Argan Sadan 2022PH11425

Q1)

a) $$a_n = \frac{1}{T}\int_{-T/2}^{+T/2} g(x)\exp(-i2\pi nx/T)\,dx$$

$T = 1$

$$\Rightarrow a_n = \int_{-0.5}^{+0.5} g(x)\exp(-i2\pi nx)\,dx$$

$$\Rightarrow a_n = \int_{-0.5}^{0.5} x e^{-i2\pi nx}\,dx$$

$$\Rightarrow a_n = \frac{x e^{-i2\pi nx}}{-i2\pi n}\Big|_{-0.5}^{0.5} + \frac{1}{i2\pi n}\int_{-0.5}^{+0.5} e^{-i2\pi nx}\,dx$$

$$\Rightarrow a_n = \frac{i}{2\pi n}\left(0.5e^{-i\pi n} + 0.5e^{i\pi n}\right) - \frac{1}{(i2\pi n)^2}$$

$(e^{i\pi n} - e^{-i\pi n})$

$$\Rightarrow \boxed{a_n = \frac{i\cos(\pi n)}{2\pi n} + \frac{i\sin(\pi n)}{2\pi^2 n^2}}$$

Aryan Sudan 2022PH11425

Q2)

$$g(x,y) = \text{rect}\left(\frac{x}{2b}\right) \times \text{rect}\left(\frac{y}{2a}\right)$$

$$G(f_x, f_y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} dx\,dy\, \text{rect}\left(\frac{x}{2b}\right) \text{rect}\left(\frac{y}{2a}\right) \exp[-i2\pi(f_x x + f_y y)]$$

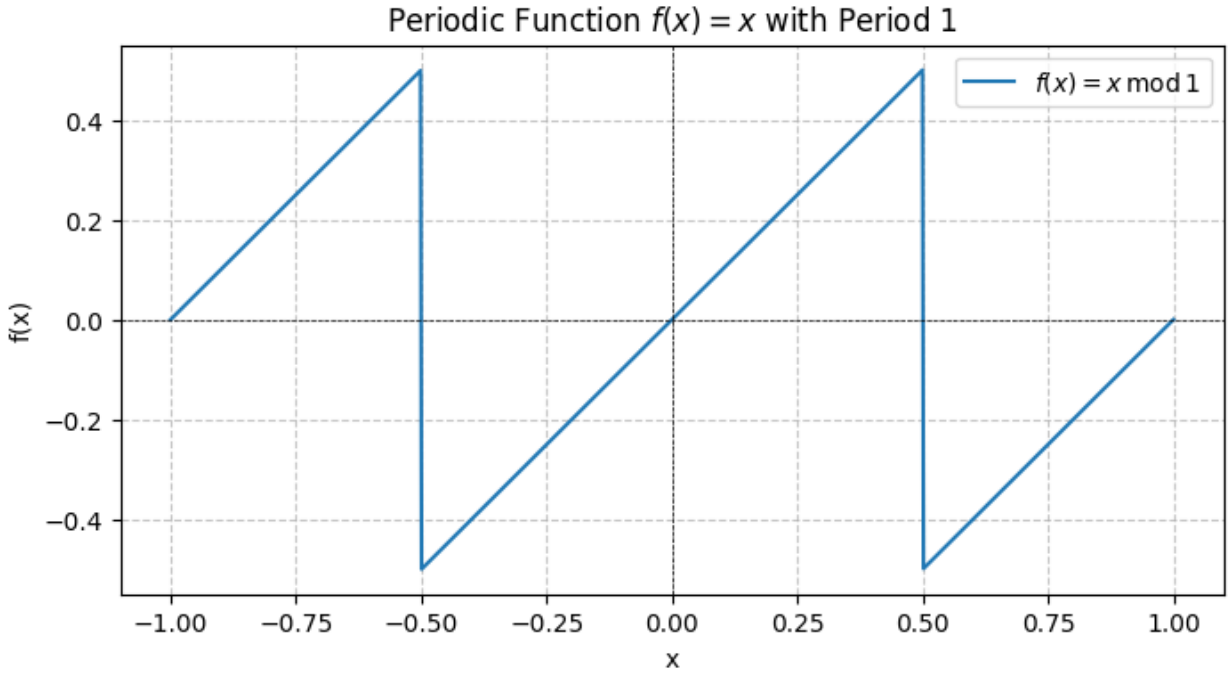
$$\Rightarrow G(f_x, f_y) = \int_{-\infty}^{\infty} dx\, \text{rect}\left(\frac{x}{2b}\right) \exp(-i2\pi f_x x) \times \int_{-\infty}^{\infty} dy\, \text{rect}\left(\frac{y}{2a}\right) \exp(i2\pi f_y y)$$

$$\Rightarrow G(f_x, f_y) = 2b\,\text{sinc}(2bf_x) \times 2a\,\text{sinc}(2af_y)$$

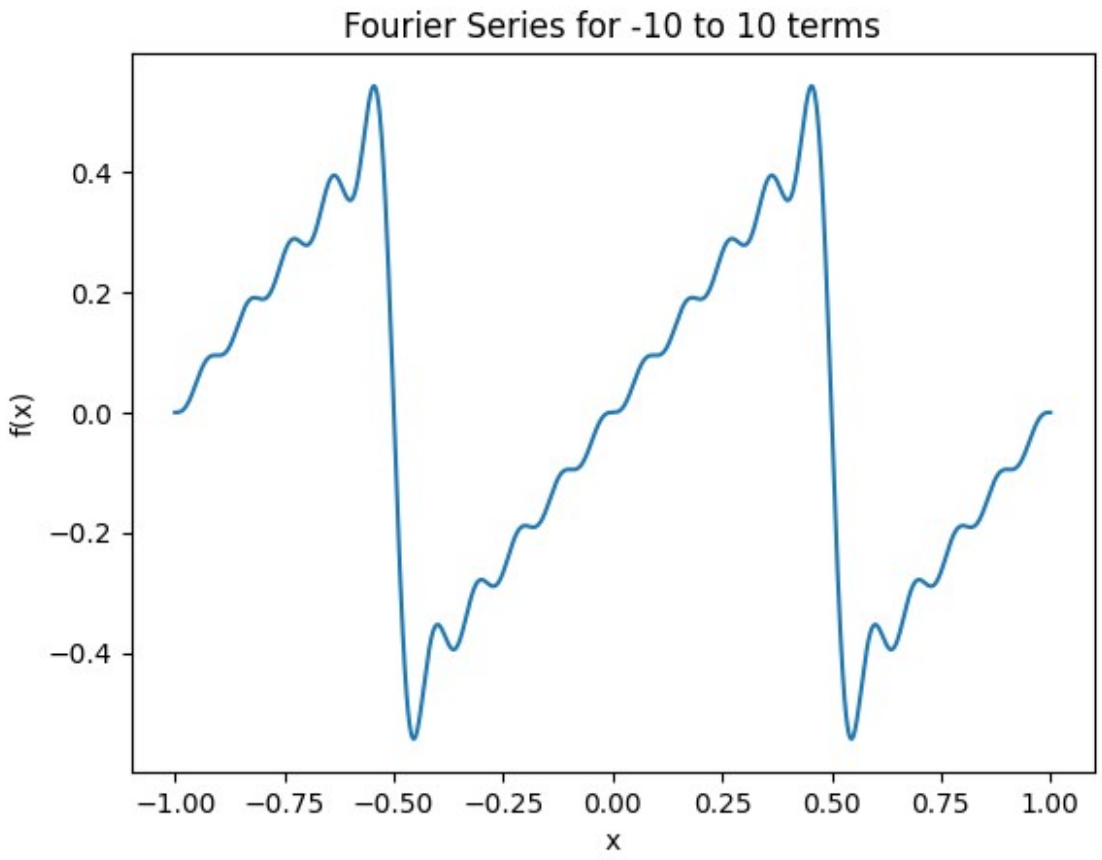
$$\Rightarrow \boxed{G(f_x, f_y) = 4ab\,\text{sinc}(2bf_x)\,\text{sinc}(2af_y)}$$

```
# This Python 3 environment comes with many helpful analytics
libraries installed
# It is defined by the kaggle/python Docker image:
https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/"
directory
# For example, running this (by clicking run or pressing Shift+Enter)
will list all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/)
that gets preserved as output when you create a version using "Save &
Run All"
# You can also write temporary files to /kaggle/temp/, but they won't
be saved outside of the current session

import matplotlib.pyplot as plt

def periodic_function(x):
    if (abs(x)<=0.5):
        return x

    decimal=x-np.floor(x)
    if (decimal<=0.5):
        return decimal
    return decimal-1



x = np.linspace(-1, 1, 1000)
y_vectorized=np.vectorize(periodic_function)
y = y_vectorized(x)

plt.figure(figsize=(8, 4))
plt.plot(x, y, label=r'$f(x) = x \, \mathrm{mod} \, 1$')
plt.axhline(0, color='black', linewidth=0.5, linestyle='--')
plt.axvline(0, color='black', linewidth=0.5, linestyle='--')
plt.title("Periodic Function $f(x) = x$ with Period 1")
plt.xlabel("x")
plt.ylabel("f(x)")
plt.grid(True, linestyle='--', alpha=0.7)
```

```
plt.legend()
plt.show()
```



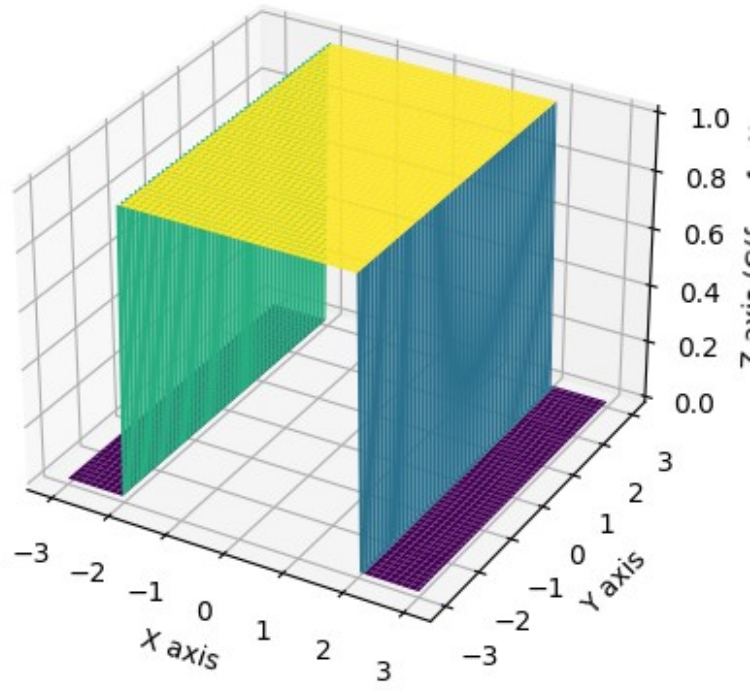

```
#  for terms -M to M
M=10
L=1
dx=0.001
x = np.linspace(-L, L, np.int32(2*L/dx)+1, endpoint=True)
s=0
for n in np.linspace(-M, M, 2*M+1, endpoint=True):
    an=0
    if (n!=0):
        an=1j*np.cos(np.pi*n)/(2*np.pi*n)
+1j*np.sin(np.pi*n)/(2*np.pi*np.pi*n*n)
    s+=an*np.exp(1j*2*np.pi*n*x)
plt.figure(1)
plt.plot(x, np.real(s))
plt.title("Fourier Series for "+str(-1*M)+" to "+str(M)+" terms")
plt.xlabel("x")
plt.ylabel("f(x)")
plt.show()
```



Question 2

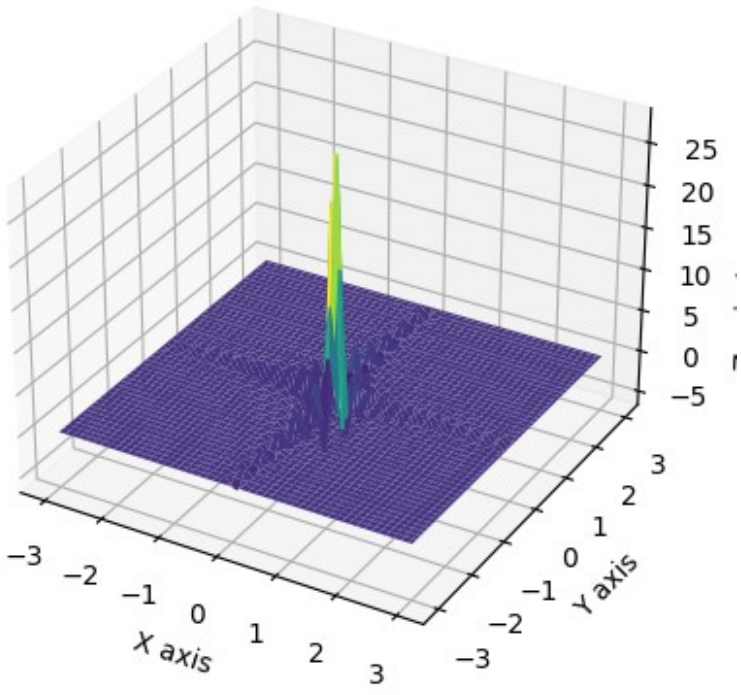
```
def rect(x, a):
    return np.where(np.abs(x) <= a, 1.0, 0.0)

def f(x, y, a, b):
    return rect(x, a)*rect(y, b)

x = np.linspace(-3, 3, 100)
y = np.linspace(-3, 3, 100)
x, y = np.meshgrid(x, y)


z = f(x, y, 2, 4)

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')


ax.plot_surface(x, y, z, cmap='viridis')


ax.set_xlabel('X axis')
ax.set_ylabel('Y axis')
```

```
ax.set_zlabel('Z axis (G(f_x, f_y))')


plt.show()
```



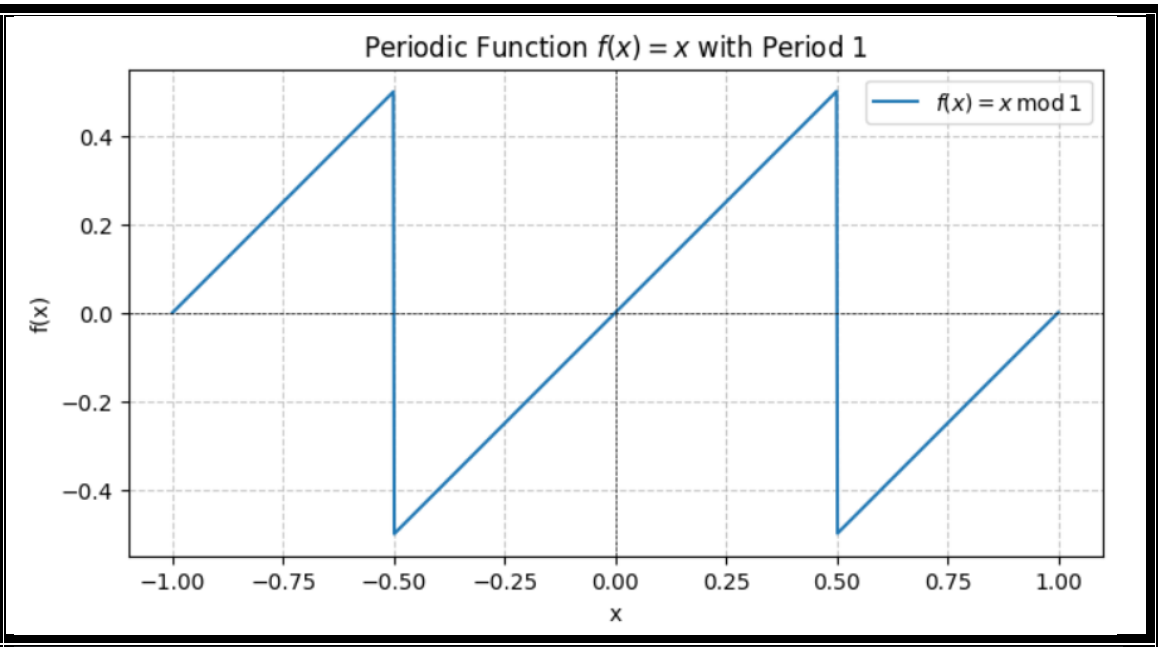
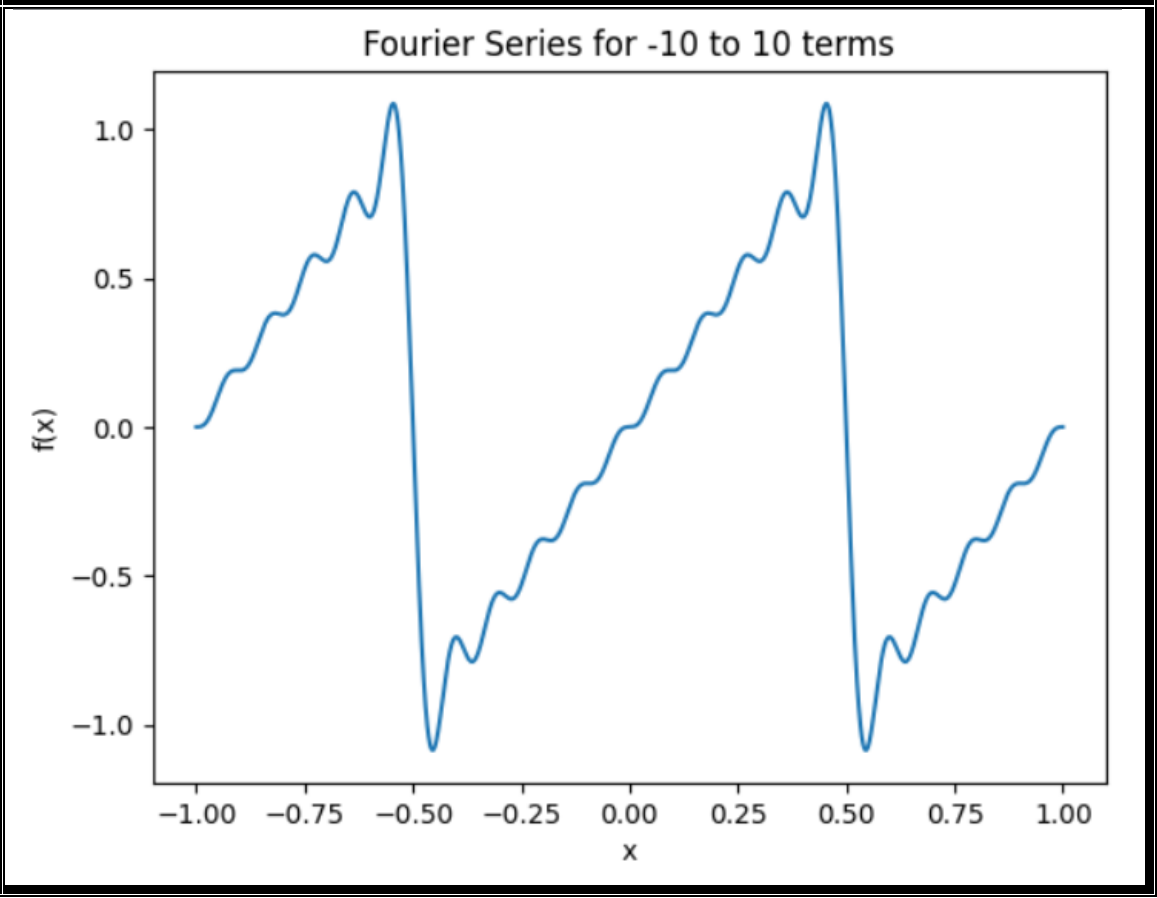

```
from mpl_toolkits.mplot3d import Axes3D

def G(f_x, f_y, a, b):
    return 4*a*b*np.sinc(2*b*f_x)*np.sinc(2*a*f_y)


x = np.linspace(-3, 3, 100)
y = np.linspace(-3, 3, 100)
x, y = np.meshgrid(x, y)


z = G(x, y, 2, 4)

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')


ax.plot_surface(x, y, z, cmap='viridis')
```

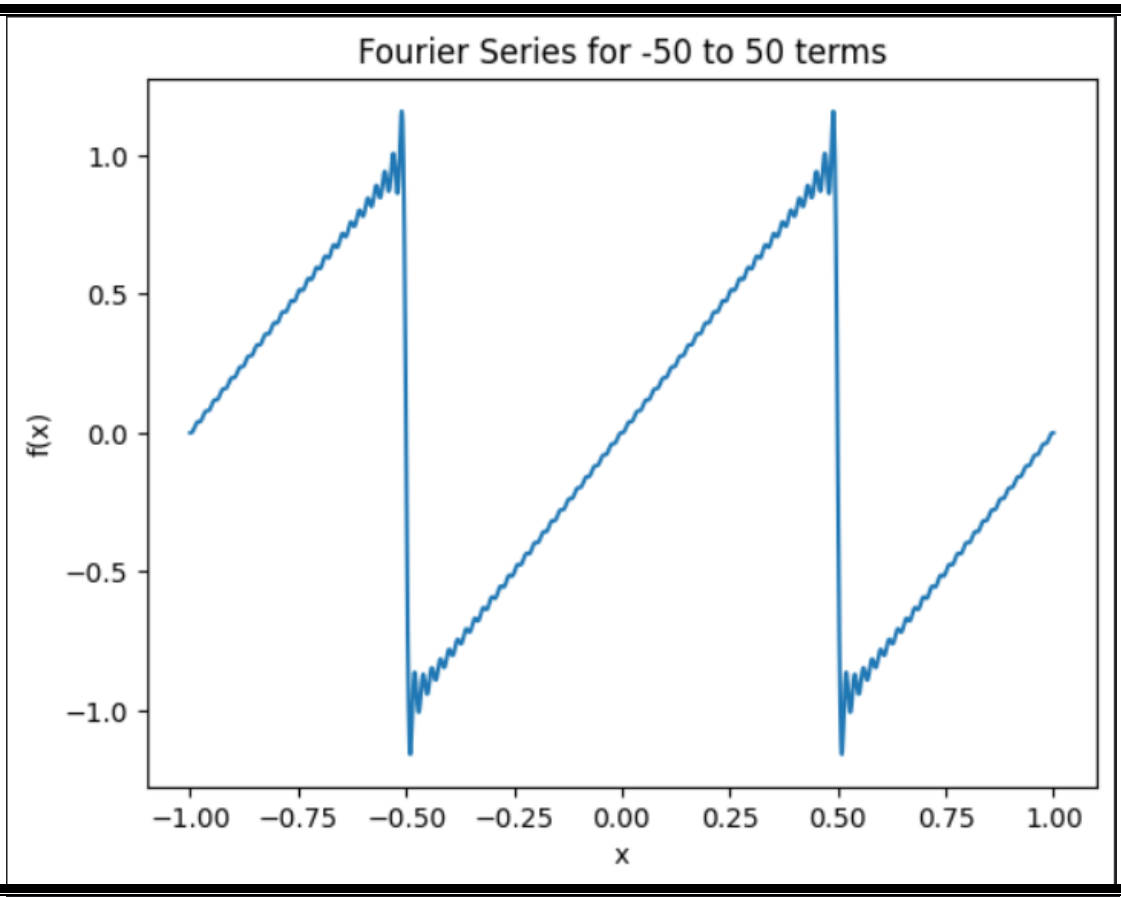
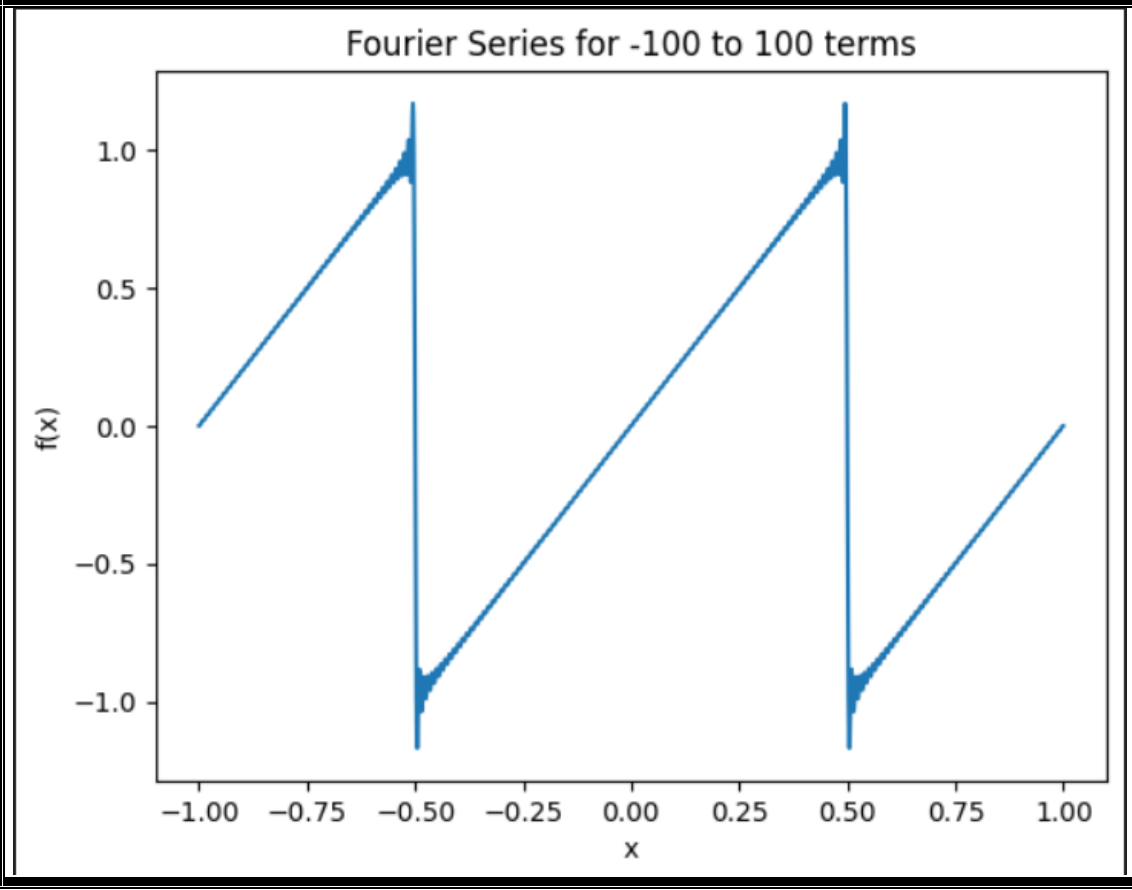
```
ax.set_xlabel('X axis')
ax.set_ylabel('Y axis')
ax.set_zlabel('Z axis (G(f_x, f_y))')

plt.show()
```



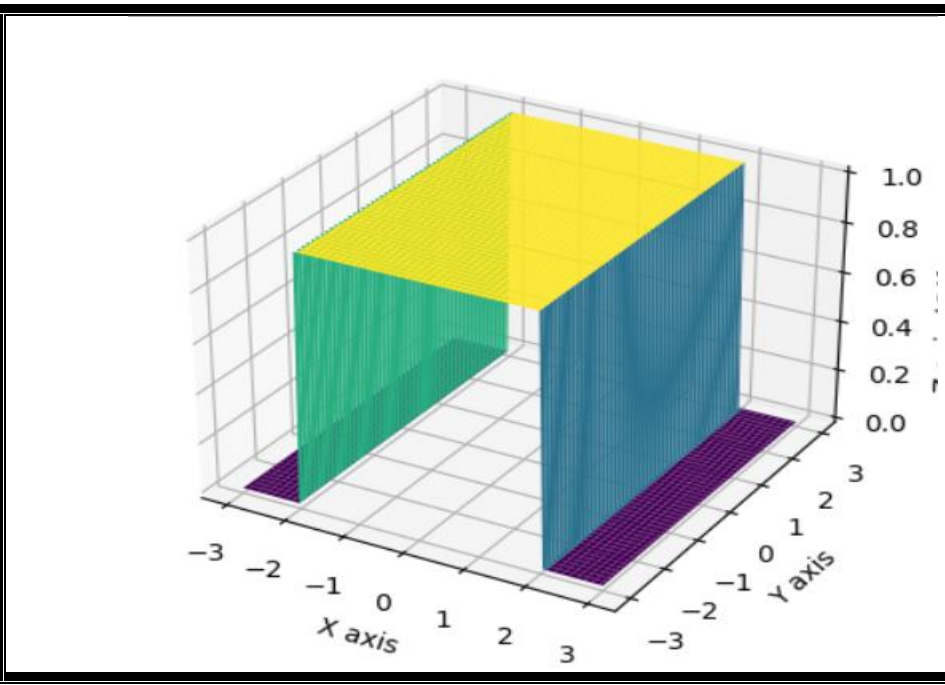
Plots

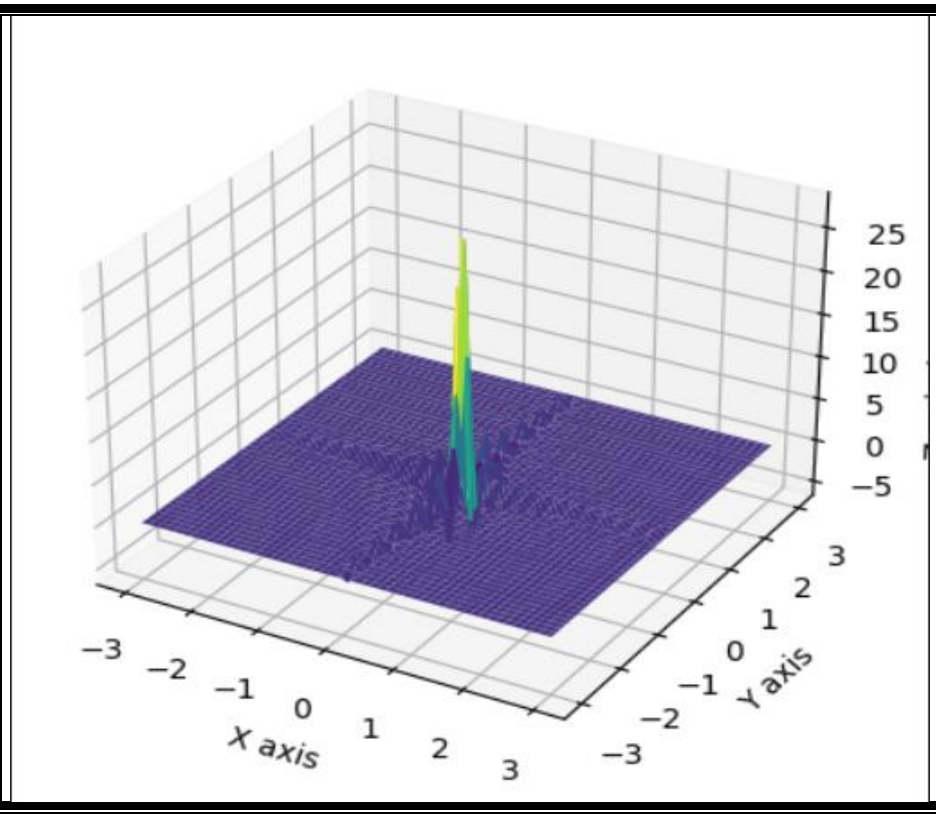
Question 1:

Original Function	
Fourier Series for f(x) for N=-10 to N=+10	

Fourier Series for f(x) for N=-50 to N=+50	
Fourier Series for f(x) for N=-100 to N=+100	

Question 2:

g(x, y)	

G(fx, fy)	 25 20 15 10 5 0 −5 −3 −2 −1 0 1 2 3 X axis −3 −2 −1 0 1 2 3 Y axis	