Computational Physics - Project 2

Johannes Scheller, Vincent Noculak, Lukas Powalla

September 23, 2015

Contents

1 Introduction And Motivation

In many fields of both mathematics and physics, we often come to the point that we have to solve so-called eigenvalue problems, which are equations of the form $\hat{A} \cdot \hat{v} = \lambda \hat{v}$, where $\hat{A}$ is a matrix of dimension $n \times n$ and v is a vector of dimension n. Equations of this kind occur not only in linear algebra, but also in mechanics and quantum mechanics and will also be a major part of this report. In this project, we are going to rewrite the Schrödinger's equation of one and two electrons in a harmonic oscillator potential in the form of an eigenvalue problem and solve it numerically by implementing Jacobi's method, an algorithm that can be used to solve any eigenvalue problem.

2 Theory

2.1 Rewriting Schrödinger's equation as eigenvalue problem

2.1.1 One electron in a harmonic oscillator potential

2.1.2 Two interacting electrons in a harmonic oscillator potential

2.2 Jacobi's method

3 Execution

3.1 Implementing the algorithm

3.2 Setting and Testing of Parameters

In both cases, whether we deal with only one particle or with two, we have three parameters to be set, resulting in two degrees of freedom that have an effect on the accuracy of our results. The first and most obvious parameter is n, the number of grid points we use. Using a higher value of n, we gain more precision as the step length decreases, but at the same time, we will end up with a larger matrix that needs more memory (proportional to n^2). Most important, the number of similarity transformations needed to calculate the eigenvalues goes like n^3, leading to a very long computation time for large matrices. The highest possible value we used for n was 1000, resulting in more than 45 minutes of computation time!

The second parameter that we can alter is ρ_{max}, the maximum value of ρ. In theory, this value should be infinite, which is just not possible for this numeric solution. In our case, the higher an eigenvalue is, the more its calculation depends on the choice of ρ_{max}. Therefore the challenge was to set this parameter to a value which resulted in stable and consistent results for the first three eigenvalues without being to high, as a higher value would also increase our step length h if we don't change n accordingly.

The last degree of freedom is to set the tolerance for the non-diagonal matrix elements that are supposed to become zero. This value determines implicitly how many similarity transformations are being operated until the non-diagonal elements are considered zero. A smaller value can lead to higher precision in the eigenvalues, but will at the same time increase the computation time again.

We tested different set-ups with different values of n, rho_{max} and $epsilon$ with the results shown in table **??**. As we wanted a precision of three leading digits for the three lowest eigenvalues, we decided to use the set-up with n =, rho_{max} = and $epsilon$ =, which seemed to be a good compromise between precision and computation time and led to the desired results.

3.3 Results

4 Comparison and discussion of the results