Numerical Method For Solving the Schrödinger Equation for Infinite and Finite Potential Wells

João Calisto Prof. Dr. José Carlos Egues

Instituto de Física de São Carlos, Universidade de São Paulo (USP), São Carlos, SP, Brasil

July 4, 2025

Abstract

This report treats about solving the Schorodinger equation independent of time for infinite and finite potential using derivative approximations. The main objective was to use the finite differences method to discretize the differential equation and converting it to a eigenvalues/eigenvectors problem. For that was used python with the numpy and scipy libraries. The theoretical values for the infinite potential was compaired with the simulated results, presenting an error of 0.12% for the first 10 states of energy. Furthermore, the probability densities graphics was ploted with the potential barrier to compair the differences between he finite and infinite potential.

1 Introduction

Solving the Schrödinger Equation problems numerically, analyzing, and comparing the results with theoretical and experimental values is an efficient way to enhance and develop a solid intuition about it.

For this, we can use different methods for interpretate, model and simulate the Schrodinger Equation.

This report present an way to use numerical approximations to deal with the Time independent Schrodinger Equation for the infinity and finite potential.

First, we will review the theoric concepts used and, in sequence, will be presented the used numerical method, and finally, the numerical results will be compaired with the theoretical calculations and experimental values.

2 Theoretical Review

2.1 The Equation

The Time-Independent Schrodinger equation given by:

$$-\frac{\hbar^2}{2m}\frac{\mathrm{d}\Psi^2(\mathbf{r})}{\mathrm{d}x^2} + V(\mathbf{r})\Psi(\mathbf{r}) = E\Psi(\mathbf{r}) \tag{1}$$

is the spacial part of the general Schrodinger equation solution.

We can rearrange this equation to a more convenient form for our numerical method:

$$-\frac{\mathrm{d}\Psi^2(\mathbf{x})}{\mathrm{d}x^2} + 2\frac{m}{\hbar^2}V(\mathbf{x})\Psi(\mathbf{x}) = 2\frac{m}{\hbar^2}E\Psi(\mathbf{x}) \tag{2}$$

This equation form is an Sturm-Liouville[1, p. 511-524] problem, used to find the eigenvalues and eigenfunctions, what solves the equation.

$$\hat{L}\Psi(\mathbf{x}) = \lambda\Psi(\mathbf{x}) \tag{3}$$

For this problem, the operator $\hat{L}$ is:

$$\hat{L} = -\frac{\mathrm{d}^2}{\mathrm{d}x^2} + 2\frac{m}{\hbar^2}V(\mathbf{x}) \tag{$\hat{L}$ Operator}$$

and the eigenvalue λ form is given by:

$$\lambda = 2\frac{m}{\hbar^2}E \tag{4}$$

As we are treating with infinite dimension, and numerically we can only work with finite dimension, we need discretize the problem to find an approximation of the theoretical eigenvalues and eigenfunctions.

In this case, instead of searching for eigenvalues using Sturm-Liouville theory (for infinite dimension space), we discretize the functions to finite vectors and solve a diagonalization problem [2, p. 135-140] to find the eigenvalues and use them to find the energy states.

2.2 Discretization

To discretize the equation we can use the second derivative approximation, which is given by:

$$\frac{d^2\Psi}{dx^2} \approx \frac{\Psi_{k+1} - 2\Psi_k + \Psi_{k-1}}{\Delta x^2} \tag{5}$$

We can rewrite x as a discrete variable $x_k = k\Delta x$, where Δx is the step size and k is the index of the discrete point.

Applying this approximation in the equation, we can rewrite the equation as:

$$-\Psi_{k+1} + 2\Psi_k - \Psi_{k-1} + 2\frac{m}{\hbar^2}V_k\Psi_k\Delta x^2 = 2\frac{m}{\hbar^2}\Delta x^2 E_k\Psi_k \tag{Discretized Equation}$$

The discrete $\hat{L}$ operator can be written as a matrix: (the term $2\frac{m}{\hbar^2}$ is defined as the parameter α)

$$M = \begin{pmatrix} 2+\alpha V_0 & -1 & 0 & \cdots & 0 \\ -1 & 2+\alpha V_1 & -1 & \cdots & 0 \\ 0 & -1 & 2+\alpha V_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & -1 \\ 0 & 0 & 0 & -1 & 2+\alpha V_{N-1} \end{pmatrix} \tag{6}$$

The eigenvalues are given by the 1×N matrix:

$$\Lambda = \begin{bmatrix} 2\frac{m}{\hbar^2}E_0 & 2\frac{m}{\hbar^2}E_1 & 2\frac{m}{\hbar^2}E_2 & \dots & 2\frac{m}{\hbar^2}E_{N-1} \end{bmatrix} \tag{7}$$

The solutions Ψ_k can be represented as a components of N×1 matrix:

$$\Psi = \begin{bmatrix} \Psi_0 \\ \Psi_1 \\ \Psi_2 \\ \vdots \\ \Psi_{N-1} \end{bmatrix} \tag{8}$$

Replacing the terms by matrices, we can rewrite the discretized equation Discretized Equation in a matrix form:

$$M\Psi = \Lambda\Psi \tag{9}$$

The direct method to find this eigenvalues and eigenvectors is to solve the characteristic polynomial of the matrix. As the matrix dimension is high, in this project we will use the numpy library to find the eigenvalues and eigenvectors of the matrix, which is a numerical method to solve the characteristic polynomial.

After finding each eigenvalue λ_k, we can find the corresponding energy eigenvalues E_k using the relation:

$$E_k = \frac{\hbar^2\lambda_k}{2m\Delta x^2} \tag{10}$$

Where each energy state E_k corresponds to the eigenvalue λ_k of the discretized operator $\hat{L}$.

3 The Infinite Potential Box

The infinite potential box describes a system where there is a particle confined in a limited region with a inifinite potential outside the region. Using the unidimensional case, we can describe the potential as:

$$V(x) = \begin{cases} 0 & \text{if } 0 < x < L \\ \infty & \text{otherwise} \end{cases} \tag{11}$$

3.1 Solving the Equation

For the infinite potential box, the second term of the $\hat{L}$ Operator is zero, as the potential is zero inside the box.

The Discretized Equation becomes:

$$-\Psi_{k+1} + 2\Psi_k - \Psi_{k-1} = 2\frac{m}{\hbar^2}\Delta x^2 E\Psi_k \tag{12}$$

And with that, the principal diagonal will have only the term 2, while the secondary diagonal will have the term −1:

$$M_{ii} = 2; \quad M_{i,i+1} = -1; \quad M_{i,i-1} = -1 \tag{13}$$

4 The Finite Potential Box

The finite potential is described by the particle confined in a region where the potential is zero and finite outside the region. Using the unidimensional case, we can describe the potential as:

$$V(x) = \begin{cases} 0 & \text{if } 0 < x < L \\ V_0 & \text{if } x < 0 \text{ or } x > L \end{cases} \tag{14}$$

4.1 Solving the Equation

For the finite potential box, the Discretized Equation becomes:

$$-\Psi_{k+1} + 2\Psi_k - \Psi_{k-1} + 2\frac{m}{\hbar^2}\Delta x^2 V_0 = 2\frac{m}{\hbar^2}\Delta x^2 E\Psi_k \tag{15}$$

The matrix form follows the same pattern, but now we have principal diagonal with the potential term:

$$M_{ii} = 2 + \alpha V_0; \quad M_{i,i+1} = -1; \quad M_{i,i-1} = -1 \tag{16}$$

5 The Numerical Method

The numerical method was implemented using python with numpy and scipy libraries.

The used mass was the electron mass $m = 9.10938356 \times 10^{-31}$kg, and the reduced Planck constant $\hbar = 1.0545718 \times 10^{-34}$J.s.

The potential box length was set to $L = 1$Å and the step size was set to $\Delta x = 10^{-12}$m.

In this code, was created a numpy array for principal diagonal and a numpy array for the secondary diagonal. The scipy.linalg eigh_tridiagonal function was used to find the eigenvalues and eigenvectors of the tridiagonal matrix, which is a numerical method to solve the characteristic polynomial of the matrix. The eigenvalues were then converted to energy values.

The code is available in the repository `https://://github.com/stdmedoth/Schrodinger_NumericalMethods`.

6 Discuss and Comparison

The results were compared with the theoretical values for the infinite potential box and finite potential box. The theoretical values for the infinite potential box are given by:

$$E_n = \frac{n^2\pi^2\hbar^2}{2mL^2} \tag{17}$$

For the first 10 energy states, the theoretical values are:

State	Energy (eV)
1	37.573
2	150.292
3	338.157
4	601.168
5	939.325
6	1352.628
7	1841.077
8	2404.672
9	3043.413
10	3757.3

The numerical results for the infinite potential box were:

State	Energy (eV)
1	37.5278
2	150.1112
3	337.7488
4	600.4388
5	938.1788
6	1350.9653
7	1838.7942
8	2401.6609
9	3039.5597
10	3752.4843

The results are very close to the theoretical values, with a medium error of 0.12% for the first 10 energy states.

As the eigenvectors are the wave functions of the system, we can calculate the wave probability density for each energy state. The wave probability density is given by:

$$P(x) = |\Psi(x)|^2 \tag{18}$$

The wave probability density for the first 3 energy states was plotted using matplotlib library.

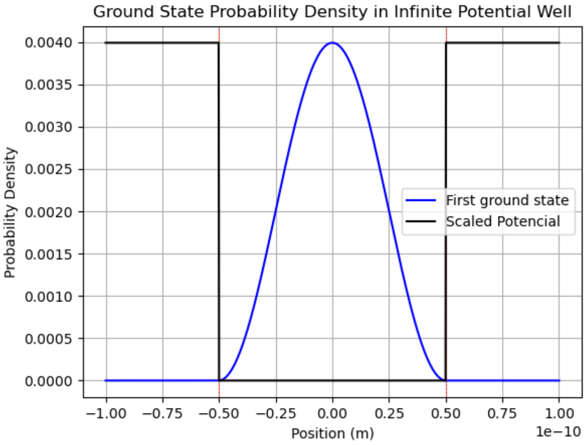

Figure 1: Probability density for the first energy state in an infinite potential box.

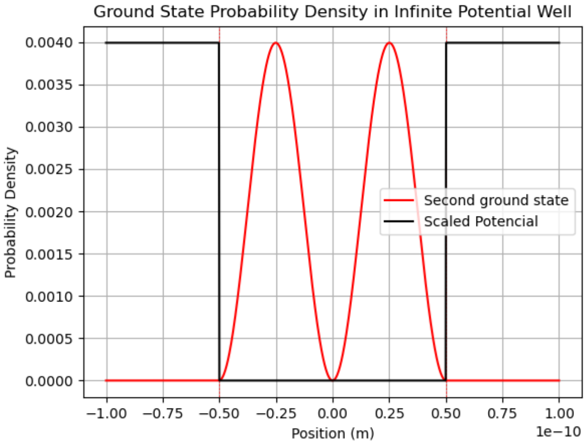

Figure 2: Probability density for the second energy state in an infinite potential box.

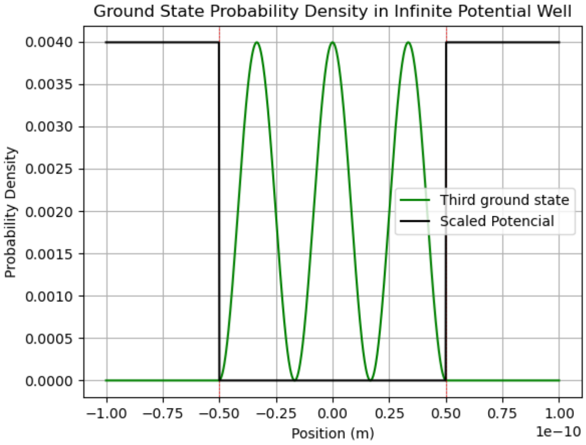

Figure 3: Probability density for the third energy state in an infinite potential box.

For the finite potential box, was used V0 = 1.0e-16 J and the theoretical values are more complicated to calculate, as they depend on the potential value and the energy states. So was calculated the values numerically for the first 10 energy states, using the same method as for the infinite potential box. The numerical results for the infinite potential box were:

State	Energy (eV)
1	$1.449e-12$
2	$5.774e-12$
3	$1.294e-11$
4	$2.295e-12$
5	$3.583e-12$
6	$5.158e-12$
7	$7.020e-12$
8	$9.167e-12$
9	$1.160e-10$
10	$1.432e-10$

This values can be more precise adjusting the Δx variable on python simulation (the infinitesimal step)

The wake probability density for the first 3 energy states was plotted using matplotlib library.

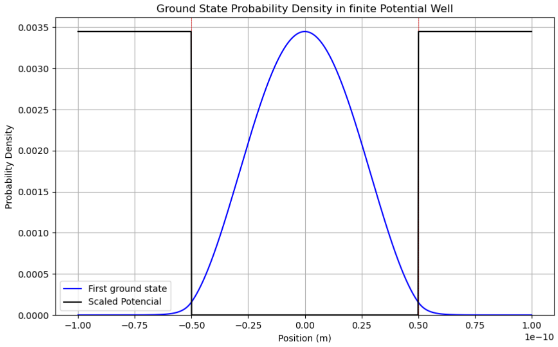

Figure 4: Probability density for the first energy state in an finite potential box.

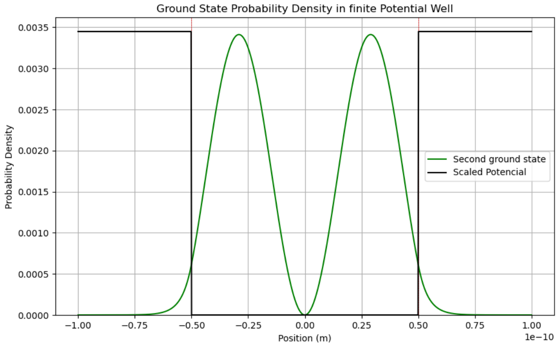

Figure 5: Probability density for the second energy state in an finite potential box.

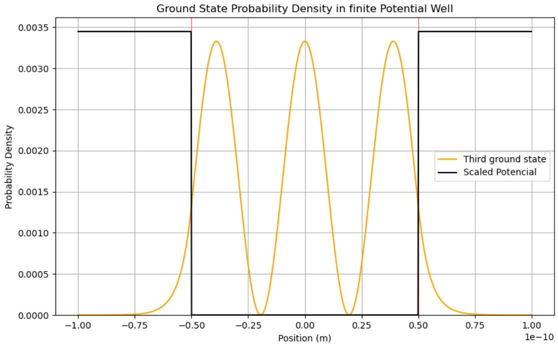

Figure 6: Probability density for the third energy state in an finite potential box.

The graphs for the finite potential reveals that the wave function penetres the potential barrier, even when the energy is lower than the potencial value. This shape is in accordding with the tunneling fenomenum. Inside the barrier, the graphs has the same shape as the infinite potential box, but with a lower amplitude, as the potential barrier reduces the wave function amplitude.

7 Conclusion

This report have demonstrated how numerical approximations can give precise results for the grounded states particles using the Schrodinger Equation derivative approximation with matrix diagonalization. For the infinite box the simulation have a medium error of 0.12%, and we can descrease it using a lower infinitesimal step for the derivative approximation. In resume, this report not only served as a pratical exercise in applying numerical methods to quantum mechanics, but also improve the intuition about energy quantization and the wave equation meaning. The use of python with numpy and scipy proved to be a good toolset for solving simple quantum problems, providing a solid foundation for exploring other more complex problems in the future.

References

[1] Richard C. DiPrima Willian E. Boyce. *Elementary Differential Equations and Boundary Value Problems*. LTC, 2010.

[2] Sergio Luis Zani. *Algebra Linear*. ICMC São Carlos, 2023.