FYS4150: Project 2

Ingrid A V Holm

September 29, 2016

Abstract

This is the abstract.

Introduction

The aim of this project was to solve the one-dimensional Schrödinger equation. We start out with the three-dimensional equation for two interacting electrons. Since we assume spherical symmetry, we begin with the radial part of the Schrodinger equation:

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

Where $V(r)=\frac{1}{2}kr^2$ is the harmonic oscillator potential.

The oscillator has energy eigenvalues:

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

Where the principal quantum number $n=0,1,2...$ and orbital angular momentum quantum number $l=0,1,2....$ So the energy levels are degenerate. In this project, however, we only look at $l=0$. By several simplifications and subsitutions we aquire the relevant equation for our project:

$$-\frac{d^2}{d\rho^2}u(\rho)+\rho^2u(\rho)=\lambda u(\rho) \tag{3}$$

Where the eigenvalues in our case ($l=0$) are $\lambda_0=3$, $\lambda_1=7$, $\lambda_2=11...$

Numerical approximation

We use the standard expression for the second derivative of a function:

$$u'' = \frac{u(\rho+h) - 2u(\rho) + u(\rho-h)}{h^2} + \mathcal{O}(h^2) \tag{4}$$

So the mathematical error in this case is of the order of $\mathcal{O}(h^2)$. We set the boundary conditions:

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

$$\rho_{min} = \rho_0 = 0$$

$$\rho_{max} = \rho_N = \infty$$

The latter can't be implemented, so we use several large values for ρ_{max}.

Numerical implementation

The mesh points of the calculation are

$$\rho_i = \rho_0 + ih \ , \ i = 1, 2, ..., N$$

So our expression becomes:

$$-\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + \rho_i^2 u_i = \lambda u_i \tag{5}$$

Where we have use that $V_i = \rho_i^2$ is the harmonic oscillator potential. We see that this can be written as a $(N-1) \times (N-1)$-matrix equation:

$$\begin{bmatrix} \frac{2}{h^2} + V_1 & -\frac{1}{h^2} & 0 & ... & 0 \\ -\frac{1}{h^2} & \frac{2}{h^2} + V_2 & -\frac{1}{h^2} & ... & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & ... & -\frac{1}{h^2} & \frac{2}{h^2} + V_{N-2} & -\frac{1}{h^2} \\ 0 & 0 & ... & -\frac{1}{h^2} & \frac{2}{h^2} + V_{N-1} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N-2} \\ u_{N-1} \end{bmatrix} = \lambda \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N-2} \\ u_{N-1} \end{bmatrix}$$

Unitary and orthogonal transformations (a)

A unitary matrix U has the properties:

$$U^{-1} = U^{\dagger} \tag{6}$$

An orthogonal matrix (the corresponding real matrix) has the properties:

$$U^T = U^{-1} \tag{7}$$

Where $U^{\dagger}$ is for matrices in Hilbert space (complex matrices). It represents the transpose of the complex conjugate, i.e. $U^{\dagger} = (U^*)^T$.

Assume we have an orthogonal basis for the n-dimensional space v_i, where $v_j^T v_i = \delta_{ij}$:

$$\mathbf{v}_i = \begin{bmatrix} v_{i1} \\ ... \\ ... \\ v_{in} \end{bmatrix} \;,\; U\mathbf{v}_i = \mathbf{w}_i$$

Under orthogonal transformations, both the dot product and orthogonality of vectors is conserved:

$$\mathbf{w}_i^T \mathbf{w}_j = (U\mathbf{v}_i)^T (U\mathbf{v}_j) = \mathbf{v}_i^T U^T U \mathbf{v}_j = \mathbf{v}_i^T U^{-1} U \mathbf{v}_j = \mathbf{v}_i^T \mathbf{v}_j = \delta_{ij}$$

Jacobi's rotation method (b)

We want to solve the eigenvalue problem for a tridiagonal matrix:

$$\begin{pmatrix} d_0 & e_0 & 0 & ... & 0 & 0 \\ e_1 & d_1 & e_1 & ... & 0 & 0 \\ 0 & e_2 & d_2 & e_2 & ... & 0 \\ \vdots & & & \ddots & & 0 \\ 0 & ... & ... & 0 & e_N & d_N \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \\ u_N \end{pmatrix} = \lambda \begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \\ u_N \end{pmatrix} \tag{8}$$

We want to use Jacobi's rotation method. We define an orthogonal matrix S to operate on our matrix A:

$$B = S^T A S \tag{9}$$

With all elements in S equal to zero, except:

$$s_{kk} = s_{ll} = \cos\theta \;,\; s_{kl} = -s_{lk} = s\sin\theta \;,\; s_{ii} = 1 \;;\; i \neq k \; i \neq l$$

1. Find $\max(a_{kl})$.

2. Compute the rotation parameters $t = \tan\theta$, $c = \cos\theta$, $s = \sin\theta$:

$$\tau = \cot\theta = \frac{a_{ll} - a_{kk}}{2a_{kl}} \tag{10}$$

$$t = -\tau \pm \sqrt{1+\tau^2} \tag{11}$$

$$c = \frac{1}{\sqrt{1+t^2}}\ ,\ s = tc \tag{12}$$

3. Calculate the elements of B:

$$b_{ii} = a_{ii}\ ,\ i \neq k\ i \neq l$$
$$b_{ik} = a_{ik}c - a_{il}s\ ,\ i \neq k\ i \neq l$$
$$b_{il} = a_{il}c + a_{ik}s\ ,\ i \neq k\ i \neq l$$
$$b_{kk} = a_{kk}c^2 - 2a_{kl}c * s + a_{ll}s^2$$
$$b_{ll} = a_{ll}c^2 + 2a_{kl}c * s + a_{kk}s^2$$
$$b_{kl} = (a_{kk} - a_{ll})c * s + a_{kl}(c^2 - s^2)$$

4. If $\max(b_{kl}) \geq \epsilon$ run the algorithm again.

Figure 1: Jacobi rotation algorithm for tridiagonal matrix

Benchmarks

Jacobi rotation

Check the Jacobi function, which finds eigenvalues of a matrix, on the matrix A:

$$A = \begin{pmatrix} 1 & 2 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}\ ,\ \det(A - \lambda\mathbb{1}) = \begin{vmatrix} 1-\lambda & 2 & 0 \\ 2 & 1-\lambda & 0 \\ 0 & 0 & 1-\lambda \end{vmatrix}$$

$$\det(A - \lambda\mathbb{1}) = (1-\lambda)^3 - 2\cdot 2(1-\lambda) = (1-\lambda)[(1-\lambda)^2 - 4]$$

$$= (1-\lambda)[\lambda^2 - 2\lambda - 3] \rightarrow \lambda = 1 \cap \lambda = 2$$

$$\rightarrow (\lambda - 1)(\lambda - 1)(\lambda - 2)$$

Results