# Mathematical Physics III

## Fourier Series

## Lab Report for Assignment _A3

| | | |
|---|---|---|
| **College Roll no.** | : | 2020PHY1097 |
| **University Roll no.** | : | 20068567031 |
| **Name** | : | Kabir Sethi |
| **Unique Paper Code** | : | 32221401 |
| **Paper Title** | : | Mathematical Physics III Lab |
| **Course and Semester** | : | B.Sc.(Hons.) Physics, Sem IV |
| **Due Date** | : | March 1, 2022 |
| **Date of Submission** | : | March 1, 2022 |
| **Lab Report File Name** | : | 2020PHY1097_A_3 |

**Partners :**

| | | |
|---|---|---|
| **Name** | : | Brahmanand Mishra |
| **Roll Number** | : | 2020PHY1184 |
| **Name** | : | Akarsh shukla |
| **Roll Number** | : | 2020PHY1216 |

*Shri Guru Tegh Bahadur Khalsa College, University of Delhi*
*New Delhi-110007, India.*

# Contents

# 1 Introduction

A Fourier series is an expansion of a periodic function f(x) in terms of an infinite sum of sines and cosines.
It is given by:

$$f(x) = a_0 + \sum_{n=1}^{\infty} \Big[a_n cos(\frac{n\pi}{L}x) + b_n sin(\frac{n\pi}{L}x)\Big]$$

# 2 Dirichlet Conditions

The **Dirichlet conditions** are **sufficient conditions** for a real-valued, periodic function f to be equal to the sum of its Fourier series at each point where **f** is continuous.
The conditions are:

1. f must be absolutely integrable over a period.
2. f must be of bounded variation in any given bounded interval.
3. f must have a finite number of discontinuities in any given bounded interval, and the discontinuities cannot be infinite.

# 3 Fourier Representation of a Periodic Function:

Instead of having a period of $2\pi$, many functions have an arbitrary period of say a period of $2L$ .
So the Fourier series for such functions is given by:

$$f(x) = a_0 + \sum_{n=1}^{\infty} \Big[a_n cos(\frac{n\pi x}{L}) + b_n sin(\frac{n\pi x}{L})\Big]$$

Where,

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

$$\boxed{a_n = \frac{1}{L}\int_{-L}^{L} f(x)cos(\frac{n\pi}{L}xdx}$$

$$\boxed{b_n = \frac{1}{L}\int_{-L}^{L} f(x)sin(\frac{n\pi}{L}xdx}$$

# 4 Derive the expressions for Fourier coefficients:

We know that :

$$f(x) = a_0 + \Big[\sum_{n=1}^{\infty} a_n cos(\frac{n\pi x}{L}) + b_n sin(\frac{n\pi x}{L})\Big]$$

Assume that the $F(x)$ has a period of $2\pi$.

$$\therefore \quad L = \pi$$

Putting this in above formula:

$$f(x) = \frac{a_0}{2} + \Big[\sum_{n=1}^{\infty} a_n cos(nx)dx + b_n sin(nx)dx\Big] \tag{1}$$

## 4.1 Derivation For $a_0$

Integrating both sides of equation 1 from $o$ to 2 $\pi$

$$\int_0^{2\pi} f(x)dx = \frac{a_0}{2}\int_0^{2\pi} dx + a_1\int_0^{2\pi} cos(x)dx + a_2\int_0^{2\pi} cos(2x)dx + ....... + a_n\int_0^{2\pi} cos(nx)dx$$
$$+ b_1\int_0^{2\pi} sin(x)dx + b_2\int_0^{2\pi} sin(2x)dx + ....... + b_n\int_0^{2\pi} sin(nx)dx$$

Since $\int_0^{2\pi} sin(nx)dx = 0$ and $\int_0^{2\pi} cos(nx)dx = 0$

$$\int_0^{2\pi} f(x)dx = \frac{a_0}{2}|x|_0^2$$

$$\therefore \boxed{a_0 = \frac{1}{\pi}\int_0^{2\pi} f(x)dx}$$

## 4.2 Derivation For $a_n$

Multiplying each side of equation 1 by cos(nx) and integrating them from $o$ to 2 $\pi$

$$\int_0^{2\pi} f(x)cos(nx)dx = \frac{a_0}{2}\int_0^{2\pi} cos(nx)dx + a_1\int_0^{2\pi} cos(x)cos(nx)dx + ....... + a_n\int_0^{2\pi} cos(nx)^2dx$$
$$+ b_1\int_0^{2\pi} sin(x)cos(nx)dx + ....... + b_n\int_0^{2\pi} sin(nx)cos(nx)dx$$

Since $\int_0^{2\pi} sin(nx)cos(nx)dx = 0$ and $\int_0^{2\pi} cos(nx)^2dx = \pi$

$$\int_0^{2\pi} f(x)cos(nx)dx = a_n\int_0^{2\pi} cos(nx)^2$$

$$\int_0^{2\pi} f(x)cos(nx)dx = a_n(\pi)$$

$$\boxed{a_n = \frac{1}{\pi}\int_0^{2\pi} f(x)cos(nx)dx}$$

### 4.3 Derivation For $b_n$

Multiplying each side of equation 1 by sin(nx) and integrating them from $o$ to 2 $\pi$

$$\int_0^{2\pi} f(x)sin(nx)dx = \frac{a_0}{2}\int_0^{2\pi} sin(nx)dx + a_1\int_0^{2\pi} cos(x)sin(nx)dx + ....... + a_n\int_0^{2\pi} cos(nx)sin(nx)dx$$
$$+ b_1\int_0^{2\pi} sin(x)sin(nx)dx + ....... + b_n\int_0^{2\pi} sin(nx)^2dx$$

Since $\int_0^{2\pi} sin(nx)^2dx = \pi$

$$\int_0^{2\pi} f(x)sin(nx)dx = b_n(\pi)$$

$$\boxed{b_n = \frac{1}{\pi}\int_0^{2\pi} f(x)sin(nx)dx}$$

If the function f(x) is **even** function then $b_n = 0$ and if the function f(x) is **odd** then $a_n = 0$.

# 5 Numericals

- $$f(x) = \begin{cases} 0, -1 < x < 0 \\ 1, 0 < x < 1 \end{cases}$$
  $$f(x+2) = f(x)$$
  Since, The period of the function given is 2.
  $\therefore l = 1$ since,
  $$a_0 = \frac{1}{2\cdot}\int_{-L}^{L} f(x)dx$$
  $$a_0 = \frac{1}{2}\Big[\int_{-1}^{0} 0dx + \int_0^1 (1)dx\Big]$$
  $$\boxed{a_0 = \frac{1}{2}}$$
  $$a_n = \frac{1}{L}\int_{-L}^{L} f(x)cos(\frac{n\pi}{L})xdx$$
  $$a_n = \frac{1}{1}\Big[\int_{-1}^{0} 0\cdot cos(\frac{n\pi}{1})xdx + \int_0^1 1\cdot cos(\frac{n\pi}{1})xdx\Big]$$
  $$a_n = \frac{1}{n\pi}\Big|sin(n\pi x)\Big|_o^1$$
  $$\therefore \boxed{a_n = 0}$$

since,

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

$$b_n = \frac{1}{1}\Big[\int_{-1}^{0} 0\cdot sin(\frac{n\pi}{1})xdx + \int_{0}^{1} 1\cdot sin(\frac{n\pi}{1})xdx\Big]$$

$$b_n = \frac{1}{n\pi}\Big| - cos(n\pi x)\Big|_{o}^{1}$$

$$b_n = \begin{cases} 0, \text{When n is even} \\ \frac{2}{n\pi}\text{When n is odd} \end{cases}$$

Therefore , the complete function is represented by:

$$f(x) = \frac{1}{2} + \sum_{n=1}^{\infty}[0 + \frac{2}{n\pi}sin(n\pi x)]$$

since, we are getting $a_n = 0$ and $a_0 \neq 0$. Therefore the given function is neither **even** nor **odd**.

-

$$f(x) = \begin{cases} 0, -1 < x < -0.5 \\ 1, -0.5 < x < 0 \\ 0, \ \ 0.5 < x < 1 \end{cases}$$

Since, The period of the function given is 2
$\therefore$ L=1

$$a_0 = \frac{1}{2}\Big[\int_{-1}^{-0.5} 0dx + \int_{-0.5}^{0.5}(1)dx + \int_{0.5}^{1} 0dx\Big]$$

$$\boxed{a_0 = \frac{1}{2}}$$

$$a_n = \frac{1}{1}\Big[\int_{-1}^{-0.5} 0\cdot cos(\frac{n\pi}{1})xdx + \int_{-0.5}^{0.5}(1)\cdot cos(\frac{n\pi}{1})xdx + \int_{0.5}^{1} 0\cdot cos(\frac{n\pi}{1})xdx\Big]$$

$$a_n = \frac{1}{n\pi}\Big|sin(n\pi x)\Big|_{-0.5}^{0.5}$$

$$a_n = \frac{1}{n\pi}[2sin(\frac{n\pi}{2}]$$

$$a_n = \boxed{\begin{cases} 0, \ \text{if n is even} \\ \frac{1}{n\pi}[2sin(\frac{n\pi}{2}], \text{if n is odd} \end{cases}}$$

$$b_n = \frac{1}{1}\Big[\int_{-1}^{-0.5} 0\cdot sin(\frac{n\pi}{1})xdx + \int_{-0.5}^{0.5}(1)\cdot sin(\frac{n\pi}{1})xdx + \int_{0.5}^{1} 0\cdot sin(\frac{n\pi}{1})xdx\Big]$$

$$b_n = \frac{1}{n\pi}\Big| - cos(n\pi x)\Big|_{-0.5}^{0.5}$$

$$b_n = \frac{1}{n\pi}[-2cos(\frac{n\pi}{2})]$$

$$\boxed{b_n = 0}$$

Therefore , the complete function is represented by:
If $n$ is odd:

$$f(x) = \frac{1}{2} + \sum_{n=1}^{\infty}[0 + 0]$$

If $n$ is even:

$$f(x) = \frac{1}{2} + \sum_{n=1}^{\infty}[\frac{1}{n\pi}(2sin(\frac{n\pi}{2})cos(n\pi x) + 0]$$

since, we are getting $b_n = 0$ and $a_0 \neq 0$. Therefore the given function is **odd** function .

-
$$f(x) = \begin{cases} -0.5, -1 < x < 0 \\ 0.5, \ \ 0 < x < 1 \end{cases}$$

$$f(x+2) = f(x)$$

Since, The period of the function given is 2
$\therefore$ L=1

$$a_0 = \frac{1}{2}\Big[\int_{-1}^{0}(-0.5)dx + \int_{0}^{1}(0.5)dx\Big]$$

$$a_0 = \frac{1}{2}\Big[(-0.5) + (0.5)\Big]$$

$$a_0 = 0$$

$\therefore$

$$a_n = \frac{1}{1}\Big[\int_{-1}^{0}(-0.5)\cdot cos(\frac{n\pi}{1})xdx + \int_{0}^{1}(0.5)\cdot cos(\frac{n\pi}{1})xdx\Big]$$

$$a_n = \Big[\frac{(-0.5)}{n\pi}\Big|sin(n\pi x)\Big|_{-1}^{0} + \frac{(0.5)}{n\pi}\Big|sin(n\pi x)\Big|_{0}^{1}\Big]$$

$$a_n = 0$$

Now,

$$b_n = \frac{1}{1}\Big[\int_{-1}^{0}(-0.5)\cdot sin(\frac{n\pi}{1})xdx + \int_{0}^{1}(0.5)\cdot cos(\frac{n\pi}{1})xdx\Big]$$

$$b_n = \Big[\frac{(-0.5)}{n\pi}\Big| - cos(n\pi x)\Big|_{-1}^{0} + \frac{(0.5)}{n\pi}\Big| - cos(n\pi x)\Big|_{0}^{1}\Big]$$

$$b_n = \frac{1}{n\pi}\Big[-1 + cos(n\pi)\Big]$$

If $n$ is odd

$$\boxed{b_n = \frac{1}{n\pi}[-2]}$$

If $n$ is even

$$\boxed{b_n = 0}$$

Therefore , the complete function is represented by:

$$f(x) = \begin{cases} 0 + \sum_{n=1}^{\infty}[\frac{1}{n\pi} - 2sin(n\pi x)], \text{when n is odd} \\ 0, \quad \text{when n is even} \end{cases}$$

Since, when n is even we are getting $a_0 = 0$ , $a_n = 0$ and $b_n = 0$. So in this case we can't say anything about the function . But when n is odd we are getting $a_0 = a_n = 0$ and $b_n \neq 0$ . Therefore the function is odd function.

# 6 Gibbs Phenomenon

According to this Phenomenon in every piece wise continuous function there is some overshoot in the approximated values at the point of discontinuity.

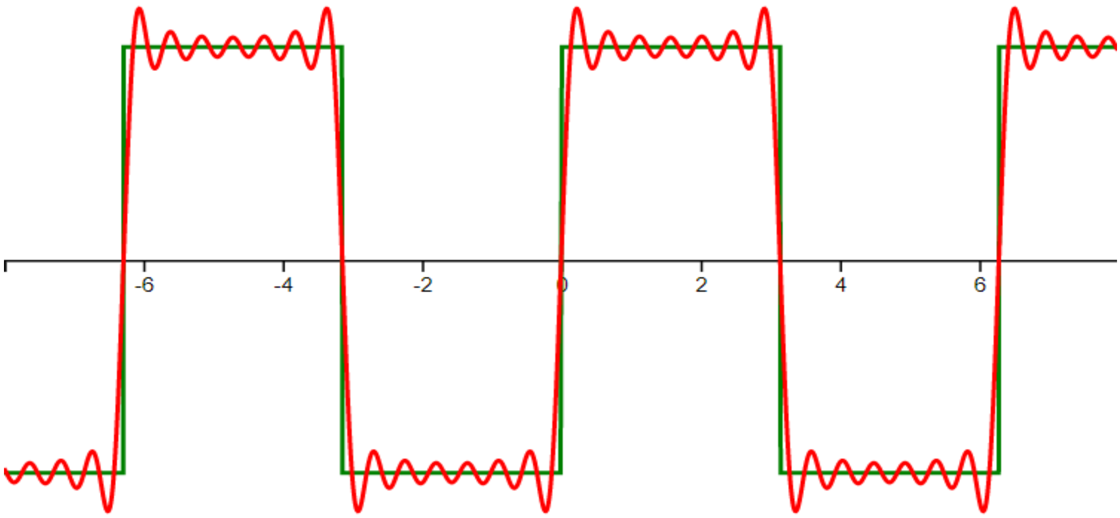


Figure 1: Gibbs Phenomenon

# 7 Half Range Fourier Series

If a function is defined over half the range, say 0 to L, instead of the full range from $L$ to L, it may be expanded in a series of sine terms only or of cosine terms only. The series produced is then called a half Range series.

## 7.1 Half Range Cosine Series

A function can be expanded using half its range from

1. 0 to $L$
2. -L to 0
3. L to 2L

That is, the range of integration is L. The Fourier series of the half range even function is given by:

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

for $n = 1, 2, 3, \ldots,$ where

$$a_0 = \frac{2}{L}\int_0^L f(x)dx$$

$$a_n = \frac{2}{L}\int_0^L f(x)cos(\frac{n\pi x}{L})dx$$

and

$$b_n = 0$$

## 7.2 Half Range sine Series

An odd function can be expanded using half its range from 0 to L, i.e. the range of integration has value L. The Fourier series of the odd function is:

Since $a_0 = 0 \quad and \quad a_n = 0$, we have:

$$\sum_{n=1}^{\infty} b_n sin(\frac{n\pi x}{L})$$

where,

$$\boxed{b_n = \frac{2}{L}\int_0^L f(x)sin(\frac{n\pi x}{L})dx}$$

## 7.3 f(x)=x

since, f(x)=x is a odd function .
$\therefore\ a_0 \text{and} a_n = 0$

$$b_n = \frac{2}{\pi}\int_0^{\pi} xsin(\frac{n\pi x}{\pi})dx$$

$$b_n = \frac{2}{\pi}\int_0^{\pi} xsin(nx)dx$$

$$b_n = \frac{2}{\pi}|-xcos(x) + sin(x)|_0^{\pi}$$

$$b_n = 2$$

Therefore, the complete function is given by:

$$f(x) =$$

# 8 cosine and sine Fourier representations

A Fourier series is given by:

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty}[a_n cos(nx) + b_n cos(nx) \tag{2}$$

Integrating the equation(2) with respect to x in the interval $-\pi$ and $\pi$.

$$\int_{-\pi}^{\pi} f(x)dx = \frac{a_0}{2} + \int_{-\pi}^{\pi}\sum_{n=1}^{\infty} a_n cos(nx)dx + \int_{-\pi}^{\pi}\sum_{n=1}^{\infty} b_n sin(nx)dx$$

$$2\int_{0}^{\pi} f(x)dx = \frac{a_0}{2}(2\pi) + 0 + 0$$

$$\therefore \quad \boxed{a_0 = \frac{2}{\pi}\int_{0}^{\pi} f(x)dx}$$

For $a_n$:
Multiply equation(2) by cos(nx) and integrating between $[-\pi,\ \pi]$ with respect to x by assuming that the function is even

$$\int_{-\pi}^{\pi} f(x)dxcos(nx) = \frac{a_0}{2}\int_{-\pi}^{\pi} cos(nx)dx + \int_{-\pi}^{\pi}\sum_{n=1}^{\infty} a_n cos(nx)^2 dx + \int_{-\pi}^{\pi}\sum_{n=1}^{\infty} b_n sin(nx)cos(nx)dx$$

since, $\int_{-\pi}^{\pi} sin(nx)cos(nx) = 0$ and $\int_{-\pi}^{\pi} cos(nx) = 0$

$$2\int_{0}^{\pi} f(x)dxcos(nx) = 2a_n(\frac{\pi}{2})$$

$$\boxed{a_n = \frac{2}{\pi}\int_{0}^{\pi} f(x)cos(nx)dx}$$

Now, for calculating the $b_n$ in half period we assume that f(x) is odd
Now, multiply equation(2) by sin(nx) and integrating between [-$\pi$,$\pi$]

$$\therefore \quad = \int_{-\pi}^{\pi} f(x)dxsin(nx) = \frac{a_0}{2}\int_{-\pi}^{\pi} sin(nx)dx + \int_{-\pi}^{\pi}\sum_{n=1}^{\infty} a_n cos(nx)sin(nx)dx + \int_{-\pi}^{\pi}\sum_{n=1}^{\infty} b_n sin(nx)^2 dx$$

$$2\int_{0}^{\pi} f(x)dxsin(nx) = 2b_n(\frac{\pi}{2})$$

$$\boxed{b_n = \frac{2}{\pi}\int_{0}^{\pi} f(x)sin(nx)dx}$$

# 9 Algorithm

**Algorithm 1** Fourier Analysis

**function** SERIES($x, coeff, period$)

▷ Defining the function to calculate the series for fourier expansion

ser = []
cosnx = lambda x,n: np.cos(n*np.pi*x/period)
sinnx = lambda x,n: np.sin(n*np.pi*x/period) **for** i in x: **do**
**end**
For loop
dummy variable d = 0 **for** j in range(len(coeff[0])): **do**
**end**
d += cosnx(i,j)*coeff[0][j]
**for** j in range(len(coeff[1])): **do**
**end**
d += sinnx(i,j)*coeff[1][j]
ser.append(temp)

**return** *ser*

**end function**

**function** FOURIER($f, value, N, d, period, method, fl = 1$)

$a\_n = []$
$b\_n = []$
meth_dict = "simp":MySimp,"trap":MyTrap,"gauss":MyLegGauss
$int\_func = meth\_dict[method]$ **if** $value == 1 or value == 0 :$ **then**
**end**
comb_func = lambda x,i: np.sin((i * np.pi * x) / period) **for** i in range(N): **do**
**end**
$f1 = lambdax : comb_func(x, i) * f(x)$
$b\_n.append(fl * (int\_func(f = f1, a = 0, b = period, n = 12, m = 50))/(period))$

**if** value == -1 or value == 0: **then**
**end**
$comb\_func = lambdax, i : np.cos((i * np.pi * x)/period)$ **for** i in range(N): **do**
**end**
f1 = lambda x: comb_func(x,i) * f(x)
a_n.append(fl*(int_func(f=f1, a=0, b= period, n=12, m=50)) / (period)) **if** i == 0: **then**
**end**
$a\_n[0] = a\_n[0]/2$

**return** [a_n,b_n]
**end function**

# 10 Discussion

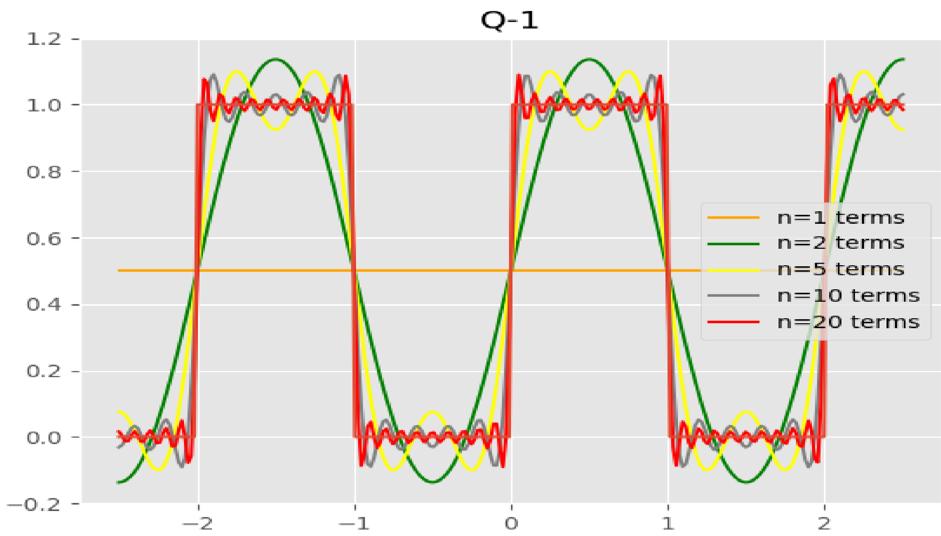


Figure 2: Question 1

FOR X = -0.5

| | n values | series values | relative error |
|---|---|---|---|
| 0 | 1 | 0.500000 | 0.500000 |
| 1 | 2 | -0.136620 | -0.136620 |
| 2 | 5 | 0.075587 | 0.075587 |
| 3 | 10 | -0.031527 | -0.031527 |
| 4 | 20 | 0.015876 | 0.015876 |

FOR X = 0

| | n values | series values | relative error |
|---|---|---|---|
| 0 | 1 | 0.5 | 0.5 |
| 1 | 2 | 0.5 | 0.5 |
| 2 | 5 | 0.5 | 0.5 |
| 3 | 10 | 0.5 | 0.5 |
| 4 | 20 | 0.5 | 0.5 |

FOR X = 0.5

| | n values | series values | relative error |
|---|---|---|---|
| 0 | 1 | 0.5 | -0.500000 |
| 1 | 2 | 0.5 | 0.136620 |
| 2 | 5 | 0.5 | -0.075587 |
| 3 | 10 | 0.5 | 0.031527 |
| 4 | 20 | 0.5 | -0.015876 |

Figure 3: Table for Question 1

From above and below graphs we observed that as the value of number of intervals increases the overshoot values at discontinuities decreases . They do not reaches to zero but they are continuously decreasing.

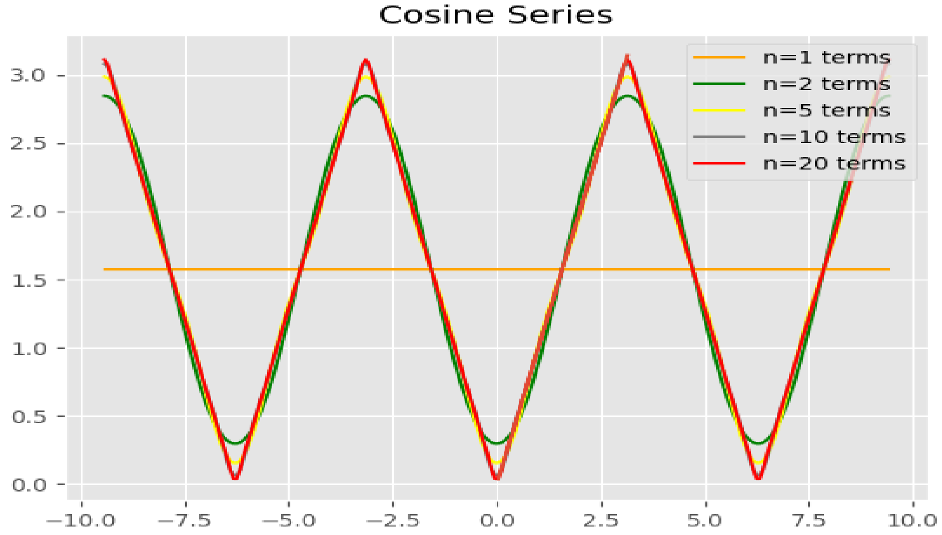


Figure 4: cosine half series plot

FOR COSINE HALF SERIES

| | n values | series values | relative error |
|---|---|---|---|
| 0 | 1 | 0.250000 | 0.250000 |
| 1 | 2 | 0.047358 | 0.047358 |
| 2 | 5 | 0.024842 | 0.024842 |
| 3 | 10 | 0.010099 | 0.010099 |
| 4 | 20 | 0.005062 | 0.005062 |

| | n values | series values | relative error |
|---|---|---|---|
| 0 | 1 | 0.250000 | -1.320796 |
| 1 | 2 | 0.205300 | -1.365496 |
| 2 | 5 | 0.219234 | -1.351563 |
| 3 | 10 | 0.213828 | -1.356968 |
| 4 | 20 | 0.214352 | -1.356445 |

| | n values | series values | relative error |
|---|---|---|---|
| 0 | 1 | 0.250000 | -2.891593 |
| 1 | 2 | 0.205300 | -2.708670 |
| 2 | 5 | 0.219234 | -2.703399 |
| 3 | 10 | 0.213828 | -2.714089 |
| 4 | 20 | 0.214352 | -2.712016 |

Figure 5: Table for cosine half series


Figure 6: sine half series plot

FOR SINE HALF SERIES

| | n values | series values | relative error |
|---|---|---|---|
| 0 | 1 | 0.0 | 0.0 |
| 1 | 2 | 0.0 | 0.0 |
| 2 | 5 | 0.0 | 0.0 |
| 3 | 10 | 0.0 | 0.0 |
| 4 | 20 | 0.0 | 0.0 |

| | n values | series values | relative error |
|---|---|---|---|
| 0 | 1 | 0.000000 | -1.570796 |
| 1 | 2 | -0.620939 | -2.191735 |
| 2 | 5 | -0.440914 | -2.011711 |
| 3 | 10 | -0.441183 | -2.011979 |
| 4 | 20 | -0.448508 | -2.019304 |

| | n values | series values | relative error |
|---|---|---|---|
| 0 | 1 | 0.000000 | -3.141593 |
| 1 | 2 | -0.620939 | -3.415531 |
| 2 | 5 | -0.440914 | -4.024827 |
| 3 | 10 | -0.441183 | -4.090837 |
| 4 | 20 | -0.448508 | -4.047510 |

Figure 7: Table for sine half series

# 11 code

```
#Akarsh Shukla
#2020PHY1216
# Kabir sethi (2020PHY1097) and Brahmanand Mishra(2020PHY1184)
import matplotlib.pyplot as plt
from  MyIntegration  import  *
from matplotlib import  style
import pandas as pd
import numpy as np
from matplotlib import use
style.use("ggplot")
use("WebAgg")
def series(x, coeff,period):
    ser = []
    cosnx = lambda x,n: np.cos(n*np.pi*x/period)
    sinnx = lambda x,n: np.sin(n*np.pi*x/period)
    for i in x:
        temp = 0
        for j in range(len(coeff[0])):
            temp += cosnx(i,j)*coeff[0][j]
        for j in range(len(coeff[1])):
            temp += sinnx(i,j)*coeff[1][j]
        ser.append(temp)
    return ser

def fourier(f,  value,  N, d, period, method, fl = 1):
    a_n = []
    b_n = []
    meth_dict = {"simp":MySimp,"trap":MyTrap,"gauss":MyLegGauss}
    int_func = meth_dict[method]
    if value == 1 or value == 0:
        comb_func = lambda x,i: np.sin((i * np.pi * x) / period)
        for i in range(N):
            f1 = lambda x: comb_func(x,i) * f(x)
            if int_func == MyLegGauss:
                b_n.append(fl*(int_func(f = f1, a=0, b = period, n =12,m = 50)) / (period))
            else:
                a_n.append(method(fl*(int_func( a,  period, 1, f1,d)) / (period)))

    if value == -1 or value == 0:
        comb_func = lambda x,i: np.cos((i * np.pi * x) / period)
        for i in range(N):
            f1 = lambda x: comb_func(x,i) * f(x)
            if int_func == MyLegGauss:
                a_n.append(fl * (int_func(f=f1, a=0, b=period, n=12, m=50)) / (period))
            else:
                an.append(method(fl * (int_func(a, period, 1, f1, d)) / (period)))
            if i == 0:
                a_n[0] = a_n[0]/2
    return(a_n,b_n)



```

```


@np.vectorize
def sawtooth(x):
    if 0 <= x <=  np.pi:
        return x


@np.vectorize
def function1(x):
    if  type(x) != list:
        if -1 < x <= 0 or 1 < x < 2 or -2.5 <= x <= -2:
            return 0
        elif 0 < x <= 1 or -2 < x <= -1 or 2 < x <= 2.5:
            return 1
    else:
        temp = []
        for i in x:
            temp.append(function1(i))
        return temp

@np.vectorize
def function2(x):
    if  type(x) != list:
        if -1.5 <= x < -1 or 0.5 <= x < 1:
            return 0
        elif -2.5 <= x < -1.5 or -0.5 <= x < 0.5 or 1.5 <= x <= 2.5:
            return 1
        elif -1 <= x < -0.5 or 1 <= x < 1.5:
            return 0
    else:
        temp = []
        for i in x:
            temp.append(function2(i))
        return temp



@np.vectorize
def function3(x):
    if  type(x) != list:
        if -2.5 <= x < -2 or -1 <= x < 0 or 1 <= x < 2:
            return -0.5
        elif -2 <= x < -1 or 0 <= x < 1 or 2 <= x <= 2.5:
            return 0.5

    else:
        temp = []
        for i in x:
            temp.append(function2(i))
        return temp





list = [1, 2, 5, 10, 20]

```

```
print("FOR  X = -0.5")

x_check = [-0.5, 0, 0.5]
out_1 = []
for i in range(len(list)):
    t =  fourier(function1,0,  list[i],1e-10, 1, "gauss", 1)
    out_1.append((series(x_check, t, 1))[0])

comp_out_1 = function1(x_check)
rel = np.array(out_1) - comp_out_1[0]

dict = {"n values " : list, "series values" : out_1,  "relative error" :
    rel}
data = pd.DataFrame.from_dict(dict)
print(data)


print("FOR  X = 0")

out_2 = []

for i in range(len(list)):
    t =  fourier(function1,0,  list[i],1e-10, 1, "gauss", 1)
    out_2.append((series(x_check, t, 1))[1])

comp_out_1 = function1(x_check)
rel2 = np.array(out_2) - comp_out_1[1]

dict2 = {"n values " : list, "series values" : out_2,  "relative error" :
     rel2}
data2 = pd.DataFrame.from_dict(dict2)
print(data2)

print("FOR  X = 0.5")

out_3 = []

for i in range(len(list)):
    t =  fourier(function1,0,  list[i],1e-10, 1, "gauss", 1)
    out_3.append((series(x_check, t, 1))[2])

comp_out_1 = function1(x_check)
rel3 = np.array(out_3) - comp_out_1[2]

dict3 = {"n values " : list, "series values" : out_2,  "relative error" :
     rel3}
data3 = pd.DataFrame.from_dict(dict3)
print(data3)




x2 = np.linspace(-2.5, 2.5, 250)




value = []
```

```python
for i in range(len(list)):
    t =  fourier(function1,0,  list[i],1e-10, 1, "gauss")
    value.append(series(x2, t, 1))
np.savetxt("Q1.txt", t)

figure, axis = plt.subplots(1,1)
plt.title("Q-1")
plt.plot(x2, value[0],label='n=1 terms',color="orange")
plt.plot(x2, value[1],label='n=2 terms',color="green")
plt.plot(x2, value[2],label='n=5 terms',color="yellow")
plt.plot(x2, value[3],label='n=10 terms',color="grey")
plt.plot(x2, value[4],label='n=20 terms',color="red")
plt.plot(x2, function1(x2))
plt.legend()



value2 = []
for i in range(len(list)):
    t =  fourier(function2,-1,  list[i],1e-10, 1, "gauss")
    value2.append(series(x2, t, 1))

figure, axis = plt.subplots(1,1)
plt.title("Q-2")
plt.plot(x2, value2[0],label='n=1 terms',color="orange")
plt.plot(x2, value2[1],label='n=2 terms',color="green")
plt.plot(x2, value2[2],label='n=5 terms',color="yellow")
plt.plot(x2, value2[3],label='n=10 terms',color="grey")
plt.plot(x2, value2[4],label='n=20 terms',color="red")
plt.plot(x2, function2(x2))
plt.legend()


value3 = []
for i in range(len(list)):
    t =  fourier(function3,0,  list[i],1e-10, 1, "gauss")
    value3.append(series(x2, t, 1))

figure, axis = plt.subplots(1,1)
plt.title("Q-3")
plt.plot(x2, value3[0],label='n=1 terms',color="orange")
plt.plot(x2, value3[1],label='n=2 terms',color="green")
plt.plot(x2, value3[2],label='n=5 terms',color="yellow")
plt.plot(x2, value3[3],label='n=10 terms',color="grey")
plt.plot(x2, value3[4],label='n=20 terms',color="red")
plt.plot(x2, function3(x2))
plt.legend()

x2 = np.linspace(-3*np.pi, 3*np.pi, 250)

value4 = []
for i in range(len(list)):
    t =  fourier(sawtooth,-1,  list[i],1e-10, np.pi, "gauss", fl = 2)
    value4.append(series(x2, t, np.pi))

figure, axis = plt.subplots(1,1)
plt.title("Cosine Series")
plt.plot(x2, value4[0],label='n=1 terms',color="orange")
```

```python
plt.plot(x2, value4[1],label='n=2 terms',color="green")
plt.plot(x2, value4[2],label='n=5 terms',color="yellow")
plt.plot(x2, value4[3],label='n=10 terms',color="grey")
plt.plot(x2, value4[4],label='n=20 terms',color="red")
plt.plot(x2, sawtooth(x2))
plt.legend()

value5 = []
for i in range(len(list)):
    t =  fourier(sawtooth,1,  list[i],1e-10, np.pi, "gauss", fl = 2)
    value5.append(series(x2, t, np.pi))

figure, axis = plt.subplots(1,1)
plt.title("Sine Series")
plt.plot(x2, value5[0],label='n=1 terms',color="orange")
plt.plot(x2, value5[1],label='n=2 terms',color="green")
plt.plot(x2, value5[2],label='n=5 terms',color="yellow")
plt.plot(x2, value5[3],label='n=10 terms',color="grey")
plt.plot(x2, value5[4],label='n=20 terms',color="red")
plt.plot(x2, sawtooth(x2))
plt.legend()

print("FOR COSINE HALF SERIES")
x_check = [0, np.pi/2, np.pi]
out_1 = []
for i in range(len(list)):
    t =  fourier(sawtooth,-1,  list[i],1e-10, 1, "gauss", 1)
    out_1.append((series(x_check, t, 1))[0])

comp_out_1 = sawtooth(x_check)
rel = np.array(out_1) - comp_out_1[0]

dict = {"n values " : list, "series values" : out_1,  "relative error" :
    rel}
data = pd.DataFrame.from_dict(dict)
print(data)

out_2 = []

for i in range(len(list)):
    t =  fourier(sawtooth,-1,  list[i],1e-10, 1, "gauss", 1)
    out_2.append((series(x_check, t, 1))[1])

comp_out_1 = sawtooth(x_check)
rel2 = np.array(out_2) - comp_out_1[1]

dict2 = {"n values " : list, "series values" : out_2,  "relative error" :
     rel2}
data2 = pd.DataFrame.from_dict(dict2)
print(data2)



out_3 = []

for i in range(len(list)):
    t =  fourier(sawtooth,-1,  list[i],1e-10, 1, "gauss", 1)
    out_3.append((series(x_check, t, 1))[2])
```

```

comp_out_1 = sawtooth(x_check)
rel3 = np.array(out_3) - comp_out_1[2]

dict3 = {"n values " : list, "series values" : out_2,  "relative error" :
     rel3}
data3 = pd.DataFrame.from_dict(dict3)
print(data3)



print("FOR SINE HALF SERIES")
x_check = [0, np.pi/2, np.pi]
out_1 = []
for i in range(len(list)):
    t =  fourier(sawtooth,1,  list[i],1e-10, 1, "gauss", 2)
    out_1.append((series(x_check, t, 1))[0])

comp_out_1 = sawtooth(x_check)
rel = np.array(out_1) - comp_out_1[0]

dict = {"n values " : list, "series values" : out_1,  "relative error" :
    rel}
data = pd.DataFrame.from_dict(dict)
print(data)

out_2 = []

for i in range(len(list)):
    t =  fourier(sawtooth,1,  list[i],1e-10, 1, "gauss", 2)
    out_2.append((series(x_check, t, 1))[1])

comp_out_1 = sawtooth(x_check)
rel2 = np.array(out_2) - comp_out_1[1]

dict2 = {"n values " : list, "series values" : out_2,  "relative error" :
     rel2}
data2 = pd.DataFrame.from_dict(dict2)
print(data2)



out_3 = []

for i in range(len(list)):
    t =  fourier(sawtooth,1,  list[i],1e-10, 1, "gauss",  fl = 2)
    out_3.append((series(x_check, t, 1))[2])

comp_out_1 = sawtooth(x_check)
rel3 = np.array(out_3) - comp_out_1[2]

dict3 = {"n values " : list, "series values" : out_2,  "relative error" :
     rel3}
data3 = pd.DataFrame.from_dict(dict3)
print(data3)
plt.show()
```

# References


[1] Introductory Methods of Numerical Analysis by S.S. Sastry

[2] Mathematical Methods for Physics and Engineering by Riley , Hobson and Bence

[3] Wolfram Mathworld
https://mathworld.wolfram.com

[4] Wikipedia
https://en.wikipedia.org/wiki