Question 2. Using 'ode-Kepler.py', verify that the escape speed is independent of initial direction, by showing the trajectories of an object with several different initial directions but with identical total initial mechanical energy (kinetic+potential).

In [2]:

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.integrate as integ
```

In [3]:

```
# A vector function. y is a two-compo vector, i.e. (x,v).
# f returns two-compo vector, i.e. (f1,f2)=(.
def F(t,z):
    x,y,vx,vy=z
    r=np.sqrt(x**2+y**2)
    fx=-x/r**3; fy=-y/r**3
    return [vx,vy,fx,fy]
```

In [4]:

```
def orbit( z0,tmax) :
    h=float(0.001)   # time step
    nStep=int(tmax/h)  # num. steps to run
    t=np.arange(0,h*nStep,h)  # an array of discretized time
    #sol = integ.odeint(F,z0,t) # sol is [[x0,y0,vx0,vy0],..]
    sol = integ.solve_ivp(F,[0,tmax],z0,method='RK45',t_eval=t) # sol is [[x0,y0,vx0,vy0],..]
    plt.plot(sol.y[0],sol.y[1]); plt.show()
    return
```

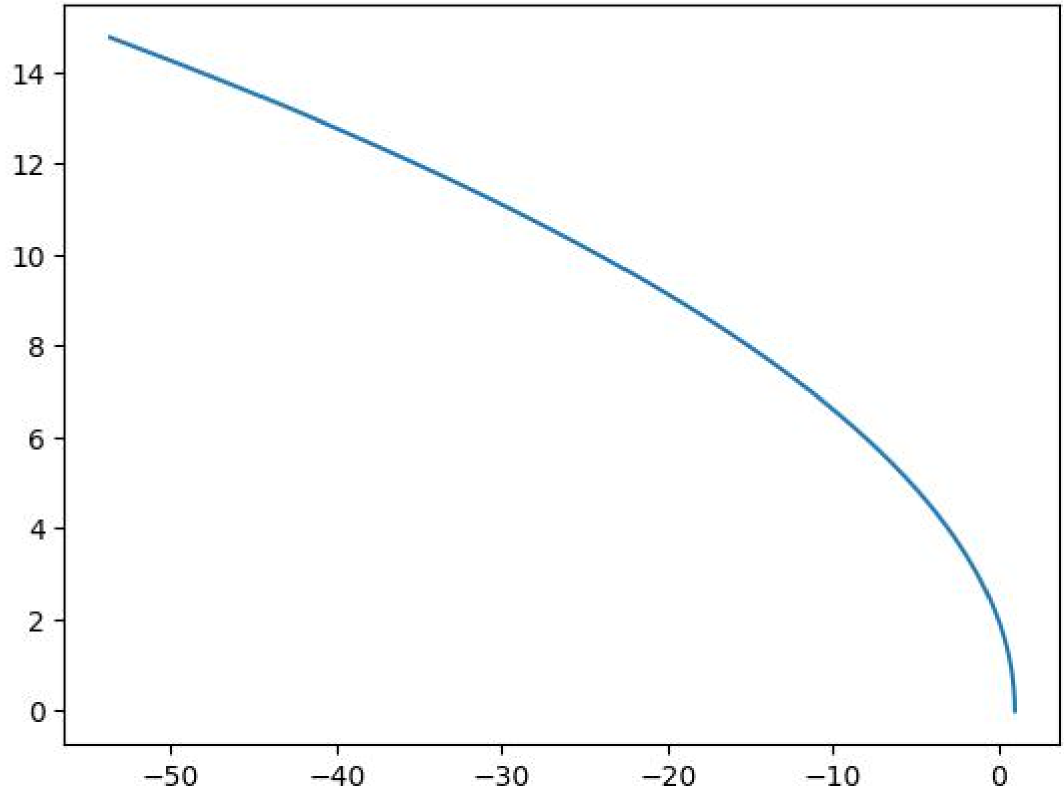
At the escape speed v, all potential energy converted to kinetic energy. So, we know $\frac{1}{2}mv_e^2 = \frac{GMm}{r^2}$

$$v_e = \sqrt{\frac{2GM}{r}}$$

For r = 1, escape speed is $\sqrt{2}$

In [5]:

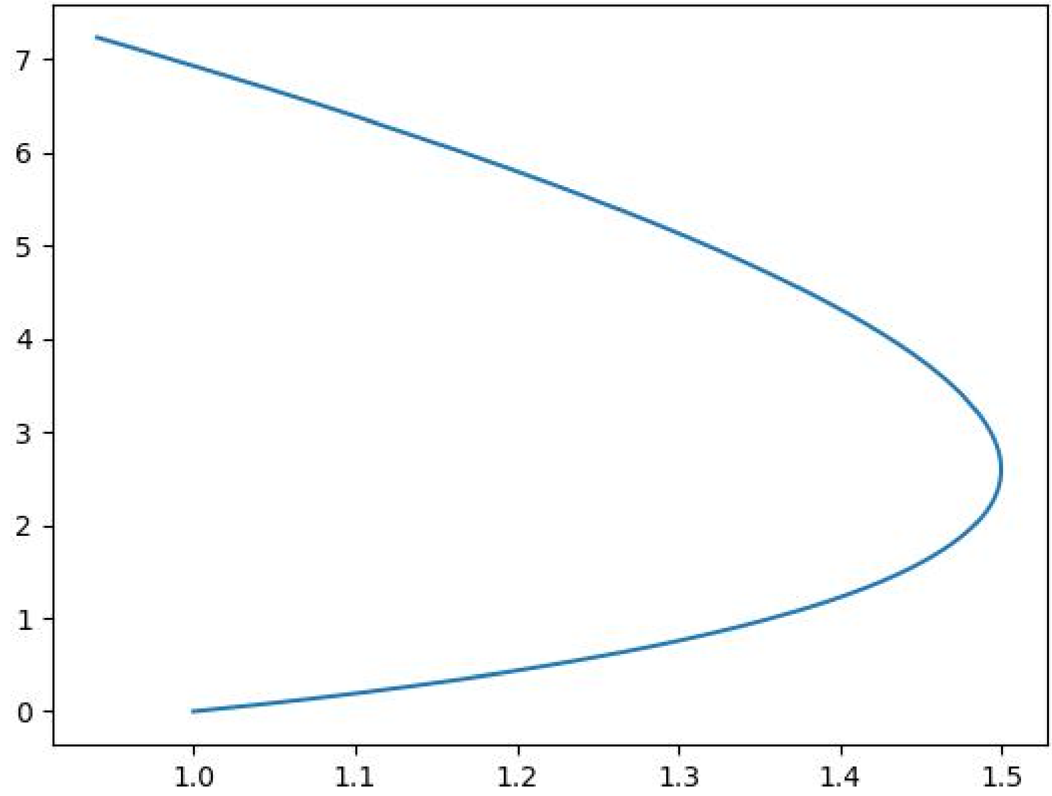
```
orbit([1., 0.0, 0.00, np.sqrt(2)],200)
```



change the direction

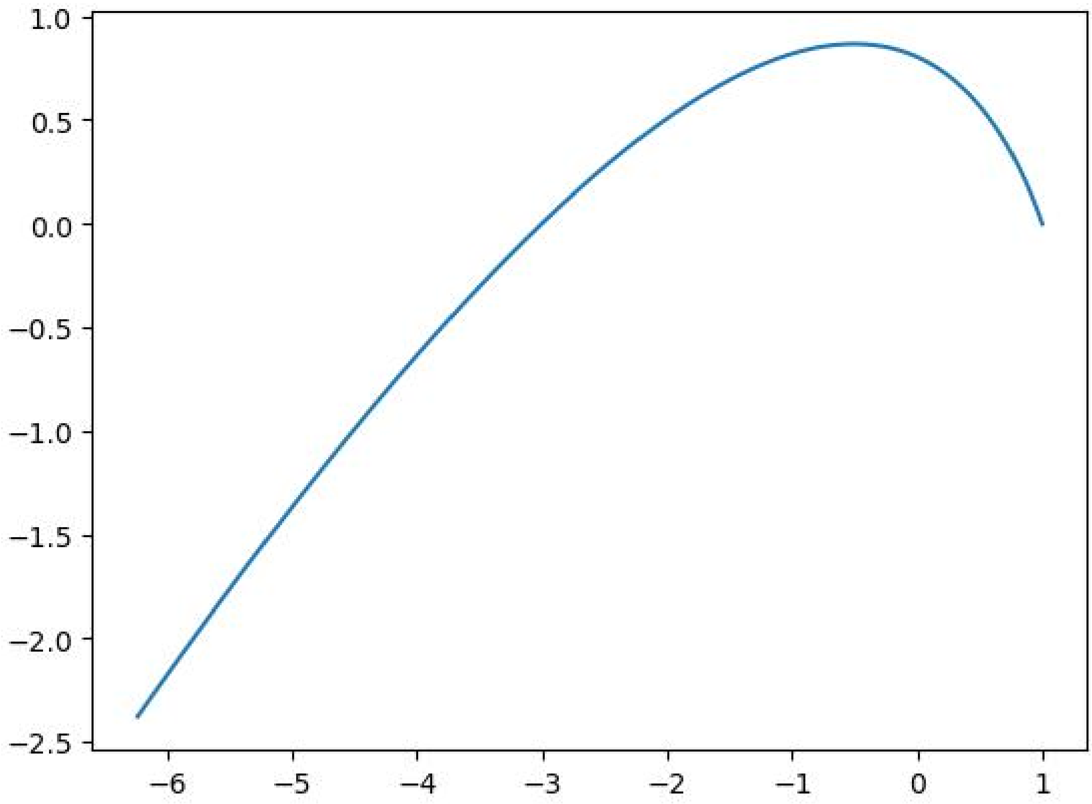
In [9]:

```
# change the direction
v0 = np.sqrt(2)
theta = np.pi/3 #direction
vx = v0*np.cos(theta)
vy = v0*np.sin(theta)
orbit([1.,0., vx,vy],10)
```



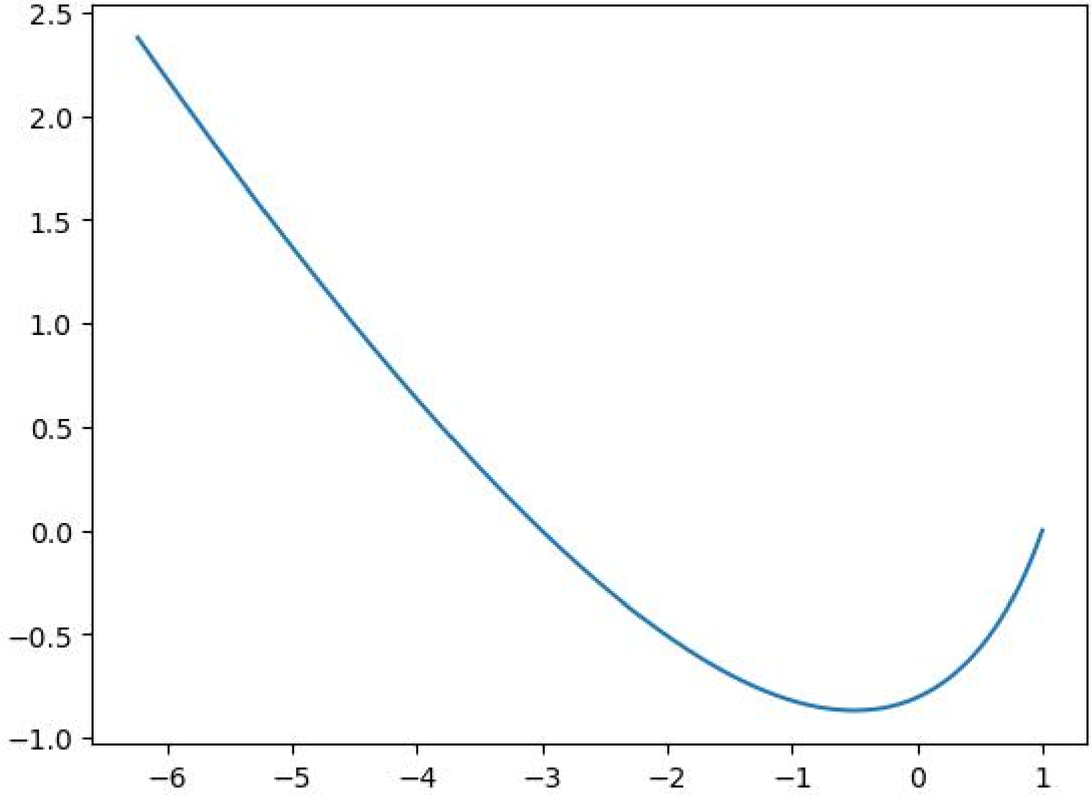
In [10]:

```
# change the direction
v0 = np.sqrt(2)
theta = np.pi*2/3
vx = v0*np.cos(theta)
vy = v0*np.sin(theta)
orbit([1.,0., vx,vy],10)
```



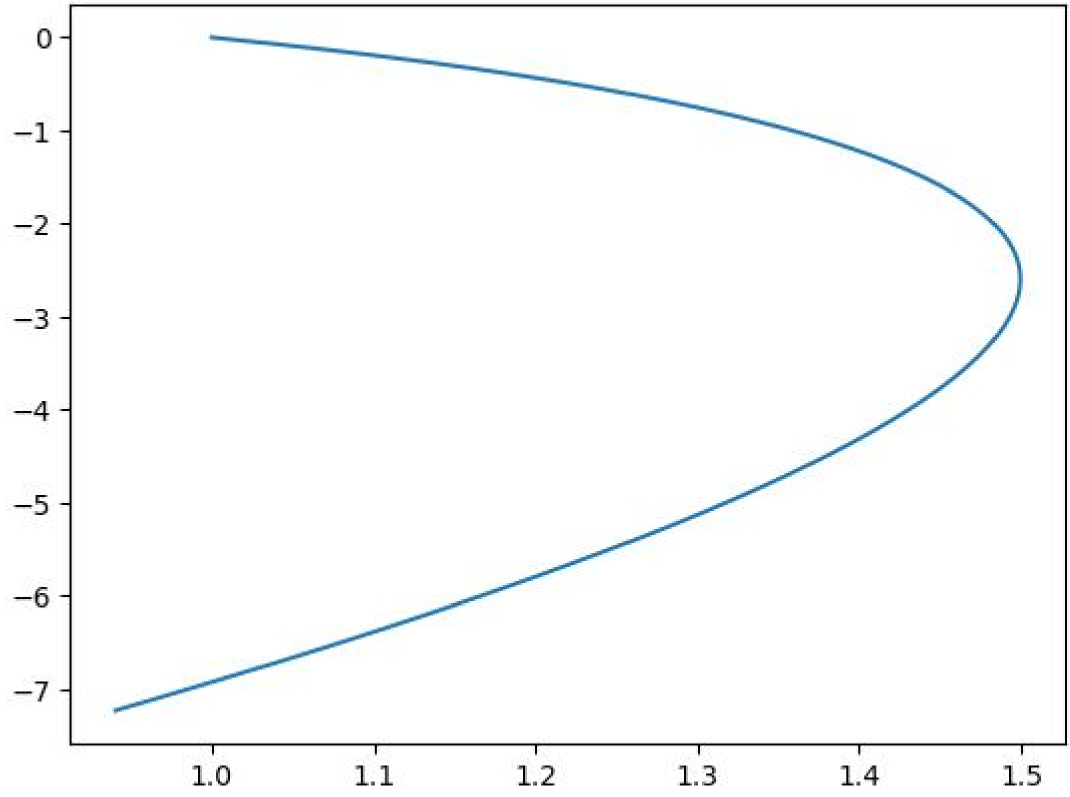
In [11]:

```
# change the direction
v0 = np.sqrt(2)
theta = np.pi*(-2/3)
vx = v0*np.cos(theta)
vy = v0*np.sin(theta)
orbit([1.,0., vx,vy],10)
```



In [12]:

```
# change the direction
v0 = np.sqrt(2)
theta = np.pi*(-1/3)
vx = v0*np.cos(theta)
vy = v0*np.sin(theta)
orbit([1.,0., vx,vy],10)
```



Adjust vx,vy