Project 2

Johanne Mehren, Marit Kollstuen, Stine Sagen

September 2018

Abstract

In this project we first implement Jacobi's method on a tridiagonal Toeplitz matrix to find its eigenvalues and corresponding eigenvectors. Further, we study numerical solutions of the radial part of Schrodinger's equation in three dimensions. First for one electron that moves in a harmonic oscillator potential and then for two electrons. Jacobi's method proves to demand $\approx 1.5n^2$ rotations to diagonalize a matrix. In order to achieve a steady solution for the one-electron problem, one should use approximately $n = 250$ integration points and $\rho_{max} \approx 5$. The same value for ρ_{max} is also applied to the two-electron case. Higher frequency ω_r reviles a higher probability of finding the electrons closer to the left corner of the radial coordinate axis.

1 Introduction

This project is an attempt to develop an algorithm for solving eigenvalue problems of tridiagonal matrix systems. The Toeplitz matrix is a type of tridiagonal matrix, which has analytical eigenpairs. The algorithm applies Jacobi's method for solving eigenvalue problems, and will be thoroughly explained in the Theory section. We can then test the result of the algorithm compared to the analytic solutions by applying unit tests.

The problems we are trying to solve are two-point boundary value problems of a spring fastened at both ends, which have analytical solutions, namely the Buckling Beam problem (Section 3.1). We can exploit this property when developing the Jacobi algorithm in order to apply it on more complicated calculations. Later, we will use the same method in studying harmonic oscillator problems in three dimensions with one or two electrons. This is a quantum mechanical problem that concerns the radial part of Schrodinger's equation (Equation 3) for one electron, which can be found in Section 2.4, and then for two electrons (Equation 9), which can be found in Section 2.5.

The Results and Discussion section (Section 3 and 4) contains the comparison of our calculations and the analytical results. The efficiency of our algorithm in terms of CPU time as well as number of rotations the algorithm requires are also considered. We are lastly discussing the ideal harmonic potential, ρ_max, and number of integration points, n, to obtain a stable solution for the one- and two electron problems. The code is written in Python version 3.6.

Our motivation for the project is to understand the advantages of solving differential equations by matrix operations. Since none of us are familiar with quantum mechanics, we use this project to learn more about other fields of physics and how we can implement this to boundary value problems in meteorology.

2 Theory

2.1 Orthogonal transformation

The main idea of orthogonal transformations is to rotate a vector about the origin and represent the same vector in a rotated coordinate system. Orthogonal transformations preserve the orthogonality of the obtained eigenvectors. This can be shown by considering a basis of a vector $\mathbf{v}_i$, where:

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

We assume that the basis is orthogonal, so that $\mathbf{v}_j^T \mathbf{v}_i = \delta_{ij}$. For an orthogonal (unitary) transformation, we have: $\mathbf{w}_i = \mathbf{U}\mathbf{v}_i$, in which $\mathbf{U}$ is a unitary, orthogonal matrix so that $\mathbf{U}^{-1} = \mathbf{U}^T$ and $\mathbf{U}^T\mathbf{U} = \delta_{ij}$. The preservation of the dot product and orthogonality by unitary transformation of the vectors $\mathbf{v}_i$ can then be shown by:

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

2.2 Jacobi's rotation method

To solve the eigenvalue problem

$$\mathbf{A}\mathbf{u} = \lambda\mathbf{u} \tag{1}$$

we can implement the Jacobi's rotation algorithm to a tridiagonal Toeplitz matrix. By applying the Jacobi rotation matrix $\mathbf{S}_{kl}$:

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

That contains ones along the diagonal except for two elements $\cos\theta$ in rows and columns k and l. The off-diagonal elements are zero except the elements $\sin\theta$ and $-\sin\theta$. The matrix $\mathbf{S}_{kl}$ has the property $\mathbf{S}^T = \mathbf{S}^{-1}$. It performs a plane rotation around an angle θ until the matrix $\mathbf{A}$ becomes almost diagonal, i.e. the non-diagonal elements are zero. The elements in the diagonal are approximations of the eigenvalues of $\mathbf{A}$.

2.3 The buckling beam problem

The buckling beam problem can be visualized as a string with length $L, x\epsilon[0, L]$ fastened at both ends. This can be written as the differential equation

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

where $u(x)$ represents the vertical displacement of the beam in y-direction, the force F applied at the beam towards origin and the parameter γ. Introducing the dimensionless variable $\rho = x/L, \quad \epsilon[0, 1]$, rewrite Equation 2 and discretize, the problem can be written as an eigenvalue problem. We can apply this theory to problems in quantum mechanics as described in the two next subsections.

2.4 One electron in three dimensions

According to **?** few one-electron problems in quantum mechanics can be solved analytically . We will now consider a system with Coulomb correlations which has an exact solution by first evaluating one single electron in three dimension. The electron has a harmonic oscillator potential. The radial part of Schrodinger's equation in spherical coordinates can be written as

$$-\frac{\hbar^2}{2m}\Big(\frac{1}{r^2}\frac{d}{dr}r^2\frac{d}{dr} - \frac{1(l+1)}{r^2}\Big)R(r) + V(r)R(r) = ER(r) \tag{3}$$

In which $V(r)$ is the harmonic oscillator potential, $(1/2)kr^2$ with $k = m\omega^2$ and E is the energy of the harmonic oscillator in three dimensions.

Let ω be the oscillator frequency and l represents the orbital momentum of the electron, the associated energies of the system is given by the relation:

$$E_{nl} = \hbar\omega\left(2n + l + \frac{3}{2}\right) \tag{4}$$

where $n = 0, 1, 2, ...$ and $l = 0, 1, 2, ...$

Performing the substitution $R(r) = \frac{1}{r}u(r)$ and introducing a dimensionless variable $\rho = (\frac{1}{\alpha})r$ Equation (3) now reads:

$$-\frac{\hbar^2}{2m\alpha^2}\frac{d^2}{d\rho^2}u(\rho) + \Big(V(\rho) + \frac{l(l+1)}{\rho^2}\frac{\hbar^2}{2m\alpha^2}\Big)u(\rho) = Eu(\rho) \tag{5}$$

which has boundary conditions $u(0) = 0, u(\infty) = 0$. Using a dimensionless variable ρ makes it simple to implement the harmonic oscillator potential to our matrix.

Throughout this project, the orbital momentum l is set to zero. Making this assumption, one can easily achieve the desirable eigenvalue problem to be studied closer.

Further, by substituting an expression for the harmonic oscillator potential $V(\rho) = \frac{1}{2}k\alpha^2\rho^2$ into Equation (9) and preform some multiplication and rearranging, the final Schrodinger's equation to be discretized now reads:

$$-\frac{d^2}{d\rho^2}u(\rho) + \rho^2 u(\rho) = \lambda u(\rho) \tag{6}$$

Where $\lambda = \frac{2m\alpha^2}{h^2}E$ represents the eigenvalues.

Applying the finite difference approximation obtained from Taylor expansion on the second order derivative, Equation (6) can be discretized as:

$$-\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + V_i u_i = \lambda u_i \tag{7}$$

With the harmonic oscillator potential defined as $V_i = \rho_i^2$

The grid size is:

$$h = \frac{\rho_N - \rho_0}{N} \tag{8}$$

where N is the amount of grid points and ρ_N and ρ_0 is the values taken at the boundary.

Equation (7) can be represented as a matrix eigenvalue problem (see Equation 1) on the same form as the buckling beam problem.

$$\begin{bmatrix} d_0 & e_0 & 0 & 0 & ... & 0 & 0 \\ e_1 & d_1 & e_1 & 0 & ... & 0 & 0 \\ 0 & e_2 & d_2 & e_2 & 0 & ... & 0 \\ ... & ... & ... & ... & ... & ... & ... \\ 0 & ... & ... & ... & e_{N-1} & d_{N-1} & e_{N-1} \\ 0 & ... & ... & ... & ... & e_N & d_N \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ ... \\ ... \\ ... \\ u_N \end{bmatrix} = \lambda \begin{bmatrix} u_0 \\ u_1 \\ ... \\ ... \\ ... \\ u_N \end{bmatrix}$$

The diagonal and non-diagonal matrix elements are defined as $d_i = \frac{2}{h^2} + V_i$ and $e_i = -\frac{1}{h^2}$. The harmonic oscillator potential V_i is thus added to the diagonal elements.

2.5 Two electrons in three dimensions

Adding an additionally electron to the potential oscillator yields a new issue to study. Two different cases are taken into account; one without Coulomb interaction and another where this interaction is present.

Schrodinger's Equation (3) when there is no Coulomb interaction is now:

$$\left(-\frac{\hbar^2}{2m}\frac{d^2}{dr_1^2} - \frac{\hbar^2}{2m}\frac{d^2}{dr_2^2} + \frac{1}{2}kr_1^2 + \frac{1}{2}kr_2^2 \right) u(r_1, r_2) = E^{(2)} u(r_1, r_2) \tag{9}$$

$E^{(2)}$ is the total energy of the system,$E^{(2)} = E_r + E_R$, where E_r is the relative energy and E_R is the center-of-mass energy. The energies must be quantized as we are looking at a quantum mechanic system with confined electrons. Since the solution of the center of mass is well known, we focus on the relative part.

When there is acting a repulsive Coulomb interacting given as:

$$V(r_1, r_2) = \frac{\beta e^2}{r} \tag{10}$$

where $\beta e^2 = 1.44$ eVnm. This term can then be added to the Schrodinger's equation. Following the same procedure as for the one-electron case in which a dimensionless variable $\rho = \frac{r}{\alpha}$ is introduced and preform some rewriting, the Schrodinger's equation for two electrons case as an eigenvalue problem reads:

$$-\frac{d^2}{d\rho^2}\psi(\rho) + \omega_r^2\rho^2\psi(\rho) + \frac{1}{\rho} = \lambda\psi(\rho) \tag{11}$$

where the frequency parameter ω_r indicates the magnitude of the oscillator potential.

The harmonic oscillator potential in this case is $\omega_r^2\rho^2 + \frac{1}{\rho}$ in which are to be added to the diagonal.

3 Results

3.1 The buckling beam problem

The Jacobi method was executed with a tolerance of $1E-8$ (i.e. iterations proceeded until all non-diagonal elements were lower than the tolerance and basically zero). Table 3.1 displays the iterations needed for the Jacobi method to converge for a given matrix of size $n \times n$. From the table, one can read that the iterations required for $n = 3$ is larger than for $n = 4$, and increasing thereafter. In general, The CPU-time (to the right) for small n is also varying, but steadily increasing after $n = 50$.

Size of matrix (n):	**Number of iterations:**	**CPU - time (sec):**
3	10	0.363
4	7	0.500
10	154	0.392
50	4349	0.519
100	17660	0.624
200	70833	4.01
300	160140	19.09

Table 1: Number of similarity transforms (iterations) required to achieve (near) zero-elements on the non-diagonals of a matrix of size $n \times n$. CPU-time is given to the right

3.1.1 Unit test

The unit test of the jacobi method is given below. Here, we are testing whether the eigenvalues yielded are the correct ones. This is done by checking if the eigenvalues yielded by the Jacobi method is the same as the ones calculated analytically and by the `numpy.linald.eig` method.

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Unit test for the jacobi method applied on a tridiagonal symmetric Toepliz
matrix.

Functions:
    test_jacobi_eigvec(n)
    test_jacobi_orthogonality(n)
"""
```

```python

from jacobi import jacobi
from jacobi import make_matrix
from jacobi import analytical_eig
import numpy as np

def test_jacobi_eigvec(n):
    """
    Tests if the Jacobi method yields the same eigenvalues as numpy.linalg.eig
    and the analytical eigenvalues

    Input:
        n - size of matrix
    """
    A = make_matrix(n,1)
    eigval = analytical_eig(A)
    eigval_np, teigvec_np = np.linalg.eig(A)
    eigval_J, eigvec_J, a, r, t= jacobi(A)
    np.testing.assert_allclose(sorted(eigval), sorted(eigval_np), rtol=1e-08, atol=0) #rtol - relative tolerance, atol - absolute tolerance
    np.testing.assert_allclose(sorted(eigval), sorted(eigval_J), rtol=1e-08, atol=0)
    print('test_jacobi_eigvec passed')


def test_jacobi_orthogonality(n):
    """
    Tests if the orthogonality is preserved for the eigenvectors calculated by
    the Jacobi method

    Input:
        n - size of matrix
    """
    A = make_matrix(n,1)
    eigval_np, teigvec_np = np.linalg.eig(A)
    eigval_J, eigvec_J, a, r, t = jacobi(A)
    err = 1E-5
    assert 1 - np.dot(eigvec_J[0], eigvec_J[0]) <= err
    assert np.dot(eigvec_J[0], eigvec_J[1]) <= err
    assert 1- np.dot(eigvec_J[3], eigvec_J[3]) <= err
    assert np.dot(eigvec_J[3], eigvec_J[1]) <= err

    print('test_jacobi_eigvec passed')

def test_largest_elem(n):
    """
    Tests if the largest element on the non-diagonal that was picked on the
    last iteration in the Jacobi method was indeed the largest element on the
    rotated matrix

    Input:
        n - size of matrix
    """
    A = make_matrix(n,1)
    eigval_J, eigvec_J,AMax,r, t = jacobi(A)
    maxelem = 0.0
    for i in range(n):
        for j in range(i+1,n):
            if abs(A[i,j]) >= maxelem:
                maxelem = abs(A[i,j])
    assert maxelem == AMax
    print('test_largest_elem passed')


if __name__ == "__main__":
    test_jacobi_eigvec(4)
    test_jacobi_orthogonality(4)
    test_largest_elem(4)
```

3.2 One electron in three dimensions

The code from the Buckling-beam problem was modified by adding the harmonic oscillator potential ρ^2 to the tridiagonal Toepliz matrix. Table 2 shows the first four eigenvalues for different n and $\rho_{max} = 5$. The line highlighted in yellow is the best approximation out of the tests in which $\lambda_1 \approx 3$, $\lambda_2 \approx 7$, $\lambda_2 \approx 11$ and $\lambda_4 \approx 15$. With $n = 250$ integration points, we get similarity down to the fourth decimal for λ_1, λ_2 ad λ_3, and to the third decimal for λ_4.

height$n$:	**Eigenvalues: λ_1, λ_2, λ_3 and λ_4:**				**CPU (sec):**	**Iterations:**
10	2.91948	6.58303	9.93656	12.91647	0.40344	131
100	2.99922	6.99609	10.99060	14.98654	0.82953	16532
200	2.99980	6.99903	10.99778	15.00057	3.76951	66618
250	2.99988	6.99938	10.99864	15.00233	8.25963	104261
300	2.99991	6.99957	10.99912	15.00331	16.45377	150728
350	2.99994	6.99968	10.99940	15.00391	30.23667	205634
400	2.99995	6.99976	10.99958	15.00431	51.26365	269129

Table 2: Tests for different integration points, n. The line highlighted in yellow is the best approximation of the eigenvalues λ_1, λ_2, λ_2 and λ_4. The CPU-time and number of rotations are listed to the right.

3.3 Two electrons in three dimensions

The code from the Buckling-beam problem was modified by adding the oscillator potential $\omega_r^2\rho^2 + 1/\rho$ to the tridiagonal Toepliz matrix. In order to investigate stable results as functions of ρ_{max}, we use $\rho_{max} = 5$, that proved to be stable in the previous problem.

Figure 1 in Appendix A displays the different radial wave functions for different ω with $\rho_{max} = 5$ and $n = 250$. Increasing the frequency ω, it can be shown from the plot that the radial distance between the ground state becomes shorter and shorter. The red curve shows the largest value of ω_r when it takes the value 5, and it is approaching zero at $\rho = 1$ compared to the blue curve for the lowest frequency which cover almost the whole range from 0 to 5.

4 Discussion and conclusion

According to **?**, one needs typically $3n^2$ to $5n^2$ rotations by the Jacobi method for the method to converge. However, from our results in Table 3.1, the method converges after a bit less than approximately $1.5n^2$ iterations. The reason for the higher required number of iterations for $n = 3$ is not obvious to us, but afterwards the required number of iterations are steadily increasing, which is expected.

Table 2 reveals that the best approximation to the analytic with scaling for the one-electron energies is achieved with $n \approx 250$ integration points and $\rho_{max} \approx 5$. From $n = 200$ to $n = 400$ integration points, the results have a closeness to the analytic results down to the fourth decimal for three out of four of the four first analytical eigenvalues, but fails to reproduce the third or the fourth.

When looking at Figure 1 in Appendix A which displays the case for the two electrons, it is of interest how the wave function varies with different ω_r. Increasing the value of ω_r results in higher peaks, but on the other hand the width is decreasing. Since the wave function $|\phi(\rho)|^2$ is plotted, a probability distribution is present. According to Section 2.5, increasing of ω_r results in a stronger oscillator potential. The likelihood of finding the electrons closer to $\rho = 0$ can be deduced.

For future work one can let the orbital momentum l differ from zero. This would make the solution to equation 3 more complicated, but would be crucial in further understanding of quantum mechanic systems. We would also investigate why our Jacobi method converges after fewer rotations than the method provided by **?**.

5 Documentation

The material used in this project can be found in the group member's repository in GitHub:
`https://github.com/marikoll/FYS4150_projects/tree/master/project_2`
The Python 3.6 algorithm for the main program is located at:
`https://github.com/marikoll/FYS4150_projects/blob/master/project_2/python/jacobi.py`
The Python 3.6 algorithm for the unit-test is located at:
`https://github.com/marikoll/FYS4150_projects/blob/master/project_2/python/test_jacobi.py`

A Appendix

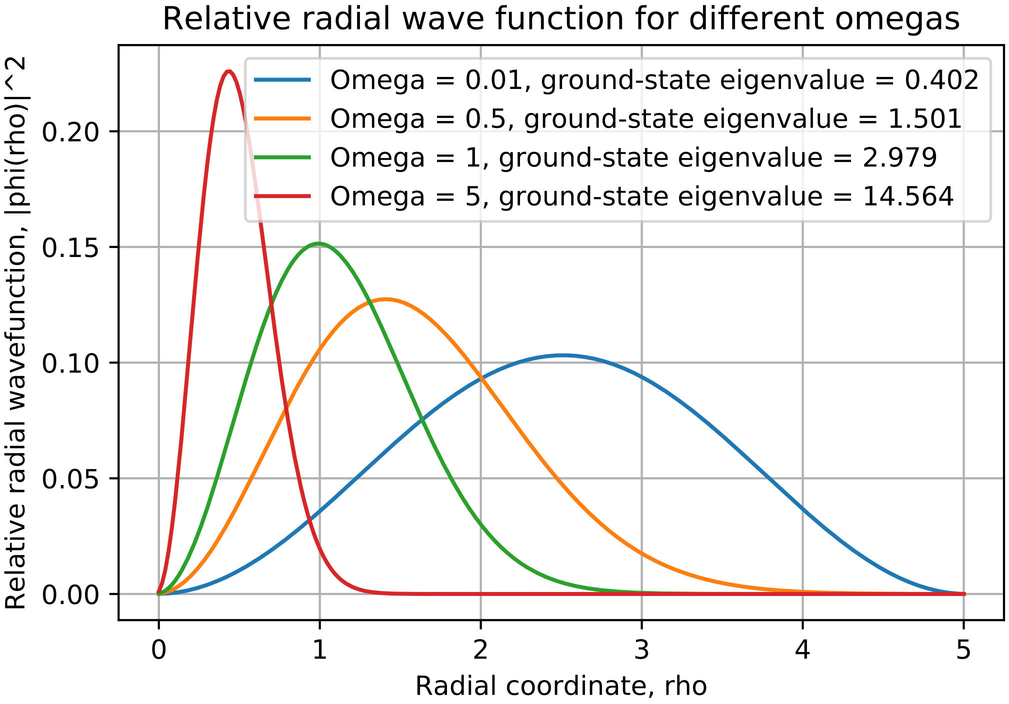

Figure 1: Plot of the radial wave function of four different frequencies ($\omega_r = 0, 01, \omega_r = 0.5, \omega_r = 1$ and $\omega_r = 5$) against the radial coordinate ρ, with ρ_{max} chosen to be ≈ 5 in order to reveal stable solutions. The number of grid points is $n = 250$. In addition to the values of ω, the different values of the corresponding eigenvalues at ground state are also indicated in the label.