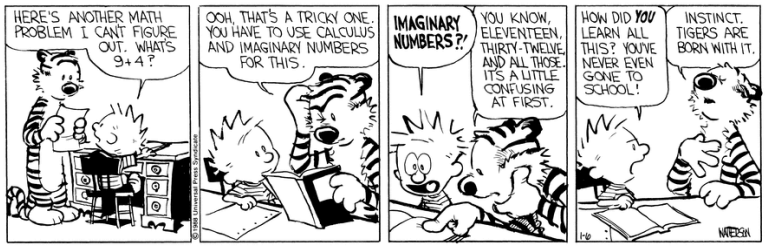
SCHRÖDINGER EQUATION FOR TWO ELECTRONS
FYS3150

IVAR HAUGALØKKEN STANGEBY

Abstract. In this project we wish to solve the Schrödinger equation for two electrons in a three-dimensional harmonic oscillator well. In approaching this problem we first wish to solve the equation for one electron, and then later expand to two electrons. In fact, it turns out that the only term differing in the two cases is the potential, hence we can develop an algorithm that runs for any given potential.
All source code can be found on my GitHub page:

github.com/qTipTip



Shamelessly stolen from [4]

Contents

1. Physical background

[1] In classical mechanics, if we know the position and momentum of a particle the system is completely determined. The analogue in quantum mechanics is the *wave-function* $\mathbf{\Psi}(\mathbf{r}, t)$ which describes the state of a system. If we know the wave-function we cannot obtain, and need not obtain any more information about the system. We will later examine the

Date: October 5, 2015.
[1]This first bit is in order for me to try to understand what I am doing, might not be as relevant for this specific project.

norm-squared of this wave-function, which represents the probability of finding a particle at a given position $\mathbf{r}$, or if you please the *probability density* of our wave function. Denoted

$$|\Psi(\mathbf{r})|^2 = \mathbb{P}(\mathbf{r}).$$

We are however typically interested in the probability of finding a particle in some neighborhood of $\mathbf{r}$, which is represented by $|\Psi(\mathbf{r})|^2 dx$. If we have two wave functions Ψ_1 and Ψ_2 that represent two different configurations of the same system, then we also have a set of *superpositions*, linear combinations of Ψ_1 and Ψ_2 that represent some configuration of the system.

We typically require that the wave function is normalized, and we will later enforce this in our implementation. This essentially means that the integral[2] over all possible positions is equal to one:

$$\int |\psi(x)|^2 \, dx = 1.$$

In other words, this tells us that the probability of finding a particle somewhere is 100%.

Heisenberg's uncertainty principle tells us that if we have great certainty in the position of a particle, then we have uncertainty in the momentum of a particle. Any wave function Ψ can be represented as a superposition of wave functions with a well defined observable quantity, namely wave-length or momentum. This is essentially what de Broglies' relations tells us:

$$\psi(x) = \int \psi(x_0)\delta(x - x_0)\, dx_0 \quad \text{definite position given by the delta-function,}$$
$$\psi(x) = \frac{1}{\sqrt{2\pi}} \int \tilde{\psi}(k) e^{ikx}\, dk \quad \text{definite wave-length given by plane-waves.}$$

1.1. **Deriving the eigenvalue problem.** The Schrödinger equation is a partial differential equation that tells us how a physical system evolves with time. The general *time-independent* Schrödinger equation reads

$$E\Psi(\mathbf{r}) = \hat{H}\Psi(\mathbf{r}), \tag{1}$$

which we see is an eigenvalue-equation. If we set up the Hamiltonian operator $\hat{H}$ for our specific system, in this case one electron, this equation then reads

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

We are interested in the radial part $R(r)$ of this equation. A series of algebraic manipulations converts this to a form we can work with, namely

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

This equation can be discretized using finite-differences which leads to

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

[2]I have often seen the notation $\int dx|\psi(x)|^2$ used. Why is this done, isn't this ambiguous?

This defines a system of equations which can be written as a single matrix equation with a tridiagonal coefficient matrix. We can then rewrite eq. (2) as a matrix eigenvalue problem:

$$\begin{bmatrix} d_1 & e_1 & 0 & 0 & \dots & 0 & 0 \\ e_1 & d_2 & e_2 & 0 & \dots & 0 & 0 \\ 0 & e_2 & d_3 & e_3 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots & \dots & d_{n-2} & e_{n-1} \\ 0 & \dots & \dots & \dots & \dots & e_{n-1} & d_{n-1} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n-2} \\ u_{n-1} \end{bmatrix} = \lambda \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n-2} \\ u_{n-1} \end{bmatrix} \tag{3}$$

We observe that each non-diagonal matrix element $e_i = -1/h^2$ is constant. This can now be solved using the Jacobi method which will be outlined in following sections.

2. Outlining the algorithm

The *Jacobi eigenvalue algorithm* is an algorithm for calculating the eigenvalues and eigenvectors of a real symmetric matrix. By performing a series of similarity transforms the matrix is transformed into an almost diagonal matrix with the eigenvalues along the diagonal. The eigenvectors are stored along the way.

2.1. **Mathematical derivation.**

We start by first considering a two-dimensional symmetric matrix

$$A = \begin{bmatrix} \alpha & \gamma \\ \gamma & \beta \end{bmatrix}. \tag{4}$$

We can then form the orthogonal matrix R given by

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix},$$

which when applied to two-dimensional vector $\mathbf{x}$ rotates it an angle θ. Letting $c = \cos\theta$ and $s = \sin\theta$ we can examine the similarity transform $R^{-1}AR$. This transform diagonalizes A given that the off diagonals are zero. We see that

$$R^{-1}AR = \begin{bmatrix} \alpha c^2 + 2\gamma cs + \beta s^2 & (\beta - \alpha)cs + \gamma(c^2 - s^2) \\ (\beta - \alpha)cs + \gamma(c^2 - s^2) & as^2 - 2\gamma cs + \beta c^2 \end{bmatrix},$$

hence we need to solve $(\beta - \alpha)cs + \gamma(c^2 - s^2) = 0$ which gives:

$$\tau = \cot 2\theta = \frac{\alpha - \beta}{2\gamma}.$$

We can now express all the trigonometric functions in terms of τ. If we let $t = \tan\theta$, $c = \cos\theta = 1/\sqrt{t^2 + 1}$ and $s = ct$. This in terms yields $\cot 2\theta = 1/2(\cot\theta - \tan\theta) = 1/2(t^{-1} - t)$; which gives us the quadratic equation

$$t^2 + 2\tau t - 1 = 0.$$

Solving this equation gives us two roots. We wish to chose the *smaller* of the two roots, in order to ensure maximum numerical stability. It is better to rotate as small an angle as possible each iteration.

We can now express the eigenvalues of A in terms of t. The diagonalized matrix looks like

$$R^{-1}AR = \begin{bmatrix} \alpha + t\gamma & 0 \\ 0 & \beta - t\gamma \end{bmatrix}.$$

Solving the characteristic equation gives us the eigenvalues. For the eigenvectors we simply perform the inverse rotation on the identity matrix I.

The algorithm outlined above for a (2×2)-matrix can very easily be extended to an $(n \times n)$-matrix case. We introduce the *Givens rotation matrix* given by

$$G(i,j,\theta) = \begin{bmatrix} 1 & \dots & 0 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \dots & c & \dots & -s & \dots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \dots & s & \dots & c & \dots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & 0 & \dots & 1 \end{bmatrix}, \tag{5}$$

where $c = \cos(\theta)$ and $s = \sin(\theta)$. The linear map this matrix represents is a rotation in the (i,j)-plane of θ radians. The reason for introducing this matrix is that is works in general as the matrix R in eq. (5) does for two dimensional cases.

Multiplying a matrix A by $G^{-1}(k,l,\theta)$ from the left and by $G(k,l,\theta)$ from the right and solving for element a_{kl} and a_{lk} equal to zero yields the following set of equations:

$$\begin{aligned} a'_{hk} &= a'_{kh} = ca_{hk} - sa_{hl} \\ a'_{hl} &= a'_{lh} = ca_{hl} - sa_{hk} \\ a'_{kl} &= a'_{lk} = (c^2 - s^2)a_{kl} + sc(a_{kk} - a_{ll}) = 0 \\ a'_{kk} &= c^2 a'_{kk} + s^2 a_{ll} - 2sca_{kl} \\ a'_{ll} &= s^2 a'_{kk} + c^2 a_{ll} + 2sca_{kl} \end{aligned} \tag{6}$$

We know that this rotation preserves the eigenvalues, however — we also wish to keep track of the corresponding eigenvectors. However, these get changed under a rotation. A way to remedy this is to do the corresponding rotations to the identity matrix I. The columns of the transformed I' will then correspond to the original eigenvectors of our matrix A.

After applying the Jacobi method to a tridiagonal matrix A, we are left with the eigenvalues $\{\lambda_i\}_{i=0}^n$ along the diagonal of the transformed matrix A', and we are left with the corresponding eigenvectors as columns in our matrix I'.

When plotting the wave-functions, we want to examine the norm-squared of the functions, however we require that $\int |\psi(x)|^2 dx = 1$. We therefore normalize our eigenvectors as following:

$$\mathbf{v}_{\lambda_i} = \frac{\mathbf{v}_{\lambda_i}}{\sqrt{\int_{\rho_{\min}}^{\rho_{\max}} |\psi(x)|^2 dx}} \quad \text{for } i = 1, \dots, n.$$

2.2. **Implementation.** We now discuss some of the implementation specific details related to this project. The implementation of this follow closely the implementation details outlined in [1, pages 217–220], however tailored to work with the `armadillo` library.

2.2.1. *Jacobi Method.* In order to perform a complete diagonalization of a matrix A using the Jacobi method we implement the auxiliary functions `rotate`, `maxoffdiag` and `solve`.

The function `rotate` performs one single rotation in the (k, l)-plane in order to zero out the elements specified. The function `maxoffdiag` returns the absolute value of the maximal off-diagonal element in our matrix and returns by pointer the position (k, l) of this given element. Finally, the function `solve` performs the Jacobi rotation until we reach our max number of iterations or the maximum off-diagonal element is sufficiently small.

2.2.2. *Unit tests.* In order to make sure that our jacobi-method does what we want it to do I implemented a set of test-functions using the UnitTesting framework *Catch*. I wanted to ensure that my Jacobi implementation satisfied some basic requirements.

i) One rotation needed to diagonalize a (2×2)-matrix
ii) The `maxoffdiag` method returned the correct indices as well as the magnitude of correct value
iii) Repeated iteration of the Jacobi method preserves orthogonality.
iv) Application to a physical system: Solving the Schrödinger equation for one electron and returning the three lowest eigenvalues exactly within some tolerance.

2.3. **Performance.** There is no way of telling a-priori how many rotations the algorithm requires in order to finish. This is due to the fact that off-diagonal elements that have been zeroed out earlier may get returned to a non-zero value by a rotation. We therefore allow our program to run until all elements are *essentially zero* given some tolerance ε, and at most we want our program to run for n^3 iterations where n is our number of time steps.

The method requires an order of $(n^2 - n)/2$ comparisons in a for-loop, with a quadratic convergence rate.

The Jacobi method — being a very slowly converging method — is however well suited for parallelization due to the fact that one can implement the method using block matrices. A numerical performance analysis of the *Block Jacobi Method* for symmetric eigenvalue problems can be found in [3].

3. Two electrons in a harmonic oscillator well

We now wish to examine the Schrödinger equation given by a two-electron wave function $u(r_1, r_2)$. Introducing the *relative coordinate* $\mathbf{r} = \mathbf{r_1} - \mathbf{r_2}$ and the *center-of-mass coordinate* $\mathbf{R} = 1/2(\mathbf{r_1} + \mathbf{r_2})$ we can — using similar manipulations as in section 1 — rewrite the Schrödinger equation as

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

The quantity ω_r reflects the strength of the oscillator potential, so intuitively the stronger the oscillator potential, the less broad should the wave-function ψ look like. We see based on our results in fig. 1 that this is indeed the case. We see that the higher ω_r we chose, the smaller $\rho_{\max}$ we need.

In **??** we see that the quantification of

If we plot the eigenstates for the same parameters with and without coloumb-interaction we see that the coloumb interaction constitutes a "widening" of the norm-squared. Ideally I should have had some plots to display this, but time ran out.

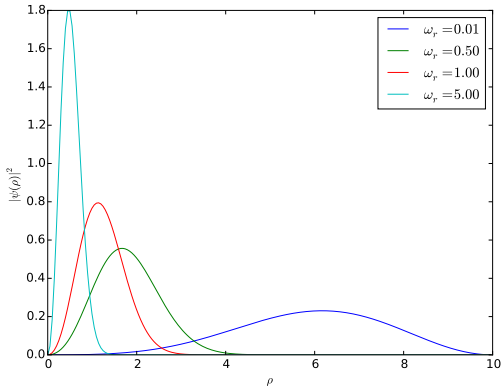


FIGURE 1. Plotting the lowest lying eigenstate (ground state) of two electrons for various oscillator-frequencies ω_r. We assume Coloumb-interaction. We see that increasing ω_r results in a more certain position closer to ρ_{min} which is reasonable since ω_r represents the strength of the harmonic oscillator. This essentially means that the stronger the potential the smaller distance between the two electrons. Here we plot for $n = 200$ and $\rho_{\text{max}} = 10$.

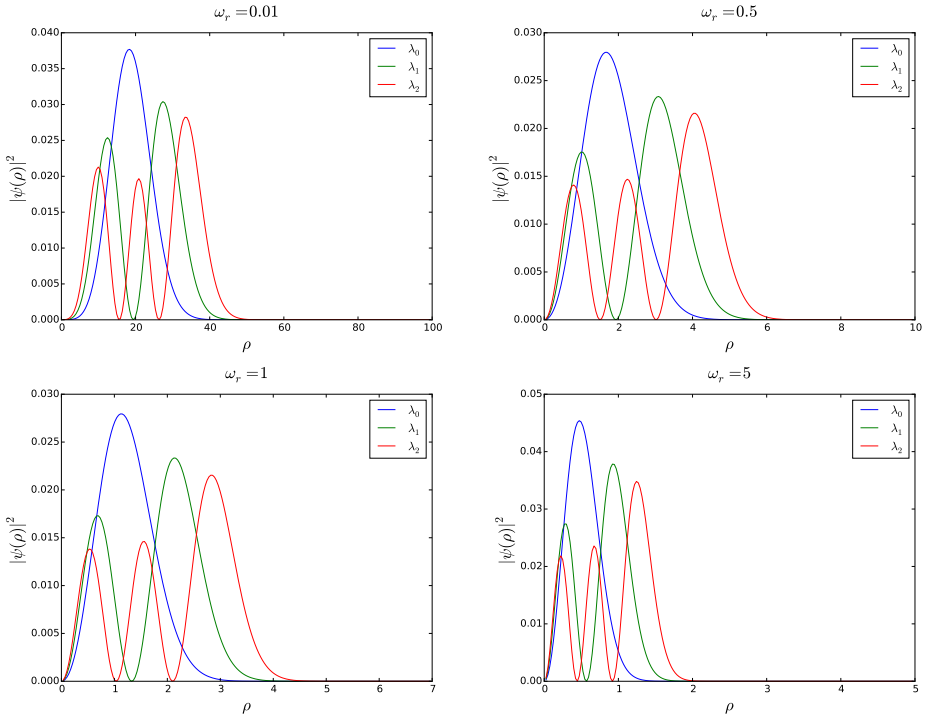


FIGURE 2. We here consider the three lowest eigenstates for each oscillator frequency, with $n = 200$ and an appropriate ρ_{max}. We see that for the various oscillator frequencies the position of the electrons are restricted to "bands" of various radii from the origin. We also notice a small boundary-error in my program.

4. Conclusion

In this project we have examined the behaviour of two electrons with Coloumb-interaction in a harmonic oscillator well. We reformulated the physical problem as a discrete eigenvalue problem and solved this using the *Jacobi Method* for diagonalizing symmetric matrices. We examine our results for various oscillator frequencies, and the results seemed to coincide with the predictions I made with the limited knowledge I have for quantum-physics.

A test-suite was implemented in order to verify and test the methods employed, and I wish that I implemented the test for the non-interacting case earlier as a test as I for a while had a bug in my potential calculatins.

No performance analysis of the algorithms were done, but I will hopefully later have some time to compare this to the *Householder's method* and the routines included in the `armadillo`-library. Since, in our case we were working with a tridiagonal symmetric matrix, we could have written a specialized tridiagonal solver instead which would have improved the efficiency significantly.

References

[1] Morten Hjort-Jensen, *Computational Physics - Lecture Notes Fall 2015*, Department of Physics, University of Oslo, August 2015

[2] M. Taut, *Two electrons in an external oscillator potential: Particular analytic solutions of a Coloumb correlation problem*, Laboratory of Atomic and Solid State Physics, Cornell University, New York, November 1993

[3] Yuusuke Takahashi, Yuusuke Hirota, Yusaku Yamamoto, *Performance of the block Jacobi method for the symmetric eigenvalue problem on a modern massively parallel computer*, Algoritmy 2012

[4] *http://www.gocomics.com/calvinandhobbes/1988/01/06*, Retrieved October 2015