NAME: ANURAG DAS

ROLL NO.: 2020PHY1116

QUANTUM MECHANICS
(LAB)

SEMESTER - V

ASSIGNMENT - 3

(a)

$$V(x) = \begin{cases} 0 & |x| < \frac{L}{2} \\ \infty & \text{otherwise} \end{cases}$$

We know that for above potential well:

$$\psi(x) = \sqrt{\frac{2}{L}} \sin kx$$

$$\text{if } k = \cancel{(2n+1)\pi}\ \frac{2n\pi}{L}$$

&

$$\psi(x) = \sqrt{\frac{2}{L}} \cos kx$$

$$\text{if } k = \frac{(2n-1)\pi}{L}$$

For ground state $n=1 \Rightarrow$ odd state

$\therefore$ probability of finding e^- :

$$= \int \psi^* \psi \, dx$$

$$= \int \frac{2}{L} \cos^2 kx \, dx$$

$$= \frac{1}{L}\int_{-L/4}^{L/4} (\cos 2kx + 1)\,dx$$

$$= \frac{1}{L}\left(\left[\frac{2\sin 2kx}{2k}\right]_{0}^{L/4} + [x]_{-L/4}^{L/4}\right)$$

$$= \frac{1}{L}\left(\frac{1}{k} \times (\cancel{\;}) + \frac{2L}{4}\right)$$

$$= \frac{1}{L}\left(\frac{L}{\pi} + \frac{L}{2}\right)$$

$$= \frac{1}{\pi} + \frac{1}{2}$$

(b) $\Delta x \, \Delta p \geq \frac{h}{4\pi}$

Uncertainity Relation

~~Ans 1~~ (a) ψ ~~For even~~

(c) (i) $\psi(x) = \sqrt{\frac{2}{L}} \sin kx$

If $k = \frac{2n\pi}{L}$

$\psi(x) = \sqrt{\frac{2}{L}} \cos kx$

If $k = \frac{(2n-1)\pi}{L}$

$\hat{p} = -i\hbar \frac{\partial}{\partial x}$

For even states:

$\langle \hat{p} \rangle = \int \psi^* \langle \hat{p} \rangle \psi \, dx$

$= \int \sqrt{\frac{2}{L}} \sin kx \left(-i\hbar \frac{\partial}{\partial x}\right) \sqrt{\frac{2}{L}} \sin kx \, dx$

~~$= \frac{2}{L} \int \sin$~~

$= -\frac{2}{L} i \int \sin kx \, \hbar k \cos kx \, dx$

$$= -\frac{2}{L} i\hbar k \int \sin kx \cos kx \, dx$$

$$= -\frac{2}{L} i\hbar k \int_{-\frac{L}{2}}^{\frac{L}{2}} \sin kx \cos kx \, dx$$

$$= \cancel{-\frac{4}{L} i\hbar k \int_{0}^{\frac{L}{2}}}$$

$$\boxed{\langle \hat{p} \rangle = 0}$$ For: even states.

Of odd states:

$$\langle \hat{p} \rangle = \int \psi^* \langle \hat{p} \rangle \psi \, dx$$

$$= \int \sqrt{\frac{2}{L}} \cos kx \left(-i\hbar \frac{\partial}{\partial x} \sqrt{\frac{2}{L}} \cos kx \right) dx$$

$$= \frac{2}{L} -i\hbar k \int -\cos kx \sin kx \, dx$$

$$= \frac{-i\hbar 2}{L} \int_{-\frac{L}{2}}^{\frac{L}{2}} \cos kx \sin kx \, dx$$

$$\boxed{\langle \hat{p} \rangle = 0}$$

$\langle \hat{p}^2 \rangle$ $\quad \hat{p} = -i\hbar \frac{\partial}{\partial x}$

$$\boxed{\hat{p}^2 = -\hbar^2 \frac{\partial^2}{\partial x^2}}$$

$$\langle \hat{p}^2 \rangle = \int \psi^* -\hbar^2 \frac{\partial^2}{\partial x^2} \psi \, dx$$

If ~~even~~ odd energy state:

$$\langle \hat{p}^2 \rangle = \int \psi^* -\hbar^2 \frac{\partial^2}{\partial x^2} \psi \, dx$$

$$= \int \left(\sqrt{\frac{2}{L}} \cos kx \, -\hbar^2 \frac{\partial^2}{\partial x^2} \sqrt{\frac{2}{L}} \cos kx \right) dx$$

$$= \cancel{\int \sqrt{\frac{2}{L}} \cos kx \, \hbar^2}$$

$$= -\frac{2}{L}\hbar^2 \int \left(\cos kx\left(-k^2 \cos kx\right)\right) dx$$

$$= +\frac{2}{L}\hbar^2 k^2 \int_{-\frac{L}{2}}^{\frac{L}{2}} \cos^2 kx \, dx$$

$$= \frac{1}{L}\hbar^2 k^2 \int_{-\frac{L}{2}}^{\frac{L}{2}} (\cos 2kx + 1) \, dx$$

$$\boxed{\langle p^2 \rangle = \frac{\hbar^2 k^2 \cancel{L}}{\cancel{L}}} = \cancel{\hbar^2 k^2}$$

If ~~odd~~ even energy state:

$$\langle \hat{p}^2 \rangle = \int \psi^* \, -\hbar^2 \frac{\partial^2}{\partial x^2} \psi \, dx$$

$$= \int \sqrt{\frac{2}{L}} \sin kx \left(-\hbar^2 \frac{\partial^2}{\partial x^2} \sqrt{\frac{2}{L}} \sin kx\right) dx$$

$$= \frac{2\hbar^2}{L}\int_{-\frac{L}{2}}^{\frac{L}{2}} \left(\sin^2 kx\right) k^2 \, dx$$

$$= \frac{2\hbar^2 k^2 \times 2}{L}\int_{0}^{\frac{L}{2}} \sin^2 kx \, dx$$

$$= \frac{4\hbar^2 k^2}{L}\int_{0}^{\frac{L}{2}} \left(\frac{1-\cos 2kx}{2}\right) dx$$

$$= \frac{4\hbar^2 k^2}{2L}\left[\left(\frac{x}{2}\right) - \left(\frac{\sin 2kx}{2k}\right)\right]_0^{\frac{L}{2}}$$

$$= \frac{4\hbar^2 k^2}{2L}\left[\frac{L}{2} - 0\right]$$

$$= \hbar^2 k^2$$

$$\boxed{\langle p^2 \rangle = \hbar^2 \times \frac{4n^2\pi^2}{L^2}}$$

$$\langle \hat{x} \rangle = \int \psi^* x \psi \, dx$$

$$= \cancel{\int \sqrt{\frac{2}{L}} \sin kx \left(-i\hbar \frac{\partial}{\partial x}\right)}$$

$$= \int \sqrt{\frac{2}{L}} \sin kx \; x \left(\sqrt{\frac{2}{L}} \sin kx\right) dx$$

$$= \int \frac{2}{L} x \sin^2 kx \, dx$$

$$= \frac{2}{L} \int^{L} x \left(\frac{1 - \cos 2kx}{2}\right) dx$$

$$\cancel{= \frac{1}{L} \left[\frac{x^2}{2} - x \cos 2kx\right]_{-\frac{L}{2}}^{+\frac{L}{2}}}$$

$$\cancel{= \frac{1}{L} \int \frac{x^2 \, dx}{2}}$$

$$= \frac{1}{L} \left[\int x \, dx - \int x \cos 2kx\right]$$

$$\cancel{= \frac{1}{L} \left[\frac{x^2}{2}\right]_{-\frac{L}{2}}^{L/2} - \frac{x \cos 2kx}{2}}$$

$$= \frac{1}{L}\left[(0) - \left[x \sin 2kx + \cos 2kx \right]_{-\frac{L}{2}}^{\frac{L}{2}} \right]$$

$$= \frac{1}{L}\left(- \right.$$

$$\int x \cos 2kx \, dx = x \frac{\sin 2kx}{2k} - \int \frac{\sin 2kx}{2k} \, dx$$

$$\int x \cos 2kx \, dx = x \frac{\sin 2kx}{2k} + \frac{1}{2k} \frac{\cos 2kx}{2k}$$

$$\int_{-\frac{L}{2}}^{\frac{L}{2}} x \cos 2kx \, dx = \left[x \frac{\sin 2kx}{2k} + \frac{1}{2k} \frac{\cos 2kx}{2k} \right]_{-\frac{L}{2}}^{\frac{L}{2}}$$

$$= \frac{L}{2} \frac{\sin \cancel{2} \frac{2n\pi}{L} \times \frac{L}{2}}{\frac{4n\pi}{L}} + \frac{1}{\frac{4n\pi}{L}} \times \frac{\cos \frac{4n\pi}{L} \times \frac{L}{2}}{\frac{4n\pi}{L}}$$

$$= \cancel{0 + \frac{L\cos}{16n^2\pi^2}} - \left(0 + \frac{\cos \frac{4n\pi}{L} \times \frac{L}{2}}{\frac{4n\pi}{L}} \right)$$

$$= 0$$

$$S = 0$$

For odd function:

$$\langle \hat{x} \rangle = \int \left(\sqrt{\frac{2}{L}} \cos kx\right) x \left(\sqrt{\frac{2}{L}} \cos kx\right) dx$$

$$= \frac{2}{L} \int x \left(\frac{\cos 2kx + 1}{2}\right) dx$$

$$\langle \hat{x} \rangle = 0$$

$$\langle \hat{x}^2 \rangle = \int \psi^* x^2 \psi \, dx$$

For odd state

$$= \int \left(\sqrt{\frac{2}{L}} \cos kx \; x^2 \sqrt{\frac{2}{L}} \cos kx\right) dx$$

$$= \frac{2}{L} \int_{-\frac{L}{2}}^{\frac{L}{2}} x^2 \cos^2 kx \, dx$$

$$= \frac{2}{L} \times 2 \int_0^{\frac{L}{2}} x^2 \cos^2 kx \, dx$$

$$= \frac{4}{L} \int_0^{\frac{L}{2}} x^2 \left(\frac{\cos 2kx + 1}{2} \right) dx$$

$$= \frac{2}{L} \int_0^{\frac{L}{2}} \left(x^2 \cos 2kx + x^2 \right) dx$$

$$= \frac{2}{L} \left(\left[\frac{x^2 \sin 2kx}{2k} + \frac{x \cos 2kx}{2k} - \frac{\sin 2kx}{4k} \right]_0^{\frac{L}{2}} + \left(\frac{x^3}{3} \right)_0^{\frac{L}{2}} \right]$$

$$= \frac{2}{L} \left(\frac{\frac{L}{2} \cos \cancel{2} \left(\frac{(2n+1)\pi}{\cancel{L}} \right) \times \frac{\cancel{L}}{2}}{2k} + \frac{L^3}{24} \right)$$

$$= \frac{-1}{2k} + \frac{L^2}{12}$$

$$\sigma_p = \sqrt{\langle p^2\rangle - \langle p\rangle^2}$$

$$= \sqrt{\frac{4n^2\pi^2\hbar^2}{L^2} - 0}$$

$$\boxed{\sigma_p = \frac{2n\pi\hbar}{L}}$$

) $\sigma_x = \sqrt{\langle x^2\rangle - \langle x\rangle^2}$

$$\boxed{\sigma_x = \frac{L}{2}\sqrt{\frac{1}{3} - \frac{2}{n\pi^2}}}$$

(ii) $\Delta x \Delta p \geq \frac{h}{4\pi}$

For $\Delta x \Delta p = \frac{h}{4\pi}$

$$\frac{n\pi\hbar}{L} \times \frac{L}{2}\sqrt{\frac{1}{3} - \frac{2}{n^2\pi^2}} = \frac{h}{4\pi}$$

$\Downarrow$

For $n = 1$

the above condition satisfies.

(iii) $-i\hbar \frac{\partial \psi}{\partial x} = p\psi$

we know that

$$\frac{\partial \psi}{\partial x} = \frac{\partial \psi}{\partial \xi} \times \frac{1}{L}$$

$$-i\frac{\hbar}{L}\frac{\partial \psi}{\partial \xi} = p\psi$$

$$-\frac{\partial \psi}{\partial \xi} = \frac{pL}{\hbar}\psi$$

let $\frac{pL}{\hbar} = p'$

$\frac{L}{\hbar}$ has dimensions of $\frac{1}{\text{momentum}}$.

Thus p' is dimensionless.

$$-\frac{\partial \psi}{\partial \xi} = p'\psi$$

Similarly momentum square eq^n.

$$-\hbar^2 \frac{\partial^2 \psi}{\partial x^2} = p^2 \psi$$

We know that $\frac{\partial^2 \psi}{\partial x^2} = \frac{1}{L^2} \frac{\partial^2 \psi}{\partial \xi^2}$

So we get, $-\frac{\hbar^2}{L^2} \frac{\partial^2 \psi}{\partial \xi^2} = p^2 \psi$

$$-\frac{\partial^2 \psi}{\partial \xi^2} = \frac{p^2 L^2}{\hbar^2} \psi$$

$$\boxed{-\frac{\partial^2 \psi}{\partial \xi^2} = (p')^2 \cdot \psi}$$

$(p')^2 = \frac{p^2 L^2}{\hbar^2}$ dimensionless momentum square

Eigen Value eqⁿ for $(p')^2$ as $-\frac{\partial^2\psi}{\partial\xi^2} = (p')^2\psi$

$$(\Delta p')^2 = \langle (p')^2\rangle - (\langle p'\rangle)^2$$

$$= \left\langle \frac{p^2L^2}{\hbar^2}\right\rangle - \left(\frac{\langle pL\rangle}{\hbar}\right)^2$$

$$(\Delta p')^2 = \frac{L^2}{\hbar^2}\left[\langle p^2\rangle - \langle p\rangle^2\right]$$

$$(\Delta\xi)^2 = \langle \xi^2\rangle - \langle\xi\rangle^2$$

$$= \left\langle\frac{x^2}{L^2}\right\rangle - \left(\frac{\langle x\rangle}{L}\right)^2$$

$$= \left[\langle x^2\rangle - \langle x\rangle^2\right]\frac{1}{L^2}$$

$$(\Delta p')^2(\Delta\xi)^2 = \frac{L^2}{\hbar^2}\left[\langle p^2\rangle - \langle p\rangle^2\right] \times \frac{1}{L^2}$$

$$\cancel{(\Delta p')^2(\Delta\xi)^2 = \theta\left[\langle p^2\rangle - \langle p\rangle^2\right]} \cdot \left[\langle x^2\rangle - \langle x\rangle\right.$$

$$(\Delta p')^2(\Delta\xi)^2 = \frac{\left[\langle p^2\rangle - \langle p\rangle^2\right]\cdot\left[\langle x^2\rangle - \langle x\rangle^2\right]}{\hbar^2}$$

We know that

$$(\Delta x)^2 (\Delta p)^2 = [\langle x^2\rangle - \langle x\rangle^2][\langle p^2\rangle - \langle p\rangle^2] \quad \text{--- (2)}$$

Substituting (2) in (1) we get:

$$(\Delta p')^2 (\Delta \xi)^2 = \frac{(\Delta x)^2 (\Delta p)^2}{\hbar^2}$$

Taking square root we get

$$(\Delta p')(\Delta \xi) = \frac{(\Delta x)(\Delta p)}{\hbar}$$

relationship b/w dimensional and dimensionless uncertainity principle.

$$\therefore (\Delta x)(\Delta p) \geq \frac{\hbar}{2}$$

$$\Rightarrow \hbar (\Delta p')(\Delta \xi) \geq \frac{\hbar}{2}$$

$$\boxed{(\Delta p')(\Delta \xi) \geq \frac{1}{2}}$$

~~Uncertainity in proper~~

Uncertainity principle in Dimensionless form.

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.integrate as integrate
import pandas as pd
from sklearn.linear_model import LinearRegression

P = []
X = []


def RK4(f, X0, tmin, tmax, N, e):
    h = (tmax - tmin) / N
    t = np.linspace(tmin, tmax, N + 1)
    X = np.zeros([N + 1, len(X0)])
    X[0] = X0
    for i in range(N):
        k1 = f(t[i], X[i], e)
        k2 = f(t[i] + h / 2, X[i] + (k1 * h) / 2, e)
        k3 = f(t[i] + h / 2, X[i] + (k2 * h) / 2, e)
        k4 = f(t[i] + h, X[i] + k3 * h, e)
        X[i + 1] = X[i] + (h / 6) * (k1 + 2 * (k2 + k3) + k4)

    return X, t


def func(x, Y, e):  # functions along with the initial conditions
    psi = 0
    y1, y2 = Y

    psider1 = y2
    psider2 = -e * y1
    return np.array([psider1, psider2])


def main_f(f, ic, tmin, tmax, N, e):
    rk4 = RK4(f, ic, tmin, tmax, N, e)
    rk4_X = rk4[0].T
    t = rk4[-1]
    return rk4_X, t

#Part (a)

initial_conditions = [0, 1]

last_u = []
x = []
e = np.arange(0, 251, 1)
for i in range(0, 251):
    Z, t = main_f(func, initial_conditions, -1 / 2, 1 / 2, 100, i)
    # NORMALIZATION
```

```
    c = integrate.simps(Z[0] ** 2, t)
    N = Z[0] / np.sqrt(c)
    last_u.append(Z[0][-1])
    x.append(t[-1])

plt.scatter(e, last_u)
plt.grid()
plt.show()

# CHECKING
last_u = np.array(last_u)
last_u2 = last_u[:-1]
last_u3 = last_u[1:]
new_list = last_u2 * last_u3
new = []
new1 = []
new2 = []

index1 = []
index2 = []
for i, j in zip(last_u2, last_u3):
    if i * j < 0:
        new1.append(i)
        new2.append(j)
        index1.append(np.where(last_u == i))
        index2.append(np.where(last_u == j))

I = index1 + index2
I.sort()
print(I)
u_req = np.array(new1) * np.array(new2)

index2 = []
for i in I:
    t = i[0][0]
    index2.append(t)
print(index2)

energy_req = []
psi_req = []
for i in index2:
    energy_req.append(e[i])
    psi_req.append(last_u[i])
print(energy_req)

# FILTERING

energy = np.array(energy_req)
energy_2 = energy[:-1]
energy_3 = energy[1:]
```

```
psi_req_2 = np.array(psi_req[:-1])
psi_req_3 = np.array(psi_req[1:])
approx = energy_3 - ((energy_3 - energy_2) / (psi_req_3 - psi_req_2)) * psi_req_3
approx2 = approx[::2]
print(approx2)

fig, axs = plt.subplots(len(approx2))
p=0

psi_approx2=[]
psi_der_approx2=[]
t_approx2=[]

for i in approx2:
    Z, t = main_f(func, initial_conditions, -1 / 2, 1 / 2, 100, i)
    # NORMALIZATION
    c = integrate.simps(Z[0] ** 2, t)
    N = Z[0] / np.sqrt(c)
    psi_approx2.append(N)

    d = integrate.simps(Z[1] ** 2, t)
    N2 = Z[1] / np.sqrt(c)
    psi_der_approx2.append(N2)
    t_approx2.append(t)

    #fig.suptitle("FOR e= " + str(i))
    axs[p].plot(t, N)
    axs[p].set_title("FOR e= " + str(i))
    axs[p].grid()
    p+=1
plt.show()

#Part (b) CURVE FITTING

nsqr = np.array(approx2 / np.pi ** 2)
print("nsqr: ",nsqr)
plt.scatter(nsqr,approx2)
model= LinearRegression()
model.fit(nsqr.reshape((-1,1)),approx2)
ypred=model.predict(nsqr.reshape((-1,1)))
print("slope: ",model.coef_)
print("Actual Value Of Slope: ", np.pi**2)
print("R_sqr: ",model.score(nsqr.reshape((-1,1)),approx2))

plt.plot(nsqr,ypred,linestyle='dashdot',color='red')
plt.grid()
plt.show()

#Part (c)
```

```
L=1
x=np.linspace(-1/2,1/2,100)

for i in range(1,6):
    e=(i**2)*(np.pi**2)
    psi=[]
    Z,t=main_f(func,initial_conditions,-1/2,1/2,100,e)
    # NORMALIZATION
    c = integrate.simps(Z[0] ** 2, t)
    N = Z[0] / np.sqrt(c)
    plt.scatter(t,N**2,label='calculated psi vs x for e='+str(i))
    if i % 2 !=0:
        k= i*(np.pi/L)
        psiodd=np.sqrt(2)*np.cos(k*x)
        plt.plot(x,psiodd**2,label='analytic solution of psi for n='+str(i))
    else:
        k = i * (np.pi / L)
        psieven = np.sqrt(2) * np.sin(k * x)
        plt.plot(x, psieven**2, label='analytic solution of psi for n=' + str(i))
plt.grid()
plt.legend()
plt.show()

#Part (d) and (e)

def eV(m,L,e_approx):
    h=6.63* 10**(-34)

    Eval=[]
    for i in range(1,6):
        Eval.append((i**2 * np.pi**2 * h**2)/(8* m * (L**2)))
    Eval=np.array(Eval)* 6.242 * (10**(17))
    Eigenval=e_approx*((h**2)/(8* m * (L**2))) * 6.242 * (10**(17))
    dtf1=pd.DataFrame({"Analytical Energy Value": Eval, "Eigen ENergy Val":
Eigenval})
    print(dtf1)

print("--------FOR ELECTRON WHEN WIDTH OF WELL= 5 Angstrom------")
eV(9.11*(10**(-31)),5 * (10**(-10)),approx2)
print()

print("--------FOR ELECTRON WHEN WIDTH OF WELL= 10 Angstrom------")
eV(9.11*(10**(-31)),10 * (10**(-10)),approx2)
print()

print("--------FOR ELECTRON WHEN WIDTH OF WELL= 5 Fermi------")
eV(1.67*(10**(-27)),5 * (10**(-15)),approx2)
print()
```

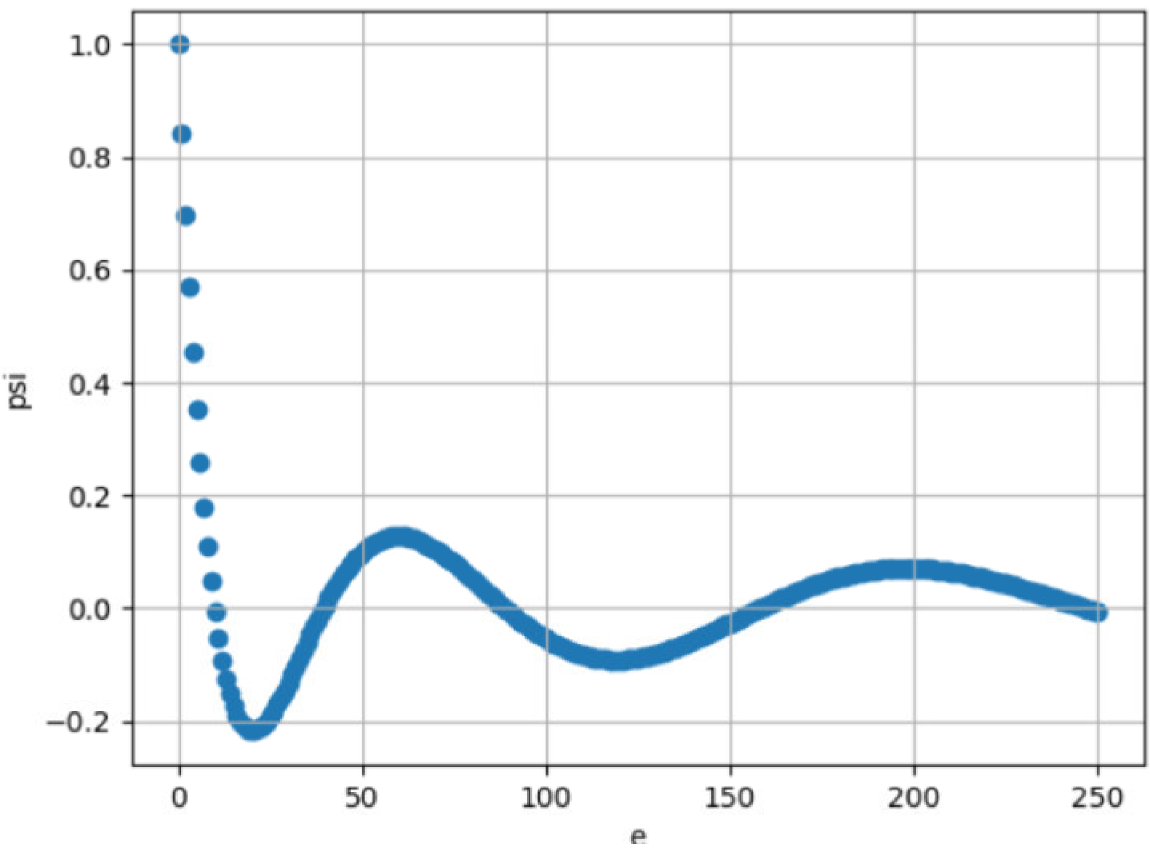
```
#EXPECTAION VALUES

#Part (f)

exp_x=[]
exp_x2=[]

for i,j in zip(psi_approx2,t_approx2):
    exp_x.append(integrate.simps(i**2 * j,j))
    exp_x2.append(integrate.simps((i ** 2) * (j ** 2), j))

psi_2der_approx2=[]
for i,j in zip(approx2,psi_approx2):

    psi_2der_approx2.append(j*i)


exp_p=[]
exp_p2=[]


for i,j,k,l in zip(psi_approx2,psi_der_approx2,psi_2der_approx2,t_approx2):
    exp_p.append(integrate.simps(j*i,l))
    exp_p2.append(integrate.simps(k *i, l))

print(np.array(exp_p) * -1j * 1.05)
print(np.array(exp_p2) * 1.11)

sigma_p=np.sqrt((np.array(exp_p2) * 1.11*10**(-34)) - (np.array(exp_p) * -1j *
1.05*10**(-34))**2)
sigma_x=np.sqrt((np.array(exp_x2)) - (np.array(exp_x)**2))
print(sigma_x*sigma_p)

#Part (g)

subr=[]
index=[]
subpsi=[]
for i in t_approx2[0]:
    if i>=-1/4 and i<=1/4:
        subr.append(i)
        index.append(np.where(t_approx2[0]==i))
psi_n1=psi_approx2[0]
print(psi_n1)
print(index)
for i in index:
    subpsi.append(psi_n1[i])
print(len(subpsi))
x=np.arange(-1/4,1/4,1)
```

```
print(len(subr))
subpsi2=[]
for i in subpsi:
    subpsi2.append(i[0])
print(subpsi2)


print(subr)
print("Probaility Of Finding The Particle Between L/4 to L/4:
",integrate.simps(np.array(subpsi2)**2,np.array(subr)))
```

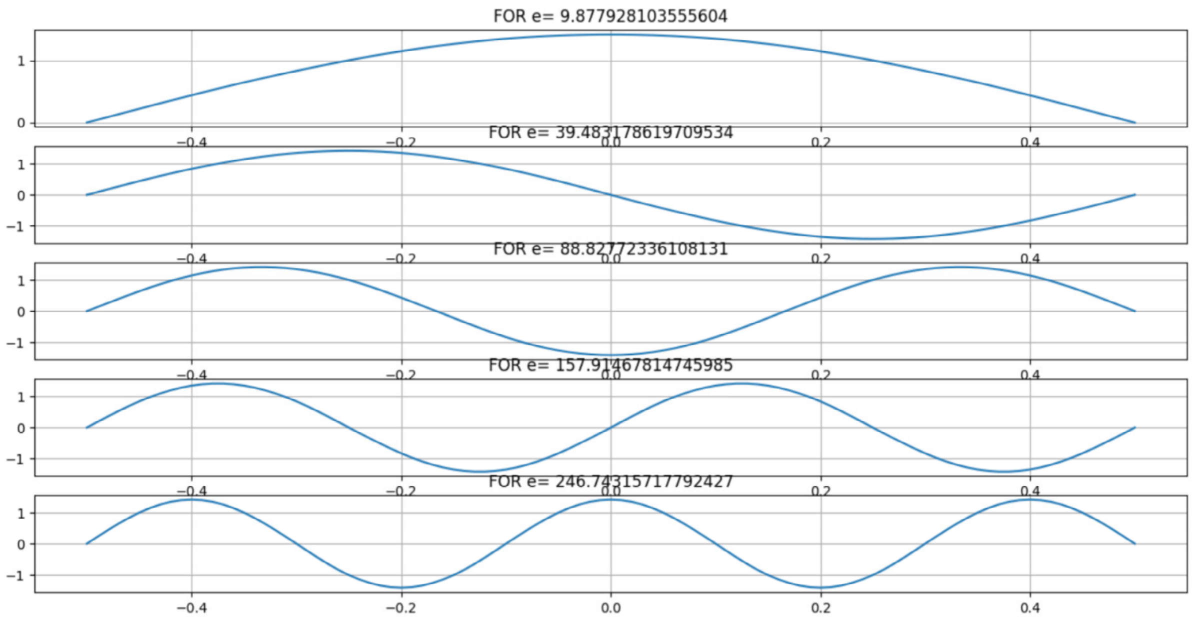
```
C:¥Users¥anura¥AppData¥Local¥Programs¥Python¥Python38¥
python.exe E:/SEM-5/A3/1116_Anurag_qmLab-A3.py
slope:  [9.8696044]
Actual Value Of Slope:  9.869604401089358
R_sqr:  1.0
--------FOR ELECTRON WHEN WIDTH OF WELL= 5 Angstrom
------
   Analytical Energy Value  Eigen ENergy Val
0                 1.486285          1.487539
1                 5.945141          5.945858
2                13.376568         13.376761
3                23.780565         23.780717
4                37.157133         37.157592

--------FOR ELECTRON WHEN WIDTH OF WELL= 10 Angstrom
------
   Analytical Energy Value  Eigen ENergy Val
0                 0.371571          0.371885
1                 1.486285          1.486465
2                 3.344142          3.344190
3                 5.945141          5.945179
4                 9.289283          9.289398

--------FOR ELECTRON WHEN WIDTH OF WELL= 5 Fermi------
   Analytical Energy Value  Eigen ENergy Val
0             8.107820e+06      8.114658e+06
1             3.243128e+07      3.243519e+07
2             7.297038e+07      7.297143e+07
3             1.297251e+08      1.297259e+08
4             2.026955e+08      2.026980e+08

   Expectation Value Of momentum
0     0.000000e+00-1.843365e-06j
```

```
1     0.000000e+00-1.949306e-07j
2     0.000000e+00-5.145239e-07j
3     0.000000e+00-2.866016e-06j
4     0.000000e+00-1.092094e-05j
   Expectation Value Of momentum square
0                      10.964500
1                      43.826328
2                      98.598773
3                     175.285293
4                     273.884904
   Expectation Value Of Uncertainity
0         5.982797e-18+0.000000e+00j
1         1.759760e-17+0.000000e+00j
2         2.767936e-17+0.000000e+00j
3         3.748614e-17+0.000000e+00j
4         4.718980e-17+0.000000e+00j
Probaility Of Finding The Particle Between L/4 to L/4:  0.
8185203577933832

Process finished with exit code 0
```

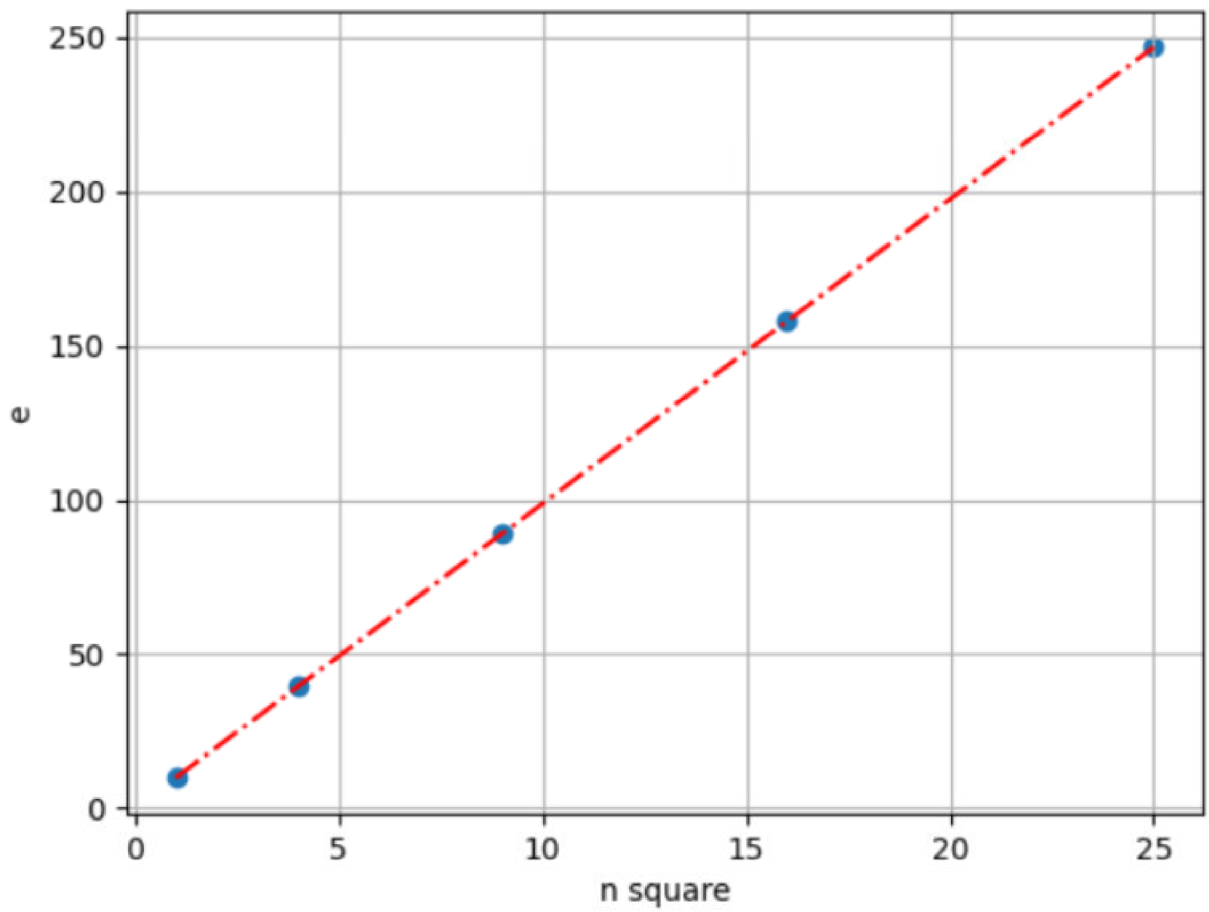
OUTPUT GRAPHS



OUTPUT GRAPHS



OUTPUT GRAPHS



OUTPUT GRAPHS