Computational Physics, Project 2

Vilde Eide Skingen
Kari Eriksen

October 6, 2013

Abstract

In project 2, we look at the harmoinc oscillator with coulomb-potential. We wish to find the eigenvalues of the hamiltonian, as well as the eigenvectors. For two electrons who repel each other, with a coulomb-potential, we get certain wavefuntions. We look at these, for different forms of potential force. Using Jacobi's method we find very good approxiamations for the first three eigenvalues, and their eigenvectors. And the wavefunctions look similar to those for we have seen for the two-particle problem.

1 Scrödinger's equation for two electrons in a three-dimensional harmonic oscillator well

1.1 Introduction

Finding the eigenvalues of the harmonic-oscillator wavefunction, can be found analytically and we already know them. But how to find them numerically?

Our task in this project is to solve Scrödinger's equation for two electrons in a three-dimensional oscillator well with and without the repulsive Coulomb interaction. We are going to solve this equation by reformulating it in a discretized form as an eigenvalue equation to be solved with Jacobi's method. We want to find how much data we need to use before getting as good an approximation as we want. First we must find the algorithm to solve this with Jocobi's method. Then we find the amount of iterations. The eigenvector problem we solve, using a build-in function from Armadillo library called eig_sym.

The method we have used, follows. Later we present our results, and last we have a discussion of our findings.

1.2 Theory

We want to solve the eigenvalue problem for both the one-particle problem and the two-particle problem. We need an expression for both cases. What we essensially need to do, is solve the Scrödinger equation by making it dimensionless.

The radial part of the wave function, $R(r)$, is a solution to

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

For the one-particel problem our V(r) is the harmonic oscillator potential $\frac{1}{1}kr^2$ with $k = m\omega^2$. E is the energy of the harmonic oscillator in three dimensions. The oscillator frequency is ω and the energies are

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

with $n = 0, 1, 2, ...$ and $l = 0, 1, 2, ...$

When we substitute $R(r) = \frac{1}{r}u(r)$ with boundary conditions $u(0) = 0$ and $u(\infty) = 0$ and introduce dimensionless variable $\rho = \frac{1}{\alpha}r$ we can rewrite Scrödinger's equation as

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

with $\alpha = \frac{\hbar^2}{mk}^{\frac{1}{4}}/4$ and $\lambda = \frac{2m\alpha^2}{\hbar^2}E$.

Our goal is to rewrite this equation as a matrix eigenvalue problem. We use the standard expression for the second derivative

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

where h is our step. With a given number of steps, n_{steps}, we define the step h as

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

If we introduce an arbitrary value of ρ as

$$\rho_i = \rho_{min} + ih$$

then the Scrödinger equation takes the form

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

or

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

Here $V_i = \rho_i^2$, and we can define the diagonal elements of a matrix as

$$d_i = \frac{2}{h^2}$$

and

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

With these definitions the Scrödinger equation takes the following form

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

where u_i is the unknown. We can write this equation as a matrix eigenvalue problem

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

When we look at the two-electrons interacting with each others, we need to change the potential in the Scrödinger equation. We put on a Coulomb force, and the solution becomes a bit different. The equation becomes

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

With relative coordinate $\mathbf{r} = \mathbf{r}_1 - \mathbf{r}_2$ and the center of mass coordinate $\mathbf{R} = 1/2(\mathbf{r}_1 + \mathbf{r}_2)$ we can separate the equation. And adding a new potensial,

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

where $\beta e^2 = 1.44$ eVnm, we can rewrite the equation to

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

This looks very much like the one for the one-electron. Only we have a new term for the potential.

Introducing a dimensionless variable $\rho = \frac{r}{\alpha}$ and the frecuency $\omega_r^2 = \frac{1}{4}\frac{mk}{\hbar^2}\alpha^4$ we end up with

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

The only differens between the first and the second solution of the Schrödinger equation, one- and two-particel problem respectivly, is a new term for the potential. For the one electron the potential was ρ^2, and for the two electrons it became $\omega_r^2 \rho^2 + \frac{1}{\rho}$.

This is essensially what we wish to solve, using Jacobi's method. To do so we must write an algorithm that can take a symmetric matrix, transform it to a diagonal matrix, where all the diagonal elements are to be the eigenvalues of the given Hamiltonian. Or the Schrödinger equation we are solving.

1.3 Method

To write the algorithm for Jacobi's method, we need to solve some equations for which are to be used in what is called Jacobi rotation. When performing the Jacobi rotation,

or the Jacobi transformation as we call it, on our symmertric matrix A, we try to put all the non-diagonal elements of the matrix to zero. This is possible by applying similarity transformation on A.

$$\mathbf{S^{-1}AS = D}$$

We see that this gives us the diagonal matrix D, where the diagonal holds the eigenvalues.

We define the quantities $tan(\theta) = t = \frac{s}{c}$, with $s = sin(\theta)$ and $c = cos(\theta)$ and

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

Using

$$cot(2\theta) = \frac{1}{2}(cot(\theta) - tan(\theta))$$

we get

$$\tau = \frac{1}{2}(\frac{1}{t} - t)$$

$$2\tau = \frac{1}{t} - t \quad | *t$$

$$t^2 + 2\tau t - 1 = 0 \Rightarrow t = -\tau \pm \sqrt{1 + \tau^2}$$

When doing the Jacobi transformation on A, we pick out the greatest non-diagonal element and set it equal to zero. When changing this element we involuntary change the other elements on the same column and row. Ideally these elements would stay zero. We want the change in these elements to be as small as possible. Thus we should choose t to be the smaller of the roots.

Largest element:

$$b_{kl} = (a_{kk} - a_{ll})cos(\theta) + a_{kl}(cos(\theta)^2 - sin(\theta)^2$$

Change in corresponding row and column:

$$b_{ik} = a_{ik}cos(\theta) - a_{il}sin(\theta)$$

$$b_{il} = a_{il}cos(\theta) + a_{ik}sin(\theta)$$

We see that the we get the lowest change when $cos(\theta) \to 1$ and $sin(\theta \to 0$.

Our goal is therefore to make $| tan(\theta) | = | \frac{sin(\theta)}{cos(\theta)} |$ as small as possible, and thereby make the change of the elements effected as little as possible. To do this we choose the smallest of the roots $tan(\theta) = t = -\tau \pm \sqrt{1 + \tau^2}$

For $\tau > 0$ the smallest t is given by $t = -\tau + \sqrt{1 + \tau^2}$.

For $\tau < 0$ the root $t = -\tau - \sqrt{1 + \tau^2}$ gives the smallest change.

To avoid loss of numerical precision we rewrite

$$\tau > 0 :$$

4

$$(-\tau + \sqrt{1+\tau^2}) * \frac{\tau + \sqrt{1+\tau^2}}{\tau + \sqrt{1+\tau^2}} = \frac{1}{\tau + \sqrt{1+\tau^2}}$$

$$\tau < 0:$$

$$(-\tau - \sqrt{1+\tau^2}) * \frac{\tau - \sqrt{1+\tau^2}}{\tau + \sqrt{1-\tau^2}} = \frac{-1}{tau + \sqrt{1+\tau^2}}$$

By this we see that the largest value we will get for $\mid tan(\theta) \mid = 1$. And thus we know that the largest value for the absolute value of the angle must be $\frac{\pi}{4}$.

In our algorithm we make a function $off - diagonal$ that takes as argument the matrix A, the index variables k and l, and the matrix dimension n. Both A and the index variables are given by address. In this function we run over the non-diagonal elements to find the biggest value of these. When having done so, we save the index for this value in the variables k and l. The function returns the maximum value.

As long as the maximum value is bigger than our chosen error limit, $\epsilon = 1 * 10^-8$, we run our $rotation$ function. This function performs the Jacobi rotation. The biggest off-diagonal value is set to zero and, as an effect of the rotation, the belonging row and column is affected as well. Our goal is to do the rotations so that all the non-diagonal elements is zero. We plead satisfied when our biggest off-diagonal matrix element is smaller than our chosen limit ϵ. We then have our eigenvalues on the diagonal of the matrix.

The eigenvalues are not ordered after value. So to find our three lowest eigenvalues we make a vector with the eigenvalues, and use the functionality $sort$ to get them into order. We then print the first three values of the matrix.

To find the eigenvalues and eigenfunctions of the two-electron wavefunction we used, as for the one-electron case, the armadillo function eig_sym. We passed the eigenvectors that corresponds to the three lowest eigenvalues to a file $data.dat$ and made a python script that read out and plotted these vectors against its index. To make it easy to change variables we compiled, ran and gave the values of n, ω and ρ from within our python script.

1.4 Results

In three dimensions the eigenvalues for $l = 0$ are $\lambda_0 = 3, \lambda_1 = 7, \lambda_2 = 11, ...$ We want to find out how many points n_{step} that is needed to get the three lowest eigenvalues with four leading digits. When running our program we see that it is first for $n_{step} = 196$ we get the wanted precision.

With our choice of tolerance ($\epsilon = 1 * 10^{-8}$) our program runs 63376 times to reach the given precision. This corresponds to the number of similarity transformations needed. We see that if we decrease our ϵ, the rotation must be performed more times to achieve the same precision.

Running for different $n - values$ we see that the number of Jacobi rotations needed increases as n increases.

5

Table 1: Needed similarity transformations for different n

n	counter
10	112
20	555
30	1322
100	16178
200	66029

We see that, roughly estimated, the number of transformations needed as function of the dimensionality of the matrix goes as n^2.

To check our results we used the armadillo function eig_sym, that gives us both the eigenvalues and the eigenvector of our symmetric matrix. We find that the three lowest eigenvalues are $2.999, 6.999$ and 10.999, which coincides with the results we found from our Jacobi method. We see that the values have four leading digit, which we needed a $n = 196$-matrix and 63376 to reach in our Jacobi rotation program. However, the armadillo program runs much faster than our Jacobi solver. We found that for a $n = 196$ matrix the armadillo program needed $0.0349s$ to do the calculations, while our program used $16.3448s$. We see that for small matrices the precision is low when using the Jacobi method, and for large matrices the time needed to reach a good precision is high.

Table 2: The three lowest eigenvalues

ω_r			
0.01	0.847	2.196	4.2768
0.5	2.231	4.171	6.402
1	4.057	7.908	11.817
5	17.443	37.047	56.793

To study two electrons in a harmonic oscillator well which also interact via a repulsive Coulomb interaction, we used our code from subproblem (a), but changed the potential to

$$V = \omega_r^2 \rho^2 + \frac{1}{\rho}$$

When running for $n = 196$ and $rho_{max} = 5$, as before, we found the lowest three eigenvalues for different ω.

Plotting the eigenvectors corresponding to the three lowest eigenvalues.

6

The first thing we notice from the plots, is how the wavefunction changes as ω and ρ_{max} changes. When we choose a big ω, the wavefunction goes faster to zero. When ω is small, we look almost at the regular wavefunctions for a harmonic oscillator.

1.5   Discussion

We used two different methods to find the eigenvalues. Both Jacobi and eig_sym gave good approximations for the eigenvalues. But eig_sym where much faster then our program. We do not know specifically how the eig_sym function workes, but there is no mystery to why the Jacobi-method is so slow. For every iteration, we must go through the entire matrix and find the lowest non-diagonal element. We then perform the rotation, and look through the matrix again. This takes a lot of time and calculations. And we must use large n's, unless we get very small step lenghts. Since we are trying to find the discutized approximation to the Schrödinger equation this would give us a very poor approximation.

When we looked at the two-paticle system, we found the eigenvalues as well. To see if the values we got where reasonable we checked it against a closed form solution. By turning off the coulomb interaction $\frac{1}{\rho}$ and using $\omega_r = 5$ we got the three lowest eigenvalues $\lambda_0 = 14,99, \lambda_1 = 34,96, \lambda_2 = 54,94$. This looks a lot like the eigenvalues we got studying the single-electron, just with a factor ω. This seems resonable, as the eigenvalues of the Hamiltonian essensially is the energy. From the expression $E_{nl} = \hbar\omega(2n + l + \frac{3}{2})$, we see that the eigenvalues for the single particle is proposional with ω. So turning of the coulomb interaction, gives us simply the eigenvalues for the single particle. When turning on the potential, the term $\omega_r^2\rho^2$ dominates for large ω's, but we get a small contribution from the term $1/\rho$.

Our final task was to plot the wavefunction. Here we could see how the wavepacket depended on both ω and ρ_{max}. Our choice of ρ_{max} depends on the width of the wavepacket. If we have a wide wavepacket we must choose a large ρ to get all the relevant data. The width of the wavepacket is dependent on the frequency, potential $V = m\omega^2$. The frequency decides the width of the curve, and thus influences our choice of ρ.

An other interesting observation was how the system behaved as we changed the potentail force. When we increase omega the potential increases strongly - and thus it will be much less likely that the particles will be located far from each other. Two electrons will repel each other, but with a very strong potensial they will be pulled more closely toward one another. This is why we see that the wavepacket reduces quite quickly to zero, and thus we do not need a too large ρ.

For the lowest state, we see that the wavefunction never crosses the x-axes. This is because the second derivative for function is never negative. The second derivative is essencially the Schrödinger equation. If we were to plot the probability distribution we would see that the probability of finding the electrons around zero would be large. The peak would actually be at zero. But as the energy for the system increases, meaning we look at the second or third state, or larger states, the probability of locating the particles around zero decreases.