Mira Mors and Elias Tidemand Ruud

Computational Physics I FYS3150/FYS4150

Department of Physics
University of Oslo
Norway

September 2020

Contents

1 Abstract

2 Introduction

3 Methods

3.1 Mathematical properties of unitary transformations

Perservation of orthonomality A unitary transformation preserves the orthogonality of the obtained eigenvectors. Given a symmetric matrix $\mathbf{A}$ with a set of eigenvectors $\mathbf{v_i}$,

$$\mathbf{v}_i = \begin{bmatrix} v_{i1} \\ \cdots \\ \cdots \\ v_{in} \end{bmatrix}$$

that are orthonormal

$$\mathbf{v}_j^T \mathbf{v}_i = \delta_{ij}, \tag{1}$$

let the unitary transformation be defined by $\mathbf{S}$ such that $\mathbf{S^T S} = I$.
Then, their inner product also preservers orthonomality

$$\mathbf{S(v_i)^T(Sv_j) = v_i^T S^T S v_j = v_i^T I v_j = v_i^T v_j} = \delta_{ij}, \tag{2}$$

there δ is the Kronecker delta.

Perservation of eigenvalues Given the matrices $\mathbf{A}$, its set of eigenvalues $\{\lambda_i\}$ and eigenvectors $\{\mathbf{v_i}\}$, and $\mathbf{S}$ as defined above, a similar matrix $\mathbf{B}$

$$\mathbf{B} = \mathbf{S}^T \mathbf{AS} \tag{3}$$

has the same eigenvalues as $\mathbf{A}$. Multiplying

$$\mathbf{Av_i} = \lambda \mathbf{v_i} \tag{4}$$

with $\mathbf{S^T}$ on the left side and inserting $\mathbf{S^T S} = I$ between $\mathbf{A}$ and $\mathbf{v_i}$ results in

$$(\mathbf{S^T AS})(\mathbf{S^T v_i}) = \lambda(\mathbf{S^T v_i}) \tag{5}$$

which is the same as

$$\mathbf{B}(\mathbf{S^T v_i}) = \lambda(\mathbf{S^T v_i}). \tag{6}$$

In other words, $\mathbf{B}$ has eigenvectors of the form $\mathbf{S^T v_i}$ with eigenvalues λ_i.

Similar transform eigenvectors er orthonormal If a matrix $\mathbf{A}$ has orthonormal eigenvectors, then a similar $\mathbf{B}$ has orthonormal eigenvectors as well. Taking use of the results above, that is using that $\mathbf{B}$ has eigenvectors of the form $\mathbf{S^T v_i}$, their dot product is given by

$$(\mathbf{S^T v_i})^T(\mathbf{S^T v_j}) = (\mathbf{v_i^T S})(\mathbf{S^T v_j}) = \mathbf{v_i^T v_j} = \delta_{ij}. \tag{7}$$

3.2 Jacobi Method

The Jacobi method guarantees a solution for all real symmetric matrices. The method rotates the initial matrix in a series of similarity transformations. By discarding the off-diagonal elements, the eigenvalues stay preserved along the diagonal. The corresponding eigenvectors, however, do change. For a real and symmetric $n \times n$ matrix $\mathbf{A}$ we get

$$\mathbf{S}^T\mathbf{A}\mathbf{S} = diag(\lambda_1\lambda_2 \dots \lambda_n). \tag{8}$$

there $\mathbf{S}$ is the transformation matrix and λ are the eigenvalues.
In order to understand this transformation, we first look at the transformation matrix $\mathbf{S}$, also called a unitary matrix. The transformation matrix is orthogonal since its inverse is equal to its transpose

$$\mathbf{S}^T = \mathbf{S}^{-1}.$$

In $\mathbb{R}^2$, the transformation matrix is given by

$$\mathbf{S} = \begin{bmatrix} cos(\theta) & sin(\theta) \\ -sin(\theta) & cos(\theta) \end{bmatrix}$$

and θ denotes the degree for the plane rotation in the Euclidean two dimensional space.
In $\mathbb{R}^3$, the rotation along the x, y and z axis respectively are

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\theta) & sin(\theta) \\ 0 & -sin(\theta) & cos(\theta) \end{bmatrix} \quad R_y = \begin{bmatrix} cos(\theta) & 0 & sin(\theta) \\ 0 & 1 & 0 \\ -sin(\theta) & 0 & cos(\theta) \end{bmatrix} \quad R_z = \begin{bmatrix} cos(\theta) & sin(\theta) & \\ -sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Generally, a rotation i $\mathbb{R}^n$ is given by

$$S(k,l,\theta) = \begin{array}{c@{}c} \begin{array}{ccccccc} & & l & & k & & \end{array} & \\ \begin{bmatrix} 1 & \dots & 0 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \dots & cos(\theta) & \dots & sin(\theta) & \dots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \dots & -sin(\theta) & \dots & cos(\theta) & \dots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & 0 & \dots & 1 \end{bmatrix} & \begin{array}{c} \\ \\ l \\ \\ k \\ \\ \\ \end{array} \end{array}$$

$cos(\theta)$ and $sin(\theta)$ are now placed in the lth row and lth column and kth row and kth column.
The similarity transformation

$$\mathbf{A'} = \mathbf{S}^T(k,l,\theta) \cdot \mathbf{A} \cdot \mathbf{S}(k,l,\theta) \tag{9}$$

rotates row and column k and l of $\mathbf{A}$ an angle θ such as the entries $\mathbf{A}'(k,l)$ and $\mathbf{A}'(l,k)$ become zero. The new entries for $\mathbf{A}'$ are given with

$$\begin{aligned} a'_{ii} &= a_{ii} \quad i \neq k, i \neq l \\ a'_{ik} &= a_{ik}cos(\theta) - a_{il}sin(\theta) \quad i \neq k, i \neq l \\ a'_{il} &= a_{il}cos(\theta) + a_{ik}sin(\theta) \quad i \neq k, i \neq l \\ a'_{kk} &= a_{kk}cos^2(\theta) - 2a_{kl}cos(\theta)sin(\theta) + a_{ll}sin^2(\theta) \\ a'_{ll} &= a_{ll}cos^2(\theta) + 2a_{kl}cos(\theta)sin(\theta) + a_{kk}sin^2(\theta) \\ a'_{kl} &= (a_{kk} - a_{ll})cos(\theta)sin(\theta) + a_{kl}(cos^2(\theta) - sin^2(\theta)) \end{aligned} \tag{10}$$

The aim is to get zero for all non-diagonal elements a'_{kl}. That is

$$a'_{kl} = (a_{kk} - a_{ll})cos(\theta)sin(\theta) + a_{kl}(cos^2(\theta) - sin^2(\theta)) = 0. \tag{11}$$

For each iteration, θ must be chosen accordingly. Given the quantity $tan(\theta) = sin(\theta)/cos(\theta)$, abbreviated with $t = s/c$, and defining

$$\tau = \frac{a_{ll} - a_{kk}}{2a_{kl}}, \tag{12}$$

we get

$$t^2 + 2\tau t - 1 = 0. \tag{13}$$

The solutions for t are

$$t = -\tau \pm \sqrt{1 + \tau^2}. \tag{14}$$

In order to ensure maximal numerical stability, the smallest root is chosen, corresponding to the smaller rotation. Consequently, c and s can be expressed as

$$c = \frac{1}{\sqrt{1 + t^2}} \tag{15}$$

and $s = t/c$.
In the case of $a_{kl} = 0$, we have $cos(\theta) = 1$ and $sin(\theta) = 0$.
The Jacobi Method is an iterative method, thus this procedure is continued until the sum over the squared non-diagonal matrix elements, $off(\mathbf{A})$, are less than a prefixed test, ϵ, (ideally equal zero). That is

$$off(\mathbf{A}) = \sqrt{\sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} = |a_{ij}|^2} < \epsilon \tag{16}$$

Since this is quite a time-consuming test, it can be replaced by finding the largest (absolute) value of the off-diagonal elements

$$max|a_{ij}| < \epsilon \quad i \neq j. \tag{17}$$

Meaning i and j become the new indices equivalent to k and l. Even though $\mathbf{A}'(k,l)$ and $\mathbf{A}'(l,k)$ are set to zero for one iteration, these entries may become different from zero in the next iteration. This is one of the reason for slowing down the algorithm. The convergence rate can be approximated with $3n^2 - 5n^2$ rotations, there each rotation requires $4n$ operations. In other words, in order to zero out the non-diagonal matrix elements, it takes about $12n^3 - 20n^3$ operations.

3.2.1 Code Implementation

Algorithm 1 Jacobi's method

```
1: while $max_{i \neq j}|a_{ij}| < \epsilon$ do
2:     $a_{kl} = max_{i \neq j}|a_{ij}$
3:     if $a_{lk} \neq 0$ then
4:         $\tau = (a_{ll} - a_{kk})/(2.0 a_{kl})$
5:         if $\tau \geq 0$ then
6:             $tan = 1.0/(\tau + \sqrt{1.0 + \tau \cdot \tau})$
7:         else
8:             $t = -1.0/(-\tau + \sqrt{1.0 + \tau \cdot \tau})$
9:         end if
10:        $cos = 1.0/\sqrt{1.0 + tan \cdot tan}$
11:        $sin = cos \cdot tan$
12:    else
13:        $cos = 1.0$
14:        $sin = 0.0$
15:    end if
16:    for $i = 0, 1, ..., n-1$ do
17:        $a'_{kk} = a_{kk} cos^2 - 2a_{kl} cos \cdot sin + a_{ll} sin^2$
18:        $a'_{ll} = a_{ll} cos^2 + 2a_{kl} cos \cdot sin + a_{kk} sin^2$
19:        $a'_{kl} = 0$
20:        $a'_{l,k} = 0$
21:        if $i \neq k \ \& \ i \neq l$ then
22:            $a'_{ik} = a_{ik} cos - a_{il} sin$
23:            $a_{ki} = a_{ik}$
24:            $a'_{il} = a_{il} cos + a_{ik} sin$
25:            $a'_{li} = a_{il}$
26:        end if
27:    end for
28: end while
```