Numerische Verfahren

Jens-Peter M. Zemke
zemke@tu-harburg.de

Institut für Numerische Simulation
Technische Universität Hamburg-Harburg

17.06.2008

TUHH
Technische Universität Hamburg-Harburg

Übersicht

Vorbetrachtungen

Da Eigenwerte und Eigenvektoren reeller Matrizen komplex sein können, betrachten wir gleich für $\boldsymbol{A} \in \mathbb{C}^{(n,n)}$ das spezielle Eigenwertproblem

$$\boldsymbol{A}\boldsymbol{x} = \lambda \boldsymbol{x}. \tag{6.1}$$

Vorbetrachtungen

Da Eigenwerte und Eigenvektoren reeller Matrizen komplex sein können, betrachten wir gleich für $\boldsymbol{A} \in \mathbb{C}^{(n,n)}$ das spezielle Eigenwertproblem

$$\boldsymbol{A}\boldsymbol{x} = \lambda \boldsymbol{x}. \tag{6.1}$$

Besitzt (6.1) eine nichttriviale Lösung $\boldsymbol{x} \in \mathbb{C}^n \setminus \{\boldsymbol{o}\}$, so heißt λ ein Eigenwert von $\boldsymbol{A}$ und $\boldsymbol{x}$ ein zugehöriger Eigenvektor.

Vorbetrachtungen

Da Eigenwerte und Eigenvektoren reeller Matrizen komplex sein können, betrachten wir gleich für $\boldsymbol{A} \in \mathbb{C}^{(n,n)}$ das spezielle Eigenwertproblem

$$\boldsymbol{A}\boldsymbol{x} = \lambda \boldsymbol{x}. \tag{6.1}$$

Besitzt (6.1) eine nichttriviale Lösung $\boldsymbol{x} \in \mathbb{C}^n \setminus \{\boldsymbol{o}\}$, so heißt λ ein Eigenwert von $\boldsymbol{A}$ und $\boldsymbol{x}$ ein zugehöriger Eigenvektor.

Die Menge aller Eigenwerte einer Matrix $\boldsymbol{A}$ heißt das Spektrum von $\boldsymbol{A}$ und wird mit $\Lambda(\boldsymbol{A})$ bezeichnet.

Vorbetrachtungen

Da Eigenwerte und Eigenvektoren reeller Matrizen komplex sein können, betrachten wir gleich für $\boldsymbol{A} \in \mathbb{C}^{(n,n)}$ das spezielle Eigenwertproblem

$$\boldsymbol{A}\boldsymbol{x} = \lambda \boldsymbol{x}. \tag{6.1}$$

Besitzt (6.1) eine nichttriviale Lösung $\boldsymbol{x} \in \mathbb{C}^n \setminus \{\boldsymbol{o}\}$, so heißt λ ein Eigenwert von $\boldsymbol{A}$ und $\boldsymbol{x}$ ein zugehöriger Eigenvektor.

Die Menge aller Eigenwerte einer Matrix $\boldsymbol{A}$ heißt das Spektrum von $\boldsymbol{A}$ und wird mit $\Lambda(\boldsymbol{A})$ bezeichnet.

Genauer heißt $\boldsymbol{x} \neq \boldsymbol{o}$ mit (6.1) ein Rechtseigenvektor von $\boldsymbol{A}$ zu λ.

Vorbetrachtungen

Ist λ ein Eigenwert von $\boldsymbol{A}$, so besitzt auch die Gleichung

$$\boldsymbol{y}^H(\boldsymbol{A} - \lambda\boldsymbol{E}) = \boldsymbol{o} \tag{6.2}$$

nichttriviale Lösungen. Jedes $\boldsymbol{y} \in \mathbb{C}^n$, das die Gleichung (6.2) erfüllt, heißt ein Linkseigenvektor von $\boldsymbol{A}$.

Vorbetrachtungen

Ist λ ein Eigenwert von $\boldsymbol{A}$, so besitzt auch die Gleichung

$$\boldsymbol{y}^H(\boldsymbol{A} - \lambda\boldsymbol{E}) = \boldsymbol{o} \tag{6.2}$$

nichttriviale Lösungen. Jedes $\boldsymbol{y} \in \mathbb{C}^n$, das die Gleichung (6.2) erfüllt, heißt ein Linkseigenvektor von $\boldsymbol{A}$.

Die Notation ist hier nicht einheitlich. In einigen Büchern werden die nichttrivialen Lösungen von $\boldsymbol{y}^T(\boldsymbol{A} - \lambda\boldsymbol{E}) = \boldsymbol{o}$ die Linkseigenvektoren von $\boldsymbol{A}$ genannt. Wenn wir nur von Eigenvektoren sprechen, so sind stets Rechtseigenvektoren gemeint.

Vorbetrachtungen

Ist λ ein Eigenwert von $\boldsymbol{A}$, so besitzt auch die Gleichung

$$\boldsymbol{y}^H(\boldsymbol{A} - \lambda\boldsymbol{E}) = \boldsymbol{o} \tag{6.2}$$

nichttriviale Lösungen. Jedes $\boldsymbol{y} \in \mathbb{C}^n$, das die Gleichung (6.2) erfüllt, heißt ein Linkseigenvektor von $\boldsymbol{A}$.

Die Notation ist hier nicht einheitlich. In einigen Büchern werden die nichttrivialen Lösungen von $\boldsymbol{y}^T(\boldsymbol{A} - \lambda\boldsymbol{E}) = \boldsymbol{o}$ die Linkseigenvektoren von $\boldsymbol{A}$ genannt. Wenn wir nur von Eigenvektoren sprechen, so sind stets Rechtseigenvektoren gemeint.

Die Eigenwerte λ von $\boldsymbol{A}$ sind die Nullstellen des charakteristischen Polynoms

$$\chi(z) := \det(z\boldsymbol{E} - \boldsymbol{A}).$$

Vorbetrachtungen

Ist λ ein k-fache Nullstelle von χ (ist also das Polynom $\chi(z)$ durch $(z-\lambda)^k$, aber nicht durch $(z-\lambda)^{k+1}$ teilbar), so heißt k die algebraische Vielfachheit $\alpha(\lambda)$ von λ.

Vorbetrachtungen

Ist λ ein k-fache Nullstelle von χ (ist also das Polynom $\chi(z)$ durch $(z-\lambda)^k$, aber nicht durch $(z-\lambda)^{k+1}$ teilbar), so heißt k die algebraische Vielfachheit $\alpha(\lambda)$ von λ.

Da χ den Grad n besitzt, besitzt also $\boldsymbol{A}$ genau n Eigenwerte, wenn man jeden Eigenwert entsprechend seiner algebraischen Vielfachheit zählt.

Vorbetrachtungen

Ist λ ein k-fache Nullstelle von χ (ist also das Polynom $\chi(z)$ durch $(z-\lambda)^k$, aber nicht durch $(z-\lambda)^{k+1}$ teilbar), so heißt k die algebraische Vielfachheit $\alpha(\lambda)$ von λ.

Da χ den Grad n besitzt, besitzt also $\boldsymbol{A}$ genau n Eigenwerte, wenn man jeden Eigenwert entsprechend seiner algebraischen Vielfachheit zählt.

Neben der algebraischen Vielfachheit betrachtet man die geometrische Vielfachheit $\gamma(\lambda)$ eines Eigenwerts λ.

Vorbetrachtungen

Ist λ ein k-fache Nullstelle von χ (ist also das Polynom $\chi(z)$ durch $(z-\lambda)^k$, aber nicht durch $(z-\lambda)^{k+1}$ teilbar), so heißt k die algebraische Vielfachheit $\alpha(\lambda)$ von λ.

Da χ den Grad n besitzt, besitzt also $\boldsymbol{A}$ genau n Eigenwerte, wenn man jeden Eigenwert entsprechend seiner algebraischen Vielfachheit zählt.

Neben der algebraischen Vielfachheit betrachtet man die geometrische Vielfachheit $\gamma(\lambda)$ eines Eigenwerts λ. Dies ist die Dimension des Lösungsraums des linearen Gleichungssystems

$$(\boldsymbol{A} - \lambda \boldsymbol{E})\boldsymbol{x} = \boldsymbol{o}.$$

Vorbetrachtungen

Ist λ ein k-fache Nullstelle von χ (ist also das Polynom $\chi(z)$ durch $(z-\lambda)^k$, aber nicht durch $(z-\lambda)^{k+1}$ teilbar), so heißt k die algebraische Vielfachheit $\alpha(\lambda)$ von λ.

Da χ den Grad n besitzt, besitzt also $\boldsymbol{A}$ genau n Eigenwerte, wenn man jeden Eigenwert entsprechend seiner algebraischen Vielfachheit zählt.

Neben der algebraischen Vielfachheit betrachtet man die geometrische Vielfachheit $\gamma(\lambda)$ eines Eigenwerts λ. Dies ist die Dimension des Lösungsraums des linearen Gleichungssystems

$$(\boldsymbol{A} - \lambda \boldsymbol{E})\boldsymbol{x} = \boldsymbol{o}.$$

Nach LA Satz 8.18 ist für jeden Eigenwert die geometrische Vielfachheit kleiner oder gleich der algebraischen Vielfachheit, es gilt

$$1 \leqslant \gamma(\lambda) \leqslant \alpha(\lambda) \leqslant n.$$

Vorbetrachtungen

Gilt $\boldsymbol{B} = \boldsymbol{X}^{-1}\boldsymbol{A}\boldsymbol{X}$ mit einer regulären Matrix $\boldsymbol{X} \in \mathbb{C}^{(n,n)}$, so heißen die Matrizen $\boldsymbol{A}$ und $\boldsymbol{B}$ ähnlich, den Übergang von $\boldsymbol{A}$ zu $\boldsymbol{B}$ und auch umgekehrt nennt man eine Ähnlichkeitstransformation.

Vorbetrachtungen

Gilt $\boldsymbol{B} = \boldsymbol{X}^{-1}\boldsymbol{A}\boldsymbol{X}$ mit einer regulären Matrix $\boldsymbol{X} \in \mathbb{C}^{(n,n)}$, so heißen die Matrizen $\boldsymbol{A}$ und $\boldsymbol{B}$ ähnlich, den Übergang von $\boldsymbol{A}$ zu $\boldsymbol{B}$ und auch umgekehrt nennt man eine Ähnlichkeitstransformation.

Ähnliche Matrizen besitzen (vgl. LA Satz 8.15) dieselben Eigenwerte einschließlich ihrer algebraischen und geometrischen Vielfachheit.

Vorbetrachtungen

Gilt $\boldsymbol{B} = \boldsymbol{X}^{-1}\boldsymbol{A}\boldsymbol{X}$ mit einer regulären Matrix $\boldsymbol{X} \in \mathbb{C}^{(n,n)}$, so heißen die Matrizen $\boldsymbol{A}$ und $\boldsymbol{B}$ ähnlich, den Übergang von $\boldsymbol{A}$ zu $\boldsymbol{B}$ und auch umgekehrt nennt man eine Ähnlichkeitstransformation.

Ähnliche Matrizen besitzen (vgl. LA Satz 8.15) dieselben Eigenwerte einschließlich ihrer algebraischen und geometrischen Vielfachheit.

Naheliegend ist es nun, die Matrix $\boldsymbol{A}$, deren Eigenwerte bestimmt werden sollen, durch geeignete Ähnlichkeitstransformationen auf eine Gestalt zu bringen, aus der man die Eigenwerte (oder jedenfalls Näherungen hierfür) von $\boldsymbol{A}$ leicht ablesen kann oder für die das Eigenwertproblem leichter gelöst werden kann.

Vorbetrachtungen

Gilt $\boldsymbol{B} = \boldsymbol{X}^{-1}\boldsymbol{A}\boldsymbol{X}$ mit einer regulären Matrix $\boldsymbol{X} \in \mathbb{C}^{(n,n)}$, so heißen die Matrizen $\boldsymbol{A}$ und $\boldsymbol{B}$ ähnlich, den Übergang von $\boldsymbol{A}$ zu $\boldsymbol{B}$ und auch umgekehrt nennt man eine Ähnlichkeitstransformation.

Ähnliche Matrizen besitzen (vgl. LA Satz 8.15) dieselben Eigenwerte einschließlich ihrer algebraischen und geometrischen Vielfachheit.

Naheliegend ist es nun, die Matrix $\boldsymbol{A}$, deren Eigenwerte bestimmt werden sollen, durch geeignete Ähnlichkeitstransformationen auf eine Gestalt zu bringen, aus der man die Eigenwerte (oder jedenfalls Näherungen hierfür) von $\boldsymbol{A}$ leicht ablesen kann oder für die das Eigenwertproblem leichter gelöst werden kann.

Das folgende Lemma ermöglicht es, die Dimension des Eigenwertproblems zu reduzieren.

Vorbetrachtungen

Lemma 6.1

Besitzt $\boldsymbol{A} \in \mathbb{C}^{(n,n)}$ eine Blockzerlegung

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{O} & \boldsymbol{A}_{22} \end{pmatrix}$$

mit $\boldsymbol{A}_{11} \in \mathbb{C}^{(m,m)}$, $\boldsymbol{A}_{22} \in \mathbb{C}^{(n-m,n-m)}$, so gilt

$$\Lambda(\boldsymbol{A}) = \Lambda(\boldsymbol{A}_{11}) \cup \Lambda(\boldsymbol{A}_{22}). \tag{6.3}$$

Vorbetrachtungen

Beweis.

Es gelte

$$\boldsymbol{A}\boldsymbol{x} = \begin{pmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{O} & \boldsymbol{A}_{22} \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{pmatrix} = \lambda \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{pmatrix}$$

mit $\boldsymbol{x}_1 \in \mathbb{C}^m$ und $\boldsymbol{x}_2 \in \mathbb{C}^{n-m}$.

Vorbetrachtungen

Beweis.

Es gelte

$$\boldsymbol{A}\boldsymbol{x} = \begin{pmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{O} & \boldsymbol{A}_{22} \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{pmatrix} = \lambda \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{pmatrix}$$

mit $\boldsymbol{x}_1 \in \mathbb{C}^m$ und $\boldsymbol{x}_2 \in \mathbb{C}^{n-m}$.

Gilt $\boldsymbol{x}_2 \neq \boldsymbol{o}$, so ist $\boldsymbol{A}_{22}\boldsymbol{x}_2 = \lambda \boldsymbol{x}_2$ und $\lambda \in \Lambda(\boldsymbol{A}_{22})$.

Vorbetrachtungen

Beweis.

Es gelte

$$\boldsymbol{A}\boldsymbol{x} = \begin{pmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{O} & \boldsymbol{A}_{22} \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{pmatrix} = \lambda \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{pmatrix}$$

mit $\boldsymbol{x}_1 \in \mathbb{C}^m$ und $\boldsymbol{x}_2 \in \mathbb{C}^{n-m}$.

Gilt $\boldsymbol{x}_2 \neq \boldsymbol{o}$, so ist $\boldsymbol{A}_{22}\boldsymbol{x}_2 = \lambda \boldsymbol{x}_2$ und $\lambda \in \Lambda(\boldsymbol{A}_{22})$.

Ist $\boldsymbol{x}_2 = \boldsymbol{o}$, so gilt $\boldsymbol{A}_{11}\boldsymbol{x}_1 = \lambda \boldsymbol{x}_1$ und $\lambda \in \Lambda(\boldsymbol{A}_{11})$.

Vorbetrachtungen

Beweis.

Es gelte

$$\boldsymbol{A}\boldsymbol{x} = \begin{pmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{O} & \boldsymbol{A}_{22} \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{pmatrix} = \lambda \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{pmatrix}$$

mit $\boldsymbol{x}_1 \in \mathbb{C}^m$ und $\boldsymbol{x}_2 \in \mathbb{C}^{n-m}$.

Gilt $\boldsymbol{x}_2 \neq \boldsymbol{o}$, so ist $\boldsymbol{A}_{22}\boldsymbol{x}_2 = \lambda \boldsymbol{x}_2$ und $\lambda \in \Lambda(\boldsymbol{A}_{22})$.

Ist $\boldsymbol{x}_2 = \boldsymbol{o}$, so gilt $\boldsymbol{A}_{11}\boldsymbol{x}_1 = \lambda \boldsymbol{x}_1$ und $\lambda \in \Lambda(\boldsymbol{A}_{11})$.

Es ist also

$$\Lambda(\boldsymbol{A}) \subset \Lambda(\boldsymbol{A}_{11}) \cup \Lambda(\boldsymbol{A}_{22}).$$

Vorbetrachtungen

Beweis.

Es gelte

$$\boldsymbol{A}\boldsymbol{x} = \begin{pmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{O} & \boldsymbol{A}_{22} \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{pmatrix} = \lambda \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{pmatrix}$$

mit $\boldsymbol{x}_1 \in \mathbb{C}^m$ und $\boldsymbol{x}_2 \in \mathbb{C}^{n-m}$.

Gilt $\boldsymbol{x}_2 \neq \boldsymbol{o}$, so ist $\boldsymbol{A}_{22}\boldsymbol{x}_2 = \lambda \boldsymbol{x}_2$ und $\lambda \in \Lambda(\boldsymbol{A}_{22})$.

Ist $\boldsymbol{x}_2 = \boldsymbol{o}$, so gilt $\boldsymbol{A}_{11}\boldsymbol{x}_1 = \lambda \boldsymbol{x}_1$ und $\lambda \in \Lambda(\boldsymbol{A}_{11})$.

Es ist also

$$\Lambda(\boldsymbol{A}) \subset \Lambda(\boldsymbol{A}_{11}) \cup \Lambda(\boldsymbol{A}_{22}).$$

Zählt man alle Eigenwerte entsprechend ihrer algebraischen Vielfachheit, so besitzt $\boldsymbol{A}$ n Eigenwerte, $\boldsymbol{A}_{11}$ m Eigenwerte und $\boldsymbol{A}_{22}$ $n-m$ Eigenwerte. Damit kann die Inklusion nicht echt sein, und es ist (6.3) bewiesen. □

Vorbetrachtungen

Ist $\boldsymbol{J}$ die Jordansche Normalform der Matrix $\boldsymbol{A}$ und $\boldsymbol{J} = \boldsymbol{X}^{-1}\boldsymbol{A}\boldsymbol{X}$, so kann man (vgl. LA Abschnitt 8.5) alle Eigenwerte und Eigenvektoren (und auch alle Hauptvektoren) von $\boldsymbol{A}$ aus $\boldsymbol{J}$ und der Transformationsmatrix $\boldsymbol{X}$ sofort ablesen.

Vorbetrachtungen

Ist $\boldsymbol{J}$ die Jordansche Normalform der Matrix $\boldsymbol{A}$ und $\boldsymbol{J} = \boldsymbol{X}^{-1}\boldsymbol{A}\boldsymbol{X}$, so kann man (vgl. LA Abschnitt 8.5) alle Eigenwerte und Eigenvektoren (und auch alle Hauptvektoren) von $\boldsymbol{A}$ aus $\boldsymbol{J}$ und der Transformationsmatrix $\boldsymbol{X}$ sofort ablesen.

Trotzdem wird die Jordansche Normalform in der numerischen Mathematik nicht verwendet. Den Grund zeigt das folgende, auf James Hardy Wilkinson zurückgehende Beispiel:

Vorbetrachtungen

Beispiel 6.2: Die Störung

$$\boldsymbol{J}_\alpha := \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \alpha & 0 & 0 & \cdots & 0 \end{pmatrix} \in \mathbb{R}^{(n,n)}$$

der Jordanschen Normalform $\boldsymbol{J}_0$ mit dem n-fachen Eigenwert 0 besitzt das charakteristische Polynom

$$\chi_\alpha(z) = (-1)^{n+1}(z^n - \alpha).$$

Vorbetrachtungen

Beispiel 6.2: Die Störung

$$\boldsymbol{J}_\alpha := \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \alpha & 0 & 0 & \cdots & 0 \end{pmatrix} \in \mathbb{R}^{(n,n)}$$

der Jordanschen Normalform $\boldsymbol{J}_0$ mit dem n-fachen Eigenwert 0 besitzt das charakteristische Polynom

$$\chi_\alpha(z) = (-1)^{n+1}(z^n - \alpha).$$

Die Matrix $\boldsymbol{J}_\alpha$ besitzt also für jedes $\alpha \neq 0$ immer n verschiedene Nullstellen, und ist daher auf Diagonalgestalt transformierbar. Dies zeigt, dass die Struktur der Jordanschen Normalform nicht stabil unter kleinen Störungen ist. □

Vorbetrachtungen

Wir betrachten ferner nicht beliebige Ähnlichkeitstransformationen, denn die nichtsinguläre Transformationsmatrix $\boldsymbol{X}$ kann schlecht konditioniert sein.

Vorbetrachtungen

Wir betrachten ferner nicht beliebige Ähnlichkeitstransformationen, denn die nichtsinguläre Transformationsmatrix $\boldsymbol{X}$ kann schlecht konditioniert sein.

In diesem Fall ist die Transformation $\boldsymbol{X}^{-1}\boldsymbol{A}\boldsymbol{X}$ anfällig für Rundungsfehler.

Vorbetrachtungen

Wir betrachten ferner nicht beliebige Ähnlichkeitstransformationen, denn die nichtsinguläre Transformationsmatrix $\boldsymbol{X}$ kann schlecht konditioniert sein.

In diesem Fall ist die Transformation $\boldsymbol{X}^{-1}\boldsymbol{A}\boldsymbol{X}$ anfällig für Rundungsfehler.

Um dies auszuschließen, beschränken wir uns auf unitäre Transformationsmatrizen $\boldsymbol{U}$, für die also gilt $\boldsymbol{U}^H\boldsymbol{U} = \boldsymbol{E}$.

Vorbetrachtungen

Wir betrachten ferner nicht beliebige Ähnlichkeitstransformationen, denn die nichtsinguläre Transformationsmatrix $\boldsymbol{X}$ kann schlecht konditioniert sein.

In diesem Fall ist die Transformation $\boldsymbol{X}^{-1}\boldsymbol{A}\boldsymbol{X}$ anfällig für Rundungsfehler.

Um dies auszuschließen, beschränken wir uns auf unitäre Transformationsmatrizen $\boldsymbol{U}$, für die also gilt $\boldsymbol{U}^H\boldsymbol{U} = \boldsymbol{E}$.

Für diese ist

$$\|\boldsymbol{U}\boldsymbol{x}\|_2^2 = (\boldsymbol{U}\boldsymbol{x})^H(\boldsymbol{U}\boldsymbol{x}) = \boldsymbol{x}^H\boldsymbol{U}^H\boldsymbol{U}\boldsymbol{x} = \boldsymbol{x}^H\boldsymbol{x} = \|\boldsymbol{x}\|_2^2$$

für alle $\boldsymbol{x} \in \mathbb{C}^n$, daher gilt $\|\boldsymbol{U}\|_2 = 1$, und da mit $\boldsymbol{U}$ auch $\boldsymbol{U}^{-1} = \boldsymbol{U}^H$ unitär ist, erhält man $\kappa_2(\boldsymbol{U}) = 1$.

Vorbetrachtungen

Wir wissen (vgl. LA Satz 8.31), dass man genau die normalen Matrizen mit unitären Matrizen auf Diagonalgestalt transformieren kann. Allgemeine Matrizen kann man auf obere Dreiecksgestalt transformieren, aus der man dann wieder die Eigenwerte unmittelbar ablesen kann.

Vorbetrachtungen

Wir wissen (vgl. LA Satz 8.31), dass man genau die normalen Matrizen mit unitären Matrizen auf Diagonalgestalt transformieren kann. Allgemeine Matrizen kann man auf obere Dreiecksgestalt transformieren, aus der man dann wieder die Eigenwerte unmittelbar ablesen kann.

Satz 6.3 (Schursche Normalform)

Es sei $\boldsymbol{A} \in \mathbb{C}^{(n,n)}$. Dann existiert eine unitäre Matrix $\boldsymbol{U}$, so dass

$$\boldsymbol{U}^H \boldsymbol{A} \boldsymbol{U} = \boldsymbol{T} \tag{6.4}$$

obere Dreiecksgestalt besitzt. $\boldsymbol{T}$ heißt Schursche Normalform der Matrix $\boldsymbol{A}$.

Vorbetrachtungen

Wir wissen (vgl. LA Satz 8.31), dass man genau die normalen Matrizen mit unitären Matrizen auf Diagonalgestalt transformieren kann. Allgemeine Matrizen kann man auf obere Dreiecksgestalt transformieren, aus der man dann wieder die Eigenwerte unmittelbar ablesen kann.

Satz 6.3 (Schursche Normalform)

Es sei $\boldsymbol{A} \in \mathbb{C}^{(n,n)}$. Dann existiert eine unitäre Matrix $\boldsymbol{U}$, so dass

$$\boldsymbol{U}^H \boldsymbol{A} \boldsymbol{U} = \boldsymbol{T} \tag{6.4}$$

obere Dreiecksgestalt besitzt. $\boldsymbol{T}$ heißt Schursche Normalform der Matrix $\boldsymbol{A}$.

Beweis.

Siehe dazu das Skript „Grundlagen der Numerischen Mathematik“ von Heinrich Voß, Abschnitt 6.1. □

Vorbetrachtungen

Bemerkung 6.4: Besitzt $\boldsymbol{A}$ die Schur-Zerlegung

$$\boldsymbol{U}^H \boldsymbol{A} \boldsymbol{U} = \operatorname{diag}\{\lambda_1, \ldots, \lambda_n\} + \boldsymbol{N}$$

mit einer strikten oberen Dreiecksmatrix $\boldsymbol{N}$, so gilt unabhängig von der Wahl von $\boldsymbol{U}$ für die Schur-Norm

$$\|\boldsymbol{N}\|_S^2 = \|\boldsymbol{U}^H \boldsymbol{A} \boldsymbol{U}\|_S^2 - \sum_{j=1}^{n} |\lambda_j|^2 = \|\boldsymbol{A}\|_S^2 - \sum_{j=1}^{n} |\lambda_j|^2.$$

Vorbetrachtungen

Bemerkung 6.4: Besitzt $\boldsymbol{A}$ die Schur-Zerlegung

$$\boldsymbol{U}^H \boldsymbol{A} \boldsymbol{U} = \operatorname{diag}\{\lambda_1, \ldots, \lambda_n\} + \boldsymbol{N}$$

mit einer strikten oberen Dreiecksmatrix $\boldsymbol{N}$, so gilt unabhängig von der Wahl von $\boldsymbol{U}$ für die Schur-Norm

$$\|\boldsymbol{N}\|_S^2 = \|\boldsymbol{U}^H \boldsymbol{A} \boldsymbol{U}\|_S^2 - \sum_{j=1}^{n} |\lambda_j|^2 = \|\boldsymbol{A}\|_S^2 - \sum_{j=1}^{n} |\lambda_j|^2.$$

Es gilt $\|\boldsymbol{N}\|_S^2 = 0$ genau dann, wenn $\boldsymbol{A}$ eine normale Matrix ist. Die Größe $\|\boldsymbol{N}\|_S =: \Delta(\boldsymbol{A})$ heißt die Abweichung von $\boldsymbol{A}$ von der Normalität. □

Vorbetrachtungen

Bei der bisherigen Behandlung der Ähnlichkeitstransformationen haben wir einen wichtigen Aspekt nicht berücksichtigt. Ist die Ausgangsmatrix $\boldsymbol{A}$ reell, so wird man nur orthogonale Matrizen $\boldsymbol{U}$ (d.h. reelle unitäre Matrizen) zur Transformation benutzen, da sonst $\boldsymbol{U}^H\boldsymbol{A}\boldsymbol{U}$ komplexe Elemente enthält.

Vorbetrachtungen

Bei der bisherigen Behandlung der Ähnlichkeitstransformationen haben wir einen wichtigen Aspekt nicht berücksichtigt. Ist die Ausgangsmatrix $\boldsymbol{A}$ reell, so wird man nur orthogonale Matrizen $\boldsymbol{U}$ (d.h. reelle unitäre Matrizen) zur Transformation benutzen, da sonst $\boldsymbol{U}^H\boldsymbol{A}\boldsymbol{U}$ komplexe Elemente enthält.

Besitzt $\boldsymbol{A}$ komplexe Eigenwerte, so kann $\boldsymbol{A}$ hiermit offenbar nicht auf obere Dreiecksgestalt transformiert werden (die Diagonale enthält ja die Eigenwerte).

Vorbetrachtungen

Bei der bisherigen Behandlung der Ähnlichkeitstransformationen haben wir einen wichtigen Aspekt nicht berücksichtigt. Ist die Ausgangsmatrix $\boldsymbol{A}$ reell, so wird man nur orthogonale Matrizen $\boldsymbol{U}$ (d.h. reelle unitäre Matrizen) zur Transformation benutzen, da sonst $\boldsymbol{U}^H \boldsymbol{A} \boldsymbol{U}$ komplexe Elemente enthält.

Besitzt $\boldsymbol{A}$ komplexe Eigenwerte, so kann $\boldsymbol{A}$ hiermit offenbar nicht auf obere Dreiecksgestalt transformiert werden (die Diagonale enthält ja die Eigenwerte).

Da mit $\lambda \in \mathbb{C} \setminus \mathbb{R}$ auch $\overline{\lambda}$ Eigenwert von $\boldsymbol{A}$ ist (das charakteristische Polynom hat reelle Koeffizienten!), kann man $\boldsymbol{A}$ auf Quasidreiecksgestalt transformieren.

Vorbetrachtungen

Dabei heißt eine Matrix $\boldsymbol{R}$ Quasidreiecksmatrix, falls

$$\boldsymbol{R} = \begin{pmatrix} \boldsymbol{R}_{11} & \boldsymbol{R}_{12} & \boldsymbol{R}_{13} & \dots & \boldsymbol{R}_{1k} \\ \boldsymbol{O} & \boldsymbol{R}_{22} & \boldsymbol{R}_{23} & \dots & \boldsymbol{R}_{2k} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \dots & \boldsymbol{R}_{kk} \end{pmatrix}$$

obere Blockdreiecksgestalt besitzt und die Diagonalblöcke $\boldsymbol{R}_{jj}$ alle die Dimension 1 oder 2 besitzen.

Vorbetrachtungen

Satz 6.5 (Reelle Schursche Normalform)

Sei $\boldsymbol{A} \in \mathbb{R}^{(n,n)}$. Dann existiert eine orthogonale Matrix $\boldsymbol{U}$, so dass $\boldsymbol{U}^T \boldsymbol{A} \boldsymbol{U}$ quasidreieckig ist. Die orthogonale Matrix $\boldsymbol{U}$ kann so gewählt werden, dass jeder 2×2-Diagonalblock $\boldsymbol{R}_{jj}$ nur komplexe Eigenwerte besitzt.

Vorbetrachtungen

Satz 6.5 (Reelle Schursche Normalform)

Sei $\boldsymbol{A} \in \mathbb{R}^{(n,n)}$. Dann existiert eine orthogonale Matrix $\boldsymbol{U}$, so dass $\boldsymbol{U}^T \boldsymbol{A} \boldsymbol{U}$ quasidreieckig ist. Die orthogonale Matrix $\boldsymbol{U}$ kann so gewählt werden, dass jeder 2×2-Diagonalblock $\boldsymbol{R}_{jj}$ nur komplexe Eigenwerte besitzt.

Beweis.

Siehe dazu das Skript „Grundlagen der Numerischen Mathematik“ von Heinrich Voß, Abschnitt 6.1. □

Vorbetrachtungen

Satz 6.5 (Reelle Schursche Normalform)

Sei $\boldsymbol{A} \in \mathbb{R}^{(n,n)}$. Dann existiert eine orthogonale Matrix $\boldsymbol{U}$, so dass $\boldsymbol{U}^T\boldsymbol{A}\boldsymbol{U}$ quasidreieckig ist. Die orthogonale Matrix $\boldsymbol{U}$ kann so gewählt werden, dass jeder 2×2-Diagonalblock $\boldsymbol{R}_{jj}$ nur komplexe Eigenwerte besitzt.

Beweis.

Siehe dazu das Skript „Grundlagen der Numerischen Mathematik“ von Heinrich Voß, Abschnitt 6.1. □

Viele Verfahren zur Bestimmung von Eigenwerten bestehen darin, Matrizen $\boldsymbol{X}_k$ zu bestimmen, so dass $\boldsymbol{X}_k^{-1}\boldsymbol{A}\boldsymbol{X}_k$ mehr und mehr einer Diagonalmatrix gleicht. Wir fragen daher, wie gut die Eigenwerte einer Matrix durch ihre Diagonalelemente approximiert werden.

Vorbetrachtungen

Satz 6.6 (Gerschgorin)

Sei

$$A := (a_{ij}) \in \mathbb{C}^{(n,n)}, \quad z_i := \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}|, \quad s_j := \sum_{\substack{i=1 \\ i \neq j}}^{n} |a_{ij}|.$$

Dann gilt

- Alle Eigenwerte von A liegen in der Vereinigung der Kreise

$$Z_i := \{z \in \mathbb{C} : |z - a_{ii}| \leqslant z_i\}.$$

- Alle Eigenwerte von A liegen in der Vereinigung der Kreise

$$S_j := \{z \in \mathbb{C} : |z - a_{jj}| \leqslant s_j\}.$$

Vorbetrachtungen

Satz 6.6 (Gerschgorin; Fortsetzung)

- Jede Zusammenhangskomponente von

$$\bigcup_{i=1}^{n} Z_i \quad \text{bzw.} \quad \bigcup_{j=1}^{n} S_j$$

enthält genau so viele Eigenwerte von A, wie Kreise an der Komponente beteiligt sind, wobei Kreise und Eigenwerte entsprechend ihrer (algebraischen) Vielfachheit gezählt sind.

Vorbetrachtungen

Satz 6.6 (Gerschgorin; Fortsetzung)

- Jede Zusammenhangskomponente von

$$\bigcup_{i=1}^{n} Z_i \quad \text{bzw.} \quad \bigcup_{j=1}^{n} S_j$$

enthält genau so viele Eigenwerte von A, wie Kreise an der Komponente beteiligt sind, wobei Kreise und Eigenwerte entsprechend ihrer (algebraischen) Vielfachheit gezählt sind.

Beweis.

Siehe dazu LA, Satz 8.84. □

Vorbetrachtungen

Bemerkung 6.7: Die dritte Aussage lässt sich nicht verschärfen zu: „In jedem Kreis S_i bzw. Z_j liegt mindestens ein Eigenwert von $\boldsymbol{A}$“, denn sei

$$\boldsymbol{A} = \begin{pmatrix} 0 & 1 \\ 2 & 0 \end{pmatrix}.$$

Dann sind die Eigenwerte gegeben durch $\lambda_{1/2} = \pm\sqrt{2} \approx \pm 1.4142$ und es liegt kein Eigenwert in $Z_1 = S_2 = \{z : |z| \leqslant 1\}$. □

Vorbetrachtungen

Bemerkung 6.7: Die dritte Aussage lässt sich nicht verschärfen zu: „In jedem Kreis S_i bzw. Z_j liegt mindestens ein Eigenwert von $\boldsymbol{A}$“, denn sei

$$\boldsymbol{A} = \begin{pmatrix} 0 & 1 \\ 2 & 0 \end{pmatrix}.$$

Dann sind die Eigenwerte gegeben durch $\lambda_{1/2} = \pm\sqrt{2} \approx \pm 1.4142$ und es liegt kein Eigenwert in $Z_1 = S_2 = \{z : |z| \leqslant 1\}$. □

Für viele Eigenwert-Routinen kann man zeigen, dass die berechneten Eigenwerte $\lambda_1, \ldots, \lambda_n$ die exakten Eigenwerte einer gestörten Matrix $\boldsymbol{A} + \boldsymbol{F}$ sind, wobei $\boldsymbol{F}$ eine kleine Norm besitzt. Wir fragen daher, wie Störungen die Eigenwerte einer Matrix beeinflussen. Beispielhaft hierfür ist:

Vorbetrachtungen

Satz 6.8 (Bauer, Fike)

Ist μ Eigenwert von $\boldsymbol{A} + \boldsymbol{F} \in \mathbb{C}^{(n,n)}$ und

$$\boldsymbol{X}^{-1}\boldsymbol{A}\boldsymbol{X} = \boldsymbol{\Lambda} = \operatorname{diag}\{\lambda_1, \ldots, \lambda_n\},$$

so gilt

$$\min_{\lambda \in \Lambda(A)} |\lambda - \mu| \leqslant \kappa_p(\boldsymbol{X}) \|\boldsymbol{F}\|_p.$$

Vorbetrachtungen

Satz 6.8 (Bauer, Fike)

Ist μ Eigenwert von $\boldsymbol{A} + \boldsymbol{F} \in \mathbb{C}^{(n,n)}$ und

$$\boldsymbol{X}^{-1}\boldsymbol{A}\boldsymbol{X} = \boldsymbol{\Lambda} = \operatorname{diag}\{\lambda_1, \ldots, \lambda_n\},$$

so gilt

$$\min_{\lambda \in \Lambda(A)} |\lambda - \mu| \leqslant \kappa_p(\boldsymbol{X}) \|\boldsymbol{F}\|_p.$$

Dabei bezeichnet $\|\cdot\|_p$ die von der Höldernorm

$$\|\boldsymbol{x}\|_p := \left(\sum_{j=1}^{n} |x_j|^p \right)^{1/p}, \quad p \geqslant 1,$$

induzierte Matrixnorm und κ_p die Kondition bzgl. $\|\cdot\|_p$.

Vorbetrachtungen

Beweis.

Für $\mu \in \Lambda(\boldsymbol{A})$ ist die Aussage trivial. Sei also $\mu \notin \Lambda(\boldsymbol{A})$.

Vorbetrachtungen

Beweis.

Für $\mu \in \Lambda(\boldsymbol{A})$ ist die Aussage trivial. Sei also $\mu \notin \Lambda(\boldsymbol{A})$.

Da $\boldsymbol{X}^{-1}(\boldsymbol{A} + \boldsymbol{F} - \mu\boldsymbol{E})\boldsymbol{X}$ singulär ist, ist auch

$$\boldsymbol{E} + (\boldsymbol{\Lambda} - \mu\boldsymbol{E})^{-1}(\boldsymbol{X}^{-1}\boldsymbol{F}\boldsymbol{X}) = (\boldsymbol{\Lambda} - \mu\boldsymbol{E})^{-1}\boldsymbol{X}^{-1}(\boldsymbol{A} + \boldsymbol{F} - \mu\boldsymbol{E})\boldsymbol{X}$$

singulär.

Vorbetrachtungen

Beweis.

Für $\mu \in \Lambda(\boldsymbol{A})$ ist die Aussage trivial. Sei also $\mu \notin \Lambda(\boldsymbol{A})$.

Da $\boldsymbol{X}^{-1}(\boldsymbol{A} + \boldsymbol{F} - \mu\boldsymbol{E})\boldsymbol{X}$ singulär ist, ist auch

$$\boldsymbol{E} + (\boldsymbol{\Lambda} - \mu\boldsymbol{E})^{-1}(\boldsymbol{X}^{-1}\boldsymbol{F}\boldsymbol{X}) = (\boldsymbol{\Lambda} - \mu\boldsymbol{E})^{-1}\boldsymbol{X}^{-1}(\boldsymbol{A} + \boldsymbol{F} - \mu\boldsymbol{E})\boldsymbol{X}$$

singulär.

Also existiert $\boldsymbol{y} \in \mathbb{C}^n$, $\|\boldsymbol{y}\|_p = 1$ mit $\boldsymbol{y} + (\boldsymbol{\Lambda} - \mu\boldsymbol{E})^{-1}(\boldsymbol{X}^{-1}\boldsymbol{F}\boldsymbol{X})\boldsymbol{y} = \boldsymbol{o}$. Hieraus folgt

$$\begin{aligned} 1 = \|\boldsymbol{y}\|_p &= \|(\boldsymbol{\Lambda} - \mu\boldsymbol{E})^{-1}(\boldsymbol{X}^{-1}\boldsymbol{F}\boldsymbol{X})\boldsymbol{y}\|_p \\ &\leqslant \|(\boldsymbol{\Lambda} - \mu\boldsymbol{E})^{-1}\boldsymbol{X}^{-1}\boldsymbol{F}\boldsymbol{X}\|_p \\ &\leqslant \Big(\min_{\lambda\in\Lambda(A)} |\lambda - \mu|\Big)^{-1}\|\boldsymbol{X}^{-1}\|_p\|\boldsymbol{F}\|_p\|\boldsymbol{X}\|_p, \end{aligned}$$

Vorbetrachtungen

Beweis.

wobei ausgenutzt wurde, dass die p-Norm einer Diagonalmatrix gleich dem Betrag des betragsmaximalen Elements der Matrix ist. □

Vorbetrachtungen

Beweis.

wobei ausgenutzt wurde, dass die p-Norm einer Diagonalmatrix gleich dem Betrag des betragsmaximalen Elements der Matrix ist. □

Für normale Matrizen erhält man insbesondere

Vorbetrachtungen

Beweis.

wobei ausgenutzt wurde, dass die p-Norm einer Diagonalmatrix gleich dem Betrag des betragsmaximalen Elements der Matrix ist. □

Für normale Matrizen erhält man insbesondere

Korollar 6.9

Es sei $\boldsymbol{A}$ eine normale Matrix und μ ein Eigenwert von $\boldsymbol{A} + \boldsymbol{F} \in \mathbb{C}^{(n,n)}$. Dann gilt

$$\min_{\lambda \in \Lambda(A)} |\lambda - \mu| \leqslant \|\boldsymbol{F}\|_2$$

Vorbetrachtungen

Beweis.

wobei ausgenutzt wurde, dass die p-Norm einer Diagonalmatrix gleich dem Betrag des betragsmaximalen Elements der Matrix ist. □

Für normale Matrizen erhält man insbesondere

Korollar 6.9

Es sei $\boldsymbol{A}$ eine normale Matrix und μ ein Eigenwert von $\boldsymbol{A} + \boldsymbol{F} \in \mathbb{C}^{(n,n)}$. Dann gilt

$$\min_{\lambda \in \Lambda(A)} |\lambda - \mu| \leqslant \|\boldsymbol{F}\|_2$$

Beweis.

Da $\boldsymbol{A}$ normal, kann man in Satz 6.8 $\boldsymbol{X}$ unitär wählen, und die Behauptung folgt dann aus $\kappa_2(\boldsymbol{X}) = 1$. □

Potenzmethode

Wir betrachten in diesem Abschnitt ein Verfahren zur Bestimmung des betragsgrößten Eigenwerts und zugehörigen Eigenvektors, namentlich die Potenzmethode, und eine Modifikation hiervon, um auch andere Eigenwerte zu berechnen.

Potenzmethode

Wir betrachten in diesem Abschnitt ein Verfahren zur Bestimmung des betragsgrößten Eigenwerts und zugehörigen Eigenvektors, namentlich die Potenzmethode, und eine Modifikation hiervon, um auch andere Eigenwerte zu berechnen.

Wir besprechen die Methoden vor allem zum besseren Verständnis des dann im nächsten Abschnitt folgenden Arbeitspferdes zur Berechnung von Eigenwerten und Eigenvektoren, des QR-Algorithmus.

Potenzmethode

Wir betrachten für $\boldsymbol{A} \in \mathbb{C}^{(n,n)}$ die spezielle Eigenwertaufgabe

$$\boldsymbol{A}\boldsymbol{x} = \lambda \boldsymbol{x}. \tag{6.5}$$

Potenzmethode

Wir betrachten für $\boldsymbol{A} \in \mathbb{C}^{(n,n)}$ die spezielle Eigenwertaufgabe

$$\boldsymbol{A}\boldsymbol{x} = \lambda \boldsymbol{x}. \tag{6.5}$$

Es sei $\boldsymbol{A}$ diagonalisierbar, $\boldsymbol{A}\boldsymbol{x}^i = \lambda_i \boldsymbol{x}^i$, $i = 1, \ldots, n$, und es besitze $\boldsymbol{A}$ einen dominanten Eigenwert λ_1, d.h., es gelte

$$|\lambda_1| > |\lambda_2| \geqslant \cdots \geqslant |\lambda_n|. \tag{6.6}$$

Potenzmethode

Es sei $\boldsymbol{u}^0 \in \mathbb{C}^n$, $\boldsymbol{u}^0 \neq \boldsymbol{o}$, gegeben. Multipliziert man

$$\boldsymbol{u}^0 =: \sum_{i=1}^{n} \alpha_i \boldsymbol{x}^i$$

m mal mit der Matrix $\boldsymbol{A}$, so folgt

$$\boldsymbol{A}^m \boldsymbol{u}^0 = \sum_{i=1}^{n} \alpha_i \lambda_i^m \boldsymbol{x}^i = \lambda_1^m \left\{ \alpha_1 \boldsymbol{x}^1 + \sum_{i=2}^{n} \alpha_i \left(\frac{\lambda_i}{\lambda_1} \right)^m \boldsymbol{x}^i \right\}. \tag{6.7}$$

Potenzmethode

Es sei $\boldsymbol{u}^0 \in \mathbb{C}^n$, $\boldsymbol{u}^0 \neq \boldsymbol{o}$, gegeben. Multipliziert man

$$\boldsymbol{u}^0 =: \sum_{i=1}^{n} \alpha_i \boldsymbol{x}^i$$

m mal mit der Matrix $\boldsymbol{A}$, so folgt

$$\boldsymbol{A}^m \boldsymbol{u}^0 = \sum_{i=1}^{n} \alpha_i \lambda_i^m \boldsymbol{x}^i = \lambda_1^m \left\{ \alpha_1 \boldsymbol{x}^1 + \sum_{i=2}^{n} \alpha_i \left(\frac{\lambda_i}{\lambda_1} \right)^m \boldsymbol{x}^i \right\}. \tag{6.7}$$

Wegen (6.6) konvergiert daher die Folge $\{\lambda_1^{-m} \boldsymbol{A}^m \boldsymbol{u}^0\}$ gegen ein Vielfaches von $\boldsymbol{x}^1$, falls $\alpha_1 \neq 0$ gilt, d.h. $\{\boldsymbol{A}^m \boldsymbol{u}^0\}$ konvergiert gegen die Eigenrichtung $\operatorname{span}\{\boldsymbol{x}^1\}$ zum dominanten Eigenwert λ_1, wenn der Startvektor $\boldsymbol{u}^0$ eine Komponente in Richtung von $\boldsymbol{x}^1$ besitzt.

Potenzmethode

Gilt $|\lambda_1| \neq 1$, so erhält man für große m Exponentenüberlauf oder -unterlauf. Man wird also die Folge $\{\boldsymbol{A}^m \boldsymbol{u}^0\}$ noch geeignet normieren und erhält die Potenzmethode oder das von-Mises-Verfahren.

Potenzmethode

Gilt $|\lambda_1| \neq 1$, so erhält man für große m Exponentenüberlauf oder -unterlauf. Man wird also die Folge $\{\boldsymbol{A}^m \boldsymbol{u}^0\}$ noch geeignet normieren und erhält die Potenzmethode oder das von-Mises-Verfahren.

Sei $\|\cdot\|$ irgendeine Norm auf $\mathbb{C}^n$ und $\boldsymbol{u}^0 \in \mathbb{C}^n \setminus \{\boldsymbol{o}\}$.

Potenzmethode

Gilt $|\lambda_1| \neq 1$, so erhält man für große m Exponentenüberlauf oder -unterlauf. Man wird also die Folge $\{\boldsymbol{A}^m \boldsymbol{u}^0\}$ noch geeignet normieren und erhält die Potenzmethode oder das von-Mises-Verfahren.

Sei $\|\cdot\|$ irgendeine Norm auf $\mathbb{C}^n$ und $\boldsymbol{u}^0 \in \mathbb{C}^n \setminus \{\boldsymbol{o}\}$.

Algorithmus 6.10 (Potenzmethode):

```
for m = 0,1,... until convergence do
   v_{m+1} = A u_m;
   k_{m+1} = ||v_{m+1}||;
   u_{m+1} = v_{m+1} / k_{m+1};
end
```

Potenzmethode

Häufig wird eine andere Skalierung der erzeugten Folge gewählt, die billiger ist als die Normierung. Sei dazu $\ell \in \mathbb{C}^n$ ein gegebener Vektor. Hiermit iterieren wir nun gemäß

Algorithmus 6.11 (Potenzmethode):

```
for m = 0,1,... until convergence do
   v_{m+1} = A u_m;
   k_{m+1} = ℓ' * v_{m+1};
   u_{m+1} = v_{m+1} / k_{m+1};
end
```

Potenzmethode

Satz 6.12

Es sei λ_1 dominanter Eigenwert von $\boldsymbol{A}$ und

$$\boldsymbol{u}^0 := \sum_{i=1}^{n} \alpha_i \boldsymbol{x}^i$$

mit $\alpha_1 \neq 0$ und $\boldsymbol{\ell} \in \mathbb{C}^n$.

Potenzmethode

Satz 6.12

Es sei λ_1 dominanter Eigenwert von $\boldsymbol{A}$ und

$$\boldsymbol{u}^0 := \sum_{i=1}^{n} \alpha_i \boldsymbol{x}^i$$

mit $\alpha_1 \neq 0$ und $\boldsymbol{\ell} \in \mathbb{C}^n$.

Es seien $\boldsymbol{u}^m$ und k_m nach Algorithmus 6.11 berechnet.

Potenzmethode

Satz 6.12

Es sei λ_1 dominanter Eigenwert von $\boldsymbol{A}$ und

$$\boldsymbol{u}^0 := \sum_{i=1}^{n} \alpha_i \boldsymbol{x}^i$$

mit $\alpha_1 \neq 0$ und $\boldsymbol{\ell} \in \mathbb{C}^n$.

Es seien $\boldsymbol{u}^m$ und k_m nach Algorithmus 6.11 berechnet.

Gilt dann $\boldsymbol{\ell}^H \boldsymbol{u}^m \neq 0$ für alle $m \in \mathbb{N}$ und $\boldsymbol{\ell}^H \boldsymbol{x}^1 \neq 0$, so ist

$$\lim_{m\to\infty} k_m = \lambda_1, \qquad \lim_{m\to\infty} \boldsymbol{u}^m = \frac{1}{\boldsymbol{\ell}^H \boldsymbol{x}^1} \boldsymbol{x}^1.$$

Potenzmethode

Beweis.

Offensichtlich gilt

$$\boldsymbol{u}^m = \frac{\boldsymbol{A}^m \boldsymbol{u}^0}{\boldsymbol{\ell}^H \boldsymbol{A}^m \boldsymbol{u}^0}.$$

Potenzmethode

Beweis.

Offensichtlich gilt

$$\boldsymbol{u}^m = \frac{\boldsymbol{A}^m \boldsymbol{u}^0}{\boldsymbol{\ell}^H \boldsymbol{A}^m \boldsymbol{u}^0}.$$

Daher ist

$$k_m = \boldsymbol{\ell}^H \boldsymbol{v}^m = \boldsymbol{\ell}^H \boldsymbol{A} \boldsymbol{u}^{m-1} = \boldsymbol{\ell}^H \boldsymbol{A} \frac{\boldsymbol{A}^{m-1} \boldsymbol{u}^0}{\boldsymbol{\ell}^H \boldsymbol{A}^{m-1} \boldsymbol{u}^0} = \frac{\boldsymbol{\ell}^H \boldsymbol{A}^m \boldsymbol{u}^0}{\boldsymbol{\ell}^H \boldsymbol{A}^{m-1} \boldsymbol{u}^0}$$

Potenzmethode

Beweis.

Offensichtlich gilt

$$\boldsymbol{u}^m = \frac{\boldsymbol{A}^m \boldsymbol{u}^0}{\boldsymbol{\ell}^H \boldsymbol{A}^m \boldsymbol{u}^0}.$$

Daher ist

$$k_m = \boldsymbol{\ell}^H \boldsymbol{v}^m = \boldsymbol{\ell}^H \boldsymbol{A} \boldsymbol{u}^{m-1} = \boldsymbol{\ell}^H \boldsymbol{A} \frac{\boldsymbol{A}^{m-1} \boldsymbol{u}^0}{\boldsymbol{\ell}^H \boldsymbol{A}^{m-1} \boldsymbol{u}^0} = \frac{\boldsymbol{\ell}^H \boldsymbol{A}^m \boldsymbol{u}^0}{\boldsymbol{\ell}^H \boldsymbol{A}^{m-1} \boldsymbol{u}^0}$$

und aus

$$\lim_{m \to \infty} \lambda_1^{-m} \boldsymbol{A}^m \boldsymbol{u}^0 = \alpha_1 \boldsymbol{x}^1$$

Potenzmethode

Beweis.

Offensichtlich gilt

$$\boldsymbol{u}^m = \frac{\boldsymbol{A}^m \boldsymbol{u}^0}{\boldsymbol{\ell}^H \boldsymbol{A}^m \boldsymbol{u}^0}.$$

Daher ist

$$k_m = \boldsymbol{\ell}^H \boldsymbol{v}^m = \boldsymbol{\ell}^H \boldsymbol{A} \boldsymbol{u}^{m-1} = \boldsymbol{\ell}^H \boldsymbol{A} \frac{\boldsymbol{A}^{m-1} \boldsymbol{u}^0}{\boldsymbol{\ell}^H \boldsymbol{A}^{m-1} \boldsymbol{u}^0} = \frac{\boldsymbol{\ell}^H \boldsymbol{A}^m \boldsymbol{u}^0}{\boldsymbol{\ell}^H \boldsymbol{A}^{m-1} \boldsymbol{u}^0}$$

und aus

$$\lim_{m \to \infty} \lambda_1^{-m} \boldsymbol{A}^m \boldsymbol{u}^0 = \alpha_1 \boldsymbol{x}^1$$

folgt

$$\lim_{m \to \infty} \lambda_1^{-1} k_m = \frac{\lim\limits_{m \to \infty} \boldsymbol{\ell}^H (\lambda_1^{-m} \boldsymbol{A}^m \boldsymbol{u}^0)}{\lim\limits_{m \to \infty} \boldsymbol{\ell}^H (\lambda_1^{-(m-1)} \boldsymbol{A}^{m-1} \boldsymbol{u}^0)} = 1,$$

Potenzmethode

Beweis.

d.h.

$$\lim_{m \to \infty} k_m = \lambda_1$$

Potenzmethode

Beweis.

d.h.

$$\lim_{m\to\infty} k_m = \lambda_1$$

und

$$\begin{aligned}\lim_{m\to\infty} \boldsymbol{u}^m &= \lim_{m\to\infty} \frac{\boldsymbol{A}^m \boldsymbol{u}^0}{\boldsymbol{\ell}^H \boldsymbol{A}^m \boldsymbol{u}^0} \\ &= \frac{\alpha_1 \boldsymbol{x}^1 + \sum_{i=2}^{n} \alpha_i (\lambda_i/\lambda_1)^m \boldsymbol{x}^i}{\alpha_1 \boldsymbol{\ell}^H \boldsymbol{x}^1 + \sum_{i=2}^{n} \alpha_1 (\lambda_i/\lambda_1)^m \boldsymbol{\ell}^H \boldsymbol{x}^i} \\ &= \frac{\boldsymbol{x}^1}{\boldsymbol{\ell}^H \boldsymbol{x}^1}.\end{aligned}$$

Potenzmethode

Bemerkung 6.13:

- Eine häufige Wahl von ℓ ist $\ell = e^k$ für ein $k \in \{1, \ldots, n\}$, d.h. man normiert im Eigenvektor die $k$te Komponente zu 1, wobei man oft nicht vorhersagen kann, welches k geeignet ist (eine Ausnahme bilden symmetrische Tridiagonalmatrizen).

Potenzmethode

Bemerkung 6.13:

- Eine häufige Wahl von ℓ ist $\ell = e^k$ für ein $k \in \{1, \ldots, n\}$, d.h. man normiert im Eigenvektor die $k$te Komponente zu 1, wobei man oft nicht vorhersagen kann, welches k geeignet ist (eine Ausnahme bilden symmetrische Tridiagonalmatrizen).
- Ist $A \in \mathbb{R}^{(n,n)}$ nichtnegativ und irreduzibel, so ist der zum Spektralradius gehörende Eigenvektor positiv und man kann $\ell = (1, 1, \ldots, 1)^T$ wählen, wenn u^0 positive Komponenten hat.

Potenzmethode

Bemerkung 6.13:

- Eine häufige Wahl von ℓ ist $\ell = e^k$ für ein $k \in \{1, \ldots, n\}$, d.h. man normiert im Eigenvektor die $k$te Komponente zu 1, wobei man oft nicht vorhersagen kann, welches k geeignet ist (eine Ausnahme bilden symmetrische Tridiagonalmatrizen).
- Ist $A \in \mathbb{R}^{(n,n)}$ nichtnegativ und irreduzibel, so ist der zum Spektralradius gehörende Eigenvektor positiv und man kann $\ell = (1, 1, \ldots, 1)^T$ wählen, wenn u^0 positive Komponenten hat.
- Wir haben $\alpha_1 \neq 0$ vorausgesetzt. Diese Voraussetzung ist nicht einschneidend, denn durch Rundungsfehler wird (fast immer) ein u^m erzeugt, das eine Komponente in Richtung von x^1 besitzt.

Potenzmethode

Bemerkung 6.13:

- Eine häufige Wahl von ℓ ist $\ell = e^k$ für ein $k \in \{1, \ldots, n\}$, d.h. man normiert im Eigenvektor die $k$te Komponente zu 1, wobei man oft nicht vorhersagen kann, welches k geeignet ist (eine Ausnahme bilden symmetrische Tridiagonalmatrizen).
- Ist $A \in \mathbb{R}^{(n,n)}$ nichtnegativ und irreduzibel, so ist der zum Spektralradius gehörende Eigenvektor positiv und man kann $\ell = (1, 1, \ldots, 1)^T$ wählen, wenn u^0 positive Komponenten hat.
- Wir haben $\alpha_1 \neq 0$ vorausgesetzt. Diese Voraussetzung ist nicht einschneidend, denn durch Rundungsfehler wird (fast immer) ein u^m erzeugt, das eine Komponente in Richtung von x^1 besitzt.
- Ist λ_1 mehrfacher Eigenwert von A und A diagonalähnlich, so bleiben alle Überlegungen richtig, wenn nur u^0 eine Komponente in Richtung des Eigenraumes von A zu λ_1 besitzt oder eine solche durch Rundungsfehler erzeugt wird.

Potenzmethode

- Ist $\boldsymbol{A} \in \mathbb{R}^{(n,n)}$ mit $|\lambda_1| = |\lambda_2| > |\lambda_3| \geqslant \ldots \geqslant |\lambda_n|$ und $\lambda_1 \neq \lambda_2$ (also $\lambda_1 = \bar{\lambda}_2$ oder $\lambda_1 = -\lambda_2$), so erhält man keine Konvergenz, sondern es treten in der Folge $\{\boldsymbol{u}^m\}$ Oszillationen auf. Auch in diesem Fall kann man aus (drei aufeinanderfolgenden Elementen) der Folge $\{\boldsymbol{u}^m\}$ Näherungen für die zu λ_1 und λ_2 gehörenden Eigenvektoren ermitteln. Eine ausführliche Diskussion findet man in Zurmühl, pp. 283 ff.

Potenzmethode

- Ist $\boldsymbol{A} \in \mathbb{R}^{(n,n)}$ mit $|\lambda_1| = |\lambda_2| > |\lambda_3| \geqslant \ldots \geqslant |\lambda_n|$ und $\lambda_1 \neq \lambda_2$ (also $\lambda_1 = \bar{\lambda}_2$ oder $\lambda_1 = -\lambda_2$), so erhält man keine Konvergenz, sondern es treten in der Folge $\{\boldsymbol{u}^m\}$ Oszillationen auf. Auch in diesem Fall kann man aus (drei aufeinanderfolgenden Elementen) der Folge $\{\boldsymbol{u}^m\}$ Näherungen für die zu λ_1 und λ_2 gehörenden Eigenvektoren ermitteln. Eine ausführliche Diskussion findet man in Zurmühl, pp. 283 ff.
- Schließlich haben wir vorausgesetzt, dass $\boldsymbol{A}$ diagonalisierbar ist. Ist dies nicht erfüllt, so muss man $\boldsymbol{u}^0$ als Linearkombination von Hauptvektoren darstellen und erhält ebenfalls Konvergenz des Verfahrens (vgl. Collatz, pp. 325 ff.).

Potenzmethode

- Für die Konvergenzgeschwindigkeit gilt mit $q := \max_{i=2,\dots,m} |\lambda_i/\lambda_1|$

$$
\begin{aligned}
|k_{m+1} - \lambda_1| &= \left| \frac{\boldsymbol{\ell}^H \boldsymbol{A}^{m+1} \boldsymbol{u}^0}{\boldsymbol{\ell}^H \boldsymbol{A}^m \boldsymbol{u}^0} - \lambda_1 \right| \\
&= \left| \frac{1}{\boldsymbol{\ell}^H \boldsymbol{A}^m \boldsymbol{u}^0} \boldsymbol{\ell}^H (\boldsymbol{A}^{m+1} \boldsymbol{u}^0 - \lambda_1 \boldsymbol{A}^m \boldsymbol{u}^0) \right| \\
&= \left| \frac{\lambda_1^{m+1}}{\boldsymbol{\ell}^H \boldsymbol{A}^m \boldsymbol{u}^0} \boldsymbol{\ell}^H \left(\sum_{i=2}^{n} \alpha_i \left(\left(\frac{\lambda_i}{\lambda_1} \right)^{m+1} - \left(\frac{\lambda_i}{\lambda_1} \right)^m \right) \boldsymbol{x}^i \right) \right| \\
&\leqslant \left| \frac{\lambda_1^{m+1}}{\boldsymbol{\ell}^H \boldsymbol{A}^m \boldsymbol{u}^0} \right| \|\boldsymbol{\ell}\|_2 \sum_{i=2}^{n} |\alpha_i| \left| \frac{\lambda_i}{\lambda_1} \right|^m \left| \frac{\lambda_i}{\lambda_1} - 1 \right| \|\boldsymbol{x}^i\|_2 \\
&\leqslant C q^m \qquad (6.8)
\end{aligned}
$$

d.h. wir haben lineare Konvergenz mit Konvergenzfaktor q.

Potenzmethode

Ist also $|\lambda_1|$ von den Beträgen der übrigen Eigenwerte gut getrennt, so erhält man rasche Konvergenz, i. A. ist das Verfahren aber sehr langsam.

Potenzmethode

Ist also $|\lambda_1|$ von den Beträgen der übrigen Eigenwerte gut getrennt, so erhält man rasche Konvergenz, i. A. ist das Verfahren aber sehr langsam.

- Bisweilen kann die Konvergenzgeschwindigkeit durch eine geschickte Spektralverschiebung verbessert werden, d.h. betrachte $\boldsymbol{A} + \alpha \boldsymbol{E}$, $\alpha \in \mathbb{C}$.

Dann gehen die Eigenwerte über in $\lambda_i + \alpha$, und die Konvergenzgeschwindigkeit wird bestimmt durch

$$q := \max_{i=2,\ldots,n} \left| \frac{\lambda_i + \alpha}{\lambda_1 + \alpha} \right| .$$

Die Verbesserung hierdurch ist meistens unwesentlich.

Potenzmethode

Eine erhebliche Beschleunigung der von-Mises-Iteration erhält man durch die inverse Iteration, die 1944 von Wielandt eingeführt wurde bei der Stabilitätsanalyse von Strukturen, die kleine Störungen bekannter Systeme sind. In diesem Fall sind gute Approximationen für die relevanten Eigenwerte bekannt und man erhält (wie wir noch sehen werden) rasche Konvergenz.

Potenzmethode

Eine erhebliche Beschleunigung der von-Mises-Iteration erhält man durch die inverse Iteration, die 1944 von Wielandt eingeführt wurde bei der Stabilitätsanalyse von Strukturen, die kleine Störungen bekannter Systeme sind. In diesem Fall sind gute Approximationen für die relevanten Eigenwerte bekannt und man erhält (wie wir noch sehen werden) rasche Konvergenz.

Heute wird die inverse Iteration vor allem verwendet zur Bestimmung von Eigenvektoren, wenn wenig Eigenwerte und Eigenvektoren gesucht sind.

Potenzmethode

Sei $\lambda \neq \lambda_i$ für alle i. Dann besitzt die Matrix $(\lambda \boldsymbol{E} - \boldsymbol{A})^{-1}$ die Eigenwerte $1/(\lambda - \lambda_i)$ und die Eigenvektoren stimmen mit denen von $\boldsymbol{A}$ überein.

Potenzmethode

Sei $\lambda \neq \lambda_i$ für alle i. Dann besitzt die Matrix $(\lambda \boldsymbol{E} - \boldsymbol{A})^{-1}$ die Eigenwerte $1/(\lambda - \lambda_i)$ und die Eigenvektoren stimmen mit denen von $\boldsymbol{A}$ überein.

Führt man die von-Mises-Iteration für $(\lambda \boldsymbol{E} - \boldsymbol{A})^{-1}$ aus und gilt

$$|\lambda - \lambda_i| < |\lambda - \lambda_j|, \quad j = 1, \ldots, n, \quad j \neq i,$$

mit einem einfachen Eigenwert λ_i von $\boldsymbol{A}$, so ist $(\lambda - \lambda_i)^{-1}$ dominanter Eigenwert von $(\lambda \boldsymbol{E} - \boldsymbol{A})^{-1}$ und die Konvergenzgeschwindigkeit wird bestimmt durch

$$q := \max_{j \neq i} \left| \frac{\lambda - \lambda_i}{\lambda - \lambda_j} \right|.$$

Potenzmethode

Sei $\lambda \neq \lambda_i$ für alle i. Dann besitzt die Matrix $(\lambda \boldsymbol{E} - \boldsymbol{A})^{-1}$ die Eigenwerte $1/(\lambda - \lambda_i)$ und die Eigenvektoren stimmen mit denen von $\boldsymbol{A}$ überein.

Führt man die von-Mises-Iteration für $(\lambda \boldsymbol{E} - \boldsymbol{A})^{-1}$ aus und gilt

$$|\lambda - \lambda_i| < |\lambda - \lambda_j|, \quad j = 1, \ldots, n, \quad j \neq i,$$

mit einem einfachen Eigenwert λ_i von $\boldsymbol{A}$, so ist $(\lambda - \lambda_i)^{-1}$ dominanter Eigenwert von $(\lambda \boldsymbol{E} - \boldsymbol{A})^{-1}$ und die Konvergenzgeschwindigkeit wird bestimmt durch

$$q := \max_{j \neq i} \left| \frac{\lambda - \lambda_i}{\lambda - \lambda_j} \right|.$$

Ist also λ eine gute Näherung für λ_i, so erhält man sehr schnelle Konvergenz.

Potenzmethode

Algorithmus 6.14 (Inverse Iteration; feste Shifts):

```
for m = 0,1,... until convergence do
   Löse (λE−A) v_{m+1} = u_m;
   k_{m+1} = ℓ' * v_{m+1};
   u_{m+1} = v_{m+1} / k_{m+1};
end
```

Potenzmethode

Algorithmus 6.14 (Inverse Iteration; feste Shifts):

```
for m = 0,1,... until convergence do
   Löse (λE−A) v_{m+1} = u_m;
   k_{m+1} = ℓ' * v_{m+1};
   u_{m+1} = v_{m+1} / k_{m+1};
end
```

Wie in Satz 6.12 erhält man im Fall $|\lambda - \lambda_i| < |\lambda - \lambda_j|$ für alle $j \neq i$

$$\boldsymbol{u}^m \to \frac{\boldsymbol{x}^i}{\boldsymbol{\ell}^H \boldsymbol{x}^i}, \quad k_m \to \frac{1}{\lambda - \lambda_i}$$

unter den entsprechenden Voraussetzungen $\alpha_i \neq 0$, $\boldsymbol{\ell}^H \boldsymbol{v}^m \neq 0$, $\boldsymbol{\ell}^H \boldsymbol{x}^i \neq 0$.

Potenzmethode

Die Konvergenz ist natürlich linear (vgl. (6.8)). Dies wird verbessert, wenn man λ gleichzeitig iteriert:

Potenzmethode

Die Konvergenz ist natürlich linear (vgl. (6.8)). Dies wird verbessert, wenn man λ gleichzeitig iteriert:

Algorithmus 6.15 (Inverse Iteration; variable Shifts):

```
for m = 0,1,... until convergence do
   Löse (λ_m E-A) v_{m+1} = u_m;
   k_{m+1} = ℓ' * v_{m+1};
   u_{m+1} = v_{m+1} / k_{m+1};
   λ_{m+1} = λ_m - 1 / k_{m+1};
end
```

Potenzmethode

Die Konvergenz ist natürlich linear (vgl. (6.8)). Dies wird verbessert, wenn man λ gleichzeitig iteriert:

Algorithmus 6.15 (Inverse Iteration; variable Shifts):

```
for m = 0,1,... until convergence do
   Löse (λ_m E-A) v_{m+1} = u_m;
   k_{m+1} = ℓ' * v_{m+1};
   u_{m+1} = v_{m+1} / k_{m+1};
   λ_{m+1} = λ_m - 1 / k_{m+1};
end
```

Es gilt $k_{m+1} \to (\lambda - \lambda_i)^{-1}$. Hieraus erhält man eine Schätzung $\lambda_{(m+1)}$ für λ_i durch

$$k_{m+1} \approx 1/(\lambda_{(m)} - \lambda_{(m+1)})$$

und damit die Aufdatierung des Shift-Parameters in Algorithmus 6.15.

Potenzmethode

Satz 6.16

Sei $\tilde{\lambda}$ ein algebraisch einfacher Eigenwert von $\boldsymbol{A}$ mit zugehörigem Eigenvektor $\tilde{\boldsymbol{u}}$ und es gelte $\boldsymbol{\ell}^H \tilde{\boldsymbol{u}} = \boldsymbol{\ell}^H \boldsymbol{u}^0 = 1$. Dann konvergiert das Verfahren in Algorithmus 6.15 lokal quadratisch gegen $\tilde{\lambda}$ bzw. $\tilde{\boldsymbol{u}}$.

Potenzmethode

Satz 6.16

Sei $\tilde{\lambda}$ ein algebraisch einfacher Eigenwert von $\boldsymbol{A}$ mit zugehörigem Eigenvektor $\tilde{\boldsymbol{u}}$ und es gelte $\boldsymbol{\ell}^H \tilde{\boldsymbol{u}} = \boldsymbol{\ell}^H \boldsymbol{u}^0 = 1$. Dann konvergiert das Verfahren in Algorithmus 6.15 lokal quadratisch gegen $\tilde{\lambda}$ bzw. $\tilde{\boldsymbol{u}}$.

Beweis.

Wir zeigen, dass das Verfahren genau die Iterationsvorschrift des Newton-Verfahrens für das Gleichungssystem

$$F(\boldsymbol{u}, \lambda) := \begin{pmatrix} \lambda \boldsymbol{u} - \boldsymbol{A}\boldsymbol{u} \\ \boldsymbol{\ell}^H \boldsymbol{u} - 1 \end{pmatrix} = \begin{pmatrix} \boldsymbol{o} \\ 0 \end{pmatrix}$$

ist und dass $F'(\tilde{\boldsymbol{u}}, \tilde{\lambda})$ regulär ist. Dann folgt die quadratische Konvergenz sowohl für den Eigenvektor als auch den Eigenwert aus einem Satz in Mathematik III. Wir werden auf das Newton-Verfahren in Kapitel 7 eingehen.

Potenzmethode

Beweis.

Es gilt

$$F'(\boldsymbol{u},\lambda) = \begin{pmatrix} \lambda \boldsymbol{E} - \boldsymbol{A} & \boldsymbol{u} \\ \boldsymbol{\ell}^H & 0 \end{pmatrix}.$$

Das Newton-Verfahren ist also gegeben durch

$$\begin{pmatrix} \lambda_{(m)}\boldsymbol{E} - \boldsymbol{A} & \boldsymbol{u}^m \\ \boldsymbol{\ell}^H & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{u}^{m+1} - \boldsymbol{u}^m \\ \lambda_{(m+1)} - \lambda_{(m)} \end{pmatrix} = - \begin{pmatrix} \lambda_{(m)}\boldsymbol{u}^m - \boldsymbol{A}\boldsymbol{u}^m \\ \boldsymbol{\ell}^H \boldsymbol{u}^m - 1 \end{pmatrix}$$

d.h.

$$\begin{aligned} (\lambda_{(m)}\boldsymbol{E} - \boldsymbol{A})\boldsymbol{u}^{m+1} + \left(\lambda_{(m+1)} - \lambda_{(m)}\right)\boldsymbol{u}^m &= \boldsymbol{o} \\ \boldsymbol{\ell}^H \boldsymbol{u}^{m+1} &= 1, \end{aligned}$$

und dies ist äquivalent zu Algorithmus 6.15.

Potenzmethode

Beweis.

Es sei nun im Gegensatz zur Annahme $F'(\tilde{\boldsymbol{u}}, \tilde{\lambda})$ singulär,

$$F'(\tilde{\boldsymbol{u}}, \tilde{\lambda}) \begin{pmatrix} \boldsymbol{v} \\ \mu \end{pmatrix} = \boldsymbol{o},$$

d.h.

$$(\tilde{\lambda}\boldsymbol{E} - \boldsymbol{A})\boldsymbol{v} + \mu\tilde{\boldsymbol{u}} = \boldsymbol{o}, \quad \boldsymbol{\ell}^H \boldsymbol{v} = 0.$$

Potenzmethode

Beweis.

Es sei nun im Gegensatz zur Annahme $F'(\tilde{\boldsymbol{u}}, \tilde{\lambda})$ singulär,

$$F'(\tilde{\boldsymbol{u}}, \tilde{\lambda}) \begin{pmatrix} \boldsymbol{v} \\ \mu \end{pmatrix} = \boldsymbol{o},$$

d.h.

$$(\tilde{\lambda}\boldsymbol{E} - \boldsymbol{A})\boldsymbol{v} + \mu\tilde{\boldsymbol{u}} = \boldsymbol{o}, \quad \boldsymbol{\ell}^H\boldsymbol{v} = 0.$$

Im Falle $\mu = 0$ gilt

$$(\tilde{\lambda}\boldsymbol{E} - \boldsymbol{A})\boldsymbol{v} = \boldsymbol{o},$$

d.h., da $\tilde{\lambda}$ ein einfacher Eigenwert von $\boldsymbol{A}$ ist, $\boldsymbol{v} = \alpha\tilde{\boldsymbol{u}}$ für ein $\alpha \in \mathbb{C}$ und es folgt $0 = \boldsymbol{\ell}^H\tilde{\boldsymbol{u}} = \alpha$.

Potenzmethode

Beweis.

Es sei nun im Gegensatz zur Annahme $F'(\tilde{\boldsymbol{u}}, \tilde{\lambda})$ singulär,

$$F'(\tilde{\boldsymbol{u}}, \tilde{\lambda}) \begin{pmatrix} \boldsymbol{v} \\ \mu \end{pmatrix} = \boldsymbol{o},$$

d.h.

$$(\tilde{\lambda}\boldsymbol{E} - \boldsymbol{A})\boldsymbol{v} + \mu\tilde{\boldsymbol{u}} = \boldsymbol{o}, \quad \boldsymbol{\ell}^H \boldsymbol{v} = 0.$$

Im Falle $\mu = 0$ gilt

$$(\tilde{\lambda}\boldsymbol{E} - \boldsymbol{A})\boldsymbol{v} = \boldsymbol{o},$$

d.h., da $\tilde{\lambda}$ ein einfacher Eigenwert von $\boldsymbol{A}$ ist, $\boldsymbol{v} = \alpha\tilde{\boldsymbol{u}}$ für ein $\alpha \in \mathbb{C}$ und es folgt $0 = \boldsymbol{\ell}^H \tilde{\boldsymbol{u}} = \alpha$.

Damit ist dann aber der Vektor aus dem Nullraum von $F'(\tilde{\boldsymbol{u}}, \tilde{\lambda})$ der Nullvektor, was die Regularität der Jacobi-Matrix ergibt.

Potenzmethode

Beweis.

Im Falle $\mu \neq 0$ gilt

$$(\tilde{\lambda}\boldsymbol{E} - \boldsymbol{A})\boldsymbol{v} = -\mu\tilde{\boldsymbol{u}} \neq \boldsymbol{o}$$

und daher

$$(\tilde{\lambda}\boldsymbol{E} - \boldsymbol{A})^2\boldsymbol{v} = -\mu(\tilde{\lambda}\boldsymbol{E} - \boldsymbol{A})\tilde{\boldsymbol{u}} = \boldsymbol{o}.$$

Der Vektor $\boldsymbol{v}$ ist also ein Hauptvektor 2. Stufe und daher besitzt $\tilde{\lambda}$ mindestens die algebraische Vielfachheit 2. Dieses ergibt einen Widerspruch zur Annahme. □

Potenzmethode

Beweis.

Im Falle $\mu \neq 0$ gilt

$$(\tilde{\lambda}\boldsymbol{E} - \boldsymbol{A})\boldsymbol{v} = -\mu\tilde{\boldsymbol{u}} \neq \boldsymbol{o}$$

und daher

$$(\tilde{\lambda}\boldsymbol{E} - \boldsymbol{A})^2\boldsymbol{v} = -\mu(\tilde{\lambda}\boldsymbol{E} - \boldsymbol{A})\tilde{\boldsymbol{u}} = \boldsymbol{o}.$$

Der Vektor $\boldsymbol{v}$ ist also ein Hauptvektor 2. Stufe und daher besitzt $\tilde{\lambda}$ mindestens die algebraische Vielfachheit 2. Dieses ergibt einen Widerspruch zur Annahme. □

Bemerkung 6.17: Man entnimmt dem Beweis sofort, dass das Verfahren in Algorithmus 6.14 auch dann lokal konvergiert, wenn $\boldsymbol{A}$ nicht diagonalisierbar ist. □

Potenzmethode

Die inverse Iteration konvergiert sogar kubisch, wenn die Matrix $\boldsymbol{A}$ symmetrisch ist und wenn die Näherung für den Eigenwert aufdatiert wird mit dem Rayleighschen Quotienten

$$\lambda_{(m)} = \frac{(\boldsymbol{u}^m)^H \boldsymbol{A}\boldsymbol{u}^m}{\|\boldsymbol{u}^m\|_2^2}.$$

Potenzmethode

Die inverse Iteration konvergiert sogar kubisch, wenn die Matrix $\boldsymbol{A}$ symmetrisch ist und wenn die Näherung für den Eigenwert aufdatiert wird mit dem Rayleighschen Quotienten

$$\lambda_{(m)} = \frac{(\boldsymbol{u}^m)^H \boldsymbol{A}\boldsymbol{u}^m}{\|\boldsymbol{u}^m\|_2^2}.$$

Diese Aufdatierung ist auch bei nicht symmetrischen Matrizen sinnvoll wegen des folgenden Lemmas. Man erhält dann wie in Satz 6.16 quadratische Konvergenz (s. Stewart, pp. 346 ff.).

Potenzmethode

Lemma 6.18

Sei $\boldsymbol{A} \in \mathbb{C}^{(n,n)}$ beliebig, $\boldsymbol{x} \in \mathbb{C}^n$ und für $\mu \in \mathbb{C}$

$$\boldsymbol{r}(\mu) := \boldsymbol{A}\boldsymbol{x} - \mu\boldsymbol{x}$$

der zugehörige Defekt. Dann gilt

$$\|\boldsymbol{r}(\mu)\|_2 = \min \quad \Leftrightarrow \quad \mu = \frac{\boldsymbol{x}^H \boldsymbol{A}\boldsymbol{x}}{\|\boldsymbol{x}\|_2^2}.$$

Potenzmethode

Lemma 6.18

Sei $\boldsymbol{A} \in \mathbb{C}^{(n,n)}$ beliebig, $\boldsymbol{x} \in \mathbb{C}^n$ und für $\mu \in \mathbb{C}$

$$\boldsymbol{r}(\mu) := \boldsymbol{A}\boldsymbol{x} - \mu\boldsymbol{x}$$

der zugehörige Defekt. Dann gilt

$$\|\boldsymbol{r}(\mu)\|_2 = \min \quad \Leftrightarrow \quad \mu = \frac{\boldsymbol{x}^H \boldsymbol{A}\boldsymbol{x}}{\|\boldsymbol{x}\|_2^2}.$$

Beweis.

Die Aufgabe $\|\boldsymbol{r}(\mu)\|_2 = \min$ ist ein lineares Ausgleichsproblem zur Bestimmung von μ mit der Koeffizientenmatrix $\boldsymbol{x}$ und der rechten Seite $\boldsymbol{A}\boldsymbol{x}$. Die Normalgleichung hierzu lautet

$$\boldsymbol{x}^H \boldsymbol{x}\mu = \boldsymbol{x}^H \boldsymbol{A}\boldsymbol{x}.$$

□

Potenzmethode

Auf den ersten Blick scheint es eine Schwierigkeit bei der inversen Iteration zu sein, dass die Koeffizientenmatrix $\lambda_{(m)}\boldsymbol{E} - \boldsymbol{A}$ des zu lösenden linearen Gleichungssystems bei Konvergenz von $\lambda_{(m)}$ gegen einen Eigenwert von $\boldsymbol{A}$ mehr und mehr singulär wird, die Kondition also wächst und damit der Fehler der Lösung des Gleichungssystems wächst.

Potenzmethode

Auf den ersten Blick scheint es eine Schwierigkeit bei der inversen Iteration zu sein, dass die Koeffizientenmatrix $\lambda_{(m)}\boldsymbol{E} - \boldsymbol{A}$ des zu lösenden linearen Gleichungssystems bei Konvergenz von $\lambda_{(m)}$ gegen einen Eigenwert von $\boldsymbol{A}$ mehr und mehr singulär wird, die Kondition also wächst und damit der Fehler der Lösung des Gleichungssystems wächst.

Man kann jedoch zeigen (vgl. Chatelin, pp. 217 ff.), dass (bei gut konditionierten Problemen) der Fehler, der bei der Lösung von $(\lambda_{(m)}\boldsymbol{E} - \boldsymbol{A})\boldsymbol{v}^{m+1} = \boldsymbol{u}^m$ gemacht wird, vornehmlich die Richtung des Eigenvektors zu λ, also die gewünschte Richtung hat.

QR-Algorithmus

Der QR-Algorithmus dient dazu, alle Eigenwerte einer Matrix $\boldsymbol{A} \in \mathbb{C}^{(n,n)}$ zu bestimmen.

QR-Algorithmus

Der QR-Algorithmus dient dazu, alle Eigenwerte einer Matrix $\boldsymbol{A} \in \mathbb{C}^{(n,n)}$ zu bestimmen.

Er wurde von Francis und Kublanovskaya unabhängig voneinander im Jahr 1961 eingeführt.

QR-Algorithmus

Der QR-Algorithmus dient dazu, alle Eigenwerte einer Matrix $\boldsymbol{A} \in \mathbb{C}^{(n,n)}$ zu bestimmen.

Er wurde von Francis und Kublanovskaya unabhängig voneinander im Jahr 1961 eingeführt.

Er wird heute als das beste Verfahren zur Lösung der vollständigen Eigenwertaufgabe für nichtsymmetrische Probleme angesehen.

QR-Algorithmus

Es sei $\boldsymbol{A}_0 := \boldsymbol{A}$. Die Grundform des QR-Algorithmus ist gegeben durch

Algorithmus 6.19 (QR-Algorithmus; Grundform):

```
for i = 0,1,... until convergence do
   Zerlege A_i = Q_i R_i;  (QR-Zerlegung)
   A_{i+1} = R_i Q_i;
end
```

QR-Algorithmus

Es sei $\boldsymbol{A}_0 := \boldsymbol{A}$. Die Grundform des QR-Algorithmus ist gegeben durch

Algorithmus 6.19 (QR-Algorithmus; Grundform):

```
for i = 0,1,... until convergence do
   Zerlege A_i = Q_i R_i; (QR-Zerlegung)
   A_{i+1} = R_i Q_i;
end
```

Die durch den QR-Algorithmus erzeugten Matrizen $\boldsymbol{A}_i$ sind einander ähnlich und damit ähnlich zu $\boldsymbol{A}$, denn es gilt

$$\boldsymbol{A}_{i+1} = \boldsymbol{R}_i \boldsymbol{Q}_i = \boldsymbol{Q}_i^H (\boldsymbol{Q}_i \boldsymbol{R}_i) \boldsymbol{Q}_i = \boldsymbol{Q}_i^H \boldsymbol{A}_i \boldsymbol{Q}_i.$$

QR-Algorithmus

Es sei $\boldsymbol{A}_0 := \boldsymbol{A}$. Die Grundform des QR-Algorithmus ist gegeben durch

Algorithmus 6.19 (QR-Algorithmus; Grundform):

```
for i = 0,1,... until convergence do
   Zerlege A_i = Q_i R_i;  (QR-Zerlegung)
   A_{i+1} = R_i Q_i;
end
```

Die durch den QR-Algorithmus erzeugten Matrizen $\boldsymbol{A}_i$ sind einander ähnlich und damit ähnlich zu $\boldsymbol{A}$, denn es gilt

$$\boldsymbol{A}_{i+1} = \boldsymbol{R}_i \boldsymbol{Q}_i = \boldsymbol{Q}_i^H (\boldsymbol{Q}_i \boldsymbol{R}_i) \boldsymbol{Q}_i = \boldsymbol{Q}_i^H \boldsymbol{A}_i \boldsymbol{Q}_i.$$

Die Konvergenz des QR-Algorithmus wird durch den folgenden Satz beschrieben.

QR-Algorithmus

Satz 6.20

Es seien die Eigenwerte von $\boldsymbol{A} \in \mathbb{C}^{(n,n)}$ dem Betrage nach voneinander verschieden und angeordnet gemäß

$$|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n| > 0. \tag{6.9}$$

Es sei $\boldsymbol{A} = \boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^{-1}$ und es besitze die Matrix $\boldsymbol{Y} := \boldsymbol{X}^{-1}$ eine LR-Zerlegung.

QR-Algorithmus

Satz 6.20

Es seien die Eigenwerte von $\boldsymbol{A} \in \mathbb{C}^{(n,n)}$ dem Betrage nach voneinander verschieden und angeordnet gemäß

$$|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n| > 0. \tag{6.9}$$

Es sei $\boldsymbol{A} = \boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^{-1}$ und es besitze die Matrix $\boldsymbol{Y} := \boldsymbol{X}^{-1}$ eine LR-Zerlegung.

Dann konvergiert der QR-Algorithmus im Wesentlichen, d.h., für die Elemente von $\boldsymbol{A}_i := (a_{jk}^{(i)})_{j,k}$ gilt

$$\lim_{i\to\infty} a_{jk}^{(i)} = 0 \quad \text{für} \quad j > k$$

$$\lim_{i\to\infty} a_{kk}^{(i)} = \lambda_k \quad \text{für} \quad k = 1, \ldots, n.$$

QR-Algorithmus

Bemerkung 6.21: Am Ende sind die Eigenwerte von A in A_i der Größe der Beträge nach geordnet. □

QR-Algorithmus

Bemerkung 6.21: Am Ende sind die Eigenwerte von $\boldsymbol{A}$ in $\boldsymbol{A}_i$ der Größe der Beträge nach geordnet. □

Beispiel 6.22: Die Matrix

$$\boldsymbol{A} = \begin{pmatrix} 1 & -1 & -1 \\ 4 & 6 & 3 \\ -4 & -4 & -1 \end{pmatrix}$$

besitzt die Eigenwerte $\lambda_1 = 3$, $\lambda_2 = 2$ und $\lambda_3 = 1$ und für die Matrix der Eigenvektoren gilt (bis auf die Normierung)

$$\boldsymbol{X} = \begin{pmatrix} 0 & -1 & -1 \\ 1 & 1 & 2 \\ -1 & 0 & -2 \end{pmatrix} \quad \text{und} \quad \boldsymbol{X}^{-1} = \begin{pmatrix} 2 & 2 & 1 \\ 0 & 1 & 1 \\ -1 & -1 & -1 \end{pmatrix}.$$

QR-Algorithmus

Die Voraussetzungen von Satz 6.20 sind also erfüllt. Man erhält

$$\boldsymbol{A}_{10} \approx \begin{pmatrix} 3.000034 & -0.577363 & 8.981447 \\ 9.78 \cdot 10^{-6} & 1.999995 & 1.4142593 \\ -6.91 \cdot 10^{-6} & 3.99 \cdot 10^{-6} & 0.999972 \end{pmatrix}. \qquad \square$$

QR-Algorithmus

Die Voraussetzungen von Satz 6.20 sind also erfüllt. Man erhält

$$\boldsymbol{A}_{10} \approx \begin{pmatrix} 3.000034 & -0.577363 & 8.981447 \\ 9.78 \cdot 10^{-6} & 1.999995 & 1.4142593 \\ -6.91 \cdot 10^{-6} & 3.99 \cdot 10^{-6} & 0.999972 \end{pmatrix}. \qquad \square$$

Bemerkung 6.23: Wegen (6.9) ist $\boldsymbol{A}$ diagonalisierbar, d.h. $\boldsymbol{X}$ und damit $\boldsymbol{Y}$ existieren. Die einzige Forderung in Satz 6.20 an die Matrix $\boldsymbol{X}$ ist also, dass $\boldsymbol{Y} := \boldsymbol{X}^{-1}$ eine LR-Zerlegung besitzt. Verzichtet man hierauf, so erhält man

$$\lim_{i \to \infty} a_{kk}^{(i)} = \lambda_{\pi(k)}$$

für eine von A abhängige Permutation π von $\{1, \ldots, n\}$ (vgl. mit dem Buch von Wilkinson, p. 519 ff.). $\square$

QR-Algorithmus

Beispiel 6.24: Für die Matrix

$$\boldsymbol{A} = \begin{pmatrix} 1 & 0 & 1 \\ 2 & 3 & -1 \\ -2 & -2 & 2 \end{pmatrix}$$

mit den Eigenwerten $\lambda_1 = 3$, $\lambda_2 = 2$, $\lambda_3 = 1$ und den Eigenvektoren

$$\boldsymbol{X} = \begin{pmatrix} -1 & -1 & -1 \\ 2 & 1 & 1 \\ -2 & -1 & 0 \end{pmatrix} \quad \text{und} \quad \boldsymbol{X}^{-1} = \begin{pmatrix} 1 & 1 & 0 \\ -2 & -2 & -1 \\ 0 & 1 & 1 \end{pmatrix}$$

ist die Voraussetzung von Satz 6.20 nicht erfüllt. Die Matrix $\boldsymbol{X}^{-1}(1:2, 1:2)$ ist singulär.

QR-Algorithmus

Der QR-Algorithmus liefert nach 30 Schritten

$$\boldsymbol{A}_{30} \approx \begin{pmatrix} 3.0000058 & -2.0000014 & 2.9999975 \\ 1.16 \cdot 10^{-6} & 0.9999989 & -0.9999978 \\ -1.16 \cdot 10^{-6} & 6.69 \cdot 10^{-5} & 1.9999953 \end{pmatrix}.$$

Die Diagonalelemente scheinen also gegen die Eigenwerte zu konvergieren, wobei sie allerdings nicht nach der Betragsgöße geordnet sind.

QR-Algorithmus

Der QR-Algorithmus liefert nach 30 Schritten

$$\boldsymbol{A}_{30} \approx \begin{pmatrix} 3.0000058 & -2.0000014 & 2.9999975 \\ 1.16 \cdot 10^{-6} & 0.9999989 & -0.9999978 \\ -1.16 \cdot 10^{-6} & 6.69 \cdot 10^{-5} & 1.9999953 \end{pmatrix}.$$

Die Diagonalelemente scheinen also gegen die Eigenwerte zu konvergieren, wobei sie allerdings nicht nach der Betragsgöße geordnet sind.

Diese Konvergenz würde auch bei exakter Rechnung eintreten (vgl. Bemerkung 6.23). Iteriert man weiter, so erhält man (durch Rundungsfehler)

$$\boldsymbol{A}_{70} \approx \begin{pmatrix} 3 + 5.2 \cdot 10^{-13} & -3.5355303 & 0.7071247 \\ 1.48 \cdot 10^{-13} & 1.9999949 & -1.0000051 \\ -7.52 \cdot 10^{-19} & -5.07 \cdot 10^{-6} & 1.0000051 \end{pmatrix},$$

also doch noch Konvergenz gegen eine Dreiecksmatrix mit der Betragsgröße nach geordneten Eigenwerten. □

QR-Algorithmus

Bemerkung 6.25: Ist $\boldsymbol{A} \in \mathbb{R}^{(n,n)}$, so gilt $\boldsymbol{Q}_k \in \mathbb{R}^{(n,n)}$, $\boldsymbol{R}_k \in \mathbb{R}^{(n,n)}$ für alle k, und der ganze Algorithmus verläuft in $\mathbb{R}$.

QR-Algorithmus

Bemerkung 6.25: Ist $\boldsymbol{A} \in \mathbb{R}^{(n,n)}$, so gilt $\boldsymbol{Q}_k \in \mathbb{R}^{(n,n)}$, $\boldsymbol{R}_k \in \mathbb{R}^{(n,n)}$ für alle k, und der ganze Algorithmus verläuft in $\mathbb{R}$.

Sind alle Eigenwerte betragsmäßig verschieden, so sind sie alle reell, und der QR-Algorithmus konvergiert. Besitzt $\boldsymbol{A}$ jedoch komplexe Eigenwerte, so kann die Grundform nicht konvergieren.

QR-Algorithmus

Bemerkung 6.25: Ist $\boldsymbol{A} \in \mathbb{R}^{(n,n)}$, so gilt $\boldsymbol{Q}_k \in \mathbb{R}^{(n,n)}$, $\boldsymbol{R}_k \in \mathbb{R}^{(n,n)}$ für alle k, und der ganze Algorithmus verläuft in $\mathbb{R}$.

Sind alle Eigenwerte betragsmäßig verschieden, so sind sie alle reell, und der QR-Algorithmus konvergiert. Besitzt $\boldsymbol{A}$ jedoch komplexe Eigenwerte, so kann die Grundform nicht konvergieren.

Ist $\boldsymbol{A} \in \mathbb{C}^{(n,n)}$ Hermitesch, so sind alle $\boldsymbol{A}_m$ Hermitesch und $\boldsymbol{A}_m$ konvergiert unter den Voraussetzungen von Satz 6.20 gegen eine Diagonalmatrix. □

QR-Algorithmus

Bemerkung 6.25: Ist $\boldsymbol{A} \in \mathbb{R}^{(n,n)}$, so gilt $\boldsymbol{Q}_k \in \mathbb{R}^{(n,n)}$, $\boldsymbol{R}_k \in \mathbb{R}^{(n,n)}$ für alle k, und der ganze Algorithmus verläuft in $\mathbb{R}$.

Sind alle Eigenwerte betragsmäßig verschieden, so sind sie alle reell, und der QR-Algorithmus konvergiert. Besitzt $\boldsymbol{A}$ jedoch komplexe Eigenwerte, so kann die Grundform nicht konvergieren.

Ist $\boldsymbol{A} \in \mathbb{C}^{(n,n)}$ Hermitesch, so sind alle $\boldsymbol{A}_m$ Hermitesch und $\boldsymbol{A}_m$ konvergiert unter den Voraussetzungen von Satz 6.20 gegen eine Diagonalmatrix. □

Ein Nachteil der Grundform des QR-Algorithmus ist, dass sie sehr aufwändig ist. Ist $\boldsymbol{A}$ voll besetzt, so benötigt man in jedem Schritt $O(n^3)$ Operationen zur Bestimmung der QR-Zerlegung.

QR-Algorithmus

Wir beschleunigen nun den QR-Algorithmus auf zwei Weisen.

QR-Algorithmus

Wir beschleunigen nun den QR-Algorithmus auf zwei Weisen.

Erstens erhöhen wir mit Hilfe von Shifts die Konvergenzgeschwindigkeit und zweitens zeigen wir, dass die Hessenberg-Gestalt einer Matrix im Laufe eines QR-Schritts erhalten bleibt.

QR-Algorithmus

Wir beschleunigen nun den QR-Algorithmus auf zwei Weisen.

Erstens erhöhen wir mit Hilfe von Shifts die Konvergenzgeschwindigkeit und zweitens zeigen wir, dass die Hessenberg-Gestalt einer Matrix im Laufe eines QR-Schritts erhalten bleibt.

Da die QR-Zerlegung einer Hessenberg-Matrix nur $\mathcal{O}(n^2)$ Operationen benötigt, werden wir Matrizen vor Anwendung des QR-Algorithmus zunächst unitär auf Hessenberg-Gestalt transformieren.

QR-Algorithmus

Wir beschleunigen nun den QR-Algorithmus auf zwei Weisen.

Erstens erhöhen wir mit Hilfe von Shifts die Konvergenzgeschwindigkeit und zweitens zeigen wir, dass die Hessenberg-Gestalt einer Matrix im Laufe eines QR-Schritts erhalten bleibt.

Da die QR-Zerlegung einer Hessenberg-Matrix nur $O(n^2)$ Operationen benötigt, werden wir Matrizen vor Anwendung des QR-Algorithmus zunächst unitär auf Hessenberg-Gestalt transformieren.

Doch zuerst präsentieren wir die angesprochene Verbesserung durch Einbau von Shifts in den QR-Algorithmus.

QR-Algorithmus

Sei $A_1 := A$. Wir betrachten dann die geshiftete Version des QR-Algorithmus:

Algorithmus 6.26 (QR-Algorithmus mit Shifts):

```
for i = 1,2,... until convergence do
   Wähle einen geeigneten Shift κ_i;
   Zerlege A_i − κ_i E = Q_i R_i;
   A_{i+1} = R_i Q_i + κ_i E;
end
```

QR-Algorithmus

Für die hierdurch erzeugten Matrizen gilt

$$\boldsymbol{R}_i = \boldsymbol{Q}_i^H(\boldsymbol{A}_i - \kappa_i \boldsymbol{E}),$$

und

$$\boldsymbol{A}_{i+1} = \boldsymbol{Q}_i^H(\boldsymbol{A}_i - \kappa_i \boldsymbol{E})\boldsymbol{Q}_i + \kappa_i \boldsymbol{E} = \boldsymbol{Q}_i^H \boldsymbol{A}_i \boldsymbol{Q}_i. \tag{6.10}$$

QR-Algorithmus

Für die hierdurch erzeugten Matrizen gilt

$$\boldsymbol{R}_i = \boldsymbol{Q}_i^H(\boldsymbol{A}_i - \kappa_i \boldsymbol{E}),$$

und

$$\boldsymbol{A}_{i+1} = \boldsymbol{Q}_i^H(\boldsymbol{A}_i - \kappa_i \boldsymbol{E})\boldsymbol{Q}_i + \kappa_i \boldsymbol{E} = \boldsymbol{Q}_i^H \boldsymbol{A}_i \boldsymbol{Q}_i. \tag{6.10}$$

Die $\boldsymbol{A}_i$ sind also alle der Matrix $\boldsymbol{A}$ ähnlich und haben dieselben Eigenwerte.

QR-Algorithmus

Für die hierdurch erzeugten Matrizen gilt

$$\boldsymbol{R}_i = \boldsymbol{Q}_i^H(\boldsymbol{A}_i - \kappa_i \boldsymbol{E}),$$

und

$$\boldsymbol{A}_{i+1} = \boldsymbol{Q}_i^H(\boldsymbol{A}_i - \kappa_i \boldsymbol{E})\boldsymbol{Q}_i + \kappa_i \boldsymbol{E} = \boldsymbol{Q}_i^H \boldsymbol{A}_i \boldsymbol{Q}_i. \tag{6.10}$$

Die $\boldsymbol{A}_i$ sind also alle der Matrix $\boldsymbol{A}$ ähnlich und haben dieselben Eigenwerte.

Um die Parameter κ_i geeignet zu wählen, überlegen wir uns einen Zusammenhang von Algorithmus 6.26 und der inversen Iteration.

QR-Algorithmus

Satz 6.27

Es seien orthogonale Matrizen $\boldsymbol{U}_i$ und obere Dreicksmatrizen $\boldsymbol{S}_i$ definiert als

$$\boldsymbol{U}_i := \boldsymbol{Q}_1\boldsymbol{Q}_2\cdots\boldsymbol{Q}_i, \qquad \boldsymbol{S}_i := \boldsymbol{R}_i\boldsymbol{R}_{i-1}\cdots\boldsymbol{R}_1.$$

Dann gilt

$$\boldsymbol{U}_i\boldsymbol{S}_i = (\boldsymbol{A}-\kappa_i\boldsymbol{E})(\boldsymbol{A}-\kappa_{i-1}\boldsymbol{E})\cdots(\boldsymbol{A}-\kappa_1\boldsymbol{E}) = p(\boldsymbol{A}). \tag{6.11}$$

QR-Algorithmus

Satz 6.27

Es seien orthogonale Matrizen $\boldsymbol{U}_i$ und obere Dreicksmatrizen $\boldsymbol{S}_i$ definiert als

$$\boldsymbol{U}_i := \boldsymbol{Q}_1\boldsymbol{Q}_2\cdots\boldsymbol{Q}_i, \qquad \boldsymbol{S}_i := \boldsymbol{R}_i\boldsymbol{R}_{i-1}\cdots\boldsymbol{R}_1.$$

Dann gilt

$$\boldsymbol{U}_i\boldsymbol{S}_i = (\boldsymbol{A}-\kappa_i\boldsymbol{E})(\boldsymbol{A}-\kappa_{i-1}\boldsymbol{E})\cdots(\boldsymbol{A}-\kappa_1\boldsymbol{E}) = p(\boldsymbol{A}). \tag{6.11}$$

Beweis.

Zunächst folgt aus (6.10) *durch Induktion*

$$\boldsymbol{A}_{i+1} = \boldsymbol{U}_i^H\boldsymbol{A}\boldsymbol{U}_i. \tag{6.12}$$

Für $i = 1$ *besagt* (6.11) $\boldsymbol{U}_1\boldsymbol{S}_1 = \boldsymbol{Q}_1\boldsymbol{R}_1 = \boldsymbol{A}-\kappa_1\boldsymbol{E}$ *und dies ist gerade die Zerlegung im ersten Schritt von Algorithmus 6.26.*

QR-Algorithmus

Beweis.

Sei (6.11) bereits für $i-1$ bewiesen. Dann folgt aus der Definition von $\boldsymbol{A}_{i+1}$

$$\begin{aligned}\boldsymbol{R}_i &= (\boldsymbol{A}_{i+1} - \kappa_i \boldsymbol{E})\boldsymbol{Q}_i^H \\ &= \boldsymbol{U}_i^H(\boldsymbol{A} - \kappa_i \boldsymbol{E})\boldsymbol{U}_i\boldsymbol{Q}_i^H \\ &= \boldsymbol{U}_i^H(\boldsymbol{A} - \kappa_i \boldsymbol{E})\boldsymbol{U}_{i-1}\end{aligned}$$

und hieraus erhält man durch Rechtsmultiplikation mit $\boldsymbol{S}_{i-1}$ und Linksmultiplikation mit $\boldsymbol{U}_i$

$$\begin{aligned}\boldsymbol{U}_i\boldsymbol{S}_i &= (\boldsymbol{A} - \kappa_i \boldsymbol{E})\boldsymbol{U}_{i-1}\boldsymbol{S}_{i-1} \\ &= (\boldsymbol{A} - \kappa_i \boldsymbol{E})(\boldsymbol{A} - \kappa_{i-1} \boldsymbol{E}) \cdots (\boldsymbol{A} - \kappa_1 \boldsymbol{E})\end{aligned}$$

nach Induktionsannahme. □

QR-Algorithmus

Aus der Darstellung (6.11) folgt sofort

$$(\boldsymbol{A}^H - \overline{\kappa_i}\boldsymbol{E})^{-1} \cdots (\boldsymbol{A}^H - \overline{\kappa_1}\boldsymbol{E})^{-1}\boldsymbol{e}^n = \boldsymbol{U}_i(\boldsymbol{S}_i^H)^{-1}\boldsymbol{e}^n.$$

QR-Algorithmus

Aus der Darstellung (6.11) folgt sofort

$$(\boldsymbol{A}^H - \overline{\kappa_i}\boldsymbol{E})^{-1} \cdots (\boldsymbol{A}^H - \overline{\kappa_1}\boldsymbol{E})^{-1}\boldsymbol{e}^n = \boldsymbol{U}_i(\boldsymbol{S}_i^H)^{-1}\boldsymbol{e}^n.$$

Da mit $\boldsymbol{S}_i^H$ auch $(\boldsymbol{S}_i^H)^{-1}$ eine untere Dreiecksmatrix ist, gilt

$$\boldsymbol{U}_i(\boldsymbol{S}_i^H)^{-1}\boldsymbol{e}^n = \sigma_i \boldsymbol{U}_i \boldsymbol{e}^n$$

für ein $\sigma_i \in \mathbb{C}$.

QR-Algorithmus

Aus der Darstellung (6.11) folgt sofort

$$(\boldsymbol{A}^H - \overline{\kappa_i}\boldsymbol{E})^{-1} \cdots (\boldsymbol{A}^H - \overline{\kappa_1}\boldsymbol{E})^{-1}\boldsymbol{e}^n = \boldsymbol{U}_i(\boldsymbol{S}_i^H)^{-1}\boldsymbol{e}^n.$$

Da mit $\boldsymbol{S}_i^H$ auch $(\boldsymbol{S}_i^H)^{-1}$ eine untere Dreiecksmatrix ist, gilt

$$\boldsymbol{U}_i(\boldsymbol{S}_i^H)^{-1}\boldsymbol{e}^n = \sigma_i \boldsymbol{U}_i \boldsymbol{e}^n$$

für ein $\sigma_i \in \mathbb{C}$.

Damit ist die letzte Spalte von $\boldsymbol{U}_i$ eine Näherung für einen Eigenvektor von $\boldsymbol{A}^H$, die man ausgehend von $\boldsymbol{e}^n$ mit der inversen Iteration mit den Shifts $\overline{\kappa_1}, \ldots, \overline{\kappa_i}$ erhält.

QR-Algorithmus

Man kann also schnelle Konvergenz erwarten, wenn man $\overline{\kappa_i}$ als Rayleigh-Quotienten von $\boldsymbol{A}^H$ an der Stelle $\boldsymbol{u}_n^{i-1}$, der letzten Spalte von $\boldsymbol{U}_{i-1}$, wählt.

QR-Algorithmus

Man kann also schnelle Konvergenz erwarten, wenn man $\overline{\kappa_i}$ als Rayleigh-Quotienten von $\boldsymbol{A}^H$ an der Stelle $\boldsymbol{u}_n^{i-1}$, der letzten Spalte von $\boldsymbol{U}_{i-1}$, wählt.

Es ist

$$\overline{\kappa_i} = (\boldsymbol{u}_n^{i-1})^H \boldsymbol{A}^H \boldsymbol{u}_n^{i-1} = \overline{(\boldsymbol{u}_n^{i-1})^H \boldsymbol{A} \boldsymbol{u}_n^{i-1}} \overset{(6.12)}{=} \overline{(\boldsymbol{e}^n)^T \boldsymbol{A}_i \boldsymbol{e}^n} =: \overline{a_{nn}^{(i)}},$$

QR-Algorithmus

Man kann also schnelle Konvergenz erwarten, wenn man $\overline{\kappa_i}$ als Rayleigh-Quotienten von $\boldsymbol{A}^H$ an der Stelle $\boldsymbol{u}_n^{i-1}$, der letzten Spalte von $\boldsymbol{U}_{i-1}$, wählt.

Es ist

$$\overline{\kappa_i} = (\boldsymbol{u}_n^{i-1})^H \boldsymbol{A}^H \boldsymbol{u}_n^{i-1} = \overline{(\boldsymbol{u}_n^{i-1})^H \boldsymbol{A} \boldsymbol{u}_n^{i-1}} \overset{(6.12)}{=} \overline{(\boldsymbol{e}^n)^T \boldsymbol{A}_i \boldsymbol{e}^n} =: \overline{a_{nn}^{(i)}},$$

d.h.

$$\kappa_i = a_{nn}^{(i)}.$$

QR-Algorithmus

Man kann also schnelle Konvergenz erwarten, wenn man $\overline{\kappa_i}$ als Rayleigh-Quotienten von $\boldsymbol{A}^H$ an der Stelle $\boldsymbol{u}_n^{i-1}$, der letzten Spalte von $\boldsymbol{U}_{i-1}$, wählt.

Es ist

$$\overline{\kappa_i} = (\boldsymbol{u}_n^{i-1})^H \boldsymbol{A}^H \boldsymbol{u}_n^{i-1} = \overline{(\boldsymbol{u}_n^{i-1})^H \boldsymbol{A} \boldsymbol{u}_n^{i-1}} \overset{(6.12)}{=} \overline{(\boldsymbol{e}^n)^T \boldsymbol{A}_i \boldsymbol{e}^n} =: \overline{a_{nn}^{(i)}},$$

d.h.

$$\kappa_i = a_{nn}^{(i)}.$$

Dieser Shift heißt Rayleigh-Quotienten-Shift.

QR-Algorithmus

Eine weitere Verbesserung des Verfahrens (Verkleinerung des Aufwandes) erhält man, wenn man $\boldsymbol{A}$ zunächst unitär auf obere Hessenberg-Gestalt transformiert.

QR-Algorithmus

Eine weitere Verbesserung des Verfahrens (Verkleinerung des Aufwandes) erhält man, wenn man $\boldsymbol{A}$ zunächst unitär auf obere Hessenberg-Gestalt transformiert.

Dabei sagt man, dass eine Matrix $\boldsymbol{A}$ Hessenberg-Gestalt hat, wenn

$$a_{ij} = 0 \quad \text{für alle} \quad i > j + 1.$$

QR-Algorithmus

Eine weitere Verbesserung des Verfahrens (Verkleinerung des Aufwandes) erhält man, wenn man $\boldsymbol{A}$ zunächst unitär auf obere Hessenberg-Gestalt transformiert.

Dabei sagt man, dass eine Matrix $\boldsymbol{A}$ Hessenberg-Gestalt hat, wenn

$$a_{ij} = 0 \quad \text{für alle} \quad i > j + 1.$$

Diese Gestalt bleibt nämlich, wie wir noch sehen werden, während des gesamten QR-Algorithmus erhalten.

QR-Algorithmus

Wir bestimmen daher eine unitäre Matrix $\boldsymbol{U}$, so dass $\boldsymbol{U}^H \boldsymbol{A} \boldsymbol{U}$ obere Hessenberg-Gestalt hat.

QR-Algorithmus

Wir bestimmen daher eine unitäre Matrix $\boldsymbol{U}$, so dass $\boldsymbol{U}^H \boldsymbol{A} \boldsymbol{U}$ obere Hessenberg-Gestalt hat.

Diese Matrix $\boldsymbol{U}$ können wir als Produkt von $n-2$ Householder-Spiegelungen der Gestalt $\boldsymbol{Q} = \boldsymbol{E} - 2\boldsymbol{u}\boldsymbol{u}^H$ aufbauen. Sei $\boldsymbol{A}$ wie folgt partioniert:

$$\boldsymbol{A} = \begin{pmatrix} a_{11} & \boldsymbol{c}^T \\ \boldsymbol{b} & \boldsymbol{B} \end{pmatrix} \quad \text{mit} \quad \boldsymbol{b} \neq \boldsymbol{o}.$$

QR-Algorithmus

Wir bestimmen daher eine unitäre Matrix $\boldsymbol{U}$, so dass $\boldsymbol{U}^H \boldsymbol{A} \boldsymbol{U}$ obere Hessenberg-Gestalt hat.

Diese Matrix $\boldsymbol{U}$ können wir als Produkt von $n-2$ Householder-Spiegelungen der Gestalt $\boldsymbol{Q} = \boldsymbol{E} - 2\boldsymbol{u}\boldsymbol{u}^H$ aufbauen. Sei $\boldsymbol{A}$ wie folgt partioniert:

$$\boldsymbol{A} = \begin{pmatrix} a_{11} & \boldsymbol{c}^T \\ \boldsymbol{b} & \boldsymbol{B} \end{pmatrix} \quad \text{mit} \quad \boldsymbol{b} \neq \boldsymbol{o}.$$

Wir bestimmen dann $\boldsymbol{w} \in \mathbb{C}^{n-1}$ mit $\|\boldsymbol{w}\|_2 = 1$ und

$$\boldsymbol{Q}_1 \boldsymbol{b} := (\boldsymbol{E}_{n-1} - 2\boldsymbol{w}\boldsymbol{w}^H)\boldsymbol{b} = k\boldsymbol{e}^1$$

und setzen hiermit

$$\boldsymbol{P}_1 = \begin{pmatrix} 1 & \boldsymbol{o}^T \\ \boldsymbol{o} & \boldsymbol{Q}_1 \end{pmatrix}.$$

QR-Algorithmus

Dann gilt

$$\boldsymbol{A}_1 := \boldsymbol{P}_1 \boldsymbol{A} \boldsymbol{P}_1 = \begin{pmatrix} a_{11} & \boldsymbol{c}^T \boldsymbol{Q}_1 \\ k & \\ 0 & \\ \vdots & \boldsymbol{Q}_1 \boldsymbol{B} \boldsymbol{Q}_1 \\ 0 & \end{pmatrix}.$$

QR-Algorithmus

Dann gilt

$$\boldsymbol{A}_1 := \boldsymbol{P}_1 \boldsymbol{A} \boldsymbol{P}_1 = \begin{pmatrix} a_{11} & \boldsymbol{c}^T \boldsymbol{Q}_1 \\ k & \\ 0 & \\ \vdots & \boldsymbol{Q}_1 \boldsymbol{B} \boldsymbol{Q}_1 \\ 0 & \end{pmatrix}.$$

Damit hat die erste Spalte schon die Gestalt, die wir in einer Hessenberg-Matrix benötigen.

QR-Algorithmus

Dann gilt

$$\boldsymbol{A}_1 := \boldsymbol{P}_1 \boldsymbol{A} \boldsymbol{P}_1 = \begin{pmatrix} a_{11} & \boldsymbol{c}^T \boldsymbol{Q}_1 \\ k & \\ 0 & \\ \vdots & \boldsymbol{Q}_1 \boldsymbol{B} \boldsymbol{Q}_1 \\ 0 & \end{pmatrix}.$$

Damit hat die erste Spalte schon die Gestalt, die wir in einer Hessenberg-Matrix benötigen.

Die Spalten $2, \ldots, n-2$ können wir dann genauso behandeln (vgl. die Vorgehensweise bei der QR-Zerlegung einer Matrix mittels Householder-Spiegelungen).

QR-Algorithmus

Sei nun $\boldsymbol{A} =: \boldsymbol{A}_1$ eine obere Hessenberg-Matrix und $\kappa_1 \in \mathbb{C}$ ein Shift-Parameter, z. B. der Rayleigh-Quotienten-Shift $\kappa_1 := a_{nn}^{(1)}$.

QR-Algorithmus

Sei nun $\boldsymbol{A} =: \boldsymbol{A}_1$ eine obere Hessenberg-Matrix und $\kappa_1 \in \mathbb{C}$ ein Shift-Parameter, z. B. der Rayleigh-Quotienten-Shift $\kappa_1 := a_{nn}^{(1)}$.

Wir multiplizieren dann für $i = 1, \ldots, n-1$ die Matrix

$$\boldsymbol{U}_{i-1,i} \ldots \boldsymbol{U}_{12}(\boldsymbol{A} - \kappa_1 \boldsymbol{E})$$

mit einer ebenen Drehung $\boldsymbol{U}_{i,i+1}$, so dass das Element an der Stelle $(i+1, i)$ annulliert wird.

QR-Algorithmus

Sei nun $\boldsymbol{A} =: \boldsymbol{A}_1$ eine obere Hessenberg-Matrix und $\kappa_1 \in \mathbb{C}$ ein Shift-Parameter, z. B. der Rayleigh-Quotienten-Shift $\kappa_1 := a_{nn}^{(1)}$.

Wir multiplizieren dann für $i = 1, \ldots, n-1$ die Matrix

$$\boldsymbol{U}_{i-1,i} \ldots \boldsymbol{U}_{12}(\boldsymbol{A} - \kappa_1 \boldsymbol{E})$$

mit einer ebenen Drehung $\boldsymbol{U}_{i,i+1}$, so dass das Element an der Stelle $(i+1, i)$ annulliert wird.

Dann besitzt

$$\boldsymbol{R}_1 := \boldsymbol{U}_{n-1,n} \ldots \boldsymbol{U}_{12}(\boldsymbol{A} - \kappa_1 \boldsymbol{E})$$

obere Dreiecksgestalt und

$$\boldsymbol{Q}_1 := \boldsymbol{U}_{12}^H \ldots \boldsymbol{U}_{n-1,n}^H$$

ist der unitäre Anteil in der QR-Zerlegung von $\boldsymbol{A} - \kappa_1 \boldsymbol{E}$ in Algorithmus 6.26.

QR-Algorithmus

Die neue Iterierte $\boldsymbol{A}_2$ erhält man dann gemäß

$$\boldsymbol{A}_2 = \boldsymbol{R}_1 \boldsymbol{U}_{12}^H \dots \boldsymbol{U}_{n-1,n}^H + \kappa_1 \boldsymbol{E}.$$

QR-Algorithmus

Die neue Iterierte $\boldsymbol{A}_2$ erhält man dann gemäß

$$\boldsymbol{A}_2 = \boldsymbol{R}_1 \boldsymbol{U}_{12}^H \dots \boldsymbol{U}_{n-1,n}^H + \kappa_1 \boldsymbol{E}.$$

Da die Multiplikation von rechts mit $\boldsymbol{U}_{i,i+1}^H$ nur die i-te und $(i+1)$-te Spalte verändert, ist klar, dass $\boldsymbol{A}_2$ wieder obere Hessenberg-Matrix ist.

QR-Algorithmus

Die neue Iterierte $\boldsymbol{A}_2$ erhält man dann gemäß

$$\boldsymbol{A}_2 = \boldsymbol{R}_1 \boldsymbol{U}_{12}^H \dots \boldsymbol{U}_{n-1,n}^H + \kappa_1 \boldsymbol{E}.$$

Da die Multiplikation von rechts mit $\boldsymbol{U}_{i,i+1}^H$ nur die i-te und $(i+1)$-te Spalte verändert, ist klar, dass $\boldsymbol{A}_2$ wieder obere Hessenberg-Matrix ist.

Die obere Hessenberg-Gestalt der Matrizen bleibt also im Verlaufe des QR-Algorithmus erhalten.

QR-Algorithmus

Die neue Iterierte $\boldsymbol{A}_2$ erhält man dann gemäß

$$\boldsymbol{A}_2 = \boldsymbol{R}_1 \boldsymbol{U}_{12}^H \dots \boldsymbol{U}_{n-1,n}^H + \kappa_1 \boldsymbol{E}.$$

Da die Multiplikation von rechts mit $\boldsymbol{U}_{i,i+1}^H$ nur die i-te und $(i+1)$-te Spalte verändert, ist klar, dass $\boldsymbol{A}_2$ wieder obere Hessenberg-Matrix ist.

Die obere Hessenberg-Gestalt der Matrizen bleibt also im Verlaufe des QR-Algorithmus erhalten.

Da ein QR-Schritt für eine Hessenberg-Matrix nur $O(n^2)$ Operationen erfordert, lohnt sich diese vorbereitende Transformation von $\boldsymbol{A}$.

QR-Algorithmus

Ist $\boldsymbol{A}$ Hermitesch, so sind alle $\boldsymbol{A}_i$ Hermitesch (vgl. (6.12)).

QR-Algorithmus

Ist $\boldsymbol{A}$ Hermitesch, so sind alle $\boldsymbol{A}_i$ Hermitesch (vgl. (6.12)).

Wenn man $\boldsymbol{A}$ auf Hessenberggestalt transformiert, so sieht man anhand der Symmetrie der Transformation,

$$\boldsymbol{H}^H = (\boldsymbol{U}^H \boldsymbol{A} \boldsymbol{U})^H = \boldsymbol{U}^H \boldsymbol{A}^H \boldsymbol{U} \overset{(\boldsymbol{A}^H = \boldsymbol{A})}{=} \boldsymbol{U}^H \boldsymbol{A} \boldsymbol{U} = \boldsymbol{H},$$

daß dann auch $\boldsymbol{H}$ Hermitesch ist, also sogar Tridiagonalgestalt hat.

QR-Algorithmus

Ist $\boldsymbol{A}$ Hermitesch, so sind alle $\boldsymbol{A}_i$ Hermitesch (vgl. (6.12)).

Wenn man $\boldsymbol{A}$ auf Hessenberggestalt transformiert, so sieht man anhand der Symmetrie der Transformation,

$$\boldsymbol{H}^H = (\boldsymbol{U}^H \boldsymbol{A} \boldsymbol{U})^H = \boldsymbol{U}^H \boldsymbol{A}^H \boldsymbol{U} \overset{(\boldsymbol{A}^H = \boldsymbol{A})}{=} \boldsymbol{U}^H \boldsymbol{A} \boldsymbol{U} = \boldsymbol{H},$$

daß dann auch $\boldsymbol{H}$ Hermitesch ist, also sogar Tridiagonalgestalt hat.

Transformiert man also $\boldsymbol{A}$ zunächst auf Tridiagonalgestalt, so bleiben alle $\boldsymbol{A}_m$ wie eben tridiagonal, und man benötigt sogar in jedem QR-Schritt nur $O(n)$ Operationen.

QR-Algorithmus

Wendet man den QR-Algorithmus mit Shift $\kappa = a_{nn}$ wiederholt an, so wird $a^{(i)}_{n,n-1}$ rasch (quadratisch; vgl. Lemma 6.18) gegen 0 gehen. Man erhält eine Matrix der Gestalt

$$\boldsymbol{A}_i = \left(\begin{array}{cccccc} + & + & \dots & + & + & \star \\ + & + & \dots & + & + & \star \\ 0 & + & \dots & + & + & \star \\ \vdots & \vdots & \vdots & \vdots & \vdots & \\ 0 & 0 & \dots & + & + & \star \\ \hline 0 & 0 & \dots & 0 & \approx 0 & \star \end{array}\right) = \boldsymbol{U}_i^H \boldsymbol{A} \boldsymbol{U}_i.$$

QR-Algorithmus

Wendet man den QR-Algorithmus mit Shift $\kappa = a_{nn}$ wiederholt an, so wird $a^{(i)}_{n,n-1}$ rasch (quadratisch; vgl. Lemma 6.18) gegen 0 gehen. Man erhält eine Matrix der Gestalt

$$\boldsymbol{A}_i = \left(\begin{array}{cccccc} + & + & \dots & + & + & \star \\ + & + & \dots & + & + & \star \\ 0 & + & \dots & + & + & \star \\ \vdots & \vdots & \vdots & \vdots & \vdots & \\ 0 & 0 & \dots & + & + & \star \\ \hline 0 & 0 & \dots & 0 & \approx 0 & \star \end{array}\right) = \boldsymbol{U}_i^H \boldsymbol{A} \boldsymbol{U}_i.$$

Der Wert $a^{(i)}_{nn}$ ist demnach eine Näherung für einen Eigenwert von $\boldsymbol{A}$ (nicht notwendig wie in Satz 6.20 für den betragskleinsten Eigenwert, da die Shifts diese Eigenschaft zerstören) und die Eigenwerte der führenden $(n-1, n-1)$ Hauptuntermatrix (oben $+$) von $\boldsymbol{A}_i$ sind Näherungen für die übrigen Eigenwerte von $\boldsymbol{A}$.

QR-Algorithmus

Man kann also mit einer verkleinerten Matrix den Algorithmus fortsetzen.

QR-Algorithmus

Man kann also mit einer verkleinerten Matrix den Algorithmus fortsetzen.

Darüber hinaus gehen auch die übrigen Subdiagonalelemente von $\boldsymbol{A}_i$ gegen 0 (vgl. Satz 6.20). Ist eines dieser Elemente klein genug, etwa

$$a_{k+1,k}^{(i)} \approx 0,$$

QR-Algorithmus

Man kann also mit einer verkleinerten Matrix den Algorithmus fortsetzen.

Darüber hinaus gehen auch die übrigen Subdiagonalelemente von $\boldsymbol{A}_i$ gegen 0 (vgl. Satz 6.20). Ist eines dieser Elemente klein genug, etwa

$$a^{(i)}_{k+1,k} \approx 0,$$

so kann man offenbar zwei kleinere Hessenberg-Matrizen

$$\begin{pmatrix} a^{(i)}_{11} & a^{(i)}_{12} & \dots & a^{(i)}_{1,k-1} & a^{(i)}_{1k} \\ a^{(i)}_{21} & a^{(i)}_{22} & \dots & a^{(i)}_{2,k-1} & a^{(i)}_{2k} \\ 0 & a^{(i)}_{32} & \dots & a^{(i)}_{3,k-1} & a^{(i)}_{3k} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & a^{(i)}_{k,k-1} & a^{(i)}_{kk} \end{pmatrix}, \begin{pmatrix} a^{(i)}_{k+1,k+1} & a^{(i)}_{k+1,k+2} & \dots & a^{(i)}_{k+1,n-1} & a^{(i)}_{k+1,n} \\ a^{(i)}_{k+2,k+1} & a^{(i)}_{k+2,k+2} & \dots & a^{(i)}_{k+2,n-1} & a^{(i)}_{k+2,n} \\ 0 & a^{(i)}_{k+3,k+2} & \dots & a^{(i)}_{k+3,n-1} & a^{(i)}_{k+3,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & a^{(i)}_{n,n-1} & a^{(i)}_{nn} \end{pmatrix}$$

mit dem QR-Algorithmus behandeln.

QR-Algorithmus

Als Kriterium für klein genug wählt man mit $\varepsilon = 10^{-t}$, wobei t die Zahl der signifikanten Stellen bezeichnet,

$$|a^{(i)}_{k+1,k}| \leqslant \varepsilon \min\{|a^{(i)}_{kk}|, |a^{(i)}_{k+1,k+1}|\}$$

QR-Algorithmus

Als Kriterium für klein genug wählt man mit $\varepsilon = 10^{-t}$, wobei t die Zahl der signifikanten Stellen bezeichnet,

$$|a^{(i)}_{k+1,k}| \leqslant \varepsilon \min\{|a^{(i)}_{kk}|, |a^{(i)}_{k+1,k+1}|\}$$

oder (weniger einschneidend)

$$|a^{(i)}_{k+1,k}| \leqslant \varepsilon(|a^{(i)}_{kk}| + |a^{(i)}_{k+1,k+1}|).$$

QR-Algorithmus

Beispiel 6.28: Wir demonstrieren den Konvergenzverlauf für die Hessenberg-Matrix

$$\boldsymbol{A} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 1 & 0 \\ 0 & 3 & 2 & 1 \\ 0 & 0 & 2 & 4 \end{pmatrix} \in \mathbb{R}^{(4,4)}.$$

QR-Algorithmus

Beispiel 6.28: Wir demonstrieren den Konvergenzverlauf für die Hessenberg-Matrix

$$\boldsymbol{A} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 1 & 0 \\ 0 & 3 & 2 & 1 \\ 0 & 0 & 2 & 4 \end{pmatrix} \in \mathbb{R}^{(4,4)}.$$

Die folgende Tabelle enthält die Subdiagonalelemente bei dem oben beschriebenen Abbruch mit $t = 16$.

QR-Algorithmus

i	a_{21}	a_{32}	a_{43}
1	2	3	2
2	1.83	-2.23	1.61
3	1.65	1.43	$3.55 \cdot 10^{-1}$
4	$6.78 \cdot 10^{-1}$	-3.65	$3.82 \cdot 10^{-2}$
5	$7.63 \cdot 10^{-1}$	1.17	$-1.63 \cdot 10^{-3}$
6	$8.32 \cdot 10^{-1}$	$-5.32 \cdot 10^{-1}$	$3.93 \cdot 10^{-6}$
7	-1.18	$1.52 \cdot 10^{-1}$	$-3.03 \cdot 10^{-11}$
8	1.64	$-4.97 \cdot 10^{-2}$	$1.62 \cdot 10^{-21}$
9	-3.22	$2.89 \cdot 10^{-5}$	
10	1.81	$5.33 \cdot 10^{-10}$	
11	$-5.32 \cdot 10^{-1}$	$-2.48 \cdot 10^{-19}$	
12	$-4.46 \cdot 10^{-3}$		
13	$5.89 \cdot 10^{-8}$		
14	$-1.06 \cdot 10^{-17}$		

QR-Algorithmus

Nach einer Anlaufphase wird die Zahl der gültigen Stellen des letzten berücksichtigten Diagonalelements von Schritt zu Schritt ungefähr verdoppelt.

QR-Algorithmus

Nach einer Anlaufphase wird die Zahl der gültigen Stellen des letzten berücksichtigten Diagonalelements von Schritt zu Schritt ungefähr verdoppelt.

Dies zeigt die quadratische Konvergenz des Verfahrens.

QR-Algorithmus

Nach einer Anlaufphase wird die Zahl der gültigen Stellen des letzten berücksichtigten Diagonalelements von Schritt zu Schritt ungefähr verdoppelt.

Dies zeigt die quadratische Konvergenz des Verfahrens.

Für die Grundform des QR-Algorithmus wird diese Genauigkeit anstatt nach 14 Schritten erst nach ca. 300 Schritten erreicht. □