QUANTUM INFORMATION AND COMPUTING

Assignment 4

Author
Marco Chiloiro
November 28, 2023

Continuous time-independent Schrödinger Equation

In this report, we explore the numerical determination of eigenstates and eigenvalues for a quantum harmonic oscillator. Additionally, we assess the correctness, numerical stability, accurate discretization, flexibility, and efficiency of the implemented scientific program. These five key points serve as essential criteria for evaluating the reliability and performance of our computational approach in the context of quantum simulations.

Theoretical background

The quantum harmonic oscillator is a fundamental concept in quantum mechanics that describes a particle subjected to a harmonic potential. The one-dimensional time-independent Hamiltonian, which represents the total energy of the system, is given by:

$$\hat{H} = -\frac{1}{2}\frac{d^2}{dx^2} + \frac{1}{2}\omega^2 x^2, \tag{1}$$

where x is the position of the particle, ω is the angular frequency of the oscillator and where we assumed the mass of the particle and the reduced Planck constant equal to 1, i.e. $\hbar = m = 1$. The corresponding continuous time-independent Schrödinger equation is

$$-\frac{1}{2}\frac{d^2\psi(x)}{dx^2} + \frac{1}{2}\omega^2 x^2 \psi(x) = E\psi(x), \tag{2}$$

whose analytical solutions lead to quantized energy levels and corresponding eigenfunctions. The energy eigenvalues are given by:

$$E_n = \frac{\omega}{2}(2n+1), \tag{3}$$

where n is a non-negative integer. The associated eigenfunctions $\psi_n(x)$ are Hermite polynomials multiplied by a Gaussian function.

The *finite difference method* is a numerical technique employed to solve differential equations by approximating derivatives with finite differences. When applied to the quantum harmonic oscillator eigenvalue problem, the Schrödinger equation is discretized. Therefore, we replace the second derivative in (2) with a finite difference approximation. Let N be the number of discrete points in the spatial domain, and h be the grid spacing. The second derivative is approximated as:

$$\frac{d^2\psi(x)}{dx^2} = \frac{\psi(x+h) - 2\psi(x) + \psi(x-h)}{h^2} + O(h^2). \tag{4}$$

Substituting (4) into (2), we get a discretized form of the latter. This leads to a system of algebraic equations for the discrete values of $\psi(x_i)$ at each grid point x_i, where $i = 1, .., N$. By collecting all the discrete values of $\psi(x_i)$ into a N-dimensional vector $\vec{\psi}$, the discretized equation becomes:

$$(K+V)\vec{\psi} = E\vec{\psi}, \tag{5}$$

where K is a tridiagonal matrix representing the finite difference approximation of the second derivative and V a diagonal matrix representing the potential term. The eigenvalues E_n and corresponding eigenvectors $\vec{\psi}_n$ can be determined by solving the above eigenvalue problem.

The accuracy of the finite difference method depends on the choice of grid spacing and the total number of grid points. Moreover, it is possible to further improve this method by considering higher orders for the second order derivative approximation.

Implementation

In this section, we describe the key steps taken in implementing the quantum harmonic oscillator problem, covering both analytical and numerical aspects.

The analytical solutions for the quantum harmonic oscillator were implemented as Python functions. These functions calculate the theoretical energy levels E_n and eigenfunctions $\psi_n(x)$ based on the quantum number n and oscillator frequency ω (since we are assuming $m = \hbar = 1$). For simplicity, we set $\omega = 1$.
Python functions were developed to generate the discretized potential and kinetic operators. Given the chosen order of approximation (second or fourth), the summation of these matrices forms the discretized Hamiltonian.
The `scipy.linalg.eigh_tridiagonal` Python function was employed to solve the eigenvalue problem associated with the discretized Hamiltonian, when using the second-order approximation. In the case of the fourth-order approximation for the kinetic operator, the more general, but less efficient, `scipy.linalg.eigh` function was employed. This approach provided numerical approximations for energy levels and corresponding eigenfunctions.
Absolute errors (AE) between the numerical and analytical energy solutions were computed. A predefined threshold of 0.01 was established to determine the maximum energy level that the numerical solution could accurately estimate. This maximum energy level, hereafter referred to as the 'critical energy level', represents the point beyond which the accuracy of the numerical estimation may be compromised.

Results

We begin by addressing the correctness of the program, focusing on the results obtained through the comparison between analytical and numerical solutions. Figures 1 and 2 illustrate that, due to the discretization of the Hamiltonian and the finite size of the considered spatial interval, the numerical solutions accurately estimate only the initial energy levels. Beyond a certain point, divergence becomes evident between the numerical and analytical solutions. This discrepancy highlights the limitations imposed by the chosen discretization and spatial constraints, emphasizing the importance of considering these factors. As an illustration of this behavior, Figure 3 depicts the amplitudes of the eigenfunctions corresponding to n immediately above and below the critical energy level. Notably, for n values below the critical threshold, the amplitudes overlap almost perfectly. However, even immediately above the critical value, differences in amplitudes become apparent.

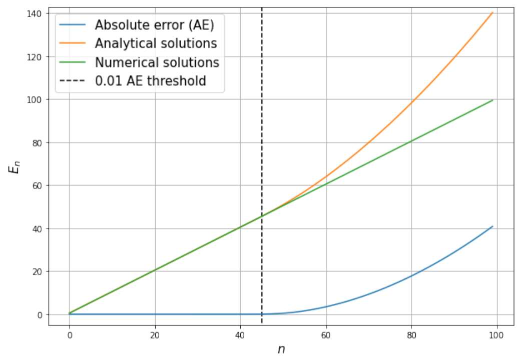

Figure 1: Analytical and numerical energy levels, along with their absolute differences. The black dotted line represents the highest energy level that adheres to the imposed threshold of 0.01 on the absolute error (AE), here it is $n = 46$. The spatial interval considered is $[-10, 10]$, with spatial steps of $h = 0.01$. Second-order approximation was considered.

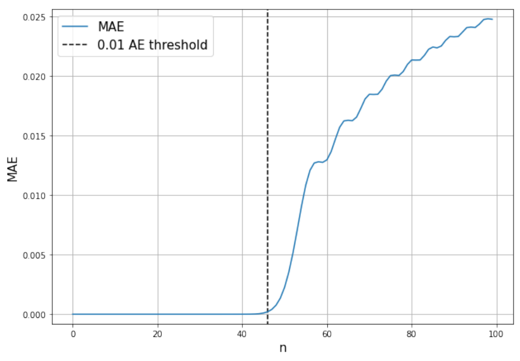

Figure 2: Mean Absolute Error (MAE) between the analytical and numerical amplitudes of the eigenfunction solutions, specifically associated with the energy levels depicted in Figure 1. The black dotted line corresponds to the same energy level showcased in Figure 1. Second-order approximation was considered.

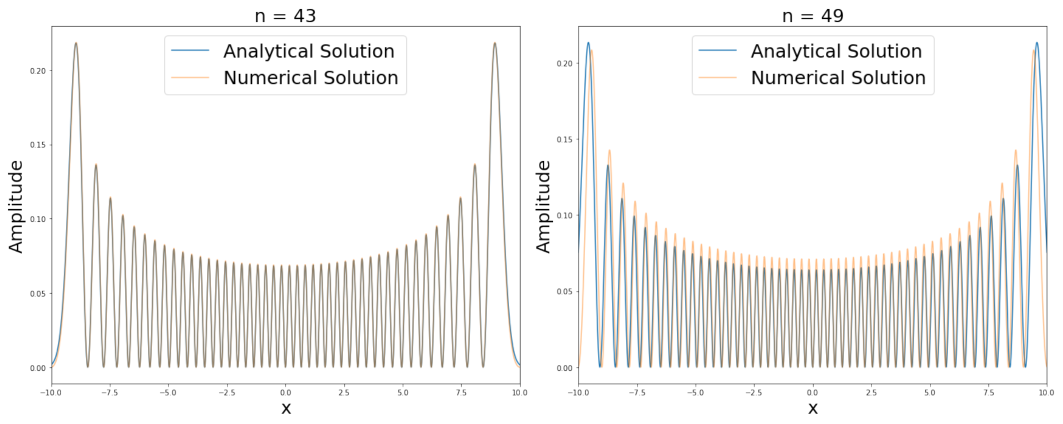

Figure 3: Comparison between the amplitude of the analytical eigenfunctions and the numerical one. On the left for $n = 43$ and on the right for $n = 49$, where both the critical energy level ($n = 46$) and the discretization considered are the same as described in Figure 1.

Continuing with the analysis of the accurate discretization, as depicted in Figure 4, we observe that, with a fixed length for the discretization interval, the variation in the step size of discretization influences the maximum energy level that can be numerically estimated with sufficient accuracy. Additionally, we notice that once the step size becomes sufficiently small, there's a point where the highest achievable energy level starts to decrease. This change is influenced by the length of the discretization interval, determining the energy levels the numerical simulation can accurately reach. Figure 5 demonstrates that by extending the length of the discretization interval, we're able to reach higher energy levels, particularly when using smaller step sizes than in the previous case.

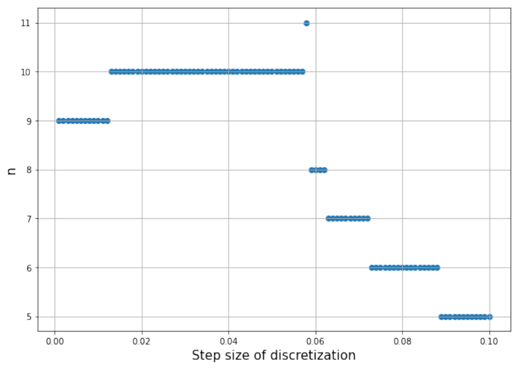

Figure 4: The quantum number n corresponding to the critical energy level, identified with a 0.01 threshold on the absolute error (AE), is depicted in relation to the step size of discretization. The analysis considers a space interval of $[-5, 5]$ and employs a second-order approximation.

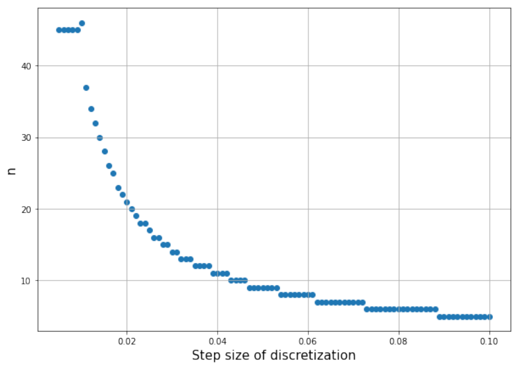

Figure 5: The quantum number n corresponding to the critical energy level, identified with a 0.01 threshold on the absolute error (AE), is depicted in relation to the step size of discretization. The analysis considers a space interval of $[-10, 10]$ and employs a second-order approximation.

Discussing program flexibility, the code is easy to extend. We've separated the functions for the potential and kinetic parts, so one can change the potential to create different time-independent Hamiltonians. In the potential function, one can adjust parameters ω and m. Moreover, if one want to find the first energy levels with higher accuracy, the kinetic matrix function includes a fourth-order approximation, and it is possible to change the error threshold in the function that locates the critical energy level. Importantly, all these functions are organized in a separate module from the main program. This means that it is possible to use them individually in future projects or extend their functionality without affecting the main program or other functions.

Now, regarding the code's efficiency, Figure 6 illustrates the fit of execution time for numerical simulations as the number of discretization steps N varies, corresponding to the size of the Hamiltonian to be diagonalized. From the fit, we see that the execution time scales as $O(N^{1.34})$, which is a good outcome. However, it's essential to note that this result is true only when considering a second-order approximation in the discretization of the kinetic energy operator, resulting in a tridiagonal Hamiltonian that is simpler to diagonalize. When higher orders of approximation are considered, such as the fourth order, the execution time scales more like $O(N^3)$. Therefore, if reliable approximations at high energy levels are sought, one must contemplate larger spatial intervals and smaller discretization steps, along with the incorporation of approximations beyond the second order. This inevitably leads to longer execution times.

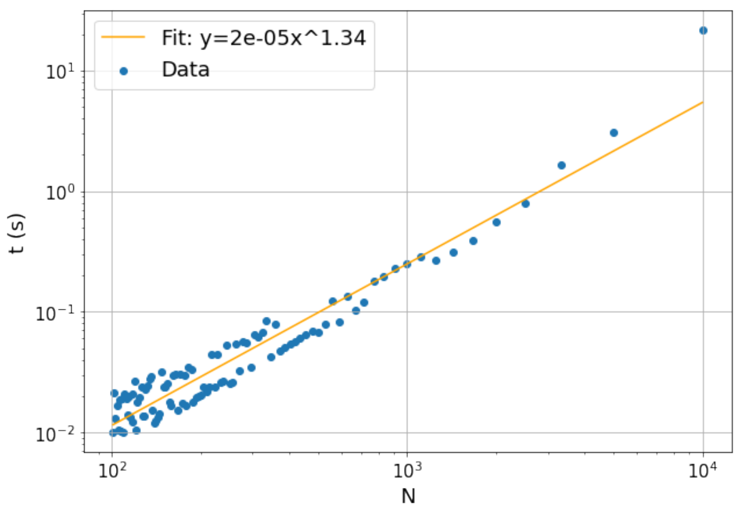

Figure 6: Power-law fit depicting the relationship between the execution time of the simulation and the number of discretization steps N.