

```

import matplotlib.pyplot as plt
import numpy as np
from matplotlib import animation

pi= np.pi
h=1
L= 1
x = np.linspace(0,L,1000)
dx= x[1]-x[0]
n=10
def psi(n,x,L):
    return (2/L)**0.5 * np.sin(n*pi*x/L)

def E(n):
    return n**2*pi**2*h**2/(2*m*L**2)

def psi_real(n,x,t):
    return (2/L)**0.5 *(np.sin(n*pi*x/L)*np.cos(E(n)*t/h))

def psi_imag(n,x,t):
    return -(2/L)**0.5 *(np.sin(n*pi*x/L)*np.sin(E(n)*t/h))

def psi_total_real(n,x,c,t):
    if n!=0:
        return (c[n-1]*psi_real(n,x,t) + psi_total_real(n-1,x,c,t))
    else:
        return 0
def psi_total_time(n,x,c,t):
    if n!=0:
        #print(c[n-1])
        #print(c[n-1]*psi_imag(n,x,t))
        return (c[n-1]*psi_imag(n,x,t) + psi_total_time(n-1,x,c,t))
    else:
        return 0

c= np.zeros(n)
m=1
c= np.ones(n)
c= c/np.sqrt(np.sum(c**2))
def prob(n,x,t,c):
    return np.vdot((((psi_total_real(n,x,c,t)+ 1j*psi_total_time(n,x,c,t))),(psi_total_r

p_val=[]

for i in x:
    #print("prob",(prob(n,i,0,c)))
    p_val=p_val+ [(prob(n,i,0,c))]
    #print(p_val)
print(p_val)

def prob_plot(x,t):
    p_plot=[]
    for i in x:
        p_plot= p_plot+ [prob(n,i,t,c)]
    return p_plot

```

```

from matplotlib import pyplot as plt
from celluloid import Camera
import numpy as np

# create figure object
fig = plt.figure()
# load axis box
ax = plt.axes()
# set axis limit
ax.set_ylim(0, float(max(p_val))+0.001)
ax.set_xlim(0, L)
plt.xlabel("x")
plt.ylabel("Probability")
camera = Camera(fig)
t=0

while t<=2:
    ax.plot(x, prob_plot(x,t)/sum(prob_plot(x,t)))
    plt.pause(0.04)
    camera.snap()
    #print(t)
    t=t+0.05

```

```

animation = camera.animate()
animation.save('animation_10_ihiu.gif', writer='PillowWriter', fps=2)

```

```

⇒ [0j, (5.987974121961714e-06+0j), (2.3938871620808336e-05+0j), (5.381364970219041
<ipython-input-40-b3e84afd312e>:66: ComplexWarning: Casting complex values to re
ax.set_ylim(0, float(max(p_val))+0.001)
/usr/local/lib/python3.10/dist-packages/matplotlib/cbook/__init__.py:1335: Compl
return np.asarray(x, float)

```



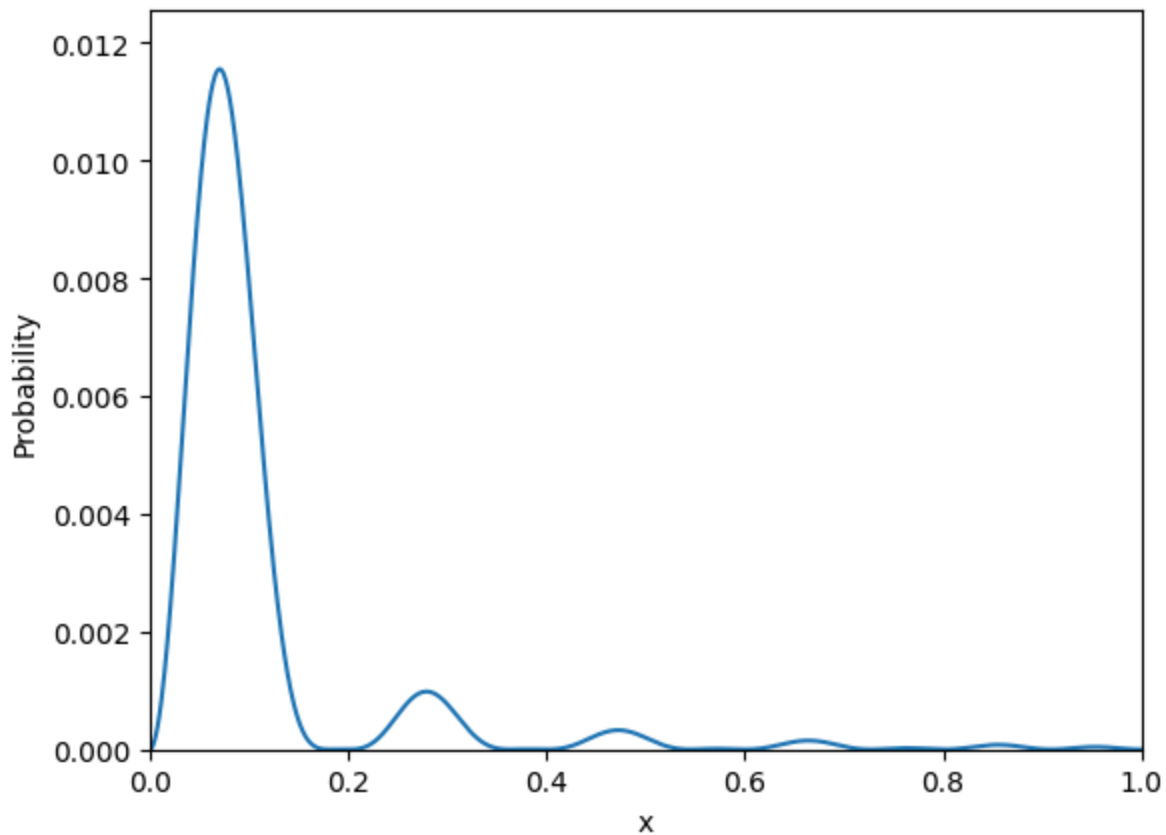
⇒ array([0.31622777, 0.31622777, 0.31622777, 0.31622777, 0.31622777,
0.31622777, 0.31622777, 0.31622777, 0.31622777, 0.31622777])

⇒ [0j, (5.987974121961714e-06+0j), (2.3938871620808336e-05+0j), (5.381364970219041

sum(np.array(p_val))

⇒ (1.0000000000000004+0j)

```
<ipython-input-34-8a49205664e9>:10: ComplexWarning: Casting complex values to re
ax.set_ylim(0, float(max(p_val))+0.001)
/usr/local/lib/python3.10/dist-packages/matplotlib/cbook/__init__.py:1335: Compl
return np.asarray(x, float)
```



```
0
0.05
0.1
0.15000000000000002
0.2
0.25
0.3
0.35
0.39999999999999997
0.44999999999999996
0.49999999999999994
0.5499999999999999
0.6
0.65
0.70000000000000001
0.75000000000000001
0.80000000000000002
0.85000000000000002
0.90000000000000002
0.95000000000000003
1.00000000000000002
1.05000000000000003
1.10000000000000003
1.15000000000000004
1.20000000000000004
1.25000000000000004
1.30000000000000005
1.35000000000000005
```

1.35000000000000005
1.40000000000000006

1.45000000000000007

!pip install celluloid



Collecting celluloid

Downloading celluloid-0.2.0-py3-none-any.whl.metadata (4.8 kB)

Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-pack

Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dis

Requirement already satisfied: cyciler>=0.10 in /usr/local/lib/python3.10/dist-pa

Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/di