

FYS4150 Project 2

Mikkel Killingmoe Christensen

October 5, 2017

GitHub repository:

<https://github.com/mikkello/FYS4150/>

Abstract

Eigenvalue problems are common in natural science, but they are often impossible to solve analytically. It is thus important to develop fast and reliable numerical solvers. In this project, an eigenvalue solver based on Jacobi's algorithm was programmed in C++ to solve the Schroedinger equation for electrons in a harmonic oscillator. Different number of grid points, approximations of the boundary condition and frequencies in the algorithm were investigated. It was found that the choice of these values greatly influenced computation time and accuracy. It was also found that the algorithm had a considerably longer computation time than the Armadillo solver.

1 Introduction

Eigenvalue problems are common in many areas of physics and chemistry. They can, however, be difficult or impossible to solve analytically for large systems. It is thus natural to develop algorithms to solve these problems numerically. The goal of this project is to develop an algorithm that uses Jacobi's method for finding eigenvalues. It will then be used to solve the Schroedinger equation for one or two electrons in a three-dimensional harmonic oscillator well. The results can potentially be used to study so-called quantum dots, which is a hot topic in solid-state physics.

2 Theory

2.1 Schroedinger's equation for one electron

The first goal will be to solve the radial part of Schroedinger's equation (SE) for one electron in a three-dimensional harmonic oscillator well. Spherical symmetry is assumed. The equation that follows (1) is the Schroedinger equation that needs to be solved, and it will in this section be simplified for easier implementation. The derivations in this project are based on the lecture notes of Morten Hjorth-Jensen [1].

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

where $r \in [0, \infty)$ because of the transformation to spherical coordinates. The potential for the harmonic oscillator is given by:

$$V(r) = \frac{1}{2}kr^2, k = m\omega^2 \quad (2)$$

By assuming $l = 0$, the energies are given by:

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

with $n = 0, 1, 2, \dots$ and $l = 0, 1, 2, \dots$. Substituting $R(r) = (1/r)u(r)$ in equation (1) and assuming $l = 0$, yields the following equation:

$$-\frac{\hbar^2}{2m} \frac{d^2}{dr^2} u(r) + V(r)u(r) = Eu(r). \quad (4)$$

A dimensionless variable $\rho = r/\alpha$ is now introduced. This changes the potential to $V(\rho) = k\alpha^2\rho^2/2$. Inserting the new variable and the potential into equation (4), yields:

$$-\frac{d^2}{d\rho^2} u(\rho) + \frac{mk}{\hbar^2} \alpha^4 \rho^2 u(\rho) = \frac{2m\alpha^2}{\hbar^2} Eu(\rho). \quad (5)$$

By defining α through:

$$\frac{mk\alpha^4}{\hbar^2} = 1, \quad (6)$$

and also defining

$$\lambda = \frac{2m\alpha^2}{\hbar^2} E, \quad (7)$$

Schroedinger's equation can now be written as:

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

This is finally on a form that is easier to solve numerically.

By using Taylor expansion, it can be shown that the second derivative can be expressed as:

$$u'' = \frac{u(\rho + h) - 2u(\rho) + u(\rho - h)}{h^2} + O(h^2), \quad (9)$$

where h is the step length. Using n grid points, $\rho_{min} = \rho_0$ and $\rho_{max} = \rho_n$ the step length becomes

$$h = \frac{\rho_n - \rho_0}{n}. \quad (10)$$

Equation (8) can be discretized with

$$\rho_i = \rho_0 + ih \quad i = 1, 2, \dots, n, \quad (11)$$

giving the discretized version:

$$-\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + V_i u_i = \lambda u_i, \quad (12)$$

where $V_i = \rho_i^2$ is the harmonic oscillator potential. Finally, this equation can be written as a linear set of equations:

$$A\mathbf{u} = \lambda\mathbf{u} \quad (13)$$

where A is the matrix

$$A = \begin{pmatrix} \frac{2}{h^2} + V_1 & -\frac{1}{h^2} & 0 & \dots & 0 & 0 \\ -\frac{1}{h^2} & \frac{2}{h^2} + V_2 & -\frac{1}{h^2} & \dots & 0 & 0 \\ 0 & -\frac{1}{h^2} & \frac{2}{h^2} + V_3 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & -\frac{1}{h^2} & \frac{2}{h^2} + V_{N-2} & -\frac{1}{h^2} \\ 0 & 0 & 0 & 0 & -\frac{1}{h^2} & \frac{2}{h^2} + V_{N-1} \end{pmatrix} \quad (14)$$

2.2 Schroedinger's equation for two electrons

To study the case where two electrons are trapped in the well, a new equation has to be set up. Repulsive Coulomb interaction will also be added. Without the repulsive interaction, the Schroedinger equation is given by:

$$\left(-\frac{\hbar^2}{2m} \frac{d^2}{dr_1^2} - \frac{\hbar^2}{2m} \frac{d^2}{dr_2^2} + \frac{1}{2}kr_1^2 + \frac{1}{2}kr_2^2 \right) u(r_1, r_2) = E^{(2)}u(r_1, r_2). \quad (15)$$

The wave function now includes two \mathbf{r} vectors and the two-electron energy $E^{(2)}$. It is convenient to work with relative coordinates, and therefore $\mathbf{r} = \mathbf{r}_1 - \mathbf{r}_2$ and the center-of-mass coordinate $\mathbf{R} = 1/2(\mathbf{r}_1 + \mathbf{r}_2)$ are introduced.

Schrödinger's equation does now read:

$$\left(-\frac{\hbar^2}{m} \frac{d^2}{dr^2} - \frac{\hbar^2}{4m} \frac{d^2}{dR^2} + \frac{1}{4}kr^2 + kR^2 \right) u(r, R) = E^{(2)}u(r, R). \quad (16)$$

Now, the equation can be split in an \mathbf{r} part and an \mathbf{R} by writing the wave function as $u(r, R) = \psi(r)\phi(R)$. The energy can then be split up into the relative energy E_r and the center-of-mass energy E_R . The total energy is then given by the sum of these two

$$E^{(2)} = E_R + E_r. \quad (17)$$

Since the repulsive Coulomb interaction is included, a new interaction potential is needed. This is given by:

$$V(r_1, r_2) = \frac{\beta e^2}{|\mathbf{r}_1 - \mathbf{r}_2|} = \frac{\beta e^2}{r}, \quad (18)$$

where $\beta e^2 = 1.44 \text{ eVnm}$. Including this in the Schroedinger equation gives the following:

$$\left(-\frac{\hbar^2}{m} \frac{d^2}{dr^2} + \frac{1}{4}kr^2 + \frac{\beta e^2}{r} \right) \psi(r) = E_r\psi(r). \quad (19)$$

Again, the equation can be simplified by introducing a dimensionless variable $\rho = r/a$. Doing this, the equation becomes:

$$-\frac{d^2}{d\rho^2}\psi(\rho) + \frac{mk}{\hbar^2}\alpha^4\rho^2\psi(\rho) + \frac{m\alpha\beta e^2}{\rho\hbar^2}\psi(\rho) = \frac{m\alpha^2}{\hbar^2}E_r\psi(\rho). \quad (20)$$

The goal is to make the equation similar to equation (8) which is suitable for numerical algorithms. This is again done by fixing α , and also introducing a new frequency and a new energy term.

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

This finally yields an equation which is easy to work with:

$$-\frac{d^2}{d\rho^2}\psi(\rho) + \omega_r^2\rho^2\psi(\rho) + \frac{1}{\rho} = \lambda\psi(\rho), \quad (21)$$

with the potential

$$V_r(\rho) = \omega_r^2\rho^2 + \frac{1}{\rho} \quad (22)$$

As with the one electron case, equation (21) can be discretized and written as a set of linear equations. The only thing that needs to be changed in matrix A is the potential.

2.3 Conservation of dot product and orthogonality

Consider an basis of vectors \mathbf{v} ,

$$\mathbf{v}_i = \begin{bmatrix} v_{i1} \\ \vdots \\ \vdots \\ v_{in} \end{bmatrix},$$

and the transformation $\mathbf{w}_i = U\mathbf{v}_i$. The basis is orthogonal, which means that

$$\mathbf{v}_j^T \mathbf{v}_i = \delta_{ij}.$$

Because

$$\mathbf{w}_j^T \mathbf{w}_i = (U\mathbf{v}_j)^T U\mathbf{v}_i = \mathbf{v}_j^T U^T U\mathbf{v}_i = \mathbf{v}_j^T \mathbf{v}_i,$$

the the dot product and orthogonality is preserved when doing unitary or orthogonal transformations.

2.4 Analytical eigenvalues

The harmonic oscillator has well-known analytical eigenvalues which can be used in this project to check how close the numerical values are to the real values. The eigenvalues are given by the following formula:

$$E_n = \hbar\omega(2n + \frac{3}{2}). \quad (23)$$

The values need to be multiplied by a factor of two to be comparable with this situation.

3 Experimental

3.1 Jacobi's method for finding eigenvalues

Jacobi's method will be used to solve the matrix equation in this project. The algorithm will transform a matrix A into a diagonal matrix D with the eigenvalues of A presented on the diagonal. To do this, the largest non-diagonal elements are repeatedly set to zero. Since this is done numerically, a tolerance ϵ will be set to indicate when the values are close enough to be called zero.

A transformation matrix S will set the elements to zero. It will work on the matrix as follows:

$$S^T A S = B, \quad (24)$$

where B is the transformed matrix with its largest non-diagonal element set to zero. S will look as follows:

$$S = \begin{pmatrix} 1 & 0 & \dots & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \cos \theta & 0 & \dots & 0 & \sin \theta \\ 0 & 0 & \dots & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & \dots & -\sin \theta & 0 & \dots & 0 & \cos \theta \end{pmatrix} \quad (25)$$

The location of the sines and cosines will have indices corresponding to the largest element of A . This will be found by a short algorithm numerically. After finding the indices, the Jacobi algorithm can be initiated. The transformed matrix B will be constructed by the following equations:

1. $b_{ii} = a_{ii}$, $i \neq k, \quad i \neq l$
2. $b_{ik} = a_{ik} \cos \theta - a_{il} \sin \theta$, $i \neq k, \quad i \neq l$
3. $b_{il} = a_{il} \cos \theta + a_{ik} \sin \theta$, $i \neq k, \quad i \neq l$
4. $b_{kk} = a_{kk} \cos^2 \theta - 2a_{kl} \cos \theta \sin \theta + a_{ll} \sin^2 \theta$
5. $b_{ll} = a_{ll} \cos^2 \theta + 2a_{kl} \cos \theta \sin \theta + a_{kk} \sin^2 \theta$
6. $b_{kl} = (a_{kk} - a_{ll}) \cos \theta \sin \theta + a_{kl}(\cos^2 \theta \sin^2 \theta)$.

In the equations above, i is an arbitrary matrix column/row, k is the index of the row having the largest non-diagonal element and l is the index of the column having the largest non-diagonal element. Repeating the algorithm will produce smaller and smaller non-diagonal elements until they all are below the tolerance ϵ .

To find the eigenvectors w_i , the following equation is used: $\mathbf{w}_i = S\mathbf{v}_i$, where S is the transformation matrix and \mathbf{v}_i is a basis of eigenvectors. This is done numerically by updating a matrix R by using the following equations:

1. $R_{il} = \cos \theta r_{il} + \sin \theta r_{ik}$
2. $R_{ik} = -\sin \theta r_{il} + \cos \theta r_{ik}$

3.2 Unit tests

One of the goals of this project is to learn to implement unit tests to make, among other things, debugging more efficient. The tests are here presented.

3.2.1 Orthogonality test

The eigenvectors can not be linear combinations of each other, and are thus orthogonal. This can be utilized in a unit test by checking that the inner product of two different eigenvalues are zero. This was shown in section (2.3).

3.2.2 Largest element test

An important part of the solver will be to find the largest non-diagonal element. A test for the function doing this will thus be implemented. It consists of a function taking the indices of the largest non-diagonal elements, and then placing the number one at this matrix position, while setting the other elements to zero. The diagonal will remain untouched. Another function will then return the indices for the largest non-diagonal element of the new matrix. If these indices corresponds to the old ones, the "largest element unit" in the algorithm works.

4 Results

4.1 Computation time

The algorithm was run for different amount of grid points N . The Armadillo eigenvalue solver was used as a benchmark. The computations times are shown in table (1).

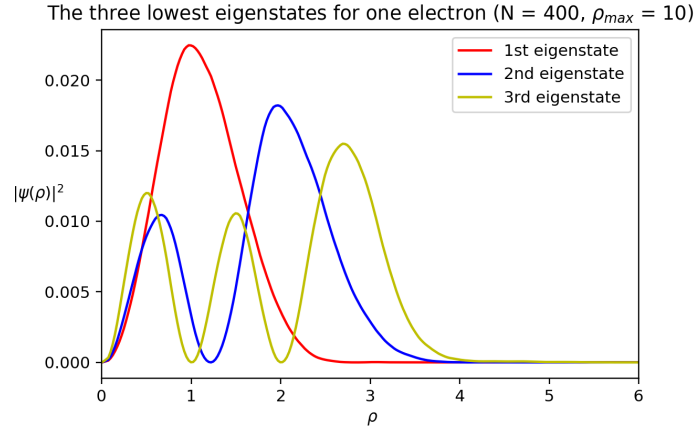
Table 1: Computation time for the different algorithms and grid points.

Grid points, N	Jacobi's method (attached code)[s]	Armadillo [s]
10	≈ 0	≈ 0
150	2.295	0.005
400	268.266	0.146
500	761.423	0.116

4.2 One-electron system

The square of the wave functions for the ground states of the one-electron system was calculated by the algorithm and plotted. It is shown in figure (1).

Figure 1: Plot of the square of the wave functions for the one-electron system. One can see that the 2nd and 3rd eigenstates has more minima/maxima than the 1st ground state (1, 2 and 3 respectively)



4.2.1 Number of iterations

The number of similarity transformations needed to calculate the eigenvalues were logged by saving the number of iterations used in the algorithm for different number of grid points, N . The relationship can be seen in table (2).

Table 2: The number of iterations in the algorithm corresponds to the number of similarity transformations. To the right, the number of iterations can be seen with respect to the number of grid points, N . ρ_{max} was set to 10. The number of iterations scales as $\approx 1.63N^2$.

Grid points, N	Iterations
10	0
150	36411
400	268846
500	422320

4.3 Choice of N and ρ

The Jacobi algorithm gives the user the opportunity to increase the accuracy by changing the number of grid points, N , and the value of ρ_{max} . The algorithm was run for different values, and the calculated lowest eigenvalues were logged. The results are presented in table (3).

4.4 Two-electron system

The square of the wave functions for the ground states of the two-electron system was found and plotted both with and without interaction. The new frequency ω was varied with the values $\omega = [0.01, 0.5, 1, 5]$. The plots can be

Table 3: Different choices of N and ρ_{max} influences the calculated eigenvalue. y .

ρ_{max}	Grid points, N	Calculated λ_0
1	10	10.0702
1	150	10.1508
1	400	10.1513
1	500	10.1511
10	10	2.68672
10	150	2.99861
10	400	2.9998
10	500	2.99987
20	10	4.49479
20	150	2.99443
20	400	2.99922
20	500	2.9995

seen in the figures (2 - 5).

Figure 2: Plot of the square of the wave functions (ground state) for the two-electron system with and without interaction ($\omega = 0.01$). One can see that the wave functions span a large part of the ρ -axis. The difference between the interacting and non-interacting case is also relatively large.

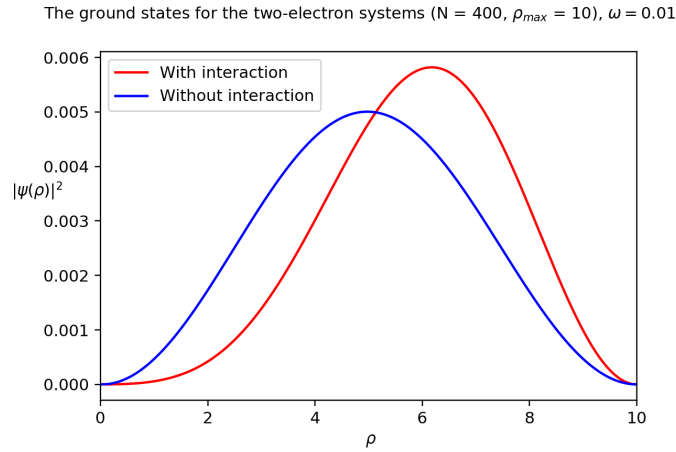


Figure 3: Plot of the square of the wave functions (ground state) for the two-electron system with and without interaction ($\omega = 0.5$).

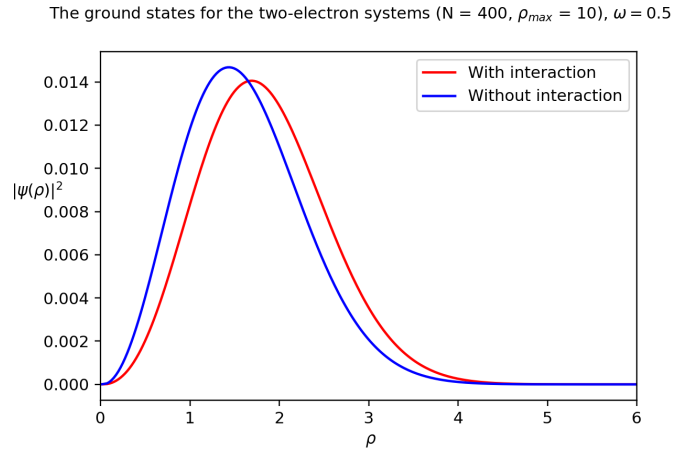


Figure 4: Plot of the square of the wave functions (ground state) for the two-electron system with and without interaction ($\omega = 1.0$).

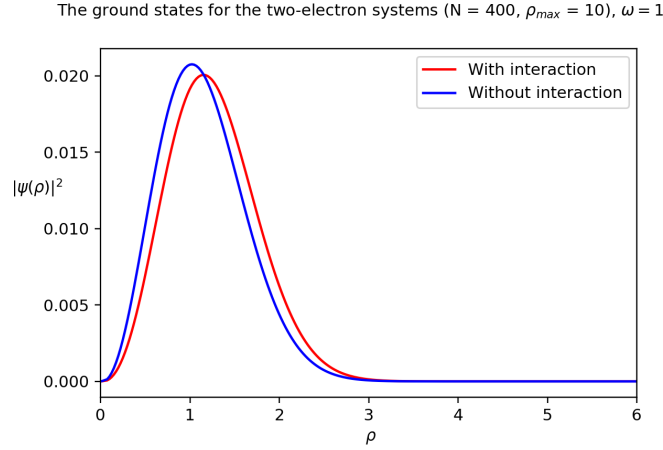
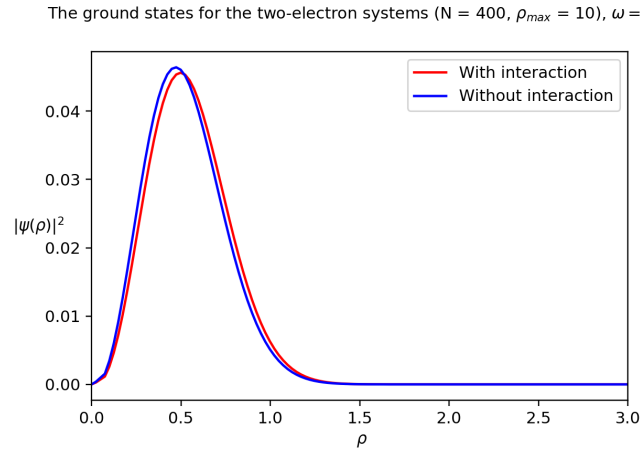


Figure 5: Plot of the square of the wave functions (ground state) for the two-electron system with and without interaction ($\omega = 5.0$). One can now see that the wave functions span a smaller part of the x-axis compared to the other figures. The difference between the interacting and non-interacting case is also smaller.



4.5 Comparison to analytical solutions

In table (4), the deviation from the analytical solutions for the one-electron case can be seen. The deviation is on the order of 10^{-4} to 10^{-3} for $N = 400$ and $\rho_{max} = 10$.

Table 4: The three first analytical eigenvalues compared to the numerical values. The number of grid points, N , were set to 400, and ρ_{max} was set to 10. The deviation is on the order of 10^{-4} to 10^{-3} .

$\lambda_{\text{Analytical}}$	$\lambda_{\text{Numerical}}$	Deviation
3	2.9998	0.0002
7	6.99902	0.00098
11	10.9976	0.0024

5 Discussion

In table (1), one could see that the Jacobi method solver made in this project was considerably slower than the solver from the Armadillo library. The time also increased quite rapidly with increasing grid points compared to the Armadillo solver. This is probably because the FLOPS of the Jacobi solver scales with a higher order of N .

In figure (1), the one-electron wavefunctions were plotted. One can here see that by increasing the energy (going from the 1st eigenstate to the 2nd and 3rd), the wavefunctions spans over a larger part of the ρ , which implies a larger radius. Furthermore, going above the ground state increases the amount of local minima and maxima. This could resemble the behaviour of electrons in an atom.

The three lowest eigenvalues were calculated with different number of grid points and values of ρ_{max} to find a satisfying accuracy. $N = 400$ and ρ_{max} were found to be sufficient with a deviation on the order of 10^{-4} to 10^{-3} . The deviations is shown in table (4), and confirms that the implemented Jacobi algorithm calculates the correct eigenvalues.

The number of iterations (similarity transformations) as a function of grid points, N , was shown in table (2). It was found that the number of iterations scaled as $\approx 1.63 \cdot N^2$. This explains the long computation times for large matrices. It is probably due to the part of the algorithm which checks for the largest element in the matrix. These types of operations scales with N^2 .

The two-electron wave functions were plotted for different values of ω , and are shown in figures ((2 - 5)). One can see that the waves becomes wider for larger values of ω . This is due to the fact that a lower value of ω makes a weaker potential. It will thus make the distance between the two electrons large, explaining the behaviour seen in this project.

In the same figures, one can also compare the interacting case with the non-interacting case. For larger values of ω , the difference between the two cases becomes lower. This is because at the stronger potential at higher values of ω , the kinetic energy of the electrons are higher, making the Coulomb interaction less important.

6 Conclusion

In this project, it was shown that the Jacobi algorithm was much slower than the Armadillo algorithm used as a benchmark. This could be due to the fact that the implemented algorithm has to search through a matrix to find the highest element which is a time consuming operation. The algorithm could probably be improved by utilizing the fact that the matrix used is already tridiagonal. Next, the choice of the number of grid points and value of ρ_{max}

was shown to influence the accuracy and computation time of the algorithm. Lastly, it was also shown that the value ω affects the probability distribution of the position of the electron by changing the strength of the potential. The influence was found to be different for the interacting and non-interacting case.

References

- [1] Morten Hjorth-Jensen, Computational Physics, August 2015