Project 2.
FYS3150. Computational Physics.

Ana Planes Hernandez.
Nikolai Sigersvold.
Faria Ghaffar.

github.com/fastfirefly

September 30, 2019

1 ABSTRACT

The aim of this project is to solve Schrödinger's equation for one and two electrons in a three dimensional harmonic oscillator well, establishing an algorithm for solving eigenvalue problems by diagonalising a tridiagonal matrix. We will develop an eigenvalue solver based on Jacobi's method, and compare it to other eigenvalue solvers contained in the armadillo library, finding these are more effective than Jacobi's method.

2 INTRODUCTION

Schrödinger's equation is the fundamental equation of physics for describing quantum mechanical behavior, giving information about how the wavefunction of a physical system evolves over time, and containing information about probabilities for the different measurement values. Its importance may be compared to that of Newton's laws in classical mechanics.
In this project, we first consider a simple buckling beam in order to study the two point boundary problem. Then, we will extend this case to quantum mechanics, studying the solution of Schrödinger's equation for the case of one and two electrons in a three dimensional harmonic oscillator well.
We will develop an eigenvalue solver based on Jacobi's method to obtain the eigenvalues and eigenvectors for both cases. Then, we will compare the results obtained using this method to those obtained making use of one of the functions in the armadillo library, in terms of effectiveness and time measurement.

3 THEORY

To study the buckling beam problem, we start with the differential equation

$$\gamma \frac{d^2u(x)}{dx^2} =_F u(x)$$

where $u(x)$ is the vertical displacement, F is the force applied, γ a known constant, and $x\epsilon[0, L]$ where L is the length of the beam.

Given the conditions $u(0) = u(L) = 0$, we define $\rho = x/L$ with $\rho\epsilon[0, 1]$ so that we can scale the equation

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

where the parameter $\lambda = FL^2/R$ is known.

If we consider h to be the stepsize, we can discretise the equation like we did in the previous project so that we have

$$\rho_i = \rho_0 + ih$$

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

which we can write in a more general form, in terms of a matrix.

$$\begin{bmatrix} d & a & 0 & 0 & 0 & ... \\ a & d & a & 0 & 0 & ... \\ 0 & a & d & a & 0 & ... \\ ... & ... & ... & ... & ... & \\ 0 & ... & ... & ... & a & d \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ ... \\ ... \\ u_{n-1} \end{bmatrix} = \lambda \begin{bmatrix} u_1 \\ u_2 \\ ... \\ ... \\ u_{n-1} \end{bmatrix}$$

Defining $d = 2/h^2$ and $a = -1/h^2$, the analytical eigenvalues are given by

$$\lambda_j = d + 2acos(\frac{j\pi}{n+1}) \; j = 1, ..., n$$

Now, we can extend this to **quantum mechanics** in order to study the cases of one and two electrons in a three dimensional harmonic oscillator.

For the case of **one electron** , we start by writing Schrödinger's equation.

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

V(r) is the harmonic oscillator potential $V(r) = (1/2)kr^2$ with $k = m\omega^2$ and E its energy. We can write the energy of the system like

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

with n,l=0,1,2... However, in this project we will only consider l=0.

Substituting $R(r) = u(r)/r$, and bearing in mind the boundary conditions $u(0) = u(\infty) = 0$, we can write

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

Now, if we introduce a dimensionless variable $\rho = (1/\alpha)r$ like we did in the previous case where α is a constant, the potential may be written as $V(\rho) = (1/2)k\alpha^2\rho^2$. Then,

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

where the constant is $\alpha = (h^2/mk)^{1/4}$ and the eigenvalues λ are given by $\lambda = \dfrac{2m\alpha^2}{h^2}E$

We can approximate the second derivative of the equation just like we did before

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

where ρ_i^2 is the potential.

Again, we will write this in a matrix.

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

And define the diagonal matrix elements as

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

and the non-diagonal elements as

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

As we can see, the non-diagonal matrix elements are nothing but a constant, meaning they are all equal.

The Schrödinger's equation takes then the form

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

the analytical values, for the case of one electron for λ are 3,7,11...

Now, for **two electrons**, we can write Schrödinger's equation, assuming, at first, that there is no Coulomb interaction.

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

We introduce the relative coordinates $r = r_1 - r_2$, the center-of-mass coordinate, $R = (r_1 + r_2)/2$, and assume that the wave function can be separated as $u(r, R) = \psi(r)\phi(r)$.

The energy is given by the sum of the relative energy and the center-of-mass energy.

$$E^{(2)} = E_r + E_R$$

Now, we add the repulsive Coulomb interaction between both electrons

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

and proceed in the same way we did for the case of one electron, but we also add a new frequency, defined as

$$\omega_r^2 = \frac{1}{4}\frac{mk}{h^2}\alpha^4$$

and eventually getting to the equation

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

It is clear to see that, the only difference from the one electron case is that now, we need to add $\omega_r^2\rho^2 + 1/\rho$ to the diagonal elements instead of only ρ^2

4 ALGORITHMS

4.1 Unitary Transform

If we consider a basis of vectors,

$$v_i = \begin{bmatrix} v_{i1} \\ ... \\ v_{in} \end{bmatrix}$$

that is orthogonal, $v_j^T v_i = \delta_{ij}$, we can show that a unitary transform $w_i = Uv_i$ preserves the dot product and orthogonality.

$$w_i^T = (Uv_i)^T = U^T v_i^T$$
$$w_i^T w_j = v_i^T U^T U v_i$$

and, since we know that $U^T U = UU^T = I$ and $w_i^T w_j = 1$, then we have:

$$v_i^T v_j = \delta_{ij}$$

4.2 Jacobi's Method

In order to solve the set of equations, we seek to write a function using Jacobi's rotation algorithm.
If we first consider a two dimensional symmetric matrix, to make it more simple,

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

we can write the rotation matrix Q

$$Q = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

so that

$$\begin{bmatrix} b_{11} & 0 \\ 0 & b_{22} \end{bmatrix} = \begin{bmatrix} c & -s \\ s & c \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix} = Q^T A Q$$

where, to simplify, we have written $c = \cos(\theta), s = \sin(\theta)$

Now, we obtain the following equations to be solved

$$b_{11} = a_{11}c^2 - 2a_{12}cs + a_{22}s^2$$
$$b_{22} = a_{22}c^2 + 2a_{12}cs + a_{11}s^2$$
$$b_{12} = b_{21} = (a_{11} - a_{22})cs + a_{12}(c^2 - s^2) = 0$$

Introducing $\tan(\theta) = t = s/c$ and $\tau = \dfrac{a_{22} - a_{11}}{2a_{12}}$

the equations take the form

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

which is immediate to resolve, and we obtain

$$c = \frac{1}{\sqrt{1+t^2}}$$

$$s = ct$$

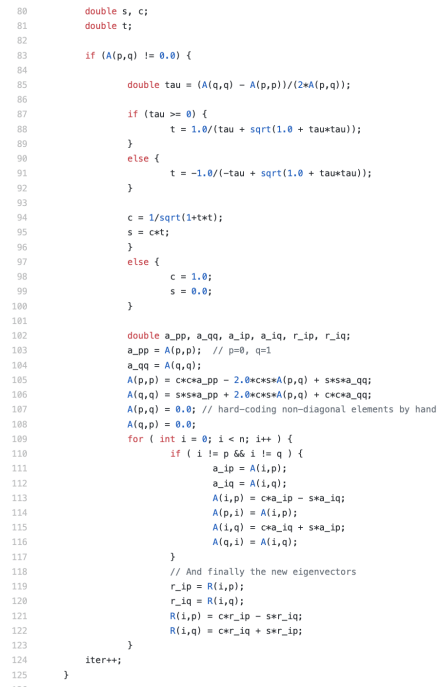
We can easily extend this to a n-dimensional case. The equations we obtain are the following.

$$\begin{gathered} b_{ii} = a_{ii} i \neq k, i \neq l \\ b_{ik} = a_{ik}c - a_{il}si \neq k, i \neq l \\ b_{il} = a_i c + a_{ik}si \neq k, i \neq l \\ b_{kk} = a_{kk}c^2 + 2a_{kl}cs + a_{ll}s^2 \\ b_{ll} = a_{ll}c^2 + 2a_{kl}cs + a_{kk}s^2 \\ b_{kl} = (a_{kk} - a_{ll})cs + a_{kl}(c^2 - s^2) \end{gathered}$$

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

4.3 Implementation

In order to resolve Schrödinger's equation for both the cases of one and two electrons, we will implement Jacobi's method.
We start by defining a tridiagonal matrix, which we work on diagonalising using this algorithm. First of all, we look for the largest off-diagonal element, which is the one we will start with. Once it is found, and we can define the values of τ, c, and s, the program is run within a defined tolerance, trying to make all of the off-diagonal elements equal to zero, ideally.
We also make use of the function $eig\_sym()$ of the armadillo library to resolve the problem, as we will compare the time measurements of both methods to see which one is more effective.



```
        double s, c;
        double t;

        if (A(p,q) != 0.0) {

                double tau = (A(q,q) - A(p,p))/(2*A(p,q));

                if (tau >= 0) {
                        t = 1.0/(tau + sqrt(1.0 + tau*tau));
                }
                else {
                        t = -1.0/(-tau + sqrt(1.0 + tau*tau));
                }

                c = 1/sqrt(1+t*t);
                s = c*t;
                }
                else {
                        c = 1.0;
                        s = 0.0;
                }

                double a_pp, a_qq, a_ip, a_iq, r_ip, r_iq;
                a_pp = A(p,p);  // p=0, q=1
                a_qq = A(q,q);
                A(p,p) = c*c*a_pp - 2.0*c*s*A(p,q) + s*s*a_qq;
                A(q,q) = s*s*a_pp + 2.0*c*s*A(p,q) + c*c*a_qq;
                A(p,q) = 0.0; // hard-coding non-diagonal elements by hand
                A(q,p) = 0.0;
                for ( int i = 0; i < n; i++ ) {
                        if ( i != p && i != q ) {
                                a_ip = A(i,p);
                                a_iq = A(i,q);
                                A(i,p) = c*a_ip - s*a_iq;
                                A(p,i) = A(i,p);
                                A(i,q) = c*a_iq + s*a_ip;
                                A(q,i) = A(i,q);
                        }
                        // And finally the new eigenvectors
                        r_ip = R(i,p);
                        r_iq = R(i,q);
                        R(i,p) = c*r_ip - s*r_iq;
                        R(i,q) = c*r_iq + s*r_ip;
                }
        iter++;
    }
```

Figure 1: Implementation of Jacobi's method

5 RESULTS

5.1 One electron

For the case of one electron, making use of the Jacobi method, we get very similar numerical and analytical results. Most of them differ only by 0.01.

5.2 Two electrons

To see the results in a more graphical way, we include two plots.
In the first one, we represent $|\psi(\rho)|^2$ with respect to ρ. We display two cases: with and without Coulomb interaction. As we can see, when there is Coulomb repulsion, the electrons are further apart, just as it was expected.

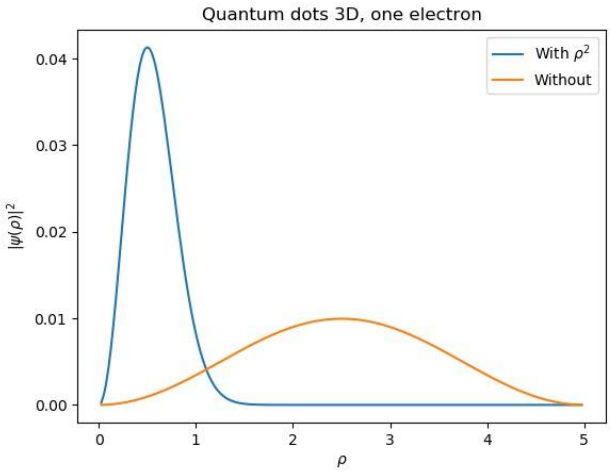


Figure 2: Plot for the behaviour of the system with and without Coulomb interaction.

Next, we represent the case of Coulomb repulsion for different frequencies ω_i. As we can see, the curves get smoother for smaller values of the frequency. This happens due to the fact that for bigger values of ω_i, the electrons are closer together as the well grows stronger and the interaction smaller.

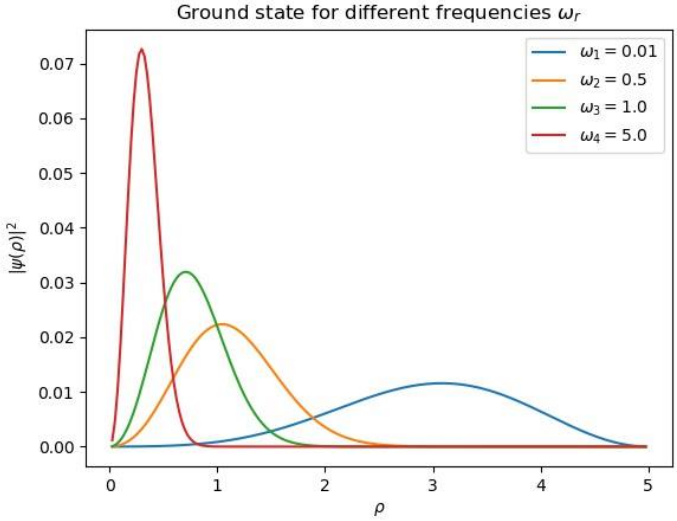


Figure 3: Plot for comparison of different frequencies on the ground state

5.3 Time measurements

In order to compare Jacobi's method with the armadillo function in terms of effectiveness, we represent different plots for the time each one of them takes with respect to a fixed value of integration points.

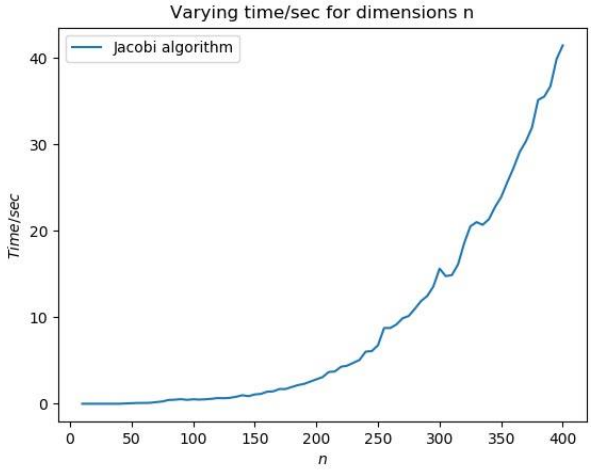


Figure 4: Plot for the time measurement of Jacobi's method.

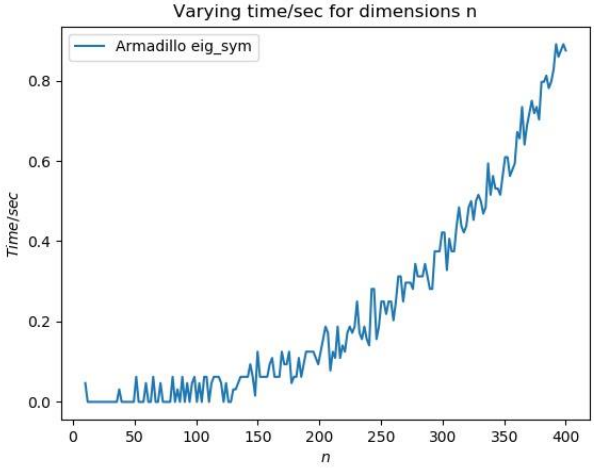


Figure 5: Plot for the time measurement of the armadillo function.

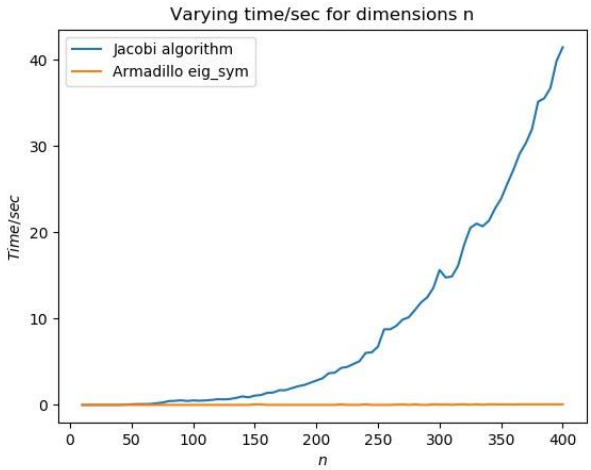


Figure 6: Plot for comparison for Jacobi's method and the armadillo function.

As we can see in figure (6), Jacobi's method takes much more time than the armadillo function to resolve the same problem. The reason for this is that, the convergence of Jacobi's method is significantly poor. In order to zero out the non-diagonal matrix elements, a total of 12 n^3 - 20 n^3 operations are needed.

6 CONCLUSION

Making use of Jacobi's rotation algorithm to find the eigenvalues and eigenvectors for both quantum cases, we have obtained very satisfactory results. However, we have also been able to prove that this is not the best nor the quickest eigenvalue solver, as other functions, in particular the one we have used from the armadillo library, are more effective for solving eigenvalue problems.
We have also implemented practical unit tests in our code, analysing the mathematical properties of our algorithm accurately.

7 REFERENCES

(1) Hjorth-Jensen, M. *Computational Physics. Lecture Notes 2015.* University of Oslo, 2015.
(2) Griffiths, David J. *Introduction to Quantum Mechanics.* Cambridge University Press, 2017.