Project 1 in FYS-3150

Gullik Vetvik Killie

September 27, 2014

1 Task a: Implement the Jacobi method to solve eigenvalue problems

1.1 Theory behind the Jacobi method

The Jacobi method is an iterative method to make an approximated diagonal matrix in an eigenvalue problem by multiplying it several times with a rotational matrix, **S**, that is chosen so it sets some off diagonal elements to 0. When multiplying it with the rotational matrix some of the already 0 elements may get a nonzero value, so by always choosing the largest off-diagonal element hopefully it will produce a near diagonal end matrix. By also doing the rotation transformation on the right and side the equation will be equal on both sides throughout and we can extract the eiganvalues easily at the end.

$$\mathbf{A}\vec{\mathbf{x}} = \lambda\vec{\mathbf{x}} \tag{1}$$

Doing a transformation with the rotation matrix $\mathbf{S} = \begin{pmatrix} 1 & & & & \\ & \ddots & & 0 & \\ & & \cos\theta & \cdots & -\sin\theta \\ & 0 & \vdots & 1 & \vdots \\ & & \sin\theta & \cdots & \cos\theta \end{pmatrix}$, in which we choose θ so the wanted elements in **A** becomes 0

$$\mathbf{S}\mathbf{A}\vec{\mathbf{x}} = \lambda\mathbf{S}\vec{\mathbf{x}} \tag{2}$$

$$\mathbf{S}\mathbf{A}(\mathbf{S}^{-1}\mathbf{S})\vec{\mathbf{x}} = \lambda\mathbf{S}\vec{\mathbf{x}} \tag{3}$$

Then we introduce a new vector $\vec{\mathbf{y}} = \mathbf{S}\vec{\mathbf{x}}$ and a new matrix $\mathbf{B} = \mathbf{S}\mathbf{A}\mathbf{S}^{-1}$, where **B** is more diagonal than **A**

$$\mathbf{B}\vec{\mathbf{y}} = \lambda\vec{\mathbf{y}} \tag{4}$$

We do this again untill desired level of diagonality of the matrix is achieved

1.2 The Jacobi method computationally

Since the matrix inversion `<http://en.wikibooks.org/wiki/LaTeX/Hyperlinks#.5Chyperref>` and multiplication $O(n^3)$ (`<http://www.ee.ucla.edu/ee236b/lectures/num-lin-alg.pdf>`) (Put in proper references later bibtex) used in the Jacobi method is unecessarily heavy for the sparse rotational matrix **S** we used a quicker method to do it.

For a simplified 3×3 symmetric system the transformation $\mathbf{B} = \mathbf{S}\mathbf{A}\mathbf{S}^{-1}$ becomes:

$$\mathbf{B} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c & -s \\ 0 & s & c \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ a_{13} & a_{23} & a_{33} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & c & s \\ 0 & -s & c \end{pmatrix} \tag{5}$$

where $c = \cos\theta, s = \sin\theta$

$$\tag{6}$$

$$= \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ ca_{12} - sa_{13} & ca_{22} - sa_{23} & ca_{23} - sa_{33} \\ sa_{12} + ca_{13} & sa_{22} + ca_{23} & sa_{23} + ca_{33} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & c & s \\ 0 & -s & c \end{pmatrix} \tag{7}$$

$$= \begin{pmatrix} a_{11} & c(a_{12}) - s(a_{13}) & s(a_{12}) + c(a_{13}) \\ ca_{12} - sa_{13} & c(ca_{22} - sa_{23}) - s(ca_{23} - sa_{33}) & s(ca_{22} - sa_{23}) + c(ca_{23} - sa_{33}) \\ sa_{12} + ca_{13} & c(sa_{22} + ca_{23}) - s(sa_{23} + ca_{33}) & s(sa_{22} + ca_{23}) + c(sa_{23} + ca_{33}) \end{pmatrix} \tag{8}$$

$$= \begin{pmatrix} a_{11} & ca_{12} - sa_{13} & sa_{12} + ca_{13} \\ ca_{12} - sa_{13} & c^2a_{22} + s^2a_33 - 2sca_{23} & a_{23}(c^2 - s^2) + sc(a_{22} - a_{33}) \\ sa_{12} + ca_{13} & a_{23}(c^2 - s^2) + sc(a_{22} - a_{33}) & s^2a_{22} + c^2a_{33} + 2sca_{23} \end{pmatrix} \tag{9}$$

Then we choose θ so that $B_{23} = 0$

$$\mathbf{B} = \begin{pmatrix} a_{11} & ca_{12} - sa_{13} & sa_{12} + ca_{13} \\ ca_{12} - sa_{13} & c^2a_{22} + s^2a_33 - 2sca_{23} & 0 \\ sa_{12} + ca_{13} & 0 & s^2a_{22} + c^2a_{33} + 2sca_{23} \end{pmatrix} \tag{10}$$

All the components of $\mathbf{B}$ is now known so they can be calculated straightforward for θ. If we precalculate $\cos\theta$ and $\sin\theta$ the flops needed to calculate $\mathbf{B}$ will be:

- First row: 4 multiplications and 2 additions
- Second row: 10 multiplications and 3 additions
- Third row: 10 multiplications and 3 additions

My implementation of the Jacobi method is in the github folder git@github.com:Gullik/Fys3150Gullik.git and is contained in the file JacobiRot.cpp. It works by calling the JacobiRot() function and it then subsequently finds the largest offdiagonal element by the function MaxOffDiag(), and then does the rotation by the Rotation function.

2 Task b: Playing with the Jacobi algorithm on the Harmonic oscillator

2.1 Steps vs well edge; N vs ρ_{max}

The number of steps, N, decides the and how finegrained our approximation of the 3D potential well is. $\rho_{max}$is also important for how good the approximation becomes, since the wave function is supposed to approach 0 at infinity distance away and at 0, and ρ_{max} is where we define that infinity to be. Since the potential is decreasing by r^2(?) it is decreasing quite fast and ρ_{max} does not need to be very large for it to cover most of the wavefunction. Unfortunately as we increase the area we cover by letting ρ_{max} grow, we need more steps, N to be able to cover the area inbetween to good enough detail and a large N slows down the Jacobi algorithm to a great extent. In table **??** (Reference numbering wrong) the needed number of steps needed to achieve 4 signaficant figures correct on the first three eigenvalues of a three dimensional harmonic oscillator, $\lambda_n = 3, 7, 11, ...$

$\rho_{\mathbf{max}}$	1	2.5	5	10	2
N	-	-	175	-	2

Table 1: The steps needed to get 4 signficant figures on the first three eignvalues for different values of ρ_{max}

N	10	50	100
eig_sym	1.25e-4s	0.001431s	0.004766s
Jacobi_Rot	4.3e-5s	0.019119s	0.349412s
Rotations	13		

Table 2: The time test where done with $\rho_{max} = 20$ and a tolerance of $\epsilon = 0.001$, value of ρ_{max} decides the sparseness of the matrix, while ϵ decides the accuracy. With the chosen tolerance the eigenvalues agreed with the ones from the Armadillo solver for 6 significant figures for most configurations. Rotations is the number of applications of the symmetric transform that was applied by the Jacobi_Rot before the offdiagonal elements was below the tolerance ϵ

2.2 Comparing it with the solver from the armadillo library