Jacobi's Method for Solving Differential Equations

Andrea Jensen Marthinussen, Kamilla Ida Julie Sulebakk
Hedvig Borgen Reiersrud

(Dated: October 1, 2020)

In this project we use Jacobi's method in order to solve three second order differential equations. The first equation is a classical problem, known as the buckling beam problem. The latter two are Schrödinger equations describing systems affected by an harmonic oscillator potential, with one and two electrons respectively. For the buckling beam problem we will see that the numerically computed eigenvalues complies with the analytical ones, and we will further observe that the number of iterations needed to diagonalize a matrix of size $n \times n$ increases for increased values of n. We will also see that the implemented unit tests in our code gives reassurance of our numerical results.

For the quantum mechanical cases, we will see how the numerical deviation from analytical eigenvalues of a matrix varies for different number of grid points n as well as for different values of the parameter $\rho_{\max}$, and we will find that for $n = 100$, it is the value $\rho_{\max} = 3.6$ which gives the most accurate result, only differing from the analytical result by 0.0004. For the quantum mechanical problem with two electrons it will be noted that the probability of finding the electrons within a smaller distance from each other increases for increased value of the parameter ω_r.

We will also observe that the algorithm has problems producing precise eigenvalues when we let the interval of ρ be large, which is going to affect our results for both the buckling beam problem as well as for the quantum mechanical case with one electron. We will that this is of less importance in the quantum mechanical case with two electrons as we are then only interested in the ground state, i.e. only the first eigenvalue. Finally we are going to discuss the limitations of numerical computing in the sense that we are unable to let $\rho_{\max} \to \infty$.

I. INTRODUCTION

Eigenvalue problems are occurring in a myriad of different contexts. Moreover, differential equations are of substantial importance in both classical and modern physics. Some differential equations can be written as an eigenvalue equation, and can be solved using properties of linear algebra. Here we will investigate differential equations in both classical physics with the buckling beam problem, as well as in two different quantum mechanical problems. Interestingly, we will use almost identical equations to solve the physical problems.

In quantum mechanics, eigenvalue problems play a stellar role in the theory. Here we will investigate the quantum dot (QD) which is a central topic in nanotechnology and has a multitude of potential applications, such as in biology, solar cell engineering and light emitting diodes (LEDs). We will study QDs for both one and two electrons in three dimensions. The electrons are effected by an harmonic oscillator potential. For the simulation with two electrons there is an additional potential, as the electrons are interacting with each other by a repulsive Coulomb potential.

In order to solve differential equations with linear algebra, we need to discretize the continuous functions, such that we end up with an eigenvalue problem. The solutions can be obtained using different numerical tools. Here, we will use Jacobi's algorithm for solving the eigenvalue problem. The method is much more effective than Cramer's rule, as it uses less floating point operations. Jacobi's method is an iterative method which diagonalizes matrices. The end result is a diagonal matrix where the diagonal elements are the eigenvalues. We will discuss the functionality of the algorithm in the physical problems we investigate.

In addition, we will discuss the accuracy of our results. This we can do by comparing the eigenvalues computed with Jacobi's method, with the given analytical eigenvalues. It is interesting to find the difference between them, to explore the numerical error sources, such as catastrophic cancellation.

It is vital to write clean code as this let us test our algorithms easily by so-called unit test. Unit-testing is broadly used in software development, and is essential to check if computer programs work as they should. Our tests study whether or not our algorithm provides reliable results.

II. ALGORITHMS

A. Jacobi's eigenvalue algorithm

Jacobi's method let us diagonalize matrices. The end result after N number of iterations is a diagonal matrix where the diagonal elements corresponds to the eigenvalues. The method involves a unitary matrix $\mathbf{S}$ which preforms a series of plane rotations.

We consider an orthonormal basis,

$$\mathbf{v}_i = \begin{bmatrix} v_{i1} \\ v_{i2} \\ . \\ . \\ . \\ v_{iN} \end{bmatrix}. \tag{1}$$

Where each element is $\in \mathbb{R}$, and its inner product is $\mathbf{v}_j \cdot \mathbf{v}_i = \delta_{ji}$.

After the unitary transformation we end up with a new basis set $\mathbf{w}_i$, such that $\mathbf{w}_i = \mathbf{U}\mathbf{v}_i$, where $\mathbf{U}$ is an unitary matrix. We can now show that the inner product and the orthogonality of $\mathbf{v}_i$ is preserved after the unitary transformation,

$$\mathbf{w}_j^T \mathbf{w}_i = \mathbf{w}_j^T \mathbf{U}\mathbf{v}_i = \mathbf{v}_i^T \mathbf{U}^T \mathbf{U}\mathbf{v}_i.$$

Since $\mathbf{U}$ is a unitary matrix, $\mathbf{U}^T\mathbf{U} = \mathbf{I}$ where $\mathbf{I}$ is the identity matrix. This gives,

$$\mathbf{w}_j^T \mathbf{w}_i = \mathbf{v}_j^T \mathbf{I}\mathbf{v}_i = \mathbf{v}_j^T \mathbf{v}_i = \delta_{jm}. \tag{2}$$

Thus the inner product and orthogonality is preserved.

The unitary matrix in Jacobi's method to diagonalize matrices is defined as

$$\mathbf{S} = \begin{bmatrix} 1 & 0 & ... & ... & ... & ... & ... & ... \\ 0 & 1 & ... & ... & ... & ... & ... & ... \\ ... & ... & ... & ... & ... & ... & ... & ... \\ 0 & ... & ... & \cos\theta & 0 & ... & 0 & \sin\theta \\ 0 & ... & ... & 0 & 1 & 0 & ... & 0 \\ ... & ... & ... & ... & ... & ... & ... & ... \\ 0 & 0 & ... & ... & ... & ... & 1 & 0 \\ 0 & 0 & ... & -\sin\theta & 0 & ... & 0 & \cos\theta \end{bmatrix}. \tag{3}$$

The non-zero elements in $\mathbf{S}$ is,

$$s_{kk} = s_{ll} = \cos(\theta), s_{kl} = -s_{lk} = \sin(\theta), s_{ii} = 1 \tag{4}$$

$$k, j \neq i. \tag{5}$$

Because $\mathbf{S}$ is unitary, the matrix has the property $\mathbf{S}^T = \mathbf{S}^{-1}$, thus $\mathbf{S}$ is orthonormal (or so called unitary). Since Jacobi's algorithm uses a series of orthogonal transformations, we know that the Frobenius norm is always preserved. It reads,

$$|| \mathbf{A} ||_F = \sqrt{\Sigma_j \Sigma_i \mid a_{ij} \mid^2}. \tag{6}$$

The idea is to reduce the norm of the non-diagonal elements of $\mathbf{A}$ after the unitary transformation. Thus, from [1], reducing off($\mathbf{B}$) which reads,

$$\text{off}(\mathbf{B}) = \sqrt{\Sigma_i^n \Sigma_{j,j\neq i}^n b_{ij}^2} = \sqrt{\text{off}(\mathbf{A})^2 - 2a_{kl}^2}. \tag{7}$$

By setting a_{kl} to be the largest non-diagonal element, we get the smallest possible value of off($\mathbf{B}$).

We perform the orthogonal transformation of the matrix $\mathbf{A}$, such that $\mathbf{B} = \mathbf{S}^T\mathbf{A}\mathbf{S}$. The iterations zeros out the largest nondiagonal element a_{kl}. In reference to the lecture notes in [1] the elements of $\mathbf{B}$ is now,

$$b_{ii} = a_{ii}, \tag{8}$$
$$b_{ik} = a_{ik}\cos\theta - a_{il}\sin\theta, \tag{9}$$
$$b_{il} = a_{il}\cos\theta + a_{ik}\sin\theta, \tag{10}$$
$$b_{kk} = a_{kk}\cos^2\theta - 2a_{kl}\cos\theta\sin\theta + a_{ll}\sin^2\theta, \tag{11}$$
$$b_{ll} = a_{ll}\cos^2\theta + 2a_{kl}\cos\theta\sin\theta + a_{kk}\sin^2\theta, \tag{12}$$
$$b_{kl} = (a_{kk} - a_{ll})\cos\theta\sin\theta + a_{kl}(\cos^2\theta - \sin^2\theta). \tag{13}$$

We know that all the non-diagonal elements needs to be zero, so that $b_{kl} = 0$. We need to find the value of θ so that $b_{kl} = 0$. Dividing b_{kl} by $a_{kl}\cos^2\theta$ let us rewrite the equation to,

$$t^2 - 2\tau t - 1 = 0. \tag{14}$$

Where $t = \tan\theta$ and $\tau = \frac{a_{ll}-a_{kk}}{2a_{kl}}$. The solution of equation 14 is,

$$t = -\tau \pm \sqrt{1+\tau}. \tag{15}$$

In addition, we define $s = \sin\theta$ and $c = \cos\theta$, such that $s = ct$ and $c = \frac{1}{\sqrt{1+t^2}}$. This let us rewrite equation 15 to $t = \frac{1}{\tau\pm\sqrt{1+\tau}}$. To minimize the catastrophic cancellation error, explained in [2], we choose $t = \frac{1}{\tau+\sqrt{1+\tau}}$ when τ is positive and $-$ for $\tau < 0$.

We perform a number of iterations until the sum over the absolute value of the non-diagonal matrix elements are less than a prefixed tolerance which we will call ϵ. Ideally the value of ϵ should be equal to zero. However, the off-diagonal elements do not necessarily ever reach zero, due to numerical errors, so we prefix the tolerance $\epsilon = 10^{-8}$.

The algorithm for diagonalizing a matrix $\mathbf{A} \in \mathbb{R}^n$ can be written as follows;

defining tolerance ϵ

while max(| $(a_{ij}$ |)) $> \epsilon$ **do**
  if $a_{kl} \neq 0$ **then**
    if $\tau > 0$ **then**
      $t = 1/(\tau + \sqrt{1+\tau^2})$;
    else
      $t = 1/(\tau - \sqrt{1+\tau^2})$;
    end if
    c = 1/ $\sqrt{1+t^2}$
    s = c*t
  else
    s = 0
    c = 1
  end if

akk = a_{kk}
all = a_{ll}
akl = a_{kl}
alk = a_{lk}

a_{kk} = akk·c^2 - 2akl·$c \cdot s$ + all·s^2
a_{ll} = all·c^2 + 2akl·$c \cdot s$ + all·s^2
a_{kl} = 0
a_{lk} = 0

for i = 0, 1, 2, ..., n-1 **do**
  if $i \neq k$ and $i \neq l$ **then**
    aik = a_{ik};
    ail = a_{il};
    a_{ik} = aik· c - ail· s;
    a_{ki} = a_{ik}
    a_{il} = ail· s + aik· c;
    a_{li} = a_{il}
  end if
end for
end while

B. Buckling beam

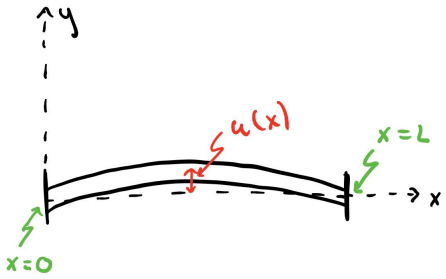

FIG. 1. The figure present a model of the bucking beam. The displacement in the vertical position is given by $u(x)$. Since the beam is fastened at $x = 0$ and $x = L$, the boundary conditions are $u(0) = u(L) = 0$.

The buckling beam problem is a classical wave function problem in one dimension [3]. We start with the following differential equation,

$$\gamma \frac{d^2u(x)}{dx^2} = -Fu(x) \tag{16}$$

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 a force acting on the beam at $x = L$ in the direction towards the origin. The parameter γ is a constant defined by properties like the rigidity of the beam. We apply the Dirichlet boundary conditions where $u(0) = u(L) = 0$. In this specific case two of the parameters γ, F and L are known. If we assume that we know F and L, then the eigenvalue problem we set up will allow us to fins γ. We define a dimensional variable,

$$\rho = \frac{x}{L}. \tag{17}$$

By this definition, we have that $\rho \in [0,1]$. We can reorder the equation such that

$$\frac{d^2u(\rho)}{d\rho^2} = -\frac{FL^2}{\gamma}u(\rho) = -\lambda u(\rho), \tag{18}$$

with $\lambda = FL^2/\gamma$ we have an equation that when discretized, becomes an eigenvalue problem. Next, we define the step length h in the approximation of the second derivative of a function u. With a given number of steps, n, $\rho_{\min} = \rho_0$ and $\rho_{\max} = \rho_N$, we obtain

$$h = \frac{\rho_n - \rho_0}{n}. \tag{19}$$

The value of ρ at a point i is then

$$\rho_i = \rho_0 + ih \tag{20}$$
$$i = 1, 2, ..., n. \tag{21}$$

With this, we can rewrite the differential equation for a value ρ_i as

$$-\frac{u(\rho_i + h) - 2u(\rho_i) + u(\rho_i - h)}{h^2} = \lambda u(\rho_i). \tag{22}$$

Written in a more compact way, we get

$$-\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} = \lambda u_i. \tag{23}$$

In order to solve the differential equation numerically for any function $u(\rho)$, we rewrite this equation in a more general form, as an eigenvalue problem;

$$\mathbf{Uv} = \begin{bmatrix} d & a & 0 & 0 & ... & 0 \\ a & d & a & 0 & ... & 0 \\ 0 & a & d & a & ... & 0 \\ ... & ... & ... & ... & ... & ... \\ 0 & 0 & 0 & ... & a & d \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ ... \\ v_{N-1} \end{bmatrix} = \lambda \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ ... \\ v_{N-1} \end{bmatrix} \tag{24}$$

The endpoints u_0 and u_N are not included here, as the endpoints are fixed to $u_0 = u_N = 0$. We define the diagonal elements as $d = 2/h^2$ and the non-diagonal ones as $a = -1/h^2$ from equation 23.
The eigenvalue problem has analytical eigenpairs, with eigenvalues given as

$$\lambda_j = d + 2a\cos(\frac{j\pi}{N}), j = 1, 2, ..., N-1. \tag{25}$$

where we have defined the matrix to run from $i = 1$ to $i = N - 1$.
The eigenvectors are

$$\mathbf{u}_j = \left[\sin(\frac{j\pi}{N}, \sin(\frac{2j\pi}{N}, ..., \sin(\frac{(N-1)j\pi}{N})\right]^T, j = 1, 2, ..., N-1 \tag{26}$$

C. Quantum dots in 3d

One electron

The potential of one electron in an harmonic oscillator is,

$$V(r) = \frac{1}{2}kr^2 = \frac{1}{2}\omega^2 r^2 \tag{27}$$

ω is the angular frequency equal to $\sqrt{\frac{k}{m}}$. Where m is the mass of the electron, and k is a positive constant. The potential is spherical symmetric such that the time independent Schrödinger equation is equal to,

$$-\frac{\hbar^2}{2m}(\frac{1}{r^2}\frac{d}{dr}r^2\frac{d}{dr}-\frac{l+1}{r^2})R(r)+V(r)R(r)=ER(r). \tag{28}$$

Where E is the energy and $\hbar$ Planck's constant. In addition, l is the quantum number corresponding to the orbital momentum of the electron. $R(r)$ is obtained by separation of variables of the wave function, $\psi(r,\theta,\phi)=R(r)Y_l^m(\theta,\phi)$. Furthermore, we define $u(r)=rR(r)$ to simplify equation 28. This gives that the Schrödinger equation with $u(r)$ is,

$$-\frac{\hbar^2}{2m}\frac{d^2}{dr^2}u(r)+(V(r)+\frac{l(l+1)}{r^2}\frac{\hbar^2}{2m})u(r)=Eu(r). \tag{29}$$

We let $r\in(0,\infty)$, and demand that the wave function has finite values. We also need to demand that the wave function approaches zero when $r\to\infty$. This implies that $u(0)=u(\infty)=0$.

The energy states of the electron is,

$$E_{nl}=\hbar\omega(2n+l+\frac{3}{2}). \tag{30}$$

In this simulation we will set $l=0$, and use the dimension-less variable $\rho=\frac{r}{\alpha}$. α is an unknown constant, with dimension length. The potential $V(\rho)$ will be equal to $\frac{1}{2}k\alpha^2\rho^2$. This gives,

$$-\frac{d^2}{d\rho^2}u(\rho)+\frac{mk}{\hbar^2}\alpha^4\rho^2u(\rho)=\frac{2m\alpha^2}{\hbar^2}Eu(\rho). \tag{31}$$

We fix the value of α by setting $\alpha=(\frac{\hbar^2}{mk})^{1/4}$. In addition we introduce a value $\lambda=\frac{2m\alpha}{\hbar^2}E$. With all this in mind, we can rewrite the Schrödinger equation to,

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

We can now discretize ρ to $\rho_i=\rho_0+ih$ where $i=1,2,3,...,N$. This let us rewrite the Schrödinger equation to,

$$-\frac{u(\rho_i+h)-2u(\rho_i)+u(\rho_i+h)}{h^2}+\rho_i^2u(\rho_i)=\lambda u(\rho_i). \tag{33}$$

Which again can be rewritten as a more compact equation,

$$-\frac{u_{i+1}-2u_i+u_{i-1}}{h^2}+V_iu_i=\lambda u_i. \tag{34}$$

The V is the harmonic oscillator potential, $V_i\propto\rho_i^2$. Since $k\&\alpha$ are unknown constants, and we are working with a dimensionless differential equation, we can set $V_i=\rho_i^2$. Similar to the bucking beam algorithm, we rewrite the equation as an eigenvalue problem. The non-diagonal elements, a, are identical to a in the bucking beam matrix. However, the diagonal elements, d, are now not constant. Hence, from equation 34, the elements equal to $d_i=\frac{2}{h^2}+V_i$. We arrive at equation 24 with an updated value of d, which we again solve using Jacobi's algorithm for diagonalizing matrices. We will then obtain as the eigenvalues. Further we can find the corresponding eigenvectors $u(\rho)$.

The analytical eigenvalues with our scaling of the energies are $\lambda_i=3+4i$. However, this is for $\rho_{\max}=\infty$. Computers cannot represent infinity, so we need to choose $\rho_{\max}$ as a finite number. We will test for different values of $\rho_{\max}$, such that the differences between the computed eigenvalues and the analytical ones are as small as feasible.

Two electrons

The Schrödinger equation for two electrons with no repulsive Coulomb interaction is,

$$(-\frac{\hbar^2}{2m}\frac{d^2}{dr_1^2}-\frac{\hbar^2}{2m}\frac{d^2}{dr_2^2}+\frac{1}{2}kr_1+\frac{1}{2}kr_2)u(r_1,r_2)=Eu(r_1,r_2). \tag{35}$$

r_1 and r_2 are the positions of the two electrons. The energy E is the sum of the relative energy E_r and the centre off mass energy E_R, $E=E_r+E_R$.

Further we will add the repulsive Coulomb interaction potential given by $V(r_1,r_2)=\frac{\beta e^2}{|r_2-r_1|}$. We define $\mid r_2-r_1 \mid = r$. The potential is only dependent on the relative distance r. Adding the potential term to equation 35 for the relative energy gives,

$$(-\frac{\hbar^2}{2m}\frac{d^2}{dr^2}+\frac{1}{4}kr+\frac{\beta e^2}{r})\psi(r)=E_r\psi(r). \tag{36}$$

Where $u(r)=\psi(r)\theta(R)$.

Similar to the one electron problem, we introduce the dimensionless variable $\rho=\frac{r}{\alpha}$. Using the same steps as for the one electron QD, we arrive at the equation,

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

Furthermore, we set $\alpha=\frac{\hbar^2}{m\beta e^2}$ and define $\lambda=\frac{m\alpha^2}{\hbar^2}E$ so that we can rewrite the Schrödinger equation to,

$$-\frac{d^2}{d\rho^2}\psi(\rho)+\omega_r^2\rho^2\psi(\rho)+\frac{1}{\rho}\psi(\rho)=\lambda E_r\psi(\rho). \tag{37}$$

Where ω_r is the new frequency equal to $\sqrt{\frac{mk\alpha^4}{\hbar^2}}$. We are now at a point in which we again can discretize the differential equation,

$$-\frac{\psi_{i+1}-2\psi_i+\psi_{i-1}}{h^2}+(\omega_r^2\rho_i^2+\frac{1}{\rho_i})\psi_i=\lambda\psi_i. \tag{38}$$

Our diagonal elements are now $d_i=\frac{2}{\hbar^2}+\frac{1}{\rho_i}+\omega_r^2\rho_i^2$, and the non-diagonal elements are equal to a in one electron QD. Again we preform the diagonalization by Jacobi's method as for the two other differential equations. We will obtain λ and the wave function ψ for two interacting electrons in an harmonic oscillator potential. In addition, we choose the frequencies $\omega_r=0.01,0.5,1,5$ and compare the numerical solutions.

III. RESULTS

The unit tests implemented tested both the conservation of eigenvalues before and after the diagonalization, and that our function for finding the maximum non-diagonal element accurately returns the correct indices for that element. Our code passed those tests. In addition, from table I we can see further proof for the accuracy of the numerically computed eigenvalues, as the analytical and numerical eigenvalues for the buckling beam reassuringly conform.

TABLE I. Table of the numerical and analytical eigenvalues for the Buckling beam problem, calculated for a size 5×5 matrix.

Numerical eigenvalues	Analytical eigenvalues
9.6462	9.6462
$3.6000\cdot10^1$	$3.6000\cdot10^1$
$7.2000\cdot10^1$	$7.2000\cdot10^1$
$1.0800\cdot10^2$	$1.0800\cdot10^2$
$1.3435\cdot10^2$	$1.3435\cdot10^2$

The plots in figure 2 display the theoretical and numerical results of $u(\rho)$ for two cases $n = 10$ and $n = 100$, which appear to match up. In addition, we see that the extension of the buckling beam appears more continuous for larger values of n. Also, we note that the maximum extension is lower for $n = 100$, than for $n = 10$.

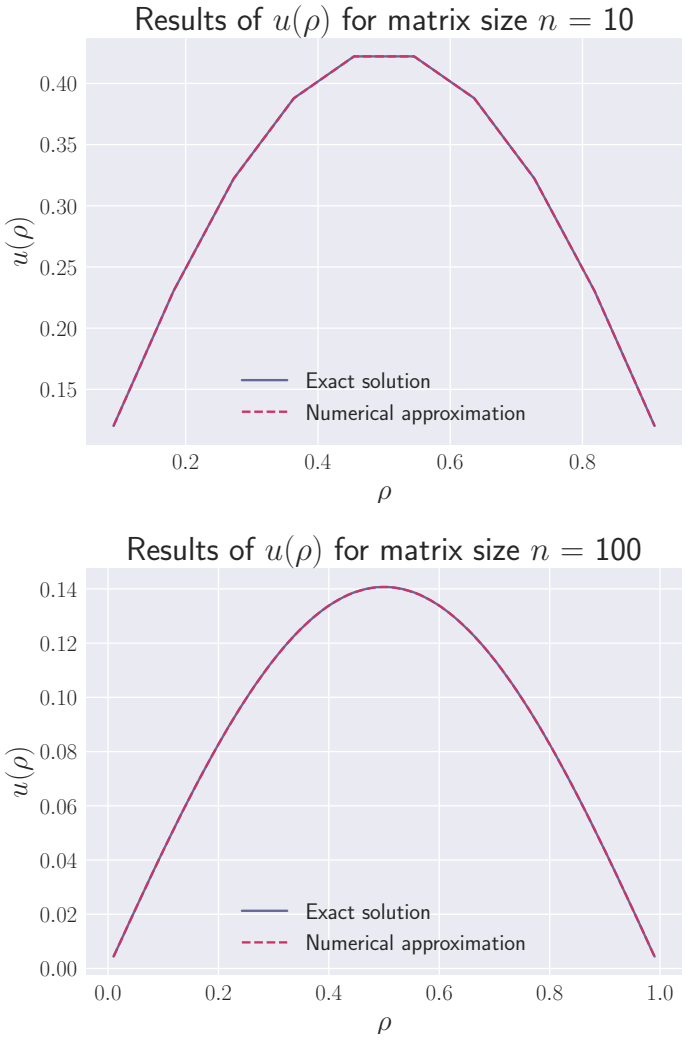

FIG. 2. The figure displays the solution $u(\rho)$ for different $\rho \in [0, 1]$ for $n = 10$ and $n = 100$.

The optimal value of the parameter $\rho_{\max}$ with respect to the accuracy of the numerically computed eigenvalues is dependent on the matrix size $n \times n$ as seen in figure 3. The $\rho_{\max}$ which produces the least deviation from analytical values is presented in table II, along with the number of iterations needed for the diagonalization. From table II we can further see that the number of integration points needed for an accuracy of three leading digits are $N \sim 4164$. The difference, which is the numerical deviation from the smallest analytical eigenvalue for the matrix, has its lowest value between $\rho_{\max} = 3$ and 4. To be more precise, the optimal value of $\rho_{\max}$ was found to be 3, 3, 3.4, 3.6 and 3.7 for $n = 5, 10, 50, 100$ and 150 respectively as presented in table II.

TABLE II. The optimal values $\rho_{\max}$ for a size $n \times n$ matrix, with number of integration points N needed to diagonalize the matrix. The deviation $\Delta\lambda$ is also presented.

n	5	10	50	100	150
$\rho_{\max}$	3	3	3.4	3.6	3.7
N	33	143	4164	16765	37858
$\Delta\lambda$	0.07	0.01	0.0006	0.0004	0.0004

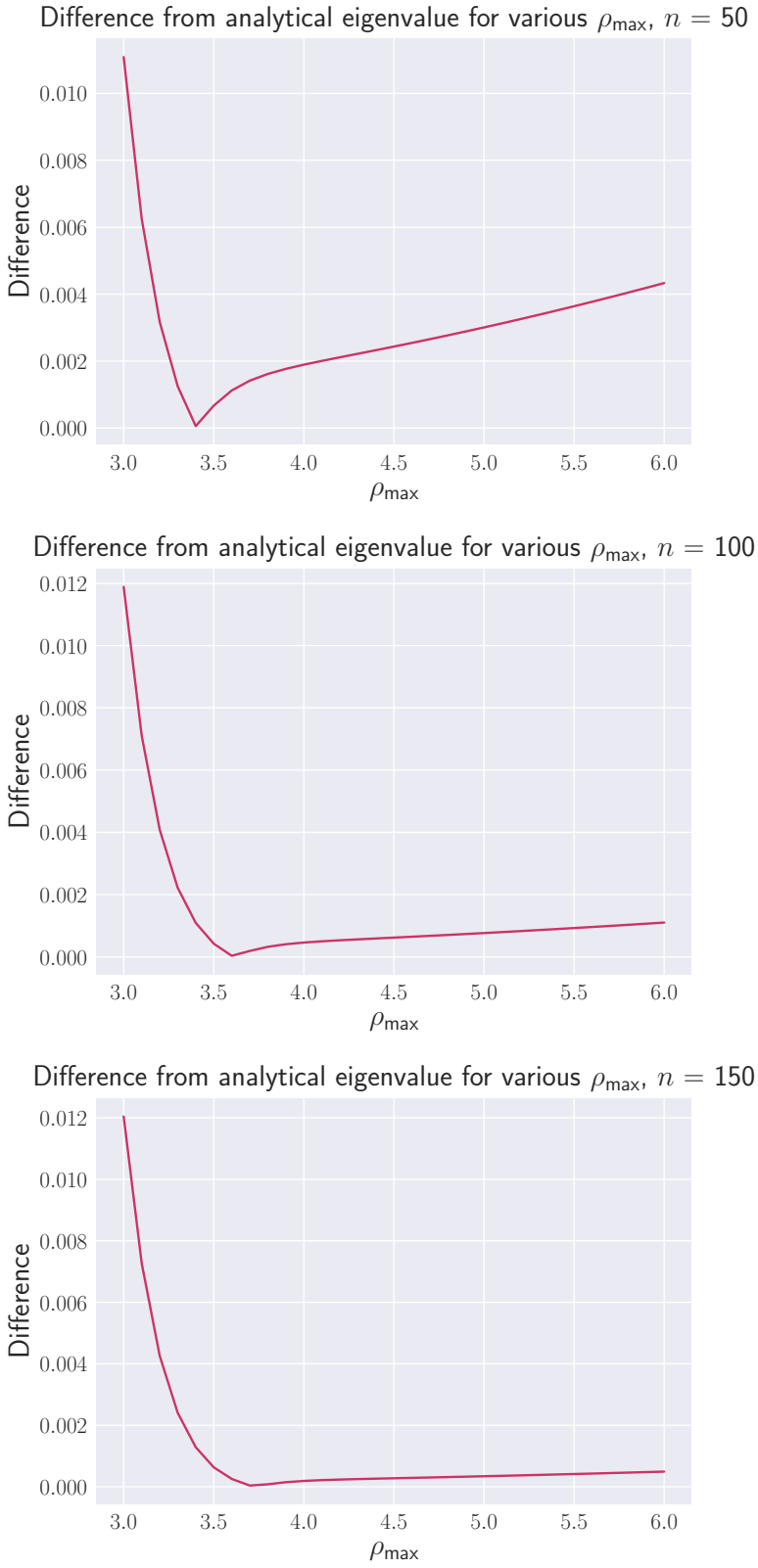

FIG. 3. The figure displays the accuracy of the numerically computed eigenvalues as a function of $\rho_{\max}$ for a size $n \times n$ matrix for different n's in the QD simulation with one electron.

From table III we see that the number of unitary iterations N needed to diagonalize a matrix of size $n \times n$ increases

with n.

TABLE III. Table of the number of similarity transformations N, needed for all non-diagonal matrix elements to be less than the tolerance ϵ for the Buckling beam problem.

n	N
5	32
10	154
50	4313
100	17702
300	161061

Table IV shows the computed eigenvalues with two different $\rho_{\max}$. The theoretical eigenvalues, which are $\lambda_i = 3+4i$, are more related to the values with $\rho_{\max} = 3.6$ for the lowest eigenvalues. However, for the higher eigenvalues we see that $\rho_{\max} = 8$ produce more accurate results.

TABLE IV. Numerically computed eigenvalues of matrix A for two different values of $\rho_{\max}$, for $n = 100$.

$\rho_{\max} = 3.6$	$\rho_{\max} = 8$
3.00004	2.99804
7.02713	6.99018
11.3681	10.976
16.7048	14.9556
23.4779	18.9288
31.7695	22.8956
41.5728	26.8561
52.8726	30.8102
65.6544	34.758
79.9037	38.6993

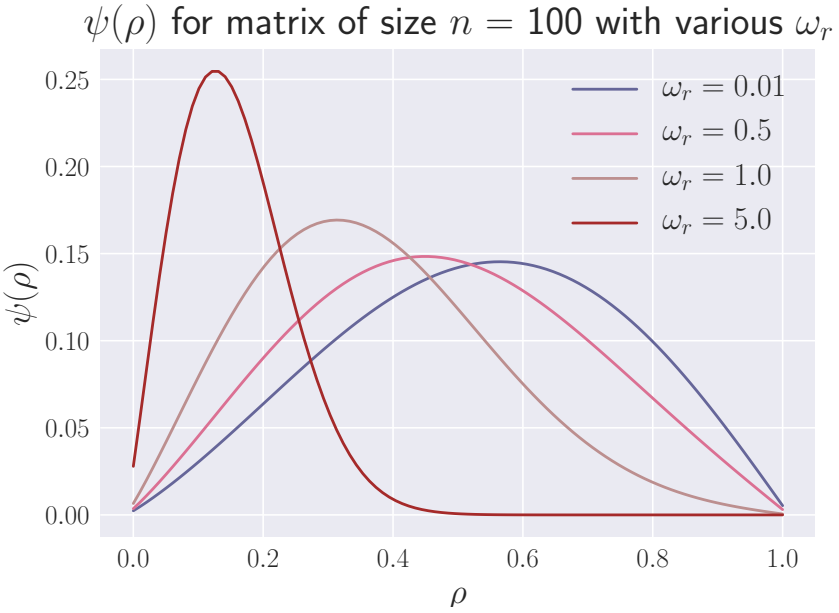

FIG. 4. The figure displays the squared wave function, which connected to the probability of observing the electron at ρ, as a function of the dimensionless length-variable ρ.

The probability of the distance between two electrons in the QD simulation is displayed in figure 4. Here we have

plotted for four different values of ω_r. It is evident that the higher value of ω_r has a higher probability of at smaller values of ρ which is the dimensionless distance between the two electrons.

IV. DISCUSSION

We have used an iterative method, Jacobi's method, in order to find the eigenvalues of three different physical differential equations. Since the method uses three $n \times n$ matrices, the number of iterations will depend on n. This is apparent in table III, where the number N of interations needed to diagonalize a matrix of size $n \times n$ increase with n.

The implemented unit tests tell us that the algorithm provide accurate eigenvalues, which is reassuring. However, from our results we observe that the calculated eigenvalues for the QDs problem with one electron depends both on the number of mesh points, n, and $\rho_{\max}$. This makes it difficult to calculate many precise eigenvalues at the same time. As presented in table IV, for $\rho_{\max} = 3.6$ the first few computed eigenvalues are more precise in comparison to the analytical values than for $\rho_{\max} = 8$. However, the accuracy soon decreases, making it seem like the algorithm skips some of the eigenvalues as the number of n increases. By setting $\rho_{\max} = 8$, the problem does not longer appear, however, it is evident that the accuracy of the first eigenvalues are lower. This problem is caused by the fact that the algorithm has problems producing precise eigenvalues when we let the interval of ρ run from $[0, \infty]$.

From the plots in figure 3 we find that the $\rho_{\max}$ which gives the smallest deviation from analytical values differs for varying values of n. This is also noticeable in table II. Since we in the QD simulation with two electrons only are interested in the ground state and thus only the first eigenvalue, the problems mentioned above such as the lack of accuracy for larger intervals ρ is insignificant. Ideally, $\rho_{\max} = \infty$, as this gives $u(\rho_{\max}) = 0$, which is our boundary condition. However, we need to make compromises as this is non-viable in a computer program. We need to set $\rho_{\max}$ to be a number which is representable on a computer. Having said that, it is not necessarily practical that its value is immense. If $\rho_{\max}$ is large and n is set, the step size $\Delta\rho$ will decrease and we will get a loss of precision. On the other hand, if $\rho_{\max}$ is too small, the wave functions will be cut off. The right balance between the two error sources is therefore changed for different values of n. As expected lower values of n has a lower optimal value of $\rho_{\max}$, and vice versa. This is clearly represented in figure 3.

The eigenvalues are normalized. This is compatible with the buckling beam plot in figure 2 which shows that the value of $u(\rho)$ is lower for $n = 50$ than for $n = 10$. Thus the higher values of n, the number of eigenvalues are increased.

The slope of the potential for the quantum mechanical problem with two electrons increases with ω_r. This implies that with an increasing ω_r, the electrons will have an increasing probability of being located with a smaller distance between the electrons, as becomes apparent in the plot of the calculated probability $\mid \psi(\rho) \mid^2$ in figure 4. Since the wave function is not properly scaled, the areas under the probability curves are not appearing to be equal to one. However, they still display the difference of probability distributions for varying ω_r. It could have been interesting to compare different values of $\rho_{\max}$ and n, here along with for one electron. However, since we found that for one electron $n = 100$ and $\rho_{\max} = 3.6$ produced a lower grade of error we continued using these values.

V. CONCLUSION

We have utilized Jacobi's method which provided viable eigenvalues for the three different differential equations. Our unit tests ran without raising errors, thus we are confident that the parts of our algorithm which they tested are producing accurate results. The unit tests is furthermore proof of the Jacobi method's adequacy.

We have found that the buckling beam problem supplied eigenvalues which complied with the analytical ones in great detail. We also found that the extension of the buckling beam decreased with an increasing value of n. This is a result of the eigenvalues being normalized. Furthermore, exploring different values of n for the buckling beam problem showed us that the number of iterations N needed to diagonalize a $n \times n$ matrix increased for increased n.

For the quantum mechanical part of the project, we can conclude that both the measure of grid points n as well as the value of $\rho_{\max}$ had an effect on the accuracy of the numerically computed eigenvalues. We have discussed

the problematicity that for large $\rho_{\max}$ and a set n, the step size $\Delta\rho$ decreases, thus giving a loss of precision, while if $\rho_{\max}$ is too small, the wave functions will be cut off. Moving on, we further saw that for varying ω_r's, the probability of both electrons being located within a smaller interval increased with increasing ω_r.

We also observed that one cannot calculate many precise eigenvalues at the same time due to the fact that the algorithm has problems producing precise eigenvalues when we let the interval of ρ run from $[0, \infty]$. This is a flaw in the Jacobi eigenvalue method which affect our results for both the buckling beam problem as well as the QD with one electron in three dimensions. As discussed, this is less problematic when we are only interested in the ground state for the QD problem with two electrons. Ideally, the upper bound of ρ would be equal to infinity especially in the quantum mechanical case, but this is not an option in numerical computing.

[1] Hjort-Jensen M. *Computational Physics, Lecture Notes.* 2015.
[2] Marthinussen Sulebakk and & Reiersrud. *Solving the one dimensional Possion equation, GitHub: https://github.com/hedvigborgen/fys3150/tree/master/project1.* 2020.
[3] Hjort-Jensen M. *Project 2 Computational Physics 3150/4150, Assignment Description.* 2020.