Landau Levels

Álvaro Cauqui

Universidad Carlos III de Madrid

Advanced Quantum Mechanics - Group 111

March 2, 2024

Contents

1 Introduction

Aim

- To numerically calculate the wavefunction and probability density of a Landau state and study convergence with classical mechanics at high level indexes

2 Landau state wavefunctions

Quantum Harmonic oscillator

The wavefunctions of a quantum harmonic oscillator are:

$$\psi_n(x) = \frac{1}{\sqrt{2^n n!}} (\frac{mw}{\hbar\pi})^{1/4} e^{-mwx^2/2\hbar} H_n\Big(\sqrt{\frac{mw}{h}}x\Big)$$

H_n being the Hermite polynomials. Thus the ground state and first excited state (ψ_0 and ψ_1) are:

$$\psi_0 = \Big(\frac{mw}{\hbar\pi}\Big)^{1/4} e^{\frac{-mwx^2}{2\hbar}}$$

$$\psi_1 = \Big(\frac{4m^3w^3}{\hbar^3\pi}\Big)^{1/4} xe^{\frac{-mwx^2}{2\hbar}}$$

Non-dimensionalization

We can introduce a variable ξ such that:

$$\xi = \frac{mw}{\hbar}x = \xi_0 x$$

Thus the ground and first excited states are written as:

$$\psi_0 = \Big(\frac{\xi_0}{\pi}\Big)^{1/4} e^{\frac{-\xi^2}{2\xi_0}}$$

$$\psi_1 = \Big(\frac{4}{\xi_0\pi}\Big)^{1/4} \xi e^{\frac{-\xi^2}{2\xi_0}}$$

In this form, the n-th wavefunction can be written in terms of the recursive equation:

$$\psi_n(\xi) = \sqrt{\frac{2}{n}}\Big(\xi\psi_{n-1}(\xi) - \sqrt{\frac{n-1}{2}}\psi_{n-2}(\xi)\Big)$$

This way, all solutions are self similar and easier to scale in the numerical simulation.
Finally, the probability distribution is just the square of the obtained wavefunctions.
.

3 Results

3.1 Wavefunctions

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

hbar=1.054571817e-34
pi=np.pi
m=9.109e-31
e=1.6e-19
B=hbar/e
w=(e*B)/m
xi0=(m*w)/hbar
```

First step is defining variables, the frequency ω is the **Cyclotron frequency**

```
def xi(pos):
    return xi0*pos
def phi0(xi_):
    return (xi0/pi)**(0.25)*np.exp(((-xi_**2)/2*xi0))
def phi1(xi_):
    return (4/xi0*pi)**(0.25)*xi_*np.exp(((-xi_**2)/2*xi0))
```

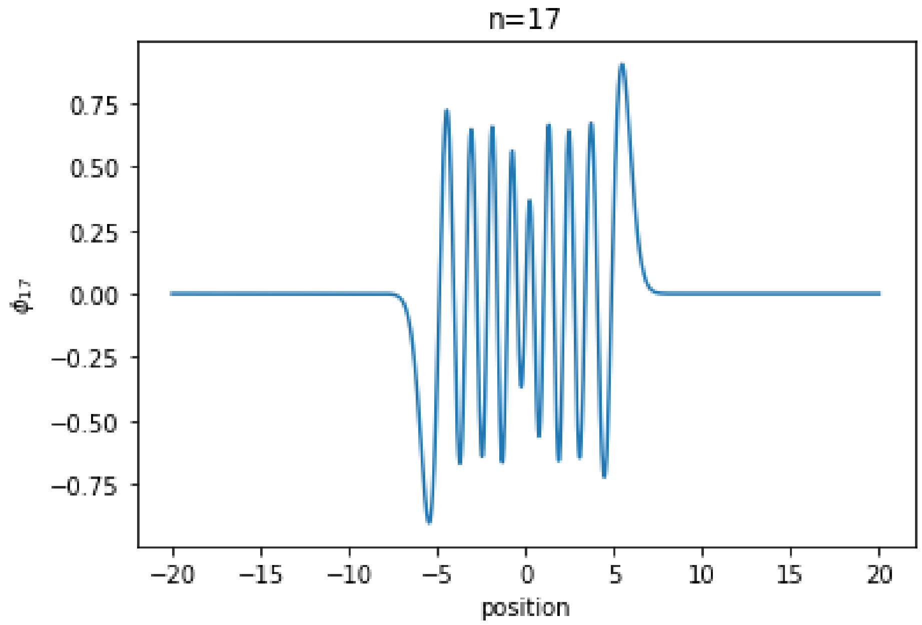
The equations are defined as above. Finally, an algorithm is constructed such that it selects applies the recursive forumula to obtain a wavefunction, and adds it to an array. The result is an array containing all wavefunctions from $n = 0$ to $n = 100$ in terms of ξ

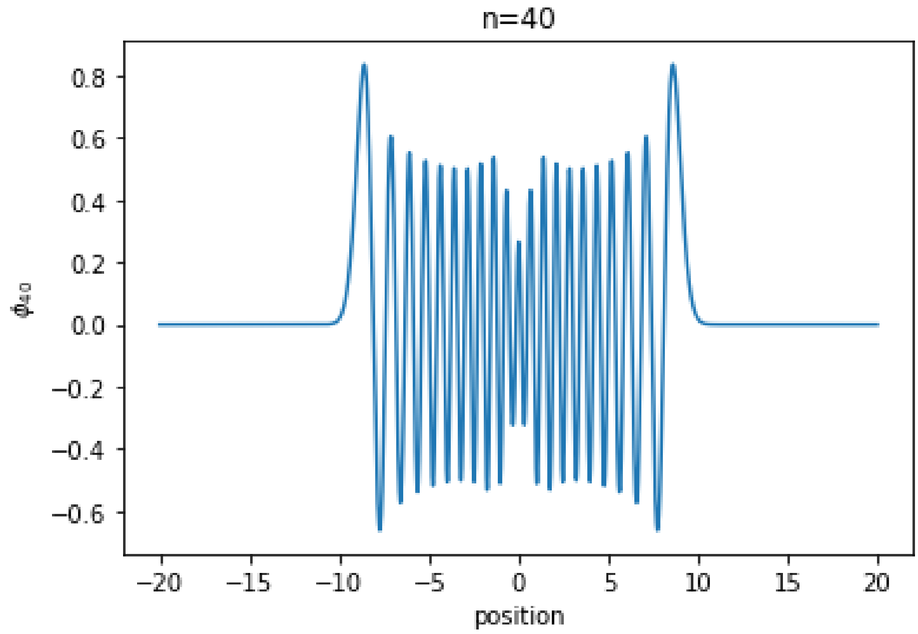
```
X=np.arange(-20,20,0.00001)
xii=xi(X)

wavefunctions=np.zeros((100,len(X)))
wavefunctions[0]=phi0(xii)
wavefunctions[1]=phi1(xii)
for i in range(2,100,1):
    wavefunctions[i]=np.sqrt(2/i)*(xii*wavefunctions[i-1]-np.sqrt((i-1)/2)*wavefunctions
    [i-2])
```

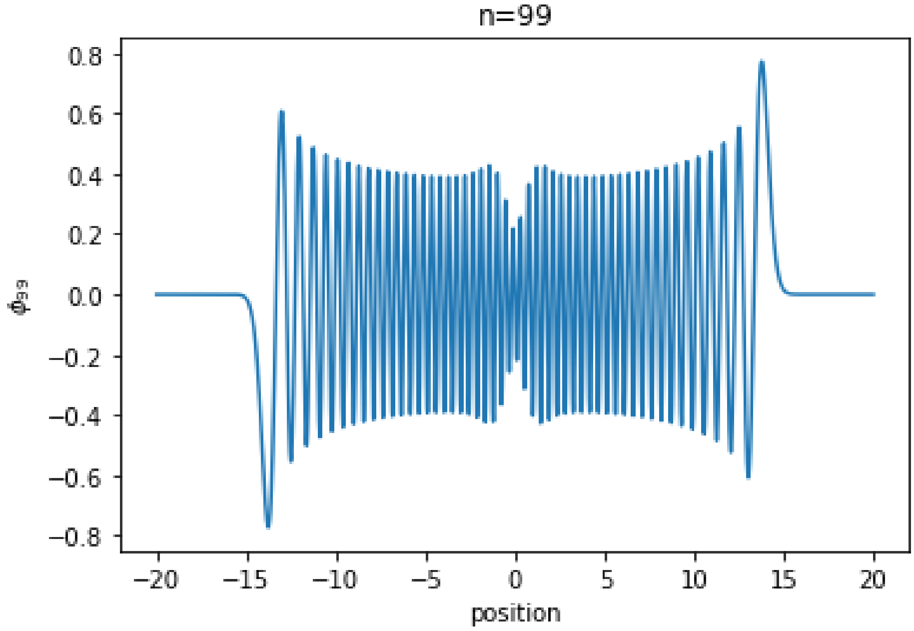
Finally,the results are plotted.

```
plt.figure()
plt.plot(xii,wavefunctions[17])
plt.title('n=17')
plt.ylabel('$\phi_{17}$')
plt.xlabel('position')
plt.figure()
plt.plot(xii,wavefunctions[40])
plt.title('n=40')
plt.ylabel('$\phi_{40}$')
plt.xlabel('position')
plt.figure()
plt.plot(xii,wavefunctions[99])
plt.title('n=99')
plt.ylabel('$\phi_{99}$')
plt.xlabel('position')
plt.figure()
plt.plot(xii,wavefunctions[17],label='n=17')
plt.plot(xii,wavefunctions[40],label='n=40')
```

```
plt.plot(xii,wavefunctions[99],label='n=99')
plt.legend()
```

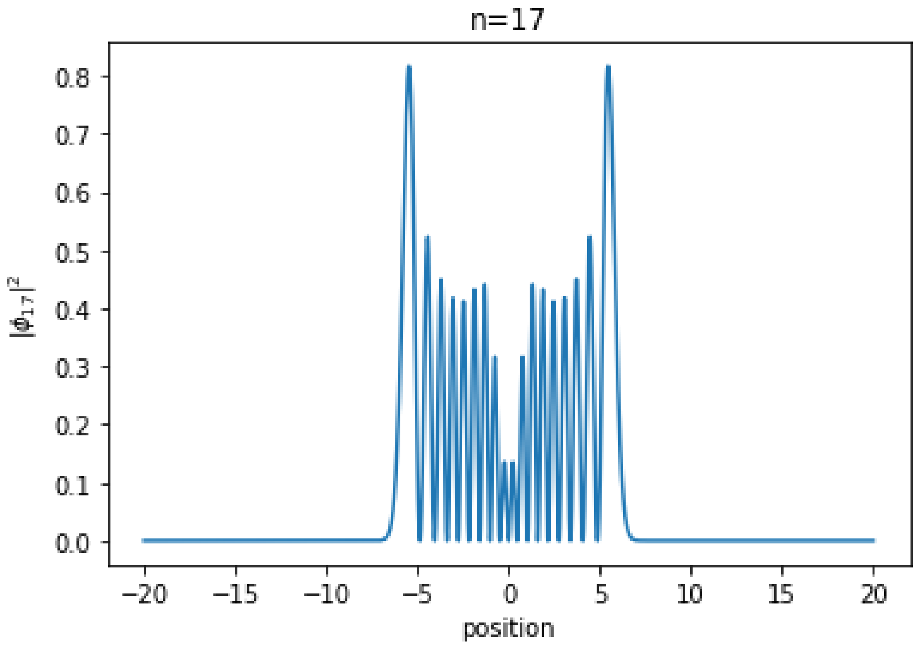


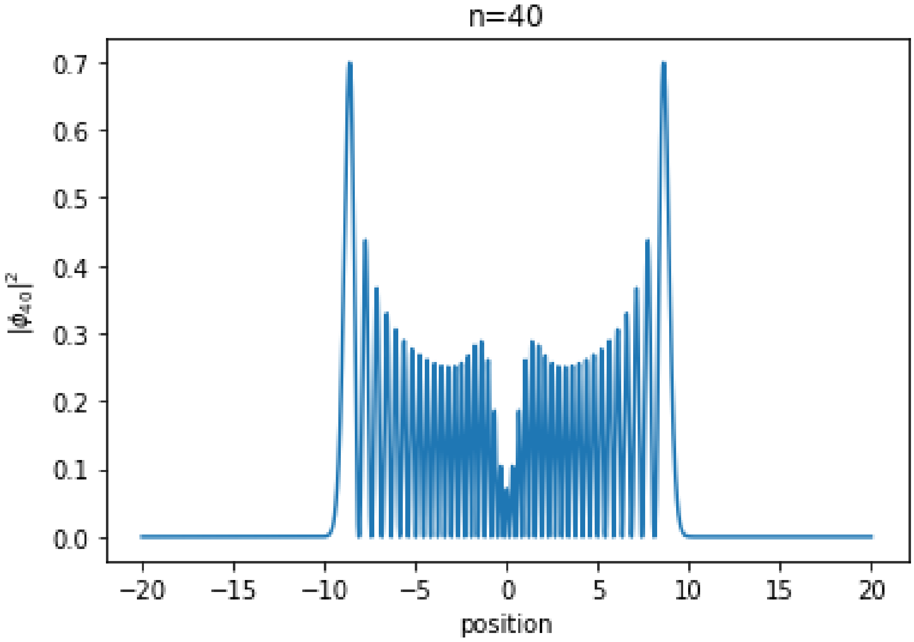


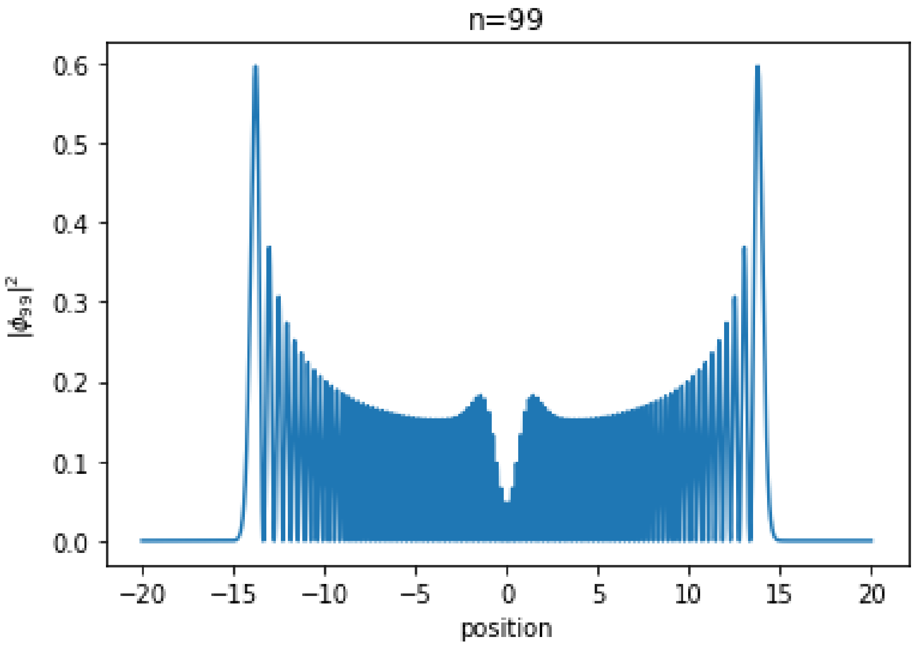


Probability distributions

Once the wavefunctions are obtained, the probability distributions are obtained straigthfoward:




n=40
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
$|\phi_{40}|^2$
−20
−15
−10
−5
0
5
10
15
20
position

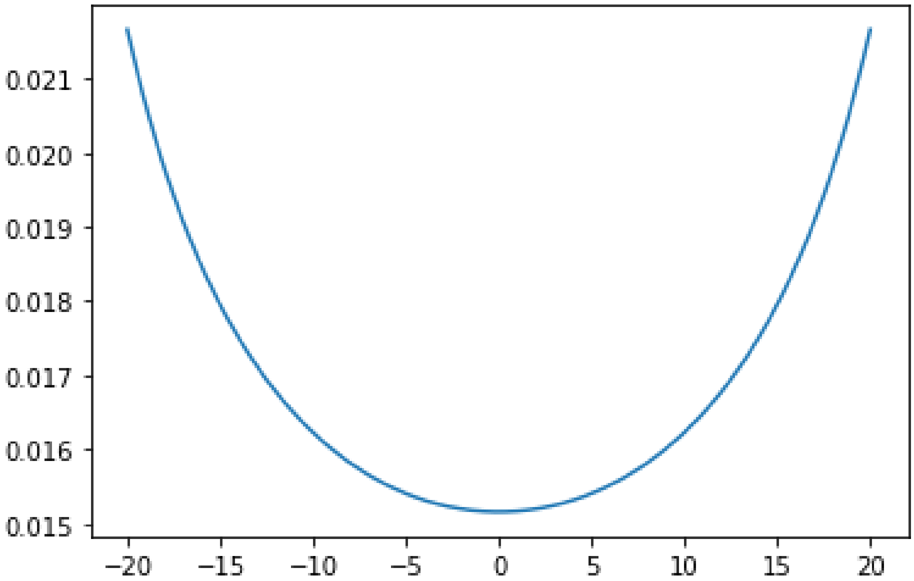

n=99
0.6
0.5
0.4
0.3
0.2
0.1
0.0
$|\phi_{99}|^2$
−20
−15
−10
−5
0
5
10
15
20
position

4 Classical limit

We can calculate the probability of finding a classical oscillating particle at a position x:

$$p(x) = \frac{1}{\pi}\frac{1}{\sqrt{A^2 - x^2}}$$

Where A is the amplitude of the oscillation.



We can see in the shape of this function that it would be reasonable to expect that the wavefunctions will converge to this limit as $n \to \infty$