

# FYS4150 - Project 2

Simen Nyhus Bastnes & Eirik Ramsli Hauge

3. October 2016

## Abstract

Wow dude, so this is like, so abstract dude. Have you ever had a dream that you, um, you had, your, you- you could, you'll do, you- you wants, you, you could do so, you- you'll do, you could- you, you want, you want them to do you so much you could do anything?

## Introduction

In project 2, we aim to solve Schrödinger's equation for two electrons with and without a repulsive Coulomb interaction in a three-dimensional harmonic oscillator well. We start by reformulating the equation in a discretized form as an eigenvalue equation, which can be solved with Jacobi's method.

We can then compare our implementation Jacobi's method fares against "smarter" algorithms for solving eigenvalue equations. Finally, we will look at implementing unit-tests, in order to test that the various parts of our program does what it's intended to do.

**Should maybe be rewritten, or changed order, or add more, dunno**

## Theory

### Schrödinger's equation

In this project, we will assume that the electrons move in a three-dimensional harmonic oscillator potential, and repel each other via the static Coulomb interaction. By assuming spherical symmetry, the solution for the radial part of Schrödinger's equation for one electron reads

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

For the rest of the project, we set the orbital momentum quantum number  $l$  to zero. In our non-interacting case, the harmonic oscillator potential  $V(r) = (1/2)kr^2$  with  $k = m\omega^2$ .

We perform a substitution for  $R(r) = (1/r)u(r)$ , introduce a dimensionless variable  $\rho = (1/\alpha)r$ , where  $\alpha$  is a constant with dimension length  $\alpha = (\hbar^2/mk)^{1/4}$ . Inserting this into the Schrödinger equation (1), we get

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

where  $\lambda = (2m\alpha^2/\hbar^2)E$ . Equation (2) is the first equation we want to solve numerically, and we will later use that for  $l = 0$ , the first eigenvalues  $\lambda$  are  $\lambda_0 = 3$ ,  $\lambda_1 = 7$  and  $\lambda_2 = 11$ .

Starting our journey to write equation (2) as an eigenvalue equation, we use the by now standard expression for the second derivative

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

where  $h$  is our step length. We define minimum and maximum values for  $\rho$ ,  $\rho_{\min} = 0$  and  $\rho_{\max}$ , as we cannot set  $\rho_{\max} = \infty$ . Next, we split this interval into  $N$  number of mesh points, so that we can define the step length as

$$h = \frac{\rho_N - \rho_0}{N}$$

where  $\rho_0 = \rho_{\min}$  and  $\rho_{\max} = \rho_N$ . This gives us an expression for  $\rho$  at point  $i$

$$\rho_i = \rho_0 + ih \quad i = 1, 2, \dots, N$$

Now we can rewrite the second derivative as

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

Using this, we can rewrite the dimensionless Schrödinger equation (2) as a discretized equation.

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

where  $V_i = \rho_i^2$  is the harmonic oscillator potential. We can represent the left-hand side as a matrix multiplied by a vector  $u$

### Interacting case

$$V(\rho) = \omega_r^2 \rho^2 + 1/\rho$$

### Jacobi's method

The aim of Jacobi's method is to use similarity transformation to make a  $n \times n$  matrix into a diagonal  $n \times n$  matrix with its eigenvalues as the diagonal. We do this by starting with a matrix  $\mathbf{A}$  which is diagonally dominant **REF TO WIKI!**. Within this matrix we find the largest element that is not on the diagonal, calling its indexes  $k$  and  $l$ . The algorithm is as follows:

---

**Algorithm 1** Maximum non-diagonal element

---

```
1:  $max = 0.0$ 
2: for  $i = 0, n$  do
3:   for  $j = i + 1, n$  do
4:     if  $|A[i][j]| > max$  then
5:        $|A[i][j]| = max$ 
6:        $k = i$ 
7:        $l = j$ 
8:     end if
9:   end for
10: end for
```

---

Now we want to rotate matrix  $\mathbf{A}$  such that the largest off-diagonal element becomes zero. This is done by using equation:

$$\mathbf{B} = \mathbf{S}^T \mathbf{A} \mathbf{S} \quad (3)$$

Where  $\mathbf{S}$  is given as:

$$\begin{pmatrix} 1 & 0 & \dots & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \cos \theta & 0 & \dots & 0 & \sin \theta \\ 0 & 0 & \dots & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & \dots & -\sin \theta & 0 & \dots & 0 & \cos \theta \end{pmatrix}$$

With properties  $\mathbf{S}^T = \mathbf{S}^{-1}$ . The positioning of  $\cos \theta$  and  $\sin \theta$  within the matrix is defined by the index of the largest off-diagonal element like this:

$$s_{kk} = s_{ll} = \cos \theta, \quad s_{kl} = -s_{lk} = \sin \theta, \quad s_{ii} = -s_{ii} = 1 \quad i \neq k, i \neq l$$

This is the same as performing a plane rotation around an angle  $\theta$  in the Euclidean  $n$ -dimensional space.

For  $\mathbf{B}$  this gives us:

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

Where  $\theta$  is set as the angle that makes  $b_{kl} = 0$ . Each transformation can be proven to bring the matrix closer to the diagonal form. [referer til forelesningsnotater](#).

When all this is done, the old maximum off-diagonal value will have become zero and a new one will have risen, but it will not be as large as the old one. We repeat our whole process until the maximum value of all off-diagonal elements are below a certain threshold value. Ideally we would want them all to be zero, but very, very small values is good enough.

By looking at our expression for  $b_{kl} = 0$  and defining

$$t = \tan \theta = \frac{\sin \theta}{\cos \theta}$$

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

We get:

$$(a_{kk} - a_{ll}) \cos \theta \sin \theta + a_{kl}(\cos^2 \theta - \sin^2 \theta) = 0$$

$$(a_{kk} - a_{ll}) \cos^2 \theta t + a_{kl} \cos^2 \theta - a_{kl} \sin^2 \theta = 0$$

$$(a_{kk} - a_{ll})t + a_{kl} - a_{kl} \frac{\sin^2 \theta}{\cos^2 \theta} = 0$$

$$(a_{kk} - a_{ll})t + a_{kl}(1 - t^2) = 0$$

$$\frac{a_{kk} - a_{ll}}{a_{kl}} t = t^2 - 1$$

$$-\tau t = \frac{t^2 - 1}{2}$$

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

Cosine and sinus are easily found to be

$$\cos \theta = \frac{1}{\sqrt{1 + t^2}}, \quad \sin \theta = \cos \theta \cdot t$$

As we can see we can choose two different values for  $t$ . The one we want is the one that is the smallest. This is evident if we look at  $b_{ik}$  and  $b_{il}$ . Hopefully these values will quickly reach values near zero and we want to keep them there. To do this we want  $\cos \theta$  to go to 1 and  $\sin \theta$  to go to 0 simultaneously. By always choosing the lowest  $t$ -value we choose the lowest possible angle and therefore we will minimize the changes to the non-diagonal elements in the matrix. Except for  $b_{kl}$  which is always set to zero.

To fully grasp the idea of this method we have imagined it as this:

Let us pretend that you have ten bars in front of you. These bars can be pushed down, but as you push down one, the others will be affected also. We want to push all the bars as far down as possible, and if you push down the highest bar, the others can't reach it's height, but they are free to go up and down. Then all you have to do is push down one bar (find the maximum non-diagonal element and rotate the matrix) see how the other bars react (view the matrix after rotation), locate the bar that is now the highest and push that one down. As you do this, gradually all the bars will come under a certain height, even though they may not be completely pushed down. When you are satisfied with the shortness of the tallest bar, your job is done and we have found our diagonal matrix with eigenvalues.

Jacobi's algorithm here?

## Experimental?

## Results

We need to make graphs of  $|\Psi(\rho)|^2$  for  $\lambda_1$  etc.

## Unit test

From the three implemented unit tests all three passed.

## Discussion

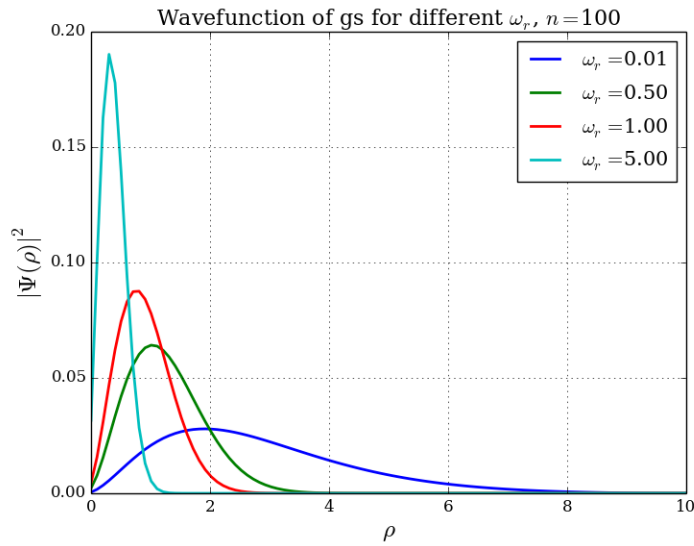


Figure 1: Plot of the eigenvectors for the interacting case for  $n = 100$

## Conclusion

## References

- [1] GitHub Repository  
<https://github.com/simennb/fys4150-project-2>
- [2] Morten Hjorth-Jensen. Project 1. [https://github.com/CompPhysics/ComputationalPhysics/blob/master/doc/Projects/2016/Project1/project1\\_2016.pdf](https://github.com/CompPhysics/ComputationalPhysics/blob/master/doc/Projects/2016/Project1/project1_2016.pdf), 2016.