JOURNAL OF
EMERGING INVESTIGATORS

Article

Solving the Schrödinger equation computationally using the Lanczos algorithm

William Longtin[1], Kristen Surrao[2]
[1]Hollis Brookline High School, Hollis, New Hampshire
[2]Department of Physics, Columbia University, New York City, New York

SUMMARY

The Schrödinger equation is a fundamental equation in quantum mechanics that describes the behavior of particles in terms of wave functions. These wave functions provide a probabilistic description of where the particle is likely to be found, making them a crucial tool for understanding how particles interact with their environment. Methods for solving the Schrödinger equation analytically tend to be very mathematically complex, leading us to look for other ways to solve the Schrödinger equation. Unfortunately, in most cases, the Schrödinger equation cannot be solved for an exact solution. There are methods, however, for approximate solutions. In this paper, we investigated the Lanczos algorithm, a computational method that can solve the 2D Schrödinger equation in cases where finding an analytical solution would not be feasible. We discretized the Schrödinger equation, then ran a Python program using the Lanczos algorithm to solve for the ground state wave function. We hypothesized that the Schrödinger equation could be solved computationally for the ground state wave function using the Lanczos algorithm. We found that this method efficiently solved the Schrödinger equation for complicated 2D potentials. Additionally, we verified the accuracy of the method by comparing the results with wave functions from problems with known, exact solutions.

INTRODUCTION

Science is essential to comprehending life, technology, and our universe, but to understand the workings of everything around us, we must delve into the behavior of microscopic particles. Quantum mechanics is a branch of physics that deals with these particles and their interactions on the atomic and subatomic scale. On these scales, particles behave in ways that differ significantly from objects in everyday life. In classical mechanics, objects can only occupy one place or state, which can be measured. However, small particles, such as atoms and electrons, do not behave according to the rules of classical mechanics. In fact, according to quantum mechanics, these particles can exist probabilistically in multiple (sometimes infinitely many) states at the same time, which contain information about the particle's position, momentum and energy. This property of a quantum mechanical system is called the superposition of states. Specifically, the superposition of states is defined as a linear combination of all the system's possible configurations with specific probabilities. When the object is observed or measured, the wave function, which is a mathematical function that describes the properties of a quantum mechanical system, collapses, reducing the described object to a single physical state. This collapse of the wave function during observation is a fundamental aspect of quantum mechanics and distinguishes it from classical mechanics (1-3).

The Schrödinger equation is the central equation of quantum mechanics that governs the wave function of a particle in a potential (3). In one dimension, the time-independent Schrödinger equation is stated as follows:

$$-\frac{\hbar^2}{2m}\frac{\partial^2\psi(x)}{\partial x^2} + V(x)\psi(x) = E\psi(x) \qquad \text{(A)}$$

where ψ is the wave function, V is the potential, m is the mass of the particle, ħ is the reduced Planck's constant *h*/2π and E is the total energy of the system (kinetic energy + potential energy). The first term relates to the kinetic energy of the system and the second term relates to the potential energy of the system. Equation (A) can also be expressed as follows:

$$\hat{H}\psi(x) = E\psi(x)$$

Where Ĥ is the Hamiltonian operator, which is associated with the total energy of the system and is defined as

$$\hat{H} = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + V(x)$$

The wave function is related to the probability density function ρ by:

$$p(x) = |\psi(x)|^2$$

The probability that the particle can be found on the interval [*a*, *b*] is given by the area underneath the curve between *a* and *b* values:

$$\int_a^b p(x)dx$$

Since the particle must be somewhere in space, the total area under the probability density function must be one, and thus, we get the following normality condition for the wave function:

$$\int_{-\infty}^{\infty} |\psi(x)|^2\, dx = 1$$

Finding exact solutions to the Schrödinger equation can be very challenging in most cases, and it may be impossible to solve the equation analytically depending on the form of the potential (3). However, we can generate approximations that are very close to the true wave function using computational methods. To do this, we must discretize the Schrödinger equation, which turns the continuous functions and operators into their discrete counterparts. We considered the values of the wave function on a lattice, a regular grid of points in space,

each representing a discrete location in space, and used a numerical approximation for the derivative. When represented in a discretized form, the Hamiltonian is a Hermitian matrix, a matrix that is equal to its own conjugate transpose and has real eigenvalues (3). Looking for a method that finds the lowest eigenvalue and corresponding eigenvector of a Hermitian matrix (representing the lowest energy state of the system and the ground state wave function respectively), which appears in the discretized version of Equation (A), we found the Lanczos algorithm, a common method for finding eigenvalues and eigenvectors of Hermitian matrices (4). The Lanczos algorithm gives us a way to solve Equation (A) computationally for the ground state wave function and associated energy E_0, the lowest possible energy state for the system. The Lanczos algorithm is much faster than direct diagonalization, which would be required for finding eigenvalues and eigenvectors of a matrix in general. This is because the Lanczos algorithm has a time complexity of $O(n^2)$, whereas direct diagonalization has a time complexity of $O(n^3)$ (4,5). The Hermitian nature and sparsity of the matrix representation of the Hamiltonian make the Lanczos algorithm an ideal method for finding the ground state wave function, producing fast and accurate results (3-5).

There are other common methods for solving the discretized Schrödinger equation computationally such as the Matrix Numerov method (6). The Numerov method gives approximate solutions to 2nd order differential equations with no first order term, such as the time-independent Schrödinger equation in Equation (A). Like the method presented in this paper, the Numerov method relies on representing the Hamiltonian as a matrix and solving the resulting eigenvalue problem. While the Numerov method has the advantage of being able to provide any energy eigenstate of the time-independent Schrödinger equation and can produce accurate solutions for complicated potentials like the Lanczos method, it is not optimized for finding the ground state wave function. Therefore, it requires more computation to find the ground state wave function than the Lanczos algorithm, which is specifically designed for this purpose (4-6). Additionally, the Numerov method is primarily applied in one dimension and would need to be adapted for use in two-dimensional problems.

The Crank-Nicholson method is another numerical method commonly used to solve the Schrödinger equation computationally, particularly for time-dependent problems (7). Unlike the Lanczos method discussed in this paper, the Crank-Nicholson method is intended for time-dependent problems and relies on a backward and forward Euler method to approximate the values of the wave function at discrete steps in time and space. While it is true that the Crank-Nicholson method can be more computationally expensive than the Lanczos method for time-independent problems, it can give the time evolution of the wave function, which can be useful in many physical applications (4,5,7). It's also worth noting that the Crank-Nicholson method can be particularly effective for simulating quantum systems in two or more dimensions (7). Ultimately, the choice of method will depend on the specific problem and the needs of the researcher.

The Lanczos algorithm is a common iterative method for computing eigenvalues and eigenvectors of a Hermitian matrix, which represents the Hamiltonian. More specifically, the Lanczos algorithm efficiently computes a few of the largest and smallest eigenvalues and corresponding eigenvectors (4,5). The Lanczos algorithm has been successfully applied to solving the Schrödinger equation in various contexts, including time-dependent problems, higher-dimensional systems, and higher energy states (5,8,9). This paper aims to provide an accessible introduction to the Lanczos method for solving the Schrödinger equation. In this paper, we focused on the Schrödinger equation in two dimensions. Therefore, we used an adapted version of Equation (A) for two dimensional potentials.

$$-\frac{\hbar^2}{2m}\left(\frac{\partial^2\psi(x,y)}{\partial x^2}+\frac{\partial^2\psi(x,y)}{\partial y^2}\right)+V(x,y)\psi(x,y)=E\psi(x,y) \qquad \text{(B)}$$

Which can also be represented as

$$\hat{H}\psi(x,y)=E\psi(x,y)$$

Where Ĥ is given by

$$\hat{H}=-\frac{\hbar^2}{2m}\left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}\right)+V(x)$$

Here, the normality condition becomes:

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}|\psi(x,y)|^2\,dydx=1 \qquad \text{(C)}$$

The term "discretized" refers to the numerical approximation of a continuous function or equation by breaking it into discrete intervals or steps. In this paper, we use the discretized version of Equation (B) to solve the Schrödinger equation numerically as derived in the Materials and Methods section:

$$-\frac{\hbar}{2m}\big(\psi(x+1,y)+\psi(x,y+1)-4\psi(x,y)+\psi(x-1,y)+\psi(x,y-1)\big)+V(x,y)\psi(x,y)=E\psi(x,y) \quad \text{(D)}$$

We implemented the Lanczos algorithm using Python3 to numerically solve the two-dimensional discretized Schrödinger equation, given by Equation (D), for the ground state wave function. We observed that running the Lanczos algorithm with more iterations yielded a more accurate approximation of the wave function. Once we obtained the wave function, we normalized it to find the probability function, which we visualized in a color plot. We tested our algorithm for accuracy and efficiency with a series of three potentials $V(x, y)$, including an infinite square well, a quantum harmonic oscillator, and a three-charge Coulomb potential. We tested accuracy by comparing wave function values generated by our algorithm to known, analytical solutions. We tested efficiency by studying the number of iterations needed for convergence and the run-time of the program. Our hypothesis was that we could effectively solve the Schrödinger equation in two dimensions using our method. We found that our model computed wave functions that matched known analytical solutions very accurately, with a difference between the approximate and exact wave functions of less than 10^{-16}, in just a few seconds. Additionally, our method produced qualitatively accurate results for complicated potentials that cannot be solved analytically as it shows a greater probability of the particle being in lower potential regions, which is expected as the ground state wave function is the lowest energy state. Our results demonstrate that this method is effective for finding the ground state wave function for complicated potentials.

RESULTS

We implemented the Lanczos algorithm using Python to numerically solve for the ground state wave function of Equation (D) for a given input potential. The wave function output by the program provided a probability distribution

for the location of an electron in our grid. We simulated an electron by using a particle with the mass and charge of an electron. We first validated the program's output for an infinite square well potential, for which the Schrödinger equation has a known analytical solution (7). Next, we tested the program on two more complicated potentials – the quantum harmonic oscillator and a three-charge Coulomb potential. Using a 128 by 128 grid, we found that the eigenvalues of the wave function converged to the true values within 50-300 iterations, while minimizing unnecessary additional computational. The results demonstrate the efficacy of our method.

Potential 1

The first case we explored was a particle in an infinite square well, also known as the problem of a particle in a box, which has been solved analytically (10). Inside the box, the wave function behaves freely, as the potential is set to zero. The potential outside the box is infinite so the particle has 0 probability of being outside the box (**Figure 1A**). We simulated this by applying the condition that the value of the Hamiltonian was 0 at the boundary of the box. In our model, the side length of our box was L = 127×10^{-9} m, with each grid space representing 10^{-9} m. After implementing the Lanczos algorithm for 300 iterations to find the ground state wave function, which took approximately 3.4 seconds, we found the probability distribution for the electron (**Figure 1B**).

We compared our approximate solution to the exact solution to verify our methodology (**Figure 1C**) (10). The exact ground state wave function that we compared our solution to was (10):

$$\psi(x,y) = \frac{2}{L}\sin\left(\frac{\pi x}{L}\right)\sin\left(\frac{\pi y}{L}\right) \qquad \text{(E)}$$

The wave function obtained from the Lanczos algorithm closely resembles the known analytical solution. Comparing the probability values of each point, we observed a maximum difference of 7.21×10^{-17} between the two functions **(Figure 1D)**. To investigate the effect of grid size on accuracy, we ran the simulation on larger grids. On a 256 by 256 grid, we obtained a higher resolution solution but it required more iterations to achieve the same degree of accuracy. Running 1200 iterations took around 64 seconds and brought the calculated probability plot within 10^{-16} of the true distribution, as in the prior case. On a 512 by 512 grid, the program took around 5000 iterations and 15 minutes to reach the same accuracy. In contrast, on a 64 by 64 grid, only 90 iterations and 0.25 seconds were necessary to obtain the same level of accuracy. These results demonstrate that the computational cost of running the algorithm increases with the square of the grid size. Therefore, when choosing the grid size, one must balance computational cost against the needed accuracy of the solution. Larger grids produce higher resolution solutions but are computationally expensive to simulate.

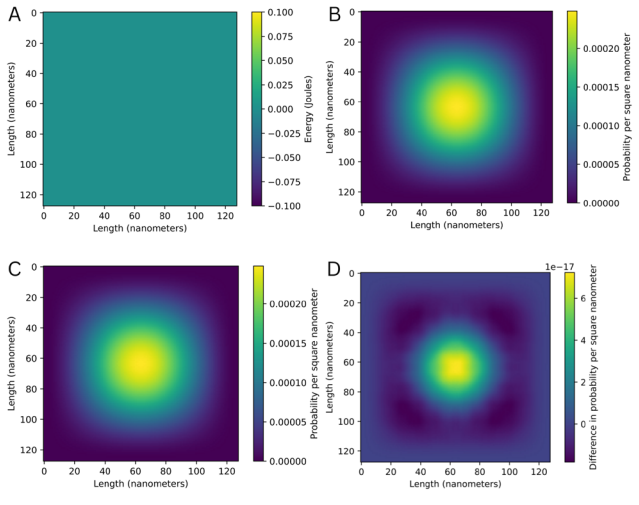

Figure 1: Graphs for Potential 1- particle in a square box. A) Potential plot showing particle confined to a box. **B)** Probability plot calculated using 300 iterations of Lanczos algorithm in 3.4 seconds. **C)** Exact probability plot given by Equation (E). **D)** Difference between calculated probability plot and exact probability plot.

Potential 2

The second potential we created was a quadratic potential well with a minimum at (64,64) (**Figure 2A**). The potential increases with square of the distance from this point, like the quantum harmonic oscillator but in two dimensions. The quantum harmonic oscillator is a case that has been solved for an exact solution where the potential energy is given by the square of the displacement from equilibrium (3). The potential is defined by an equation of the form

$$V(r) = \frac{1}{2}kr^2 \qquad \text{(F)}$$

In this problem, r is the distance from (64,64) in meters and k, the wave factor, is set as 10^{16} J/m^2.

Additionally, we placed a barrier at the edges of the 128 by 128 box by creating a boundary condition that the wave function must be 0 at the boundaries of the box. Here we used a scale of 1 μm per grid space so that the side length of the grid is 128 μm.

We ran the Lanczos algorithm 60 times, taking 0.8 seconds, and generated a probability plot (**Figure 2B).** After 60 iterations, the eigenvalues generated by the Lanczos algorithm had converged to 10 decimal places, indicating that the algorithm had produced a highly accurate approximation of the eigenvalues. This potential is likely impossible to solve analytically as it increases from the center in a radial nature, but the boundary is a square. However, the probability plot is consistent with intuition. Since we solved for the ground state wave function, which is the lowest energy state of the system, we would expect the particle to be localized in regions with lowest potential energy, as demonstrated **(Figure 2B)**. Comparing this probability plot **(Figure 2B)** with the plot from the previous

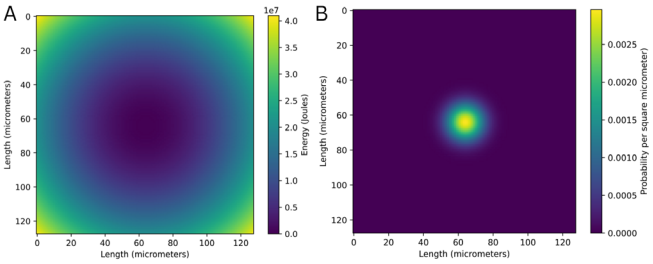

Figure 2: Graphs for Potential 2- 2D quantum harmonic oscillator. A) 2D quantum harmonic oscillator potential given by Equation (F). **B)** Probability plot calculated using 100 iterations of Lanczos algorithm.

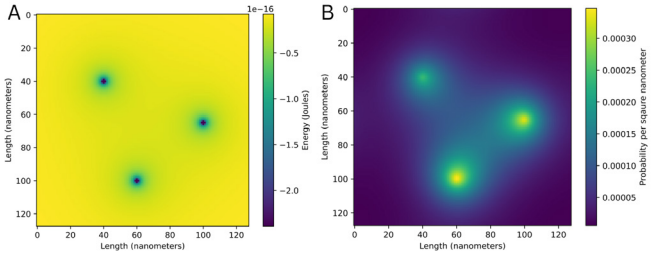

Figure 3: Graphs for the three point charge potential. Each grid space represents one nanometer. A) Three point charge potential given by Equation (G). **B)** Probability plot calculated using 100 iterations of Lanczos algorithm.

problem **(Figure 1C)**, we can see that the probability drops off faster as we move away from the center, which makes sense as this problem has a potential well in the center. Additionally, this graph is similar in form to the probability plot found by Wai Kui Wong for another quadratic potential well using a different computational method for solving the Schrödinger equation, which contains ground state wave function plots for the infinite square well and 2D harmonic oscillator potentials (11).

Potential 3

Finally, we decided to create a more complicated potential to really test the strengths of this method to solve the Schrödinger equation. This potential was an electric potential generated using three protons, particles with a charge of $+1.6\text{x}10^{-19}$ C, placed around the 128 by 128 grid **(Figure 3A)**. The contribution from each of the protons to the potential was

$$V(r) = \frac{k \cdot e^2}{r} \qquad \text{(G)}$$

where $k = 9\text{x}10^9$ N/m^{2}C^2 (Coulomb's constant), $e = 1.6\text{x}10^{-19}$ C, the magnitude of charge of an electron, and r is the distance to the point charge, where each grid space represents 1 nm. The protons were positioned at the following coordinates: (40,40) (65,100) (100,60).

This Schrödinger equation for this potential would be very difficult or impossible to solve analytically as it consists of three separate components that make up the overall potential (3). Our method gave us a fast and qualitatively accurate solution to this problem **(Figure 3B)**. We ran this program for 100 iterations, which took around 1.6 seconds. After 100 iterations, the eigenvalues generated by the Lanczos algorithm had converged to 10 decimal places, indicating that the algorithm had produced a highly accurate approximation of the eigenvalues. Solving for the ground state wave function, the algorithm predicted the electron to most likely be near the three positively charged particles and very unlikely to be near the boundary, as the particle has a negative charge. This is consistent with our understanding of the Schrödinger equation, as the particle is more likely to be in a region of lower potential energy since it is in its lowest energy state.

DISCUSSION

Our program simulated the interactions of an electron with an input potential. It output a colored display of the probability plot, showing where the electron was likely to be. First, we ran the program on a potential that has been solved analytically, the infinite square well, which demonstrated the validity of the algorithm as the probability plot fell within 10^{-16} of the exact solution in 300 iterations (10). Additionally, it produced accurate results in only 3.4 seconds, demonstrating the efficiency of the method. Then, we applied our method to two potentials that would be incredibly difficult or impossible to solve analytically, producing results consistent with those of other computational methods and with our understanding of the ground state wave function of the Schrödinger equation being the lowest energy state (3,11). The results establish the strengths of this method.

In the future, this method could be extended to three dimensional potentials. To do this, the 3D Schrödinger would have to be discretized in a similar fashion. The program could be altered to use a 3D version of the Hamiltonian and a 3D grid. However, this extra dimension would greatly increase the computational cost of this program (8). The results would also have to be visualized differently. Additionally, we could add time dependency (9). This would not affect our results for the probability plots of the ground state wave function. However, it would allow us to find nonstationary states formed by the superposition of eigenfunctions (3).

In most real-world cases, we must deal with a complicated potential where the Schrödinger equation cannot be solved exactly. For example, for larger atoms, it is impossible to find an exact solution to the Schrödinger equation describing the electrons (12). However, implementing this program in three dimensions could give us insights into the behavior of electrons in atoms. Additionally, the Schrödinger equation has applications in the study of semiconductors. Semiconductors often have complicated, piecewise potentials that would be very difficult to solve analytically (13).

Although this method works well in the given examples, it has its limitations. Since we used a discretized version of the Schrödinger equation, this method will not produce an exact solution, no matter how many iterations we use or how large we make the grid. Additionally, if we make the grid larger, the computational complexity of our model increases, and we require more iterations to produce accurate results. The Lanczos algorithm has complexity $O(n^2)$, which is faster than other common algorithms such as exact diagonalization, which scales exponentially (4,5). However, on larger grids of thousands or millions of points per side, this algorithm would take a considerable amount of time to run as each iteration would take longer and we would need more to produce accurate results. Overall, it appears that the computational time increases with the square of the grid size. Even if run on more powerful computers, the computational time would still increase quickly with larger grid sizes.

MATERIALS AND METHODS

To solve the Schrödinger equation, we developed a Python program (accessible through https://github.com/williamlongtin/Lanczos) based on the Lanczos algorithm.

Hamiltonian operator

The Hamiltonian is an operator that gives the total energy of a system (kinetic plus potential). Acting on the wave function, the Hamiltonian, is given by the following:

$$\hat{H}\psi(x,y) = E\psi(x,y)$$

This is the time-independent Schrödinger equation, which has solutions called basis states. If a function $\psi(x, y)$ is a solution to the time independent Schrödinger equation, it is called

an energy eigenstate with eigenvalue E. These eigenstates are the allowed states the system may be in, and the eigenvalue gives the corresponding energy. Solutions to the time independent Schrödinger equation are discrete, so there are only energy eigenstates for certain energy levels. The lowest energy eigenstate is called the ground state wave function (2, 3). In this project, we solved for the ground state wave function of an electron in a potential to make predictions about its location.

Lanczos algorithm

The Lanczos algorithm is an algorithm to find the eigenvalues and eigenvectors of a Hermitian matrix, a matrix that is equal to its conjugate transpose. The Hamiltonian can be discretized and approximated by a Hermitian matrix. We can solve for the eigenvalues and eigenvectors of this matrix using the Lanczos algorithm which proceeds as follows (4).

Given an n x n Hermitian matrix H, we would like to find the lowest eigenvalue and the corresponding eigenvector. We begin with an arbitrary vector $v_0 \in C^n$ with Euclidean norm one Initial iteration:

$$w_0' = Hv_0$$
$$\alpha_0 = w_0'^* v_0$$

(Here, $w'_0{}^*$ is the conjugate transpose of w'_0.)

$$w_0 = w_0' - \alpha_0 v_0$$

Then for j = 1, 2, …, n – 1,

$$\beta_j = |w_{j-1}|$$
$$v_j = w_{j-1}\beta_j$$
$$w_j' = Hv_j$$
$$\alpha_j = w_j'^* v_j$$
$$w_j = w_j' - \alpha_j v_j - \beta_j v_{j-1}$$

Let V be the matrix with columns $v_0, v_1, ..., v_{n-1}$ and T be the tridiagonal matrix with $\alpha_0, \alpha_1, \ldots, \alpha_{n-1}$ on the main diagonal, $\beta_1, \beta_2, \ldots, \beta_{n-1}$ on the lower and upper sub-diagonals, and zeros elsewhere. After we have found the matrix T, we can compute its eigenvectors and eigenvalues. There are many efficient algorithms for this, but we can do it using a Python function (SciPy eigh_tridiagonal). After the least eigenvalue λ and corresponding eigenvector x are computed, we can find the corresponding eigenvector of H as $v = Vx$ which has eigenvalue λ. The eigenvalue and eigenvector are approximate. If we desire more accurate values, we can plug v back into the algorithm as v_0 until the lowest eigenvalue converges to a fixed quantity (4).

Discretization of Schrödinger equation

In order to implement the Lanczos algorithm, we must represent the Hamiltonian discretely. The Schrödinger equation in two dimensions (Equation B) gives us:

$$\hat{H}\psi(x,y) = -\frac{\hbar}{2m}\left(\frac{\partial^2\psi(x,y)}{\partial x^2} + \frac{\partial^2\psi(x,y)}{\partial y^2}\right) + V(x,y)\psi(x,y) = E\psi(x,y)$$

We can numerically approximate the following:

$$\frac{\partial^2\psi(x,y)}{\partial x^2} \approx \frac{\psi(x+\Delta x,y) - 2\psi(x,y) + \psi(x-\Delta x,y)}{\Delta x^2}$$

Setting Δx = 1, we can have the following. We take Δx to be one as that is our spacing in the grid on which we solve for the ground state wave function.

$$\frac{\partial^2\psi(x,y)}{\partial x^2} \approx \psi(x+1,y) - 2\psi(x,y) + \psi(x-1,y)$$

It then follows that for Δy = 1:

$$\frac{\partial^2\psi(x,y)}{\partial y^2} = \psi(x,y+1) - 2\psi(x,y) + \psi(x,y-1)$$

Substituting these expressions into Equation (B), we get:

$$-\frac{\hbar}{2m}\big(\psi(x+1,y) + \psi(x,y+1) - 4\psi(x,y) + \psi(x-1,y) + \psi(x,y-1)\big) + V(x,y)\psi(x,y) = E\psi(x,y)$$

This is the discrete version of the Schrödinger equation (8). To solve for the function ψ, we must find the eigenvectors of the Hamiltonian. In particular, we are looking for the lowest eigenvalue and eigenvector, corresponding to the ground state of the system.

$$\hat{H}\psi(x,y) = -\frac{\hbar}{2m}\big(\psi(x+1,y) + \psi(x,y+1) - 4\psi(x,y) + \psi(x-1,y) + \psi(x,y-1)\big) + V(x,y)\psi(x,y)$$

Although the Hamiltonian here is not expressed as a matrix, we can still find its eigenvectors and eigenvalues using the Lanczos algorithm.

Writing a python program to solve the Schrödinger equation

To solve the discrete version of the Schrödinger equation, we encoded it in Python and implemented the Lanczos algorithm to find the ground state wave function. First, we encoded the Hamiltonian as a function that took the wave function and the potential as inputs. We set up the potential within a 128x128 box, assigning a value to each of the 16384 points. We varied the scale of our model in the examples tested.

In this program, we used NumPy, Matplotlib, and SciPy modules. We built this program to solve the Schrödinger equation on a 128 by 128 grid. First, we created a function that applies the potential to a square array. This allows the user to easily change the potential. Then, we implemented the Lanczos algorithm as a function, taking a starting vector, mass, and charge as inputs. After applying the steps described in Reference (9), this function returns an approximation of the lowest eigenvector of a Hermitian matrix, using the SciPy eigh_tridiagonal function. Then, we defined a function that represents a discrete version of the Hamiltonian. This function is used when running the Lanczos algorithm. We ran the Lanczos algorithm using a particle with a charge of $-1.6\text{x}10^{-19}$ C (the charge of an electron) and a mass of $9.11\text{x}10^{-31}$ kg (the mass of an electron) to simulate the interactions of an electron with the input potential. Additionally, we used ħ as $1.055\text{x}10^{-34}$ *Js*. We used the output of the Lanczos algorithm as the input for the next iteration to produce more accurate results. We normalized the resulting wavefunction by dividing by the square root of the sum of the squared values, ensuring that the sum of the probability values in the probability density function would be one. Then, we plotted the potential and the probability density function using matplotlib. We repeated this process, testing the method for other potentials by editing the potential function.

Potential 1: Potential value inside 128 x 128 box is zero. The potential value outside the box is infinite, simulated by boundary condition that Hamiltonian of wave function is zero at boundary. In this example, the scale was one nanometer, , per grid space.

Potential 2: The potential was given by the equation:

$$V(r) = \frac{1}{2}kr^2$$

Where r is the distance from (64,64). In this example, each grid space represents 1μm. Here we took the value of k, the wave factor, to be 10^{16} J/m^2.

Potential 3: The potential was created by three point charges with charge +1.6x10^{-19} C positioned at the following coordinates: (40,40) (65,100) (100,60). The potential due to each of the point charges is given by

$$V(r) = \frac{k \cdot e^2}{r}$$

where k = 9x10^9 N/m^{2}C^2 (Coulomb's constant), e = 1.6x10^{-19} C, the magnitude of charge of an electron, and r is the distance to the point charge. Here we took each grid space to represent 1 nm.To get the overall potential, we summed the three potentials due to the three point charges.

APPENDICES

GitHub link:
https://github.com/williamlongtin/Lanczos

ACKNOWLEDGEMENTS

W.L. would like to thank Kristen Surrao for mentoring him throughout the research process.

Received: September 29, 2022
Accepted: February 24, 2023
Published: June 29, 2023

REFERENCES

1. Morin, David. "Introduction to Quantum Mechanics - Harvard University." *Havard University*, scholar.harvard.edu/files/david-morin/files/waves_quantum.pdf.
2. Nave, Carl. "Quantum Physics." *Hyperphysics Concepts*, hyperphysics.phy-astr.gsu.edu/hbase/quacon.html.
3. Zwiebach, Barton. "Quantum Physics I: Physics." *MIT OpenCourseWare*, MIT OpenCourseWare, ocw.mit.edu/courses/8-04-quantum-physics-i-spring-2016/.
4. "Lanczos Algorithm." Wikipedia, Wikimedia Foundation, 21 Apr. 2022, en.wikipedia.org/wiki/Lanczos_algorithm.
5. Koch, Erik. *The Lanczos Method*. www.cond-mat.de/events/correl11/manuscripts/ koch.pdf .
6. Dongjiao, Tang. *Generalized Matrix Numerov Solutions to the Schrödinger Equation*. www.physics.nus.edu.sg/wp-content/uploads/sites/5/2020/08/hyp-201314-32.pdf.
7. Mena, Arturo. "Solving the 2D Schrödinger Equation Using the Crank-Nicolson Method." *Singularities*, 26 Feb. 2021, artmenlope.github.io/solving-the-2d-Schrödinger-equation-using-the-crank-nicolson-method/.
8. Groenenboom, Gerrit C. and Hm Henk Buck. "Solving the discretized time□independent Schrödinger equation with the Lanczos procedure." *Journal of Chemical Physics* 92 (1990): 4374-4379.
9. Weiss, Daniel. "Chapter 3." *Numerical Methods for the Time-Dependent Schrödinger Equation*, na.uni-tuebingen.de/ex/num4_ss10/lecture/chap3.pdf.
10. Tuckerman, Mark E. "Particle in a 2-Dimensional Box." *LibreTexts Chemistry*, chem.libretexts.org/Bookshelves/Physical_and_Theoretical_Chemistry_Textbook_Maps/Supplemental_Modules_(Physical_and_Theoretical_Chemistry)/Quantum_Mechanics/05.5%3A_Particle_in_Boxes/Particle_in_a_2-Dimensional_Box.
11. Wong, Wai Kui. (2021). Solving 2D Time Independent Schrödinger Equation Using Numerical Method. 10.13140/RG.2.2.12835.99360.
12. Zielinski, Theresa. "9: The Electronic States of the Multielectron Atoms." *Chemistry LibreTexts*, 21 Apr. 2022, chem.libretexts.org/Bookshelves/Physical_and_Theoretical_ Chemistry_Textbook_Maps/Book%3A_Quantum_States_of_Atoms_and_Molecules_(Zielinksi_et_al)/09%3A_The_Electronic_States_of_the_Multielectron_Atoms.
13. Filikhin, Igor. *QUANTUM MECHANICS OF SEMICONDUCTOR QUANTUM DOTS AND RINGS*. ArXiv, 6 June 2012, arxiv.org/ftp/arxiv/papers/1206/1206.1354.pdf.