Part 1: Fourier Analysis of a Square Wave

Thursday, 12 January 2023 09:27

Monday 9th of January

The aim of Today's lab session was to get started with Task 1, and ensure it is finished before the next lab session.

Task 1.1

Below is the proof for the amplitude-phase form of the Fourier series which will be used in this experiment.

GIVEN A FUNCTION $T(t)$ WITH PERIOD τ:

$$T(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty}\left[a_n \cos\left(\frac{2\pi n}{\tau} t\right) + b_n \sin\left(\frac{2\pi n}{\tau} t\right)\right] \quad (EQ.1)$$

WHERE:

$$a_n = \frac{2}{\tau}\int_0^{\tau} T(t)\cos\left(\frac{2\pi n}{\tau} t\right) dt$$

$$b_n = \frac{2}{\tau}\int_0^{\tau} T(t)\sin\left(\frac{2\pi n}{\tau} t\right) dt$$

WE CAN WRITE (EQ1) IN AMPLITUDE PHASE FORM

$$T(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \beta_n \sin\left(\frac{2\pi n}{\tau} t - \Delta\phi_n\right) \quad (EQ.2)$$

TO FIND β_n AND $\Delta\phi_n$ WE EQUATE (EQ.1) WITH (EQ.2)

$$\beta_n \sin\left(\frac{2\pi n t}{\tau} - \Delta\phi_n\right) = a_n \sin\left(\frac{2\pi n t}{\tau}\right) + b_n \sin\left(\frac{2\pi n t}{\tau}\right)$$

$$\beta_n \cos(\Delta\phi_n)\sin\left(\frac{2\pi n t}{\tau}\right) - \beta_n \sin(\Delta\phi_n)\sin\left(\frac{2\pi n t}{\tau}\right) = a_n \cos\left(\frac{2\pi n t}{\tau}\right) + b_n \sin\left(\frac{2\pi n t}{\tau}\right)$$

EQUATING COEFFICIENTS, WE GET:

$$\beta_n \sin(\Delta\phi_n) = a_n$$

$$\beta_n \cos(\Delta\phi_n) = b_n$$

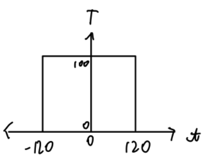
$$\therefore \boxed{\begin{aligned}\beta_n &= \sqrt{a_n^2 + b_n^2} \\ \Delta\phi_n &= -\arctan\left(\frac{a_n}{b_n}\right)\end{aligned}}$$

Task 1.2

Below is the pen and paper calculation for the Fourier Series of a square wave, up to n=3.

SQUARE FUNCTION PERIOD: $\tau = 240$

AMPLITUDE: 50 $(0 \to 100)$



TAKE RANGE $0 < t < 240$

$$T(t) = \begin{cases} 100 & 0 \le t \le 120 \\ 0 & \text{OTHERWISE} \end{cases}$$

$T(t)$ IS AN ODD FUNCTION

$\therefore$ INTEGRATES TO 0 OVER EVEN INTERVAL

$\therefore a_n = 0 \ \forall n$

$$b_n = \frac{2}{\tau}\int_0^{\tau} T(\tau)\sin\left(\frac{2\pi n}{\tau} t\right) dt$$

$$= \frac{2}{\tau}\int_0^{\tau/2} 100 \sin\left(\frac{2\pi n}{\tau} t\right) dt$$

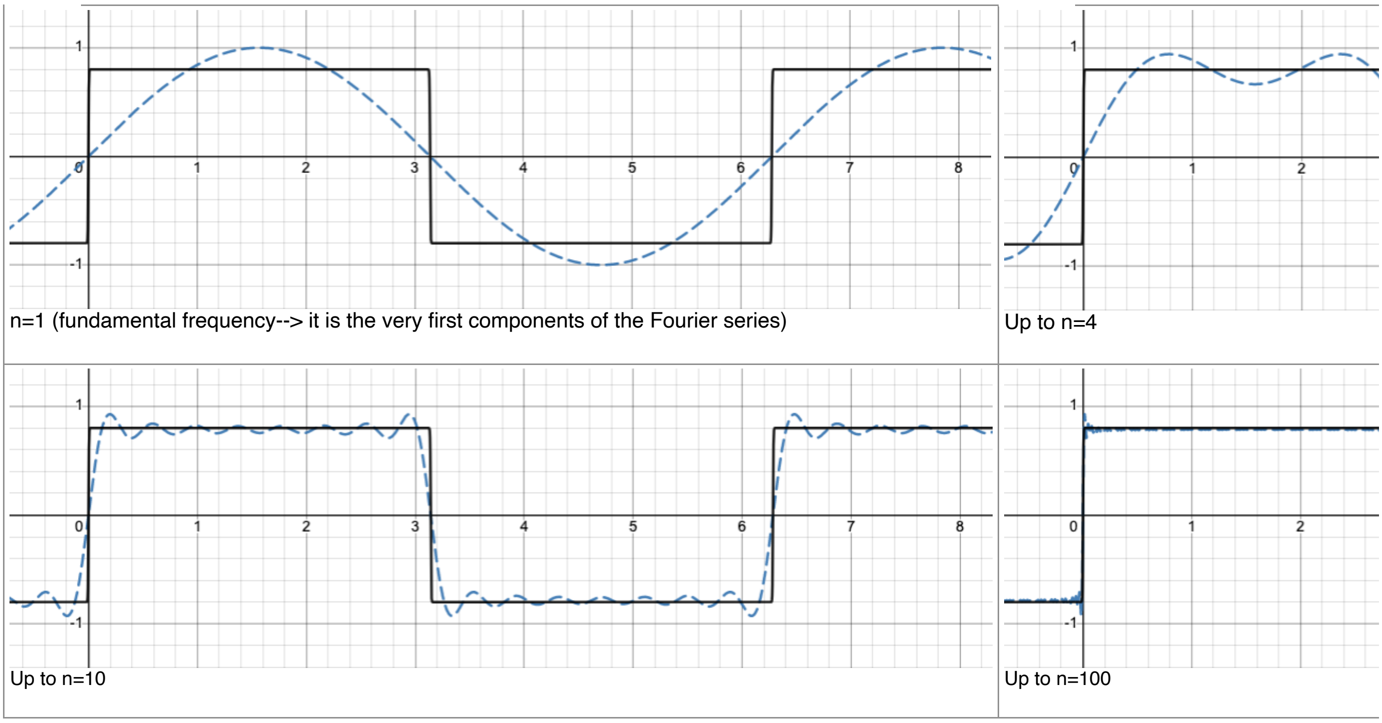
$$= \frac{200}{\tau}\cdot\frac{\tau}{2\pi n}\left[-\cos\left(\frac{2\pi n t}{\tau}\right)\right]_0^{\tau/2}$$

$b_n = \frac{200}{\pi n} \ \forall$ ODD n

$b_n = 0 \ \forall$ EVEN n

$$\therefore \boxed{T(t) \sim 50 + \frac{200}{\pi}\sin\left(\frac{2\pi}{240} t\right) + \frac{200}{3\pi}\sin\left(\frac{4\pi}{240} t\right)} \quad \text{UP TO } n=3$$

Plotting the Fourier series of a square wave up to different values of n gives the following graphs:



n=1 (fundamental frequency--> it is the very first components of the Fourier series)

Up to n=4

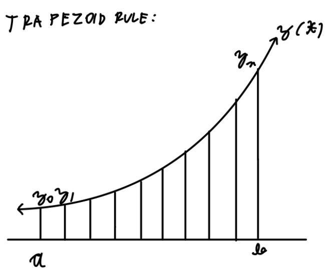
Up to n=10

Up to n=100

As it can clearly be seen above, the Fourier Series becomes an increasingly better fit for the square wave as we go up to higher n, and going up to n=100 models the square wave very well. However, in this case the square wave is a continuous function and its Fourier coefficients can be easily calculated. However, we will be dealing with data sets that are not continuous functions, and hence it is more appropriate to use numerical methods, which the next part will discuss.

Task 1.3

We have set up a function which allows us to calculate the area under a set of discrete data points, as if they were on a continuous function. This is done using the trapezoid rule, as shown below:



$$\boxed{\text{AREA} = \int_a^b y\,dx \approx \frac{1}{2}h\left[y_0 + 2(y_1 + y_2 + \dots + y_{n-1}) + y_n\right]}$$

WHERE $h = \frac{b-a}{n}$

Using this rule in Python we have used numerical integration for both the sine and the semi circle functions and have obtained the following results:

Sine data

Analytical Value	0
Low-Resolution Data Numerical Integration	$-6.72*10^{-16}$
High-Resolution Data Numerical Integration	$2.33*10^{-16}$

Circle data

Analytical Value	1.57
Low-Resolution Data Numerical Integration	1.49
High-Resolution Data Numerical Integration	1.56

As it can be seen from the tables above, the numerical integration algorithm gave us results that were very close to the analytical value calculated, indicating that the algorithm is valid. Furthermore, the high-resolution data yielded better results when numerical integration was ran on it, as distances between points come smaller and approach infinitesimal distances, hence approaching exact integration. As such, we will attempt to use high-resolution data when performing numerical integration in our experiment.