Quantum Harmonic Oscillator

GROUP NUMBER:1

MEMBERS:

Deepak K.Keshari(21CY40015)
Subhadip Saha(21CY40046)
Kishor Das(21CY40022)

OUTLINE

- **The Harmonic Oscillator Eigenstates**
- **The Classical Harmonic Oscillator**
- **Comparing Classical vs. Quantum Harmonic Results**
- **Plot Eigenvalues and Eigen functions of harmonic oscillator**

Methods Imported

```
import numpy as np
import matplotlib.pyplot as plt
import math
```

```
import matplotlib
import matplotlib.pyplot as plt
import numpy
import numpy.polynomial.hermite as Herm
```

```
import scipy
from scipy.special import hermite
```

```
from celluloid import Camera
from IPython.display import HTML
```

PIP Used

```
!pip install  celluloid
```

- **The Harmonic Oscillator Eigenstates**

```python
def evenHn(x,n):
  n1 = int(n/2)
  sum = 0
  for l in range(0,n1+1):

    s_term = ((-1)**(n1-l))/(fac(2*l)*fac(n1-l))
    sum += s_term * (2*x)**(2*l)
  Hn = fac(n)*sum
  return Hn

def oddHn(x,n):
  n1 = int((n-1)/2)
  sum = 0
  for l in range(0,n1+1):

    s_term = ((-1)**(n1-l))/((fac(2*l))*fac(n1-l))
    sum += s_term * (2*x)**(2*l+1)

  Hn = fac(n)*sum
  return Hn

def gauss(Hn,x):
  f = Hn*math.exp((-1*(x**2))/2)
  return f
```

For Hermite Polynomial when n is even

For Hermite Polynomial when n is odd

- **The Harmonic Oscillator Eigenstates**

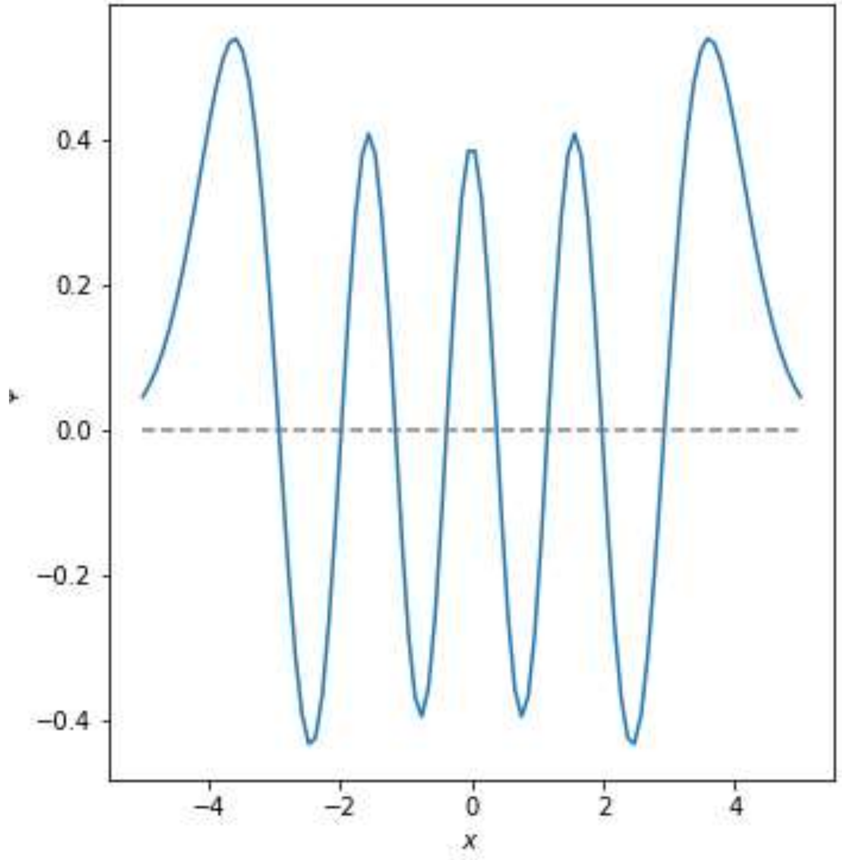
```
def main(n):
  x =np.linspace(-5,5,100)
  h = x[1]-x[0]
  f = []
  f2 = []
  for _ in x:

    if n%2==0:
      Hn = evenHn(_,n)
    else:
      Hn = oddHn(_,n)
    f1 = gauss(Hn,_)
    f.append(f1)
    f2.append(f1**2)

  No =  N(n)
```

Corresponding value of n,

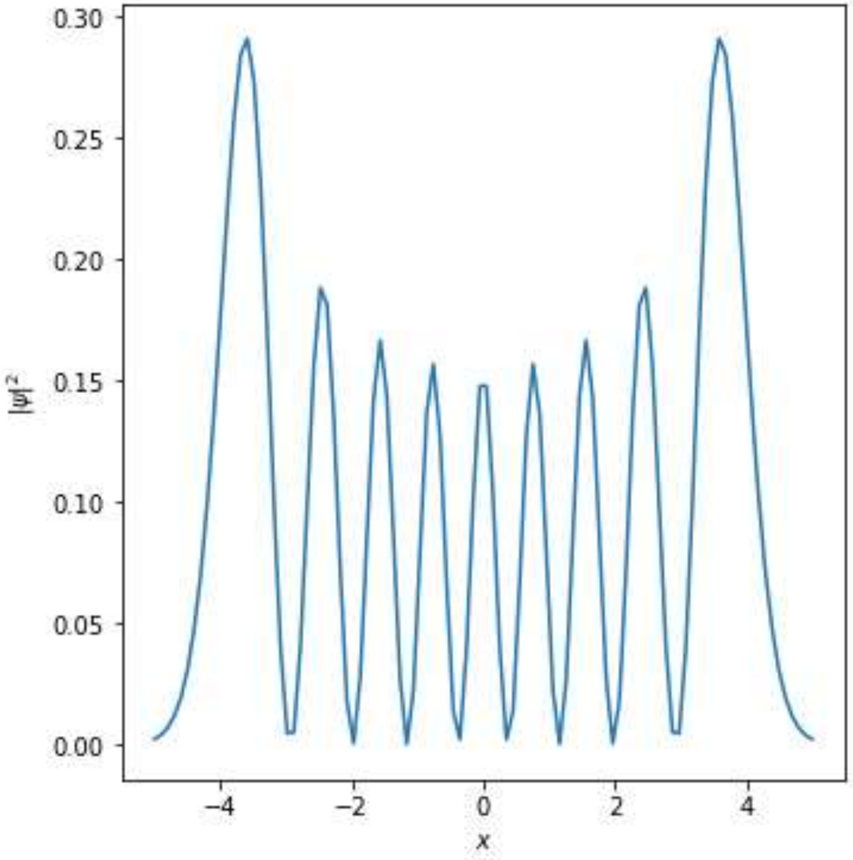
```
  plt.figure(figsize=(12,6))
  plt.subplot(1,2,1)
  plt.plot(x,No*np.array(f))
  plt.plot([-5,5],[0,0], color='gray', linestyle='--')
  plt.xlabel("$x$")
  plt.ylabel("$ψ$")
  plt.subplot(1,2,2)
  plt.plot(x,No**2*np.array(f2))
  plt.xlabel("$x$")
  plt.ylabel("$|ψ|^2$")
n = int(input("Enter the value of n: "))
```

- **The Harmonic Oscillator Eigenstates**

Ψ **VS x**

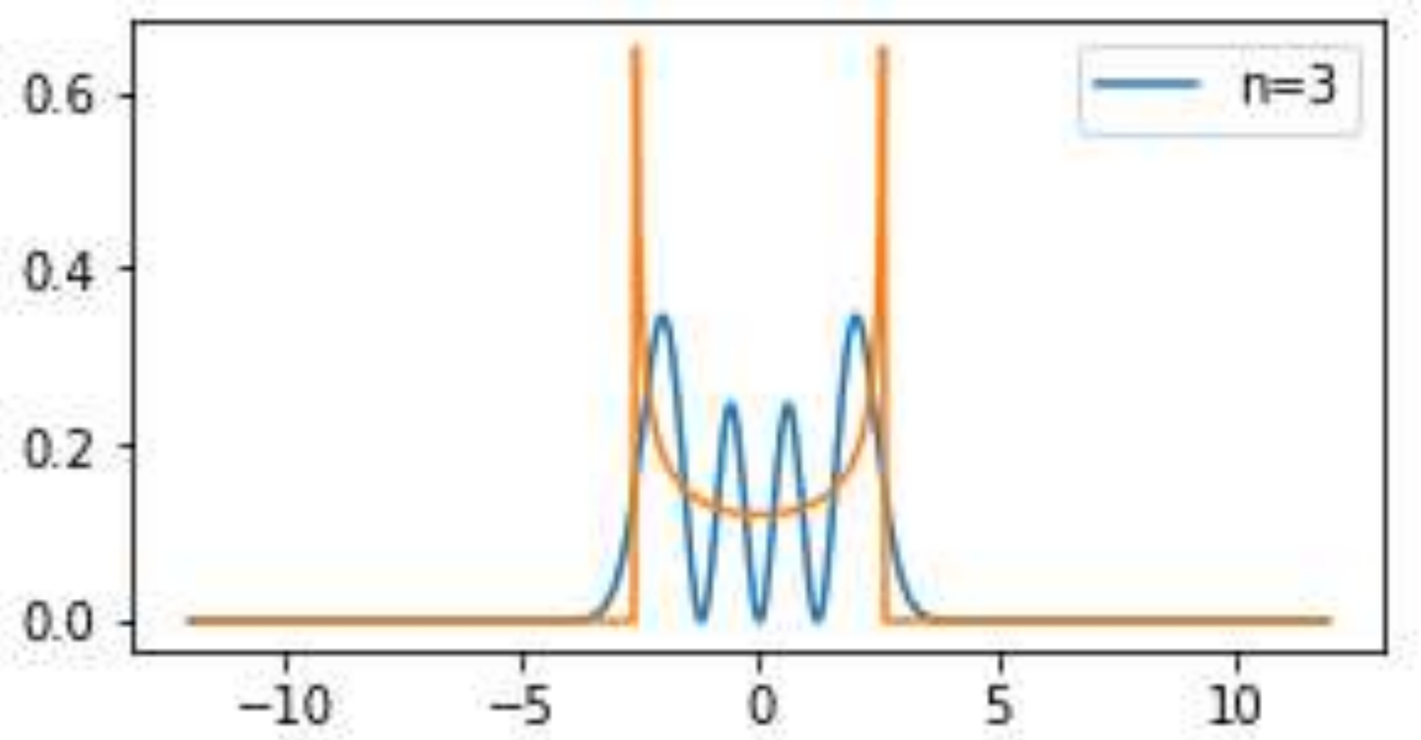


Ψ^2 **VS x**



The Classical Harmonic Oscillator

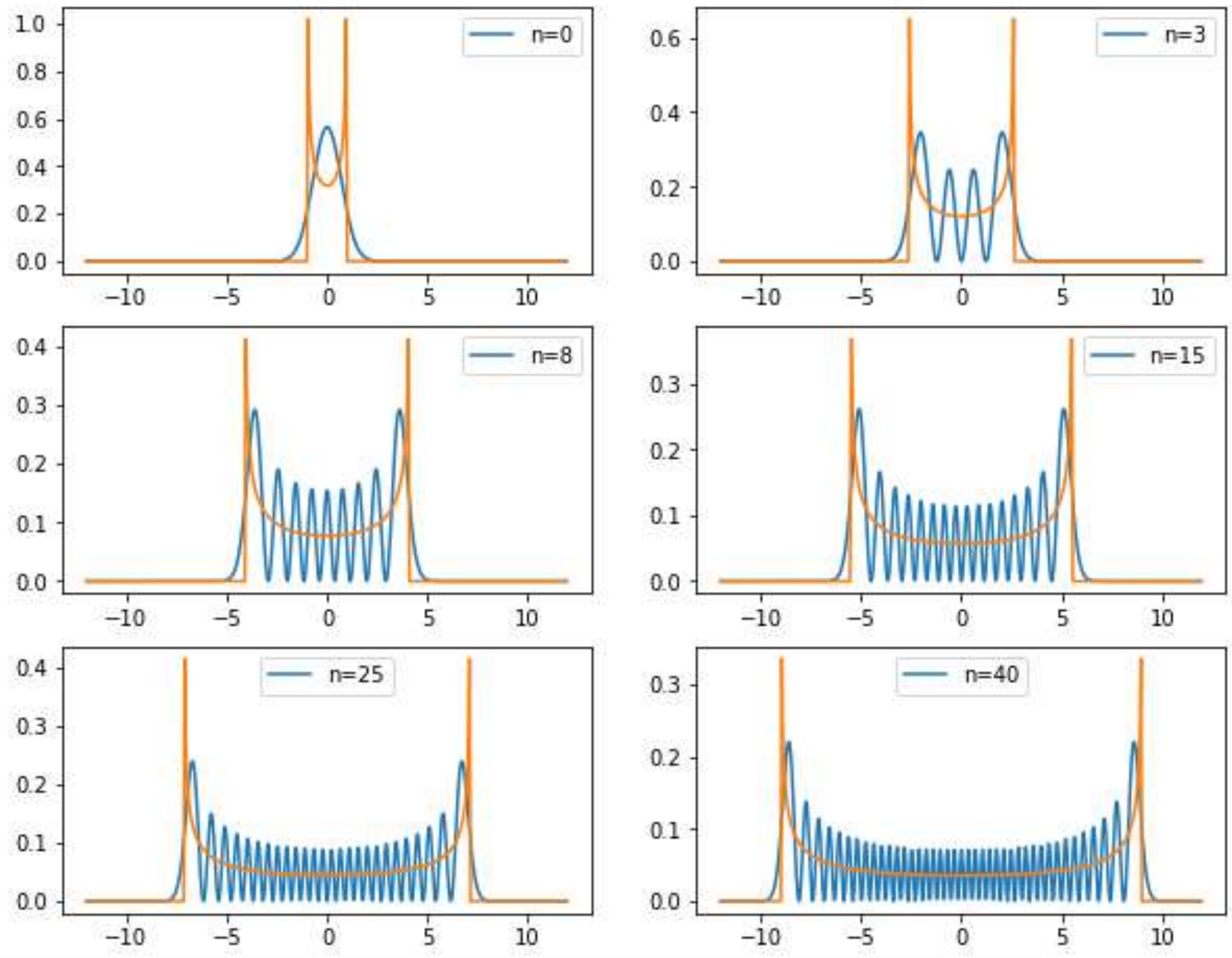
The quantum harmonic oscillator is the quantum-mechanical analog of the classical harmonic oscillator. Because **an arbitrary smooth potential can usually be approximated as a harmonic potential at the vicinity of a stable equilibrium point**, it is one of the most important model systems in quantum mechanics.



The Classical Harmonic Oscillator

```
def N(n):
    '''Normalization constant '''

    return 1./np.sqrt(np.sqrt(np.pi)*2**n*factorial(n))

def psi(x, n ):
    """Harmonic oscillator wavefunction for level n computed on grid of points x"""

    Hr=hermite(n)

    ψx = N(n)*Hr(x)*np.exp(-0.5*x**2)

    return ψx

def classical_P(x,n):
    E = hbar*w*(n+0.5)
    x_max = numpy.sqrt(2*E/(m*w**2))
    classical_prob = numpy.zeros(x.shape[0])
    x_inside = abs(x) < (x_max - 0.025)
    classical_prob[x_inside] = 1./numpy.pi/numpy.sqrt(x_max**2-x[x_inside]*x[x_inside])
    return classical_prob
```

Comparing Classical vs. Quantum Harmonic Results



Plot Eigenvalues and eigenfunctions of harmonic oscillator

```
def E(n): #Eigen values

    '''Eigenvalues in units of h'''

    return (n + 0.5)

def V(x):  #potential energy
    """Potential energy function"""

    return 0.5*x**2
```

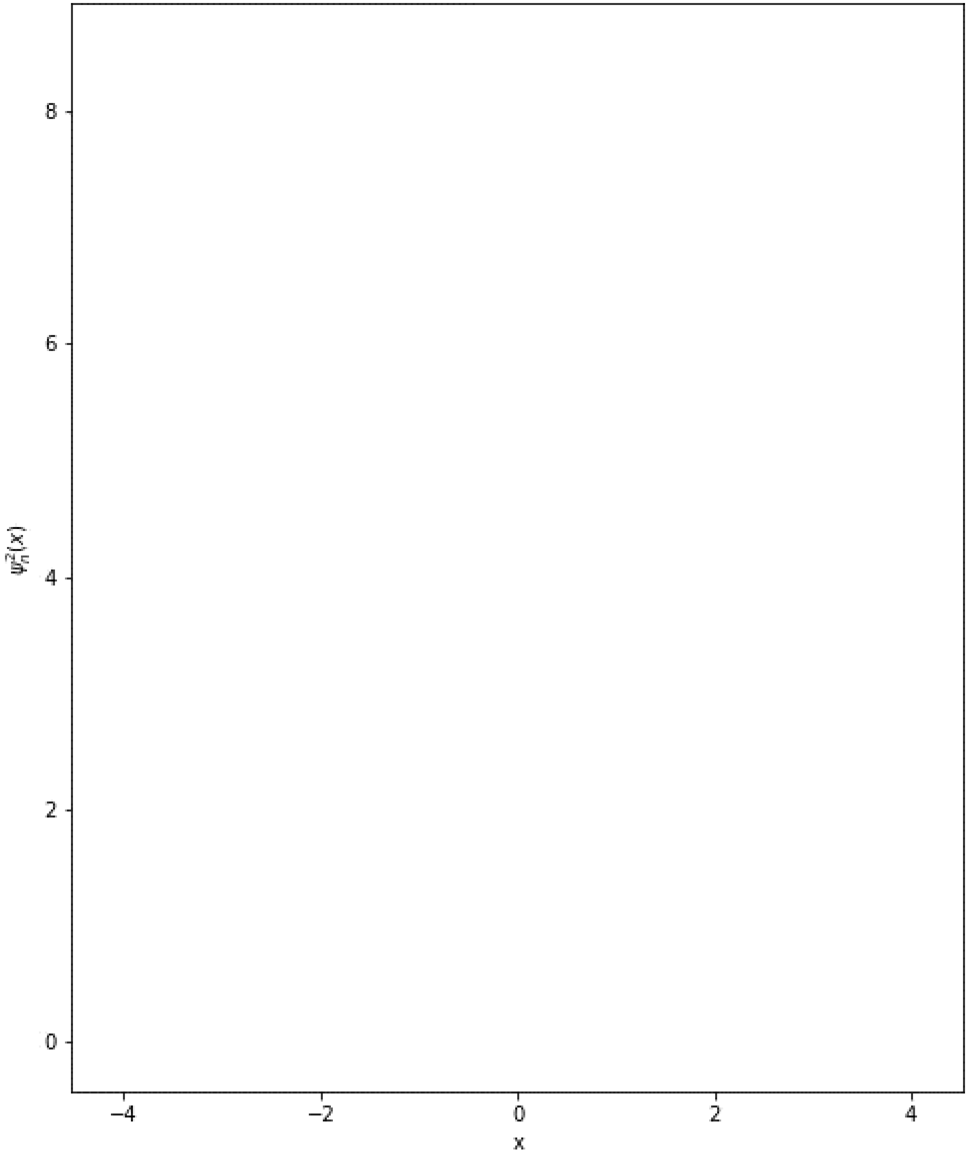
Plot Eigenvalues and eigenfunctions of harmonic oscillator

Module used:

```
!pip install  celluloid

import numpy as np
import matplotlib.pyplot as plt

from celluloid import Camera
from IPython.display import HTML

fig, bx = plt.subplots(figsize = (8,10))
camera = Camera(fig)
plt.close()
```

Plot of Probability densities of harmonic oscillator

On Increasing the level,
the probability shift more toward
classical probability density



Future prospects and Applications

The harmonic oscillator is a model which has several important applications in both classical and quantum mechanics.

It serves as a prototype in the mathematical treatment of such diverse phenomena as

elasticity, acoustics, AC circuits,
molecular and crystal vibrations,
electromagnetic fields and
optical properties of matter.

References

- *https://scipython.com/blog*
- *https://youtube.com/shorts/p-plveDPwfc?feature=share*
- *https://helentronica.com/2014/12/28/qm-with-python-swing-on-the-quantum-harmonic-oscillator/*
- *https://chem.libretexts.org/Ancillary_Materials/Interactive_Applications*
- *https://opentextbc.ca/universityphysicsv3openstax/chapter/the-quantum-harmonic-oscillator/*
- https://github.com/agarret7/QHO-Visualizer/tree/master/src
- *https://en.wikipedia.org/wiki/Quantum_harmonic_oscillator*

Thank You