Fourier Series Representation
Lab Report for Assignment No. 3

SHASHVAT JAIN (2020PHY1114)
ANKUR KUMAR (2020PHY1113)
HARSH SAXENA (2020PHY1162)

S.G.T.B. Khalsa College, University of Delhi, Delhi-110007, India.

March 1, 2022

Submitted to Dr. Mamta
"32221401 - MATHEMATICAL PHYSICS III"

Contents

1 Theory

Not all functions can be represented in the form of Fourier Series. The particular conditions that a function f must fulfil in order that it may be expanded as a Fourier series are known as *Dirichlet conditions*, and may be summarised by the following four points:

1. The function must be periodic.
2. The function must be single-valued and continuous throughout the domain except at a finite number of finite discontinuities.
3. It must have only a finite number of maxima and minima within the period.
4. The integral over one period of $|f(x)|$ must converge.

If the above mentioned conditions are satisfied then the Fourier series representation of f converges to the true value $f(x)$ if f is continuous at x and if x is a point of discontinuity then the value of series representation at x is the average of the values of the function obtained at the neighbouring points of x, also called the *Dirichlet theorem.*. The Dirichlet conditions are sufficient conditions for f to be represented as a Fourier series but they are **not** all necessary.
The Fourier series of a periodic function $f(x+2L) = f(x)$ can be written as,

$$\boxed{f(x) = \sum_{n=0}^{\infty} a_n \cos\left(\frac{n\pi x}{L}\right) + b_n \sin\left(\frac{n\pi x}{L}\right)} \tag{1}$$

where the Fourier coefficients $a_n, b_n \in R$, To obtain a_n, b_n we make use of the following orthogonality properties of $\sin(x)$ and $\cos(x)$ functions,

$$\int_{-L}^{L} \cos\left(\frac{m\pi x}{L}\right) \sin\left(\frac{n\pi x}{L}\right) dx = 0 \tag{2}$$

$$\int_{-L}^{L} \sin\left(\frac{m\pi x}{L}\right) \sin\left(\frac{n\pi x}{L}\right) dx = \begin{cases} 0 & n = m = 0 \\ L\delta_{mn} & n, m \neq 0 \end{cases}$$

$$\int_{-L}^{L} \cos\left(\frac{m\pi x}{L}\right) \cos\left(\frac{n\pi x}{L}\right) dx = \begin{cases} 2L & n = m = 0 \\ L\delta_{mn} & n, m \neq 0 \end{cases} \tag{3}$$

To obtain a_k multiply both sides of the equality 1 by $\cos\left(\frac{k\pi x}{L}\right)$ and integrate both sides with respect to x in the interval $[-L, L]$, we get,

$$\begin{aligned}
\int_{-L}^{L} \cos\left(\frac{k\pi x}{L}\right) f(x) dx &= \int_{-L}^{L} \cos\left(\frac{k\pi x}{L}\right) \left[\sum_{n=0}^{\infty} a_n \cos\left(\frac{n\pi x}{L}\right) + b_n \sin\left(\frac{n\pi x}{L}\right)\right] dx \\
&= \sum_{n=0}^{\infty} \left[\int_{-L}^{L} \cos\left(\frac{k\pi x}{L}\right) a_n \cos\left(\frac{n\pi x}{L}\right) dx + \int_{-L}^{L} \cos\left(\frac{k\pi x}{L}\right) b_n \sin\left(\frac{n\pi x}{L}\right) dx\right] \\
&= \sum_{n=0}^{\infty} \left[a_n \int_{-L}^{L} \cos\left(\frac{k\pi x}{L}\right) \cos\left(\frac{n\pi x}{L}\right) dx + b_n \int_{-L}^{L} \cos\left(\frac{k\pi x}{L}\right) \sin\left(\frac{n\pi x}{L}\right) dx\right]
\end{aligned}$$

Using property 2 the terms with b_n disappear,

$$\int_{-L}^{L} \cos\left(\frac{k\pi x}{L}\right) f(x) dx = \sum_{n=0}^{\infty} \left[a_n \int_{-L}^{L} \cos\left(\frac{k\pi x}{L}\right) \cos\left(\frac{n\pi x}{L}\right) dx\right]$$

Using property 3 all terms for which $n \neq k$ disappear, leaving behind,

$$\int_{-L}^{L} \cos\left(\frac{k\pi x}{L}\right) f(x)dx = \begin{cases} a_k L & k \neq 0 \\ a_k 2L & k = 0 \end{cases}$$

$$\implies \boxed{a_k = \frac{1}{L}\int_{-L}^{L} \cos\left(\frac{k\pi x}{L}\right) f(x)dx \quad \text{for } k \neq 0} \tag{4}$$

$$\boxed{a_0 = \frac{1}{L}\int_{-L}^{L} f(x)dx \quad \text{for } k = 0} \tag{5}$$

Similarly, to obtain b_k multiply both sides of the equality 1 by $\sin\left(\frac{k\pi x}{L}\right)$ and integrate both sides with respect to x in the interval $[-L, L]$, we get,

$$\begin{aligned}
\int_{-L}^{L} \sin\left(\frac{k\pi x}{L}\right) f(x)dx &= \int_{-L}^{L} \sin\left(\frac{k\pi x}{L}\right) \left[\sum_{n=0}^{\infty} a_n \cos\left(\frac{n\pi x}{L}\right) + b_n \sin\left(\frac{n\pi x}{L}\right)\right] dx \\
&= \sum_{n=0}^{\infty} \left[\int_{-L}^{L} \sin\left(\frac{k\pi x}{L}\right) a_n \cos\left(\frac{n\pi x}{L}\right) dx + \int_{-L}^{L} \cos\left(\frac{k\pi x}{L}\right) b_n \sin\left(\frac{n\pi x}{L}\right) dx\right] \\
&= \sum_{n=0}^{\infty} \left[a_n \int_{-L}^{L} \sin\left(\frac{k\pi x}{L}\right) \cos\left(\frac{n\pi x}{L}\right) dx + b_n \int_{-L}^{L} \sin\left(\frac{k\pi x}{L}\right) \sin\left(\frac{n\pi x}{L}\right) dx\right]
\end{aligned}$$

Using property 2 the terms with a_n disappear,

$$\int_{-L}^{L} \sin\left(\frac{k\pi x}{L}\right) f(x)dx = \sum_{n=0}^{\infty} \left[b_n \int_{-L}^{L} \cos\left(\frac{k\pi x}{L}\right) \cos\left(\frac{n\pi x}{L}\right) dx\right]$$

Using property 3 all terms for which $n \neq k$ disappear, leaving behind,

$$\int_{-L}^{L} \sin\left(\frac{k\pi x}{L}\right) f(x)dx = b_k L$$

$$\implies \boxed{b_k = \frac{1}{L}\int_{-L}^{L} \sin\left(\frac{k\pi x}{L}\right) f(x)dx} \implies \qquad \boxed{b_0 = 0} \tag{6}$$

- If the function is even $b_n = 0$
- If the function is odd $a_0 = 0$, $a_n = 0$

α.

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

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

The coefficients are calculated as:

$$\begin{aligned}
a_0 &= \frac{2}{2}\int_{-1}^{1} f(x)dx \\
a_0 &= \int_{-1}^{0} f(x)dx + \int_{0}^{1} f(x)dx \\
a_0 &= \int_{0}^{1} 1dx \\
a_0 &= 1
\end{aligned}$$

$$
\begin{aligned}
a_n &= \frac{2}{2}\int_{-1}^{1} f(x)\cos\left(\frac{2n\pi x}{2}\right)dx \\
a_n &= \int_{-1}^{1} f(x)\cos\left(n\pi x\right)dx \\
a_n &= \int_{-1}^{0} f(x)\cos\left(n\pi x\right)dx + \int_{0}^{1} f(x)\cos\left(n\pi x\right)dx \\
a_n &= \int_{0}^{1} 1.\cos\left(n\pi x\right)dx \\
a_n &= 0
\end{aligned}
$$

$$
\begin{aligned}
b_n &= \frac{2}{2}\int_{-1}^{1} f(x)\sin\left(\frac{2n\pi x}{2}\right)dx \\
b_n &= \int_{-1}^{1} f(x)\sin\left(n\pi x\right)dx \\
b_n &= \int_{-1}^{0} f(x)\sin\left(n\pi x\right)dx + \int_{0}^{1} f(x)\sin\left(n\pi x\right)dx \\
b_n &= \int_{0}^{1} 1.\sin\left(n\pi x\right)dx \\
b_n &= -\frac{\cos n\pi x}{n\pi}\Bigg|_{0}^{1} \\
b_n &= \frac{1}{n\pi}[\cos 0 - \cos n\pi] \\
b_n &= \frac{1}{n\pi}[1-(-1)^n]
\end{aligned}
$$

Therefore the Fourier series of this function is given as:

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

β.

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

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

Since this is an even function $b_n = 0$, the coefficients are calculated as:

$$
\begin{aligned}
a_0 &= \frac{2}{2}\int_{-1}^{1} f(x)dx \\
a_0 &= \int_{-1}^{-0.5} f(x)dx + \int_{-0.5}^{0.5} f(x)dx + \int_{0.5}^{1} f(x)dx \\
a_0 &= \int_{-0.5}^{0.5} 1dx \\
a_0 &= 1
\end{aligned}
$$

$$a_n = \frac{2}{2}\int_{-1}^{1} f(x)\cos\left(\frac{2n\pi x}{2}\right)dx$$
$$a_n = \int_{-1}^{1} f(x)\cos(n\pi x)dx$$
$$a_n = \int_{-1}^{-0.5} f(x)\cos(n\pi x)dx + \int_{-0.5}^{0.5} f(x)\cos(n\pi x)dx + \int_{0.5}^{1} f(x)\cos(n\pi x)dx$$
$$a_n = \int_{-0.5}^{0.5} 1\cos(n\pi x)dx$$
$$a_n = \frac{\sin n\pi x}{n\pi}\Bigg|_{-0.5}^{0.5}$$
$$a_n = \frac{2}{n\pi}\sin\frac{n\pi}{2}$$

Therefore the Fourier series of this function is given as:

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

γ.

$$f(x) = \begin{cases} -0.5, & -1 < x < 0 \\ 0.5, & 0 < x < 1 \end{cases}$$
$$f(x+2) = f(x)$$

Since this is an odd function $a_0, a_n = 0$, the coefficients are calculated as:

$$b_n = \frac{2}{2}\int_{-1}^{1} f(x)\sin\left(\frac{2n\pi x}{2}\right)dx$$
$$b_n = \int_{-1}^{1} f(x)\sin(n\pi x)dx$$
$$b_n = \int_{-1}^{0} f(x)\sin(n\pi x)dx + \int_{0}^{1} f(x)\sin(n\pi x)dx$$
$$b_n = (-0.5)\frac{-\cos n\pi x}{n\pi}\Bigg|_{-1}^{0} + (0.5)\frac{cosn\pi x}{n\pi}\Bigg|_{0}^{1}$$
$$b_n = \frac{1}{2n\pi}[1-(-1)^n] - \frac{1}{2n\pi}[(-1)^n - 1]$$
$$b_n = \frac{1}{n\pi}[1-(-1)^n]$$

Therefore the Fourier series of this function is given as:

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

Even Functions

A function is an even function if $f(x)$ is equal to $f(-x)$ for all the values of x. This means that the function is the same for the positive x-axis and the negative x-axis, or graphically, symmetric about the y-axis.

Odd Functions

The odd functions are functions that return their negative inverse when x is replaced with $-x$. This means

that $f(x)$ is an odd function when $f(-x) = -f(x)$.The graph of an odd function will be symmetrical about the origin.

- α is neither even nor odd
- β is even
- γ is odd

Gibbs Phenomenon

The Gibbs phenomenon was discovered by Henry Wilbraham in 1848 and then rediscovered by J. Willard Gibbs in 1899.

For a periodic signal with discontinuities, if the signal is reconstructed by adding the Fourier series, then overshoots appear around the edges. These overshoots decay outwards in a damped oscillatory manner away from the edges. This is known as Gibbs phenomenon and is shown in the figure below.

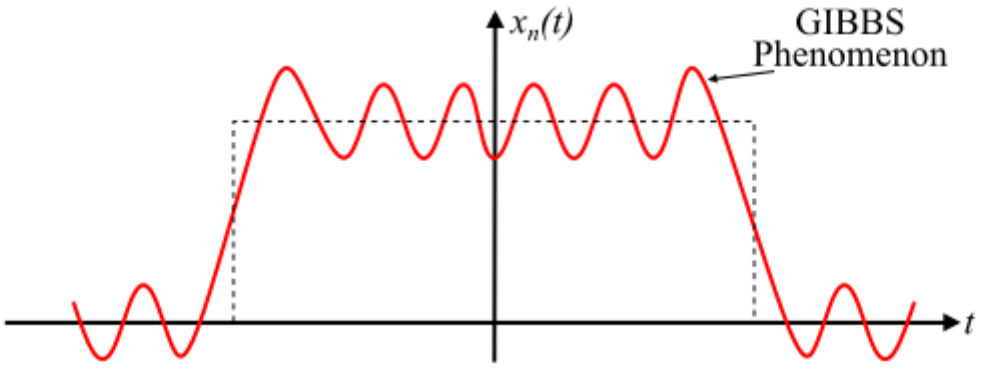

Figure 1: Gibbs Phenomenon

The amount of the overshoots at the discontinuities is proportional to the height of discontinuity and according to Gibbs, it is found to be around 9% of the height of discontinuity irrespective of the number of terms in the Fourier series. The exact proportion is given by the Wilbraham-Gibbs Constant.

$$\frac{1}{\pi}\int_0^{\pi}\frac{\sin t}{t}dt - \frac{1}{2} = 0.089489...$$

It may also be noted that as more number of terms in the series are added, the frequency increases and the overshoots become sharper, but the amplitude of the adjoining oscillation reduces, i.e., the error between the original signal $x(t)$ and the truncated signal $x_n(t)$ reduces except at edges as the n increases. Hence, the truncated Fourier series approaches the original signal $x(t)$ as the number of terms in approximation increases.

Half Range Expansion

Property of even and odd functions

1. Let $f(x)$ be a Odd function

$$a_0 = 0 \tag{7}$$

$$a_k = \frac{1}{L}\int_{-L}^{L} \underbrace{\cos\left(\frac{k\pi x}{L}\right) f(x)}_{\text{odd function}}\, dx = 0 \tag{8}$$

$$b_k = \frac{1}{L}\int_{-L}^{L} \underbrace{\sin\left(\frac{k\pi x}{L}\right) f(x)}_{\text{even function}}\, dx = \frac{2}{L}\int_{0}^{L} \sin\left(\frac{k\pi x}{L}\right) f(x)dx \tag{9}$$

2. Let $f(x)$ be a Even function

$$a_0 = \frac{1}{L}\int_{0}^{L} f(x)dx \tag{10}$$

$$a_k = \frac{1}{L}\int_{-L}^{L} \underbrace{\cos\left(\frac{k\pi x}{L}\right) f(x)}_{\text{Even function}}\, dx = \frac{2}{L}\int_{0}^{L} \cos\left(\frac{k\pi x}{L}\right) f(x)dx \tag{11}$$

$$b_k = \frac{1}{L}\int_{-L}^{L} \underbrace{\sin\left(\frac{k\pi x}{L}\right) f(x)}_{\text{Odd function}}\, dx = 0 \tag{12}$$

Then we can represent a odd function using only the terms consisting of sin and even function using only the terms consisting of cos.

Odd and Even expansion

If we are given a function $f(x)$ in a interval $[0, L]$,Clearly the function is not periodic but we want to represent the function in terms of Fourier series,We have two choices a Even Extension of the original function and a Odd Extension of the original function.i.e,
If a function is defined in half range say $[0, L]$ instead of $[-L, L]$,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 Fourier series.

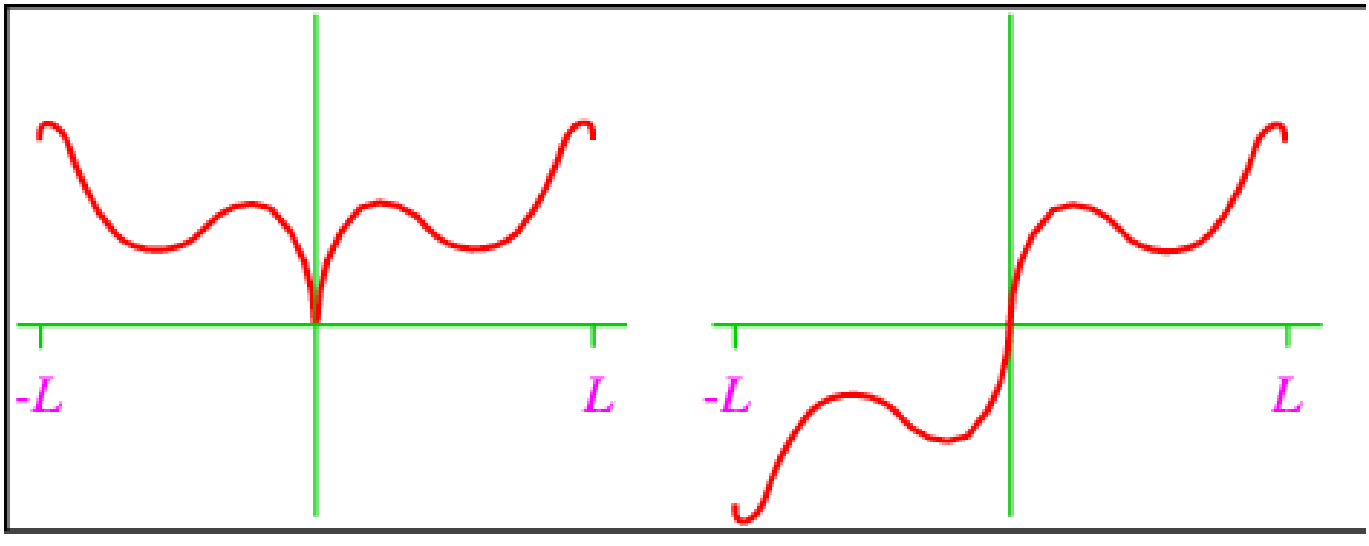

Figure 2: Even and odd extension from left to right

This even and odd extension of the original function helps us form the fourier expansion of the function,Since the original function is defined only in the interval $[0, L]$,We only integrate over this interval,see the equation 9,11,12

Half range expansion of $f(x) = x$, $x \in [0, \pi]$

1. Even periodic extension

$$a_0 = \frac{1}{\pi}\int_0^\pi f(x)dx = \frac{1}{\pi}\int_0^\pi xdx \tag{13}$$

$$a_0 = \frac{\pi}{2} \tag{14}$$

$$a_n = \frac{2}{\pi}\int_0^\pi \cos(nx)xdx \tag{15}$$

$$a_n = \frac{2((-1)^n - 1)}{\pi n^2} \tag{16}$$

Therefore the function is given by -

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

2. Odd periodic extension

$$b_n = \frac{2}{\pi}\int_0^\pi \sin(nx)xdx \tag{17}$$

$$b_n = \frac{2(-1)^{n+1}}{n} \tag{18}$$

Therefore the function is given by-

$$f(x) = \sum_{n=1}^{\infty} \frac{2(-1)^{n+1}}{n}\sin(nx)$$

2 Algorithm

Algorithm 1 Use Fourier series to represent piecewise continuous functions

procedure FOURIERCOEFF(f,L,N,method,$func\_flag$,d)

Input: f is the function which is to be represented as a fourier series,L is the half period of the given periodic function,N is the no. of terms whose fourier coefficient is to be determined,method is the name of the method that is to be used for integration,$func\_flag$ takes input one of 3 variables which tells us if the function is even or odd or nor even neither odd,and d is the number of significant digits required in the numerical approximation of the integral.

Output: Returns,a_0, a_n, b_n

▷ We set flags for the method used and type of the function,and check if the input flag is present in our defined flag,if the input flag is present then we calculate the only the coefficient required for that function.

$methods \leftarrow$ A dictionary that stores the key for the method used.
$func\_type \leftarrow$ A tuple that contains three variable 0,1,-1,for the evenness of the function.
$a_n, b_n \leftarrow$ A vector of zeros of length of N
if method not in methods and func_flag not in func_type **then**
  Return Error in parsing the flags
$Integrate \leftarrow$ it has the function corresponding to the method key in methods
for $i = 1, 2, 3 \dots N$ **do**
  if func_flag is 0 or -1 **then**

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

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

  if func_flag is 1 or -1 **then**

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

Return a_0, a_n, b_n
EXIT

Note:
Gaussian quadrature method is far superior method of integration than simpson 1/3 and trapezoidal method because of its unique technique of determining the weights and x points,we can easily integrate the piecewise continuous functions like shown above,if we choose the subinterval to be the interval where function is defined and continuous,we don't require function to be defined at its endpoints,where there might be any discontinuity.
Further the output wave form of a **full wave rectifier** is very similar to curve **|sin(x)|**,Since we know that simpson and trapezoidal method can accurately integrate upto 3rd order and 2nd order respectively.Whereas gaussian quadrature method can integrate accurately upto polynomial of degree 2n-1 in its n-point formula. Hence it is a better choice.

3 Programming

```python
from Myintegration import *

def FourierCoeff(func,n,L,d=None,method="quad",even_flag=-1,hr_flag = 0):
    methods = {"trap":MyTrap,"simp":MySimp,"quad":MyLegQuadrature}
    eveness = (0,1,-1)
    if method not in methods.keys():
        raise ValueError(f"Passed method not found, select method from one of {methods} ")
    if even_flag not in eveness:
        raise ValueError(f"Passed even_flag not found, select flag value from {eveness} ")

    n_arr = np.arange(0,n)
    a_n=np.zeros(n)
    b_n=np.zeros(n)
    inte = methods[method]

    uppr_limit = 2*L
    coef_div = L
    if hr_flag == 1:
        uppr_limit/=2
        coef_div /= 2

    calc_a_n = np.vectorize(lambda k: (1/coef_div)*inte(lambda x : np.cos(k*np.pi*x/L)*func(x),
    a=0,b=uppr_limit,d=d)[0])
    calc_b_n = np.vectorize(lambda k: (1/coef_div)*inte(lambda x : np.sin(k*np.pi*x/L)*func(x),
    a=0,b=uppr_limit,d=d)[0])

    if even_flag == 0 or even_flag ==-1:
        a_n = calc_a_n(n_arr)
    if even_flag == 1 or even_flag ==-1:
        b_n = calc_b_n(n_arr)

    a_n[0] /= 2
    return(a_n,b_n)

def Partials(func,n,L,d,method="quad",even_flag=-1,hr_flag=0):
    cosnx = lambda x,i : np.cos(i*np.pi*x/L)
    sinnx = lambda x,i : np.sin(i*np.pi*x/L)

    a_i,b_i = FourierCoeff(func,n,L,d,method,even_flag,hr_flag)
    print(a_i,b_i)
    na = np.arange(0,len(a_i))
    nb = np.arange(1,len(b_i))
    return (np.vectorize(lambda x :np.sum(cosnx(x,na)*(a_i))+np.sum(sinnx(x,nb)*(b_i[1:]))))

def main():
    terms_arr = [1,2,5,10,20]
    def p1(x):
        if -1<=x and x<=0:
            return(0)
        elif 0<=x and x<=1:
            return(1)
        elif x>1 :
            return(p1(x-2))
        elif x<-1 :
            return(p1(x+2))
```

```
    def p2(x):
        if -1<=x and x<=-0.5:
            return(0)
        elif -0.5<=x and x<=0.5:
            return(1)
        elif 0.5<=x and x<=1:
            return(0)
        elif x>1 :
            return(p2(x-2))
        elif x<-1 :
            return(p2(x+2))
    def p3(x):
        if -1<=x and x<=0:
            return(-0.5)
        elif 0<=x and x<=1:
            return(0.5)
        elif x>1 :
            return(p3(x-2))
        elif x<-1 :
            return(p3(x+2))
    p1 = np.vectorize(p1)
    p2 = np.vectorize(p2)
    p3 = np.vectorize(p3)
    #p1 = np.vectorize(lambda x: np.piecewise(x,[-1<=x and x<0,0<=x and x<=1],[lambda
    x: 0,lambda x: 1 ]))
    #p2 = np.vectorize(lambda x: np.piecewise(x,[-1<=x and x<-0.5,-0.5<=x and x
    <0.5,0.5<=x and x<=1],[lambda x: 0,lambda x: 1,lambda x: 0 ]))
    #p3 = np.vectorize(lambda x: np.piecewise(x,[-1<=x and x<0,0<=x and x<=1],[lambda
    x: -0.5,lambda x: 0.5 ]))

    x_space = np.linspace(-2.5,2.5,300)
    cols = ["x","f(x)"]
    table1 = np.array([-0.5,0,0.5])
    table2 = table1.copy()
    table3 = table1.copy()

    fig1,ax1 = plt.subplots(1,1)
    plt.plot(x_space,p1(x_space),label= "$f(x)$")
    table1 = np.column_stack((table1,p1(table1)))
    for terms in terms_arr:
        cols.extend(["$S_{"+f"{terms}"+"}$","$RE_{"+f"{terms}"+"}$"])
        f1 = Partials(p1,n=terms,L=1,d=7,method="quad",even_flag=-1)
        table1 = ad2table(table1,f1)
        plt.plot(x_space,f1(x_space),label= "$S_{"+f"{terms}"+"}$")
    ax1.legend()


    fig2,ax2 = plt.subplots(1,1)
    plt.plot(x_space,p2(x_space),label= "$f(x)$")
    table2 = np.column_stack((table2,p2(table2)))

    for terms in terms_arr:
        f2 = Partials(p2,n=terms,L=1,d=7,method="quad",even_flag=0)
        table2 = ad2table(table2,f2)
        plt.plot(x_space,f2(x_space),label= "$S_{"+f"{terms}"+"}$")
    ax2.legend()

    fig3,ax3 = plt.subplots(1,1)
    plt.plot(x_space,p3(x_space),label= "$f(x)$")
    table3 = np.column_stack((table3,p3(table3)))

    for terms in terms_arr:
```

```
        f3 = Partials(p3,n=terms,L=1,d=7,method="quad",even_flag=-1)
        table3 = ad2table(table3,f3)
        plt.plot(x_space,f3(x_space),label= "$S_{"+f"{terms}"+"}$")
    ax3.legend()

    dfs = []
    for i in range(1,4) :
        dfs.append(pd.DataFrame(eval(f"table{i}"),columns=cols))
        dfs[i-1].to_csv(f"p{i}.csv")
        print(dfs[i-1])

    plt.show()

def ad2table(t,f):
    t = np.column_stack((t,f(t[:,0])))
    t = np.column_stack((t,(f(t[:,0])-t[:,1])/t[:,1]))
    return t

if __name__=="__main__":
    from scipy.fft import fft
    import matplotlib.pyplot as plt
    from matplotlib import use
    import pandas as pd
    use("WebAgg")
    plt.style.use("bmh")
    main()
```

```
from A3a_2020PHY1114 import *


def main():
    def p4(x):
        if 0<=x and x<=np.pi:
            return(x)

    p4 = np.vectorize(p4)
    terms_arr = [1,2,5,10,20]

    x_space = np.linspace(-3*np.pi,3*np.pi,300)

    cols = ["x","f(x)"]
    table1 = np.array([0,np.pi/2,np.pi])
    table2 = table1.copy()

    fig1,ax1 = plt.subplots(1,1)
    ax1.plot(x_space,p4(x_space),label= "$f(x)$")
    table1 = np.column_stack((table1,p4(table1)))
    for terms in terms_arr:
        cols.extend(["$S_{"+f"{terms}"+"}$","$RE_{"+f"{terms}"+"}$"])
        f4 = Partials(p4,n=terms,L=np.pi,d=7,method="quad",even_flag=0,hr_flag=1)
        table1 = ad2table(table1,f4)
        ax1.plot(x_space,f4(x_space),label= "$S_{"+f"{terms}"+"}$",marker="1")
    ax1.legend()

    fig2,ax2 = plt.subplots(1,1)
    ax2.plot(x_space,p4(x_space),label= "$f(x)$")
    table2 = np.column_stack((table2,p4(table2)))
    for terms in terms_arr:
        f4 = Partials(p4,n=terms,L=np.pi,d=7,method="quad",even_flag=1,hr_flag=1)
        table2 = ad2table(table2,f4)
        ax2.plot(x_space,f4(x_space),label= "$S_{"+f"{terms}"+"}$",marker="1")
    ax2.legend()

```

```
    dfs = []
    for i in range(1,3) :
        dfs.append(pd.DataFrame(eval(f"table{i}"),columns=cols))
        dfs[i-1].to_csv(f"h{i}.csv")
        print(dfs[i-1])

    plt.show()


if __name__ == "__main__":
    import matplotlib.pyplot as plt
    from matplotlib import use
    import pandas as pd
    use("WebAgg")
    plt.style.use("bmh")
    main()
```

4 Discussion

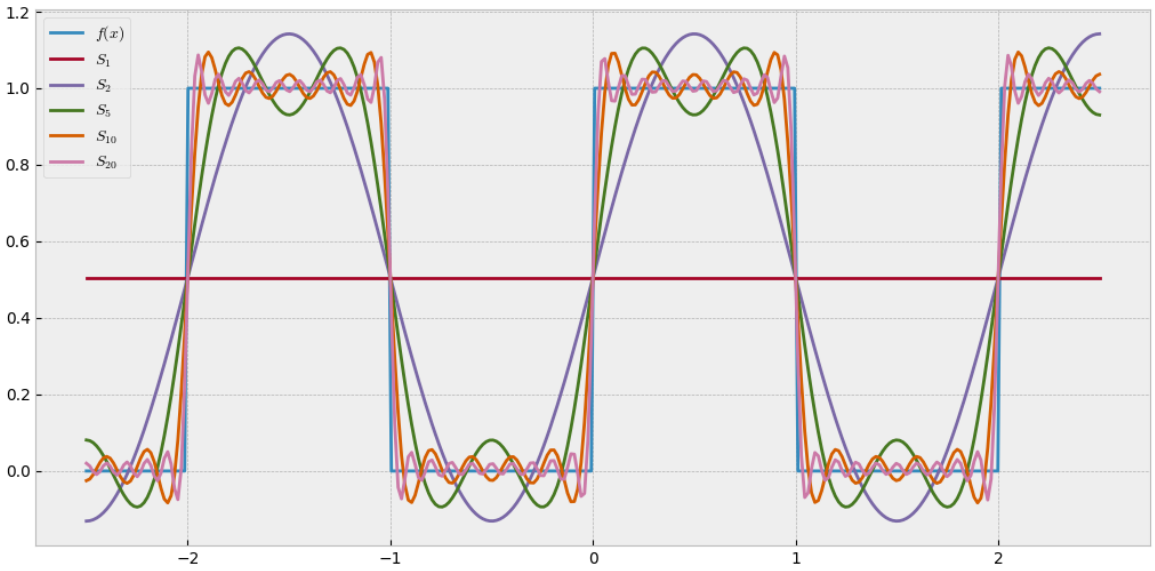

Figure 3: Function 1

	x	f(x)	S_1	RE_1	S_2	RE_2	S_5	RE_5	S_{10}	RE_{10}	S_{20}	RE_{20}
0	-0.5	0.0	0.5056271733868211	inf	-0.1308595846453956	-inf	0.08094769222639486	inf	-0.02522926424989791	-inf	0.02012627673017864	inf
1	0.0	1.0	0.5056271733868211	-0.4943728266131894	0.5056271733868211	-0.4943728266131894	0.5056271733868204	-0.4943728266131796	0.505627173386 82	-0.4943728266131805	0.5056271733868183	-0.4943728266131817
2	0.5	1.0	0.5056271733868211	-0.4943728266131894	1.1421139314190376	0.1421139314190376	0.9303066545472473	-0.06969334545275274	1.0364836110235391	0.03648361102353914	0.991128070043463	-0.008871929956536961

Table 1

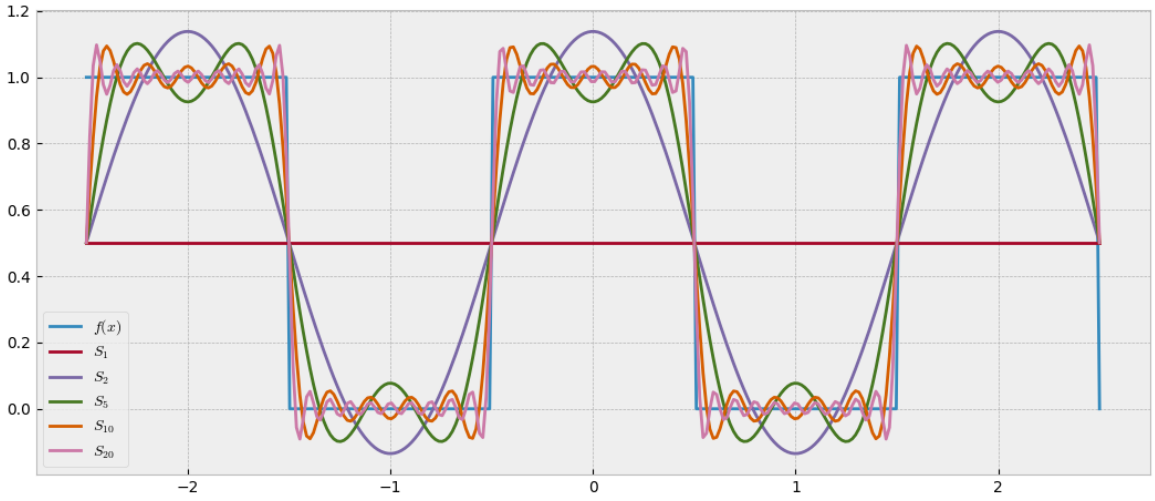

Figure 4: Function 2

	x	f(x)	S_1	RE_1	S_2	RE_2	S_5	RE_5	S_{10}	RE_{10}	S_{20}	RE_{20}
0	-0.5	1.0	0.5015687104105051	-0.4984312895894949	0.5015687104105051	-0.4984312895894949	0.501568710410505	-0.498431289589495	0.5015687104105055	-0.49843128958949445	0.5015687104105044	-0.4984312895894956
1	0.0	1.0	0.5015687104105051	-0.4984312895894949	1.1382825643618426	0.1382825643618426	0.9257934726589976	-0.07420652734100242	1.0335712725368005	0.03357127253680048	0.9847076330744672	-0.015292366925532819
2	0.5	0.0	0.5015687104105051	inf	0.5015687104105051	inf	0.501568710410505	inf	0.5015687104105055	inf	0.5015687104105044	inf

Table 2

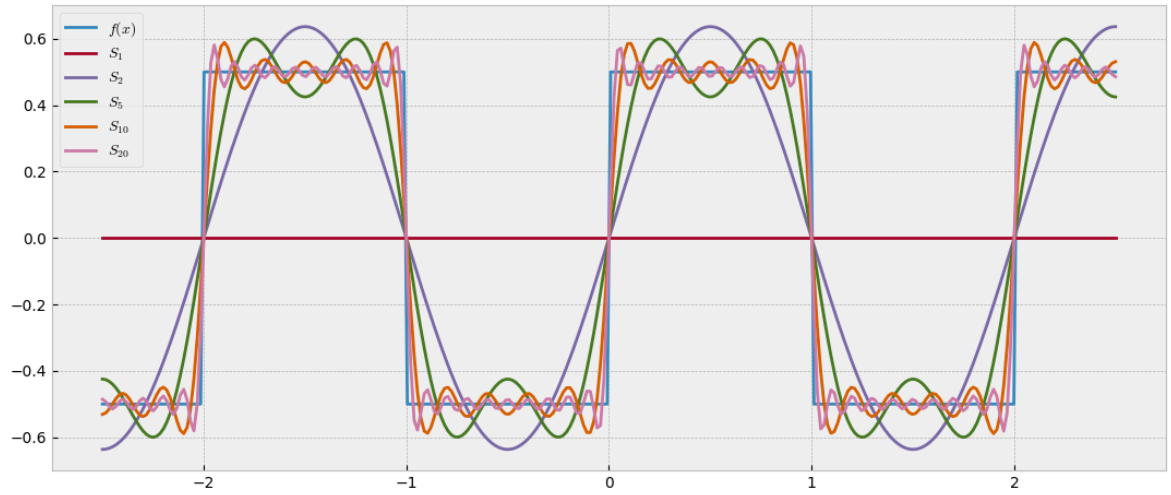

Figure 5: Function 3

	x	f(x)	S_1	RE_1	S_2	RE_2	S_5	RE_5	S_{10}	RE_{10}	S_{20}	RE_{20}
0	-0.5	-0.5	0.0	-1.0	-0.6364867580322167	0.2729735160644333	-0.4246794811604259	-0.15064103767914816	-0.5308564376367183	0.06171287527343661	-0.48550089665664253	-0.028998206686714934
1	0.0	-0.5	0.0	-1.0	6.071532165918825e-17	-1.0000000000000002	1.97758476261356e-16	-1.0000000000000004	1.1032841307212493e-15	-1.0000000000000022	-1.366962099069724e-15	-0.9999999999999972
2	0.5	0.5	0.0	-1.0	0.6364867580322167	0.2729735160644333	0.4246794811604265	-0.15064103767914705	0.5308564376367187	0.061712875273437495	0.4855008966566421	-0.02899820668671582

Table 3

- Note that in all the three functions we see there are finite discontinuities at 0,-0.5 or 0.5, we see that the fourier expansion of these piecewise continuous function attain the average value at the discontinuity as theorized by the dirichlet theorem.
  This fact is evident from table 1,2,3 also.
- As the value of n increases the fourier series goes on to become a better approximation of the function.
- Gibbs phenomenon as described in the theory 1 is also evident from the graphs,as we increase the no. of terms in partial fourier series of the function the we can observe the kinks at the edges.
- We can speculate just from the estimation of fourier series for these piecewise continuous function that,how powerful a tool is the fourier series,we can approximate any physical vibrations in terms of sine and cosine only,which is very handy because we have studied the simple harmonic motion of waves which can be represented in terms of sine or cosine.
- Note that function 3 is a odd function hence its fourier coefficient are non-zero only for sine terms,as we can see that $a_0 = 0$ since S1 is a constant line at $y = 0$.
- Note that at the discontinuities all the partial series of the fuction satisfies the Dirichlet theoram.

Part(b)

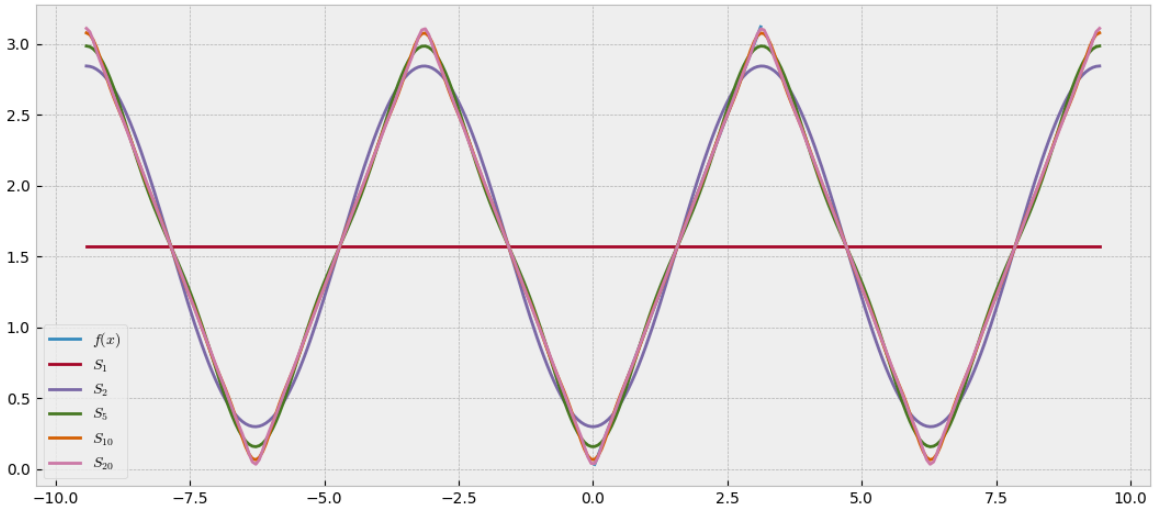

Figure 6: Even half range extension of f(x) = x,using only fourier series of cos

	x	f(x)	S_1	RE_1	S_2	RE_2	S_5	RE_5	S_{10}	RE_{10}	S_{20}	RE_{20}
0	0.0	0.0	1.5707963267948966	inf	0.29755678205973446	inf	0.15608572153360545	inf	0.06345265251427483	inf	0.03180455491101664	inf
1	1.5707963267948966	1.5707963267948966	1.5707963267948966	0.0	1.5707963267948966	0.0	1.5707963267948963	-1.4135798584282297e-16	1.5707963267948963	-1.4135798584282297e-16	1.5707963267948957	-5.654319433712919e-16
2	3.141592653589793	3.141592653589793	1.5707963267948966	-0.5	2.8440358715300587	-0.09471526543064908	2.985506932056187	-0.0496836282627704	3.078140001075517	-0.020197606599878886	3.109788098678776	-0.010123704253852039

Table 4

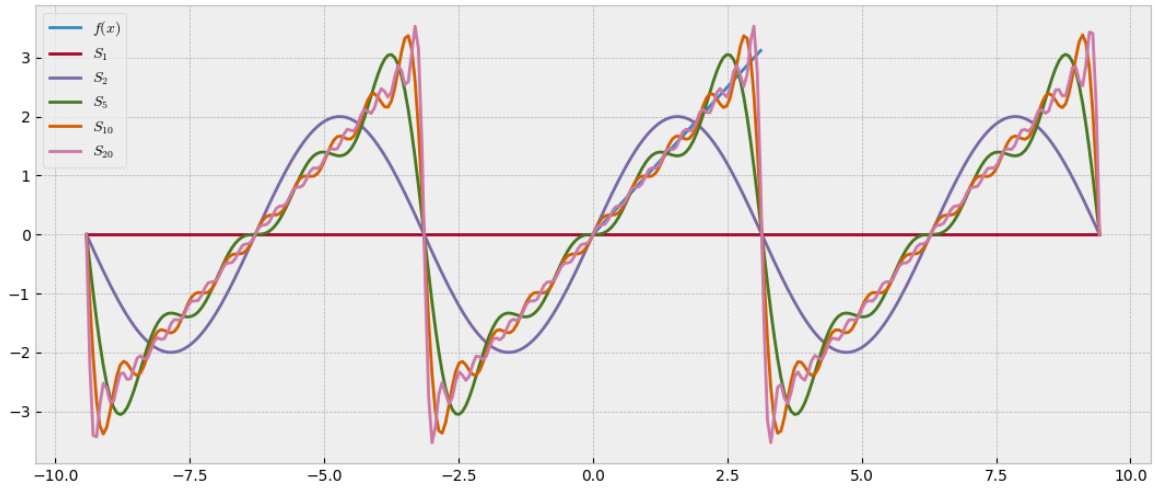

Figure 7: Odd half range extension of f(x) = x,using only fourier series of sin

	x	f(x)	S_1	RE_1	S_2	RE_2	S_5	RE_5	S_{10}	RE_{10}	S_{20}	RE_{20}
0	0.0	0.0	0.0		0.0		0.0		0.0		0.0	
1	1.5707963267948966	1.5707963267948966	0.0	-1.0	2.0	0.2732395447351627	1.333333333333333	-0.15117363684322502	1.6698412698412706	0.06305396909634269	1.5209198094647003	-0.03175237710923728
2	3.141592653589793	3.141592653589793	0.0	-1.0	2.4492935982947064e-16	-0.9999999999999999	9.797174393178824e-16	-0.9999999999999997	2.9149069742253323e-15	-0.999999999999999	6.603818617626961e-15	-0.9999999999999979

Table 5

- Similar conclusions can be drawn for the half range series. As the value of n increases the approximation becomes better and at the point of discontinuities the average value is taken.
- Even half range extension of the function does not show Gibbs phenomenon