Solving Schrödinger's equation for two electrons in a three-dimensional harmonic oscillator potential with a repulsive Coulomb interaction by using the Jacobi rotation algorithm

Project 2, FYS-3150

Ina K. B. Kullmann

Abstract

The aim of this project is to numerically solve Schrödinger's equation for two electrons in a three-dimensional harmonic oscillator with a repulsive Coulomb interaction by using the Jacobi rotation algorithm.

Before solving the problem for two electrons we will look at a simpler system, one electron in a three-dimensional harmonic oscillator potential. Then we will move on to the two electrons in a three-dimensional harmonic oscillator, but first without the Coulomb interaction. For each case we will reformulate the Schrödingers equation for this system to a dimentionless form and then to a descrete eigenvalue equation. This eigenvalue problem will we solve numerically with the Jacobi rotation algorithm and compare with the Armadillo library functions for the one electron system.

The Jacobi rotation algorithm will be implemented numerically on a general form so that it can be applied to any eigenvalue problem with a symetrix matix. We will test the implementation of the Jacobi method on an arbitrary symetric matrix before moving on to solve the physical problems.

When the procedure used on the two simplest cases are tested and understood we will apply the metods on the two electron system with a Coulomb interaction and plot the probability distrubution for different strengths of the interaction.

All source codes can be found at: `https://github.com/inakbk/Project_2`.

Contents

1 Motivation and purpose

Together with linear equations and least squares, the third major problem in matrix computations deals with the algebraic eigenvalue problem.[1] It is therefore of great help to be able to solve eigenvalue problems numerically when we are dealing with large matices and complex systems. Such complex systems without any analytical solution occur often in physics.

Electrons confined in small areas in semiconductors, so-called quantum dots, form a hot research area in modern solid-state physics, with applications spanning from such diverse fields as quantum nano-medicine to the contruction of quantum gates. [2] In this article we will study two electrons moving in a three-dimensional harmonic oscillator potential that repel each other via the Coulomb interaction. Throughout this paper we will assume spherically symmetry and let the angular momentum be $l = 0$ (study the ground state only).

For the numerical methods we have chosen to limit our attention to solving eigenvalue problems with symmetric matrices as meny problems can be written on this form. In particulary we focus on similarity transformations and the Jacobi rotation algorithm where we want to obtain both the eigenvalues and the eigenvectors for the systems.

[1] cite from `lectures2015` p. 225
[2] cite from `lectures2015` p. 225

Part I
Theory

2 Numerical method

First we will describe how similarity tansformations in general can be used to solve eigenvalue problems and then in detail how the Jacobi rotation algorithm solves this problem.

Let us assume that we have the eigenvalue problem

$$\mathbf{A}\mathbf{x}^{(\mathbf{v})} = \lambda\mathbf{x}^{(\mathbf{v})}$$

where $\lambda^{(v)}$ are the eigenvalues and $\mathbf{x}^{(\mathbf{v})}$ the corresponding eigenvectors.[3] Assuming that the matrix $\mathbf{A}$ is real and symetric we can use the Jacobi rotation algoritm to solve the eigenvalue problem.

2.1 Solving eigenvalueproblems using similarity transformations

Let $\mathbf{D}$ be the diagonal matrix with the eigenvalues of $\mathbf{A}$ on the diagonal:

$$\mathbf{D} = \begin{pmatrix} \lambda_1 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \lambda_3 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots & \dots & \lambda_{n-1} & 0 \\ 0 & \dots & \dots & \dots & \dots & 0 & \lambda_n \end{pmatrix} \tag{1}$$

We say that a matrix $\mathbf{B}$ is a similarity transform of $\mathbf{A}$ if

$$\mathbf{B} = \mathbf{S^T A S}$$

Where $\mathbf{S}$ is a unitary matrix so $\mathbf{S^T S} = \mathbf{S^{-1} S} = \mathbf{I}$[4]. The point of this is that the matrix $\mathbf{B}$ has the same eigenvalues as $\mathbf{A}$. Since $\mathbf{A}$ is real and symetric there exists a real orthonogal matrix $\mathbf{S'}$ such that:

$$\mathbf{S'^T A S'} = \mathbf{D}$$

So the strategy is then to perform a series of similarity transformations on the original matrix $\mathbf{A}$ so that the matrix reduces to the diagonal matrix $\mathbf{D}$ with the eigenvalues on the diagonal:

$$\mathbf{S_N^T} \dots \mathbf{S_1^T A S_1} \dots \mathbf{S_N} = \mathbf{D}$$

where $\mathbf{S_1} \dots \mathbf{S_N} = \mathbf{S'}$. This must be done on both sides of the equation. Only one transformation is given by:

$$(\mathbf{S^T A S})(\mathbf{S^T x}) = \lambda \mathbf{S^T x}$$
$$\mathbf{B}(\mathbf{S^T x}) = \lambda(\mathbf{S^T x})$$

using that $\mathbf{S}$ is unitary. We see that the eigenvalue of $\mathbf{B}$ is the same as for $\mathbf{A}$, but the eigenvector is changed to $\mathbf{S^T x}$.

[3] The introduction to this section is based on the lecturenotes"lectures2015.pdf"

[4] `http://mathworld.wolfram.com/UnitaryMatrix.html`

2.2 Jacobi's rotation algorithm

We will now see how we can find the eigenvalues of a real and symetric matrix $\mathbf{A}$ by using the Jacobi's rotation algoritm, or Jacobi's method.

The Jacobi's method introduces the $(n \times n)$ unitary orthogonal transformation matrix

$$\mathbf{S} = \begin{pmatrix} 1 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \cos\theta & 0 & 0 & \dots & \sin\theta \\ 0 & 0 & 0 & 1 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots & \dots & 1 & 0 \\ 0 & \dots & -\sin\theta & \dots & \dots & 0 & \cos\theta \end{pmatrix} \tag{2}$$

that performs a plane rotation around an angle θ in the Euclidean n-dimensional space. For simpler notation we define the quantities $\tan\theta = t = s/c$, with $s = \sin\theta$ and $c = \cos\theta$. The elements of the matrix $\mathbf{S}$ that differ from zero is given by the indexes k, l:

$$s_{kk} = s_{ll} = c, \quad s_{kl} = -s_{lk} = -s, \quad s_{ii} = 1$$

where $i \neq k$, $i \neq l$ and i, j are the indexes of the matrix. Then the similarity transformation

$$\mathbf{B} = \mathbf{S^T A S}$$

can be written on component form as:

$$\begin{aligned} b_{ii} &= a_{ii} \\ b_{ik} &= ca_{ik} - sa_{il} \\ b_{il} &= ca_{il} - sa_{ik} \\ b_{kk} &= c^2 a_{kk} - scsa_{kl} + s^2 a_{ll} \\ b_{ll} &= c^2 a_{ll} + 2csa_{kl} + s^2 a_{kk} \\ b_{kl} &= cs(a_{kk} - a_{ll}) + (c^2 - s^2)a_{kl} \end{aligned}$$

We want to choose the angle θ so that all the non-diagonal matric elements becomes zero, that is $b_{kl} = 0$. Introducing a new variable τ dependent on k, l:

$$\begin{aligned} b_{kl} &= cs(a_{kk} - a_{ll}) + (c^2 - s^2)a_{kl} = 0 \\ &\Rightarrow \frac{a_{ll} - a_{kk}}{a_{kl}} = \frac{c^2 - s^2}{cs} = \frac{2\cos 2\theta}{\sin 2\theta} = 2\cot 2\theta \\ &\Rightarrow \tau = \cot 2\theta = \frac{a_{ll} - a_{kk}}{2a_{kl}} \end{aligned}$$

Then $b_{kl} = 0$ can be written as a second order equation

$$\begin{aligned} 2\tau = \frac{c^2 - s^2}{cs} &= \frac{c^2}{cs} - \frac{s^2}{cs} = \frac{1}{t} - t \\ &\Rightarrow 2\tau t = 1 - t^2 \\ &\Rightarrow t^2 + 2\tau t - 1 = 0 \end{aligned}$$

with the solution

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

We obtain c ans s from

$$\begin{aligned} c^2 + s^2 &= 1 \\ \Rightarrow c &= \frac{1}{\sqrt{1+t^2}}, \end{aligned}$$

and using that $s = tc$.

But which of the roots of t should we choose? To make shure that the the similarity transformation does not make bigger changes to the other elements of $\mathbf{A}$ while making another element zero we minimize the difference between the matrices $\mathbf{B}$ and $\mathbf{A}$ by letting $|\theta| \leq \pi/4$. This also makes shure that the iterations goes faster toward the solution, else the iterations might not converge at all after a reasonable number of iterations.

Separating the inequality gives $\theta \leq \pi/4$ and $\theta \geq -\pi/4$. Starting with the first inequality:

$$\begin{aligned} \theta = \arctan t &\leq \pi/4 \\ t &\leq \tan(\pi/4) = 1 \end{aligned}$$

and

$$\begin{aligned} \theta = \arctan t &\geq -\pi/4 \\ t &\geq \tan(-\pi/4) = -1 \end{aligned}$$

so we see that $|t| \leq 1 \Rightarrow -\tau \pm \sqrt{1+\tau^2} \leq 1$. If we look at the case where $\tau > 0$ and try the positive root of t we see that:

$$\begin{aligned} -\tau + \sqrt{1+\tau^2} &\leq -|\tau| + 1 + |\tau| = 1 \\ &\Rightarrow t \leq 1 \Rightarrow \theta \leq \pi/4 \end{aligned}$$

must be satisfied.

If we then try the case when $\tau < 0$ and the negative root of t we see that:

$$\begin{aligned} -\tau - \sqrt{1+\tau^2} = |\tau| - \sqrt{1+\tau^2} &\geq |\tau| - (1 + |\tau|) = -1 \\ &\Rightarrow t \geq -1 \Rightarrow \theta \geq -\pi/4 \end{aligned}$$

must be satisfied. So the choice of the root of t is dependent on τ. When $\tau > 0$ we choose the positive root, and when $\tau < 0$ we choose the negative root to make shure that $|\theta| \leq \pi/4$ is satisfied.

The Jacobi algorithm can then be described as follows:

1. Find the indexes k, l of the maximum element of the matrix $\mathbf{A}$.
2. Obtain c and s, the matrix elements of $\mathbf{S}$, given by k, l.
3. Calculate the similarity transformation $\mathbf{B} = \mathbf{S^T A S}$.

4. Start on [1.] again, setting $\mathbf{A} = \mathbf{B}$, untill all off-diagonal elements are esentially zero (less than a given threshold). When the iterations stop the eigenvalues are given by the matrix $\mathbf{D} = \mathbf{B}$.

We see that all information needed to perform a Jacobi rotation is given by the indexes k, l of the maximum elements of $\mathbf{A}$.

Retriving the eigenvectors

In this particular problem we also want to find the eigenvectors $\mathbf{x}^{(\mathbf{v})}$ of the matrix $\mathbf{A}$. The Jacobi's rotation method described in section 2.2 does not return the eigenvectors. For every similarity transformation in the Jacobi's method the eigenvector is changed from $\mathbf{x}$ to $\mathbf{S^T x}$. So it is possible to add a routine to the Jacobi's algorithm which ceeps track of the changes to the eigenvectors so that the eigenvectors can be returned when the iterations stop. The changes to the eigenvectors can be saved in a $n \times n$ matrix that is changed for every rotation. The matrix are initialized to the identity matrix and returns the eigenvectors on the columns when all the rotations are done. The returned eigenvectors are normalized.

3 One electron in a three-dimensional harmonic oscillator potential.

We will first look at a simple system with known analytical solutions and later apply the methods obtained on a more complex system. First we need to write the Schrödinger's equation on a dimensionless form and then discretize the equation to solve it numerically.

3.1 Rewriting the Schrödinger's equation to a dimensionless form

We look at the Schödinger's equation for one electron at a radius $r \in [0, \infty)$ in a harmonic oscillator potential given by:

$$V(r) = (1/2)kr^2$$

where $k = m\omega^2$ and ω is the oscillator frequence. The energy E of the harmonic oscillator in three dimensions is given by:

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

with $n = 0, 1, 2, \ldots$ and $l = 0, 1, 2, \ldots$ is the orbital momentum of the electron.

We are only interested in the solution of the radial part of the Schrödinger equation given by

$$-\frac{\hbar^2}{2m}\left(\frac{1}{r^2}\frac{d}{dr}r^2\frac{d}{dr} - \frac{l(l+1)}{r^2}\right)R(r) + V(r)R(r) = ER(r). \tag{3}$$

We then use the substitution $R(r) = (1/r)u(r)$ and obtain

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

The boundary conditions are $u(0) = 0$ and $u(\infty) = 0$.

We will modify equation 3 further by introducing a dimensionless variable $\rho = (1/\alpha)r$ where α is a constant with dimension length and get

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

Now inserting $l = 0$ and the rewritten potential $V(\rho) = (1/2)k\alpha^2\rho^2$ we obtain:

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

and then multiply by $2m\alpha^2/\hbar^2$ on both sides

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

We fix the constant α so that

$$\frac{mk}{\hbar^2}\alpha^4 = 1 \Rightarrow \alpha = \left(\frac{\hbar^2}{mk}\right)^{1/4}.$$

and then define

$$\lambda = \frac{2m\alpha^2}{\hbar^2}E,$$

Finally equation 3 can be written as

$$-\frac{d^2}{d\rho^2}u(\rho) + \rho^2 u(\rho) = \lambda u(\rho). \tag{4}$$

which is the equation we want to solve numerically. We know[5] that this equation has the eigenvalues $\lambda_0 = 3, \lambda_1 = 7, \lambda_2 = 11, \ldots$. for $l = 0$. These analytical values will be used compare with the numerical solution in this one electron system.

3.2 Discretizing the Schrödingers equation to solve the equation numerically.

We will now rewrite the Schrödingers equation on a discretized form to be able to solve it as an matrix eigenvalue problem.

We use the expression for the second derivative of the function $u(\rho)$

$$u'' = \frac{u(\rho+h) - 2u(\rho) + u(\rho-h)}{h^2} + O(h^2), \tag{5}$$

where h is our step. We define the minimum and maximum values for the variable ρ, $\rho_{\min} = 0$ and $\rho_{\max}$ so that:

$$h = \frac{\rho_{\max} - \rho_{\min}}{n_{\text{step}}}.$$

where n_{step} is a given number of steps. Since $\rho \propto r$ and $r \in [0, \infty)$ the maximum value of ρ should be $\rho_{\max} = \infty$. But we cannot set a infinite value when computing the solution numerically. We will therefore have to choose a significally large enough ρ.

[5]This is given in the assignment text for Project 2, FYS3150..

We can then define an arbitrary value of ρ as

$$\rho_i = \rho_{\min} + ih \qquad i = 0, 1, 2, \ldots, n_{\text{step}}$$

so that the Schrödinger equation for ρ_i reads

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

or in a more compact way with the harmonic oscillator potential $V_i = \rho_i^2$:

$$-\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + V_i u_i = \lambda u_i, \tag{6}$$

Then we can define first the diagonal matrix element

$$d_i = \frac{2}{h^2} + V_i,$$

and the non-diagonal matrix element:

$$e_i = -\frac{1}{h^2}.$$

We observe that in this case all the non-diagonal matrix element is given by a constant, so we can denote them all as $e = -\frac{1}{h^2}$ instead.

With these definitions the Schrödinger equation (4) takes the following form

$$d_i u_i + e_{i-1} u_{i-1} + e_{i+1} u_{i+1} = \lambda u_i, \tag{7}$$

where u_i is unknown. This can be written as a matrix eigenvalue problem

$$\begin{pmatrix} d_1 & e_1 & 0 & 0 & \ldots & 0 & 0 \\ e_1 & d_2 & e_2 & 0 & \ldots & 0 & 0 \\ 0 & e_2 & d_3 & e_3 & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & \ldots & \ldots & \ldots & \ldots & d_{n_{\text{step}}-2} & e_{n_{\text{step}}-1} \\ 0 & \ldots & \ldots & \ldots & \ldots & e_{n_{\text{step}}-1} & d_{n_{\text{step}}-1} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \ldots \\ \ldots \\ \ldots \\ u_{n_{\text{step}}-1} \end{pmatrix} = \lambda \begin{pmatrix} u_1 \\ u_2 \\ \ldots \\ \ldots \\ \ldots \\ u_{n_{\text{step}}-1} \end{pmatrix} \tag{8}$$

This is the final equation that we will solve numerically. We will test the implementation of the problem by comparing with the analytical eigenvalues, but we are also interested in the eigenvector of the ground state so that we can plot the probability distrubution.

4 Two electrons in a three-dimensional harmonic oscillator potential with Coulomb interaction.

We will now study two electrons in a harmonic oscillator well which also interact via a repulsive Coulomb interaction.

We now rewrite the single-electron equation as

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

where $E^{(1)}$ stands for the energy with one electron only. We then want to obtain a equation for the two electron system that is written on the same form as the one electron equation that we can solve numerically. First we look at the two electron syste without Coulomb interaction and we then add the interaction. Again we are only interested in the ground state with $l = 0$.

4.1 The two electron system without Coulomb interaction

The Schrödingers equation for two electrons in a harmonic oscillator potential with no Coulomb interaction can be written as

$$\left(-\frac{\hbar^2}{2m}\frac{d^2}{dr_1^2} - \frac{\hbar^2}{2m}\frac{d^2}{dr_2^2} + \frac{1}{2}kr_1^2 + \frac{1}{2}kr_2^2\right)u(r_1, r_2) = E^{(2)}u(r_1, r_2) \tag{9}$$

with a two-electron wave function $u(r_1, r_2)$ and the two-electron energy $E^{(2)}$.

Since there is no electron interaction equation 9 is separabel, it can be written as the product of two single-electron wave functions. We introduce the relative coordinate $\mathbf{r} = \mathbf{r}_1 - \mathbf{r}_2$ and the center-of-mass coordinate $\mathbf{R} = 1/2(\mathbf{r}_1 + \mathbf{r}_2)$. With these new coordinates, equation 9 reads

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

We can separate this equation into equations for r and R since the wave function $u(r, R) = \psi(r)\phi(R)$ is a product. The energy is given by the sum of the relative energy E_r and the center-of-mass energy E_R, that is

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

For this article we will omit the center-of-mass energy in the calculations. Then the equation of interest becomes:

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

By following the same approach as in section 3.2 equation 10 can written on a dimensionless form as:

$$\left(-\frac{d^2}{d\rho^2} + \frac{1}{\rho}\right)\psi(\rho) = \lambda\psi(\rho).$$

We can then use equation 8 to solve the problem numerically if the potential is replaced:

$$V_i = \rho^2 \rightarrow V_i = \frac{1}{\rho}$$

We will plot the probability distrubution to see if the results are reasonable before implementing the Coulumb interaction.

4.2 The two electron system with Coulomb interaction

We will now include the repulsive Coulomb interaction between two electrons. Introducing the term

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

with $\beta e^2 = 1.44$ eVnm. With this extra term representing the interaction the r-dependent Schrödinger equation becomes

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

We want to manipulate this equation further to make it as similar as possible to the one electron equation (3) we obtained earlier. Again we introduce the dimensionless variable $\rho = r/\alpha$ and repeating the same steps as in section 3.1, we arrive at

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

We manipulate this equation further by defining a 'frequency' which reflects the strength of the oscillator potential

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

and fix the constant α by requiring

$$\frac{m\alpha\beta e^2}{\hbar^2} = 1$$

or

$$\alpha = \frac{\hbar^2}{m\beta e^2}.$$

Then defining

$$\lambda = \frac{m\alpha^2}{\hbar^2}E,$$

so we finally can write the Schrödinger's equation as

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

We can again use equation 8 to solve the problem numerically if the potential is replaced:

$$V_i = \rho^2 \rightarrow V_i = w_r^2\rho^2 + \frac{1}{\rho}$$

For specific oscillator frequencies, the above equation has answers in an analytical form found by M. Taut[6]. We will plot the probability distrubution for the cases $\omega_r = 0.01$, $\omega_r = 0.5$, $\omega_r = 1$, and $\omega_r = 5$ for the ground state only ($l = 0$).

[6]M. Taut, Phys. Rev. A 48, 3561 - 3566 (1993).The article can be retrieved from the following web address `http://prola.aps.org/abstract/PRA/v48/i5/p3561_1`

Part II
Implementation, testing and finding the solutions of the Schrödinger equations

5 Implementation and testing of the Jacobi algorithm

Implementation

The Jacobi rotation algorithm is structured into separate files which can be run once from `main.cpp` to solve one eigenvalueproblem. All files refered to in this section can be found in the folder `project_2_test_jacobi` at:
`https://github.com/inakbk/Project_2/tree/master/project_2_test_jacobi`

The `jacobi.h` file contains all the routines for one rotation. The code finds the maximum element of the matrix, obtains the transformation matrix and does one rotation. The `jacobisolver.h` file does all the rotations for a given matrix $\mathbf{B}$ by calling the `jacobi.h` code and returns the eigenvalues and the first eigenvector. The parameter `tolerance` sets the limit of how small all the off diagonal elements in the matrix $\mathbf{B}$ needs to be before the rotations stop and the eigenvalues and the first eigenvector are returned. If the number of rotations (`numberOfIterations`) exceeds the maximum number of rotations (`maxNumberOfIterations`) before the off diagonal elements are smaller than the tolerance, the opreations are aborted with a warning message that the routine did not converge. The code also computes the execution time of all iterations for one matrix.

We then see that the code in `jacobisolver.h` does the whole job of solving the eigenvalue problem with the Jacobi method and need only be given the matrix $\mathbf{B}$ with its dimension and the maximum number of iterations. Then the `main.cpp` program only needs to constructs the matrix $\mathbf{B}$ and call the `jacobisolver.h` code. With the help of `writetofile.h` the main program will write all parameters and the eigenvalues and eigenvectors to a file with filenames that distinguish the parameters that where run. These datafiles will then later be read by a python script and plot the data.

Testing of the code

Before using the Jacobi algorithm solve the physical problems the code was tested to make sure it worked properly. The purpose of the files in the folder `project_2_test_jacobi` is to test the implementation of the Jacobi algorithm.

To do this the matrix $\mathbf{B}$ is initialized to a random symetric matrix. Then the solution is compared to the solutions generated by the `eig_sym` function in the armadillo library.

To initialize $\mathbf{B}$ as a symetrix matrix we let $\mathbf{C}$ be a random matrix and set $\mathbf{B} = \mathbf{C}\mathbf{C}^\mathbf{T}$. Then $\mathbf{B}$ is a symetric matrix:

$$\mathbf{B}^\mathbf{T} = (\mathbf{C}\mathbf{C}^\mathbf{T})^\mathbf{T} = (\mathbf{C}^\mathbf{T})^\mathbf{T}\mathbf{C}^\mathbf{T} = \mathbf{C}\mathbf{C}^\mathbf{T} = \mathbf{B}$$

using that $(\mathbf{a}\mathbf{b})^\mathbf{T} = \mathbf{b}^\mathbf{T}\mathbf{a}^\mathbf{T}$[7].

[7] `https://en.wikipedia.org/wiki/Transpose` (I dont have the mat1120 book here so wikipedia refrences it is then..)

6 Using the Jacobi rotation algorithm to solve the Schrödinger equations

Equation 8 is the matrix equation that we can use to solve both the one and two electron systems. The only difference between the systems are the potential V_i and hence the matrix $\mathbf{B}$.

We can solve equation 8 numerically using the Jacobi rotation algorithm as described in section 5 by using the files `jacobisolver.h` and `jacobi.h` and `writetofile.h` to save the results. But in `main.cpp` the matrix would have to be initialized to the matrix in equation 8:

$$\mathbf{B} = \begin{pmatrix} d_1 & e & 0 & 0 & \dots & 0 & 0 \\ e & d_2 & e & 0 & \dots & 0 & 0 \\ 0 & e & d_3 & e & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots & \dots & d_{n_{\text{step}}-2} & e \\ 0 & \dots & \dots & \dots & \dots & e & d_{n_{\text{step}}-1} \end{pmatrix}$$

This is done in the folders `project_2_1e`, `project_2_2e` and `project_2_2e_w_col`[8] for respectively the one electron system and the two electron system without Coulomb interaction and with Coulumb interaction. The only difference between the folders are the potentials, or the matrix $\mathbf{B}$, the `.h` files are identical. The only exception is that the parameter w_r is included in the last two electron case.

But before the Schrödinger equations can be solved numerically the value of $\rho_{\max}$ and n_{step} that we will use needs to be decided.

6.1 Finding reasonable values for $\rho_{\max}$ and n_{step}

The matrix $\mathbf{B}$ is only dependent on the elements d_i and e which is only dependent on $\rho_{\max}$ and the dimension of the matrix n_{step}. So which values of $\rho_{\max}$ and n_{step} should we choose? We need to test the algorithm so that we know which $\rho_{\max}$ that gives stable results and which values of n_{step} it is reasonable to use.

First we have to decide which $\rho_{\max}$ to choose as this might affect the choice of n_{step}. This can be done by looking at the first eigenvalues and choose the $\rho_{\max}$ which gives the smallest error when comparing to the analytical values. For the two electron case there are no analytical values to compare with so it is reasonable to choose a value which gives the most stable eigenvalues as a finction of n_{step}. We will therefore plot the eigenvalues as a function of $\rho_{\max}$.

Then we have to decide which precition of the eigenvalues we want to set a limit on how large n_{step} needs to be. This can be done by choosing a value of n_{step} that gives a small enough absolute error in the first three eigenvalues and that also have a reasonable execution time. We will plot the absolute error in the eigenvalues as a function of n_{step} to investigate this.

The execution time and precition is also related to how close to zero we force the off diagonal elements of $\mathbf{B}$ to be before reading off the eigenvalues. If the tolerance is too low we end up doing a lot of extra rotations that are very time consuming. We therefore should

[8]The reason the folders are separated are to not mix up the result files and be shure to be able to reproduce the results and look back if something is strange at a later point.

choose a tolerance that is low enough, but not too small. The number of transformations and the execution time will also be plotted as a function of n_{step}.

When we are done deciding on the values of $\rho_{\max}$ and n_{step} the the Schrödinger equations can be solved numerically as described in the beginning of this section.

Part III
Results and discussion

The one electron system

First the relative error between the three first numerical eigenvalues and the analytical eigenvalues where plotted as a function of $\rho_{\max} \in [3, 6]$. This plot was generated for four different values of $n_{\text{step}} \in [50, 100, 150, 200]$.

In figure 1 we see that the absolute error decreases drastically as $\rho_{\max}$ increases, here for $n_{\text{step}} = 100$, and does only increase sligtly after a minimum, for some of the eigenvalues. This was the case for all the plots for the different n_{step}.

In figure 2 we see the plots with the four different n_{step} values zoomed in on the minimum for the error in the eigenvalues. We see that the minimum for the relative error reads different values of $\rho_{\max}$ for the eigenvalues and for different choice of n_{step}. We see that for $n_{\text{step}} = 50$ the error in the eigenvalues are large, 0.01 for the second eigenvalue. For $n_{\text{step}} = 100$ the error decreases to the half, given the same range of $\rho_{\max}$.

The task is then to choose the value of $\rho_{\max}$ which gives the smallest error in all three eigenvalues combined. In every plot in figure 2 the point $(4.7, 0.0005)$ is marked. This is the point which gives the smallest absolute error in all of the eigenvalues for $n_{\text{step}} = 150$ and is the choice for $\rho_{\max}$ used in the numerical calculations.

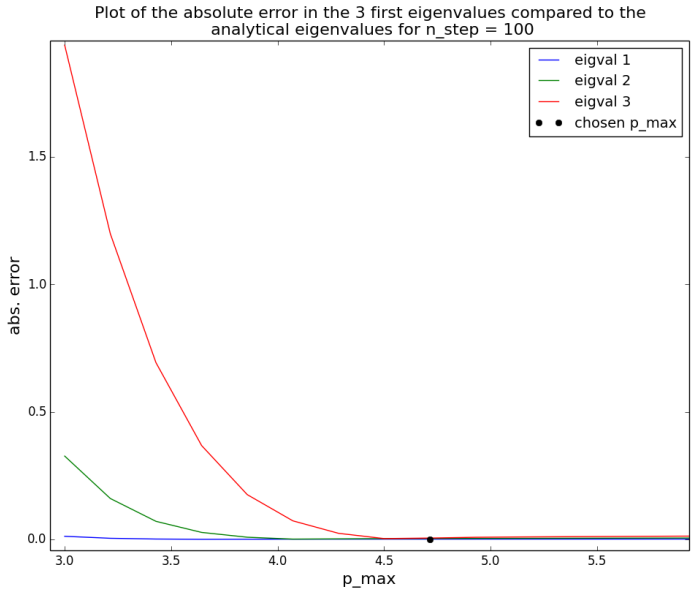


Figure 1: This should be a very long caption text, but I am too lazy to write anything right now so I would please ask you to read the text that refers to this figure.

To see if $n_{\text{step}} = 150$ is a wise choice we look at figure 3 where the absolute error is plotted as a function of n_{step}. In the top plot we see that the error decreases drastically as a function of n_{step} and is relatively small even for $n_{\text{step}} = 50$, but not for the three first eigenvalues at the same time. In the bottom plot we see that for $n_{\text{step}} = 150$ the error is maximum 0.02

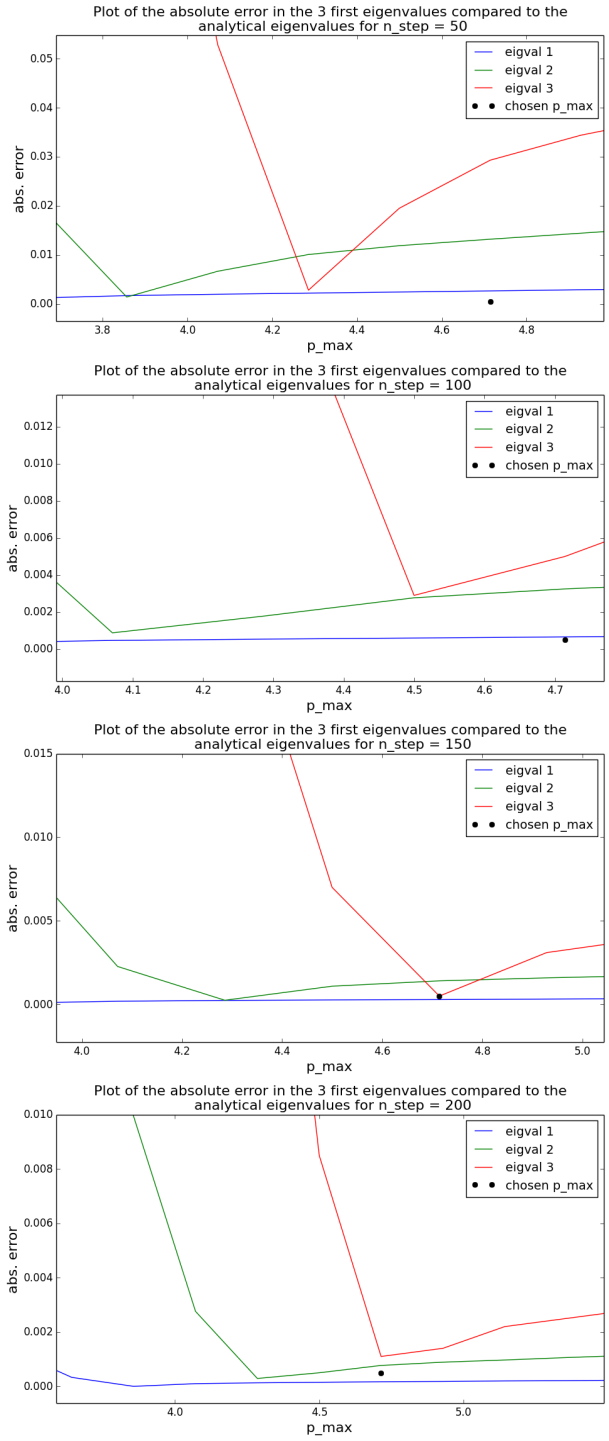


Figure 2: This should be a very long caption text, but I am too lazy to write anything right now so I would please ask you to read the text that refers to these figures.

and the effect of increasing n_{step} to 200 is small, the maximum error only decreases to 0.01.

In figure 4 we see how the dimention of the matrix effects the execution time for the Jacobi method compared to the Armadillo library. We see that the execution time increases drastically as n_{step} is increased from 150 to 200. For the Jacobi method, while the Armadillo library apeares to use the same time for all values of n_{step}. The choice og $n_{\text{step}} = 150$ seems therefore reasonable. In figure 5 we see the number of iterations as a function of n_{step}. We see that the number of iterations is more evenly increasing as a function of n_{step} than the execution time.

Finally in figure 6 we see the probability distribution, the eigenvector for the ground state squared plotted as a function of the dimensionless position ρ.

The two electron system

In figure 8 we see the value of the first, second and the third eigenvalue plotted as a function of $\rho_{\max}$ where each line in the plot corresponds to a value of n_{step}. We wish to choose a value for $\rho_{\max}$ so that the results for the eigenvalues are stable as a function of n_{step}. If we discard $n_{\text{step}} = 50$ we see that as long as we choose a value of $\rho_{\max}$ larger than 5 the values of the first, second and third eigenvalue are independent on the choice of n_{step}. Therefore $n_{\text{step}} = 150$ and $\rho_{\max} = 6$ seems like a reasonable choice.

The final desired result, the probability distribution for the interaction two electron system is plotted in figure 6. The eigenvector for the ground state squared plotted as a function of the dimensionless position ρ for different values of the strength of the Coulumb interaction w_r. We see that the shape of the distribution changes for the different values of w_r.

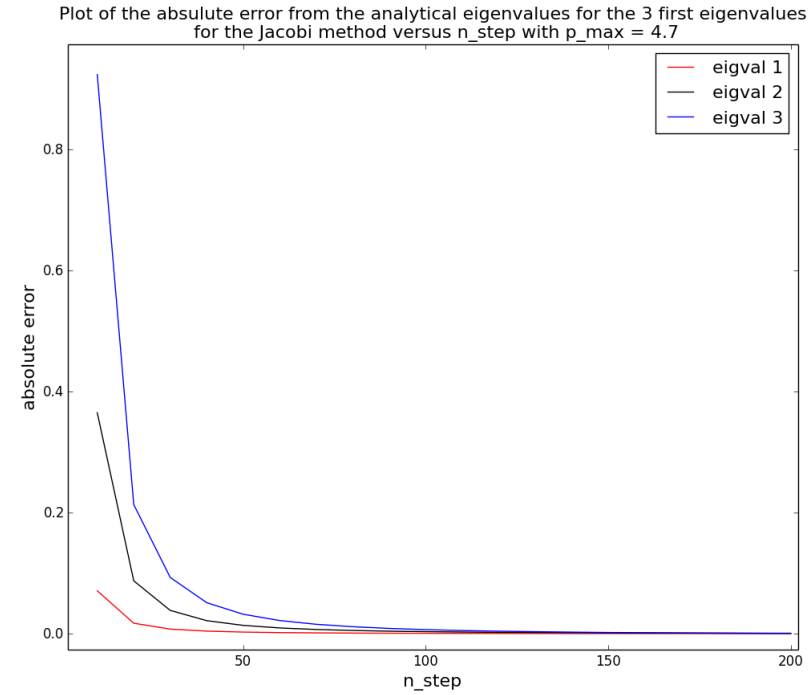
Discussion and Experiences

This problem has been both fun and frustrating to work with. It was very hard to get a owerview of the problem and a lot of time was spent on details that was not even useful for the problem. It was frustrating to 'blindfoldedly' looking for the values of n_{step} and $\rho_{\max}$ as i did not know what to expect.

The results generated for the one electron system was as expected, the solutions are already known analytically. But it was quite unexpected to see that the execution time for the Jacobi method incrased so rapidly and that the execution time for the Armadillo library did not. Later realizing that this is because the tolerance was once set to a very low value for testing, but forgotten to reajust again. So the value of the tolerance was set to 10^-14 instead of 10^-8 and this have affected the results. It would have been interesting to rerun and see if the execution time and number of iterations for the Jacobi method did change, but there was no time to do this. The Armadillo code must also have a very fast algorithm, or something have gone wrong in my routines when obtaining the execution time.

It was very satisfying to finally obtain the plot of the probability distribution for the interacting two electron system! But very frustrating to have nothing to compare with, only a insecure gut feeling that the results seems reasonable and physical.

I have also used to much time on this project, but I have learned a lot that will be useful for the next projects!



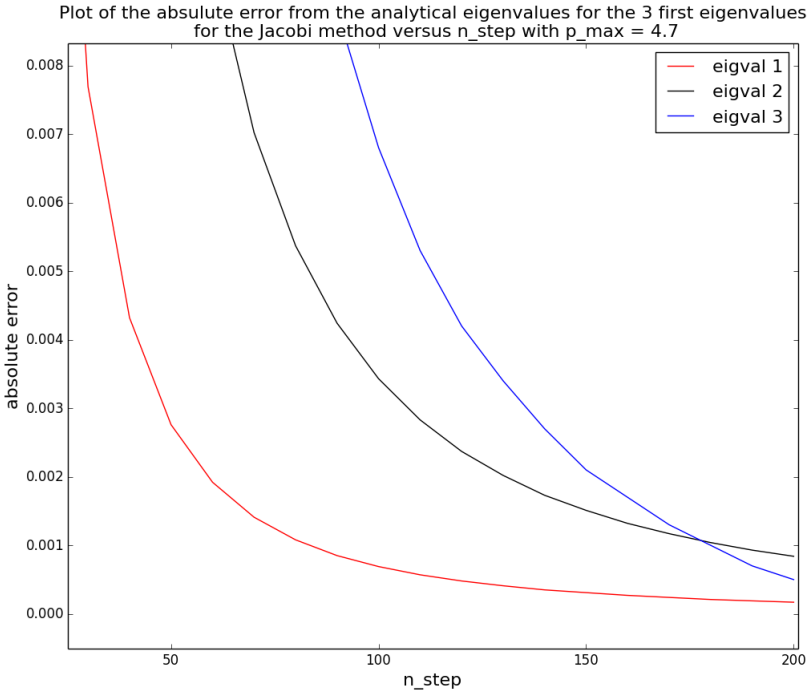


Figure 3: This should be a very long caption text, but I am too lazy to write anything right now so I would please ask you to read the text that refers to this figure.

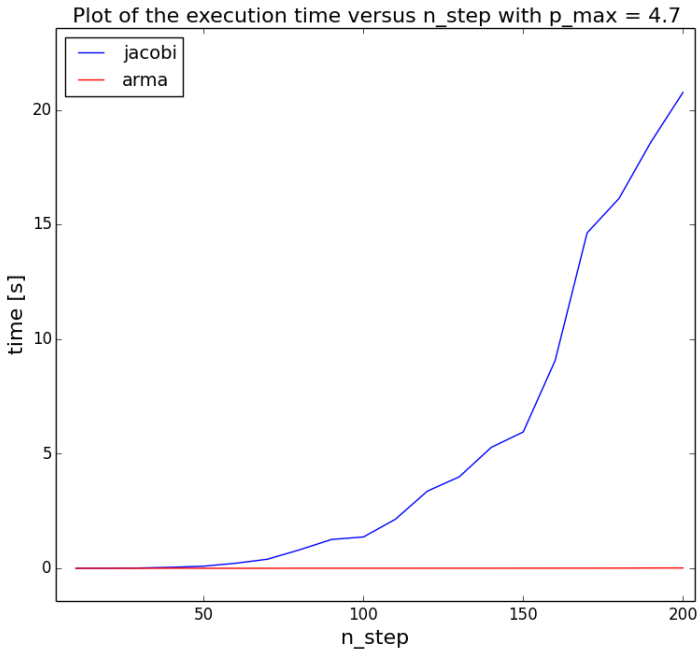


Figure 4: This should be a very long caption text, but I am too lazy to write anything right now so I would please ask you to read the text that refers to this figure.

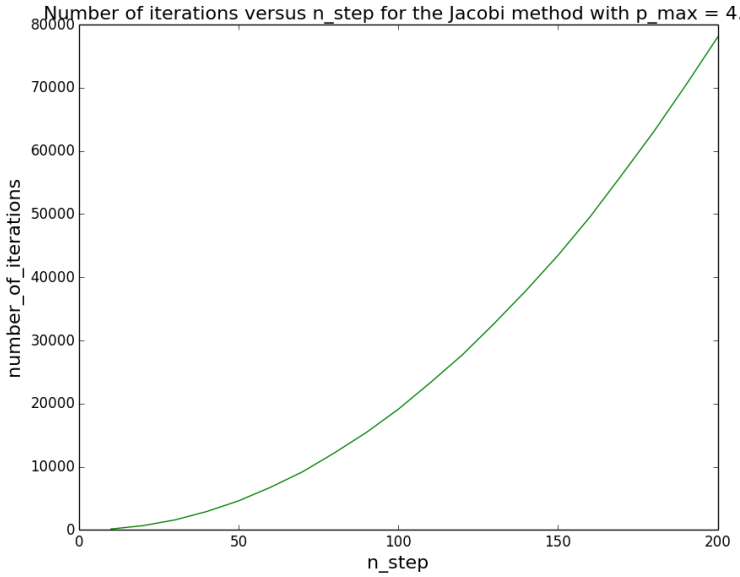


Figure 5: This should be a very long caption text, but I am too lazy to write anything right now so I would please ask you to read the text that refers to this figure.

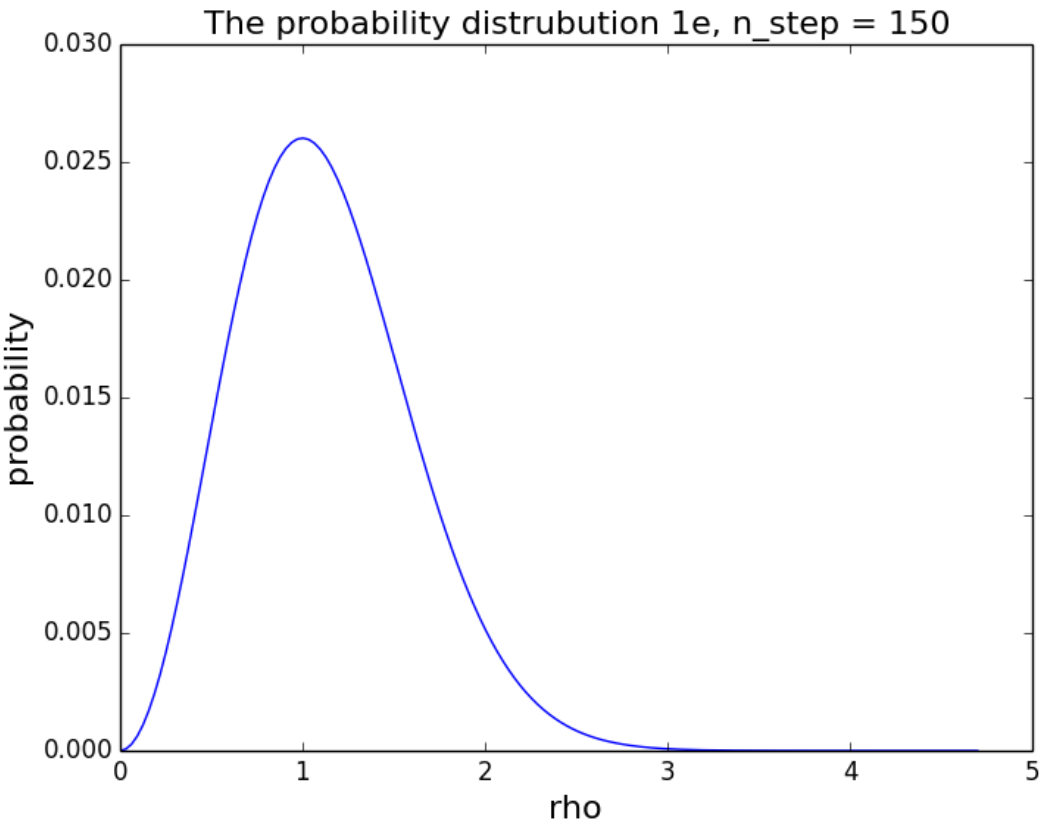


Figure 6: noe langt

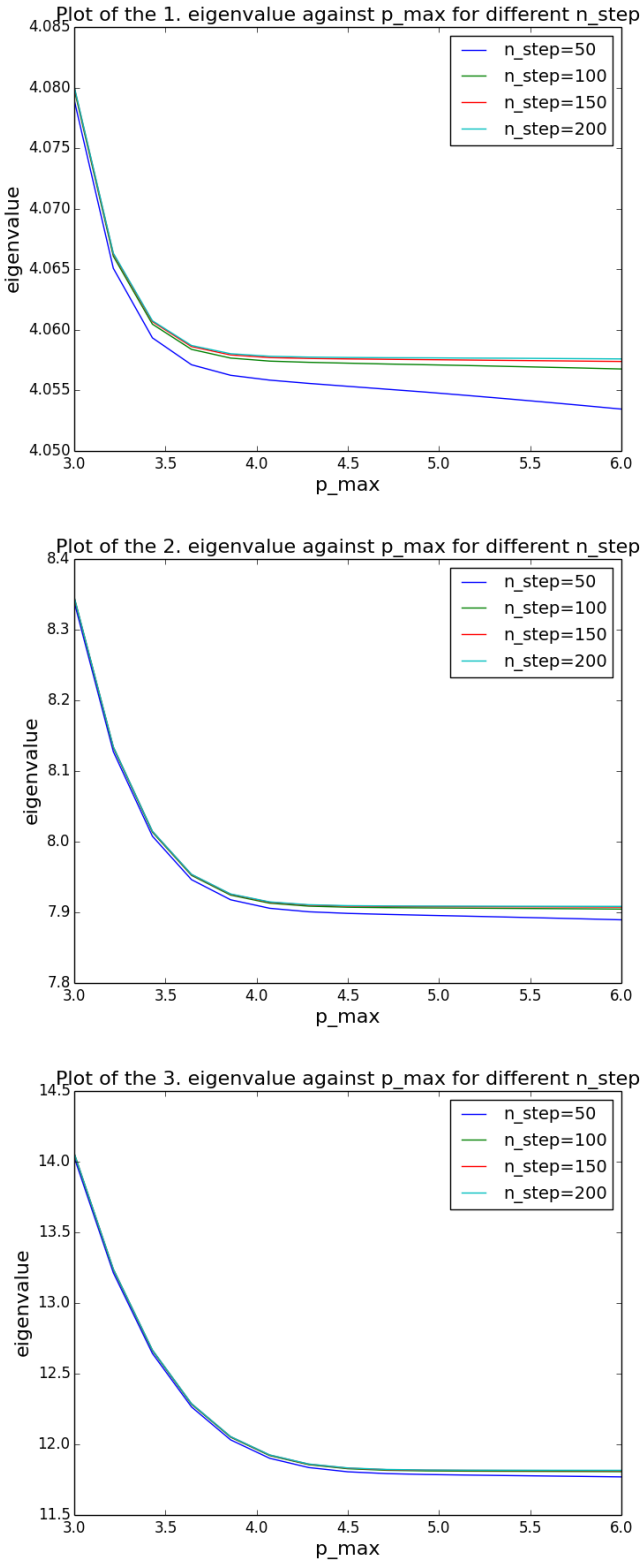


Figure 7: This should be a very long caption text, but I am too lazy to write anything right now so I would please ask you to read the text that refers to this figure.

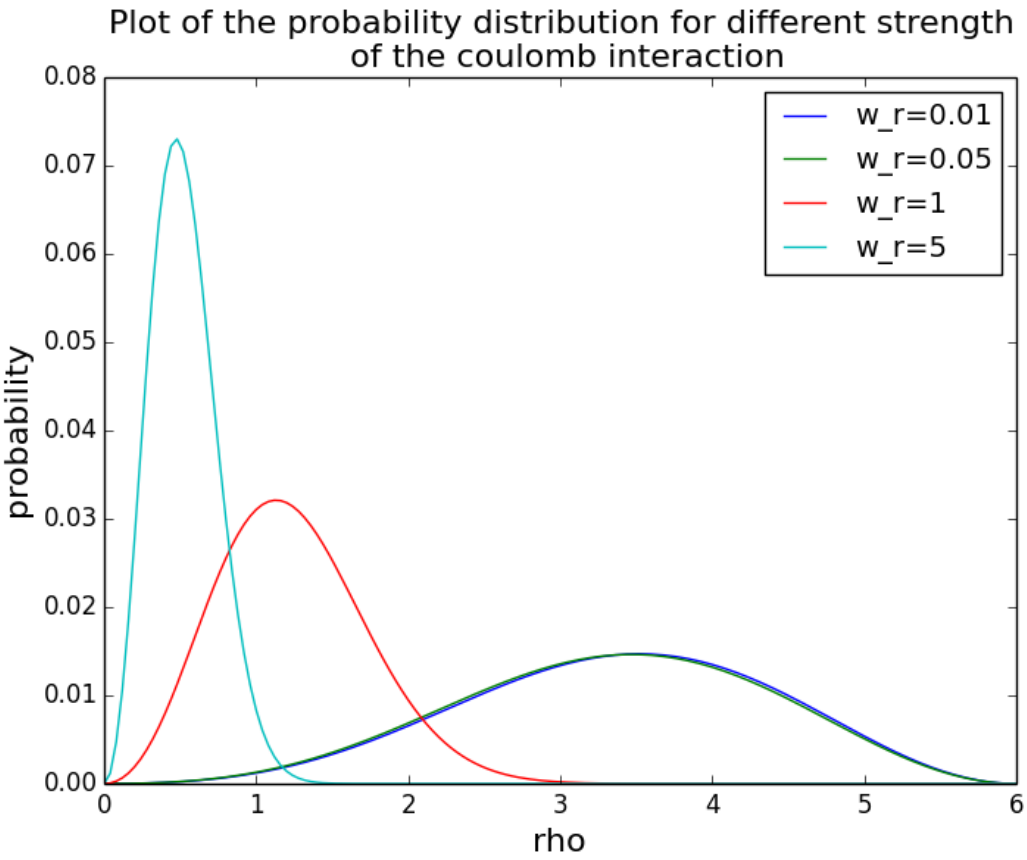


Figure 8: This should be a very long caption text, but I am too lazy to write anything right now so I would please ask you to read the text that refers to this figure.