Project 2 - FYS3150*

Andreas G. Lefdalsnes
Student: University of Oslo, Department of Physics
email-address: andregl@student.matnat.uio.no

Tellef Storebakken
Student: University of Oslo, Department of Physics
email-address: tellefs@student.matnat.uio.no
(Dated: October 3, 2016)

In this project we solve the Schrodinger equation for two electrons in a 3D harmonic oscillator potential. We solve with and without electron repulsion, and compare the results. To accomplish this we apply a general method of discretizing the domain and reducing the problem to an eigenvalue equation. We thereafter apply Jacobi's rotation algorithm to obtain the eigenvalues of the matrix. We also apply the principles of unit testing by testing the algorithm for some simple problems with known solutions.

I. INTRODUCTION

In this project we aim to solve the Schrodinger equation for two electrons in a 3D harmonic oscillator potential. We will be solving with and without the repulsive Coulomb potential, and comparing the results. For the case of no repulsion we have an analytical expression for the energies, and this will be useful in determining the accuracy of our results. Assume spherical symmetry.

II. THEORY AND METHODS

A. The radial equation

We begin by studying the radial part of Schrodingers' equation for a single electron in a harmonic oscillator potential [1].

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

The potential $V(r)=\frac{1}{2}kr^2$ is the harmonic oscillator potential with $k=m\omega^2$ and E is the energy of the electron. ω is the oscillator frequency and the allowed energies are

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

Where the quantum number $n=0,1,2..$ is the energy quantum number and $l=0,1,2...$ is the orbital momentum quantum number. Introducing $R(r)=(1/r)u(r)$ our equation can be rewritten in terms of the second derivative d^2/dr^2:

$$-\frac{\hbar^2}{2m}\frac{d^2}{dr^2}u(r)+(V(r)+\frac{l(l+1)}{r^2}-\frac{\hbar^2}{2m})u(r)=Eu(r) \quad (3)$$

We introduce a dimensionless variable $\rho=(1/\alpha)r$ where alpha is a constant of dimension length and obtain

$$-\frac{\hbar^2}{2m\alpha^2}\frac{d^2}{d\rho^2}u(\rho)+(V(\rho)+\frac{l(l+1)}{\rho^2}\frac{\hbar^2}{2m\alpha^2})u(\rho)=Eu(\rho) \quad (4)$$

In this project we will be interested in the case $l=0$. Now since we are working in spherical coordinates, $r\in[0,\infty)$. Since we require $R(r)$ to go to zero at the boundaries, when we make the substituion $R(r)=(1/r)u(r)=(1/r)u(\alpha\rho)$ we obtain the boundary conditions for $u(\rho)$: $u(0)=u(\infty)=0$.

We insert $V(\rho)=\frac{1}{2}k\alpha^2\rho^2$ and obtain

$$-\frac{\hbar^2}{2m\alpha^2}\frac{d^2}{d\rho^2}u(\rho)+\frac{1}{2}k\alpha^2\rho^2u(\rho)=Eu(\rho) \quad (5)$$

To obtain a simpler expression we multiply by $2m\alpha^2\rho^2/\hbar^2$ and fix α such that

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

and define

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

so we can rewrite our equation as

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

* Computational Physics, autumn 2016, University of Oslo

To solve this equation we discretize the domain and define minimum and maximum values for ρ, $\rho_{min} = \rho_0 = 0$ and ρ_{max}. ρ_{max} cannot be chosen to be ∞ so we must take care to set it sufficiently large in order to obtain the correct solution. With N mesh points let

$$h = \frac{\rho_{max} - \rho_0}{N} \tag{9}$$

and we obtain a discrete set of values for ρ,

$$\rho_i = \rho_0 + ih \qquad i = 0, 1, 2..., N \tag{10}$$

Replacing the second order derivative by the 2nd order central difference we can write our equation as

$$-\frac{u_{i+1} + u_{i-1} - 2u_i}{h^2} + \rho_i^2 u_i = \lambda u_i \tag{11}$$

Where $u_i = u(\rho_i)$ is the discretized version of our function. We let $V_i = \rho_i^2$ and rewrite this as a matrix equation

$$\boldsymbol{Au} = \lambda \boldsymbol{u} \tag{12}$$

Since the endpoints are known we let A be a matrix of dimension $(n-2)\cdot(n-2)$ and u a vector of dimension (n-2).

$$\boldsymbol{Au} = \begin{pmatrix} \frac{2}{\hbar^2} + V_1 & -\frac{1}{\hbar^2} & 0 & \cdots & \cdots & 0 \\ -\frac{1}{\hbar^2} & \frac{2}{\hbar^2} + V_2 & -\frac{1}{\hbar^2} & 0 & \cdots & \cdots \\ 0 & -\frac{1}{\hbar^2} & \frac{2}{\hbar^2} + V_3 & -\frac{1}{\hbar^2} & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & -\frac{1}{\hbar^2} & \frac{2}{\hbar^2} + V_{N-3} & -\frac{1}{\hbar^2} \\ 0 & \cdots & \cdots & \cdots & -\frac{1}{\hbar^2} & \frac{2}{\hbar^2} + V_{N-2} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ \\ u_{N-2} \end{pmatrix} = \lambda \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ \\ u_{N-2} \end{pmatrix} \tag{13}$$

where $u_0 = u_{N-1} = 0$.

B. Coulomb Interaction

The single electron equation can be written as

III. RESULTS AND DISCUSSION

A. Programming precision

First we checked how many mesh points N we needed. We ran the program for a system where we knew the lowest eigenvalue and wanted the difference to be less than 10^{-4}. When doing this for a known problem where the lowest eigenvalue should have been $\lambda_{theory} = 3$ we found that we needed $N = 200$ mesh points to get this presicion.
Now that we found our right mesh point precision, we timed our algorith and compared it to the $c++$ library, Armadillo.

B. Comparison between different ω_r

[1] All theory in this project adapted from FYS3150 Project 2 (Fall 2016) < *link* >.

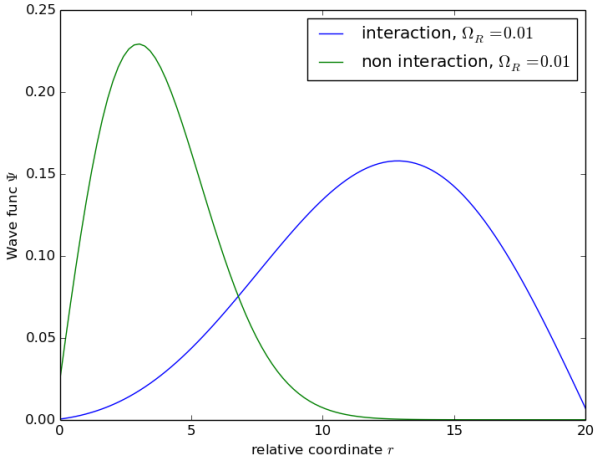

FIG. (1)

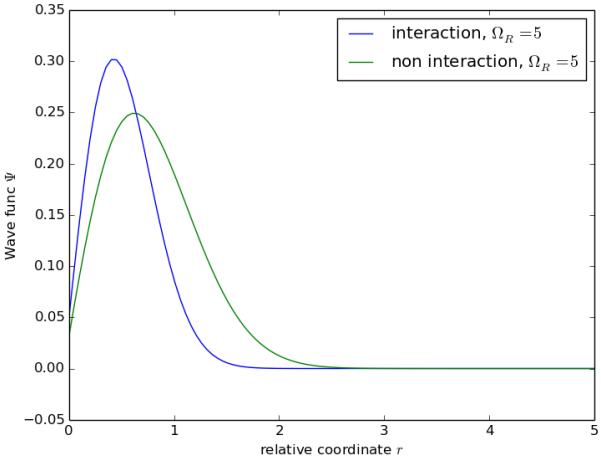

FIG. (4)

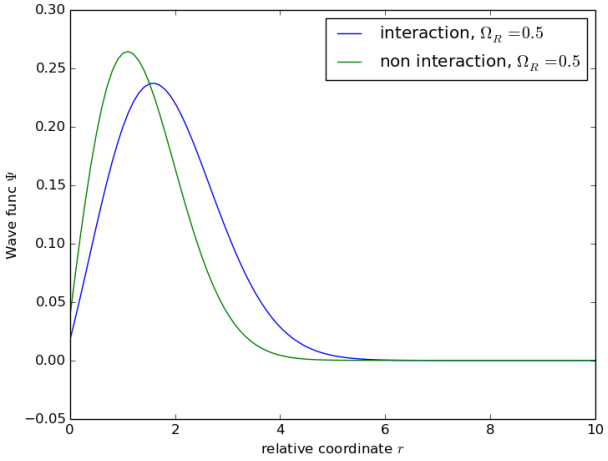

FIG. (2)

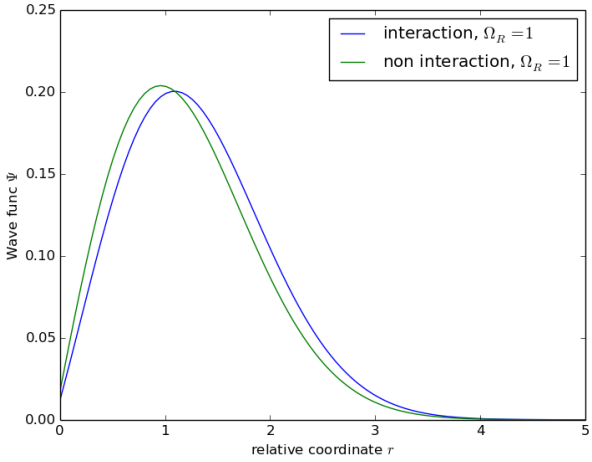

FIG. (3)