Project 2 FYS4150

Kjetil Karlsen and Vilde Mari Reinertsen

September 30, 2017

Abstract

Abstract
Intro
V Teori
K Resultat 1b (2 tab): N, lambda jacboi, N, lambda arma kommentar: lambda svarer til energi ... 1e (tab) omega, lambda, E tid tab: time (N, Jacobi v armadillo, Similarity transforms (iterations))
Diskusjon
Konklusjon

Contents

1 Introduction

2 Theory

2.1 Dimensionless and scaled Schrödinger Equation

In this report we are looking at how to solve an eigenvalue problem using Jacobi's method. The eigenvalue problem we will solve is the two particle Schrödinger Equation for the ground state and the harmonic oscillator potential. We then start out with the one particle radial Schrödinger Equation (Equation 1 for the ground state which means that the quantum number $l = 0$. Here R is the radial part of the wave function, r is the distance from the origin to the electron, m is the mass of the particle, $\hbar$ is Planck's constant and E is the energy. The harmonic oscillator potential is shown in Equation 2. Here k is the wave number and ω is the oscillator frequency.

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

$$V(r) = \frac{1}{2}kr^2 \quad \text{where } k = m\omega^2 \quad (2)$$

After substituting $R(r) = \frac{1}{r}u(r) = \frac{\alpha}{\rho}u(\rho)$ where $\rho = \frac{r}{\alpha}$, a dimensionless variable, and α is subsequently a constant with dimension length, the radial Schrödinger equation with the harmonic oscillator potential looks like Equation 3.

$$-\frac{\hbar^2}{2m\alpha^2}\frac{d^2}{d\rho^2}u(\rho) + \frac{1}{2}k\alpha^2\rho^2u(\rho) = Eu(\rho) \quad (3)$$

To make the equation a pure eigenvalue problem, we need to scale the equation properly, that we can to with the help of the inserted α. We start by multiplying Equation 3 with $\frac{2m\alpha^2}{\hbar^2}$:

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

If we set $\frac{m\alpha^4}{\hbar^2}k = 1$ then $\alpha = \left(\frac{\hbar^2}{mk}\right)^{\frac{1}{4}}$ and the equation is then written as Equation 4.

$$-\frac{d^2}{d\rho^2}u(\rho) + \rho^2u(\rho) = \lambda u(\rho) \quad \text{where } \lambda = \frac{2m\alpha^2}{\hbar^2}E \quad (4)$$

2.2 An eigenvalue problem

We have the scaled and dimensionless Schrödinger Equation (Equation 4) and the next move is to make it into a numerical eigenvalue problem. We start by making the continuous function $u(\rho)$ a discrete number of values $u_i = u(\rho_i)$. Here $\rho_i = \rho_0 + ih$ where h is the step length $h = \frac{\rho_{max}}{N}$, N is the number of mesh points and $\rho_0 = 0$.

ρ_{max} should be infinitely big which is not possible, since $r \in [0, \infty)$, but we will come back to that in the method part of the report. Using the expression for the second derivative we get:

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

from Equation 4.

This equation can be rewritten into a matrix eigenvalue equation shown in Equation 5

$$\begin{bmatrix} d_1 & e_1 & 0 & \cdots & 0 & 0 \\ e_1 & d_2 & e_2 & \cdots & 0 & 0 \\ 0 & e_2 & d_3 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & d_{N-1} & e_{N-1} \\ 0 & 0 & 0 & \cdots & e_{N-1} & d_N \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{N-1} \\ u_N \end{bmatrix} = \lambda \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{N-1} \\ u_N \end{bmatrix} \quad (5)$$

where $d_i = \frac{2}{h^2} + \rho_i^2$ and $e_i = -\frac{1}{h^2}$.

2.3 Jacobi's method

We are using Jacobi's method to solve the eigenvalue problem from Equation 5 in the previous section. We know that for a symmetric matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ there exists a real orthogonal matrix $\mathbf{S}$ so that $\mathbf{S}^T AS = D$ where D is a diagonal matrix with the eigenvalues on the diagonal.

The idea of eigenvalue problem solving is to do a series of similarity transformations of the matrix $\mathbf{A}$ so that eventually the matrix A is reduced to the matrix D and we have the eigenvalues. $\mathbf{B}$ is a similarity transformation of $\mathbf{A}$ if $\mathbf{B} = \mathbf{S}^T\mathbf{A}\mathbf{S}$.

In Jacobi's method the matrix $\mathbf{S}$ used in the similarity transformations is the orthogonal transformation matrix on the form:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \cos\theta & 0 & 0 & \sin\theta \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -\sin\theta & 0 & 0 & \cos\theta \end{bmatrix}$$

Let s_{ij} be an element in the matrix $\mathbf{S}$ and then $s_{ii} = 1$, $s_{kk} = s_{ll} = \cos\theta$ and $s_{kl} = -s_{kl} = -\sin\theta$ where $i \neq k$ and $i \neq l$. Here k, l are the number of a row or a column and i is the index of the columns and row.

The similarity transformation of a matrix will then change the elements in the matrix. The clue of the method is to choose the angle, θ, so that the elements that are not on the diagonal become zero. Because of the nature of the similarity transformation the eigenvalues stay the same:

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

The eigenfunctions change though, but their orthogonality remains because of $\mathbf{S}$ is an orthogonal matrix and then $\mathbf{S}^T \mathbf{x} = \mathbf{U}\mathbf{x}$, a unitary transformation. It can be shown like this:

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

After a similarity transformation on the matrix $\mathbf{A}$ the elements of the similarity transformation $\mathbf{B}$ becomes:

$$b_{ik} = a_{ik}\cos\theta - a_{il}\sin\theta \qquad i \neq k, i \neq l$$
$$b_{il} = a_{il}\cos\theta + a_{ik}\sin\theta \qquad i \neq k, i \neq l$$
$$b_{kk} = a_{kk}\cos^2\theta - 2a_{kl}\cos\theta\sin\theta + a_{ll}\sin^2\theta$$
$$b_{ll} = a_{ll}\cos^2\theta - 2a_{kl}\cos\theta\sin\theta + a_{kk}\sin^2\theta$$
$$b_{kl} = (a_{kk} - a_{ll})\cos\theta\sin\theta + a_{kl}(\sin^2\theta - \cos^2\theta)$$

Jacobi's method is to reduce the norm of the non-diagonal elements of the matrix with similarity transformations. We change define $c = \cos\theta$ and $s = \sin\theta$. When we require the non-diagonal elements to be zero that implies that:

$$b_{kl} = (a_{kk} - a_{ll})cs + a_{kl}(s^2 - c^2) = 0$$

We can rearrange some variables and get:

$$(a_{ll} - a_{kk})cs = a_{kl}(s^2 - c^2) \implies \frac{(a_{ll} - a_{kk})}{2a_{kl}} = \frac{1}{2}\left(\frac{s^2}{sc} - \frac{c^2}{sc}\right)$$

We define $\tan\theta = t = s/c$ and $\cot 2\theta = \tau = \frac{a_{ll} - a_{kk}}{2a_{kl}}$ and insert that to the equation:

$$\frac{(a_{ll} - a_{kk})}{2a_{kl}} = \frac{1}{2}\left(\frac{s}{c} - \frac{c}{s}\right) \implies \cot 2\theta = \frac{1}{2}(\tan\theta - \cot\theta)$$

$$\cot\theta = \tan\theta - 2\cot 2\theta = (\tau - 2t)$$

$$\tau = \frac{1}{2}(t - (\tau - 2t)) \implies t^2 + 2\tau t - 1 = 0$$

Which then comes down to:

$$t = \tau \pm \sqrt{1 + \tau^2}$$
$$c = \cos\theta = \frac{1}{\sqrt{1 + \tan^2\theta}} = \frac{1}{\sqrt{1 + t^2}}$$
$$s = \sin\theta = \tan\theta\cos\theta = tc$$

3 Method

3.1 Jacobi's method

$$t = \tau \pm \sqrt{1 + \tau^2}$$
$$c = \cos\theta = \frac{1}{\sqrt{1 + \tan^2\theta}} = \frac{1}{\sqrt{1 + t^2}}$$
$$s = \sin\theta = \tan\theta\cos\theta = tc$$

3.2 The algorithm

Metode Algorigmer: Jacobi, rhomax valg av epsilon ("0")

3.3 Unit tests

In this project we used two unit tests to ensure that the program performs as expected and delivers accurate enough results.

One way to be sure that our algorithm gives correct answers is to task it to find the eigenvalues of a matrix with known eigenvalues. These values had been pre calculated by Matlab.

The other unit test we utilized was to check that our algorithm to find the largest off-diagonal elements actually found the largest off-diagonal elements. This was done by setting up a known matrix and tasking our algorithm to find the largest off-diagonal element and checking it against the manually found largest element.

Other tests: Orthogonality, norm(A) = norm(B)

HVORFOR SØKER VI BARE ØVRE HALVDEL????????

4 Result

Table 4.1: Convergence of eigenvalues for the non-interacting case using the Jacobi method

N	λ_1	λ_2	λ_3
10	2.9263	6.61967	10.0351
100	2.99928	6.99642	10.9919
200	2.99982	6.99911	10.9986
400	2.99996	6.99979	11.0003

Table 4.2: Comparisson of the time used between the Jacobi-algorithm and the Armadillo function eig_sym as a function of the mesh size N of matrix A. "Transforms" refers to the amount of similarity transforms the Jacobi method uses.

N	Transforms	time Jacobi (s)	time Armadillo (ms)
10	144	0.000	0.043
100	17681	0.196	1.436
200	71500	3.211	5.572
400	287490	58.802	23.937

Table 4.3: Eigenvalues for the interacting case. All the calculations are done with a mesh size of N=400. ρ_{max} was set to 7.9 for all ω_r, except for $\omega_r = 0.01$ ($\rho_{max} = 50$).

ω_r	λ_0	λ_1	λ_2
0.01	0.106	0.142	0.178
0.25	1.250	2.193	3.192
0.5	2.230	4.134	6.074
1	4.058	7.909	11.818
5	17.445	37.057	56.817

x

5   Discussion

6   Conclusion

References