plots









3.5.3 Observation

- We can notice that there is some deviation of the Fourier coefficients as estimated using **Least Squares** and **Direct Integration**. We can treat the coefficients estimated using direct integration as the true values as we got those values from the function itself whereas the Least squares coefficients are just estimates.
- There is more deviation in the case of the exponential function as compared to $\cos(\cos(x)))$ as the latter is a periodic function but we are taking periodic extension of the former which increases the error factor.
- The magnitude of coefficients of e^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 $-2\log(n)$ for large n. This explains why the **loglog** plot of e^x is **linear** for large n.
- The magnitude of coefficients for $\cos(\cos(x))$ vary exponentially with n. Hence, the **semilog** plot for $\cos(\cos(x))$ is **linear**.

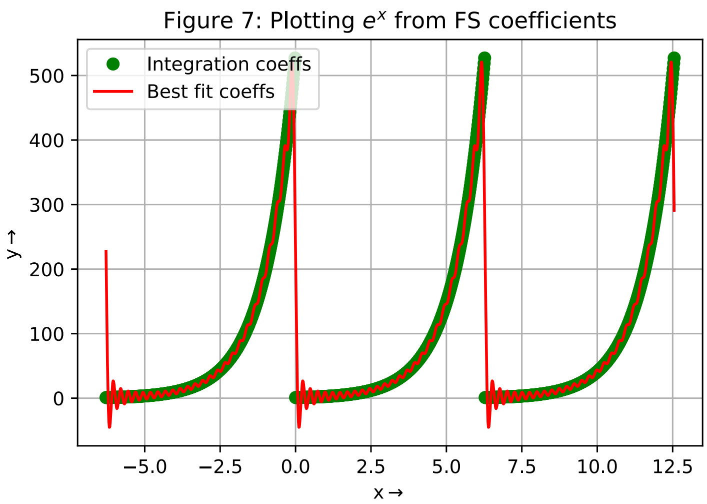
3.6 Task 7

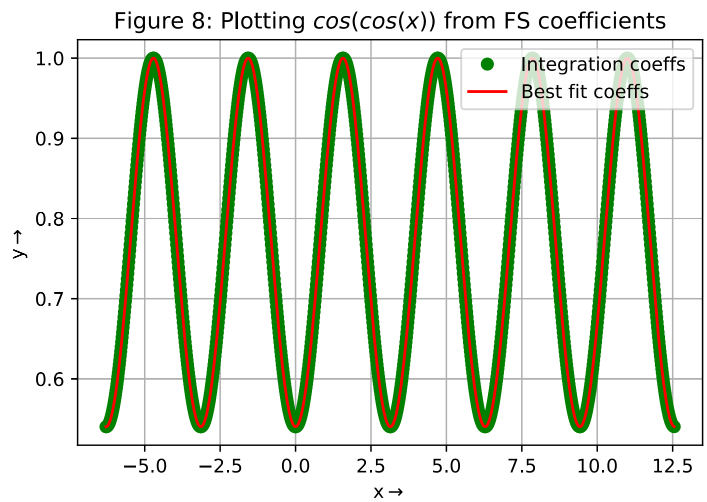
- Computing $A \cdot c$ from the estimated values of c by **Least Squares Method**.
- Plotting and comparing the reconstructed function with the original function.

3.6.1 Code

```
new_x = linspace(-2*pi,4*pi,1201)
new_x = new_x[:-1] # Removing the last elements to include periodic elements only

A_new = zeros((1200,51))
A_new[:,0] = 1 # 1st col is all 1

range_col = [int(el) for el in linspace(1,25,25)]

for k in range_col :
    A_new[:,2*k-1] = cos(k*new_x) #cos(kx) column
    A_new[:,2*k] = sin(k*new_x) #sin(kx) column

# Plotting exp(x) from its FS coefficients by integration and best fit
figure('Figure 7: Plotting exp(x) from FS coefficients')
plot(new_x, dot(A_new,c1), 'go', label='Best fit coeffs')
legend(loc='best')
title('Figure 7: Plotting $e^x$ from FS coefficients')
xlabel(r'x$\rightarrow$')
ylabel(r'y$\rightarrow$')
grid(True)

# Plotting cos(cos(x)) from its FS coefficients by integration and best fit
figure('Figure 7: Plotting cos(cos(x)) from FS coefficients')
plot(new_x, dot(A_new,coscos_coeff), 'go', label='Integration coeffs')
plot(new_x, dot(A_new,c2), 'r', label='Best fit coeffs')
legend(loc='upper right')
title('Figure 7: Plotting $cos(cos(x))$ from FS coefficients')
xlabel(r'x$\rightarrow$')
ylabel(r'y$\rightarrow$')
grid(True)
```

3.6.2 Plots





3.6.3 Observation

- It can be seen that there is considerable deviation for the the case of e^x whereas there is negligible deviation for the case of $\cos(\cos(x))$.
- This phenomenon is called **Gibbs Phenomenon** which states that *ripples* are generated at the points of discontinuity of a discontinuous functions and the ripple amplitude decreases as we move closer to the point of discontinuity.
- In our case, ripples can be seen -2π, 0, 2π, 4π in the plot of e^x. This happens because we approximate e^x using low frequency components. To approximate it with greater accuracy, we need to consider components with higher frequencies.
- However, this is not the case with $\cos(\cos(x))$ because it is a **periodic function**. Hence, it can be accurately represented using low frequencies.

4 Conclusion

- In this assignment, we observed that the the **Fourier estimation** of e^x does not exactly match the original function unlike in the case of $\cos(\cos(x))$ where we get a perfect match.
- The presence of discontinuity in the periodically extended plot of e^x gives rise to **Gibbs Phenomenon** leading to non-uniform convergence of the Fourier estimation of e^x.
- We conclude that the **Least Squares Method** gives us a computationally **efficient** way of approximating functions even though it is less accurate than **Direct Integration Method**.
- Thus, we can conclude that the **Fourier Series Approximation** method works well for smooth *periodic functions*, but is errorneous for discontinuous non-periodic functions.
