Fys4150
Project 2

Peter Killingstad and Karl Jacobsen

https://github.com/kaaja/fys4150

September 29, 2017

Abstract

1 Introduction

2 Theory

2.1 Scaling

What is scaling , how is it done, what is its effect? Finally the unscaled and scaled equations of the project.

2.2 Vectorization

Couple of sentences about what vectorization inveolves, and how it is done in c++ (compiler commands).

2.3 Orthogonal tranformation properties

In at least one of the algorithms that will be used in this report, orthogonal transformations will be applied. A central property of orthogonal transformations is that they preserve the dot product and orthogonality. This will now be shown.

We have an orthogonal tranformation

$$A = Q^T U Q, \tag{1}$$

$$Q^T Q = I. \tag{2}$$

The orthogonal tranformation of our orthogonal unit vectors is

$$\mathbf{w}_i = \mathbf{U}\mathbf{v}_i,$$

Now lets calculate the dot product of the orthogonally transformed unit vectors.

$$\begin{aligned}
\mathbf{w}_j^T \mathbf{w}_j &= (\mathbf{U}\mathbf{v}_j)^T \mathbf{U}\mathbf{v}_i && \text{(3a)}\\
&= \mathbf{v}_j^T \mathbf{U}^T \mathbf{U}\mathbf{v}_i && \text{(3b)}\\
&= (\mathbf{v}_j)^T \mathbf{v}_i && \text{(3c)}\\
&= \delta_{ij}, && \text{(3d)}
\end{aligned}$$

so the dot-product of the transformed orthogonal unit vectors is unchanged by the transformation.

2.4 Jacoi's algorithm

2.5 The Sturm Bisection method

This is a method for computing the eigenvalues of a symmetric tridiagonal matrices. The eigenvalues are given by the solutions to the characteristic polynomial, which is the determinant of $A - \lambda I$. This is normally inefficient to compute, but in the case of a symmetric trdiagonal matrix, this is not necessarely the case.

The characteristic polynomial can be written as a Sturm sequence, which sloppily put is a recursive equation of the two previous polynomials. By 'previuous' polynomials we mean the polynomials for the determinant of the submatrices of one and two orders below.

Sturms theorem says that the number of sign changes in the polynomial, when calculating all Sturm sequences, which means for all matrix sizes, for a given guess of the eigenvalue, equals the number of eigenvalues less than the guess eigenvalue.

3 Results

3.1 Algorithm efficiency

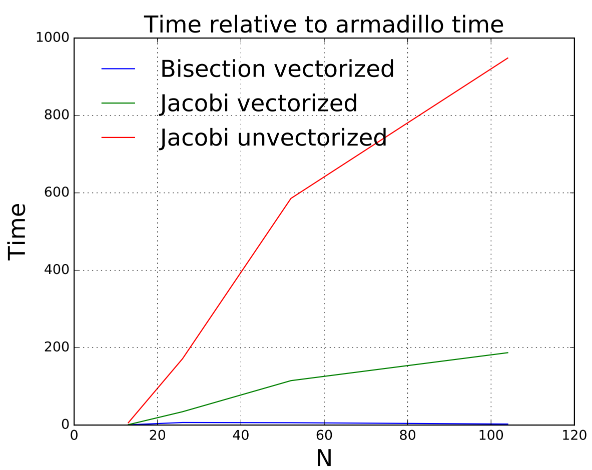

Figure 1: Algorithm times divided by Armadillo time

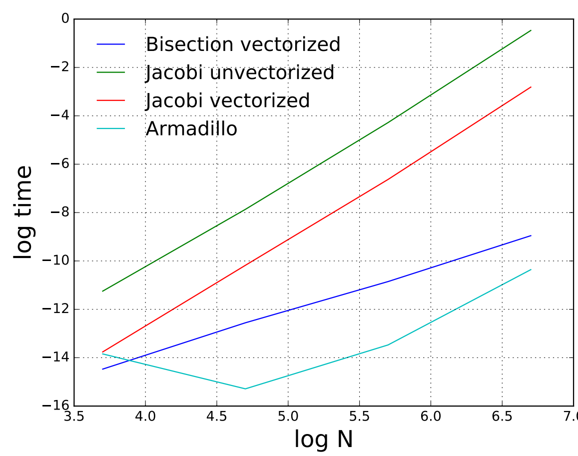

Figure 2: log Algorithm times

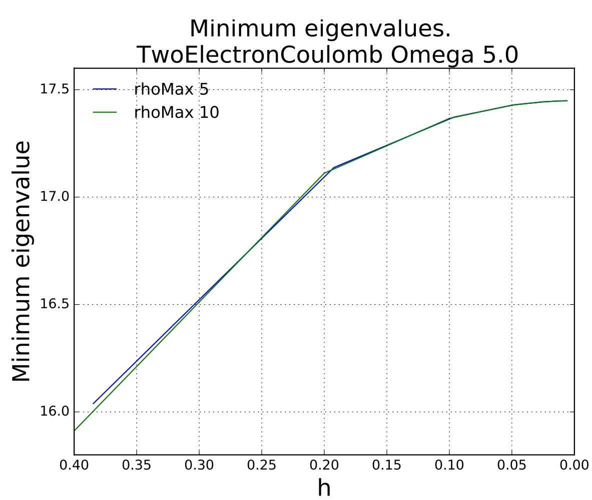

Figure 3: Comparison rhoMax. Minimum eigenvalue. Coulomb

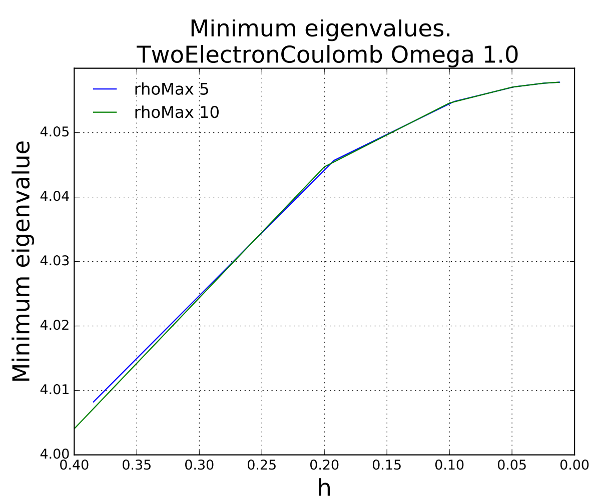

Figure 4: Comparison rhoMax. Minimum eigenvalue. Coulomb

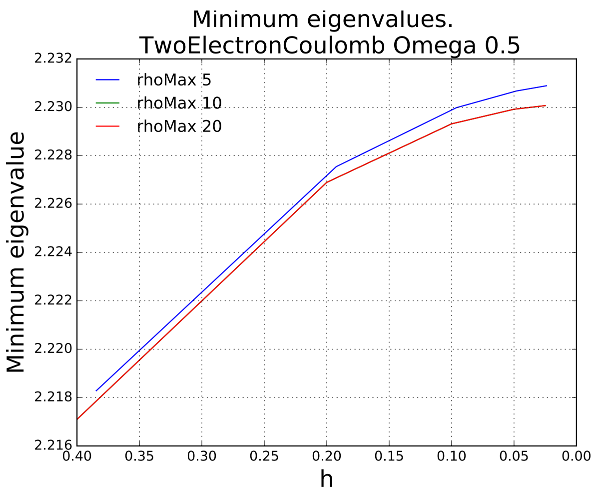

Figure 5: Comparison rhoMax. Minimum eigenvalue. Coulomb

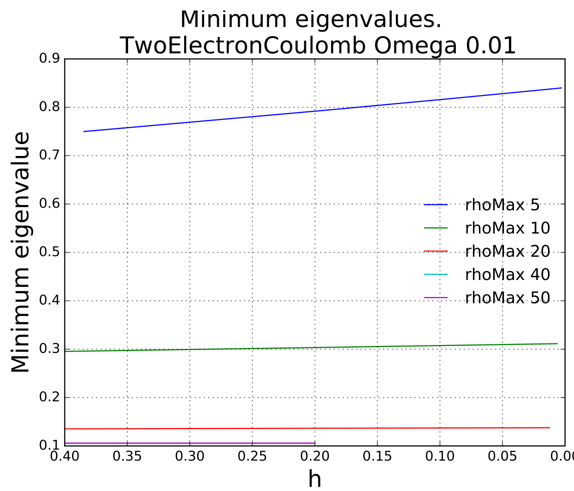

Figure 6: Comparison rhoMax. Minimum eigenvalue. Coulomb

3.2 Sturm bisection

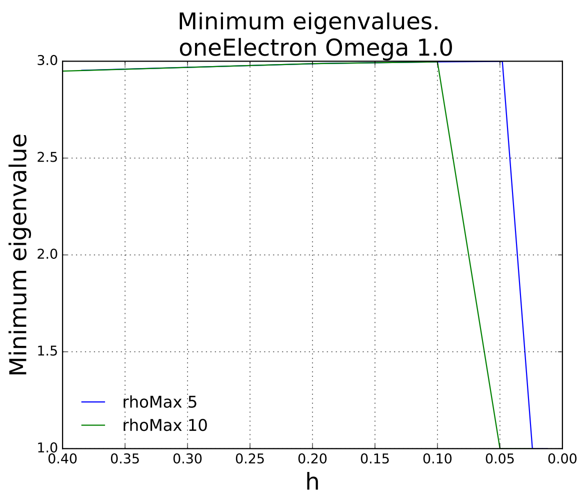

Figure 7: Minimum eigenvalue. One electron Sturm Bisection.

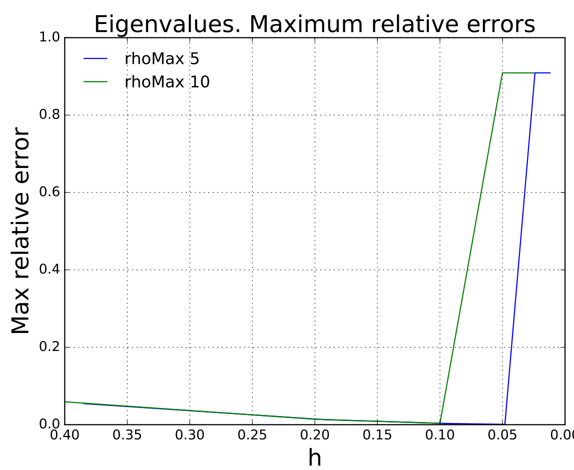

Figure 8: Maximum relative error of three smallest eigenvalues. One electron. Sturm Bisection

3.3 Comparison with theory

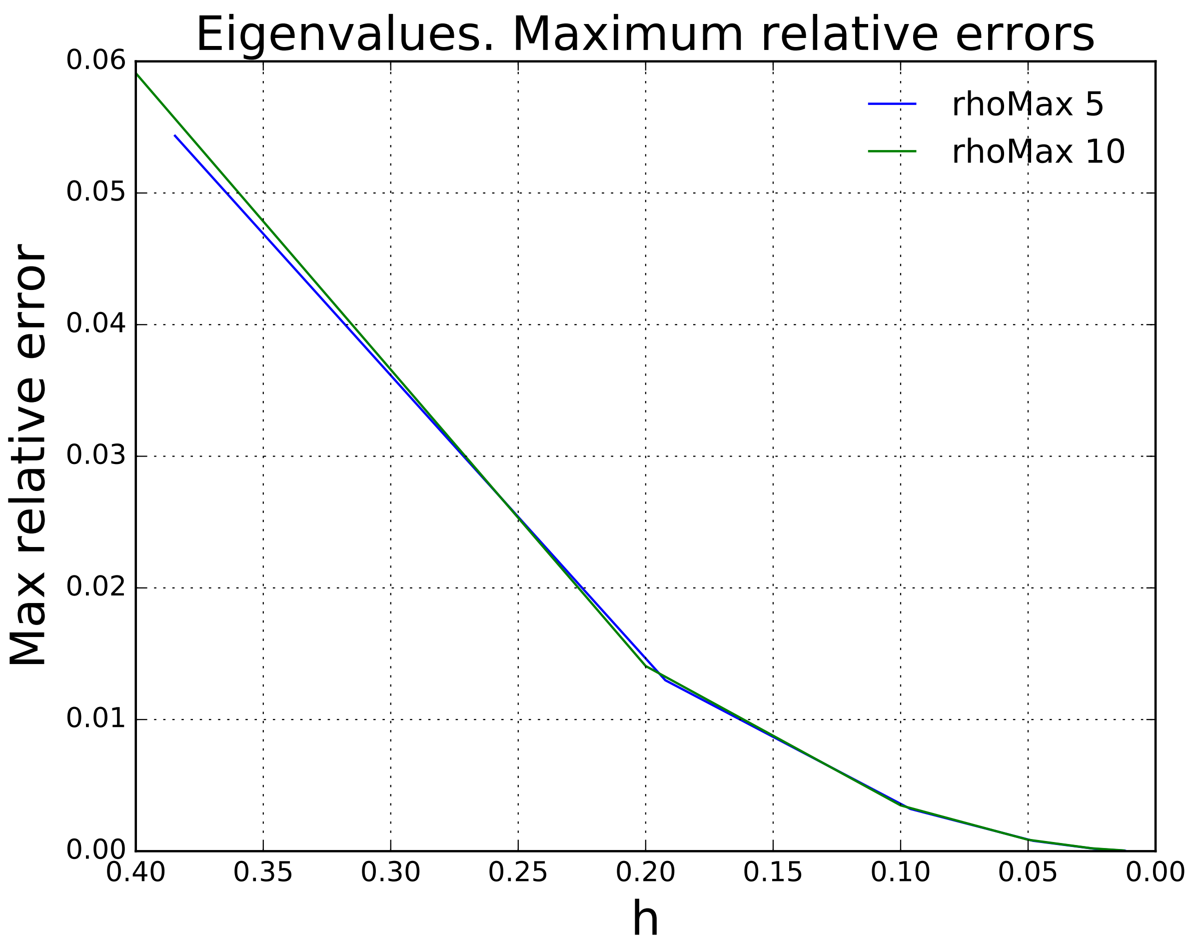

Figure 9: Maximum relative error of the three smallest eigenvalues, one electron.

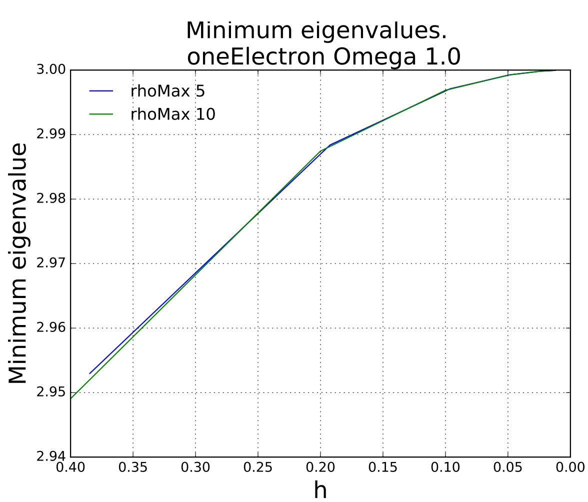

Figure 10: Minimum eigenvalue. 1 electron.

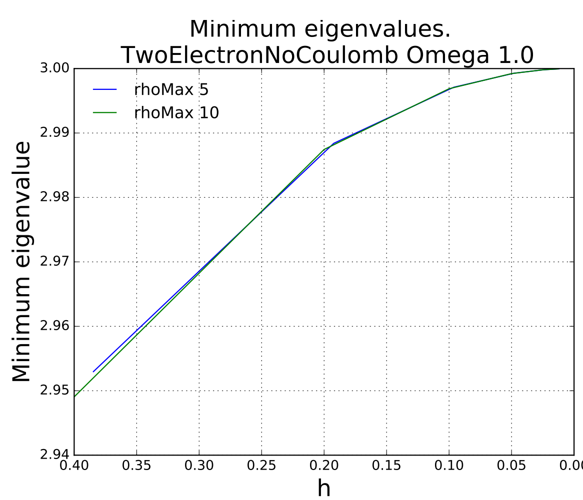

Figure 11: Minimum eigenvalue. 2 electrons no Coulomb.

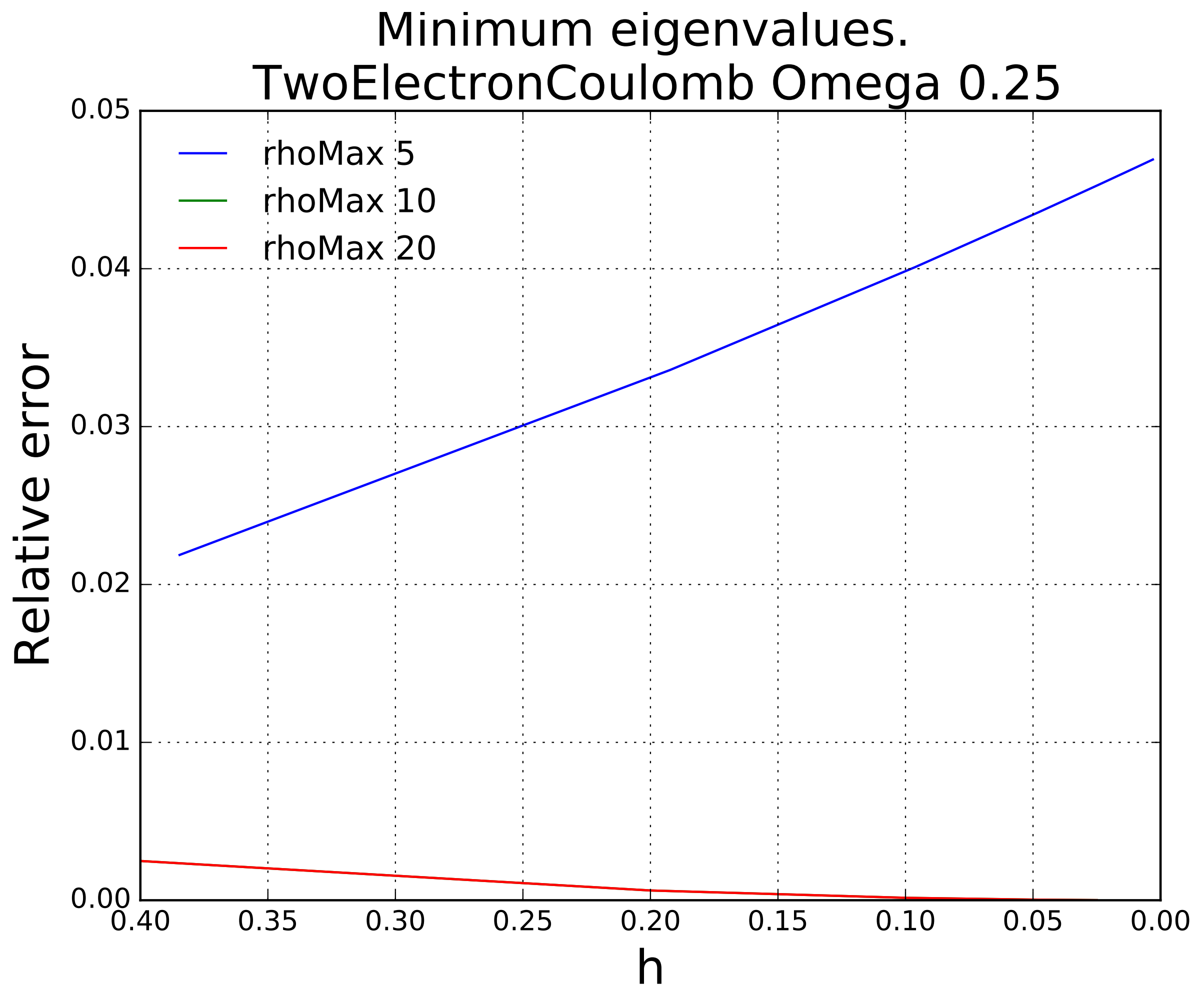

Figure 12: Relative error smallest eigenvalue.

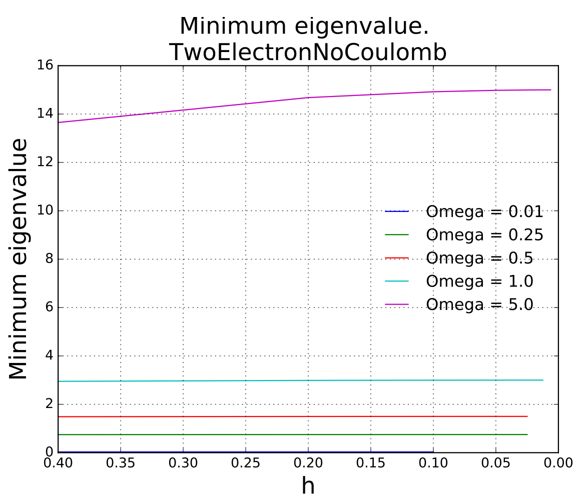

Figure 13: Minimum eigenvalue. 2 electron. No Coulomb.

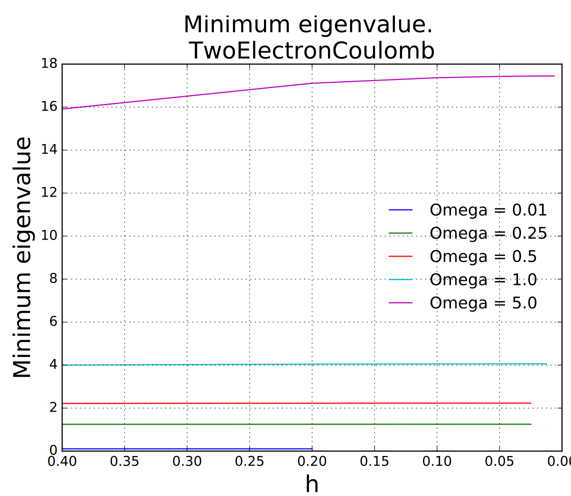

Figure 14: Minimum eigenvalue. 2 electrons Coulomb.

4 Conclusions

5 Feedback

5.1 Project 1

This project has been extremely educational. We learned about about c++, especially pointers and dynamic memory allocoation. Also which for us was a well forgotten subject, we learned about dangerous of numerical round-off errors.

We feel the size of the project is large, much larger than typical assignments in other courses. However, the quality and quantity of the teaching without a doubt made the workload managable. The detailed lectures, combined with the fast and good respones on Piazze helped a lot!

We think the project could have gone even smoother, if we on the 2nd lab-session had learned basic branching in Github. We used a considerable amount of time finding out of this.

All in all, two thumbs up!

5.2 Project 2

6 Bibliography

[1] Hjorth-Jensen, M.(2015) *Computational physics. Lectures fall 2015.* `https://github.com/CompPhysics/ComputationalPhysics/tree/master/doc/Lectures`

[2] Watkins, D.S.(2002) *Fundamentals of matrix computations. 2nd edition.*

[3] Kiusalaas, J.(2013) *Numerical Methods in Engineering with Python 3. 3rd edition.*