Assignment on Fourier Series Approximation

Overview

In this offline, you will need to implement a Fourier Series approximation of a periodic function from scratch in Python. The assignment is divided into modular steps where each component of the Fourier Series calculation is isolated for easier implementation and comprehension.

Background

The Fourier Series is a way to represent a periodic function $f(x)$ as an infinite sum of sines and cosines:

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left(a_n \cos\left(\frac{n\pi x}{L}\right) + b_n \sin\left(\frac{n\pi x}{L}\right) \right)$$

where:

- L is half the period of the function.
- a_0 is the average value (DC component) of the function over one period.
- a_n and b_n are Fourier coefficients that describe the contribution of cosine and sine terms, respectively, at the n-th harmonic.

The Fourier coefficients a_0, a_n, and b_n are calculated as follows:

$$a_0 = \frac{1}{2L} \int_{-L}^{L} f(x)\, dx$$

$$a_n = \frac{1}{L} \int_{-L}^{L} f(x) \cos\left(\frac{n\pi x}{L}\right) dx$$

$$b_n = \frac{1}{L} \int_{-L}^{L} f(x) \sin\left(\frac{n\pi x}{L}\right) dx$$

Problem Description

You are tasked with implementing a Python class, `FourierSeries`, that calculates and plots the Fourier Series approximation of a given periodic function. The class must be modular, and each method should handle a specific part of the Fourier Series computation.

Requirements

1. Implement a method `calculate_a0()` to compute a_0 using numerical integration.
2. Implement a method `calculate_an()` to compute a_n for the n-th cosine term.
3. Implement a method `calculate_bn()` to compute b_n for the n-th sine term.
4. Implement a method `approximate()` that combines the above coefficients to approximate $f(x)$.
5. Implement a `plot()` method to visualize the original function and its Fourier Series approximation over one period.
6. Define several periodic functions as target functions (square wave, sawtooth, triangle, sine, and cosine) for the Fourier approximation.

Code Skeleton

A Python skeleton code is provided to guide the implementation. Some sections are already implemented, while others require you to add code.

Instructions

- Use numerical integration (e.g., `np.trapz`) to compute the integrals for a_0, a_n, and b_n. - Implement each function in the skeleton code provided in the corresponding placeholders. - Use `target_function(x, function_type)` to define different types of periodic functions. Test the Fourier series approximation for each type. - Ensure proper labeling in your plots.

Mark Distribution

The marks for this assignment are distributed as follows:

Submission

Submit a single Python file containing the completed class and all necessary function implementations. Rename it with your student id and then submit the python file.

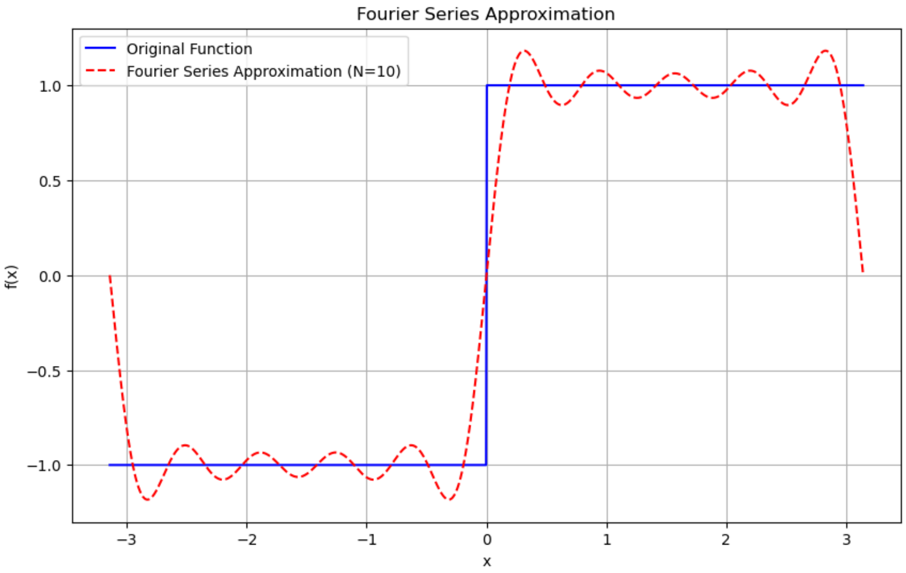

Figure 1: Fourier Series Approximation Example

Component	**Points**
Initialization	5 points
Method `calculate_a0()`	15 points
Method `calculate_an()`	15 points
Method `calculate_bn()`	15 points
Method `approximate()`	20 points
Method `plot()`	10 points
Target Function Implementation	20 points
Total	**100 points**

Table 1: Mark Distribution for Fourier Series Assignment