The Two-Dimensional Time-Independent Schrödinger Equation. Solution by Direct Matrix Method and Lanczos Algorithm.

Reinaldo Magallanes Saunders*

July 2021

1 Introduction

The main objective of this assignment is to apply the direct matrix method to the two-dimensional time-independent Schrödinger equation and estimate the lowest bound-state eigenvalues of the Hamiltonian. We will consider three common two-dimensional potentials, namely, the rectangular well, the harmonic oscillator and the Gaussian well. This will be done using a script in Python.

2 The Schrödinger Equation in 2 Dimensions

The time-independent Schrödinger equation for a particle of mass m in a two dimensional potential V can be written as:

$$\left[-\frac{\hbar^2}{2m}\left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}\right)+V(x,y)\right]\psi(x,y)=E\psi(x,y) \tag{1}$$

The potentials considered will be the rectangular well

$$V_r(x,y)=\begin{cases}0 & \text{if } |x|<a/2,\ |y|<b/2\\ \infty & \text{elsewhere}\end{cases}, \quad a,b>0 \tag{2}$$

the harmonic oscillator

$$V_h(x,y)=\frac{1}{2}m\omega^2(x^2+y^2) \tag{3}$$

and the Gaussian well

$$V_g(x,y)=-V_0e^{-\alpha(x^2+y^2)}, \quad V_0,\alpha>0 \tag{4}$$

*e-mail: rei.magallanes@gmail.com

Of these potentials, only the rectangular well and the harmonic oscillator have exact solutions.

2.1 Solution for the Rectangular Well

Since V_r can be written as

$$V_r(x,y) = V_r^{(x)}(x) + V_r^{(y)}(y)$$

for

$$V_r^{(x)}(x) = \begin{cases} 0 & \text{if } |x| < a/2 \\ \infty & \text{if } |x| > a/2 \end{cases}, \quad V_r^{(y)}(y) = \begin{cases} 0 & \text{if } |y| < b/2 \\ \infty & \text{if } |y| > b/2 \end{cases},$$

separation of variables can be used to obtain two independent one-dimensional Schrödinger equations.

Thus, since we know the solution for a one-dimensional potential well, we get

$$\psi_{n_x,n_y}(x,y) = \frac{2}{\sqrt{ab}} \sin\left[\frac{n_x\pi}{a}\left(x+\frac{a}{2}\right)\right] \sin\left[\frac{n_y\pi}{b}\left(y+\frac{b}{2}\right)\right], \quad n_x, n_y = 1,2,3,\ldots$$

$$E_{n_x,n_y} = \frac{\hbar^2\pi^2}{2m}\left(\frac{n_x^2}{a^2} + \frac{n_y^2}{b^2}\right)$$

2.2 Solution for the Harmonic Oscillator

This potential is already written as a sum of single-variable terms. Using separation of variables and the solution for the one-dimensional harmonic oscillator, it's straightforward to see that

$$\psi_{n_x,n_y}(x,y) = \frac{e^{-\frac{x^2+y^2}{2l_0^2}}}{l_0\sqrt{\pi(n_x)!(n_y)!2^{n_x+n_y}}} H_{n_x}\left(\frac{x}{l_0}\right) H_{n_y}\left(\frac{y}{l_0}\right), \quad l_0 = \sqrt{\frac{\hbar}{m\omega}}$$

with

$$E_{n_x,n_y} = \hbar\omega\left(n_x + n_y + 1\right), \quad n_x, n_y = 0,1,2,\ldots$$

or

$$E_n = \hbar\omega(n+1), \quad n = n_x + n_y, \quad n_x, n_y = 0,1,2,\ldots$$

We note that the degeneracy of E_n is $n+1$.

2.3 The Gaussian well.

The Gaussian well has no analytical solution, but it always admits at least one bound state[1][2]. The number of bound states will depend on the depth V_0 and the parameter α. As seen in Figure 1, the parameter α determines the width of the well.

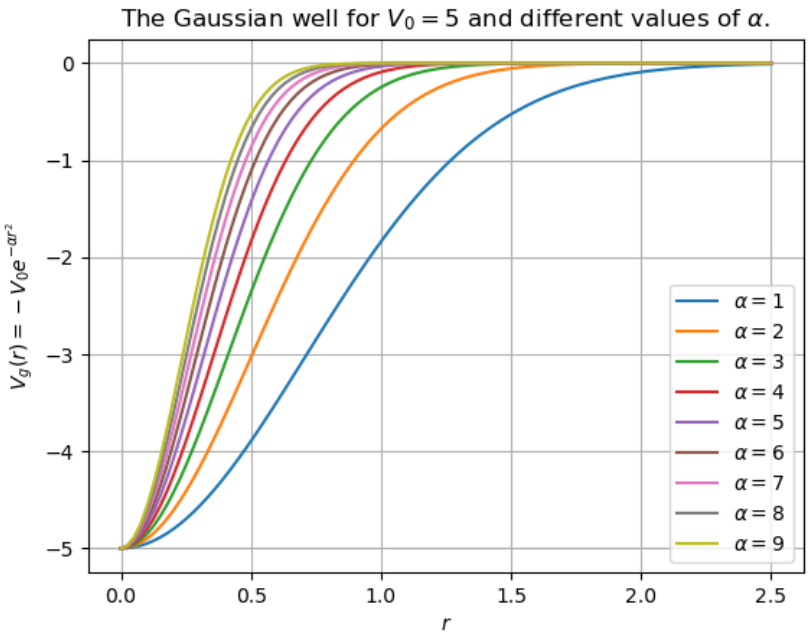

Figure 1: How the Gaussian well changes with the parameter α.

3 The Direct Matrix Method

The direct matrix method involves using finite differences to approximate the second-order derivatives in the Hamiltonian. By discretizing over a suitable finite grid, a discrete form of the Schrödinger equation can be obtained. The main issue lies in the fact that the wave function will be described by two indices, i.e., a two-dimensional array. So, by 'flattening' the wave function array a one-dimensional array is obtained for the wave function and a two-dimensional array is obtained for the Hamiltonian, which then can be diagonalized.

3.1 The Second Derivative Central Difference

Finite difference formulas can be obtained by using Taylor series expansions. For a function $f(x)$ and a step size Δx, we can write

$$f(x + \Delta x) = f(x) + \Delta x f'(x) + \frac{1}{2}(\Delta x)^2 f''(x) + \frac{1}{6}(\Delta x)^3 f'''(x) + \mathcal{O}((\Delta x)^4)$$

Solving for $f'(x)$ we get forwards difference formula:

$$f'(x) = \frac{f(x + \Delta x) - f(x)}{\Delta x} + \mathcal{O}(\Delta x)$$

Similarly, we can write

$$f(x - \Delta x) = f(x) - \Delta x f'(x) + \frac{1}{2}(\Delta x)^2 f''(x) - \frac{1}{6}(\Delta x)^3 f'''(x) + \mathcal{O}((\Delta x)^4)$$

to get the backwards difference formula:

$$f'(x) = \frac{f(x) - f(x - \Delta x)}{\Delta x} + \mathcal{O}(\Delta x)$$

If, instead, we first subtract $f(x-\Delta x)$ from $f(x+\Delta x)$ to get:

$$f(x+\Delta x)-f(x-\Delta x)=2\Delta x f'(x)+\frac{1}{3}(\Delta x)^3 f'''(x)+\mathcal{O}((\Delta x)^5)$$

we can get the central difference formula:

$$f'(x)=\frac{f(x+\Delta x)-f(x-\Delta x)}{2\Delta x}+\mathcal{O}((\Delta x)^2)$$

If $\Delta x < 1$, the central difference formula is a better approximation for the first-order derivative, since the error in the approximation is of order $(\Delta x)^2$.

For the second-order derivative, the forwards and backwards difference formulas also have errors of order Δx, so the central difference formula will be used since it's error is of order $(\Delta x)^2$. It can be obtained by adding $f(x-\Delta x)$ and $f(x+\Delta x)$ to get:

$$f(x-\Delta x)+f(x+\Delta x)=2f(x)+(\Delta x)^2 f''(x)+\mathcal{O}((\Delta x)^4)$$

Solving for $f''(x)$ we get:

$$f''(x)=\frac{f(x-\Delta x)-2f(x)+f(x+\Delta x)}{(\Delta x)^2}+\mathcal{O}((\Delta x)^2) \tag{5}$$

This is the second derivative central difference formula, which is accurate to second order in Δx.

3.2 The One-Dimensional Case

In order to make the two-dimensional case easier to deal with, the one-dimensional case will be tackled, since the 'flattening' process can be somewhat challenging.

For the one-dimensional Schrödinger equation with appropriate boundary conditions

$$\left[-\frac{\hbar^2}{2m}\frac{d^2}{dx^2}+V(x)\right]\psi(x)=E\psi(x),\quad \psi(a)=\alpha,\quad \psi(b)=\beta,$$

a finite grid of N equispaced points x_i is introduced, such that

$$x_0=a,\quad x_{N-1}=b \quad\Longrightarrow\quad \begin{cases}\psi(a)=\psi(x_0)\equiv\psi_0=\alpha\\ \psi(b)=\psi(x_{N-1})\equiv\psi_{N-1}=\beta\end{cases}$$

If we define

$$x_{j+1}-x_j=\Delta x$$

we can apply the second derivative central difference formula to the Schrödinger equation to obtain

$$-\frac{\hbar^2}{2m}\frac{\psi_{j-1}-2\psi_j+\psi_{j+1}}{(\Delta x)^2}+V_j\psi_j=E\psi_j+\mathcal{O}((\Delta x)^2),\quad j=1,2,\ldots,N-2.$$

By setting

$$\gamma = \frac{\hbar^2}{2m(\Delta x)^2}$$

we can write the system of equations in the following form:

$$\begin{cases} -\gamma\alpha + (2\gamma + V_1)\psi_1 - \gamma\psi_2 = E\psi_1 \\ -\gamma\psi_1 + (2\gamma + V_2)\psi_2 - \gamma\psi_3 = E\psi_2 \\ -\gamma\psi_2 + (2\gamma + V_3)\psi_3 - \gamma\psi_4 = E\psi_3 \\ \quad\vdots \qquad\qquad \vdots \qquad\quad \vdots \qquad \vdots \\ -\gamma\psi_{N-4} + (2\gamma + V_{N-3})\psi_{N-3} - \gamma\psi_{N-2} = E\psi_{N-3} \\ -\gamma\psi_{N-3} + (2\gamma + V_{N-2})\psi_{N-2} - \gamma\beta = E\psi_{N-2} \end{cases}$$

This can be rewritten as a matrix equation, by moving the terms $-k\alpha$ and $-k\beta$ to the right-hand side:

$$\begin{bmatrix} 2\gamma + V_1 & -\gamma & 0 & \dots & 0 & 0 \\ -\gamma & 2\gamma + V_2 & -\gamma & \dots & 0 & 0 \\ 0 & -\gamma & 2\gamma + V_3 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 2\gamma + V_{N-3} & -\gamma \\ 0 & 0 & 0 & \dots & -\gamma & 2\gamma + V_{N-2} \end{bmatrix} \begin{bmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \\ \vdots \\ \psi_{N-3} \\ \psi_{N-2} \end{bmatrix} = E \begin{bmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \\ \vdots \\ \psi_{N-3} \\ \psi_{N-2} \end{bmatrix} + \gamma \begin{bmatrix} \alpha \\ 0 \\ 0 \\ \vdots \\ 0 \\ \beta \end{bmatrix}$$

Introducing H as the discretized Hamiltonian and $\boldsymbol{\psi}$ as the discretized wave function, with $\mathbf{b}$ as the boundary condition vector, this can be expressed as:

$$H\boldsymbol{\psi} = E\boldsymbol{\psi} + k\mathbf{b}$$

Since we are interested in wave functions of bound states, for which $\psi(x \to \pm\infty) = 0$, our boundary conditions will be

$$\psi_0 = 0, \quad \psi_{N-1} = 0 \quad \Longrightarrow \quad \mathbf{b} = 0$$

Thus, the matrix equation reduces to an eigenvalue problem

$$H\boldsymbol{\psi} = E\boldsymbol{\psi}$$

where we can diagonalize H to find it's eigenvectors $\boldsymbol{\psi}$ along with their eigenvalues E.

This method will provide accurate results for small eigenvalues, while for highly excited states it may be that the error increases with the step size[5]. This could happen in cases where the wave function no longer satisfies the boundary conditions.

3.3 The Two-Dimensional Case and Natural Ordering

For the two-dimensional Schrödinger equation

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

a two-dimensional $N \times N$ grid is introduced, such that

$$\psi(x_0, y) = \psi(x_{N-1}, y) = 0, \quad \psi(x, y_0) = \psi(x, y_{N-1}) = 0 \tag{6}$$

and

$$x_j - x_{j-1} = \Delta x, \quad y_j - y_{j-1} = \Delta y \tag{7}$$

Applying the second derivative central difference formula to both second-order derivatives in the Schrödinger equation we get

$$-\frac{\hbar^2}{2m}\left[\frac{\psi_{i-1,j} - 2\psi_{i,j} + \psi_{i+1,j}}{(\Delta x)^2} + \frac{\psi_{i,j-1} - 2\psi_{i,j} + \psi_{i,j+1}}{(\Delta y)^2}\right] + V_{i,j}\psi_{i,j} = E\psi_{i,j} + \mathcal{O}(\max[(\Delta x)^2, (\Delta y)^2])$$

where

$$\psi_{i,j} = \psi(x_i, y_j), \quad V_{i,j} = V(x_i, y_j)$$

Setting

$$\eta = \frac{\hbar^2}{2m(\Delta x)^2}, \quad \mu = \frac{\hbar^2}{2m(\Delta y)^2}, \quad \nu_{i,j} = 2(\eta + \mu) + V_{i,j} \tag{8}$$

we can write

$$-\eta\psi_{i-1,j} - \mu\psi_{i,j-1} + \nu_{i,j}\psi_{i,j} - \eta\psi_{i+1,j} - \mu\psi_{i,j+1} = E\psi_{i,j} + \mathcal{O}(\max[(\Delta x)^2, (\Delta y)^2]) \tag{9}$$

At this point, since the discretized wave function is represented by a two-dimensional array, we introduce the concept of natural ordering in order to re-frame the wave function as a one-dimensional array. This is done by setting up a new index k such that:

- The value $k = 0$ is assigned to the pair (x_1, y_1).
- An increment in the x index by one translates to an increment in k by 1.
- An increment in the y index by one translates to an increment in k by $N - 2$.

Essentially, for the discretized wave function, this is a column by column indexing.

$$\begin{bmatrix} \psi_{1,1} & \psi_{1,2} & \psi_{1,3} & \cdots & \psi_{1,N-2} \\ \psi_{2,1} & \psi_{2,2} & \psi_{2,3} & \cdots & \psi_{2,N-2} \\ \psi_{3,1} & \psi_{3,2} & \psi_{3,3} & \cdots & \psi_{3,N-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \psi_{N-2,1} & \psi_{N-2,2} & \psi_{N-2,3} & \cdots & \psi_{N-2,N-2} \end{bmatrix}$$

$$\downarrow$$

$$\boldsymbol{\psi} = (\psi_{1,1}, \psi_{2,1}, \ldots, \psi_{N-2,1}, \psi_{1,2}, \psi_{2,2}, \ldots, \psi_{N-2,2}, \psi_{3,1}, \ldots\ldots, \psi_{1,N-2}, \psi_{2,N-2}, \ldots, \psi_{N-2,N-2})$$

Thus, this new index k allows for the discretized wave function to be represented by a one-dimensional array with $(N-2)^2$ entries. It's not hard to see that for an arbitrary $\psi_{i,j}$ and it's corresponding $\boldsymbol{\psi}_k$, the following is satisfied:

$$k = (i-1) + (j-1)(N-2) \iff \begin{cases} i = [k \mod (N-2)] + 1 \\ j = \left\lfloor \dfrac{k}{N-2} \right\rfloor + 1 \end{cases}$$

where $\lfloor x \rfloor$ denotes the floor function.

With this new index k, Equation (9) can be written as

$$-\mu\boldsymbol{\psi}_{k-(N-2)} - \eta\boldsymbol{\psi}_{k-1} + \nu_{i,j}\boldsymbol{\psi}_k - \eta\boldsymbol{\psi}_{k+1} - \mu\boldsymbol{\psi}_{k+(N-2)} = E\boldsymbol{\psi}_k, \quad k = 0, 1, \ldots, (N-2)^2 - 1. \tag{10}$$

where it is assumed that

$$m \notin \{0, 1, \ldots, (N-2)^2 - 1\} \implies \boldsymbol{\psi}_m = 0$$

To understand how this can be expressed as an eigenvalue problem, we make a couple key observations:

- Since the wave function has been 'flattened' to a $(N-2)^2$-dimensional vector, the Hamiltonian will be represented as a $(N-2)^2 \times (N-2)^2$ matrix.
- If $\mu = 0$, we would have a system of equations quite similar to the one-dimensional problem. Thus, the terms $-\eta\boldsymbol{\psi}_{k-1} + \nu_{i,j}\boldsymbol{\psi}_k - \eta\boldsymbol{\psi}_{k+1}$ correspond to a tridiagonal part of the Hamiltonian.
- There is a sort of 'discontinuity' in the new indexing scheme for values of $k+1$ that are a multiple of $(N-2)$ for $\boldsymbol{\psi}_{k+1}$ terms that came from $\psi_{i+1,j}$ terms that were introduced by the finite difference approximation for the second-order x derivative. This is as a result of how boundary conditions have been handled; the k-indexing scheme does not include them.

  To see this, let's take the case for $k = (N-2) - 1$, corresponding to $i = N-2$ and $j = 1$. Thus, by equations (9) and (10):

  $$-\mu\psi_{N-2,0} - \eta\psi_{(N-2)-1,1} + \nu_{N-2,1}\psi_{N-2,1} - \eta\psi_{(N-2)+1,1} - \mu\psi_{N-2,2} = E\psi_{N-2,1}$$

  $$-\mu\boldsymbol{\psi}_{-1} - \eta\boldsymbol{\psi}_{(N-2)-2} + \nu_{(N-2),1}\boldsymbol{\psi}_{(N-2)-1} - \eta\boldsymbol{\psi}_{(N-2)} - \mu\boldsymbol{\psi}_{2(N-2)-1} = E\boldsymbol{\psi}_{(N-2)-1}$$

  As mentioned previously $\boldsymbol{\psi}_{-1} = 0$, since $\psi_{N-2,0} = 0$, by boundary conditions. The issue lies with the terms $\psi_{(N-2)+1,1}$ and $\boldsymbol{\psi}_{(N-2)}$. They do not represent the same value, since $\psi_{(N-2)+1,1} = \psi_{N-1,1}$, which is zero by boundary conditions, and $\boldsymbol{\psi}_{(N-2)} \mapsto \psi_{1,2}$. So, in this case, the term $-\eta\boldsymbol{\psi}_{(N-2)}$ should not be present.

  The same issue arises for values of k that are multiples of $(N-2)$ for $\boldsymbol{\psi}_{k-1}$ terms.

  This is best seen in a concrete example; the case for $N = 5$ is presented in Appendix A.

So, the cases where $k+1$ or k are a multiple of $(N-2)$ should be treated with care.

Fortunately, this allows for a straightforward way to describe the Hamiltonian by considering it as a block matrix. The issues discussed previously define $(N-2) \times (N-2)$ sub-matrices such that the Hamiltonian can be described as a block-tridiagonal matrix.

In the main diagonal (block-wise) we have $(N-2) \times (N-2)$ matrices similar to those of the one-dimensional case, given by the terms $-\eta\boldsymbol{\psi}_{k-1} + \nu_{i,j}\boldsymbol{\psi}_k - \eta\boldsymbol{\psi}_{k+1}$. In the off-diagonals (block-wise) we have $(N-2) \times (N-2)$ identity matrices multiplied by $-\mu$, given by the terms $-\mu\boldsymbol{\psi}_{k-(N-2)}$ and $-\mu\boldsymbol{\psi}_{k+(N-2)}$.

And so, we can express the Hamiltonian matrix in the following way:

$$H = \begin{bmatrix} \mathbb{V}_1 & -\mu\mathbb{I}_{N-2} & 0 & 0 & \dots & 0 & 0 \\ -\mu\mathbb{I}_{N-2} & \mathbb{V}_2 & -\mu\mathbb{I}_{N-2} & 0 & \dots & 0 & 0 \\ 0 & -\mu\mathbb{I}_{N-2} & \mathbb{V}_3 & -\mu\mathbb{I}_{N-2} & \dots & 0 & 0 \\ 0 & 0 & -\mu\mathbb{I}_{N-2} & \mathbb{V}_4 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \mathbb{V}_{N-3} & -\mu\mathbb{I}_{N-2} \\ 0 & 0 & 0 & 0 & \dots & -\mu\mathbb{I}_{N-2} & \mathbb{V}_{N-2} \end{bmatrix} \tag{11}$$

where $\mathbb{I}_{N-2}$ is the $(N-2)\times(N-2)$ identity matrix and

$$\mathbb{V}_j = \begin{bmatrix} \nu_{1,j} & -\eta & 0 & 0 & \dots & 0 & 0 \\ -\eta & \nu_{2,j} & -\eta & 0 & \dots & 0 & 0 \\ 0 & -\eta & \nu_{3,j} & -\eta & \dots & 0 & 0 \\ 0 & 0 & -\eta & \nu_{4,j} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \nu_{N-3,j} & -\eta \\ 0 & 0 & 0 & 0 & \dots & -\eta & \nu_{N-2,j} \end{bmatrix}$$

Thus, we now have a matrix eigenvalue problem

$$H\boldsymbol{\psi} = E\boldsymbol{\psi}.$$

3.3.1 Different number of points in each direction

One thing that should be noted is that it's not necessary for the grid to be $N \times N$.

For a $N_x \times N_y$ grid the overall structure of the Hamiltonian matrix is the same. Major differences with this grid are listed below:

- Boundary conditions are now

$$\psi(x_0, y) = \psi(x_{N_x-1}, y) = 0, \quad \psi(x, y_0) = \psi(x, y_{N_y-1}) = 0$$

- The k-indexing scheme satisfies

$$k = (i-1) + (j-1)(N_x-2) \iff \begin{cases} i = [k \mod (N_x-2)] + 1 \\ j = \left\lfloor \dfrac{k}{N_x-2} \right\rfloor + 1 \end{cases}$$

  and the system of equations is

  $$-\mu\boldsymbol{\psi}_{k-(N_x-2)} - \eta\boldsymbol{\psi}_{k-1} + \nu_{i,j}\boldsymbol{\psi}_k - \eta\boldsymbol{\psi}_{k+1} - \mu\boldsymbol{\psi}_{k+(N_x-2)} = E\boldsymbol{\psi}_k, \quad k = 0, 1, \dots, (N_x-2)(N_y-2)-1.$$

  The 'discontinuity' issue discussed previously now arises for values of k or $k+1$ that are a multiple of (N_x-2).

- The Hamiltonian is a $(N_x-2)(N_y-2) \times (N_x-2)(N_y-2)$ block-tridiagonal matrix. Now, $(N_x-2)\times(N_x-2)$ are the dimensions of the sub-matrices and, block-wise, there are (N_y-2) matrices $\mathbb{V}_j$ in the main diagonal, where $j = 1, 2, \dots, (N_y-2)$.

4 The Lanczos Algorithm

The Lanczos algorithm is an iterative diagonalization algorithm particularly useful when one is interested in a subset of eigenvalues of a Hermitian matrix which are near the extremes of it's spectrum. Strictly speaking, the algorithm does not need access to the matrix explicitly, but rather only to a function that computes the product of said matrix by an arbitrary vector.

For a $M \times M$ Hermitian matrix H, the starting point of the algorithm is a M-dimensional vector $|q_0\rangle = 0$ and another arbitrary M-dimensional vector $|q_1\rangle$, which is normalised. Then, generate the Lanczos vectors $|q_j\rangle$ as follows:

$$|w_{j+1}\rangle = H\,|q_j\rangle - \alpha_j\,|q_j\rangle - \beta_j\,|q_{j-1}\rangle\,, \quad |q_{j+1}\rangle = \frac{1}{\beta_{j+1}}\,|w_{j+1}\rangle \tag{12}$$

where

$$\alpha_j = \langle q_j|H|q_j\rangle\,, \quad \beta_j = \langle w_j|w_j\rangle^{\frac{1}{2}}$$

After k steps, with $k \leq M$, the algorithm yields an orthonormal set of vectors $\{|q_j\rangle\}$. Furthermore, after M steps, H, when expressed with respect to the set $\{|q_j\rangle\}$, is a tridiagonal symmetric matrix with elements α_1, α_2, ..., α_M on the main diagonal and elements β_2, β_3, ..., β_M on the off-diagonals.

When $k < M$, a $k \times k$ tridiagonal matrix T_k can be constructed in the same fashion. The eigenvalues of the matrix T_k obtained after k iterations will converge to the eigenvalues of the original matrix. It can be shown that the lowest (and highest) eigenvalues converge first. This is why the Lanczos algorithm is used for these parts of the spectrum. Since the diagonalization of a tridiagonal matrix is computationally quick, this algorithm gives a useful way to compute the lowest eigenvalues of a large matrix.

The main drawback of the Lanczos algorithm is that it is prone to numerical instability. This stems from the fact that the orthogonality between the vectors $\{|q_j\rangle\}$ is lost rather quickly and, in some cases, linear dependence between vectors is present, both of which affect the eigenvalues obtained after diagonalization.

5 Algorithm and Implementation

First, we need to discuss grid selection and nondimensionalization. Since the boundary conditions (6) must be satisfied, the overall size of the grid will depend upon the potential. Furthermore, given the potentials we are interested in, the grid will always be centred at the origin. With this in mind, two options were considered.

One option is for the code to require certain inputs relating to the grid and potential and then estimate a suitable grid, while using Rydberg atomic units. For instance, for the harmonic oscillator, the code would have the frequency ω and step sizes in both directions Δx, Δy. With these parameters and certain considerations (which will be discussed later) a suitable grid, centred at the origin, would have been estimated.

The second option, which is the one chosen, is to first use nondimensionalization or scaling, which will somewhat fix the size of the grid, and then the code will have necessary

parameters as inputs. This may be more convenient since, with this procedure, we know the overall size of the grid before we choose the step sizes Δx and Δy.

5.1 Nondimensionalization

5.1.1 The Rectangular Well

First, for the rectangular well we set

$$x' = \frac{x}{a}, \quad y' = \frac{y}{b}$$

And so, with a simple change of variables

$$\frac{\partial^2}{\partial x^2} = \frac{1}{a^2}\frac{\partial^2}{\partial x'^2}, \quad \frac{\partial^2}{\partial y^2} = \frac{1}{b^2}\frac{\partial^2}{\partial y'^2}$$

we arrive at

$$\left[-\left(\frac{1}{a^2}\frac{\partial^2}{\partial x'^2} + \frac{1}{b^2}\frac{\partial^2}{\partial y'^2}\right) + \widetilde{V}_r(x', y')\right]\widetilde{\psi}(x', y') = \frac{2mE}{\hbar^2}\widetilde{\psi}(x', y') \tag{13}$$

where

$$\widetilde{V}_r(x', y') = \begin{cases} 0 & \text{if } |x'| < 1/2,\ |y'| < 1/2 \\ \infty & \text{elsewhere} \end{cases}$$

We note that, since the two-dimensional wave function has units of $length^{-1}$, every term has units of $length^{-3}$.

5.1.2 The Harmonic Oscillator

For the harmonic oscillator, the Schrödinger equation can be written as

$$\left[-\frac{\hbar^2}{2m}\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right) + \frac{1}{2}m\omega^2 x^2 + \frac{1}{2}m\omega^2 y^2\right]\psi(x, y) = E\psi(x, y)$$

Introducing the following quantity

$$l_0 = \sqrt{\frac{\hbar}{m\omega}}$$

and setting

$$x' = \frac{x}{l_0}, \quad y' = \frac{y}{l_0}$$

we can write

$$\left[-\frac{\hbar^2}{2m}\left(\frac{m\omega}{\hbar}\frac{\partial^2}{\partial x'^2} + \frac{m\omega}{\hbar}\frac{\partial^2}{\partial y'^2}\right) + \frac{1}{2}m\omega^2\frac{\hbar}{m\omega}x'^2 + \frac{1}{2}m\omega^2\frac{\hbar}{m\omega}y'^2\right]\widetilde{\psi}(x', y') = E\widetilde{\psi}(x', y')$$

And thus:

$$\left[-\frac{1}{2}\left(\frac{\partial^2}{\partial x'^2} + \frac{\partial^2}{\partial y'^2}\right) + \frac{1}{2}x'^2 + \frac{1}{2}y'^2\right]\widetilde{\psi}(x', y') = \frac{E}{\hbar\omega}\widetilde{\psi}(x', y') \tag{14}$$

5.1.3 The Gaussian Well

For the Gaussian well we have

$$\left[-\frac{\hbar^2}{2m}\left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}\right)-V_0 e^{-\alpha(x^2+y^2)}\right]\psi(x,y)=E\psi(x,y)$$

By setting

$$x'=\sqrt{\alpha}x, \quad y'=\sqrt{\alpha}y$$

we can write

$$\left[-\alpha\left(\frac{\partial^2}{\partial x'^2}+\frac{\partial^2}{\partial y'^2}\right)-\widetilde{V}_0 e^{-(x'^2+y'^2)}\right]\widetilde{\psi}(x',y')=\frac{2mE}{\hbar^2}\widetilde{\psi}(x',y'), \quad \widetilde{V}_0=\frac{2mV_0}{\hbar^2} \tag{15}$$

5.2 Grid selection

As mentioned previously, the grid must be chosen so that the boundary conditions (6) are satisfied. From this point forward, variables x', y' will be referred to as x, y.

For the rectangular well, the choice is straightforward. Since the potential goes to infinity at the boundaries of the well, a grid should be set over the square $[-1/2, 1/2] \times [-1/2, 1/2]$. The code will have both a and b, as well as the step size in each direction Δx, Δy, as inputs. With these values, an estimation for the number of points in each direction is made

$$N_x=\frac{1}{\Delta x}+1, \quad N_y=\frac{1}{\Delta y}+1$$

For the harmonic oscillator and the Gaussian well, a systematic way of defining a grid is needed. Both present rotational symmetry about the origin, since they are functions of the distance from the origin. So, a square centred at the origin is an appropriate selection to set up the grid.

As for the size of the grid, since the wave function of a bound state decays exponentially outside the classically allowed region, the classical turning points can be used to determine the size of the grid so that the boundary conditions are satisfied. If we are interested in bound states with energy E, it suffices that

$$V(x_{boundary}, y_{boundary}) >> E$$

So, for both the harmonic oscillator and the Gaussian well, we know that the smallest value for the potential at the boundaries is at the intersection of the boundary and the coordinate axes. So, if l is the side length of the square grid, we need

$$\frac{1}{2}\left(\frac{l}{2}\right)^2 >> E \quad \text{and} \quad -\widetilde{V}_0 e^{-\left(\frac{l}{2}\right)^2} >> E$$

for the harmonic oscillator and the Gaussian well, respectively.

For the harmonic oscillator we know the exact solution. After scaling, the energies of bound states are $E_n = n + 1$ for $n = 0, 1, 2, \ldots$, with degeneracy $n + 1$. So, for the 10 or 15 lowest-energy bound states, at most, we consider n to be around 4 or 5. So

$$\frac{l^{(HO)}}{2} \geq 6 \implies l^{(HO)} \geq 12$$

should be adequate. The code has $l^{(HO)}$, Δx and Δy as inputs.

For the Gaussian well, since we don't know the exact solution, we want the potential to be as close to zero as possible over the boundaries without choosing too big of a grid. So, a criterion was set based on the 65-95-99 rule for normal distributions, which states that approximately 65%, 95% and 99% of the population lies within one, two and three standard deviations, respectively, of the mean. As a refresher, the probability density function of a normal distribution with mean equal to zero and standard deviation σ is

$$f(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{z^2}{2\sigma^2}}$$

Thus, we can set a parameter ς such that

$$\frac{1}{2\varsigma^2} = 1 \implies \varsigma = \frac{1}{\sqrt{2}}$$

and argue that an acceptable value for $l^{(GW)}/2$ would be around 5ς or 6ς. For reference, for a normal distribution, since the expected percentage of the population within $p\sigma$ of the mean is given by $\text{erf}(p/\sqrt{2})$, we can estimate that 99.9999% and 99.999999% of the population lies within five and six standard deviations, respectively. Taking

$$\frac{l^{(GW)}}{2} \approx \frac{5}{\sqrt{2}}, \frac{6}{\sqrt{2}} \implies l^{(GW)} \approx 8$$

ensures that the exponential part of the potential is sufficiently small. The code has $\widetilde{V_0}$, $l^{(GW)}$, Δx and Δy as inputs.

If $\widetilde{V_0}$ were to be exceedingly large, the potential would not be small at the boundaries. But, this would mean the potential would now admit a large number of bound states, and we are only interested in the first few.

For both potentials, the number of points in each direction is given by

$$N_x = \frac{l}{\Delta x} + 1, \quad N_y = \frac{l}{\Delta y} + 1, \quad l = l^{(HO)}, l^{(GW)}$$

And so, appropriate grid arrays in each direction are set up.

5.3 Diagonalization

5.3.1 Set up

Given that, potentially, N_x and N_y could be numbers that would cause arrays to have a big number of entries, the code will not store in memory any arrays whose entries can be computed once they are necessary. In particular, neither the Hamiltonian matrix nor the potential array over the grid will be stored as a whole.

Since the Lanczos algorithm only needs a way to compute the product of the Hamiltonian matrix H with an arbitrary vector and we know that

$$[H\boldsymbol{\psi}]_k = -\mu\boldsymbol{\psi}_{k-(N_x-2)} - \eta\boldsymbol{\psi}_{k-1} + \nu_{i,j}\boldsymbol{\psi}_k - \eta\boldsymbol{\psi}_{k+1} - \mu\boldsymbol{\psi}_{k+(N_x-2)}, \quad k = 0, 1, \ldots, (N_x-2)(N_y-2)-1.$$

Potential	η	μ	$\nu_{i,j}$
Rectangular well	$\dfrac{1}{a^2(\Delta x)^2}$	$\dfrac{1}{b^2(\Delta y)^2}$	$2(\eta+\mu)$
Harmonic oscillator	$\dfrac{1}{2(\Delta x)^2}$	$\dfrac{1}{2(\Delta y)^2}$	$2(\eta+\mu)+\dfrac{1}{2}(x_i^2+y_j^2)$
Gaussian well	$\dfrac{\alpha}{(\Delta x)^2}$	$\dfrac{\alpha}{(\Delta y)^2}$	$2(\eta+\mu)-\widetilde{V}_0 e^{-(x_i^2+y_j^2)}$

Table 1: Parameters relating to the Hamiltonian matrix for each potential.

we could compute both said product and the elements $\nu_{i,j}$ on the fly.

The parameters η, μ, and $\nu_{i,j}$ will depend upon the potential and their explicit value can be seen in Table 1.

An attempt was made to implement the Lanczos algorithm by computing the product of H and an arbitrary vector instead of storing the Hamiltonian matrix. However, convergence was never achieved. After multiple revisions, the reason for convergence failure was not found.

Thus, we were forced to somehow store the Hamiltonian matrix. This was done by using *SciPy*'s `sparse` package. This package allows to somewhat store sparse matrices by specifying where the non-zero values are located. In the script this was done by setting arrays for the non-zero diagonals of the Hamiltonian matrix.

5.3.2 Lanczos algorithm - Misconvergence[3][4][5]

The Lanczos algorithm as detailed in the previous section has two distinct but somewhat related misconvergence problems.

The first one is related to the choice of the initial Lanczos vector q_1 to start the iteration. If q_1 is nearly orthogonal to an eigenvector ψ_λ of the Hamiltonian matrix H, the algorithm will not converge to it's corresponding eigenvalue λ, that is, none of the eigenvalues of the tridiagonal matrix T_k will approximate λ, until one of the Lanczos vectors q_k generated by the algorithm has a significant component in the direction of ψ_λ. If q_1 were to be exactly orthogonal to ψ_λ, none of the eigenvalues of T_k will ever converge to λ, no approximation is computed, since all subsequent Lanczos vectors q_i will also be orthogonal to ψ_λ.

The most straightforward solution to this issue is to choose q_1 at random, via some random number generator function. In that case, it's unlikely that q_1 will be orthogonal to any of the eigenvectors of the Hamiltonian matrix. If the q_1 is randomly generated each time the iteration is run, more than one run can be done to verify that no eigenvalues are missing.

The second problem is that it could happen that multiple eigenvalues of T_k approximate an eigenvalue of the Hamiltonian matrix H. This is a consequence of the loss of orthogonality between the Lanczos vectors, and thus, if accurate multiplicities are important, the Lanczos vectors need to be kept orthogonal. It's worth noting that a symmetric tridiagonal matrix whose off-diagonal elements are positive will always have distinct eigenvalues. So, the Lanczos algorithm will never find eigenvalues of the Hamiltonian with the correct multiplicity. This has it's benefits, however, as spotting identical eigenvalues would mean that

that particular eigenvalue has already converged.

The most straightforward, and computationally expensive, solution is full re-orthogonalization; re-orthogonalize every new generated Lanczos vector to all previously generated Lanczos vectors. This could be done via the Gram-Schmidt algorithm, either up to a tolerance value or via multiple applications of the algorithm. A less expensive approach would be local re-orthogonalization; re-orthogonalize every new generated Lanczos vector to a fixed number of previously generated Lanczos vectors.

A third approach, selective orthogonalization, can be used by taking advantage of the fact that the Lanczos vectors q_i lose their orthogonality in a systematic way.

5.3.3 Lanczos algorithm - Selective Orthogonalization[3][4]

To see how the Lanczos vectors lose their orthogonality, we will take a look at how the Lanczos vectors relate the Hamiltonian matrix to the T_k matrix and their eigenvalues.

So, if H is the matrix whose eigenvalues we are interested in, after k iterations, the Lanczos algorithm generates a set of vectors q_i such that

$$T_k = Q_k^T H Q_k, \quad Q_k = [q_1, q_2, \ldots, q_k]$$

where T_k is a symmetric tridiagonal matrix whose eigenvalues approximate the eigenvalues of the matrix H. Now, T_k has an eigendecomposition of the form

$$T_k = V_k \Lambda_k V_k^T, \quad V_k = [v_1^{(k)}, v_2^{(k)}, \ldots, v_k^{(k)}], \quad \Lambda_k = \mathrm{diag}(\theta_1^{(k)}, \theta_2^{(k)}, \ldots, \theta_k^{(k)})$$

where $v_i^{(k)}$ are the eigenvectors of T_k and $\theta_i^{(k)}$ it's eigenvalues and V_k is an orthogonal matrix. These eigenvalues approximate the eigenvalues of the matrix H and are known as Ritz values. Also, the vectors that approximate the eigenvectors of H, known as Ritz vectors, can be found by using both expressions for T_k:

$$T_k = V_k \Lambda_k V_k^T = Q_k^T H Q_k \implies \Lambda_k = V_k^T Q_k^T H Q_k V_k$$

So, the Ritz vectors $y_i^{(k)}$ are the columns of the matrix $Q_k V_k$:

$$Q_k V_k = [y_1^{(k)}, y_2^{(k)}, \ldots, y_k^{(k)}], \quad y_i^{(k)} = Q_k v_i^{(k)}$$

With the concept of Ritz vectors and Ritz values we can now appropriately assert that Lanczos vectors start to lose orthogonality as soon as a Ritz value is close to convergence, causing the subsequent Lanczos vectors to develop a component in the direction of the corresponding Ritz vector, which leads to multiple copies of converged Ritz values.

To assess convergence, it can be shown that, when using the Lanczos algorithm, the error bound is given by:

$$\left\| H y_i^{(k)} - \theta_i^{(k)} y_i^{(k)} \right\| = \beta_{k+1} \left| [v_i^{(k)}]_k \right|$$

where $[v_i^{(k)}]_k$ is the k-th entry of $v_i^{(k)}$.

Thus, by monitoring the error bounds, it can be predicted which q_i will have large components in the direction of a specific Ritz vector, which allows to selectively orthogonalize q_i

against those Ritz vectors. This keeps the Lanczos vectors nearly orthogonal for very little extra computational work.

To do this, the following result will be used, which relates the component of the computed Lanczos vector q_{k+1} in the direction of a particular Ritz vector $y_i^{(k)}$:

$$y_i^{(k)} \cdot q_{k+1} = \frac{\mathcal{O}(\varepsilon_M \|H\|)}{\beta_{k+1}\left|[v_i^{(k)}]_k\right|}$$

where ε_M is the machine epsilon[1].

So, when a Ritz value $\theta_i^{(k)}$ converges, it's error bound $\beta_{k+1}\left|[v_i^{(k)}]_k\right|$ goes to zero and the Lanczos vector q_{k+1} acquires a large component in the direction of the corresponding Ritz vector $y_i^{(k)}$. Thus, when the error bound is small enough, orthogonalization against $y_i^{(k)}$ is used. It's enough to check for when

$$\beta_{k+1}\left|[v_i^{(k)}]_k\right| \leq \sqrt{\varepsilon_M}\|H\| \approx \sqrt{\varepsilon_M}\|T_k\|.$$

where $\|T_k\| = \max_i \left|\theta_i^{(k)}\right|$, the absolutely largest eigenvalue, is used since $\|H\|$ is not known and we know the fastest converging eigenvalues are towards the extreme of the spectrum.

As for ε_M, the simplest way to calculate it is by running the following calculation

$$\varepsilon_M = \frac{7}{3} - \frac{4}{3} - 1$$

Another way, is to use *NumPy*'s `finfo` function. The default numerical data type is float64, 64-bit precision floating-point number type: sign bit, 11 bits exponent, 52 bits mantissa. And thus:

$$\varepsilon_M = 2^{-52} \approx 2.22 \times 10^{-16}$$

6 Results and Discussion

The first thing we wanted to analyse was the direct matrix method itself. This was done using known results for the harmonic oscillator and analysing how the results change as inputs change. Then, using the infinite rectangular well, the Lanczos method was analysed. Once the overall behaviour of both methods was understood, the Gaussian well was analysed. Unless explicitly stated otherwise, the Lanczos method without selective orthogonalization was used. Finally, some comparisons were drawn between the Lanczos method with selective orthogonalization and without selective orthogonalization.

So, first, we took the harmonic oscillator, since the code requires the least number of inputs, namely l, Δx and Δy. We fixed $\Delta x = \Delta y = 0.1$ and varied l to check which values of l give an acceptable estimate for the first 5 energy levels. Results can be seen in Table 2. For low values of l it was noticed that degeneracy breaks for the lower levels, so via conventional diagonalization it was checked which values correspond to which level. If this

[1]Machine epsilon is the smallest value ε_M such that $1 + \varepsilon_M > 1$.

	ε_0	ε_1	ε_2	ε_3	ε_4	$\dim(H)$
$l = 4$	1.07432	2.29909	3.923747*	6.08097*	8.93039*	1521×1521
$l = 6$	1.00016	2.00462	3.03681*	4.13955*	5.43735*	3481×3481
$l = 8$	0.99937	1.99813	2.99582	3.99390	4.99514*	6241×6241
$l = 10$	0.99937	1.99812	2.99561	3.99245	4.98802*	9801×9801
$l = 12$	0.99937	1.99812	2.99561	3.99185	4.98683*	14161×14161
$l = 14$	0.99937	1.99812	2.99561	3.99185	4.98683*	19321×19321
$l = 16$	0.99937	1.99812	2.99561	3.99185	4.98683*	25281×25281
Exact	1	2	3	4	5	-

Table 2: Eigenvalues for different values of l. Harmonic Oscillator, $\Delta x = \Delta y = 0.1$ and $\hbar\omega\varepsilon_n = E_n$. Split levels are marked with $\star$.

	ε_0	ε_1	ε_2	ε_3	ε_4	$\dim(H)$
$\Delta x = \Delta y = 0.5$	0.98411	1.95180*	2.88565*	3.78357*	4.64292*	529×529
$\Delta x = \Delta y = 0.25$	0.99607	1.98820	2.97237*	3.94848*	4.91644*	2209×2209
$\Delta x = \Delta y = 0.1$	0.99937	1.99812	2.99561	3.99185	4.98683*	14161×14161
$\Delta x = \Delta y = 0.05$	0.99984	1.99953	2.99890	3.99796	4.99765	57121×57121
$\Delta x = \Delta y = 0.01$	0.99999	1.99998	2.99995	3.99993	4.99988	1437601×1437601
$\Delta x = 0.5$, $\Delta y = 0.01$	0.99174	1.95942*	2.89328*	3.79120*	4.65055*	2737×2737
$\Delta x = 0.25$, $\Delta y = 0.05$	0.99796	1.99008	2.97425*	3.95037*	4.91833*	11233×11233
$\Delta x = 0.1$, $\Delta y = 0.01$	0.99968	1.99843	2.99592	3.99216	4.98714*	142681×142681
Exact	1	2	3	4	5	-

Table 3: Eigenvalues for different values of Δx and Δy. Harmonic Oscillator, $l = 12$ and $\hbar\omega\varepsilon_n = E_n$. Split levels are marked with $\star$.

happened for a level, it's denoted and the highest value was included. This was done only for split levels that the Lanczos algorithm treated as split but if the difference was on the fourth significant figure it was treated as being the same value. Via conventional diagonalization it was checked that sometimes a level was split with near values, but the Lanczos algorithm approximated it by a single eigenvalue whose value was in-between these values.

It can be seen that increasing the value of l affects the converged values up to a certain point. By taking bigger values of l, higher levels stabilise but the difference between the converged values and the exact values keeps increasing. This is because a bigger grid ensures the boundary conditions are satisfied for higher energy states but for their wave function to be appropriately sampled, a smaller step size is needed.

Next, the value of l was fixed at 12, and for a few values of the step size Δx and Δy, the converged values are shown in Table 3. Same as before, some levels were split, but in this case the lowest value was included.

First thing to notice is that reducing the step size gives a better approximation each time. Again, this is because by reducing the step size, the wave function is better approximated by discretizing. Also, if the step sizes are different, the bigger step size dominates. While reducing only one of the step sizes gives a slightly better approximation, it's not signifi-

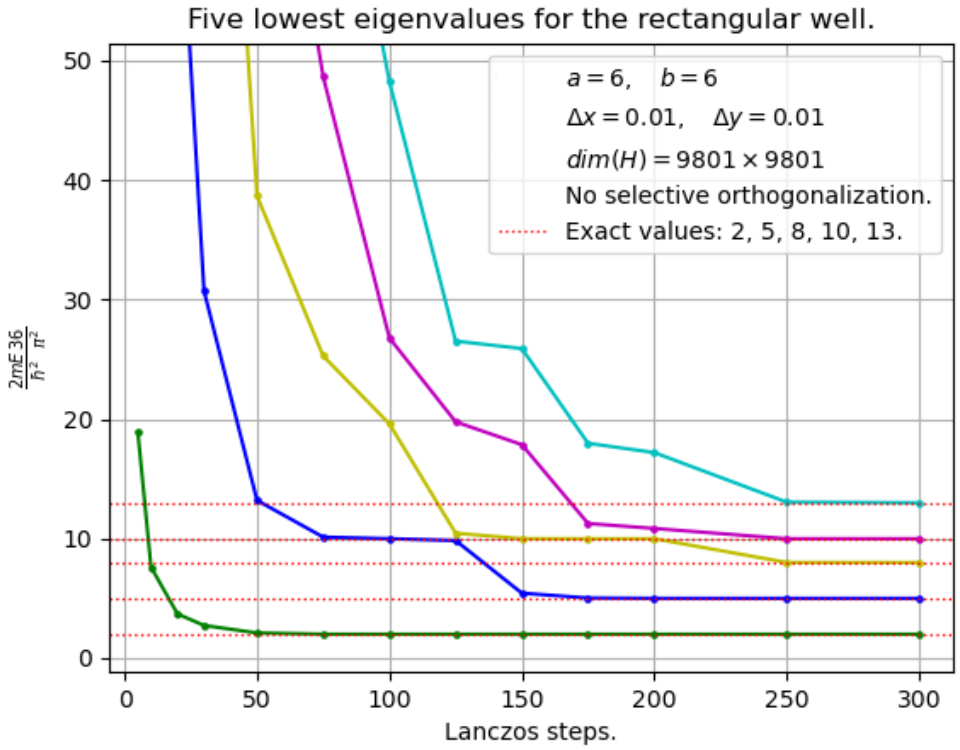

Figure 2: Five lowest eigenvalues of the infinite rectangular well as a function of the number of iterations for $a = b = 6$ and $\Delta x = \Delta y = 0.01$.

cantly better. This corroborates the claim that the error in the approximation is of order $\max[(\Delta x)^2, (\Delta y)^2]$

One thing that was quickly noticed is that the dimensions of the Hamiltonian quickly get quite large, which causes the Lanczos algorithm to achieve convergence for a larger number of iterations. This was also observed for the rectangular and Gaussian wells.

For the infinite rectangular well, the size of the grid is fixed, but the code requires a, b, Δx, Δy as inputs. First, we set $a = b = 6$ and $\Delta x = \Delta y = 0.01$ and eigenvalues as a function of number of iterations was plotted in Figure 2. As expected, lower eigenvalues converge first and at around 300 iterations all five values converged. It's also worth noticing that it seems that eigenvalues tend to mistakenly converge to higher values, before being knocked off to a lower value. This can be seen for the second and third values.

To compare between the dimensions of the Hamiltonian matrix and the number of iterations needed, $a = b = 6$ was taken and, by taking different values of $\Delta x = \Delta y$, how many iterations were needed for convergence. Results are shown in Table 4. While the number of iterations certainly grows, it doesn't grow as fast as the dimensions of the matrix. In fact, as a percentage of the dimension of the total vector space, it goes down. It should be noted, however, that each iteration is more costly as the size of the matrix grows. So, even though a bigger matrix may yield more accurate results, the gained precision may not be worth it.

Next we wanted to see whether taking different step sizes would be advantageous in some

$\dim(H)$	Iterations	ε_0	ε_1	ε_2	ε_3	ε_4
361×361	~ 60	1.99589	4.96515	7.93441	9.83262	12.80188
1521×1521	~ 120	1.99897	4.99126	7.98356	9.95792	12.95022
9801×9801	~ 275	1.99983	4.99860	7.99736	9.99325	12.99202
39601×39601	~ 625	1.99995	4.99965	7.99934	9.99831	12.99805
249001×249001	~ 1600	1.99999	4.99994	7.99989	9.99973	12.99972
Exact value	-	2	5	8	10	13

Table 4: Five lowest eigenvalues of the rectangular well, $a = b = 6$, for different values of $\Delta x = \Delta y$ where $(\pi^2/36)\varepsilon_n = 2mE_n/\hbar^2$. Comparison between $\dim(H)$ and iterations until convergence.

	$\dim(H)$	ε_0	ε_1	ε_2	ε_3	ε_4
$\Delta x = 0.01$, $\Delta y = 0.01$	9801×9801	1.29793	1.90194	2.90797	4.31502	4.58645
$\Delta x = 0.005$, $\Delta y = 0.005$	39601×39601	1.29801	1.90221	2.90904	4.31826	4.58754
$\Delta x = 0.01$, $\Delta y = 0.005$	19701×19701	1.29800	1.90201	2.90804	4.31508	4.58753
$\Delta x = 0.005$, $\Delta y = 0.01$	19701×19701	1.29794	1.90214	2.90898	4.31820	4.58646
Exact value	-	1.298043	1.902305	2.909407	4.319351	4.587911

Table 5: Five lowest eigenvalues of the rectangular well, $a = 7$, $b = 3$, for different values of Δx and Δy where $\varepsilon_n = 2mE_n/\hbar^2$. Comparison between $\dim(H)$ and results.

situations. As mentioned previously, for the harmonic oscillator there was no need to take different step sizes but for a rectangular potential, taking different step sizes may provide similar results with a smaller matrix. For this purpose, we took $a = 7$ and $b = 3$ and tried some step sizes combinations. Results are shown in Table 5. It can be seen that for ε_0 and ε_4 better results are obtained by taking a smaller y-step, while for the other three better results are obtained by taking a smaller x-step. This stems from the fact that, while ε_0 corresponds to the pair $(n_x, n_y) = (1, 1)$, ε_1, ε_2 and ε_3 correspond to $(2, 1)$, $(3, 1)$ and $(4, 1)$, respectively and ε_4 corresponds to $(1, 2)$. A bigger n implies a higher number of nodes for the wave function, and so, a smaller step size will give a better sampling in that direction.

With all of this in mind, we took the Gaussian well. As mentioned in a previous section, this potential doesn't admit an analytic solution to compare to. So, by taking appropriate values for l and $\Delta x = \Delta y$, we got it's eigenvalues for different values of $\widetilde{V}_0$ and α, which can be seen in Table 6. First thing to note is that since the Lanczos algorithm does not provide accurate multiplicities, we can only state how many bound state energy levels the Gaussian well admits, without stating their degeneracy. It can be seen that bigger values of $\widetilde{V}_0$ and smaller values of α favour more bound state energy levels. While one may be inclined to reduce the value of l for bigger values of α, given the shape of the well, it should be treated with care. Bound states whose energy is near zero require a bigger grid so that boundary conditions are satisfied, which, as seen in Table 6, is not a rare occurrence.

Finally, selective orthogonalization was used to assess it's utility. It was found that it had no significant impact in convergence. The number of iterations needed for convergence was slightly lower, but not worth it. Since storage of all Lanczos vectors is needed and

	$\Delta x = \Delta y$	Levels	ε_0	ε_1	ε_2	ε_3	ε_4
$\widetilde{V_0} = 2$, $\alpha = 1$	0.05	1	-0.19783	-	-	-	-
$\widetilde{V_0} = 5$, $\alpha = 1$	0.05	1	-1.58103	-	-	-	-
$\widetilde{V_0} = 7$, $\alpha = 1$	0.05	1	-2.75629	-	-	-	-
$\widetilde{V_0} = 10$, $\alpha = 1$	0.05	2	-4.71721	-0.78366	-	-	-
$\widetilde{V_0} = 15$, $\alpha = 1$	0.05	3	-8.28998	-2.82822	-0.17451	-	-
$\widetilde{V_0} = 7$, $\alpha = 0.2$	0.05	5	-4.83950	-2.90684	-1.45034	-1.24039	-0.34319
$\widetilde{V_0} = 7$, $\alpha = 0.5$	0.05	3	-3.77675	-1.18459	-0.01880	-	-
$\widetilde{V_0} = 7$, $\alpha = 1$	0.05	1	-2.75629	-	-	-	-
$\widetilde{V_0} = 7$, $\alpha = 2$	0.03	1	-1.57426	-	-	-	-
$\widetilde{V_0} = 7$, $\alpha = 5$	0.05	1	-0.38678	-	-	-	-

Table 6: Gaussian well eigenvalues for different values of $\widetilde{V_0}$ and α.

diagonalization of T_k occurs in every iteration, it gets slow quite fast. It's faster and less costly to do a large number of iterations without selective orthogonalization and check for multiple eigenvalues.

Overall, the direct matrix method with Lanczos algorithm for diagonalization works fine for lower energy levels. It works fast enough for the first few levels with acceptable precision. The biggest issue stems from the direct matrix method. The way it's formulated inevitably leads to absurd sizes for the Hamiltonian matrix as a product of the relationship between it's size and the grid. This ultimately undermines the Lanczos algorithm.

A good first step towards mitigating this would be to take a better approximation for the second-order derivatives. The fourth-order approximation is given by

$$f''(x) = \frac{-f(x+2\Delta x) + 16f(x+\Delta x) - 30f(x) + 16f(x-\Delta x) - f(x-2\Delta x)}{12(\Delta x)^2} + \mathcal{O}((\Delta x)^4)$$

This would give a somewhat less sparse but smaller matrix with the advantage that the error is now of order four in the step size and a smaller number of iterations in the Lanczos algorithm would be needed to achieve convergence.

As for the Lanczos algorithm, one possible improvement could be to implement a variation that yields accurate multiplicities, such as the block Lanczos method or the implicitly restarted Lanczos method. The block Lanczos method, instead of yielding a tridiagonal matrix, yields a block tridiagonal matrix, which can have multiple eigenvalues. The implicitly restarted Lanczos method uses the fact that all quantities are a function of the initial vector q_1 to apply the method multiple times, each time with a better guess for that initial vector.

References

[1] K. Chadan, N.N. Khuri, A. Martin and T.T. Wu, "*Bound States in one and two Spatial Dimensions*", J. Math. Phys. 44 (2003) 406

[2] K. Yang and M. de Llano, "*Simple variational proof that any two-dimensional potential well supports at least one bound state*", Am. J. Phys. 57 (1989) 85.

[3] J. W. Demmel, (2007) "*Applied Numerical Linear Algebra*". Society for Industrial and Applied Mathematics. Philadelphia, USA.

[4] V. Hernández, J. E. Román, A. Tomás and V. Vidal. "*STR-5: Lanczos Methods in SLEPc*", [Online]. Available: `https://slepc.upv.es/documentation` [October 2006]. Universitat Politècnica de València, Valencia, Spain.

[5] J. Izaac and J. Wang, (2018) "*Computational Quantum Mechanics*". Springer. Cham, Switzerland.

[6] P. Giannozzi, (2019) "*Numerical Methods in Quantum Mechanics*". Lecture notes. University of Udine, Italy. Academic year 2018/2019.

[7] J. M. Thijssen, (2007) "*Computational Physics*". 2nd ed. Cambridge University Press. Cambridge, England.

A Two-Dimensional Direct Matrix Method - The Hamiltonian Matrix step by step for $N = 5$

For $N = 5$, boundary conditions are

$$\psi_{0,j} = \psi_{4,j} = \psi_{i,0} = \psi_{i,4} = 0$$

and

$$N - 2 = 3 \implies (N-2)^2 = 9$$

This means that the Hamiltonian will be represented by a 9×9 matrix while the wave function will be represented by a 9-dimensional vector $\boldsymbol{\psi}$ whose components satisfy

$$-\mu\boldsymbol{\psi}_{k-3} - \eta\boldsymbol{\psi}_{k-1} + \nu_{i,j}\boldsymbol{\psi}_k - \eta\boldsymbol{\psi}_{k+1} - \mu\boldsymbol{\psi}_{k+3} = E\boldsymbol{\psi}_k, \quad k = 0, 1, \ldots, 8.$$

where $i, j = 1, 2, 3$.

Thus, this system of equations is as follows:

$$\begin{cases}
k=0 & \to & \varnothing & \varnothing & +\nu_{1,1}\boldsymbol{\psi}_0 & -\eta\boldsymbol{\psi}_1 & -\mu\boldsymbol{\psi}_3 & = & E\boldsymbol{\psi}_0 \\
k=1 & \to & \varnothing & -\eta\boldsymbol{\psi}_0 & +\nu_{2,1}\boldsymbol{\psi}_1 & -\eta\boldsymbol{\psi}_2 & -\mu\boldsymbol{\psi}_4 & = & E\boldsymbol{\psi}_1 \\
k=2 & \to & \varnothing & -\eta\boldsymbol{\psi}_1 & +\nu_{3,1}\boldsymbol{\psi}_2 & +0 & -\mu\boldsymbol{\psi}_5 & = & E\boldsymbol{\psi}_2 \\
k=3 & \to & -\mu\boldsymbol{\psi}_0 & +0 & +\nu_{1,2}\boldsymbol{\psi}_3 & -\eta\boldsymbol{\psi}_4 & -\mu\boldsymbol{\psi}_6 & = & E\boldsymbol{\psi}_3 \\
k=4 & \to & -\mu\boldsymbol{\psi}_1 & -\eta\boldsymbol{\psi}_3 & +\nu_{2,2}\boldsymbol{\psi}_4 & -\eta\boldsymbol{\psi}_5 & -\mu\boldsymbol{\psi}_7 & = & E\boldsymbol{\psi}_4 \\
k=5 & \to & -\mu\boldsymbol{\psi}_2 & -\eta\boldsymbol{\psi}_4 & +\nu_{3,2}\boldsymbol{\psi}_5 & +0 & -\mu\boldsymbol{\psi}_8 & = & E\boldsymbol{\psi}_5 \\
k=6 & \to & -\mu\boldsymbol{\psi}_3 & +0 & +\nu_{1,3}\boldsymbol{\psi}_6 & -\eta\boldsymbol{\psi}_7 & \varnothing & = & E\boldsymbol{\psi}_6 \\
k=7 & \to & -\mu\boldsymbol{\psi}_4 & -\eta\boldsymbol{\psi}_6 & +\nu_{2,3}\boldsymbol{\psi}_7 & -\eta\boldsymbol{\psi}_8 & \varnothing & = & E\boldsymbol{\psi}_7 \\
k=8 & \to & -\mu\boldsymbol{\psi}_5 & -\eta\boldsymbol{\psi}_7 & +\nu_{2,3}\boldsymbol{\psi}_8 & \varnothing & \varnothing & = & E\boldsymbol{\psi}_8
\end{cases}$$

where we have explicitly included terms that are null; $\varnothing$ denotes values $\boldsymbol{\psi}_m$ of the wave function vector where $m \notin \{0, 1, \ldots, 8\}$, while 0 denotes values of the wave function vector that are zero by boundary conditions and should be ignored by the k-indexing scheme.

Writing this system of equations as a matrix eigenvalue problem of the form

$$H\boldsymbol{\psi} = E\boldsymbol{\psi}$$

we can see that

$$H = \left[\begin{array}{ccc:ccc:ccc}
\nu_{1,1} & -\eta & 0 & -\mu & 0 & 0 & 0 & 0 & 0 \\
-\eta & \nu_{2,1} & -\eta & 0 & -\mu & 0 & 0 & 0 & 0 \\
0 & -\eta & \nu_{3,1} & 0 & 0 & -\mu & 0 & 0 & 0 \\
\hdashline
-\mu & 0 & 0 & \nu_{1,2} & -\eta & 0 & -\mu & 0 & 0 \\
0 & -\mu & 0 & -\eta & \nu_{2,2} & -\eta & 0 & -\mu & 0 \\
0 & 0 & -\mu & 0 & -\eta & \nu_{3,2} & 0 & 0 & -\mu \\
\hdashline
0 & 0 & 0 & -\mu & 0 & 0 & \nu_{1,3} & -\eta & 0 \\
0 & 0 & 0 & 0 & -\mu & 0 & -\eta & \nu_{2,3} & -\eta \\
0 & 0 & 0 & 0 & 0 & -\mu & 0 & -\eta & \nu_{3,3}
\end{array}\right]$$

With the aid of the dashed lines, we can see that H can be easily described as a block-tridiagonal matrix. If we denote the 3×3 identity matrix by $\mathbb{I}_3$, we can write

$$H = \begin{bmatrix} \mathbb{V}_1 & -\mu\mathbb{I}_3 & 0 \\ -\mu\mathbb{I}_3 & \mathbb{V}_2 & -\mu\mathbb{I}_3 \\ 0 & -\mu\mathbb{I}_3 & \mathbb{V}_3 \end{bmatrix}, \quad \text{where,} \quad \mathbb{V}_j = \begin{bmatrix} \nu_{1,j} & -\eta & 0 \\ -\eta & \nu_{2,j} & -\eta \\ 0 & -\eta & \nu_{3,j} \end{bmatrix}$$