

[https://github.com/einaur/fys4150/blob/master/project2/main\\_code/main.cpp](https://github.com/einaur/fys4150/blob/master/project2/main_code/main.cpp)  
[https://github.com/einaur/fys4150/blob/master/project2/main\\_code/arma\\_eig\\_solver.cpp](https://github.com/einaur/fys4150/blob/master/project2/main_code/arma_eig_solver.cpp)  
<https://github.com/einaur/fys4150/tree/master/project2/report>

## PROJECT2: EIGENVALUE PROBLEMS, EQUATIONS OF A BUCKLING BEAM, FYS4150: COMPUTATIONAL PHYSICS

SAJJAD AHMADIGOLTAPPEH, EINAR AURBAKKEN  
BASTIAN SKJELSTAD, PER HARALD BARKOST

ABSTRACT. In this project we have solved the three-dimensional Schrödinger equation for two particles with and without a repulsive Coulomb potential in a harmonic oscillator well. By discretizing the equation for one and, later, two electrons we could solve it numerically as an eigenvalue problem using the Jacobi method. By implementing the algorithm, we observe that the numerical solution coincides well with the analytical prediction. Furthermore, the only difference between the cases with and without Coulomb repulsion is the potential term, hence we can conclude that our algorithm can be modified for any potential.

### CONTENTS

1. Introduction	1
2. Theory	1
2.1. Buckling beam problem	1
2.2. Schrödinger equation: radial equation for single electron	3
2.3. Schrodinger equation: two electron in well by Coulomb interaction	4
2.4. Preservation of orthogonality	4
3. Algorithm	5
3.1. Jacobi's method	5
3.2. Algorithm implementation	6
3.3. Making matrix $A$	6
3.4. Diagonalizing	6
4. Results and discussion	7
5. Conclusion	8
References	9
Appendix A. Laguerre Polynomials	9

## 1. INTRODUCTION

The aim of this study is solving an eigenvalue problem such as the buckling beam problem and the three dimensional (3D) Schrödinger equation utilizing Jacobi's method. Actually, the two-point boundary value problem of a buckling beam and the Schrödinger equation could be reformulated in a discretized form, which eventually will lead to a tridiagonal Toeplitz matrix. This matrix has analytical eigenvalues and eigenvectors, and provides a suitable condition for testing the algorithm. Furthermore, equation scaling technique is introduced in this project. In fact, making use of this technique, it is possible to define new dimensionless variables or to introduce suitable units.

## 2. THEORY

Two well-known problems in physics leading to an eigenvalue problem, the *Buckling beam problem* and the *Schrödinger equation*, are discussed in this section.

**2.1. Buckling beam problem.** The buckling beam equation is given as follows:

$$\gamma \frac{d^2 u(x)}{dx^2} = -Fu(x), \quad (1)$$

where  $u(x)$  is the vertical displacement of the beam in the  $y$  direction. The beam has length  $L$ ,  $x \in [0, L]$  and  $F$  is the force applied at  $(L, 0)$  in the direction towards the origin. The parameter  $\gamma$  is a constant defined by properties such as the rigidity of the beam. Dirichlet boundary conditions,  $u(0) = u(L) = 0$ , are present. In the buckling beam equation, all the parameters  $\gamma$ ,  $F$  and  $L$  are known. To implement scaling, a dimensional variable  $\rho$  is introduced as:

$$\rho = \frac{x}{L},$$

where  $\rho \in [0, 1]$ . Then, equation (1) must be reformulated based on the derivative with respect to  $\rho$ . This means that  $x$  must be replaced by  $\rho$ . Considering the following mathematical formula for variable change:

$$\frac{d^2 u}{dx^2} = \frac{d^2 u}{d\rho^2} \left( \frac{d\rho}{dx} \right)^2 + \frac{du}{d\rho} \cdot \frac{d^2 \rho}{dx^2},$$

equation (1) may be rewritten as:

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

where

$$\lambda = FL^2/R.$$

As shown, equation (1) could successfully be rewritten as equation (2) after scaling, and may now be discretized. By discretization of equation (2), it will be converted to an eigenvalue problem. To do so, a Taylor expansion must be used to approximate equation (2) by a linear equation with  $\mathcal{O}(h^2)$ ,

$$u'' = \frac{u(\rho + h) - 2u(\rho) + u(\rho - h)}{h^2} + \mathcal{O}(h^2), \quad (3)$$

where  $h$  is the mesh size. Then, the minimum and maximum values of  $\rho$ ,  $\rho_{\min}$  and  $\rho_{\max}$ , are given:

$$\rho_{\min} = 0 \quad \text{and} \quad \rho_{\max} = 1.$$

By a given number of mesh points,  $N$ ,  $\rho_{\min}$  is given by  $\rho_0$  and  $\rho_{\max}$  is given by  $\rho_N$ , so that  $h$  will be given as

$$h = \frac{\rho_N - \rho_0}{N}. \quad (4)$$

Therefore, the value of  $\rho$  can be calculated at each point of  $i$  as:

$$\rho_i = \rho_0 + ih, \quad \text{where} \quad i = 1, 2, \dots, N. \quad (5)$$

By the definition in equation (5), equation (3) may be rewritten as:

$$-\frac{u(\rho_i + h) - 2u(\rho_i) + u(\rho_i - h)}{h^2} = \lambda u(\rho_i), \quad (6)$$

which leads to:

$$-\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} = \lambda u_i. \quad (7)$$

Illustrating equation (7) in matrix form gives:

$$\begin{bmatrix} d & a & 0 & 0 & \dots & 0 & 0 \\ a & d & a & 0 & \dots & 0 & 0 \\ 0 & a & d & a & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots & a & d & a \\ 0 & \dots & \dots & \dots & \dots & a & d \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \dots \\ u_{N-2} \\ u_{N-1} \end{bmatrix} = \lambda \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \dots \\ u_{N-2} \\ u_{N-1} \end{bmatrix}. \quad (8)$$

It is important to note that the end points,  $u_0$  and  $u_N$ , which are subject to Dirichlet boundary conditions, are not included in equation (8). In this equation, the diagonal elements,  $d$ , and the non-diagonal elements,  $a$ , are defined as

$$d = 2/h^2 \quad \text{and} \quad a = -1/h^2.$$

The eigenvalue problem presented in equation (8) has analytical eigenpairs (consisting of eigenvalues and eigenvectors) given by:

$$\lambda_j = d + 2a \cos\left(\frac{j\pi}{N+1}\right), \quad \text{where} \quad j = 1, 2, \dots, N-1. \quad (9)$$

## 2.2. Schrödinger equation: radial equation for single electron.

The radial part of Schrödinger's equation for a single electron in a harmonic oscillator potential is:

$$-\frac{\hbar^2}{2m}\left(\frac{1}{r^2}\frac{d}{dr}r^2 - \frac{l(l+1)}{r^2}\right)R(r) + V(r)R(r) = ER(r), \quad (10)$$

where the harmonic oscillator potential is  $V(r) = \frac{1}{2}kr^2$ , with  $k = m\omega^2$  and  $E$  being the energy of the electron and  $\omega$  the oscillator frequency. Allowed energy levels are:

$$E_{nl} = \hbar\omega\left(2n + l + \frac{3}{2}\right), \quad (11)$$

where  $n = 0, 1, 2, \dots$  is the energy quantum number and  $l = 0, 1, 2, \dots$  is the orbital momentum quantum number. Assuming  $R(r) = (1/r)u(r)$ , equation (10) is rewritten as

$$-\frac{\hbar^2}{2m}\frac{d^2}{dr^2}u(r) + \left(V(r) + \frac{l(l+1)}{r^2} - \frac{\hbar^2}{2m}\right)u(r) = Eu(r). \quad (12)$$

By defining a dimensionless variable  $\rho = (1/\alpha)r$  equation (12) could be scaled to equation (13), where  $\alpha$  is a constant with length as dimension.

$$-\frac{\hbar^2}{2m\alpha^2}\frac{d^2}{d\rho^2}u(\rho) + \left(V(\rho) + \frac{l(l+1)}{\rho^2} - \frac{\hbar^2}{2m\alpha^2}\right)u(\rho) = Eu(\rho). \quad (13)$$

To simplify the calculation, it is assumed that  $l = 0$ . Plugging  $V(\rho) = \frac{1}{2}k\alpha^2\rho^2$  into equation (12) gives

$$-\frac{\hbar^2}{2m\alpha^2}\frac{d^2}{d\rho^2}u(\rho) + \frac{1}{2}k\alpha^2\rho^2u(\rho) = Eu(\rho), \quad (14)$$

where  $\frac{mk}{\hbar^2}\alpha^4 = 1$  and  $\lambda = \frac{2m\alpha^2}{\hbar^2}E$ . By multiplying both sides of equation (14) by

$$2m\alpha^2\rho^2/\hbar^2,$$

leads to the following equation:

$$-\frac{d^2}{d\rho^2}u(\rho) + \rho^2u(\rho) = \lambda u(\rho). \quad (15)$$

To solve this equation, it should be discretized first. Using the same discretizing technique that was used in the previous section, the Schrödinger equation can be rewritten as:

$$-\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + V_i u_i = \lambda u_i. \quad (16)$$

Further simplification gives:

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

where

$$d_i = \frac{2}{h^2} + V_i \text{ and } e_i = \frac{1}{h^2}.$$

**2.3. Schrodinger equation: two electron in well by Coulomb interaction.** Considering two electrons in a well that are interacting with each other by repulsive Coulomb interaction, gives equation (17) after scaling.

$$-\frac{d^2}{d\rho^2}\psi(\rho) + \frac{1}{4}\frac{mk}{\hbar^2}\alpha^4\rho^2\psi(\rho) + \frac{m\alpha\beta e^2}{\rho\hbar^2}\psi(\rho) = \frac{m\alpha^2}{\hbar^2}E_r\psi(\rho). \quad (17)$$

Defining some of the constants in equation (17),

$$\omega_r^2 = \frac{1}{4}\frac{mk}{\hbar^2}\alpha^4, \quad \frac{m\alpha\beta e^2}{\hbar^2} = 1, \quad \alpha = \frac{\hbar^2}{m\beta e^2} \quad \text{and} \quad \lambda = \frac{m\alpha^2}{\hbar^2}E,$$

allows Schrödinger's equation to be presented as

$$-\frac{d^2}{d\rho^2}\psi(\rho) + \omega_r^2\rho^2\psi(\rho) + \frac{1}{\rho} = \lambda\psi(\rho). \quad (18)$$

The general matrix form which is introduced for the dimensionless equations (14) and (18) is illustrated as:

$$Au = \begin{bmatrix} \frac{2}{\hbar^2} + V_1 & -\frac{1}{\hbar^2} & 0 & \cdots & \cdots & 0 \\ -\frac{1}{\hbar^2} & \frac{2}{\hbar^2} + V_2 & -\frac{1}{\hbar^2} & 0 & \cdots & \cdots \\ 0 & -\frac{1}{\hbar^2} & \frac{2}{\hbar^2} + V_3 & -\frac{1}{\hbar^2} & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & -\frac{1}{\hbar^2} & \frac{2}{\hbar^2} + V_{N-2} & -\frac{1}{\hbar^2} \\ 0 & \cdots & \cdots & \cdots & -\frac{1}{\hbar^2} & \frac{2}{\hbar^2} + V_{N-1} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N-1} \end{bmatrix} = \lambda \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N-1} \end{bmatrix}, \quad (19)$$

containing all the potentials of the harmonic oscillators. In the case where the harmonic oscillator potential doesn't exist, the potential terms will be  $V_i = \rho_i^2$ . However, by considering the repulsive Coulomb potential, the total potential becomes  $V_i = \omega_r\rho_i^2 + 1/\rho_i^2$ , where  $\omega_r$  is the frequency describing the strength of the Coulomb interaction.

**2.4. Preservation of orthogonality.** The basis vectors  $\mathbf{v}_i$  are defined as:

$$\mathbf{v}_i = \begin{bmatrix} v_{i1} \\ \vdots \\ v_{in} \end{bmatrix}, \quad (20)$$

and  $\mathbf{v}_i$  is assumed to be an orthogonal basis,

$$\mathbf{v}_j^T \mathbf{v}_i = \delta_{ij}. \quad (21)$$

It can be shown that for a unitary transformation  $\mathbf{w}_i = U\mathbf{v}_i$ , where  $U$  is a unitary matrix and  $U^T U = I$ , the dot product and orthogonality are preserved in the unitary transformation,

$$\begin{aligned} \mathbf{w}_i^T \mathbf{w}_j &= (U\mathbf{v}_i)^T U\mathbf{v}_j = \mathbf{v}_i^T U^T U\mathbf{v}_j = \mathbf{v}_i^T I\mathbf{v}_j \\ &\rightarrow \mathbf{w}_i^T \mathbf{w}_j = \mathbf{v}_i^T \mathbf{v}_j = \delta_{ij}, \end{aligned} \quad (22)$$

which means that dot product and orthogonality both are preserved. The necessity of this proof will be clarified in the next section. For equations of fifth order or higher there does not exist an analytical solution, however, Abel and Ruffini's theorem provides a solution for these equations utilizing iterative processes. In other words, such equations that do not have analytical solutions may be solved numerically.

### 3. ALGORITHM

For estimation of eigenvalues and eigenvectors of a symmetric matrix such as  $A$ , the Jacobi method was selected as algorithm. Using this technique, matrix  $A$  is transformed to a diagonal matrix which has its eigenvalues on the diagonal.

**3.1. Jacobi's method.** The symmetric matrix  $A$  is defined as:

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}. \quad (23)$$

Since  $A$  is symmetric, it is true that  $a_{12} = a_{21}$ . Furthermore, the rotational matrix  $R$  is defined as

$$R = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}. \quad (24)$$

The outcome of multiplication of  $R$  with any vector such as  $\mathbf{v}$  is the rotation of the vector by an angle  $\theta$ . To simplify the operation,  $c$  and  $s$  are defined as  $c = \cos \theta$  and  $s = \sin \theta$ , respectively. Then, similarity transforms are defined as

$$R^{-1}AR = \begin{bmatrix} a_{11}c^2 + 2a_{12}cs + a_{22}s^2 & (a_{22} - a_{11})cs + a_{12}(c^2 - s^2) \\ (a_{22} - a_{11})cs + a_{12}(c^2 - s^2) & a_{11}s^2 - 2a_{12}s + a_{22}c^2 \end{bmatrix}. \quad (25)$$

It is worth to note that the matrix of equation (14) is a symmetric matrix. The goal of this section is rotating with  $\theta$  so that the off-diagonal elements become zero. Hence we need to solve  $(a_{22} - a_{11})cs + a_{12}(c^2 - s^2) = 0$ , which gives

$$\tau = \cot 2\theta = \frac{a_{11} - a_{22}}{2a_{12}}. \quad (26)$$

It is possible to express all the trigonometric functions in terms of  $\tau$ . Setting  $t = \tan \theta$ ,  $c = \cos \theta = 1/\sqrt{t^2 + 1}$  and  $s = ct$ , yields  $\cot 2\theta = 1/2(\cot \theta - \tan \theta) = 1/2(2^{-1} - t)$ , and finally the following quadratic equation is concluded,

$$t^2 - 2\tau t - 1 = 0, \quad (27)$$

which has two roots. In order to ensure maximum numerical stability of these two roots, the smallest one is picked. Then, at each iteration it is rotated by a small angle.

Then relevant eigenvalues can at this point be expressed in terms of  $t$  as follows:

$$R^{-1}AR = \begin{bmatrix} a_{11} & 0 \\ 0 & a_{22} \end{bmatrix}. \quad (28)$$

Eigenvalues are obtained by solving the characteristic equation, then by an inverse rotation of the identity matrix, eigenvectors can be obtained.

The algorithm for the  $2 \times 2$ -case is a simple example and in the case of real, practical problems, it can be extended to an  $n \times n$ -matrix which is called the Given's rotation matrix:

$$G(i, j, \theta) = \begin{bmatrix} 1 & \dots & 0 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \dots & c & \dots & -s & \dots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \dots & s & \dots & c & \dots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & 0 & \dots & 1 \end{bmatrix}, \quad (29)$$

where  $s = \sin \theta$  and  $c = \cos \theta$ . This matrix represents a rotation in the  $(i, j)$ -plane with angle  $\theta$ . Matrix  $G(k, l, \theta)$  of equation (18) works as  $R$  in equation (17). Therefore, if matrix  $A$  is multiplied by  $G(i, j, \theta)^{-1}$  from the left side and by  $G(k, l, \theta)$  from the right side, where  $a_{kl}$  and  $a_{lk}$  are equal to zero, the following system of equations is produced:

$$\begin{aligned} a'_{hk} &= a'_{kh} = ca_{hk} - sa_{hl}, \\ a'_{hl} &= a'_{lh} = ca_{hl} - sa_{hk}, \\ a'_{kl} &= a'_{lk} = (c^2 - s^2)a_{kl} + sc(a_{kk} - a_{ll}) = 0, \\ a'_{kk} &= c^2 a'_{kk} + s^2 a_{ll} - 2sca_{kl}, \\ a'_{ll} &= s^2 a'_{kk} + c^2 a_{ll} + 2sca_{kl}. \end{aligned} \quad (30)$$

**3.2. Algorithm implementation.** To make use of the algorithm explained above, a matrix  $A$  must be made, then by applying the Jacobi method to construct a diagonalized matrix, and finally after several unit tests, it will be possible to estimate the eigenvectors.

**3.3. Making matrix  $A$ .** The first strategy to solve the current differential equation which is an eigenvalue problem, is to create matrix  $A$ . Matrix  $A$  is an  $n \times n$  matrix, and to be able to construct it, the elements along the diagonal must be filled.

**3.4. Diagonalizing.** The strategy in this section is to utilize symmetry transformations to force the off-diagonal elements of matrix  $A$  to reach the value zero.

#### 4. RESULTS AND DISCUSSION

The code was written in C++. First, the number of mesh points,  $N$ , was specified, and the program was executed to solve equation (16). Initially, the maximum allowed difference between the lowest analytical eigenvalue and the numerical eigenvalue,  $\varepsilon = |\lambda_{analytic} - \lambda_{num}|$ , was set to  $10^{-4}$ . The lowest analytical eigenvalue is 3, and a mesh size of at least  $N = 200$  is required to obtain a sufficiently low value of the tolerance,  $\varepsilon$ . To obtain results with higher precision, the mesh size must be further increased, which in turn results in longer run times for the program.



FIGURE 1. Number of iterations as a function of number of matrix elements for  $\varepsilon = 10^{-8}$ .

By performing such transformations, it is possible to set all the non-diagonal elements to a value lower than  $\varepsilon$ , which approximates to zero. In this program, a counter variable is embedded to check how many times the matrix elements are rotated, the so-called number of iterations. The tolerance was at this point set to  $\varepsilon = 10^{-8}$ , meaning that all



the off-diagonal elements would reach values lower than that of  $\varepsilon$ . For example, calculations were performed for a  $500 \times 500$  matrix (250 000 matrix elements), which resulted in approximately 400 000 iterations. Figure 1 shows the relationship between the number of matrix elements,  $N \times N$ , and the number of iterations. From the plot, it is evident that the relation between the number of iterations and the number of matrix elements is linear while keeping the tolerance,  $\varepsilon$ , constant. Therefore, in this case, if  $\varepsilon$  is kept constant, the number of iterations depends solely on the matrix size.

Calculations done by the Jacobi method were compared to corresponding calculations performed by the Armadillo library in C++. The mesh size,  $N$ , was set to 200 with  $\varepsilon = 10^{-8}$ . The run times were collected for the calculation performed by the Jacobi method and by Armadillo, and can be seen in Table 1.

TABLE 1. Time comparison of the Jacobi method and the Armadillo library in C++

Method	Time [s]
Jacobi algorithm	$4.55 \times 10^6$
Armadillo	74956

As can be seen from Table 1, the run time is significantly longer for the Jacobi algorithm than that of Armadillo.

## 5. CONCLUSION

The solution to the Schrödinger equation for two electrons in an external harmonic oscillator was found using Jacobi's eigenvalue algorithm and the Armadillo library in C++. Compared with Armadillo, the Jacobi method is arguably slower. However, the Jacobi method is able to make the numerical eigenvalues converge to the analytical eigenvalues with great precision. Therefore, the Jacobi method is numerically stable over the working boundaries and potential in question.

## REFERENCES

- [1] Hjorth-Jensen, M., *Project 2, Computational Physics I FYS3150/FYS4150*, University of Oslo, 2016.
- [2] Abel, N.H., Memoire sur le équations algébriques, ou l'on démontre l'impossibilité de la résolution de l'équation générale de cinquième degré, In Sylow, L. and Lie, S., *Æuvres Complètes de Niels Henrik Abel*, 2nd ed., Grøndahl & Søn, pp. 28-33, 1881.

## APPENDIX A. LAGUERRE POLYNOMIALS

Generally, the name Laguerre polynomials is used for solutions to

$$x \frac{d^2 y}{dx^2} + (\alpha + 1 - x) \frac{dy}{dx} + ny = 0. \quad (31)$$

These polynomials, usually denoted,  $L_0$ ,  $L_1$ ,  $L_2$  etc are a polynomial sequence which may be defined by the Rodriguez formula

$$L_n(x) = \frac{e^x}{n!} \frac{d^n}{dx^n} (e^{-x} x^n) = \frac{1}{n!} \left( \frac{d}{dx} - 1 \right) x^n. \quad (32)$$