

Project 2 - FYS3150 Computational Physics

Fredrik Østrem (`fredost`)
Joseph Knutson (`josephkn`)

September 28, 2016

Abstract

INSERT ABSTRACT HERE

Contents

1	Introduction	2
2	Methods	2
2.1	2
2.2	3
3	Results	4
4	Conclusions	4
A	Appendix	5
B	Bibliography	6

1 Introduction

In this project we are going to solve Schroedinger's equation for two electrons in a three-dimensional harmonic oscillator well. This will be done with and without a repulsive Coulomb potential while we assume spherical symmetry. Our method consists of the Jacobi method where we take advantage of similarity to solve our problem.

Here we present the solution of the radial part of Schroedinger's equation for one electron

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

In our case $V(r)$ is the harmonic oscillator potential $(1/2)kr^2$ with $k = m\omega^2$. We substitute $R(r) = (1/r)u(r)$, set $l = 0$ and obtain

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

Now we introduce a dimensionless variable $\rho = (1/\alpha)r$ where α is a constant with dimension length. We do this to remove unnecessary factors later. We also set $V(\rho)$ equal to the HO potential $(1/2)k\alpha^2\rho^2$ and rewrite our equation:

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

The constant α can now be fixed so that

$$\alpha = \left(\frac{\hbar^2}{mk} \right)^{1/4}.$$

Defining

$$\lambda = \frac{2m\alpha^2}{\hbar^2} E,$$

we can rewrite Schroedinger's equation as

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

2 Methods

2.1

Assume that \mathbf{U} is an orthogonal transformation. Then by definition, $\mathbf{U}^T \mathbf{U} = \mathbf{I}$, so for any vectors \mathbf{u} and \mathbf{v} we see that

$$(\mathbf{U}\mathbf{u}) \cdot (\mathbf{U}\mathbf{v}) = (\mathbf{U}\mathbf{u})^T (\mathbf{U}\mathbf{v}) = \mathbf{u}^T \underbrace{\mathbf{U}^T \mathbf{U}}_{=\mathbf{I}} \mathbf{v} = \mathbf{u}^T \mathbf{v} = \mathbf{u} \cdot \mathbf{v}$$

Since the dot product of vectors are preserved by \mathbf{U} , orthogonality (that is, the dot product being zero) is also preserved.

2.2

We have the differential equation

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

where $V(\rho) = \rho^2$, that we want to solve. By using Taylor expansion of $u(\rho)$ and discretizing u and ρ with step length h , we get the equation

$$-\frac{1}{h^2}u_{i-1} + \left(\frac{2}{h^2} - V_i\right)u_i - \frac{1}{h^2}u_{i+1} = \lambda u_i$$

We can write this as a matrix eigenvector equation, $\mathbf{A}\mathbf{u} = \lambda\mathbf{u}$, where

$$\mathbf{A} = \begin{bmatrix} d_0 & e_0 & 0 & 0 & \dots & 0 & 0 \\ e_1 & d_1 & e_1 & 0 & \dots & 0 & 0 \\ 0 & e_2 & d_2 & e_2 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots & \dots e_{N-2} & d_{N-2} & e_{N-2} \\ 0 & \dots & \dots & \dots & \dots & e_{N-1} & d_{N-1} \end{bmatrix}, \quad d_i = \frac{2}{h^2} - V_i, \quad e_i = -\frac{1}{h^2}$$

We can solve this equation by finding an orthonormal matrix \mathbf{S} and a diagonal matrix \mathbf{B} such that

$$\mathbf{A} = \mathbf{S}^T \mathbf{B} \mathbf{S}$$

where the diagonal elements of \mathbf{B} , b_{ii} , are the eigenvalues λ_i of \mathbf{A} , and the row vectors \mathbf{s}_i of \mathbf{S} are the eigenvectors of \mathbf{A} . We can see this since, if $\mathbf{y} = \mathbf{e}_i$, then \mathbf{y} is an eigenvector of \mathbf{B} with eigenvalue λ_i ; if then $\mathbf{x} = \mathbf{S}^T \mathbf{y}$, then

$$\mathbf{A}\mathbf{x} = \mathbf{S}^T \mathbf{B} \mathbf{S} \mathbf{x} = \mathbf{S}^T \mathbf{B} \mathbf{S} \mathbf{S}^T \mathbf{y} = \mathbf{S}^T \mathbf{B} \mathbf{y} = \mathbf{S}^T \lambda_i \mathbf{y} = \lambda_i \mathbf{S}^T \mathbf{y} = \lambda_i \mathbf{x}$$

where $\mathbf{x} = \mathbf{S}^T \mathbf{e}_i$ is the i 'th column of \mathbf{S}^T , and thus also the i 'th row of \mathbf{S} .

We'll be using the Jacobi method, in which we in multiple steps will remove the off-diagonal elements of \mathbf{A} with orthonormal transformations, until we have a matrix where the off-diagonal elements are sufficiently small. Start with $\mathbf{A}_0 = \mathbf{A}$ and $\mathbf{P}_0 = \mathbf{I}$.

For each step with matrices \mathbf{A}_n and \mathbf{P}_n , we find the element a_{kl} that is the largest off-diagonal element in terms of absolute value. We define

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

and the orthonormal similarity transformation

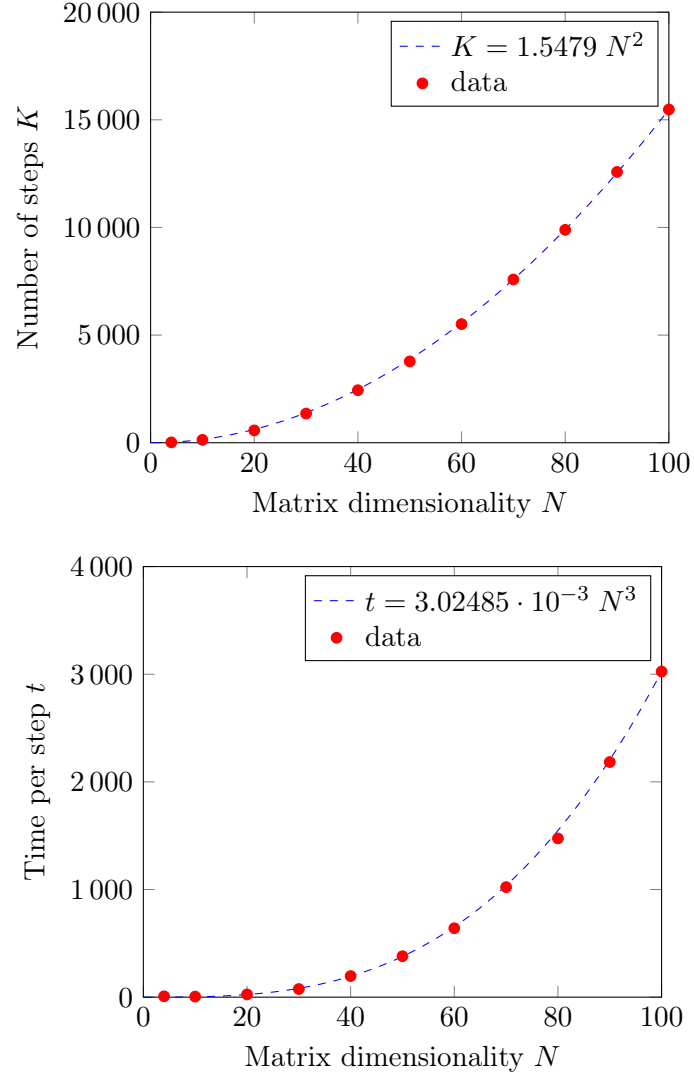
$$\mathbf{S}_n = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \cos \theta & 0 & \dots & 0 & \sin \theta & \dots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

where $(\mathbf{S}_n)_{kk} = (\mathbf{S}_n)_{ll} = \cos \theta$ and $(\mathbf{S}_n)_{kl} = -(\mathbf{S}_n)_{lk} = \sin \theta$.

We then find $\mathbf{A}_{n+1} = \mathbf{S}_n^T \mathbf{A}_n \mathbf{S}_n$ and $\mathbf{P}_{n+1} = \mathbf{S}_n^T \mathbf{P}_n$, and use these for the next step.

Once the off-diagonal elements of \mathbf{A}_n are sufficiently small, the diagonal elements of \mathbf{A}_n will contain the eigenvalues of \mathbf{A} , and the columns vectors of \mathbf{P}_n will contain the corresponding eigenvectors.

3 Results



4 Conclusions

A Appendix

All files used in this project can be found at <https://github.com/frxstrem/fys3150/tree/master/project2>. The following code files are used:

- *none*

B Bibliography

- [1] Computational Physics I FYS3150. *Project 2*. 2016. URL: https://github.com/CompPhysics/ComputationalPhysics/blob/master/doc/Projects/2016/Project2/project2_2016.pdf.
- [2] Conrad Sanderson and Ryan Curtin. “Armadillo: a template-based C++ library for linear algebra.” In: *Journal of Open Source Software* vol. 1 (2016), p. 26.