EE2703: Applied Programming Lab
Assignment 4
Fourier Series Approximations

Arjun Menon Vadakkeveedu
EE18B104
Electrical Engineering, IIT Madras

February 26, 2020

Contents

1 Introduction

Fourier Series representation of periodic functions is used to determine the harmonics present in the sequence. In this assignment, the Fourier Series (FS) coefficients of exp(x) and cos(cos(x)) are obtained and the frequency characteristics of these functions are analysed.
The Fourier Series Representation of a periodic function f(x) is given as (aka Synthesis Equation):

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

The coefficients can be computed using the following integrals (Analysis Equations):

$$a_0 = \frac{1}{T}\int_0^T f(x)dx$$

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

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

where, T is the fundamental period of the function and the integration may be over any interval of length T. Here the integration is done over [0, 2π] for convenience.
In this program, the first 51 FS coefficients are computed and the function obtained by the FS approximation using these coefficients is compared with the original function.
It must be noted that while FS approximation is defined only for periodic signals, the function exp(x) defined over the domain $(-\infty, +\infty)$ is not periodic. Thus, for computing its FS coefficients in the interval [0, 2π], we define a 2π periodic function which is equal to exp(x) at all x in the interval [0, 2π]. This does not change the behaviour of the function in this interval. In order to obtain the frequency characteristics of exp(x) over its complete domain, the FS coefficients may be computed by considering the period of the function to tend to ∞ (Fourier Transform).

1.1 Approach

1. Numeric Integration- Implementing the analysis equations in the range $[0, 2\pi]$ using scipy's quadrature integration function (scipy.integrate.quad())

2. Least Squares Regression- The function values are computed at 401 points in the range $[0, 2\pi]$. Then, the problem is reduced to that of least squares regression where the function is written as (truncated Fourier Series Representation):

$$f(x) = a_0 + \sum_{n=1}^{25}(a_n cos(nx) + b_n sin(nx))$$

   scipy's lstsq() function is used to evaluate the linear coefficients a_n and b_n

---

2 Code Structure

The program has the following blocks:

1. Vectorised code (using numpy functions) to compute f(x) and the integrands of the FS analysis equations
   Numpy functions are computationally efficient for operations over vectors and matrices in comparison to iterative operations.

2. Computing the FS coefficients
   For this purpose two approaches have been followed- by numerically implementing the analysis equations (using scipy's quadrature integrator) and by least squares regression over predetermined values of f(x).

3. Error Analysis and Data Visualisation

2.1 Important blocks used in the program

1: Compute FS Coefficients using scipy.integrate.quad for f(x) = exp(x):

```
coeff_exp = np.zeros(51)
coeff_exp[0] = quad(u_exp, 0, 2*pi, args = (k[0]))[0]/(2*pi)
for n in range(1, len(k)):
    coeff_exp[2*n - 1] = quad(u_exp, 0, 2*pi, args = (k[n]))[0]/(pi)
    coeff_exp[2*n] = quad(v_exp, 0, 2*pi, args = (k[n]))[0]/(pi)
```

2: Regenerate the function f(x) from the FS Coefficients by using the synthesis equation:

```
def regen(fn_type):
  if(fn_type == "exp"):
    est = np.ones(x.shape)*coeff_exp[0]
    for n in range(1, len(k)):
        est += coeff_exp[2*n -1]*np.cos(k[n]*x) + coeff_exp[2*n]*np.sin(k[n]*x)
  else:
    est = np.ones(x.shape)*coeff_ccos[0]
    for n in range(1, len(k)):
      est += coeff_ccos[2*n -1]*np.cos(k[n]*x) + coeff_ccos[2*n]*np.sin(k[n]*x)
  return est
```

3: Compute FS Coefficients using lstsq regression:

```
b_exp = gen_exp(x)
b_ccos = gen_ccos(x)
A= np.zeros((400,51))
A[:,0]=1        # column 1 is all ones
for n in range(1,len(k)):
  A[:,2*n-1] = np.cos(k[n]*x) # cos(kx) column
  A[:,2*n] = np.sin(k[n]*x)   # sin(kx) column
c_exp = lstsq(A,b_exp)[0]
c_ccos = lstsq(A, b_ccos)[0]
```

2.2 Input and Output formats of the program

Format for running file on Terminal:

```
python3 EE18B104_Assign4_code.py
```

Output format:

```
Mean Square Errors in FS coefficients of exp(x) =  44.91552619718075
Mean Square Errors in FS coefficients of cos(cos(x)) =  2.749484542615631e-29
Max. Deviation in exp(x) =  1.3327308703353111
Max. Deviation in cos(cos(x)) =  2.622704669603838e-15
```

Note that the error in the FS coefficients of exp(x) obtained by numeric integration and least squares regression is quite large. The regenerated function obtained from the FS coefficients of exp(x) tends to oscillate about the true value, with the deviation being large for smaller values of x. This problem is

not observed in case of cos(cos(x)).
The subsequent sections cover the plots and discuss the inferences drawn from them.

---

3 Plots

3.1 Data Visualisation

First, the functions exp(x) and cos(cos(x)) are generated over the interval [-2π, 4π] and then plotted in semilog and linear scales respectively:

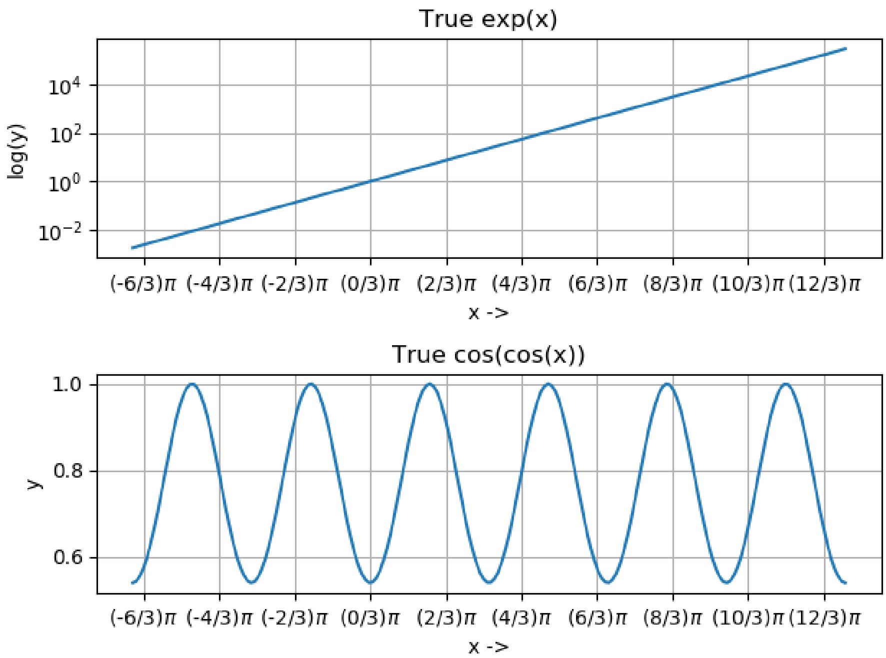
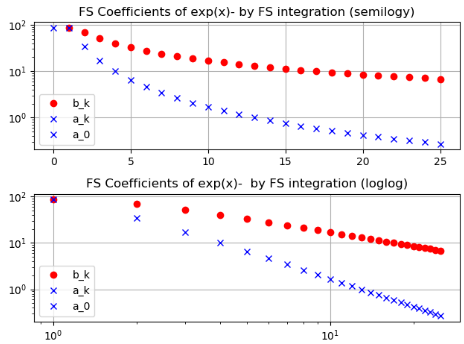


Figure 1: True functions

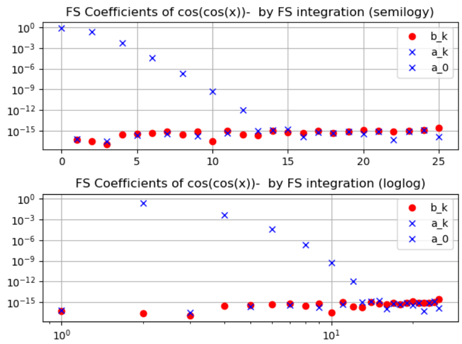
3.2 Fourier Series Coefficients

As mentioned above, the FS coefficients were obtained in two different ways:

3.2.1 scipy.integrate.quad()

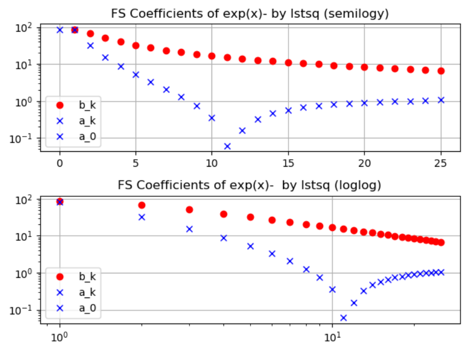


(a) FS coefficients of exp(x) in semilog and log-log scales

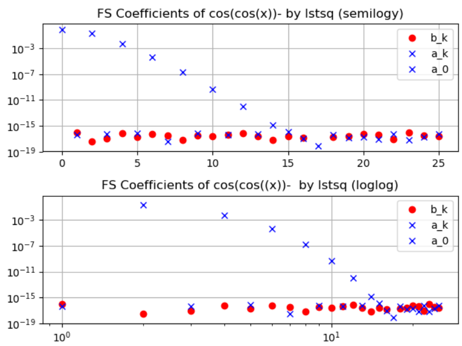


(b) FS coefficients of cos(cos(x)) in semilog and log-log scales

3.2.2 Least Squares Regression



(c) FS coefficients of exp(x) in semilog and log-log scales

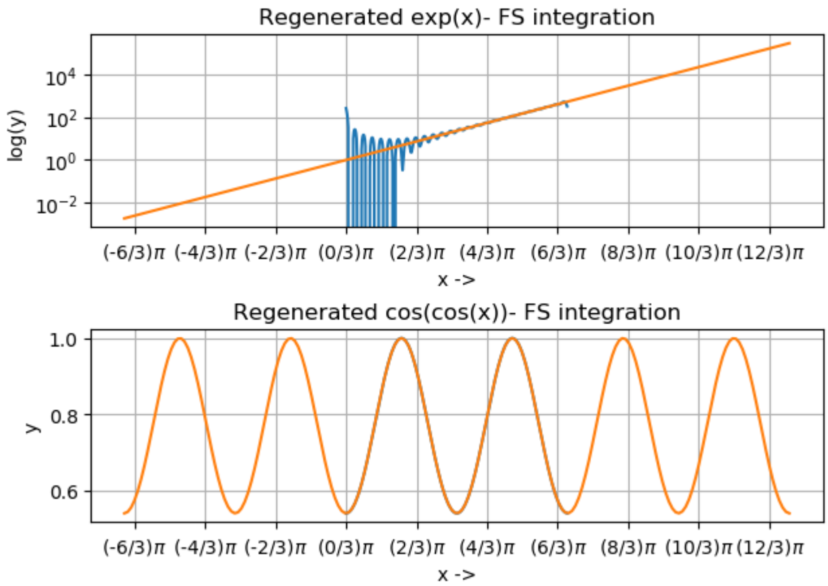


(d) FS coefficients of cos(cos(x)) in semilog and log-log scales

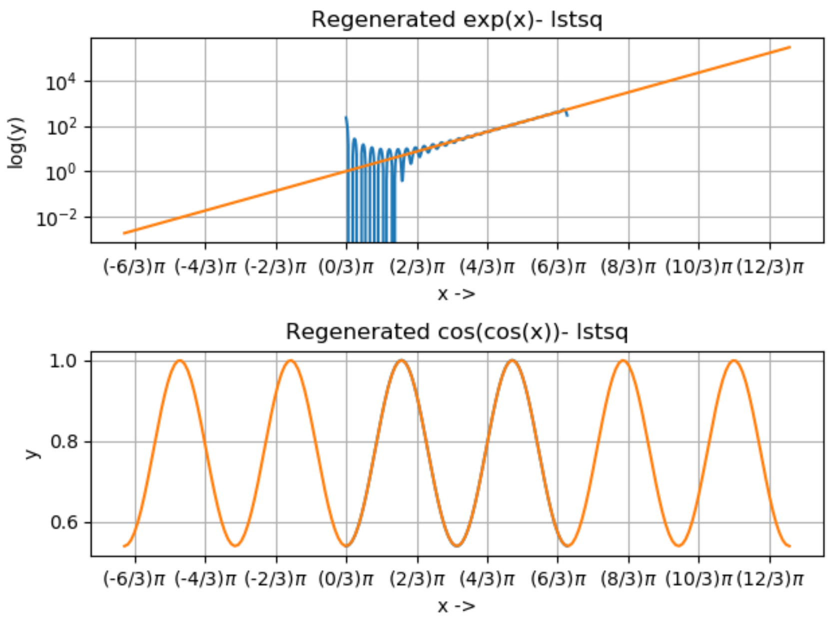
3.3 Regeneration of the functions (synthesis)

The function was regenerated back from its FS coefficients.

3.3.1 FS Coefficients obtained by numeric integration



3.3.2 FS Coefficients obtained by Least Square Regression



Note that at the current resolution the true cos(cos(x)) signal and the regenerated signal cannot be distinguished simply by looking.

---

4 Inferences

4.1 Behaviour of FS Coefficients

1. The Fourier Series coefficients of cos(cos(x)) decay rapidly. One may note that cos(cos(x)) is an even function unlike sin(x) and its nature closely resembles that of cos(kx) (k is any integer). Thus, most values of b_n is 0. That is, the contribution of sin(kx) in synthesising cos(cos(x)) is negligible in comparison to that of cos(kx).

2. Most harmonics of cos(cos(x)) correspond to low frequencies. This also explains why very low errors were obtained for the regenerated signal despite having only 25 harmonics.

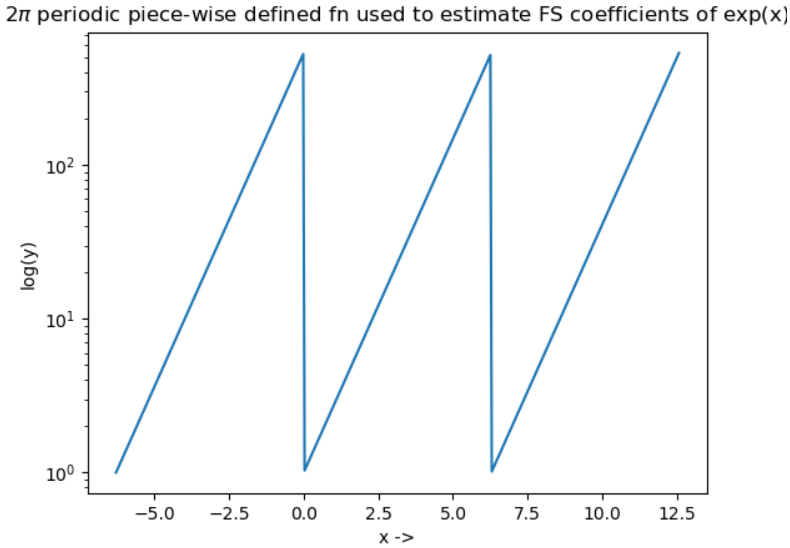
3. In comparison, the Fourier Series coefficients for exp(x) do not decay quite rapidly. In addition, the coefficients a_n and b_n seem to tend towards a non-zero value as n gets larger. This partially explains the relatively high error obtained in the regenerated estimate of exp(x).

4. Similar values and behaviour of FS coefficients were obtained for the two methods used (numeric integration and lstsq regression). It is worth noting however that the coefficients of exp(x) vary in value significantly (by 1 order of magnitude) near n = 10 for the two methods.

4.2 Behaviour of the Regenerated Functions

1. A ringing effect, similar to Gibbs phenomenon, is observed in case of the regenerated exp(x) function. Remember that the FS coefficients correspond to a periodic function with period 2π defined as exp(x) in the range $[0, 2\pi]$. Since $\exp(2\pi)$ is not equal to exp(0), the function is discontinuous at the edges of each period (0 and 2π in the fundamental period interval).

2. Discontinuities in a periodic signal cannot be perfectly represented by continuous sinusoidal functions. This gives rise to the ringing (Gibbs) phenomenon in the regenerated function.



3. Thus, the error in the FS representation of exp(x) is due to both Gibbs phenomenon and truncation of the Fourier Series. While taking more coefficients would reduce the truncation error, the ringing phenomenon would still be present.

4. As expected, the FS representation of cos(cos(x)) gives accurate results. Here, ringing phenomenon is not observed as there are no discontinuities and truncation error is negligible as the FS coefficients decay with n.

⋆ ⋆ ⋆