Assignment 4 - Quantum harmonic oscillator.

Calandra Buonaura Lorenzo
(Dated: November 22, 2024)

This assignment investigates the quantum harmonic oscillator, a fundamental model in quantum mechanics, through numerical approaches. The focus lies on solving the time-independent Schrödinger equation for the quantum harmonic oscillator using the finite difference method; by discretizing the spatial domain and approximating the differential operators, we compute the eigenvalues and eigenfunctions of the Hamiltonian in the first quantization framework. This study explores the influence of grid size, discretization step, and finite difference order on the accuracy and convergence of the numerical solutions. Additionally, we analyze the statistical properties of the computed eigenvalues and eigenfunctions, emphasizing the stability of the approach over different runs and the agreement with analytical results. Finally we show the efficiency of the approach, studying the compilation time when reducing the size of the grid step.

1. INTRODUCTION

The quantum harmonic oscillator is one of the simplest quantum system, where a single particle is not free but it is subject to a simple harmonic potential. This system can be analytically solved (both for energy levels and eigenfunctions) without approximations: this property makes it a good toy system to study the correct implementation of a numerical procedure used for solving the Schrödinger equation.

The method used in this Assignment for solving the time-independent Schrödinger equation numerically is called the "finite difference method", which approximates differential operators by discretizing the spatial domain into a finite grid. This approach transforms the continuous eigenvalue problem into a matrix eigenvalue problem, making it computationally tractable; depending on the degree of the approximation, we can reach a different accuracy when compared with the analytical results.

In order to see whether the method is efficiently and correctly solving the equation, we need to modify the parameters of the simulation: by varying the grid size, discretization step, and the finite difference order, we examine the stability and accuracy of the computed eigenvalues and eigenfunctions. Through these analyses, we aim to provide a comprehensive understanding of the quantum harmonic oscillator and the practical application of computational methods in quantum mechanics.

2. THEORETICAL FRAMEWORK

A. Quantum harmonic oscillator (QHO)

The quantum harmonic oscillator describes a particle bound by a quadratic potential. Its Hamiltonian is given by:

$$\hat{H} = -\frac{\hbar^2}{2m}\frac{d^2}{dx^2} + \frac{1}{2}m\omega^2 x^2, \tag{2.1}$$

where m is the mass of the particle, ω is the angular frequency, and x is the position [1].

Solving the time-independent Schrödinger equation, $\hat{H}\psi(x) = E\psi(x)$, yields the eigenvalues and eigenfunctions. The eigenvalues are discrete and given by:

$$E_n = \hbar\omega\left(n + \frac{1}{2}\right), \quad n = 0, 1, 2, \dots \tag{2.2}$$

The eigenfunctions, $\psi_n(x)$, are expressed in terms of Hermite polynomials $H_n(x)$:

$$\psi_n(x) = \sqrt{\frac{1}{2^n n!}}\left(\frac{m\omega}{\pi\hbar}\right)^{\frac{1}{4}} e^{-\frac{m\omega x^2}{2\hbar}} H_n\left(\sqrt{\frac{m\omega}{\hbar}}x\right) \tag{2.3}$$

The quantum harmonic oscillator, as the classical counterparts, is as a crucial model due to its exact solvability and, in particular, its relevance in approximating physical systems near equilibrium.

B. Finite difference method (FDM)

The finite difference method is a numerical technique for solving differential equations by approximating derivatives using finite differences. To solve the time-independent Schrödinger equation, the spatial domain $[a, b]$ is discretized into a grid of N points, and the second derivative in the kinetic energy term is approximated using a central difference scheme [2]. The grid spacing is defined as:

$$\Delta x = \frac{b-a}{N} \tag{2.4}$$

For a second-order scheme, the second derivative of the wavefunction $\psi(x)$ at a grid point x_i is approximated as:

$$\frac{d^2\psi}{dx^2} \approx \frac{\psi(x_{i+1}) - 2\psi(x_i) + \psi(x_{i-1})}{\Delta x^2} \tag{2.5}$$

Higher-order finite difference schemes can improve accuracy by incorporating more grid points into the approximation, which are gathered in Table I.

Order	−4	−3	−2	−1	0	1	2	3	4
2				1	−2	1			
4			−1/12	4/3	−5/2	4/3	−1/12		
6		1/90	−3/20	3/2	−49/18	3/2	−3/20	1/90	
8	−1/560	8/315	−1/5	8/5	−205/72	8/5	−1/5	8/315	−1/560

TABLE I: Coefficients for FDM at different orders. The table shows the increasing range of non-zero coefficients for higher approximation orders, leading to more accurate yet computationally demanding discretizations.

The continuous Schrödinger equation is thus transformed into a matrix eigenvalue problem:

$$\mathbf{H}\boldsymbol{\Psi} = E\boldsymbol{\Psi}, \tag{2.6}$$

where $\mathbf{H}$ is the discretized Hamiltonian matrix, $\boldsymbol{\Psi}$ is the vector of wavefunction values at grid points, and E represents the eigenvalues. The Hamiltonian matrix $\mathbf{H}$ has the following tridiagonal structure (at order 2, otherwise other diagonals appear):

$$\mathbf{H} = \begin{bmatrix} a_1 & b & 0 & \cdots & 0 \\ b & a_2 & b & \cdots & 0 \\ 0 & b & a_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & b \\ 0 & 0 & 0 & b & a_n \end{bmatrix}, \tag{2.7}$$

where:

$$a_i = \frac{1}{\Delta x^2} + \frac{\omega^2 x_i^2}{2}, \tag{2.8}$$

$$b = -\frac{1}{2\Delta x^2}. \tag{2.9}$$

using atomic units ($m = \hbar = 1$) [2]. The eigenvalues and eigenvectors of $\mathbf{H}$ correspond to the energies and wavefunctions of the quantum system, respectively. The method's accuracy depends on the grid size N, the domain length L, and the finite difference order, which are taken into consideration in the analysis.

C. Evaluation parameters

Developing reliable scientific software requires following some ideal principles that ensure correctness, stability, accuracy, flexibility, and efficiency [3]:

- **Correctness:** Software correctness ensures that the implementation matches the theoretical model. This requires extensive validation against analytical solutions, which are known for the QHO.
- **Stability:** Numerical stability is crucial for obtaining meaningful results, particularly when solving differential equations. Algorithms must handle round-off errors and avoid numerical instabilities, without showing distant results for consecutive runs.
- **Accurate discretization:** Discretization parameters, such as the grid size and step size, should be chosen to balance accuracy and computational cost, considering that higher-order methods may improve precision but can be more resource-intensive. Thus, is necessary to choose the correct grid size that makes the computation efficient and correct at the same time.
- **Flexibility:** Code should be modular and adaptable, allowing changes to parameters or methods without extensive rewrites, facilitating, for example, exploration of different numerical schemes.
- **Efficiency:** Computational efficiency is critical for handling large-scale problems; leveraging optimized libraries, parallelization, and efficient algorithms can significantly reduce computation time.

These principles are exhaustively discussed for our implementation of the FDM for the quantum harmonic oscillator.

3. CODE DEVELOPMENT

All the functions needed for the analysis and the plotting have been collected into a python script, `qho_functions.py`, which is then imported in a jupyter notebook for better visualization.

A. FDM and analytical solution

The first functions implemented are those for the actual FDM solution and the analytical solution, along with their visualization:

- `kinetic_matrix`: Computes the kinetic energy matrix for the finite difference method, with a specified order of approximation (2, 4, 6, or 8), using the coefficient that are presented in Table I.
- `hamiltonian`: Constructs the Hamiltonian matrix for the harmonic oscillator using the finite difference numerical method (Equation (2.7)).
- `harmonic_oscillator_spectrum`: Computes the eigenvalues and eigenfunctions of the harmonic oscillator using the finite difference method.

- `hermite`: Computes the Hermite polynomial of a given order over a real space grid.
- `harmonic_en`: Calculates the energy levels for a harmonic potential given the angular frequency and the energy level index.
- `harmonic_wfc`: Computes the wavefunction of a given order for a harmonic potential, defined over a real space grid.
- `generate_colors`: Generates a colormap with a specified number of distinct colors for plotting.
- `plot_wf_en`: Plots wavefunctions and energy levels of the quantum harmonic oscillator, with the option to use either numerical or analytical results.

B. Program evaluation

The other functions present in the file are used for the program evaluation, taking into consideration the principles explained before in Section 2 C.

Correctness

- `check_schroedinger`: Verifies how well the computed wavefunctions and energies satisfy the Schrödinger equation for the harmonic oscillator.
- `energy_difference`: Computes the absolute or relative energy differences between analytical and computed eigenvalues for the harmonic oscillator, handling different finite difference orders.
- `wfc_difference`: Computes the differences between analytical and computed eigenstates for the harmonic oscillator using the dot product. The function ensures that the number of eigenstates is sufficient and returns the 1 - absolute dot product as the difference.
- `plot_schroedinger`: Generates a heatmap of the differences in how well the computed wavefunctions and energies satisfy the Schrödinger equation, leveraging the `check_schroedinger` function.
- `plot_energy_orders`: Generates a heatmap of energy differences between analytical and computed eigenvalues for different orders, leveraging the `energy_difference` function.
- `plot_wfc_orders`: Generates a heatmap of differences between analytical and computed eigenstates for different orders of finite differences, leveraging the `wfc_difference` function.
- `plot_loglog_fit`: Performs a log-log regression on the energy differences for a specific order and plots the data along with the regression fit.

Stability

- `check_stability`: Checks the stability of the finite difference solution by computing mean and standard deviation of eigenvalues, mean deviation of eigenvector dot products, and a dot matrix for deviations across runs.
- `plot_stability`: Visualizes stability metrics (mean and standard deviation of eigenvalues, eigenvector dot products) for various finite difference orders using heatmaps.

Accurate discretization

- `check_discretization`: Computes discretization errors in the eigenvalues and eigenfunctions for various grid sizes and finite difference orders.
- `plot_discretization_heatmaps`: Visualizes discretization errors as heatmaps, showing relative errors of eigenvalues and eigenfunctions for different grid sizes and finite difference orders.

Flexibility

No specific functions have been implemented for the flexibility analysis; nevertheless, all the functions have adjustable parameters (like choosing the approximation order), which makes them flexible and adaptable to different studies. One thing that could have been improved is a function capable to cope with a non-quadratic potential, but, due to lack of time, it has not been done.

Efficiency

- `measure_efficiency`: Measures computational times for solving the quantum harmonic oscillator problem across various grid resolutions and finite difference orders, optionally averaged over multiple repetitions.
- `plot_efficiency_heatmap`: Visualizes computation times as a heatmap, with finite difference orders and grid resolutions represented as axes.

C. Code availability

All the data and the code for the data analysis can be found in this public Github repository: https://github.com/Kallo27/QIC/tree/main/Assignment4

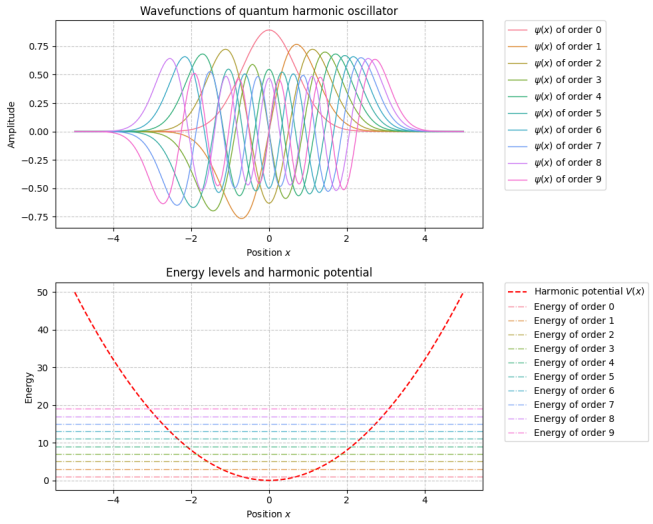

FIG. 1: (Top) First 10 wavefunctions for the quantum harmonic oscillator in the domain [-5, 5] with $\omega = 2$. (Bottom) First 10 energy levels and harmonic potential. The approximation order is 2 and $dx = 0.01$.

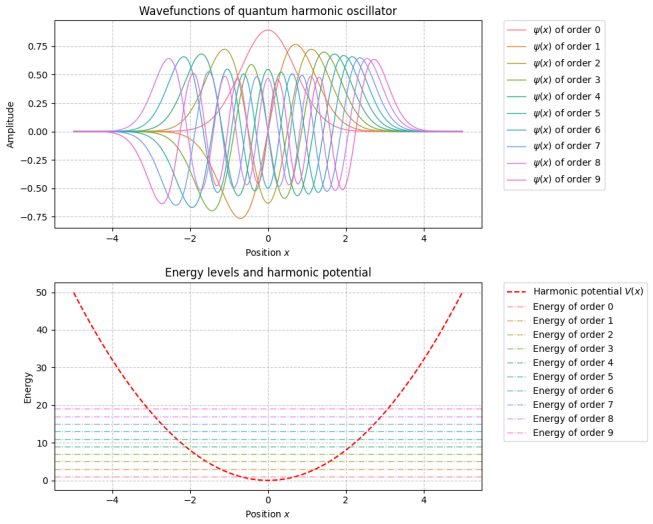

FIG. 3: (Top) First 10 wavefunctions for the quantum harmonic oscillator in the domain [-5, 5] with $\omega = 2$. (Bottom) First 10 energy levels and harmonic potential. Analytical solutions.

4. RESULTS

The first step of the analysis consists in building the finite difference approximation and plotting energy levels and wavefunctions. Figure 1 shows the first ten wavefunctions and energy levels for the second order approximation for the quantum harmonic oscillator. As we can see, the energy levels appear to be pretty equally spaced and with the correct values; at the same time the wavefunctions are similar to the ones expected (see Figure 3). It's obvious that these observations are purely descriptive, not analytical: in order to prove that the FDM implementation is working well we need to proceed with some calculations. Thus, in the following subsections, we'll analyze the principles presented in Section 2 C. Most of the simulations have the same parameters ($N = 1000$, $L = 5$, $\omega = 2$) and it's specified when something changes.

A. Correctness

The first thing to do is to see if the obtained results are consistent, meaning that the eigenvalue-eigenstate problem has been solved correctly. In order to do so, we can compare the expected value of the energy, given by $\psi H \psi$ (where ψ is an eigenstate), with the obtained eigenvalue. Figure 2 shows the difference between obtained and expected energy for the first fifty eigenstates. We can easily conclude that the computation is right, as all these differences are of the order of magnitude of 10^{-12} or smaller. We can also notice that in general, for higher order of approximation, we have a smaller error, as expected.

After this, we can proceed with the comparison with the analytical results. For the energy levels we study the relative difference, while for the wavefunctions we compute the hermitian product with corresponding eigen-

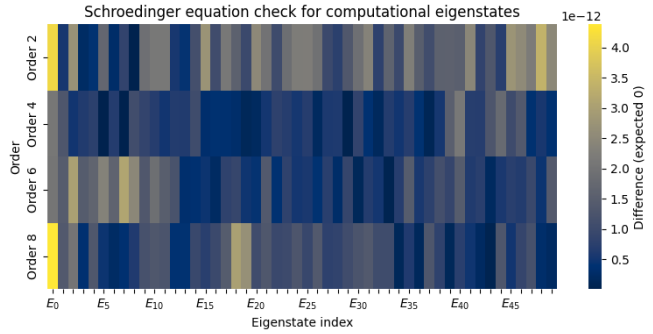

FIG. 2: Difference between expected and obtained energies for the the first 50 eigenvalues.

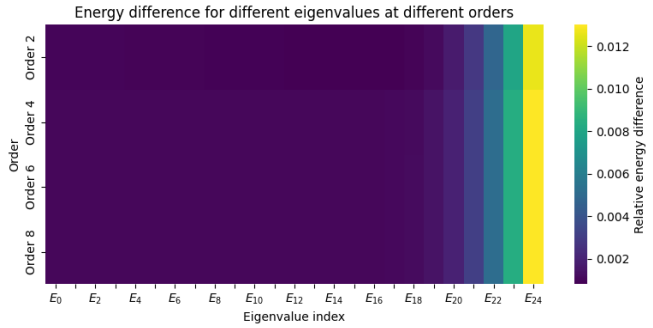

FIG. 4: Relative difference between expected and obtained energy levels.

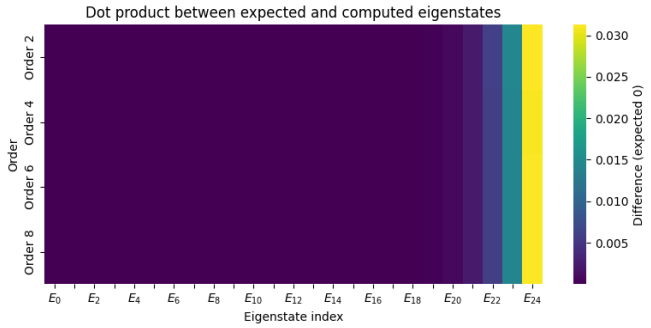

FIG. 5: Difference between 1 and hermitian product between expected and obtained wavefunctions.

states: the expected value is obviously 1, as the wavefunctions are a set of orthonormal vectors, thus we consider the difference between 1 and this product. Figure 4 and Figure 5 depict these analysis for the first twentyfive energy levels and eigenstates. As we can see for the energy levels we are always under the 1.2% error, which highlights the agreement with the analytical solution. Regarding the eiegnstates, we have also in this case a good agreement, with a maximum difference of 0.03 from 1. In both cases we can easily see that the error increases with the index taken into consideration, which means that we have a better approximation for low energy states (near the bottom of the potential).

In order to see how the absolute energy difference scales with the eigenvalue index we can perform a fit with a known distribution; in particular, it turns out to be polinomial, so we can perform a linear regression in log-log space (fixing the highest order of approximation, 8). It turn out that the absolute energy difference scales as $k^{2.44}$, where k is the eigenvalue index. From Figure 6

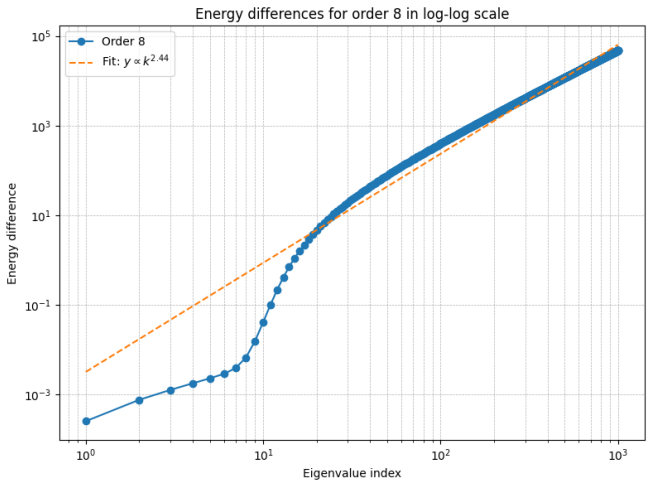

FIG. 6: Linear regression in log-log space for absolute energy difference

we can notice that for the first 20 eigenvalues the behaviour is different: this deviates from the trend of the other energy levels and it' probably due to their smaller absolute magnitude, which makes them less oscillatory and better approximating the analytical result.

B. Stability

After proving the correctness of the approximation, it's necessary to verify whether the implementation is stable over different runs or not. For this, we can analyze the mean standard deviation of the eigenvalues over a high number of runs, in order to see if there is a broadening of the distribution. Regarding the eigenstates we can proceed similarly, studying the value of the hermitian product between corresponding ones and the mean deviation over the number of runs. From this analysis, it turns out that the eigenvalue mean and the hermitian product of

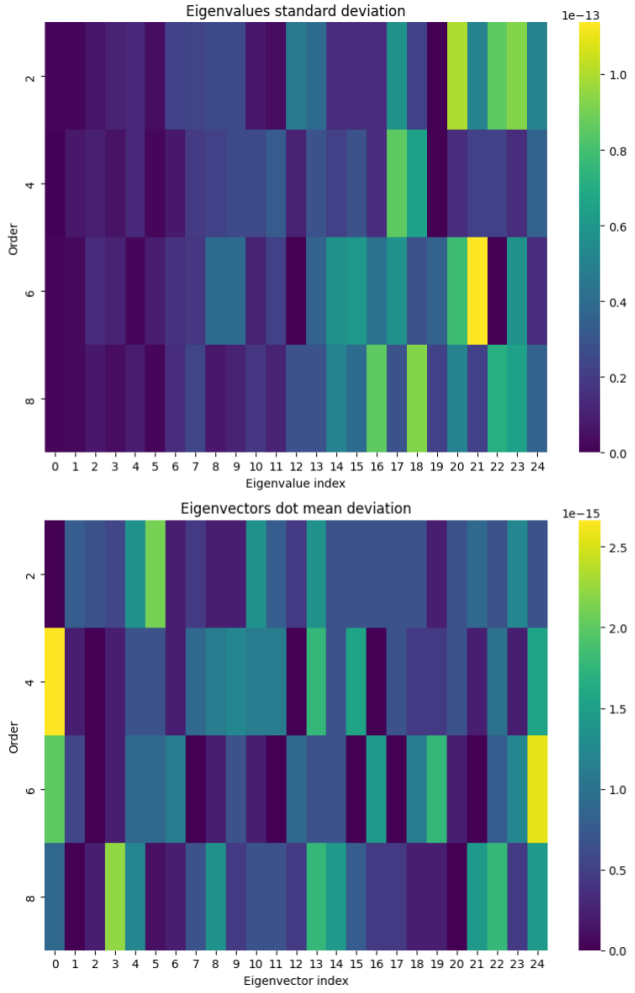

FIG. 7: (Top) Eigenvalues standard deviation. (Bottom) Eigenvectors hermitian product deviation. Both heatmaps are realized with `n_runs = 100`.

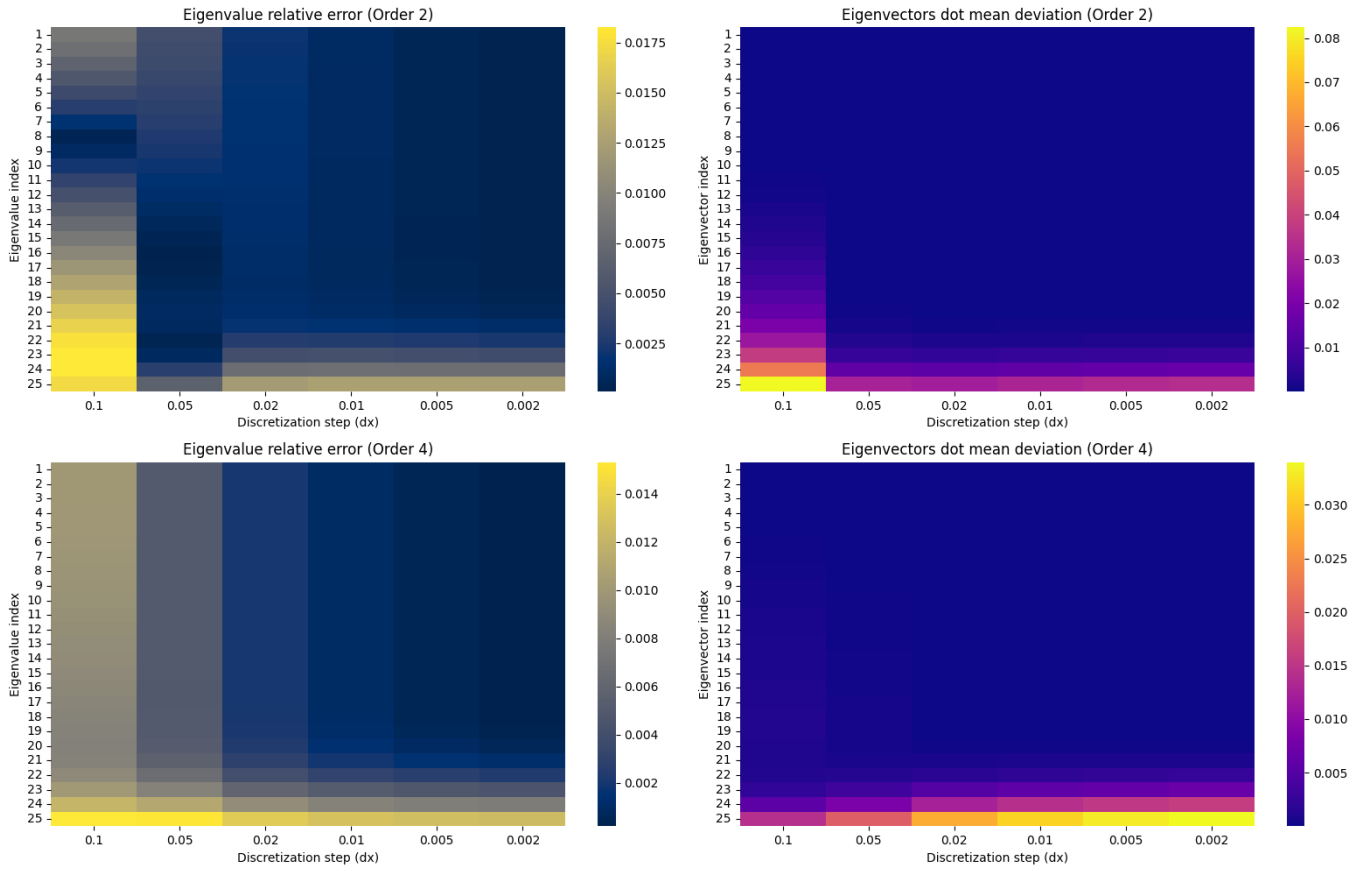

FIG. 8: Analysis of the relative energy difference (left) and the hermitian product (right) between corresponding eigenstates as a function of the grid spacing dx, for second and fourth order finite difference approximations.

corresponding eigenstates are stable over an hundred of runs, which highlights the stability of the FDM implementation. Figure 7 presents the two the summarizing heatmaps for the eigenvalues and eigenstates hermitian products deviation (the other heatmaps are not present so as not to overload the report with unnecessary images). As it's clear from them, we don't observe any significative deviation, we can simply notice that the eigenvalues distribution seems to be broadening as the energy level index increases, in agreement with what observed before.

C. Accurate discretization

It is now time to study how the grid size influences the accuracy of the FDM implementation: due to the fact that we want to solve a differential equation discretizing the spatial domain, we can suspect that a finer grid will be more accurate (even if more computationally demanding). Thus, the analysis consider how the relative energy difference and the hermitian product between corresponding eigenstates change with the discretization step. Figure 8 represent the results for the first two order of approximation (for order 6 and 8 the result are similar to order 4): it clearly stands out that a finer grid, up to $dx = 0.002$, has in general a better agreement with the expected results than a larger one, as expected. We can also notice that an higher order of approximation lowers the general error of some quantity (0.35%), but in general the behaviour is the same; also in this case we see that for higher eigenvalue indexes we lose some accuracy. For the eigenvector hermitian product we have the same result for the second order approximation, while for the fourth order we observe something different: the error is indeed higher for a finer grid for the higher eigenstates. This can be due to the fact that the higher eigenstates are more sensitive to numerical errors introduced by truncation and discretization, even when using a finer grid.

D. Flexibility

For the flexibility analysis, as alredy mentioned previously, no specific functions have been implemented. The FDM approximation has been implemented up to the eighth order of approximation, as can be evinced from the previous pictures, demonstrating the inherent flexi-

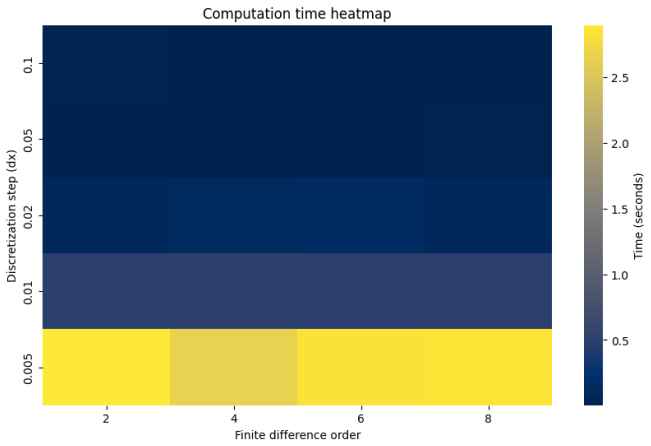

FIG. 9: Computation time as a function of grid size for different orders of approximation.

bility of the finite difference method (FDM), as it allows for straightforward extensions to even higher orders of approximation if required. The modular design of the implementation ensures that modifying the order of the approximation or adapting the grid parameters can be achieved with minimal effort. Future work could include implementing automated routines for optimizing the grid size and order of approximation based on a desired level of accuracy, or extending the method to support time-dependent problems, such as solving the time-dependent Schrödinger equation.

E. Efficiency

Finally, regarding efficiency, we studied the computation time for different grid sizes (see Figure9), benchmarking the `harmonic_oscillator_spectrum` function. We can easily see that for all the orders of approximation the behaviour is the same, with a very low computational time for large grid size and what seems to be a polinomial scaling for finer grids, in agreement with what expected. These results were obtained averaging over 20 runs every combination of parameters; due to the lack of computational time and resources, no further analysis was conducted to investigate the scaling behavior in more detail.

It is also worth noting that while higher-order approximations tend to introduce a larger prefactor in the computational cost due to the increased complexity of the stencil, their efficiency remains comparable for the same grid size. This indicates that the choice of approximation order primarily affects accuracy rather than drastically altering computational performance, at least at this level.

5. CONCLUSION

In this study, we implemented and analyzed a finite difference method (FDM) approach for solving the quantum harmonic oscillator, focusing on its correctness, stability, accuracy, flexibility, and efficiency.

The correctness of the method was verified through comparisons between computed and expected results for eigenvalues and eigenfunctions: relative energy differences and the hermitian product with analytical solutions confirm the consistency of the results, especially for low-energy states, with errors consistently below 1.2% and 0.03, respectively.

Stability analysis further reinforced the reliability of the implementation, showing negligible deviations in eigenvalues and eigenstates over 100 runs.

In terms of accuracy, we observed that a finer grid leads to better agreement with analytical results, as expected, and that higher-order approximations reduce general errors. However, the higher eigenstates are more sensitive to discretization and truncation errors, which manifest as reduced accuracy for these states.

Finally, efficiency analysis showed a polynomial scaling of computational time with grid size, consistent with theoretical expectations. Higher-order approximations introduce some additional computational cost due to the complexity of the stencil but maintain comparable efficiency for the same grid size. This trade-off between accuracy and efficiency should guide practical applications, depending on the available computational resources and desired precision.

In conclusion, the finite difference method provides a robust, flexible, and efficient framework for solving the quantum harmonic oscillator.

[1] Griffiths, D. J., & Schroeter, D. F., *Introduction to Quantum Mechanics*. Cambridge University Press, 2018.
[2] Kahaner, D., Moler, C., & Nash, S., *Numerical Methods and Software*. Prentice Hall, 1989.
[3] Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P., *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, 2007.