

```

import sys
import numpy as np
import matplotlib.pyplot as plt
from math import cos,sin,pi

fx=float(sys.argv[1])
fy=float(sys.argv[2])
ax=float(sys.argv[3])
ay=float(sys.argv[4])
phi=float(sys.argv[5])
delT=float(sys.argv[6])
N=int(sys.argv[7])
funcCall=int(sys.argv[8]) # if 0, call the lissajous function
                           # else call beats function

def withNumpy():
    n=(np.arange(1+N,dtype=float))

    x=ax*np.cos(2*pi*fx*n*delT)
    y=ay*np.sin(2*pi*fy*n*delT+phi)
    z=x+y

    temp_table=np.append([x],[y],axis=0)
    table=np.append(temp_table,[z],axis=0)
    np.savetxt('lissajous_numpy.dat',table)

def withList():
    f = open('lissajous_list.dat','w')
    z= [None]*(N+1)
    for n in range(N+1):
        x=ax*cos(2*pi*fx*n*delT)
        z[n]=x
        f.write(str(x)+' ')
    f.write('\n')
    for n in range(N+1):
        y=ay*sin(2*pi*fy*n*delT+phi)
        z[n]=z[n]+y
        f.write(str(y)+' ')
    f.write('\n')
    for nums in z:
        f.write(str(nums)+' ')
    f.write('\n')

    f.close()

def xyzPlot():

    plt.figure(figsize=(11,7))
    plt.subplot(211)
    plt.plot(n*delT,funcs[0], 'r-',label='X(t)')

```

```

plt.plot(n*delT,funcs[1],'y-',label='Y(t)')
if funcCall !=0:
    plt.plot(n*delT,funcs[2],'b-',label='Z(t)')
plt.xlabel('time (t)')
plt.legend()

def lissajous():

    plt.subplot(212)
    plt.plot(funcs[0],funcs[1],label='X(t) vs Y(t)')
    plt.xlabel('X(t)')
    plt.ylabel('Y(t)')
    plt.text(-8, -18, '$f_X$ = '+str(fx)+' , $f_Y$ = '+str(fy)+' , $A_X$
= '+str(ax)+' , $A_Y$ = '+str(ay)+' , $\phi$ = '+str(phi)+' , $\Delta$ t
= '+str(delT)+' , N = '+ str(N))
    plt.subplots_adjust(hspace=0.4)

    plt.show()

def beats():

    plt.subplot(212)
    plt.plot(n*delT,funcs[2],label='Z(t)')
    plt.xlabel('time (t)')
    plt.ylabel('Z(t)')
    plt.text(-4, -35, '$f_X$ = '+str(fx)+' , $f_Y$ = '+str(fy)+' , $A_X$
= '+str(ax)+' , $A_Y$ = '+str(ay)+' , $\phi$ = '+str(phi)+' , $\Delta$ t
= '+str(delT)+' , N = '+ str(N)+' , $\omega_1$ = '+ str(2*pi*fx)+' , $
\omega_2$ = '+ str(2*pi*fy))
    plt.subplots_adjust(hspace=0.4)

    plt.show()

withNumpy()
# withList()

funcs=np.loadtxt('lissajous_numpy.dat')
# funcs=np.loadtxt('lissajous_list.dat')

n=(np.arange(1+N,dtype=float))

xyzPlot()

if funcCall:
    lissajous()
else:
    beats()

```