Project 2:Jacobis method, Fys 4150

Ask J. Markestad, Thorbjørn V. Larsen

Abstract

Jacobis method

Introduction

Theory and Algorithms

$$\begin{pmatrix} 2 & -1 & 0 & ... & ... & 0 \\ -1 & 2 & -1 & 0 & ... & 0 \\ 0 & -1 & 2 & -1 & 0 & ... \\ ... & ... & ... & ... & ... & ... \\ 0 & ... & 0 & -1 & 2 & -1 \\ 0 & 0 & ... & 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ ... \\ ... \\ u_n \end{pmatrix} = \begin{pmatrix} f_1h^2 \\ f_2h^2 \\ f_3h^2 \\ ... \\ ... \\ f_nh^2 \end{pmatrix} \tag{1}$$

A. Memory handling and algorithms

Source code and accompanying codes can be found at the git hub address:
https://github.com/ajmarkestad/Fys4150/tree/master/Project2

```
//general forward algorithm
for (int i=1; i<=n; i++)
        {
        b[i]=b[i]-c[i-1]*b[i-1]/b[i-1];
        f[i]=f[i]-c[i-1]*f[i-1]/b[i-1];
        }
```

B. Unit-tests

There exist certain mathematical properties that could be exploited to make sure the program and the algorithms run correctly. Since the transformations that occur in the Jacobi's method are either orthogonal or unitary, one can see that the inner product of a given matrix will stay invariant. Under the orthogonal transformation U one has

$$v^Tv = v^TU^TUv = (Uv)^T(Uv) = w^Tw \tag{2}$$

while for a unitary transformation W and general complex v

$$v^\dagger v = v^\dagger W^\dagger Wv = (Wv)^\dagger(Wv) = w^\dagger w \tag{3}$$

We see that under these transformation the inner product is conserved. We can also check whether orthogonality also is conserved. In the initial basis $\{u_i\}$ the orthogonality relation $u_j^\dagger u_i = \delta_{ij}$. We transform as earlier

$$\delta_{ij} = u_j^\dagger u_i = u_j^\dagger W^\dagger W u_i = w_j^\dagger w_i \tag{4}$$

which we see has the same property. We can use these identities to construct tests during Jabobis method calculation, to ensure that machine error in representing numbers not will perturb the results after running a certain time. In practice this means that for every n'th iteration of the transformation we calculate the inner product of 2 random nonequal eigenvectors to check that they are orthonormal, and 2 random equal eigenvectors to check that the inner product is conserved. In this setting conserved is a test that the difference is smaller that a given tolerance (in our case $\epsilon = 10^{-4}$). There is also another set of tests that are useful while constructing the programs. These tests rely on simple constructed problems that are solved analytically and compared to the solutions the algorithms give. As we want to find eigenvalues and vectors, we can construct the matrix

$$\begin{pmatrix} 3 & \sqrt{2} \\ 0 & -1 \end{pmatrix}, \quad \lambda_1 = -1, \quad \lambda_2 = 3, \quad v_1 = \begin{pmatrix} -\frac{1}{2\sqrt{2}} \\ 1 \end{pmatrix}, \quad v_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{5}$$

As a solution the program gave

Results

Conclusion

References

[1] Morten Hjort-Jensen *Computational Physics Lecture Notes Fall 2015* Department of Physics, University of Oslo 2015 https://github.com/CompPhysics/ComputationalPhysics/blob/master/doc/Lectures/lectures2015.pdf