Quantum Harmonic Oscillator

Assignment 4

Gabriel Amorosetti[1*]

[1]MSc Physics of Data, Università degli Studi di Padova

Abstract

In this study, we investigate the quantum harmonic oscillator, which is a fundamental quantum mechanical system that describes a particle in a quadratic potential well. Using the Schrödinger equation, we employ the Finite Difference Method (FDM) to discretize and solve for the eigenvalues and eigenvectors of the system. We focus on approximating the second derivative term in the Hamiltonian and extend the method to higher-order approximations for improved accuracy. The results of this numerical approach are compared with analytical solutions, and we evaluate the stability and efficiency of the method. This work provides a reliable framework for solving quantum mechanical problems where analytical solutions may not be feasible.

Introduction

The quantum harmonic oscillator is the quantum analogue of the classic harmonic oscillator and is one of the few quantum systems for which an exact analytical solution is known. This fundamental problem in quantum mechanics models the behaviour of a particle in a quadratic potential well. The Hamiltonian of this particle is :

$$\hat{H} = \frac{\hat{p}^2}{2m} + \frac{1}{2}m\omega^2\hat{q}^2 \tag{1}$$

with the mass m, the angular frequency ω, the position $\hat{q}$ and the impulsion $\hat{p} = -i\hbar\nabla$. In order to study this problem, one should solve the Schrödinger equation :

$$\hat{H}\Psi(x) = E\Psi(x) \tag{2}$$

where $\hat{H}$ is the Hamiltonian operator, $\Psi(x)$ are the eigenfunctions, and E are the corresponding eigenvalues (energy levels), that are crucial to describe the quantum states of the system.

As analytical solutions are known for this problem, it makes it suitable to implement and test numerical methods to solve the equation 2. This report describes a numerical approach to compute the first k eigenvalues E_k and eigenvectors $|\Psi_k\rangle$ by discretizing the Schrödinger equation. This study implements the finite difference method to find approximate solutions (also extended to higher order), includes a comparison with analytical solutions, and tries to evaluate the stability and efficiency of the code.

Methodology

Model setup

In this context,

- $\hbar$ and m will be set as natural units, therefore equals to 1,
- the problem will be limited to the case of one particle in one spatial dimension, $\hat{p} = -i\hbar\frac{\partial}{\partial x}$.

The Hamiltonian can now be expressed as [1]:

$$\hat{H} = \hat{T} + \hat{V} = -\frac{1}{2}\frac{\partial^2}{\partial x^2} + \frac{1}{2}\omega^2 x^2 \tag{3}$$

In order to find the first k eigenvalues E_k and eigenvectors $|\Psi_k\rangle$, one needs to solve the Schrödinger equation :

$$H\left|\Psi_k\left(x_i\right)\right\rangle = E_k\left|\Psi_k\left(x_i\right)\right\rangle \tag{4}$$

$$\left[\frac{1}{2}\left(-\nabla_x^2 + \omega^2 x^2\right)\right]\left|\Psi_k\left(x_i\right)\right\rangle = E_k\left|\Psi_k\left(x_i\right)\right\rangle \tag{5}$$

Finite Difference Method

The Finite Difference Method (FDM) is a numerical technique for approximating solutions by transforming the differential equation into a matrix equation, useful when an analytical solution is not available.

The x axis is discretized : instead of using a continuous variable, a discrete grid of N points will be used. The study will be limited to $x \in [a, b]$ and the interval will be divided by N points, with $\Delta x = \frac{b-a}{N-1}$ the discretization space between each point. Therefore, $x_i = a + i\Delta$ where $i = 0, 1, 2, ..., N-1$ and the wave function $\Psi(x)$ is represented as $[\Psi(x_0), \Psi(x_1), ..., \Psi(x_{N-1})]^T$.

In our case, $\hat{V} = \frac{1}{2}\omega^2 x^2$, the potential term of the Hamiltonian (3), is easy to evaluate, but the kinetic term $\hat{T} = -\frac{1}{2}\frac{\partial^2}{\partial x^2}$ needs to be approximated numerically. Hence, this second derivative term in the Schrödinger equation is approximated as :

*gabriel.amorosetti@studenti.unipd.it

Due by : November 18, 2024

[1] For convenience, we will set later $\omega = 1$ during computation

$$-\frac{1}{2}\frac{\partial^2\Psi}{\partial x^2}\bigg|_{x_i} \approx \frac{-\Psi(x_{i+1})+2\Psi(x_i)-\Psi(x_{i-1})}{2\Delta x^2} + \mathcal{O}\left(\Delta x^2\right) \tag{6}$$

which is represented as a tridiagonal matrix :

$$\hat{K} = \frac{1}{2\Delta x^2}\begin{pmatrix} 2 & -1 & 0 & \dots & 0 & 0 \\ -1 & 2 & -1 & \dots & 0 & 0 \\ 0 & -1 & 2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 2 & -1 \end{pmatrix}$$

and the potential term is represented by a diagonal matrix :

$$\hat{V} = \frac{\omega^2}{2}\begin{pmatrix} x_1^2 & 0 & 0 & \dots & 0 & 0 \\ 0 & x_2^2 & 0 & \dots & 0 & 0 \\ 0 & 0 & x_3^2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & x_N^2 \end{pmatrix}$$

The Finite Difference Method can be extended to a higher order version in order to have a more accurate approximation of the second derivative term. By using Taylor expansions, the formula can be extended to the fourth order :

$$-\frac{1}{2}\frac{\partial^2\Psi}{\partial x^2}\bigg|_{x_i} \approx \frac{-\Psi(x_{i+2})+16\Psi(x_{i+1})-30\Psi(x_i)+16\Psi(x_{i-1})-\Psi(x_{i-2})}{12\Delta x^2} + \mathcal{O}\left(\Delta x^4\right) \tag{7}$$

which is represented as a pentadiagonal matrix (while the potential term remains the same) :

$$\hat{K} = \frac{1}{2\Delta x^2}\begin{pmatrix} 5/2 & -4/3 & 1/12 & \dots & 0 & 0 \\ -4/3 & 5/2 & -4/3 & \dots & 0 & 0 \\ 1/12 & -4/3 & 5/2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 5/2 & -4/3 \end{pmatrix}$$

The problem being now represented as a matrix equation, it can be solved using matrix diagonalization, as explained in the following section.

Implementation

In order to solve the problem formulated as a matrix equation using the Finite Difference Method, we implement in Python a function `quantum_harmonic_oscillator(N, a, b, omega, higher_order = False)`, with N the number of points of the x discrete grid, a and b its left and right boundaries, *omega* the angular frequency ω, and `higher_order` a flag to choose to use the second order or higher (fourth) order FDM to compute the solutions.

The grid of N points is created with `x = np.linspace(a, b, N)`, followed by the definition of Δx : `dx = x[1] - x[0]`.

Then, the $\hat{V}$ and $\hat{T}$ matrices can be defined, both for the second and higher order methods, following their descriptions introduced in the last section :

```
# Potential matrix V
V = 0.5 * omega**2 * np.diag(x**2)

# Kinetic energy matrix T, using the finite difference method
if higher_order :

    main_diag = (5/2) * np.ones(N) / (dx**2)
    off_diag1 = -4/3 * np.ones(N-1) / (dx**2)
    off_diag2 = 1/12 * np.ones(N-2) / (dx**2)
    T = np.diag(main_diag) + np.diag(off_diag1, k=1) + np.diag(off_diag1, k=-1)
    T += np.diag(off_diag2, k=2) + np.diag(off_diag2, k=-2)

else :

    main_diag = 2 * np.ones(N) / (dx**2)
    off_diag = -1 * np.ones(N-1) / (dx**2)
    T = np.diag(main_diag) + np.diag(off_diag, k=1) + np.diag(off_diag, k=-1)

T *= 0.5  # Factor of 1/2 of the kinetic energy term
```

Figure 1: *$\hat{V}$ and $\hat{T}$*

They are summed up to create the Hamiltonian matrix `H = T + V` and the eigenfunctions and eigenvalues can be finally computed with a function from the `scipy.linalg` package :
`eigenvalues, eigenfunctions = eigh(H)`.

Results

The methods described in the previous sections allow us to compute n eigenfunctions and their associated eigenvalues.

Table 1: *Comparison of computed and analytical eigenvalues*

n	Computed eigenvalue	Analytical eigenvalue
0	0.5000	0.5000
1	1.5000	1.5000
2	2.5000	2.5000
3	3.5000	3.5000
4	4.5000	4.5000
5	5.5001	5.5000
6	6.5006	6.5000
7	7.5029	7.5000
8	8.5113	8.5000
9	9.5360	9.5000
10	10.5950	10.5000
11	11.7108	11.5000
12	12.9048	12.5000
13	14.1916	13.5000
14	15.5788	14.5000
15	17.0693	15.5000

We represented on the figure 2 all the computed eigenfunctions and on the table 1 all the computed eigenvalues, for $n = 0$ to 15.

The analytical solutions are also represented in order to discuss our results in the next section.

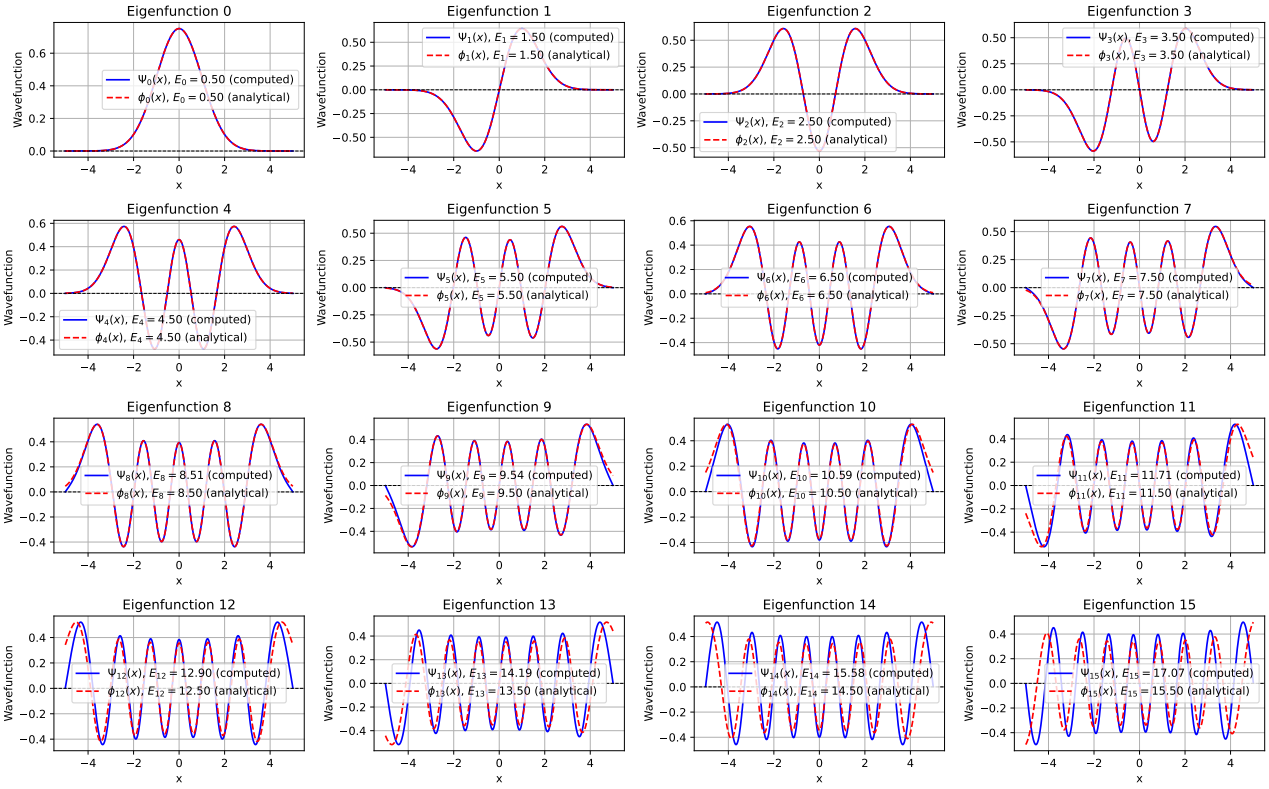

Figure 2: *Computed eigenfunctions from* 0 *to* 15 *(Higher order FDM)*

Discussion

Correctness

In order to evaluate the correctness of our solutions, we need to evaluate the analytical solutions of :

$$\hat{H}\,|\psi_n\rangle = \left(\frac{\hat{p}^2}{2m} + \frac{1}{2}m\omega^2\hat{x}^2\right)|\psi_n\rangle = E_n\,|\psi_n\rangle$$

In this case, the analytical solutions exist and are known. The eigenvalues can be analytically described by :

$$E_n = \hbar\omega\left(n + \frac{1}{2}\right)$$

and the eigenfunctions by the following equation :

$$\psi_n(x) = \frac{1}{\sqrt{2^n n!}}\left(\frac{m\omega}{\pi\hbar}\right)^{1/4} e^{-\frac{m\omega x^2}{2\hbar}} H_n\left(\sqrt{\frac{m\omega}{\hbar}}x\right)$$

with

$H_n(x) = (-1)^n e^{z^2}\dfrac{\mathrm{d}^n}{\mathrm{d}z^n}\left(e^{-z^2}\right)$ the Hermite polynomial.

The analytical solutions are represented and listed along with the computed ones, for $n = 0$ to 15 on the figure 2 and the table 1.

At first, the computed solutions seem to be coherent with the analytical ones. However, one can notice that as n grows, the difference between the two solutions increases.

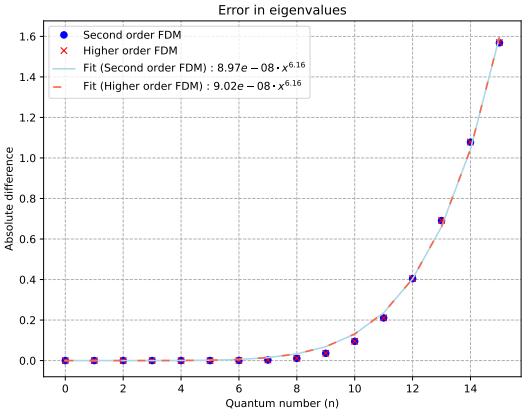

Figure 3: *Eigenvalue error depending on the quantum number*

Indeed, as depicted in the figure 3, the error (absolute difference between the computed and analytical eigenvalue) seems to follow a polynomial behaviour as n grows. This shows us a limit in the correctness of our computed solutions, as a more precise approximation should be implemented in order to stay coherent with the analytical solutions.

Stability

In order to evaluate the stability of the solutions provided by our algorithm, we compare the results of several runs of the code. Moreover, we qualitatively evaluate the influence of the number of grid points on the stability of the solutions.

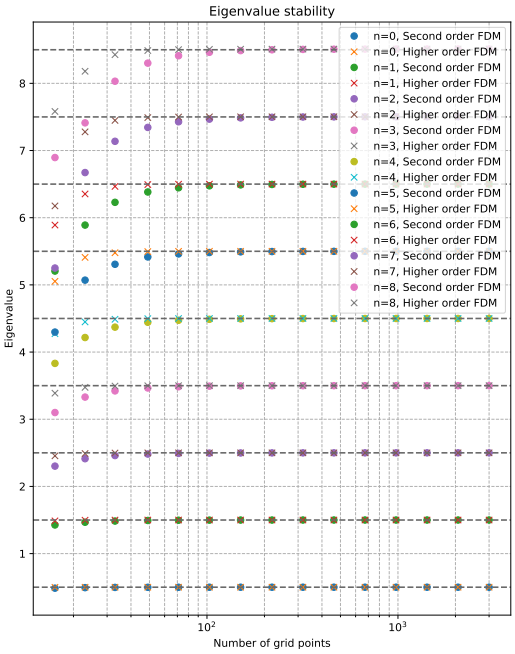

Figure 4: *Eigenvalues for different discretization steps*

For a low number of grid points ($N < 100$), one cannot claim with confidence the validity of the solutions, as they differ from one N to another, especially when the quantum number n increases. However, over $N > 100$ the solutions appear to be stable for every n.

Accurate discretization

The influence of the size of the discretization step is evaluated by computing the absolute difference between the computed and the analytical eigenvalues, only for the first and the fifteenth ones. One first observation from the figures 5 and 6 is that it seems that the bigger the quantum number n, the more prone it is to have a bigger difference to the analytical solution. Indeed, the first eigenvalue error is, at least, two orders of magnitude lower than the one of the 15^{th} eigenvalue. This phenomenon is also seen on the figure 4, where higher n eigenvalues show a more important difference to the analytical solutions (grey dashed lines). It's worth noticing from figure 4 that using the higher-order FDM lowers the errors, especially for low numbers of grid points. Furthermore, the bigger the number of grid points, the lower the error is for the first eigenvalue, whereas, after a certain threshold (around $N = 10^2$), the error for the fifteenth eigenvalue does not seem to improve. One can generalize from this section and the previous one that a large enough number of discretization points, above $N = 10^2$, is necessary in order to ensure stability of the solutions and closeness to the analytical solutions.

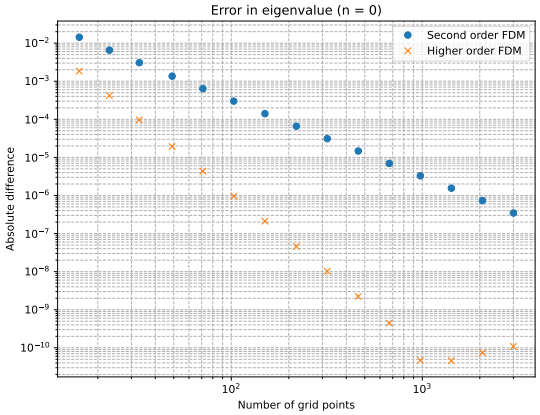

Figure 5: *Eigenvalue error depending on the number of grid points ($n = 0$)*

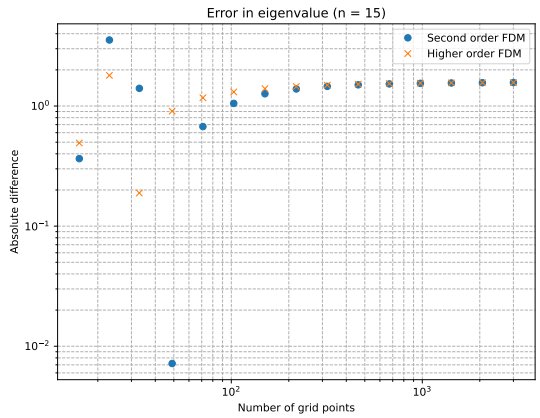

Figure 6: *Eigenvalue error depending on the number of grid points ($n = 15$)*

Flexibility

The Python code realized to compute the eigenfunctions and eigenvalues can be considered flexible as it allows to change the input parameters and the FDM method used in an easy way. Indeed the function `quantum_harmonic_oscillator(N, a, b, omega, higher_order = False)` leaves to the user the freedom to change all the numerical parameters used for the approximation of the solutions.

Efficiency

To evaluate the efficiency of the implemented code, the runtimes of the execution of the function `quantum_harmonic_oscillator(N, a, b, omega, higher_order = False)` have been recorded and represented on the figure 7, for different numbers of grid points.

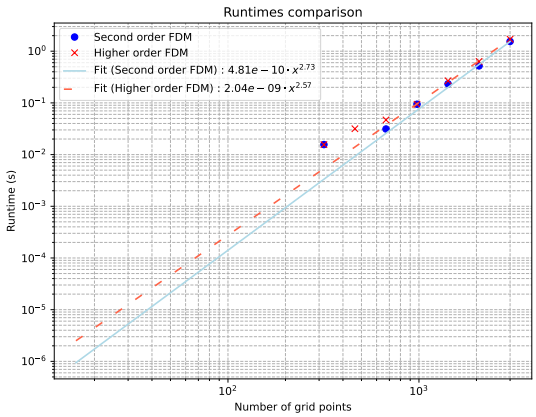

Figure 7: *Runtimes for different numbers of grid points*

The runtimes follow a polynomial scaling according to the number of grid points. Nevertheless, these times are still relatively low and allow to use a large number of grid points while still being efficient.

Conclusion

The implementation of the Finite Difference Method to solve the Schrödinger equation for the quantum harmonic oscillator successfully yields numerical solutions for the eigenvalues and eigenfunctions. By discretizing the system and applying higher-order approximations for the second derivative, we significantly improve the accuracy of the results. The comparison with analytical solutions demonstrates the validity of the approach. Additionally, the stability and efficiency of the method were evaluated, showing that higher-order methods can be computationally intensive but offer enhanced precision. This approach can be extended to more complex quantum systems, providing a versatile tool for computational quantum mechanics.