Question 2. Find up to three energy levels from the ground state of a particle trapped in a two-dimensional potential box. The width and depth of the box are Lx = 1 and Ly = 1, respectively. The potential at the wall of the box is V0 = 10. You may start with separation of variables to reduce the 2D problem into a set of 1D problems.

In [1]:

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

In [2]:

```
h=0.005
t=np.arange(0,1.0,h)
```

$$-\frac{\hbar^2}{2m}(\frac{\partial^2\psi}{\partial x^2}+\frac{\partial^2\psi}{\partial y^2}) = (E-V)\psi$$

using separation of variables and taking

$$m = 1, \hbar = 1$$

$$\frac{1}{X}\frac{\partial^2 X}{\partial x^2}+\frac{1}{Y}\frac{\partial^2 Y}{\partial y^2} = -2(E-V)\psi$$

$$\frac{1}{X}\frac{\partial^2 X}{\partial x^2} = -2(E_x-V_x)\psi$$
$$\frac{1}{Y}\frac{\partial^2 Y}{\partial y^2} = -2(E_y-V_y)\psi$$

But we don't know the information of potential energy for separation of variables. I questioned it to professor and professor answered

"내가 제시한 방법은 근사해를 구하는 방법 정도로 이해하면 될 것같습니다. 포텐셜이 아주 크면 무한 포텐셜과 크게 다르지 않을텐데, 무한 포텐셜에서는 변수분리를 확실히 사용 가능하니, 그 방법을 유한 포텐셜에 적용하여 근사해를 구한 것으로 보면 어떨까 생각됩니다."

Therefore, we will assume the infinite potential well. (V is very huge)

Therefore,

$$\frac{1}{X}\frac{\partial^2 X}{\partial x^2} = -2E_x\psi$$
$$\frac{1}{Y}\frac{\partial^2 Y}{\partial y^2} = -2E_y\psi$$

We will find

$$E_x$$

and

$$E_y$$

for

$$E = E_x + E_y$$

Differential equations for x and y are same equations, whose difference is only variables x and y. That is, allowed energy of x and y for bounded states are same. We can know the total energy by solving original 1d infinite potential well differential equation.

Problem is to find three energy levels. If we find energy of the ground state and first excited state for 1D problem, we can solve the 2d problem from combination of them.

In [3]:

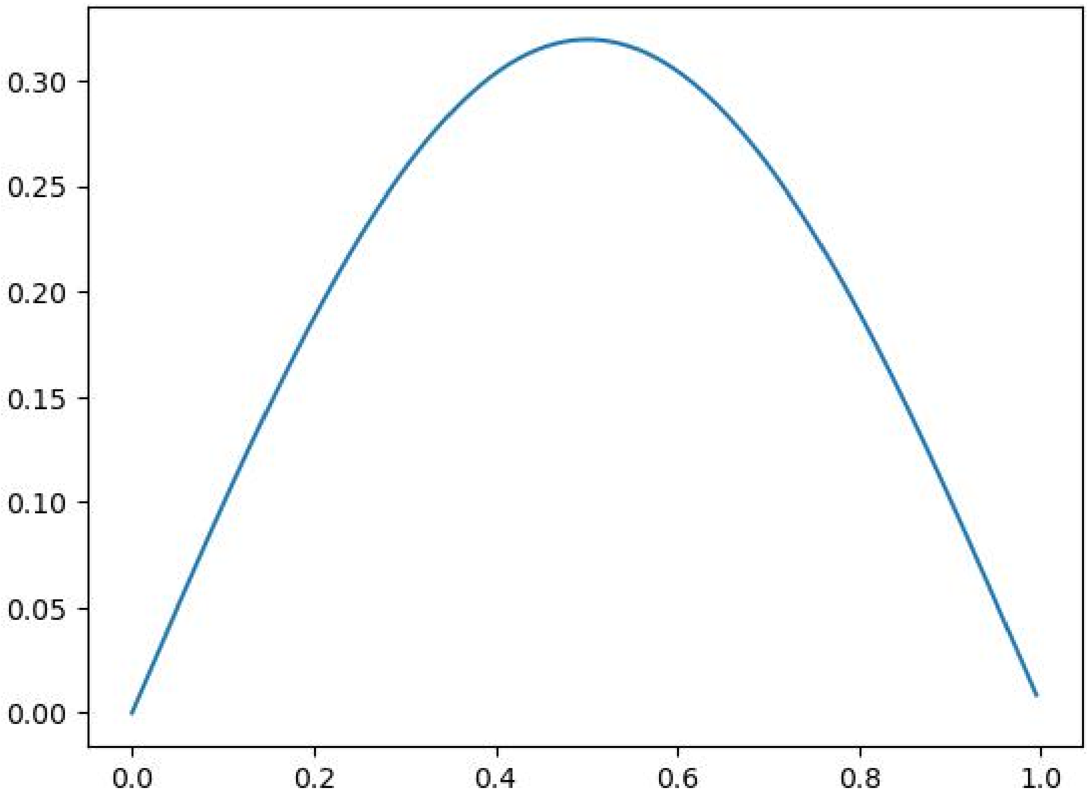
```
def f(y,t,E):
    x,v=y
    return [v,-2.0*E*x]
```

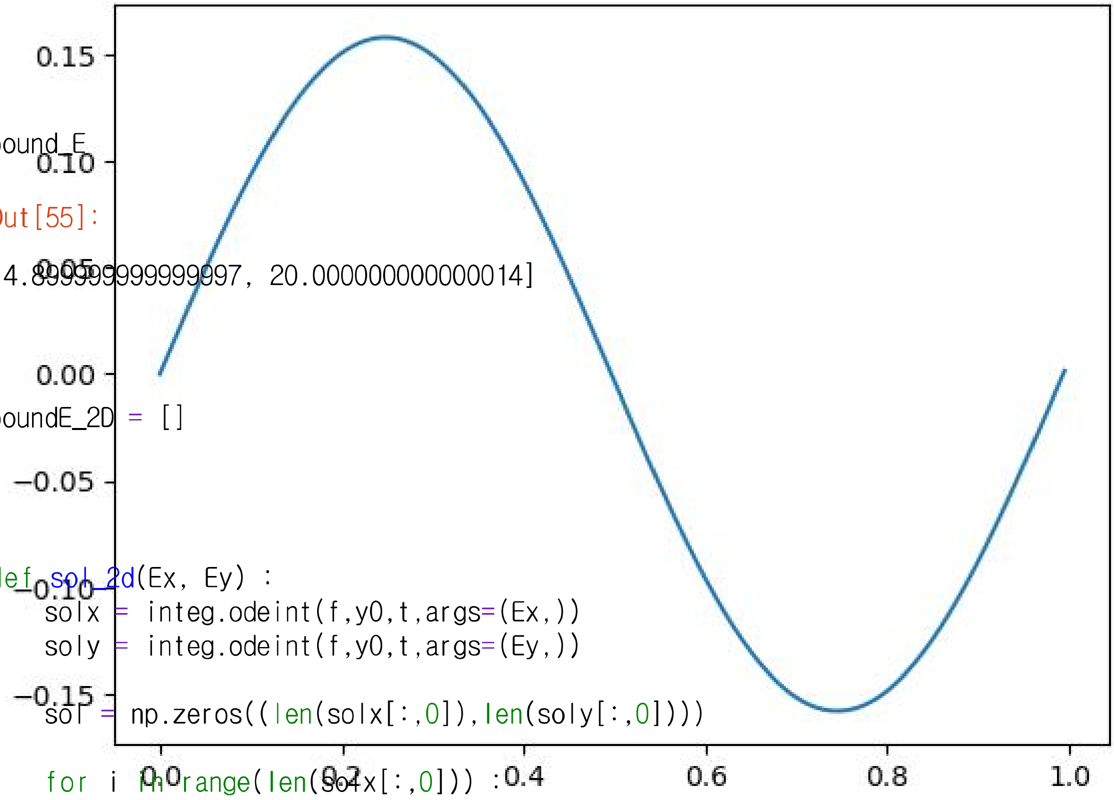
In [4]:

```
y0=[0.0,1.0]
```

In [32]:

```
i = 0
E = 4
bound_E = []
while i<2:
    sol = integ.odeint(f,y0,t,args=(E,))
    if sol[-1,0] < 0.01 and sol[-1,0]>0 :
        bound_E.append(E)
        i+=1
        plt.plot(t,sol[:,0]); plt.show()
    E += 0.1
```





```
bound_E
```

Out[55]:

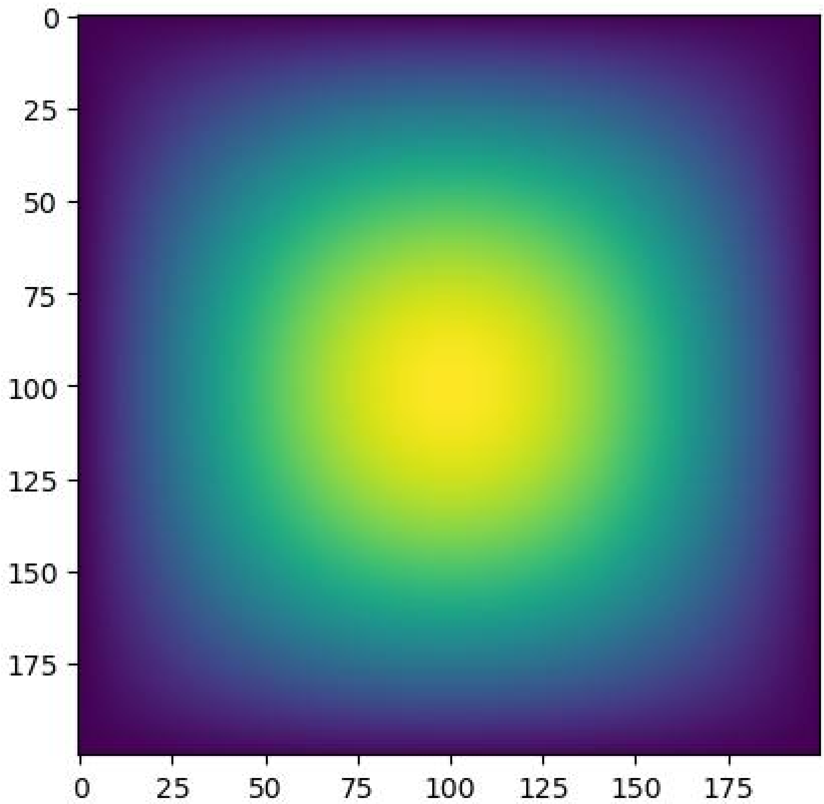
```
[4.899999999999997, 20.000000000000014]
```

```
boundE_2D = []
```

```
def sol_2d(Ex, Ey) :
    solx = integ.odeint(f,y0,t,args=(Ex,))
    soly = integ.odeint(f,y0,t,args=(Ey,))

    sol = np.zeros((len(solx[:,0]),len(soly[:,0])))

    for i in range(len(solx[:,0])) :
        for j in range(len(soly[:,0])):
            sol[i,j] = solx[i,0] * soly[j,0] # psi = X*Y

    plt.imshow(sol)

    return Ex+Ey
```

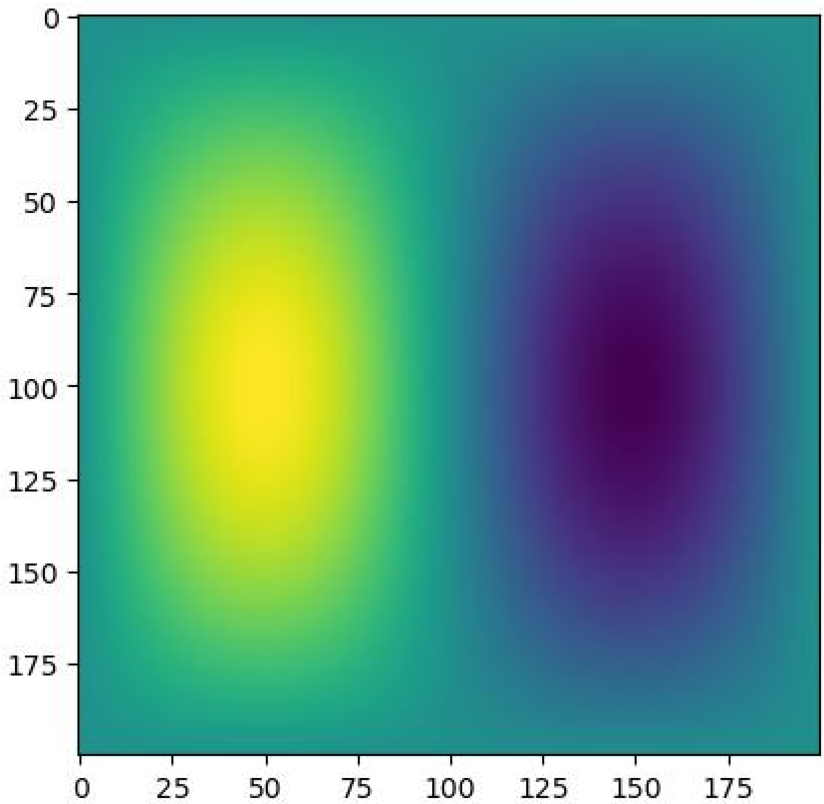
In [58]:

```
# ground state for 2d  : Ex and Ey both has minimum energy
boundE_2D.append(sol_2d(bound_E[0], bound_E[0]))
```



In [59]:

```
# first excite state for 2d
boundE_2D.append(sol_2d(bound_E[0], bound_E[1]))
#boundE_2D.append(sol_2d(bound_E[1], bound_E[0]))
```



In [60]:

```
# second excited state for 2d
boundE_2D.append(sol_2d(bound_E[1], bound_E[1]))
```