Numerik IIA

Endlichdimensionale Probleme

Bernhard Schmitt

Wintersemester 1998/99

Inhaltsverzeichnis

1 Best-Approximationen, Orthogonalbasen

Bei vielen wissenschaftlichen Problemen weiß man theoretisch (oder versucht dies durch einen Ansatz zu schätzen), wie die Abhängigkeit einer meßbaren Größe $y \in \mathbb{R}$ von Versuchsparametern $a = (a_1, \dots, a_p)^\mathsf{T} \in \mathbb{R}^p$ aussieht,

$$y = f(a_1, \dots, a_p; x_1, \dots, x_n).$$

Die Variablen $(x_1, \dots, x_n)^\mathsf{T} =: x$ sind hier unbekannte Konstanten (Modellparameter) einer Funktion f von bekannter Form, die aufgrund von Messungen zu bestimmen sind. Nach m Messungen mit unterschiedlichen Versuchsparametern $(a_{k1}, \dots, a_{kp})$ sollte im Idealfall gelten

$$y_k = f(a_{k1}, \dots, a_{kp}; x_1, \dots, x_n), \ k = 1, \dots, m. \tag{1.0.1}$$

Für $m > n$ existiert aber i.a. keine Lösung dieses Gleichungssystems (1.0.1), da die Meßwerte normalerweise Fehler enthalten oder das Modell f doch nicht exakt ist. Daher beschränkt man sich darauf, (1.0.1) mit möglichst kleinem *Defekt* zu lösen:

$$\sum_{k=1}^{m} \Big| y_k - f(a_{k1}, \dots, a_{kp}; x_1, \dots, x_n) \Big|^2 \stackrel{!}{=} \min . \tag{1.0.2}$$

Die Verwendung der Quadratsumme geht auf *Gauß* zurück. Diese *Methode der kleinsten Quadrate* ist einerseits einfach zu behandeln, hat aber auch besonders gute statistische Eigenschaften. Im folgenden werden nur lineare Funktionen f und die Euklidnorm betrachtet.

1.1 Kleinste-Quadrate-Lösung

Für eine lineare Funktion f ist (1.0.1) ein lineares Gleichungssytem, bei dem die Zahl der Gleichungen und Unbekannten i.a. nicht gleich sind. Daher muß zunächst für beliebige lineare Gleichungssysteme ein verallgemeinerter Lösungsbegriff definiert werden.

Definition 1.1.1 *Zu $A \in \mathbb{C}^{m\times n}$, $m, n \in \mathbb{N}$, und $y \in \mathbb{C}^m$ heißt $\hat{x} \in \mathbb{C}^n$* Kleinste-Quadrate-Lösung, *wenn $\hat{x}$ das Funktional*

$$\|Ax - y\|_2^2 = (Ax - y)^*(Ax - y) \tag{1.1.1}$$

minimiert. Ist das Minimum von (1.1.1) nicht eindeutig, so wird mit x^+ diejenige Kleinste-Quadrate-Lösung mit kleinster Norm bezeichnet,

$$\|x^+\|_2 = \min\{\|\hat{x}\|_2 : \ \hat{x} \in \mathbb{C}^n \ \textit{minimiert (1.1.1)}\}. \tag{1.1.2}$$

Um die Lösungen aus Definition 1.1.1 genauer charakterisieren zu können, werden folgende Begriffe benötigt. Zu $A \in \mathbb{C}^{m\times n}$ heißen

$$\begin{array}{rcll} N(A) & := & \{x \in \mathbb{C}^n : \ Ax = 0\} & \text{Nullraum bzw. Kern von } A, \\ R(A) & := & \{y \in \mathbb{C}^m : \ \exists x \in \mathbb{C}^n \text{ mit } Ax = y\} = A\mathbb{C}^n & \text{Bildraum von } A. \end{array} \tag{1.1.3}$$

Aus der Linearen Algebra wird später folgende wichtige Beziehung verwendet:

$$R(A)^{\perp} = N(A^*). \tag{1.1.4}$$

Satz 1.1.2 *Es sei $P : \mathbb{C}^m \mapsto R(A)$ der Orthogonal-Projektor auf $R(A)$, d.h., es gilt $P^2 = P,\ P^* = P$. Dann sind die folgenden Aussagen äquivalent:*

$$\begin{aligned}
\|A\hat{x} - y\|_2 &= \min\{\|Ax - y\|_2 :\ x \in \mathbb{C}^n\}, && (1.1.5)\\
A\hat{x} &= Py && (1.1.6)\\
A^*A\hat{x} &= A^*y, \quad \text{bzw.,} \quad A^*(A\hat{x} - y) = 0 \quad \text{‘Normalengleichung’}. && (1.1.7)
\end{aligned}$$

Beweis a) (Äquivalenz der beiden ersten Aussagen) Wegen $Ax = PAx\,\forall x$ gilt

$$\begin{aligned}
\|Ax - y\|_2^2 &= \|Ax - Py - (I - P)y\|^2\\
&= \|Ax - Py\|_2^2 - 2\mathrm{Re} y^* \underbrace{(I - P^*)P}_{=0}(Ax - y) + \|(I - P)y\|_2^2.
\end{aligned}$$

Da der letzte Summand nicht von x abhängt, wird das Minimum für $\|Ax - Py\|_2 = 0$ angenommen. Das Minimum wird erreicht, da $Py \in R(A)$.

b) (Äquivalenz der beiden letzten Gleichungen)

$$\left.\begin{array}{lll}
(1.1.6) \iff & \text{Defekt } d := A\hat{x} - y = (P - I)y & \in R(A)^{\perp}\\
(1.1.7) \iff & \text{Defekt } d := A\hat{x} - y & \in N(A^*)
\end{array}\right\} \begin{array}{l}\text{äquivalent wegen}\\ (1.1.4)\end{array} \qquad \blacksquare$$

Aus diesem Satz folgen Existenz von $\hat{x}$ und Eindeutigleit von x^+.

Satz 1.1.3 *Das Ausgleichsproblem (1.1.1) besitzt (mind.) eine Kleinste-Quadrate-Lösung $\hat{x}$. Für $m \geq n$, Rang$(A) = n$, ist sie eindeutig. Die allgemeine Lösung von (1.1.1) hat die Form $\hat{x} + N(A)$. Die Lösung x^+ mit kleinster Norm (1.1.2) ist eindeutig bestimmt.*

Beweis a) Die Existenz folgt aus (1.1.6) und $Py \in R(A)$ oder aus (1.1.7): A^*A ist positiv semidefinit, denn $x^*A^*Ax = \|Ax\|_2^2 \geq 0$. Es gilt $A^*Ax = 0 \ \Rightarrow x^*A^*Ax = \|Ax\|_2^2 = 0 \iff Ax = 0$, also $N(A^*A) = N(A)$ und daher $R(A^*A) = N(A^*A)^{\perp} = N(A)^{\perp} = R(A^*)$. Daher ist (1.1.7) lösbar, die allgemeine Lösung ist $\hat{x} + N(A)$.

b) Eindeutigkeit von x^+: Im affinen Unterraum $\hat{x} + N(A)$ existiert genau ein Element kleinster Norm, x^+, die Projektion des Nullpunkts auf diesen Unterraum. $\blacksquare$

1.2 Die Verallgemeinerte Inverse

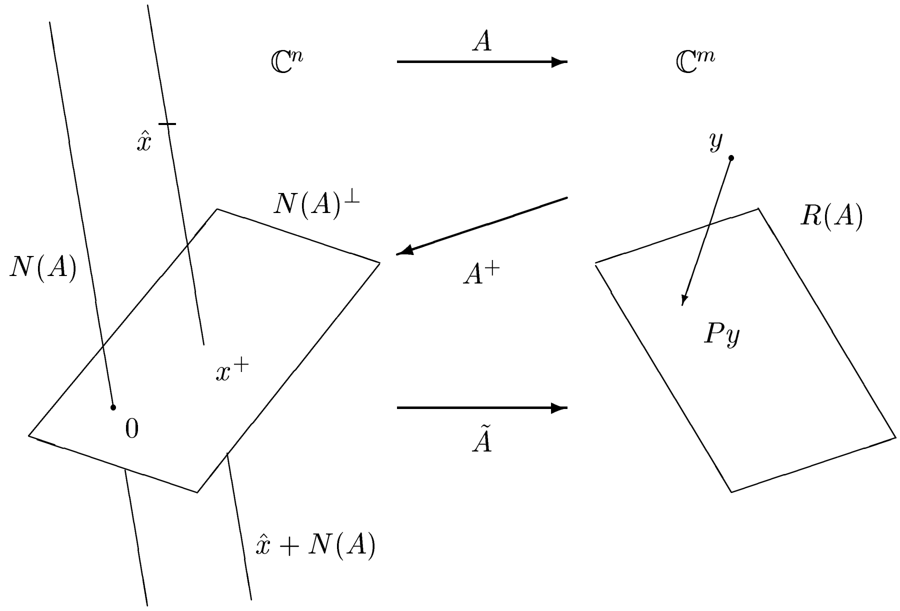
Im regulären Fall mit quadratischer Matrix A läßt sich *die* Lösung in der Form $x = A^{-1}y$ angeben. Da auch die Zuordnung $y \mapsto x^+$ linear ist, gibt es auch hier eine zugehörige Matrix.

Definition 1.2.1 *Mit $A \in \mathbb{C}^{m\times n}$, der Orthogonalprojektion $P: \mathbb{C}^m \mapsto R(A)$ und der restringierten Abbildung $\tilde{A} := A\Big|_{N(A)^\perp}$ heißt*

$$A^+ : \begin{cases} \mathbb{C}^m & \mapsto \quad \mathbb{C}^n \\ y & \mapsto \quad A^+y := (\tilde{A})^{-1}Py \end{cases} \tag{1.2.1}$$

Verallgemeinerte Inverse *von A (bzw. Moore-Penrose-Inverse, Pseudo-Inverse).*

Skizze:



Satz 1.2.2 *Die Minimallösung x^+ von (1.1.1), (1.1.2) stimmt mit*

$$x^+ = A^+y \tag{1.2.2}$$

überein, es gilt $R(A^+) = N(A)^\perp$.

Beweis Jedes $x \in \mathbb{C}^n$ ist eindeutig zerlegbar in $x = x_0 + x_\perp$ mit $x_0 \in N(A)$ und $x_\perp \in N(A)^\perp$. Damit ist $Ax = \tilde{A}x_\perp$. Mit dieser Zerlegung einer Minimallösung $\hat{x} = x_0 + x^+$ von (1.1.1) ergibt sich aus (1.1.6) und Satz 1.1.3 die Aussage:

$$\tilde{A}x^+ = A\hat{x} = Py. \qquad \blacksquare$$

Bemerkung: a) Jede Matrix besitzt eine Verallgemeinerte Inverse (auch die Nullmatrix).

b) Für $\text{Rang}(A) = n,\ m \geq n$, folgt insbesondere aus (1.1.7)

$$A^+ = (A^*A)^{-1}A^*.$$

Ist außerdem noch $m = n$, stimmen A^+ und A^{-1} überein.

Die Verallgemeinerte Inverse kann unabhängig von Definition 1.2.1 als Lösung der folgenden Gleichungen (1.2.3) charakterisiert werden (Moore-Penrose-Bedingungen).

Satz 1.2.3 *Die Verallgemeinerte Inverse A^+ aus (1.2.1) ist die eindeutig bestimmte Lösung der Gleichungen*

$$\begin{array}{lrcl} (a) & A^+A & = & (A^+A)^* \\ (b) & AA^+ & = & (AA^+)^* \\ (c) & AA^+A & = & A \\ (d) & A^+AA^+ & = & A^+ \end{array} \tag{1.2.3}$$

Es gilt $A^{++} = A$ und $(A^+)^* = (A^*)^+$.

Beweis a) Eine entscheidende Bedeutung besitzen die Orthogonalprojektoren

$$\begin{aligned} P: &\ \mathbb{C}^m \mapsto R(A) \\ U: &\ \mathbb{C}^n \mapsto N(A)^\perp, \quad I-U: \ \mathbb{C}^n \mapsto N(A). \end{aligned}$$

Als erstes wird nämlich gezeigt, daß $AA^+ = P$ und $A^+A = U$ ist. Dazu wird definiert

$$R := A^+A = \tilde{A}^{-1}PA = \tilde{A}^{-1}A: \ \mathbb{C}^n \mapsto N(A)^\perp. \tag{1.2.4}$$

Mit $I = U + (I-U)$ gilt $A = AU + A(I-U) = AU = \tilde{A}U$. Also ist $R = \tilde{A}^{-1}A = \tilde{A}^{-1}AU = \tilde{A}^{-1}\tilde{A}U = U$, und da U hermitesch ist, gilt insgesamt

$$U = A^+A = (A^+A)^*, \qquad \text{d.h. (1.2.3,a).} \tag{1.2.5}$$

Analog folgt

$$P = AA^+ = (AA^+)^*, \qquad \text{d.h. (1.2.3,b).} \tag{1.2.6}$$

Daraus ergeben sich aber auch die restlichen Gleichungen, denn

$$\begin{array}{rcll} (AA^+)A & = & PA = A & \text{wegen (1.2.6),} \qquad \text{d.h. (1.2.3,c),} \\ (A^+A)A^+ & = & UA^+ = A^+ & \text{wegen (1.2.5),} \qquad \text{d.h. (1.2.3,d).} \end{array}$$

Also löst A^+ aus (1.2.1) die Gleichungen (1.2.3).

b) Zum Nachweis der Eindeutigkeit sei $\hat{A}$ eine Lösung von (1.2.3). Dann gelten für $\hat{P} := A\hat{A}$ und $\hat{U} = \hat{A}A$ die Beziehungen $\hat{P}^* = \hat{P}$ und $\hat{U}^* = \hat{U}$, sowie

$$\hat{P}^2 = (A\hat{A}A)\hat{A} = A\hat{A} = \hat{P}, \quad \hat{U}^2 = (\hat{A}A\hat{A})A = \hat{A}A = \hat{U}. \tag{1.2.7}$$

Beide Abbildungen sind also Orthogonal-Projektoren, für die jetzt die Eigenschaften

$$N(\hat{U}) = N(A), \quad R(\hat{P}) = R(A)$$

nachgewiesen werden. Für $\hat{U}$ etwa folgt aus (1.2.3,c) die Identität $A = A\hat{A}A = A\hat{U}$, also $N(\hat{U}) \subseteq N(A)$. Andererseits ist für $x \in N(A)$ auch $\hat{U}x = \hat{A}Ax = 0$ und somit $N(\hat{U}) = N(A)$. Analog ergibt sich die Aussage für $R(\hat{P})$.

Da die Bildräume der Orthogonalprojektoren $\hat{P}$ bzw. $\hat{U}$ mit denen von P bzw. U übereinstimmen, sind auch die Projektoren gleich, $\hat{P} = A\hat{A} = P = AA^+$ und $\hat{U} = \hat{A}A = U = A^+A$. Aus (1.2.3,d) folgt daher

$$\hat{A} = (\hat{A}A)\hat{A} = A^+A\hat{A} = A^+AA^+ = A^+.$$

c) Die beiden letzten Aussagen folgen aus der Eindeutigkeit nach Ersetzung von A in (1.2.3) durch A^+ bzw. A^*. ■

Der nächste Abschnitt behandelt die numerische Berechnung der verallgemeinerten Inversen A^+.

1.3 QR-Zerlegung und Berechnung von A^+

In der Numerik I wurde gezeigt, daß die Konditionszahl $\kappa(B) = \|B\|\|B^{-1}\|$ entscheidend ist für die Fehleranfälligkeit eines linearen Gleichungssystems $Bx = b$. Die Lösung des Ausgleichsproblems (1.1.1) kann für Rang$(A) = n$ zwar über die Normalengleichung (1.1.7), $A^*Ax = A^*y$, berechnet werden. Die Kondition der dabei auftretenden Matrix $B = A^*A$ ist allerdings i.a. viel schlechter als beim Ausgangsproblem. Im regulären Fall $m = n$ gilt nämlich $\kappa_2(B) = \kappa_2(A^*A) = \varrho(A^*A)\varrho(A^{-1}A^{*-1}) = \kappa_2(A)^2$. Beim Übergang zur Normalengleichung wird die Konditionszahl des Ausgangssystems $Ax = y$ also quadriert und für $\kappa(A) \gg 1$ extrem vergrößert.

Eine Abhilfe bietet die Zerlegung der Matrix in eine unitäre Matrix Q und eine obere Dreieckmatrix R, die sogenannte *QR-Zerlegung*,

$$A = QR, \quad Q \in \mathbb{C}^{m\times m},\ Q^*Q = I, \quad R = \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ & r_{22} & \cdots & r_{2n} \\ & & \ddots & \end{pmatrix} \in \mathbb{C}^{m\times n}. \tag{1.3.1}$$

Mit dieser Zerlegung wird später das Ausgleichsproblem gelöst. Aber auch bei einer regulären, quadratischen Matrix A kann diese Zerlegung zur Lösung des Linearen Gleichungssystems $Ax = y$ verwendet werden, es gilt $x = R^{-1}Q^*y$. Diese Darstellung ist oft sogar numerisch stabiler als die übliche LR-Zerlegung, $A = L\tilde{R}$ (vgl. Numerik I), da $\kappa_2(R) = \kappa_2(QR) = \kappa_2(A)$, aber i.a. $\kappa_2(\tilde{R}) \gg \kappa_2(L\tilde{R}) = \kappa_2(A)$ gilt. Allerdings ist die Berechnung der QR-Zerlegung etwas teurer als die der LR-Zerlegung.

Die QR-Zerlegung (1.3.1) läßt sich im Prinzip mit dem Orthogonalisierungsverfahren von E.Schmidt (angewendet auf die Spaltenvektoren) berechnen. Dabei können jedoch große Rundungsfehler auftreten, etwa wenn Spaltenvektoren von A fast linear abhängig sind. Es ist numerisch günstiger, die Matrix durch elementare Spiegelungen (Householder-Transformationen, vgl. Lemma 1.3.1) umzuformen. Dabei wird jeweils, analog zum Gauß-Algorithmus, der erste Spaltenvektor der (Rest-) Matrix auf ein Vielfaches des ersten Einheitsvektors abgebildet.

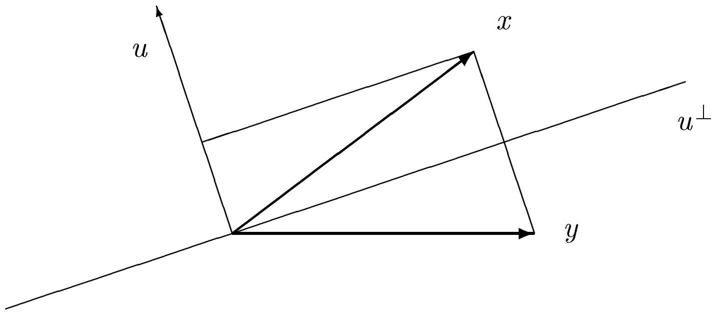
Lemma 1.3.1 *Mit* $u \in \mathbb{C}^m$, $\|u\|_2 = 1$, *sei*

$$H := I - 2uu^*. \tag{1.3.2}$$

Die Matrix H ist hermitesch ($H^* = H$), unitär ($H^*H = I$) und involutorisch ($H^2 = I$).

Beweis $H^* = I - 2uu^* = H \Rightarrow H^*H = H^2$, $H^2 = I - 4uu^* + 4(u^*u)uu^* = I$. ■

Geometrisch:
$y = Hx = x - 2(u^*x)u$
Spiegelung an $u^\perp$.



Anwendung: u ist so zu bestimmen, daß x auf die $e^{(1)}$-Achse gespiegelt wird, $e^{(1)} = (1, 0, 0, \ldots)^\mathsf{T}$.

$$Hx = x - 2(u^*x)u \overset{!}{=} \beta e^{(1)} \quad (= y). \tag{1.3.3}$$

$$\left.\begin{array}{lll} H \text{ unitär} & \Rightarrow & |\beta| = \|x\|_2 \\ H \text{ hermitesch} & \Rightarrow & \mathbb{R} \ni x^*Hx = \beta x^* e^{(1)} = \beta \bar{x}_1 \end{array}\right\} \Rightarrow \beta = \pm\sigma\|x\|_2,$$

wobei $x_1 = \sigma|x_1|$, $|\sigma| = 1$. Mit (1.3.3) und $\|u\|_2 = 1$ folgt die Darstellung

$$u = \frac{x - \beta e^{(1)}}{\|x - \beta e^{(1)}\|}.$$

In der ersten Komponente von u tritt die Differenz $x_1 - \beta$ auf. Um bei der Berechnung dieser Differenz eine Auslöschung führender Ziffern zu vermeiden, wird das Vorzeichen bei β so gewählt, daß tatsächlich eine Addition der Beträge stattfindet:

$$x_1 - \beta = \sigma|x_1| \mp \sigma\|x\|_2 \overset{!}{=} \sigma(|x_1| + \|x\|_2),$$

also $\beta = -\sigma\|x\|_2$. Damit wird

$$\|x - \beta e^{(1)}\|_2^2 = (|x_1| + \|x\|_2)^2 + |x_2|^2 + \ldots + |x_m|^2 = 2\|x\|(|x_1| + \|x\|).$$

Das Lemma faßt diese Konstruktion zusammen.

Lemma 1.3.2 *Der Vektor $0 \neq x \in \mathbb{C}^m$ mit $x_1 = \sigma|x_1|$, $|\sigma| = 1$, wird durch die Matrix*

$$H = I - \frac{vv^*}{\|x\|_2(|x_1| + \|x\|_2)}, \quad v := x + \sigma\|x\|_2 e^{(1)},$$

auf $-\sigma\|x\|_2 e^{(1)}$ abgebildet.

Mit dieser Konstruktion können in der ersten Spalte der Matrix A die Subdiagonalelemente eliminiert werden durch die Wahl $HAe^{(1)} = \beta e^{(1)}$. Wie beim Gaußalgorithmus wird dieser Schritt mit den Restmatrizen wiederholt. Im j-ten Schritt ($A^{(1)} := A$) hat man also

$$A^{(j)} = \left(\begin{array}{ccc|ccc} * & \cdots & * & * & \cdots & * \\ & \ddots & \vdots & \vdots & & \vdots \\ & & * & * & \cdots & * \\ \hline & & & a_{jj}^{(j)} & \cdots & a_{jn}^{(j)} \\ & 0 & & \vdots & & \vdots \\ & & & a_{mj}^{(j)} & \cdots & a_{mn}^{(j)} \end{array}\right) =: \begin{pmatrix} R^{(j)} & B^{(j)} \\ 0 & \tilde{A}^{(j)} \end{pmatrix},$$

mit $R^{(j)} \in \mathbb{C}^{(j-1)\times(j-1)}$, $\tilde{A}^{(j)} \in \mathbb{C}^{(m-j+1)\times(n-j+1)}$. Da Produkte von Spiegelungen unitär sind, kann A so unitär auf Dreieckgestalt transformiert werden. Dazu sind die Transformationsmatrizen in der Form

$$H^{(j)} = \left(\begin{array}{c|c} I_{j-1} & 0 \\ \hline 0 & \tilde{H}^{(j)} \end{array}\right) \qquad \Longleftrightarrow \qquad {u^{(j)}}^{\mathsf{T}} = (\underbrace{0,\ldots,0}_{j-1}, \underbrace{*,\ldots,*}_{m-j+1})$$

zu wählen. Beim Produkt $H^{(j)}A^{(j)} = A^{(j+1)}$ bleiben dann die ersten $j-1$ Zeilen unverändert. Die Anwendung von Lemma 1.3.2 erfolgt mit

$$\tilde{H}^{(j)} \begin{pmatrix} a_{jj}^{(j)} \\ \vdots \\ a_{mj}^{(j)} \end{pmatrix} \overset{!}{=} \begin{pmatrix} * \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{1.3.4}$$

In $k := \min\{m-1, n\}$ Schritten bekommt man so

$$A^{(k+1)} = R^{(k+1)} =: R = H^{(k)} \cdots H^{(2)} H^{(1)} A \qquad \Longleftrightarrow \qquad A = H^{(1)} H^{(2)} \cdots H^{(k)} R =: QR.$$

Der folgende Satz faßt diese Konstruktion zusammen.

Satz 1.3.3 *Die QR-Zerlegung einer Matrix $A \in \mathbb{C}^{m\times n}$ in das Produkt einer unitären Matrix $Q \in \mathbb{C}^{m\times m}$ und einer oberen Dreieckmatrix $R \in \mathbb{C}^{m\times n}$ ergibt sich in $k = \min\{m-1, n\}$ Schritten in folgender Weise.*

$$A^{(1)} := A, \quad A^{(j)} = (a_{\mu\nu}^{(j)})_{\mu=1,\nu=1}^{m,n}. \tag{1.3.5}$$

Für $j = 1, 2, \ldots, k$ berechne man

$$\left.\begin{array}{r} c_j^2 := \sum_{\mu=j}^m |a_{\mu j}^{(j)}|^2, \ c_j > 0, \quad K_j := c_j^2 + |a_{jj}^{(j)}| c_j \\ {v^{(j)}}^{\mathsf{T}} := \left(0,\ldots,0, \sigma_j c_j + a_{jj}^{(j)}, a_{j+1,j}^{(j)}, \ldots, a_{mj}^{(j)}\right), \ \textit{wobei } \sigma_j |a_{jj}^{(j)}| = a_{jj}^{(j)}, \end{array}\right\} \tag{1.3.6}$$

und bilde

$$\left.\begin{array}{l} H^{(j)} := I - \frac{1}{K_j} v^{(j)} v^{(j)*} \\ A^{(j+1)} := H^{(j)} A^{(j)} = A^{(j)} - v^{(j)} \left(\frac{1}{K_j} v^{(j)*} A^{(j)}\right) \end{array}\right\} \tag{1.3.7}$$

Dann ist $A = QR$ mit

$$Q = H^{(1)} \cdots H^{(k)}, \quad R = A^{(k+1)}, \tag{1.3.8}$$

$$\textit{wobei}\quad R=\begin{pmatrix} r_{11} & \dots & r_{1m} & \dots & r_{1n}\\ & \ddots & \vdots & & \vdots\\ 0 & & r_{mm} & \dots & r_{mn}\end{pmatrix},\quad R=\begin{pmatrix} r_{11} & \dots & r_{1n}\\ & \ddots & \vdots\\ 0 & & r_{nn}\\ & 0 & \end{pmatrix}$$

$$\textit{wenn}\qquad m\le n, k:=m-1, \qquad\qquad m>n, k:=n.$$

Praktische Durchführung, Rechenaufwand

Wegen des hohen Aufwands werden keine Produkte von Matrizen ausgerechnet, der Übergang $A^{(j+1)} = H^{(j)}A^{(j)}$ wird vielmehr entsprechend der Klammerung in (1.3.7) durchgeführt. Auch Q wird nicht explizit berechnet. Bei der Auflösung mehrerer Gleichungssysteme mit der Matrix A kann das Produkt Q^*y mit gleichem Aufwand in der Form $H^{(k)}\cdots H^{(1)}y$ ausgewertet werden. Dazu kann man die Komponenten der Vektoren $v^{(j)}$ im freien Platz unter und einschließlich der Hauptdiagonale von R unterbringen. Nur die Elemente r_{jj} werden zur Vereinfachung anderweitig gespeichert. So erfordert die Durchführung von (1.3.6),(1.3.7) folgenden *Rechenaufwand*:

Der Aufwand für c_j, K_j ist geringfügig und wird vernachlässigt. In (1.3.7) werden berechnet:

$$z^{(j)^*} := v^{(j)^*}A^{(j)} = (\underbrace{0\dots0}_{j-1}\underbrace{*\dots*}_{m-j+1})\begin{pmatrix} \ddots & \\ \underbrace{0}_{j-1} & \Big| \underbrace{\tilde A^{(j)}}_{n-j+1}\end{pmatrix}\Big\}_{m-j+1} = (\underbrace{0\dots0}_{j-1}\bullet\underbrace{*\dots*}_{n-j}).$$

Da das Ergebnis in der j-ten Spalte ('$\bullet$') nach Konstruktion (1.3.4) bekannt ist, sind für die Produktbildung $2(m-j+1)(n-j)$ Operationen nötig. Der Vektor $v^{(j)}$ besitzt höchstens $m-j+1$, $z^{(j)}$ nicht mehr als $n-j$ nichttriviale Elemente. Daher benötigt die Bildung von $A^{(j+1)} = A^{(j)} - v^{(j)}[\frac{1}{K_j}z^{(j)^*}]$ ebenfalls $2(m-j+1)(n-j)$ Operationen (Spalte j ist bekannt). Im Fall $n<m$ erhält man so den Gesamtaufwand

$$4\sum_{j=1}^{n}(m-j+1)(n-j) = 4\sum_{\nu=0}^{n-1}(m-n+\nu+1)\nu = \frac{4}{3}n^3 + 2(m-n)n^2 + \dots$$

Ein analoges Ergebnis gilt für $n\ge m$ und führt auf die Aussage

<u>**Rechenaufwand**</u> für die QR-Zerlegung einer $m\times n$-Matrix:

$$\frac{4}{3}\min\{m^3,n^3\} + 2|m-n|\min\{m^2,n^2\} + \dots\ \text{Operationen.}$$

Bemerkung: Im regulären Fall mit $m=n$ ist die QR-Zerlegung also ungefähr doppelt so teuer wie eine LR-Zerlegung ohne Pivotisierung und noch 33% teurer als dieselbe mit Spaltenpivotisierung. Dafür wird die Konditionszahl κ_2 von R bei der QR-Zerlegung nicht vergrößert.

Mit Hilfe der QR-Zerlegung kann bei rechteckigen Matrizen von vollem Rang, $\mathrm{Rang}(A) = \min\{m,n\}$, das Kleinste-Quadrate-Problem (1.1.1) einfach gelöst und die *verallgemeinerte Inverse* A^+ einfach berechnet werden. Für $m\ge n$, $\mathrm{Rang}(A)=n$, gilt

$$A = QR, \quad \text{mit } R = \begin{pmatrix} R_0\\ 0\end{pmatrix},\ R_0\in\mathbb{C}^{n\times n}\ \text{regulär.} \tag{1.3.9}$$

Durch Aufspaltung der transformierten rechten Seite $Q^*y = z = \begin{pmatrix} \hat{z} \\ \tilde{z} \end{pmatrix}$, $\hat{z} = (z_1, \ldots, z_n)^\mathsf{T}$, wird (1.1.1) zerlegt:

$$\|Ax - y\|_2^2 = \|Q(Rx - Q^*y)\|_2^2 = \|Rx - z\|_2^2 = \|R_0 x - \hat{z}\|_2^2 + \|\tilde{z}\|_2^2. \tag{1.3.10}$$

Die zweite Norm ist jetzt unabhängig von x. Daher wird das Minimum bei $R_0 x = \hat{z}$ angenommen, also ist $\hat{x} = R_0^{-1}\hat{z}$. Zu dieser Zuordnung $y \mapsto \hat{x} = x^+$ gehört die Matrix

$$A^+ = R_0^{-1} E_{nm} Q^*, \quad E_{nm} := \left(\begin{array}{ccc|c} 1 & & & 0\cdots \\ & \ddots & & \vdots \\ & & 1 & 0\cdots \end{array}\right) = (I_n|0) \in \mathbb{R}^{n\times m}. \tag{1.3.11}$$

Die Matrix E_{nm} wählt dabei die ersten n Komponenten eines m-Vektors aus. Die Darstellung (1.3.11) ist äquivalent zu $A^+ = (A^*A)^{-1}A^*$, numerisch aber besser konditioniert (vgl. die Diskussion am Anfang des Abschnitts) und billiger! Für die Projektoren aus §2.2 gilt hier übrigens

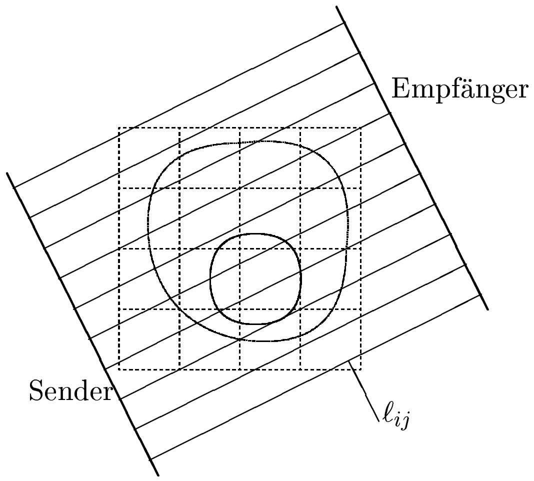
$$\begin{gathered} U = A^+A = I_n : \ \mathbb{C}^n \mapsto \mathbb{C}^n = N(A)^\perp, \\ P = AA^+ = QE_{nm}^\mathsf{T} R_0 R_0^{-1} E_{nm} Q^* = Q \begin{pmatrix} I_n & 0 \\ 0 & 0 \end{pmatrix} Q^* : \ \mathbb{C}^m \mapsto R(A). \end{gathered}$$

Im Fall $m < n$, $\mathrm{Rang}(A) = m$, kann analog vorgegangen werden durch Berechnung der QR-Zerlegung von $A^* = QR$. Hier gilt $A^+ = QE_{mn}^\mathsf{T}(R_0^*)^{-1} : \ \mathbb{C}^m \mapsto \mathbb{C}^n$.

Der allgemeine Fall, $\mathrm{Rang}(A) < \min\{m, n\}$, läßt sich mit Spalten- bzw. Zeilenvertauschungen auf eine Kombination der beiden behandelten Fälle zurückführen. Eine weiteres Lösungsverfahren mit Hilfe der aufwendigeren Singulärwert-Zerlegung, die allerdings auch zusätzliche Optionen bietet, wird in §3.7 im Anschluß an das Eigenwertproblem behandelt.

Beispiel 1.3.4 Computer-Tomographie

Die Dichte-Verteilung in einem Körper kann aus der von außen gemessenen Absorption von (Röntgen-Gamma-) Strahlen rekonstruiert werden. Zur Rekonstruktion wird eine Schicht des Untersuchungsbereichs, z.B., in Quadrate unterteilt. Der Bereich wird dann mit einer Reihe von parallelen Meßstrahlen durchleuchtet und die Messung unter verschiedenen Drehwinkeln der Apparatur wiederholt. Nimmt man in jedem Teilquadrat des Untersuchungsbereichs eine konstante Dichte d_j an, dann ergibt sich für die Absorption a_i im i-ten Meßstrahl die Beziehung



$$a_i = \sum_j \ell_{ij} d_j \,.$$

Dabei gibt der Koeffizient ℓ_{ij} die Weglänge des i-ten Meßstrahls im j-ten Quadrat an. Je nach Anzahl der Messungen erhält man ein (unter- oder überbestimmtes) Lineares Gleichungssystem für die unbekannten Dichten d_j. Dieses Rekonstruktionsproblem (Radon-Transformation) reagiert allerdings empfindlich auf Störungen. Daher sind in der Praxis Zusatzmaßnahmen erforderlich, um akzeptable Lösungen zu bekommen. Auch diese Problematik wird noch einmal in §3.7 angesprochen.

1.4 Trigonometrische Interpolation

Die Vorteile von Orthogonal-Transformationen bzw. -Basen wurden im letzten Abschnitt klar. Bei der trigonometrischen Approximation bzw. Interpolation von periodischen Funktionen kommt als weiterer Vorteil ein extrem schnelles Berechnungsverfahren hinzu. Zur Darstellung periodischer Funktionen betrachtet man zweckmäßigerweise Polynome auf dem komplexen Einheitskreis bzw. direkt trigonometrische Polynome

$$p(x) := a_0 + a_1 e^{ix} + a_2 e^{2ix} + \ldots + a_{n-1} e^{(n-1)ix}, \; x \in [0, 2\pi). \tag{1.4.1}$$

Hier ist $i = \sqrt{-1}$ die imaginäre Einheit. Bei der Interpolation werden aus praktischen Gründen nur äquidistante Stützstellen $x_k = 2\pi \frac{k}{n}$, $k = 0, \ldots, n-1$, behandelt. Zu gegebenen Stützwerten $y_k \in \mathbb{C}$ ist das Interpolationspolynom p gesucht mit

$$p(x_k) = p\Big(2\pi \frac{k}{n}\Big) = y_k, \quad k = 0, 1, \ldots, n-1. \tag{1.4.2}$$

Wegen der 2π-Periodizität von p ist dann auch $p(2\pi k/n) = y_k \forall k \in \mathbb{Z}$, wenn definiert wird $y_{k+jn} := y_k \forall j \in \mathbb{Z}$. Nach Einführung der Größe $\omega_n := \exp(2\pi i/n)$ entspricht (1.4.2) dem linearen Gleichungssystem

$$y_k = p(x_k) = \sum_{j=0}^{n-1} a_j e^{2\pi i jk/n} = \sum_{j=0}^{n-1} a_j \, \omega_n^{jk}, \quad k = 0, 1, \ldots, n-1. \tag{1.4.3}$$

Da die Zahl ω_n eine *n-te primitive Einheitswurzel* ist mit $\omega_n^n = 1$, folgen die Beziehungen

$$\begin{aligned} \omega_n^{k+jn} &= \omega_n^k \; \forall j \in \mathbb{Z} \\ \sum_{k=0}^{n-1} \omega_n^{jk} \, \omega_n^{-\ell k} &= n\delta_{j\ell} = \begin{cases} n & \text{für } j = \ell \\ 0 & \text{für } j \neq \ell \end{cases}. \end{aligned} \tag{1.4.4}$$

Beweis Die Nachweise für Periodizität und Summenwert bei $j = \ell$ sind trivial, für $j \neq \ell$ gilt

$$\sum_{k=0}^{n-1} \omega_n^{(j-\ell)k} = \frac{1 - \omega_n^{(j-l)n}}{1 - \omega_n^{j-\ell}} = 0, \quad \text{da } \omega_n^n = 1. \qquad \blacksquare$$

Die Identität (1.4.4) besagt, daß die Vektoren $w^{(k)} := \left(\omega_n^{jk}\right)_{j=0}^{n-1} \in \mathbb{C}^n$ eine *Orthonormalbasis* bilden im Innenprodukt

$$(y, z) := \frac{1}{n} \sum_{j=0}^{n-1} y_j \, \bar{z}_j. \tag{1.4.5}$$

Daher läßt sich die Lösung des Interpolationsproblems (1.4.3) explizit angeben, denn dazu sind nur die Koeffizienten a_j in der Orthonormalentwicklung $y = \sum_{j=0}^{n-1} a_j w^{(j)}$ zu berechnen.

Satz 1.4.1 *Die Koeffizienten a_j des Interpolationspolynoms (1.4.1) mit (1.4.2) sind*

$$a_j = (y, w^{(j)}) = \frac{1}{n} \sum_{k=0}^{n-1} y_k \, \omega_n^{-jk}, \quad j = 0, 1, \ldots, n-1. \tag{1.4.6}$$

Beweis Da $\{w^{(j)}\}$ eine Orthonormalbasis ist, gilt $(y, w^{(j)}) = (\sum_{k=0}^{n-1} a_k w^{(k)}, w^{(j)}) = a_j$. ∎

Bemerkung: Man beachte die Ähnlichkeit von (1.4.3) und (1.4.6).

Die Entwicklung nach einer Orthonormalbasis hat zur Folge, daß Teilsummen $\sum_j a_j w^{(j)}$ auch jeweils Bestapproximierende an y sind:

Satz 1.4.2 *Für beliebiges $\ell \leq n-1$ minimiert das Abschnittspolynom*

$$p_\ell(x) := a_0 + a_1 e^{ix} + \ldots + a_\ell e^{i\ell x}$$

unter allen Polynomen q_ℓ vom Grad ℓ die Fehlerquadratsumme

$$\sum_{k=0}^{n-1} |y_k - q_\ell(x_k)|^2.$$

Beweis (Pythagoras) p_ℓ ist Orthogonalprojektion von p bzgl. $(\cdot, \cdot)$ auf die Menge der trigon. Polynome vom Grad ℓ. ∎

Reelle Trigonometrische Interpolation:

Für reelle Daten $y_k \in \mathbb{R}$ läßt sich das reelle trigonometrische Polynom

$$\varphi(x) := \frac{\alpha_0}{2} + \sum_{k=1}^{m-1} (\alpha_k \cos kx + \beta_k \sin kx) + \frac{\alpha_m}{2} \cos mx, \tag{1.4.7}$$

das die Werte y_k interpoliert, über (1.4.6) bestimmen.

Satz 1.4.3 *Die Koeffizienten a_j seien mit $n := 2m$, $y_k \in \mathbb{R}$, nach (1.4.6) definiert und*

$$\begin{aligned}
\alpha_j &:= \quad 2\mathit{Re}\; a_j = a_j + a_{n-j} &&= \frac{2}{n} \sum_{k=0}^{n-1} y_k \cos kx_j, \ j = 0, \ldots, m, \\
\beta_j &:= -2\mathit{Im}\; a_j = \tfrac{1}{i}(a_{n-j} - a_j) &&= \frac{2}{n} \sum_{k=0}^{n-1} y_k \sin kx_j, \ j = 1, \ldots, m-1.
\end{aligned}$$

Dann gilt für das trigonometrische Polynom (1.4.7) mit diesen Koeffizienten die Aussage

$$\varphi(x_k) = y_k \quad k = 0, \ldots, n-1.$$

Beweis Für reelle y_k und mit $2\mathrm{Re}\,(a_j\omega_n^{jk}) = \alpha_j \cos jx_k + \beta_j \sin jx_k$ folgt

$$\begin{aligned} a_{n-j} &= \frac{1}{n}\sum_{k=0}^{n-1} y_k\omega_n^{k(j-n)} = \frac{1}{n}\sum_{k=0}^{n-1} y_k\omega_n^{kj} = \bar{a}_j \quad \text{und daher} \\ y_k &= p(x_k) = \sum_{j=0}^{n-1} a_j\omega^{jk} = \sum_{j=0}^{m-1} a_j\omega_n^{jk} + \sum_{\ell=1}^{m} a_{n-\ell}\omega_n^{kn-k\ell} \\ &= a_0 + \sum_{j=1}^{m-1}(a_j\omega_n^{jk} + a_{n-j}\omega_n^{-jk}) + a_m\omega^{mk} \\ &= a_0 + \sum_{j=1}^{m-1} 2\mathrm{Re}\,(a_j\omega_n^{jk}) + a_m(-1)^k = \varphi(x_k). \qquad \blacksquare \end{aligned}$$

Die schnelle Fourier-Transformation (FFT)

Sowohl zur Bestimmung der Koeffizienten (1.4.6) als auch zur Auswertung des Polynoms (1.4.3) sind Summen der Form

$$z_k := \sum_{j=0}^{n-1} b_j\omega_n^{jk}, \quad k = 0,\dots,n-1;\ \omega_n^n = 1,$$

zu berechnen (*diskrete Fourier-Transformation*). Im Normalfall werden dafür n^2 Operationen benötigt. Wenn n aber eine Zweierpotenz ist, kann der Aufwand auf $n\log_2 n$ Operationen gesenkt werden ($\to$ *schnelle* Fourier-Tr.) aufgrund der hohen Symmetrie der Koeffizienten ω_n^{jk}. Das Prinzip ist schon bei $n = 2m$ erkennbar, dort gilt

$$\left.\begin{aligned} z_{2k} &= \sum_{j=0}^{n-1} b_j\omega_{2m}^{2jk} = \sum_{j=0}^{m-1}(b_j + b_{j+m})\omega_m^{jk} &&=: \sum_{j=0}^{m-1} b_j^g\omega_m^{jk}, \\ &\quad \text{da}\quad \omega_{2m}^2 = \omega_m,\ \omega_m^{(j+m)k} = \omega_m^{jk}, \\ z_{2k+1} &= \sum_{j=0}^{n-1} b_j\omega_{2m}^{2jk+j} = \sum_{j=0}^{m-1}[(b_j - b_{j+m})\omega_n^j]\omega_m^{jk} &&=: \sum_{j=0}^{m-1} b_j^u\omega_m^{jk}, \\ &\quad \text{da}\quad \omega_{2m}^2 = \omega_m,\ \omega_{2m}^m = -1. \end{aligned}\right\} \qquad (1.4.8)$$

Durch den Übergang vom Gesamtvektor $(b_j)_{j=0}^{n-1}$ zu den beiden Teilvektoren $(b_j^g, b_j^u)_{j=0}^{m-1}$, der $2m$ Additionen und m Multiplikationen kostet, kann die Gesamt-Fourier-Transformation zu $n = 2m$ auf 2 Transformationen der halben Ordnung m zurückgeführt werden. Für den Aufwand A_n der Fourier-Transformation gilt also

$$A_{2m} = 2A_m + 3m, \quad A_1 = 0.$$

Für $n = 2^\ell$ ist die Methode (1.4.8) rekursiv anwendbar. Man sieht sofort, daß dann die Rekursionsformel durch $A_{2^\ell} = 3\ell 2^{\ell-1} = \frac{3}{2}n\log_2 n$ erfüllt wird, denn $A_{2^{\ell+1}} = 3(\ell+1)2^\ell = 2[3\ell 2^{\ell-1}] + 3\,2^\ell$.

Satz 1.4.4 *Für $n = 2^\ell$ können die Summen (1.4.3) bzw. (1.4.6) mit $\frac{3}{2}n\log_2 n$ komplexen Operationen berechnet werden.*

Praktische Durchführung bei $n = 2^\ell$ [Cooley-Tuckey]:

Die Durchführung der ℓ Transformations-Stufen bei Anwendung von (1.4.8) kann ohne zusätzlichen Speicherbedarf, d.h., am Platz von (b_j), stattfinden. Dazu werden die Koeffizienten b_j^g, $j = 0,\ldots,m-1$, in die vordere Hälfte und die b_j^u, $j = 0,\ldots,m-1$, in die hintere Hälfte eingeordnet, die nächste Transformations-Stufe arbeitet getrennt auf beiden Hälften:

$$\left\{\begin{array}{rlll} b_j^{(1)} := & b_j^g = b_j + b_{j+m}, & j = 0,\ldots,m-1, & \text{Berechnung } \{z_{2j}\} \\ b_{j+m}^{(1)} := & b_j^u = (b_j - b_{j+m})\omega_n^j, & j = 0,\ldots,m-1, & \text{Berechnung } \{z_{2j+1}\}. \end{array}\right. \tag{1.4.9}$$

Zur korrekten Zuordnung werden die Indizes der Koeffizienten $b_j^{(1)}$ und die der Zielkoeffizienten z_k anhand ihrer Binärdarstellung (Binärstellen s_j in den Kästchen) verglichen:

$$\begin{array}{lccc} \text{Koeff. mit} & \left\{\begin{array}{l} k = 2j \\ k = 2j+1 \end{array}\right. & \text{kommt in Position} & \left\{\begin{array}{l} j \\ m+j \end{array}\right. \\ \text{Index binär:} & \boxed{s_{\ell-1}}\boxed{\cdots}\boxed{s_1}\boxed{s_0} & \longrightarrow & \boxed{s_0}\,\boxed{s_{\ell-1}}\boxed{\cdots}\boxed{s_1} \end{array}$$

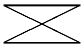
Die weiteren Stufen lassen die höchste Binärstelle unverändert und ändern die restlichen analog. Daher haben sich nach $\ell = \log_2 n$ Schritten die Binärdarstellungen der Indizes umgedreht:

$$z_k \text{ mit } k = \boxed{s_{\ell-1}}\boxed{\cdots}\boxed{s_1}\boxed{s_0} \text{ steht in } b_j^{(\ell)} \text{ mit } j = \boxed{s_0}\boxed{s_1}\boxed{\cdots}\boxed{s_{\ell-1}}.$$

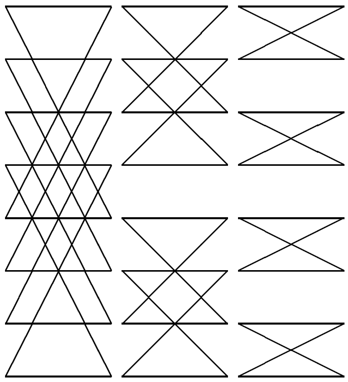
Zur Korrektur ist am Ende der FFT ein Permutationsschritt durchzuführen.

Beispiel 1.4.5 Schnelle Fourier-Transformation bei $n = 8 = 2^3$

	Index binär	1.Schritt	für Koeff	2.Schritt	für Koeff	3.Schritt	Index	Koeff
b_0	000	$b_0 + b_4$	xx0	$b_0 + b_2$	x00	$b_0 + b_1$	000	z_0
b_1	001	$b_1 + b_5$	xx0	$b_1 + b_3$	x00	$(b_0 - b_1)\omega^0$	100	z_4
b_2	010	$b_2 + b_6$	xx0	$(b_0 - b_2)\omega^0$	x10	$b_2 + b_3$	010	z_2
b_3	011	$b_3 + b_7$	xx0	$(b_1 - b_3)\omega^2$	x10	$(b_2 - b_3)\omega^0$	110	z_6
b_4	100	$(b_0 - b_4)\omega^0$	xx1	$b_4 + b_6$	x01	$b_4 + b_5$	001	z_1
b_5	101	$(b_1 - b_5)\omega^1$	xx1	$b_5 + b_7$	x01	$(b_4 - b_5)\omega^0$	101	z_5
b_6	110	$(b_2 - b_6)\omega^2$	xx1	$(b_4 - b_6)\omega^0$	x11	$b_6 + b_7$	011	z_3
b_7	111	$(b_3 - b_7)\omega^3$	xx1	$(b_5 - b_7)\omega^2$	x11	$(b_6 - b_7)\omega^0$	111	z_7

Eine übersichtliche Darstellung des Ablaufs gibt das Schmetterlingsdiagramm rechts (bei $n = 8$, 3 Zyklen von links nach rechts). Dabei entspricht ein *Schmetterling*  einem Paar von Operationen (1.4.9) mit zwei Spaltenelementen.

Das eigentliche Problem bei der Realisierung des Gesamt-Algorithmus ist die Ablaufsteuerung. Dazu reichen drei Schleifen aus. In der äußeren werden die $\log_2 n$ Zyklen der Transformation durchlaufen, außerdem sind je eine Schleife zur Zählung der Tabellenteile und der einzelnen Komponentenpaare erforderlich.



Anwendungen der FFT:

a) Approximation von Koeffizienten der Fourier-Entwicklung

$$c_j = \frac{1}{2\pi}\int_0^{2\pi} f(x)e^{-ijx}dx,\ j \in \mathbb{Z}, \tag{1.4.10}$$

von 2π-periodische Funktionen f. Für reelles f ist

$$\operatorname{Re} c_j = \frac{1}{2\pi}\int_0^{2\pi} f(x)\cos jx\,dx, \quad \operatorname{Im} c_j = -\frac{1}{2\pi}\int_0^{2\pi} f(x)\sin jx\,dx.$$

Mit den Werten $y_k = f(x_k) = f(2\pi k/n),\ k \in \mathbb{Z}$, in (1.4.6) gilt wegen der Periodizität

$$\begin{aligned} a_j &= \frac{1}{n}\sum_{k=0}^{n-1} f(x_k)\exp(-2\pi i\frac{jk}{n}) \\ &= \frac{h}{2\pi}\Big(\frac{1}{2}f(0) + \sum_{k=1}^{n-1} f(x_k)\exp(-2\pi i\frac{jk}{n}) + \frac{1}{2}f(2\pi)\Big), \end{aligned}$$

mit $h = 2\pi/n$. Dies entspricht genau der Trapeznäherung an das Integral (1.4.10). Aus der Euler-McLaurin-Summenformel (Numerik I, Satz 5.5.3) folgt, daß die Trapezregel für periodische und genügend oft differenzierbare Integranden eine beliebig hohe Ordnung besitzt, für $f \in C^m(\mathbb{R})$ gilt:

$$|a_j - c_j| \le k_m\Big(\frac{j+1}{n}\Big)^m \max_{\nu\le m}\|f^{(\nu)}\|_\infty,\ 0 \le j < n.$$

Daher bilden die ersten diskreten Fourierkoeffizienten a_j (z.B. mit $j \le \alpha n,\ \alpha < 1$) sehr gute Approximationen an die c_j. Dies gilt nicht mehr für $j \cong n$, da für eine reelle, stetige Funktion f zwar gilt $c_j \to 0,\ j \to \infty$, aber $a_{n-j} = \bar{a}_j$.

b) Schnelle Berechnung *diskreter Faltungen* von Vektoren. Zu Vektoren $u = (u_0, u_1, \dots, u_{n-1})^\mathsf{T}$, $y = (y_0, y_1, \dots, y_{n-1})^\mathsf{T} \in \mathbb{C}^n$, deren Komponenten bei Bedarf n-periodisch fortgesetzt werden, wird folgendes Produkt definiert,

$$z = u * y \quad :\Longleftrightarrow \quad z_k = \sum_{j=0}^{n-1} u_{k-j}\,y_j,\ k = 0, \dots, n-1, \tag{1.4.11}$$

bzw. $k \in \mathbb{Z}$. Dieses *Faltungsprodukt* $*$ ist kommutativ und assoziativ. Werden die diskreten Fouriertransformierten zu $u, y, z \in \mathbb{C}^n$ mit $\widehat{u}, \widehat{y}, \widehat{z}$ bezeichnet, etwa $\widehat{y} = (\widehat{y}_k) = (\sum_{j=0}^{n-1} y_j\omega_n^{jk})$, so gilt

$$\widehat{z} = \widehat{(u * y)} = \widehat{u} \circ \widehat{y} := (\widehat{u}_k\widehat{y}_k)_{k=0}^{n-1}. \tag{1.4.12}$$

Beweis Wegen der Periodizität ist

$$\widehat{z}_k = \sum_{j=0}^{n-1} z_j\omega_n^{jk} = \sum_{j=0}^{n-1}\sum_{m=0}^{n-1} u_{j-m}y_m\omega_n^{(j-m)k}\omega_n^{mk} = \sum_{m=0}^{n-1} y_m\omega_n^{mk}\sum_{l=-m}^{n-1-m} u_l\omega_n^{lk} = \widehat{y}_k\widehat{u}_k. \blacksquare$$

Wegen (1.4.12) läßt sich die Berechnung einer Faltung (Aufwand direkt: $2n^2$) durch komponentenweise Multiplikation der Fourierkoeffizienten realisieren. Diese und die drei erforderlichen Fouriertransformationen kosten insgesamt nur ca. $\frac{9}{4}n\log_2 n$ Operationen. Sind in (1.4.11) u und z bekannt, kann umgekehrt das Gleichungssystem für y mit der *zyklischen* Matrix $(u_{k-j})_{k,j=0}^{n-1}$ mit gleichem Aufwand (statt n^3 bei Gauß-Elimination) gelöst werden.

Anwendungen von (1.4.11), (1.4.12):

— *digitale Filter*, z.B. gleitender Mittelwert $z_j := \frac{1}{4}(y_{j-1} + 2y_j + y_{j+1})$, *Tiefpaß-Filter*,

— Bildverarbeitung (2-dimens. Filter), mehrdimensionale Fouriertransformation, Bildtelephon, Bild-Komprimierung

— Polynom- bzw. Langzahl-Multiplikation (statt in $\mathbb{C}$ in endlichen Zahlkörpern, auch dort gibt es primitive Einheitswurzeln),

— Statistik

c) Polynom-Interpolation in Tschebyscheff-Knoten: vgl. Numerik I, §3.6.

Verallgemeinerungen, Weiterentwicklungen

- Permutationsfreie FFT: Durch geeignete Zusammenfassung von Schritten in den letzten $\lfloor \ell/2 \rfloor$ Stufen der FFT erscheinen die Koeffizienten am Ende in der richtigen Reihenfolge.
- Für allgemeinere n mit Primfaktorzerlegung $n = p_1 p_2 \cdots p_\ell$ ist analog eine Aufspaltung in ℓ Stufen möglich (z.B. $n = 1000 = 2^3 5^3$).
- Das Berechnungsprinzip ist mit Abwandlungen auch auf einige andere Transformationen übertragbar. Die Cosinus-Transformation (*reelle* trigonometrische Entwicklung, reell durchführbar) erhält man, wenn man nur spiegelsymmetrische Funktionen betrachtet. Z.B. bekommt man aus einer in $[0, \pi]$ definierten stetigen Funktion f mit $f(2\pi - x) := f(x)$ eine 2π-periodische überall stetige Funktion. Dies reduziert i.a. die Anteile der hohen Frequenzen. Diese Transformation wird beim Bildtelefon und der Bild-Komprimierung eingesetzt (JPEG-Bild-Format). Bei der Walsh-Transformation erfolgt eine reelle Entwicklung nach Treppenfunktionen, hier treten i.w. nur Additionen und Subtraktionen auf.

2 Lineare Optimierung

2.1 Problemstellung, Normalform

Die lineare Optimierung befaßt sich mit einem, v.a. im kaufmännischen Bereich, sehr wichtigen Problem der numerischen Datenverarbeitung, dem der Optimierung einer linearen Funktion unter linearen Nebenbedingungen in Form von Ungleichungen ('Lineare Programme').

Beispiel 2.1.1 Eine gewünschte Metallegierung L_0 aus 30% Blei, 30% Zink und 40% Zinn soll möglichst billig produziert werden. Als Ausgangsmaterial stehen 5 Legierungen $L_1, \dots, L_5$ zur Verfügung mit folgenden Daten:

	Legierung	L_1	L_2	L_3	L_4	L_5	L_0
Gewichts-	Blei	0.1	0.1	0.6	0.4	0.3	0.3
Anteile	Zink	0.5	0.4	0.1	0.2	0.6	0.3
	Zinn	0.4	0.5	0.3	0.4	0.1	0.4
	Preis/kg	40	45	50	52	20	?

Zur mathematischen Formulierung dieses Problems werden die unbekannten Gewichtsanteile x_i der Legierungen L_i an L_0 eingeführt. Bei der Produktion sind die Nebenbedingungen

$$\begin{aligned} & x_i \geq 0, \quad i = 1, \dots, 5, \\ \text{Blei:}\quad & 0.1x_1 + 0.1x_2 + 0.6x_3 + 0.4x_4 + 0.3x_5 \quad = 0.3 \\ \text{Zink:}\quad & 0.5x_1 + 0.4x_2 + 0.1x_3 + 0.2x_4 + 0.6x_5 \quad = 0.3 \\ \text{Zinn:}\quad & 0.4x_1 + 0.5x_2 + 0.3x_3 + 0.4x_4 + 0.1x_5 \quad = 0.4 \end{aligned}$$

einzuhalten, das Ziel ist aber letztendlich die Minimierung der Kosten

$$40x_1 + 45x_2 + 50x_3 + 52x_4 + 20x_5 \stackrel{!}{=} \min .$$

Viel Optimierungsprobleme besitzen eine ähnliche Form oder können auf diese umgeformt werden. Sie wird als Normalform verwendet.

Definition 2.1.2 *Gegeben sei eine Matrix* $A \in \mathbb{R}^{m \times n}, m < n$, *mit Rang* $A = m$, *und* $b \in \mathbb{R}^m, c \in \mathbb{R}^n$. *Die Aufgabe, ein* $\hat{x} \in \mathbb{R}^n$ *zu bestimmen, das unter den* Nebenbedingungen

$$Ax = b, \quad x \geq 0,$$

(d.h. $x_i \geq 0 \ \forall i = 1, \dots, n$*) die* Zielfunktion

$$c^{\mathsf{T}} x$$

maximiert, heißt **L***ineares* **O***ptimierungsproblem in* **N***ormalform (LON). Kurzformulierung*

$$c^{\mathsf{T}} \hat{x} = \max\{c^{\mathsf{T}} x : \ x \in M\}, \qquad M := \{x \in \mathbb{R}^n : \ Ax = b, \, x \geq 0\}. \tag{2.1.1}$$

Die Menge M *wird die der* zulässigen Punkte *genannt.*

Andere lineare Probleme können in die Form (2.1.1) übergeführt werden. Außer trivialen Umformungen (Multiplikation mit -1) kommen auch in Frage:

- Ungleichungen als Nebenbedingungen $\alpha_1 x_1 + \ldots + \alpha_n x_n \leq \beta$ werden durch Einführung einer *Schlupfvariablen* y ersetzt durch
$$\alpha_1 x_1 + \ldots + \alpha_n x_n + y = \beta, \quad y \geq 0.$$
- Variablen x ohne Vorzeichenbeschränkung werden ersetzt durch zwei nichtnegative, $x = y_1 - y_2,\ y_1, y_2 \geq 0$.

2.2 Eigenschaften und Grundlagen

Das (schrittweise) Konstruktionsverfahren zur Berechnung einer Lösung von (2.1.1) nutzt die speziellen geometrischen Eigenschaften von M aus.

Satz 2.2.1 *Die Menge M aus (2.1.1) ist konvex. Es treten folgende Fälle auf:*

a) Für $M = \emptyset$ existiert keine Lösung des LON.

b) Für $M \neq \emptyset$, M beschränkt, existiert eine Lösung des LON.

c) Für $M \neq \emptyset$ existiert eine Lösung genau dann, wenn $\sup\{c^\mathsf{T} x : \ x \in M\} < \infty$.

Beweis M ist konvex als Durchschnitt des affinen Unterraums $\{x \in \mathbb{R}^n : \ Ax = b\}$ und der Halbräume $H_i = \{x \in \mathbb{R}^n : \ x_i \geq 0\}$. Die Aussage a) ist trivial, b) gilt, da M kompakt und die Zielfunktion stetig ist. Teil c) wird später konstruktiv gezeigt. ■

Definition 2.2.2 *Ein Punkt $x \in M$ heißt* Ecke, *wenn er nicht konvexe Linearkombination anderer Punkte von M ist, d.h. wenn gilt*
$$x = \lambda y + (1-\lambda) z,\ \lambda \in (0,1),\ y, z \in M \quad \Rightarrow \quad x = y = z.$$

Da die Zielfunktion linear ist, kann man sich auf die Betrachtung der Ecken von M beschränken. Diese können durch jeweils m Spalten von A beschrieben werden. Dazu werden folgende Bezeichnungen benötigt. Zu einer Indexmenge $J = \{j_1, \ldots, j_k\} \subseteq N := \{1, \ldots, n\}$ sei $|J| := k$ und
$$x_J := (x_{j_1}, \ldots, x_{j_k})^\mathsf{T}, \quad A_J := (a^{(j_1)}, \ldots, a^{(j_k)}). \tag{2.2.1}$$
Zunächst sei angenommen, daß die Komponenten bzw. Spalten nach aufsteigenden Indizes angeordnet sind. Bei der tatsächlichen Rechnung ist diese Festlegung aber i.a. nicht mehr sinnvoll. Dabei sind $a^{(j)} := Ae^{(j)}$ die Spaltenvektoren der Matrix $A = (a^{(1)}, \ldots, a^{(n)})$, mit $e^{(j)} := (\delta_{ij})_i$.

Umgekehrt wird zu einem zulässigen Punkt ($x \in M$) definiert
$$I(x) := \{i \in N : \ x_i > 0\}.$$
Eine besondere Bedeutung besitzen die regulären $m \times m$-Untermatrizen von A.

Definition 2.2.3 *Für $J = \{j_1, \ldots, j_m\} \subseteq N$ heißt A_J* Basis, *wenn A_J regulär ist. Zu A_J wird eine Basislösung $\bar{x}$ definiert durch $\bar{x}_J := A_J^{-1}b$, $\bar{x}_K = 0$, wobei $K = N \setminus J$ die komplementäre Indexmenge ist. Wenn $\bar{x}$ zulässig ist (d.h. für $A_J^{-1}b \geq 0$), heißt A_J zulässige Basis.*

Der Begriff der Basis stellt den Zusammenhang her zwischen dem geometrischen Objekt *Ecke* und der algebraischen Definition der Menge M. Die zulässigen Basislösungen sind nämlich genau die Ecken von M. Hier wird allerdings nur eine Richtung dieser Aussage gezeigt.

Satz 2.2.4 *Jede zulässige Basislösung ist Ecke von M.*

Beweis Von zentraler Bedeutung in diesem Kapitel ist die folgende *Eckendarstellung* von M, bei der M von einer Ecke her parametrisiert wird. Zur Basis A_J, $J \subseteq N$, sei $K := N \setminus J$. Für ein beliebiges $x \in M$ gilt

$$b = Ax = \sum_{j=1}^{n} a^{(j)} x_j = \sum_{j \in J} a^{(j)} x_j + \sum_{j \in K} a^{(j)} x_j = A_J x_J + A_K x_K. \tag{2.2.2}$$

Multiplikation mit A_J^{-1} führt auf

$$x_J = A_J^{-1}b - A_J^{-1}A_K x_K = \bar{x}_J - A_J^{-1}A_K\, x_K. \tag{2.2.3}$$

Hier sind die Basisvariablen x_J eines Punktes $x \in M$ in Abhängigkeit von den Nichtbasisvariablen x_K parametrisiert.

Zum Eckennachweis sei $\bar{x} = \lambda y + (1-\lambda)z$, $y, z \in M$, also insbesondere $0 = \bar{x}_K = \lambda y_K + (1-\lambda)z_K$. Da aber $y, z \geq 0$, und $\lambda, 1-\lambda > 0$ ist, folgt $y_K = z_K = 0 = \bar{x}_K$, aus (2.2.3) auch die Gleichheit der J-Komponenten, $y_J = \bar{x}_J - 0, z_J = \bar{x}_J - 0$, also ist $\bar{x} = y = z$ Ecke von M. ■

Insgesamt gibt es $\binom{n}{m}$ Untermatrizen von A der Größe $m \times m$, daher kann es höchstens so viele Ecken von M geben, also insbesondere nur endlich viele. Das im folgenden beschriebene *Simplexverfahren* arbeitet die Ecken von M zielgerichtet ab, beginnend mit einer Startecke. Wichtig ist dabei die möglichst eindeutige Zuordnung von Basen und Ecken. Daher nennt man eine Basislösung bzw. Ecke *nicht ausgeartet*, wenn $\bar{x}_J > 0$ gilt. Denn eine ausgeartete Ecke x mit $|I(x)| < m$ kann durch verschiedene Basen erzeugt werden. Dies kann im folgenden Algorithmus auf Zyklen führen, die durch zusätzliche Maßnahmen unterdrückt werden müssen (vgl. Collatz/Wetterling). Dieser (seltene) Fall wird hier ausgeschlossen durch die Forderung, daß das LON nicht ausgeartet ist, d.h., keine ausgearteten Ecken enthält.

2.3 Das Simplex-Verfahren

Das Grundprinzip ist ein schrittweises Vorgehen von einer Ausgangsecke $\bar{x}$ mit Basis A_J in eine Nachbarecke mit einem größeren Wert der Zielfunktion. Da M nur endlich viele Ecken besitzt, ist das Verfahren daher endlich.

Es sei also $\bar{x}$ eine nicht ausgeartete Ecke mit Basis A_J und $K := N \setminus J$, also $\bar{x}_J = A_J^{-1}b > 0$, $\bar{x}_K = 0$. Die Basisdarstellung (2.2.3) wird etwas abgekürzt, für jedes $x \in M$ gilt

$$x_J = \bar{x}_J - A_J^{-1}A_K x_K = \bar{x}_J + \sum_{j\in K} v_J^{(j)} x_j, \quad x_j \geq 0 \ (j \in K), \tag{2.3.1}$$

wobei $v_J^{(j)} := -A_J^{-1}a^{(j)}$, $j \in K$. Diese Darstellung (2.3.1) beschreibt aber nur solange Elemente von M wie $x_J \geq 0$ gilt, d.h.,

$$0 \leq x_i = \underbrace{\bar{x}_i}_{>0} + \sum_{j\in K} v_i^{(j)} \underbrace{x_j}_{\geq 0}, \ i \in J.$$

Für feste Indizes $i \in J$, $j \in K$, bedeutet dies

$$v_i^{(j)} \begin{cases} \geq 0 & \Rightarrow \quad \text{Forderung } x_i \geq 0 \text{ ist keine Restriktion für } x_j, \\ < 0 & \Rightarrow \quad x_j \text{ darf als Maximalwert } -\bar{x}_i/v_i^{(j)} \text{ annehmen.} \end{cases}$$

Daher führt die Vorzeichenbedingung $x \geq 0$ in (2.3.1) auf die Einschränkung

$$x_j \leq \min\{-\frac{\bar{x}_i}{v_i^{(j)}} : \ i \in J, \ v_i^{(j)} < 0\} =: \vartheta_j > 0, \tag{2.3.2}$$

wenn nur x_j vergrößert werden soll. Falls $v_J^{(j)} \geq 0$ ist, werde dabei $\vartheta_j = \infty$ gesetzt. Aus (2.3.2) ergibt sich, wie weit von der Ecke $\bar{x}$ aus in einer der Richtungen x_j, $j \in K$, gegangen werden kann ohne M zu verlassen. Zur Auswahl einer Richtung, in der die Zielfunktion ansteigt, wird diese Funktion mit (2.3.1) betrachtet:

$$\begin{aligned} c^\mathsf{T} x &= c_J^\mathsf{T} x_J + c_K^\mathsf{T} x_K = \underbrace{c_J^\mathsf{T}\bar{x}_J}_{\text{alter Wert}} + \sum_{j\in K}(c_J^\mathsf{T} v_J^{(j)} + c_j)x_j \\ &= c^\mathsf{T}\bar{x} + \sum_{j\in K}\gamma_j x_j, \qquad \gamma_j := c_J^\mathsf{T} v_J^{(j)} + c_j, \ j \in K. \end{aligned} \tag{2.3.3}$$

Die Parameter γ_j geben an, wie die Zielfunktion proportional zu x_j anwächst. Sie werden daher als *reduzierte Gewinne* (bzw. $-\gamma_j$ als reduzierte Kosten) bezeichnet. Eine Vergrößerung der Variablen x_j, $j \in K$, ist offensichtlich nur für $\gamma_j > 0$ sinnvoll. Außerdem kann an (2.3.3) folgendes abgelesen werden:

a) Wenn $\gamma_j \leq 0 \, \forall j \in K$ gilt, ist $\bar{x}$ Maximallösung.

b) Gilt für ein $j \in K$ mit $\gamma_j > 0$ wegen $\vartheta_j = \infty$ an x_j keine Einschränkung, dann gibt es kein Maximum, da $\sup\{c^\mathsf{T}x : x \in M\} = \infty$.

c) Für $\gamma_j > 0$ gibt das Produkt $\gamma_j\vartheta_j$ die in x_j-Richtung erzielbare Vergrößerung der Zielfunktion an. Als 'beste' lokale Strategie liegt hier daher die Wahl derjenigen Richtung mit maximalem Wert von $\gamma_j\vartheta_j$ nahe. Dies erfordert aber einen zu hohen Rechenaufwand, wie weiter unten überlegt wird.

Es wird nun der Vektor $x \in M$ betrachtet, der bei maximaler Vergrößerung einer einzelnen Komponente $x_j : \ 0 \to \vartheta_j$ mit $\gamma_j > 0$ entsteht,

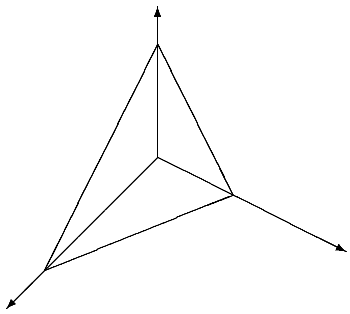
$$x_J^{(1)} := \bar{x}_J + v_J^{(j)} \vartheta_j, \quad x_K^{(1)} := e_K^{(j)} \vartheta_j.$$

Wenn $r \in J$ derjenige Index ist, in dem das Minimum in (2.3.2) angenommen wird, so verschwindet nach Konstruktion genau die Komponente $x_r^{(1)} = 0$. Die Indexmenge $I(x)$ der nichtverschwindenden Komponenten ändert sich beim Übergang zu $x^{(1)}$ daher gemäß

$$I(x^{(1)}) = I(\bar{x}) \setminus \{r\} \cup \{j\}. \tag{2.3.4}$$

Daher ist $x^{(1)}$ eine Nachbarecke von $\bar{x}$. Daß dazu $A_{I(x^{(1)})}$ auch eine Basis ist, wird später gezeigt. Durch die Beziehung (2.3.4) zwischen den Indexmengen kann aber auch umgekehrt das Verfahren vollständig beschrieben werden.

Beispiel, $n = 3$: **1)** $m = 1$:



$$\begin{aligned} 2x_1 + 3x_2 + 2x_3 &= 6, \quad x_i \geq 0 \\ x_1 + x_2 &= \max \end{aligned}$$

Basen: $J = \{1\}, \bar{x} = (3,0,0)^T$, $J = \{2\}, \bar{x} = (0,2,0)^T$, $J = \{3\}, \bar{x} = (0,0,3)^T$.

2) $m = 2$,

$$\begin{aligned} \begin{pmatrix} 0 & -1 & 2 \\ 1 & 1 & 0 \end{pmatrix} x &= \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \quad x_i \geq 0 \\ x_1 + x_3 &= \max \end{aligned}$$

Mögliche Basis-Indexmengen: $\{1,2\}$, $\{1,3\}$, $\{2,3\}$.

a) $J = \{1,2\}$, $K = \{3\}$, also $\bar{x}_3 = 0$,

$$A_J \bar{x}_J = \begin{pmatrix} 0 & -1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \end{pmatrix} \Rightarrow \bar{x}_J = \begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \end{pmatrix} = \begin{pmatrix} 4 \\ \boxed{-2} \end{pmatrix} \quad \text{unzulässig!}$$

b) $J = \{1,3\}$, $K = \{2\}$, also $\bar{x}_2 = 0$,

$$A_J \bar{x}_J = \begin{pmatrix} 0 & 2 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \bar{x}_1 \\ \bar{x}_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \end{pmatrix} \Rightarrow \bar{x}_J = \begin{pmatrix} \bar{x}_1 \\ \bar{x}_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \ c^T \bar{x} = 3.$$

c) $J = \{2,3\}$, $K = \{1\}$, also $\bar{x}_1 = 0$,

$$A_J \bar{x}_J = \begin{pmatrix} -1 & 2 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \bar{x}_2 \\ \bar{x}_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \end{pmatrix} \quad \bar{x}_J = \begin{pmatrix} \bar{x}_2 \\ \bar{x}_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \ c^T \bar{x} = 2.$$

Basisdarstellungen von M:

b) $J = \{1,3\}$, $\bar{x}_J = \begin{pmatrix}2\\1\end{pmatrix}$: $\begin{pmatrix}0 & 2\\1 & 0\end{pmatrix}\begin{pmatrix}x_1-2\\x_3-1\end{pmatrix} = -\begin{pmatrix}-1\\1\end{pmatrix}x_2 \Rightarrow \begin{pmatrix}x_1\\x_3\end{pmatrix} = \begin{pmatrix}2-x_2\\1+\frac{1}{2}x_2\end{pmatrix} =$ $\bar{x}_J + v_J^{(2)}x_2 \overset{!}{\geq} 0$, d.h., $x_2 \leq 2 = \vartheta_2$. Wegen $\gamma_2 = c_2 + c_1v_1^{(2)} + c_3v_3^{(2)} = -\frac{1}{2} \leq 0$ ist das Optimum hier.

c) $J = \{2,3\}$, $\bar{x}_J = \begin{pmatrix}2\\2\end{pmatrix}$: $\begin{pmatrix}-1 & 2\\1 & 0\end{pmatrix}\begin{pmatrix}x_2-2\\x_3-2\end{pmatrix} = -\begin{pmatrix}0\\1\end{pmatrix}x_1 \Rightarrow \begin{pmatrix}x_2\\x_3\end{pmatrix} = \begin{pmatrix}2-x_1\\2-\frac{1}{2}x_1\end{pmatrix} =$ $\bar{x}_J + v_J^{(1)}x_1 \overset{!}{\geq} 0$, d.h., $x_1 \leq \vartheta_1 = \min\{\frac{2}{1}, 1/(\frac{1}{2})\} = 2$. Wegen $\gamma_1 = c_1 + c_2v_2^{(1)} + c_3v_3^{(1)} = \frac{1}{2} > 0$ kann Zielfunktion um $\gamma_1\vartheta_1 = \frac{1}{2}2 = 1$ verbessert werden. Die neue Ecke ist $x^{(1)} = (2,0,1)^T$ mit $J = \{1,3\}$, vgl. b).

Praktische Durchführung

Die Berechnung aller Vektoren $v^{(j)}$ in (2.3.1) ist recht aufwendig, die Multiplikation $A_J^{-1}A_K$ kostet ca. $2m^2(n-m)$ Operationen, und ist im Falle $\gamma_j \leq 0$ auch überflüssig. Der Gewinne-Vektor γ kann vielmehr ohne Kenntnis der $v^{(j)}$ berechnet werden:

$$\gamma_j = c_j - \underbrace{(c_J^\mathsf{T}A_J^{-1})}_{p^\mathsf{T}}a^{(j)} = c_j - p^\mathsf{T}a^{(j)},\ j \in K.$$

Anhand dieser Werte läßt sich feststellen, welche $v^{(j)}$ überhaupt benötigt werden zu Bestimmung des lokal besten Schritts mit $\max_j \gamma_j\vartheta_j$. In der Praxis entscheidet man sich für irgendeine Richtung mit $\gamma_j > 0$ und berechnet nur das zugehörige $v^{(j)}$. Die jetzt beschriebene Verfahrensvariante wird in der Literatur das *revidierte Simplexverfahren* genannt.

In einer Ausgangsecke $x^{(0)} \in M$ wird die zugehörige Basis $A_J,\ J = I(x^{(0)})$, benötigt. Damit kann folgendes Verfahren durchgeführt werden.

Algorithmus 2.3.1 Simplex-Verfahren

Eingabe:	Zulässige Basis $A_J,\ J \subseteq N$	
Schritt 1	$x_J := A_J^{-1}b,\ K := N \setminus J,\ x_K := 0$	
2	berechne $p^\mathsf{T} := c_J^\mathsf{T}A_J^{-1},\ \gamma_j := c_j - p^\mathsf{T}a^{(j)},\ j \in K.$	
3	wenn $\gamma_j \leq 0\,\forall j \in K\ \Rightarrow$	**STOP**, Optimum!
4	wähle $s \in K$ mit $\gamma_s > 0$, berechne $v_J^{(s)} := -A_J^{-1}a^{(s)}$.	
5	wenn $v_i^{(s)} \geq 0\,\forall i \in J\ \Rightarrow$	**STOP**, unbeschränkt!
6	Bestimme $r \in J$: $-x_r/v_r^{(s)} = \vartheta_s = \min\{-x_i/v_i^{(s)}\ :\ v_i^{(s)} < 0, i \in J\}$	
7	$J := J \setminus \{r\} \cup \{s\}$, gehe nach 1	

Einige Eigenschaften des Verfahrens enthält

Satz 2.3.2 *Das LON sei nicht ausgeartet, $x^{(0)} \in M$ eine Ecke mit Basis A_J.*

a) Wenn in Schritt 3 alle $\gamma_j \leq 0$ sind, dann ist $x^{(0)}$ Optimallösung.

b) Es existiert keine endliche Lösung, wenn mit einem $\gamma_s > 0$ auch $v_J^{(s)} \geq 0,\ s \in K$.

c) Wenn in Schritt 6 gilt $\vartheta_s = -x_r/v_r^{(s)} < \infty$, dann wird mit $J^{(1)} := J \setminus \{r\} \cup \{s\}$ durch $A_{J^{(1)}}$ eine zulässige Basis definiert. Für die zugehörige Ecke $x^{(1)}$ ist $x_r^{(1)} = 0,\ x_s^{(1)} = \vartheta_s > 0$.

Beweis Die Aussagen a) und b) wurden schon im Anschluß an (2.3.3) behandelt. Zum Nachweis von c) wird der Strahl $x(\vartheta)$ betrachtet, der sich bei Variation von $x_s \in [0, \vartheta_s]$ ergibt, d.h.,

$$x_J(\vartheta) := x_J^{(0)} + \vartheta v_J^{(s)}, \quad x_K(\vartheta) := \vartheta e_K^{(s)}.$$

Offensichtlich ist $x(\vartheta_s) = x^{(1)}$, insbesondere $x_s^{(1)} = x_s(\vartheta_s) = \vartheta_s$, und

$$x_r^{(1)} = x_r^{(0)} + \vartheta_s v_r^{(s)} = x_r^{(0)} + (-x_r^{(0)}/v_r^{(s)})v_r^{(s)} = 0.$$

Dies bestätigt die Aussage zu $J^{(1)}$. Das eigentliche Problem ist nun die Regularität von $A_{J^{(1)}}$. Nach Schritt 7 wird bei $A_{J^{(1)}}$ die Spalte $a^{(r)}$ durch $a^{(s)}$ ersetzt. Bei geeigneter Einordnung ist

$$\begin{aligned} A_{J^{(1)}} &= A_J + (a^{(s)} - a^{(r)}){e^{(r)}}^\mathsf{T} \quad \iff \\ A_J^{-1}A_{J^{(1)}} &= I + A_J^{-1}(a^{(s)} - a^{(r)}){e^{(r)}}^\mathsf{T}. \end{aligned} \tag{2.3.5}$$

Eine Matrix der Gestalt $I + u{e^{(r)}}^\mathsf{T}$ ist regulär für ${e^{(r)}}^\mathsf{T} u + 1 = u_r + 1 \neq 0$. Denn für beliebiges $y \in \mathbb{R}^n$ folgt aus

$$0 \stackrel{!}{=} (I + u{e^{(r)}}^\mathsf{T})y = y + uy_r$$

durch Betrachtung der r-ten Komponente zunächst $y_r = 0$ (für $1 + u_r \neq 0$) und damit $y = 0$. Also ist $I + u{e^{(r)}}^\mathsf{T}$ tatsächlich regulär. Speziell bei (2.3.5) lautet diese Regularitätsbedingung

$$0 \neq 1 + {e^{(r)}}^\mathsf{T} A_J^{-1}(a^{(s)} - a^{(r)}) = 1 - v_r^{(s)} - 1 = -v_r^{(s)},$$

da $A_J^{-1}a^{(r)} = e^{(r)}$. Da nach Konstruktion $v_r^{(s)} > 0$ gilt, ist $A_{J^{(1)}}$ regulär und damit Basis. ■

Da die endlich vielen Ecken von M mit streng wachsender Zielfunktion durchlaufen werden, gilt

Satz 2.3.3 *Für ein nicht ausgeartetes Lineares Optimierungsproblem in Normalform erzeugt das Simplexverfahren, ausgehend von einer Startecke, eine endliche Folge von Ecken mit streng wachsender Zielfunktion. Das Verfahren endet mit einer optimalen Ecke oder der Aussage, daß eine solche nicht existiert.*

Praktische Durchführung des Basiswechsels

In der Diskussion wurde bisher noch nicht auf die Auflösung der drei Gleichungssysteme

$$A_J x_J = b, \quad p^\mathsf{T} A_J = c_J^\mathsf{T}, \quad A_J v_J^{(s)} = -a^{(s)},$$

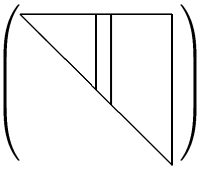
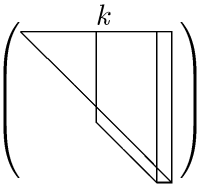
eingegangen, die in jedem Schritt des Simplexverfahrens auftreten. Betrachtete man jeden Schritt isoliert, wäre hierzu ein Aufwand von $O(m^3)$ Operationen notwendig. Allerdings unterscheiden sich die Basen $A_{J^{(0)}}$ und $A_{J^{(1)}}$ aus aufeinanderfolgenden Schritten nur wenig, da $J^{(1)} =$

$J^{(0)} \setminus \{r\} \cup \{s\}$, vgl. (2.3.5). Dieser Zusammenhang kann auf verschiedene Weise ausgenutzt werden. Beim (für Handrechnung) üblichen Tabellenverfahren werden die Inversen der Basismatrizen mitgeführt und ein Austauschschritt von $A^{-1}_{J^{(0)}}$ zu $A^{-1}_{J^{(1)}}$ verwendet. Hierbei können sich allerdings Rundungsfehler in unkontrollierter Weise ausbreiten, z.B., wenn eine der auftretenden Basismatrizen schlecht konditioniert ist.

Alternativ kann eine QR-Zerlegung der Basismatrizen mitgeführt und bei jedem Basiswechsel angepaßt werden. Dieses Verfahren ist i.a. numerisch stabiler. Abweichend von der früheren Vereinbarung wird jetzt die Freiheit in der Zuordnung der Basisindizes auf Spaltenpositionen der Matrizen A_J ausgenutzt. Daher werden die Indizes der Mengen J als m-Tupel geschrieben, d.h. angeordnet. Beim Austausch der Spalte $a^{(r)}$ durch $a^{(s)}$ in der Form

$$\begin{array}{ccc} J^{(0)} \sim (j_1, \dots, j_k = r, \dots, j_m) & \mapsto & J^{(1)} \sim (j_1, \dots, j_{k-1}, j_{k+1}, \dots, j_m, s) \\ A_{J^{(0)}} = \begin{pmatrix} \boxed{} & \begin{matrix} a_1^{(r)} \\ \vdots \\ a_m^{(r)} \end{matrix} & \boxed{} \end{pmatrix} & \mapsto & A_{J^{(1)}} = \begin{pmatrix} \boxed{} & \boxed{} & \begin{matrix} a_1^{(s)} \\ \vdots \\ a_m^{(s)} \end{matrix} \end{pmatrix} \end{array}$$

wird die k-te Spalte von $A_{J^{(0)}}$ gestrichen und $a^{(s)}$ als letzte Spalte neu eingefügt. Bei bekannter QR-Zerlegung von $A_{J^{(0)}} = QR_0$ ergibt sich nach dieser Operation im R-Faktor folgende Struktur:

$$R_0 = Q^{\mathsf{T}} A_{J^{(0)}} =$$  $$\mapsto \quad Q^{\mathsf{T}} A_{J^{(1)}} =$$  .

In der letzten Spalte von $Q^{\mathsf{T}} A_{J^{(1)}}$ steht als neuer Vektor $Q^{\mathsf{T}} a^{(s)}$, ab Spalte k wurde jeweils ein neues Subdiagonalelement eingeführt. Diese $m - k$ Elemente können mit ebenso vielen unitären Zeilenumformungen (2×2-Rotationen) wieder eliminiert werden. Daher benötigt ein einzelner Simplexschritt tatsächlich nur $O(m^2)$ Operationen.

Bestimmung einer Startecke

Als Ausgangspunkt des Simplexverfahrens wird eine Ecke $x^{(0)} \in M$ benötigt. Eine solche kann bei einer speziellen Klasse von Problemen explizit angegeben werden. Das Optimierungsproblem

$$By \le b, \ y \ge 0, \quad d^{\mathsf{T}} y \stackrel{!}{=} \max,$$

mit $B \in \mathbb{R}^{m \times k}$ geht nach Einführung von Schlupfvariablen $z \in \mathbb{R}^m$ über in die Normalform

$$By + z = b, \ y \ge 0, z \ge 0, \quad (d^{\mathsf{T}}, 0^{\mathsf{T}}) \begin{pmatrix} y \\ z \end{pmatrix} \stackrel{!}{=} \max .$$

In den Bezeichnungen von (2.1.1) gilt also

$$A = (B, I) \in \mathbb{R}^{m \times (k+m)}, \ x = \begin{pmatrix} y \\ z \end{pmatrix}, \ c = \begin{pmatrix} d \\ 0 \end{pmatrix}.$$

Für $b \geq 0$ ist hier der Vektor $x^{(0)} = \begin{pmatrix} 0 \\ b \end{pmatrix}$ eine zulässige Ecke mit Basis

$$A_J = I_m, \quad J = \{k+1, \ldots, k+m\}.$$

Diese Eigenschaft läßt sich auch bei allgemeinen Problemen zur Konstruktion einer Startecke im Rahmen einer *Anlaufrechnung* verwenden. Dazu betrachtet man folgendes Hilfsproblem für $u \in \mathbb{R}^{n+m}$:

$$(A, I_m)u = b, \ u \geq 0, \quad -u_{n+1} - \ldots - u_{n+m} \stackrel{!}{=} \max. \tag{2.3.6}$$

Der Punkt $u^\mathsf{T} = (0, b^\mathsf{T})$ ist eine zulässige Ecke von (2.3.6), wenn $b \geq 0$ gilt. Diese Vorzeichenbedingung kann beim LON durch Multiplikation der i-ten Gleichung mit $\mathrm{sign} b_i$ erreicht werden. Eine Optimallösung von (2.3.6) existiert, da die Zielfunktion durch Null nach oben beschränkt ist. Der Wert der Hilfs-Zielfunktion in der Optimallösung $\hat{u}$ von (2.3.6) liefert dann folgende Information

$$-\hat{u}_{n+1} - \ldots - \hat{u}_{n+m} \begin{cases} = 0 & : \quad (\hat{u}_1, \ldots, \hat{u}_n)^\mathsf{T} \text{ ist Ecke von } M, \\ < 0 & : \quad M = \emptyset. \end{cases}$$

Mit Anlaufrechnung kann das Simplexverfahren das LON also vollständig behandeln.

Fazit: *Das Simplexverfahren liefert entweder in endlich vielen Schritten eine Lösung des Linearen Optimierungsproblems, oder eine Begründung dafür, weshalb eine solche nicht existiert.*

Zum *Rechenaufwand:* Obwohl Beispiele konstruiert werden können, bei denen das Simplexverfahren alle $\binom{n}{m}$ Ecken des Simplex M absucht und daher einen exponentiellen Aufwand besitzt, tritt diese Situation in praktischen Problemen kaum auf, dort arbeitet das Verfahren sehr effizient. Neuere theoretische Ergebnisse beweisen, daß das LON mit polynomiell wachsendem Aufwand gelöst werden kann, die auf dem konstruktiven Beweisprinzip beruhenden Verfahren sind allerdings (noch) keine Konkurrenz für das Simplexverfahren.

Seit einiger Zeit werden auch sog. Innere-Punkte-Verfahren untersucht, die Ähnlichkeiten mit den in §5 behandelten Fortsetzungsverfahren besitzen.

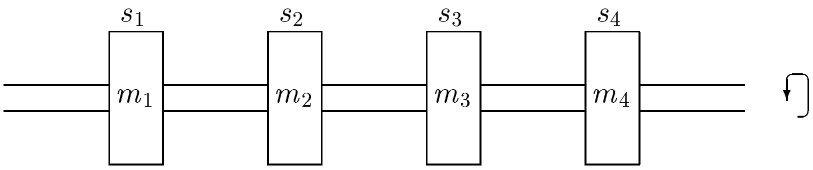
3 Eigenwertprobleme von Matrizen

Bei Systemen von gewöhnlichen Differentialgleichungen mit konstanter Koeffizientenmatrix $A \in \mathbb{R}^{n\times n}$ führt der Standardansatz

$$y(t) = e^{\lambda t} y_0 \quad \text{bei} \quad y'(t) = Ay(t) \;\Rightarrow\; Ay_0 = \lambda y_0.$$

auf ein Eigenwertproblem. Analog lassen sich lineare Schwingungsprobleme in der Mechanik (Dgln zweiter Ordnung) in der Regel über Eigenwertprobleme behandeln. Einen einfacheren Hintergrund hat folgendes Beispiel:

Generatorturbinen werden mit extrem hohen Drehzahlen betrieben. Eine Turbine mit n Schaufelrädern kann als biegsame Achse mit n an den Stellen s_j aufgesetzten Gewichten (Massen m_j) modelliert werden.



Wenn eine Einheitskraft, die an der Stelle s_j wirkt, an einer Stelle s_i die Auslenkung b_{ij} verursacht und die Auslenkungen zu verschiedenen Kräften sich linear überlagern, ergibt sich die Maximal-Drehzahl $\omega_{\max}$ aus folgender Überlegung. Eine Auslenkung der Größe x_j an der Stelle s_j erzeugt bei Rotation mit Drehzahl ω die radiale Kraft $m_j\omega^2 x_j$. Die Achse wird dann zerstört, wenn die sich überlagernden Kräfte insgesamt die bestehende Auslenkung aufrecht erhalten:

$$\left\{ \begin{array}{rcl} x_1 & = & \omega^2(b_{11}m_1x_1 + \ldots + b_{1n}m_nx_n) \\ & & \cdots \\ x_n & = & \omega^2(b_{n1}m_1x_1 + \ldots + b_{nn}m_nx_n) \end{array} \right. ,$$

d.h.,

$$x = \omega^2 Ax, \quad A := (b_{ij}m_j)_{i,j}.$$

3.1 Grundlagen, Matrix-Normalformen

Definition 3.1.1 *Gegeben sei eine Matrix $A \in \mathbb{C}^{n\times n}$. Ein Wert $\lambda \in \mathbb{C}$ heißt* Eigenwert *von A, wenn ein $x \in \mathbb{C}^n$ existiert mit*

$$Ax = \lambda x, \qquad x \neq 0. \tag{3.1.1}$$

Der Vektor x heißt (Rechts-) Eigenvektor *von A zum Eigenwert λ, der Nullraum $N(A-\lambda I) = \{x : Ax = \lambda x\}$ Eigenraum, seine Dimension die (geometrische) Vielfachheit des Eigenwerts ('Anzahl der linear unabhängigen Eigenvektoren').*

Die Eigenwerte von A sind Nullstellen des *charakteristischen Polynoms*

$$\begin{aligned} p(z) &= \det(A - zI) = (-1)^n[z^n + b_{n-1}z^{n-1} + \ldots + b_0] \qquad (3.1.2) \\ &= (-1)^n(z-\lambda_1)^{\gamma_1}(z-\lambda_2)^{\gamma_2}\cdots(z-\lambda_k)^{\gamma_k}. \end{aligned}$$

Die Vielfachheit γ_j der Nullstelle λ_j heißt *algebraische Vielfachheit* von λ_j. Außer bei Vielfachheit $\gamma_1 = 1$ kann die geometrische Vielfachheit kleiner sein als die algebraische:

Beispiel 3.1.2 Es sei

$$C_m(\lambda) := \begin{pmatrix} \lambda & 1 & & 0 \\ & \lambda & 1 & \\ & & \ddots & \ddots \\ & & & \lambda \end{pmatrix} \in \mathbb{C}^{m\times m}. \tag{3.1.3}$$

Das charakteristische Polynom ist $p(z) = (\lambda - z)^m$, λ ist also m-facher algebraischer Eigenwert von $C_m(\lambda)$. Die Lösungen des Gleichungssystems

$$\Big(C_m(\lambda) - \lambda I\Big)x = \begin{pmatrix} x_2 \\ \vdots \\ x_m \\ 0 \end{pmatrix} = 0$$

besitzen die Form $x = ce^{(1)}$, $c \in \mathbb{C}$, die geometrische Vielfachheit von λ ist also nur eins.

Definition 3.1.3 *Für $\lambda \in \mathbb{C}$, $k \in \mathbb{N}$ und $x \in \mathbb{C}^n, x \neq 0$, gelte*

$$(A - \lambda I)^k x = 0, \quad \textit{aber } (A - \lambda I)^{k-1} x \neq 0. \tag{3.1.4}$$

Dann heißt x Hauptvektor vom Grad k zum Eigenwert λ.

Satz 3.1.4 *Haupt- und Eigenvektoren zu verschiedenen Eigenwerten sind linear unabhängig. Hauptvektoren verschiedenen Grades sind linear unabhängig. Wenn $x^{(k)}$ ein Hauptvektor vom Grad k ist, dann ist $x^{(k-j)} := (A - \lambda I)^j x^{(k)}$ ein Hauptvektor vom Grad $k - j$. Der Unterraum*

$$U := [x^{(k)}, (A - \lambda I)x^{(k)}, \ldots, (A - \lambda I)^{k-1} x^{(k)}]$$

ist ein invarianter Unterraum *von A, d.h., es gilt $AU \subseteq U$.*

Ein Basiswechsel im $\mathbb{C}^n$ entspricht einer *Ähnlichkeitstransformation* der Matrix A mit einer regulären Matrix $T \in \mathbb{C}^{n\times n}$,

$$A \ \mapsto \ TAT^{-1} =: B. \tag{3.1.5}$$

B besitzt offensichtlich die gleichen Eigenwerte wie A, während die Haupt- und Eigenvektoren transformiert werden,

$$x \text{ ist Haupt-/Eigenvektor von } A \iff Tx \text{ ist Haupt-/Eigenvektor von } B.$$

Ähnlichkeitstransformationen übertragen sich auf Funktionen von A, es gilt etwa

$$(TAT^{-1})^{-1} = T(A^{-1})T^{-1}, \quad (TAT^{-1})^k = T(A^k)T^{-1}.$$

Von besonderem Interesse sind Ähnlichkeitstransformationen, die eine Matrix auf einfache Gestalt bringen, an der wesentliche Eigenschaften ablesbar sind.

Satz 3.1.5 (Jordan-Normalform) *Zu jeder Matrix $A \in \mathbb{C}^{n\times n}$ gibt es eine reguläre Matrix X so, daß $A = XJX^{-1}$ ist mit*

$$J = \begin{pmatrix} C_{m_1}(\lambda_1) & & 0 \\ & \ddots & \\ 0 & & C_{m_k}(\lambda_k) \end{pmatrix} = \begin{pmatrix} \lambda_1 & 1 & & & & & & \\ & \ddots & \ddots & & & & & \\ & & \lambda_1 & 1 & & & & \\ & & & \lambda_1 & 0 & & & \\ & & & & \lambda_2 & 1 & & \\ & & & & & \ddots & \ddots & \end{pmatrix}, \tag{3.1.6}$$

mit $m_1 + \ldots + m_k = n$. Eigenwerte in verschiedenen Jordanblöcken C_{m_j} können dabei übereinstimmen. Die Spalten von X enthalten die (Rechts-) Haupt- und Eigenvektoren von A, $X = (x^{(1)}, \ldots, x^{(n)})$. Die Zeilen von X^{-1} sind die Links- Haupt- bzw. Eigenvektoren, $(y^{(1)}, \ldots, y^{(n)})^* = X^{-1}$, daher sind diese zueinander orthogonal, ${y^{(i)}}^* x^{(j)} = \delta_{ij}$.

Eine Matrix, bei der alle Jordanblöcke die Größe eins haben (J hat in der Nebendiagonale keine Einsen), heißt *diagonalisierbar*. Sie besitzt also n linear unabhängige Eigenvektoren. Jede Matrix mit nur algebraisch einfachen Eigenwerten ist offensichtlich diagonalisierbar.

Ungeachtet ihrer theoretischen Bedeutung wird die Jordan-Normalform bei numerischen Berechnungen kaum verwendet, da sie sehr empfindlich auf Störungen der Matrix reagiert. So ist etwa die Matrix

$$C_2(1) = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$

in Jordan-Normalform. Schon bei einer beliebig kleinen Änderung in einem der Hauptdiagonalelemente wird die Matrix aber diagonalisierbar und ihre Jordan-Normalform hat eine Null über der Diagonalen, weicht also dort um den Wert eins von $C_2(1)$ ab. Eine 'stabilere' Normalform ist die folgende, bei der Orthonormalbasen zugrundegelegt werden.

Satz 3.1.6 (Schur-Normalform) *Jede Matrix $A \in \mathbb{C}^{n\times n}$ kann durch eine unitäre Matrix U, $U^* = U^{-1}$, auf obere Dreieckgestalt transformiert werden,*

$$A = USU^*, \quad S = \begin{pmatrix} \lambda_1 & s_{12} & \ldots & s_{1n} \\ & \lambda_2 & & s_{2n} \\ & & \ddots & \vdots \\ & & & \lambda_n \end{pmatrix}. \tag{3.1.7}$$

In der Hauptdiagonalen von S stehen die Eigenwerte (in beliebiger Reihenfolge).

Die Reihenfolge der Eigenwerte ist zwar beliebig, im folgenden wird zur Standardisierung der Bezeichnungen aber oft eine bestimmte vorausgesetzt, z.B., $|\lambda_1| \geq |\lambda_2| \geq \ldots |\lambda_n|$.

Beweis Mit einer QR-Zerlegung der Matrix $X = UR$ aus der Jordan-Normalform zeigt sich

$$A = XJX^{-1} = U(RJR^{-1})U^*.$$

Da $RJR^{-1} =: S$ als Produkt oberer Dreieckmatrizen wieder obere Dreieckmatrix ist, deren Hauptdiagonalelemente die von J sind, folgt die Behauptung. ∎

Eine besonders einfache Normalform besitzen normale Matrizen.

Definition 3.1.7 *Eine Matrix A heißt* normal, *wenn gilt*

$$A^*A = AA^*. \tag{3.1.8}$$

Spezielle normale Matrizen sind die reell symmetrischen ($A^\mathsf{T} = A$), die hermiteschen ($A^* = A$), schief-hermiteschen ($A^* = -A$) und unitären ($A^* = A^{-1}$).

Satz 3.1.8 *a) Eine Matrix ist genau dann normal, wenn sie unitär diagonalisierbar ist.*

b) Eine normale Matrix besitzt eine Orthonormalbasis aus Eigenvektoren.

c) Die Eigenwerte einer hermiteschen Matrix sind reell, die einer schief-hermiteschen rein imaginär, die einer unitären liegen auf dem Einheitskreis.

Beweis Trivial bei Einsatz der Schur-Normalform.

Für quantitative Aussagen werden meist Normabschätzungen herangezogen. In der Spektralnorm $\|.\|_2$ ergibt sich für die Schur-Normalform

$$\|A\|_2 = \|USU^*\|_2 = \|S\|_2. \tag{3.1.9}$$

Für normale Matrizen ist daher $\|A\|_2 = \varrho(A) = \max_{i=1}^n |\lambda_i|$. Für allgemeine Matrizen sei an Satz 2.2.1 aus der Numerik I erinnert:

Satz 3.1.9 *a) Für jede Matrixnorm $\|\cdot\|$ gilt $\rho(A) \le \|A\|$.*

b) Für jede Matrix A und jedes $\varepsilon > 0$ existiert eine (spezielle) Norm mit

$$\|A\|_M \le \rho(A) + \varepsilon. \tag{3.1.10}$$

c) Für diagonalisierbare Matrizen A kann in (3.1.10) $\varepsilon = 0$ gewählt werden.

Beispiel 3.1.10 Die wichtigsten Verfahren zur Eigenwert-Bestimmung verwenden (implizit) hohe Potenzen A^k von A. Bei

$$C_2(\lambda) = \begin{pmatrix} \lambda & 1 \\ 0 & \lambda \end{pmatrix} \quad \text{ist} \quad C_2^k = \begin{pmatrix} \lambda^k & k\lambda^{k-1} \\ 0 & \lambda^k \end{pmatrix}.$$

Hier sieht man, daß für $\lambda \ne 0$ die Norm nicht durch $|\lambda|^k$ beschränkt ist, mit $x = e^{(2)}$ etwa gilt

$$\frac{\|C_2^k x\|_2}{(|\lambda| + \varepsilon)^k} = \frac{|\lambda^{k-1}|\sqrt{k^2 + |\lambda|^2}}{(|\lambda| + \varepsilon)^k} = \frac{\sqrt{1 + k^2/|\lambda|^2}}{(1 + \varepsilon/|\lambda|)^k} \to \begin{cases} \infty, & \varepsilon = 0, \\ 0, & \varepsilon > 0, \end{cases} \quad (k \to \infty).$$

In der Schranke aus Satz 3.1.9 ist also $\varepsilon > 0$ erforderlich.

3.2 Reduktion auf einfachere Gestalt

Die wichtigsten Verfahren zur Behandlung des Eigenwertproblems erzeugen iterativ eine Ähnlichkeitstransformation der Matrix auf obere Dreieck- oder Diagonalgestalt. Der Rechenaufwand pro Iterationsschritt kann sich dabei ganz erheblich verringern, wenn die Struktur der Ausgangsmatrix nahe bei der Zielgestalt ist.

Definition 3.2.1 *Die Matrix $A = (a_{ij}) \in \mathbb{C}^{n\times n}$ ist eine*

$$\begin{array}{lcl} \textit{Hessenberg-Matrix} & \iff & a_{ij} = 0 \textit{ für } \quad j < i-1, \\ \textit{Tridiagonal-Matrix} & \iff & a_{ij} = 0 \textit{ für } \quad |i-j| > 1. \end{array}$$

Ausführlich:

$$\begin{pmatrix} a_{11} & a_{12} & \cdot & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdot & & a_{2n} \\ 0 & a_{32} & a_{33} & & \\ & & \ddots & \ddots & \vdots \\ 0 & & & a_{n,n-1} & a_{nn} \end{pmatrix} \qquad \begin{pmatrix} a_{11} & a_{12} & 0 & \cdots & 0 \\ a_{21} & a_{22} & a_{23} & & \\ & \ddots & \ddots & \ddots & \\ 0 & & \cdot & \cdot & a_{n-1,n} \\ 0 & \cdots & 0 & a_{n,n-1} & a_{nn} \end{pmatrix}$$

Hessenberg-Form   Tridiagonal-Gestalt

Satz 3.2.2 *Zu jeder Matrix $A \in \mathbb{C}^{n\times n}$ gibt es eine unitäre Matrix U so, daß $B := U^*AU$ Hessenberg-Form besitzt. U kann als Produkt von Householder-Spiegelungen (1.3.2) dargestellt werden, $U^* = H^{(n-2)} \cdots H^{(1)}$. Der Rechenaufwand für diese Transformation beträgt $\frac{10}{3}n^3$ Operationen. Wenn A symmetrisch ist, dann ist dies auch B, also tridiagonal. Der Aufwand reduziert sich in diesem Fall auf $\frac{4}{3}n^3$ Operationen.*

Beweis durch Konstruktion: Der Unterschied zum Vorgehen bei der QR-Zerlegung ist die Einschränkung, daß auf $A =: A^{(1)}$ ausschließlich Ähnlichkeitstransformationen anzuwenden sind,

$$A^{(j+1)} := H^{(j)}A^{(j)}H^{(j)}, \quad H^{(j)} = I - 2u^{(j)}u^{(j)*}. \tag{3.2.1}$$

Damit bei der Rechts-Multiplikation mit $H^{(j)}$ die gerade eliminierte j-te Spalte von $H^{(j)}A^{(j)}$ nicht wieder aufgefüllt wird, ist auch $u_j^{(j)} = 0$ zu wählen. Es sei jetzt

$$A^{(j)} = \begin{pmatrix} & & & & \\ & a_{j+1,j}^{(j)} & * & \cdots & * \\ & \vdots & \vdots & & \vdots \\ & a_{nj}^{(j)} & * & \cdots & a_{nn}^{(j)} \end{pmatrix} \begin{array}{l} \}\, j \\ \\ \}\, n-j \end{array} \in \mathbb{C}^{n\times n}$$

Nach Lemma 1.3.2 wird eine Spiegelung $H^{(j)}$ mit $u^{(j)} = (0, \dots, 0, u_{j+1}^{(j)}, \dots, u_n^{(j)})^\mathsf{T}$ bestimmt, d.h., mit der Struktur

$$H^{(j)} = \left(\begin{array}{c|c} I_j & 0 \\ \hline 0 & \tilde{H}^{(j)} \end{array} \right) \quad \text{so, daß} \quad \tilde{H}^{(j)} \begin{pmatrix} a_{j+1,j}^{(j)} \\ \vdots \\ a_{nj}^{(j)} \end{pmatrix} = \begin{pmatrix} \bullet \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Dann wird diese j-te Spalte bei Rechtsmultiplikation $A^{(j+1)} = \left(H^{(j)} A^{(j)} \right) H^{(j)}$ nicht wieder verändert. Nach $n-2$ Schritten ist $A^{(n-1)} =: B$ in Hessenbergform. Die Vektoren können wieder, i.w., in den freien Plätzen von B untergebracht werden.

Aufwand: Wie bei Satz 1.3.3 sind nur die Matrix-Umformungen relevant. Die erste,

$$\tilde{A}^{(j)} := H^{(j)} A^{(j)} = A^{(j)} - \underbrace{2u^{(j)} \cdot}_{2(n-j)^2} [\underbrace{u^{(j)^*} A^{(j)}}_{2(n-j)^2}]$$

benötigt $4(n-j)^2$ Operationen, die zweite

$$A^{(j+1)} = \tilde{A}^{(j)} H^{(j)} = \tilde{A}^{(j)} - \underbrace{2[\tilde{A}^{(j)} u^{(j)}]}_{2n(n-j)} \underbrace{\cdot u^{(j)^*}}_{2(n-j)n}$$

$4n(n-j)$ Operationen, da $\tilde{A}^{(j)} u^{(j)}$ vollbesetzt ist. Der Gesamtaufwand ist somit

$$4 \sum_{j=1}^{n-2} (n-j)^2 + 4 \sum_{j=1}^{n-2} n(n-j) = \frac{4}{3} n^3 + 2n^3 + \ldots = \frac{10}{3} n^3 + \mathcal{O}(n^2).$$

Die Symmetrie von B folgt aus der von A: $A = A^* \Rightarrow B^* = U^* A^* U = U^* A U = B$. ■

Bemerkung: A kann auch durch eine untere Dreieckmatrix L auf Hessenbergform LAL^{-1} transformiert werden. Diese Methode erfordert nur den halben Rechenaufwand der unitären Reduktion und kann mit Hilfe von Pivotisierungen auch numerisch stabil durchgeführt werden. Allerdings führt dieses Verfahren nicht auf die Schur-Normalform von A bei Anwendung der später folgenden Verfahren (QR-Iteration).

3.3 Das unsymmetrische Eigenwertproblem

Die hier behandelten Verfahren, Vektor-, orthogonale und QR-Iteration, arbeiten nach einem gemeinsamen Prinzip, einer Verallgemeinerung der zunächst behandelten Vektoriteration. Da in der Praxis unterschiedliche Fragestellungen auftreten können, sind alle drei Varianten von Interesse. Das allgemeinste Verfahren ist die QR-Iteration, mit der die volle Schur-Normalform (alle Eigenwerte) berechnet werden kann. Oft sind aber nur bestimmte Eigenwerte von Interesse, z.B., die größten, die (betrags-)kleinsten oder die einer Stelle $\hat{\lambda}$ am nächsten gelegenen (im Turbinenbeispiel die bei der Solldrehzahl liegenden kritischen Drehzahlen). Hier ist die Vektor-oder orthogonale Iteration vorzuziehen. Dies gilt vor allem dann, wenn die Matrix A groß und dünn besetzt ist, da diese Verfahren A nur über Matrix-Vektor-Multiplikationen verwenden.

Die Vektoriteration beruht darauf, daß sich in der Folge $(A^k)_{k\in\mathbb{N}}$ der betragsgrößte Eigenwert gegen die anderen durchsetzt. Bei der numerischen Berechnung hoher Potenzen A^k sind allerdings Vorsichtsmaßnahmen zu treffen.

Die Vektor-Iteration (nach von Mises)

Ausgehend von $0 \neq z^{(0)} \in \mathbb{C}^n$ wird eine Vektorfolge berechnet durch

$$\left.\begin{array}{rcll} q^{(k-1)} & := & z^{(k-1)}/|z_{i_k}^{(k-1)}|, \quad \text{wo } |z_{i_k}^{(k-1)}| = \|z^{(k-1)}\|_\infty & \text{(Normierung)} \\ z^{(k)} & := & Aq^{(k-1)}, & \text{(Iteration)} \\ \lambda^{(k)} & := & z_{i_k}^{(k)}/q_{i_k}^{(k-1)} & \end{array}\right\} k = 1, 2, \ldots \tag{3.3.1}$$

Bemerkung: a) $q^{(k)}$ ist aus praktischen Gründen auf eins normiert, stimmt aber bis auf die Normierung mit dem Vektor $A^k z^{(0)}$ überein, d.h., es gilt

$$q^{(k)} = v^{(k)}/\|v^{(k)}\|_\infty, \quad v^{(k)} := A^k z^{(0)}. \tag{3.3.2}$$

b) Bei Iterationsverfahren $x^{(k+1)} := Ax^{(k)} + b$ zur Lösung Linearer Gleichungssysteme der Form $(I - A)\hat{x} = b$ gilt für den Fehler $v^{(k)} := x^{(k)} - \hat{x}$ und den Defekt $d^{(k)} := (I - A)x^{(k)} - b = (I - A)v^{(k)}$ die gleiche Beziehung, $v^{(k)} = Av^{(k-1)} = \ldots = A^k v^{(0)}$. Daher ist der nächste Satz auch in diesem Zusammenhang von grundlegender Bedeutung (vgl. §4).

c) Die Bestimmung von $\lambda^{(k)}$ in (3.3.1) erfolgt auf eine etwas komplizierte Weise, da hierbei gleiche Komponenten von $z^{(k)}$ und $q^{(k-1)}$ verglichen werden müssen. Zahlenbeispiel:

$$A := \begin{pmatrix} 0 & 1 & -3 \\ 1 & 2 & -1 \\ -3 & -1 & 0 \end{pmatrix}, \; z^{(0)} := \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad x^{(1)} = \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix}, \; \lambda_1 = 4, \; \lambda_2 = -3.$$

Vektoriteration, die betragsmaximale Komponente ist unterstrichen:

$k =$	1	2	3	...	10	11	12	13	14
	0	$\underline{3.33}$	0.8		$\underline{3.898}$	3.455	$\underline{3.940}$	3.682	$\underline{3.966}$
$z^{(k)}$	1	1.66	2.1		3.595	3.688	3.762	3.819	3.862
	$\underline{-3}$	-0.33	$\underline{-3.5}$		-3.291	$\underline{-3.922}$	-3.583	$\underline{-3.955}$	-3.759
$\lambda^{(k)}$	0	0.33	0.8		3.291	3.455	3.583	3.682	3.759

Offensichtlich konvergieren $z^{(k)} \to \lambda_1 x^{(1)}$, $\lambda^{(k)} \to \lambda_1$, allerdings recht langsam. Für den Quotienten aufeinanderfolgender Fehler gilt $|\lambda^{(k)} - \lambda_1|/|\lambda^{(k-1)} - \lambda_1| \cong \frac{3}{4} = |\lambda_2/\lambda_1|$.

Da Eigenvektoren nur bis auf Konstanten bestimmt sind, müssen in der folgenden Konvergenzaussage die Vektoren $q^{(k)}$ und $x^{(1)}$ gleichartig normiert werden.

Satz 3.3.1 *Für die Eigenwerte der Matrix A gelte $|\lambda_1| > |\lambda_2| \geq \ldots \geq |\lambda_n|$. An dem Startvektor $z^{(0)}$ sei der Eigenvektor $x^{(1)}$ beteiligt, d.h., $z^{(0)} = \alpha_1 x^{(1)} + \ldots$ mit $\alpha_1 \neq 0$. Dann konvergiert die Vektor-Iteration in dem Sinn, daß eine skalare Folge $\sigma_k \in \mathbb{C}$ existiert mit*

$$\sigma_k q^{(k)} \to x^{(1)}, \quad \lambda^{(k)} \to \lambda_1, \quad k \to \infty.$$

Für eine quantitative Aussage sei $0 < \varepsilon < |\lambda_1| - |\lambda_2|$. Zu einem festen Index i mit $x_i^{(1)} \neq 0$ sei $\sigma_k = x_i^{(1)}/q_i^{(k)}$. Dann existieren Konstanten c, k_0 so, daß

$$\left.\begin{array}{c} |\lambda^{(k)} - \lambda_1| \\ \|\sigma_k q^{(k)} - x^{(1)}\| \end{array}\right\} \leq c \left(\frac{|\lambda_2| + \varepsilon}{|\lambda_1|}\right)^k, \ k \geq k_0.$$

Beweis Zur Vereinfachung des Beweises wird zunächst die Diagonalisierbarkeit von A vorausgesetzt. Der Startvektor besitze die Entwicklung

$$z^{(0)} = \sum_{j=1}^n \alpha_j x^{(j)}, \quad \alpha_1 \neq 0.$$

Dann gilt für die Vektoren $v^{(k)}$ aus (3.3.2) die Darstellung

$$v^{(k)} = \sum_{j=1}^n \lambda_j^k \alpha_j x^{(j)} = \lambda_1^k \alpha_1 \Big(x^{(1)} + \sum_{j=2}^n \frac{\alpha_j}{\alpha_1} \Big(\frac{\lambda_j}{\lambda_1}\Big)^k x^{(j)}\Big). \tag{3.3.3}$$

Mit dem normierten Vektor $q^{(k-1)} := v^{(k-1)}/\|v^{(k-1)}\|$ folgt dann

$$\lambda^{(k)} = \frac{z_{i_k}^{(k)}}{q_{i_k}^{(k-1)}} = \frac{v_{i_k}^{(k)}}{v_{i_k}^{(k-1)}} = \lambda_1 \frac{x_{i_k}^{(1)} + \sum_{j=2}^n \frac{\alpha_j}{\alpha_1} \big(\frac{\lambda_j}{\lambda_1}\big)^k x_{i_k}^{(j)}}{x_{i_k}^{(1)} + \sum_{j=2}^n \frac{\alpha_j}{\alpha_1} \Big(\frac{\lambda_j}{\lambda_1}\Big)^{k-1} x_{i_k}^{(j)}}$$

Nach Konstruktion ist $|q_{i_k}^{(k-1)}| = 1$. Da $(\lambda_j/\lambda_1)^k \to 0 \, \forall j > 1$ ist daher auch $x_{i_k}^{(1)} \neq 0$ für $k \geq k_0$. Daher folgt $|\lambda^{(k)} - \lambda_1| \leq c|\frac{\lambda_2}{\lambda_1}|^k, \ k \geq k_0$.

Für $\sigma := \lambda_1/|\lambda_1| \neq 1$ gilt asymptotisch nur $q^{(k)} \simeq \sigma^k x^{(1)}$, d.h., Konvergenz liegt nur für die von $q^{(k)}$ aufgespannten Unterräume vor. Eine normale Konvergenzaussage kann nach Einfrieren einer Komponente gemacht werden. Für $x_i^{(1)} \neq 0$ gilt auch $v_i^{(k)} \neq 0$ für $k \geq k_0$ nach (3.3.3). Daher ist

$$\frac{q^{(k)}}{q_i^{(k)}} = \frac{v^{(k)}}{v_i^{(k)}} = \frac{x^{(1)} + \sum_{j=2}^n \frac{\alpha_j}{\alpha_1} \big(\frac{\lambda_j}{\lambda_1}\big)^k x^{(j)}}{x_i^{(1)} + \sum_{j=2}^n \frac{\alpha_j}{\alpha_1} \Big(\frac{\lambda_j}{\lambda_1}\Big)^k x_i^{(j)}} = \frac{x^{(1)}}{x_i^{(1)}} + \mathcal{O}\Big(\Big|\frac{\lambda_2}{\lambda_1}\Big|^k\Big).$$

Bei einer nicht diagonalisierbaren Matrix kann das Wachstum der nichttrivialen Jordanblöcke nach einer Aufspaltung

$$J = X^{-1} A X = \begin{pmatrix} \lambda_1 & 0 \\ 0 & M \end{pmatrix}$$

mit Satz 3.1.9 abgeschätzt werden. Dann tritt in der Aussage tatsächlich ein $\varepsilon > 0$ auf. ∎

Der zum betragsgrößten Eigenwert λ_1 gehörige Eigenvektor $x^{(1)}$ setzt sich im Verlauf der Iteration also gegenüber den anderen Eigenvektoren durch. Daher werden λ_1 bzw. $x^{(1)}$ als *dominanter* Eigenwert bzw. Eigenvektor bezeichnet. Die Voraussetzung $\alpha_1 = y^{(1)^*}x^{(1)} \neq 0$ kann oft aus theoretischen Gründen garantiert werden. Aber selbst ohne Vorkenntnis wird durch Rundungsfehler i.a. ein Beitrag mit $\alpha_1 \neq 0$ eingeschleppt.

Die Konvergenz der einfachen Vektoriteration ist nur linear mit dem Konvergenzfaktor $|\lambda_2/\lambda_1|$. Wenn λ_1 und λ_2 nur geringe Betragsunterschiede aufweisen, konvergiert die Iteration nur sehr langsam. Folgende Beobachtungen führen aber zu merklichen Verbesserungen:

- Die Eigenwerte der Matrix $A - \lambda I$ sind $\lambda_j - \lambda$. Eine solche *Spektralverschiebung* kann die Konvergenz von (3.3.1) verbessern, z.B., wenn $\lambda_1\lambda_2 < 0$ oder $0 < \ldots \leq \lambda_2 < \lambda_1$ gilt bei reellen Eigenwerten.

- Die Eigenwerte von A^{-1} sind λ_j^{-1}, die von $(A-\lambda I)^{-1}$ also $(\lambda_j-\lambda)^{-1}$. Der bei dieser Matrix etwa zur Berechnung von λ_1 entscheidende Quotient
$$\max_{j\neq 1}\left|\frac{\lambda_1-\lambda}{\lambda_j-\lambda}\right|$$
wird bei geeigneter Wahl von λ beliebig klein. Dies ist die Basis des folgenden Verfahrens.

Inverse Iteration (nach Wielandt)

$\lambda \in \mathbb{C}$ sei kein Eigenwert von A, mit $z^{(0)} \in \mathbb{C}^n$ wird iterativ berechnet

$$\left.\begin{array}{rcll} q^{(k-1)} & := & z^{(k-1)}/|z^{(k-1)}_{i_k}|, & \text{wo } |z^{(k-1)}_{i_k}| = \|z^{(k-1)}\|_\infty \\ z^{(k)} & := & (A-\lambda I)^{-1}q^{(k-1)}, & \\ \lambda^{(k)} & := & \lambda + q^{(k-1)}_{i_{k+1}}/z^{(k)}_{i_{k+1}} & \end{array}\right\} k = 1,2,\ldots \tag{3.3.4}$$

Für die Konvergenz dieses Verfahrens gilt

Satz 3.3.2 *Der Eigenwert λ_m von A sei in Bezug auf λ isoliert, d.h., es gelte*
$$|\lambda_m - \lambda| < |\lambda_j - \lambda| \ \forall j \neq m.$$
Wenn der Eigenvektor $x^{(m)}$ an $z^{(0)}$ beteiligt ist, also $y^{(m)^*}z^{(0)} \neq 0$ gilt, dann konvergieren die Werte $\lambda^{(k)}$ gegen λ_m und (im verallgemeinerten Sinn) die Vektoren $q^{(k)}$ gegen $x^{(m)}$. Quantitativ gilt für jedes $\varepsilon > 0$ mit $|\lambda_m - \lambda| + \varepsilon < |\lambda_j - \lambda| \ \forall j \neq m$ eine Schranke
$$\left.\begin{array}{r} |\lambda^{(k)} - \lambda_m| \\ \min_{\sigma\in\mathbb{C}} \|\sigma q^{(k)} - x^{(m)}\| \end{array}\right\} \leq c\left(\frac{|\lambda_m - \lambda|}{\min_{j\neq m}|\lambda_j - \lambda| - \varepsilon}\right)^k,\ k \geq k_0.$$

Durch geeignete Wahl von λ läßt sich also gezielt ein einzelner Eigenwert der Matrix bestimmen. Zur weiteren Beschleunigung der Konvergenz kann in (3.3.4) für λ jeweils die aktuellste Eigenwert-Schätzung, $\lambda^{(k-1)}$, zur Spektralverschiebung verwendet werden:

$$z^{(k)} := (A - \lambda^{(k-1)} I)^{-1} q^{(k-1)}.$$

Bei dieser Variante ist nun in jedem Schritt eine LR- oder QR-Zerlegung durchzuführen. Daher sollte hier die Matrix vorher in Hessenbergform gebracht werden, da jede Zerlegungen dann nur einen $\mathcal{O}(n^2)$-Aufwand erfordern. Diese Variante der Inversen Iteration hat in gutartigen Situationen hervorragende Eigenschaften (vgl. symmetrisches Eigenwertproblem). In schwierigen Fällen kann es aber vorkommen, daß nur der erste Schritt eine Verbesserung bringt. Der wichtigste Einsatzbereich der Inversen Iteration ist die Berechnung einzelner Eigenvektoren, wenn Eigenwert-Approximationen mit einem anderen Verfahren (z.B. QR-Verfahren, Bisektion) geliefert wurden. Zu den problematischen Fällen gilt

Beispiel 3.3.3 Es sei $A = \begin{pmatrix} 1 & 1 \\ \varepsilon & 1 \end{pmatrix}$ mit kleinem ε, z.B., $\varepsilon = 10^{-10}$. Eine solche Matrix kann etwa bei Maschinenrechnung anstelle eines 2×2-Jordanblocks entstehen. Sie hat die Eigenwerte $1 \pm \sqrt{\varepsilon}$ und die Eigenvektoren $x^{(1/2)} = \begin{pmatrix} 1 \\ \pm\sqrt{\varepsilon} \end{pmatrix}$. Als Eigenwert-Näherung der Inversen Iteration werde $\lambda = 1$ verwendet. Dann ist

$$(A - \lambda I)^{-1} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ \varepsilon & 0 \end{pmatrix}^{-1} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} x_2/\varepsilon \\ x_1 \end{pmatrix}.$$

Für die 'meisten' Startvektoren $q^{(0)} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$, die mit $\varepsilon |x_1| < |x_2|$, ist $q^{(1)} = \begin{pmatrix} 1 \\ \varepsilon x_1/x_2 \end{pmatrix}$ eine gute Näherung der Eigenvektoren, denn $\|Aq^{(1)} - \lambda q^{(1)}\| = \varepsilon\sqrt{1 + |x_1/x_2|^2}$ ist klein. Dagegen ist die nächste Iterierte $z^{(2)} = \begin{pmatrix} x_1/x_2 \\ 1 \end{pmatrix}$ wieder weit davon entfernt, da $\|Az^{(2)} - \lambda z^{(2)}\| = \|q^{(1)}\| = 1$.

Die Vektor-Iteration (in der besprochenen Form) ist nur bei einfachen Eigenwerten mit genügend großem gegenseitigen Abstand praktisch verwendbar. Zur Behandlung mehrfacher, oder eng zusammenliegender Eigenwerte muß die Iteration gleichzeitig mit mehreren Vektoren durchgeführt werden. Die im Vektor $z^{(k)}$ enthaltene Information wird bei der Vektor-Iteration zerlegt in einen Richtungsvektor $q^{(k)}$ mit Norm eins und einen Multiplikator, die Eigenwert-Schätzung $\lambda^{(k)}$. Bei Iteration mit $p > 1$ Vektoren, d.h., mit Matrizen $Z^{(k)} \in \mathbb{C}^{n \times p}$, in der Form $Z^{(k)} = AQ^{(k-1)}$ kann diese Aufspaltung

$$z^{(k)} = q^{(k)} \|z^{(k)}\| \quad \text{durch die QR-Zerlegung} \quad Z^{(k)} = Q^{(k)} R^{(k)}$$

verallgemeinert werden. In Satz 1.3.3 war dabei Q als quadratische und R als $n \times p$-Matrix in Dreieckform betrachtet worden. Da aber die letzten $n - p$ Zeilen von R verschwinden, ist es hier ($p \ll n$) geschickter, R als quadratische $p \times p$-Matrix und Q als orthogonale $n \times p$-Matrix zu betrachten. Zur Vermeidung von Doppelbezeichnungen (vgl. QR-Iteration) wird der unitäre Faktor in diesem Verfahren U genannt. Die so verallgemeinerte Vektor-Iteration bietet außerdem einen einfachen Zugang zum wichtigsten Eigenwert-Verfahren, der QR-Iteration.

Orthogonale Iteration

Ausgehend von einer orthogonalen Matrix $U^{(0)} \in \mathbb{C}^{n\times p}$ werden Matrizen $Z^{(k)}, U^{(k)} \in \mathbb{C}^{n\times p}$, $R^{(k)} \in \mathbb{C}^{p\times p}$ berechnet nach der Vorschrift

$$\left.\begin{array}{rcll} Z^{(k)} & := & AU^{(k-1)} & \\ U^{(k)}R^{(k)} & := & Z^{(k)} & \text{(QR-Zerlegung)} \end{array}\right\} k = 1, 2, \ldots \tag{3.3.5}$$

Da die Spalten von $U^{(k)}$ zueinander orthogonal sind, können sie i.a. natürlich nicht alle gegen Eigenvektoren von A konvergieren. Es zeigt sich aber, daß bei einer Betragstrennung der Eigenwerte nach dem p-ten gemäß $|\lambda_1| \geq \ldots \geq |\lambda_p| > |\lambda_{p+1}| \geq \ldots$ der aufgespannte Unterraum $R(U^{(k)})$ gegen den *invarianten Unterraum* H_p von A mit $AH_p \subseteq H_p$ konvergiert, der von den ersten p Eigen-/Haupt-Vektoren aufgespannt wird, $H_p = [x^{(1)}, \ldots, x^{(p)}]$.

Daher heißt (3.3.5) auch Unterraum-Iteration. Die orthogonale Iteration ist vor allem dann effektiv, wenn nur ein Teil ($p \ll n$) der Eigenwerte einer großen Matrix berechnet werden soll. Wenn A dünn besetzt ist, kann dies bei der Multiplikation AU direkt ausgenutzt werden. Dagegen zerstört das im folgenden besprochene QR-Verfahren, das eng mit (3.3.5) zusammenhängt, eine eventuell vorhandene Struktur von A.

Bemerkung: Alle bei der orthogonalen Iteration (3.3.5) auftretenden Matrizen können für beliebige $p < n$ als vordere Spalten quadratischer $n \times n$-Matrizen betrachtet werden. Denn

— beim Produkt AU sind die einzelnen Spalten voneinander völlig unabhängig,

— bei der QR-Zerlegung nach Householder hängen die j-ten Spalten von R und U nicht von den nachfolgenden Spalten von Z ab, sie werden durch die Transformationen $H^{(j+1)}, H^{(j+2)}, \ldots$ nicht mehr verändert, denn $Ue^{(j)} = H^{(1)} \cdots H^{(n-1)}e^{(j)} = H^{(1)} \cdots H^{(j)}e^{(j)}$.

Aufgrund dieser Bemerkung muß, etwa bei Konvergenzaussagen, nur der Fall $p = n$ behandelt werden. Da diese Aussagen dann für die orthogonale Iteration und das QR-Verfahren aufgrund des folgenden Zusammenhangs völlig analog sind, werden sie beim QR-Verfahren behandelt. Zu diesem kommt man zwangsläufig, wenn man (im Fall $p = n$) die Matrizen

$$S^{(k)} := U^{(k)*}AU^{(k)} \tag{3.3.6}$$

betrachtet, die zu A ähnlich sind und daher die Eigenwert-Information der orthogonalen Iteration enthalten. Überlegt man nämlich, wie $S^{(k)}$ aus $S^{(k-1)}$ hervorgeht, so ergibt sich aus (3.3.5) $R^{(k)} = U^{(k)*}AU^{(k-1)}$ und daher einerseits

$$S^{(k)} = U^{(k)*}AU^{(k-1)}U^{(k-1)*}U^{(k)} = R^{(k)}\underbrace{U^{(k-1)*}U^{(k)}}_{=:\,Q^{(k)}}, \tag{3.3.7}$$

aber auch

$$S^{(k-1)} = U^{(k-1)*}AU^{(k-1)} = U^{(k-1)*}Z^{(k)} = \underbrace{U^{(k-1)*}U^{(k)}}_{Q^{(k)}}R^{(k)}.$$

Also bekommt man $S^{(k)}$ durch Berechnung einer QR-Zerlegung von $S^{(k-1)}$ und anschließender Multiplikation der beiden Faktoren in umgekehrter Reihenfolge. Dies ist

Das QR-Verfahren

Mit $S^{(0)} := A$ werden Matrizen $Q^{(k)}, S^{(k)}, R^{(k)} \in \mathbb{C}^{n\times n}$ berechnet nach

$$\left.\begin{array}{rcll} Q^{(k)}R^{(k)} & := & S^{(k-1)} & \text{(QR-Zerlegung)} \\ S^{(k)} & := & R^{(k)}Q^{(k)} & \end{array}\right\} k = 1, 2, \ldots \tag{3.3.8}$$

Die folgenden Ergänzungen beruhen auf der Tatsache, daß die Subdiagonalelemente von $S^{(k)}$ in der Regel gegen null konvergieren. Präziser gilt dazu die Aussage

$$s_{ij}^{(k)} = \mathcal{O}\Big(\Big|\frac{\lambda_i}{\lambda_j}\Big|^k\Big), \quad k \to \infty,\ j < i, \tag{3.3.9}$$

bei der Anordnung $|\lambda_1| \geq \ldots \geq |\lambda_n|$, die in Satz 3.3.7 gezeigt wird. In der einfachen Form ist das Verfahren (3.3.8) zu aufwendig, da jeder Schritt $\mathcal{O}(n^3)$ Operationen kostet (Satz 1.3.3ff). Die Effizienz des Verfahrens kann durch die drei folgenden Maßnahmen dramatisch gesteigert werden.

- **Satz 3.3.4** *Besitzt $S^{(0)} = A$ Hessenbergform, dann gilt dies auch für alle Iterierten $S^{(k)}$. In diesem Fall beträgt der Aufwand für einen Schritt (3.3.8) nur noch $8n^2$ Operationen.*

  Beweis Die für die QR-Zerlegung (1.3.7) benutzten Spiegelungen $H^{(j)} = I - 2u^{(j)}u^{(j)^*}$ verwenden Vektoren $u^{(j)}$, in denen nur die j-te und $j+1$-te Komponente nicht null sind. Bei der Rückmultiplikation $S = RH^{(1)} \ldots H^{(n-1)}$ werden daher in S nur Elemente direkt links von der Hauptdiagonale von R eingeführt. ■

- **Satz 3.3.5** *Bei Spektralverschiebung*

$$\left.\begin{array}{rcl} Q^{(k)}R^{(k)} & := & S^{(k-1)} - \lambda I \\ S^{(k)} & := & R^{(k)}Q^{(k)} + \lambda I \end{array}\right\} k = 1, 2, \ldots \tag{3.3.10}$$

  verändert sich mit der neuen Anordnung $|\lambda_1 - \lambda| \geq \ldots \geq |\lambda_n - \lambda|$ die Konvergenzaussage (3.3.9) zu

$$s_{ij}^{(j)} = \mathcal{O}\Big(\Big|\frac{\lambda_i - \lambda}{\lambda_j - \lambda}\Big|^k\Big),\ j < i$$

  Beweis Die Eigenwerte von $S^{(k)}$ werden nicht verändert, denn

$$S^{(k)} = R^{(k)}Q^{(k)} + \lambda I = Q^{(k)^*}(S^{(k-1)} - \lambda I)Q^{(k)} + \lambda I = Q^{(k)^*}S^{(k-1)}Q^{(k)}, \tag{3.3.11}$$

  konvergenzbestimmend sind aber die verschobenen Eigenwerte $\lambda_j - \lambda$. ■

  Formel (3.3.11) zeigt übrigens ebenfalls, daß $S^{(k)}$ und $S^{(k-1)}$ ähnliche Matrizen sind.

- **Satz 3.3.6** *Ist S Hessenberg-Matrix mit $s_{n,n-1} = 0$, dann ist s_{nn} ein Eigenwert von S. Die restlichen Eigenwerte sind die von S_1 in der Partitionierung*

$$S = \left(\begin{array}{c|c} S_1 & \begin{matrix} * \\ \vdots \end{matrix} \\ \hline 0 \cdots 0 & s_{nn} \end{array}\right).$$

Beweis $e^{(n)}$ ist Links-Eigenvektor von S mit Eigenwert s_{nn}. Die Links-Eigen- bzw. Haupt-Vektoren von S_1 können durch n-te Komponenten zu solchen von S ergänzt werden. ■

Wenn im Verlauf der Iteration das letzte Subdiagonalelement $s_{n,n-1}^{(k)}$ sehr klein wird, kann daher die QR-Iteration mit der kleineren Matrix S_1 fortgesetzt werden (*Deflation*).

Zur praktischen Anwendung von Satz 3.3.5 ist eine Strategie für die Wahl der (des) Spektralparameter(s) erforderlich, um eine Verbesserung der Konvergenz zu erreichen. Im allgemeinen ist die Konvergenz des letzten Diagonalelements gegen den betragskleinsten Eigenwert, $s_{nn}^{(k)} \to \lambda_n$, am schnellsten. Nach einer Anlaufphase, etwa wenn

$$\left|1 - \frac{s_{nn}^{(k-2)}}{s_{nn}^{(k-1)}}\right| < \frac{1}{3}$$

gilt, läßt sich dieses Verhalten durch folgende Wahl für λ in Satz 3.3.5 unterstützen:

$$\lambda^{(k-1)} := s_{nn}^{(k-1)}. \tag{3.3.12}$$

Bei **reellen** Matrizen mit nicht-reellen Eigenwerten treten diese in **konjugiert-komplexen Paaren** auf. Wenn genau zwei Eigenwerte gleichen Betrag haben, hat $S^{(k)}$ für $k \to \infty$ die Gestalt

$$S^{(k)} \cong \begin{pmatrix} \ddots & \cdot & \cdot & \cdots \\ & * & * & \cdots \\ & * & * & \cdots \\ & & & \ddots \end{pmatrix}.$$

Auch bei reeller Rechnung können die komplexen Eigenwerte aus dem 2×2-Block

$$\begin{pmatrix} s_{pp} & s_{p,p+1} \\ s_{p+1,p} & s_{p+1,p+1} \end{pmatrix}$$

berechnet werden, wenn $s_{p+1,p}$ das auch für große k nicht verschwindende Subdiagonalelement in $S^{(k)}$ ist. Die Spektralverschiebung nach Satz 3.3.5 bewirkt aber nur mit komplexen Spektralparametern λ eine brauchbare Konvergenzbeschleunigung. Die (aufwendige) komplexe Rechnung läßt sich dabei umgehen, wenn in zwei aufeinander folgenden Schritten zueinander konjugierte Parameter $\lambda^{(k+1)} = \bar{\lambda}^{(k)}$ verwendet werden. Dafür geeignet sind die beiden Eigenwerte von

$$\begin{pmatrix} s_{n-1,n-1}^{(k-1)} & s_{n-1,n}^{(k-1)} \\ s_{n,n-1}^{(k-1)} & s_{nn}^{(k-1)} \end{pmatrix}.$$

Man überlegt sich leicht, daß dann die Matrizen $S^{(k+1)}, Q^{(k+1)}$ wieder reell sind. Sie können mit reeller Rechnung aus $S^{(k-1)}, Q^{(k-1)}$ durch einen sogenannten *QR-Doppelschritt* berechnet werden. Dieser Spezialfall wird wegen seiner Komplexität nicht behandelt.

Ein umfassender, globaler Konvergenzbeweis für das QR-Verfahren ist nur beim symmetrischen Eigenwertproblem möglich. Im unsymmetrischen Fall gilt

Satz 3.3.7 *Die Eigenwerte der Matrix A seien betragsmäßig verschieden und $|\lambda_1| > \ldots > |\lambda_n|$. Für die Matrix $Y := X^{-1}$ aus der Jordan-Normalform $A = XJX^{-1}$ existiere die LR-Zerlegung. Dann konvergieren die Matrizen $S^{(k)}$ des einfachen QR-Verfahrens (3.3.8) wie in (3.3.9) gegen eine obere Dreieckmatrix, insbesondere gilt*

$$s_{ii}^{(k)} \to \lambda_i \quad (k \to \infty).$$

Mit Spektralverschiebung (3.3.12) konvergiert die QR-Iteration (3.3.10) lokal quadratisch, mit

$$|s_{n,n-1}^{(k+1)}| \le c|s_{n,n-1}^{(k)}|^2,$$

sobald $|s_{n,n-1}^{(k)}|$ 'klein genug' ist.

Beweis Zur Hervorhebung des grundlegenden Beweisprinzips wird der Beweis nur unter Einschränkungen und nur für das einfache Verfahren (3.3.8) geführt. Es sei A reell und $|\lambda_n| > 0$, A also regulär. Ein wesentliches Argument im Beweis ist die weitgehende Eindeutigkeit und stetige Abhängigkeit der QR-Zerlegung. Dazu sind folgende Vorbemerkungen erforderlich:

— Die QR-Zerlegung nach Satz 1.3.3 enthält eine Vorzeichenentscheidung zur Vermeidung von Rundungsfehlern. Wählt man davon abweichend in R positive Hauptdiagonalelemente ($\beta := \|x\|_2$ in (1.3.3)), dann ist die QR-Zerlegung eindeutig.

— Diese QR-Zerlegung hängt auch stetig von der zerlegten Matrix ab, z.B., gilt für die QR-Faktoren Q_k, R_k einer Folge $R + F_k = Q_k R_k$, mit einer oberen Dreieckmatrix R (positive Hauptdiagonale) und $F_k \to 0$ $(k \to \infty)$ die Aussage $Q_k \to I,\ R_k \to R$. Zur Unterscheidung von den Matrizen des QR-Verfahrens sind diese hier ausnahmsweise unten indiziert.

a) Als erstes werden die Transfomationsmatrizen der QR-Iteration (3.3.8) aufmultipliziert

$$U^{(k)} := Q^{(1)}Q^{(2)} \cdots Q^{(k)},\ k > 0, \quad U^{(0)} := I, \tag{3.3.13}$$

vgl. auch (3.3.7). Mit diesen gilt wegen (3.3.11)

$$\begin{aligned} AU^{(k-1)} &= AQ^{(1)} \cdots Q^{(k-1)} = Q^{(1)}S^{(1)}Q^{(2)} \cdots Q^{(k-1)} = \ldots \\ &= Q^{(1)} \cdots Q^{(k-1)}S^{(k-1)} = U^{(k-1)}S^{(k-1)} \qquad (3.3.14)\\ &= U^{(k-1)}Q^{(k)}R^{(k)} = U^{(k)}R^{(k)}. \qquad (3.3.15) \end{aligned}$$

und daher

$$A^k = A^k U^{(0)} = A^{k-1}U^{(1)}R^{(1)} = \ldots = U^{(k)}\underbrace{R^{(k)} \cdots R^{(2)}R^{(1)}} =: U^{(k)}\hat{R}^{(k)}. \tag{3.3.16}$$

$U^{(k)}$ ist also der eindeutige (s.o.) unitäre Faktor der QR-Zerlegung von A^k.

b) In der Jordan-Normalform von A sei nun $UR_X = X$ die QR-Zerlegung von X (U ist die unitäre Matrix der Schur-Normalform, vgl. Beweis von Satz 3.1.6) und $L_Y R_Y = Y$ die LR-Zerlegung von $Y = X^{-1}$. Dann gilt für A^k auch

$$A^k = XJ^kX^{-1} = UR_XJ^kL_YR_Y = UR_X \underbrace{(J^kL_YJ^{-k})}_{=:\, I+F_k} J^kR_Y. \tag{3.3.17}$$

Da J diagonal ist, hat die untere Dreieckmatrix $J^kL_YJ^{-k}$ die Elemente $l_{ij}(\lambda_i/\lambda_j)^k$. Die Subdiagonalen dieser Matrix verschwinden für $k \to \infty$, da $|\lambda_i/\lambda_j| \le q < 1$ ist für $j < i$. Daher gilt in (3.3.17) tatsächlich

$$J^kL_YJ^{-k} = I + F_k \quad \text{mit} \quad \|F_k\| \le cq^k \to 0 \ (k \to \infty).$$

Nach den einleitenden Bemerkungen hat die Folge der QR-Zerlegungen von $R_X(I+F_k) = Q_kR_k$ die Eigenschaft $Q_k \to I, R_k \to R_X$. Als letzter Schritt seien

$$D_k := \operatorname{diag}(\sigma_i^k \operatorname{sign}(r_{ii}^{(Y)})), \ \sigma_i := \operatorname{sign}(\lambda_i),$$

Vorzeichenmatrizen mit $D_k^2 = I$. Aus (3.3.17) wird nun

$$A^k = UQ_kR_kJ^kR_Y = (UQ_kD_k)(D_kR_kJ^kR_Y). \tag{3.3.18}$$

Dabei ist $D_kR_kJ^kR_Y$ eine obere Dreieckmatrix, deren Diagonalelemente als Produkte der Diagonalen ihrer Faktoren positiv sind, und UQ_kD_k ist unitär.

c) Durch Vergleich der beiden (eindeutigen) QR-Zerlegungen (3.3.16) und (3.3.18) von A^k folgt somit

$$U^{(k)} = UQ_kD_k \quad \text{und} \quad \hat{R}^{(k)} = D_kR_kJ^kR_Y. \tag{3.3.19}$$

Wegen $Q_k \to I$ konvergiert daher $U^{(k)}D_k \to U$, die Konvergenz von $U^{(k)}$ gegen U aus der Schur-Normalform tritt also bis auf Vorzeichen der Spalten ein. Diese Vorzeichen sind aber irrelevant, für die Matrizen $S^{(k)}$ gilt nach (3.3.14) und (3.3.19) nämlich

$$S^{(k)} \ = \ U^{(k)^*}AU^{(k)} = D_k \underbrace{Q_k^*}_{\to I} \underbrace{U^*AU}_{=S} \underbrace{Q_k}_{\to I} D_k.$$

Demnach konvergiert $S^{(k)}$ gegen eine obere Dreieckmatrix, präzise gilt $D_kS^{(k)}D_k \to S$ mit der Matrix S aus der Schur-Normalform. In der Hauptdiagonale kompensieren sich die alternierenden Faktoren: $\sigma_i^k s_{ii}^{(k)} \sigma_i^k = s_{ii}^{(k)} \to s_{ii} = \lambda_i$, $k \to \infty$. ■

Beispiel 3.3.8 Das QR-Verfahren wird auf die Hessenbergmatrix

$$A = \begin{pmatrix} 3 & -2 & -0.5 & 0 & 0 \\ -2 & 3 & -2 & -0.5 & 0 \\ 0 & -2 & 3 & -2 & -0.5 \\ 0 & 0 & -2 & 3 & -2 \\ 0 & 0 & 0 & -2 & 3 \end{pmatrix}$$

angewendet, ab Schritt 5 (unter der horizontalen Linie) mit Spektralverschiebung (3.3.12). Die Tabelle zeigt die Diagonale der Matrizen $S^{(k)}$ und das Subdiagonalelement $s_{m,m-1}^{(k)}$, wobei $m \leq n$ die aktuelle Dimension der Restmatrix ist (vgl. Satz 3.3.6) angedeutet durch die Treppenlinie.

k	s_{11}	s_{22}	s_{33}	s_{44}	s_{55}	$s_{m,m-1}$
1	4.846154	4.088911	3.139369	1.843980	1.081586	−1.653266
2	5.401216	4.602085	4.064473	0.835122	0.097104	0.740867
3	5.698527	4.970917	3.902994	0.882349	−0.454788	−0.965393
4	5.878893	5.162691	3.570136	0.797329	−0.409049	0.467165
5	5.991053	5.211672	3.418541	1.126695	−0.747961	−0.132251
6	6.064557	5.203066	3.356221	1.095888	−0.719731	−0.001994
7	6.114333	5.179253	3.331381	1.094180	−0.719147	−0.000001
8	6.154318	5.149222	3.322309	1.093298	−0.719147	−0.000002
9	6.171288	5.130994	3.323566	1.093298	−0.719147	−0.000000
10	6.171301	5.122938	3.331609	1.093298	−0.719147	−0.000354
11	6.156498	5.137690	3.331661	1.093298	−0.719147	−0.000000
12	6.103079	5.191109	3.331661	1.093298	−0.719147	−0.007077
13	6.098903	5.195285	3.331661	1.093298	−0.719147	−0.000032
14	6.098883	5.195305	3.331661	1.093298	−0.719147	−0.000000

Das QR-Verfahren ist **das** Standardverfahren zur Lösung des vollständigen unsymmetrischen Eigenwertproblems. Wie im Beispiel werden nach der Anlaufphase für jeden Eigenwert in der Regel 2 QR-Schritte benötigt. Daraus leitet man einen *empirischen* Mittelwert für den Rechenaufwand zur Bestimmung aller Eigenwerte (ohne Transformationsmatrix U) von $16n^3$ Operationen her, die volle Schur-Normalform (S und U) benötigt analog

$$\cong 30n^3 \text{Operationen.}$$

Das QR-Verfahren vereinigt in sich die anderen Iterationsverfahren. Man sieht leicht, daß hier u.a. sowohl eine Vektoriteration als auch eine Inverse Iteration ausgeführt werden:

a) Zunächst entspricht nach (3.3.14) ein QR-Schritt einem Schritt der orthogonalen Iteration (3.3.5) (vgl. auch (3.3.7)). Da $R^{(k)}$ obere Dreieckmatrix ist, wird somit in der *ersten* Spalte von $U^{(k)}$ eine Vektoriteration durchgeführt: $AU^{(k-1)}e^{(1)} = r_{11}^{(k)}U^{(k)}e^{(1)}$.

b) Bei Spektralverschiebung ändert sich (3.3.15) zu

$$AU^{(k-1)} = U^{(k)}R^{(k)} + \lambda^{(k-1)}U^{(k-1)} \quad \iff \quad (A - \lambda^{(k-1)}I)U^{(k-1)} = U^{(k)}R^{(k)}.$$

Da $U^{(k)}$ unitär ist, folgt nach Inversion und Transposition

$$(A^* - \bar{\lambda}^{(k-1)}I)^{-1}U^{(k-1)} = U^{(k)}(R^{(k-1)})^{-1^*}.$$

Hier ist $(R^{(k-1)})^{-1^*}$ ein untere Dreieckmatrix und daher gilt $(R^{(k-1)})^{-1^*}e^{(n)} = e^{(n)}/\bar{r}_{nn}^{(k)}$. Analog zu oben folgt

$$U^{(k)}e^{(n)} = \bar{r}_{nn}^{(k)}\,(A^* - \bar{\lambda}^{(k-1)}I)^{-1}\,(U^{(k-1)}e^{(n)}).$$

Dies ist aber ein Schritt der Inversen Iteration (3.3.4) mit der *letzten* Spalte von $U^{(k-1)}$!

Das QR-Verfahren führt also in der ersten Spalte der akkummulierten Transformationsmatrizen $U^{(k)}$ eine Vektor-Iteration und in deren letzter Spalte eine Inverse Iteration durch.

Der *Vorläufer* des QR-Verfahrens war das *LR-Verfahren*

$$A^{(0)} := A, \quad L^{(k)}R^{(k)} := A^{(k-1)}, \quad A^{(k)} := R^{(k)}L^{(k)},$$

bei dem statt der QR-Zerlegung die LR-Zerlegung der Iterierten verwendet wird. Auch dieses Verfahren bewahrt die Hessenbergform von $A^{(0)}$ und erfordert pro Schritt sogar weniger Aufwand als das QR-Verfahren. Allerdings kann die Existenz der LR-Zerlegungen (ohne Pivotisierung) nicht garantiert werden, andererseits kann Pivotisierung die Konvergenz zerstören. Daher ist das Verfahren weniger robust als das QR-Verfahren.

3.4 Das symmetrische Eigenwertproblem

Für diesen Spezialfall können bei den besprochenen Verfahren verbesserte Varianten angegeben und präzisere Aussagen gemacht werden. Außerdem sind einige Spezialverfahren einsetzbar, die nicht oder nur schwer auf den unsymmetrischen Fall verallgemeinert werden können. Zwei Eigenschaften des hermiteschen Eigenwert-Problems

$$Ax = \lambda x, \quad A = A^*, \tag{3.4.1}$$

$0 \neq x \in \mathbb{C}^n$, erleichtern den Zugang erheblich. Eine hermitesche Matrix besitzt nämlich

- nur reelle Eigenwerte,
- eine Orthonormalbasis von Eigenvektoren.

Dies ergibt sich direkt aus der Schur-Normalform, vgl. Satz 3.1.8. Etwas direkter ist allerdings der Zugang über die quadratische Form.

Definition 3.4.1 *Zu einer beliebigen Matrix $A \in \mathbb{C}^{n\times n}$ und $0 \neq x \in \mathbb{C}^n$ heißt*

$$R_A[x] := \frac{x^*Ax}{x^*x} \tag{3.4.2}$$

der Rayleigh-Quotient *an der Stelle x.*

Wenn x, y auf eins normierte Eigenvektoren zu verschiedenen Eigenwerten $\lambda \neq \mu$ sind, folgen die beiden Aussagen zum symmetrischen Fall aus

$$\lambda = x^*Ax = (x^*Ax)^\mathsf{T} = \overline{x^*A^*x} = \overline{\lambda} \qquad \text{und}$$

$$0 = y^*Ax - (Ay)^*x = \lambda y^*x - \mu y^*x = (\lambda - \mu)y^*x.$$

Mit Hilfe des Rayleigh-Quotienten kann bei der Vektor-Iteration eine wesentlich bessere Eigenwert-Näherung als im unsymmetrischen Fall berechnet werden.

Satz 3.4.2 *Es sei $x \in \mathbb{C}^n$ mit $\|x\|_2 = 1$.*

a) Für eine beliebige Matrix $A \in \mathbb{C}^{n\times n}$ ist

$$\|Ax - R_A[x]x\|_2 = \min_{\lambda\in\mathbb{C}} \|Ax - \lambda x\|_2.$$

b) Wenn $A \in \mathbb{C}^{n\times n}$ normal ist, gilt für jeden Index k mit $d_k := \min_{j\neq k} |\lambda_j - \lambda_k| > 0$ die Aussage

$$\Big|R_A[x] - \lambda_k\Big| \leq \frac{1}{d_k}\|Ax - \lambda_k x\|_2^2.$$

Bemerkung: Wenn x als eine Näherung für den Eigenvektor $x^{(k)}$ angesehen werden kann, weil $\hat{\varepsilon} := \|Ax - \lambda_k x\|$ klein ist, dann ist nach Teil a) der Rayleighquotient dazu die beste Eigenwert-Näherung. Der Fehler dieser Eigenwert-Approximation entspricht nach Teil b) bei einfachen Eigenwerten und normalen Matrizen dem Quadrat des Fehlers $\hat{\varepsilon}$ der Eigenvektor-Näherung x.

Beweis a) Die Aussage folgt direkt aus folgender quadratischer Ergänzung

$$\begin{aligned}\|Ax - \lambda x\|_2^2 &= \|Ax\|_2^2 - 2\Re(\bar{\lambda} x^* Ax) + |\lambda|^2\|x\|_2^2 \\ &= |\lambda - R_A[x]|^2\|x\|_2^2 + \|Ax\|_2^2 - |R_A[x]|^2\|x\|_2^2.\end{aligned}$$

b) Da die Eigenvektoren $\{x^{(j)}\}$ einer normalen Matrix eine Orthonormalbasis bilden, gilt

$$x = \sum_{j=1}^n \xi_j x^{(j)}, \quad Ax = \sum_{j=1}^n \lambda_j \xi_j x^{(j)}, \quad R_A[x] = \sum_{j=1}^n \lambda_j |\xi_j|^2,$$

denn $\|x\|_2^2 = \sum_{j=1}^n |\xi_j|^2 = 1$. Die Behauptung folgt aus

$$\Big|R_A[x] - \lambda_k\Big| = \Big|\sum_{j\neq k}(\lambda_j - \lambda_k)|\xi_j|^2\Big| \leq \frac{1}{d_k}\sum_{j\neq k}|\lambda_j - \lambda_k|^2|\xi_j|^2 \quad \text{und}$$

$$\sum_{j\neq k}|\lambda_j - \lambda_k|^2|\xi_j|^2 = \|Ax - \lambda_k x\|_2^2. \qquad \blacksquare$$

Insbesondere für hermiteschen Matrizen kann die Eigenwert-Näherung bei der *Vektor-Iteration* durch Verwendung des Rayleigh-Quotienten verbessert werden. Für die modifizierte Iteration

$$\left.\begin{aligned} z^{(k)} &:= Aq^{(k-1)}, \\ \lambda^{(k)} &:= q^{(k-1)*}z^{(k)} \quad = R_A[q^{(k-1)}], \\ q^{(k)} &:= z^{(k)}/\|z^{(k)}\|_2 \end{aligned}\right\} k = 1, 2, \ldots \tag{3.4.3}$$

mit $\|q^{(0)}\|_2 = 1$ steht in der Konvergenzaussage

$$|\lambda^{(k)} - \lambda_1| \leq c\left|\frac{\lambda_2}{\lambda_1}\right|^{2k}, \; k \geq k_0,$$

im Vergleich zu Satz 3.3.1 der doppelte Exponent $2k$. Die verbesserte Eigenwert-Schätzung wirkt sich noch viel günstiger auf die *Inverse Iteration* (3.3.4) aus, wenn der Rayleigh-Quotient bei der Spektralverschiebung eingesetzt wird. Es sei $\|q^{(0)}\|_2 = 1,\ \lambda^{(0)} := R_A[q^{(0)}]$, und

$$\left.\begin{array}{rcl} z^{(k)} & := & (A-\lambda^{(k-1)}I)^{-1}q^{(k-1)}, \\ \lambda^{(k)} & := & \lambda^{(k-1)} + z^{(k)*}q^{(k-1)}/\|z^{(k)}\|_2^2, \\ q^{(k)} & := & z^{(k)}/\|z^{(k)}\|_2 \end{array}\right\} \ k = 1,2,\ldots \tag{3.4.4}$$

Dabei ist in der zweiten Zeile

$$\lambda^{(k)} = R_A[z^{(k)}] = \frac{z^{(k)*}Az^{(k)}}{\|z^{(k)}\|_2^2} = \frac{1}{\|z^{(k)}\|_2^2} z^{(k)*}(A-\lambda^{(k-1)}I)z^{(k)} + \lambda^{(k-1)} = \frac{z^{(k)*}q^{(k-1)}}{\|z^{(k)}\|_2^2} + \lambda^{(k-1)}.$$

Dieses Verfahren (3.4.4) hat jetzt extrem gute Konvergenzeigenschaften.

Satz 3.4.3 *Bei einer hermiteschen Matrix A konvergiert die Inverse Iteration (3.4.4) für fast alle Startvektoren $q^{(0)}$ gegen einen Eigenvektor x von A mit Eigenwert λ (globale Konvergenz). Bei einfachen Eigenwerten ist die Konvergenz in der Grenze kubisch, d.h., es gibt $k_0 \in \mathbb{N}, c \in \mathbb{R}$ und $\sigma_k \in \mathbb{C}$ so, daß für $k \geq k_0$ gilt*

$$\begin{array}{rcl} |\lambda^{(k)} - \lambda| & \leq & c|\lambda^{(k-1)} - \lambda|^3 \\ \|\sigma_k q^{(k)} - x\|_2 & \leq & c\|\sigma_{k-1}q^{(k-1)} - x\|_2^3 \end{array}$$

Beweis [Parlett, Math. Comp. 28(74), 679-693]

Zur Erläuterung dient ein einfacher Spezialfall und eine Querverbindung zum Newtonverfahren.

Beispiel 3.4.4 $n = 2,\ A = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$ (oBdA), $q^{(k-1)} = \begin{pmatrix} c_{k-1} \\ s_{k-1} \end{pmatrix}$ mit $c_{k-1}^2 + s_{k-1}^2 = 1$. Dann ist (vgl. Beweis von Satz 3.4.2)

$$\lambda^{(k-1)} = R_A[z^{(k-1)}] = R_A[q^{(k-1)}] = \lambda_1 c_{k-1}^2 + \lambda_2 s_{k-1}^2,$$

$$z^{(k)} = \begin{pmatrix} c_{k-1}/(\lambda_1 - \lambda^{(k-1)}) \\ s_{k-1}/(\lambda_2 - \lambda^{(k-1)}) \end{pmatrix} = \frac{1}{\lambda_1 - \lambda_2}\begin{pmatrix} c_{k-1}/s_{k-1}^2 \\ -s_{k-1}/c_{k-1}^2 \end{pmatrix},$$

da, z.B., $\lambda_1 - \lambda^{(k-1)} = \lambda_1(1 - c_{k-1}^2) - \lambda_2 s_{k-1}^2 = (\lambda_1 - \lambda_2)s_{k-1}^2$. Die Normierung führt dann auf

$$q^{(k)} = \frac{\pm 1}{\sqrt{c_{k-1}^6 + s_{k-1}^6}}\begin{pmatrix} c_{k-1}^3 \\ -s_{k-1}^3 \end{pmatrix} \longrightarrow \pm\left\{\begin{array}{ll} e^{(1)} & \text{für } |c_0| > |s_0| \\ e^{(2)} & \text{für } |c_0| < |s_0| \end{array}\right. .$$

Mit Ausnahme der Fälle mit $|c_0| = |s_0|$ konvergiert die Inverse Iteration also kubisch gegen einen der beiden Einheits- und Eigenvektoren.

Beispiel 3.4.5 Wird zur Eigenwert-Gleichung (3.4.1) noch eine Normierungsbedingung für den Eigenvektor hinzugefügt, ergibt sich ein nichtlineares Gleichungssystem für die $n+1$ Unbekannten x, λ. Zu

$$F(x,\lambda) := \begin{pmatrix} Ax - \lambda x \\ \frac{1}{2}(1 - x^*x) \end{pmatrix} = 0, \quad \text{ist } F'(x,\lambda) = \begin{pmatrix} A - \lambda I & -x \\ -x^* & 0 \end{pmatrix}.$$

Das Gleichungssystem eines Newton-Schritts,

$$F'\Big(x^{(k-1)},\lambda^{(k-1)}\Big)\begin{pmatrix} x^{(k)}-x^{(k-1)} \\ \lambda^{(k)}-\lambda^{(k-1)}\end{pmatrix} = -F\Big(x^{(k-1)},\lambda^{(k-1)}\Big) \quad \Leftrightarrow$$

$$\begin{cases} (A-\lambda^{(k-1)}I)x^{(k)} & -\lambda^{(k)}x^{(k-1)} & = -\lambda^{(k-1)}x^{(k-1)} \\ \quad x^{(k-1)^*}x^{(k)} & & = \frac{1}{2}+\frac{1}{2}\|x^{(k-1)}\|_2^2 \end{cases}$$

kann in folgender Weise gelöst werden

$$\begin{cases} z^{(k)} & := & (A-\lambda^{(k-1)}I)^{-1}x^{(k-1)}, \\ \lambda^{(k)} & := & \lambda^{(k-1)}+\frac{1}{2}(1+\|x^{(k-1)}\|^2)/(x^{(k-1)*}z^{(k)}), \\ x^{(k)} & := & (\lambda^{(k)}-\lambda^{(k-1)})z^{(k)}. \end{cases}$$

Das (quadratisch konvergente) Newtonverfahren unterscheidet sich hier also nur durch die fehlende exakte Normierung (i.a. $\|x^{(k)}\| \neq 1$) von der kubisch konvergenten Inversen Iteration.

Auch beim *QR-Verfahren* können die Aussagen gegenüber dem unsymmetrischen Fall verbessert werden. Nach Satz 3.2.2 kann oBdA für A Tridiagonalgestalt angenommen werden. Daher reduziert sich schon der Rechenaufwand pro QR-Schritt auf ca. $28n$ Operationen und n Quadratwurzeln.

Wegen des engen Zusammenhangs zwischen Vektoriteration und QR-Verfahren überträgt sich die lokal kubische Konvergenz auf (3.3.10),(3.3.12), denn es gilt

$$\lambda^{(k-1)} = s_{nn}^{(k-1)} = R_{S^{(k-1)}}[e^{(n)}].$$

Allerdings kann bei dieser Wahl in bestimmten Fällen zu Beginn nur lineare Konvergenz eintreten. In der Startphase läßt sich die Konvergenz noch weiter verbessern dadurch, daß als Spektralparameter derjenige Eigenwert des letzten 2×2-Blocks

$$\begin{pmatrix} s_{n-1,n-1}^{(k-1)} & s_{n-1,n}^{(k-1)} \\ s_{n,n-1}^{(k-1)} & s_{n,n}^{(k-1)} \end{pmatrix}$$

gewählt wird, der näher an $s_{nn}^{(k-1)}$ liegt. Mit $\delta_{k-1} := \frac{1}{2}(s_{nn}^{(k-1)} - s_{n-1,n-1}^{(k-1)})$ ist dies

$$\lambda^{(k-1)} = \frac{1}{2}(s_{nn}^{(k-1)} + s_{n-1,n-1}^{(k-1)}) + sign(\delta_{k-1})\sqrt{\delta_{k-1}^2 + {s_{n-1,n}^{(k-1)}}^2}. \tag{3.4.5}$$

Für diese Wahl kann sogar folgende globale Konvergenzaussage gemacht werden. Eine Tridiagonalmatrix heißt dabei *unzerlegbar*, wenn $a_{i,i-1}\neq 0\ \forall i = 2,\dots,n$. Unzerlegbare Tridiagonalmatrizen besitzen nur einfache Eigenwerte.

Satz 3.4.6 (Wilkinson) *Für eine unzerlegbare symmetrische Tridiagonalmatrix A konvergiert das QR-Verfahren (3.3.10) mit Spektralverschiebung (3.4.5) global, also ohne Startschritte. Die Konvergenz ist mindestens quadratisch, d.h., $\exists \varepsilon_0 > 0$ so, daß*

$$|s_{n-1,n}^{(k-1)}| \le \varepsilon \le \varepsilon_0 \quad \Rightarrow \quad |s_{n-1,n}^{(k)}| \le c\frac{\varepsilon^3}{s_{n-1,n-1}^{(k-1)} - \lambda^{(k-1)}} \le c\,\varepsilon^2.$$

Mit dieser QR-Variante können (als empirischer Wert) alle Eigenwerte mit *ca.* $\frac{4}{3}n^3$ Operationen und die volle Schur-Normalform mit ca. $10n^3$ Operationen berechnet werden.

Außer den Modifikationen der allgemein anwendbaren Verfahren gibt es weitere Spezialverfahren für das symmetrische Eigenwert-Problem. Diese besitzen in bestimmten Situationen Vorteile gegenüber den Standardverfahren und sind daher ebenfalls von Interesse.

Bisektionsverfahren

Dieses Verfahren erlaubt die gezielte Berechnung einzelner Eigenwerte der Matrix. Dabei läßt sich die Nummer oder die Lage des Eigenwerts vorgeben. Die Grundlage des Verfahrens ist die Tatsache, daß die Unterdeterminanten (Haupt-Minoren)

$$p_k(\lambda) := \det \begin{pmatrix} a_{11}-\lambda & a_{12} & & & \\ a_{21} & a_{22}-\lambda & a_{23} & & \\ & \ddots & \ddots & \ddots & \\ & & \cdot & \cdot & a_{k-1,k} \\ & & & a_{k,k-1} & a_{kk}-\lambda \end{pmatrix}, \quad k = 1, \ldots, n, \tag{3.4.6}$$

$p_0 :\equiv 1$, einer symmetrischen Tridiagonalmatrix eine sogenannte *Sturmsche Kette* bilden.

Definition 3.4.7 *Die Folge $p_0 \equiv 1, p_1, \ldots, p_n$ von reellen Polynomen wachsenden Grads bildet eine* Sturmsche Kette, *wenn folgende Bedingungen gelten:*

1. *Die Nullstellen von p_n sind einfach.*
2. *Für jede reelle Nullstelle ξ von p_n gilt sign $p_{n-1}(\xi) = -$sign $p'_n(\xi)$ $(\neq 0)$.*
3. *Für jede reelle Nullstelle ξ von p_i, $i = 1, \ldots, n-1$, gilt*

$$p_{i-1}(\xi)p_{i+1}(\xi) < 0.$$

Für die Polynome einer Sturmschen Kette kann die Verteilung der reellen Nullstellen an der Vorzeichenverteilung der Folge $(p_j(\hat{x}))_j$ an einer festen Stelle $\hat{x}$ abgelesen werden. Dazu sei

$$w(\hat{x}) := \text{ Anzahl der Vorzeichenwechsel in } (p_0(\hat{x}), p_1(\hat{x}), \ldots, p_n(\hat{x})) \tag{3.4.7}$$

mit der Verabredung, daß zuvor alle verschwindenden Werte $p_j(\hat{x}) = 0$ aus der Folge entfernt werden. Dann gilt

Satz 3.4.8 *Die Anzahl der reellen Nullstellen des Polynoms p_n einer Sturmschen Kette im Intervall $[a, b]$ ist gleich*

$$w(b) - w(a).$$

Beweis [Stoer, Numer. Math. 1]

Um den Satz auf das Eigenwertproblem anwenden zu können, ist nachzuweisen, daß die Haupt-Minoren p_k aus (3.4.6) tatsächlich die Bedingungen der Definition 3.4.7 erfüllen. Dazu, und zur numerischen Auswertung dient folgende Rekursionsformel.

$$\begin{aligned} p_0(\lambda) &:= 1,\ p_1(\lambda) := a_{11} - \lambda, \\ p_{k+1}(\lambda) &:= (a_{k+1,k+1} - \lambda)p_k(\lambda) - |a_{k+1,k}|^2 p_{k-1}(\lambda),\ k = 1, \dots, n-1. \end{aligned} \tag{3.4.8}$$

Beweis Entwicklung nach der letzten Zeile in (3.4.6):

$$p_{k+1} = (a_{k+1,k+1} - \lambda)p_k - a_{k+1,k} \begin{vmatrix} \ddots & & \vdots \\ & a_{k-1,k-1} - \lambda & 0 \\ \cdots & a_{k,k-1} & a_{k,k+1} \end{vmatrix}. \qquad \blacksquare$$

Also kann insbesondere der Wert $p_n(\lambda)$ mit $4n$ Operationen bestimmt werden.

Beispiel 3.4.9

$$T = \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 3 & -1 \\ 0 & 0 & -1 & 4 \end{pmatrix} \quad \Rightarrow \{\lambda_j\} \doteq \{0.25, 1.82, 3.18, 4.74\}.$$

In $\hat{x} = 2$ ist $(p_0(2), \dots, p_4(2)) = (1, -1, -1, 0, 1)$. Bei der Zählung der Vorzeichenwechsel wird die Null gestrichen, es ergibt sich $w(2) = 2$. Wegen $w(-\infty) = 0$ liegen 2 Eigenwerte links von $\hat{x} = 2$.

Satz 3.4.10 *Die Matrix $A \in \mathbb{C}^{n\times n}$ sei eine unzerlegbare, hermitesche Tridiagonalmatrix. Dann gelten für die Polynome p_k, $k = 0, \dots, n$, aus (3.4.6), (3.4.8) die Aussagen*

a) $p_k \in \Pi_k, \quad p_k(\lambda) = (-\lambda)^k + \dots,$ *d.h.,* $p_k(\lambda) \to \begin{cases} \infty, & \lambda \to -\infty \\ (-1)^k\infty, & \lambda \to \infty \end{cases}$

b) Das Polynom p_k, $k > 0$, hat k reelle, einfache Nullstellen $\lambda_1^{(k)} < \dots < \lambda_k^{(k)}$, die durch die Nullstellen von p_{k-1} getrennt werden,

$$\lambda_1^{(k)} < \lambda_1^{(k-1)} < \lambda_2^{(k)} < \dots < \lambda_{k-1}^{(k-1)} < \lambda_k^{(k)}.$$

c) $p_k(\xi) = 0 \quad \Rightarrow \quad p_{k-1}(\xi)p_{k+1}(\xi) < 0.$

d) $\mathit{sign}\ p_{k-1}(\lambda_j^{(k)}) = -\mathit{sign}\ p_k'(\lambda_j^{(k)}),\ j = 1, \dots, k.$

Beweis a) Folgt aus (3.4.8).

c) Aus (3.4.8) folgt für $p_k(\xi) = 0$ direkt $p_{k+1}(\xi) = -|a_{k+1,k}|^2 p_{k-1}(\xi)$. Dabei ist $a_{k+1,k} \neq 0$ n.V., also gilt $p_{k+1}(\xi)p_{k-1}(\xi) \le 0$. Keiner der beiden Werte kann aber verschwinden, da sonst aus (3.4.8) $p_0(\xi) = 0$ folgen würde.

b) Die Nullstellen sind reell, da die p_k charakteristische Polynome symmetrischer Matrizen sind.

Die Trennungseigenschaft folgt induktiv dadurch, daß für jedes Polynom die Vorzeichenverteilung in und zwischen den Nullstellen aufgelistet wird:

$$\left\{\begin{array}{lccccccc}
p_0: & & & & + & & & \\
p_1: & & & + & 0 & - & & \\
p_2: & & + & 0 & - & 0 & + & \\
p_3: & + & 0 & - & 0 & + & 0 & -
\end{array}\right. \tag{3.4.9}$$

Dabei folgt die Vorzeichenverteilung am linken und rechten Rand aus a), die vertikalen Pfeile deuten die Folgerung c) an, auf die Existenz der Nullstellen schließlich kann aus den Vorzeichenwechseln geschlossen werden. Da bei p_k jeweils k Nullstellen auftreten, sind alle einfach.

d) Folgt ebenfalls aus (3.4.9): $\operatorname{sign} p_{k-1}(\lambda_j^{(k)}) = \operatorname{sign}\left(p_k(\lambda_j^{(k)} - \epsilon) - p_k(\lambda_j^{(k)} + \epsilon)\right)$. ∎

Mit Hilfe von Satz 3.4.8 läßt sich ein sehr einfaches Verfahren zur Bestimmung eines beliebigen Eigenwerts λ_m, $1 \le m \le n$, angeben. Ausgehend von einem Startintervall $[a, b]$, z.B., $[a, b] = [-\|A\|, \|A\|]$, wird die Einschließung von λ_m durch *Bisektion* (vgl. Numerik I §6.1) anhand der Vorzeichenwechsel w schrittweise verbessert:

$$\boxed{\begin{array}{l}
\text{wiederhole:}\\
\quad \lambda := (a+b)/2;\\
\quad \text{falls } w(\lambda) \ge m \text{ setze } b := \lambda\\
\qquad \text{sonst } a := \lambda;\\
\text{bis } |a-b| \le \mathcal{E}(|a|+|b|);
\end{array}} \tag{3.4.10}$$

Bessere a-priori-Einschließungen der Eigenwerte werden im nächsten Teilparagraphen behandelt. Jedes der erzeugten Teilintervalle enthält wegen $w(-\infty) = 0$ den Eigenwert λ_m. Die Konvergenz des Bisektionsverfahrens (3.4.10) ist im Vergleich zur Endkonvergenz der anderen Verfahren recht langsam, da die Intervallänge in jedem Schritt nur halbiert wird (⇒ Konvergenzfaktor $= \frac{1}{2}$). Es bietet sich aber an, gerade bei nahe zusammenliegenden Eigenwerten, zur Bestimmung eines Startintervalls, das nur λ_m enthält. Die genaue Berechnung dieses Eigenwertes und des zugehörigen Eigenvektors kann dann schneller mit der Inversen Iteration vorgenommen werden.

Jacobi-Verfahren

Das Grundprinzip dieses Verfahrens besteht darin, durch einfache unitäre Ähnlichkeitstransformationen einzelne Nebendiagonalelemente der Ausgangsmatrix (d.h. ohne Vortransformation auf Tridiagonalgestalt) zu Null zu machen. Das Verfahren ist zwar mit ca. $32n^3$ Operationen aufwendiger als das QR-Verfahren mit $\frac{4}{3}n^3$ Operationen, hat aber den Vorteil eines einfachen Aufbaus und der Parallelisierbarkeit (→ Superrechner). Zur Transformation werden bei reellen

Matrizen elementare Drehungen der Gestalt

$$J(p,q;\vartheta) := \begin{pmatrix} 1 & & & & & & & & \\ & \ddots & & & & & & & \\ & & 1 & & & & & & \\ & & & c & & & s & & \\ & & & & 1 & & & & \\ & & & & & \ddots & & & \\ & & & & & & 1 & & \\ & & & -s & & & c & & \\ & & & & & & & 1 & \\ & & & & & & & & \ddots \\ & & & & & & & & & 1 \end{pmatrix}, \tag{3.4.11}$$

mit $c := \cos\vartheta,\ s := \sin\vartheta$, benutzt. Die Einträge c und $\pm s$ stehen dabei in den Zeilen und Spalten p und q. Den Winkel ϑ bestimmt man so, daß die Elemente $b_{pq} = b_{qp}$ der transformierten Matrix

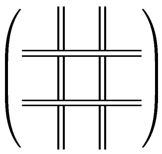
$$B := J^* A J \tag{3.4.12}$$

verschwinden. Die Wirkung der beiden Multiplikationen ist, mit $A' := J^* A$,

$$\left.\begin{array}{l} a'_{pj} = c a_{pj} - s a_{qj} \\ a'_{qj} = s a_{pj} + c a_{qj} \end{array}\right\} \; j = 1, \ldots, n, \quad a'_{ij} = a_{ij} \text{ für } i \neq p, q,$$

$$\left.\begin{array}{l} b_{ip} = c a'_{ip} - s a'_{iq} \\ b_{iq} = s a'_{ip} + c a'_{iq} \end{array}\right\} \; i = 1, \ldots, n, \quad b_{ij} = a'_{ij} \text{ für } j \neq p, q,$$

Die Transformation (3.4.12) ändert A also nur im schraffierten Bereich

$B :=$  .

Insbesondere ergeben sich

$$\begin{aligned} b_{pp} &= c^2 a_{pp} - 2cs a_{pq} + s^2 a_{qq}, \\ b_{qq} &= s^2 a_{pp} + 2cs a_{pq} + c^2 a_{qq}, \\ b_{qp} = b_{pq} &= (a_{pp} - a_{qq})cs + a_{pq}(c^2 - s^2) \stackrel{!}{=} 0. \end{aligned} \tag{3.4.13}$$

Natürlich gilt $b_{pp} + b_{qq} = a_{pp} + a_{qq}$. Das Verschwinden von b_{pq} in der dritten Gleichung ist äquivalent mit

$$\frac{1}{2}(a_{pp} - a_{qq}) \sin 2\vartheta + a_{pq} \cos 2\vartheta = 0.$$

Für den nichttrivialen Fall $a_{pq} \neq 0$ definiert dies den Winkel ϑ (oBdA $|\vartheta| \leq \frac{\pi}{4}$) durch

$$\cot 2\vartheta = \frac{a_{qq} - a_{pp}}{2a_{pq}} =: z. \tag{3.4.14}$$

Für die Transformation werden nicht ϑ, sondern $\cos\vartheta$ und $\sin\vartheta$ benötigt. Für $t := \tan\vartheta$ ist

$$\cot 2\vartheta = \frac{1-t^2}{2t} = z \quad \Longleftrightarrow \quad t^2 - 2zt - 1 = 0.$$

Die betragskleinere Lösung dieser Gleichung (mit $0 < |t| \leq 1$) ist

$$t = \begin{cases} 1, & z = 0, \\ \dfrac{\operatorname{sign} z}{|z| + \sqrt{1+z^2}}, & z \neq 0, \end{cases}$$

aus der sich die Transformationsparameter ableiten,

$$c = \frac{1}{\sqrt{1+t^2}} = \cos\vartheta, \quad s = tc = \sin\vartheta.$$

Die Transfomationsformeln (3.4.13) können noch vereinfacht werden,

$$b_{pp} = a_{pp} - ta_{pq}, \; b_{qq} = a_{qq} + ta_{pq}, \; b_{pq} = b_{qp} = 0, \tag{3.4.15}$$

während sich für die restlichen Elemente wegen der Symmetrie ergibt

$$\left.\begin{array}{rcl} b_{pi} = b_{ip} & = & ca_{ip} - sa_{iq} \\ b_{qi} = b_{iq} & = & sa_{ip} + ca_{iq} \end{array}\right\} \begin{array}{c} i = 1, \ldots, n, \\ i \neq p, q. \end{array} \tag{3.4.16}$$

Rechenaufwand pro Jacobi-Schritt: $6n$ Operationen.
Zur iterativen Durchführung des Jacobi-Verfahrens ist eine Auswahl der Matrix-Elemente a_{pq} durchzuführen. In Frage kommen (u.a.) zwei Strategien

- klassische Strategie: Das betragsgrößte Nebendiagonalelement wird auf Null transformiert,

  $$|a_{pq}| = \max_{j \neq i} |a_{ij}|.$$

  Die Maximumbestimmung kann allerdings aufwendiger sein als die Transformation.

- Zyklische Strategie: Alle Nebendiagonalelemente werden der Reihe nach bearbeitet, $p = 1, 2, \ldots, n-1, \; q = p+1, \ldots, n$. Modifikation: zu kleine Elemente werden übersprungen.

Eine Analyse der **Konvergenz** des Jacobi-Verfahrens ist einfach möglich durch Betrachtung der Quadratsumme aller Nebendiagonalelemente,

$$S(A) := \sum_{j>i} a_{ij}^2. \tag{3.4.17}$$

Wegen der Orthogonalität der Matrix J gilt nach (3.4.16)

$$\left.\begin{array}{rcl} b_{ip}^2 + b_{iq}^2 & = & a_{ip}^2 + a_{iq}^2 \\ b_{pj}^2 + b_{qj}^2 & = & a_{pj}^2 + a_{qj}^2 \end{array}\right\} i, j \neq p, q.$$

Da durch die Transformation a_{pq} eliminiert wird, ist

$$S(B) = S(J^*AJ) = S(A) - a_{pq}^2. \tag{3.4.18}$$

Insbesondere wird die nichtnegative Größe S verkleinert. Daraus folgt

Satz 3.4.11 *Beim Jacobi-Verfahren,* $A^{(0)} := A$,

$$A^{(k+1)} := J^{(k)*}A^{(k)}J^{(k)},\ \ J^{(k)} := J(p,q,\vartheta),$$

mit klassischer oder zyklischer Strategie konvergiert die Folge $A^{(k)}$ *gegen eine Diagonalmatrix der Eigenwerte von* A. *Beim klassischen Verfahren gilt die a-priori-Schranke*

$$S(A^{(k)}) \le \Big(1 - \frac{2}{n(n-1)}\Big)^k S(A).$$

Beweis (klassische Strategie): Nach (3.4.18) ist $S(A^{(k)})$ eine streng monoton fallende, nach unten durch null beschränkte Folge. Außerdem gilt hier, bei Wahl des maximalen Matrixelements

$$S(A^{(k)}) \le \frac{n(n-1)}{2} a_{pq}^{(k)2} \ \Rightarrow\ S(A^{(k+1)}) \le \Big(1 - \frac{2}{n(n-1)}\Big) S(A^{(k)}).$$

Die Hauptdiagonalelemente konvergieren, da nach (3.4.15) ihre Änderung pro Schritt durch $|a_{pq}^{(k)}| \le \sqrt{S(A^{(k)})}$ beschränkt ist. ■

Bei einer Matrix mit nur einfachen Eigenwerten, bei denen also

$$d := \min_{i \ne j} |\lambda_i - \lambda_j| > 0$$

positiv ist, tritt bei beiden Strategien sogar quadratische Konvergenz ein, wenn volle Zyklen, d.h., $N := \frac{1}{2}n(n-1)$ Schritte betrachtet werden. Dann gilt

$$S(A^{(k+N)}) \le \frac{c_n}{d^2}\, S(A^{(k)})^2.$$

Wenn $S(A^{(k)})$ klein ist im Vergleich zu $\|A\|$ können mit den Sätzen 3.4.2, 3.6.3 oder 3.6.5 Schranken für die Eigenwerte von A angegeben werden.

Beispiel 3.4.12 Die Eigenwerte der Matrix

$$A := \begin{pmatrix} 20 & -7 & 3 & -2 \\ -7 & 5 & 1 & 4 \\ 3 & 1 & 3 & 1 \\ -2 & 4 & 1 & 2 \end{pmatrix}$$

sind $\{\lambda_j\} \doteq \{23.5, 6.46, 1.17, -1.16\}$, also $d \doteq 2.3$. Das zyklische Verfahren liefert

$k =$	0	6	12	18	24
$S(A^{(k)}) =$	80	2.9	$6.9_{10}-3$	$5.5_{10}-10$	$1.9_{10}-31$

3.5 Überblick der Eigenwert-Verfahren

Die folgende Tabelle faßt die Eigenschaften und die Einsatzbereiche der behandelten Verfahren zusammen. Das als letztes erwähnte Lanczos-Verfahren wird im nächsten Kapitel besprochen.

Verfahren für Eigenwertprobleme

Verfahren	Unsym. EWP	Symm. EWP	für Poblemklasse
Vektor-Iteration	+	+	größter EW u. EV, v.a. dünnbesetzte Matrix
Inverse Iteration	evtl. problematisch	kubisch u. global konvergent	zu λ_0 nächster EW u. EV, z.B. nach QR-Verfahren, Bisektion
orthogonale Iteration	+	+	Gruppe größter EWe, invarianter Unterraum
QR-Verfahren	+	kubisch u. global konvergent	Standardverfahren für volle Matrix: Schur-Normalform
Bisektion	$-$	lineare Konvergenz	für k-ten EW, EW-Haufen, kein EV ($\rightarrow$ Inverse Iteration)
Jacobi-Verfahren	$-$	quadratische Konvergenz	alle EWe: Schurnormalform, parallelisierbar
Lanczos-Verfahren	$-$	+	größte EWe, schnelle Approximation bei dünnbesetzter Matrix

3.6 Eigenwert- und Fehler-Schranken

Wie in der Numerik I wird ein Algorithmus als *gutartig* bezeichnet, wenn er die exakte Lösung eines nur leicht gestörten Ausgangsproblems liefert. Große Fehler in den berechneten Lösungen sind bei solchen gutartigen Verfahren auf eine zu große Fehleranfälligkeit (schlechte Kondition) des Ausgangsproblem zurückzuführen.

In diesem Sinn sind die behandelten Verfahren gutartig. Bei den Transformationsverfahren mit unitären Matrizen (Hessenberg-Transf., QR-, Jacobi-Verfahren) etwa gilt, daß das Ergebnis bei numerischer Rechnung statt der exakten Matrix $T = Q^*AQ$ eine Matrix $\tilde{T}$ ist mit

$$\tilde{T} = Q^*(A + F)Q, \quad \|F\|_2 \le c\|A\|_2 \cdot \mathcal{E}. \tag{3.6.1}$$

Dabei ist $\mathcal{E}$ die Maschinengenauigkeit und c eine harmlose Konstante, die i.w. von der Zahl der Rechenoperationen abhängt. Diese überschaubaren Störungen des *Problems* können, abhängig von den Eigenschaften der Matrix, sehr unterschiedliche Störungen der *Lösungen* beim Eigenwertproblem zur Folge haben. Das symmetrische Eigenwertproblem ist dabei sehr unempfindlich gegen Störungen, also gut konditioniert. Dies folgt aus

Satz 3.6.1 *(Courantsches Maximum-Minimum-Prinzip) Die Eigenwerte der hermiteschen Matrix $A \in \mathbb{C}^{n\times n}$ seien geordnet nach $\lambda_1 \ge \lambda_2 \ge \ldots \ge \lambda_n$. Dann gilt*

$$\lambda_j = \max_{\dim S = j} \min_{0 \ne y \in S} R_A[y], \quad j = 1, \ldots, n,$$

wobei das Maximum über lineare Unterräume $S \subseteq \mathbb{C}^{n\times n}$ zu bilden ist.

Sobald in den folgenden Aussagen verschiedene Matrizen $A, B, \ldots$ auftreten, werden deren Eigenwerte mit $\lambda_j(A), \lambda_j(B), \ldots$ bezeichnet, sie seien wie in Satz 3.6.1 fallend angeordnet. Eine direkte Folgerung des Minimax-Prinzips ist

Satz 3.6.2 *Sind A und $A+F$ hermitesch, dann gilt für $j = 1, \ldots, n$*

$$\lambda_j(A) + \lambda_n(F) \le \lambda_j(A+F) \le \lambda_j(A) + \lambda_1(F).$$

Beweis für die untere Schranke: Da $R_{A+F}[y] = R_A[y] + R_F[y]$ ist, gilt

$$\begin{aligned} \lambda_j(A+F) &= \max_{\dim S=j} \min_{y\in S} \Big(R_A[y] + R_F[y]\Big) \ge \max_{\dim S=j} \Big(\min_{y\in S} R_A[y] + \min_{z\in S} R_F[z]\Big) \\ &\ge \max_{\dim S=j} \min_{y\in S} R_A[y] + \min_{z\in\mathbb{C}^n} R_F[z]. \end{aligned}$$

Die letzte Größe stimmt mit der linken Seite der Behauptung überein. ■

Die Störung der Eigenwerte von A hat insbesondere höchstens den Betrag $\varrho(F) = \|F\|_2$, es gilt also $|\lambda_j(A+F) - \lambda_j(A)| \le \|F\|_2$. Bei den Eigenvektoren kann die Störung allerdings größer ausfallen. Im folgenden Satz wird davon ausgegangen, daß eine Näherung x für einen Eigenvektor und λ für den zugehörigen Eigenwert (z.B. durch den Rayleigh-Quotienten) bekannt ist.

Satz 3.6.3 *Gegeben sei eine hermitesche Matrix A und ein Paar (x, λ) mit $x \ne 0$. Damit sei*

$$\|Ax - \lambda x\|_2 \le \varepsilon \|x\|_2 \quad \textit{bzw.} \quad \|Ax - \lambda x\|_2 \le \delta \|Ax\|_2. \tag{3.6.2}$$

a) Dann gilt

$$\min_j |\lambda_j - \lambda| \le \varepsilon \quad \textit{bzw.} \quad \min_{\lambda_j \ne 0} |1 - \frac{\lambda}{\lambda_j}| \le \delta.$$

b) Ist λ_k der zu λ nächstgelegene Eigenwert, bei dem also $\min_j |\lambda_j - \lambda|$ angenommen wird, und ist dieser einfach mit $d_k := \min_{j\ne k} |\lambda_j - \lambda_k| > 0$, dann gilt

$$\min_{\xi\in\mathbb{C}} \|x - \xi x^{(k)}\|_2 \le 2\frac{\varepsilon}{d_k} \|x\|_2.$$

Beweis Da die Eigenvektoren eine Orthonormalbasis bilden, gilt für $x = \sum_{j=1} \xi_j x^{(j)}$ in Teil a):

$$\min_j (\lambda_j - \lambda)^2 \sum_j |\xi_j|^2 \le \sum_{j=1}^n (\lambda_j - \lambda)^2 |\xi_j|^2 = \|Ax - \lambda x\|_2^2 \le \varepsilon^2 \|x\|_2^2 = \varepsilon^2 \sum_j |\xi_j|^2.$$

Für die zweite Ungleichung ergibt sich mit $\eta_j := \xi_j \lambda_j$:

$$\begin{aligned} \min_{\lambda_j\ne 0} (1 - \lambda/\lambda_j)^2 \sum_j |\eta_j|^2 &\le \sum_{\lambda_j \ne 0} (1 - \lambda/\lambda_j)^2 |\lambda_j \xi_j|^2 + \lambda^2 \sum_{\lambda_j = 0} |\xi_j|^2 \\ &= \|Ax - \lambda x\|_2^2 \le \delta^2 \|Ax\|_2^2 = \delta^2 \sum_j |\eta_j|^2. \end{aligned}$$

b) Nach Teil a) folgt

$$\varepsilon\|x\|_2 \geq \|Ax - \lambda_k x\|_2 - |\lambda - \lambda_k|\|x\|_2 \geq \|Ax - \lambda_k x\|_2 - \varepsilon\|x\|_2.$$

Dies bedeutet

$$4\varepsilon^2\|x\|_2^2 \geq \|Ax - \lambda_k x\|_2^2 = \sum_{j\neq k}(\lambda_j - \lambda_k)^2|\xi_j|^2 \geq d_k^2 \sum_{j\neq k}|\xi_j|^2 = d_k^2\|x - \xi_k x^{(k)}\|_2^2.$$

Wegen der Orthogonalität der Eigenvektor-Basis wird $\min_\xi \|x - \xi x^{(k)}\|_2$ in ξ_k angenommen. ■

Bemerkung: 1) Nach Satz 3.4.2 ist der Rayleighquotient $R_A[x]$ die beste Eigenwert-Schätzung zur Eigenvektor-Näherung x. Aus dem Beweis dieses Satzes folgt (bei $\|x\|_2 = 1$)

$$\varepsilon^2 = \|Ax - R_A[x]x\|_2^2 = \|Ax\|_2^2 - R_A[x]^2,$$

und für beliebige λ die Beziehung $\|Ax - \lambda x\|_2^2 = \varepsilon^2 + |\lambda - R_A[x]|^2$. Daher gilt

$$\min_j |\lambda_j - R_A[x]| \leq \frac{1}{d_k}\|Ax - \lambda_k x\|_2^2 = \frac{1}{d_k}(|\lambda_k - R_A[x]|^2 + \varepsilon^2) \leq 2\frac{\varepsilon^2}{d_k}. \qquad (3.6.3)$$

Für $\varepsilon < d_k/2$ ist diese Schranke besser als die aus Satz 3.6.3.

2) Die in der ersten Bemerkung verwendete Größe ε ist bei der Vektoriteration (3.4.3) bzw. (3.4.4) einfach verfügbar. Bei der Inversen Iteration (3.4.4) etwa gilt mit $\lambda^{(k)} = R_A[z^{(k)}]$ und $B_{k-1} := A - \lambda^{(k-1)}I$:

$$\begin{aligned}
\varepsilon^2\|z^{(k)}\|_2^2 &= \|Az^{(k)} - \lambda^{(k)}z^{(k)}\|_2^2 = \|Az^{(k)} - \lambda^{(k-1)}z^{(k)} - (\lambda^{(k)} - \lambda^{(k-1)})z^{(k)}\|_2^2 \\
&= \|B_{k-1}z^{(k)} - R_{B_{k-1}}[z^{(k)}]z^{(k)}\|_2^2 = \|q^{(k-1)}\|_2^2 - (z^{(k)*}q^{(k-1)}/\|z^{(k-1)}\|_2)^2 \\
&= 1 - (z^{(k)*}q^{(k-1)}/\|z^{(k-1)}\|_2)^2.
\end{aligned}$$

Alle hier auftretenden Größen werden bei der Inversen Iteration sowieso berechnet, mit Satz 3.6.3 steht also ein einfaches Abbruchkriterium zur Verfügung.

3) Beim symmetrischen QR-Verfahren sind die Einheitsvektoren Näherungs-Eigenvektoren der Matrizen $S^{(k)}$. Für $x = e^{(n)}$ ist $R_S[x] = s_{nn}$ und $\|Sx - s_{nn}x\|_2 = |s_{n-1,n}| =: \varepsilon$. Ein Eigenwert von S unterscheidet sich von s_{nn} also höchstens um den Betrag $|s_{n-1,n}|$.

Beispiel 3.6.4 Die Eigenwerte von

$$A = \begin{pmatrix} 1 & 2 & 7/3 \\ 2 & 2 & 3 \\ 7/3 & 3 & 3 \end{pmatrix}$$

sind $\{\lambda_j\} \doteq \{7.09, -0.5293, -0.562\}$, daher $d_2 \cong 1/30$. Für $x = (0.529, 0.439, -0.723)^\mathsf{T}$ ist $R_A[x] \doteq -0.529326$ eine Eigenwert-Näherung der Genauigkeit $\varepsilon = 7.3_{10}-4$. Für den Eigenvektor ergibt sich die gröbere Schranke $\|x - x^{(2)}\| \leq 1.2_{10} - 2$.

Im Gegensatz zum symmetrischen Eigenwertproblem können bei *unsymmetrischen* (genauer: nicht-normalen) Matrizen die Eigenwerte und -Vektoren sehr empfindlich auf Störungen reagieren. Zunächst wird aber eine wichtige globale Aussage über die Lage der Eigenwerte behandelt.

Satz 3.6.5 (Gerschgorin) *Zur Matrix $A \in \mathbb{C}^{n\times n}$ sei definiert*

$$r_i := \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}|, \; i = 1, \ldots, n. \tag{3.6.4}$$

Dann liegen alle Eigenwerte von A in der Vereinigung der Kreisscheiben

$$K_{r_i}(a_{ii}) := \{z \in \mathbb{C} : \; |z - a_{ii}| \le r_i\}, \; i = 1, \ldots, n.$$

Beweis Es werde angenommen, ein Eigenwert λ liege außerhalb aller Kreisscheiben, sein Eigenvektor sei x mit $1 = \|x\|_\infty = |x_k|$. Dann folgt aber in der k-ten Gleichung aus

$$0 = |(a_{kk} - \lambda)x_k + \sum_{j\neq k} a_{kj}x_j| \ge |a_{kk} - \lambda||x_k| - \sum_{j\neq k} |a_{kj}|\|x\|_\infty = (|a_{kk} - \lambda| - r_k)\|x\|_\infty > 0$$

ein Widerspruch. ∎

Beispiel 3.6.6 Für die Matrix des letzten Beispiels ist $r_1 \doteq 4.33, r_2 = 5, r_3 \doteq 5.33$. Im Reellen gilt $K_1 \doteq [-3.33, 5.33], K_2 = [-3, 7], K_3 \doteq [-2.33, 8.33]$ und $K_1 \cup K_2 \cup K_3 = [-3.33, 8.33]$.

Bemerkung: 1) Wenn $\bigcup_{i=1}^n K_{r_i}(a_{ii})$ in mehrere, nicht-zusammenhängende Teile zerfällt, kann gezeigt werden, daß in jeder Komponente dieser Menge soviele Eigenwerte liegen, wie Kreise dazugehören.

2) Eine Verbesserung der Schranken (für einzelne Eigenwerte) ist eventuell möglich durch Anwendung des Satzes auf A^T oder

$$D^{-1}AD = \Big(\frac{d_j}{d_i}a_{ij}\Big)_{i,j}, \; D = \operatorname{diag}(d_i).$$

Besonders empfindlich können mehrfache Eigenwerte auf Störungen reagieren.

Beispiel 3.6.7 Der doppelte Eigenwert 1 der Matrix $A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ spaltet sich bei Addition der Matrix $F = \begin{pmatrix} 0 & 0 \\ \varepsilon & 0 \end{pmatrix}$ auf in die beiden Eigenwerte $\lambda_{1/2}(A + F) = 1 \pm \sqrt{\varepsilon}$. Die Eigenwerte hängen hier also nicht-differenzierbar von ε ab!

Zur Verallgemeinerung der obigen Störungssätze kann im unsymmetrischen Fall die Jordan- oder die Schur-Normalform herangezogen werden.

Satz 3.6.8 *Ist die Matrix $A = XJX^{-1}$ diagonalisierbar, so gilt für $j = 1, \ldots, n$ die Schranke*

$$\min_i |\lambda_j(A + F) - \lambda_i(A)| \le \|F\|\kappa(X).$$

Diese Aussage gilt in der $1, 2, \infty$-Norm mit der zugehörigen Konditionszahl $\kappa(X) = \|X\|\|X^{-1}\|$.

Beweis Es sei λ Eigenwert von $A+F$ und x ein zugehöriger Eigenvektor mit Norm eins, $\|x\|=1$. λ sei aber (oBdA) kein Eigenwert von A. Dann gilt

$$\begin{aligned} \lambda x - Ax = Fx \quad &\iff \quad x = (\lambda I - A)^{-1} F x \quad \Rightarrow \\ 1 \quad = \quad &\|x\| \le \|(\lambda I - A)^{-1} F\| \le \|(\lambda I - A)^{-1}\| \|F\| \\ \le \quad &\|(\lambda I - J)^{-1}\| \kappa(X) \|F\| = \frac{1}{\min_i |\lambda_i - \lambda|} \kappa(X) \|F\| \quad \blacksquare \end{aligned} \tag{3.6.5}$$

Der allgemeinste Fall wird erst durch den folgenden Satz abgedeckt, bei dem von der Schur-Normalform $A = USU^*$ ausgegangen wird. Dabei wird die Dreieckmatrix S zerlegt in die Diagonale $\Lambda = \operatorname{diag}(\lambda_j)$ und den Rest M, $S = \Lambda + M$. Die Norm $\|M\|_2$ ist ein Maß dafür, wieweit sich A von einer normalen Matrix unterscheidet.

Satz 3.6.9 *Es sei $U^*AU = S = \Lambda + M$ die Schur-Normalform der Matrix A und $p \le n$ so, daß $|M|^p = 0$ ist. Es sei*

$$\vartheta := \|F\|_2 \sum_{j=0}^{p-1} \|M\|_2^j .$$

Dann gilt für $j = 1, \ldots, n$ die Schranke

$$\min_i |\lambda_j(A+F) - \lambda_i(A)| \le \max\{\vartheta, \vartheta^{1/p}\}.$$

Beweis Hier wird in (3.6.5) eine andere Abschätzung der Inversen verwendet,

$$\|(\lambda I - A)^{-1}\|_2 = \|(\lambda I - \Lambda - M)^{-1}\|_2 \le \|(\lambda I - \Lambda)^{-1}\| \| \Big(I - (\lambda I - \Lambda)^{-1} M \Big)^{-1} \|_2 .$$

Mit $d := \|(\lambda I - \Lambda)^{-1}\|_2^{-1} = \min_i |\lambda_i - \lambda|$ und dem Satz von Neumann folgt

$$\| \Big(I - (\lambda I - \Lambda)^{-1} M \Big)^{-1} \|_2 \le \sum_{j=0}^{p-1} d^{-j} \|M\|_2^j ,$$

da mit $|M|^p = 0$ auch $[(\lambda I - \Lambda)^{-1} M]^p = 0$. Die Beziehung (3.6.5) führt daher hier auf

$$1 \le \|F\|_2 \frac{1}{d} \sum_{j=0}^{p-1} \frac{1}{d^j} \|M\|_2^j \le \|F\|_2 \sum_{j=1}^{p-1} \|M\|_2^j \max\{d^{-1}, d^{-p}\}.$$

Also folgt mit $\min\{d, d^p\} \le \vartheta \iff d \le \max\{\vartheta, \vartheta^{1/p}\}$ die Behauptung. $\blacksquare$

Beide Sätze zeigen, daß für große Werte von $\kappa(X)$ bzw. $\|M\|_2$ kleine Änderungen der Matrix A große Störungen in den Eigenwerten zur Folge haben können. Nach (3.6.1) ist bei allen numerischen Algorithmen mit einer Störmatrix mindestens in der Größenordnung der Maschinengenauigkeit $\mathcal{E}$ zu rechnen. Nach Satz 3.6.9 können daraus Eigenwertstörungen der Größe $\mathcal{E}^{1/p}$ erwachsen, die erheblich über der Maschinengenauigkeit liegen.

Allerdings können verschiedene Eigenwerte der gleichen Matrix sehr unterschiedlich auf Störungen reagieren. Für einen einfachen Eigenwert $\lambda(A)$ der Matrix A läßt sich zeigen, daß bei Störung durch eine Matrix εF, $\|F\|_2 = 1$, gilt

$$\lim_{\varepsilon \to 0} \left| \frac{\lambda(A + \varepsilon F) - \lambda(A)}{\varepsilon} \right| \le \frac{1}{|y^* x|}, \tag{3.6.6}$$

wobei x und y die Rechts- und Links-Eigenvektoren der Länge eins zum Eigenwert $\lambda(A)$ sind. Der Eigenwert λ ist daher dann gut (schlecht) konditioniert, wenn x und y einen kleinen (großen) Winkel miteinander bilden. Da bei normalen Matrizen gilt $x = y$, folgt auch aus diesem Ergebnis die Robustheit des entsprechenden Eigenwertproblems.

Beispiel 3.6.10 $A = \begin{pmatrix} 1 & 1 \\ \varepsilon & 1 \end{pmatrix}$, $\lambda_{1/2} = 1 \pm \sqrt{\varepsilon}$.

$$x^{(1/2)} = \frac{1}{\sqrt{1+\varepsilon}} \begin{pmatrix} 1 \\ \pm\sqrt{\varepsilon} \end{pmatrix}, \quad y^{(1/2)} = \frac{1}{\sqrt{1+\varepsilon}} \begin{pmatrix} \pm\sqrt{\varepsilon} \\ 1 \end{pmatrix}.$$

Die Eigenwerte sind sehr schlecht konditioniert, denn

$$\frac{1}{|y^* x|} = \frac{1+\varepsilon}{2\sqrt{\varepsilon}}.$$

Diese Aussage paßt zu der, daß die um ε veränderte Matrix $\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ den um $\sqrt{\varepsilon}$ gestörten doppelten Eigenwert 1 besitzt.

Vor diesem Hintergrund kann man im *unsymmetrischen* Fall nicht erwarten, daß Eigenwerte oder Eigenvektoren bis auf Maschinengenauigkeit exakt berechnet werden können. Man nennt sie aber *numerisch akzeptabel*, wenn sie *exakte* Eigenwerte/Eigenvektoren einer Matrix $A + F$ sind mit $\|F\| \cong \mathcal{E}\|A\|$. In diesem Sinn liefern die behandelten Transformationsverfahren (Hessenberg-, QR-, Jacobi-) wegen (3.6.1) numerisch akzeptable Approximationen.

3.7 Die Singulärwert-Zerlegung

Die Eigenwerttheorie führt bei reellen, symmetrischen Matrizen A auf die einfache Faktorisierung $A = UDU^\top$, wobei D eine (reelle) Diagonalmatrix ist (der Eigenwerte) und die Basismatrizen U und $U^\top$ orthogonal und daher zueinander invers sind. Für beliebige Matrizen A muß bei einer analogen Faktorisierung $A = UDV^\top$ mindestens eine der genannten Eigenschaften aufgegeben werden. Überblick (mit teilweise abweichenden Standard-Bezeichnungen):

Faktorisierung $A = UDV^\top$

Matrix A	Normalform		Eigenschaft der Faktoren		
			D diag.	U, V orth.	$V^\top = U^{-1}$
A symmetrisch		$A = UDU^\top$	•	•	•
A beliebig	Jordan–	$A = UDU^{-1}$	(•)	U regulär	•
	Schur–	$A = USU^\top$	S Dreieck	•	•
	Singulärwert–	$A = U\Sigma V^\top$	•	•	—

Die Eigenschaft $V^\top = U^{-1}$ in der letzten Spalte besagt, daß A und D ähnlich sind und ist für den Einsatz beim Eigenwertproblem unabdingbar. In anderen Aufgabenbereichen, wie zum Beispiel der Ausgleichsrechnung, ist dagegen die letzte Form, die Singulärwertzerlegung

$$A = U\Sigma V^\top, \quad U, V \text{ orthogonal}, \quad \Sigma \text{ reell diagonal (nicht-negativ)}$$

interessant. Insbesondere kann man hier beliebige (rechteckige) Matrizen $A \in \mathbb{R}^{m\times n}$ und $U \in \mathbb{R}^{m\times k}$, $V \in \mathbb{R}^{n\times k}$, $k = \min\{m, n\}$, betrachten. Zunächst werden grundlegende Eigenschaften und zwei typische Anwendungen der Singulärwertzerlegung besprochen, danach ihre Berechnung.

Eigenschaften und Anwendungen der Singulärwertzerlegung

Bevor die Existenz dieser Zerlegung gezeigt wird, sollen zwei einfache Zusammenhänge diskutiert werden, die Ansätze für den Existenzbeweis liefern. Für die Diagonalelemenete $\sigma_1, \ldots, \sigma_k$ von $\Sigma \in \mathbb{R}^{k\times k}$ gilt in Verbindung zur Matrixnorm:

$$\max_j \sigma_j = \|A\|_2 = \max_{x\neq 0} \frac{\|Ax\|_2}{\|x\|_2}, \tag{3.7.1}$$

Diese Beziehung folgt direkt aus der Orthogonalität von U, V, denn mit $x \in \mathbb{R}^n$ ist

$$\|U\Sigma V^\top x\|_2 = \|\Sigma(V^\top x)\|_2 \le \|\Sigma\|\|V^\top x\|_2 = \max_j \sigma_j \|x\|_2.$$

Die Beziehung (3.7.1) folgt daraus, da das Maximum tatsächlich angenommen wird.

Betrachtet man außerdem mit $A = U\Sigma V^\top$ die Produkte $AA^\top = U\Sigma^2 U^\top$ und $A^\top A = V\Sigma^2 V^\top$ wird der Zusammenhang mit dem Eigenwertproblem schnell klar: Die Matrix U ist die Orthonormalbasis der Eigenvektoren der symmetrischen, positiv semi-definiten Matrix $AA^\top$ und V die zu $A^\top A$. Da $AA^\top$ und $A^\top A$ die gleichen, nichtnegativen Eigenwerte besitzen, enthält Σ^2 gerade diese Eigenwerte. Ausführlich formuliert:

Satz 3.7.1 (und Definition) *Es sei $A \in \mathbb{R}^{m\times n}$ und $k := \min\{m, n\}$. Dann existiert die* Singulärwertzerlegung *von A, d.h., es gibt nichtnegative reelle Zahlen*

$$\sigma_1 \ge \sigma_2 \ge \ldots \sigma_k \ge 0$$

(die sog. singulären Werte*) und orthogonale Matrizen $U \in \mathbb{R}^{m\times k}$, $V \in \mathbb{R}^{n\times k}$ so, daß gilt*

$$A = U\Sigma V^\top = \sum_{j=1}^{k} \sigma_j u^{(j)} v^{(j)\top}, \quad \Sigma = \begin{pmatrix} \sigma_1 & & & 0 \\ & \sigma_2 & & \\ & & \ddots & \\ 0 & & & \sigma_k \end{pmatrix}. \tag{3.7.2}$$

mit den Spaltenvektoren $u^{(j)} = Ue^{(j)} \in \mathbb{R}^m$, $v^{(j)} = Ve^{(j)} \in \mathbb{R}^n$, $j = 1, \ldots, k$.

Beweis Da das Maximum in (3.7.1) angenommen wird, existieren Vektoren $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$ mit $\|x\|_2 = \|y\|_2 = 1$ so, daß $Ax = \sigma_1 y$ gilt. Beide Vektoren können jeweils durch $\tilde{U} \in \mathbb{R}^{m\times(k-1)}$ bzw. $\tilde{V} \in \mathbb{R}^{n\times(k-1)}$ zu orthogonalen Matrizen $U = (y, \tilde{U}) \in \mathbb{R}^{m\times k}$ bzw. $V = (x, \tilde{V}) \in \mathbb{R}^{n\times k}$ ergänzt werden. Wegen $\tilde{U}^\mathsf{T} y = 0$ folgt damit

$$U^\mathsf{T} A V = \begin{pmatrix} \sigma_1 & z^\mathsf{T} \\ 0 & B \end{pmatrix} =: A^{(1)}.$$

Da speziell für $v^\mathsf{T} := (\sigma_1, z^\mathsf{T})$ gilt

$$\|A^{(1)} v\|_2 \geq \sigma_1^2 + z^\mathsf{T} z = \|v\|_2^2,$$

ist natürlich $\|A\|_2 = \|A^{(1)}\|_2 \geq \|v\|_2 = \sqrt{\sigma_1^2 + z^\mathsf{T} z}$. Da aber schon $\sigma_1 = \|A\|_2$ war, folgt $z = 0$. Eine k-malige Anwendung dieses Arguments liefert $V^\mathsf{T} A U = \Sigma$. Durch Einschränkung von A auf $R(A)$ im Fall $m \geq n$ bzw. $N(A)^\perp$ im Fall $n > m$ folgt daraus (3.7.2). ∎

Beispiel: Zahlbeispiel

$$\begin{pmatrix} 4 & 2 & 1 \\ 0 & 0 & 2 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \frac{\sqrt{5}}{3} & \frac{2}{3} & 0 \\ \frac{2}{3\sqrt{5}} & -\frac{1}{3} & \frac{2}{\sqrt{5}} \\ \frac{4}{3\sqrt{5}} & -\frac{2}{3} & -\frac{1}{3\sqrt{5}} \end{pmatrix} \begin{pmatrix} 5 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{4}{3\sqrt{5}} & \frac{2}{3} & -\frac{1}{3\sqrt{5}} \\ \frac{2}{3\sqrt{5}} & \frac{1}{3} & \frac{2}{\sqrt{5}} \\ \frac{\sqrt{5}}{3} & -\frac{2}{3} & 0 \end{pmatrix}^\mathsf{T}.$$

Auch die kleinen Singulärwerte lieferen Informationen über A, für $m = n$ und

$$\sigma_n > 0 \text{ gilt } \sigma_n = 1/\|A^{-1}\|_2. \tag{3.7.3}$$

Im allgemeinen Fall mit $\sigma_r > 0 = \sigma_{r+1} = \ldots = \sigma_k$, folgt dagegen

$$\operatorname{Rang}(A) = r, \quad N(A) = \operatorname{span}(v^{(r+1)} \ldots, v^{(k)}), \quad R(A) = \operatorname{span}(u^{(1)}, \ldots, u^{(r)}).$$

Der Rang einer Matrix hängt nun allerdings nicht stetig von Störungen der Matrix A ab (die Menge der Matrizen von vollem Rang ist dicht). Die praktische Bedeutung der Singulärwertzerlegung beruht auch darauf, daß mit ihrer Hilfe die strenge, qualitative Rang-Aussage quantitativ gefaßt werden kann so, daß kleine Störungen unerheblich sind. Dazu sei mit $i \leq k = \min\{m, n\}$ eine Teilmatrix A_i definiert durch

$$A_i := \sum_{j=1}^{i} \sigma_j u^{(j)} {v^{(j)}}^\mathsf{T} =: U_i \Sigma_i V_i^\mathsf{T}. \tag{3.7.4}$$

Nach Definition besitzt A_i höchstens Rang i. Es ist

$$\|A - A_i\|_2 = \sigma_{i+1}, \tag{3.7.5}$$

da für $y \in \mathbb{C}^n$ gilt $\|(A - A_i)y\| = \sum_{j=i+1}^{k} \sigma_j^2 ({v^{(j)}}^\mathsf{T} y)^2 \leq \sigma_{i+1} \|y\|^2$. Der $i{+}1$-te singuläre Wert σ_{i+1} gibt also an, in welchem Abstand zur Gesamtmatrix A sich eine Matrix vom Rang $\leq i$ befindet, insbesondere kann für $\sigma_{i+1} > 0$ gezeigt werden, daß die Matrix A_i die zu A nächstgelegene Matrix vom Rang i ist,

$$\|A - A_i\|_2 = \min_{\mathrm{Rg}(B)=i} \|A - B\|_2.$$

Ist also A durch Meß- oder Rundungsfehler der Gesamtgröße $\|A - \hat{A}\| = \varepsilon$ aus einer (unbekannten) Matrix $\hat{A}$ hervorgegeangen, läßt sich aufgrund von (3.7.5) dann ein Rang r für $\hat{A}$ garantieren, wenn bei A der Wert $\sigma_r > \varepsilon$ ist. Umgekehrt kann man davon ausgehen, daß die Vektoren $u^{(j)}, v^{(j)}$ zu singulären Werten $\sigma_j \le \varepsilon$ keine erkennbare Informationen über die Ausgangsmatrix $\hat{A}$ enthalten, sondern nur „Rauschen". Man definiert daher einen gegen Störungen der Matrix unempfindlicheren *numerischen Rang* durch

$$\mathrm{Rg}_\varepsilon(A) := \sum_{\sigma_j > \varepsilon} 1 = (\text{Anzahl der singulären Werte } \sigma_j > \varepsilon). \tag{3.7.6}$$

Die Schwelle $\varepsilon \ge 0$ sollte bei einem konkreten Problem oberhalb der zu erwartenden (Meß-) Fehler gewählt werden. Wenn nur Rundungsfehler der relativen Größe $\mathcal{E}$ zu erwarten sind, ist die Wahl $\varepsilon = \mathcal{E}\|A\|_2$ sinnvoll.

Ganz allgemein gibt die Größe des Wertes σ_j an, wie stark die Vektoren $u^{(j)}, v^{(j)}$ Information über die Matrix A enthalten.

Kleinste-Quadrate-Lösung

Mit der Singulärwertzerlegung kann die in §1 besprochene Kleinste-Quadrate-Lösung x^+ direkt angegeben werden. Mit der Aufspaltung

$$\begin{aligned} \|Ax - y\|_2^2 &= \|\Sigma(V^\mathsf{T} x) - (U^\mathsf{T} y)\|_2^2 + \|(I - UU^\mathsf{T})y\|^2 \\ &= \sum_{j=1}^{r} |\sigma_j {v^{(j)}}^\mathsf{T} x - {u^{(j)}}^\mathsf{T} y|^2 + \sum_{j=r+1}^{k} |{u^{(j)}}^\mathsf{T} y|^2 + \|(I - UU^\mathsf{T})y\|^2 \end{aligned}$$

ergibt sich diese Lösung (für beliebige Dimensionen m und n) sofort. Die Komponenten $\xi_j = {v^{(j)}}^\mathsf{T} x$ mit $j \le r$ erhält man durch Minimierung von $\|Ax - y\|$, die restlichen $\xi_{r+1} = \ldots = \xi_k = 0$ durch anschließende Minimierung von $\|x\|$. Dies ergibt explizit die Lösung

$$x^+ = \sum_{j=1}^{r} \frac{{u^{(j)}}^\mathsf{T} y}{\sigma_j}\, v^{(j)} \tag{3.7.7}$$

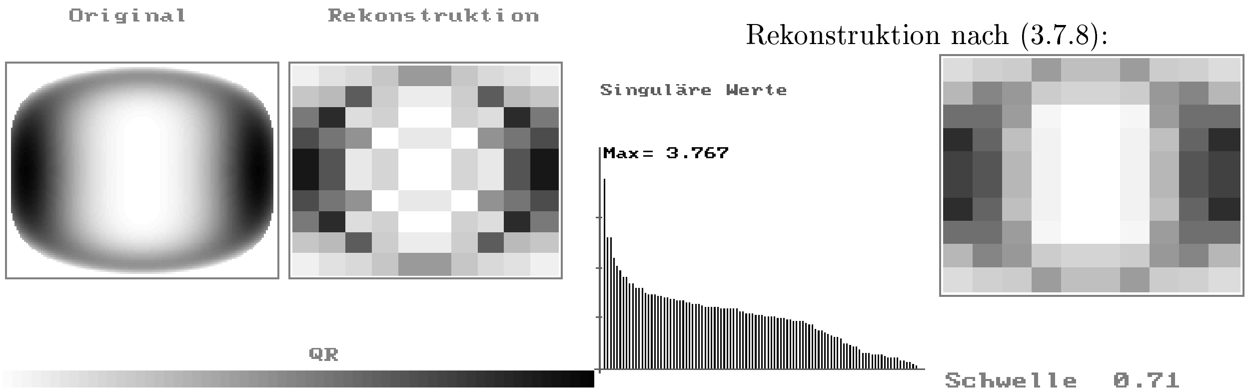
und entspricht der expliziten Darstellung der Pseudo-Inversen

$$A^+ = V_r \Sigma_r^{-1} U_r^\mathsf{T} = V \Sigma^+ U^\mathsf{T}, \quad \Sigma^+ = \mathrm{diag}(\frac{1}{\sigma_1}, \ldots, \frac{1}{\sigma_r}, 0, \ldots, 0).$$

Allerdings bietet dieser Zusammenhang wegen des hohen Rechenaufwands der SWZ noch keinen praktischen Vorteil gegenüber den in §1 behandelten Verfahren. Dies ändert sich allerdings, wenn man die Ergebnisse des vorhergehenden Abschnitts heranzieht. Denn (3.7.7) garantiert nicht in allen Fällen eine sinnvolle Lösung des Problems $Ax = y$, vor allem, wenn A beinahe einen Rangdefekt besitzt, d.h., wenn $\mathrm{Rg}_\varepsilon(A) < k$ mit $\varepsilon = \mathcal{E}\|A\|$ gilt. Nach der obigen Erläuterung enthalten Komponenten zu kleinen singulären Werten $0 < \sigma_i \ll 1$ i.a. wenig relevante Information zum Problem. Wegen der Division durch σ_i in (3.7.7) wird aber gerade dieser Anteil in x^+ stark hervorgehoben. Hier ist es sinnvoller, in der Kleinste-Quadrate-Lösung (3.7.7) nur soviele Komponenten zu berücksichtigen, wie es dem numerischen Rang entspricht, etwa in der Form

$$x_\varepsilon^+ := \sum_{j=1}^{r_\varepsilon} \frac{u^{(j)\mathsf{T}} y}{\sigma_j} v^{(j)}, \quad r_\varepsilon := \mathrm{Rg}_\varepsilon(A). \tag{3.7.8}$$

Beispiel: Anwendung aus Beispiel 1.3 (Computer-Tomographie) mit $n = 10 * 10$ Quadraten und $m = 10 * 14$ Messungen. Neben der Originaldichteverteilung links zeigt die Rekonstruktion mit x^+ aus §1 (oszillierende) Störungen. Das mittlere Diagramm zeigt die Singulärwerte nach abfallender Größe, das rechte die Rekonstruktion x_ε^+ mit der Schwelle $\varepsilon = 0.7$.



Komponentenanalyse

In Anwendungen, bei denen es darauf ankommt aus großen Datenmengen die wesentlichen Zusammenhänge („Komponenten") herauszufiltern, kann die Singulärwertzerlegung eingesetzt werden. Einer Menge von Objekten (Text-Dateien, Bilder,..) läßt sich dadurch eine Matrix zuordnen, daß man bei dem Objekt Nummer j verschiedene quantitative Angaben (Wort-Vorkommen in Datei, Pixel-Grauwerte im Bild, ..) zu einem Vektor zusammenfaßt und diesen als die j-te Spalte der Matrix schreibt.

Die Bedeutung der in den singulären Vektoren enthaltenen Information über die Matrix A wird bei den kleineren singulären Werten immer geringer. Daher stellt die Folge

$$A_1, A_2, A_3, \ldots$$

der in (3.7.4) definierten Abschnittsmatrizen eine Folge von Modellen für A mit wachsender Komplexität (Rang) und monoton fallendem Fehler $\|A - A_i\|$ dar. Die Auswahl des im konkreten Fall geeigneten Modellrangs hängt vom Anwendungsproblem ab. Mit einem solchen Modell $A_i = U_i \Sigma_i V_i^\mathsf{T}$, $i \ll k$, können dann folgende Aufgaben behandelt werden:

1. Datenkompression: approximiere Spalte $Ae^{(j)} \in \mathbb{R}^m$ (d.h. Objekt j) durch die wenigen Komponenten in der U-Basis, $s^{(j)} := U_i^{\mathsf{T}} A e^{(j)} = \Sigma_i V_i^{\mathsf{T}} e^{(j)} \in \mathbb{R}^i$.

2. Identifikation: Finde zu einem neuem Objekt mit den Merkmalen $y \in \mathbb{R}^m$ das(die) ähnlichste(n) Objekt(e) der Datenbasis, d.h., den(die) ähnlichsten Spaltenvektor(en) von A,

$$\min_j \|U_i^{\mathsf{T}} y - \Sigma_i V_i^{\mathsf{T}} e^{(j)}\|.$$

   Diese Suche ist also nur im Raum $\mathbb{R}^i$ durchzuführen.

3. Weitere Anwendungen im Dokumenten-Bereich (vgl. Berry/Dumais/O'Brien: SIAM Review, 37/1995), z.B., Informationsfilter, Identifikation ähnlicher Begriffe, etc.

Berechnung der Singulärwertzerlegung

Die Berechnung der singulären Werte der Matrix erfolgt durch eine Iteration, die zur QR-Iteration bei der symmetrischen Matrix $A^{\mathsf{T}} A$ äquivalent ist. Wie dort kann Arbeit durch Reduktion auf einfachere Gestalt eingespart werden. Da jetzt bei A unabhängig voneinander orthogonale Umformungen von links und rechts zulässig sind, kann A durch Spiegelungen sogar auf Bidiagonalgestalt transformiert werden. Für sehr unterschiedliche Dimensionen m und n ist es aber effizienter, zunächst eine QR-Zerlegung von A (im Fall $m \gg n$) oder A^{T} (Fall $m \ll n$) durchzuführen. Da sich dann auch die Bezeichnungen vereinfachen, wird nur noch der Fall $m \geq n$ betrachtet. Aus einer Singulärwertzerlegung $R = \tilde{U} \Sigma V^{\mathsf{T}}$ bei $A = QR$ (mit $Q \in \mathbb{R}^{m \times n}, R \in \mathbb{R}^{n \times n}$) ergibt sich die von A mit $U = Q\tilde{U}$ nach $A = QR = (Q\tilde{U}) \Sigma V^{\mathsf{T}}$.

Es sei also nun oBdA $R \in \mathbb{R}^{n \times n}$ obere Dreieckmatrix, in der ersten Spalte sind unter der Hauptdiagonale also schon Nullen. Da die Multiplikation von rechts mit einer Spiegelung $K^{(2)} = I - 2w^{(2)} w^{(2)\mathsf{T}}$ die erste Spalte von $R =: R^{(2)}$ unverändert lassen soll, ist $w_1^{(2)} = 0$ zu wählen. Bei $\tilde{R}^{(2)} = RK^{(2)}$ treten dann ab der zweiten Spalte nichttriviale Elemente auf. Im j-ten Schritt ($j \geq 2$) besitze $R^{(j)}$ daher die Gestalt

$$R^{(j)} = \left(\begin{array}{ccc|ccc} * & * & 0 & & & \\ & \ddots & \ddots & & 0 & \\ & & * & * & \cdots & * \\ \hline & & & * & \cdots & * \\ & & & \vdots & & \vdots \\ & & & * & \cdots & * \end{array}\right) \begin{array}{l} \left.\vphantom{\begin{array}{c} * \\ \ddots \\ * \end{array}}\right\} j-1 \\ \left.\vphantom{\begin{array}{c} * \\ \vdots \\ * \end{array}}\right\} n-j+1 \end{array} \tag{3.7.9}$$

Nun werden nach Lemma 1.3.2 mit $R^{(j)}$ die Spiegelung $K^{(j)}$ und mit $\tilde{R}^{(j)} = R^{(j)} K^{(j)}$ die

Spiegelung $H^{(j)}$ so bestimmt, daß gilt

$$\begin{aligned}
(\dots, r^{(j)}_{j-1,j-1}, \dots, r^{(j)}_{j-1,n}) K^{(j)} &= (\dots, r^{(j)}_{j-1,j-1}, *, 0, \dots), \\
\text{und} \quad H^{(j)} \begin{pmatrix} \vdots \\ \tilde{r}^{(j)}_{j-1,j} \\ \tilde{r}^{(j)}_{jj} \\ \vdots \\ \tilde{r}^{(j)}_{nj} \end{pmatrix} &= \begin{pmatrix} \vdots \\ \tilde{r}^{(j)}_{j-1,j} \\ * \\ 0 \\ \vdots \end{pmatrix}.
\end{aligned} \tag{3.7.10}$$

Dann hat $R^{(j+1)} = H^{(j)} \tilde{R}^{(j)} = H^{(j)} R^{(j)} K^{(j)}$ wieder die Gestalt (3.7.9). Das Verfahren führt in $n-2$ Schritten zum Ziel, d.h.,

$$R^{(n-1)} = \underbrace{H^{(n-1)} \cdots H^{(2)}}_{=: \hat{H}^{\mathsf{T}}} R \underbrace{K^{(2)} \cdots K^{(n-1)}}_{=: \hat{K}} = B = \begin{pmatrix} d_1 & b_1 & & \\ & d_2 & b_2 & \\ & & \ddots & \ddots \\ & & & d_n \end{pmatrix}$$

ist obere Bidiagonalmatrix. Der Rechenaufwand für diese Bidiagonalisierung beträgt i.w. das doppelte der QR-Zerlegung einer $n \times n$-Matrix, nämlich $\frac{8}{3} n^3$ Operationen. Zusammen mit der QR-Zerlegung von A ergibt dies

$$2(m+n)n^2 \text{ Operationen für } B \text{ (und auch für } \Sigma).$$

Die Berechnung der Basismatrizen erfordert einen Zusatzaufwand von $2mn^2 + 11n^3$ für U und $9n^3$ für V. Wie beim symmetrischen Eigenwertproblem ist im Vergleich dazu der Aufwand zur abschließenden Berechnung der Singulärwertzerlegung von B nur gering.

Berechnung der Singulärwertzerlegung einer Bidiagonalmatrix

Dem folgenden Verfahren liegt die QR-Iteration bei der Tridiagonalmatrix

$$T := B^{\mathsf{T}} B = \begin{pmatrix} d_1^2 & d_1 b_1 & & & \\ d_1 b_1 & d_2^2 + b_1^2 & d_2 b_2 & & \\ & \ddots & \ddots & \ddots & \\ & & d_{n-2} b_{n-2} & d_{n-1}^2 + b_{n-2}^2 & d_{n-1} b_{n-1} \\ & & & d_{n-1} b_{n-1} & d_n^2 + b_{n-1}^2 \end{pmatrix} \tag{3.7.11}$$

zugrunde. Da die explizite Berechnung von $B^{\mathsf{T}} B$ numerisch ungünstig ist (vgl. Einleitung §1.3), vermeidet man diese und führt den QR-Schritt auf folgende Weise implizit aus. Für $b_1 \neq 0$ ist die erste Operation des QR-Schritts bei (3.7.11) die Anwendung einer 2×2-Drehung oder Spiegelung H_1^{T}, die das Element $t_{21} = d_1 b_1$ eliminiert und dann die entsprechende Rechtsmultiplikation:

$$T = B^{\mathsf{T}} B \quad \mapsto \quad H_1^{\mathsf{T}} B^{\mathsf{T}} B H_1 = (BH_1)^{\mathsf{T}} (BH_1).$$

Nach Anwendung von $n-2$ weiteren solchen unitären 2×2-Transformationen ist die Matrix wieder in Tridiagonalgestalt. Man kann nun das gleiche Ergebnis dadurch erreichen, daß man nur den Faktor BH_1 behandelt und durch (andere) unitäre Umformungen wieder auf Bidiagonalgestalt bringt. Da

$$B^{(1)} := BH_1 = \begin{pmatrix} * & * & & & \\ \bullet & * & * & & \\ & & * & * & \\ & & & & \ddots \end{pmatrix}$$

gilt, hat $B^{(1)}$ in der ersten Spalte eine „Beule" bekommen. Diese Beule läßt sich nun durch weitere 2×2-Spiegelungen F_j^T von links und G_{j+1} von rechts zum unteren Ende der Matrix treiben. Im j-ten Schritt ändert F_j^T nur die Zeilen j und $j+1$, G_{j+1} die Spalten $j+1$ und $j+2$, daher gilt

$$\begin{array}{ccccc} B^{(j)} & \mapsto & F_j^\mathsf{T} B^{(j)} & \mapsto & F_j^\mathsf{T} B^{(j)} G_{j+1} =: B^{(j+1)} \\ \begin{pmatrix} \ddots & \ddots & & & \\ & * & * & & \\ & \bullet & * & * & \\ & & & * & * \\ & & & & \ddots \quad \ddots \end{pmatrix} & \mapsto & \begin{pmatrix} \ddots & \ddots & & & \\ & * & * & \bullet & \\ & & * & * & \\ & & & * & * \\ & & & & \ddots \quad \ddots \end{pmatrix} & \mapsto & \begin{pmatrix} \ddots & \ddots & & & \\ & * & * & & \\ & & * & * & \\ & & \bullet & * & * \\ & & & & \ddots \quad \ddots \end{pmatrix} \end{array}$$

Dann ist

$$\hat{B} := (F_1 \cdots F_{n-1})^\mathsf{T} B H_1 G_2 \cdots G_{n-1} \tag{3.7.12}$$

wieder in Bidiagonalgestalt und $\hat{B}^\mathsf{T}\hat{B} =: \hat{T}$ daher tridiagonal. Mit $Q := H_1 G_2 \cdots G_{n-1}$ gilt insbesondere $\hat{T} = Q^\mathsf{T} B^\mathsf{T} B Q = Q^\mathsf{T} T Q$. Die erste Spalte von Q ist dabei $Qe^{(1)} = H_1 G_2 \cdots G_{n-1} e^{(1)} = H_1 e^{(1)}$, also nur durch H_1 bestimmt, da die Matrizen G_j Änderungen erst ab Komponente j verursachen. Man kann nun zeigen (sog. 'Implizites Q-Theorem'), daß schon aufgrund dieser Eigenschaften Q und $\hat{T}$ (bis auf Vorzeichen der Nebendiagonalen) mit den Matrizen aus dem QR-Schritt $QR = T$, $\hat{T} = RQ$, übereinstimmen.

Bei der praktischen Durchführung ist die Iteration auf den Bereich der Matrix B zwischen dem ersten und letzten nichtverschwindenden Nebendiagonalelement b einzuschränken. Außerdem kann nach (3.4.5) eine Spektralverschiebung mit demjenigen Eigenwert λ von

$$\begin{pmatrix} d_{n-1}^2 + b_{n-2}^2 & d_{n-1} b_{n-1} \\ d_{n-1} b_{n-1} & d_n^2 + b_{n-1}^2 \end{pmatrix},$$

gemacht werden, der näher an $t_{n,n} = d_n^2 + b_{n-1}^2$ liegt. Der einzige Unterschied zum bisherigen Vorgehen ist dabei, daß H_1 jetzt aus $(d_1^2 - \lambda, d_1 b_1, 0, \ldots)^\mathsf{T}$ abgeleitet wird.

Da mit dieser Strategie die globale Konvergenz des QR-Verfahrens garantiert ist, vgl. Satz 3.4.6, liefert das Verfahren eine Singulärwertzerlegung von B. Insgesamt kann somit die Singulärwertzerlegung einer allgemeinen Matrix A in drei Schritten erzeugt werden:

$$\begin{array}{ccccccc} A & \overset{\text{QR-Zerl.}}{\longrightarrow} & R = Q^\mathsf{T} A & \overset{\text{Bidiag.}}{\longrightarrow} & B = \hat{H}^\mathsf{T} R \hat{K} & \overset{\text{Iter. (3.7.12)}}{\longrightarrow} & \Sigma = \hat{U}^\mathsf{T} B \hat{V} \\ A = (Q\hat{H}\hat{U})\Sigma(\hat{K}V)^\mathsf{T} & \leftarrow & R = \hat{H}\hat{U}\Sigma(\hat{K}V)^\mathsf{T} & & \leftarrow & & B = \hat{U}\Sigma\hat{V}^\mathsf{T} \end{array}$$

4 Iterationsverfahren für große Matrizen

Die stetige Entwicklung bei Computern zu immer höheren Rechenleistungen und Speicherkapazitäten hat dazu geführt, daß immer feinere und komplexere Modelle realer Prozesse aufgestellt und numerisch gelöst werden können. Gerade dann machen sich aber auch Effizienzunterschiede verschiedener Verfahren (unterschiedliche Komplexitätsordnungen) immer stärker bemerkbar. Außerdem spielen Struktureigenschaften ein stärkere Rolle. Bei vielen Modellen treten lineare Gleichungssysteme oder Eigenwertprobleme mit sehr großen Matrizen als Unterprobleme auf. Wegen der meist lokal beschränkten Kopplung zwischen den Unbekannten (z.B. nur lokale Nachbarschaften) sind diese Matrizen oft dünn besetzt.

Bei einer Reihe von Iterationsverfahren kann diese Eigenschaft einfach ausgenutzt werden, etwa dann, wenn die Matrix A nicht vollständig explizit bekannt sein muß, sondern vor allem über Matrix-Vektor-Produkte Ax eingeht. Dies war in den Verfahren der Numerik I, z.B., bei Gesamtschritt-, Einzelschritt- und SOR-Verfahren schon der Fall. Bei diesen Iterationsverfahren wird unglücklicherweise für wichtige Problemklassen die Konvergenz bei wachsender Problemgröße n immer schlechter (d.h. $\varrho(B_n) \to 1,\ n \to \infty$, vgl. Numerik-I, §2.5, Demo-Beispiel). Das früher ebenfalls behandelte Relaxationsverfahren läßt sich aber auf einfache Weise verbessern.

4.1 Polynomielle Beschleunigungsverfahren

Ohne Einschränkung wird die Fixpunktgleichung

$$x = Bx + r \quad \iff \quad Ax = r \ \text{ mit } A = I - B, \tag{4.1.1}$$

$x, r \in \mathbb{R}^n, A, B \in \mathbb{R}^{n\times n}$ betrachtet, die Lösung sei z. In der Numerik I wurde zu deren Lösung das *Relaxationsverfahren*

$$x^{(k+1)} = (1-\omega)x^{(k)} + \omega(Bx^{(k)} + r) = x^{(k)} + \omega(r - Ax^{(k)}),\ k = 0, 1, \dots, \tag{4.1.2}$$

betrachtet mit dem Parameter $\omega \in \mathbb{R}$. Die Iterationsmatrix des Verfahrens war

$$B(\omega) = (1-\omega)I + \omega B = I - \omega A$$

und besaß bei reellen Eigenwerten $0 < \lambda_n(A) \le \ldots \le \lambda_1(A)$ einen minimalen Spektralradius

$$\varrho(B(\hat\omega)) = \frac{\lambda_1(A) - \lambda_n(A)}{\lambda_1(A) + \lambda_n(A)} \quad \text{in} \quad \hat\omega = \frac{2}{\lambda_1(A) + \lambda_n(A)}. \tag{4.1.3}$$

Der (asymptotische) Konvergenzfaktor $\varrho(B(\hat\omega))$ hängt alleine von dem Verhältnis λ_1/λ_n der Eigenwerte von A ab. In Anlehnung an die Numerik I wird daher die *Spektral-Konditionszahl* eine regulären Matrix definiert durch

$$\hat\kappa(A) := \varrho(A)\varrho(A^{-1}) = \frac{\max\{|\lambda_j(A)| :\ j = 1, \dots, n\}}{\min\{|\lambda_j(A)| :\ j = 1, \dots, n\}}. \tag{4.1.4}$$

Offensichtlich gilt $1 \leq \hat{\kappa}(A) \leq \kappa_p(A),\ p \geq 1$. Für symmetrische Matrizen, $A = A^\mathsf{T}$, ist sogar $\hat{\kappa}(A) = \kappa_2(A)$. Der Konvergenzfaktor aus (4.1.3) läßt sich also in der Form

$$\min_{\omega \in \mathbb{R}} \varrho(B(\omega)) = \frac{\hat{\kappa}(A) - 1}{\hat{\kappa}(A) + 1} = 1 - \frac{2}{\hat{\kappa}(A) + 1} \tag{4.1.5}$$

schreiben. Die Optimalitätsaussage gilt genau genommen nur bei isolierter Betrachtung eines einzelnen Schrittes (4.1.2). Faßt man dagegen mehrere Schritte mit *wechselnden* Parametern ω_k zusammen, ergibt sich für den Fehler $x^{(m)} - z$ nach m Schritten

$$\begin{aligned} x^{(m)} - z &= x^{(m-1)} - z + \omega_m(Az - Ax^{(m-1)}) = (I - \omega_m A)(x^{(m-1)} - z) \\ &= (I - \omega_m A)(I - \omega_{m-1}A) \cdots (I - \omega_1 A)(x^{(0)} - z). \end{aligned} \tag{4.1.6}$$

Der Vorfaktor des Startfehlers $x^{(0)} - z$ besteht also aus einem *Polynom* der Matrix A. Dessen Parameter ω_k sollen im folgenden so bestimmt werden, daß dieses Matrixpolynom "möglichst klein" wird. Für den Fehler (4.1.6) erhält man

Satz 4.1.1 *Bei einer Matrix A mit reellen Eigenwerten*

$$0 < a \leq \lambda_n \leq \ldots \leq \lambda_1 \leq b$$

gilt für den Fehler bei der Richardson-Iteration

$$x^{(k)} := x^{(k-1)} + \omega_k(r - Ax^{(k-1)}),\ \ k = 1, \ldots, m, \tag{4.1.7}$$

mit Parametern ω_k nach m Schritten die Beziehung

$$x^{(m)} - z = p_m(A)\,(x^{(0)} - z). \tag{4.1.8}$$

Dabei ist $p_m \in \Pi_m$, $p_m(0) = 1$, $p_m(t) = (1 - \omega_1 t) \cdots (1 - \omega_m t)$ und

$$\varrho(p_m(A)) \leq \max\{|p_m(t)| :\ t \in [a, b]\}. \tag{4.1.9}$$

Beweis Die Identität (4.1.8) entspricht (4.1.6), die Reihenfolge der Faktoren ist beliebig. Für jeden Eigenvektor x von A mit Eigenwert λ gilt

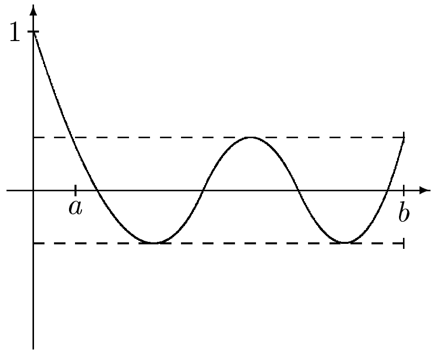
$$p_m(A)x = (I - \omega_m A) \ldots (I - \omega_2 A)(1 - \omega_1 \lambda)x = \ldots = p_m(\lambda)x.$$

Aus der Voraussetzung $\lambda_j \in [a, b]$ folgt die Schranke (4.1.9). ■

Statt für einen einzelnen Schritt kann der Konvergenzfaktor jetzt für einen festen *Zyklus* von m Schritten minimiert werden. Da die Eigenwerte der Matrix i.a. nicht bekannt sind, Eigenwert-Schranken aber mit akzeptablem Aufwand bestimmt werden können (vgl., z.B., Satz 3.6.5), minimiert man die Schranke (4.1.9),

$$\max\{|p_m(t)| :\ t \in [a, b]\} \stackrel{!}{=} \min,\ \ p_m \in \Pi_m,$$

unter der Nebenbedingung $p_m(0) = 1$. Die Lösung dieses Problems ist von der Suche optimaler Interpolationsknoten aus der Numerik I bekannt. Es geht auch hier darum, einen möglichst kleinen Funktionswert von p_m im Intervall $[a,b]$ zu erreichen, nur die Normierungsbedingung unterscheidet sich hier vom genannten Problem.



Satz 4.1.2 *Unter allen Polynomen $p_m \in \Pi_m$ mit $p_m(0) = 1$ wird*

$$\|p\|_{[a,b]} := \max\{|p(t)| : \ t \in [a,b]\}$$

minimal bei

$$\hat{p}_m(t) := T_m\Big(\frac{b+a-2t}{b-a}\Big)\Big/T_m\Big(\frac{b+a}{b-a}\Big), \tag{4.1.10}$$

wobei $T_m(s)$ das Tschebyscheff-Polynom vom Grad m ist. Es wird durch die Rekursion $T_0 = 1$, $T_1(s) := s$,

$$T_{k+1}(s) := 2s\,T_k(s) - T_{k-1}(s), \ k = 1,2,\ldots \tag{4.1.11}$$

definiert und besitzt die explizite Darstellung

$$T_m(s) = \begin{cases} \cos(m \arccos s), & |s| \le 1 \\ \cosh(m \operatorname{arcosh} s), & s > 1. \end{cases}$$

Die Parameter $\hat{\omega}_k$ von $\hat{p}_m(t) = \prod_{k=1}^m (1 - \hat{\omega}_k t)$ sind daher

$$\frac{1}{\hat{\omega}_k} = \frac{a+b}{2} - \frac{b-a}{2} \cos \frac{2k-1}{2m}\pi, \ k = 1,\ldots,m. \tag{4.1.12}$$

Der Minimalwert der Maximumnorm ist

$$\|\hat{p}_m\|_{[a,b]} = 1\Big/T_m\Big(\frac{b+a}{b-a}\Big) \le 2\left(\frac{\sqrt{b/a}-1}{\sqrt{b/a}+1}\right)^m. \tag{4.1.13}$$

Beweis Nach Satz 3.6.1 aus der Numerik I minimiert $\hat{q}_m(t) := T_m\Big(\frac{b+a-2t}{b-a}\Big)$ die Norm $\|q_m\|_{[a,b]}$ unter allen Polynomen $q_m(t) = c_m t^m + \ldots$ mit festem höchsten Koeffizienten $c_m = -2^{m+1}/(b-a)^m$. Die neue Nebenbedingung der Form $p_m(0) = 1$ wird durch die Wahl $\hat{p}_m = \hat{q}_m(t)/\hat{q}_m(0)$ erfüllt. Die Kehrwerte $1/\hat{\omega}_k$ entsprechen daher den Nullstellen von $\hat{p}_m$. Die erste Schranke (4.1.13) gilt nun wegen $\|\hat{q}_m\|_{[a,b]} = 1$.

Zur Normierung wird in (4.1.10) ein Funktionswert $T_m(s)$ mit $s > 1$ verwendet. Um auch für diesen Fall eine explizite Darstellung der Polynome aus (4.1.11) zu erhalten, wird $s = \cosh\sigma$ gesetzt. Wegen

$$\cosh(k \pm 1)\sigma = \cosh(k\sigma)\ \cosh(\sigma) \pm \sinh(k\sigma)\ \sinh(\sigma),$$

sowie $\cosh 0 = 1$ gilt daher hier die explizite Beziehung $T_m(s) = \cosh(m \operatorname{arcosh} s)$. Die letzte Ungleichung in (4.1.13) folgt aus einer weiteren Darstellung der Tschebyscheff-Polynome mit Argument $q = (b+a)/(b-a) = (\kappa+1)/(\kappa-1)$, $\kappa = b/a$, nämlich

$$T_m\Big(\frac{\kappa+1}{\kappa-1}\Big) = \frac{1}{2}\Big[\Big(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\Big)^m + \Big(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\Big)^m\Big] \geq \frac{1}{2}\Big(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\Big)^m. \qquad \blacksquare$$

Der Konvergenz-Vorteil bei Wahl der Relaxationsparameter nach Satz 4.1.2 für einen Zyklus der Länge $m > 1$ gegenüber dem einfachen, stationären Relaxationsverfahren (4.1.2) der Zykluslänge 1 soll jetzt bei $[a, b] = [\lambda_n, \lambda_1]$ analysiert werden. Dann hängt (4.1.13) wieder nur von der Konditionszahl $\hat{\kappa}(A)$ ab und für große Werte derselben, $\hat{\kappa}(A) \gg 1$, ist

$$\|p_m\|_{[a,b]} \leq 2\left(\frac{\sqrt{\hat{\kappa}}-1}{\sqrt{\hat{\kappa}}+1}\right)^m \doteq 2\left(1 - \frac{2}{\sqrt{\hat{\kappa}}}\right)^m.$$

Im Vergleich dazu liefern k Schritte mit festem Parameter $\hat{\omega}$ nach (4.1.5) den Fehlerfaktor

$$\varrho(B(\hat{\omega}))^k \doteq \Big(1 - \frac{2}{\hat{\kappa}}\Big)^k.$$

Prüft man im wichtigen Fall $\hat{\kappa} \gg 1$ für welche Schrittzahlen eine Fehlerreduktion um $1/2$ erreicht wird, kommt man auf

$$m \doteq \sqrt{\hat{\kappa}} \ln 2, \qquad \text{bzw} \qquad k \doteq \hat{\kappa}\frac{\ln 2}{2},$$

d.h., $k \cong m^2$. Die Konvergenz der Richardson-Iteration (4.1.7) mit optimalen Parametern (4.1.12) ist also wesentlich schneller, ein m-Zyklus bei ihr entspricht ungefähr m^2 Schritten der einfachen Relaxation!

Optimale Parameter für die Richardson-Iteration können auch noch bei komplexen Eigenwerten angegeben werden, wenn man Ellipsen in der komplexen Ebene zur Einschließung verwendet, die den Nullpunkt nicht enthalten. In noch allgemeineren Situationen, z.B., bei reellen Eigenwerten mit beiden Vorzeichen, verwendet man oft ersatzweise Polynome, die ein quadratisches Fehlerkriterium minimieren, wie

$$\int_S |p(z)|^2 g(z)dz, \quad S \supset \lambda_j \forall j = 1, \ldots, n.$$

Der Grund ist der, daß solche Probleme einerseits explizit gelöst werden können und bei geeigneter Gewichtsfunktion g auch annehmbare Polynome erwarten lassen, da die Standard-Tschebyscheff-Polynome T_k sowohl Minimalpolynome in der Supremumsnorm $\sup\{|T_k(x)| : x \in [-1, 1]\}$ (vgl. Num-I, §3.6) als auch der gewichteten L_2-Norm $\int_{-1}^{1} T_k(x)^2(1-x^2)^{-1/2}dx$ sind (vgl. Num-I, §5.3).

Ein wesentlicher *Nachteil* des Richardson-Verfahrens ist der, daß die Konvergenzaussage (4.1.9), (4.1.13) erst nach dem Durchlaufen eines vollen m-Zyklus zutrifft, bei periodischer Wahl $\omega_{k+jm} := \omega_k$ also nur für $x^{(m)}, x^{(2m)}, \ldots$. Der Fehler der restlichen Iterierten kann sogar erheblich über dem Startfehler liegen. Daher muß man sich vor Beginn für eine Mindestzahl m von Iterationen entscheiden.

Dies läßt sich im Fall des positiven reellen Spektrums (und einigen anderen Fällen, aber nicht im indefiniten Fall) auf folgende Weise umgehen. Mit den skalierten Tschebyscheff-Polynomen war der Fehler (4.1.8)

$$x^{(m)} - z = \hat{p}_m(A)(x^{(0)} - z).$$

Für *Orthogonal-Polynome* gilt nun eine zweistufige *Rekursionsformel* (vgl. Numerik I, §3.4)

$$\hat{p}_{m+1}(t) = \alpha_m(t - \gamma_m)\hat{p}_m(t) + \beta_m \hat{p}_{m-1}(t), \; m = 1, 2, \ldots . \tag{4.1.14}$$

Betrachtet man für beliebiges k die neuen Vektoren

$$y^{(k)} := z + \hat{p}_k(A)(x^{(0)} - z) = \hat{p}_k(A)x^{(0)} + (I - \hat{p}_k(A))z, \; k = 0, 1, \ldots,$$

die nur für $k = 0$ bzw. $k = m$ mit denen der Richardson-Iteration übereinstimmen, so folgt aus (4.1.14)

$$\begin{aligned} y^{(k+1)} &= \hat{p}_{k+1}(A)(x^{(0)} - z) + z \\ &= \alpha_k(A - \gamma_k I)\hat{p}_k(A)(x^{(0)} - z) + \beta_k \hat{p}_{k-1}(A)(x^{(0)} - z) + z \\ &= \alpha_k(A - \gamma_k I)(y^{(k)} - z) + \beta_k(y^{(k-1)} - z) + z \\ &= \alpha_k(Ay^{(k)} - r) - \alpha_k\gamma_k y^{(k)} + \beta_k y^{(k-1)} + \underbrace{(1 - \beta_k + \alpha_k\gamma_k)}_{=0} z. \end{aligned}$$

Der Vorfaktor bei z verschwindet dabei, da $\hat{p}_k(0) = 1 \,\forall k$, vgl. (4.1.8). Dies entspricht aber genau der Beziehung $1 = \beta_k - \alpha_k\gamma_k$ in (4.1.14). Der Vektor $y^{(k+1)}$, der nun in jedem Schritt $k+1$ einen minimalen Fehlerfaktor $\varrho(p_{k+1}(A))$ besitzt, kann aus den Vektoren $y^{(k)}$ und $y^{(k-1)}$ berechnet werden. Für die Koeffizienten der Rekursion (4.1.14) der Polynome (4.1.10) ergeben sich mit

$$q := \frac{b+a}{b-a}, \quad d := \frac{2}{b+a}, \quad \varrho_k := 2q\frac{T_k(q)}{T_{k+1}(q)} \tag{4.1.15}$$

die Beziehungen $\alpha_k = -\varrho_k d,\ \alpha_k\gamma_k = -\varrho_k,\ \beta_k = 1 - \varrho_k$. Denn es ist $\hat{p}_k(t) = T_k(q(1-dt))/T_k(q)$ und aus der Rekursion (4.1.11) folgt

$$\begin{aligned} \hat{p}_{k+1}(t) &= \frac{T_{k+1}(q(1-dt))}{T_{k+1}(q)} = 2q(1-dt)\frac{T_k(q(1-dt))}{T_{k+1}(q)} - \frac{T_{k-1}(q(1-dt))}{T_{k+1}(q)} \\ &= 2q(1-dt)\frac{T_k(q)}{T_{k+1}(q)}\,\hat{p}_k(t) - \frac{T_{k-1}(q)}{T_{k+1}(q)}\,\hat{p}_{k-1}(t) \\ &= \varrho_k(1-dt)\hat{p}_k(t) + (1-\varrho_k)\hat{p}_{k-1}(t). \end{aligned}$$

Beim letzten Koeffizienten $1 - \varrho_k$ wurde (4.1.11) noch einmal verwendet. Mit diesen Bezeichnungen wird nun die *Tschebyscheff-Iteration* formuliert. Wegen der Norm-Konvergenzaussage (4.1.17) wird dabei nur der symmetrische Fall betrachtet.

Satz 4.1.3 *Die Eigenwerte der symmetrischen $n \times n$-Matrix A seien im Intervall $[a, b]$, $0 < a \le b$, enthalten, also ist $\hat{\kappa}(A) \le \kappa := b/a$. Für die Vektoren der* Tschebyscheff-Iteration

$$\left\{ \begin{aligned} y^{(1)} &:= y^{(0)} + d(r - Ay^{(0)}), \\ y^{(k+1)} &:= \varrho_k\Big(y^{(k)} + d(r - Ay^{(k)})\Big) + (1-\varrho_k)y^{(k-1)}, \quad k = 1, 2, \ldots, \end{aligned} \right. \tag{4.1.16}$$

mit Startvektor $y^{(0)} \in \mathbb{R}^n$ *und den in (4.1.15) definierten Parametern gilt die Fehlerschranke*

$$\|y^{(k)} - z\|_2 \le \frac{1}{T_k(q)} \|y^{(0)} - z\|_2 \le 2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k \|y^{(0)} - z\|_2, \ k \ge 1. \tag{4.1.17}$$

Beweis Nach obiger Konstruktion gilt für jedes k

$$y^{(k)} - z = \hat{p}_k(A)\,(y^{(0)} - z).$$

Wegen der Symmetrie von A ist $\|\hat{p}_k(A)\|_2 = \varrho(\hat{p}_k(A)) \le 1/T_k(q)$, vgl. (4.1.13). ■

Bemerkung: Die Tschebyscheff-Iteration (4.1.16) besteht offensichtlich aus einer zusätzlichen Extrapolation ($1 < \varrho_k \le 2$) bei der einfachen, optimalen Relaxation (4.1.2), (4.1.3):

$$\begin{aligned} \tilde{y}^{(k+1)} &:= y^{(k)} + \frac{2}{a+b}(r - Ay^{(k)}), \\ y^{(k+1)} &= \varrho_k \tilde{y}^{(k+1)} + (1 - \varrho_k) y^{(k-1)}. \end{aligned}$$

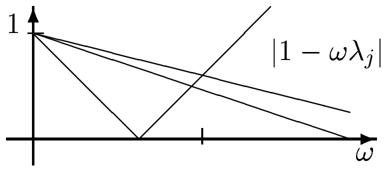
Praktische Durchführung

Sowohl bei der Richardson- als auch der Tschebyscheff-Iteration werden zur Durchführung gute (Ab-) Schätzungen der extremalen Eigenwerte $\lambda_1(A), \lambda_n(A)$ benötigt. Nach den Ergebnissen von §3 bietet sich dazu die Vektor-Iteration an. Diese kann sogar ohne zusätzlichen Aufwand mit der einfachen Relaxation (4.1.2) kombiniert werden. Sowohl bei Betrachtung des Fehlers $x^{(k)} - z$ als auch des Defekts $d^{(k)} := r - Ax^{(k)}$ entspricht diese nämlich einer Vektor-Iteration,

$$d^{(k+1)} := r - Ax^{(k+1)} = r - A(x^{(k)} + \omega d^{(k)}) = (I - \omega A) d^{(k)} \tag{4.1.18}$$

mit der Matrix $I - \omega A$. Diese Beziehung kann im symmetrisch-definiten Fall, $0 < \lambda_n \le \ldots \le \lambda_1$ ausgenutzt werden. Dazu sei an die Aussage

$$\varrho(I - \omega A) = \left\{ \begin{array}{ll} \omega\lambda_1 - 1 & \text{für} \quad \omega > \\ 1 - \omega\lambda_n & \text{für} \quad \omega \le \end{array} \right\} \hat{\omega} = \frac{2}{\lambda_n + \lambda_1}.$$



erinnert. Durch Variation von ω können daher λ_n und λ_1 bei (4.1.18) geschätzt werden:

a) Für $\omega := 1/\|A\| \le 1/\lambda_1$ ist $\varrho(I - \omega A) = 1 - \omega\lambda_n > 0$. Analog zu (3.4.3) bekommt man aus (4.1.18) eine Schätzung für λ_n.

b) Für beliebiges $x \in \mathbb{R}^n$ ist $\omega := 2/(R_A[x] + \lambda_n) \ge \hat{\omega}$. Mit diesem Wert ω ist daher $\varrho(I - \omega A) = \omega\lambda_1 - 1$ und aus der Eigenwert-Schätzung bei (4.1.18) kann eine Schätzung von λ_1 abgeleitet werden. Bei ungünstiger Wahl von x kann allerdings ω zu groß sein und $\varrho(I - \omega A) > 1$ gelten, was bei der Iteration zu wachsenden Fehlern gegenüber der Startnäherung führt.

c) Mit diesen Eigenwert-Schätzungen für λ_n und λ_1 kann jetzt auf die Richardson- oder Tschebyscheff-Iteration umgeschaltet werden.

Wegen des engen Zusammenhangs mit dem *Eigenwert-Problem* können Richardson- und Tschebyscheff-Methode auch zur Beschleunigung der Vektor-Iteration eingesetzt werden. Statt der Folge (3.3.2)

$$v^{(m)} := A^m z^{(0)} \text{ wird betrachtet } v^{(m)} := p_m(A) z^{(0)}, \ p_m \in \Pi_m.$$

Dabei wird p_m im Fall $0 \le \lambda_n \le \ldots \le \lambda_1$ so optimiert, daß der Konvergenzfaktor

$$\|p_m\|_{[0,\lambda_2]} \Big/ |p_m(\lambda_1)|$$

minimal wird. Dazu wird allerdings eine Schätzung für λ_2 benötigt. Bei Verwendung der optimalen Tschebyscheff-Polynome verbessert sich die Konvergenz von

$$\left|\frac{\lambda_2}{\lambda_1}\right|^m \quad \text{auf} \quad 1 \Big/ \left|T_m\Big(2\frac{\lambda_1}{\lambda_2} - 1\Big)\right| \le \left|\frac{\lambda_2}{\lambda_1}\right|^m \frac{2}{(1 + \sqrt{1 - \lambda_2/\lambda_1})^{2m}}$$

Ein zur Tschebyscheff-Iteration ähnliches Verfahren mit ‘eingebauter’ Parameterbestimmung wird im folgenden Abschnitt besprochen.

4.2 Das Verfahren der Konjugierten Gradienten

Bei einer postiv definiten symmetrischen Matrix kann die Lösung z des linearen Gleichungssystems $Ax = b$ als eindeutiges Minimum eines konvexen Funktionals $\varphi: \mathbb{R}^n \mapsto \mathbb{R}$ charakterisiert und daher durch ein Abstiegsverfahren approximiert werden.

Satz 4.2.1 *Die Matrix $A \in \mathbb{R}^{n \times n}$ sei symmetrisch und positiv definit. Dann wird durch*

$$(x, y)_A := y^\mathsf{T} A x, \quad \|x\|_A := \sqrt{(x,x)_A} = \sqrt{x^\mathsf{T} A x}, \tag{4.2.1}$$

$x, y \in \mathbb{R}^n$, ein Innenprodukt und eine Norm definiert. Die Lösung z des linearen Gleichungssystems $Ax = b$ ist eindeutiges Minimum der Funktion

$$\varphi(x) := \frac{1}{2} x^\mathsf{T} A x - b^\mathsf{T} x, \quad x \in \mathbb{R}^n. \tag{4.2.2}$$

Beweis Wegen der Definitheit von A ist $(x,x)_A = x^\mathsf{T} A x > 0 \ \forall x \ne 0$ und $\|x\|_A$ daher eine Norm. Für φ gilt mit $b = Az$ die Darstellung

$$\begin{aligned} 2\varphi(x) &= x^\mathsf{T} A x - 2b^\mathsf{T} x = x^\mathsf{T} A x - 2z^\mathsf{T} A x = (x,x)_A - 2(x,z)_A \\ &= \|x - z\|_A^2 - \|z\|_A^2. \end{aligned} \tag{4.2.3}$$

Also ist z eindeutiges Minimum von φ. ■

Die Funktion φ ist streng konvex. An der Stelle x ist die Richtung des steilsten Abstiegs

$$-\mathrm{grad}\varphi(x) = b - Ax =: d$$

gerade der Defekt von x. Bei einer Liniensuche in dieser Richtung kann das Minimum auf dem Strahl $s(t) := x + td,\ t \in \mathbb{R}$, exakt bestimmt werden:

$$\begin{aligned}
\varphi(s(t)) &= \frac{1}{2}(x+td)^\top A(x+td) - b^\top(x+td)\\
&= \frac{1}{2}t^2 d^\top Ad + td^\top Ax + \frac{1}{2}x^\top Ax - b^\top x - td^\top b\\
&= \frac{1}{2}t^2 d^\top Ad - td^\top d + \varphi(x)\\
&= \frac{1}{2}d^\top Ad\Big(t - \frac{d^\top d}{d^\top Ad}\Big)^2 + \varphi(x) - \frac{1}{2}\frac{(d^\top d)^2}{d^\top Ad}.
\end{aligned} \tag{4.2.4}$$

Das Minimum wird also in $\hat{t} = d^\top d/d^\top Ad = 1/R_A[d]$ angenommen. Diese *Methode des steilsten Abstiegs* läßt sich auch als Iterationsverfahren anwenden,

$$d^{(k)} := b - Ax^{(k)}, \quad x^{(k+1)} := x^{(k)} + \frac{1}{R_A[d^{(k)}]}d^{(k)}, \quad k = 0, 1, \ldots\ . \tag{4.2.5}$$

Allerdings ist ihre Konvergenz oft sehr schlecht. Dies sieht man, wenn man die verschobene Zielfunktion $2\varphi(y) + \|z\|_A^2 = \|y - z\|_A^2$ im Minimum von (4.2.4) betrachtet. Hier ist

$$2\varphi(s(\hat{t})) + \|z\|_A^2 = \|s(\hat{t}) - z\|_A^2 = \|x - z\|_A^2 - \frac{(d^\top d)^2}{d^\top Ad}.$$

Da zwischen der A-Norm und der Euklid-Norm der Zusammenhang

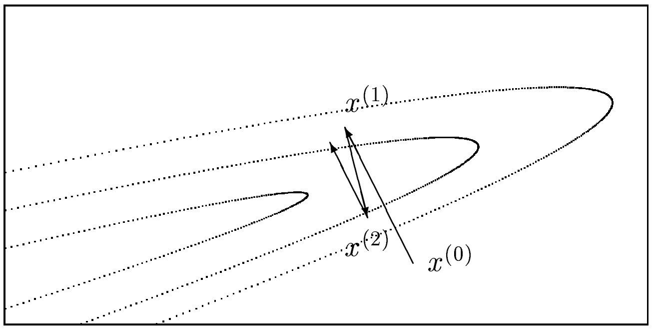
$$\lambda_{\min}(A)\|x\|_2^2 \le \|x\|_A^2 \le \lambda_{\max}(A)\|x\|_2^2 \tag{4.2.6}$$

(Satz 3.6.1) besteht, folgt mit $d^{(k)} = b - Ax^{(k)} = A(z - x^{(k)})$ weiter

$$\begin{aligned}
\|x^{(k+1)} - z\|_A^2 &= \|x^{(k)} - z\|_A^2 - \frac{\|A(x^{(k)} - z)\|_2^4}{\|A(x^{(k)} - z)\|_A^2} \le \|x^{(k)} - z\|_A^2 - \frac{\|A(x^{(k)} - z)\|_2^2}{\|A\|_2}\\
&\le \Big(1 - \frac{\lambda_{\min}(A)}{\lambda_{\max}(A)}\Big)\|x^{(k)} - z\|_A^2 = \Big(1 - \frac{1}{\hat{\kappa}(A)}\Big)\|x^{(k)} - z\|_A^2.
\end{aligned} \tag{4.2.7}$$

Für $\hat{\kappa}(A) \gg 1$ ergibt sich somit eine ähnlich langsame Konvergenz wie beim einfachen Relaxationsverfahren (4.1.5).

Geometrischer Hintergrund: Die Niveau-Flächen von φ sind Ellipsoide mit den Halbachsen $1/\sqrt{\lambda_j(A)}$ und Zentrum z. Für $\hat{\kappa} \gg 1$ sind die Ellipsoide sehr flach. Das Verfahren springt dann von einer Wand des Tals (im Bild: $\mathbb{R}^2$) zur anderen, kommt aber längs des Tals nur langsam voran.



Eine anschauliche Erklärung für die langsame Konvergenz ist das fehlende 'Gedächtnis' des Verfahrens, da in späteren Schritten beinahe in schon früher benutzte Richtungen gegangen wird.

Die Suchrichtung müßte also zur Verbesserung des Verfahrens modifiziert werden. Diese Richtung wird jetzt mit $p^{(k)}$ bezeichnet im Schritt k. Ein Verfahren mit vollem Gedächtnis sollte im k-ten Schritt mit Startvektor $x^{(0)} = 0$ sogar das Minimalproblem

$$\min\{\varphi(x) :\ x \in [p^{(1)}, \dots, p^{(k)}]\} \tag{4.2.8}$$

über dem von allen bisherigen Suchrichtungen aufgespannten Unterraum berechnen. Dazu seien

$$P_j := (p^{(1)}, \dots, p^{(j)}) \in \mathbb{R}^{n\times j},\ j \le k,$$

die mit den Suchrichtungen gebildeten Matrizen. Mit $t \in \mathbb{R}$, $y \in \mathbb{R}^{k-1}$, sei

$$s(t,y) := P_{k-1}y + tp^{(k)}$$

der Suchvektor. Für den Wert φ gilt dann

$$\begin{aligned} \varphi(s(t,y)) &= \frac{1}{2}(P_{k-1}y + tp^{(k)})^{\mathsf{T}} A (P_{k-1}y + tp^{(k)}) - b^{\mathsf{T}}(P_{k-1}y + tp^{(k)}) \\ &= \varphi(P_{k-1}y) + t(p^{(k)\mathsf{T}} A P_{k-1} y - b^{\mathsf{T}} p^{(k)}) + \frac{1}{2}t^2 (p^{(k)}, p^{(k)})_A. \end{aligned} \tag{4.2.9}$$

In dieser Beziehung kommt nur durch den Ausdruck $tp^{(k)\mathsf{T}} A P_{k-1} y = t(p^{(k)}, P_{k-1}y)_A$ eine Kopplung der Unbekannten t und y zustande. Gilt dagegen

$$(p^{(k)}, P_{k-1}y)_A = 0\ \forall y \quad \iff \quad (p^{(k)}, p^{(j)})_A = 0\ \forall j < k, \tag{4.2.10}$$

dann zerfällt das Minimalproblem (4.2.8) in zwei unabhängige Teilprobleme, dessen Minimum bzgl. t nach (4.2.4) in

$$t_k = b^{\mathsf{T}} p^{(k)} / p^{(k)\mathsf{T}} A p^{(k)} \tag{4.2.11}$$

angenommen wird. Als erstes Zwischenergebnis läßt sich also zusammenfassen:

Wenn schon $x^{(k-1)} \in [p^{(1)}, \dots, p^{(k-1)}]$ das $k-1$-dimensionale Minimalproblem (4.2.8) löst und $p^{(k)}$ A-orthogonal *zu $p^{(1)}, \dots, p^{(k-1)}$ ist, vgl. (4.2.10), dann führt die exakte, eindimensionale Liniensuche*

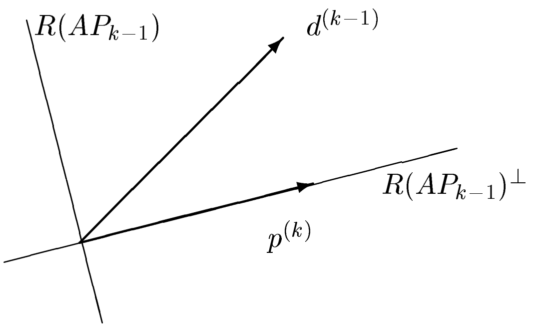
$$x^{(k)} := x^{(k-1)} + t_k p^{(k)} \tag{4.2.12}$$

auf die k-dimensionale Minimallösung in (4.2.8).

Die Abnahme der Zielfunktion φ ist dabei

$$\varphi(x^{(k)}) = \varphi(x^{(k-1)}) - \frac{1}{2}\frac{(b^{\mathsf{T}} p^{(k)})^2}{p^{(k)\mathsf{T}} A p^{(k)}} = \varphi(x^{(k-1)}) - \frac{1}{2}\frac{(d^{(k-1)\mathsf{T}} p^{(k)})^2}{p^{(k)\mathsf{T}} A p^{(k)}},$$

mit $d^{(k-1)} = b - Ax^{(k-1)}$, denn nach (4.2.10) ist $p^{(k)\mathsf{T}} A x^{(k-1)} = 0$. Damit dieser Iterationsschritt einen möglichst tiefen Abstieg erbringt, sollte die Suchrichtung $p^{(k)}$ in die Richtung von $d^{(k-1)}$ zeigen, unter Beachtung der Nebenbedingung (4.2.10). Diese Richtung kann folgendermaßen charakterisiert werden.



Lemma 4.2.2 *Der Vektor $p^{(k)}$, in dem*

$$\min\{\|p - d^{(k-1)}\|_2 : \ p \in \mathbb{R}^n, \, p^\mathsf{T} A P_{k-1} = 0\}$$

angenommen wird, hat die Form

$$p^{(k)} = d^{(k-1)} - AP_{k-1}v^{(k-1)}, \ v^{(k-1)} \in \mathbb{R}^{k-1}, \tag{4.2.13}$$

mit der Lösung $v^{(k-1)}$ des linearen Ausgleichsproblems

$$\min\{\|d^{(k-1)} - AP_{k-1}v\|_2 : \ v \in \mathbb{R}^{k-1}\}. \tag{4.2.14}$$

Beweis Nach Satz 1.1.2 erfüllt die Lösung von (4.2.14) die Normalengleichung

$$0 = (AP_{k-1})^\mathsf{T}(d^{(k-1)} - AP_{k-1}v^{(k-1)}) = (AP_{k-1})^\mathsf{T} p^{(k)},$$

d.h., (4.2.10). Daher ist auch $p^{(k)}$ die Orthogonalprojektion von $d^{(k-1)}$ auf $R(AP_{k-1})^\perp$, also der nächstgelegene Vektor. ■

Der Vektor $p^{(k)}$ aus (4.2.13), (4.2.14) kann nur dann als Suchrichtung dienen, wenn er nicht null ist. Nun ist die Bedingung $p^{(k)} = 0$ (für $x^{(k-1)} = P_{k-1}y$) äquivalent mit

$$\begin{aligned} 0 &= \min_v \|d^{(k-1)} - AP_{k-1}v\|_2 = \min_v \|b - Ax^{(k-1)} - AP_{k-1}v\|_2 \\ &= \min \|b - AP_{k-1}(y+v)\|_2. \end{aligned}$$

Dann gilt aber $b \in R(AP_{k-1}) = [Ap^{(1)}, \ldots, Ap^{(k-1)}] \iff z = A^{-1}b \in [p^{(1)}, \ldots, p^{(k-1)}]$. Dies bedeutet

$$p^{(k)} = 0 \iff d^{(k-1)} = 0 \iff x^{(k-1)} = z. \tag{4.2.15}$$

Die Konstruktion (4.2.14) der Suchrichtung bricht also nur zusammen, wenn die Lösung z schon durch $x^{(k-1)}$ angenommen wird ('günstiger Abbruch'). Insbesondere liefert (bei exakter Rechnung) spätestens die n-te Iterierte die exakte Lösung $x^{(n)} = z$. Die bisherigen Ergebnisse werden nun zur Formulierung einer 'Nullversion' des Verfahrens benutzt, bevor später überlegt wird, wie die Suchrichtungen $p^{(k)}$ effizient zu berechnen sind.

CG-0-Version:

(4.2.16)

$x^{(0)} := 0;\ d^{(0)} := b;\ k := 0;$

solange $d^{(k)} \neq 0$:

  $k := k + 1;$

  falls $k = 1$ dann $p^{(1)} := d^{(0)}$

   sonst $p^{(k)} := d^{(k-1)} - AP_{k-1}v^{(k-1)}$ nach (4.2.14);

  $t_k := p^{(k)\mathsf{T}} d^{(k-1)} / p^{(k)\mathsf{T}} A p^{(k)};$

  $x^{(k)} := x^{(k-1)} + t_k p^{(k)};$

  $d^{(k)} := b - Ax^{(k)};$

Für das Verfahren der Konjugierten Gradienten (conjugate-gradient-method, cg-method) können die wichtigsten theoretischen Eigenschaften schon aus der Nullversion abgeleitet werden.

Satz 4.2.3 *Nach Ausführung der Schritte $j = 1, \dots, k$ in (4.2.16) mit $d^{(j)} \neq 0$ gilt*

$$d^{(j)} = d^{(j-1)} - t_j A p^{(j)}, \quad P_j^\mathsf{T} d^{(j)} = 0, \; j = 1, \dots, k. \tag{4.2.17}$$

Die verwendeten Suchrichtungen und Defekte bilden Basen der Krylov-Räume

$$\mathcal{K}_j := [b, Ab, A^2 b, \dots, A^{j-1} b],$$

$$[p^{(1)}, \dots, p^{(j)}] = [d^{(0)}, \dots, d^{(j-1)}] = \mathcal{K}_j, \; j = 1, \dots, k. \tag{4.2.18}$$

Beweis a) Aus $x^{(j)} = x^{(j-1)} + t_j p^{(j)}$ folgt $d^{(j)} = b - Ax^{(j)} = d^{(j-1)} - t_j A p^{(j)}$. Im Unterraum $[p^{(1)}, \dots, p^{(j)}] = R(P_j)$ hat die Funktion φ die Gestalt

$$\varphi(P_j y) = \frac{1}{2} y^\mathsf{T} (P_j^\mathsf{T} A P_j) y - y^\mathsf{T} (P_j^\mathsf{T} b), \; y \in \mathbb{R}^j.$$

Die Minimalstelle $\hat{y}$ dieser Funktion erfüllt nach Satz 4.2.1, angewendet auf die ebenfalls positiv definite Matrix $P_j^\mathsf{T} A P_j$, das Gleichungssystem $0 = P_j^\mathsf{T} A P_j \hat{y} - P_j^\mathsf{T} b = P_j^\mathsf{T}(A P_j \hat{y} - b)$. Nach Konstruktion ist $P_j \hat{y} = x^{(j)}$, also gilt $P_j^\mathsf{T} d^{(j)} = 0$.

b) Für $j = 1$ ist (4.2.18) offensichtlich erfüllt. Unter der Annahme, (4.2.18) gelte für $j < k$, folgt durch Induktion aus (4.2.17) $d^{(j)} \in \mathcal{K}_{j+1}$ und daher nach (4.2.13)

$$p^{(j+1)} = \underbrace{d^{(j)}}_{\in \mathcal{K}_{j+1}} - A \underbrace{P_j v^{(j)}}_{\in \mathcal{K}_j} \in \mathcal{K}_{j+1}.$$

Da $d^{(j)} \perp \mathcal{K}_j$ und $p^{(j+1)} \perp_A \mathcal{K}_j$ ist, haben $[d^{(0)}, \dots, d^{(j)}]$ und $[p^{(1)}, \dots, p^{(j+1)}]$ jeweils Dimension $j + 1$ und stimmen daher mit $\mathcal{K}_{j+1}$ überein. ■

Die folgende Eigenschaft gab dem Konjugierte-Gradienten-Verfahren seinen Namen ('orthogonal'='konjugiert', $-\mathrm{grad}\,\varphi$=Defekt).

Satz 4.2.4 *Nach k Schritten von (4.2.16) (mit $d^{(j)} \neq 0 \forall j \leq k$) sind die Defekte $d^{(0)}, \dots, d^{(k)}$ paarweise orthogonal.*

Beweis Nach (4.2.17) und (4.2.18) gilt für jedes $j = 1, \dots, k$,

$$d^{(j)\mathsf{T}} P_j = 0 \iff d^{(j)} \perp [p^{(1)}, \dots, p^{(j)}] = \mathcal{K}_j = [d^{(0)}, \dots, d^{(j-1)}]. \quad ■$$

Somit konstruiert das cg-Verfahren sowohl Orthogonalbasen $\{d^{(0)}, \dots, d^{(j-1)}\}$ als auch A-Orthogonalbasen $\{p^{(1)}, \dots, p^{(j)}\}$ für $\mathcal{K}_j$, $j \leq k$. Offensichtlich ist (4.2.16) unabhängig von der Länge $\|p^{(k)}\|_2$. Daher kann im k-ten Schritt *ein* $p^{(k)}$ in dem durch (4.2.13) festgelegten eindimensionalen

Unterraum $\mathcal{K}_k \cap (A\mathcal{K}_{k-1})^\perp$ aus dem schon bekannten $d^{(k-1)}$, $\mathcal{K}_k \ni d^{(k-1)} \perp_2 \mathcal{K}_{k-1}$ durch Projektion berechnet werden bei Verwendung des A-Innenprodukts:

$$p^{(k)} := d^{(k-1)} - \sum_{j=1}^{k-1} \frac{(d^{(k-1)}, p^{(j)})_A}{(p^{(j)}, p^{(j)})_A} p^{(j)} = d^{(k-1)} - \frac{d^{(k-1)^\mathsf{T}} A p^{(k-1)}}{p^{(k-1)^\mathsf{T}} A p^{(k-1)}} p^{(k-1)}. \tag{4.2.19}$$

Dabei tritt nur noch $p^{(k-1)}$ auf, da $d^{(k-1)} \perp \mathcal{K}_{k-1} \supseteq A\mathcal{K}_{k-2} \ni Ap^{(j)}$, $j < k-1$. Mit (4.2.19) wird (4.2.16) ein sehr effizientes Verfahren, da pro Schritt nur zwei Matrix-Vektor-Multiplikationen benötigt werden bei $d^{(k)} = b - Ax^{(k)}$ und $Ap^{(k)}$. Die erste davon kann ebenfalls noch eingespart werden durch Berechnung des Defekts über (4.2.17). Wegen (4.2.19) gilt

$$t_k = \frac{p^{(k)^\mathsf{T}} d^{(k-1)}}{p^{(k)^\mathsf{T}} A p^{(k)}} = \frac{\|d^{(k-1)}\|_2^2}{p^{(k)^\mathsf{T}} A p^{(k)}}.$$

Aus (4.2.17) folgt für den Koeffizienten in (4.2.19) außerdem

$$\frac{t_{k-1} d^{(k-1)^\mathsf{T}} A p^{(k-1)}}{t_{k-1} p^{(k-1)^\mathsf{T}} A p^{(k-1)}} = -\frac{\|d^{(k-1)}\|_2^2}{\|d^{(k-2)}\|_2^2} =: -\beta_{k-1}.$$

Diese Überlegungen liefern die endgültige Form des *Verfahrens der Konjugierten Gradienten.*

CG-1-Version:

$$\boxed{\begin{array}{l} x^{(0)} := 0;\ d^{(0)} := b;\ p^{(1)} := b;\ k := 0; \\ \text{solange } d^{(k)} \neq 0: \\ \quad k := k+1; \\ \quad t_k := \|d^{(k-1)}\|_2^2 / p^{(k)^\mathsf{T}} A p^{(k)}; \\ \quad x^{(k)} := x^{(k-1)} + t_k p^{(k)}; \\ \quad d^{(k)} := d^{(k-1)} - t_k A p^{(k)}; \\ \quad \beta_k := \|d^{(k)}\|_2^2 / \|d^{(k-1)}\|_2^2; \\ \quad p^{(k+1)} := d^{(k)} + \beta_k p^{(k)}; \end{array}} \tag{4.2.20}$$

In dieser Form benötigt das Verfahren eine Matrix-Vektor-Multiplikation, drei Vektor-Linearkombinationen und zwei Innenprodukte, außerdem Platz für drei n-Vektoren. Zusammenfassend werden die wichtigsten Eigenschaften im nächsten Satz aufgeführt. Zusätzlich enthält dieser eine Fehlerschranke für den praktisch interessantesten Fall $k < n$.

Satz 4.2.5 *Nach k Schritten mit $d^{(j)} \neq 0, j < k$, gelten für die im CG-Verfahren (4.2.20) berechneten Größen folgende Aussagen.*

a) Die Suchrichtungen $\{p^{(1)}, \ldots, p^{(k)}\}$ sind A-orthogonal, die Defekte $\{d^{(0)}, \ldots, d^{(k-1)}\}$ orthogonal und jeweils Basen des Krylovraums $\mathcal{K}_k$,

$$[p^{(1)}, \ldots, p^{(k)}] = [d^{(0)}, \ldots, d^{(k-1)}] = \mathcal{K}_k = [b, Ab, A^2 b, \ldots, A^{k-1} b].$$

b) Die Näherung $x^{(k)}$ minimiert φ über $\mathcal{K}_k$, d.h.,

$$\varphi(x^{(k)}) = \min_{x \in \mathcal{K}_k} \varphi(x).$$

Daher ist $x^{(k)}$ exakt, wenn $z \in \mathcal{K}_k$, also spätestens für $k = n$.

c) In der A-Norm (4.2.1) gilt mit $q := (\hat\kappa(A)+1)/(\hat\kappa(A)-1)$ die Fehlerschranke

$$\begin{aligned} \|x^{(k)} - z\|_A &= \min\{\, \|g_k(A)(x^{(0)} - z)\|_A :\ g_k \in \Pi_k,\ g_k(0) = 1\} \\ &\le \frac{1}{T_k(q)}\|x^{(0)} - z\|_A \le 2\left(\frac{\sqrt{\hat\kappa(A)}-1}{\sqrt{\hat\kappa(A)}+1}\right)^k \|x^{(0)} - z\|_A. \end{aligned} \tag{4.2.21}$$

Beweis Die Behauptungen a) und b) fassen die Sätze 4.2.3-4.2.4 zusammen.

c) Nach (4.2.3) ist die Minimumsuche bei $\varphi(x)$ äquivalent zur Minimierung von $\|x - z\|_A^2 = 2\varphi(x) + \|z\|_A^2$. Da die Elemente von $\mathcal{K}_k$ die Gestalt $\sum_{j=0}^{k-1} y_j A^j b = f_{k-1}(A)b,\ y_j \in \mathbb{R},\ f_{k-1} \in \Pi_{k-1}$, besitzen, folgt aus b)

$$\|x^{(k)} - z\|_A = \min_{f_{k-1} \in \Pi_{k-1}} \|f_{k-1}(A)b - z\|_A.$$

Mit $b = Az$ kann die letzte Differenz auch in der Form $f_{k-1}(A)Az - z = -(I - f_{k-1}(A)A)z = -g_k(A)z$ geschrieben werden mit $g_k \in \Pi_k,\ g(0) = 1$, da $g_k(t) = 1 - tf_{k-1}(t)$. Daher entspricht die letzte Charakterisierung der ersten Zeile in (4.2.21), da auch $x^{(0)} = 0$.

Da A symmetrisch positiv definit ist, läßt es sich zeigen, daß A eine ebensolche Quadratwurzel $A^{1/2}$ besitzt. Mit dieser ist $\|x\|_A^2 = x^\mathsf{T} Ax = (A^{1/2}x)^\mathsf{T} A^{1/2}x = \|A^{1/2}x\|_2^2$. Damit kann die A-Norm bei (4.2.21) umgeformt werden,

$$\|g_k(A)z\|_A = \|A^{1/2}g_k(A)z\|_2 = \|g_k(A)A^{1/2}z\|_2 \le \|g_k(A)\|_2\|z\|_A = \varrho(g_k(A))\|z\|_A.$$

Die letzte Gleichung gilt wegen der Symmetrie von A, vgl. (3.1.9). Die Fehlerschranken in (4.2.21) ergeben sich nun dadurch, daß für das zu minimierende Polynom g_k in

$$\|x^{(k)} - z\|_A \le \min\{\max\{|g_k(\lambda)| :\ \lambda \in [\lambda_n, \lambda_1]\} :\ g_k \in \Pi_k, g_k(0) = 1\}\|x^{(0)} - z\|_A$$

analog zu Satz 4.1.2 das Tschebyscheff-Polynom zum optimalen Intervall $[\lambda_n, \lambda_1]$ eingesetzt wird. Die Schranken (4.2.21) folgen damit aus den entsprechenden Nachweisen von (4.1.13). ■

Die Fehlerschranke für das CG-Verfahren (in der A-Norm) stimmt mit derjenigen für die Tschebyscheff-Iteration überein, wenn dort ebenfalls die beste Eigenwert-Einschließung verwendet wird. Die Interpretation des CG-Verfahrens als eine Tschebyscheff-Iteration mit *eingebauter* Parameterbestimmung wird durch eine andere Formulierung noch hervorgehoben, bei der die Suchrichtungen $p^{(j)}$ aus (4.2.20) eliminiert wurden.

CG-2-Version:

$$
\boxed{
\begin{array}{l}
x^{(-1)} := 0;\ x^{(0)} := 0;\ d^{(0)} := b;\ k := 0; \\
\text{solange } d^{(k)} \neq 0: \\
\quad k := k + 1; \\
\quad \gamma_{k-1} := \|d^{(k-1)}\|_2^2 / \|d^{(k-1)}\|_A^2; \\
\quad \text{falls } k = 1 \text{ dann } \omega_1 := 1 \text{ sonst} \\
\qquad \beta_{k-1} := \|d^{(k-1)}\|_2^2 / \|d^{(k-2)}\|_2^2; \\
\qquad \omega_k := 1 \Big/ \Big(1 - \frac{\gamma_{k-1}}{\gamma_{k-2}} \beta_{k-1} / \omega_{k-1}\Big); \\
\quad x^{(k)} := \omega_k (x^{(k-1)} + \gamma_{k-1} d^{(k-1)}) + (1 - \omega_k) x^{(k-2)}; \\
\quad d^{(k)} := r - A x^{(k)};
\end{array}
}
\tag{4.2.22}
$$

Der eigentliche Iterationsschritt hat die gleiche Struktur wie bei der Tschebyscheff-Iteration (4.1.16), denn (4.2.22) besteht aus

$$
\left\{
\begin{array}{lcll}
\tilde{x}^{(k)} & := & x^{(k-1)} + d^{(k-1)} / R_A[d^{(k-1)}], & \text{einfacher Gradientenschritt (4.2.5)} \\
x^{(k)} & := & \omega_k \tilde{x}^{(k)} + (1 - \omega_k) x^{(k-2)}, & \text{+Extrapolationsschritt}
\end{array}
\right.
$$

Der wesentliche Unterschied zwischen beiden Verfahren ist also der, daß bei der Tschebyscheff-Iteration die Parameter aus vorab bekannten Eigenwert-Schranken berechnet werden müssen, während das CG-Verfahren alle Parameter automatisch bestimmt. Dabei minimieren die Parameter der Tschebyscheff-Iteration die Fehler*schranke* (4.1.9), während sich die CG-Koeffizienten an den *tatsächlichen* Fehler anpassen. Daher konvergiert das CG-Verfahren in der Praxis meist schneller als die Tschebyscheff-Iteration, die Fehlerschranke (4.2.21) ist ziemlich pessimistisch (vgl. Demo-Beispiel am Ende). Sie liefert aber den Ansatzpunkt für folgende Verbesserungen.

4.3 Vorkonditionierung

In der Numerik I wurde bei der Konstruktion von Iterationsverfahren der Begriff der regulären Zerlegung $A = M - N$ einer Matrix A eingeführt. Der reguläre *Hauptteil* M war dabei so zu wählen, daß mit dem Rest N die Iterationsmatrix $B = M^{-1}N$ kontraktiv war, $\varrho(M^{-1}N) < 1$. In der Sprechweise der letzten Abschnitte entspricht eine solche Zerlegung dem Übergang vom Linearen Gleichungssystem $Ax = b$ mit Matrix A zu dem äquivalenten mit der Matrix $I - B = I - M^{-1}N = M^{-1}A$, also zu

$$(M^{-1}A)\, x = M^{-1} b. \tag{4.3.1}$$

Der alten Forderung bei der Fixpunktiteration nach einem möglichst kleinen Konvergenzfaktor $\varrho(M^{-1}N)$ entspricht bei den hier besprochenen Verfahren aufgrund der Konvergenzaussagen der Sätze 4.1.2, 4.1.3, 4.2.5 die Forderung nach einer möglichst kleinen Kondition $\hat{\kappa}(M^{-1}A)$ (≥ 1). Daher nennt man den Übergang zum System (4.3.1) auch *Vorkonditionierung.*

Bei Anwendung dieses Ansatzes ist aber darauf zu achten, daß die Verfahren bei (4.3.1) mit $M^{-1}A$ noch einsetzbar sind. Bei Richardson- und Tschebyscheff-Iteration wurden reelle Eigenwerte gefordert, das CG-Verfahren setzt eine symmetrisch-definite Matrix voraus. Diese Voraussetzungen können durch Umformulierungen erfüllt werden, wenn gilt

$$M \text{ in (4.3.1) ist symmetrisch, positiv definit.} \tag{4.3.2}$$

Dann besitzt M nämlich eine *Cholesky-Zerlegung* $M = LL^\mathsf{T}$ mit einer unteren Dreieckmatrix L, die positive Diagonalelemente besitzt (vgl. Numerik I, Satz 2.3.4). Durch Multiplikation von (4.3.1) mit L^T ergibt sich das neue Gleichungssystem $L^{-1}Ax = L^{-1}b \iff$

$$(L^{-1}AL^{-1^\mathsf{T}})(L^\mathsf{T}x) = L^{-1}b$$

für die Hilfsvariable $y = L^\mathsf{T}x$. Da die Matrix $L^{-1}AL^{-1^\mathsf{T}} = L^\mathsf{T}(M^{-1}A)L^{-1^\mathsf{T}}$ dieses Systems symmetrisch, positiv definit und ähnlich zu $M^{-1}A$ ist, sind mit (4.3.2) Richardson- und Tschebyscheff-Iteration bei (4.3.1) einsetzbar. Die *vorkonditionierte Tschebyscheff-Iteration* etwa lautet

$$\begin{array}{rcl} M(\tilde{y}^{(k+1)} - y^{(k)}) & := & \dfrac{2}{\alpha+\beta}(r - Ay^{(k)}) \\ y^{(k+1)} & := & \varrho_k \tilde{y}^{(k+1)} + (1-\varrho_k)y^{(k-1)} \end{array}. \tag{4.3.3}$$

Dabei muß M so beschaffen sein, daß die Lösung des Gleichungssystems für $\tilde{y}^{(k+1)} - y^{(k)}$ (nicht M^{-1}!) einfach berechnet werden kann. Die Koeffizienten sind jetzt aus Eigenwert-Schranken $[\alpha, \beta] \ni \lambda_j(M^{-1}A)\,\forall j$ zu bestimmen, die Konvergenz hängt von β/α ab.

Das CG-Verfahren aber ist nur bei symmetrisch und definiten Problemen einsetzbar. Diese Eigenschaft besitzt die zu $M^{-1}A$ gehörige lineare Abbildung unter der Voraussetzung (4.3.2), wenn das ℓ_2-Innenprodukt durch das Produkt $(\cdot,\cdot)_M$ ersetzt wird, denn

$$(x, M^{-1}Ay)_M = (y^\mathsf{T}AM^{-1})Mx = y^\mathsf{T}Ax = y^\mathsf{T}MM^{-1}Ax = (M^{-1}Ax, y)_M.$$

In der Fehlerabschätzung (4.2.21) ist dann auch der Wert $\hat{\kappa}(A)$ durch $\hat{\kappa}(M^{-1}A)$ zu ersetzen.

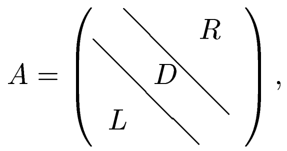
Präkonditioniertes CG-Verfahren:

$$\boxed{\begin{array}{l} x^{(0)} := 0;\ d^{(0)} := b;\ \text{löse } Mp^{(1)} := b;\ k := 0; \\ \text{solange } d^{(k)} \neq 0: \\ \quad \text{löse } Mc^{(k)} = d^{(k)}; \quad k := k+1; \\ \quad t_k := d^{(k-1)^\mathsf{T}}c^{(k-1)}/p^{(k)^\mathsf{T}}Ap^{(k)}; \\ \quad x^{(k)} := x^{(k-1)} + t_k p^{(k)}; \\ \quad d^{(k)} := d^{(k-1)} - t_k Ap^{(k)}; \\ \quad \beta_k := d^{(k)^\mathsf{T}}c^{(k)}/d^{(k-1)^\mathsf{T}}c^{(k-1)}; \\ \quad p^{(k+1)} := c^{(k)} + \beta_k p^{(k)}; \end{array}} \tag{4.3.4}$$

Strategien zur Wahl von M:

• Diagonale von A, $M = D := \mathrm{diag}(a_{ii})$, analog zum Gesamtschrittverfahren (Numerik I). Dies ist nur sinnvoll, wenn die Diagonalelemente von A sehr unterschiedliche Größen besitzen.

• Die Wahl '$M = D + L$' aus der Zerlegung

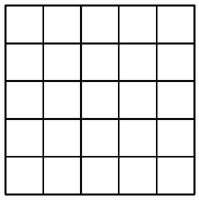


wie beim Einzelschrittverfahren scheidet aus, da $D+L$ nicht symmetrisch ist. Das Einzelschrittverfahren ergab sich bei schrittweiser Auflösung der i-ten Gleichung nach $x_i^{(k+1)}$, $i = 1, \ldots, n$. Wird daran ein umgekehrter Durchlauf $i = n, n-1, \ldots, 1$, angeschlossen, ergibt sich wegen $R = L^\mathsf{T}$ doch wieder eine symmetrische Iterationsmatrix. Gleiches gilt beim beschleunigten Einzelschrittverfahren, dem SOR-Verfahren, bei dem $D + L$ ersetzt wird durch $D + \omega L$ mit einem Parameter $0 < \omega < 2$. Ein Vorwärts- Rückwärts- Zyklus führt auf die symmetrische Matrix

$$M_\omega = (D + \omega L) D^{-1} (D + \omega L^\mathsf{T}). \tag{4.3.5}$$

Dieses Verfahren heißt SSOR ('symmetric successive overrelaxation'). Da dessen Konvergenz auf einem anderen Prinzip beruht als Tschebyscheff- und CG-Iteration, kann die SSOR-Iteration durch letztere weiter beschleunigt werden.

Beispiel 4.3.1 Zur Approximation einer Minimalfläche (Seifenhaut), die auf den Rändern eines Rechtecks gegebene Werte annimmt, kann diese Fläche durch ein elastisches Rechtecknetz angenähert werden. Die unbekannte Höhe am Netzknoten in Zeile i und Spalte j sei u_{ij}, dabei werden je $m+2$ Zeilen und Spalten $i, j \in \{0, \ldots, m+1\}$ verwendet. Im Knoten (i,j) erfüllt die Höhe bei der (approximierten) Minimalfläche die Bedingung



$$u_{ij} = \text{Mittelwert der Nachbarn.}$$

Insgesamt gilt also das Lineare Gleichungssystem

$$\begin{aligned} u_{ij} - \tfrac{1}{4}(u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1}) &= 0, && i, j = 1, \ldots, m, \\ u_{ij} &= \text{Randwert}, && i \text{ oder } j \in \{0, m+1\}, \end{aligned}$$

mit $n = m^2$ Unbekannten. Die Spektral-Kondition der zugehörigen Matrix ist $\hat\kappa \doteq 4m^2/\pi^2 \cong 0.4n$. In der folgenden Tabelle sind Kontraktionsfaktoren (bezogen auf einen Iterationsschritt) bei diesem Beispiel für die verschiedenen Verfahren aufgeführt. Dabei stehen in der ersten Zeile Approximationsformeln (für große m) der theoretisch hergeleiteten Schranken, in den folgenden Zeilen die bei zwei verschiedenen Problemgrößen praktisch beobachteten Faktoren (gemittelt über die Iterationsschritte). Die so berechnete Näherung u stimmt umso besser mit der tatsächlichen Minimalfläche überein, je größer m ist. Für genaue Approximationen sind daher große Gleichungssystem mit großer Kondition zu lösen. Auf diese Tatsache reagieren die verschiedenen Verfahren unterschiedlich empfindlich.

Kontraktionsfaktoren

Verfahren:	Relaxat.	SSOR	Richards. 32-Zyklus	Tscheb.	CG-Verf.	SSOR+ Tscheb.
Theorie:	$1-\frac{\pi^2}{2m^2}$	$1-\frac{\pi}{m}$	$1-\frac{16\pi^2}{m^2}$	$1-\frac{\pi}{m}$	$1-\frac{\pi}{m}$	$1-2\sqrt{\frac{\pi}{m}}$
$n=m^2=400$:	0.9775	0.8845	0.9278	0.9277	0.7460	0.6501
$n=m^2=1600$:	0.9887	0.9288	0.9564	0.9559	0.9159	0.7293

Im Vergleich der drei diskutierten Verfahren, Richardson- und Tschebyscheff-Iteration sowie CG-Verfahren gibt es im betrachteten Fall der symmetrisch–definiten Matrizen eine klare Abstufung nach Effizienz und Flexibilität. Hier ist auf jeden Fall das CG-Verfahren der Tschebyscheff-Iteration, und diese der Richardson-Iteration vorzuziehen (außer bei Speicherplatzbeschränkungen). Allerdings sind die Verfahren unterschiedlich gut in allgemeineren Situationen einsetzbar, die effizienteren Verfahren sind die weniger flexiblen. Die Entwicklung eines verallgemeinerten CG-Verfahrens für nicht-definite oder unsymmetrische Probleme ist ein sehr aktuelles Forschungsgebiet, keiner der vielen Ansätze kommt aber bisher ohne gravierende Nachteile aus. Die Tschebyscheff-Iteration ist auf Fälle verallgemeinerbar, in denen das Spektrum in einer Halbebene der komplexen Zahlen liegt, die die Null nicht enthält. Nur für die Richardson-Iteration gibt es keine solchen prinzipiellen Einschränkungen, die Konstruktion guter Parameter kann im Einzelfall allerdings schwierig sein.

Gegenüberstellung

	Sym.Pos.Def.	einseit. Spektrum	allgemein
Richardson	+	+	+
Tschebysch.	++	++	—
CG	+ + +	—	—

$\longrightarrow$ Schwierigkeitsgrad

$\downarrow$ Effizienz

4.4 Lanczos-Verfahren zur Eigenwert-Berechnung

Das Verfahren der Konjugierten Gradienten wurde im letzten Abschnitt als Minimierungsverfahren eingeführt. Diese Iteration erzeugt aber gleichzeitig auch ein *Modell* der Matrix A von niederem Rang, das dazu geeignet ist, *Eigenwert-Approximationen* für A zu berechnen ohne diese Matrix explizit zu kennen. Zur Herleitung werden k Schritte der CG-Iteration (4.2.20) ohne die Näherung $x^{(j)}$ betrachtet,

$$\begin{aligned} d^{(0)} &= b, & p^{(1)} &= b, \\ d^{(j)} &= d^{(j-1)} - t_j A p^{(j)}, & p^{(j+1)} &= d^{(j)} + \beta_j p^{(j)}, \quad j = 1, 2, \ldots, k. \end{aligned}$$

Darin können die Suchrichtungen $p^{(j)}$ eliminiert werden. Aus der ersten Gleichung folgt

$$Ap^{(j)} = \frac{1}{t_j}(d^{(j-1)} - d^{(j)}). \tag{4.4.1}$$

Wird diese Gleichung mit dem nächsthöheren Index verwendet, kann dort $p^{(j+1)}$ mit der zweiten Rekursionsformel ersetzt werden,

$$\frac{1}{t_{j+1}}(d^{(j)} - d^{(j+1)}) = Ap^{(j+1)} = Ad^{(j)} + \beta_j A p^{(j)} = Ad^{(j)} + \frac{\beta_j}{t_j}(d^{(j-1)} - d^{(j)}).$$

Für $j = 0$ ist $Ad^{(0)} = \frac{1}{t_1}(d^{(0)} - d^{(1)})$, da $p^{(1)} = d^{(0)}$. Setzt man $\beta_0 := 0$, gilt also

$$Ad^{(j)} = -\frac{\beta_j}{t_j} d^{(j-1)} + \Big(\frac{\beta_j}{t_j} + \frac{1}{t_{j+1}}\Big) d^{(j)} - \frac{1}{t_{j+1}} d^{(j+1)}, \quad j = 0, \ldots, k-1. \tag{4.4.2}$$

Nach Satz 4.2.4 bilden die Defekte $\{d^{(j)}\}_{j=0}^{k}$ eine Orthogonalbasis des Krylovraums $\mathcal{K}_{k+1}$. Faßt man die normierten Defekte zusammen zu einer orthogonalen Matrix Q_k, $Q_k^\mathsf{T} Q_k = I_k$, d.h.,

$$q^{(j)} := \frac{d^{(j-1)}}{\|d^{(j-1)}\|_2}, \ j = 1, \ldots, k, \quad Q_k := (q^{(1)}, \ldots, q^{(k)}) \in \mathbb{R}^{n \times k},$$

lautet die Beziehung (4.4.2) in Matrixschreibweise kompakt folgendermaßen

$$AQ_k = Q_k M_k + \gamma_k q^{(k+1)} {e^{(k)}}^\mathsf{T}. \tag{4.4.3}$$

Die Matrix M_k ist dabei einfach aufgebaut. Die Spalte $j > 0$ dieser Identität zeigt wegen (4.4.2), daß der Vektor $Aq^{(j)}$ eine Linearkombination von $q^{(j-1)}, q^{(j)}, q^{(j+1)}$ ist und M_k somit eine *Tridiagonalmatrix*. Die Symmetrie von A überträgt sich wegen

$$\underbrace{Q_k^\mathsf{T} A Q_k^\mathsf{T}}_{\text{sym.}} = \underbrace{Q_k^\mathsf{T} Q_k}_{=I_k} M_k + \gamma_k \underbrace{Q^\mathsf{T} q^{(k+1)}}_{=0} {e^{(k)}}^\mathsf{T} = M_k,$$

auf M_k. Daher gilt (4.4.3) mit der Matrix

$$M_k = \begin{pmatrix} \alpha_1 & \gamma_1 & & \\ \gamma_1 & \alpha_2 & \gamma_2 & \\ & \ddots & \ddots & \ddots \\ & & \gamma_{k-1} & \alpha_k \end{pmatrix} \in \mathbb{R}^{k \times k}. \tag{4.4.4}$$

Diese Beziehung (4.4.3) kann wegen der Orthonormalität der $q^{(j)}$ auch direkt zur Berechnung von M_k und Q_k dienen. Dazu wird die j-te Spalte von (4.4.3) betrachtet und nach $q^{(j+1)}$ aufgelöst:

$$\gamma_j q^{(j+1)} = (A - \alpha_j I) q^{(j)} - \gamma_{j-1} q^{(j-1)}.$$

Für die Koeffizienten dieser Rekursion ergeben sich die Werte

$$\begin{aligned}
\alpha_j &= {q^{(j)}}^\mathsf{T} A q^{(j)}, \quad \text{da } 0 = \gamma_j {q^{(j)}}^\mathsf{T} q^{(j+1)} = {q^{(j)}}^\mathsf{T} A q^{(j)} - \alpha_j, \\
\gamma_j &= \|(A - \alpha_j I) q^{(j)} - \gamma_{j-1} q^{(j-1)}\|_2.
\end{aligned}$$

Daher liefert das folgende Verfahren (bis auf Vorzeichen) die Größen aus der Identität (4.4.3), (4.4.4). Der Startvektor ist nun beliebig, $d^{(0)} \neq 0$.

Lanczos-Verfahren:

$$\boxed{\begin{array}{l}
\gamma_0 := \|d^{(0)}\|_2;\ q^{(0)} := 0;\ j := 0; \\
\text{solange } \gamma_j \neq 0: \\
\quad q^{(j+1)} := d^{(j)}/\gamma_j;\ j := j+1;\ \alpha_j := {q^{(j)}}^\mathsf{T} A q^{(j)}; \\
\quad d^{(j)} = (A - \alpha_j I) q^{(j)} - \gamma_{j-1} q^{(j-1)}; \\
\quad \gamma_j := \|d^{(j)}\|_2;
\end{array}} \qquad (4.4.5)$$

Falls dieses Verfahren erst mit $\gamma_n = 0$ abbricht, liefert es (bei exakter Rechnung) eine *Tridiagonalisierung* $M_n = Q_n^\mathsf{T} A Q_n$ von A nach einem zu Satz 3.2.2 unterschiedlichen Prinzip. Da die Vektoren $\{q^{(1)}, \ldots, q^{k)}\}$ aber eine Basis des Krylovraums $\mathcal{K}_k$ bilden, was an (4.4.5) direkt ablesbar ist, sind schon für $k \ll n$ die Eigenwerte der kleineren Matrix M_k gute Approximationen der *extremalen* Eigenwerte von A.

Der Grund dafür ist das Courantsche Maximum-Minimum-Prinzip, Satz 3.6.1. Zur Diskussion seien $\lambda_1 \geq \ldots \geq \lambda_n$ die Eigenwerte von A und

$$\lambda_1^{(k)} \geq \ldots \geq \lambda_k^{(k)} \quad \text{die Eigenwerte von } M_k \text{ aus (4.4.4),(4.4.5).}$$

Letztere können mit den Verfahren aus §3.4 effizient berechnet werden. Wegen $M_k = Q_k^\mathsf{T} A Q_k$, $Q_k \mathbb{R}^k = \mathcal{K}_k$, gilt z.B. für den maximalen Eigenwert

$$\lambda_1^{(k)} = \max_{y \in \mathbb{R}^k} \frac{y^\mathsf{T} M_k y}{y^\mathsf{T} y} = \max_{y \in \mathbb{R}^k} \frac{y^\mathsf{T} Q_k^\mathsf{T} A Q_k y}{y^\mathsf{T} Q_k^\mathsf{T} Q_k y} = \max_{x \in \mathcal{K}_k} \frac{x^\mathsf{T} A x}{x^\mathsf{T} x} \leq \max_{x \in \mathbb{R}^n} \frac{x^\mathsf{T} A x}{x^\mathsf{T} x} = \lambda_1. \qquad (4.4.6)$$

Ähnlich wie beim CG-Verfahren minimiert $\lambda_k^{(k)}$ bzw. maximiert $\lambda_1^{(k)}$ den Rayleighquotienten $R_A[x]$ auf dem Krylov-Unterraum. Eine Fehlerschranke für diese beiden Eigenwerte läßt sich daher analog zu Satz 4.2.5 herleiten.

Satz 4.4.1 *Die symmetrische Matrix A besitze die Eigenwerte $\lambda_1 \geq \ldots \geq \lambda_n$ und die orthonormalen Eigenvektoren $x^{(1)}, \ldots, x^{(n)}$. Nach k Schritten des Lanczos-Verfahrens (4.4.5) gilt für den größten Eigenwert $\lambda_1^{(k)}$ der Matrix M_k aus (4.4.4) die Fehlerschranke*

$$\lambda_1 \geq \lambda_1^{(k)} \geq \lambda_1 - (\lambda_1 - \lambda_n) \frac{\tan^2 \varphi}{T_{k-1}^2 (1 + 2\varrho_1)}, \qquad (4.4.7)$$

wobei $\cos\varphi_1 = |q^{(1)}{}^\mathsf{T} x^{(1)}|$, $\varrho_1 = (\lambda_1-\lambda_2)/(\lambda_2-\lambda_n)$ *und* T_{k-1} *das Tschebyscheff-Polynom (4.1.11) vom Grad* $k-1$ *ist.*

Beweis Jedes Element des Krylovraums $x \in \mathcal{K}_k$ hat die Gestalt $x = p(A)d^{(0)} = \sum_{j=1}^n p(\lambda_j)\xi_j x^{(j)}$, $p \in \Pi_{k-1}$, wobei $d^{(0)} = \sum_{j=1}^n \xi_j x^{(j)}$. Nach (4.4.6) ist

$$\lambda_1^{(k)} = \max_{p\,\in\,\Pi_{k-1}} \frac{d^{(0)}{}^\mathsf{T} p(A)Ap(A)d^{(0)}}{d^{(0)}{}^\mathsf{T} p(A)^2 d^{(0)}} = \max_{p\,\in\,\Pi_{k-1}} \frac{\sum_{j=1}^n \xi_j^2 p(\lambda_j)^2\lambda_j}{\sum_{j=1}^n \xi_j^2 p(\lambda_j)^2}.$$

Mit beliebigen Polynomen $p \in \Pi_{k-1}$ gilt also

$$\frac{d^{(0)}{}^\mathsf{T} p(A)Ap(A)d^{(0)}}{d^{(0)}{}^\mathsf{T} p(A)^2 d^{(0)}} = \lambda_1 - \frac{\sum_{j=2}^n \xi_j^2 p(\lambda_j)^2(\lambda_1-\lambda_j)}{\sum_{j=1}^n \xi_j^2 p(\lambda_j)^2} \ge \lambda_1 - (\lambda_1-\lambda_n)\frac{\sum_{j=2}^n \xi_j^2 p(\lambda_j)^2}{\xi_1^2 p(\lambda_1)^2}. \quad (4.4.8)$$

Das Verhältnis $|p(\lambda_j)/p(\lambda_1)|$, $j>1$, läßt sich dadurch auf einen minimalen Wert bringen, daß man p als Tschebyscheff-Polynom zum Intervall $[\lambda_n,\lambda_2]$ wählt. Nach Satz 4.1.2 gilt mit

$$p(\lambda) = T_{k-1}\Big(\frac{\lambda_2+\lambda_n-2\lambda}{\lambda_2-\lambda_n}\Big) \;\Rightarrow\; |p(\lambda_j)| \le 1,\ j=2,\dots,n.$$

Da $|p(\lambda_1)| = |T_{k-1}(-1-2\varrho_1)| = |T_{k-1}(1+2\varrho_1)|$ ist, folgt damit $|p(\lambda_j)/p(\lambda_1)| \le 1/|T_{k-1}(1+2\varrho_1)|$, $j>1$. Außerdem ist

$$\frac{\sum_{j=2}^n \xi_j^2}{\xi_1^2} = \frac{\|d^{(0)}\|_2^2}{(x^{(1)}{}^\mathsf{T} d^{(0)})^2} - 1 = \frac{1}{\cos^2\varphi_1} - 1 = \tan^2\varphi_1.$$

Mit diesen Schranken folgt aus (4.4.8) die Behauptung. ■

Für $k \ll n$ fällt der Aufwand zur Berechnung des größten Eigenwerts von M_k gegenüber (4.4.5) nicht ins Gewicht, der Aufwand für k Schritte ist daher ungefähr so groß wie bei der Vektor-Iteration (3.4.3). Bis auf Konstanten ist der Fehler bei definiten Matrizen ($\lambda_n \ge 0 \Rightarrow 1+2\varrho_1 \ge 2\lambda_1/\lambda_2 - 1$) nach k Matrix-Vektor-Multiplikationen jeweils

$$\begin{array}{ll} \Big(\frac{\lambda_2}{\lambda_1}\Big)^{2k} & \text{bei sym. Vektoriteration,} \\ \frac{1}{T_k(2\lambda_1/\lambda_2-1)^2} \le 4\Big(\frac{\lambda_2}{\lambda_1}\Big)^{2k}\Big(1+\sqrt{1-\lambda_2/\lambda_1}\Big)^{-4k} & \text{bei Lanczos.} \end{array} \quad (4.4.9)$$

Zum Vergleich der Schranken aus (4.4.9) seien Zahlenwerte für $k=20$ und verschiedene Verhältnisse λ_1/λ_2 angegeben:

$\lambda_1/\lambda_2 =$	1.5	1.1	1.01
Vekt-It.	$9_{10}-8$	$2_{10}-2$	0.7
Lanczos	$5_{10}-23$	$6_{10}-11$	$1_{10}-3$

Das Lanczos-Verfahren benutzt i.w. die gleiche Information wie die Vektoriteration, nämlich die Krylov-Vektoren $\{A^j q^{(1)}\}$, leitet daraus aber wesentlich bessere Eigenwert-Schätzungen ab. Beim praktischen Einsatz ist das Lanczos-Verfahren aber recht anfällig gegen Rundungsfehler und erfordert Zusatzmaßnahmen zur Absicherung der Ergebnisse.

5 Fortsetzungsverfahren für nichtlineare Probleme

5.1 Die numerische Verfolgung von Lösungskurven

Das Standardverfahren zur Lösung des nichtlinearen Gleichungssystems

$$F(x) = 0, \qquad F: \mathbb{R}^n \mapsto \mathbb{R}^n, \tag{5.1.1}$$

ist das *Newton-Verfahren* (vgl. Numerik I, §6.2): $x^{(0)} \in \mathbb{R}^n$,

$$F'(x^{(k)})\Big(x^{(k+1)} - x^{(k)}\Big) = -t_k F(x^{(k)}),\ k = 0, 1, \dots, \tag{5.1.2}$$

mit Parametern $t_k > 0$. Beim klassischen Newtonverfahren mit $t_k \equiv 1$ kann die Konvergenz nur in einer, evtl. sehr kleinen, Umgebung der Lösung z von (5.1.1) garantiert werden. Eine Verbesserung war die Einführung des Parameters t_k, der mit Hilfe einer *Liniensuche* so bestimmt wird, daß der Funktionswert in $x^{(k+1)}$ kleiner wird,

$$\varphi(x^{(k+1)}) < \varphi(x^{(k)}) \quad \text{mit} \quad \varphi(x) := \|F(x)\|_2^2.$$

Das Prinzip entspricht dem des einfachen Gradientenverfahrens (4.2.5). Aber auch bei der Liniensuche ist die Konvergenz gegen z nicht sicher, da die Folge $\{x^{(k)}\}$ aus (5.1.2) gegen ein lokales Minimum $\bar{x}$ mit $\varphi(\bar{x}) > 0$ konvergieren kann.

Ein alternativer Zugang nutzt die lokal garantierte Konvergenz des Newtonverfahrens aus. Man kann nämlich erwarten, daß die Lösung eines leicht *deformierten* Problems (5.1.1) eine gute Startnäherung für die Iteration (5.1.2) liefert. Konkret wird ein zusätzlicher Parameter λ eingeführt, etwa $\lambda \in [0, 1]$, und die *Problemschar*

$$0 = G(x, \lambda) := \lambda F(x) + (1 - \lambda) F_0(x),\ \lambda \in [0, 1], \tag{5.1.3}$$

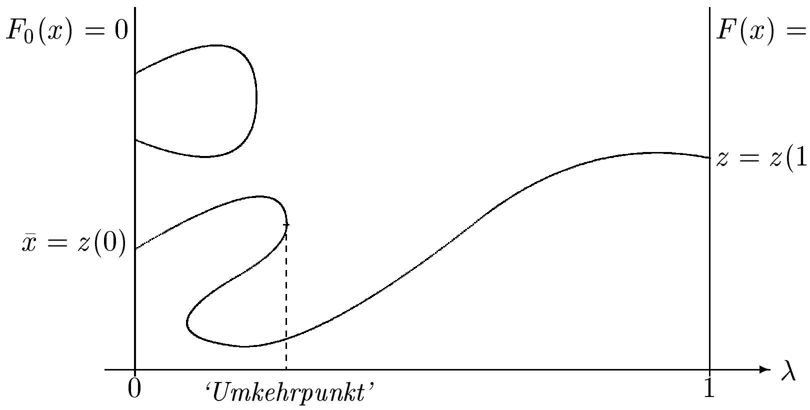
betrachtet ('Einbettung'). Dabei ist F_0 eine Abbildung, für die eine Nullstelle $\bar{x}$, $F_0(\bar{x}) = 0$, bekannt ist, z.B.,

$$\begin{aligned} F_0(x) := x \quad &\Rightarrow \quad \bar{x} = 0, \\ F_0(x) := F(x) - F(\tilde{x}) \quad &\Rightarrow \quad \bar{x} = \tilde{x}. \end{aligned}$$

Unter geeigneten Voraussetzungen hängen die Lösungen von (5.1.3) stetig vom Parameter λ ab, sie bilden eine *Lösungskurve*. Daher kann man versuchen, dieser Kurve zu folgen,

vom bekannten Startpunkt $\quad z(0) = \bar{x} \quad$ mit $\quad G(x, 0) = F_0(x) = 0,$
zum gesuchten Endpunkt $\quad z(1) = z \quad$ mit $\quad G(x, 1) = F(x) = 0.$

Im folgenden Bild sind mögliche Situationen skizziert, insbesondere muß $z(\lambda)$ keine (eindeutige) Funktion von λ sein.



Parameterabhängige Gleichungssysteme $G(x, \lambda) = 0$ müssen in der Praxis oft auch unabhängig von der Motivation (5.1.3) behandelt werden, etwa als Abhängigkeitsanalysen über Material- oder Modellparameter. Daher wird nun das abstrakte Problem

$$G(x, \lambda) = 0, \quad G: \ D \times [0,1] \mapsto \mathbb{R}^n, \ D \subset \mathbb{R}^n, \tag{5.1.4}$$

betrachtet. Der Satz über implizite Funktionen liefert hier ein einfaches Kriterium für die Existenz einer glatten und bezüglich λ eindeutigen Lösungskurve.

Satz 5.1.1 *D sei offen und $G \in C^1(D \times [0,1])$. Die Ableitung $G_x = \partial G/\partial x$ sei überall invertierbar und es geltefür alle $(x, \lambda) \in D \times [0,1]$:*

$$\|G_x(x, \lambda)^{-1}\| \le \alpha, \ \|G_\lambda(x, \lambda)\| \le \beta_1. \tag{5.1.5}$$

Es sei $z_0 \in D$ eine Lösung von $G(x, 0) = 0$ so, daß gilt

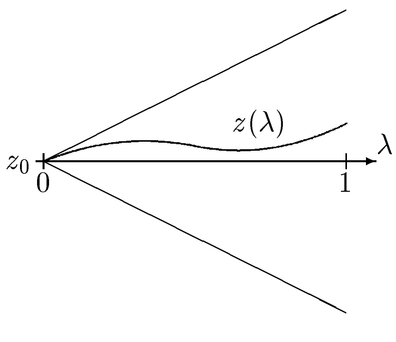
$$\{x \in \mathbb{R}^n : \ \|x - z_0\| \le \alpha\beta_1\} \subseteq D.$$

Dann gibt es eine in z_0 beginnende, stetig differenzierbare Lösungsfunktion $z(\lambda)$ mit $z(0) = z_0$, $G(z(\lambda), \lambda) \equiv 0$, für $\lambda \in [0,1]$. Ihre Ableitung z' erfüllt die Beziehung

$$z'(\lambda) = \frac{dz}{d\lambda}(\lambda) = -G_x(z(\lambda), \lambda)^{-1} G_\lambda(z(\lambda), \lambda). \tag{5.1.6}$$

Beweis Aus der Regularität von G_x und dem Satz über implizite Funktionen folgt die lokale Existenz von $z(\lambda)$ und die Gültigkeit von (5.1.6), da

$$G(z(\lambda), \lambda) \equiv 0 \ \Rightarrow \ G_x z' + G_\lambda \equiv 0.$$



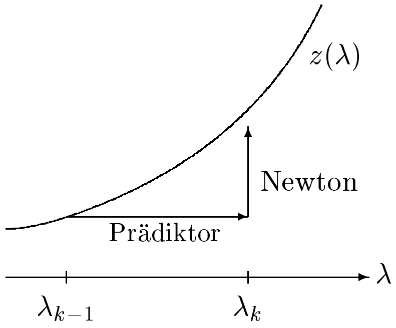
Die daraus abgeleitete Gleichung (5.1.6) ist eine Differentialgleichung $z' = f(z, \lambda)$ mit stetiger und beschränkter rechter Seite f, $\|f\| \le \alpha\beta_1$ in $D \times [0,1]$. Nach dem Existenzsatz von Peano existiert daher eine Lösung $z(\lambda)$, die in dem Kegel $\|z(\lambda) - z(0)\| \le \alpha\beta_1\lambda$ verläuft. Wenn dieser ganz in $D \times [0,1]$ liegt, erreicht die Lösung $z(\lambda)$ den rechten Rand, $\lambda = 1$. ∎

Unter der Voraussetzung (5.1.5) können insbesondere keine *Umkehrpunkte* wie im obigen Bild auftreten, da in diesen die Ableitung G_x singulär wird. Zunächst wird die Verfolgung von Kurven ohne Umkehrpunkte behandelt. Tatsächlich berechnet werden dabei allerdings nur einzelne Punkte der Lösungskurve in Parameterwerten $0 = \lambda_0, \lambda_1, \ldots$. Mögliche

Strategien

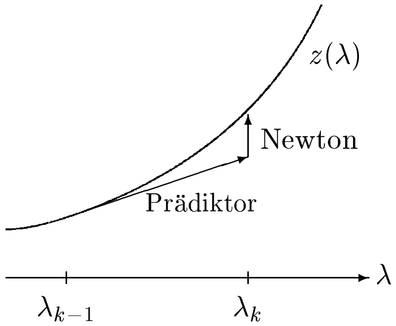
- *Klassische Fortsetzung*:

  Der Wert $z(\lambda_{k-1})$ wird als Startwert der Newton-Iteration zur Lösung von $G(x, \lambda_k) = 0$ im Nachbarpunkt $\lambda_k > \lambda_{k-1}$ verwendet. Bei großen Werten der Ableitung z' sind hierbei aber sehr kleine Schritte $h_k = \lambda_k - \lambda_{k-1}$ notwendig.



- *Tangentiale Fortsetzung*:

  Wegen (5.1.6) ist in λ_{k-1} mit $z(\lambda_{k-1})$ auch die Ableitung $z'(\lambda_{k-1})$ bekannt. Eine Startnäherung $x^{(0)}(\lambda_k)$ für $z(\lambda_k)$ im Nachbarpunkt $\lambda_k = \lambda_{k-1} + h_k$ kann daher mit der Tangente in λ_{k-1} konstruiert werden. Dies ergibt ein zweistufiges Verfahren



$$\text{Prädiktor-Schritt:} \quad \left\{ \begin{array}{l} x^{(0)}(\lambda_k) := z(\lambda_{k-1}) + h_k z'(\lambda_{k-1}), \quad \text{mit} \\ G_x\Big(z(\lambda_{k-1}), \lambda_{k-1}\Big) z'(\lambda_{k-1}) = -G_\lambda\Big(z(\lambda_{k-1}), \lambda_{k-1}\Big) \end{array} \right. \tag{5.1.7}$$

$$\text{Korrektor-Schritt:} \quad \left\{ \begin{array}{l} G_x\Big(x^{(j)}(\lambda_k), \lambda_k\Big)\Big(x^{(j+1)}(\lambda_k) - x^{(j)}(\lambda_k)\Big) = -G\Big(x^{(j)}(\lambda_k), \lambda_k\Big), \\ j = 0, 1, \ldots \,. \end{array} \right. \tag{5.1.8}$$

Im folgenden wird nur die zweite, bessere Strategie untersucht. Auch bei dieser ist grundsätzlich die Schrittweite h_k so klein zu wählen, daß die Newton-Iteration (5.1.8) mit dem Startwert $x^{(0)}(\lambda_k)$ konvergiert, d.h., in Abhängigkeit von der Krümmung der Kurve. Meist ist man aber am genauen Verlauf der Kurve $z(\lambda)$ für $\lambda < 1$ nicht interessiert und will schnell zur Stelle $\lambda = 1$ kommen, etwa durch möglichst große Schritte. Der folgende Satz charakterisiert die größte Schrittweite, für die noch die Konvergenz des Prädiktor-Korrektor-Verfahrens garantiert ist.

Satz 5.1.2 *Es sei D offen, $G \in C^2(D \times [0,1])$ und*

$$\|G_x(x,\lambda)^{-1}\| \le \alpha, \ \|G_\lambda\| \le \beta_1, \ \max\{\|G_{xx}\|, \|G_{x\lambda}\|, \|G_{\lambda\lambda}\|\} \le \beta_2 \tag{5.1.9}$$

$\forall (x, \lambda) \in D \times [0,1]$. *Damit sei*

$$\bar{h} := \frac{1}{\alpha(1 + \alpha\beta_1)\beta_2}. \tag{5.1.10}$$

Die Lösung $z(\lambda_{k-1})$, $\lambda_{k-1} \in [0,1)$, von (5.1.4) sei einschließlich der Kugel

$$\{x \in \mathbb{R}^n : \|x - z(\lambda_{k-1})\| \le \alpha\beta_1\bar{h}\}$$

in D enthalten. Dann konvergiert das Prädiktor-Korrektor-Verfahren (5.1.7), (5.1.8) gegen $z(\lambda_k)$ wenn $\lambda_k \le 1$ gilt und

$$h_k = \lambda_k - \lambda_{k-1} \le \bar{h}.$$

Beweis Nach Satz 6.2.2 aus der Numerik I konvergiert die Newton-Iteration (5.1.8), wenn für den Startwert $x^{(0)}$ gilt

$$\|x^{(0)}(\lambda_k) - z(\lambda_k)\| < \frac{2}{3\alpha\beta_2}. \tag{5.1.11}$$

Um diese Aussage hier anzuwenden, ist die Abweichung von Kurve und Tangente abzuschätzen, der Satz von Taylor zeigt dazu

$$\|z(\lambda_{k-1}) + h_k z'(\lambda_{k-1}) - z(\lambda_k)\| \le \frac{1}{2}h_k^2 \max_{s\in[0,h]} \|z''(\lambda_{k-1} + s)\|.$$

Daher wird zunächst eine Schranke für z'' in D benötigt. Nach Satz 5.1.1 ist z differenzierbar und $\|z'\| \le \alpha\beta_1$, demnach liegen $x^{(0)}(\lambda_k)$ und $z(\lambda_k)$ in D. Wegen $G \in C^2$ ist die Ableitung z'' stetig. Sie ergibt sich aus der Kettenregel

$$G_x z' + G_\lambda \equiv 0 \;\Rightarrow\; G_x z'' + G_{xx}z'z' + 2G_{x\lambda}z' + G_{\lambda\lambda} \equiv 0,$$

durch Multiplikation mit G_x^{-1}. Daher ist

$$\|z''\| \le \alpha\beta_2(1 + \|z'\|)^2 \le \alpha\beta_2(1 + \alpha\beta_1)^2.$$

Somit läßt sich die Konvergenzbedingung (5.1.11) mit der Taylor-Schranke durch die Forderung

$$\|x^{(0)}(\lambda_k) - z(\lambda_k)\| \le \frac{1}{2}h_k^2\alpha\beta_2(1 + \alpha\beta_1)^2 \overset{!}{<} \frac{2}{3\alpha\beta_2}$$

erzwingen. Letztere ist aber unter der Voraussetzung $h_k \le \bar{h}$ erfüllt. ■

Dieser Satz garantiert, daß man mit Schrittweiten $h_k \le \bar{h}$ die Kurve $z(\lambda)$ tatsächlich durch das ganze Intervall $[0,1]$ verfolgen kann, also mit ca. $1/\bar{h}$ Schritten. Die Aussage des Satzes hat aber kaum eine praktische Bedeutung, da die Schranken (5.1.9) selten bekannt sind. Aber auch aus Effizienzgründen ist es i.a. nicht sinnvoll, mit maximalen Schrittweiten $h_k \cong \bar{h}$ vorzugehen. Dies liegt an der quadratischen Konvergenz des Newtonverfahrens, da der Startfehler einer guten Näherung wesentlich stärker reduziert wird als der einer schlechten. Für die Fehler $e_j := \|x^{(j)}(\lambda_k) - z(\lambda_k)\|$ lautet die Konvergenzaussage von Satz I.6.2.2

$$e_j \le \frac{3}{2}\alpha\beta_2\, e_{j-1}^2. \tag{5.1.12}$$

Für $h \cong \bar{h}$ wird der Fehler am Anfang nur langsam kleiner, da $e_1 \le (\frac{3}{2}\alpha\beta_2 e_0)e_0 \cong e_0$ nach Konstruktion im Beweis des letzten Satzes. Um wieder nahe genug an die Lösungskurve zu kommen, werden daher viele Newton-Schritte benötigt. Da der Prädiktor-Schritt (5.1.7) den gleichen Aufwand wie eine Newton-Korrektor (5.1.8) erfordert, ist es günstiger, und sicherer, mehrere kleine λ-Schritte mit wenig Korrekturen pro Schritt zu machen. Dazu muß eine gute *Kontraktion* der Newton-Iteration gesichert werden, etwa $\frac{3}{2}\alpha\beta_2 e_0 \overset{!}{\leq} \frac{1}{10} \quad \Rightarrow \quad e_1 \leq \frac{1}{10}e_0$, $e_2 \leq \frac{1}{1000}e_0, \ldots$. Die unbekannte tatsächliche Kontraktion e_1/e_0 kann dabei aus den Newton-Korrekturen $\|x^{(j+1)} - x^{(j)}\|$ geschätzt werden. Wegen der quadratischen Konvergenz ist nämlich $\|x^{(j+1)} - x^{(j)}\| = \|(x^{(j)} - z) - (x^{(j+1)} - z)\| = e_j + \mathcal{O}(e_j^2)$. Mit $q \ll 1$ kann daher aus

$$\frac{e_1}{e_0} \cong \frac{\|x^{(2)} - x^{(1)}\|}{\|x^{(1)} - x^{(0)}\|} \overset{!}{\leq} q \quad (q = \frac{1}{10} \cdots \frac{1}{2}) \tag{5.1.13}$$

auf eine akzeptable Konvergenz der Newton-Korrektur (5.1.8) geschlossen werden.

In der Praxis läßt sich diese Forderung mit Hilfe einer *Schrittweitensteuerung* realisieren. Dazu werden zunächst mit einer geschätzten Schrittweite h_k (z.B. der aus dem letzten λ-Schritt) der Prädiktor- (5.1.7) und zwei Korrektorschritte (5.1.8) durchgeführt. Entspricht dabei die Kontraktion nicht der Bedingung (5.1.13), wird dieser Versuch verworfen und die Schrittweite verkleinert, etwa $h_k \to h_k/2$ (z.B., 'in Kurven bremsen!'). Umgekehrt kann bei einer zu guten Kontraktion $\|x^{(2)} - x^{(1)}\|/\|x^{(1)} - x^{(0)}\| < \frac{1}{4}q$ die Schrittweite für den nächsten Schritt vergrößert werden, $h_{k+1} := 2h_k$ ('beschleunigen auf gerader Strecke'). Der folgende Algorithmus skizziert dieses Vorgehen.

Fortsetzungsverfahren mit Schrittweitensteuerung

(5.1.14)

$\lambda := 0;\ h := h_{Start};\ z := z_0,$	
löse $G_x(z_0, 0)t = -G_\lambda(z_0, 0);$	{Tangente}
wiederhole:	
$y := z + ht;\ j := 0;$	{Prädiktor}
wiederhole:	{Newton-Iteration}
löse $G_x(y, \lambda + h)d = -G(y, \lambda + h);$	
$y := y + d;\ j := j + 1;\ c_j := \|d\|;$	
bis $(c_j \leq \epsilon)$ oder $(j \geq j_{\max});$	{z.B. $j_{\max} = 2$}
falls $(c_j > \epsilon)$ oder $(c_2/c_1 > q)$ dann	{Schritt verwerfen!}
$h := h/2$; falls $h < h_{\min}$ dann **Abbruch**;	
sonst	{Schritt akzeptieren!}
$\lambda := \lambda + h;\ z := y;$	
löse $G_x(z, \lambda)t = -G_\lambda(z, \lambda);$	{neue Tangente}
falls $c_2/c_1 < q/4$ dann $h := 2h;$	
falls $\lambda + h > 1$ dann $h := 1 - \lambda;$	
bis $\lambda \geq 1;$	

Im Prädiktor-Schritt kann übrigens die QR-Zerlegung von G_x aus dem letzten Newton-Schritt verwendet werden.

Zu einem Abbruch des Verfahrens kommt es in der Regel nur dann, wenn die Voraussetzungen von Satz 5.1.1 verletzt sind, bei stetig differenzierbarem G also i.w., wenn G_x singulär wird. Ein einfaches Beispiel dazu ist die Kreisgleichung

$$0 = G(x,\lambda) := x^2 + \lambda^2 - 1$$

im $\mathbb{R}^1$. Eine Parameterdarstellung der Lösung durch $(1,0)$ nach λ, wie $z(\lambda) = \sqrt{1-\lambda^2}$, stößt im Punkt $(x,\lambda) = (0,1)$ auf Schwierigkeiten, da z' dort nicht beschränkt ist. Parametrisiert man den Kreis aber nach der *Bogenlänge* s, z.B.,

$$(z(s),\lambda(s)) = (\cos s, \sin s),$$

hat der *Umkehrpunkt* $(0,1)$ keinerlei besondere Bedeutung mehr.

Beim allgemeinen Problem kann analog die Hervorhebung des $n+1$-ten Parameters λ fallengelassen werden dadurch, daß alle Komponenten zu einem $n+1$-Vektor y zusammengefaßt werden, $y_i = x_i \ (i \le n),\ y_{n+1} := \lambda$. Das Problem lautet dann

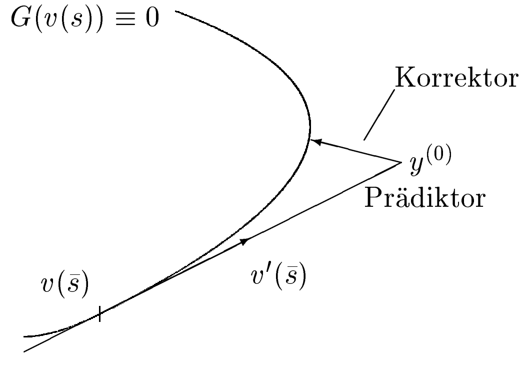
$$G(y) = 0, \quad G : D \subseteq \mathbb{R}^{n+1} \mapsto \mathbb{R}^n. \tag{5.1.15}$$

Wenn G in einer Umgebung eines Punktes $\bar{v}$ mit $G(\bar{v}) = 0$ stetig differenzierbar ist und $G'(\bar{v}) = \frac{\partial G}{\partial y}$ den *Maximalrang* n besitzt, garantiert der Satz über implizite Funktionen die lokale Existenz einer Lösungskurve $v(s),\ s \in \mathbb{R}$. Diese ist sogar stetig differenzierbar und es gilt

$$G(v(s)) \equiv 0 \quad \Rightarrow \quad G'(v(s))v'(s) \equiv 0. \tag{5.1.16}$$

Die Ableitung und Tangentenrichtung $v' \in \mathbb{R}^{n+1}$ ist somit ein Element des eindimensionalen Nullraums von $G' : \mathbb{R}^{n+1} \mapsto \mathbb{R}^n$. Wenn insbesondere als Kurvenparameter die Bogenlänge s gewählt wird mit $\|\partial v/\partial s\|_2 \equiv 1$, ist v' durch (5.1.16) bis auf das Vorzeichen (d.h. die Kurvenrichtung) eindeutig bestimmt. In Verallgemeinerung des Verfolgungsverfahrens (5.1.7), (5.1.8) kann also mit Hilfe von (5.1.16) ein *Prädiktor*-Schritt in Richtung der Kurventangente durchgeführt werden.

In Verallgemeinerung von (5.1.8) sollte der *Korrektor* idealerweise zum nächstgelegenen Punkt, $v(\hat{s})$, auf der Kurve führen. Die Korrektur $d^{(0)} = y^{(1)} - y^{(0)}$ des durch den Prädiktor berechneten Startvektors $y^{(0)}$ sollte daher senkrecht zu $v'(\hat{s})$ sein, d.h. $d^{(0)} \in N\Big(G'(v(\hat{s}))\Big)^{\perp}$. Die Gleichung für die Newtonkorrektur, $G'(y^{(0)})d^{(0)} = -G(y^{(0)})$ ist ein *unterbestimmtes* Gleichungssystem. Ändert man jedoch die erwähnte Orthogonalitätsbedingung etwas ab, so wird durch



$$G'(y^{(0)})d^{(0)} = -G(y^{(0)}), \quad d^{(0)} \in N\Big(G'(y^{(0)})\Big)^{\perp}$$

die eindeutige *Kleinste-Quadrate-Lösung* (vgl. Def. 1.1.1) definiert,

$$d^{(0)} = -G'(y^{(0)})^{+} G(y^{(0)}),$$

$(G')^+$ ist dabei die *verallgemeinerte Inverse*. So ergibt sich das

Prädiktor-Korrektor-Verfahren für (5.1.15):

$$\text{Prädiktor-Schritt:} \quad \begin{cases} s_k := s_{k-1} + h_k, \\ y^{(0)}(s_k) := v(s_{k-1}) + h_k t_{k-1}, \quad \text{mit} \\ G'\Big(v(s_{k-1})\Big) t_{k-1} = 0, \ \|t_{k-1}\|_2 = 1, \end{cases} \tag{5.1.17}$$

$$\text{Korrektor-Schritt:} \quad \begin{cases} y^{(j+1)}(s_k) = y^{(j)}(s_k) - G'\Big(y^{(j)}(s_k)\Big)^{+} G(y^{(j)}(s_k)), \\ j = 0, 1, \ldots \ . \end{cases} \tag{5.1.18}$$

Die Richtung der Tangente t_k in (5.1.17) wird am einfachsten durch $t_{k-1}^{\mathsf{T}} t_{k-2} > 0$ festgelegt. Zur Berechnung der Lösungen in (5.1.17) und (5.1.18) dient die *QR-Zerlegung* (vgl. Satz 1.3.3) der Matrix

$$(G')^{\mathsf{T}} = QR = Q \begin{pmatrix} R_0 \\ 0 \end{pmatrix}, \quad Q \in \mathbb{R}^{(n+1)\times(n+1)}, \ R_0 \in \mathbb{R}^{n \times n},$$

wobei R_0 eine reguläre obere Dreieckmatrix ist. Die Tangente t ist dann nämlich gerade der $n+1$-te Spaltenvektor von Q, denn

$$0 = G't = (R_0^{\mathsf{T}}, 0) \underbrace{Q^{\mathsf{T}} t}_{=:\, r}, \ r \in \mathbb{R}^{n+1} \quad \Longleftrightarrow \quad r_i = \pm \delta_{i,n+1} \ \Rightarrow \ t = \pm Q e^{(n+1)}.$$

Die Kleinste-Quadrate-Lösung zu $G'd = -G$ ergibt sich ebenfalls aus dieser QR-Zerlegung $d = -QE_{n,n+1}^{\mathsf{T}}(R_0^{\mathsf{T}})^{-1}G$, vgl. (1.3.11 ff). Auch hier kann i.a. (außer im Anfangspunkt) für den Prädiktorschritt (5.1.17) die QR-Zerlegung des letzten Korrektorschritts zu Stelle s_{k-1} benutzt werden.

Bei einer Schrittweitensteuerung wie in (5.1.14) für das Verfahren (5.1.17),(5.1.18) ist es zweckmäßig, außer der anfänglichen Kontraktion $\|y^{(2)} - y^{(1)}\| / \|y^{(1)} - y^{(0)}\| \overset{!}{\le} q < 1$ auch den Winkel zwischen aufeinanderfolgenden Tangenten zu überwachen, $t_{k-1}^{\mathsf{T}} t_{k-2} \overset{!}{\ge} \tau > 0$.

Beispiel 5.1.3 Ein beliebtes Testbeispiel, dessen Schwierigkeit mit wachsender Dimension größer wird, ist

$$F(x) = 0, \quad F_i(x) := x_i - \exp\cos(i \sum_{j=1}^{n} x_j), \ i = 1, \ldots, n.$$

Mit $G(x, \lambda) = \lambda F(x) + (1 - \lambda)x$ ergibt sich dafür die Einbettung

$$G(x, \lambda) = 0, \quad G_i(x, \lambda) := x_i - \lambda \exp\cos(i \sum_{j=1}^{n} x_i), \ i = 1, \ldots, n,$$

mit dem Startwert $z(0) = 0$. Die Bogenlängen der Lösungskurven wachsen mit der Dimension (stark) an von ca. $1.6\,(n = 2)$ auf ca. $60\,(n = 9)$.

Index