From the Buckling Beam Problem to Quantum Mechanics: Solving Eigenvalue Problems with Jacobi's Method

Aaby I, Steinnes L, Rashid S.

Institute of Physics, University of Oslo

isakaab@student.matnat.uio.no, lasse.steinnes@fys.uio.no, sidrar@student.matnat.uio.no

September 30, 2020

Abstract

Implementation of Jacobi's method shows a slow algorithm with high demand of CPU power compared to other numerical EVP solvers like Armadillo. Simulations of the method agrees with , where the method converges quadratically. Solutions to the buckling beam problem matches the associated analytical solutions with small relative errors. Approximations to $\rho_{\max}$ in the one-electron problem gives $\rho_{\max} = 10.82$ while numerical eigenvalues differs from the analytical eigenvalues. Two-electron ground state results for $\omega_r = 0.01$, $\omega_r = 0.50$, $\omega_r = 1.00$ and $\omega_r = 5.00$ shows larger ground states for larger ω_r. This agrees with results from, which says that larger ω_r gives increase in electron density and by that larger ground states.

I Introduction

Solving systems of linear equations was first described in Ancient China, before it became known to the outside world in the 17.th century due to Rene Descartes' Cartesian geometry [6]. In its theoretical infancy, a structured way of solving linear systems used determinants - a method credited to Gottfried Leibniz [8]. In the coming centuries, the theoretical framework was expanded upon with Gauss elimination, and set in a rigorous structure. At the beginning of the 20th. century, this new way of describing equations and solving them, emerged as the mathetmatical field known today as linear algebra.

Yet a versatile mathematical tool, linear algebra first reached it full potential with the development of electronic computational devices during and after the Second World War [2]. One area of interest that was heavily influenced by increased computational capacity was eigenvalue and eigenvector calculations. The processes in which they are obtained, are generally more complex than simple linear algebraic operations. In addition they are computationally costly, with $O(n^3)$ for direct methods, creating a bottleneck for matrix dimensionality [4].

Amongst the methods aiming at obtaining these so-called eigenpairs, are direct methods and iterative methods [4]. These methods use similarity transformations to reduce a matrix first to a tridiagonal matrix, and then to diagonal form. In this paper, a focus will be given to Jacobi's method for eigenvalues. It is a direct method for finding eigenvalues.

The aim of this paper is to apply the Jacobi method to solve differential equations in the form of an eigenvalue problem, to obtain its eigenpair-solutions. This paper is laid out by first presenting the theory behind eigenvalue problems and how Jacobi's method offers a strategy at solving them.

Thereafter, the differential equations to solve are introduced, all which are known problems in Physics. They include the buckling beam problem, and two equations describing quantum mechanical systems with one and two electrons respectively. The results are presented with graphs and benchmarks, followed by a discussion and critical evaluation of the Jacobi method and its application to these problems. The paper is ended with some brief concluding remarks.

All programmes of relevance to solving the eigenvalue problems are available at https://github.com/lasse-steinnes/FYS4150-Project2.

II Theory and Methods

The theory section is mainly based upon Morten Hjorth-Jensen's lecture notes in the Computational Physics course (FYS4150) at the University of Oslo [4].

II.I Eigenvalues and Eigenvectors

Eigenvalues and eigenvectors arise from linear transformations. An eigenvector of matrix $\mathbf{A} \in \mathbb{C}^{n\times n}$, is a non-zero vector $\mathbf{x} \in \mathbb{C}^{n\times 1}$, which A maps onto its own vector space with a scaling factor λ. It's scaling factor is the corresponding eigenvalue of A. The eigenvalue equation describe this transformation as

$$\mathbf{A}\mathbf{x}^{(v)} = \lambda^{(v)}\mathbf{x}^{(v)}. \tag{1}$$

The challenge consists of finding the set of scalars $\lambda^{(v)}$ and vectors $\mathbf{x}^{(v)}$ that satisfy eq. 1. Thus, for each eigenvalue $\lambda^{(v)}$ there are n unknown $x_i^{(v)}$'s to be determined.

A common strategy obtaining $\lambda^{(v)}$ and $\mathbf{x}^{(v)}$, is rewriting eq. 1 to a zero left hand side and taking its determinant. An eigenvector $\mathbf{x}^{(v)}$ is non-zero by definition, thus the determinant yields a solution only iff

$$\left|\mathbf{A} - \lambda^{(v)}\mathbf{I}\right| = 0. \tag{2}$$

To satisfy the above equality, λ_i must be the i-th root of the characteristic polynomial

$$P(\lambda) = \prod_{i=1}^{n}(\lambda - \lambda_i). \tag{3}$$

Having found these roots, one can easily find the corresponding $\mathbf{x}^{(i)}$'s. However, if $n \geq 5$ there is no guarantee one can find solutions to these polynomials with radicals, i.e. n-th roots and closed form expressions. This was proven by Niels Henrik Abel in 1824 (Abel-Ruffini impossibility theorem), and further explained by Galois theory with symmetry groups (S_2, S_3 and S_4 solvable).

Hence, other viable methods are preferred when solving large eigenvalue problems. These methods use properties of the EVP, together with similarity transformations, to obtain the eigenvalues and eigenvectors. One of these methods is the Jacobi Algorithm.

II.II Mathematical intermezzo

First of all we want to show that an orthogonal or unitary transformation

$$\mathbf{w}_i = \mathbf{U}\mathbf{v}_i$$

preserves the dot product and orthogonality. Let us consider an orthogonal basis of vectors $\mathbf{v}_i$,

$$\mathbf{v}_i = \begin{bmatrix} v_{i1} \\ \dots \\ \dots \\ v_{in} \end{bmatrix}$$

where orthogonality is noted as

$$\mathbf{v}_j^T \mathbf{v}_i = \delta_{ij}$$

For a matrix U to be unitary it must have the property that $U^T = U^{-1}$. Computing the dot product yields

$$\mathbf{w}_j^T \mathbf{w}_i = \mathbf{U}\mathbf{v}_j^T \mathbf{U}\mathbf{v}_i = \mathbf{v}_j^T (\mathbf{U}^T \mathbf{U})\mathbf{v}_i = \mathbf{v}_j^T (\mathbf{U}_1^- \mathbf{U})\mathbf{v}_i = \mathbf{v}_j^T \mathbf{v}_i = \delta_{ij}$$

and thus we have proved that a unitary transformation preserves the dot product and the orthogonality of the obtained eigenvectors.

II.III Jacobi Method

Jacobi's algorithm is a direct method for solving EVPs. In constrast to the formal method described in sec. II.I, the matrix $\mathbf{A}$ is changed to another matrix $\mathbf{B}$, in a series of transformations. The diagonal matrix $\mathbf{D} = (\lambda_1, \lambda_2, ...\lambda_n)$, is the target, so that a matrix $\mathbf{S}$ transforms $\mathbf{A} \mapsto \mathbf{B} \mapsto \mathbf{D}$. In the transformation, λ_i's of $\mathbf{A}$ are conserved, but its basis changes. As a result, eigenvectors of $\mathbf{A}$ are altered in in the process.

If $\mathbf{A} \in \mathbb{R}^{n\times n}$ is symmetric, there exists orthogonal matrices $\mathbf{S}_\mathrm{i} \in \mathbb{R}^{n\times n}$ ($\mathbf{S}_\mathrm{i}^T = \mathbf{S}_\mathrm{i}^{-1}$), so that

$$\mathbf{S}_m^T \mathbf{S}_{m-1}^T ... \mathbf{S}_1^T \mathbf{A} \mathbf{S}_1 ... \mathbf{S}_{m-1} \mathbf{S}_m = \mathbf{D}. \tag{4}$$

One transformation performed on eq. 1 yields

$$\mathbf{S}^T \mathbf{A}\mathbf{S}(\mathbf{S}^T \mathbf{x}) = \mathbf{B}(\mathbf{S}^T \mathbf{x}) = \lambda (\mathbf{S}^T \mathbf{x}), \tag{5}$$

in which $\mathbf{S}\mathbf{S}^T = \mathbf{I}$. Hence, the eigenpairs of $\mathbf{B} \in \mathbb{R}^{n\times n}$ are the λ's and $\mathbf{S}^T\mathbf{x}$ vectors satisfying eq. 5.

In the algorithm, a rotation matrix is used, where each set of row and column vectors forms an orthonormal basis for the Euclidean space $\mathbb{R}^n$. In a rotation, the inner product is conserved, hence it is by definition a orthogonal matrix. A transformed vector is rotated in the Euclidean n-dimensional space by a planar angle θ. Thus, the only non-zero elements of $\mathbf{S}$ are

$$S_{kk} = S_{ll} = \cos\theta, \tag{6}$$

$$S_{kl} = -S_{lk} = \sin\theta, \tag{7}$$

and

$$S_{ii} = 1, \tag{8}$$

for i $\neq$ k, i $\neq$ l. The challenge is to choose θ so that non-diagonal elements of **B** goes to zero, $B_{lk} = 0$. Fortunately, for each orthogonal transformation, the Frobenius norm

$$\text{norm}(\mathbf{A})_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} |A_{ji}|^2} \tag{9}$$

is conserved. It can be shown that the off-diagonal Frobenius norm of B, off(**B**), becomes

$$\text{off}(\mathbf{B})^2 = \text{off}(\mathbf{A})^2 - 2A_{kl}^2, \tag{10}$$

Thus, for each transformation, matrix **B** has off-diagonals closer to zero. The following parametrization ensures minimization of the difference between **B** and **A**,

$$\tan\theta = t \tag{11}$$

$$\cot 2\theta = \tau = \frac{A_{ll} - A_{kk}}{2A_{kl}}, \tag{12}$$

$$t = -\tau \pm \sqrt{1+\tau^2} = \frac{1}{\tau \pm \sqrt{1+\tau^2}}. \tag{13}$$

So that $\cos\theta = c = 1/\sqrt{1+t^2}$ and $\sin\theta = s = t \cdot c$. Thus, a direct calculation of s and c through θ is never neeeded in the algorirthm. In eq. 13, the last equality is rewritten by multiplication and division by $\tau \pm \sqrt{1+\tau^2}$ to avoid round-off error. The algorithm to rotate the elements of A and obtain the diagonal matrix is described below (algo. 1).

A presentation of the problems which the Jacobi method is applied to in this paper is given in the following subsections (sec. II.IV-II.VII).

II.IV Solving the buckling beam problem

Now we want to solve a specific problem using Jacobi's method. We have the following differential equation

$$\gamma \frac{d^2u(x)}{dx^2} = -Fu(x), \tag{14}$$

where $u(x)$ is the vertical displacement of a beam with length L. F is a force applied in the direction towards the origin and γ is a constant defined by the rigidity of the beam. In our case F and L are known, while γ is a unknown quantity.

Algorithm 1 Jacobi Algorithm: The basic outline of Jacobi's algorithm for solving an eigenvalue problem by finding the diagonal matrix $\mathbf{D} = (\lambda_1, ...\lambda_n)$.

$A_{kl} = \max_{i \neq j} |A_{ij}|$ ▷ Choose maximum non-diag element
while $A_{kl} \geq \varepsilon$ **do** ▷ Check if non-diag is larger than tolerance
  $\tau = (A_{ll} - A_{kk})/2A_{kl}$ ▷ Calculate rotation parameters
  $t = 1/(\tau \pm \sqrt{1+\tau^2})$
  $c = 1/\sqrt{1+t^2}$
  $s = t \cdot c$
  procedure ROTATE(**A**) ▷ Use the orthogonal transformation
    for int i=0; i< n;i++ **do**
      if i $\neq$ k & i $\neq$ l **then**
        $B_{ik} = A_{ik}c - A_{il}s$
        $B_{il} = A_{il}c + A_{ik}s$
      end if
      $B_{kk} = A_{kk}c^2 - 2A_{kl}c \cdot s + A_{ll}s^2$
      $B_{ll} = A_{ll}c^2 + 2A_{kl}c \cdot s + A_{kk}s^2$
      $B_{kl} = 0$ ▷ $t(\theta)$ chosen so that non-diagonal set to 0
    end for
  end procedure
  A $\Leftarrow$ **B** ▷ Update Matrix
end while

We start by defining a dimensionless variable $\rho = x/L$ and get

$$\frac{d^2u(\rho)}{d\rho^2} = -\lambda u(\rho), \qquad \lambda = FL^2/\gamma, \qquad u(\rho_{\min}) = u(\rho_{\max}) = 0. \tag{15}$$

where we set $\rho_{\min} = \rho_0 = 0$ and $\rho_{\max} = \rho_N = 1$. Solving this eigenvalue problem will give us the eigenvalues λ, and by that we can solve for the unknown quantity γ. Having also applied the Dirichlet boundary conditions, we approximate the second derivative

$$-\frac{u(\rho_i + h) - 2u(\rho_i) + u(\rho_i - h)}{h^2} = \lambda u(\rho_i), \qquad \rho_i = \rho_0 + ih, \quad h = \frac{\rho_N - \rho_0}{N} \tag{16}$$

to get a discretized equation which in general gives us the following eigenvalue problem

$$\begin{bmatrix} d & a & 0 & 0 & \dots & 0 & 0 \\ a & d & a & 0 & \dots & 0 & 0 \\ 0 & a & d & a & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots & a & d & a \\ 0 & \dots & \dots & \dots & \dots & a & d \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \dots \\ u_{N-2} \\ u_{N-1} \end{bmatrix} = \lambda \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \dots \\ u_{N-2} \\ u_{N-1} \end{bmatrix} \tag{17}$$

where $d = 2/h^2$ and $a = -1/h^2$. We solve equation (17) by implementing Jacobi's algorithm (1) to find the eigenvalues λ and eigenvectors $u(\rho_i)$. This problem also has analytical solutions with eigenvalues and eigenvectors given by

$$\lambda_j = d + 2a\cos\left(\frac{j\pi}{N}\right), \quad j = 1,2,...N-1 \tag{18}$$

$$\mathbf{u}_j = [\sin\left(\frac{j\pi}{N}\right), \sin\left(\frac{2j\pi}{N}\right), ..., \sin\left(\frac{(N-1)j\pi}{N}\right)]^T, \quad j = 1,2,\ldots N-1. \tag{19}$$

. We therefore also want to compare our results with the analytical results.

II.V Electrons in a harmonic oscillator potential

When assuming a three-dimensional harmonic oscillator potential with spherical symmetry, electrons move in a way so that they repel each other via the static Coulomb interaction. When using spherical coordinates, one can show that the radial part of Schrödinger's equation with one electron is given by

$$-\frac{\hbar^2}{2m}\left(\frac{1}{r^2}\frac{d}{dr}r^2\frac{d}{dr} - \frac{l(l+1)}{r^2}\right)R(r) + V(r)R(r) = ER(r). \tag{20}$$

where E is the energy of the harmonic oscillator in three dimensions, and $V(r) = (1/2)kr^2$ is the harmonic oscillator potential with the constant $k = m\omega^2$. The constant ω is the oscillator frequency. It can also be shown that the different energy levels for an electron in a harmonical oscillator is given by

$$E_{nl} = \hbar\omega(2n + l + 3/2) \tag{21}$$

In both equations (20) and (21), we have $n = 0,1,2,...$ and $l = 0,1,2,....$ The quantum number l is the orbital momentum of a given electron.

A classical problem when having Schrödinger's equation, is to find the energies from equation (21) and the wavefunction $R(r)$. This can be done by standard integration to solve the differential equation. Another way of doing this is to approximate the derivative and use Jacobi's method (1).

II.VI Solving radial part of Schrödinger's equation for one electron

Now we want to solve the radial part of Scrödinger's equation by applying Jacobi's method. Having equation (20) we substitute $R(r) = (1/r)u(r)$ and introduce a dimensionless variable $\rho = r/\alpha$ where α is a constant with dimension length. In our case we

are only going to study the equation for $l = 0$ which gives

$$-\frac{\hbar^2}{2m\alpha^2}\frac{\mathrm{d}^2}{\mathrm{d}\rho^2}u(\rho)+V(\rho)u(\rho)=Eu(\rho). \tag{22}$$

Inserting the modified potential $V(\rho) = (1/2)k\alpha^2\rho^2$ and reorganize the equation we have

$$-\frac{\mathrm{d}^2}{\mathrm{d}\rho^2}u(\rho)+\frac{mk}{\hbar^2}\alpha^4\rho^2u(\rho)=\frac{2m\alpha^2}{\hbar^2}Eu(\rho). \tag{23}$$

By fixing the constant in front of the second term to 1, the variable α can be determined

$$\frac{mk}{\hbar^2}\alpha^4=1 \quad \Rightarrow \quad \alpha=\left(\frac{\hbar^2}{mk}\right)^{1/4}, \qquad \lambda=\frac{2m\alpha^2}{\hbar^2}E \tag{24}$$

where we have also defined λ to get

$$-\frac{\mathrm{d}^2}{\mathrm{d}\rho^2}u(\rho)+\rho^2u(\rho)=\lambda u(\rho). \tag{25}$$

By approximating the second derivative, in the same way as we did with the Buckling beam problem, we solve equation (25) with Jacobi's algorithm (1) to find the eigenvalues λ and eigenstates $u(\rho)$. The only difference from the buckling beam problem is that we had to add the potential ρ^2 to the main diagonal elements in the matrix (17).

Fortunately for this case, we know the analytical scaled one-electron energies λ which are given by

$$\lambda = 3, 7, 11, 15, 19, ... \tag{26}$$

Having the analytical results, we approximate $\rho_{\max}$ to get our calculated energy eigenvalues close to these results.

II.VII Solving radial part of Schrödinger's equation for two electrons

Having schödinger's equation (22) for one electron, the equation for two electrons with no Coulomb interaction i given by

$$\left(-\frac{\hbar r^2}{2m}\frac{d^2}{dr_1^2}-\frac{\hbar^2}{2m}\frac{d^2}{dr_2^2}+\frac{1}{2}kr_1^2+\frac{1}{2}kr_2^2\right)u(r_1,r_2)=E^{(2)}u(r_1,r_2). \tag{27}$$

where $u(r_1, r_2)$ and $E^{(2)}u(r_1, r_2)$ is the two-electron wavefunction and and energy. We define the vector $\mathbf{r} = \mathbf{r_1} - \mathbf{r_2}$ and center-of-mass coordinate $\mathbf{R} = 1/2(\mathbf{r_1} + \mathbf{r_2})$ which gives

$$\left(-\frac{\hbar^2}{m}\frac{d^2}{dr^2}-\frac{\hbar^2}{4m}\frac{d^2}{dR^2}+\frac{1}{4}kr^2+kR^2\right)u(r,R)=E^{(2)}u(r,R), \qquad E^{(2)}=E_r+E_R \tag{28}$$

where the wavefunction $u(r,R) = \psi(r)\phi(R)$ is given by the ansatz where we have seperated the equations for r and R. The total energy is given by the sum of relative energy E_r and the center-of-mass energy E_R. Adding the repulsive Coulomb interaction between the electrons we get

$$\left(-\frac{\hbar^2}{m}\frac{d^2}{dr^2} + \frac{1}{4}kr^2 + V(r_1,r_2)\right)\psi(r) = E_r\psi(r), \qquad V(r_1,r_2) = \frac{\beta e^2}{|\mathbf{r_1}-\mathbf{r_2}|} = \frac{\beta e^2}{r} \tag{29}$$

where $\beta e^2 = 1.44\,\text{eVnm}$. Introducing the same dimensionless variable $\rho = r/\alpha$

$$-\frac{d^2}{d\rho^2}\psi(\rho) + \frac{1}{4}\frac{mk}{\hbar^2}\alpha^4\rho^2\psi(\rho) + \frac{m\alpha\beta e^2}{\rho\hbar^2}\psi(\rho) = \frac{m\alpha^2}{\hbar^2}E_r\psi(\rho). \tag{30}$$

and defining the variables

$$\omega_r^2 = \frac{1}{4}\frac{mk}{\hbar^2}\alpha^4, \qquad \frac{m\alpha\beta e^2}{\hbar^2} = 1 \quad \Rightarrow \quad \alpha = \frac{\hbar}{m\beta e^2}, \qquad \lambda = \frac{m\alpha^2}{\hbar^2}E \tag{31}$$

where ω_r is a new frequency. This gives the following equation

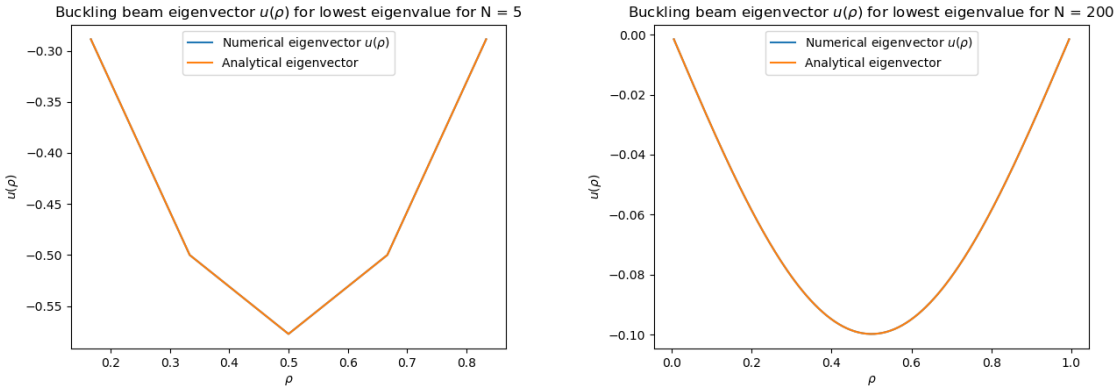
$$-\frac{d^2}{d\rho^2}\psi(\rho) + \omega_r^2\rho^2\psi(\rho) + \frac{1}{\rho}\psi(\rho) = \lambda\psi(\rho). \tag{32}$$

By adding the potential $\omega_r^2\rho^2 + 1/\rho$ to the main diagonal elements in the matrix (17), we solved equation (32) using Jacobi's method (1) for different values of $\omega_r = 0.01$, $\omega_r = 0.5$, $\omega_r = 1$ and $\omega_r = 5$.

III Results

Table 1: **Relative error and CPU comparison**: Relative error for the calculated buckling beam eigenvalues relative to the analytical eigenvalues from (18), for $n = 10, 50, 200$ and 250. In addition to this, the table also shows comparable CPU times for Jacobi's method and Armadillo's eig sym method for the same number of grid points. Armadillo's eig sym method was gathered from [7].

n	Relative error	CPU time (ms) (Jacobi)	CPU time (ms) (Armadillo)
10	18.580	0.615	3
50	2.902	63	2
200	0.0138	8505	7
250	0.064	57754	24



(a) n = 5 (b) n = 200

Figure 1: The ground state for the buckling beam problem in comparison with the analytical ground state from (19) (when inserting $j = 0$), for both $N = 5$ and $N = 200$. Both numerical and analytical ground states are normalized, and the numerical solution was calculated using $\rho_{\max} = 1$.

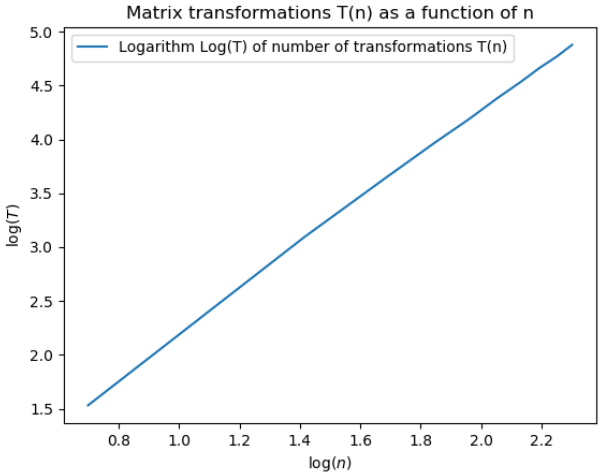
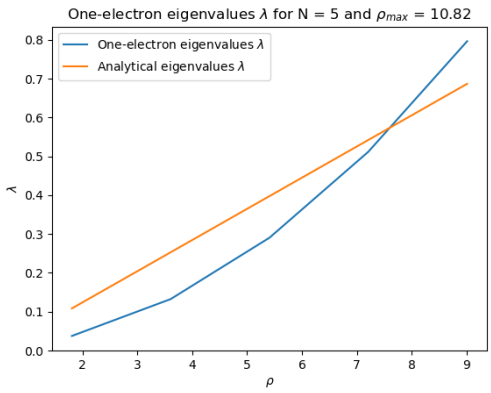
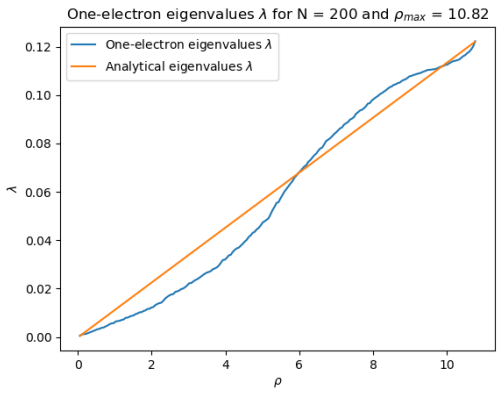


Figure 2: A logarithmic plot showing number of matrix transformations $T(n)$ from Jacobi's method (1) as a function of n. Using polyfit [1] from numpy, the calculated slope number of $\log(T)$ was $a \approx 2.190$.

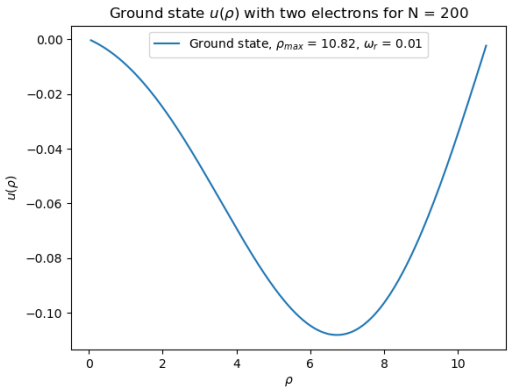


(a) n = 5

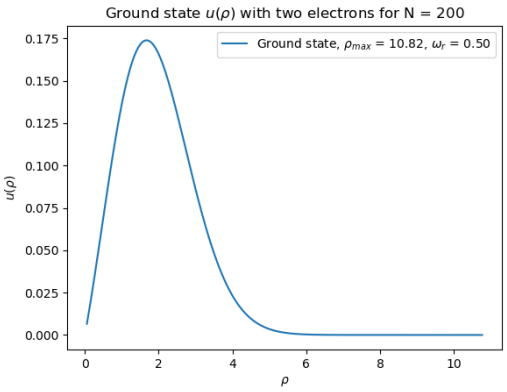


(b) n = 200

Figure 3: Comparing numerical and analytical (26) one-electron scaled eigenvalues λ for $N = 5$ and $N = 200$. After optimizing $\rho_{\max}$ for the one-electron problem we received the lowest maximal absolute error for eigenvalues using $\rho_{\max} \approx 10.82$.

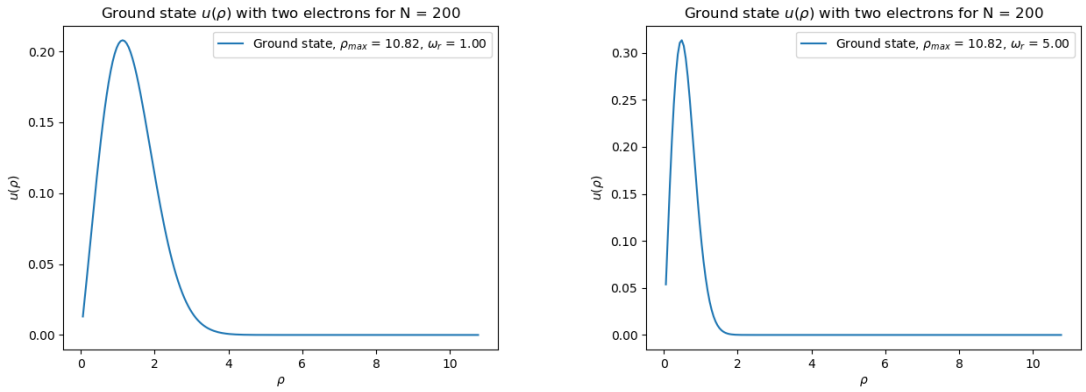


(a) $\omega_r = 0.01$



(b) $\omega_r = 0.50$

Figure 4: Numerical results for the two-electron ground state $u(\rho)$, for oscillating frequencies $\omega_r = 0.01$ and $\omega_r = 0.5$. Both numerical solutions are normalized and calculated using $N = 200$.



(a) $\omega_r = 1.00$ (b) $\omega_r = 5.00$

Figure 5: Numerical results for the two-electron ground state $u(\rho)$, for oscillating frequencies $\omega_r = 1.00$ and $\omega_r = 5.00$. Both numerical solutions are normalized and calculated using $N = 200$.

IV Discussion

As we wanted to point out in the introduction to this article, there exists several numerical methods for solving EVP's, and some of them are much faster compared to others. Jacobi's method performs a series of linear matrix transformations and the amount of CPU-time needed increases rapidly, as n increases. This is shown in Tab.(1), where the amount of CPU time for Jacobi increases drastically, while Armadillo's solver from [7] is much faster, with only 24 ms for $N = 250$. As expected the relative error between the numerical and analytical buckling beam eigenvalues decreases for larger n, but eventually it starts to increase again for $n > 200$. One could argue that this is is due to a small steplength h and thereby round of errors during the simulations.

The book from Gene H.Golub and Charles F.Van Loan [3] argues that the number of linear matrix transformations in Jacobi's method converges quadratically. If we look at Fig.(2), one can see that the logarithm of number of transformations is a linear function of $\log(n)$, and we found that the slope number was $a \approx 2.190$. This agrees with [3], having quadratic convergence.

The comparison between the buckling beam numerical and analytical ground states in Fig.(1) shows very small variations even for a small $N = 5$. This suggests that the Jacobi algorithm (1) has been implemented correctly relative to the analytical solution. In some contrast, the normalized comparison of numerical and analytical one-electron eigenvalues in Fig.(3), shows a larger unexpected deviation. This might be because we only opti-

mzed $\rho_{\max}$ without actually optimizing n, and we are then left with a less accurate result.

For the two-electron problem we studied the ground states $u(\rho)$ for different values of ω_r. Using the optimized $\rho_{\max} = 10.82$ and $N = 200$ we received Fig.(4) for $\omega_r = 0.01$ and $\omega_r = 0.50$, and Fig.(5) for $\omega_r = 1.00$ and $\omega_r = 5.00$. The different figures show that for larger values of oscillating frequency ω_r, the peak of the ground states increases. This looks reasonable when comparing with the article from M. Taut, Phys. Rev. A 48, 3561 (1993) [5], which argues that the electron density increases with higher ω_r. This is something we would expect, since the ground state of two electrons describes the distribution of the electrons for the lowest energy λ.

An important thing to note is that we are dealing with scaled numerical solutions for the different eigenvalue problems. To study the original eigenstates and eigenvalues, one should re-scale the solutions back, using the definitions of α and λ for the different EVP's.

V Conclusion

We have now studied and solved common physics related eigenvalue problems by implementing Jacobi's algorithm. The implementation of the method shows that it very quickly requires a lot of CPU time when n increases. However the algorithm is fairly easy to implement and teaches you how to transform a typical linear algebra problem into a numerical solver for EVP's. The results for the buckling beam problem, shows very small deviations from analytical solutions (19) and our implementation agrees with [3], having quadratic convergence.

We found that the best approximation to $\rho_{\max}$ in the one-electron problem was $\rho_{\max} = 10.82$. However the lack of optimizing both $\rho_{\max}$ and n might be the reason for having so large deviations from analytical eigenvalues (26). The numerical ground state solutions for the two-electron problem with different ω_r agrees with [5]. The agreement implies that larger oscillating frequency ω_r gives rise to the electron density and thereby larger ground states.

References

[1] Numpy library polyfit. Accessed: September 2020.

[2] Gene H. Golub and Henk A. Eigenvalue computation in the 20th century. *Journal of Computational and Applied Mathematics*, pages 35–65, 2000.

[3] Gene H. Golub; Charles F. Van Loan;. *Matrix Computations*. The Johns Hopkins University Press, 4 edition, 2013.

[4] Morten Hjorth-Jensen. Computational Physics Lectures: Eigenvalue problems, 2020. Accessed: September 2020.

[5] M.Taut. Two electrons in an external oscillator potential: Particular analytic solutions of a coulomb correlation problem. 1993.

[6] John Stillway. *Analytic Geometry. Mathematics and its History*. Springer, 2004.

[7] The SciPy community. Armadillo library eig sym . Accessed: September 2020.

[8] University of Oregon. A Brief History of Linear Algebra and Matrix Theory. Accessed: September 2020.