Quantum Harmonic Oscillator

Ross McPhee

October 3, 2024

1 Introduction

The quantum harmonic oscillator is a fundamental model in quantum mechanics, describing a particle bound in a quadratic potential. This project provides a numerical and visual exploration of the quantum harmonic oscillator using Hermite polynomials. The goal is to gain insights into the wavefunctions, probability distributions, and physical properties of the oscillator for various quantum states.

This project is implemented in Python and utilizes numerical methods to compute wavefunctions and visualize their properties. It includes the generation of Hermite polynomials, computation of wavefunctions for different quantum states, calculation of expectation values, and visualization of probability densities.

2 Features

2.1 Hermite Polynomial Generation

The Hermite polynomials are a set of orthogonal polynomials that arise in the wavefunctions of the quantum harmonic oscillator. This project uses the recurrence relation:

$$H_{n+1}(x) = 2xH_n(x) - 2nH_{n-1}(x)$$

to generate Hermite polynomials up to a specified order. This method is computationally efficient and forms the basis for constructing the oscillator's wavefunctions.

2.2 Wavefunction Calculation

The wavefunctions of the harmonic oscillator are given by:

$$\psi_n(x) = N_n H_n(x) e^{-\frac{x^2}{2}}$$

where $H_n(x)$ is the Hermite polynomial of order n, and N_n is a normalization factor ensuring the wavefunction's total probability is unity. These wavefunctions are essential in quantum mechanics as they represent the possible states of a particle in the oscillator's potential.

In this project, the wavefunctions are calculated for a range of quantum numbers n. The normalization process is handled numerically, which allows for accurate probability density computations.

2.3 Expectation Value Computation

The expectation value of a physical quantity in a given quantum state provides insights into the average outcome of measurements. This project calculates the expectation values of the position $\langle x \rangle$ and the uncertainty (Δx) using:

$$\langle x \rangle = \int_{-\infty}^{\infty} x |\psi_n(x)|^2 \, dx$$

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

These quantities provide information about the particle's localization in different quantum states, showcasing the probabilistic nature of quantum mechanics. Numerical integration methods are used to compute these values over a defined spatial range.

2.4 Visualization

Visualization is key to understanding the quantum harmonic oscillator. This project includes functions to plot the wavefunctions $\psi_n(x)$ and their corresponding probability densities $|\psi_n(x)|^2$ for various quantum states. Users can explore how the shape and spread of the wavefunctions change with increasing quantum numbers.

Optional features include time evolution animations, showing how wave packets evolve in the oscillator's potential, further illustrating quantum superposition and dynamics.

3 Installation

To run this project, you need Python 3.x and several scientific libraries. Follow these steps to set up the environment:

1. Clone the repository from GitHub:

```
git clone https://github.com/RossMcphee23/quantum-harmonic-oscillator.git
```

2. Navigate to the project directory:

```
cd quantum-harmonic-oscillator
```

3. Install the required dependencies using pip:

```
pip install -r requirements.txt
```

4 Project Structure

The project is organized into the following modules and files:

- **main.py**: The main script that allows users to interact with the program. It provides options to select quantum states, plot wavefunctions, and calculate physical properties.
- **hermite.py**: Contains functions for generating Hermite polynomials using the recurrence relation. This module is the foundation for constructing the oscillator's wavefunctions.
- **wavefunction.py**: Computes the harmonic oscillator wavefunctions and performs normalization. It also includes functions for calculating expectation values and uncertainties.
- **visualization.py**: Provides functions for visualizing wavefunctions, probability densities, and (optionally) time evolution animations.
- **requirements.txt**: Lists the Python libraries necessary to run the project, ensuring a consistent environment for all users.

5 Dependencies

This project relies on the following Python libraries:

- **NumPy**: For numerical computations, including the evaluation of Hermite polynomials and integration for expectation values.
- **Matplotlib**: To plot wavefunctions, probability densities, and create visualizations.
- **SciPy**: For numerical integration and additional scientific computations.

Ensure that these dependencies are installed using the 'requirements.txt' file to maintain a consistent environment.

6 Contact

For questions, suggestions, or contributions, please contact Ross McPhee at rm378@st-andrews.ac.uk. Contributions are welcome through pull requests on the GitHub repository.