ECE 351 - Lab 8

Khoi Nguyen
https://github.com/3khoin

28 October 2021

Contents

1 Introduction

The goal of this lab was to compute the coefficients of a Fourier series and verify these calculations with Python.

2 Equations

$$x(t) = \frac{1}{2}a_0 + \sum_{k=1}^{\infty} a_k cos(k\omega_o t) + b_k sin(k\omega_o t)$$

$$\omega_0 = \frac{2\pi}{T}$$

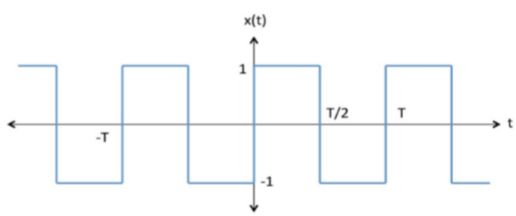
$$a_k = \frac{2}{T} * 2\int_0^{T/2} cos(k\omega_o t)\, dt = 0$$

$$b_k = \frac{2}{T} * 2\int_0^{T/2} sin(k\omega_o t)\, dt = \frac{4}{k\pi} sin^2(\frac{k\pi}{2})$$

$$x(t) = \sum_{k=1}^{\infty} \frac{4}{k\pi} sin^2(\frac{k\pi}{2}) sin(k\omega_o t) = \frac{4}{\pi}\sum_{k=1}^{\infty} \frac{1}{k} sin^2(\frac{k\pi}{2}) sin(k\omega_o t)$$

3 Methodology

The Fourier series for this lab was calculated using the following square wave function.



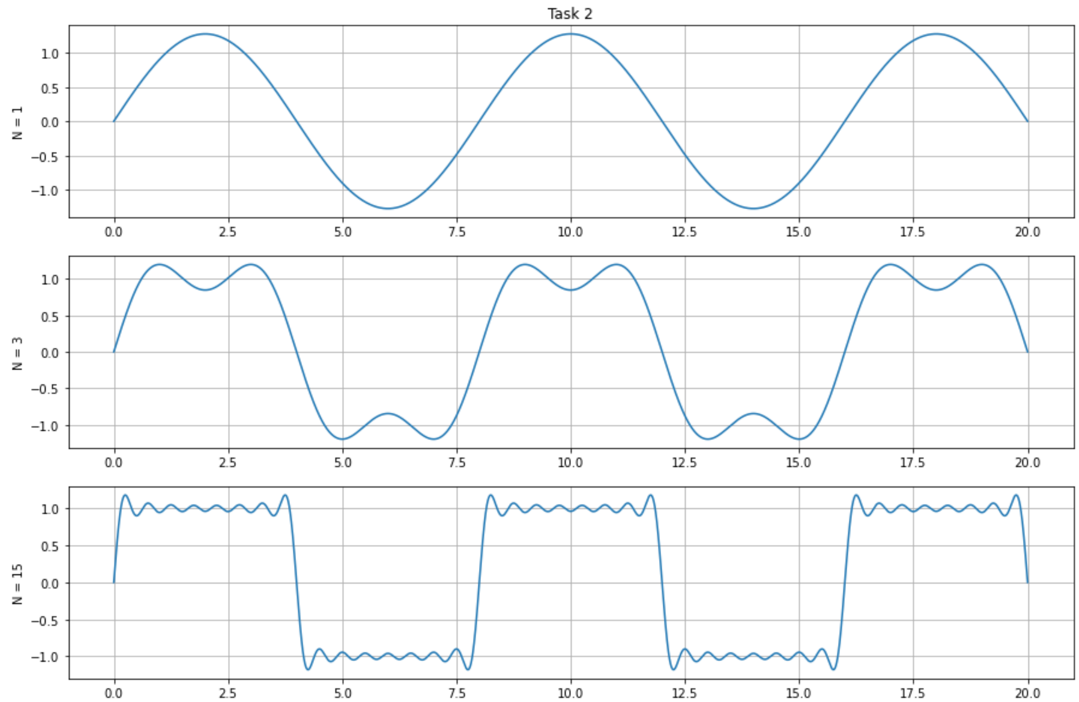
In the prelab, we calculated the expressions for a_k and b_k; we found that the a_k term reduces to 0 because the function is entirely odd, with no even terms. There was also no offset, so the a_0 term also reduced to 0. Hence, the final Fourier series was just the sum of the b_k terms.

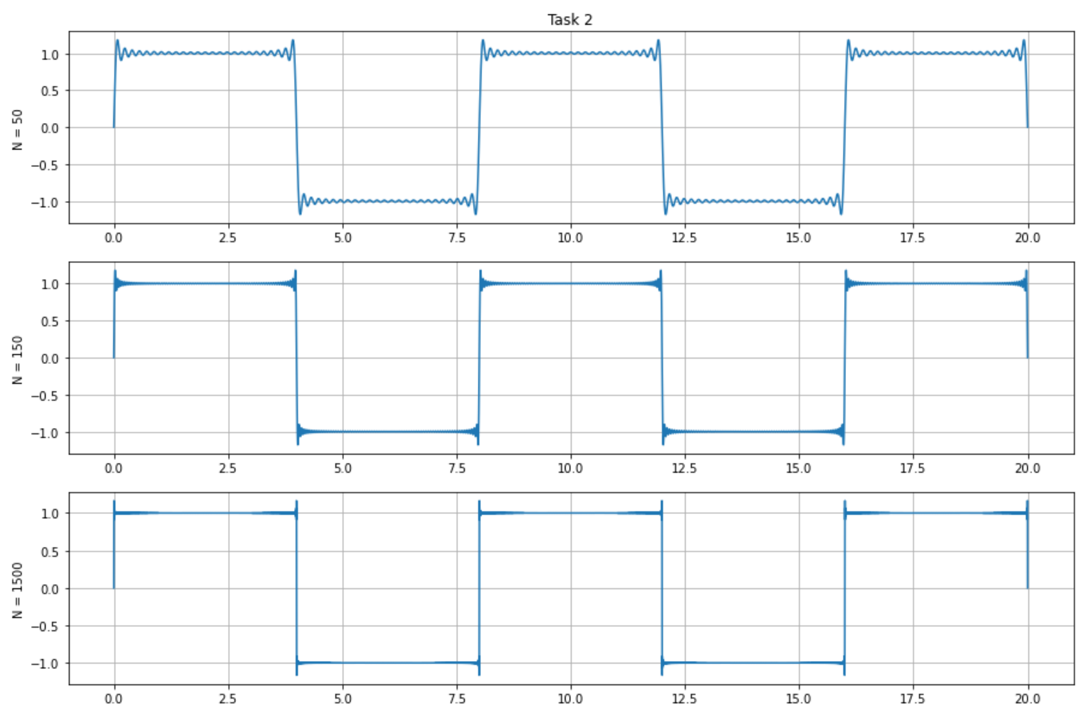
The b_k term was then inserted into Spyder and the values for b_1, b_2 b_3 were found. Calculations for a_0 and a_1 were not necessary since we know them to be 0. We then plotted the Fourier series approximation for the square wave using values of N = {1, 3, 15, 50, 150, 1500}, using T = 8 s and a domain of 0 ≤ t ≤ 20 s.

4 Results

```
b(1): 1.2732395447351628
b(2): 9.547767314442451e-33 # This approximates to 0.
b(3): 0.4244131815783876
```

Implementing a user-defined function for b_k, we printed the values of the first 3 terms. Again, a_k is known to be 0 for all terms, so no terms were printed.





The plots above represent the square wave Fourier series for N = $\{1, 3, 15, 50, 150, 1500\}$. The series was implemented using the following function:

```python
def x(t, N):
  y = 0
  for k in range(1, N + 1):
    y += b(k) * np.sin(k * 2 * np.pi * t/ T)
  return y
```

The function returns the sum of terms of $x_k(t)$ all the way up to N.

5 Error Analysis

No particular errors or matters of concern were brought up during the lab.

6 Questions

1. x(t) is an odd function. Examining x(t), we can see that it consists of the b_k term, $\frac{4}{k\pi}sin^2(\frac{k\pi}{2})$, and the sine term $sin(k\omega_o t)$. The first term is a squared function, and is even; x(-t) = x(t). The sine function is odd; x(-t) = -x(t). An odd function multiplied by an even function will be odd; hence, x(t) is odd.

2. As was demonstrated in the prelab, a_k will be 0 for all values of k, since the function is not even, so there is no relevant $a_k cos(k\omega ot)$ term.

3. The square wave Fourier series approximation grows closer and closer to being shaped like an actual square wave as N increases. One limitation of the approximation can be seen at the edges of the rectangular waveform; before entering and exiting the approximated top edge of the waveform, a sudden jump must be made.

4. As N increases, the additional coefficient terms being added are increasing in frequency while decreasing in magnitude, which translates graphically to a flatter waveform.

5. The clarity of the tasks, expectations, and lab deliverables was exceptionally clear for this lab.

7 Conclusion

In this lab, although no new tools in particular was explored in Python, we learned to use Python to assist in the calculation of the coefficients in a Fourier series and plot the ensuing approximations. Although the Fourier series does have limitations, it is very close in its ability to replicate just about any periodic waveform.