

Project 2 FYS4150

Eigenvalue problems

Audun Tahina Reitan and Marius Holm

October 2, 2018

1 Abstract

2 Introduction

In this project we'll develop code for solving eigenvalue problems. Our eigenvalue solver will be based on Jacobi's method, while the matrix we need to diagonalize is the tridiagonal Toeplitz matrix. This matrix has analytical eigenvalues and eigenvectors, which makes it easier for us to test our algorithms.

The first problem we'll look at is the two-point boundary value problem of a buckling beam or a spring fastened at both ends. This problem has analytical solutions, and by adding a new variable along the diagonal we can study quantum mechanical problems. From quantum mechanics we will study the harmonic oscillator problem, with one or two electrons. For the two electron problem we can study the effects of Coulomb interaction and extract some interesting physics results. The two electron problem even has analytical solutions for selected frequencies.[\[Taut, 1993\]](#)

We introduce the relevant methods along with a brief explanation of our implementation.

Structure of report

3 Methods

3.1 Jacobi's method

Jacobi's method is a simple method for solving linear algebra problems of the sort

$$\hat{A}\mathbf{x} = \mathbf{b} \tag{1}$$

where \hat{A} is a matrix and \mathbf{x} and \mathbf{b} are vectors. The Jacobi method is an iterative scheme where we make a guess for the unknown, \mathbf{x} , and after $k + 1$ iterations we have

$$\mathbf{x}^{(k+1)} = \hat{D}^{-1}(\mathbf{b} - (\hat{L} + \hat{U})\mathbf{x}^{(k)}) \tag{2}$$

where we have defined $\hat{A} = \hat{D} + \hat{U} + \hat{L}$. \hat{D} is the diagonal matrix, \hat{U} is an upper triangular matrix, and \hat{L} is a lower triangular matrix. [Hjort-Jensen, 2015]

3.2 Eigenvalue problems

Given the eigenvalue problem

$$\mathbf{A}\mathbf{x}^{(v)} = \lambda^{(v)}\mathbf{x}^{(v)} \quad (3)$$

where \mathbf{A} is a matrix of dimension n . We also have that $\lambda^{(v)}$ are the eigenvalues and $\mathbf{x}^{(v)}$ the corresponding eigenvectors. From this we find that the eigenvalues of \mathbf{A} are given by the n roots of the characteristic polynomial given by:

$$P(\lambda) = \det(\lambda\mathbf{I} - \mathbf{A}) = \prod_{i=1}^n (\lambda_i - \lambda) \quad (4)$$

This procedure is only valid for problems where we only need a small fraction of the eigenvalues and eigenvectors, or if the matrix is on a tridiagonal form. As our matrix is tridiagonal we can use the above procedure which leads us to the Jacobi method. A general procedure for solving equation (3) is to perform similarity transformations until the original matrix \mathbf{A} is either on diagonal form or is a tridiagonal matrix which then easily can be diagonalized. The general procedure leads to the well known Householder's algorithm. [Hjort-Jensen, 2015]

3.3 Jacobi's method

Given an $n \times n$ orthogonal transformation matrix

$$\mathbf{S} = \begin{pmatrix} 1 & 0 & \dots & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & 0 & \dots \\ 0 & 0 & \dots & \cos \theta & 0 & \dots & 0 & \sin \theta \\ 0 & 0 & \dots & 0 & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & 0 & \dots \\ 0 & 0 & \dots & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & \dots & \sin \theta & \dots & \dots & 0 & \cos \theta \end{pmatrix} \quad (5)$$

with the property $\mathbf{S}^T = \mathbf{S}^{-1}$. This matrix performs a plane rotation around an angle θ in the Euclidean n -dimensional space. Which means that the non-zero matrix elements are given by

$$s_{kk} = s_{ll} = \cos \theta, \quad s_{kl} = -s_{lk} = -\sin \theta, \quad s_{ii} = -s_{ii} = 1 \quad i \neq k \quad i \neq l, \quad (6)$$

A similarity transformation

$$\mathbf{B} = \mathbf{S}_u \mathbf{T} \mathbf{A} \mathbf{S}, \quad (7)$$

results in

$$\begin{aligned}
b_{ii} &= a_{ii}, \quad i \neq k, i \neq l \\
b_{ik} &= a_{ik} \cos \theta - a_{il} \sin \theta, \quad i \neq k, i \neq l \\
b_{il} &= a_{il} \cos \theta + a_{ik} \sin \theta, \quad i \neq k, i \neq l \\
b_{kk} &= a_{kk} \cos^2 \theta - 2a_{kl} \cos \theta \sin \theta + a_{ll} \sin^2 \theta \\
b_{ll} &= a_{ll} \cos^2 \theta + 2a_{kl} \cos \theta \sin \theta + a_{kk} \sin^2 \theta \\
b_{kl} &= (a_{kk} - a_{ll}) \cos \theta \sin \theta + a_{kl}(\cos^2 \theta - \sin^2 \theta)
\end{aligned}$$

We can choose the angle θ as we wish, while making sure that all the non-diagonal matrix elements, b_{kl} become zero. The algorithm which follows is then quite simple. We do a number of similarity transformations until the sum over the squared non-diagonal matrix elements are less than some chosen tolerance, typically less than 10^{-8} . We therefore seek to minimize

$$\text{off}(\mathbf{A}) = \sqrt{\sum_{i=1}^n \sum_{j=1, j \neq i}^n a_{ij}^2} \quad (8)$$

3.4 Scaling equations

3.5 Implementation

All our code, benchmark calculations, and plots used can be found in [Auduns GitHub](#).

4 Results

4.1 Preservation of dot product

We want to show that an orthogonal transformation preserves the dot product and orthogonality. Given a basis of vectors \mathbf{v}_i

$$\mathbf{v}_i = \begin{bmatrix} v_{i1} \\ v_{i2} \\ \vdots \\ v_{in} \end{bmatrix} \quad (9)$$

And assuming that the basis is orthogonal.

$$\mathbf{v}_j^T \mathbf{v}_i = \delta_{ij} \quad (10)$$

The transformation is given by

$$\mathbf{w}_i = \mathbf{U} \mathbf{v}_i \quad (11)$$

As the transformation matrix \mathbf{U} is orthogonal we have $\mathbf{U}^T \mathbf{U} = \mathbf{I}$.

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

5 Discussion

5.1 Jacobi method

The Jacobi method is generally a slower algorithm than those based tridiagonalization. However the Jacobi method is easy to parallelize which can improve performance significantly. The main reason for the Jacobi methods slow convergence is that for each new rotation, matrix elements which were zero might change to non-zero values.

6 Appendix A

References

- [Hjort-Jensen, 2015] Hjort-Jensen, M. (2015). Computational physics. lecture notes. Accessible at course github repository. 551 pages.
- [Taut, 1993] Taut, M. (1993). Two electrons in an external oscillator potential: Particular analytic solutions of a coulomb correlation problem. *Phys. Rev. A*, 48:3561–3566.