Week 4: Fourier Approximations

February 11, 2019

The Assignment

We will fit two functions, $\exp(x)$ and $\cos(\cos(x))$ over the interval $[0, 2\pi)$ using the fourier series

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

As you know from earlier courses, the coefficients a_n and b_n are given by

$$\begin{aligned} a_0 &= \frac{1}{2\pi}\int_0^{2\pi} f(x)dx \\ a_n &= \frac{1}{\pi}\int_0^{2\pi} f(x)\cos(nx)\,dx \\ b_n &= \frac{1}{\pi}\int_0^{2\pi} f(x)\sin(nx)\,dx \end{aligned}$$

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.

   Note: Since $\exp(x)$ grows rapidly, use semilogy for that plot.

   Note: Add grids and labels to figures.

2. Obtain the first 51 coefficients for the two functions above. **Note:** The built in integrator in Python integrates a scalar function from a to b. So you will have to compute the coefficients in a for loop. Also note that you will have to create two new functions to be integrated, namely $u(x,k) = f(x)\cos(kx)$ and $v(x,k) = f(x)\sin(kx)$. To integrate these, use the option in *quad* to pass extra arguments to the function being integrated:

   ```
   rtnval=quad(u,0,2*pi,args=(k))
   ```

   What this does is it accepts a function $u(x,\ldots)$; the integration is over x, but the k values is passed to the function by quad as the second argument. So you will have to define the two functions as having k as their second (not first) argument.

3. For each of the two functions, make two different plots using "semilogy" and "loglog" and plot the magnitude of the coefficients vs n. The values should be plotted with red circles. Note that the answer should be a vector of the form

$$\begin{pmatrix} a_0 \\ a_1 \\ b_1 \\ \ldots \\ a_{25} \\ b_{25} \end{pmatrix}$$

Note: Plot the coefficients for $f_1(x) = \exp(x)$ in Figures 3 and 4, and the coefficients for $f_2(x) = \cos(\cos x)$ in Figures 5 and 6.

(a) If you did Q1 correctly, the b_n coefficients in the second case should be nearly zero. Why does this happen?

(b) In the first case, the coefficients do not decay as quickly as the coefficients for the second case. Why not?

(c) Why does *loglog* plot in Figure 4 look linear, wheras the *semilog* plot in Figure 5 looks linear?

4. We instead do a "Least Squares approach" to the problem. Define a vector x going from 0 to 2π in 400 steps (remember *linspace*). Evaluate the function $f(x)$ at those x values and call it b. Now this function is to be approximated by Eq. (1). So for each x_i we want

$$a_0 + \sum_{n=1}^{25} a_n \cos nx_i + \sum_{n=1}^{25} b_n \sin nx_i \approx f(x_i) \tag{2}$$

Turn this into a matrix problem:

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

Create the matrix on the left side and call it A. We want to solve

$$Ac = b$$

where c are the fourier coefficients.

Note: The matrix can be constructed using zeros((400,51)) and then filling in the columns in a for loop. Do not use a double for loop. Here is the way to build up x, b and A without using nested loop:

```
x=linspace(0,2*pi,401)
x=x[:-1] # drop last term to have a proper periodic integral
b=f(x)   # f has been written to take a vector
A=zeros((400,51))      # allocate space for A
A[:,0]=1               # col 1 is all ones
for k in range(1,26):
  A[:,2*k-1]=cos(k*x) # cos(kx) column
  A[:,2*k]=sin(k*x)   # sin(kx) column
#endfor
c1=lstsq(A,b)[0]       # the '[0]' is to pull out the
# best fit vector. lstsq returns a list.
```

5. Use *lstsq* to solve this problem. You execute

```
c=lstsq(A,b)[0]
```

What this does is to find the "best fit" numbers that will satisfy Eq. (2) at exactly the points we have evaluated $f(x)$. Obtain the coefficients for both the given functions. Plot them with green circles in the corresponding plots.

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?)

7. Compute Ac 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?

8. Write a report on this assignment in latex and submit the same along with your code.