,

FYS4150
Project 2, deadline October 2.

Authors: Robin D. Kifle, Sander W. Losnedahl and Vemund S. Thorkildsen
University of Oslo, Autumn 2017

Abstract

Introduction

The problem we will deal with in this project is of quantum mechanical nature. As none of the three authors have had any quantum mechanics courses we will focus on the mathematical and numerical side of this problem. In this project we are going to develop our own eigenvalue-solver by using Jacobi's method. We will study two different cases, the first is for one electron moving in a harmonic oscillator. The second case is for two electrons moving in a harmonic oscillator with and without repulsive coulomb interaction.

,

Method

To create our eigenvalue solver we first have to take a look at the matrix at hand, to get an understanding of the problem.

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

This equation will be solved numerically, and has given eigenvalues $\lambda_0 = 3$, $\lambda_1 = 7$ and $\lambda_2 = 11$. We can use the expression for the second derivative to rewrite this. This expression is in our case given by:

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

The last part of this expression, namely $O(h^2)$, is the truncation error and will not be used further. By using the first part, our expression now looks like this:

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

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

Our h is given by $h = \frac{\rho_n - \rho_0}{n}$, as we want our ρ_i to vary with step length h. By using this h, ρ_i will take the form: $\rho_i = \rho_o + ih$. The oscillator potential is given by $(\rho_i)^2$ and will be denoted as V_i in the rest of this article. Now we have everything we need to rearrange this problem as a a matrix eigenvalue problem.

From our expression it is easy to see that the matrix we are looking for takes the negative of element $i+1$ and $i-1$, divided by the step length squared. It also need to take two times the positive of element i divided by h^2 plus element i multiplied by the oscillator potential V_i. This means that we are once again faced by a problem that involves a tridiagonal matrix.

$$\text{Main diagonal} = \frac{2}{h^2} + V_i, \text{ first diagonal above and below} = -\frac{1}{h^2}$$

On matrix form this looks like:

$$
\begin{bmatrix}
\frac{2}{h^2} + V_1 & -\frac{1}{h^2} & 0 & 0 & \ldots & 0 & 0 \\
-\frac{1}{h^2} & \frac{2}{h^2} + V_2 & -\frac{1}{h^2} & 0 & \ldots & 0 & 0 \\
0 & -\frac{1}{h^2} & \frac{2}{h^2} + V_3 & -\frac{1}{h^2} & 0 & \ldots & 0 \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
0 & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
0 & \ldots & \ldots & \ldots & \ldots & \ldots & \frac{2}{h^2} + V_{N-1}
\end{bmatrix}
\begin{bmatrix}
u_0 \\ u_1 \\ u_2 \\ \ldots \\ \ldots \\ u_n
\end{bmatrix}
= \lambda
\begin{bmatrix}
u_0 \\ u_1 \\ u_2 \\ \ldots \\ \ldots \\ u_n
\end{bmatrix}
$$

We are going to solve this by using a Jacobi rotation algorithm. This uses a lot of similarity transformations. As long as:

$$v_j^T v_i = \delta_{ij}$$

$$U^T U = I$$

The transformations can be shown to preserve the dot product and orthogonality by multiplying with the transpose of the matrix:

$$w = Uv$$

$$w^T w = (Uv)^T Uv = v^T U^T Uv = v^T v = \delta_{ij}$$

¡¡¡¡¡¡¡ HEAD As stated earlier in this article we are going to solve this problem by using the Jacobi eigenvalue algorithm. Consider a symmetric matrix A and Givens rotation matrix R:

$$A' = RAR^T$$

A' is symmetric and similar. A' has entries:

$$A'_{ii} = c^2 A_{ii} - 2sc A_{ij} + s^2 Ajj$$

$$A'_{jj} = s^2 A_{ii} + 2sc A_{ij} + c^2 A_{jj}$$

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

$$A'_{ik} = A'_{ki} = cA_i k - sA_{jk} \qquad\qquad k \neq i, j$$

$$A'_{jk} = A'_{kj} = sA_{ik} + cA_{jk} \qquad\qquad k \neq i, j$$

$$A'_{kl} = S_{kl} \qquad\qquad k, l \neq i, j$$

where $s = sin(\theta)$ and $c = cos(\theta)$ We choose θ such that $A'_{ij} = 0$ Rewrite the 3rd equation:

$$A'_{ij} = c(2\theta)A_{ij} + \frac{1}{2}s(2\theta)(A_{ii} - A_{jj})$$

And set this equal to zero:

$$t(2\theta) = \frac{2A_{ij}}{A_{jj} - A_{ii}}$$

if $A_{jj} = A_{ii}$, then $\theta = \frac{\pi}{4}$ This is optimized by using the off-diagonal element with the largest absolute value as A_{ij} The Jacobi eigenvalue method repeats this rotation until the matrix becomes close to diagonal. The elements on the diagonal are then approximations of the real eigenvalues of A. ======= The algorithm was developed by first creating a matrix U as shown above, letting the matrix size n run from 40 to 400 in 10 steps so that the three lowest eigenvalues produced by a later algorithm would approach 3, 7 and 11. The first algorithm/function ($maxoffdiag.cpp$) finds the maximum values of the upper triangular matrix of U and then located the k and l coordinates. The max value of the upper triangular matrix is returned.

The Jacobi rotation is defined as a function in our script and is later called upon in the $jacobi.cpp$. In this script, the matrix is rotated by and angle θ until the maximum value of the upper, and also the lower diagonal, has a lower value than a set tolerance ϵ of 10^{-8}. When all the elements of besides the diagonal is less than ϵ, the rotation stops and the number of iterations is printed for the user to see.

The last script $lowesteigen.cpp$ finds the three lowest eigenvalues with the corresponding eigenvectors. The eigenvalues and eigenvectors were saved as a txt file and implemented into a MatLab script to plot.

Two tests were implemented using user input. The first one is testing if the rotation matrix R is orthogonal so that $R^T R = I$.

Results

We have run our algorithm for different cases and focused on different parameters. The different cases have been for two electrons with and without repulsive coulomb interaction. In the first four figures we have focused on the eigenvalues created by the eigenvalue solver. This has been plotted against the number of iterations n for 4 different values of ρ_{max}. This is compared to the known values of the three lowest eigenvalues, $\lambda_1 = 3$, $\lambda_2 = 7$, $\lambda_3 = 11$, to find a sensible value of ρ_{max}.
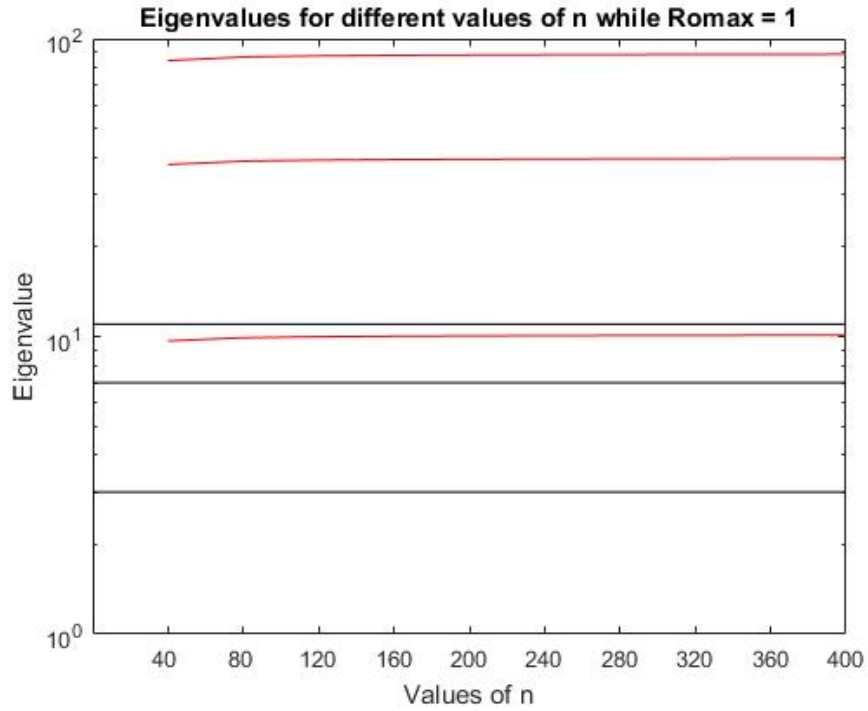
$Figure - 1$: Here we have chosen ρ_{max} to be 1. As the three lowest eigenvalues are known, this is obviously not a good fit.
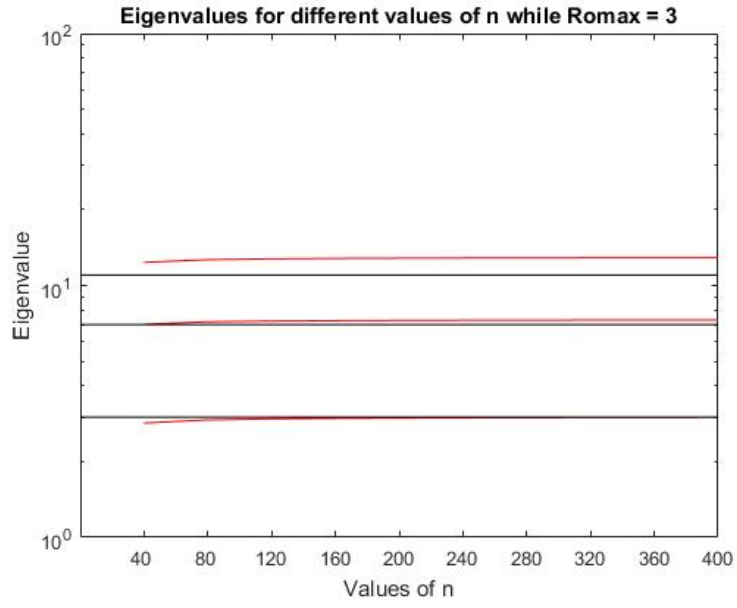
Figure − 2: Here we have chosen ρ_{max} to be 3. The eigenvalues are much closer than in figure 1, but they are still not correct.
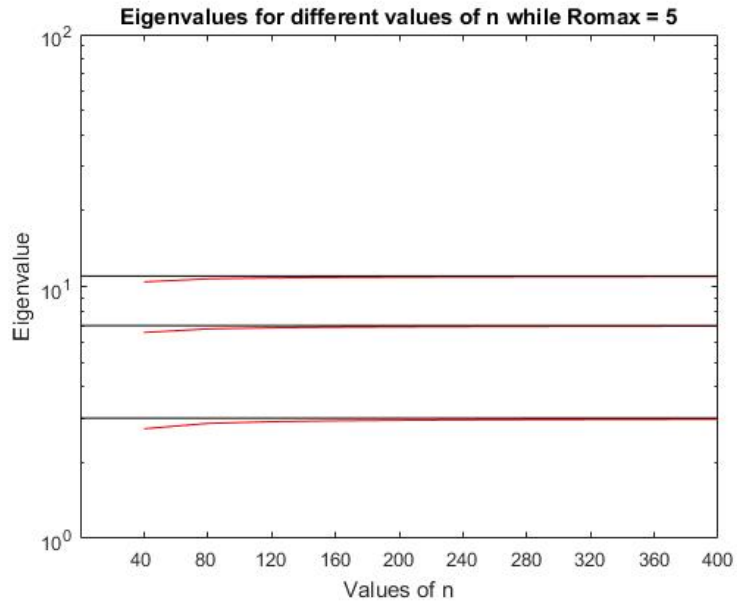

Figure − 3: Here we have chosen ρ_{max} to be 5. In this case all the eigenvalues converge to the correct values.
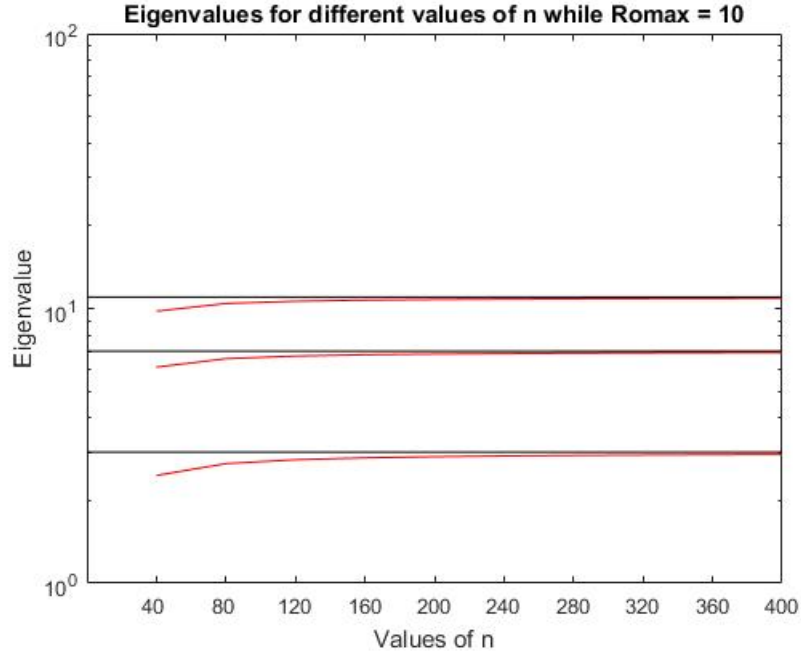
$Figure-4$: Here we have chosen ρ_{max} to be 10. In this case all the eigenvalues converge to the correct values, but slower than in the case with $\rho_{max} = 5$

From these plots it is easy to see that $\rho_{max} = 1$ and $\rho_{max} = 3$ is to low(Figure 1, Figure 2), while the two larger values seem to produce the correct results (Figure 3, Figure 4). The reason for this will be discussed later in the article. Now we can plot the wave functions produced by the same runs, i.e the case with 2 non-interacting electrons.
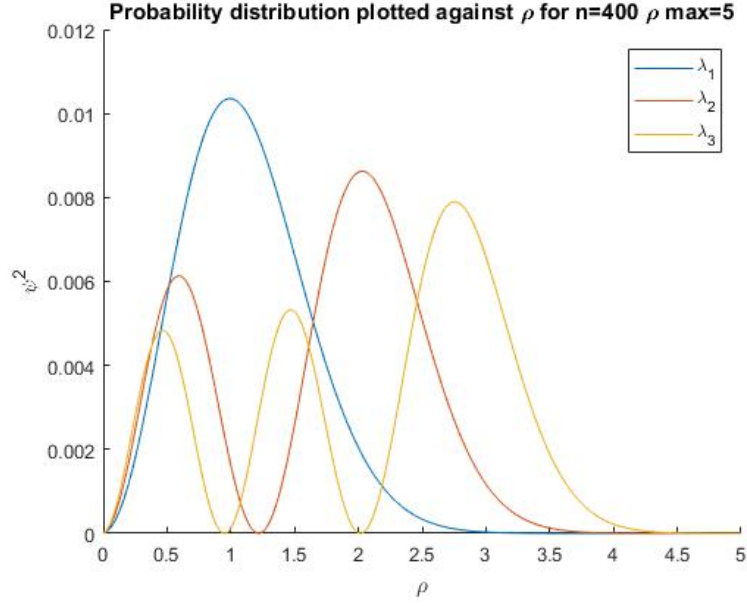
Figure − 5: Here we have chosen ρ_{max} to be 5. The probability dies out before we reach ρ_{max}. We also see that λ_1 only has one top, λ_2 has two tops and λ_3 has three tops.
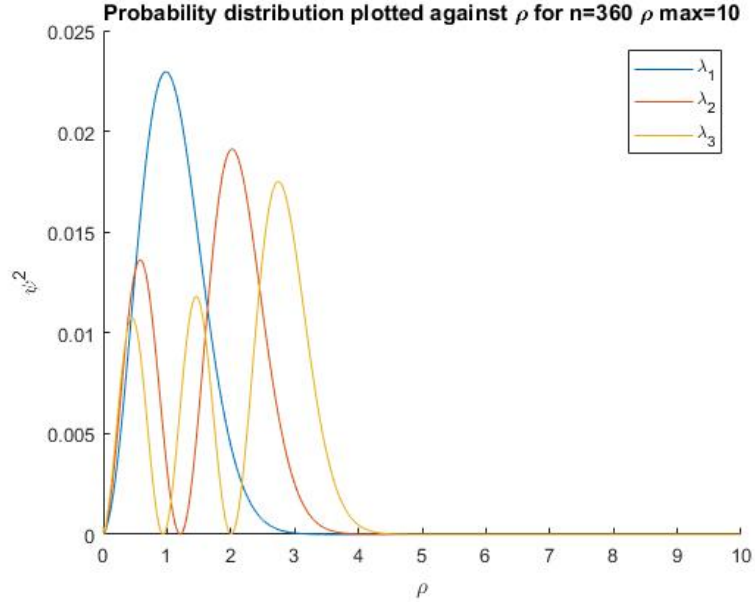

Figure − 6: Here we have chosen ρ_{max} to be 10. The wave function is of similar shape, and dies out at the approximately the same place as for $\rho_{max} = 5$.

As the wave function dies out at approximately the same place in both Figure 5 and Figure 6, the most sensible ρ_{max} will be 5.

Discussion

Concluding remarks