MATH 101A — ASSIGNMENT 3

Learning goals

- Apply the Trapezoidal Rule to functions defined by data.
- Witness the basic ideas of Fourier analysis.

Contributors

On the first page of your submission, list the student numbers and full names (with the last name in **bold***) of all team members. Indicate members who have not contributed using the comment "(non-contributing)".*

Assignment questions: Spectral Messages

The questions in this section contribute to your assignment grade*. Stars indicate the difficulty of the questions, as described on Canvas.*

Before starting work on this assignment, please read through all the questions and take note of the special instructions in the section headed "Notes and Comments" below.

Overview. Figure 1 below shows the graph of the function:

$$f(x) = 72\sin(x) + 101\sin(2x) + 108\sin(3x) + 108\sin(4x) + 111\sin(5x).$$

The coefficients $(72, 101, 108, 108, 111)$ are the numbers for the characters ("H","e","l","l","o") used in

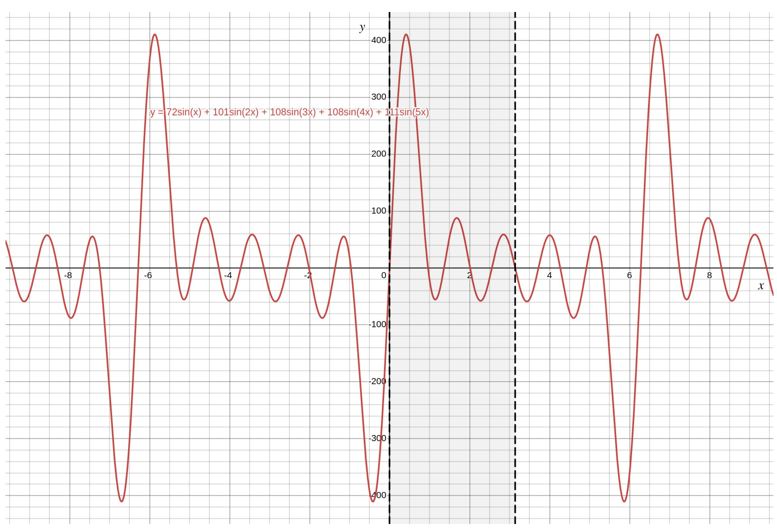


Figure 1: This function says "Hello"

computer systems worldwide (UTF-8). This assignment will guide you through the integrations needed to reveal short messages encoded in functions like this one, using only the information in their graphs.

Notation. The five sinusoids combined to produce Figure 1 are plotted in Figure 2. Each one is odd and 2π-periodic, and these two properties are inherited by the function in Figure 1. For this reason, that entire graph can be reconstructed from the shaded segment, where $0 \le x \le \pi$. We will focus on just this interval in all that follows. (This explains the choice of axes in Figure 2.)
The sinusoids used in Figures 1 and 2 are the first five members of the infinite family

$$y_n(x) = \sin(nx), \qquad n = 1,2,3,.... \tag{1}$$

This assignment concerns various functions of the form exemplified above:

$$f(x) = b_1 \sin(x) + b_2 \sin(2x) + \ldots = \sum_{k=1}^{\infty} b_k \sin(kx) \; . \tag{2}$$

Each of these is called a *Fourier Sine Series*; the real-valued constants $b_1, b_2, \ldots$ are called the *coefficients*. Later in MATH 101 we will develop a general theory for "infinite sums" (officially, *series*). For this assignment, the informal and intuitive approach of treating them just like ordinary finite sums will suffice.

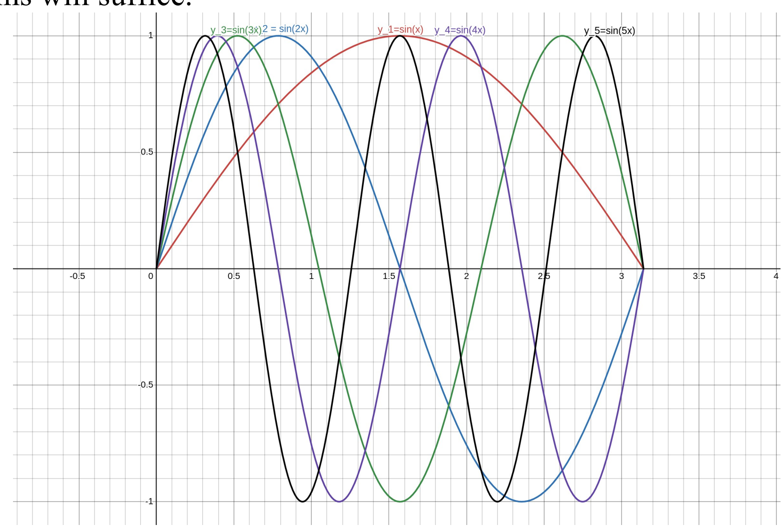


Figure 2: Basis functions $y_n = \sin(nx)$ for $n = 1,2,3,4,5$

2. Use the definitions in line (1) to complete the following.
   (a) (1 mark) ★☆☆☆ For arbitrary constants K, D, m, n, calculate:

$$\frac{d}{dx}\Big[K \sin(mx)\cos(nx) + D\cos(mx)\sin(nx)\Big].$$

Then use wise choices of K and D to find the following indefinite integral, assuming $m^2 \neq n^2$:

$$\int \sin(mx)\sin(nx)dx.$$

Solution:

Using product and chain rules we can calculate the following,

$$K[mcos(mx)\cos(nx) - nsin(mx)\sin(nx)] + D[ncos(mx)\cos(nx) - msin(mx)\sin(nx)]$$

Simplifying we get,

$$\cos(mx)\cos\,(nx)(Km + Dn) - sin(mx)\sin(nx)\,(Kn + Dm).$$

Setting $Kn + Dm$ = -1 and $Km + Dn$ = 0, we can equate the indefinite integral to the derivative.

$$\text{K} = \frac{-1-Dm}{n} = \frac{-Dn}{m},$$

$$m(1 + Dm) = Dn^2,$$

$$D(m^2 - n^2) = -m,$$

$$D = \frac{-m}{m^2-n^2}.$$

$$K = \frac{m}{m^2-n^2} \cdot \frac{n}{m},$$

$$K = \frac{n}{m^2-n^2}.$$

So,

$$\int \sin(mx)\sin(nx)dx = \frac{n}{m^2-n^2}\sin(mx)\cos(nx) - \frac{m}{m^2-n^2}\cos(mx)\sin\,(nx).$$

(b) (1 mark) ★☆☆☆ Determine the constant R that makes the following equation valid for all positive integers m and n. Explain why the stated equation holds.

$$\int_0^\pi \sin(mx)\sin(nx)\,dx = \begin{cases} 0, & \text{if } m \neq n, \\ R, & \text{if } m = n. \end{cases}$$

Solution:

Case 1: if $m \neq n$

It's already stated that for $m \neq n$, the integral is 0.

Case 2: if $m = n$

For $m = n$ we need to calculate the integral

$$\int_0^\pi sin^2(mx)dx$$

$$= \int_0^\pi \frac{1-\cos(2mx)}{2}\,dx,$$

$$= \frac{1}{2}\int_0^\pi (1-\cos(2mx))dx,$$

$$R = \frac{\pi}{2}$$

For any given function $f(x)$ integrable on $[0, \pi]$, we now use the constant R found above to define

$$B_k(f) = \frac{1}{R}\int_0^\pi f(x)\sin(kx)\,dx, \qquad k = 1, 2, 3, \ldots. \tag{3}$$

(c) (1 mark) ★☆☆☆ Consider the specific function plotted in Figure 1, namely,

$$f(x) = 72\sin(x) + 101\sin(2x) + 108\sin(3x) + 108\sin(4x) + 111\sin(5x).$$

Calculate $B_n(f)$ for each integer $n \geq 1$.

Solution:

$$B_k(f) = \frac{2}{\pi}\int_0^\pi f(x)\sin(kx)\,dx, \qquad k = 1,2,3, \ldots.$$

Given the properties we went over before when calculating $B_k(f)$ we can use the property that the integral of $\sin(mx)\sin(nx)$ over 0 to π is 0 unless $m = n$, in which case it is $R = \frac{\pi}{2}$.

$$B_1(f) = \frac{2}{\pi}\int_0^\pi 72sin^2xdx = \frac{2}{\pi} * 72 * \frac{\pi}{2} = 72.$$

$$B_2(f) = \frac{2}{\pi}\int_0^\pi 101sin^2 2xdx = \frac{2}{\pi} * 101 * \frac{\pi}{2} = 101.$$

$$B_3(f) = \frac{2}{\pi}\int_0^\pi 108sin^2 3xdx = \frac{2}{\pi} * 108 * \frac{\pi}{2} = 108.$$

$$B_4(f) = \frac{2}{\pi}\int_0^\pi 108sin^2 4xdx = \frac{2}{\pi} * 108 * \frac{\pi}{2} = 108.$$

$$B_5(f) = \frac{2}{\pi}\int_0^\pi 111sin^2 5xdx = \frac{2}{\pi} * 111 * \frac{\pi}{2} = 111.$$

(d) (1 mark) ★☆☆☆ Now think about a general function f with the form shown below, where each b_k is a constant:

$$f(x) = b_1 \sin(x) + b_2 \sin(2x) + \ldots = \sum_{k=1}^{\infty} b_k \sin(kx).$$

Find a formula that expresses b_n in terms of one or more of the numbers $B_k(f)$ defined in line (3). Your result should be valid for every integer $n \geq 1$. Explain why your formula holds.

Solution:

In 2c, it is clear that each coefficient b_k is equal to $B_k(f)$ for k = 1,2,3,…, thus,

$$b_n = B_n(f).$$

This result holds for every integer n≥1 due to the orthogonality of the sine functions over the specified interval. This is a crucial aspect of Fourier analysis, allowing us to decompose a function into its sine components and directly recover the coefficients that define its behavior on the interval [0,π]. Essentially this function we're effectively filtering out all components of $f(x)$ except the one that matches the frequency kx.

For smooth periodic functions like the one graphed in Figure 1, the integrals shown in line (3) are excellent candidates for approximate evaluation by the Trapezoidal Rule. Specialized analysis reveals that the actual difference between the exact value and its Trapezoidal approximation is much, much smaller than the standard estimate shown in class suggests. So, we choose the Trapezoidal Rule for all our integral approximations below.

3. Imagine dividing $[0,\pi]$ into $N = 8$ subintervals, with endpoints

$$x_0 = 0,\ x_1 = \frac{\pi}{8},\ \ldots,\ x_i = \frac{i\pi}{8},\ \ldots,\ x_8 = \frac{8\pi}{8},$$

and evaluating the numbers $f_i = f(x_i)$ for $i = 0, 1, \ldots, 8$. For $f(x)$ as shown in Figure 1, the 9 numbers f_i, in order, are approximately

0.0, 409.3, 149.8, −53.9, 75.0, 19.3, −52.2, 50.5, 0.0.

(a) (1 mark) Use all of these values to calculate Trapezoidal-rule approximations for the integrals $B(f),\ldots,B_5(f)$. (Call these $B_n^{\text{trap}}(f)$.) Report each answer with 3 decimal places.

Solution:

mathassignment3.m

```matlab
% Define the provided function values at x_i
f = [0.0, 409.3, 149.8, -53.9, 75.0, 19.3, -52.2, 50.5, 0.0];

% Number of subintervals
N = 8;

% Delta x value, which is pi/N
deltaX = pi/N;

% Pre-allocate Bk(f) array for k = 1 to 5
Bk = zeros(1, 5);

% Loop over k = 1 to 5 to calculate each B_k(f)
for k = 1:5
    % Initialize the sum for the trapezoidal rule
    trapezoidalSum = 0;

    % Calculate the trapezoidal sum, skipping the first and last terms
    % since they are multiplied by 0 due to the sine function
    for i = 1:N-1
        x_i = i * deltaX;
        trapezoidalSum = trapezoidalSum + f(i+1) * sin(k*x_i);
    end

    % Multiply by 2 for the middle terms and add the first and last terms
    % directly since f_0 and f_N are part of the provided f array
    trapezoidalSum = 2 * trapezoidalSum + f(1) * sin(k*0) + f(N+1) * sin(k*pi)

    % Apply the trapezoidal rule formula
    Bk(k) = (2/pi) * (deltaX/2) * trapezoidalSum;
end

% Display the results
disp('Bk(f) values for k = 1 to 5:');
disp(Bk);
```

Command Window

```
>> mathassignment3
Bk(f) values for k = 1 to 5:
   72.0013  100.9874  108.0136  108.0000  111.0068
```

f-array: Contains the provided function values at each x_i

deltaX: Calculated as $\frac{\pi}{N}$ since the interval $[0, \pi]$ is divided into N subintervals.

Bk-array: Stores the calculated $B_k(f)$ values for each k

Trapezoidal Sum Calculation: For each k, we calculate the trapezoidal sum by iterating through each x_i.

From this we get,

$$
\begin{aligned}
B_1(f) &= 72.001,\\
B_2(f) &= 100.987,\\
B_3(f) &= 108.014,\\
B_4(f) &= 108.000,\\
B_5(f) &= 111.007.
\end{aligned}
$$

(b) (1 mark) Report, to 3 significant digits, the 5 numbers $B_n^{\text{trap}}(f) - B_n(f)$, $n = 1, \ldots, 5$. Your writeup for this problem should describe your method for computing the numbers $B_n^{\text{trap}}(f)$ in enough detail that a patient reader could reproduce your work. Show documentation appropriate for whatever approach you used: a code listing (any language is acceptable), a spreadsheet sample, or selected screenshots, etc.

Solution:

```matlab
% Define the provided function values at x_i
f = [0.0, 409.3, 149.8, -53.9, 75.0, 19.3, -52.2, 50.5, 0.0];

% Number of subintervals
N = 8;

% Delta x value, which is pi/N
deltaX = pi/N;

% Pre-allocate Bk(f) array for k = 1 to 5
Bk = zeros(1, 5);

% Loop over k = 1 to 5 to calculate each B_k(f)
for k = 1:5
    % Initialize the sum for the trapezoidal rule
    trapezoidalSum = 0;

    % Calculate the trapezoidal sum
    for i = 1:N-1
        x_i = i * deltaX;
        trapezoidalSum = trapezoidalSum + f(i+1) * sin(k*x_i);
    end

    % Multiply by 2 for the middle terms and add the first and last terms
    trapezoidalSum = 2 * trapezoidalSum + f(1) * sin(k*0) + f(N+1) * sin(k*p

    % Apply the trapezoidal rule formula
    Bk(k) = (2/pi) * (deltaX/2) * trapezoidalSum;
end

% Exact Bn(f) values for n = 1 to 5
% Update this array with the exact Bn(f) values for your specific function
B_exact = [72, 101, 108, 108, 111];

% Calculate the differences Bn_trap(f) - Bn(f) for n = 1...5
B_diff = Bk - B_exact;

% Display the Bk(f) values
disp('Bk(f) values for k = 1 to 5 (Trapezoidal Rule Approximations):');
disp(Bk);

% Display the exact Bn(f) values
disp('Exact Bn(f) values for n = 1 to 5:');
disp(B_exact);
```

Command Window

```
>> mathassignment31
Bk(f) values for k = 1 to 5 (Trapezoidal Rule Approximations):
   72.0013   100.9874   108.0136   108.0000   111.0068

Exact Bn(f) values for n = 1 to 5:
    72   101   108   108   111

Differences Bn_trap(f) - Bn(f) for n = 1 to 5:
    0.0013   -0.0126    0.0136         0    0.0068
```

f-array: Contains the provided function values at each x_i

deltaX: Calculated as $\frac{\pi}{N}$ since the interval $[0, \pi]$ is divided into N subintervals.

Bk-array: Stores the calculated trapezoidal rule approximation $B_k(f)$ values for each k= 1 to 5

B_exact array: Contains the exact values of $B_n(f)$

B_diff calculation: Computes the difference between the trapezoidal rule approximations and the exact values for each k

From this we get,

$$B_1^{trap}(f) - B_1(f) = 0.00131,$$
$$B_2^{trap}(f) - B_2(f) = -0.0126,$$
$$B_3^{trap}(f) - B_3(f) = 0.0136,$$
$$B_4^{trap}(f) - B_4(f) = 0.000,$$
$$B_5^{trap}(f) - B_5(f) = 0.00676.$$

The next question applies the ideas from Question 3 at scale. It involves rather large numbers N and R, so computer assistance in evaluating the individual Trapezoidal-Rule approximations will be essential. You will need to apply the Trapezoidal Rule to $R+1$ different integrals, so a systematic computer-assisted approach is recommended also for this.

4. A special message for the students in your small class section has been UTF-8 encoded in a list of positive integers $b_1, b_2, \ldots, b_R$. These numbers provide the coefficients that define a new function

$$f(x) = b_1 \sin(x) + b_2 \sin(2x) + \ldots + b_R \sin(Rx).$$

(a) Acquire the list of numbers f_i, $i = 0,\ldots,N$, specific to your small class. These are available on Canvas, in an appropriately-named CSV file. The given numbers are approximate values $f_i \approx f(x_i)$, where $x_i = \frac{i\pi}{N},\ i = 0, 1, 2, \ldots, N$. (There is nothing to hand in for this part.)

(b) (1 mark) Determine and report your specific value of N. Then produce a computer-generated plot of the curve $y = f(x)$ on the interval $0 \le x \le \pi$.

Solution:

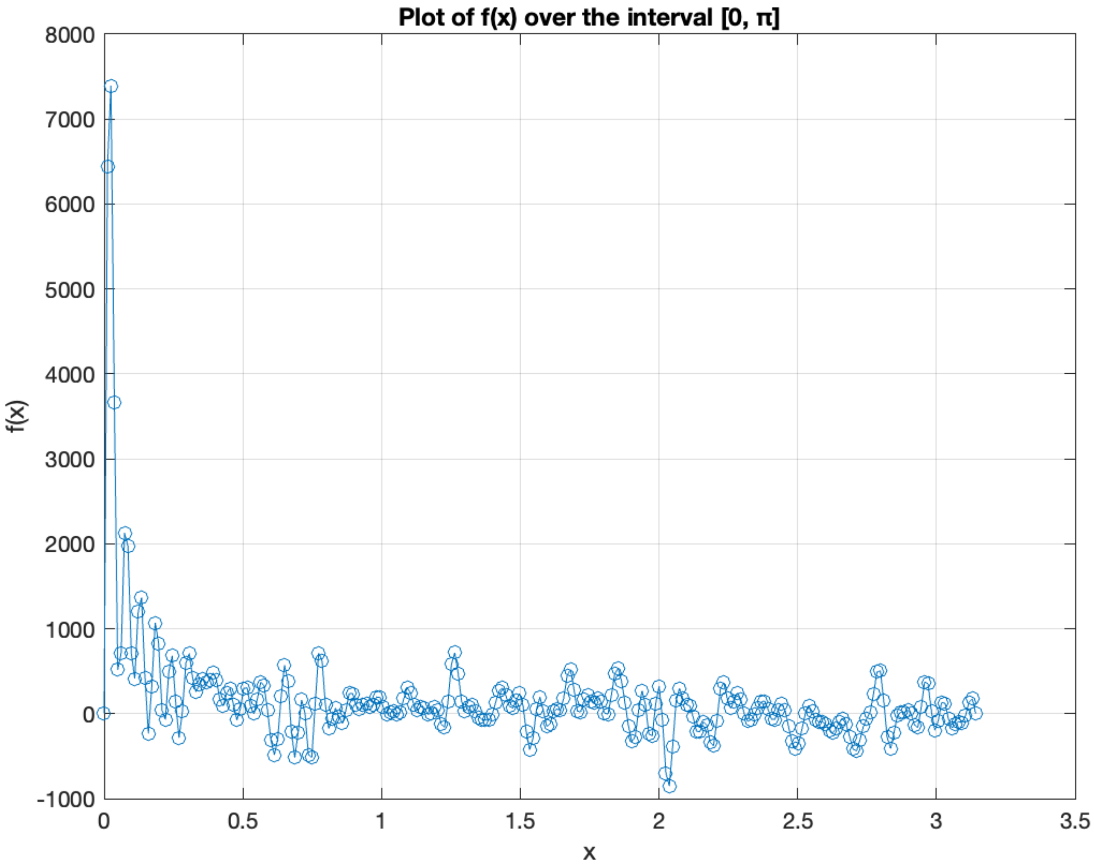
```
% Assuming A42 is a table and the column of interest is the first (and only) c
% Extract the column as a vector
f_i = A42.Variables; % For generic table access
% Or if the column has a name, for example, 'Values', you can use A42.Values

% Determine N based on the length of the extracted column
N = length(f_i) - 1;

% Generate x_i values from 0 to pi, with N+1 points
x_i = linspace(0, pi, N+1);

% Plot f(x) using the x_i and the extracted f_i values
figure; % Create a new figure window
plot(x_i, f_i, '-o'); % Use the extracted f_i for plotting
title('Plot of f(x) over the interval [0, π]');
xlabel('x');
ylabel('f(x)');
grid on; % Enable grid for better readability

```



In our case the specific value of N was 256

Size (A42, 1): This gets the number of rows in 'A42', which is $N + 1$. Subtracting 1 gives us N.

linspace(0, pi, $N + 1$): This generates $N + 1$ linearly spaced points between 0 and π, corresponding to our x_i values.

Plot(x_i, A42, '-o'): This plots the $f(x)$ values stored in A42 against the generated x_i values. The -o option creates a line plot with circle markers at the data points, making it clear where each f_i value is positioned.

(c) (5 marks) Use the Trapezoidal Rule to find the number R and the coefficients $b_1,\ldots,b_R,b_{R+1}$. To recognize R, note that $b_1,\ldots,b_R$ are all positive, whereas $b_{R+1} = 0$. Report the coefficients $b_1,\ldots,b_R$ as a comma-separated list of positive integers.
Just as in Question 3, your submission should describe your methods in enough detail that a patient reader could reproduce your work.

Solution:

For the next question we switched from MATLAB to python using pandas and numpy instead as they were able to process the array operations more seamlessly than MATLAB. The following is our python script used to calculate the coefficients.

```python
import pandas as pd
import numpy as np

# Load the data from the CSV file.
data_frame = pd.read_csv('/Users/randyren/Downloads/A42.csv', header=None)
flat_data = data_frame.to_numpy().ravel()

# Calculate the number of points.
num_points = len(flat_data) - 1
# Generate x values between 0 and pi, inclusive.
x_vals = np.linspace(0, np.pi, num_points + 1)

def calculate_integral_approximations(data, x_points, iterations):
    """Calculate the integral approximations using the trapezoidal rule."""
    coefficients = []
    for i in range(1, iterations + 1):
        # Calculate sine values for current iteration.
        sine_vals = np.sin(i * x_points)
        # Apply the trapezoidal rule.
        integral = np.pi / num_points * np.sum((data[:-1] * sine_vals[:-1] + data[1:] * sine_vals[1:]) / 2)
        # Adjust the result according to the given formula and round it.
        coefficient = round((2 / np.pi) * integral)
        coefficients.append(coefficient)
    return coefficients

# Compute b values using the modified function.
b_vals = calculate_integral_approximations(flat_data, x_vals, num_points)

# Write b values to a text file.
with open('b_values_modified.txt', 'w') as file:
    for value in b_vals:
        file.write(f'{value} ')
```

To break this down into its most essential components:

pandas, which is extensively used for data manipulation and analysis. It offers data structures and operations for manipulating numerical tables and time series.
numpy, which is used for a wide variety of mathematical operations on arrays. It provides a high-performance multidimensional array object, and tools for working with these arrays.

data_frame = pd.read_csv('/Users/randyren/Downloads/A42.csv', header=None)
flat_data = data_frame.to_numpy().ravel()

The data is loaded from a CSV file into a pandas DataFrame. This file contains the function values f(x) which we want to integrate. We then convert this DataFrame into a numpy array and flatten it, as we are interested in a one-dimensional array of function values.

This function computes the trapezoidal rule approximation of the integral

$$\int_0^{\pi} f(x)\sin(nx)\,dx \text{ for a given } n.$$

sine_values contains $\sin(nx)$ for each x in x_values.

The integral approximation is calculated by averaging adjacent $f(x)$ and $\sin(nx)$ products, multiplying by the width of each subinterval $\frac{\pi}{N}$ and summing these across the interval.
The final approximation is scaled by $\frac{2}{\pi}$ and rounded to the nearest integer, as per the specific requirements of the problem.

Next, we can compute and write b_n values to a file using the following code:
b_vals = calculate_integral_approximations(flat_data, x_vals, num_points)

This list comprehension applies the trapezoidal_rule function for each n from 1 to N, copmuting the coefficients b_n for the Fourier sine series approximation of $f(x)$. Next the following code writes these values to a text file named b_values.txt:

```
with open('b_values_modified.txt', 'w') as file:
    for value in b_vals:
        file.write(f'{value} ')
```

Finally, this gives us the following numbers:

78, 111, 32, 97, 109, 111, 117, 110, 116, 32, 111, 102, 32, 101, 120, 112, 101, 114, 105, 109, 101, 110, 116, 97, 116, 105, 111, 110, 32, 99, 97, 110, 32, 101, 118, 101, 114, 32, 112, 114, 111, 118, 101, 32, 109, 101, 32, 114, 105, 103, 104, 116, 59, 32, 97, 32, 115, 105, 110, 103, 108, 101, 32, 101, 120, 112, 101, 114, 105, 109, 101, 110, 116, 32, 99, 97, 110, 32, 112, 114, 111, 118, 101, 32, 109, 101, 32, 119, 114, 111, 110, 103, 46, 32, 45, 32, 65, 108, 98, 101, 114, 116, 32, 69, 105, 110, 115, 116, 101, 105, 110, 32, 91, 83, 71, 68, 93, 0.

(d) (1 mark) ☆☆☆☆ Turn the UTF-8 encodings $b_1,...,b_R$ into a readable message and report it. This task needs no Calculus, so you are invited simply to paste your list of numbers from part (c) into the online tool linked here: https://personal.math.ubc.ca/~loew/utf8-to-text.html.

Solution:

No amount of experimentation can ever prove me right; a single experiment can prove me wrong. - Albert Einstein [SGD]

5. The integrals defined in line (3) are meaningful numbers for any integrable function f. Treating a given function as if it were a combination of periodic functions opens the door to *Fourier analysis*, a powerful tool with many applications in science, engineering, and mathematics. Present exact calculations where they are requested below, but use suitable software to produce the corresponding plots.

(a) (1 mark) ★☆☆☆ Let $f(x) = 1$ for $0 \leq x \leq \pi$. Find a formula for $B_n(f)$ valid for every integer $n \geq 0$.

Solution:

To calculate the Fourier sine series $B_n(f)$ for the function $f(x) = 1$ over the interval $[0, \pi]$, the formula used is:

$$B_n = \frac{2}{\pi}\int_0^{\pi} \sin(nx)\, dx,$$

$$B_n = \frac{2}{\pi}[-\frac{1}{n}\cos\,(nx)]_0^{\pi},$$

$$B_n = \frac{2(1-\cos(n\pi))}{n\pi}.$$

Now to interpret this equation, for odd values of n: $\cos(n\pi) = -1$, so the expression simplifies to $B_n = \frac{4}{n\pi}$ and for even values of n: $\cos(n\pi) = 1$, so the expression simplifies to $B_n = 0$.

This gives us the piecewise function:

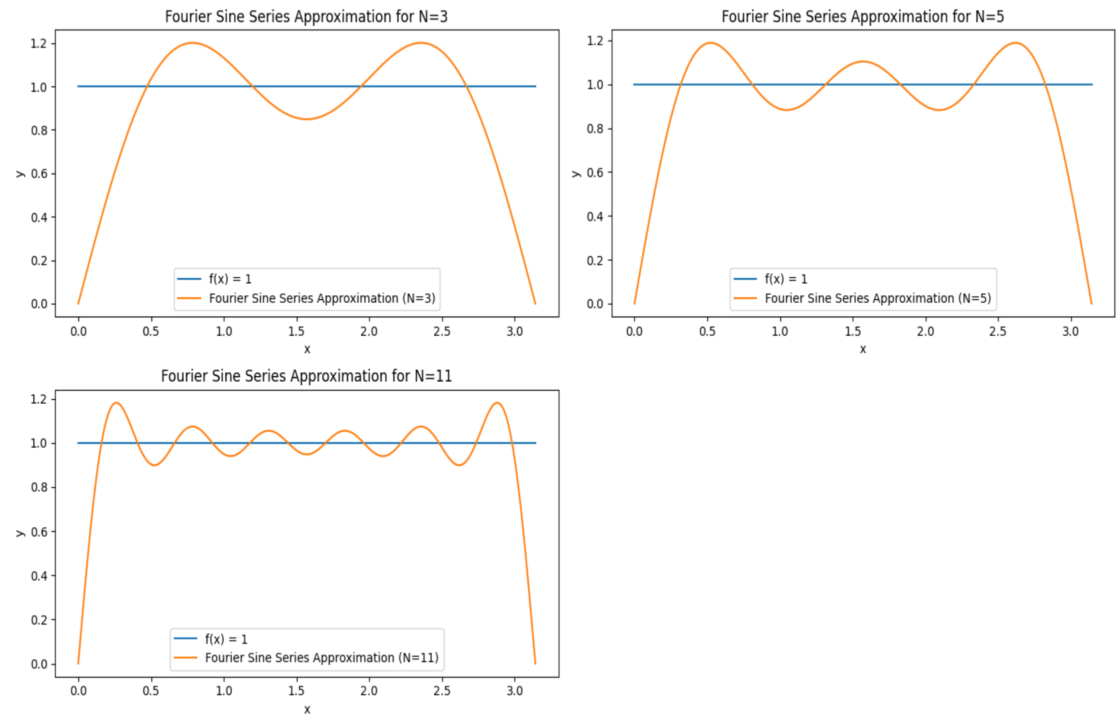
$$B_n(f) = \begin{cases} \frac{4}{n\pi}, \text{if n is odd} \\ 0, \text{if n is even} \end{cases}.$$

(b) (1 mark) Make three plots showing $y = f(x)$ and $y = S_N(x)$ on the same axes, where

$$S_N(x) = \sum_{n=1}^{N} B_n(f)\sin(nx)$$

Use $N = 3$ for the first plot, $N = 5$ for the second, and $N = 11$ for the third.

Solution:




```python
import numpy as np
import matplotlib.pyplot as plt

# Define the Fourier series function
def fourier_series(n_terms, x):
    S_n = np.zeros_like(x)
    for n in range(1, n_terms + 1):
        if n % 2 != 0:  # n is odd
            B_n = 4 / (n * np.pi)
            S_n += B_n * np.sin(n * x)
    return S_n

# Generate x values from 0 to pi
x = np.linspace(0, np.pi, 1000)

# Plot settings
plt.figure(figsize=(14, 8))

# N values for the plots
N_values = [3, 5, 11]

# Create plots for each N
for i, N in enumerate(N_values, start=1):
    plt.subplot(2, 2, i)  # Create a subplot for each N
    plt.plot(x, np.ones_like(x), label='f(x) = 1')  # Plot the original function
    plt.plot(x, fourier_series(N, x), label=f'Fourier Sine Series Approximation (N={N})')  # Plot the Fourier series
    plt.title(f'Fourier Sine Series Approximation for N={N}')
    plt.xlabel('x')
    plt.ylabel('y')
    plt.legend()

# Adjust layout to prevent overlap
plt.tight_layout()

# Display the plot
plt.show()
```

These 3 plots were generated using a python script that I will outline here. The Python code uses NumPy and Matplotlib libraries to compute and visualize the Fourier sine series approximations of a constant function $f(x) = 1$ over the interval $[0, \pi]$. A custom function `fourier_series` is defined to calculate the series up to a specified number of terms, exploiting the fact that only the coefficients for odd terms are non-zero. The code then iterates over three specified series lengths $N = 3,5,11$ to create subplots for each case within a single figure. For each subplot, the constant function and its approximation are plotted on the same axes, providing a visual comparison of the Fourier series' ability to approximate the function as more terms are included.

(c) (1 mark) Let $g(x) = x$ for $0 \le x \le \pi$. Find a formula for $B_n(g)$ valid for every integer $n \ge 0$.

Solution:

To calculate the Fourier sine series coefficient $B_n(g)$ for the function $g(x) = x$ over the interval $[0, \pi]$, the formula used is:

$$B_n = \frac{2}{\pi}\int_0^{\pi} x \sin(nx)\, dx,$$

$$B_n = \frac{2}{\pi}[-\frac{x}{n}\cos(nx)]_0^{\pi} + \frac{2}{n^2\pi}[\sin(nx)]_0^{\pi},$$

$$B_n = \frac{2(\sin(n\pi) - n\pi\cos(n\pi))}{n^2\pi}.$$

This however can be further simplified since the term $\sin(n\pi) = 0$ for all values of n and $\cos(n\pi)$ will alternate signs depending on whether n is even or odd. The formula thus simplifies to,

$$B_n(g) = \frac{2}{n}(-1)^{n+1}$$

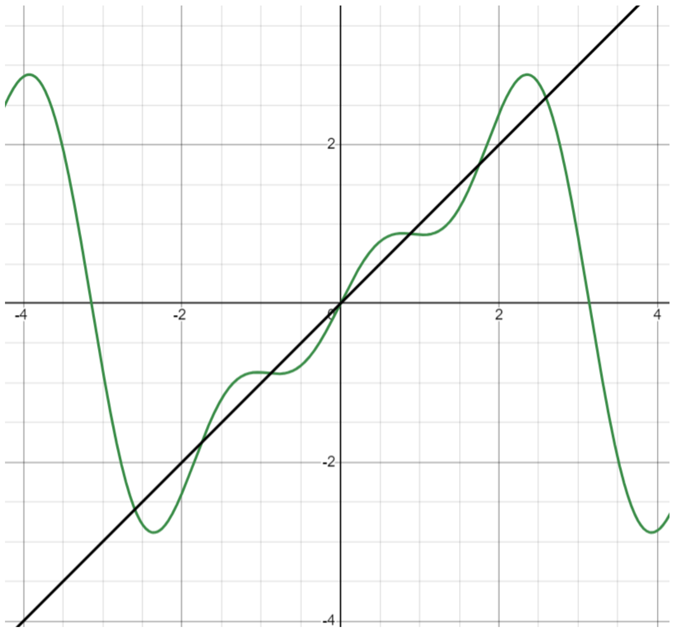
(d) (1 mark) Make three plots showing $y = g(x)$ and $y = \tilde{S}_N(x)$ on the same axes, where

$$\widetilde{S}_N(x) = \sum_{n=1}^{N} B_n(g)\sin(nx)$$

Use $N = 3$ for the first plot, $N = 5$ for the second, and $N = 11$ for the third.

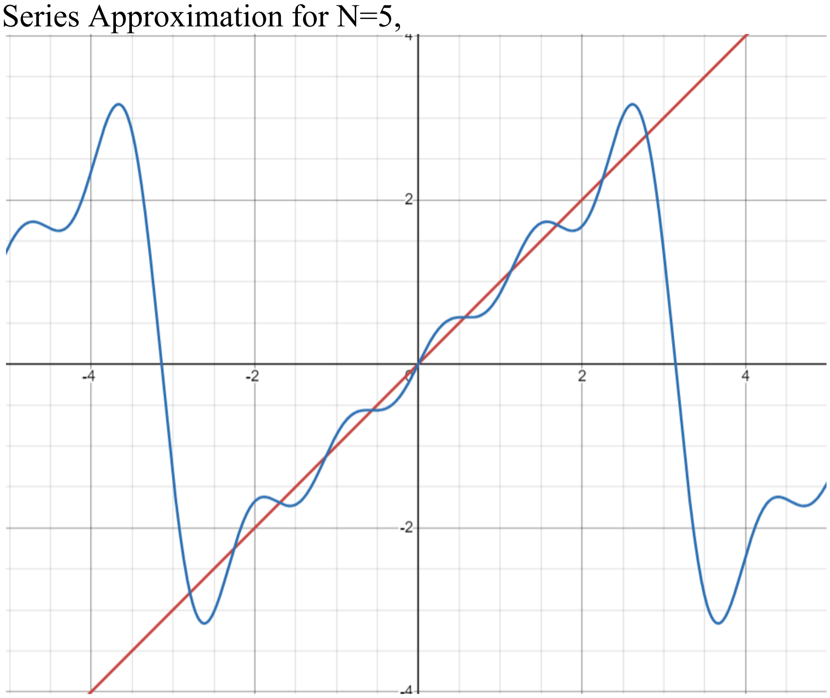
Solution:

Fourier Series Approximation for N=3,



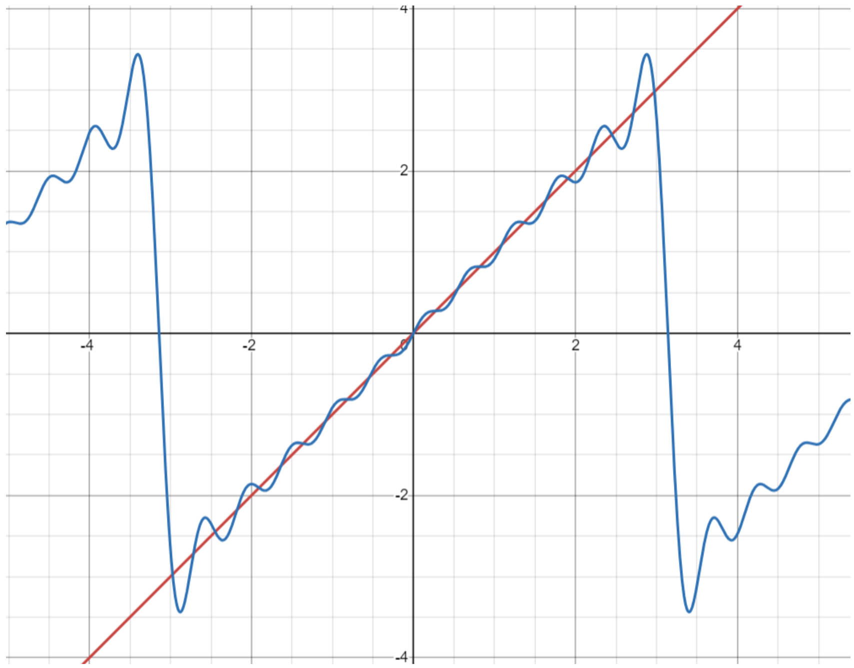
Green: $y = \tilde{S}_N(x)$; black: y = g(x)

Fourier Series Approximation for N=5,



Blue: $y = \tilde{S}_N(x)$; black: y = g(x)

Fourier Series Approximation for N=11,



Blue: $y = \tilde{S}_N(x)$; black: y = g(x)

Notes and Comments

- Question 4 calls for powerful general methods, which should be capable of handling the tasks in Question 3 with ease. This makes the simple problem in Question 3 an ideal test-case for developing a general method. However, solvers are *not required* to use the same methods in these two problems. A simple, direct, no-code solution to Question 3 is fully acceptable.

Credits

- The plots shown on the question sheet were made with Desmos.