Numerische Lineare Algebra

Oliver Ernst

Professur Numerische Mathematik

Wintersemester 2017/18

Mathematik!
TU Chemnitz

Inhalt I

Inhalt II

Inhalt

Nach dem Satz von Abel-Ruffini sind Polynomgleichungen ab Grad 5 nicht durch Wurzelziehen lösbar. Die Nullstellen des Polynoms

$$p(z) = z^n + a_{n-1}z^{n-1} + \cdots + a_1 z + a_0$$

sind die Eigenwerte seiner Frobenius-Begleitmatrix

$$\boldsymbol{A} = \begin{bmatrix} 0 & & & & -a_0 \\ 1 & 0 & & & -a_1 \\ & \ddots & \ddots & & \vdots \\ & & 1 & 0 & -a_{n-2} \\ & & & 1 & -a_{n-1} \end{bmatrix} \in \mathbb{C}^{n\times n}.$$

Ein Verfahren zur Berechnung von Eigenwerten mit endlich vielen Schritten würde somit zu einer „Formel“ für die Nullstellen eines Polynomes führen.

Das in diesem Kapitel betrachtete Verfahren, die QR-Iteration, berechnet die Schur-Zerlegung einer $n \times n$ Matrix in $O(n^3)$ Operationen.

- „Direkte“ Verfahren zur Berechnung der Eigenwerte einer Matrix $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ beruhen darauf, die Matrix schrittweise in eine Form zu überführen, an der man die Eigenwerte ablesen kann.
- Da Ähnlichkeitstransformationen die Eigenwerte unverändert lassen, kommen diese hierfür in Frage.
- Ähnlichkeitstransformation auf Diagonalform ist nur für diagonalisierbare Matrizen möglich.
- Unitäre Transformation auf Diagonalform (attraktiver, weil numerisch stabiler) ist im Allgemeinen nur für normale Matrizen möglich.

Satz 6.1 (Schur-Zerlegung)

Jede Matrix $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ läßt sich durch unitäre Ähnlichkeitstransformation in obere Dreiecksgestalt überführen, d. h. es existieren eine unitäre Matrix $\boldsymbol{U}$ sowie eine obere Dreiecksmatrix $\boldsymbol{T}$ mit

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{T}\boldsymbol{U}^{\mathsf{H}} \quad \text{bzw.} \quad \boldsymbol{U}^{\mathsf{H}}\boldsymbol{A}\boldsymbol{U} = \boldsymbol{T} \qquad \text{(Schur-Form)}.$$

Ist eine Schur-Zerlegung $\boldsymbol{U}^{\mathsf{H}}\boldsymbol{A}\boldsymbol{U} = \boldsymbol{T}$ vorhanden und gilt $\boldsymbol{T}\boldsymbol{x} = \lambda\boldsymbol{x}$, so ist $\boldsymbol{U}\boldsymbol{x}$ Eigenvektor von $\boldsymbol{A}$ zum Eigenwert λ. Es genügt also, Eigenvektoren von $\boldsymbol{T}$ zu bestimmen.
Ist $\lambda = t_{i,i}$ einfacher Eigenwert von $\boldsymbol{T}$ so gilt $(\boldsymbol{T} - \lambda\boldsymbol{I})\boldsymbol{x} = \boldsymbol{0}$, oder

$$\boldsymbol{0} = \begin{bmatrix} \boldsymbol{T}_{11} - \lambda\boldsymbol{I} & \boldsymbol{T}_{12} & \boldsymbol{T}_{13} \\ \boldsymbol{0}^{\mathsf{T}} & 0 & \boldsymbol{T}_{23} \\ \boldsymbol{O} & \boldsymbol{0} & \boldsymbol{T}_{33} - \lambda\boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ x_2 \\ \boldsymbol{x}_3 \end{bmatrix} = \begin{bmatrix} (\boldsymbol{T}_{11} - \lambda\boldsymbol{I})\boldsymbol{x}_1 + \boldsymbol{T}_{12}x_2 + \boldsymbol{T}_{13}\boldsymbol{x}_3 \\ \boldsymbol{T}_{23}\boldsymbol{x}_3 \\ (\boldsymbol{T}_{33} - \lambda\boldsymbol{I})\boldsymbol{x}_3 \end{bmatrix}$$

Da λ einfach ist, sind $\boldsymbol{T}_{11} - \lambda\boldsymbol{I}$ und $\boldsymbol{T}_{33} - \lambda\boldsymbol{I}$ invertierbar.

Insbesondere folgt aus $(\boldsymbol{T}_{33} - \lambda\boldsymbol{I})\boldsymbol{x}_3 = \boldsymbol{0}$ auch $\boldsymbol{x}_3 = \boldsymbol{0}$ und somit

$$(\boldsymbol{T}_{11} - \lambda\boldsymbol{I})\boldsymbol{x}_1 = -\boldsymbol{T}_{12}x_2.$$

Setzt man $x_2 = 1$, so ergibt sich

$$\boldsymbol{x} = \begin{bmatrix} -(\boldsymbol{T}_{11} - \lambda\boldsymbol{I})^{-1}\boldsymbol{T}_{12} \\ 1 \\ \boldsymbol{0} \end{bmatrix}.$$

Das QR-Verfahren für Eigenwertaufgaben

Vorbemerkungen

- Das QR-Verfahren besteht prinzipiell aus einer (geschickt gewählten) Folge unitärer Ähnlichkeitstransformationen, die $\boldsymbol{A}$ immer näher an Schur-Form bringen.
- Man kann den Rechenaufwand für diese Transformationen reduzieren, indem man $\boldsymbol{A}$ zunächst (ebenfalls unitär ähnlich) auf obere Hessenberg-Gestalt bringt. Der Aufwand hierfür beträgt $O(n^3)$ Rechenoperationen.
- Für unitäre Ähnlichkeitstransformation angewandt auf Hessenberg-Matrizen beträgt der Aufwand $O(n^2)$ Operationen.
- Im Allgemeinen konvergiert das QR-Verfahren dann in $O(n)$ Schritten, woraus sich ein Gesamtaufwand von $O(n^3)$ ergibt.
  (Ohne Hessenberg-Reduktion $O(n^4)$.)
- Auch das QR-Verfahren schaffte es in die Liste der zehn wichtigsten numerischen Algorithmen des 20. Jahrhunderts.

Inhalt

Eine Matrix $\boldsymbol{A} = [a_{i,j}] \in \mathbb{C}^{n\times n}$ heißt obere Hessenberg-Matrix, falls

$$a_{i,j} = 0 \quad \text{falls } i > j+1.$$

Im 5×5-Beispiel:
($\times$ bezeichnet beliebige Einträge, i.A. $\neq 0$)

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix}$$

Satz 6.2

Zur jeder Matrix $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ existiert eine unitäre Matrix $\boldsymbol{U} \in \mathbb{C}^{n\times n}$ sodass $\boldsymbol{U}^{\mathsf{H}}\boldsymbol{A}\boldsymbol{U} = \boldsymbol{H}$ obere Hessenberg-Matrix ist.

Die unitäre Transformation auf obere Hessenberg-Gestalt ist in endlich vielen Schritten möglich (s.u.). Der Schritt zur Schur-Form, d.h. die Elimination der verbleibenden unteren Nebendiagonalen mittels unitärer Transformationen, ist jedoch ein iterativer, i.A. unendlicher Prozess.

Algorithmus 14: Transformation auf Hessenberg-Gestalt.

Gegeben: $\boldsymbol{A} \in \mathbb{C}^{n\times n}$.

1 **for** $k = 1$ **to** $n-2$ **do**
2 $\quad [\boldsymbol{v}, \beta] \leftarrow \text{house}(\boldsymbol{A}(k+1:n, k))$
3 $\quad \boldsymbol{A}(k+1:n, k:n) \leftarrow (\boldsymbol{I} - \beta \boldsymbol{v}\boldsymbol{v}^{\mathsf{H}})\boldsymbol{A}(k+1:n, k:n)$
4 $\quad \boldsymbol{A}(1:n, k+1:n) \leftarrow \boldsymbol{A}(1:n, k+1:n)(\boldsymbol{I} - \beta \boldsymbol{v}\boldsymbol{v}^{\mathsf{H}})$

Dabei liefert die Funktion $[\boldsymbol{v}, \beta] = \text{house}(\boldsymbol{x})$ zu $\boldsymbol{x} \in \mathbb{C}^n$ einen Vektor $\boldsymbol{v} \in \mathbb{C}^n$, den sog. Householder-Vektor, mit $v_1 = 1$ sowie $\beta \in \mathbb{R}$ sodass

$$\boldsymbol{P} = \boldsymbol{I} - \beta \boldsymbol{v}\boldsymbol{v}^{\mathsf{H}} \text{ unitär} \qquad \text{und} \qquad \boldsymbol{P}\boldsymbol{x} = \|\boldsymbol{x}\|_2 \frac{x_1}{|x_1|} \boldsymbol{e}_1.$$

Dieser Algorithmus erfordert $10n^3/3$ Flops, $4n^3/3$ zusätzliche Flops falls $\boldsymbol{U}$ explizit benötigt wird.

Wir zeigen das Vorgehen der Reduktion auf Hessenberg-Gestalt durch Householder-Transformationen anhand einer beliebigen 6×6-Matrix $\boldsymbol{A}$ auf.

Bezeichnet $\tilde{\boldsymbol{P}}_1 \in \mathbb{C}^{(n-1)\times(n-1)}$ eine Householder-Transformation, welche den Vektor $\boldsymbol{A}(2:n,1)$ auf ein Vielfaches von $\boldsymbol{e}_1 \in \mathbb{C}^{n-1}$ abbildet, so ergibt

$$\boldsymbol{P}_1^{\mathsf{H}}\boldsymbol{A} = \begin{bmatrix} 1 & \boldsymbol{0}^{\mathsf{T}} \\ \boldsymbol{0} & \tilde{\boldsymbol{P}}_1 \end{bmatrix} \begin{bmatrix} \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \end{bmatrix} = \begin{bmatrix} \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \end{bmatrix}$$

Anschließende Multiplikation von rechts mit $\boldsymbol{P}_1$ lässt die erste Spalte unverändert, sodass wir mit $\boldsymbol{A}_1 := \boldsymbol{P}_1^{\mathsf{H}}\boldsymbol{A}\boldsymbol{P}_1$ eine zu $\boldsymbol{A}$ unitär ähnliche Matrix erhalten mit Nullen unterhalb der ersten Nebendiagonalen in der ersten Spalte.

Mit der zweiten Spalte verfahren wir analog: bildet die Householder-Matrix $\tilde{\boldsymbol{P}}_2 \in \mathbb{C}^{(n-2)\times(n-2)}$ den Vektor $\boldsymbol{A}_1(3:n,2)$ auf ein Vielfaches von $\boldsymbol{e}_1 \in \mathbb{C}^{n-3}$ ab, so wirkt die unitäre Matrix von links multipliziert

$$\boldsymbol{P}_2^{\mathsf{H}} = \begin{bmatrix} \boldsymbol{I}_2 & \boldsymbol{O} \\ \boldsymbol{O} & \tilde{\boldsymbol{P}}_2 \end{bmatrix}$$

nur auf die Zeilen 3 bis n, analog $\boldsymbol{P}_3$ von rechts multipliziert nur auf Spalten 3 bis n, und wir erhalten

$$\boldsymbol{A}_2 = (\boldsymbol{P}_1\boldsymbol{P}_2)^{\mathsf{H}}\boldsymbol{A}(\boldsymbol{P}_1\boldsymbol{P}_2) = \begin{bmatrix} \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \end{bmatrix}$$

In 2 weiteren Schritten mit immer kleineren Householder-Transformationen erhalten wir

$$\boldsymbol{A}_2 \xrightarrow{\boldsymbol{P}_3} \boldsymbol{A}_3 = \begin{bmatrix} \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times & \times \end{bmatrix} \xrightarrow{\boldsymbol{P}_4} \boldsymbol{A}_4 = \begin{bmatrix} \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times & \times \end{bmatrix}.$$

Die unitäre Transformation auf obere Hessenberg-Gestalt lautet insgesamt

$$\boldsymbol{A}_4 = \boldsymbol{U}^{\mathsf{H}} \boldsymbol{A} \boldsymbol{U}, \qquad \boldsymbol{U} = \boldsymbol{P}_1 \boldsymbol{P}_2 \boldsymbol{P}_3 \boldsymbol{P}_4.$$

Inhalt

Der Prototyp aller numerischer Eigenwertverfahren ist die sog. Vektoriteration nach von Mises, auch Potenzmethode (engl. power method) genannt. Es beruht auf der Tatsache, dass für (nahezu) jeden Vektor $\boldsymbol{q} \in \mathbb{C}^n$ die Vektorfolge

$$\boldsymbol{q}, \boldsymbol{A}\boldsymbol{q}, \boldsymbol{A}^2\boldsymbol{q}, \boldsymbol{A}^3\boldsymbol{q}, \ldots$$

asymptotisch in Richtung eines Eigenvektors von $\boldsymbol{A}$ zum betragsgrößten Eigenwert zeigt. In seiner einfachsten Form lautet das Verfahren wie folgt:

Algorithmus 15: Vektoriteration nach von Mises.

Gegeben: $\boldsymbol{A} \in \mathbb{C}^{n\times n}$, $\boldsymbol{q}_0 \in \mathbb{C}^n$.

1 **for** $k = 1$ **to** $\ldots$ **do**
2 $\quad \boldsymbol{z}_k \leftarrow \boldsymbol{A}\boldsymbol{q}_{k-1}$
3 $\quad \boldsymbol{q}_k \leftarrow \boldsymbol{z}_k / \|\boldsymbol{z}_k\|_2$
4 $\quad \mu_k \leftarrow \boldsymbol{q}_k^{\mathsf{H}} \boldsymbol{A}\boldsymbol{q}_k$

Wie man sieht, wird, allein schon um Gleitpunktüber-/unterlauf zu vermeiden, der Vektor zusätzlich in jedem Schritt normiert.

Lemma 6.3

Sei $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ und $\boldsymbol{u} \in \mathbb{C}^n \setminus \{\boldsymbol{0}\}$. Der Ausdruck $\|\boldsymbol{A}\boldsymbol{u} - \mu\boldsymbol{u}\|_2$ wird minimiert für

$$\mu = \frac{\boldsymbol{u}^{\mathsf{H}}\boldsymbol{A}\boldsymbol{u}}{\boldsymbol{u}^{\mathsf{H}}\boldsymbol{u}} \qquad \text{(Rayleigh-Quotient)}.$$

Ferner ist $\boldsymbol{A}\boldsymbol{u} - \mu\boldsymbol{u} \perp \boldsymbol{u}$.

Ist $\boldsymbol{A}$ diagonalisierbar mit Eigenwerten

$$|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_{n-1}| \geq |\lambda_n|$$

und einer Basis $\{\boldsymbol{v}_j\}_{j=1}^n$ aus zugehörigen Eigenvektoren, so gilt mit Startvektor

$$\boldsymbol{q} = \gamma_1 \boldsymbol{v}_1 + \cdots + \gamma_n \boldsymbol{v}_n,$$

für die Richtung der Iterierten im k-ten Schritt

$$\begin{aligned}
\boldsymbol{A}^k \boldsymbol{q} &= \lambda_1^k \gamma_1 \boldsymbol{v}_1 + \cdots + \lambda_n^k \gamma_n \boldsymbol{v}_n \\
&= \lambda_1^k \left(\gamma_1 \boldsymbol{v}_1 + \left(\frac{\lambda_2}{\lambda_1}\right)^k \gamma_2 \boldsymbol{v}_2 + \cdots + \left(\frac{\lambda_n}{\lambda_1}\right)^k \gamma_n \boldsymbol{v}_n \right) \\
&\approx \lambda_1^k \gamma_1 \boldsymbol{v}_1.
\end{aligned}$$

Sofern nur $\gamma_1 \neq 0$ klingen die übrigen Komponenten der Iterierten (linear) ab mit mindestens der Rate $|\lambda_2/\lambda_1|$.

Wir betrachten die Matrix (MATLAB-Notation)

$$\boldsymbol{A} = \mathrm{diag}(100 : -1 : 1) + \frac{1}{3}\,\mathrm{triu}(\mathrm{ones}(100), 1)$$

mit Eigenwerten

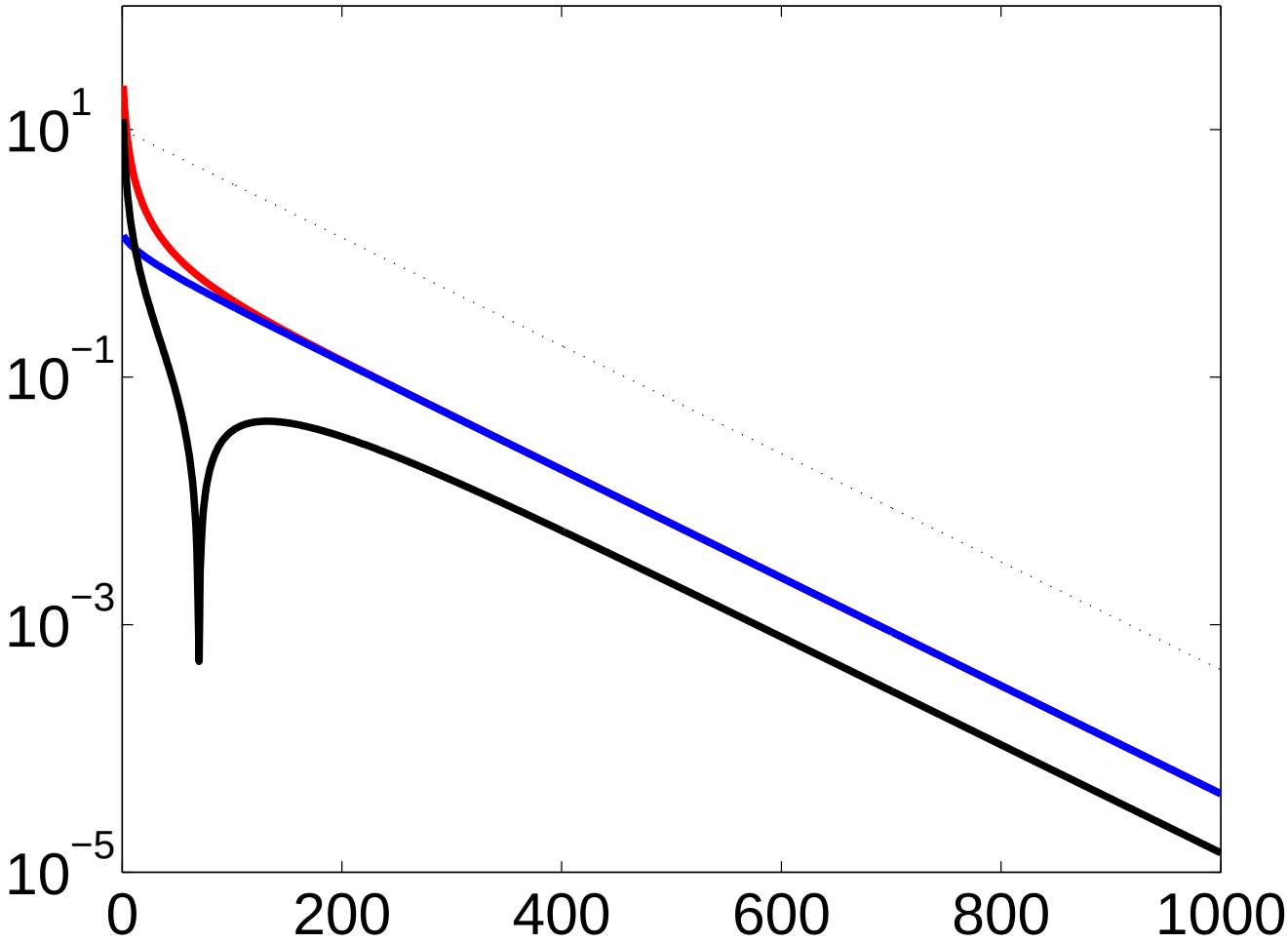
$$\Lambda(\boldsymbol{A}) = \{1, 2, \ldots, 100\}.$$

und wenden die Vektoriteration an mit Startvektoren

- $\boldsymbol{q}_0 = \mathrm{ones}(100, 1)$ bzw.
- $\boldsymbol{q}_0 = \sum_{j=1}^{99} \boldsymbol{v}_j$.

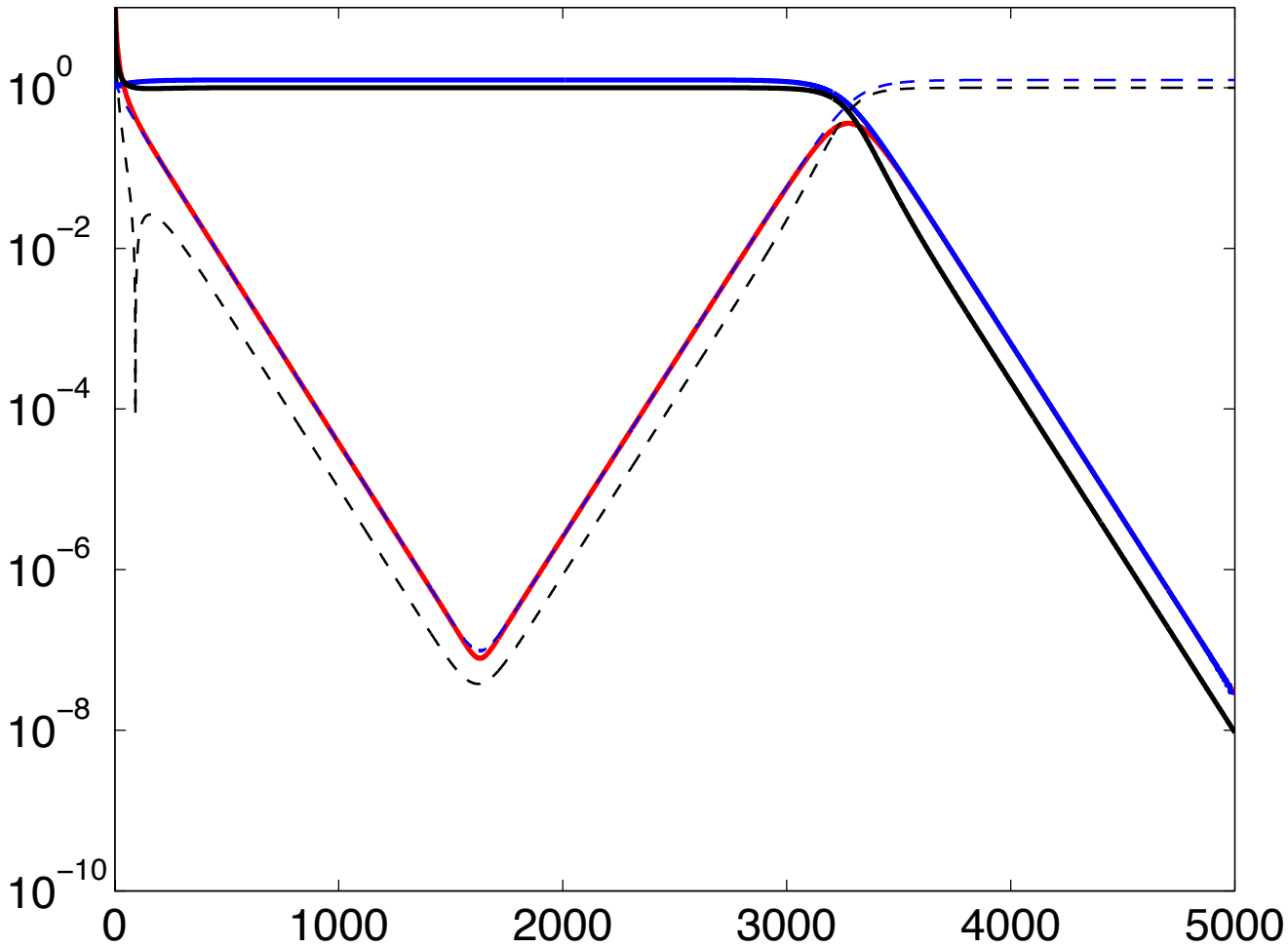
$\boldsymbol{q}_0 = \mathrm{ones}(100,1)$:

$\|\boldsymbol{A}\boldsymbol{q}_k - \mu_k \boldsymbol{q}_k\|_2 \quad \measuredangle(\boldsymbol{q}_k, \boldsymbol{v}_{100}) \quad |100 - \mu_k|$



$\boldsymbol{q}_0 = \boldsymbol{v}_1 + \cdots + \boldsymbol{v}_{99}$: $\|\boldsymbol{A}\boldsymbol{q}_k - \mu_k \boldsymbol{q}_k\|_2$ $\quad \measuredangle(\boldsymbol{q}_k, \boldsymbol{v}_{100})$ $\quad |100 - \mu_k|$

$\measuredangle(\boldsymbol{q}_k, \boldsymbol{v}_{99})$ $\quad |99 - \mu_k|$



Das QR-Verfahren für Eigenwertaufgaben

Inverse Iteration

- Ist man an einem anderen als dem dominanten Eigenpaar interessiert, leistet die einfache Vektoriteration nicht das Gewünschte.
- Ist man jedoch bereit, in jedem Schritt ein lineares Gleichungssystem zu lösen, so kann man die Vektoriteration anstatt auf $\boldsymbol{A}$ auf die Matrix

$$(\boldsymbol{A} - \sigma \boldsymbol{I})^{-1}$$

anwenden. Den Parameter σ bezeichnet man dabei als Shift („Verschiebung“).

- Der betragsgrößte Eigenwert der geshifteten Matrix ist gegeben durch

$$\frac{1}{\lambda(\sigma) - \sigma} \qquad \lambda(\sigma) := \operatorname*{arg\,min}_{\lambda \in \Lambda(\boldsymbol{A})} |\lambda - \sigma|.$$

- Durch geeignete Wahl von σ kann man so die Konvergenz der Vektoriteration steuern.

Algorithmus 16: Inverse Iteration.

Gegeben: $\boldsymbol{A} \in \mathbb{C}^{n\times n}$, $\boldsymbol{q}_0 \in \mathbb{C}^n$, $\sigma \in \mathbb{C} \setminus \Lambda(\boldsymbol{A})$.

1 **for** $k = 1$ **to** $\ldots$ **do**

2 $\quad \boldsymbol{z}_k \leftarrow (\boldsymbol{A} - \sigma \boldsymbol{I})^{-1} \boldsymbol{q}_{k-1}$

3 $\quad \boldsymbol{q}_k \leftarrow \boldsymbol{z}_k / \|\boldsymbol{z}_k\|_2$

4 $\quad \mu_k \leftarrow \boldsymbol{q}_k^{\mathsf{H}} \boldsymbol{A} \boldsymbol{q}_k$

Inhalt

Die Erweiterung der Vektoriteration auf mehrere (orthogonal gehaltene) Vektoren ist als simultane Iteration (simultaneous/subspace/orthogonal iteration) bekannt.

Algorithmus 17: Simultane Iteration.

Gegeben: $\boldsymbol{A} \in \mathbb{C}^{n\times n}$, $\boldsymbol{Q}_0 \in \mathbb{C}^{n\times p}$ mit orthonormalen Spalten.

1 **for** $k = 1$ **to** ... **do**
2 $\quad \boldsymbol{Z}_k \leftarrow \boldsymbol{A}\boldsymbol{Q}_{k-1}$
3 $\quad$ QR-Zerlegung $\boldsymbol{Z}_k = \boldsymbol{Q}_k\boldsymbol{R}_k$

Satz 6.4

Sei $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ diagonalisierbar mit $\boldsymbol{A} = \boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^{-1}$, die Beträge der Eigenwerte von $\boldsymbol{A}$ paarweise verschieden und alle Hauptuntermatrizen von $\boldsymbol{X}$ invertierbar. Dann konvergiert bei simultaner Iteration mit $p = n$ und Startvektor $\boldsymbol{Q}_0 = \boldsymbol{I}$ die Folge der Matrizen $\boldsymbol{A}_k := \boldsymbol{Q}_k^{\mathsf{H}}\boldsymbol{A}\boldsymbol{Q}_k$ gegen eine obere Dreiecksmatrix, auf deren Hauptdiagonalen die Eigenwerte von $\boldsymbol{A}$ in Reihenfolge abnehmenden Betrags erscheinen.

Algorithmus 18: QR-Iteration.

Gegeben: $\boldsymbol{A} \in \mathbb{C}^{n\times n}$, $\boldsymbol{A}_0 := \boldsymbol{A}$.

1 **for** $k = 0$ **to** $\ldots$ **do**

2 $\quad \boldsymbol{A}_k = \boldsymbol{Q}_k \boldsymbol{R}_k$ (QR-Zerlegung)

3 $\quad \boldsymbol{A}_{k+1} \leftarrow \boldsymbol{R}_k \boldsymbol{Q}_k$.

Wegen $\boldsymbol{A}_{k+1} = \boldsymbol{R}_k \boldsymbol{Q}_k = \boldsymbol{Q}_k^{\mathsf{H}} \boldsymbol{Q}_k \boldsymbol{R}_k \boldsymbol{Q}_k = \boldsymbol{Q}_k^{\mathsf{H}} \boldsymbol{A}_k \boldsymbol{Q}_k$ sind $\boldsymbol{A}_{k+1}$ und $\boldsymbol{A}_k$ unitär ähnlich.

Lemma 6.5

Für die durch Algorithmus 18 erzeugte Folge $\{\boldsymbol{A}_k\}$ gilt $\boldsymbol{A}_k = \tilde{\boldsymbol{Q}}_k^{\mathsf{H}} \boldsymbol{A} \tilde{\boldsymbol{Q}}_k$, wobei $\tilde{\boldsymbol{Q}}_k$ die durch simultane Iteration mit Startmatrix $\tilde{\boldsymbol{Q}}_0 = \boldsymbol{I}$ bezeichnet. Somit konvergiert $\boldsymbol{A}_k$ gegen obere Dreicksform, sofern die Eigenwerte von $\boldsymbol{A}$ paarweise verschiedene Beträge besitzen.

Als nächstes beschleunigen wir die Konvergenz der QR-Iteration durch Einbringen von Shifts:

Algorithmus 19: QR-Iteration mit Shifts.

Gegeben: $\boldsymbol{A} \in \mathbb{C}^{n\times n}$, $\boldsymbol{A}_0 := \boldsymbol{A}$.

1 **for** $k = 0$ **to** $\ldots$ **do**
2 Wähle Shift σ_k nahe an einem Eigenwert von $\boldsymbol{A}$
3 $\boldsymbol{A}_k - \sigma_k \boldsymbol{I} = \boldsymbol{Q}_k \boldsymbol{R}_k$ (QR-Zerlegung)
4 $\boldsymbol{A}_{k+1} \leftarrow \boldsymbol{R}_k \boldsymbol{Q}_k + \sigma_k \boldsymbol{I}$.

Lemma 6.6

$\boldsymbol{A}_k$ und $\boldsymbol{A}_{k+1}$ sind unitär ähnlich.

Bemerkung 6.7

1. Ist $\boldsymbol{R}_k$ regulär, so gilt auch $\boldsymbol{A}_{k+1} = \boldsymbol{R}_k \boldsymbol{A}_k \boldsymbol{R}_k^{-1}$.
2. Ist σ_k ein Eigenwert von $\boldsymbol{A}_k$, so konvergiert die QR-Iteration in einem Schritt.
3. Ist σ_k kein (exakter) Eigenwert, so sehen wir $[\boldsymbol{A}_{k+1}]_{n,n}$ dann als konvergiert an, wenn die restlichen Einträge der Zeile $\boldsymbol{A}_{k+1}(n, 1 : n-1)$ hinreichend klein geworden sind.
4. Wahl geeigneter Shifts: Konvergenz gegen Schur-Form beginnt bei $\boldsymbol{A}_k(n,n)$; genauer: $\sigma_k = \boldsymbol{A}_k(n,n)$ führt zu quadratischer Konvergenz von $\boldsymbol{A}_k(n,n)$ gegen einen Eigenwert von $\boldsymbol{A}$.
   Bei reeller Matrix $\boldsymbol{A}$: Konvergenz gegen komplexen Eigenwert möglich.

Besitzt $\boldsymbol{A}_k$ Hessenberg-Gestalt, so kann man den QR-Schritt

$$\boldsymbol{A}_k - \sigma_k \boldsymbol{I} = \boldsymbol{Q}_k \boldsymbol{R}_k, \qquad \boldsymbol{A}_{k+1} \leftarrow \boldsymbol{R}_k \boldsymbol{Q}_k + \sigma_k \boldsymbol{I}$$

so durchführen, dass die Hessenberg-Struktur erhalten bleibt, indem man die QR-Zerlegung mit Givens-Rotationen durchführt.

Im folgenden 5×5-Beispiel wird die obere Dreiecksstruktur nach sukzessiver Multiplikation mit Givens-Rotationen

$$\boldsymbol{G}_{j,j+1}, \qquad j = 1, \ldots, n-1,$$

erreicht:

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix} \xrightarrow{\boldsymbol{G}_{1,2}} \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix} \xrightarrow{\boldsymbol{G}_{2,3}} \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix}$$

$$\xrightarrow{\boldsymbol{G}_{3,4}} \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix} \xrightarrow{\boldsymbol{G}_{4,5}} \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & \times \end{bmatrix}$$

Fasst man alle $n-1$ Givens-Rotationen zusammen als

$$\boldsymbol{Q}^{\mathsf{H}} := \boldsymbol{G}_{n-1,n}\,\boldsymbol{G}_{n-2,n-1}\cdots\boldsymbol{G}_{1,2},$$

so ergibt sich

$$\boldsymbol{Q}^{\mathsf{H}}\boldsymbol{A} = \boldsymbol{R}, \quad \text{oder} \quad \boldsymbol{A} = \boldsymbol{Q}\boldsymbol{R}.$$

Die Matrix $\boldsymbol{Q} = \boldsymbol{G}_{1,2}^{\mathsf{H}} \cdots \boldsymbol{G}_{n-1,n}^{\mathsf{H}}$ ist aufgrund des Besetzungsmusters ihrer Faktoren ebenfalls obere Hessenberg-Matrix. (Dies folgt auch aus dem zu $\boldsymbol{A} = \boldsymbol{QR}$ äquivalenten Ausdruck $\boldsymbol{Q} = \boldsymbol{AR}^{-1}$.)

Demzufolge ist mit $\boldsymbol{A}_k$ auch

$$\boldsymbol{A}_{k+1} = \boldsymbol{R}_k \boldsymbol{Q}_k + \sigma_k \boldsymbol{I}$$

obere Hessenberg-Matrix.

Satz 6.8

Beim QR-Schritt mit einfachem Shift bleibt die obere Hessenberg-Struktur erhalten.

Eine effiziente Implementierung des QR-Verfahrens arbeitet mit impliziten Shifts, d.h. die QR-Zerlegung der Hessenberg-Matrizen wird implizit als Produkt von elementaren unitären Matrizen konstruiert. Hierzu ist eine Tatsache hilfreich, die als implizites Q-Theorem bekannt ist.

Definition 6.9

Eine obere Hessenberg-Matrix $\boldsymbol{H} = [h_{i,j}]_{i,j=1}^{n} \in \mathbb{C}^{n\times n}$ heißt unreduziert (engl. unreduced), falls

$$h_{k+1,k} \neq 0 \qquad \forall k = 1, 2, \ldots, n-1.$$

Satz 6.10

Ein Eigenwert $\lambda \in \Lambda(\boldsymbol{A})$ einer unreduzierten Hessenberg-Matrix $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ besitzt geometrische Vielfachheit Eins.

Satz 6.11 (Implizites Q-Theorem)

Seien $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ sowie zwei unitäre Matrizen $\boldsymbol{Q} = [\boldsymbol{q}_1, \ldots, \boldsymbol{q}_n]$ und $\boldsymbol{V} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n]$ gegeben, für welche die Matrizen

$$\boldsymbol{H} = [h_{i,j}] = \boldsymbol{Q}^{\mathsf{H}}\boldsymbol{A}\boldsymbol{Q} \quad \text{und} \quad \boldsymbol{G} = [g_{i,j}] = \boldsymbol{V}^{\mathsf{H}}\boldsymbol{A}\boldsymbol{V}$$

obere Hessenberg-Gestalt besitzen. Ferner sei $k := \min\{j : h_{j+1,j} = 0\}$ bzw. $k = n$ falls $\boldsymbol{H}$ unreduziert. Gilt dann $\boldsymbol{v}_1 = \boldsymbol{q}_1$, so gelten auch

$$\boldsymbol{v}_j = \vartheta_j \boldsymbol{q}_j \quad \text{mit} \quad |\vartheta_j| = 1, \quad |h_{j+1,j}| = |g_{j+1,j}|, \qquad j = 2, \ldots, k.$$

Falls $k < n$, so ist auch $g_{k+1,k} = 0$.

Interpretation: Sind

$$\boldsymbol{G} = \boldsymbol{V}^{\mathsf{H}} \boldsymbol{A} \boldsymbol{V} \quad \text{und} \quad \boldsymbol{H} = \boldsymbol{Q}^{\mathsf{H}} \boldsymbol{A} \boldsymbol{Q}$$

zwei unreduzierte Hessenberg-Matrizen und besitzen die unitären Matrizen $\boldsymbol{V}$ und $\boldsymbol{Q}$ die gleiche erste Spalte, so sind $\boldsymbol{G}$ und $\boldsymbol{H}$ im Wesentlichen gleich, d.h. es gilt $\boldsymbol{G} = \boldsymbol{D}^{-1} \boldsymbol{H} \boldsymbol{D}$ mit einer Diagonalmatrix $\boldsymbol{D} = \operatorname{diag}(\vartheta_1, \ldots, \vartheta_n)$, $|\vartheta_j| = 1$.

Die implizite Berechnung der QR-Schritts $\boldsymbol{A}_{k+1} = \boldsymbol{Q}_k^{\mathsf{H}} \boldsymbol{A}_k \boldsymbol{Q}_k$ geschieht nun wie folgt:

(1) Berechne die erste Spalte von $\boldsymbol{Q}_k$ (skalares Vielfaches der ersten Spalte von $\boldsymbol{A}_k - \sigma_k \boldsymbol{I}$, ergibt sich also durch Nomierung).

(2) Bestimme restliche Spalten von $\boldsymbol{Q}_k$ so, dass $\boldsymbol{Q}_k$ unitär und $\boldsymbol{A}_{k+1}$ unreduzierte obere Hessenberg-Matrix.

Das implizite Q-Theorem stellt dann sicher, dass wir dann $\boldsymbol{A}_{k+1}$ bis auf Faktoren vom Betrag eins berechnet haben.

Wir illustrieren den impliziten QR-Schritt an folgendem 5×5 Beispiel:
$\boldsymbol{A}_0$ liege in unreduzierter Hessenberg-Gestalt vor.

$$\boldsymbol{G}_1^{\mathsf{H}} = \begin{bmatrix} c_1 & s_1 & & & \\ -\overline{s_1} & c_1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix}, \quad \boldsymbol{G}_1^{\mathsf{H}} \boldsymbol{A}\, \boldsymbol{G}_1 = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ + & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix}$$

$$\boldsymbol{G}_2^{\mathsf{H}} = \begin{bmatrix} 1 & & & & \\ & c_2 & s_2 & & \\ & -\overline{s_2} & c_2 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix}, \quad \boldsymbol{G}_2^{\mathsf{H}} \boldsymbol{A}_1\, \boldsymbol{G}_2 = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & + & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix}$$

$$\boldsymbol{G}_3^{\mathsf{H}} = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & c_3 & s_3 & \\ & & -\overline{s_3} & c_3 & \\ & & & & 1 \end{bmatrix}, \qquad \boldsymbol{G}_3^{\mathsf{H}} \boldsymbol{A}_2 \boldsymbol{G}_3 = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & + & \times & \times \end{bmatrix}$$

$$\boldsymbol{G}_4^{\mathsf{H}} = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & c_4 & s_4 \\ & & & -\overline{s_4} & c_4 \end{bmatrix}, \qquad \boldsymbol{G}_4^{\mathsf{H}} \boldsymbol{A}_3 \boldsymbol{G}_4 = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix}$$

Diese sukzessive Wahl der Givens-Rotationen nennt man auch bulge chasing, da die von der ersten orthogonalen Ähnlichkeitstransformation erzeugte 'Beule' in der (3,1)-Position schrittweise entlang der zweiten unteren Nebendiagonalen 'gescheucht' wird.

Insgesamt besitzt nun $\boldsymbol{Q}^{\mathsf{H}}\boldsymbol{A}\boldsymbol{Q}$ wieder obere Hessenberg-Gestalt, mit

$$\boldsymbol{Q} := \boldsymbol{G}_1\boldsymbol{G}_2\boldsymbol{G}_3\boldsymbol{G}_4 = \begin{bmatrix} c_1 & \times & \times & \times & \times \\ s_1 & \times & \times & \times & \times \\ & s_2 & \times & \times & \times \\ & & s_3 & \times & \times \\ & & & s_4 & \times \end{bmatrix}.$$

- Die erste Spalte von $\boldsymbol{Q}$ lautet $[c_1, s_2, 0, \ldots, 0]^{\mathsf{T}}$.
- Nach dem impliziten Q-Theorem bestimmt diese (bis auf Faktoren vom Betrag Eins) die restlichen Spalten von $\boldsymbol{Q}$.
- Wir wählen nun die erste Spalte von $\boldsymbol{Q}$ als skalares Vielfaches der ersten Spalte von $\boldsymbol{A} - \sigma\boldsymbol{I}$, also $[a_{11} - \sigma, a_{21}, 0, \ldots, 0]^{\mathsf{T}}$.
- Damit ist $\boldsymbol{Q}$ bereits der linke Faktor aus der QR-Zerlegung von $\boldsymbol{A} - \sigma\boldsymbol{I}$.

- Komplexe Arithmetik ist etwa viermal teurer als reelle.
- Daher wäre es von Vorteil, für reelle Matrizen die QR-Iteration auch in reeller Arithmetik durchzuführen.
- Man nutzt dabei die Tatsache aus, dass nichtreelle Eigenwerte reeller Matrizen als konjugiert-komplexe Paare auftreten.
- Der Doppelshift-Algorithmus von Francis nutzt gleichzeitig die konjugiert-komplexen Shifts σ und $\overline{\sigma}$:

$$
\begin{aligned}
\boldsymbol{A}_0 - \sigma \boldsymbol{I} &= \boldsymbol{Q}_1 \boldsymbol{R}_1 \\
\boldsymbol{A}_1 &= \boldsymbol{R}_1 \boldsymbol{Q}_1 + \sigma \boldsymbol{I}, \quad \text{d.h. } \boldsymbol{A}_1 = \boldsymbol{Q}_1^{\mathsf{T}} \boldsymbol{A}_0 \boldsymbol{Q}_1, \\
\boldsymbol{A}_1 - \overline{\sigma} \boldsymbol{I} &= \boldsymbol{Q}_2 \boldsymbol{R}_2, \\
\boldsymbol{A}_2 &= \boldsymbol{R}_2 \boldsymbol{Q}_2 + \overline{\sigma} \boldsymbol{I}, \quad \text{d.h. } \boldsymbol{A}_2 = \boldsymbol{Q}_2^{\mathsf{T}} \boldsymbol{A}_1 \boldsymbol{Q}_2 = \boldsymbol{Q}_2^{\mathsf{T}} \boldsymbol{Q}_1^{\mathsf{T}} \boldsymbol{A}_0 \boldsymbol{Q}_1 \boldsymbol{Q}_2.
\end{aligned}
$$

Lemma 6.12

Die orthogonalen Matrizen $\boldsymbol{Q}_1$, $\boldsymbol{Q}_2$ können so gewählt werden, dass sie (und damit $\boldsymbol{A}_2$) reell sind und die erste Spalte von $\boldsymbol{Q}_1 \boldsymbol{Q}_2$ leicht zu berechnen ist.

Zur Wahl von (einfachen und doppelten) Shifts:

- Beim einfachen Shift hatten für $\boldsymbol{A}_k$ hatten wir $\sigma = \boldsymbol{A}_k(n,n)$ gewählt.
- Für einen Doppelshift wählt man σ und $\overline{\sigma}$ als die beiden Eigenwerte des rechten unteren 2-Blocks $\boldsymbol{A}_k(n-1:n, n-1:n)$.
  Dieser Block konvergiert dann entweder gegen eine 2×2 Diagonalmatrix oder gegen eine 2×2-Matrix deren (konjugiert-komplexe) Eigenwerte auch Eigenwerte von $\boldsymbol{A}$ sind.
- Eigenwerte werden als konvergiert betrachtet sobald $a_{n,n-1}$ bzw. $a_{n-1,n-2}$ hinreichend klein geworden sind.
- Man kann zeigen: im Allgemeinen konvergieren diese Einträge quadratisch gegen Null. In der Praxis sind nicht mehr als zwei Iterationen pro Eigenwert erforderlich.
- Es sei erwähnt, dass trotz erfolgreichen Einsatzes dieses Verfahrens seit 60 Jahren noch einige Detailfragen bei der Konvergenzanalyse des QR-Verfahrens offen bleiben.

Konvergenz der QR-Iteration:

- Unter realistischen Annahmen kann man zeigen, dass das QR-Verfahren quadratisch konvergiert.
- Für hermitesche Matrizen erhöht sich diese Konvergenzordnung auf kubisch.
- Das hermitesche QR-Verfahren kann auch zur Berechnung der Singulärwertzerlegung eingesetzt werden.