

Using Jacobi's Rotation Algorithm to Solve the Two Electron Schrödinger Equation

Kosar Nozari Mirarkolaei

September 2018

1 Abstract

Abstract

In this project, I have implemented the Jacobi rotation algorithm to solve eigenvalue problem for one and two electrons in the Schrödinger equation. I have found that the eigenvalues for the four lowest states to a very high precision. In the one electron case, they were found to be $\lambda_1 = 2.898$, $\lambda_2 = 6.8446$, $\lambda_3 = 10.8016$ and $\lambda_4 = 14.7629$ compared to the analytical 3, 7, 11 and 15 respectively.

There is also made a discussion of the speed of the algorithm compared to other methods, and in relation, pros and cons connected with the Jacobi method compared to one other method. The main idea here is that the Jacobi method is intrinsically slow, but easy to parallelize.

The source code for this project can be found at my github¹.

2 Introduction

Eigenvalue problems are of great importance in the modern sciences. In this project, I will solve the radial part of the Schrödinger equation for one and two electrons in a harmonic oscillator potential with and without repulsive Coulomb interaction using the Jacobi's rotation method for finding eigenvalues.

3 Theory

3.1 The Buckling Beam Problem

From the bucking beam problem we have the following equation:

$$\gamma \frac{d^2 u(x)}{dx^2} = -Fu(x)$$

¹<https://github.com/Kosarnm/FYS4150>

We also have the Dirichlet boundary condition which is $u(0) = 0$ and $u(L) = 0$ for $x \in [0, L]$.

In mechanical view L is the beam length and F is the force that apply at the $(L, 0)$ toward the origin and γ is a constant that define by properties like rigidity of the beam.

Now we change the scale of the equation by defining dimensional variable ρ :

$$\rho = \frac{x}{L}$$

Now by reordering the equation in $\rho \in [0, 1]$ interval we have the following equation:

$$\frac{d^2 u(\rho)}{d\rho^2} = -\frac{FL^2}{R}u(\rho) = -\lambda u(\rho)$$

With discretizing the equation the problem change to the eigenvalue problem. Now we can use the expression for second derivative :

$$u'' = \frac{u(\rho + h) - 2u(\rho) + u(\rho - h)}{h^2} + O(h^2), \quad (1)$$

In the above equation h is our step. And we also should define the maximum and minimum value for the variable ρ , ρ_{min} and ρ_{max} . We consider $\rho_{min} = 0$ and $\rho_{max} = 1$. Now by considering $\rho_{min} = \rho_0$ and $\rho_{max} = \rho_N$ and with given number of mesh point N we can define step length as:

$$h = \frac{\rho_N - \rho_0}{N}$$

And we can calculate ρ at the different point of i by using the following relation:

$$\rho_i = \rho_0 + ih \quad i = 1, 2, \dots, N$$

Then we rewrite the differential equation with ρ_i :

$$-\frac{u(\rho_i + h) - 2u(\rho_i) + u(\rho_i - h)}{h^2} = \lambda u(\rho_i)$$

This equation also can be written as:

$$-\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} = \lambda u_i$$

Now we can write this equation in the form of tridiagonal matrix for different value of i :

$$\begin{pmatrix} d & a & 0 & 0 & \dots & 0 & 0 \\ a & d & a & 0 & \dots & 0 & 0 \\ 0 & a & d & a & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots & a & d & a \\ 0 & \dots & \dots & \dots & \dots & a & d \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ \dots \\ u_{N-2} \\ u_{N-1} \end{pmatrix} = \lambda \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ \dots \\ u_{N-2} \\ u_{N-1} \end{pmatrix} \quad (2)$$

As we can see the boundary condition u_0 and u_N are not included in the matrix. Now we have the diagonal element $d = \frac{-2}{h^2}$ and non diagonal elements $a = \frac{1}{h^2}$ For finding analytical eigenpair for eigenvalue problem we can use following equation:

$$\lambda_j = d + 2a \cos\left(\frac{j\pi}{N+1}\right) \quad j = 1, 2, \dots, N-1 \quad (3)$$

3.2 Schrödinger equation for one electron

Following equation shows the radial part of Schrödinger equation for one electron. We start with solving this equation:

$$-\frac{\hbar^2}{2m} \left(\frac{1}{r^2} \frac{d}{dr} r^2 - \frac{l(l+1)}{r^2} \right) R(r) + V(r)R(r) = ER(r) \quad (4)$$

In above equation $V(r) = \frac{1}{2}kr^2$ is harmonic oscillator potential and $k = m\omega^2$ and E is the energy of harmonic oscillator in three dimension. ω is the oscillator frequency. Energies are calculated by the following equation for $n = 0, 1, 2, \dots$ and $l = 0, 1, 2, \dots$ n and l are energy quantum number and orbital momentum quantum number respectively.

$$E_{nl} = \hbar\omega(2n + l + \frac{3}{2})$$

Then we assume that $R(r) = (1/r)u(r)$ and rewrite the equation 4:

$$-\frac{\hbar^2}{2m} \frac{d^2}{dr^2} u(r) + \left(V(r) + \frac{l(l+1)}{r^2} - \frac{\hbar^2}{2m} \right) u(r) = Eu(r) \quad (5)$$

The boundary condition for this equation are $u(0) = 0$ and $u(\infty) = 0$. Now we define a dimensionless variable $\rho = (1/\alpha)r$ where α is a constant and then rewrite the equation:

$$-\frac{\hbar^2}{2m\alpha^2} \frac{d^2}{d\rho^2} u(\rho) + \left(V(\rho) + \frac{l(l+1)}{\rho^2} - \frac{\hbar^2}{2m\alpha^2} \right) u(\rho) = Eu(\rho) \quad (6)$$

In this project for easier calculation we set $l = 0$ and substitute $V(\rho) = \frac{1}{2}k\alpha^2\rho^2$ in equation (6):

$$-\frac{\hbar^2}{2m\alpha^2} \frac{d^2}{d\rho^2} u(\rho) + \frac{1}{2}k\alpha^2\rho^2 u(\rho) = Eu(\rho) \quad (7)$$

Now we multiply the both side of equation(7) by $2m\alpha^2\rho^2/\hbar^2$ and define some constant as below:

$$\frac{mk}{\hbar^2} \alpha^4 = 1 \quad \lambda = \frac{2m\alpha^2}{\hbar^2} E$$

Then we can rewrite the Schrödinger equation as:

$$-\frac{d^2}{d\rho^2}u(\rho) + \rho^2u(\rho) = \lambda u(\rho) \quad (8)$$

Just like the previous section we can solve this equation by discretizing the equation. First we define the minimum and maximum value for the ρ , $\rho_{min} = 0$ and ρ_{max} . Since we cannot set $\rho_{max} = \infty$ we can solve the problem for different value of ρ_{max} . By setting $\rho_{min} = \rho_0$ and $\rho_{max} = \rho_N$ and having the given number of mesh point N , the step length can define as:

$$h = \frac{\rho_N - \rho_0}{N}$$

The value of ρ for different value of i :

$$\rho_i = \rho_0 + ih \quad i = 1, 2, \dots, N$$

Now we can rewrite the Schrödinger equation for value of ρ_i :

$$-\frac{u(\rho_i + h) - 2u(\rho_i) + u(\rho_i - h)}{h^2} + \rho_i^2 u(\rho_i) = \lambda u(\rho_i)$$

Or it can rewrite in more simple way:

$$-\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + \rho_i^2 u_i = -\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + V_i u_i = \lambda u_i$$

In the above equation $V_i = \rho_i^2$ is the harmonic oscillator potential.

We define the diagonal and non-diagonal matrix element as $d_i = \frac{2}{h^2} + V_i$ and $e_i = \frac{1}{h^2}$ respectively. Now by using these simplification the Schrödinger equation can be written as:

$$d_i u_i + e_{i-1} u_{i-1} + e_{i+1} u_{i+1} = \lambda u_i$$

Now we can write this equation as matrix eigenvalue problem:

$$\begin{pmatrix} d_0 & e_0 & 0 & 0 & \dots & 0 & 0 \\ e_1 & d_1 & e_1 & 0 & \dots & 0 & 0 \\ 0 & e_2 & d_2 & e_2 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots & e_{N-1} & d_{N-1} & e_{N-1} \\ 0 & \dots & \dots & \dots & \dots & d_N & \dots \end{pmatrix} \begin{pmatrix} u_0 \\ u_2 \\ u_3 \\ \dots \\ u_{N-1} \\ u_N \end{pmatrix} = \lambda \begin{pmatrix} u_0 \\ u_2 \\ u_3 \\ \dots \\ u_{N-1} \\ u_N \end{pmatrix} \quad (9)$$

3.3 Schrödinger equation for two electron

The Schrödinger equation for one electron has been studied in the previous section and it had been modeled with following equation. $E^{(1)}$ is the energy for one electron only:

$$-\frac{\hbar^2}{2m} \frac{d^2}{dr^2} u(r) + \frac{1}{2} k r^2 u(r) = E^{(1)} u(r),$$

Now we want to study two electron in harmonic oscillator well using repulsive Coulomb interaction. For two electron with no repulsive Coulomb interaction the Schrödinger equation can be written as:

$$\left(-\frac{\hbar^2}{2m}\frac{d^2}{dr_1^2}-\frac{\hbar^2}{2m}\frac{d^2}{dr_2^2}+\frac{1}{2}kr_1^2+\frac{1}{2}kr_2^2\right)u(r_1,r_2)=E^{(2)}u(r_1,r_2).$$

For solving this equation we introduced two new coordinate $\mathbf{r} = \mathbf{r}_1 - \mathbf{r}_2$ the relative coordinate and $\mathbf{R} = 1/2(\mathbf{r}_1 + \mathbf{r}_2)$ the center of mass coordinate. Then we rewrite the Schrödinger equation as:

$$\left(-\frac{\hbar^2}{m}\frac{d^2}{dr^2}-\frac{\hbar^2}{4m}\frac{d^2}{dR^2}+\frac{1}{4}kr^2+kR^2\right)u(r,R)=E^{(2)}u(r,R).$$

Then we can divided the equation into two r-dependent and R-dependent equation. Then we define the repulsive Coulomb interaction between two electrons:

$$V(r_1,r_2)=\frac{\beta e^2}{|\mathbf{r}_1-\mathbf{r}_2|}=\frac{\beta e^2}{r},$$

and by adding it to the r-dependent part of equation:

$$\left(-\frac{\hbar^2}{m}\frac{d^2}{dr^2}+\frac{1}{4}kr^2+\frac{\beta e^2}{r}\right)\psi(r)=E_r\psi(r).$$

This equation is similar to the equation for the one electron case, so here is we also introduce a dimensionless variable $\rho = r/\alpha$ and by repeating the same steps we have:

$$-\frac{d^2}{d\rho^2}\psi(\rho)+\frac{1}{4}\frac{mk}{\hbar^2}\alpha^4\rho^2\psi(\rho)+\frac{m\alpha\beta e^2}{\rho\hbar^2}\psi(\rho)=\frac{m\alpha^2}{\hbar^2}E_r\psi(\rho).$$

For making this equation more like the one electron case we define new frequency ω_r :

$$\omega_r^2=\frac{1}{4}\frac{mk}{\hbar^2}\alpha^4,$$

We also define some new constant as we did in previous section:

$$\alpha=\frac{\hbar^2}{m\beta e^2}\quad\quad\lambda=\frac{m\alpha^2}{\hbar^2}E$$

Now the Schrödinger equation can be rewrite as:

$$-\frac{d^2}{d\rho^2}\psi(\rho)+\omega_r^2\rho^2\psi(\rho)+\frac{1}{\rho}=\lambda\psi(\rho). \quad (10)$$

The corresponding matrix form of this equation can be written as:

$$Au = \begin{pmatrix} \frac{2}{h^2} + V_1 & -\frac{1}{h^2} & 0 & \cdots & \cdots & 0 \\ -\frac{1}{h^2} & \frac{2}{h^2} + V_2 & -\frac{1}{h^2} & 0 & \cdots & \cdots \\ 0 & -\frac{1}{h^2} & \frac{2}{h^2} + V_3 & -\frac{1}{h^2} & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & -\frac{1}{h^2} & \frac{2}{h^2} + V_{N-2} & -\frac{1}{h^2} \\ 0 & \cdots & \cdots & \cdots & -\frac{1}{h^2} & \frac{2}{h^2} + V_{N-1} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N-1} \end{pmatrix} = \lambda \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N-1} \end{pmatrix} \quad (11)$$

3.4 Jacobi eigenvalue algorithm

Jacobi eigenvalue algorithm is an iterative method for calculation of the eigenvalues and eigenvectors of a symmetric matrix:

3.4.1 Givens rotation

A Givens rotation is represented by a matrix of the form:

$$G(i, j, \theta) = \begin{pmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & c & \cdots & -s & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & s & \cdots & c & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{pmatrix} \quad (12)$$

where $c = \cos\theta$, $s = \sin\theta$ appear intersection of i th and j th rows and columns. That is the non zero elements of Givens matrix is given by $g_{kk} = 1$ for $k \neq i, j$

$$g_{ii} = c$$

$$g_{jj} = -s$$

$$g_{ij} = s \quad \text{for } i > j$$

3.4.2 Description of Jacobi algorithm

$S' = G^T S G$ where G be a Givens rotation matrix and S be a symmetric matrix then S' is also a symmetric matrix. S' has entries:

$$S'_{ij} = S'_{ji} = (c^2 - s^2)S_{ij} + sc(S_{ii} - S_{jj})$$

$$S'_{ik} = S'_{ki} = cS_{ik} - sS_{jk} \quad \text{for } k \neq i, j$$

$$S'_{jk} = S'_{kj} = sS_{ik} + cS_{jk} \quad \text{for } k \neq i, j$$

$$S'_{ii} = c^2 S_{ii} - 2sc S_{ij} + s^2 S_{jj}$$

$$S'_{jj} = s^2 S_{ii} + 2scS_{ij} + c^2 S_{jj}$$

where $c = \cos\theta$, $s = \sin\theta$, G is orthogonal, S and S' have the same Frobenious norm, however we can choose θ so that $S'_{ij} = 0$, in which case S' has a large sum of squares on the diagonal.

$$S'_{ij} = (\cos 2\theta)S_{ij} + \frac{1}{2}(\sin 2\theta)(S_{ii} - S_{jj})$$

Then we set above equation equal to zero:

$$\tan 2\theta = \frac{2S_{ij}}{S_{jj} - S_{ii}}$$

The Jacobi eigenvalue method repeatedly performs rotation until the matrix becomes almost diagonal. Then the diagonal elements are approximations of the eigenvalues of S .

3.5 Orthogonality Preservation

Unity transformation preserve orthogonality. For proving this, first, we consider a basis of vector v_i

$$\mathbf{v}_i = \begin{pmatrix} v_{i_1} \\ \vdots \\ v_{i_n} \end{pmatrix} \quad (13)$$

We assume that v_i is orthogonal:

$$\mathbf{v}_j^T \mathbf{v}_i = \delta_{ij}. \quad (14)$$

For unity transformation $\mathbf{w}_i = U\mathbf{v}_i$ we can write:

$$\begin{aligned} \mathbf{w}_i^T \mathbf{w}_j &= (U\mathbf{v}_i)^T U\mathbf{v}_j = \mathbf{v}_i^T U^T U\mathbf{v}_j = \mathbf{v}_i^T I\mathbf{v}_j \\ &\rightarrow \mathbf{w}_i^T \mathbf{w}_j = \mathbf{v}_i^T \mathbf{v}_j = \delta_{ij}, \end{aligned} \quad (15)$$

It shows that both orthogonality and dot product has been preserved. This fact will be used further in this project.

4 Implementation

I have implemented the above methods in the python programming language. The values for the potential in both cases are known at $i = 0$ and $i = n$, namely $V_0 = 0$ and $V_n = 0$. To create the matrix in equation (2), we therefore only need values for ρ_i for $i = 1, 2, \dots, n-1$, which are plugged into the diagonal elements d_i .

Listing 1: Jacobi algorithm

```
def jacobi_method(A, R, n):
```

```

for i in range(0,n):
    for j in range(0,n):
        if i==j:
            R[i][j] = 1.0
        else:
            R[i][j] = 0.0
tol = 1.0e-10
max_iteration = n**4
iteration = 0
max_offdiagonal,l,k = maxoff(A,n)
while abs(max_offdiagonal) > tol and iteration < max_iteration :
    max_offdiagonal,l,k = maxoff(A,n)
    rotate(A,R,k,l,n)
    iteration +=1
print('number of iteration: ', iteration, '\n')

```

where the function **rotate** follows the rotation algorithm presented in section (2.4.2). The function **maxoff** finds and returns the largest element in the current matrix A , also changing the values of k and l so that these refer to the largest element in A .

5 Results and Discussion

5.1 Setting up a code for tridiagonal Toeplitz matrix

Using the code which was describe in the previous section, We define $\tan \theta = \frac{s}{c}$, $\sin \theta = s$ and $\cos \theta = c$.

$$\cot 2\theta = \tau = \frac{a_{ll} - a_{kk}}{2a_{kl}}.$$

Now we can define the angel of θ in order that the non-diagonal element of the matrix become zero. Given the facts $\cot 2\theta = 1/2(\cot \theta - \tan \theta)$ and $\cot \theta = \frac{1}{\tan \theta}$ and since $\tan \theta = t$:

$$\tau = 1/2\left(\frac{1}{t} - t\right) \Rightarrow t^2 + 2\tau t - 1 = 0$$

Solving above equation:

$$t = -\tau \pm \sqrt{1 + \tau^2},$$

Now the c and s obtain easily:

$$c = \frac{1}{\sqrt{1 + t^2}}, \quad s = tc$$

Table 1 shows numerical and analytical eigenvalues for $N = 10$:

5.2 Unit Test

In this section I implemented two unit test for the code. The *maxoff* function has been checked whether it return the correct value for the maximum off-diagonal element for a given simple test matrix. The function returns the expected value so it shows that this part of program works properly. The eigenvalues of a matrix do not change after rotations. We should get the same eigenvalues for both the original matrix and the matrix after some rotations. Get the same eigenvalues for matrix before and after rotation shows that our *rotate* function works as expected.

5.3 Quantum dots in three dimensions, one electron

In this section I use Schrödinger equation for one electron (2.2) for solving the eigenvalue problem. Here we can use same code which we develop using the Jacobi's rotation algorithm. But we need to add the harmonic oscillator potential $V_i = \rho^2$ to the diagonal element of the matrix. Figure 1 shows the eigenstates of the electron plotted as a probability distribution as a function of ρ . The respective eigenvalues are indicated, and resemble those found theoretically, building confidence in the reliability of the program. The value of ρ_{\max} should, ideally, be ∞ , but as Figure 1 shows, the probability of finding the electron outside of $\rho_{\max} \approx 5$ is almost, or equal to, zero for the four lowest eigenstates. One can therefore increase the accuracy by setting ρ_{\max} lower, without needing to increase the number of grid points. Table 2 shows such a relation. Note that decreasing ρ_{\max} into an area where there is a larger than zero probability of finding the electron ($\rho_{\max} < \sim 4$), has the effect of decreasing the accuracy. By the method of trial and error, it was found that $n = 220$ is enough to get the three first eigenvalues stable to four leading digits. It is also of interest to find the number of transformations needed as a function of the number of grid points. The number of iterations needed goes like $\sim O(n^2)$.

Table 1: Comparing numerical and analytical values.

<i>Analytical</i>	<i>Numerical</i>
-0.512890	-0.51289
3.64507	3.64507
0.16574	0.16574
2.88523	2.88523
2.20157	2.20157
1.45647	1.45647
1.78628	1.78628
1.92128	1.92128
1.84952	1.45647
1.13062	1.13062

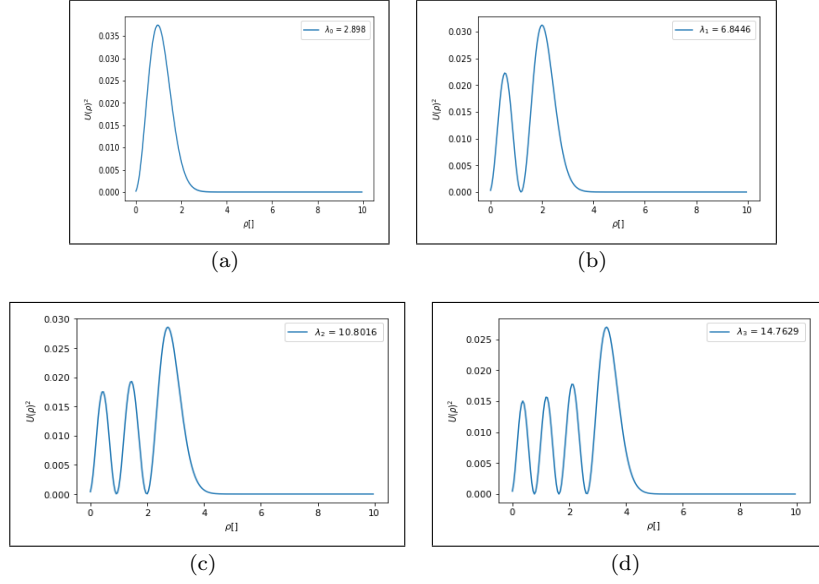


Figure 1: Probability distribution of the eigenstates for one electron in the harmonic oscillator potential. As we can see, the eigenvalues are not completely equal to the theoretical results. In this example, $n = 220$ grid points and $\rho_{\max} = 10$ were used.

ρ_{max}	λ_0	λ_1	λ_2	λ_3
10	2.8984	6.8446	10.8016	14.7629
8	2.919	6.86	10.842	14.812
6	2.939	6.907	10.882	14.861
4	2.959	6.941	10.997	15.442

Table 2: The accuracy of λ increases as the density of grid points increases, here represented by decreasing values of ρ_{\max} . $n = 220$.

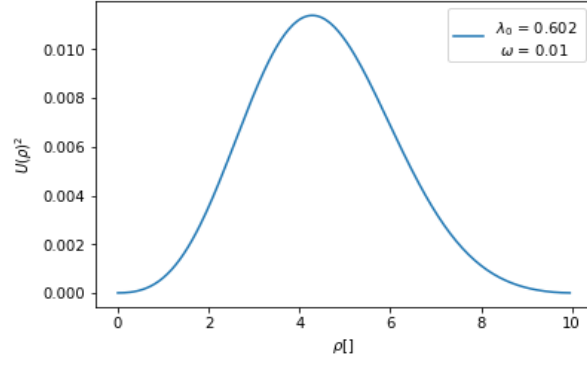
5.4 Quantum dots in three dimensions, two electron

For solving eigenvalue problem ,I use Schrödinger equation for two electrons. The algorithm is same as the one we had for the tridiagonal Toeplitz matrix. However, we need to add potential $\omega_r^2 \rho^2 + 1/\rho$ to the diagonal elements.

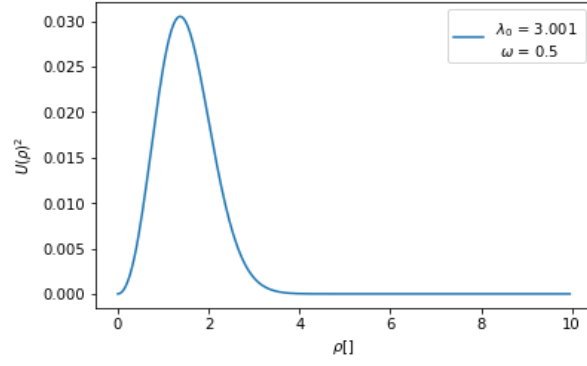
For the case of two electrons, we are interested in the wavefunction for the ground state (lowest eigenvalue) for four different values of the oscillator "frequency" ω_r . The results are shown in Figure 2. As the plot show, the oscillator frequency greatly influences the probability distribution of the position of the center of mass for the electron pair, higher frequencies leading to a narrower distribution, much closer to the origin. One other thing to note is that higher frequency leads to larger eigenvalue. This makes direct sense, as the eigenvalues $\lambda \propto E$, and higher frequencies give higher energies. figure 2

6 Conclusion

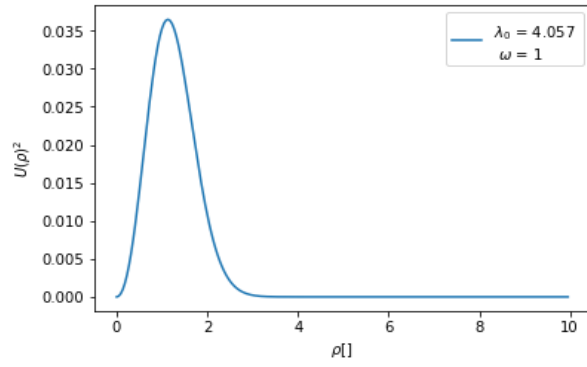
The Jacobi method is a very slow method of finding eigenvalues compared to the method embedded in the python `numpy` library. On average, it is needed around $O(n^2)$ Jacobi rotations to find the eigenvalues of an $n \times n$ matrix, each rotation using around 30-40n FLOPS, leading to $\sim O(n^3)$ FLOPS. We have seen that both methods reaches convergence for the three lowest eigenvalues relatively quickly at $n \approx 200$. A good reason to use the Jacobi method that is easily parallelizable, so for systems where a relatively small number of grid points n is sufficient, the Jacobi method may be the preferred method.



(a)



(b)



(c)

Figure 2: Plot of the probability distributions for two electrons with four different oscillator "frequencies" ω_r .