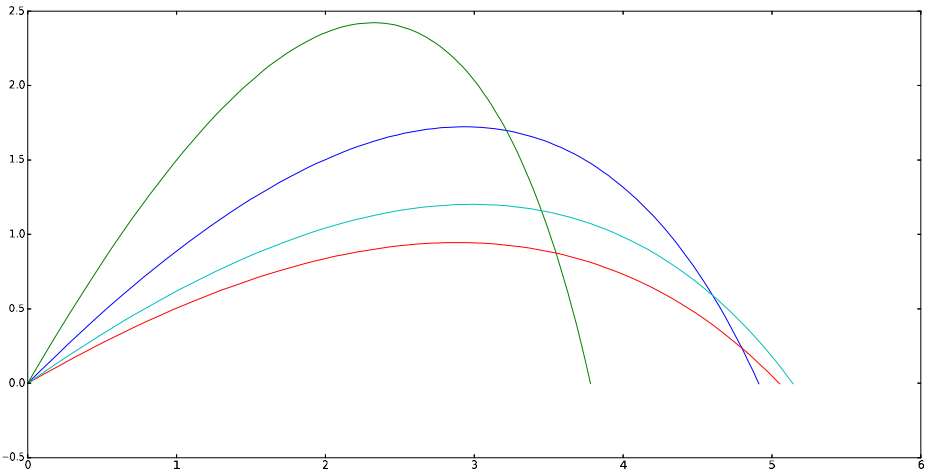
```
import numpy as np
import matplotlib.pyplot as plt
# Section I
# Question 2
def FrotLin(a,v0,alpha,dt=0.001):
    X=[0]
    Y=[0]
    t=0
    g=10
    while Y[-1]>=0:
        t=t+dt
        X.append(v0*np.cos(a)/alpha*(1-np.exp(-alpha*t)))
        Y.append((v0*np.sin(a)*alpha+g)/alpha**2*(1-np.exp(-alpha*t))-g*t/alpha)
    return X,Y
# Question 3
v0=10
alpha=1
plt.plot(*FrotLin(np.pi/4,v0,alpha))
plt.plot(*FrotLin(np.pi/3,v0,alpha))
plt.plot(*FrotLin(np.pi/6,v0,alpha))
plt.axis("scaled")
# Question 4
def AngleOptimal(v0,alpha,da=np.pi/180):
    maxdist=0
    aopt=0
    a=da
    while a<np.pi/2:
        X,Y=FrotLin(a,v0,alpha)
        if X[-1]>maxdist:
            aopt=a
            maxdist=X[-1]
        a+=da
    return aopt,maxdist
# Question 5
a,d=AngleOptimal(v0,alpha)
print("v0=",v0,"alpha=",alpha)
print("Angle optimal : ",a,"Distance parcourue :",d)
plt.plot(*FrotLin(a,v0,alpha))
plt.axis("scaled")

plt.figure()
v0=5
alpha=1
plt.plot(*FrotLin(np.pi/4,v0,alpha))
plt.plot(*FrotLin(np.pi/3,v0,alpha))
plt.plot(*FrotLin(np.pi/6,v0,alpha))
a,d=AngleOptimal(v0,alpha)
print("v0=",v0,"alpha=",alpha)
print("Angle optimal : ",a,"Distance parcourue :",d)
plt.plot(*FrotLin(a,v0,alpha))
```

```
plt.axis("scaled")

plt.figure()
v0=50
alpha=1
plt.plot(*FrotLin(np.pi/4,v0,alpha))
plt.plot(*FrotLin(np.pi/3,v0,alpha))
plt.plot(*FrotLin(np.pi/6,v0,alpha))
a,d=AngleOptimal(v0,alpha)
print("v0=",v0,"alpha=",alpha)
print("Angle optimal : ",a,"Distance parcourue :",d)
plt.plot(*FrotLin(a,v0,alpha))
plt.axis("scaled")

plt.figure()
v0=10
alpha=0.2
plt.plot(*FrotLin(np.pi/4,v0,alpha))
plt.plot(*FrotLin(np.pi/3,v0,alpha))
plt.plot(*FrotLin(np.pi/6,v0,alpha))
a,d=AngleOptimal(v0,alpha)
print("v0=",v0,"alpha=",alpha)
print("Angle optimal : ",a,"Distance parcourue :",d)
plt.plot(*FrotLin(a,v0,alpha))
plt.axis("scaled")
```




```
# Section II
def FrotLin2(a,v0,alpha,dt=0.001):
    X=[0]
    Y=[0]
    vx=[v0*np.cos(a)]
    vy=[v0*np.sin(a)]
    g=10
    while Y[-1]>=0:
        X.append(X[-1]+vx[-1]*dt)
        Y.append(Y[-1]+vy[-1]*dt)
```

```
        vx.append(vx[-1]-alpha*vx[-1]*dt)
        vy.append(vy[-1]-alpha*vy[-1]*dt-g*dt)
    return X,Y
plt.figure()
v0=10
alpha=1
plt.plot(*FrotLin(np.pi/4,v0,alpha))
plt.plot(*FrotLin2(np.pi/4,v0,alpha,dt=0.001))
plt.title('Trajectoires theoriques et numeriques\nFrottements lineaires')
plt.legend(['Theorique','Methode d\'Euler'])
plt.axis('scaled')

def AngleOptimal2(v0,alpha,da=np.pi/180,dt=0.001):
    maxdist=0
    aopt=0
    a=da
    while a<np.pi/2:
        X,Y=FrotLin2(a,v0,alpha,dt=dt)
        if X[-1]>maxdist:
            aopt=a
            maxdist=X[-1]
        a+=da
    return aopt,maxdist
a,d=AngleOptimal(v0,alpha)
print("Angle optimal, calculé avec la primitivation theorique :",a,d)
a,d=AngleOptimal2(v0,alpha)
print("Angle optimal, calculé avec la methode d'Euler :",a,d)

# Section III
def FrotNonLin(a,v0,alpha,beta,dt=0.001):
    X=[0]
    Y=[0]
    vx=[v0*np.cos(a)]
    vy=[v0*np.sin(a)]
    g=10
    while Y[-1]>=0:
        X.append(X[-1]+vx[-1]*dt)
        Y.append(Y[-1]+vy[-1]*dt)
        v=(vx[-1]**2+vy[-1]**2)**0.5
        vx.append(vx[-1]-alpha*vx[-1]*dt-beta*vx[-1]*v*dt)
        vy.append(vy[-1]-alpha*vy[-1]*dt-beta*vy[-1]*v*dt-g*dt)
    return X,Y

plt.figure()
v0=10
alpha=0.2
beta=0.05
plt.plot(*FrotLin2(np.pi/4,v0,alpha,dt=0.001))
plt.plot(*FrotNonLin(np.pi/4,v0,alpha,beta,dt=0.001))

def AngleOptimal3(v0,alpha,beta,da=np.pi/180,dt=0.001):
```

```
    maxdist=0
    aopt=0
    a=da
    while a<np.pi/2:
        X,Y=FrotNonLin(a,v0,alpha,beta,dt=dt)
        if X[-1]>maxdist:
            aopt=a
            maxdist=X[-1]
        a+=da
    return aopt,maxdist

a,d=AngleOptimal3(v0,alpha,beta)
plt.plot(*FrotNonLin(a,v0,alpha,beta,dt=0.001))
plt.axis('scaled')
plt.title('Trajectoire numerique\nFrottements non lineaires et lineaires\nAngles : 45° et op
plt.legend(['Frottements lineaires 45°','Frottements non lineaires 45°','Non lineaires, distan
```