APPENDIX D Plotting Graphs in Python

D.1 INTRODUCTION TO GRAPH PLOTTING

The `matplotlib` is a module used for plotting images using Python scripts, the Python and IPython shells, etc. To plot graphs, you must install the `matplotlib` module.

For details on `matplotlib` module, visit the links given below.
`https://pythonprogramming.net/matplotlib-python-3-basics-tutorial/`
`https://www.digitalocean.com/community/tutorials/how-to-plot-data-in-python-3-using-matplotlib`

Note All the programs should be written using user-defined functions, wherever possible.

Program D.1 Write a menu driven program to create mathematical 3D objects.

```
I. curve      II. sphere      III. cone
IV. arrow     V. ring         VI. cylinder

from visual import *

def show_curve():
     curve(x=arange(100), y=arange(100)**0.5,color=color.red)

def show_sphere():
     ball = sphere(pos=(1,2,1), radius=0.5)

def show_cone():
     cone(pos=(5,2,0), axis=(12,0,0),radius=2)

def show_arrow():
     pointer = arrow(pos=(0,2,1),axis=(5,0,0), shaftwidth=1)

def show_rings():
     ring(pos=(1,1,1), axis=(0,1,0), radius=0.5, thickness=0.1)
def show_cylinder():
     rod = cylinder(pos=(0,2,1),axis=(5,0,0), radius=1)

def menu():
     print('1. Curve'
     print('2. Sphere')
```

```
    print('3. Cone')
    print('4. Arrow')
    print('5. Rings')
    print('6. Cylinder')


ch = 'y'
while(ch=='y'):
    menu()
    choice = int(input('Enter choice..'))
    if(choice == 1):
        show_curve()
    elif(choice == 2):
        show_sphere()
    elif(choice == 3):
        show_cone()
    elif(choice == 4):
        show_arrow()
    elif(choice == 5):
        show_rings()
    elif(choice ==6):
        show_cylinder()
    else:
        print('Wrong choice')
    ch = input('Do you wish to continue(y/n)..')
```

OUTPUT

The desired shapes of the 3D objects will be obtained.

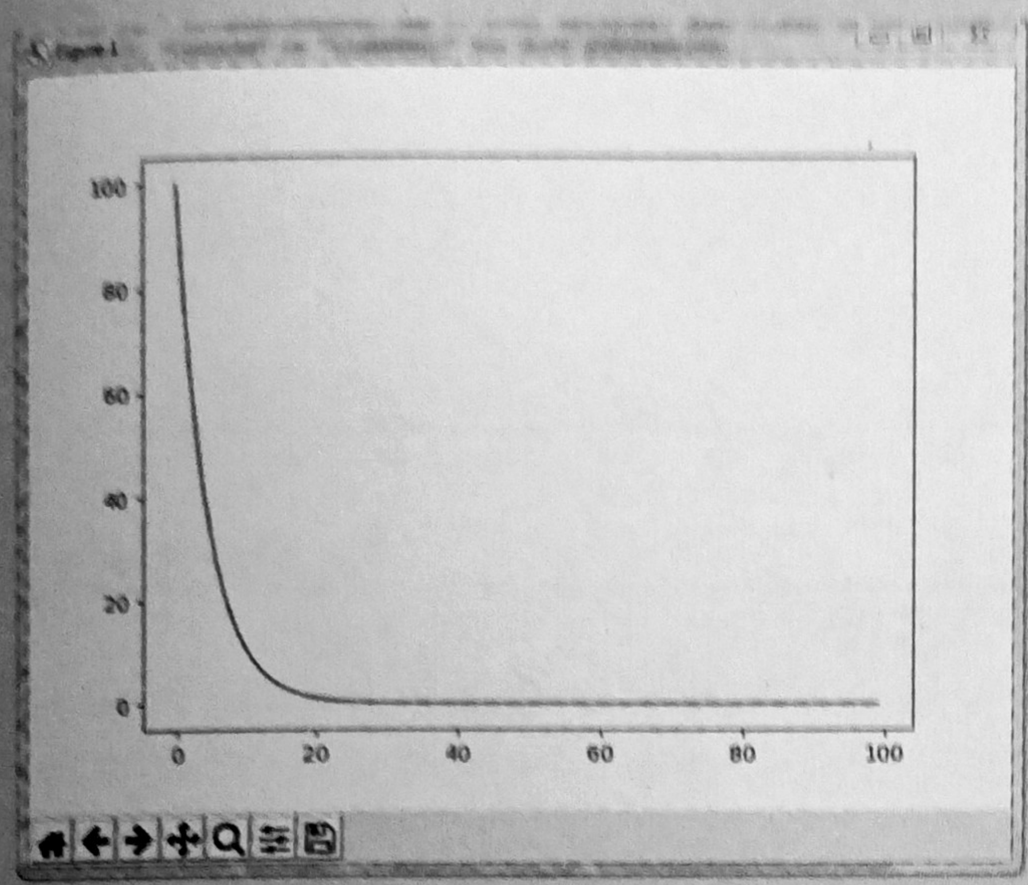
Program D.2 Write a program to read n integers and display them as histogram.

```
import matplotlib.pyplot as plt
import numpy as np
from mpl_toolkits.mplot3d import Axes3D

def sine_curve():
    Fs = 8000
    f = 5
    sample = 8000
    x = np.arange(sample)
    y = np.sin(2 * np.pi * f * x / Fs)
    plt.plot(x, y)
    plt.xlabel('voltage(V)')
    plt.ylabel('sample(n)')
    plt.show()

def cosine_curve():
```

```
    Fs = 8000
    f = 5
    sample = 8000
    x = np.arange(sample)
    y = np.cos(2 * np.pi * f * x / Fs)
    plt.plot(x, y)
    plt.xlabel('voltage(V)')
    plt.ylabel('sample(n)')
    plt.show()

def polynomial_curve():
    x = np.arange(-5, 5, 0.25)
    y = np.arange(-5, 5, 0.25)
    X, Y = np.meshgrid(x, y)
    F = 3 + 2*X + 4*X*Y + 5*X*X
    fig = plt.figure()
    ax = fig.add_subplot(111, projection='3d')
    ax.plot_surface(X, Y, F)
    plt.show()

def exponential_curve(func, x_range):
    x = np.arange(*x_range)
    y = func(x)
    plt.plot(x, y)
    plt.show()

def menu():
    print('1. Sine Curve')
    print('2. Cosine Curve')
    print('3. Polynomial Curve')
    print('4. Exponential Curve')

ch = 'y'
while(ch=='y'):
    menu()
    choice = int(input('Enter choice..'))
    if(choice == 1):
        sine_curve()
    elif(choice == 2):
        cosine_curve()
    elif(choice == 3):
        polynomial_curve()
    elif(choice == 4):
        exponential_curve(lambda x: 100*(np.power(0.8, x)), (0,100))
    else:
        print('Wrong choice')
    ch = input('Do you wish to continue(y/n)..')
```

OUTPUT

```
1. Sine Curve
2. Cosine Curve
3. Polynomial Curve
4. Exponential Curve
Enter choice..4
```



Program D.3 Input initial velocity and acceleration, and plot the following graphs depicting equations of motion:

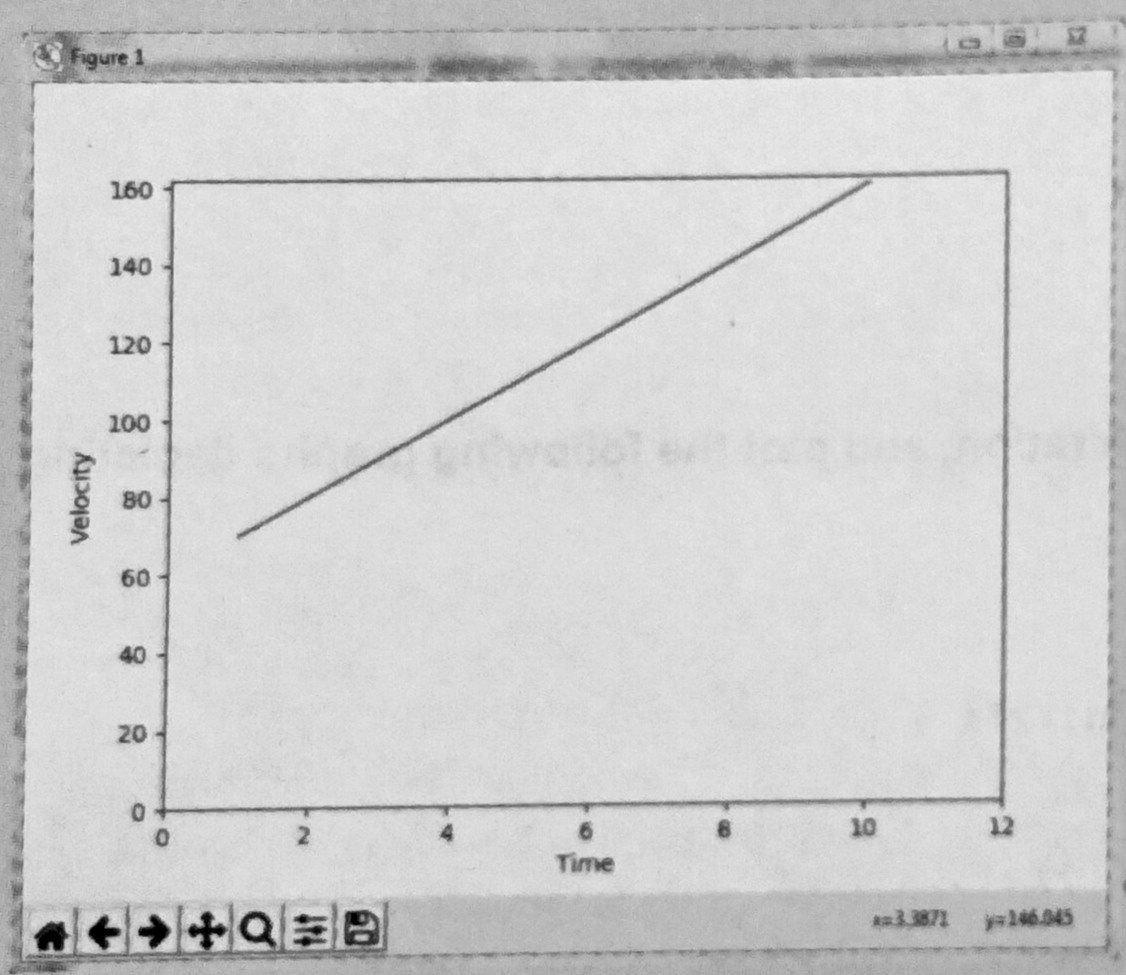
```
I. velocity w.r.t time (v=u+at)
II. distance w.r.t time (s=u*t+0.5*a*t*t)
III. distance w.r.t. velocity (s=(v*v-u*u)/2*a )
```

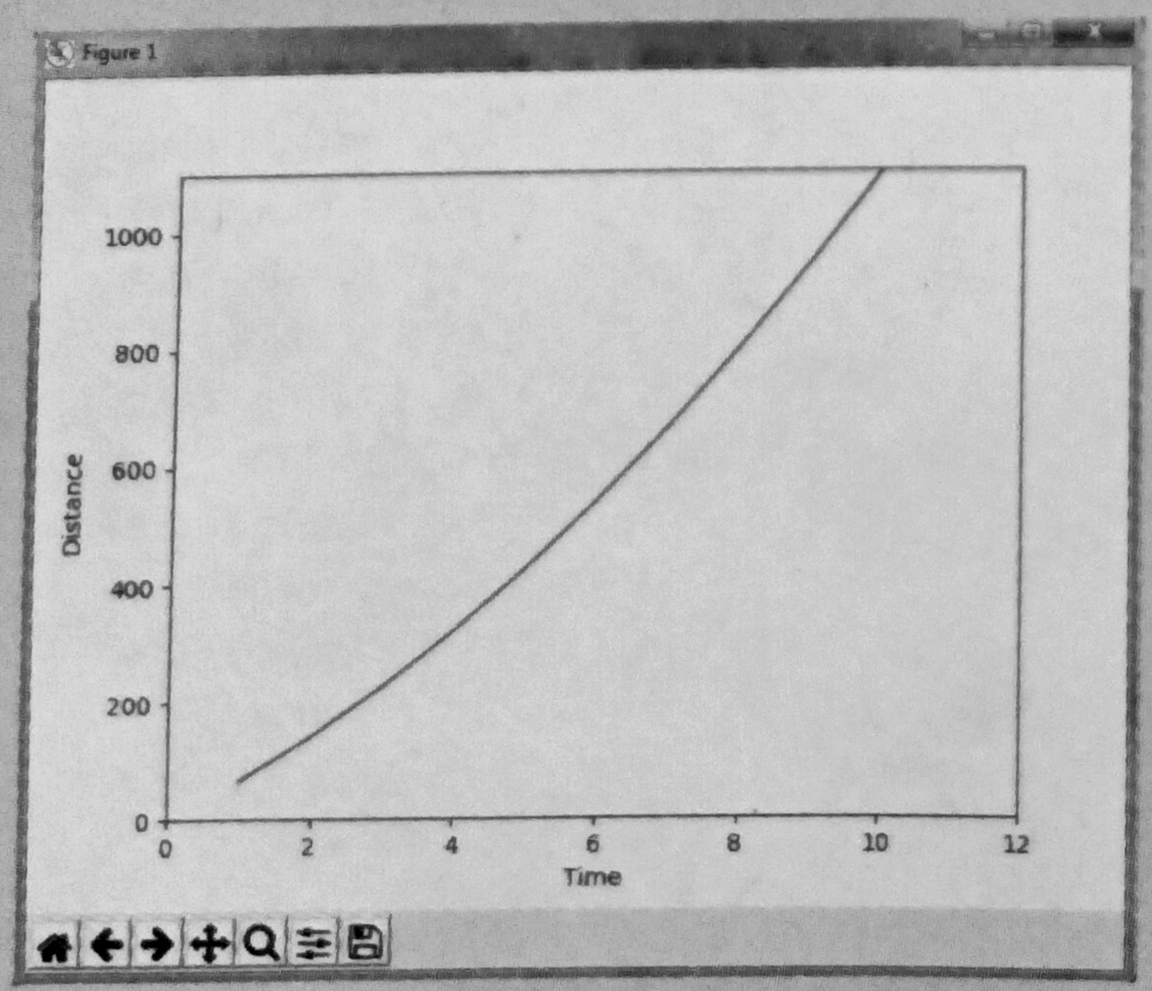
```
import matplotlib.pyplot as plt
u = int(input('Enter initial velocity : '))
a = int(input('Enter acceleration : '))
v = []
t = [1,2,3,4,5,6,7,8,9,10]
for i in t:
    v.append(u + (a*i))
plt.plot(t, v)
plt.axis([0, max(t)+2, 0, max(v)+2])
plt.xlabel('Time')
plt.ylabel('Velocity')
plt.show()

s = []
for i in t:
    s.append(u*i + (0.5)*a*i*i)
plt.plot(t, s)
plt.axis([0, max(t)+2, 0, max(s)+2])
plt.xlabel('Time')
```

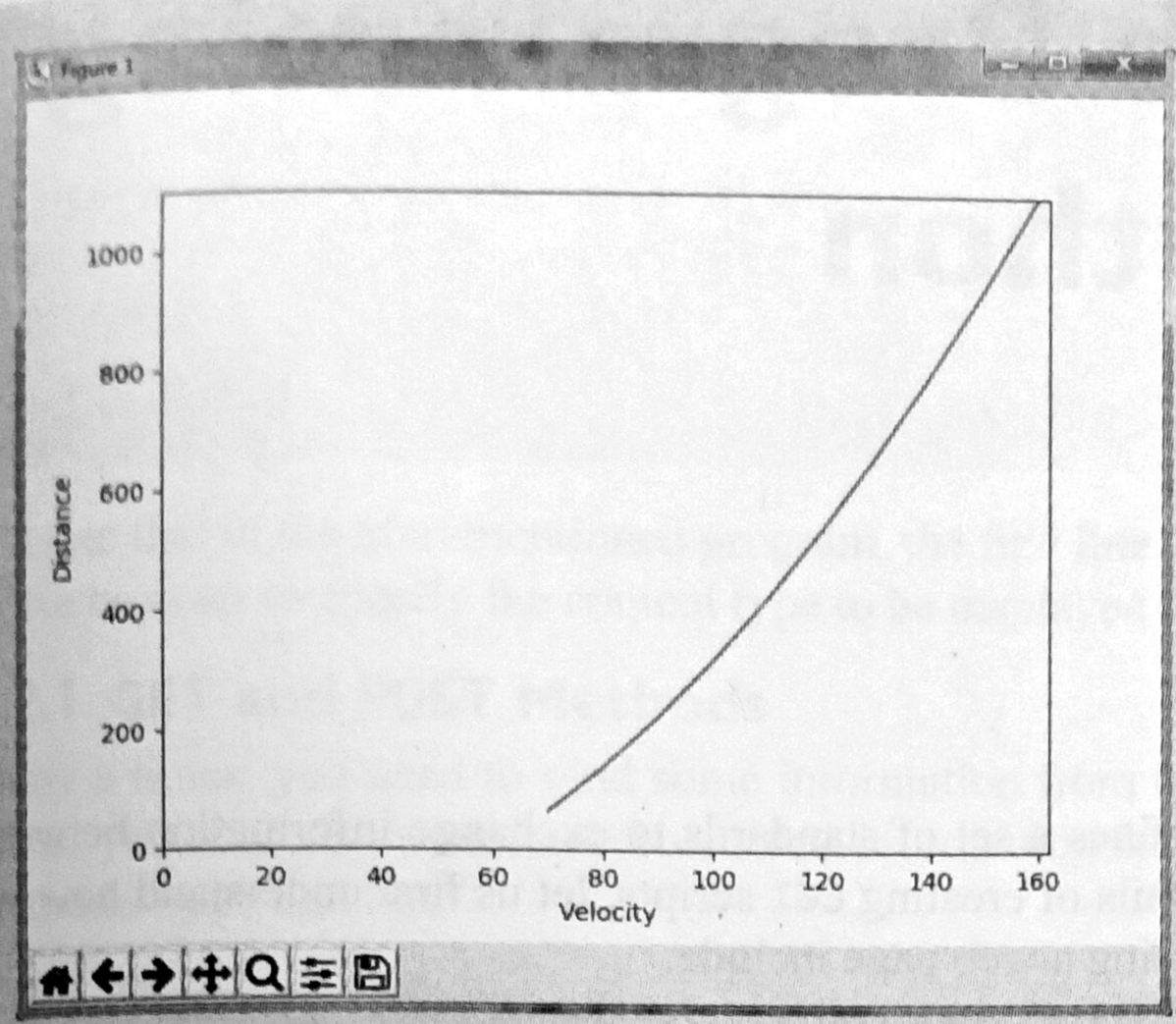
```
plt.ylabel('Distance')
plt.show()

s = []
for i in v:
    s.append((i*i - u*u)/(2*a))
plt.plot(v, s)
plt.axis([0, max(v)+2, 0, max(s)+2])
plt.xlabel('Velocity')
plt.ylabel('Distance')
plt.show()
```

OUTPUT

```
Enter initial velocity : 60
Enter acceleration : 10
```



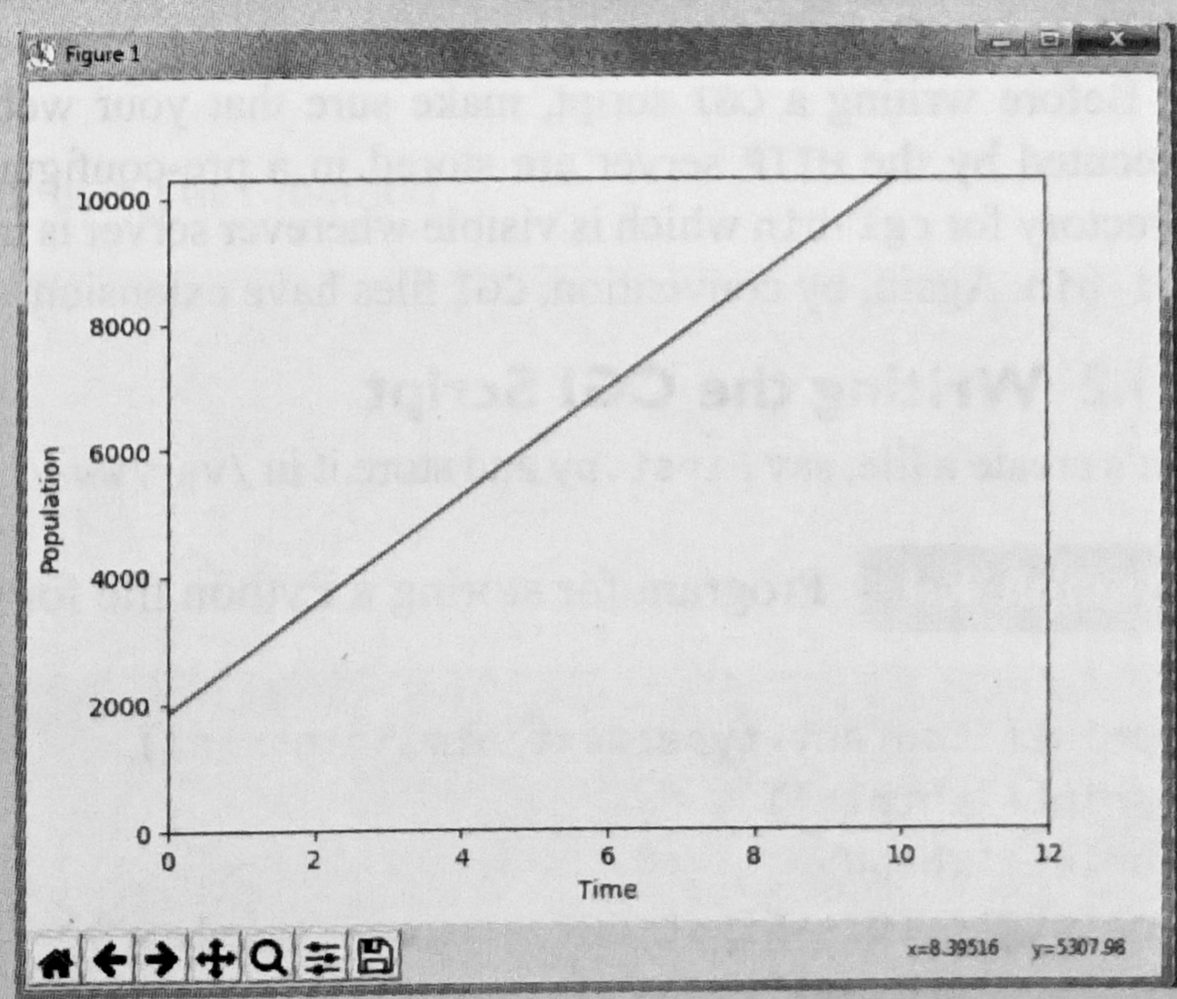




Program D.4 Population of 1000 bacteria is introduced into a nutrient medium. The population p grows as follows:

```
p(t) = (15000(1+t))/(15+ e)
```

where, the time t is measured in hours. Write a program to determine the size of the population at given time t and plot a graph for p vs t for the specified time interval.

```
import matplotlib.pyplot as plt
e = 2.718
def size_of_population(t):
    return (15000 * (1+t))/(15 + e

t = [0,1,2,3,4,5,6,7,8,9,10]
p = []
for i in t:
    p.append(size_of_population(i) +
1000)
plt.plot(t, p)
plt.axis([0, max(t)+2, 0, max(p)+2])
plt.xlabel('Time')
plt.ylabel('Population')
plt.show()
```

OUTPUT