FYS3150 - Project 2

Alfred Alocias Mariadason

22.09.2014

Introduction

The problem at hand is to solve the Schrödinger's equation for two electrons in a three-dimensional harmonic oscillator well. The method used for solving this is to transform(reformulate) the equation to an eigenvalue problem and then use Jacobi's method to find the eigenvalue solutions.

This report focuses mainly on the implementation of Jacobi's method with a critical analysis of the results and holds some intuitive interpretation of the physics behind the results. There are also some comments on how well(bad) Jacobi's method is compared to other similar algorithms for eigenvalue problems.

The language used for the implementation of Jacobi's method is C++ while the visualization(plot) is in python.
Programs used in this project are newprog and plotting.py. Github link for programs: `https://github.com/Oo1Insane1oO/project2`.

Theory and Algorithm

The eigenvalue problem:

As mentioned the aim of the project is to solve the Schrödinger equation. This can be archieved by transforming the equation into an eigenvalue problem and just use one of various methods to grab the eigenvalues and eigenvectors as solutions. If we assume spherical symmetry and look at one electron the equation is:

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

Here V is our harmonic oscillator potensial and E is the energy in three dimensions. l is just the orbital momentum(energy state) of the electron. Now we can do a substitution, $R=(1/r)u(r)$ and the equation reads:

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

We know the solution at the boundaries which gives $u(0)=0$, $u(\infty)=0$. Introducing a dimensionless variable $\rho=r/\alpha$ with α as a variable of dimension length the equation can be further manipulated into:

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

We know that $V(r)=(1/2)kr^2, k=m\omega^2$, with the substitution we have $V(\rho)=(1/2)k\alpha^2\rho^2$. Inserting this into the equation and multiplying with $2m\alpha^2/\hbar^2$ gives:

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

Now we can fix α to work in unit lengths:

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

Now we define $\lambda = (2m\alpha^2/\hbar^2)E$ and the final equation can be written as:

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

Here λ simply represents the energy levels and ρ is the distance in dimension length. The derivated part of the equation can be approximated with the standard equation for second derivative. If we do this and define $\rho_i = rho_min + ih, i = 1, 2 \ldots, n_{step}$ where h is the step given as $h = (\rho_{max} - \rho_{min})/n_{step}$. The approximated equation is then:

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

Here $V_i = \rho_i^2$ is the potensial. To write this into a matrix we set the diagonal elements $d_i = 2/h^2 + V_i$ as the mid-diagonal and $e_i = -1/h^2$ as the lower- and upper diagonals. Now all non-diagonal elements are given by a constant are thereby equal. The equation is finally:

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

Here u_i is the unknown and the equation can be written as a matrix eigenvalue problem with the tridiagonal matrix as follows:

$$\begin{pmatrix}
\frac{2}{h^2} + V_1 & -\frac{1}{h^2} & 0 & 0 & \ldots & 0 & 0 \\
-\frac{1}{h^2} & \frac{2}{h^2} + V_2 & -\frac{1}{h^2} & 0 & \ldots & 0 & \\
0 & -\frac{1}{h^2} & \frac{2}{h^2} + V_3 & -\frac{1}{h^2} & 0 & \ldots & 0 \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
0 & \ldots & \ldots & \ldots & \ldots & \frac{2}{h^2} + V_{n_{step}-2} & -\frac{1}{h^2} \\
0 & \ldots & \ldots & \ldots & \ldots & -\frac{1}{h^2} & \frac{2}{h^2} + V_{n_{step}-1}
\end{pmatrix}$$

Now we can use an eigenvalue algorithm like Jacobi's method, covered in the next section, to solve this matrix.

Jacobi's method:

Jacobi's method bases itself on transforming a tridiagonal matrix, making it converge towards a tridiagonal form with the eigenvalues along the mid-diagonal. This is archived by rotating the matrix with an angle theta. The rotation process is made by multiplying our matrix with an orthogonal transformation matrix. More specifically we have an $(n \times n)$ orthogonal

transformation matrix:

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

This matrix(as mentioned above) executes a plane rotation around an angle θ. We also have that $S^T = S^{-1}$. Considering this rotation the elements not 0 are given by:

$$\begin{aligned} s_{kk} &= s_{ll} = \cos\theta \\ s_{kl} &= -s_{lk} = -\sin\theta \\ s_{ii} &= -s_{ii} = 1 \\ i &\neq k \\ i &\neq l \end{aligned}$$

As mentioned the rotation process is basically multiplying the transformation matrix S with our matrix. Let's call our matrix for A and the new matrix for B. The similarity transformation is then given as:

$$B = S^T A S$$

The resulting algorithm is then:

$$\begin{aligned} b_{ii} &= a_{ii} \\ b_{ik} &= a_{ik}\cos\theta - a_{il}\sin\theta \\ b_{il} &= a_{il}\cos\theta + a_{ik}\sin\theta \\ b_{kk} &= a_{kk}\cos^2\theta - 2a_{kl}\cos\theta\sin\theta + a_{ll}\sin^2\theta \\ b_{ll} &= a_{ll}\cos^2\theta + 2a_{kl}\cos\theta\sin\theta + a_{kk}\sin^2\theta \\ b_{kl} &= (a_{kk} - a_{ll})\cos\theta\sin\theta + a_{kl}(\cos^2\theta - \sin^2\theta) \\ i &\neq k, i \neq l \end{aligned}$$

The point here is to choose a θ that makes b_{kl}(the non-digonal matrix elements) become zero. This is archieved by defining a t as $\tan\theta = \frac{s}{c}$ and $\tau = \frac{a_{ll}-a_{kk}}{2a_{kl}}$. We then obtain a quadratic equation:

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

s and c is respectivly $\sin\theta$ and $\cos\theta$ and from the expression for t, $c = \frac{1}{\sqrt{1+t^2}}$ and $s = tc$. It is easy to see from the expression of c that chosing the smaller

t will result in a θ value which is smaller than 1, this in turn gives the angle $\theta \leq \pi/4$. This last result basically ensures that the difference between our original matrix and the rotated matrix is as small as possible.

Jacobi's method is alot slower than other eigenvalue solvers. Jacobi's is also a bit unstable because one might not get zeros outside of the diagonal, that is, next iteration might change a non-diagonal element which is zero to non-zero. This means that increasing the number of iterations can actually give non-zero elements where you don't want.

Results of Jacobi's method

Running new_prog with an $n_{step} = 280$ gives these eigenvalues(5 first):

$\boldsymbol{\rho_{max} = 5}$	$\boldsymbol{\rho_{max} = 10}$
2.999901	2.999604
6.999508	6.998021
10.998990	10.995170
15.003610	14.991050
19.069403	18.985662

With some intuitive experimentation the lowest n_{step} in order to get the lowest three eigenvalues with four leading digits is 280. This was archieved with $\rho_{max} = 10$. Increasing ρ_{max} beyond this only gave worse results. Decreasing it didn't yield any better results either.

Number of similarity transformations:

n	Transformations
10	113
100	17293
150	39377
200	70538
280	139511
500	447415

The only behaviour one can see here is that the number of similarity transformations needed is determined by a factor about $1.75n^2$.

Jacobi compared to armadillo solve:

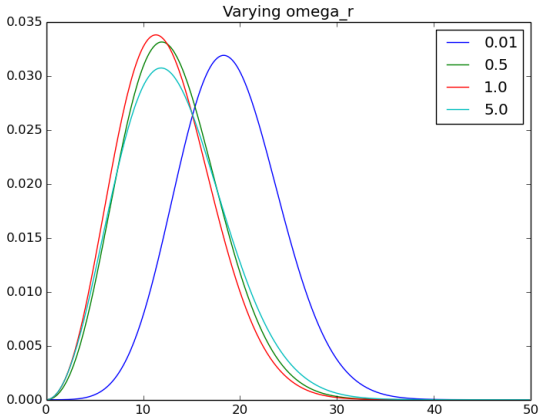
The time usage for both methods:

n_{step}	Jacobi method	Armadillo solve
10	418μs	128μs
100	230ms	1ms
150	1s	2ms
200	3s	5ms
280	16s	11ms
500	168s	60ms

One can see here that the time usage for Jacobi's method is ridiculous compared to armadillo's solve. The conclusion here is that Jacobi's method is very(very) slow, but fairly simple to implement and aslong as we ensure $\theta \leq \pi/4$ it does give a good result. The reason behind why the method is so slow is because of the number of operations needed for the algorithm to function. We first need to do the similarity transformation, and check for the right t and on top of this the maximal off-diagonal element needs to be found. Just by looking at the program where the method is implemented it is easy to see that it is a fairly long code to chew through.

Two electrons in a harmonic oscillator well

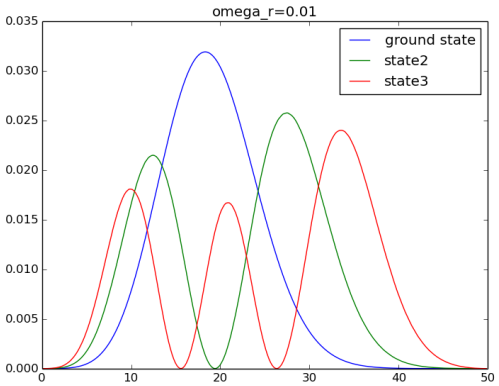
We can use the Schrodinger equation to study electrons in an oscillator well by looking at the wavefunction for the groundstate with varying ω_r. Now our potensial is on form $V = \omega_r^2\rho^2 + 1/\rho$. The procedure for aquiring the new equation(and the new potensial form) follows the same as in the first form, but we now use the relative distance between the two electrons. The equation will have the same form as previously. Let's first look at the wavefunction as potensial over distance(potensial on y-axis and distanse ρ on x-axis) for varying ω_r:



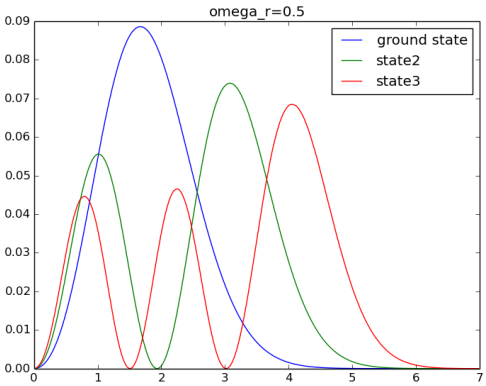
We can see here that the shape of the curve changes, an ω_r close to 1 gives a higher maximum potensial and a larger ω_r makes the potensial curves converge closer to eachother. This is a fairly logical result since ω_r is the parameter which determines the strength of the oscillator potensial. A large ω_r gives a large potensial, with other words the repulsion in the wall is large and the oscillation will ofcourse happen over a smaller distance.

Wavefunction for two electrons

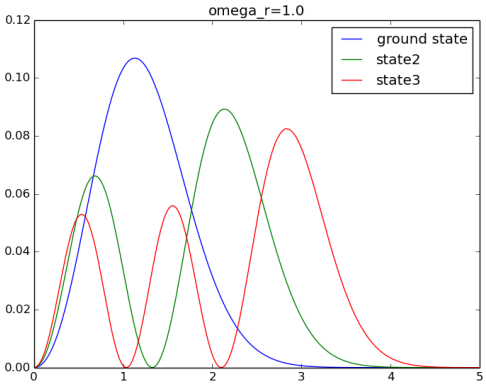
In this section we want to plot the wavefunction for two electrons. With the repulsion(Coulomb interaction) the plots look something like this:



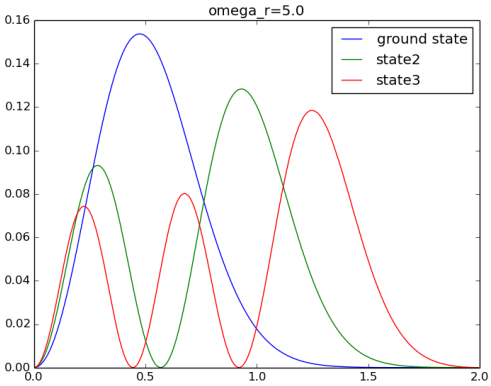
$\omega_r = 0.01$



$\omega_r = 0.5$



$\omega_r = 1.0$



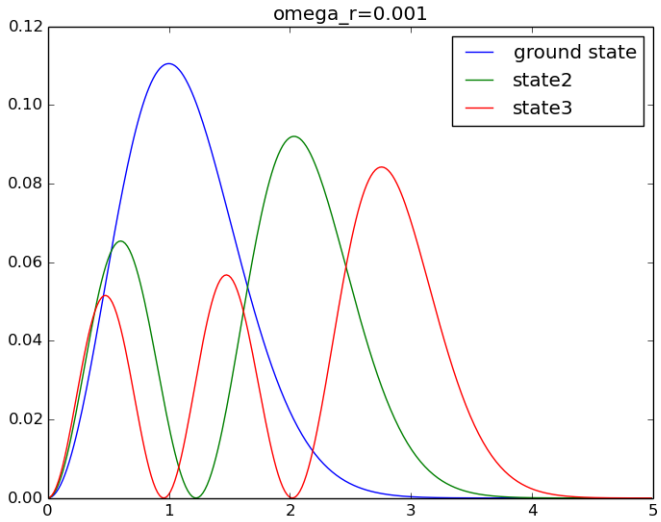
$\omega_r = 5.0$

Note: the plots are of the probability distribution(the wavefunction squared).

Increasing ω_r makes the graph shorter, which does make sense since this

is our defined frequency(and also strength of potensial). So increasing the frequancy reduces the distance at which the particle travels, with other words, the maximum potensial is archieved at a shorter distance.

Without the repulsion we have:



Note that this is basically the result from the results of Jacobi's method.

Without any interaction the graph doesn't change for different ω_r(obviously since the potensial isn't dependant on ω_r in this case). The curve also spans a much smaller distance since the potensial is only determined by the distance squared($V = \rho^2$).

Sources

Project text:

`http://www.uio.no/studier/emner/matnat/fys/FYS3150/h14/undervisningsmateriale/projects/project-2-deadline-september-22/project2_2014.pdf`

The whole theory section is essentially just the text from here.

Lecture Notes(section 7.4 Jacobi's Method):

`http://www.uio.no/studier/emner/matnat/fys/FYS3150/h14/undervisningsmateriale/Lecture%20Notes/lecture2014.pdf`

The section on Jacobi's method is taken from here.