Prof. Dr. Manuel Torrilhon
Dr. Robert Speck
Tuan Vo
Edilbert Christhuraj

Applied and Computational Mathematics

RWTH AACHEN UNIVERSITY

Mathematische Grundlagen III (CES) | WS 2022
Programmierübung 3 | 15.12.2022

Hinweise zur Abgabe einer Programmieraufgabe (bitte sorgfältig lesen!):

- Die Programmieraufgaben sind in **Dreier- bis Fünfergruppen** zu bearbeiten.
- Vor Abgabe der Aufgabe müssen alle Programmverifikationen erfolgreich sein.
- Zum Testieren tragen Sie sich bitte ab **heute bis 12.01.2023 14:00 Uhr** in den Terminplaner im Moodle ein.
- Der Testattermin findet am **13.01.2023 (Fr) 14:30–18:30 Uhr + Moodle-Termine** in **Seminarraum 328, Rogowski Gebäude** statt.
- Bringen Sie zu dem Testattermin bitte Ihr lauffähiges `matlab`-Programm mit.
- Sie werden dazu aufgefordert werden Ihren Bildschirm zu teilen.

Computational eigenvalues based on QR-Algorithm with and without spectral shift

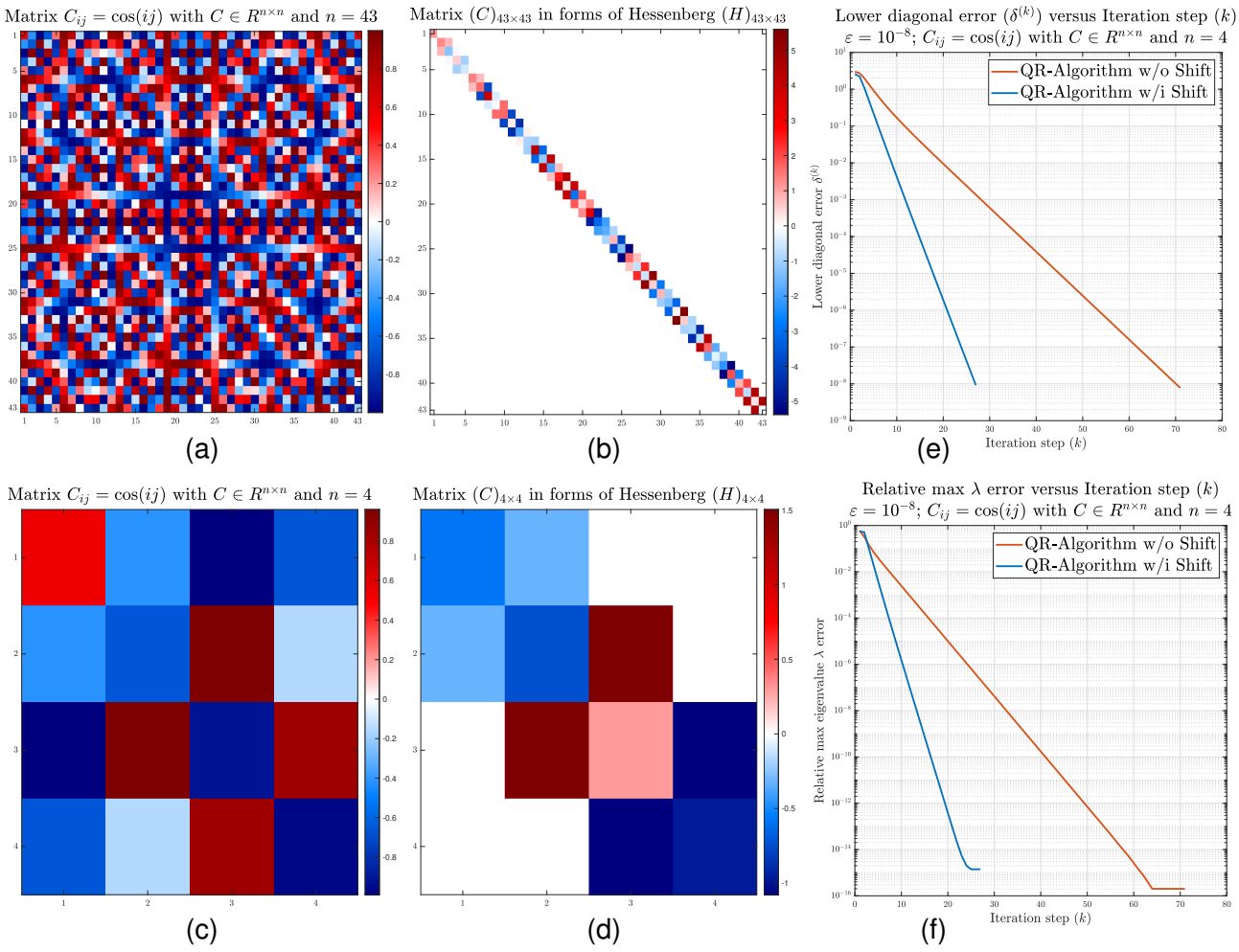

Abbildung 1: An example of QR-Algorithm with and without spectral shift for $C_{ij} = \cos(ij)$:
(a) Matrix C with $C \in \mathbb{R}^{43\times 43}$;
(b) *Hessenberg*-matrix form of $(C)_{43\times 43}$;
(c) Matrix C with $C \in \mathbb{R}^{4\times 4}$;
(d) *Hessenberg*-matrix form of $(C)_{4\times 4}$;
(e) Lower diagonal error $\delta^{(k)}$ versus Iteration step (k);
(f) Relative maximal eigenvalue error versus Iteration step (k).

Given a matrix $A \in \mathbb{R}^{n \times n}$ which we would like to compute its eigenvalues numerically. The eigenvalues of the matrix A will be computed by using QR-Algorithm either with or without spectral shift. Our program will include two sub programs

$$\text{(i)} \rightarrow \text{(ii) (a)} \rightarrow \text{(iii)}$$
$$\text{(i)} \rightarrow \text{(ii) (b)} \rightarrow \text{(iii)}$$

where the step (i), (ii), and (iii), with (a) and (b) are described in details, as follows

(i) **Hessenberg**-matrix transformation
First, we transform the matrix A to a matrix $A^{(0)}$ similar to A, such that the transformed matrix $A^{(0)}$ becomes a *Hessenberg*-matrix, using *Householder*-Refection.

(ii) **Upper triangular matrix** computation
Then, we will compute iteratively the upper triangular matrix $A^{(k)}$ where

$$A^{(k)} \in \mathbb{R}^{n \times n} \quad \text{for} \quad k = 1, 2, 3, \ldots,$$

by taking into consideration the following two schemes

a) QR-Algorithm **without** spectral shift

$$\boxed{A^{(k)} := R^{(k-1)} Q^{(k-1)},}$$

where $Q^{(k-1)}$ and $R^{(k-1)}$ are the orthogonal matrix and right upper triangular matrix, respectively, obtained by QR-decomposition of the matrix $A^{(k-1)}$ performed at a previous step $(k-1)$, meaning that

$$A^{(k-1)} = Q^{(k-1)} R^{(k-1)}.$$

b) QR-Algorithm **with** spectral shift

$$\boxed{A^{(k)} := R^{(k-1)} Q^{(k-1)} + \gamma^{(k-1)} I,}$$

where $Q^{(k-1)}$ and $R^{(k-1)}$ are the orthogonal matrix and right upper triangular matrix, respectively, obtained by QR-decomposition of the matrix $A^{(k-1)}$ performed at a previous step $(k-1)$, meaning that

$$A^{(k-1)} - \gamma^{(k-1)} I = Q^{(k-1)} R^{(k-1)}.$$

Note in passing that the shift parameter $\gamma^{(k-1)}$ defined as

$$\gamma^{(k-1)} := a_{nn}^{(k-1)} \in \mathbb{R}$$

which is the n-th diagonal entry chosen from matrix $A^{(k-1)}$.

(iii) **Termination criterion** definition
After a number of repetitive executions the *Hessenberg*-matrix $A^{(0)}$ will be transformed into a right upper triangular matrix $A^{(k)}$, at which all eigenvalues will now stay along the principal diagonal. As k progresses the lower diagonal shoud converge to zero. Therefore, the termination criterion is defined as follows

$$\boxed{\delta^{(k)} := \sum_{i=2}^{n} \left| a_{i,i-1}^{(k)} \right| < \varepsilon,}$$

where tolerance limit $\varepsilon = 10^{-8}$ is chosen.

The tasks for this programming exercise is that we will use the QR-Algorithms, with and without spectral shift, described above to compute the eigenvalues of matrices `A`, `B`, and `C` given below.

Programming building
Define matrices `A`, `B`, and `C`, and the tolerance limit `eps` in the main program called `main.m`. Besides, the two following functions

```
function [step,error,Ak] = QRwoShift(A,eps) % QR without shift
function [step,error,Ak] = QRwiShift(A,eps) % QR with shift
```

are called from the main program. Furthermore, the vectors `step` and `error` indicate the iteration step k and the corresponding error $\delta^{(k)}$ of the lower diagonal. Likewise, the three-dimensional matrix `Ak` indicates the matrix $A^{(k)}$ in every iteration step.

Hints:

i) Any other programming language is very much welcome, such as Julia, Python, C/C++, Java, and so on. The `matlab` procedure given above is one example.

ii) Routines available in `matlab` such as `hess(A)` and `qr(A)` can be used to compute the *Hessenberg*-matrix and to perform the QR-decomposition, respectively.

Matrices to be tested
We shall test our program by examining the following three matrices:

i) A naïve matrix $A \in \mathbb{R}^{3\times 3}$ given by

$$\boxed{A = \begin{pmatrix} 2 & -3 & 1 \\ 3 & 1 & 3 \\ -5 & 2 & -4 \end{pmatrix}.}$$

ii) A disturbed matrix $B \in \mathbb{R}^{4\times 4}$ with floating points, given by

$$\boxed{B = \begin{pmatrix} 3.5488 & 15.593 & 8.5775 & -4.0123 \\ 2.3595 & 24.524 & 14.596 & -5.8157 \\ 0.0899 & 27.599 & 21.438 & -5.8415 \\ 1.9227 & 55.667 & 39.717 & -10.558 \end{pmatrix}.}$$

iii) A large-size matrix $C \in \mathbb{R}^{n\times n}$, $n \in \mathbb{N}$ where n can be chosen arbitrarily, and entries of the matrix C is followed by

$$\boxed{C_{ij} = \sin(ij).}$$

To-do list:

(a) Compute the eigenvalues of the above **three** matrices by using QR-Algorithm followed by schemes given above, i.e. QR-Algorithm with and without spectral shift.

(b) Plot the convergence of the two QR-Algorithms where an illustration of error $\delta^{(k)}$ depending on the iteration step (k) is shown.

(c) Plot the convergence of the relative error between the maximal eigenvalue computed by the two QR-Algorithms and the maximal eigenvalue computed by the available `matlab` routine `qr(A)`, depending on the iteration step (k) is shown.

(d) Visualize the formation of the right upper triangular matrix $A^{(k)}$ depending on the iteration step (k).

8+5+5+2 Punkte

Hints:

(a) While programming by other languages beyond `matlab`, the *Hessenberg*-matrix transformation and QR-decomposition rountine in such programming languages can be acquired accordingly.

(b) Regarding the convergence plot in task (b) and (c) the `matlab`-command

```
semilogy
```

can be used. This command will help us to generate a figure whose one of its axes is in log-scale and another is non-log-scale.

(c) By defining

$\lambda_{\text{max}}^{\text{os}}$ Maximal eigenvalue computed by available open-source library

$\lambda_{\text{max}}^{\text{wi}}$ Maximal eigenvalue computed by QR-Algorithm with shift

$\lambda_{\text{max}}^{\text{wo}}$ Maximal eigenvalue computed by QR-Algorithm without shift

the aforementioned relative error in task (c) is computed as follows

$$\boxed{\begin{aligned} E^{\text{wi}} &= \frac{\lambda_{\text{max}}^{\text{wi}} - \lambda_{\text{max}}^{\text{os}}}{\lambda_{\text{max}}^{\text{os}}} \\ E^{\text{wo}} &= \frac{\lambda_{\text{max}}^{\text{wo}} - \lambda_{\text{max}}^{\text{os}}}{\lambda_{\text{max}}^{\text{os}}} \end{aligned}}$$

where E^{wi} and E^{wo} are the corresponding aforementioned relative error for QR-Algorithm with and without shift, respectively.

(d) The following command sequence may be helpful for task (d)

```
imagesc(log(abs(A)));
colormap(turbo); % e.g. We would like to use turbo
colorbar;
drawnow;
```

(e) As an alternative option for task (d), 2 points are also offered for answering extra questions during Testat, and/or for coding the programming exercise by any other languagues beyond `matlab`.