Numerik IIA

Endlichdimensionale Probleme

Bernhard Schmitt

Wintersemester 2007/08

Inhaltsverzeichnis

1 Eigenwertprobleme von Matrizen

Der erste Teil dieser Vorlesung beschäftigt sich mit der Berechnung der Eigenwerte und -Vektoren von Matrizen. Diese Daten haben bekannterweise deshalb eine grundlegende Bedeutung, weil die Wirkungsweise linearer Abbildungen in der Eigenvektorbasis besonders einfach aussieht. Insbesondere dominieren bei wiederholter Anwendung einer Abbildung (Matrix) $A \in \mathbb{R}^{n\times n}$ die Eigenvektoren zu bestimmten Eigenwerten das Erscheinungsbild. Daher vermittelt die Betrachtung der Eigensysteme u.a. in folgenden Anwendungen grundlegende Einsichten:

- *Schwingungsprobleme:* Schon aus der Schulphysik ist bekannt, dass lineare, schwingende Systeme ausgewählte Eigenfrequenzen und zugehörige Eigenmoden besitzen (Grundton und Obertöne), mit denen sie auf äußere Anregungen reagieren. Denn für das lineare System von Differentialgleichungen $y''(t) = Ay(t)$ mit $A \in \mathbb{R}^{n\times n}$ führt der Ansatz

$$y(t) = e^{\lambda t}y_0 \;\longrightarrow\; y''(t) = \lambda^2 e^{\lambda t}y_0 \stackrel{!}{=} e^{\lambda t}Ay_0 \;\Rightarrow\; Ay_0 = \lambda^2 y_0$$

  auf ein Eigenwertproblem. Wenn A nur negative reelle Eigenwerte besitzt, ist $\lambda^2 < 0$ und daher λ imaginär. Also setzt sich $e^{\lambda t}$ aus Sinus- und Cosinus-Schwingungen zusammen.

- *Dynamische Systeme, Differentialgleichungen:* Bei linearen Systemen von Dgln erster Ordnung $y'(t) = Ay(t)+g(t)$ führt beim homogenen System der Ansatz von oben auf das EWP $Ay_0 = \lambda y_0$. Anhand der Realteile der Eigenwerte λ_i entscheidet sich, ob mit t wachsende oder (nur) fallende Lösungen auftreten. Aber auch für das Erscheinungsbild nichtlinearer Prozesse

$$y'(t) = f(y(t))$$

  in der Nähe eines Gleichgewichts $\bar{y}$ mit $f(\bar{y}) = 0$ gilt diese Aussage mit den Eigenwerten von $A = f'(\bar{y})$.

- *Iterationen, diskrete dynamische Systeme:* In der Numerik I wurden Iterationsverfahren für lineare Gleichungssysteme der Form

$$y^{(k+1)} = Ay^{(k)} + b, \quad k = 0, 1, \ldots$$

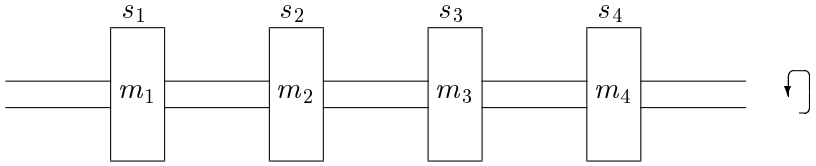
  betrachtet. Man kann die Vorschrift aber auch als diskretes, dynamisches System ansehen, wo die Zeit nur an diskreten Punkten $k \in \mathbb{N}$ (Tage, Jahre) betrachtet wird. Beim homogenen Problem führt nun der Ansatz

$$y^{(k)} = \lambda^k y_0 \;\longrightarrow\; y^{(k+1)} = \lambda y^{(k)} \stackrel{!}{=} Ay^{(k)} \Rightarrow Ay_0 = \lambda y_0$$

  auf das gleiche Eigenwertproblem wie zuvor. Allerdings entscheidet jetzt der Betrag $|\lambda_i|$ der Eigenwerte darüber ob Lösungen wachsen oder fallen. Bei der Iteration war tatsächlich die Bedingung $\varrho(A) < 1$ hinreichend für Konvergenz. Auch für nichtlineare diskrete Systeme $y^{(k+1)} := f(y^{(k)})$, $k \in \mathbb{N}_0$, liefern die Eigenwerte der Ableitung $A = f'(\bar{y})$ in einem Fixpunkt $\bar{y} = f(\bar{y})$ die Information über die Dynamik in dessen Umgebung.

In diesen Anwendungen zeigt sich auch, dass man oft nicht die Kenntnis aller Eigenwerte benötigt, sondern nur die von ausgewählten, etwa mit kleinstem oder größtem Betrag oder Realteil. Zur Veranschaulichung wird folgendes Beispiel diskutiert:

Generatorturbinen in Kraftwerken 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 Grenz-Drehzahl $\omega_{\max}$, welche nicht erreicht werden darf, 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:

$$\begin{cases} x_1 &= \omega^2(b_{11}m_1x_1 + \ldots + b_{1n}m_nx_n), \\ & \ldots \\ x_n &= \omega^2(b_{n1}m_1x_1 + \ldots + b_{nn}m_nx_n), \end{cases}$$

d.h.,

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

Dies ist offensichtlich ein Eigenwertproblem für den Eigenwert $\lambda = 1/\omega^2$. Die Grenzdrehzahl gehört somit zum größten (positiven!) Eigenwertvon A.

1.1 Grundlagen, Matrix-Normalformen

Da man auch bei allgemeinen, rellen Matrizen mit komplexen Eigenwerten zu rechnen hat, wird gleich der Fall komplexer Matrizen betrachtet.

Definition 1.1.1 *Gegeben sei eine Matrix $A \in \mathbb{C}^{n\times n}$. Ein Wert $\lambda \in \mathbb{C}$ heißt* Eigenwert *von A, wenn ein Vektor $x \in \mathbb{C}^n$ existiert mit*

$$Ax = \lambda x, \qquad x \neq 0. \tag{1.1.1}$$

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

Es gibt aber zwei Begriffe für die Vielfachheit. Denn die Eigenwerte von A sind auch Nullstellen des *charakteristischen Polynoms*

$$\begin{aligned} p(z) &= \det(zI - A) = z^n + b_{n-1}z^{n-1} + \ldots + b_0 \\ &= (z-\lambda_1)^{\alpha_1}(z-\lambda_2)^{\alpha_2}\cdots(z-\lambda_k)^{\alpha_k}. \end{aligned} \tag{1.1.2}$$

Die Vielfachheit α_j der Nullstelle λ_j heißt *algebraische Vielfachheit* von λ_j. Außer bei Vielfachheit $\alpha_1 = 1$ kann die geometrische Vielfachheit γ_j kleiner sein als die algebraische, $1 \le \gamma_j \le \alpha_j$:

Beispiel 1.1.2 Es sei

$$C_m(\lambda) := \begin{pmatrix} \lambda & 1 & & 0 \\ & \lambda & 1 & \\ & & \ddots & \ddots \\ & & & \lambda \end{pmatrix} \in \mathbb{C}^{m\times m}. \tag{1.1.3}$$

Das charakteristische Polynom ist $p(z) = (\lambda - z)^m$, λ ist also algebraisch m-facher 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: $\gamma = 1 \le \alpha = m$.

Um bei Abweichung von geometrischer und algebraischer Vielfachheit die Eigenvektoren zu einer Basis des Gesamtraums ergänzen zu können, werden weitere Vektoren benötigt.

Definition 1.1.3 *Für $\lambda \in \mathbb{C}$, $k \in \mathbb{N}$ und $x \in \mathbb{C}^n, x \neq 0$, gelte*

$$(A - \lambda I)^k x = 0, \quad \text{aber } (A-\lambda I)^{k-1}x \neq 0. \tag{1.1.4}$$

Dann heißt x Hauptvektor vom Grad k zum Eigenwert λ.

Satz 1.1.4 *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*

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

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 $T \in \mathbb{C}^{n\times n}$,

$$A \;\mapsto\; TAT^{-1} =: B. \tag{1.1.5}$$

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 solche Ähnlichkeitstransformationen, die eine Matrix auf einfache Gestalt bringen, an der wesentliche Eigenschaften in Bezug auf die einleitenden Fragestellungen ablesbar sind.

Satz 1.1.5 (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*

$$J = \begin{pmatrix} C_{m_1}(\lambda_1) & & 0 \\ & \ddots & \\ 0 & & C_{m_k}(\lambda_k) \end{pmatrix} = \begin{pmatrix} \lambda_1 & 1 & & & & & \\ & \ddots & \ddots & & & & \\ & & \lambda_1 & 1 & & & \\ & & & \lambda_1 & 0 & & \\ & & & & \lambda_2 & 1 & \\ & & & & & \ddots & \ddots \end{pmatrix}, \tag{1.1.6}$$

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 Links- und Rechts-Eigen-/Hauptvektoren zueinander orthogonal, ${y^{(i)}}^* x^{(j)} = \delta_{ij}$.

Dabei ist $y^* = (\bar{y})^\mathsf{T}$. Die j-te Spalte der Identität $AX = XJ$ entspricht dabei (1.1.1) bei einem EV $x^{(j)}$, aber $Ax^{(j)} = \lambda x^{(j)} + x^{(j-1)}$ wenn $x^{(j)}$ Hauptvektor ist. Eine Matrix, bei der alle Jordanblöcke die Größe eins haben (J hat in der Nebendiagonale keine Einsen), heißt *diagonalisierbar*. Sie besitzt also eine Basis aus n Eigenvektoren. Jede Matrix mit nur algebraisch einfachen Eigenwerten ist offensichtlich diagonalisierbar.

Trotz ihrer theoretischen Bedeutung legt man die Jordan-Normalform kaum bei numerischen Rechnungen zugrunde, da sie sehr empfindlich auf Störungen der Matrix reagiert. Bei

$$C_2(1) = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \quad \text{und} \quad \tilde{C} = \begin{pmatrix} 1+\epsilon & 1 \\ 0 & 1 \end{pmatrix} = X \begin{pmatrix} 1+\epsilon & \mathbf{0} \\ 0 & 1 \end{pmatrix} X^{-1}$$

ist, z.B. die erste Matrix $C_2(1)$ in Jordan-Normalform. Schon bei einer beliebig kleinen Änderung $\epsilon \neq 0$ ist die gestörte Matrix $\tilde{C}$ 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 1.1.6 (Schur-Normalform) *Jede Matrix $A \in \mathbb{C}^{n\times n}$ kann durch eine unitäre Matrix U, $U^* = U^{-1}$, ähnlich auf obere Dreieckgestalt transformiert werden,*

$$A = USU^*, \quad S = \begin{pmatrix} \lambda_1 & s_{12} & \dots & s_{1n} \\ & \lambda_2 & & s_{2n} \\ & & \ddots & \vdots \\ & & & \lambda_n \end{pmatrix}. \tag{1.1.7}$$

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

Die Reihenfolge der Eigenwerte ist zwar beliebig, im folgenden wird zur Standardisierung der Bezeichnungen aber oft eine Numerierung nach der Größe angenommen, etwa $|\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_n|$.

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 1.1.7 *Eine Matrix A heißt* normal, *wenn gilt*

$$A^*A = AA^*. \tag{1.1.8}$$

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 1.1.8 *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 wegen der Isometrie unitärer Abbildungen für die Schur-Normalform

$$\|A\|_2 = \|USU^*\|_2 = \|S\|_2. \tag{1.1.9}$$

Für normale Matrizen ist daher $\|A\|_2 = \varrho(A) = \max_{i=1}^n |\lambda_i|$. Für allgemeine Matrizen sei an Satz 4.2.1 aus der Numerik I erinnert:

Satz 1.1.9 *a) Für jede Matrixnorm $\|\cdot\|$ gilt $\rho(A) \le \|A\|$.*

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

$$\|A\|_M \le \rho(A) + \varepsilon. \tag{1.1.10}$$

c) Für diagonalisierbare Matrizen A kann in (1.1.10) $\varepsilon = 0$ gewählt werden.

Beispiel 1.1.10 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 $\lambda \neq 0$ die Norm nicht durch $|\lambda|^k$ beschränkt ist, mit $x = e^{(2)}$ etwa gilt

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

In der Schranke aus Satz 1.1.9 ist also tatsächlich $\varepsilon > 0$ erforderlich.

1.2 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 1.2.1 *Die Matrix $A = (a_{ij}) \in \mathbb{C}^{n\times n}$ ist eine*

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

Ausführlich:

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

Hessenberg-Form Tridiagonal-Gestalt

Zur Umformung auf diese einfacheren Gestalten eignen sich die Householder-Spiegelungen $H = I - 2uu^*$, $\|u\|_2 = 1$, aus der Numerik I. Die Konstruktion dieser Spiegelungen zur Elimination der Elemente $x_2, \dots, x_n$ einer Matrixspalte x wird hier noch einmal zusammengefaßt.

Satz 1.2.2 *Der Vektor $0 \neq x \in \mathbb{C}^m$ mit $x_1 = \sigma|x_1|$, $|\sigma| = 1$, wird durch die unitäre Matrix*

$$H = I - \frac{vv^*}{\|x\|_2(|x_1| + \|x\|_2)}, \quad v := x + \sigma\|x\|_2 e^{(1)}, \tag{1.2.1}$$

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

Der Unterschied zum Vorgehen bei der QR-Zerlegung ist die Einschränkung, dass auf die Matrix A ausschließlich Ähnlichkeitstransformationen anzuwenden sind.

Satz 1.2.3 *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 $n-2$ Householder-Spiegelungen (1.2.1) dargestellt werden, $U^* = H^{(n-2)} \cdots H^{(1)}$. Der Rechenaufwand für diese Transformation beträgt $\frac{10}{3}n^3$ Operationen. Wenn A hermitesch 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: Beginnend mit $A^{(1)} := A$ werden Umformungen durch Ähnlichkeitstransformation

$$A^{(j+1)} =: H^{(j)}A^{(j)}H^{(j)}, \quad H^{(j)} = I - 2u^{(j)}u^{(j)^*} \tag{1.2.2}$$

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

$$A^{(j)} = \begin{pmatrix} & & \\ & \ddots & \\ & a_{j+1,j}^{(j)} \quad * \quad \cdots \quad * & \\ & \vdots \quad \vdots \quad \quad \vdots & \\ & a_{nj}^{(j)} \quad * \quad \cdots \quad a_{nn}^{(j)} & \end{pmatrix} \begin{matrix} \left.\vphantom{\begin{matrix}a\\a\\a\end{matrix}}\right\} j \\ \left.\vphantom{\begin{matrix}a\\a\\a\end{matrix}}\right\} n-j \end{matrix} \in \mathbb{C}^{n\times n}$$

Nach Satz 1.2.2 wird eine Spiegelung $H^{(j)}$ bestimmt mit $u^{(j)} = (0, \ldots, 0, u_{j+1}^{(j)}, \ldots, u_n^{(j)})^\mathsf{T}$, d.h., mit der Struktur

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

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

Aufwand: Wie im Satz über die QR-Zerlegung in der Numerik I bestimmen die Matrix-Umformungen den Aufwand. Die Rechnungen werden natürlich nur mit nichttrivialen Elementen

durchgeführt, außerdem wird dabei die Rang-1-Struktur $2u^{(j)}u^{(j)^*} = v^{(j)}v^{(j)^*}/d_j$ (vgl. Satz 1.2.2) ausgenutzt. Damit benötigt die erste Umformung,

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

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

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

$4n(n-j)$ Operationen, da $\tilde{A}^{(j)}v^{(j)}$ 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: $A = A^* \Rightarrow B^* = U^*A^*U = U^*AU = B$. ∎

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 besprochenen Standard-Verfahren (QR-Iteration).

1.3 Das unsymmetrische Eigenwertproblem

Die hier besprochenen 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 werden aber nur bestimmte Eigenwerte benötigt, 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

$$\left.\begin{array}{rcll} q^{(k-1)} & := & z^{(k-1)}/|z_{i_k}^{(k-1)}|, \quad \text{wo } |z_{i_k}^{(k-1)}| = \|z^{(k-1)}\|_\infty & \text{(Normierung)} \\ z^{(k)} & := & Aq^{(k-1)}, & \text{(Iteration)} \\ \lambda^{(k)} & := & z_{i_k}^{(k)}/q_{i_k}^{(k-1)} & \end{array}\right\} k = 1, 2, \ldots \quad (1.3.1)$$

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

$$q^{(k)} = v^{(k)}/\|v^{(k)}\|_\infty, \quad v^{(k)} := A^k z^{(0)}. \quad (1.3.2)$$

b) Mit dem Iterationsverfahren $x^{(k+1)} := Ax^{(k)} + b$ (vgl. Numerik I, §4.5) kann man für $\varrho(A) < 1$ das Lineare Gleichungssystem $(I - A)\hat{x} = b$ lösen. Hier hat der (unbekannte) Fehler $v^{(k)} := x^{(k)} - \hat{x}$ eine identische Rekursion

$$v^{(k)} = x^{(k)} - \hat{x} = Ax^{(k-1)} + b - (A\hat{x}^{(k)} + b) = Av^{(k-1)} = \ldots = A^k v^{(0)}.$$

Dies gilt auch für den Defekt $d^{(k)} := (I - A)x^{(k)} - b = (I - A)v^{(k)} = Ad^{(k-1)}$, der in der Regel sowieso berechnet wird (Abbruchkriterium). Daher ist der nächste Satz auch in diesem Zusammenhang von grundlegender Bedeutung (vgl. §2).

c) Die Bestimmung von $\lambda^{(k)}$ in (1.3.1) erfolgt auf eine etwas komplizierte Weise, da man hierbei gleiche Komponenten von $z^{(k)}$ und $q^{(k-1)}$ vergleichen und Division durch Null vermeiden muß. 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	<u>3.33</u>	0.8		<u>3.898</u>	3.455	<u>3.940</u>	3.682	<u>3.966</u>
$z^{(k)}$	1	1.66	2.1		3.595	3.688	3.762	3.819	3.862
	<u>-3</u>	-0.33	<u>-3.5</u>		-3.291	<u>-3.922</u>	-3.583	<u>-3.955</u>	-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 $|\lambda^{(k)} - \lambda_1|/|\lambda^{(k-1)} - \lambda_1| \cong \frac{3}{4} = |\lambda_2/\lambda_1|$.

Für die Konvergenz entscheidend ist, dass der größte Eigenwert betragsmäßig von den andern separiert ist. Da Eigenvektoren nur bis auf Konstanten bestimmt sind, müssen in der folgenden Konvergenzaussage die Vektoren $q^{(k)}$ und $x^{(1)}$ gleichartig normiert werden.

Satz 1.3.1 *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., es gelte $z^{(0)} = \alpha_1 x^{(1)} + \ldots$ 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 (1.3.2) die Darstellung

$$v^{(k)} = A^k z^{(0)} = \sum_{j=1}^{n} \lambda_j^k \alpha_j x^{(j)} = \lambda_1^k \alpha_1 \Big(x^{(1)} + \sum_{j=2}^{n} \frac{\alpha_j}{\alpha_1} \Big(\frac{\lambda_j}{\lambda_1}\Big)^k x^{(j)} \Big). \tag{1.3.3}$$

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 $|q_{i_k}^{(k-1)}| = 1$. Wegen $(\lambda_j/\lambda_1)^k \to 0 \forall j > 1$ 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 (1.3.3). 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

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

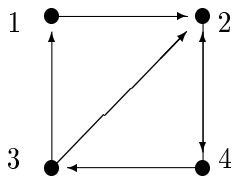
mit Satz 1.1.9 abgeschätzt werden, $\|M\| \leq \rho(M) + \epsilon = |\lambda_2| + \epsilon$. Dann tritt in der Aussage tatsächlich ein $\varepsilon > 0$ auf. ∎

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 $\alpha_1 = y^{(1)^*} z^{(0)} \neq 0$ kann oft aus theoretischen Gründen garantiert werden wie im nächsten Beispiel. Aber selbst ohne Vorkenntnis wird durch Rundungsfehler i.a. ein Beitrag mit $\alpha_1 \neq 0$ eingeschleppt.

Beispiel 1.3.2 Die Suchmaschine Google behandelt Details ihres Suchalgorithmus als Geschäftsgeheimnis um Manipulationen zu erschweren ("Link-Farmen"). Ein wesentlicher Bestandteil des Algorithmus ist dabei aber das patentierte "Page-Ranking". Dazu betrachtet man die WWW-Seiten (zu einer Anfrage) als Knoten eines Graphen und die Links zwischen Seiten als gerichtete Kanten. Seiten sind dann wichtig, wenn viele Links von anderen wichtigen Seiten hinzeigen. Die Bedeutung einer Seite wird also implizit durch die unbekannte Bedeutung der auf sie zeigenden Seiten bestimmt. Gibt man jeder Seite ein Gewicht x_j (Summe auf eins normiert $\mathbb{1}^\mathsf{T} x = 1$), dann erhält man die Page-Ranks aller n Knoten aus dem System

$$x_i = \sum_{j=1}^{n} g_{ij} x_j, \ i = 1, \ldots, n, \quad g_{ij} := \begin{cases} \frac{1}{m_j} & \exists \text{ Link } j \to i, \\ 0 & \text{kein Link } j \to i. \end{cases}$$

Dabei werden die Einträge in G mit $1/m_j$ gewichtet, wobei m_j die Anzahl der von Seite j ausgehenden Links ist. Dann hat die nichtnegative Matrix $G \geq 0$ Spaltensummen eins, $\mathbb{1}^\mathsf{T} G = \mathbb{1}^\mathsf{T}$, und das Problem ist tatsächlich ein Eigenwertproblem, denn x ist der Rechts-Eigenvektorvon G zum Eigenwert 1. Beim Einsatz wird zur Vermeidung betragsgleicher Eigenwerte allerdings mit einem Faktor $d \in (0,1)$ die modifizierte positive Matrix $A := \frac{1-d}{n} \mathbb{1}\mathbb{1}^\mathsf{T} + dG$ verwendet, die ebenfalls den gleichen Links-Eigenvektor besitzt, $\mathbb{1}^\mathsf{T} A = \mathbb{1}^\mathsf{T}$. Das Diagramm zeigt ein kleines Netz und die zugehörige (modifizierte Adjazenz-) Matrix G



$$G := \begin{pmatrix} 0 & 0 & \frac{1}{2} & 0 \\ 1 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & \frac{1}{2} \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Die Funktion des Algorithmus beruht auf der Perron-Frobenius-Theorie positiver Matrizen $A > 0$. Für diese ist $\lambda_1 = \varrho(A) > 0$ ein einfacher dominanter Eigenwert und der zugehörige Eigenvektor x positiv, also mit der Vektor-Iteration und einem positiven Startvektor gut berechenbar. Im Beispiel ist $\frac{1}{11}(1,4,2,4)^\mathsf{T}$ der normierte Eigenvektor von G und gibt den Knoten 2 und 4 den höchsten Rang, für $d = 0.85$ bekommt man bei A eine realistischere Rangwerte $x = (0.124, 0.357, 0.184, 0.335)^\mathsf{T}$ mit maximalem Rang x_2.

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 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 (1.3.1) geringfügig 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)

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

$$\left.\begin{array}{rcll} q^{(k-1)} & := & z^{(k-1)}/|z_{i_k}^{(k-1)}|, & \text{wo } |z_{i_k}^{(k-1)}| = \|z^{(k-1)}\|_\infty \\ z^{(k)} & := & (A-\lambda I)^{-1} q^{(k-1)}, & \\ \lambda^{(k)} & := & \lambda + q_{i_{k+1}}^{(k-1)}/z_{i_{k+1}}^{(k)} & \end{array}\right\} k = 1, 2, \ldots \qquad (1.3.4)$$

Für die Konvergenz dieses Verfahrens gilt

Satz 1.3.3 *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 (1.3.4) 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. Dies ist bei dünnbesetzten Matrizen effizient machbar, bei vollbesetzten sollte die Matrix aber vorher in Hessenbergform gebracht werden, da jede Zerlegung dann nur einen $\mathcal{O}(n^2)$-Aufwand erfordern. Diese Variante der Inversen Iteration hat in gutartigen Situationen hervorragende Eigenschaften (vgl. symmetrisches Eigenwertproblem). Wichtige Einsatzbereiche der Inversen Iteration sind sehr große Probleme, bei denen die QR-Iteration nicht praktikabel ist und die Berechnung einzelner Eigenvektoren, wenn Eigenwert-Approximationen mit einem anderen Verfahren

(z.B. QR-Verfahren, Bisektion) geliefert wurden. In schwierigen Fällen mit eng benachbarten Eigenwerten kann es aber vorkommen, dass nur der erste Schritt eine Verbesserung bringt.

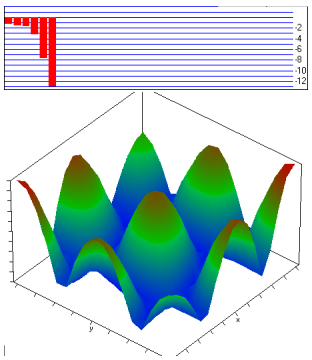
Beispiel 1.3.4 Die Schwingung eines Trommelfells ("Membran"), das die Form eines Gebietes $\Omega \subset \mathbb{R}^2$ hat, wird durch die Schwingungsgleichung $u_{tt}(t,x,y) = \Delta u(t,x,y),\ (x,y) \in \Omega$, beschrieben. Dabei ist $\Delta u = u_{xx} + u_{yy}$ der Laplace-Operator. Der Ansatz aus §1 führt auf das Eigenwertproblem

$$\Delta u(x,y) = \lambda^2 u(x,y),\ (x,y) \in \Omega,\ (x,y) = 0 \text{ auf dem Rand von } \Omega.$$

Approximiert man dieses Problem durch Differenzen (vgl. Numerik 1, Beisp. 4.3.5) oder andere Verfahren, ist bei der Wielandt-Iteration (1.3.4) in jedem Schritt ein Gleichungssystem mit dünnbesetzter (z.B. Band-) Matrix zu lösen.

Die Graphiken zeigen die schnelle (quadratische) Konvergenz der Iteration (Differenzdiagramm oben) und ein Bild der zugehörigen Eigenfunktion (Schwingungsform) zum Eigenwert $\cong 1.048$. Die Matrix hat Dimension $n = 289$.

0	1.0
1	1.0587700503394779
2	1.0273143219123218
3	1.0473673848444942
4	1.0480047826707237
5	1.0480047613117915
6	1.0480047613117927



Die moderne Sichtweise des Problemfalls bei eng benachbarten Eigenwerten ist, dass man nicht versuchen sollte, diese fehleranfällige Situation im Einzelnen aufzulösen. Man versucht statt dessen, größere, *invariante Unterräume* genau zu bestimmen. Zur Behandlung mehrfacher, oder eng zusammenliegender Eigenwerte konvergiert auch die einfache Vektor-Iteration schlecht. Die folgende Verallgemeinerung führt Schritte gleichzeitig mit mehreren Vektoren (d.h., Unterraum-Basen) durch. 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

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

verallgemeinert werden. In der Numerik I war dabei Q als quadratische und R als $n \times p$-Matrix in Dreieckform behandelt worden. Da aber die letzten $n - p$ Zeilen von R verschwinden, spielen die letzten Spalten von Q keine Rolle und es ist hier ($p \ll n$) ökonomischer, R als quadratische $p \times p$-Matrix zu betrachten und nur den vorderen, orthogonalen $n \times p$-Teil zu verwenden. Zur Vermeidung von Doppelbezeichnungen (vgl. QR-Iteration) wird dieser vordere Teil im 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

$$\left.\begin{array}{rcll} Z^{(k)} & := & AU^{(k-1)} & \\ U^{(k)}R^{(k)} & := & Z^{(k)} & \text{(QR-Zerlegung)} \end{array}\right\} \; k = 1, 2, \ldots \tag{1.3.5}$$

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 (1.3.5) 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. Denn wenn A dünn besetzt ist, lassen sich bei der Multiplikation AU Produkte mit $a_{ij} = 0$ einsparen (durch Verwaltung der nichttrivialen Einträge). Dagegen zerstört das im folgenden besprochene QR-Verfahren, das eng mit (1.3.5) zusammenhängt, eine eventuell vorhandene Struktur von A. Daher haben beide Verfahren in Bezug auf die praktische Durchführung durchaus unterschiedliche Eigenschaften. In Bezug auf Konvergenzfragen sind sie allerdings äquivalent aufgrund der folgenden Überlegungen.

Bemerkung: Alle bei der orthogonalen Iteration (1.3.5) auftretenden Matrizen können für beliebige $p < n$ als vordere Spalten quadratischer $n \times n$-Matrizen betrachtet werden. Denn

— bei der Produktbildung AU sind die einzelnen Spalten völlig unabhängig voneinander,

— 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 $Ue^{(j)} = H^{(1)} \cdots H^{(n-1)}e^{(j)} = H^{(1)} \cdots H^{(j)}e^{(j)}$.

Daher kann man bei Konvergenzaussagen nur den Fall $p = n$ betrachten. Dann sind diese Aussagen aber für die orthogonale Iteration und das später folgende QR-Verfahren völlig analog und werden daher dort behandelt (Satz 1.3.8). Zum QR-Verfahren kommt man zwangsläufig, wenn man sich (im Fall $p = n$) die Matrizen

$$S^{(k)} := U^{(k)^*} A U^{(k)} \tag{1.3.6}$$

ansieht, 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 (1.3.5) $R^{(k)} = U^{(k)^*} A U^{(k-1)}$ und daher einerseits

$$S^{(k)} = U^{(k)^*} A U^{(k-1)} U^{(k-1)^*} U^{(k)} = R^{(k)} \underbrace{U^{(k-1)^*} U^{(k)}}_{=: Q^{(k)}}, \tag{1.3.7}$$

andererseits aber auch

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

Also entsteht $S^{(k)}$ auch direkt aus einer QR-Zerlegung von $S^{(k-1)}$ und anschließender Multiplikation der beiden Faktoren in *umgekehrter Reihenfolge*. Dies ist ein neues Verfahren:

Das QR-Verfahren

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

$$\left.\begin{array}{rcll} Q^{(k)} R^{(k)} & := & S^{(k-1)} & \text{(QR-Zerlegung)} \\ S^{(k)} & := & R^{(k)} Q^{(k)} & \end{array}\right\} \; k = 1, 2, \ldots \tag{1.3.8}$$

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

$$s_{ij}^{(k)} = \mathcal{O}\Big(\Big|\frac{\lambda_i}{\lambda_j}\Big|^k\Big), \quad k \to \infty, \; j < i, \tag{1.3.9}$$

bei der Anordnung $|\lambda_1| \geq \ldots \geq |\lambda_n|$, die später in Satz 1.3.8 gezeigt wird. Denn in der einfachen Form ist das Verfahren (1.3.8) noch zu aufwendig, da jeder Schritt $\mathcal{O}(n^3)$ Operationen kostet (Numerik I). Die Effizienz des Verfahrens kann durch die drei folgenden Maßnahmen dramatisch gesteigert werden.

- **Satz 1.3.5** *Besitzt $S^{(0)} = A$ Hessenbergform, dann gilt dies auch für alle Iterierten $S^{(k)}$. In diesem Fall beträgt der Aufwand für einen Schritt (1.3.8) nur noch $8n^2$ Operationen.*

  Beweis Die für die QR-Zerlegung benutzten Spiegelungen $H^{(j)} = I - 2u^{(j)}u^{(j)^*}$ verwenden hier 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 1.3.6** *Bei Spektralverschiebung*

$$\left.\begin{array}{rcl} Q^{(k)} R^{(k)} & := & S^{(k-1)} - \lambda I \\ S^{(k)} & := & R^{(k)} Q^{(k)} + \lambda I \end{array}\right\} \; k = 1, 2, \ldots \tag{1.3.10}$$

  verändert sich mit der neuen Anordnung $|\lambda_1 - \lambda| \geq \ldots \geq |\lambda_n - \lambda|$ die Konvergenzaussage (1.3.9) zu

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

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

$$S^{(k)} = R^{(k)}Q^{(k)} + \lambda I = Q^{(k)^*}(S^{(k-1)} - \lambda I)Q^{(k)} + \lambda I = Q^{(k)^*} S^{(k-1)} Q^{(k)}, \tag{1.3.11}$$

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

Formel (1.3.11) zeigt übrigens ebenfalls, dass $S^{(k)}$ und $S^{(k-1)}$ ähnliche Matrizen sind.

- **Satz 1.3.7** *Ist S Hessenberg-Matrix und $s_{n,n-1} = 0$, dann ist s_{nn} ein Eigenwert von S. Die restlichen Eigenwerte sind die von S_1 in der Unterteilung*

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

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 $s^{(k)}_{n,n-1}$ sehr klein wird, kann daher die QR-Iteration mit der kleineren Matrix S_1 fortgesetzt werden (*Deflation*).

Zur praktischen Anwendung von Satz 1.3.6 ist eine Strategie für die Wahl der (des) Spektralparameter(s) erforderlich, um eine Verbesserung der Konvergenz zu erreichen. Oft ist die Konvergenz des letzten Diagonalelements gegen den betragskleinsten Eigenwert, $s^{(k)}_{nn} \to \lambda_n$, am schnellsten. Nach einer Anlaufphase, etwa wenn

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

gilt, läßt sich dieses Verhalten durch folgende Wahl für λ in Satz 1.3.6 unterstützen:

$$\lambda^{(k-1)} := s^{(k-1)}_{nn}. \tag{1.3.12}$$

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

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

Auch bei reeller Rechnung können die komplexen Eigenwerte dort 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 1.3.6 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 $\lambda^{(k+1)} = \overline{\lambda}^{(k)}$ verwendet werden. Dafür geeignet sind die beiden Eigenwerte des letzten 2×2-Blocks

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

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 1.3.8 *Die Eigenwerte der Matrix A seien betragsmäßig verschieden, $|\lambda_1| > \ldots > |\lambda_n|$. 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 (1.3.8) wie in (1.3.9) gegen eine obere Dreieckmatrix, insbesondere gilt für $i = 1, \ldots, n$:*

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

Mit Spektralverschiebung (1.3.12) konvergiert die QR-Iteration (1.3.10) lokal quadratisch, mit

$$|s_{n,n-1}^{(k+1)}| \le c|s_{n,n-1}^{(k)}|^2, \quad k \ge k_0,$$

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

Beweis Zur Verdeulichung des grundlegenden Beweisprinzips wird der Beweis nur unter Einschränkungen und nur für das einfache Verfahren (1.3.8) geführt. Es sei A reell und $|\lambda_n| > 0$, also A regulär. Ein wesentliches Argument im Beweis ist die weitgehende Eindeutigkeit und stetige Abhängigkeit der QR-Zerlegung. Dazu sind folgende Vorbemerkungen erforderlich:

— Die QR-Zerlegung enthält eine Vorzeichenentscheidung, vgl. Satz 1.2.2, zur Vermeidung von Rundungsfehlern. Wählt man davon abweichend in R positive Hauptdiagonalelemente (Abbildung der Spalte x auf $+\|x\|_2 e^{(1)}$ in (1.2.1)), 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.

Zum Beweis werden zwei verschiedenen QR-Zerlegungen der Matrix A^k betrachtet, welche aber durch die Eindeutigkeit dann doch miteinander identifiziert werden können.

a) Als erstes werden die Transfomationsmatrizen der QR-Iteration (1.3.8) aufmultipliziert

$$U^{(k)} := Q^{(1)}Q^{(2)}\cdots Q^{(k)},\ k > 0, \quad U^{(0)} := I, \tag{1.3.13}$$

vgl. auch (1.3.7). Mit diesen gilt wegen (1.3.11)

$$\begin{aligned} AU^{(k-1)} &= AQ^{(1)}\cdots Q^{(k-1)} = Q^{(1)}S^{(1)}Q^{(2)}\cdots Q^{(k-1)} = \ldots \\ &= Q^{(1)}\cdots Q^{(k-1)}S^{(k-1)} = U^{(k-1)}S^{(k-1)} \qquad (1.3.14) \\ &= U^{(k-1)}Q^{(k)}R^{(k)} = U^{(k)}R^{(k)}. \qquad (1.3.15) \end{aligned}$$

Dies entspricht gerade einem Schritt der orthogonalen Iteration (1.3.5). Es folgt

$$A^k = A^k U^{(0)} = A^{k-1} U^{(1)} R^{(1)} = \ldots = U^{(k)} \underbrace{R^{(k)} \cdots R^{(2)} R^{(1)}}_{=: \hat{R}^{(k)}} =: U^{(k)} \hat{R}^{(k)}. \tag{1.3.16}$$

Die Matrix $U^{(k)}$ ist also der eindeutige (s.o.) unitäre Faktor der QR-Zerlegung von A^k, die Dreieck-Faktoren wurden zu $\hat{R}^{(k)}$ zusammengefasst.

b) In der Jordan-Normalform $A = XJX^{-1}$ sei nun $UR_X = X$ die QR-Zerlegung von X (U ist die unitäre Matrix der Schur-Normalform, vgl. Beweis von Satz 1.1.6) und $L_Y R_Y = Y$ die LR-Zerlegung von $Y = X^{-1}$. Dann gilt für A^k auch

$$A^k = XJ^kX^{-1} = UR_X J^k L_Y R_Y = UR_X \underbrace{(J^k L_Y J^{-k})}_{=: I + F_k} J^k R_Y. \tag{1.3.17}$$

Hier haben R_X, R_Y schon obere Dreieckgestalt, J^k ist diagonal. Es stört die untere Dreieckmatrix $J^k L_Y J^{-k}$ mit den Elementen $l_{ij}(\lambda_i/\lambda_j)^k$. In dieser konvergieren aber die Subdiagonalen gegen null für $k \to \infty$, da $|\lambda_i/\lambda_j| \le q < 1$ ist nach Voraussetzung für $j < i$. Daher gilt in (1.3.17) tatsächlich

$$J^k L_Y J^{-k} = I + F_k \quad \text{mit} \quad \|F_k\| \le c\, q^k \to 0 \ (k \to \infty).$$

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

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

Vorzeichenmatrizen mit $\bar{D}_k D_k = I$. Aus (1.3.17) wird nun

$$A^k = UQ_k R_k J^k R_Y = (UQ_k \bar{D}_k)(D_k R_k J^k R_Y). \tag{1.3.18}$$

Dabei ist $D_k R_k J^k R_Y$ eine obere Dreieckmatrix, deren Diagonalelemente als Produkte der Diagonalen ihrer Faktoren positiv sind, und $UQ_k D_k$ ist unitär.

c) Durch Vergleich der beiden (eindeutigen!) QR-Zerlegungen (1.3.16) und (1.3.18) von A^k folgt somit

$$U^{(k)} = UQ_k \bar{D}_k \quad \text{und} \quad \hat{R}^{(k)} = D_k R_k J^k R_Y. \tag{1.3.19}$$

Wegen $Q_k \to I$ konvergiert daher $U^{(k)} D_k \to U$ ($k \to \infty$), die Konvergenz von $U^{(k)}$ gegen U aus der Schur-Normalform tritt also bis auf Vorzeichen der Spalten ein. Diese Vorzeichen sind aber unwichtig, denn für die Matrizen $S^{(k)}$ gilt nach (1.3.14) und (1.3.19)

$$S^{(k)} \ = \ U^{(k)*} A U^{(k)} = D_k \underbrace{Q_k^*}_{\to I} \underbrace{U^* A U}_{= S} \underbrace{Q_k}_{\to I} \bar{D}_k.$$

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

Beispiel 1.3.9 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 (1.3.12). Die Tabelle zeigt die Diagonale der Matrizen $S^{(k)}$ und das Subdiagonalelement $s^{(k)}_{m,m-1}$, wobei $m \leq n$ die aktuelle Dimension der Restmatrix ist (vgl. Satz 1.3.7) 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 wegen der quadratischen Konvergenz für jeden Eigenwert in der Regel nur 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, dagegen benötigt man für S und U, die

$$\text{volle Schur-Normalform } \cong 30n^3 \text{Operationen.}$$

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

a) Zunächst entspricht nach (1.3.14) ein QR-Schritt einem Schritt der orthogonalen Iteration (1.3.5), vgl. (1.3.15)). Da $R^{(k)}$ obere Dreieckmatrix ist, wird somit in der *ersten* Spalte von $U^{(k)}$ eine Vektoriteration durchgeführt: $AU^{(k-1)}e^{(1)} = r^{(k)}_{11}U^{(k)}e^{(1)}$.

b) Bei Spektralverschiebung ändert sich (1.3.15) zu

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

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

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

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

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

Dies ist gerade ein Schritt der Inversen Iteration (1.3.4) mit der *letzten* Spalte von $U^{(k-1)}$!

Fazit:

Das QR-Verfahren führt 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, und unglücklicherweise kann Pivotisierung die Konvergenz zerstören. Daher ist das Verfahren weniger verläßlich als das QR-Verfahren und wird daher kaum noch verwendet.

1.4 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 übertragbar sind. Zwei Eigenschaften des hermiteschen Eigenwert-Problems

$$Ax = \lambda x, \quad A = A^*, \tag{1.4.1}$$

$0 \neq x \in \mathbb{C}^n$, 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 1.1.8. Etwas direkter ist allerdings der Zugang über die quadratische Form.

Definition 1.4.1 *Zu einer beliebigen Matrix $A \in \mathbb{C}^{n\times n}$ und $0 \neq x \in \mathbb{C}^n$ heißt*

$$R_A(x) := \frac{x^*Ax}{x^*x} \tag{1.4.2}$$

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 hermiteschen 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. Der folgende Satz ist eigentlich eine Fehlerschranke, welche in §1.6 näher betrachtet werden.

Satz 1.4.2 *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 $\hat{\varepsilon} := \|Ax - \lambda_k x\|$ klein ist und x daher als Näherung für den Eigenvektor $x^{(k)}$ angesehen werden kann, 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 $\hat{\varepsilon}^2$ des Fehlers 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 hermitesche Matrizen kann die Eigenwert-Näherung bei der *Vektor-Iteration* durch Verwendung des Rayleigh-Quotienten verbessert werden. Für die modifizierte Iteration

$$\left.\begin{array}{lcll} z^{(k)} & := & Aq^{(k-1)}, & \\ \lambda^{(k)} & := & q^{(k-1)^*} z^{(k)} & = R_A(q^{(k-1)}), \\ q^{(k)} & := & z^{(k)}/\|z^{(k)}\|_2 & \end{array}\right\} \; k = 1, 2, \ldots \tag{1.4.3}$$

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 1.3.1 der doppelte Exponent $2k$. Die verbesserte Eigenwert-Schätzung wirkt sich noch viel günstiger auf die *Inverse Iteration* (1.3.4) aus, wenn der Rayleigh-Quotient bei der Spektralverschiebung eingesetzt wird. Es sei $\|q^{(0)}\|_2 = 1$, $\lambda^{(0)} := R_A(q^{(0)})$, und

$$\left.\begin{array}{lcl} z^{(k)} & := & (A - \lambda^{(k-1)} I)^{-1} q^{(k-1)}, \\ \lambda^{(k)} & := & \lambda^{(k-1)} + z^{(k)^*} q^{(k-1)} / \|z^{(k)}\|_2^2, \\ q^{(k)} & := & z^{(k)}/\|z^{(k)}\|_2 \end{array}\right\} \; k = 1, 2, \ldots \tag{1.4.4}$$

Dabei ist in der zweiten Zeile tatsächlich

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

Dieses Verfahren (1.4.4) hat jetzt extrem gute Konvergenzeigenschaften.

Satz 1.4.3 *Bei einer hermiteschen Matrix A konvergiert die Inverse Iteration (1.4.4) 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*

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

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

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

Beispiel 1.4.4 $n = 2$, $A = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$ (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 1.4.2)

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

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

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

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

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

Beispiel 1.4.5 Wird zur Eigenwert-Gleichung (1.4.1) noch eine Normierungsbedingung für den Eigenvektor hinzugefügt, erhält man ein nichtlineares Gleichungssystem für die $n+1$ Unbekannten x, λ. Zu

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

Das Gleichungssystem eines Newton-Schritts,

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

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

kann in folgender Weise gelöst werden

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

Die Vorschrift für $z^{(k)}$ sieht aus wie bei der Inversen Iteration (1.4.4). Daher unterscheidet sich das (quadratisch konvergente) Newtonverfahren nur durch die fehlende exakte Normierung (i.a. ist hier $\|x^{(k)}\| \neq 1$) von der kubisch konvergenten Inversen Iteration.

Auch beim *QR-Verfahren* können die Aussagen gegenüber dem unsymmetrischen Fall verbessert werden. Nach Satz 1.2.3 kann man oBdA für A Tridiagonalgestalt annehmen. 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 (1.3.10), (1.3.12), denn der Spektralparameter ist ein Rayleigh-Quotient

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

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

$$\lambda^{(k-1)} = \frac{1}{2}(s_{nn}^{(k-1)} + s_{n-1,n-1}^{(k-1)}) + sign(\delta_{k-1})\sqrt{\delta_{k-1}^2 + |s_{n-1,n}^{(k-1)}|^2}. \tag{1.4.5}$$

Für diese Wahl kann sogar folgende globale Konvergenzaussage gemacht werden. Eine Tridiagonalmatrix heißt dabei *unzerlegbar*, wenn $a_{i,i-1} \neq 0 \ \forall i = 2, \ldots, n$. Unzerlegbare Tridiagonalmatrizen besitzen nur einfache Eigenwerte, bei zerlegbaren Matrizen können die Eigenwerte aus unzerlegbaren Teilmatrizen bestimmt werden.

Satz 1.4.6 (Wilkinson) *Für eine unzerlegbare hermitesche Tridiagonalmatrix A konvergiert das QR-Verfahren (1.3.10) mit Spektralverschiebung (1.4.5) global, also ohne Startschritte. Die Konvergenz ist mindestens quadratisch, d.h.,* $\exists \varepsilon_0 > 0$ *so, dass*

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

Mit dieser QR-Variante können (als empirischer Wert) alle Eigenwerte einer hermiteschen Matrix mit *ca.* $\frac{4}{3}n^3$ Operationen (d.h., dem Aufwand für die Reduktion auf Tridiagonalgestalt) und die volle hermitesche Schur-Normalform mit ca. $10n^3$ Operationen berechnet werden.

Außer den Modifikationen der allgemein anwendbaren Verfahren gibt es noch Spezialverfahren für den symmetrischen Fall, welche in bestimmten Situationen Vorteile aufweisen.

Bisektionsverfahren

Dieses verblüffende Verfahren erlaubt die gezielte Berechnung einzelner Eigenwerte der Matrix, da man die Anzahl der Eigenwerte in einem beliebigen Intervall exakt angeben kann. Die Grundlage des Verfahrens ist die Tatsache, dass die Unterdeterminanten (Haupt-Minoren)

$$p_k(\lambda) := \det \begin{pmatrix} a_{11} - \lambda & a_{12} & & & \\ a_{21} & a_{22} - \lambda & a_{23} & & \\ & \ddots & \ddots & \ddots & \\ & & . & . & a_{k-1,k} \\ & & & a_{k,k-1} & a_{kk} - \lambda \end{pmatrix}, \ k = 1, \ldots, n, \tag{1.4.6}$$

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

Definition 1.4.7 *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_k, $k = 1, \ldots, n-1$, *gilt*

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

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

$$w(\hat{x}) := \text{ Anzahl der Vorzeichenwechsel in } (p_0(\hat{x}), p_1(\hat{x}), \ldots, p_n(\hat{x})) \tag{1.4.7}$$

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

Satz 1.4.8 *Die Anzahl der reellen Nullstellen des Polynoms p_n einer Sturmschen Kette im halboffenen Intervall $[a, b)$ ist gleich*

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

Beweis [Stoer, Numer. Math. 1]

Zur Anwendung auf das Eigenwertproblem ist nachzuweisen, dass die Haupt-Minoren p_k aus (1.4.6) tatsächlich eine Sturm-Kette bilden und die Bedingungen der Definition 1.4.7 erfüllen. Dazu, und zur numerischen Auswertung dient folgende Rekursionsformel,

$$\begin{aligned} p_0(\lambda) &:= 1,\ p_1(\lambda) := a_{11} - \lambda, \\ p_{k+1}(\lambda) &:= (a_{k+1,k+1} - \lambda)p_k(\lambda) - |a_{k+1,k}|^2 p_{k-1}(\lambda),\ k = 1, \ldots, n-1, \end{aligned} \tag{1.4.8}$$

die durch Entwicklung nach der letzten Zeile in (1.4.6) verifiziert wird:

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

Also kann man insbesondere den Wert $p_n(\lambda)$ mit $4n$ Operationen bestimmen.

Beispiel 1.4.9

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

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

Satz 1.4.10 *Die Matrix $A \in \mathbb{C}^{n\times n}$ sei eine unzerlegbare, hermitesche Tridiagonalmatrix. Dann gelten für die Polynome p_k , $k = 0, \ldots, n$, aus (1.4.6), (1.4.8) die Aussagen*

a) $p_k \in \Pi_k, \quad p_k(\lambda) = (-\lambda)^k + \ldots,$ *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 $\lambda_1^{(k)} < \ldots < \lambda_k^{(k)}$, die durch die Nullstellen von p_{k-1} getrennt werden,

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

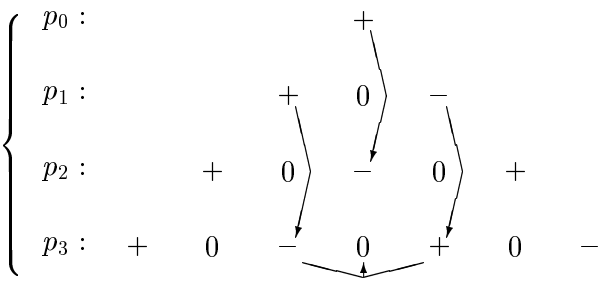
c) $p_k(\xi) = 0 \quad \Rightarrow \quad p_{k-1}(\xi)p_{k+1}(\xi) < 0.$

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

Beweis a) Folgt induktiv aus (1.4.8).

c) Aus (1.4.8) folgt für $p_k(\xi) = 0$ direkt $p_{k+1}(\xi) = -|a_{k+1,k}|^2 p_{k-1}(\xi)$. Da $a_{k+1,k} \neq 0$ ist n.V., gilt $p_{k+1}(\xi)p_{k-1}(\xi) \leq 0$. Keiner dieser beiden Werte kann aber verschwinden, da sonst aus (1.4.8) $p_0(\xi) = 0$ folgen würde im Widerspruch zur Definition $p_0 \equiv 1$.

b) Die Nullstellen sind reell, da die p_k charakteristische Polynome hermitescher Matrizen sind. Die Trennungseigenschaft folgt induktiv dadurch, dass für jedes Polynom die Vorzeichenverteilung in und zwischen den Nullstellen aufgelistet wird:



(1.4.9)

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 (1.4.9): sign $p_{k-1}(\lambda_j^{(k)}) = \text{sign}\left(p_k(\lambda_j^{(k)} - \epsilon) - p_k(\lambda_j^{(k)} + \epsilon)\right)$. ■

Mit Satz 1.4.8 läßt sich eine sehr einfache Methode zur Bestimmung eines beliebigen Eigenwerts λ_m, $1 \leq m \leq n$, angeben, da insbesondere alle EWe einfach sind, $\lambda_1 < \ldots < \lambda_n$. Ausgehend von einem Startintervall $[a, b]$, z.B., $[a, b] = [-\|A\|, \|A\|]$, wird die Einschließung von λ_m durch *Bisektion* (vgl. Numerik I §4.1) anhand der Vorzeichenwechsel w iterativ verbessert:

```
wiederhole:
   λ := (a + b)/2;
   falls w(λ) ≥ m setze b := λ
      sonst a := λ;
bis |a − b| ≤ ε(|a| + |b|);
```

(1.4.10)

Bessere a-priori-Einschließungen der Eigenwerte werden im nächsten Abschnitt behandelt. Jedes der erzeugten Teilintervalle enthält wegen $w(-\infty) = 0$ den Eigenwert λ_m. Die Konvergenz des Bisektionsverfahrens (1.4.10) ist im Vergleich zur Endkonvergenz der anderen Verfahren recht langsam, da die Intervallänge in jedem Schritt nur halbiert wird ($\Rightarrow$ 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 erfolgen.

Es gibt noch weitere Spezialverfahren für den symmetrischen Fall, die aber nur in sehr speziellen Fällen Vorteile gegenüber den besprochenen haben. So arbeitet etwa das Jacobi-Verfahren mit Rotationen, die iterativ *einzelne* Nebendiagonalelemente von A auf null transformieren. In gewissem Ausmaß sind dabei verschiedene Rotationen unabhängig voneinander und können daher parallel ausgeführt werden. Mögliche Vorteile dieser Parallelisierbarkeit hängen aber stark von der Rechnerstruktur ab.

1.5 Überblick der Eigenwert-Verfahren

Die folgende Tabelle faßt die Eigenschaften und die Einsatzbereiche der behandelten Verfahren zusammen. Die als letztes erwähnten Krylov-Verfahren werden 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	+	kubisch u. global konvergent	zu λ_0 nächster EW u. EV, z.B. nach QR-Verfahren, Bisektion
orthogonale Iteration	+	+	Gruppe größter EWe, invarianter Unterraum, dünne Matrizen
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)
Krylov-Verfahren	+	+	größte EWe, schnelle Approximation bei dünnbesetzter Matrix

1.6 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-Verfahren) etwa gilt, dass das Ergebnis bei numerischer Rechnung statt der exakten Matrix $T = Q^* A Q$ eine Matrix $\tilde{T}$ ist mit

$$\tilde{T} = Q^*(A + F)Q, \quad \|F\|_2 \leq c\|A\|_2 \cdot \mathcal{E}. \tag{1.6.1}$$

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

$$\lambda_j = \max_{\dim S = j} \min_{0 \neq y \in S} R_A(y), \quad j = 1, \ldots, n,$$

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

Insbesondere gilt also $\lambda_n = \min_{y\neq 0} R_A(y)$, $\lambda_1 = \max_{s\neq 0} R_A(s)$.

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 1.6.1 fallend angeordnet. Eine direkte Folgerung des Minimax-Prinzips ist

Satz 1.6.2 *Sind A und $A + F$ hermitesch, dann gilt für $j = 1, \ldots, n$*

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

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

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

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 $|\lambda_j(A+F) - \lambda_j(A)| \leq \|F\|_2$. Bei den Eigenvektoren kann die Störung allerdings größer ausfallen. Der folgende Satz setzt voraus, dass eine Näherung x für einen Eigenvektor und λ für den zugehörigen Eigenwert (z.B. durch den Rayleigh-Quotienten) bekannt ist.

Satz 1.6.3 *Gegeben sei eine hermitesche Matrix A und ein Paar (x, λ) mit $x \neq 0$. Damit sei*

$$\|Ax - \lambda x\|_2 \leq \varepsilon \|x\|_2 \quad \textit{bzw.} \quad \|Ax - \lambda x\|_2 \leq \delta \|Ax\|_2. \tag{1.6.2}$$

a) Dann gilt

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

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

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

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 entfallen mit $\eta_j := \xi_j\lambda_j$ in allen Summen die Terme mit $\eta_j = 0$:

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

b) Mit Teil a) folgt aus der Voraussetzung auch $\|A - \lambda_k x\| \le 2\varepsilon$, denn

$$\varepsilon\|x\|_2 \ge \|Ax - \lambda_k x\|_2 - |\lambda - \lambda_k|\|x\|_2 \ge \|Ax - \lambda_k x\|_2 - \varepsilon\|x\|_2 \quad \Rightarrow$$

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

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

Bemerkung: 1) Nach Satz 1.4.2 ist der Rayleighquotient $R_A(x)$ die beste Eigenwert-Schätzung zur Eigenvektor-Näherung x. Im Beweis dieses Satzes wurde gezeigt, dass (bei $\|x\|_2 = 1$) gilt

$$\|Ax - \lambda x\|_2^2 = |\lambda - R_A(x)|^2 + \|Ax\|_2^2 - R_A(x)^2.$$

Mit dem optimalen $\lambda = R_A(x)$ ist in der Voraussetzung des Satzes $\varepsilon^2 = \|Ax - R_A(x)x\|_2^2$ (auch $= \|Ax\|_2^2 - R_A(x)^2$). In der Schranke von Satz 1.4.2 folgt mit obiger Identität bei $\lambda = \lambda_k$, dass

$$\min_j |\lambda_j - R_A(x)| \le \frac{1}{d_k}\|Ax - \lambda_k x\|_2^2 = \frac{1}{d_k}(\underbrace{|\lambda_k - R_A(x)|^2}_{S.1.6.3a} + \varepsilon^2) \le 2\frac{\varepsilon^2}{d_k}. \tag{1.6.3}$$

Für $\varepsilon < d_k/2$ ist die alte Schranke besser als die aus dem Satz 1.6.3 a).

2) Die in der ersten Bemerkung verwendete Größe ε ist bei der Vektoriteration (1.4.3) bzw. (1.4.4) einfach verfügbar. Bei der Inversen Iteration (1.4.4) etwa gilt mit $\lambda^{(k)} = R_A(z^{(k)})$ und $B_{k-1} := A - \lambda^{(k-1)}I$:

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

Alle hier auftretenden Größen werden bei der Inversen Iteration sowieso berechnet, mit Satz 1.6.3 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 1.6.4 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 $Ax \doteq (-0.28, -0.233, 0.3823)^\mathsf{T}$, $\|Ax\|^2/\|x\|^2 \doteq 0.280187$ und daher $R_A(x) \doteq -0.529326$ eine Eigenwert-Näherung der Genauigkeit $\varepsilon = 1.2_{10} - 3$ nach Satz 1.6.3. Die bessere Aussage (1.6.3) liefert $8.3_{10} - 5$. Für den Eigenvektor ergibt sich die gröbere Schranke $\|x - x^{(2)}\| \leq 7.2_{10} - 2$.

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

$$r_i := \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}|, \ \ i = 1, \ldots, n. \tag{1.6.4}$$

Dann liegen alle Eigenwerte von A in der Vereinigung der Kreisscheiben

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

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

$$\begin{aligned} 0 &= |(a_{kk} - \lambda)x_k + \sum_{j\neq k} a_{kj}x_j| \geq |a_{kk} - \lambda||x_k| - \sum_{j\neq k} |a_{kj}|\|x\|_\infty \\ &= (|a_{kk} - \lambda| - r_k)\|x\|_\infty > 0. \quad \blacksquare \end{aligned}$$

Beispiel 1.6.6 Für die Matrix des letzten Beispiels ist $r_1 \doteq 4.34, r_2 = 5, r_3 \doteq 5.34$. Im Reellen gilt $K_1 \doteq [-3.34, 5.34], K_2 = [-3, 7], K_3 \doteq [-2.34, 8.34]$ und $K_1 \cup K_2 \cup K_3 \subseteq [-3.34, 8.34]$.

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

$$D^{-1}AD = \left(\frac{d_j}{d_i}a_{ij}\right)_{i,j},\ D = \mathrm{diag}(d_i).$$

3) Mit Satz 1.6.5 können die für andere Verfahren erforderlichen Anfangsschätzungen für (alle) Eigenwerte bestimmt werden, etwa für Bisektion (1.4.10) oder Inverse Iteration (1.3.4).

Bei Störungen können insbesondere mehrfache Eigenwerte sehr empfindlich reagieren.

Beispiel 1.6.7 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 zwar stetig, aber nicht-differenzierbar von ε ab!

Für Störungssätze im unsymmetrischen Fall kann die Jordan- oder die Schur-Normalform herangezogen werden. Diese liefern aber keine Schranken für den Fehler einzelner Eigenwerte, sondern nur den Maximalfehler.

Satz 1.6.8 *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

$$\begin{aligned} & \lambda x - Ax = Fx \iff x = (\lambda I - A)^{-1}Fx \quad \Rightarrow \\ 1 \;=\; & \|x\| \le \|(\lambda I - A)^{-1}F\| \le \|(\lambda I - A)^{-1}\|\|F\| \\ \le\; & \|(\lambda I - J)^{-1}\|\kappa(X)\|F\| = \frac{1}{\min_i |\lambda - \lambda_i|}\kappa(X)\|F\| \quad \blacksquare \end{aligned} \tag{1.6.5}$$

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 $\Lambda = \mathrm{diag}(\lambda_j)$ und den Rest M, $S = \Lambda + M$. Die Norm $\|M\|_2$ ist ein Maß dafür, wieweit sich A von einer normalen Matrix unterscheidet.

Satz 1.6.9 *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 (1.6.5) eine andere Abschätzung der Inversen verwendet,

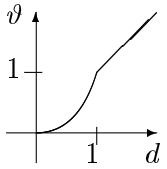
$$\|(\lambda I - A)^{-1}\|_2 = \|(\lambda I - \Lambda - M)^{-1}\|_2 \le \|(\lambda I - \Lambda)^{-1}\| \| \Big(I - (\lambda I - \Lambda)^{-1} M \Big)^{-1} \|_2 .$$

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 (1.6.5) führt daher hier auf

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



Die Behauptung folgt mit $\min\{d, d^p\} \le \vartheta \iff d \le \max\{\vartheta, \vartheta^{1/p}\}$. ■

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 (1.6.1) ist bei allen numerischen Algorithmen mit einer Störmatrix mindestens in der Größenordnung der Maschinengenauigkeit $\mathcal{E}$ zu rechnen. Nach Satz 1.6.9 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

$$\lim_{\varepsilon \to 0} \Big| \frac{\lambda(A + \varepsilon F) - \lambda(A)}{\varepsilon} \Big| \le \frac{1}{|y^* x|}, \tag{1.6.6}$$

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 Eigenwertproblems mit diesen Matrizen.

Beispiel 1.6.10 $A = \begin{pmatrix} 1 & 1 \\ \varepsilon & 1 \end{pmatrix}$, $\lambda_{1/2} = 1 \pm \sqrt{\varepsilon}$, vgl. Bsp. 1.6.7.

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

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 $\|F\| \cong \mathcal{E}\|A\|$ ("Rückwärtsanalyse"). In diesem Sinn liefern die behandelten Transformationsverfahren (Hessenberg-, QR-) wegen (1.6.1) numerisch akzeptable Approximationen.

1.7 Die Singulärwert-Zerlegung

Die Eigenwerttheorie führt bei reellen, symmetrischen Matrizen A auf die einfache Faktorisierung $A = UDU^{\mathsf{T}}$, wobei D eine (reelle) Diagonalmatrix ist (der Eigenwerte) und die Basismatrizen U und U^{T} orthogonal und daher zueinander invers sind. Anhand dieser Faktorisierung kann die Wirkung der zu A gehörigen linearen Abbildung einfach charakterisiert und in die verschiedenen Anteile zerlegt werden. Für beliebige Matrizen A muß bei einer analogen Faktorisierung $A = UDV^{\mathsf{T}}$ mindestens eine der genannten Eigenschaften aufgegeben werden. Überblick (mit teilweise abweichenden Standard-Bezeichnungen):

Faktorisierung $A = UDV^{\mathsf{T}}$

Matrix A	Normalform	Eigenschaft der Faktoren		
		D diagonal?	U, V orthog?	$V^{\mathsf{T}} = U^{-1}$?
A symmetrisch	$A = UDU^{\mathsf{T}}$	$\surd$	$\surd$	$\surd$
A beliebig	Jordan– $A = UDU^{-1}$	$(\surd)$	U regulär	$\surd$
	Schur– $A = US\bar{U}^{\mathsf{T}}$	S Dreieck	$\surd$	$\surd$
	Singulärwert– $A = U\Sigma V^{\mathsf{T}}$	$\surd$	$\surd$	nein

Die Eigenschaft $V^{\mathsf{T}} = U^{-1}$ 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

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

interessant. Insbesondere kann man hier beliebige (nicht-quadratische) 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:

$$\max_j \sigma_j = \|A\|_2 = \max_{x\neq 0} \frac{\|Ax\|_2}{\|x\|_2}, \tag{1.7.1}$$

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

$$\|U\Sigma V^\mathsf{T} x\|_2 = \|\Sigma(V^\mathsf{T} x)\|_2 \leq \|\Sigma\|\|V^\mathsf{T} x\|_2 = \max_j \sigma_j \|x\|_2.$$

Die Beziehung (1.7.1) folgt daraus, da das Maximum tatsächlich angenommen wird.

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

Satz 1.7.1 (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*

$$\sigma_1 \geq \sigma_2 \geq \ldots \sigma_k \geq 0$$

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

$$A = U\Sigma V^\mathsf{T} = \sum_{j=1}^{k} \sigma_j u^{(j)} v^{(j)\mathsf{T}}, \quad \Sigma = \begin{pmatrix} \sigma_1 & & & 0 \\ & \sigma_2 & & \\ & & \ddots & \\ 0 & & & \sigma_k \end{pmatrix}. \tag{1.7.2}$$

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 (1.7.1) 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

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

Für den speziellen Vektor $v := (\sigma_1, z^\mathsf{T})^\mathsf{T}$ gilt

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

daher ist $\|A\|_2 = \|A^{(1)}\|_2 \geq \|v\|_2 = \sqrt{\sigma_1^2 + z^\mathsf{T} z}$. Da aber schon $\sigma_1 = \|A\|_2$ war, folgt $z = 0$. Eine k-malige Anwendung dieses Arguments liefert $U^\mathsf{T} AV = \Sigma$ und daher $U\Sigma V^\mathsf{T} = UU^\mathsf{T} AVV^\mathsf{T}$. Die Behauptung (1.7.2) folgt aus der Überlegung, dass UU^T Projektor auf $R(A)$ im Fall $m \geq n$ ist bzw. VV^T Projektor auf $R(A^\mathsf{T})$ im Fall $n > m$. ∎

Beispiel 1.7.2 Es ist

$$A = \begin{pmatrix} 6.2 & -1.6 \\ -2.8 & 10.4 \\ 7.6 & -6.8 \end{pmatrix} = U\Sigma V^\mathsf{T}$$

mit den Faktoren

$$U = \frac{1}{3}\begin{pmatrix} 1 & 2 \\ -2 & 2 \\ 2 & 1 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} 15 & \\ & 6 \end{pmatrix}, \quad V = \frac{1}{5}\begin{pmatrix} 3 & 4 \\ -4 & 3 \end{pmatrix}.$$

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

$$\sigma_n > 0 \quad \text{gilt} \quad \|A^{-1}\|_2 = 1/\sigma_n. \tag{1.7.3}$$

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

$$\begin{aligned} R(A) &= \text{span}(u^{(1)}, \ldots, u^{(r)}), \quad \text{Rang}(A) = r, \\ N(A) &= \text{span}(v^{(r+1)} \ldots, v^{(k)}) = \text{span}(v^{(1)}, \ldots, v^{(r)})^{\perp}. \end{aligned}$$

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 definiert man mit $i \leq k = \min\{m, n\}$ eine approximierende Teilmatrix A_i, die nur die i *größten* Singulärwerte berücksichtigt durch

$$A_i := \sum_{j=1}^{i} \sigma_j u^{(j)} {v^{(j)}}^{\mathsf{T}} =: U_i \Sigma_i V_i^{\mathsf{T}}. \tag{1.7.4}$$

Nach Definition besitzt A_i höchstens Rang i. Der Fehler zu A ist

$$\|A - A_i\|_2 = \sigma_{i+1}, \tag{1.7.5}$$

da für $y \in \mathbb{C}^n$ gilt $\|(A - A_i)y\|^2 = \sum_{j=i+1}^{k} \sigma_j^2 ({v^{(j)}}^{\mathsf{T}} y)^2 \leq \sigma_{i+1}^2 \|y\|^2$. Der $i+1$-te singuläre Wert σ_{i+1} gibt also an, in welchem Abstand zur Gesamtmatrix A eine Matrix vom Rang $\leq i$ existiert, insbesondere kann für $\sigma_{i+1} > 0$ gezeigt werden, dass die in (1.7.4) definierte Matrix A_i die zu A nächstgelegene Matrix vom Rang i ist,

$$\|A - A_i\|_2 = \min_{\text{Rg}(B)=i} \|A - B\|_2 = \sigma_{i+1}.$$

Ist also A durch Meß- oder Rundungsfehler der Gesamtgröße $\|A - \widehat{A}\| = \varepsilon$ aus einer (unbekannten) Matrix $\widehat{A}$ hervorgegeangen, läßt sich aufgrund von (1.7.5) dann ein Rang r für $\widehat{A}$ 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 $\sigma_j \leq \varepsilon$ keine erkennbare Informationen über die Ausgangsmatrix $\hat{A}$ enthalten, sondern nur "Rauschen". Man definiert daher einen gegen Störungen der Matrix unempfindlicheren *numerischen Rang* durch

$$\text{Rg}_{\varepsilon}(A) := \sum_{\sigma_j > \varepsilon} 1 = (\text{Anzahl der singulären Werte } \sigma_j > \varepsilon). \tag{1.7.6}$$

Die Schwelle $\varepsilon \geq 0$ 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.

Allgemein formuliert ist die Größe σ_j ein Maß dafür, wie stark die Rang-1-Matrix (das "Merkmal") $u^{(j)}v^{(j)\mathsf{T}}$ in der Matrix (Datentabelle) A enthalten ist.

Kleinste-Quadrate-Lösung

Mit der Singulärwertzerlegung kann die in der Numerik 1, §4.6, besprochene Kleinste-Quadrate-Lösung x^+ in allen Situationen explizit angegeben werden. In der Numerik 1 wurde die verallgemeinerte Lösung eines LGS $Ax = y$ durch Minimierung des Defekts $\|Ax - y\|_2$ gesucht und bei Nichteindeutigkeit nochmals die Norm $\|x\|_2$ selbst minimiert. In den Orthonormalbasen U, V bei $A = U\Sigma V^{\mathsf{T}}$ sind diese Minimierungen direkt durchführbar. Mit der orthogonalen Aufspaltung $y = UU^{\mathsf{T}}y + (I - UU^{\mathsf{T}})y$ gilt

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

Da x nur noch in der ersten Summe vorkommt, wird das Minimum in den Komponenten $\xi_j := v^{(j)\mathsf{T}}x = u^{(j)\mathsf{T}}y/\sigma_j$, $j \leq r$, der V-Basis angenommen. Die restlichen Komponenten von $\xi = V^{\mathsf{T}}x$ sind beliebig. Wegen $x = V\xi + (I - VV^{\mathsf{T}})x$ wird aber die Lösung minimaler Norm $\|x\|_2^2 = \|\xi\|_2^2 + \|(I - VV^{\mathsf{T}})x\|_2^2$ mit $\xi_{r+1} = \ldots = \xi_k = 0$ und $x = V\xi$ erreicht. Dies ergibt explizit die Lösung

$$x^+ = \sum_{j=1}^{r} \frac{u^{(j)\mathsf{T}}y}{\sigma_j}\, v^{(j)} \tag{1.7.7}$$

des Kleinste-Quadrate-Problems und entspricht der expliziten Darstellung vom Rang r für die Pseudo-Inverse

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

Dieses Ergebnis korrespondiert auf folgende Weise mit denen aus der Numerik 1. Jede Kleinste-Quadrate-Lösung erfüllt das Gleichungssystem $Ax = Py$ mit dem Orthogonalprojektor $P : \mathbb{R}^m \to R(A)$ und die Minimallösung x^+ ist außerdem durch die Eigenschaft $x^+ \in N(A)^\perp$ charakterisiert. Für $\mathrm{Rang}(A) = r$ ist $A = U_r\Sigma_r V_r^{\mathsf{T}}$ und $R(A) = \mathrm{span}(u^{(1)}, \ldots, u^{(r)}) = R(U_r)$, sowie $N(A)^\perp = \mathrm{span}(v^{(1)}, \ldots, v^{(r)}) = R(V_r)$. Daher ist $U_rU_r^{\mathsf{T}} = P$ der Orthogonalprojektor auf $R(A)$ und für die Lösung gilt der Ansatz $x^+ = V_r\xi$, $\xi \in \mathbb{R}^r$, dessen Komponenten ξ sich, wie oben vorgerechnet, aus dem System $0 = Ax^+ - Py = U_r(\Sigma_r\xi - U_r^{\mathsf{T}}y)$ ergeben.

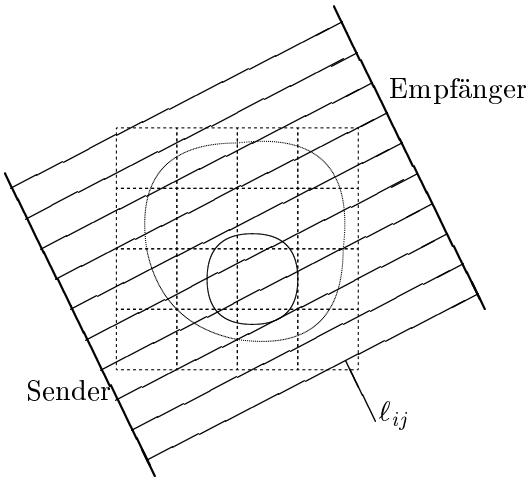
Dieser grundlegende Zusammenhang bietet allerdings wegen des hohen Rechenaufwands für die SWZ (s.u.) noch keinen praktischen Vorteil gegenüber der früher verwendeten QR-Zerlegung. Dies ändert sich allerdings, wenn man die Ergebnisse des vorhergehenden Abschnitts heranzieht. Denn (1.7.7) ist nicht immer eine sinnvolle Lösung des Problems $Ax = y$, vor allem dann, wenn A

sehr kleine Singulärwerte in der Größenordnung der Problemfehler ε besitzt. Dann hat A beinahe einen Rangdefekt, es gilt $\mathrm{Rg}_\varepsilon(A) < r$, etwa mit $\varepsilon = \mathcal{E}\|A\|$. Daher enthalten aber Komponenten zu kleinen singulären Werten $0 < \sigma_j \ll \sigma_1$ kaum sinnvolle Information zum Problem ("Rauschen"). Aber ausgerechnet diese Anteile werden wegen der Division $1/\sigma_j$ in x^+ aus (1.7.7) sehr stark hervorgehoben, der größte Summand in x^+ etwa ist $v^{(r)}{u^{(r)}}^{\mathsf{T}} y/\sigma_r$. Hier ist es sinnvoller, in der Kleinste-Quadrate-Lösung (1.7.7) nur die wichtigsten Komponenten innerhalb des numerischen Rangs zu berücksichtigen und den Rest abzuschneiden. Dann hat die Lösung die Form

$$x_\varepsilon^+ := \sum_{j=1}^{r_\varepsilon} \frac{{u^{(j)}}^{\mathsf{T}} y}{\sigma_j}\, v^{(j)}, \quad r_\varepsilon := \mathrm{Rg}_\varepsilon(A). \tag{1.7.8}$$

Anwendungen

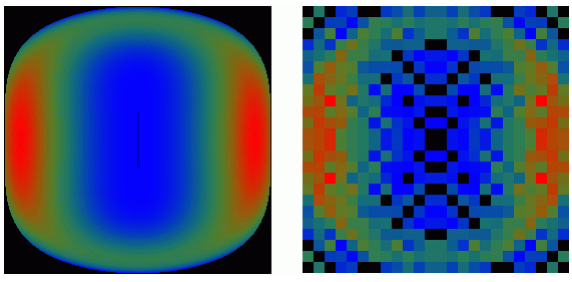
Beispiel: (Computer-Tomographie) Die Dichte-Verteilung in einem Körper läßt sich aus der von außen gemessenen Absorption von (Röntgen-, Gamma-) Strahlen rekonstruieren. Dazu wird eine Schicht des Untersuchungsbereichs in Zellen, z.B. Quadrate unterteilt. Dieser Bereich wird dann mit einer Reihe von parallelen Meßstrahlen durchleuchtet und die Messung unter verschiedenen Drehwinkeln der Apparatur wiederholt. Unter Annahme einer konstanten Dichte d_j in einzelnen Zellen des Untersuchungsbereichs ergibt sich für die Absorption a_i im i-ten Meßstrahl die Beziehung



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

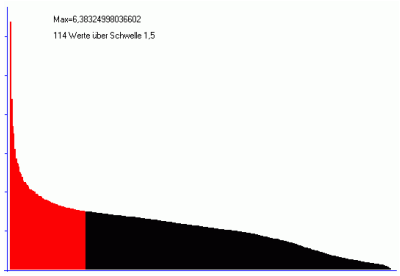
Dabei gibt der Koeffizient ℓ_{ij} die Weglänge des i-ten Meßstrahls im j-ten Quadrat an. Je nach Zahl der Messungen erhält man ein unter- oder überbestimmtes Lineares Gleichungssystem für die unbekannten Dichten d_j. Dieses Rekonstruktionsproblem (Radon-Transformation) reagiert allerdings empfindlich auf Störungen. Eine feinere Auflsung des Gebiets durch Verwendung kleinerer Zellen verschärft diese Problematik leider noch ("schlecht gestelltes Problem"). Daher werden im Beispiel viele Messungen verwendet und in der Praxis sind Zusatzmaßnahmen erforderlich, um akzeptable Lösungen zu bekommen.

Das Bild rechts zeigt die Kleinste-Quadrate-Lösung mit $n = 24 * 24 = 576$ Quadraten und $m = 24 * 40$ Messungen. Neben der Originaldichteverteilung (Falschfarben) wird die Rekonstruktion x^+ dargestellt, die offensichtlich von (oszillierenden) Störungen überlagert ist.

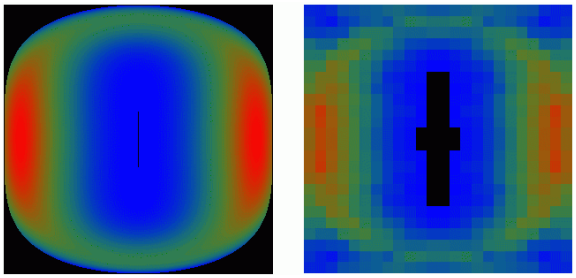


Diese Störungen gehören zu kleinen Singulärwerten und können jetzt in der Lösung x_ε^+ (1.7.8) ausgeblendet werden. Das folgende linke Diagramm zeigt die Singulärwerte nach abfallender Größe. Im rechten Bild ist das Original und die Rekonstruktion x_ε^+ mit der Schwelle $\varepsilon = 1.5$, wodurch 80% der Singulärwerte (im linken Bild schwarz) ausgeblendet wurden.

Singulärwerte



Rekonstruktion nach (1.7.8), $\varepsilon = 1.5$:



Komponentenanalyse

Jede Sammlung von numerischen Daten in Tabellenform kann als eine Matrix A angesehen werden. In Anwendungen, bei denen es darauf ankommt, aus großen Datenmengen die wesentlichen Zusammenhänge (*Merkmale,Komponenten*) herauszufiltern, ist daher die Singulärwertzerlegung von Interesse. Denn die Darstellung (1.7.2) der Datentabelle A kann als eine Zerlegung in ,,Merkmale" $u^{(j)}v^{(j)\mathsf{T}}$ interpretiert werden, deren Wichtigkeit durch den zugehörigen Singulärwert σ_j angegeben wird. Einer Menge nicht-numerischer Objekte (Text-Dateien, Bilder,..) läßt sich eine Matrix dadurch zuordnen, dass man bei dem Objekt Nummer j verschiedene quantitative Angaben (Wort-Häufigkeiten in Datei Nr. j, 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 (1.7.4) definierten Abschnittsmatrizen eine Folge von Modellen für A mit wachsender Komplexität (Rang) und monoton fallendem Fehler $\|A - A_i\| = \sigma_{i+1}$ dar. Die Auswahl des im konkreten Fall geeigneten Modellrangs hängt vom Anwendungsproblem ab, bei großen Datenmengen sind kompakte Modelle $A_i = U_i\Sigma_i V_i^{\mathsf{T}}$ mit $i \ll k$ von besonderem Interesse. Mit solchen Modellen können dann, u.a., folgende Aufgaben behandelt werden:

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

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.

Eine spektakuläre Anwendung ist die zur Erkennung von Gesichtern [Moghaddam, Pentland 1995]. Jedes Gesicht j wird durch eine Bilddatei (Grauwerte, als Vektor) repräsentiert. Die Matrix A entspricht dann der Sammlung von Bildern und in den ersten singulären Vektoren $u^{(1)}, u^{(2)}, \ldots$ der SWZ $A = U\Sigma V^\mathsf{T}$ sind bestimmte Merkmale der Gesichtszüge enthalten (vgl. VL-Vorführung). Als Beispiel wir hier das Haupt-*"Eigen-Gesicht"* $u^{(1)}$ gezeigt:

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, dieses Produkt aber nicht verwendet (hätte größere Kondition). Wie dort kann Arbeit durch Reduktion auf einfachere Gestalt eingespart werden. Bei A sind jetzt unabhängig voneinander orthogonale Umformungen von links und rechts zulässig. Daher 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 also schon Nullen unter r_{11}. Für die schrittweise Umformung wird von folgender Struktur ausgegangen

$$R^{(j)} = \left(\begin{array}{ccc|ccc} * & * & 0 & & & \\ & \ddots & \ddots & & 0 & \\ & & * & * & \cdots & * \\ \hline & & & * & \cdots & * \\ & & & \vdots & & \vdots \\ & & & * & \cdots & * \end{array}\right) \begin{array}{l} \left.\vphantom{\begin{array}{c}a\\a\\a\end{array}}\right\} j-1 \\ \left.\vphantom{\begin{array}{c}a\\a\\a\end{array}}\right\} n-j+1 \end{array} \tag{1.7.9}$$

wobei $R = R^{(2)}$ ist. Bei der ersten Multiplikation von rechts mit einer Spiegelung $K^{(2)} = I - 2w^{(2)}w^{(2)\mathsf{T}}$ soll die erste Spalte von R unverändert bleiben. Daher ist $w_1^{(2)} = 0$ zu wählen und nur $r_{13}, \ldots, r_{1n}$ können eliminiert werden. Dann hat $\tilde{R}^{(2)} = RK^{(2)}$ ab der zweiten Spalte wieder nichttriviale Elemente. Allgemein werden im j-ten Schritt nach Satz 1.2.2 zu $R^{(j)}$ die Spiegelung $K^{(j)}$ und zu $\tilde{R}^{(j)} = R^{(j)}K^{(j)}$ die Spiegelung $H^{(j)}$ so bestimmt, dass gilt

$$\begin{aligned} (\ldots, 0, r^{(j)}_{j-1,j-1}, r^{(j)}_{j-1,j}, \ldots, r^{(j)}_{j-1,n})K^{(j)} &= (\ldots, 0, r^{(j)}_{j-1,j-1}, \tilde{r}^{(j)}_{j-1,j}, 0, \ldots), \\ \text{und} \quad H^{(j)} \begin{pmatrix} \vdots \\ \tilde{r}^{(j)}_{j-1,j} \\ \tilde{r}^{(j)}_{jj} \\ \vdots \\ \tilde{r}^{(j)}_{nj} \end{pmatrix} &= \begin{pmatrix} \vdots \\ \tilde{r}^{(j)}_{j-1,j} \\ r^{(j+1)}_{jj} \\ 0 \\ \vdots \end{pmatrix}. \end{aligned} \tag{1.7.10}$$

Dann hat $R^{(j+1)} = H^{(j)}\tilde{R}^{(j)} = H^{(j)}R^{(j)}K^{(j)}$ wieder die zu (1.7.9) analoge Gestalt. Das Verfahren führt in $n-2$ Schritten zum Ziel, d.h., die letzte Matrix

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

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 im betrachteten fall $m \geq n$. 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

$$T := B^{\mathsf{T}}B = \begin{pmatrix} d_1^2 & d_1b_1 & & & \\ d_1b_1 & d_2^2 + b_1^2 & d_2b_2 & & \\ & \ddots & \ddots & \ddots & \\ & & d_{n-2}b_{n-2} & d_{n-1}^2 + b_{n-2}^2 & d_{n-1}b_{n-1} \\ & & & d_{n-1}b_{n-1} & d_n^2 + b_{n-1}^2 \end{pmatrix} \tag{1.7.11}$$

zugrunde allerdings ohne die explizite Berechnung von $B^{\mathsf{T}}B$! Denn diese Multiplikation würde die Kondition des Problems quadrieren. Da dies numerisch ungünstig ist, führt man den entsprechenden QR-Schritt auf folgende Weise implizit aus. Für $b_1 \neq 0$ ist die erste Operation

des QR-Schritts bei (1.7.11) die Anwendung einer $2 \times 2-$Drehung oder Spiegelung H_1^T, die das Element $t_{21} = d_1 b_1$ eliminiert und danach 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 gilt

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

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

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

Dann ist

$$\hat{B} := (F_1 \cdots F_{n-1})^\mathsf{T} B \underbrace{H_1 G_2 \cdots G_{n-1}}_{=:Q} \tag{1.7.12}$$

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 $j \geq 2$ 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 (1.4.5) 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.

Mit dieser Strategie ist die globale Konvergenz des QR-Verfahrens garantiert, vgl. Satz 1.4.6. Daher liefert das Verfahren sehr schnell eine Singulärwertzerlegung $B = \hat{U}\Sigma\hat{V}^{\mathsf{T}}$. Insgesamt kann also die Singulärwertzerlegung einer allgemeinen Matrix A in drei Schritten erzeugt werden:

$$A \overset{\text{QR-Zerl.}}{\longrightarrow} R = Q^{\mathsf{T}}A \overset{\text{Bidiag.}}{\longrightarrow} B = \hat{H}^{\mathsf{T}}R\hat{K} \overset{\text{Iter. (1.7.12)}}{\longrightarrow} \Sigma = \hat{U}^{\mathsf{T}}B\hat{V}$$

$$A = (Q\hat{H}\hat{U})\Sigma(\hat{K}\hat{V})^{\mathsf{T}} \longleftarrow R = \hat{H}\hat{U}\Sigma(\hat{K}\hat{V})^{\mathsf{T}} \longleftarrow B = \hat{U}\Sigma\hat{V}^{\mathsf{T}}$$

2 Iterative Verfahren 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, §4.5, Demo-Beispiel). Das früher ebenfalls behandelte Relaxationsverfahren läßt sich aber auf einfache Weise verbessern.

2.1 Polynomielle Beschleunigungsverfahren

Ohne Einschränkung wird die Fixpunktgleichung

$$x = Bx + r \quad \Longleftrightarrow \quad Ax = r \ \text{ mit } A = I - B, \tag{2.1.1}$$

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

$$x^{(k+1)} = (1-\omega)x^{(k)} + \omega(Bx^{(k)} + r) = x^{(k)} + \omega(r - Ax^{(k)}),\ k = 0, 1, \ldots, \tag{2.1.2}$$

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

$$\varrho(B(\hat\omega)) = \frac{\lambda_1(A) - \lambda_n(A)}{\lambda_1(A) + \lambda_n(A)} \quad \text{in} \quad \hat\omega = \frac{2}{\lambda_1(A) + \lambda_n(A)}. \tag{2.1.3}$$

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

$$\hat\kappa(A) := \varrho(A)\varrho(A^{-1}) = \frac{\max\{|\lambda_j(A)| :\ j = 1, \ldots, n\}}{\min\{|\lambda_j(A)| :\ j = 1, \ldots, n\}}. \tag{2.1.4}$$

Offensichtlich gilt $1 \le \hat{\kappa}(A) \le \kappa_p(A),\ p \ge 1$. Für symmetrische Matrizen, $A = A^{\mathsf{T}}$, ist sogar $\hat{\kappa}(A) = \kappa_2(A)$. Der Konvergenzfaktor aus (2.1.3) läßt sich also in der Form

$$\min_{\omega\in\mathbb{R}} \varrho(B(\omega)) = \frac{\hat{\kappa}(A)-1}{\hat{\kappa}(A)+1} = 1 - \frac{2}{\hat{\kappa}(A)+1} \tag{2.1.5}$$

schreiben. Diese Optimalitätsaussage gilt genau genommen aber nur bei isolierter Betrachtung eines einzelnen Schrittes (2.1.2). Faßt man dagegen mehrere Schritte mit *wechselnden* Parametern ω_k zusammen, ergibt sich für den Fehler $x^{(m)} - z$ nach m Schritten

$$\begin{aligned} x^{(m)} - z &= x^{(m-1)} - z + \omega_m(Az - Ax^{(m-1)}) = (I - \omega_m A)(x^{(m-1)} - z) \\ &= \underbrace{(I - \omega_m A)(I - \omega_{m-1}A)\cdots(I - \omega_1 A)}_{p_m(A)}(x^{(0)} - z). \end{aligned} \tag{2.1.6}$$

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 (2.1.6) erhält man

Satz 2.1.1 *Bei einer Matrix A mit reellen Eigenwerten*

$$0 < a \le \lambda_n \le \ldots \le \lambda_1 \le b$$

gilt für den Fehler bei der Richardson-Iteration

$$x^{(k)} := x^{(k-1)} + \omega_k(r - Ax^{(k-1)}),\ k = 1, \ldots, m, \tag{2.1.7}$$

mit Parametern ω_k nach m Schritten die Beziehung

$$x^{(m)} - z = p_m(A)\,(x^{(0)} - z). \tag{2.1.8}$$

Dabei ist $p_m \in \Pi_m$, $p_m(0) = 1,\ p_m(t) = (1 - \omega_1 t)\cdots(1 - \omega_m t)$ und

$$\varrho(p_m(A)) \le \max\{|p_m(t)| :\ t \in [a, b]\}. \tag{2.1.9}$$

Beweis Die Identität (2.1.8) entspricht (2.1.6), die Reihenfolge der Faktoren ist beliebig. Für jeden Eigenvektor x von A mit Eigenwert λ gilt

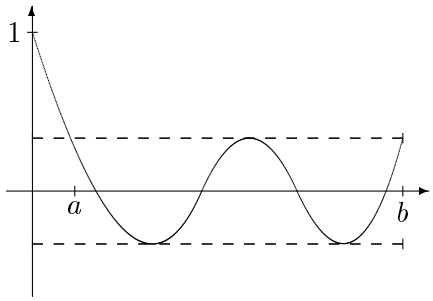
$$p_m(A)x = (I - \omega_m A)\ldots(I - \omega_2 A)(1 - \omega_1\lambda)x = \ldots = p_m(\lambda)x.$$

Aus der Voraussetzung $\lambda_j \in [a, b]$ folgt die Schranke (2.1.9). ∎

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 1.6.5), minimiert man die Schranke (2.1.9),

$$\max\{|p_m(t)| :\ t \in [a, b]\} \stackrel{!}{=} \min,\ p_m \in \Pi_m,$$

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 vom genannten Problem.



Satz 2.1.2 *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

$$\hat{p}_m(t) := T_m\Big(\frac{b + a - 2t}{b - a}\Big)\Big/T_m\Big(\frac{b + a}{b - a}\Big), \tag{2.1.10}$$

wobei $T_m(s)$ das Tschebyscheff-Polynom vom Grad m ist. Es wird durch die Rekursion $T_0 = 1$, $T_1(s) := s$,

$$T_{k+1}(s) := 2s\,T_k(s) - T_{k-1}(s), \ k = 1, 2, \ldots \tag{2.1.11}$$

definiert und besitzt die explizite Darstellung

$$T_m(s) = \begin{cases} \cos(m \arccos s), & |s| \le 1 \\ \cosh(m\,\mathrm{arcosh}\, s), & s > 1. \end{cases}$$

Die Parameter $\hat{\omega}_k$ von $\hat{p}_m(t) = \prod_{k=1}^{m}(1 - \hat{\omega}_k t)$ sind daher

$$\frac{1}{\hat{\omega}_k} = \frac{a + b}{2} - \frac{b - a}{2}\cos\frac{2k - 1}{2m}\pi, \ k = 1, \ldots, m. \tag{2.1.12}$$

Der Minimalwert der Maximumnorm ist

$$\|\hat{p}_m\|_{[a,b]} = 1\Big/T_m\Big(\frac{b + a}{b - a}\Big) \le 2\left(\frac{\sqrt{b/a} - 1}{\sqrt{b/a} + 1}\right)^m. \tag{2.1.13}$$

Beweis Für die Tschebyscheffpolynome gilt nach Definition $|T_m(s)| \le 1$, $s \in [-1, 1]$, und ihre Supremumsnorm $\|T_m\|_{[-1,1]} = 1$ ist minimal unter allen Polynomen der Form $p_m(s) = 2^{m-1}s^m + \ldots$ (vgl. Numerik 1). Die Minimierungsaufgabe des Satzes unterscheidet sich aber von der aus Satz 2.1.13 der Numerik I durch die andere Normierung des Polynoms $p_m(0) = 1$, statt $p_m^{(m)}/m! = c$. Mit dem verschobenen Polynom $\hat{q}_m(t) := T_m\Big(\frac{b+a-2t}{b-a}\Big)$ erfüllt $\hat{p} = \hat{q}/\hat{q}(0)$ nach Konstruktion die Nebenbedingung $\hat{p}_m(0) = 1$, die Normidentität in (2.1.13) folgt aus $\|\hat{q}_m\|_{[a,b]} = 1$. Der Optimalitätsbeweis aus der Numerik I läßt sich hierauf sinngemäß übertragen, er beruht auf der Tatsache, dass $|\hat{p}(t)|$ den Maximalwert (2.1.13) in den $m + 1$ Stellen mit $2t_j = a + b - (b - a)\cos\frac{j\pi}{m}$, $j = 0, \ldots, m$ alternierend annimmt. Jedes andere Polynom $p(t)$ mit $p(0) = 1$, das eine kleinere Supremumsnorm besitzt, würde $\hat{p}$ in m Stellen $t_j \in (a, b)$ schneiden. Damit stimmen $p, \hat{p} \in \Pi_m$ in $m + 1$ Stellen (auch $t = 0$) überein und sind daher identisch. Die Kehrwerte $1/\hat{\omega}_k$ entsprechen den Nullstellen von $\hat{p}_m$ bzw. $\hat{q}_m$.

Zur Normierung wird in (2.1.10) ein Funktionswert $T_m(\hat{s})$ mit $\hat{s} > 1$ verwendet. Um auch für diesen Fall eine explizite Darstellung der Polynome aus (2.1.11) zu erhalten, wird $s = \cosh\sigma = \frac{1}{2}(e^{\sigma} + e^{-\sigma})$ gesetzt. Wegen des Additionstheorems

$$\cosh(k \pm 1)\sigma = \cosh(k\sigma)\,\cosh(\sigma) \pm \sinh(k\sigma)\,\sinh(\sigma),$$

und $\cosh 0 = 1$ gilt daher die Rekursion (2.1.11) mit $s \geq 1$ für die Funktionen

$$T_m(s) = \cosh(m \operatorname{arcosh} s) = \frac{1}{2}(e^{\sigma m} + e^{-\sigma m}) \tag{2.1.14}$$

$$= \frac{1}{2}(w^m + w^{-m}), \quad w^{\pm 1} = s \pm \sqrt{s^2 - 1}. \tag{2.1.15}$$

Die Größe $w = e^{\sigma}$, $\sigma > 0$, in (2.1.15) ist dabei die größere Lösung der quadratischen Gleichung $\frac{1}{2}(w + w^{-1}) = s$ und w^{-1} die kleinere. Die letzte Ungleichung in (2.1.13) folgt aus der Darstellung (2.1.15), denn für das Argument $\hat{s} = (b + a)/(b - a) = (\kappa + 1)/(\kappa - 1)$, $\kappa = b/a \geq 1$, wird $w = (\kappa + 1 + 2\sqrt{\kappa})/(\kappa - 1)$. Hier kann ein Linearfaktor gekürzt werden und man erhält

$$T_m\Big(\frac{\kappa+1}{\kappa-1}\Big) = \frac{1}{2}\Big[\Big(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\Big)^m + \Big(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\Big)^m\Big] \geq \frac{1}{2}\Big(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\Big)^m. \qquad \blacksquare$$

Der Konvergenz-Vorteil bei Wahl der Relaxationsparameter nach Satz 2.1.2 für einen Zyklus der Länge $m > 1$ gegenüber dem einfachen, stationären Relaxationsverfahren (2.1.2) der Zykluslänge 1 soll jetzt bei $[a, b] = [\lambda_n, \lambda_1]$ analysiert werden. Dann hängt (2.1.13) 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 (2.1.5) 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

$$m \doteq \sqrt{\hat{\kappa}}\ln 2, \qquad \text{bzw} \qquad k \doteq \hat{\kappa}\frac{\ln 2}{2} \cong m^2.$$

Die Richardson-Iteration (2.1.7) mit optimalen Parametern (2.1.12) konvergiert 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. Diese Situation wird im nächsten Abschnitt bei den Krylov-Verfahren diskutiert, wo die in diesem Fall schwierige Eigenwert-Schätzung nicht vor dem Einsatz der Verfahren erforderlich ist, sondern nur für Konvergenzaussagen. Auch in noch allgemeineren Situationen, z.B., bei reellen Eigenwerten mit beiderlei Vorzeichen, kann Richardson-Iteration eingesetzt werden (vgl. Spezialliteratur).

Ein wesentlicher *Nachteil* des Richardson-Verfahrens ist der, dass die Konvergenzaussage (2.1.9), (2.1.13) 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 $\hat{p}_m$ hatte der Fehler (2.1.8) die Form

$$x^{(m)} - z = \hat{p}_m(A)(x^{(0)} - z).$$

Für diese speziellen Polynome gilt wegen (2.1.11) eine zweistufige *Rekursionsformel*

$$\hat{p}_{m+1}(t) = \alpha_m(t - \gamma_m)\hat{p}_m(t) + \beta_m\hat{p}_{m-1}(t), \;\; m = 1, 2, \ldots, \tag{2.1.16}$$

deren Koeffizienten später ermittelt werden. 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 (2.1.16)

$$\begin{aligned} y^{(k+1)} &= \hat{p}_{k+1}(A)\,(x^{(0)} - z) + z \\ &= \alpha_k(A - \gamma_k I)\,\hat{p}_k(A)\,(x^{(0)} - z) + \beta_k\hat{p}_{k-1}(A)\,(x^{(0)} - z) + z \\ &= \alpha_k(A - \gamma_k I)(y^{(k)} - z) + \beta_k(y^{(k-1)} - z) + z \\ &= \alpha_k(Ay^{(k)} - r) - \alpha_k\gamma_k y^{(k)} + \beta_k y^{(k-1)} + \underbrace{(1 - \beta_k + \alpha_k\gamma_k)}_{=0} z. \end{aligned}$$

Der Vorfaktor bei der unbekannten Lösung z verschwindet dabei, da $\hat{p}_k(0) = 1\,\forall k$, vgl. (2.1.8), denn dies entspricht genau der Beziehung $1 = \beta_k - \alpha_k\gamma_k$ in der Rekursion (2.1.16). 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 (2.1.16) der Polynome (2.1.10) ergeben sich mit

$$q := \frac{b + a}{b - a}, \quad d := \frac{2}{b + a}, \quad \varrho_k := 2q\frac{T_k(q)}{T_{k+1}(q)} \tag{2.1.17}$$

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 (2.1.11) 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 (2.1.11) noch einmal verwendet. Mit diesen Bezeichnungen wird nun die *Tschebyscheff-Iteration* formuliert. Wegen der Norm-Konvergenzaussage (2.1.19) wird dabei nur der symmetrische Fall betrachtet.

Satz 2.1.3 *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

$$\left\{ \begin{array}{rcl} y^{(1)} & := & y^{(0)} + d(r - Ay^{(0)}), \\ y^{(k+1)} & := & \varrho_k \Big(y^{(k)} + d(r - Ay^{(k)}) \Big) + (1-\varrho_k) y^{(k-1)}, \quad k = 1,2,\ldots, \end{array} \right. \tag{2.1.18}$$

mit Startvektor $y^{(0)} \in \mathbb{R}^n$ und den in (2.1.17) definierten Parametern gilt die Fehlerschranke

$$\|y^{(k)} - z\|_2 \le \frac{1}{T_k(q)} \|y^{(0)} - z\|_2 \le 2 \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} \right)^k \|y^{(0)} - z\|_2, \ k \ge 1. \tag{2.1.19}$$

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. (2.1.13). ∎

Bemerkung: Die Tschebyscheff-Iteration (2.1.18) besteht offensichtlich aus einer zusätzlichen linearen Extrapolation ($1 < \varrho_k \le 2$) bei der einfachen, optimalen Relaxation (2.1.2), (2.1.3):

$$\begin{array}{rcl} \tilde{y}^{(k+1)} & := & y^{(k)} + \dfrac{2}{a+b}(r - Ay^{(k)}), \\ y^{(k+1)} & = & \varrho_k \tilde{y}^{(k+1)} + (1-\varrho_k) y^{(k-1)}. \end{array}$$

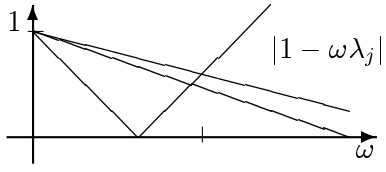
Praktische Durchführung

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 §1 bietet sich dazu die Vektor-Iteration an. Diese kann sogar ohne zusätzlichen Aufwand mit der einfachen Relaxation (2.1.2) 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,

$$d^{(k+1)} := r - Ax^{(k+1)} = r - A(x^{(k)} + \omega d^{(k)}) = (I - \omega A) d^{(k)} \tag{2.1.20}$$

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

$$\varrho(I - \omega A) = \left\{ \begin{array}{lll} \omega\lambda_1 - 1 & \text{für} & \omega > \\ 1 - \omega\lambda_n & \text{für} & \omega \le \end{array} \right\} \hat{\omega} = \frac{2}{\lambda_n + \lambda_1}.$$



erinnert. Durch Variation von ω können daher λ_n und λ_1 bei (2.1.20) geschätzt werden:

a) Für $\omega := 1/\|A\| \le 1/\lambda_1$ ist $\varrho(I-\omega A) = 1-\omega\lambda_n > 0$. Analog zu (1.4.3) bekommt man aus (2.1.20) 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 (2.1.20) 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* lassen sich Richardson- und Tschebyscheff-Methode auch zur Beschleunigung der Vektor-Iteration einsetzen. Statt der Folge (1.3.2)

$$v^{(m)} := A^m z^{(0)} \quad \text{wird betrachtet } v^{(m)} := p_m(A)z^{(0)},\ p_m \in \Pi_m.$$

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 jetzt eine Schätzung für λ_2 benötigt. Bei Verwendung der optimalen Tschebyscheff-Polynome verbessert sich die Konvergenz von

$$\Big|\frac{\lambda_2}{\lambda_1}\Big|^m \quad \text{auf} \quad 1\Big/\Big|T_m\Big(2\frac{\lambda_1}{\lambda_2}-1\Big)\Big| \le \Big|\frac{\lambda_2}{\lambda_1}\Big|^m \frac{2}{(1+\sqrt{1-\lambda_2/\lambda_1})^{2m}}$$

Prinizipiell stellt die Erfordernis guter Eigenwert-Schätzungen ein Hindernis für den praktischen Einsatz der Tschebyscheff-Iteration dar. Ein ähnliches Verfahren mit "eingebauter" Parameterbestimmung wird als Spezialfall der folgenden Krylov-Verfahren auftreten. Bei unsymmetrischen Matrizen sind diese neuen Verfahren aber teurer als die Tschebyscheff-Iteration.

2.2 Krylov-Verfahren

Die beiden letzten Abschnitte haben gezeigt, dass die Betrachtung von Matrix-Polynomen $p(A)$ weitreichende Hilfsmittel erschließt. Zur Verminderung des Rechenaufwand wird aber $p(A)$ nur als Vorfaktor $p(A)q$ auf einen festen Vektor $q \in \mathbb{R}^n$ angewendet. Da jedes Polynom Linearkombination von Potenzen A^k ist, ist das Produkt $p(A)q$ daher ein Element des linearen Unterraums

$$\mathcal{K}_m(A,q) := \operatorname{span}\{q, Aq, A^2q, \ldots, A^{m-1}q\}, \quad m \ge 1, \tag{2.2.1}$$

den man *Krylov-Raum* nennt. Die erzeugenden Elemente dieses Raums sind übrigens genau die Vektoren $v^{(k)} = A^k q$ aus der Vektor-Iteration (1.3.1). Dabei wird aber die in den Krylov-Vektoren $v^{(k)}$ steckende Information nur zu einem geringen Teil ausgenutzt, wie sich im Folgenden zeigt.

Die gesamte in $\mathcal{K}_m(A,q)$ enthaltene Information bewahrt man durch Berechnung eine Orthonormalbasis $U_m \in \mathbb{R}^{n\times m}$ von $\mathcal{K}_m$. In diesem Unterraum wird die Abbildung A durch die Matrix

$$H_m := U_m^{\mathsf{T}} A U_m \in \mathbb{R}^{m\times m}, \quad R(U_m) = \mathcal{K}_m(A,q), \quad m = 1,2,\ldots, \tag{2.2.2}$$

repräsentiert. Aufgrund der (später gezeigten) guten Approximationseigenschaften von Krylovräumen kann man nun die Matrix H_m bzw. die Rang-m-Matrix $U_m H_m U_m^{\mathsf{T}} \in \mathbb{R}^{n\times n}$ als "Modell" für die Gesamt-Matrix A sowohl bei Gleichungssystemen als auch Eigenwertproblemen einsetzen.

Das folgende Verfahren führt die Berechnung von U_m und H_m effizient durch, indem der Aufbau des Krylovraums geschickt mit der Gram-Schmidt-Orthogonalisierung verschränkt wird. Die Matrizen werden unten indiziert, da sie schrittweise aufgebaut werden und sich die Anfangsspalten bei Vergrößerung von m nicht ändern. Beide Matrizen sind daher Ausschnitte von größeren Matrizen, etwa bei $U_{m+1} = (U_m, u^{(m+1)})$. Wegen (2.2.2) ist $(q, Aq, \ldots, A^{m-1}q) = U_m R_m$ eine QR-Zerlegung.

Satz 2.2.1 (Arnoldi-Verfahren) *Gegeben sei $A \in \mathbb{R}^{n\times n}$ und $q \in \mathbb{R}^n \setminus \{0\}$. Beginnend mit $d := q$ und $h_{10} := \|q\|_2$ wird für $j = 1,2,\ldots$ und solange $h_{j,j-1} \neq 0$ ist, berechnet:*

$$\begin{aligned} u^{(j)} &:= d/h_{j,j-1} \\ v &:= Au^{(j)}, \quad h_{ij} := v^{\mathsf{T}} u^{(i)}, \ i = 1,\ldots,j \\ d &:= v - \sum_{i=1}^{j} u^{(i)} h_{ij}, \quad h_{j+1,j} := \|d\|_2. \end{aligned} \tag{2.2.3}$$

Wenn dieser Algorithmus über m Schritte durchführbar war, dann gelten für die Matrizen $U_m := (u^{(1)},\ldots,u^{(m)})$ und $H_m := (h_{ij})_{i,j=1}^m$ folgende Aussagen:

$$R(U_m) = \mathcal{K}_m(A,q), \quad U_m^{\mathsf{T}} U_m = I_m, \quad H_m = U_m^{\mathsf{T}} A U_m \text{ ist Hessenberg-Matrix},$$

und es gilt

$$AU_m = U_m H_m + h_{m+1,m} u^{(m+1)} e^{(m)\mathsf{T}}. \tag{2.2.4}$$

Beweis Der j-te Schritt im Verfahren (2.2.3) kann zusammengefaßt werden zur Identität

$$h_{j+1,j} u^{(j+1)} = Au^{(j)} - \sum_{i=1}^{j} u^{(i)} h_{ij}, \quad h_{ij} := u^{(i)} A u^{(j)}. \tag{2.2.5}$$

Daran sind die Aussagen direkt abzulesen. Multiplikation von (2.2.5) mit $u^{(k)\mathsf{T}}$, $k \le j$, ergibt

$$h_{j+1,j}\, u^{(k)\mathsf{T}} u^{(j+1)} = h_{kj} - \sum_{i=1}^{j} u^{(k)\mathsf{T}} u^{(i)} h_{ij}, \quad \text{da } u^{(k)\mathsf{T}} A u^{(j)} = h_{kj}.$$

Unter der Induktionsvoraussetzung $u^{(k)\mathsf{T}} u^{(i)} = \delta_{ki}$, $k,i \le j$, folgt daraus die Orthogonalität $u^{(k)\mathsf{T}} u^{(j+1)} = 0$, $k \le j$. Die Normierung ist $\|u^{(j+1)}\|_2 = 1$ (wenn $h_{j+1,j} \neq 0$) und für $k = j+1$ gilt tatsächlich $h_{j+1,j}\|u^{(j+1)}\|_2^2 = u^{(j+1)\mathsf{T}} A u^{(j)} - 0$.

Des weiteren folgt aus (2.2.5), dass $u^{(j+1)} \in A\mathcal{K}_j(A,q)+\mathcal{K}_j(A,q) \subseteq \mathcal{K}_{j+1}(A,q),\ j \le m$ gilt (dies ist das Konstruktionsprinzip). Daher ist $R(U_m) \subseteq \mathcal{K}_m(A,q)$. Die ONB U_m spannt aber einen m-dimensionalen Raum auf und daher sind beide Unterräume gleich. Als letzten Schritt schreibt man (2.2.5) in der Form $Au^{(j)} = (AU_m)e^{(j)} = \sum_{i=1}^{j+1} u^{(i)}h_{ij}$, $j \le m$. Diese Identität verifiziert die Darstellung (2.2.4) spaltenweise, $1 \le j \le m$, und dabei ist $h_{ij} = 0$ für $j < i-1$, also $H_m = (h_{ij})$ Hessenbergmatrix. ∎

Bemerkung: a) Der Aufwand für das Arnoldiverfahren besteht aus einer Matrix-Vektor-Multiplikation pro Schritt und dem Aufwand der Orthogonalisierung ($4jn$ Operationen in Schritt j). Bis zum Schritt m sind das m Matrix-Vektor-Multiplikationen und $2m^2n$ Operationen. Bei großen, dünnbesetzten Matrizen wird daher der Aufwand mit wachsendem m von der Orthogonalisierung dominiert.

b) Aus Nutzersicht haben Krylov-Verfahren in der Praxis den Vorteil, "Matrix-frei" zu arbeiten. Da A nur über die Produkte Au in (2.2.3) verwendet wird, muß in Anwendungen tatsächlich nur die lineare Abbildung $u \mapsto Au$ implementiert werden und nicht deren Matrixdarstellung $A = (a_{ij})$. Besonders vorteilhaft ist das etwa bei nichtlinearen Problemen $f(y) = 0$, wo im Newtonverfahren lineare Systeme mit der Ableitung $A = f'(y)$ zu lösen sind. Hier wird gerne eine Differenzenapproximation verwendet,

$$f'(y)u \cong \frac{1}{h}\Big(f(y+hu) - f(y)\Big),$$

mit geeignetem $h > 0$, wodurch man die Programmierung von f' überhaupt umgeht.

c) Es ist a priori nicht klar, bis zu welcher Dimension $m \le n$ das Arnoldiverfahren durchführbar ist. Wenn das Verfahren aber bei $j = m$ abbricht mit $h_{m+1,m} = \|d\|_2 = 0$, dann gilt in (2.2.4) $AU_m = U_mH_m$. Damit wurde aber ein *invarianter Unterraum* von A gefunden und die Eigenwerte von H_m sind auch Eigenwerte von A. Wenn A sogar nur Rang m besitzt, wurde damit die Zerlegung

$$\text{Rang}(A) = m \quad \Rightarrow \quad A = U_mH_mU_m^{\mathsf{T}} \tag{2.2.6}$$

berechnet. Der vermeintlich ungünstige Fall eines vorzeitigen Abbruchs kann bei Gleichungssystemen durch eine geschickte Wahl ins Gegenteil verkehrt werden, s.u. (*lucky breakdown*).

Die Identität (2.2.6) kann auch für $h_{m+1,m} \neq 0$ als Motivation von Krylov-Näherungsverfahren dienen für verschiedene Probleme, in der die Matrix A auftritt, indem man dort die Matrix A durch ihr Rang-m-Modell $U_mH_mU_m^{\mathsf{T}} \cong A$ ersetzt. Bei Linearen Gleichungssystemen $Ax = r$ betrachtet man dann die Näherung $x^{(m)} \in \mathbb{R}^n$ aus

$$U_mH_mU_m^{\mathsf{T}}x^{(m)} = r \quad \longrightarrow \quad H_m(U_m^{\mathsf{T}}x^{(m)}) = U_m^{\mathsf{T}}r. \tag{2.2.7}$$

Für reguläres H_m ist zwar das rechte System immer lösbar, das linke aber nur für $r \in \mathcal{K}_m(A,q)$. Letzteres läßt sich trivialerweise durch den Startvektor $q := r$ erzwingen. Eine andere Interpretation von (2.2.7) ist, dass man mit dem Ansatz $x^{(m)} = U_my$, $y \in \mathbb{R}^m$, in das LGS $Ax = r$ geht.

Mit (2.2.4) und $R(U_m) = \mathcal{K}(A,r)$ und $\beta := r^\mathsf{T} u^{(1)} = \|r\|_2$ liest sich der Defekt dieses Problem als

$$\begin{aligned} r - Ax^{(m)} &= \beta u^{(1)} - AU_m y = U_m(\beta e^{(1)} - H_m y) - h_{m+1,m} u^{(m+1)} y_m \\ &= U_{m+1} \begin{pmatrix} \beta e^{(1)} - H_m y \\ -h_{m+1,m} y_m \end{pmatrix}. \end{aligned} \tag{2.2.8}$$

Statt, wie in (2.2.7) den unteren Teil des Defekts zu vernachlässigen, kann man die Kleinste-Quadrate-Lösung von $AU_m y - r = 0$ bestimmen, indem man die Norm des Gesamtdefekts (2.2.8) minimiert. Dies führt auf das folgende Verfahren (Generalized Minimum RESiduum method).

Satz 2.2.2 (GMRES-Verfahren) *a) Zu dem Linearen Gleichungssystem $Ax = r$ mit regulärer Matrix $A \in \mathbb{R}^{n\times n}$ werden Näherungen $x^{(m)} \in \mathbb{R}^n$ definiert durch*

$$\delta_m := \|r - Ax^{(m)}\|_2 = \min\{\, \|r - Au\|_2 :\ u \in \mathcal{K}_m(A,r)\}, \quad m = 1,2,\ldots.$$

Dann ist $(\delta_m)_m$ eine endliche, monoton gegen null fallende Folge mit $\delta_{\bar m} = 0$, $\bar m \le n$. Es gilt

$$\delta_m = \min\{\, \|p(A)r\|_2 :\ p \in \Pi_m,\ p(0) = 1\}. \tag{2.2.9}$$

b) Die Näherungen $x^{(m)} = U_m y^{(m)}$ können als Kleinste-Quadrate-Lösungen mit einer QR-Zerlegung der Matrix

$$\bar H_m := \begin{pmatrix} H_m \\ 0..0\, h_{m+1,m} \end{pmatrix} = Q_m \bar R_m = Q_m \begin{pmatrix} R_m \\ 0^\mathsf{T} \end{pmatrix},\ Q_m \in \mathbb{R}^{(m+1)\times(m+1)},\ R_m \in \mathbb{R}^{m\times m},$$

berechnet werden, denn R_m ist regulär. Mit dem Vektor $\beta Q_m^\mathsf{T} e^{(1)} =: \bar g = (g^\mathsf{T}, \gamma_{m+1})^\mathsf{T}$ sind die Koeffizienten und der Defekt der Lösung gegeben durch

$$y^{(m)} = R_m^{-1} g, \quad \delta_m = \|r - Ax^{(m)}\|_2 = |\gamma_{m+1}|. \tag{2.2.10}$$

Beweis a) Die Monotonie $\delta_m \searrow$ folgt allein aus der Definition, da der Unterraum $\mathcal{K}_m$ mit wachsendem m nicht kleiner wird. Da jedes Element $u \in \mathcal{K}_m(A,r)$ Linearkombination von Vektoren $A^j r$ ist, gilt $u = p_{m-1}(A)r$, $p_{m-1} \in \Pi_{m-1}$. Der Defekt ist also $r - Au = r - Ap_{m-1}(A)r = p_m(A)r$ mit dem Polynom $p_m(z) = 1 - z p_{m-1}(z)$. Dies zeigt (2.2.9). Es sei nun $\bar m \le n$ die maximale Dimension der Unterraum-Folge $\mathcal{K}_m$ mit $\mathcal{K}_{\bar m} = \mathcal{K}_{\bar m - 1}$ bzw. $A^{\bar m - 1} r \in \mathcal{K}_{\bar m - 1}$. Dann gilt aber mit einem Polynom $q \in \Pi_{\bar m - 2}$ dass $A^{\bar m - 1} r = q(A)r$ ist. Für das Polynom $p(z) := z^{\bar m - 1} - q(z)$ wird dann aber in (2.2.9) der Minimalwert $\delta_{\bar m} = 0$ angenommen, die zugehörige Lösung ist exakt: $x \in \mathcal{K}_{\bar m}(A,r)$.

b) Die Darstellung (2.2.4) läßt sich kompakt in der Form $AU_m = U_{m+1}\bar H_m$ schreiben. Wegen der Orthogonalität von U_{m+1} und mit $r = \beta u^{(1)}$ gilt für den Defekt

$$\begin{aligned} \|r - AU_m y\|_2^2 &= \|U_{m+1}(\beta e^{(1)} - \bar H_m y)\|_2^2 = \|\beta e^{(1)} - Q_m \bar R_m y\|_2^2 = \|\bar g - \bar R_m y\|_2^2 \\ &= \|g - R_m y\|_2^2 + \gamma_{m+1}^2. \end{aligned} \tag{2.2.11}$$

Für eine reguläre Matrix A hat AU_m vollen Rang m und somit auch $\bar{R}_m = Q_m^{\mathsf{T}} A U_m$. Die quadratische Untermatrix R_m ist daher regulär und (2.2.11) nimmt sein eindeutiges Minimum in $R_m^{-1} g$ an. Der Minimalwert ist γ_{m+1}^2. ∎

Der Satz 2.2.2 beschreibt in Teil a) den theoretischen Hintergrund von GMRES, wobei die Defektdarstellung (2.2.9) mit Polynomen später Basis für eine Konvergenzanalyse sein wird. In Teil b) wird die praktische Durchführung beschrieben. Dazu gilt weiter:

- Parallel zur Aufstellung der Basen $U_1, U_2, \ldots$ im Arnoldi-Verfahren kann die QR-Zerlegung der Matrix H bis zur Spalte m, also von $\bar{H}_m$, berechnet werden. Wegen der Hessenbergform verändert die m-te Householder-Spiegelung bei $\bar{H}_m$ nur die beiden letzten Zeilen $m, m+1$. Dies betrifft auch die mittransformierte rechte Seite $\bar{g}$. Daher bleiben die ersten m Spalten bei $\bar{R}_m$ und bei $\bar{g}$ die ersten m Elemente ab Schritt $m+1$ unverändert.

- Direkt im Anschluß an die m-te Spiegelung kann nach (2.2.10) der *Defekt* $\delta_m = |\gamma_{m+1}|$ bestimmt werden, ohne die Lösung $x^{(m)}$ überhaupt zu kennen. Damit kann ein Abbruchkriterium $\delta_m \overset{!}{\leq} tol$ ohne Zusatzkosten geprüft werden.

- Die Näherungen $x^{(1)}, x^{(2)}, \ldots$ werden daher nicht fortlaufend bestimmt, erst zum Schluß wird nach dem Abbruch einmal die letzte Näherung $x^{(m)} = U_m R_m^{-1} g$ berechnet.

- Ein vorzeitiger Abbruch des Arnoldi-Verfahrens wegen $h_{\bar{m}+1,\bar{m}} = 0$ entspricht dem in Teil a) behandelten Fall. Dann entfällt die letzte Householder-Spiegelung und die letzte Zeile von Q_m ist der Einheitsvektor ${e^{(m+1)}}^{\mathsf{T}}$, also ist $\gamma_{m+1} = 0$. Daher ist auch der Defekt $\delta_{\bar{m}} = |\gamma_{\bar{m}+1}| = 0$ und $x^{(\bar{m})}$ schon die exakte Lösung ("lucky breakdown").

Nach diesen Bemerkungen kann das GMRES-Verfahren effizient durchgeführt werden, solange die Dimension m des Krylovraums nicht zu groß wird. Um eine Vorstellung von den erforderlichen Größenordnungen zu bekommen ist eine Analyse der Konvergenzeigenschaften erforderlich. Hier kann man an der Minimalbedingung (2.2.9) ansetzen und analog zur Richardson-Iteration (2.1.8) "gute Polynome" konstruieren. Allerdings muß man beachten, dass sich Aussagen über die Defektnorm δ_m nur mit Vorsicht auf den Fehler

$$A^{-1} r - x^{(m)} = A^{-1}(r - A x^{(m)}) \quad \Rightarrow \quad \|A^{-1} r - x^{(m)}\|_2 \leq \|A^{-1}\|_2 \; \delta_m$$

übertragen. Bei einer symmetrischen Matrix bekommt man zwar einfache Schranken von der Form (2.1.13) für δ_m. Für diesen Fall gibt es aber einen effizienteren Vorgänger des GMRES-Verfahrens, der im nächsten Abschnitt behandelt wird.

Das wichtigste Einsatzgebiet von GMRES ist der unsymmetrische Fall mit evtl. komplexen Eigenwerten. Zur theoretischen Abschätzung der Minimalschranke (2.2.9), die mit jedem eingesetzten Polynom p_m eine obere Schranke liefert, konzentriert man sich wieder auf die Eigenwerte des Matrixpolynoms, d.h. auf $\max_i |p(\lambda_i)|$. Je nach Annahme an die Matrix kann man

dazu verschiedene Aussagen herleiten, einen gößeren Problemkreis deckt man aber wieder mit Tschebyscheff-Polynomen T_m ab. Dazu ist zunächst das Verhalten dieser Polynome $T_m(z)$ im Komplexen, $z \in \mathbb{C}$, zu untersuchen. Dazu eignet sich die Darstellung (2.1.15) mit $\cosh(\zeta) = z$ und $T_m(z) = \cosh(m\zeta) = \frac{1}{2}(e^{m\zeta} + e^{-m\zeta}) = \frac{1}{2}(w^m + w^{-m})$. Für reelle $z = s$, $|s| > 1$, waren $w = s + \sqrt{s^2 - 1}$ und $w^{-1} = s - \sqrt{s^2 - 1}$ die beiden Lösungen der Gleichung $w + w^{-1} = 2s$. Diese Darstellung läßt sich direkt ins Komplexe übertragen. Eine wichtige Rolle spielt dabei die Abbildung $J(w) := \frac{1}{2}(w + w^{-1})$. Damit lautet die allgemeine Darstellung der Tschebyscheff-Polynome

$$T_m(z) = J(w^m) = \frac{1}{2}(w^m + w^{-m}), \quad \text{wobei} \quad J(w) = \frac{1}{2}(w + w^{-1}) = z \in \mathbb{C}. \tag{2.2.12}$$

Die Gleichung $\frac{1}{2}(w+w^{-1}) = z$ hat jeweils zwei Lösungen, welche zueinander invers sind, $w_1 w_2 = 1$. Daher liefern beide in $J(w^m)$ den gleichen Wert. Für große m dominiert in der Summe $\frac{1}{2}(w^m+w^{-m})$ der betragsgrößere Summand, es gilt $|T_m(z)| \geq \frac{1}{2}(|w|^m-1)$ mit der Lösung $|w| > 1$. Für das Verhalten der Tschebyscheffpolynome ist daher der Betrag der Funktion $z \mapsto w = J^{-1}(z)$ entscheidend, also der Umkehrabbildung von J. Werte auf einem Kreis $\{w : |w| = \rho > 0\}$ führen zu gleichem asymptotischem Verhalten von $|T_m|$, $m \to \infty$, insbesondere für $\rho > 1$. Das Bild eines solchen Kreises unter J ist eine *Ellipse* in der z-Ebene. Die Zusammenhänge lassen sich im folgenden Diagramm darstellen

$$\begin{array}{ccccc} z & \longrightarrow & w & \longrightarrow & T_m(z) \\ & & & & \\ \boxed{\text{Ellipsen}} & \xleftarrow[J(w)]{} & \boxed{\text{Kreise}} & \xrightarrow[J(w^m)]{} & \boxed{\text{Kreise asympt.}} \end{array}$$

Eine Ellipse mit Mittelpunkt μ und horizontaler bzw. vertikaler Halbachse α, β wird beschrieben durch

$$E(\mu, \alpha, \beta) := \{z \in \mathbb{C} : \ (\mathrm{Re}\,(z - \mu)/\alpha)^2 + (\mathrm{Im}\,(z - \mu)/\beta)^2 = 1\}.$$

Satz 2.2.3 *Für $\rho > 0$ wird der Kreis $\{w \in \mathbb{C} : |w| = \rho\}$ durch die Abbildung $J(w) = \frac{1}{2}(w+w^{-1})$ abgebildet auf die Ellipse $E(0, \alpha, \beta)$ (in der z-Ebene) mit*

$$\alpha = \frac{1}{2}(\rho + \frac{1}{\rho}) = J(\rho), \quad \beta = \frac{1}{2}|\rho - \frac{1}{\rho}|.$$

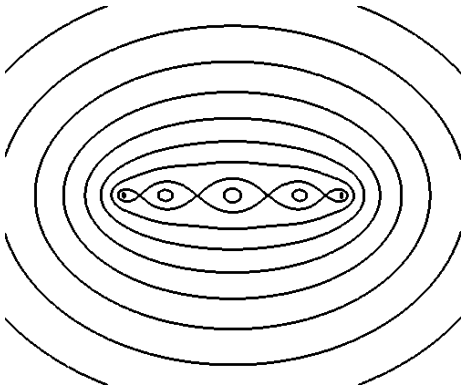
Beweis Mit $w = \rho e^{i\phi}$, $\phi \in [0, 2\pi)$ wird der Kreis parametrisiert. Hieraus folgt

$$J(w) = \frac{1}{2}(\rho e^{i\phi} + \frac{1}{\rho} e^{-i\phi}) = \frac{1}{2}(\rho + \frac{1}{\rho})\cos\phi + i\frac{1}{2}(\rho - \frac{1}{\rho})\sin\phi$$

Dies ist die Parameterdarstellung der angegebenen Ellipse $E(0, \alpha, \beta)$. ∎

In der Definition ist α die horizontale und $\beta < \alpha$ die vertikale Halbachse der Ellipse $E(\alpha, \beta)$, die Brennpunkte im Satz sind ± 1. Kreise mit Radius ρ und $1/\rho$ werden auf die gleiche Ellipse

abgebildet. Im Grenzübergang $\rho \to 1$ wird aus der Ellipse das reelle Intervall $[-1, 1]$.
Im Bild rechts sind die Höhenlinien von $|T_5(z)|$ in $[-2, 2] \times [-2, 2]$ gezeigt. Die Höhenlinie $|T_5| = 1$ ist das knotenförmige Gebilde und hat keine Ähnlichkeit mit Ellipsen. Für wachsenden Betrag $r > 1$ ähneln die Höhenlinien $|T_5| = r$ aber immer mehr den Ellipsen aus Satz 2.2.3. Der Radius $\rho > 1$ des Bildkreises unter J^{-1} kann direkt aus dem Achsenverhältnis der Ellipse E berechnet werden. Aus Satz 2.2.3 folgt für $\alpha > \beta$



$$\frac{\beta}{\alpha} = \frac{\rho^2 - 1}{\rho^2 + 1} \quad \Longleftrightarrow \quad \rho = \sqrt{\frac{\alpha/\beta + 1}{\alpha/\beta - 1}}. \tag{2.2.13}$$

Kreise ($\alpha = \beta$) fallen hier aus dem Rahmen ($\rho \to \infty$). Man kann zeigen, dass bei Verteilung des Spektrums von A in einem Kreis mit Mittelpunkt μ das Polynom $p_m(z) = (1 - z/\mu)^m$ kleinste Norm besitzt. In diesem Fall ist also das einfache Relaxationsverfahren (2.1.2) mit festem Parameter $\omega = 1/\mu$ optimal.

Im Fall $\alpha > \beta$ kann Satz 2.2.3 analog zu Satz 2.1.2 zur Konstruktion eines speziellen Polynoms p_m verwendet werden, welches in der Defektdarstellung (2.2.9) einen kleinen Wert ergibt. Dazu soll p_m auf den Eigenwerten von A möglichst kleine Werte im Vergleich zum Normierungswert $p_m(0) = 1$ annehmen. Der folgende Satz enthält dabei keine Optimalitätsaussage mehr, die angegebenen (Tschebyscheff-) Polynome sind nur noch asymptotisch optimal ($m \to \infty$).

Satz 2.2.4 *Die Eigenwerte der Matrix $A \in \mathbb{R}^{n\times n}$ seien in der Ellipse $E(\mu, \alpha, \beta)$, $\mu \in \mathbb{R}$, enthalten und es sei $|\mu| > \alpha > \beta$, d.h. $0 \notin E(\mu, \alpha, \beta)$. Mit $\gamma := \sqrt{\alpha^2 - \beta^2}$ gilt dann für das Polynom*

$$p_m(z) := \frac{T_m((z - \mu)/\gamma)}{T_m(-\mu/\gamma)}$$

die Schranke

$$\max_{z\in E} |p_m(z)| \le 2\Big(\frac{\alpha + \beta}{|\mu| + \sqrt{\mu^2 - \alpha^2 + \beta^2}}\Big)^m.$$

Beweis Mit der verwendeten Variablentranformation ändern sich in (2.2.12) die Definition für w zu $z - \mu = \gamma J(w)$. Für die Achsen der Ellipse $E(\mu, \alpha, \beta)$ folgt aus Satz 2.2.3 der Zusammenhang $\alpha = \gamma J(\rho)$, $\beta = \frac{\gamma}{2}|\rho - 1/\rho|$, das Verhältnis bleibt also unverändert und ergibt mit (2.2.13) den Wert ρ. Mit diesem folgt die Gestalt von $\gamma = \alpha/J(\rho) = \sqrt{\alpha^2 - \beta^2}$. Mit den jeweils betragsgrößeren Lösungen w aus $\gamma J(w) = z - \mu$ und $w_0 \in \mathbb{R}$ aus $\gamma J(w_0) = -\mu$ folgt aus (2.2.12)

$$|p_m(z)| = \frac{|w^m + w^{-m}|}{|w_0^m + w_0^{-m}|} \le 2\Big(\frac{\rho}{|w_0|}\Big)^m.$$

Im Zähler wurde nach oben abgeschätzt mit $|w| = \rho$ und im Nenner der betragskleinere Wert

vernachlässigt. Mit der Darstellung (2.2.13) von ρ und $|w_0| = (|\mu| + \sqrt{\mu^2 - \gamma^2})/\gamma$ folgt nun

$$\gamma\rho = \sqrt{\alpha^2 - \beta^2}\sqrt{\frac{\alpha+\beta}{\alpha-\beta}} = \alpha + \beta \text{ und } \quad \frac{\rho}{|w_0|} = \frac{\gamma\rho}{|\mu| + \sqrt{\mu^2-\gamma^2}} = \frac{\alpha+\beta}{|\mu| + \sqrt{\mu^2-\gamma^2}} < 1. \quad \blacksquare$$

Mit komplexen Werten μ, γ können auch andere Ellipsen behandelt werden. Entscheidend für die Güte der Konvergenz bei Krylov-Verfahren ist überraschenderweise die Exzentrizität des Gebiets, in dem die Eigenwerte liegen. Im Extremfall einer Linie (kleine Halbachse $\beta = 0$, $\alpha = (b-a)/2$, $\mu = (a+b)/2$) kann die in Satz 2.1.2 gezeigte maximale Beschleunigung gegenüber der einfachen Relaxation erreicht werden. Bei einer kreisförmigen Verteilung ($\alpha = \beta$) der Eigenwerte ist dagegen keine Verbesserung möglich.

Krylovverfahren haben den entscheidenden Vorteil, dass sie in allen Fällen selbsttätig die optimalen Parameter verwenden, allerdings erkauft durch einen im Vergleich zu den Verfahren aus §2.1 viel höheren Rechenaufwand. Es gibt einige Ansätze, den mit $O(m^2 n)$ wachsenden Rechenaufwand zu vermindern. Dazu gehren ein GMRES-Neustart nach jeweils m Iterationen (GMRES(m)), oder eine nur teilweise Orthogonalisierung in(2.2.3). Diesen Verfahren fehlt aber das sichere theoretische Fundament von GMRES.

Lanczos-Verfahren und Konjugierte Gradienten

Die Effizienz der Krylov-Verfahren wächst dramatisch im symmetrischen (und definiten) Fall. Die zugehörigen Verfahren sind historisch allerdings früher entstanden. Wie in §1.2 ist bei symmetrischer Matrix A die Hessenberg-Matrix $H_m = U_m^{\mathsf{T}} A U_m$ ebenfalls symmetrisch, also tridiagonal. Mit $\alpha_j := h_{jj}$, $\gamma_j := h_{j+1,j} = h_{j,j+1}$ verkürzt sich die Orthogonalisierungsschleife im Arnoldi-Verfahren auf zwei Innenprodukte pro Schritt. Damit ist der Aufwand pro Schritt konstant und wächst daher nur linear mit der Zahl der Schritte. Die Identität (2.2.5) lautet jetzt kürzer

$$\gamma_j u^{(j+1)} = (A - \alpha_j I) u^{(j)} - \gamma_{j-1} u^{(j-1)}$$

und kann zur Konstruktion der Basis U_m dienen. Das zugehörige **Lanczos-Verfahren** lautet:

$$\begin{array}{|l|}\hline d := q;\ \gamma_0 := \|q\|_2;\ u^{(0)} := 0;\ j := 0; \\ \text{solange } \gamma_j \neq 0\text{:} \\ \quad u^{(j+1)} := d/\gamma_j;\ j := j+1;\ \alpha_j := {u^{(j)}}^{\mathsf{T}} A u^{(j)}; \\ \quad d := (A - \alpha_j I) u^{(j)} - \gamma_{j-1} u^{(j-1)}; \\ \quad \gamma_j := \|d\|_2; \\ \hline \end{array} \tag{2.2.14}$$

Mit dieser Tridiagonalmatrix $T_m := H_m$ kann wieder analog zu (2.2.7) eine Näherung $x^{(m)}$ aus dem Krylovraum $\mathcal{K}_m(A,b) = R(U_m)$ berechnet werden. Diese Lösung hat jetzt aber zusätzliche Eigenschaften und läßt sich auch durch eine einfachere Rekursion bestimmen. Zur Herleitung

wird zunächst mit der Basisidentität (2.2.4), $AU_m = U_m T_m + \gamma_{m+1} u^{(m+1)} e^{(m)\mathsf{T}}$, für die Lösung $x^{(m)} = U_m y^{(m)}$ zu (2.2.7), $T_m y^{(m)} = \beta e^{(1)}$ der Defekt betrachtet:

$$d^{(m)} = r - Ax^{(m)} = \beta u^{(1)} - AU_m y^{(m)} = U_m(\underbrace{\beta e^{(1)} - H_m y^{(m)}}_{=0}) - \gamma_{m+1} y_m^{(m)} u^{(m+1)}. \qquad (2.2.15)$$

Also sind die Defekte $d^{(m)}$ in den einzelnen Schritten Vielfache des jeweiligen Basisvektors $u^{(m+1)}$ und daher zueinander orthogonal ("konjugiert"). Der vollständige Name des Verfahrens der "Konjugierten Gradienten" (*conjugate gradients*) beruht darauf, dass im definiten Fall die Defekte auch Gradienten eines bestimmten Funktionals sind.

Satz 2.2.5 *Die Matrix $A \in \mathbb{R}^{n\times n}$ sei symmetrisch und positiv definit. Dann werden durch*

$$(x,y)_A := y^{\mathsf{T}} Ax, \quad \|x\|_A := \sqrt{(x,x)_A} = \sqrt{x^{\mathsf{T}} Ax}, \qquad (2.2.16)$$

$x, y \in \mathbb{R}^n$, ein Innenprodukt und eine Norm definiert. Die Lösung z des linearen Gleichungssystems $Ax = r$ ist eindeutiges Minimum des Funktionals $\varphi : \mathbb{R}^n \to \mathbb{R}$ mit

$$\varphi(x) := \frac{1}{2} x^{\mathsf{T}} Ax - r^{\mathsf{T}} x, \quad x \in \mathbb{R}^n. \qquad (2.2.17)$$

Beweis Wegen der Definitheit von A ist $(x,x)_A = x^{\mathsf{T}} Ax > 0 \ \forall x \neq 0$ und $\|x\|_A$ daher eine Norm. Für φ gilt mit $r = Az$ die Darstellung

$$\begin{aligned} 2\varphi(x) &= x^{\mathsf{T}} Ax - 2r^{\mathsf{T}} x = x^{\mathsf{T}} Ax - 2z^{\mathsf{T}} Ax = (x,x)_A - 2(x,z)_A \\ &= \|x-z\|_A^2 - \|z\|_A^2. \end{aligned} \qquad (2.2.18)$$

Also ist z eindeutiges Minimum von φ. ∎

Die Funktion φ ist streng konvex. An der Stelle x ist die Richtung des steilsten Abstiegs

$$-\mathrm{grad}\varphi(x) = r - Ax =: d$$

gerade der Defekt von x. Man kann jetzt zusätzlich untersuchen, wie die Krylov-Lösungen der Reihe nach entstehen. Dazu sei $T_m = L_m R_m$ die LR-Zerlegung in Bidiagonalmatrizen(!) und

$$x^{(m)} = U_m y^{(m)} = U_m T_m^{-1} e^{(1)} \beta = \underbrace{U_m R_m^{-1}}_{=:P_m} \underbrace{L_m^{-1} e^{(1)} \beta}_{=:v^{(m)}} = P_m v^{(m)}. \qquad (2.2.19)$$

Der Vektor $v^{(m)}$ enthält einfach die ersten m Komponenten der Lösung der unteren Dreiecksysteme $L_m v^{(m)} = \beta e^{(1)}$, die sich der Reihe nach ergeben durch $v_1 = \beta$, $v_k = -l_{k,k-1} v_{k-1}$, $k \geq 2$, und dann unverändert bleiben. Daher gilt insbesondere

$$x^{(m)} = P_m v^{(m)} = P_{m-1} v^{(m-1)} + v_m p^{(m)} = x^{(m-1)} + v_m p^{(m)}, \qquad (2.2.20)$$

wenn wieder $P_m = (p^{(1)}, \ldots, p^{(m)})$ gesetzt wird. Also enthält die Matrix P_m die "Suchrichtungen" des Verfahrens. Auch für diese gilt aufgrund der Definition $U_m = P_m R_m$ die einfache Rekursion

$$p^{(m)} = (u^{(m)} - p^{(m-1)} r_{m-1,m}) / r_{mm}. \qquad (2.2.21)$$

Diese Suchrichtungen besitzen ebenfalls eine bemerkenswerte Eigenschaft, denn nach Definition ist die folgende Matrix symmetrisch und nach (2.2.19) gilt

$$P_m^{\mathsf{T}} A P_m = (U_m R_m^{-1})^{\mathsf{T}} A U_m R_m^{-1} = (R_m^{-1})^{\mathsf{T}} T_m R_m^{-1} = (R_m^{-1})^{\mathsf{T}} L_m.$$

Das letzte Produkt ist eine symmetrische, untere Dreieckmatrix, also diagonal. Daher gilt für die Suchrichtungen

$$p^{(i)\mathsf{T}} A p^{(j)} = (p^{(i)}, p^{(j)})_A = 0 \quad \text{für } i \neq j,$$

sie sind *A-orthogonal* in dem Innenprodukt (2.2.16).

Mit diesen beiden Eigenschaften, Orthogonalität der Defekte $d^{(m)}$ und A-Orthogonalität der Suchrichtungen $p^{(m)}$ können die Krylov-Lösungen $x^{(m)}$ auch direkt konstruiert werden (mit neuen Koeffizienten). Iterations- und Defektschritt hängen zusammen durch

$$x^{(m)} = x^{(m-1)} + t_m p^{(m)}, \quad d^{(m)} = d^{(m-1)} - t_m A p^{(m)}. \tag{2.2.22}$$

Orthogonalität bei den Defekten erfordert insbesondere

$$d^{(m)} = d^{(m-1)} - t_m A p^{(m)} \perp d^{(m-1)} \Rightarrow \quad t_m = \frac{\|d^{(m-1)}\|_2^2}{(d^{(m-1)}, p^{(m)})_A}.$$

Zur Verminderung des Aufwands vermeidet man gemischte Innenprodukte von d und p. Dazu kann man wegen (2.2.21) den Ansatz für (umskalierte) Suchrichtungen

$$p^{(m)} = d^{(m-1)} + \beta_{m-1} p^{(m-1)} \tag{2.2.23}$$

machen. Daraus folgt die Vereinfachung

$$(A p^{(m)})^{\mathsf{T}} d^{(m-1)} = (A p^{(m)})^{\mathsf{T}} (p^{(m)} - \beta_{m-1} p^{(m-1)}) = (p^{(m)}, p^{(m)})_A,$$

da $(p^{(m)}, p^{(m-1)})_A = 0$ ist. Aus t_m wird daher der Quotient $t_m = \|d^{(m-1)}\|_2^2 / (p^{(m)}, p^{(m)})_A$. Die A-Orthogonalität für $p^{(m+1)}$ liefert $\beta_m = -(d^{(m)}, p^{(m)})_A / (p^{(m)}, p^{(m)})_A$. Das Produkt im Zähler kann noch mit (2.2.22) auf Bekanntes reduziert werden, denn mit $A p^{(m)} = (d^{(m-1)} - d^{(m)})/t_m$ folgt $t_m d^{(m)\mathsf{T}} A p^{(m)} = d^{(m)\mathsf{T}} (d^{(m-1)} - d^{(m)}) = -\|d^{(m)}\|_2^2$. Mit der Formel für t_m führt dies auf den Wert $\beta_m = \|d^{(m)}\|_2^2 / \|d^{(m-1)}\|_2^2$. Damit hat man die Kosten des Verfahrens auf eine Matrix-Vektor-Multiplikation, drei Vektor-Linearkombinationen und zwei Innenprodukte reduziert, Speicherplatz wird nur für drei n-Vektoren benötigt.

Dieses *Verfahren der Konjugierten Gradienten* (CG-Verfahren)wird in (2.2.24) vorgestellt. Seine wichtigsten Eigenschaften folgen in einem Satz, welcher zusätzlich eine wichtige Minimaleigenschaft der Näherungen und eine Fehlerschranke für den praktisch interessantesten Fall $m < n$ angibt. Denn das CG-Verfahren wurde von Hestenes und Stiefel 1952 eigentlich als Alternative zum Gauß-Algorithmus präsentiert, großen Erfolg hatte es später aber als Iterationsverfahren.

CG-Verfahren:

$$
\begin{array}{|l|}
\hline
x^{(0)} := 0;\ d^{(0)} := r;\ p^{(1)} := r;\ m := 0; \\
\text{solange } d^{(m)} \neq 0\text{:} \\
\quad m := m+1; \\
\quad t_m := \|d^{(m-1)}\|_2^2 / {p^{(m)}}^{\mathsf{T}} A p^{(m)}; \\
\quad x^{(m)} := x^{(m-1)} + t_m p^{(m)}; \\
\quad d^{(m)} := d^{(m-1)} - t_m A p^{(m)}; \\
\quad \beta_m := \|d^{(m)}\|_2^2 / \|d^{(m-1)}\|_2^2; \\
\quad p^{(m+1)} := d^{(m)} + \beta_m p^{(m)}; \\
\hline
\end{array}
\qquad (2.2.24)
$$

Satz 2.2.6 *Nach m Schritten mit $d^{(j)} \neq 0, j < m$, gelten für die im CG-Verfahren (2.2.24) berechneten Größen folgende Aussagen.*

a) Die Suchrichtungen $p^{(1)}, \ldots, p^{(m)}$ sind paarweise A-orthogonal, die Defekte $d^{(0)}, \ldots, d^{(m-1)}$ orthogonal und jeweils Basen des Krylovraums $\mathcal{K}_m(A, r)$,

$$[p^{(1)}, \ldots, p^{(m)}] = [d^{(0)}, \ldots, d^{(m-1)}] = \mathcal{K}_m(A, r) = [r, Ar, A^2 r, \ldots, A^{m-1} r].$$

b) Die Näherung $x^{(m)}$ minimiert das Funktional φ (2.2.17) über $\mathcal{K}_m$, es gilt

$$\varphi(x^{(m)}) = \min_{x \in \mathcal{K}_m} \varphi(x).$$

Daher ist $x^{(m)}$ exakt, wenn $z \in \mathcal{K}_m$ gilt, also spätestens für $m = n$.

c) In der A-Norm (2.2.16) gilt mit $q := (\hat{\kappa}(A)+1)/(\hat{\kappa}(A)-1)$ die Fehlerschranke

$$
\begin{aligned}
\|x^{(m)} - z\|_A &= \min\{\, \|g_m(A)(x^{(0)} - z)\|_A :\ g_m \in \Pi_m,\, g_m(0) = 1\} \\
&\leq \frac{1}{T_m(q)} \|x^{(0)} - z\|_A \leq 2 \left(\frac{\sqrt{\hat{\kappa}(A)} - 1}{\sqrt{\hat{\kappa}(A)} + 1} \right)^m \|x^{(0)} - z\|_A. \qquad (2.2.25)
\end{aligned}
$$

Beweis (kurz) a) kann mit der Herleitung und den Definitionen induktiv gezeigt werden.

b) Mit $P_k = (p^{(1)}, \ldots, p^{(k)})$, $k \leq m$, hat jedes $x \in \mathcal{K}_m$ die Darstellung $x = P_m y = P_{m-1} v + t p^{(m)}$, $y = (v^{\mathsf{T}}, t)^{\mathsf{T}} \in \mathbb{R}^m$. Daher gilt

$$\phi(x) = \phi(P_{m-1} v + t p^{(m)}) = \phi(P_{m-1} v) + t \underbrace{(P_{m-1} v)^{\mathsf{T}} A p^{(m)}}_{=0} - t r^{\mathsf{T}} p^{(m)} + \frac{t^2}{2} (p^{(m)}, p^{(m)})_A.$$

Durch die A-Orthogonalität ${P_{m-1}}^{\mathsf{T}} A p^{(m)} = 0$ entfällt aber der Anteil, der die Minimierungen bezgl. v und t koppelt. Daher wird das Minimum von $\phi(P_{m-1} v)$ in $x^{(m-1)}$ und für den Rest in $\hat{t} = r^{\mathsf{T}} p^{(m)} / (p^{(m)}, p^{(m)})_A$ angenommen. Dabei gilt tatsächlich $\hat{t} = t_m$ aus (2.2.24), denn wegen $r = d^{(0)}$ und (2.2.23) ist $r^{\mathsf{T}} p^{(m)} = \beta_{m-1} \beta_{m-2} \cdots \beta_1 \|d^{(0)}\|_2^2 = \|d^{(m-1)}\|_2^2$.

c) Mit (2.2.18) kann Aussage b) auch formuliert werden als $\|x^{(m)} - z\|_A = \min\{\|x - z\|_A : x \in \mathcal{K}_m(A, r)\}$. Wie in Satz 2.2.2 werden Elemente des Krylovraums dargestellt als $x = f_{m-1}(A)r$, $f_{m-1} \in \Pi_{m-1}$. Mit $r = Az$ wird $x - z = f_{m-1}(A)Az - z = -g_m(A)z$, $g_m(t) = 1 - tf_{m-1}(t)$. Das Minimum kann nun durch Einsetzen des Tschebyscheff-Polynoms aus Satz 2.1.3 wie dort abgeschätzt werden, wobei Zusatzhilfsmittel zum Umgang mit der A-Norm erforderlich sind. ■

Der wesentliche Unterschied zwischen CG-Verfahren und Tschebyscheff- bzw. Richardson-Iteration ist im symmetrischen Fall also der, dass bei letzteren die Parameter aus vorab bekannten Eigenwert-Schranken berechnet werden müssen, während das CG-Verfahren alle Parameter selbst bestimmt. Dabei minimieren die Parameter der Tschebyscheff-Iteration die Fehler*schranke* (2.1.9), 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 (2.2.25) ist oft pessimistisch (vgl. Demo-Beispiel am Ende). Die Schranke zeigt aber die alleinige Abhängigkeit von der Kondition $\hat{\kappa}(A)$ und bildet deher den Ansatzpunkt für die im folgenden besprochenen Verbesserungen. Tschebyscheff-, Richardson-Iteration und GMRES-Verfahren sind auch bei unsymmetrischen Matrizen einsetzber, bei GMRES gilt unter den Voraussetzungen von Satz 2.2.6 die gleiche Schranke, allerdings nur für den Defekt δ_m, nicht für den tatsächlichen Fehler (in der A-Norm bei CG).

2.3 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 $Ax = r$ mit Matrix A zu dem äquivalenten mit der Matrix $I - B = I - M^{-1}N = M^{-1}A$, also zum System

$$(M^{-1}A)\, x = M^{-1}r. \tag{2.3.1}$$

Der alten Forderung bei der Fixpunktiteration nach einem möglichst kleinen Konvergenzfaktor $\varrho(M^{-1}N)$ entspricht jetzt aufgrund der Konvergenzaussagen der Sätze 2.1.2, 2.1.3, 2.2.6 die Forderung nach einer möglichst kleinen Kondition $\hat{\kappa}(M^{-1}A)$ (≥ 1). Daher nennt man den Übergang zum System (2.3.1) auch *Vorkonditionierung*.

Bei Anwendung dieses Ansatzes ist aber darauf zu achten, dass die Verfahren bei (2.3.1) mit $M^{-1}A$ noch einsetzbar sind. Bei Richardson- und Tschebyscheff-Iteration wurden (zur Vereinfachung) reelle Eigenwerte gefordert, das CG-Verfahren setzt sogar eine symmetrisch-definite Matrix voraus. Diese Voraussetzungen können durch Umformulierungen erfüllt werden, wenn

$$M \text{ in (2.3.1) symmetrisch, positiv definit ist.} \tag{2.3.2}$$

Dann besitzt M nämlich eine *Cholesky-Zerlegung* $M = LL^{\mathsf{T}}$ mit einer unteren Dreieckmatrix L, die positive Diagonalelemente besitzt (symmetrische LR-Zerlegung). Durch Multiplikation von

(2.3.1) mit L^T ergibt sich das neue Gleichungssystem $L^{-1}Ax = L^{-1}r \iff$

$$(L^{-1}AL^{-1^T})(L^T x) = L^{-1}r$$

für die Hilfsvariable $y = L^T x$. Da die Matrix $L^{-1}AL^{-1^T} = L^T(M^{-1}A)L^{-1^T}$ dieses Systems symmetrisch, positiv definit und ähnlich zu $M^{-1}A$ ist, sind mit (2.3.2) Richardson- und Tschebyscheff-Iteration bei (2.3.1) einsetzbar. Die *vorkonditionierte Tschebyscheff-Iteration* etwa lautet

$$\begin{array}{rcl} M(\tilde{y}^{(k+1)} - y^{(k)}) & := & \frac{2}{\alpha+\beta}(r - Ay^{(k)}) \\ y^{(k+1)} & := & \varrho_k \tilde{y}^{(k+1)} + (1-\varrho_k)y^{(k-1)} \end{array} . \tag{2.3.3}$$

Aus Effizienzgründen 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. Statt aber mit dem umformulierten System mit Matrix $L^{-1}AL^{-1^T}$ zu arbeiten, kann man diese Eigenschaft auch für $M^{-1}A$ durch Änderung des Innenprodukts erreichen. Unter der Voraussetzung (2.3.2) gilt in dem neuen Innenprodukt $(\cdot,\cdot)_M$ die Symmetrie, denn

$$(x, M^{-1}Ay)_M = (y^T AM^{-1})Mx = y^T Ax = y^T MM^{-1}Ax = (M^{-1}Ax, y)_M.$$

In der Fehlerabschätzung (2.2.25) ist dann auch der Wert $\hat{\kappa}(A)$ durch $\hat{\kappa}(M^{-1}A)$ zu ersetzen. Im Verfahren wird ein Zusatzvektor $c^{(m)} = M^{-1}d^{(m)}$ verwendet.

Präkonditioniertes CG-Verfahren:

$x^{(0)} := 0;\ d^{(0)} := r;$ löse $Mc^{(0)} := r;\ p^{(1)} := c^{(0)};\ m := 0;$
solange $d^{(m)} \neq 0$:
  $m := m + 1;$
  $t_m := {d^{(m-1)}}^T c^{(m-1)} / {p^{(m)}}^T A p^{(m)};$
  $x^{(m)} := x^{(m-1)} + t_m p^{(m)};$
  $d^{(m)} := d^{(m-1)} - t_m A p^{(m)};$
  löse $Mc^{(m)} = d^{(m)};$
  $\beta_m := {d^{(m)}}^T c^{(m)} / {d^{(m-1)}}^T c^{(m-1)};$
  $p^{(m+1)} := c^{(m)} + \beta_m p^{(m)};$

(2.3.4)

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

$$A = \begin{pmatrix} & & R \\ & D & \\ L & & \end{pmatrix},$$

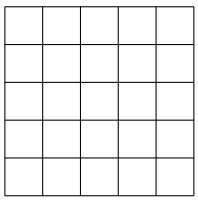
wie beim Einzelschrittverfahren scheidet aus, da $D + L$ nicht symmetrisch ist. Das Einzelschrittverfahren ergab sich bei schrittweiser Auflösung der i-ten Gleichung nach $x_i^{(k+1)}$, $i = 1, \ldots, n$. Wird daran ein umgekehrter Durchlauf $i = n, n-1, \ldots, 1$, angeschlossen, erhält man wegen $R = L^\mathsf{T}$ isngesamt 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

$$M_\omega = (D + \omega L) D^{-1} (D + \omega L^\mathsf{T}). \tag{2.3.5}$$

Das beschriebene Verfahren heißt SSOR ("symmetric successive overrelaxation"). Da dessen Konvergenz auf einem anderen Prinzip beruht als Tschebyscheff- und CG-Iteration, können diese durch Kombination mit SSOR weiter beschleunigt werden.

- Weitere gängige Präkonditionierer basieren auf *unvollständigen LR-Zerlegungen* oder *Gebietszerlegung*. Die Wahl einer guten Vorkonditionierung ist aber i.d.R. problemabhängig und ein reges Forschungsgebiet.

Beispiel 2.3.1 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 (pro Matrix-Vektor-Operation, das präkondit.Tschebyscheff-Verfahren benötigt 2 pro Iterationsschritt, CG-SSOR sogar 3) bei diesem Beispiel für die verschiedenen Verfahren aufgeführt. Außer beim CG-Verfahren, welches keine Zusatzangaben benötigt, wurden bei den Verfahrensparametern exakte Eigenwertangaben verwendet, im praktischen Einsatz ist das natürlich unrealistisch. In der ersten Zeile der Tabelle stehen Approximationsformeln (für große m) der theoretisch hergeleiteten Schranken, in

den folgenden Zeilen die bei zwei verschiedenen Problemgrößen praktisch beobachteten Faktoren (gemittelt über maximal 384 Iterationsschritte). Die mit den Verfahren berechnete Näherung u stimmt umso besser mit der tatsächlichen Minimalfläche überein, je größer m ist. Für genaue Approximationen sind daher große Gleichungssysteme mit großer Kondition zu lösen. Auf diese Tatsache reagieren die verschiedenen Verfahren unterschiedlich empfindlich.

Kontraktionsfaktoren

Verfahren:	Relaxat.	SSOR	Richards. 32-Zykl.	Tscheb.	CG-Verf.	Tscheb+ SSOR	CG+ SSOR
Theorie:	$1-\frac{\pi^2}{2m^2}$	$1-\frac{\pi}{m}$	$1-\frac{16\pi^2}{m^2}$	$1-\frac{\pi}{m}$	$1-\frac{\pi}{m}$	$1-2\sqrt{\frac{\pi}{m}}$	$1-\frac{4}{3}\sqrt{\frac{\pi}{m}}$
$n=m^2=1024$:	0.9876	0.9260	0.9537	0.9536	0.8142	0.6753	0.7394
$n=m^2=16384$:	0.9926	0.9740	0.9845	0.9818	0.9504	0.8173	0.8594

Im Vergleich der vier diskutierten Verfahren, GMRES, Richardson- und Tschebyscheff-Iteration sowie CG-Verfahren gibt es eine klare Abstufung nach Effizienz und Flexibilität. Im hauptsächlich betrachteten Fall der symmetrisch–definiten Matrizen ist auf jeden Fall das CG-Verfahren der Tschebyscheff-Iteration, und diese der Richardson-Iteration vorzuziehen (außer bei gravierenden Speicherplatzbeschränkungen). Allerdings sind die Verfahren unterschiedlich gut in allgemeineren Situationen einsetzbar, die effizienteren Verfahren sind die weniger flexiblen. Das CG-Verfahren ist nur im symmetrisch-definiten Fall verwendbar. Die Tschebyscheff-Iteration ist auf Fälle verallgemeinerbar, in denen das Spektrum in einer Halbebene der komplexen Zahlen liegt, die die Null nicht enthält (vgl. Satz 2.2.4). Nur für die (billigste) Richardson-Iteration gibt es keine solchen prinzipiellen Einschränkungen, die Konstruktion guter Parameter (durch den Anwender!) kann im Einzelfall allerdings schwierig sein. Die Diskussion kann durch folgende Tabelle zusammengefaßt werden. Darin wird zwischen selbssteuernden Verfahren (+) und Verfahren mit Parametervorgabe durch Anwender (•) unterschieden.

Gegenüberstellung

	Sym.Pos.Def.	einseit. Spektrum	allgemein
GMRES	+	+	+
Richardson	•	•	•
Tschebysch.	••	••	—
CG	+++	—	—

$\longrightarrow$ Schwierigkeitsgrad

$\downarrow$ Effizienz

Das GMRES-Verfahren ist am flexibelsten, es ist auch im unsymmetrischen Fall einsetzbar und profitiert ohne Nutzereingriff von einer günstigen Eigenwert-Verteilung (etwa reell). Allerdings wird diese Flexibilität durch einen mit m wachsenden Aufwand bei der Berechnung der Basis U_m erkauft. Die Entwicklung eines günstigeren, verallgemeinerten CG-Verfahrens mit kurzen Rekursionen für nicht-definite oder unsymmetrische Probleme ist ein sehr aktuelles For-

schungsgebiet, keiner der vielen Ansätze kommt aber bisher ohne gravierende Nachteile aus.

Entsprechend der am Beginn erläuterten Motivation können die durch das Arnoldi-Verfahren (2.2.3) bzw. Lanczos-Verfahren (2.2.14) berechneten Modelle $U_m H_m U_m^{\mathsf{T}}$ bzw. $U_m T_m U_m^{\mathsf{T}}$ der Matrix A auch zur Approximation von Eigenwerten eingesetzt werden. Gerade im symmetrischen Fall sind die berechneten Eigenwert-Approximationen auch erheblich genauer als etwa die der Vektor-Iteration (vgl. Ende §2.1). Allerding ist gerade das Lanczos-Verfahren sehr fehleranfällig und benötigt in der Praxis Zusatzmaßnahmen bei der Anwendung. Diese sind, aus anderen Gründen auch im unsymmetrischen Fall erforderlich und würden den Rahmen der Vorlesung sprengen.

3 Die diskrete Fourier-Transformation

3.1 Trigonometrische Interpolation

Die Verwendung von Orthogonal-Basen bzw. unitären Transformationen steht im Zentrum vieler moderner Verfahren. Bei Interpolation bzw. Approximation von (periodischen) Funktionen kommen diese Vorteile bei Verwendung trigonometrischer Polynome zum Tragen. Darüber hinaus gibt es hier ein extrem schnelles Berechnungsverfahren. Zur Darstellung periodischer Funktionen betrachtet man zweckmäßigerweise Polynome auf dem komplexen Einheitskreis bzw. direkt trigonometrische Polynome im Komplexen

$$p(x) := a_0 + a_1 e^{ix} + a_2 e^{2ix} + \ldots + a_{n-1} e^{(n-1)ix}, \;\; x \in [0, 2\pi). \tag{3.1.1}$$

Hier ist $i = \sqrt{-1}$ die imaginäre Einheit. Bei der Interpolation werden aus praktischen Gründen (Anwendbarkeit der FFT) 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

$$p(x_k) = p\Big(2\pi \frac{k}{n}\Big) = y_k, \quad k = 0, 1, \ldots, n-1. \tag{3.1.2}$$

Wegen der 2π-Periodizität von p ist dann auch $p(2\pi k/n) = y_k \forall k \in \mathbb{Z}$, wenn man definiert $y_{k+jn} := y_k \forall j \in \mathbb{Z}$. Nach Einführung der Größe $\omega_n := \exp(2\pi i/n)$ entspricht (3.1.2) dem linearen Gleichungssystem

$$y_k = p(x_k) = \sum_{j=0}^{n-1} a_j e^{2\pi i j k/n} = \sum_{j=0}^{n-1} a_j \, \omega_n^{jk}, \quad k = 0, 1, \ldots, n-1. \tag{3.1.3}$$

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

$$\begin{aligned} \omega_n^{k+jn} &= \omega_n^k \;\forall j \in \mathbb{Z} \\ \sum_{k=0}^{n-1} \omega_n^{jk} \, \omega_n^{-\ell k} &= n\delta_{j\ell} = \begin{cases} n & \text{für} \quad j = \ell \\ 0 & \text{für} \quad j \neq \ell \end{cases} . \end{aligned} \tag{3.1.4}$$

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

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

Die Identität (3.1.4) besagt, dass die Vektoren $w^{(k)} := \left(\omega_n^{jk}\right)_{j=0}^{n-1} \in \mathbb{C}^n$ eine *Orthonormalbasis* bilden im Innenprodukt

$$(y, z) := \frac{1}{n} \sum_{j=0}^{n-1} y_j \, \bar{z}_j. \tag{3.1.5}$$

Daher läßt sich die Lösung des Interpolationsproblems (3.1.3) 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 3.1.1 *Die Koeffizienten a_j des Interpolationspolynoms (3.1.1) mit (3.1.2) sind*

$$a_j = (y, w^{(j)}) = \frac{1}{n} \sum_{k=0}^{n-1} y_k \, \omega_n^{-jk}, \quad j = 0, 1, \ldots, n-1. \tag{3.1.6}$$

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 (3.1.3) und (3.1.6).

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

Satz 3.1.2 *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 trigonometrischen Polynome vom Grad ℓ. ∎

Reelle Trigonometrische Interpolation:

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

$$\varphi(x) := \frac{\alpha_0}{2} + \sum_{k=1}^{m-1} (\alpha_k \cos kx + \beta_k \sin kx) + \frac{\alpha_m}{2} \cos mx, \tag{3.1.7}$$

das die Werte y_k interpoliert, über (3.1.6) bestimmen.

Satz 3.1.3 *Die Koeffizienten a_j seien mit $n := 2m$, $y_k \in \mathbb{R}$, nach (3.1.6) definiert und*

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

Dann interpoliert das trigonometrische Polynom (3.1.7) mit diesen Koeffizienten alle Daten,

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

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

$$a_{n-j} = \frac{1}{n}\sum_{k=0}^{n-1} y_k \omega_n^{k(j-n)} = \frac{1}{n}\sum_{k=0}^{n-1} y_k \omega_n^{kj} = \bar{a}_j$$

und daher

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

3.2 Die schnelle Fourier-Transformation (FFT)

Sowohl zur Bestimmung der Koeffizienten (3.1.6) als auch zur Auswertung des Polynoms (3.1.3) sind Summen der Form

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

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 ($\rightarrow$ *schnelle* Fourier-Tr.) aufgrund der hohen Symmetrie der Koeffizienten ω_n^{jk}. Das Prinzip ist schon bei $n = 2m$ erkennbar, dort gilt

$$\left.\begin{aligned} z_{2k} &= \sum_{j=0}^{n-1} b_j \omega_{2m}^{2jk} = \sum_{j=0}^{m-1} (b_j + b_{j+m})\omega_m^{jk} \qquad =: \sum_{j=0}^{m-1} b_j^g \omega_m^{jk}, \\ &\quad \text{da} \quad \omega_{2m}^2 = \omega_m,\ \omega_m^{(j+m)k} = \omega_m^{jk}, \\ z_{2k+1} &= \sum_{j=0}^{n-1} b_j \omega_{2m}^{2jk+j} = \sum_{j=0}^{m-1} [(b_j - b_{j+m})\omega_n^j]\omega_m^{jk} \quad =: \sum_{j=0}^{m-1} b_j^u \omega_m^{jk}, \\ &\quad \text{da} \quad \omega_{2m}^2 = \omega_m,\ \omega_{2m}^m = -1. \end{aligned}\right\} \tag{3.2.1}$$

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 $n = 2m$ Daten auf 2 Transformationen der halben Ordnung m zurückgeführt werden. Für den Aufwand A_n der Fourier-Transformation gilt also die Beziehung

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

Für $n = 2^\ell$ ist die Methode (3.2.1) 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 3.2.1 *Für $n = 2^\ell$ können die Summen (3.1.3) bzw. (3.1.6) mit $\frac{3}{2} n \log_2 n$ komplexen Operationen berechnet werden.*

Praktische Durchführung bei $n = 2^\ell$ (Cooley-Tuckey-FFT):

Die Durchführung der ℓ Transformations-Stufen bei Anwendung von (3.2.1) 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:

$$\begin{cases} b_j^{(1)} := & b_j^g = b_j + b_{j+m}, & j = 0, \ldots, m-1, & \text{Berechnung } \{z_{2j}\} \\ b_{j+m}^{(1)} := & b_j^u = (b_j - b_{j+m})\omega_n^j, & j = 0, \ldots, m-1, & \text{Berechnung } \{z_{2j+1}\}. \end{cases} \tag{3.2.2}$$

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:

$$\text{Koeff. mit } \begin{cases} k = 2j \\ k = 2j+1 \end{cases} \quad \text{kommt in Position } \begin{cases} j \\ m+j \end{cases}$$

$$\text{Index binär: } \boxed{s_{\ell-1}} \boxed{\cdots} \boxed{s_1} \boxed{s_0} \quad \longrightarrow \quad \boxed{s_0} \boxed{s_{\ell-1}} \boxed{\cdots} \boxed{s_1}$$

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:

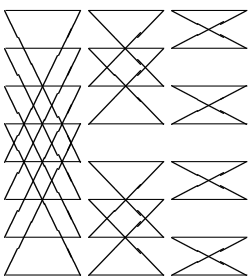
$$z_k \text{ mit } k = \boxed{s_{\ell-1}} \boxed{\cdots} \boxed{s_1} \boxed{s_0} \text{ steht in } b_j^{(\ell)} \text{ mit } j = \boxed{s_0} \boxed{s_1} \boxed{\cdots} \boxed{s_{\ell-1}}.$$

Zur Korrektur ist am Ende der FFT ein Permutationsschritt durchzuführen.

Beispiel 3.2.2 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 (3.2.2) 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

$$c_j = \frac{1}{2\pi}\int_0^{2\pi} f(x)e^{-ijx}dx,\ j \in \mathbb{Z}, \tag{3.2.3}$$

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

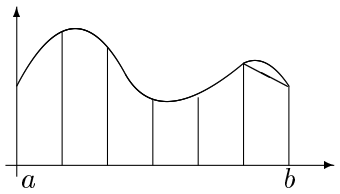
$$\operatorname{Re} c_j = \frac{1}{2\pi}\int_0^{2\pi} f(x)\cos jx\,dx, \quad \operatorname{Im} c_j = -\frac{1}{2\pi}\int_0^{2\pi} f(x)\sin jx\,dx.$$

Mit den Werten $y_k = f(x_k) = f(2\pi k/n),\ k \in \mathbb{Z}$, in (3.1.6) 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 $h = 2\pi/n$. Diese Formel entspricht einer Näherung an das Integral (3.2.3) mit Integrand $g(x) = f(x)e^{-ijx}$ durch *Quadratur* mit der Trapezregel

$$T_n := \frac{2\pi}{n}\sum_{k=0}^{n-1}\frac{1}{2}\Big(g(x_k) + g(x_{k+1})\Big).$$



Die Euler-McLaurin-Summenformel (o.Bew.)

$$T_n = \int_0^{2\pi} g(x)dx + \sum_{l=1}^{q} b_l h^{2l}\Big(g^{(2l-1)}(2\pi) - g^{(2l-1)}(0)\Big) + \dots$$

zeigt, dass der Fehler normalerweise nur wie $O(h^2) = O(n^{-2})$ gegen null geht, für *periodische* und genügend oft differenzierbare Integranden aber eine beliebig hohe Ordnung besitzt. Für $f \in C^m(\mathbb{R})$ gilt daher

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

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 $a_{n-j} = \bar{a}_j$, vgl. Satz 3.1.3.

b) Mehrdimensionale Fourier-Transformation: Für Funktionen $f(x,y)$ von zwei Variablen, die 2π-periodisch in x und y sind, lautet die Verallgemeinerung von (3.2.3)

$$c_{jk} = \frac{1}{4\pi^2}\int_0^{2\pi}\int_0^{2\pi} f(x,y)e^{-i(jx+ky)}\,dx\,dy = \frac{1}{2\pi}\int_0^{2\pi}\Big(\frac{1}{2\pi}\int_0^{2\pi} f(x,y)e^{-ijx}dx\Big)e^{-iky}dy.$$

Das Integral läßt sich also in zwei eindimensionale Integrationen auftrennen und die mehrdimensionale Fourier-Transformation somit auf eindimensionale Transformationen zurückführen. Das gilt auch für die diskrete FT. Ordnet man etwa die Funktionswerte $f(x_k, y_l)$ in einer $n \times m$-Matrix $F = (f(x_k, y_l))_{k,l=0}^{n-1,m-1}$ an und bezeichnet mit $\Omega_n = (\omega_n^{jk}) \in \mathbb{C}^{n\times n}$ die eindimensionale Fouriermatrix, dann erhält man das Ergebnis der zweidimensionalen diskreten Transformation durch getrennte, zeilen- und spaltenweise Anwendung

$$\Omega_n F \Omega_m.$$

Für Zweierpotenzen n und m läßt sich diese Operation daher mit $mn(\log_2 m + \log_2 n)$ Operationen durchführen. Dies ist für die Bildverarbeitung von großem Interesse.

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

$$z = u * y \quad :\Longleftrightarrow \quad z_k = \sum_{j=0}^{n-1} u_{k-j}\, y_j,\ k = 0, \ldots, n-1, \tag{3.2.4}$$

bzw. $k \in \mathbb{Z}$. Dieses *Faltungsprodukt* $*$ ist kommutativ und assoziativ. Werden die diskreten Fouriertransformierten zu $u, y, z \in \mathbb{C}^n$ wie üblich 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

$$\widehat{z} = \widehat{(u * y)} = \widehat{u} \circ \widehat{y} := (\widehat{u}_k \widehat{y}_k)_{k=0}^{n-1}. \tag{3.2.5}$$

Beweis Wegen der Periodizität ist

$$\widehat{z}_m \;=\; \sum_{k=0}^{n-1} z_k \omega_n^{km} = \sum_{k=0}^{n-1}\sum_{j=0}^{n-1} u_{k-j} y_j \omega_n^{(k-j)m} \omega_n^{jm} = \sum_{j=0}^{n-1} y_j \omega_n^{jm} \sum_{l=-j}^{n-1-j} u_l \omega_n^{lm} = \widehat{y}_m \widehat{u_m}. \;\blacksquare$$

Wegen (3.2.5) 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 (3.2.4) 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 (3.2.4), (3.2.5):

— *digitale Filter* haben die Gestalt einer Faltung, etwa beim *Tiefpaß-Filter* des gleitenden Mittelwerts $z_j := \frac{1}{4}(y_{j-1} + 2y_j + y_{j+1})$. Ein solcher Tiefpaß kann zum Entrauschen gestörter Signale dienen. Der Umweg über die FFT lohnt sich insbesondere bei längeren Filtern.

— Bei der Bildverarbeitung kommt die zweidimensionale Fouriertransformation zum Einsatz, Anwendungen sind wieder Filterung (2-dimensional), aber auch Datenkompression.

— Polynom- bzw. Langzahl-Multiplikation: Die Koeffizienten im Cauchy-Produkt von Polynomen haben die Form einer Faltung. Die Multiplikation großer Zahlen $x = \sum_{j=0}^{m} \xi_j B^j$, $0 \le \xi_j < B$, dargestellt in einer Basis B, kann daher über die FFT beschleunigt werden. Wenn man dabei in endlichen Zahlkörpern (Ringen) arbeitet, und dort primitive Einheitswurzeln einsetzt, treten keine Rundungsfehler auf.
— Statistik

c) Polynom-Interpolation in Tschebyscheff-Knoten: vgl. Numerik I, §2.1.

Verallgemeinerungen, Weiterentwicklungen, Alternativen

- 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. wird 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 bei der Bild-Komprimierung eingesetzt (JPEG-Bild-Format). Bei der Walsh- bzw. Hadamard-Transformation erfolgt eine reelle Entwicklung nach Treppenfunktionen, hier treten i.w. nur Additionen und Subtraktionen auf.

  Die Hadamard-Matrizen der Ordnung $n = 2^m$ können rekursiv definiert werden durch

$$H_1 = (1), \quad H_{2n} = \frac{1}{2} \begin{pmatrix} H_n & H_n \\ H_n & -H_n \end{pmatrix}, \ n \ge 1.$$

  Sie sind symmetrisch und sogar unitär, also ist $H_n^2 = I_n$. Bis auf den gemeinsamen Vorfaktor haben alle Einträge den Betrag eins. Es gilt, z.B.,

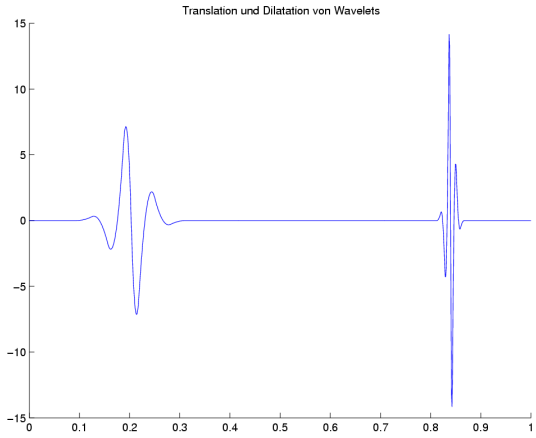
$$H_4 = \frac{1}{4} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}.$$

  Aufgrund dieser Eigenschaften kann die Multiplikation mit H_n und $H_n^{-1} = H_n$, $n = 2^\ell$, mit $n\ell$ Additionen + n Divisionen ausgeführt werden durch Einsatz der Rekursionsformel.

- Die Fourier-Entwicklung ist für viele natürliche Vorgänge gut geeignet, da Schwingungsvorgänge sich gesetzmäßig aus trigonometrischen Funktionen zusammensetzen. Ein wesentlicher Nachteil ist aber ihr global einheitliches Verhalten. Viele Prozesse verhalten sich

zwar ein Stück weit wie reine Schwingungen, ändern aber ihre Charakteristik gelegentlich (z.B., mit der Zeit). Dieses hat man schon früh durch Verwendung einer "gefensterten" DFT berücksichtigt. Allerdings handelt man sich dabei aber i.d.R. Verfälschungen ein.

Einige der Nachteile lassen sich mit *Wavelets* umgehen. Dieses sind Ansatzfunktionen $\psi(t)$, die zwar ebenfalls Wellencharakter haben (können), allerdings wegen ihres beschränkten Trägers nur einen lokal begrenzten Beitrag zur Gesamtfunktion liefern. Diese Lokalisation wird durch Verschiebung (Translation) einer Grundfunktion (*Mutter-Wavelet*) im Argument erreicht, $\psi(t-k)$, $k \in \mathbb{Z}$.



An die Stelle der Frequenz bei den Fourier-Funktionen tritt jetzt die Skala, die zur Darstellung unterschiedlicher Detailinformation im Signal dient. Auch hierbei bedient man sich des Mutter-Wavelets, indem man es staucht (Dilatation). Somit haben die Ansatzfunktionen hier (bis auf Skalierungsfaktoren) die Form

$$\psi_{jk}(t) = \psi(2^j t - k).$$

Die Ansatzfunktionen $\{\psi_{jk},\, k \in \mathbb{Z}\}$ bilden dann eine sog. Mehrfachauflösung, d.h., eine Zerlegung des Raums aller Signale (L_2) in Unterräume unterschiedlicher Detailinformation, wenn das Wavelet bestimmte Eigenschaften hat. Dazu gehört insbesondere eine sog. Zweiskalengleichung mit der Funktionen auf einer gröberen Skala durch solche auf einer feineren dargestellt werden können. Mit Hilfe der Zweiskalengleichung können einfache Algorithmen, ähnlich der FFT, zur Berechnung der Wavelet-Koeffizienten einer Funktion formuliert werden. Der Basisschritt ist dabei, ähnlich zu (3.2.2) die Aufspaltung des Ausgangssignals in einen Grobanteil (*Tiefpaß*) und Detailinformation (*Hochpaß*).

4 Fortsetzungsverfahren für nichtlineare Probleme

4.1 Die numerische Verfolgung von Lösungskurven

Das Standardverfahren zur Lösung des nichtlinearen Gleichungssystems

$$F(x) = 0, \qquad F: \ \mathbb{R}^n \mapsto \mathbb{R}^n, \tag{4.1.1}$$

ist das *Newton-Verfahren* (vgl. Numerik I, §5.2): $x^{(0)} \in \mathbb{R}^n$,

$$F'(x^{(k)})\Big(x^{(k+1)} - x^{(k)}\Big) = -t_k F(x^{(k)}), \ k = 0, 1, \ldots, \tag{4.1.2}$$

mit Parametern $t_k > 0$. Beim klassischen Newtonverfahren mit $t_k \equiv 1$ kann Konvergenz nur in einer, evtl. sehr kleinen, Umgebung der Lösung z von (4.1.1) 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,

$$\varphi(x^{(k+1)}) < \varphi(x^{(k)}) \ \text{ mit } \ \varphi(x) := \|F(x)\|_2^2.$$

Das Prinzip entspricht dem eines einfachen Gradientenverfahrens.Aber auch bei der Liniensuche ist die Konvergenz gegen z nicht sicher, da die Folge $\{x^{(k)}\}$ aus (4.1.2) 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 (4.1.1) eine gute Startnäherung für die Iteration (4.1.2) liefert. Konkret wird ein zusätzlicher Parameter λ eingeführt, etwa $\lambda \in [0, 1]$, und die *Problemschar*

$$0 = G(x, \lambda) := \lambda F(x) + (1 - \lambda) F_0(x), \ \lambda \in [0, 1], \tag{4.1.3}$$

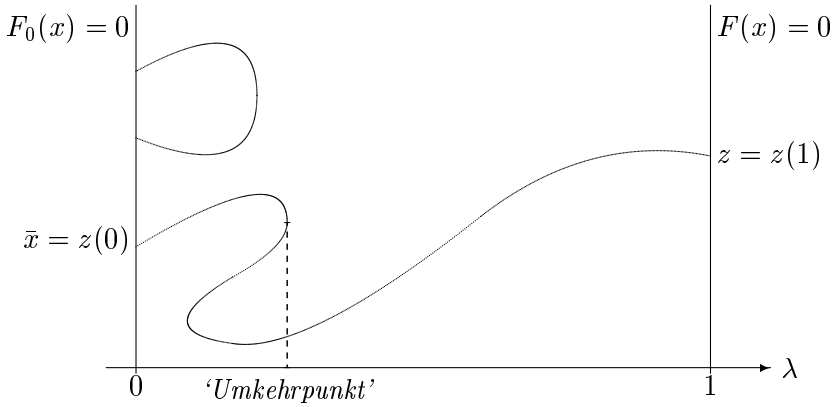
betrachtet ("Einbettung"). Dabei ist F_0 eine Abbildung, für die eine Nullstelle $\bar{x}$, $F_0(\bar{x}) = 0$, bekannt ist, z.B.,

$$\begin{aligned} F_0(x) := x \quad &\Rightarrow \quad \bar{x} = 0, \\ F_0(x) := F(x) - F(\tilde{x}) \quad &\Rightarrow \quad \bar{x} = \tilde{x}. \end{aligned}$$

Unter geeigneten Voraussetzungen hängen die Lösungen von (4.1.3) stetig vom Parameter λ ab, sie bilden eine *Lösungskurve* $z(\lambda)$. Daher kann man versuchen, dieser Kurve zu folgen,

vom bekannten Startpunkt $\quad z(0) = \bar{x} \quad$ mit $\quad G(\bar{x}, 0) = F_0(\bar{x}) = 0,$
zum gesuchten Endpunkt $\quad z(1) = z \quad$ mit $\quad G(z, 1) = F(z) = 0.$

Im folgenden Bild sind mögliche Situationen skizziert, insbesondere muß $z(\lambda)$ keine (eindeutige) Funktion von λ sein.



Parameterabhängige Gleichungssysteme $G(x, \lambda) = 0$ treten in der Praxis oft auch unabhängig von der Motivation (4.1.3) auf, etwa als Abhängigkeitsanalysen über Material- oder Modellparameter. Daher wird nun das abstrakte Problem

$$G(x, \lambda) = 0, \quad G: \ D \times [0, 1] \mapsto \mathbb{R}^n, \ D \subset \mathbb{R}^n, \tag{4.1.4}$$

betrachtet. Der Satz über implizite Funktionen liefert hier ein einfaches Kriterium für die Existenz einer glatten und bezüglich λ eindeutigen Lösungskurve.

Satz 4.1.1 *D sei offen und $G \in C^1(D \times [0, 1])$. Die Ableitung $G_x = \partial G/\partial x$ (Jacobimatrix) sei überall invertierbar und es gelte für alle $(x, \lambda) \in D \times [0, 1]$:*

$$\|G_x(x, \lambda)^{-1}\| \leq \alpha, \ \|G_\lambda(x, \lambda)\| \leq \beta_1. \tag{4.1.5}$$

Es sei $z_0 \in D$ eine Lösung von $G(x, 0) = 0$ so, dass gilt

$$\{x \in \mathbb{R}^n : \ \|x - z_0\| \leq \alpha\beta_1\} \subseteq D.$$

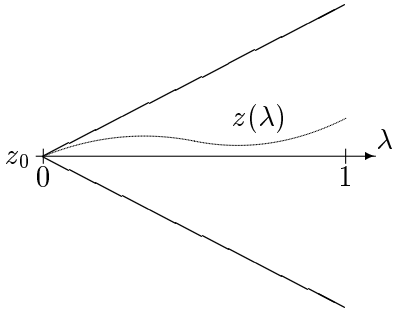
Dann gibt es eine in z_0 beginnende, stetig differenzierbare Lösungsfunktion $z(\lambda)$ mit $z(0) = z_0$, $G(z(\lambda), \lambda) \equiv 0$, für $\lambda \in [0, 1]$. Ihre Ableitung z' erfüllt die Beziehung

$$z'(\lambda) = \frac{dz}{d\lambda}(\lambda) = -G_x(z(\lambda), \lambda)^{-1} G_\lambda(z(\lambda), \lambda). \tag{4.1.6}$$

Beweis Aus der Regularität von G_x und dem Satz über implizite Funktionen folgt die lokale Existenz von $z(\lambda)$ und die Gültigkeit von (4.1.6), da

$$G(z(\lambda), \lambda) \equiv 0 \ \Rightarrow \ G_x z' + G_\lambda \equiv 0.$$

Die daraus abgeleitete Gleichung (4.1.6) ist eine Differentialgleichung $z' = f(z, \lambda)$ mit stetiger und beschränkter rechter Seite f, $\|f\| \leq \alpha\beta_1$ in $D \times [0, 1]$. Nach dem Existenssatz von Peano existiert daher eine Lösung $z(\lambda)$, die in dem Kegel $\|z(\lambda) - z(0)\| \leq \alpha\beta_1\lambda$ verläuft. Wenn dieser, wie gefordert, ganz in $D \times [0, 1]$ liegt, erreicht die Lösung $z(\lambda)$ den rechten Rand bei $\lambda = 1$. ∎

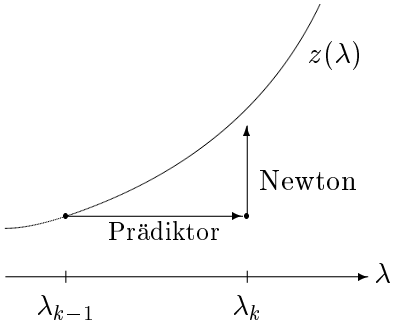


Unter der Voraussetzung (4.1.5) können insbesondere keine *Umkehrpunkte* wie im obigen Bild auftreten, da in solchen Punkten die partielle Ableitung G_x singulär wird. Zunächst wird die Verfolgung von Kurven ohne Umkehrpunkte behandelt. Tatsächlich berechnet werden dabei allerdings nur einzelne Punkte der Lösungskurve in Parameterwerten $0 = \lambda_0, \lambda_1, \ldots$. Mögliche

Strategien

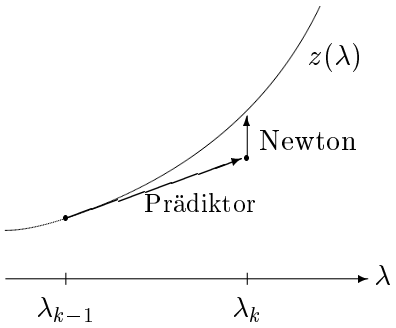
- *Klassische Fortsetzung*:

  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 (4.1.6) ist in λ_{k-1} mit $z(\lambda_{k-1})$ auch die Ableitung $z'(\lambda_{k-1})$, also die Tangentenrichtung 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



$$\text{Prädiktor-Schritt:} \quad \begin{cases} x^{(0)}(\lambda_k) := z(\lambda_{k-1}) + h_k z'(\lambda_{k-1}), & \text{mit} \\ G_x\Big(z(\lambda_{k-1}), \lambda_{k-1}\Big) z'(\lambda_{k-1}) = -G_\lambda\Big(z(\lambda_{k-1}), \lambda_{k-1}\Big) \end{cases} \tag{4.1.7}$$

$$\text{Korrektor-Schritt:} \quad \begin{cases} G_x\Big(x^{(j)}(\lambda_k), \lambda_k\Big)\Big(x^{(j+1)}(\lambda_k) - x^{(j)}(\lambda_k)\Big) = -G\Big(x^{(j)}(\lambda_k), \lambda_k\Big), \\ j = 0, 1, \ldots \ . \end{cases} \tag{4.1.8}$$

Im folgenden wird nur die bessere tangentiale Fortsetzung untersucht. Auch hier ist die Schrittweite h_k grundsätzlich immer so klein zu wählen, dass die Newton-Iteration (4.1.8) 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 zeigt, dass man $\lambda = 1$ erreichen kann und charakterisiert die größte Schrittweite, mit der noch die Konvergenz des Prädiktor-Korrektor-Verfahrens garantiert ist.

Satz 4.1.2 *Es sei D offen, $G \in C^2(D \times [0,1])$ und*

$$\|G_x(x,\lambda)^{-1}\| \le \alpha, \ \ \|G_\lambda\| \le \beta_1, \ \ \max\{\|G_{xx}\|, \|G_{x\lambda}\|, \|G_{\lambda\lambda}\|\} \le \beta_2 \tag{4.1.9}$$

$\forall (x,\lambda) \in D \times [0,1]$. *Damit sei*

$$\bar{h} := \frac{1}{\alpha(1 + \alpha\beta_1)\beta_2}. \tag{4.1.10}$$

Die Lösung $z(\lambda_{k-1})$, $\lambda_{k-1} \in [0,1)$, von (4.1.4) 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 (4.1.7), (4.1.8) gegen $z(\lambda_k)$ wenn $\lambda_k \le 1$ gilt und

$$h_k = \lambda_k - \lambda_{k-1} \le \bar{h}.$$

Beweis Nach Satz 5.2.2 aus der Numerik I konvergiert die Newton-Iteration (4.1.8), wenn für den Startwert $x^{(0)}$ gilt

$$\|x^{(0)}(\lambda_k) - z(\lambda_k)\| < \frac{2}{3\alpha\beta_2}. \tag{4.1.11}$$

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 4.1.1 ist z differenzierbar und überall $\|z'\| \le \alpha\beta_1$, 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 wieder 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 (4.1.11) 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. Diese ist unter der Voraussetzung $h_k \le \bar{h}$ wegen $\frac{1}{2} < \frac{2}{3}$ erfüllt. ∎

Dieser Satz zeigt, 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 hat aber eine geringe praktische Bedeutung, da die Schranken (4.1.9) kaum bekannt sind. Aber auch aus Effizienzgründen ist es i.a. nicht sinnvoll, mit maximalen Schrittweiten $h_k \cong \bar{h}$ vorzugehen. Denn durch die quadratische Konvergenz des Newtonverfahrens wird der Startfehler einer guten Näherung wesentlich stärker reduziert als der einer schlechten. Für die Fehler $e_j := \|x^{(j)}(\lambda_k) - z(\lambda_k)\|$ lautet die Konvergenzaussage von Satz I.5.2.2

$$e_j \le \frac{3}{2}\alpha\beta_2\, e_{j-1}^2. \tag{4.1.12}$$

Nun wird für $h \cong \bar{h}$ 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. Dann werden also viele Newton-Schritte benötigt,

um wieder nahe genug an die Lösungskurve zu kommen. Da der Prädiktor-Schritt (4.1.7) den gleichen Aufwand wie eine Newton-Korrektor (4.1.8) erfordert, ist es günstiger und sicherer mehrere kleine λ-Schritte zu machen, wenn weniger Korrekturen pro Schritt anfallen. Dazu muß eine gute *Kontraktion* der Newton-Iteration gesichert werden, etwa $\frac{3}{2}\alpha\beta_2 e_0 \overset{!}{\leq} \frac{1}{10} \Rightarrow e_1 \leq \frac{1}{10}e_0$, $e_2 \leq \frac{1}{1000}e_0, \ldots$. 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

$$\frac{e_1}{e_0} \cong \frac{\|x^{(2)} - x^{(1)}\|}{\|x^{(1)} - x^{(0)}\|} \overset{!}{\leq} q \quad (q = \frac{1}{10} \cdots \frac{1}{2}) \tag{4.1.13}$$

auf eine akzeptable Konvergenz der Newton-Korrektur (4.1.8) 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- (4.1.7) und zwei Korrektorschritte (4.1.8) durchgeführt. Entspricht dabei die Kontraktion nicht der Bedingung (4.1.13), wird dieser Versuch verworfen und die Schrittweite verkleinert, etwa $h_k \to h_k/2$ ("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

```
λ := 0; h := h_Start; z := z_0,
löse G_x(z_0, 0)t = −G_λ(z_0, 0);                    //Tangente t
wiederhole {
  y := z + ht; j := 0;                               //Prädiktor
  wiederhole {                                       //Newton-Iteration
    löse G_x(y, λ + h)d = −G(y, λ + h);
    y := y + d; j := j + 1; c_j := ‖d‖;
  } bis (c_j ≤ ε) oder (j ≥ j_max);                  //z.B. j_max = 2
  falls (c_j > ε) oder (c_2/c_1 > q) dann            //Schritt verwerfen!
    { h := h/2; falls h < h_min dann Abbruch; }
  sonst {                                            //Schritt akzeptieren!
    λ := λ + h; z := y;
    löse G_x(z, λ)t = −G_λ(z, λ);                    //neue Tangente t
    falls c_2/c_1 < q/4 dann h := 2h;
    falls λ + h > 1 dann h := 1 − λ;
  }
} bis λ ≥ 1;
```

(4.1.14)

Im Prädiktor-Schritt kann übrigens die QR-Zerlegung von G_x aus dem letzten Newton-Schritt wiederverwendet werden.

Zu einem Abbruch des Verfahrens kommt es in der Regel nur dann, wenn die Voraussetzungen von Satz 4.1.1 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 den Punkt $(x, \lambda) = (1, 0)$ 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.

Analog kann man beim allgemeinen Problem die Hervorhebung des $n+1$-ten Parameters λ dadurch fallen lassen, dass alle Komponenten zu einem $n+1$-Vektor y zusammengefaßt werden, $y_i = x_i$ $(i \leq n)$, $y_{n+1} := \lambda$. Das Problem lautet dann

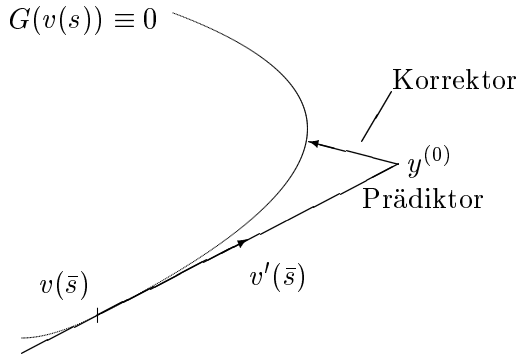
$$G(y) = 0, \quad G: D \subseteq \mathbb{R}^{n+1} \mapsto \mathbb{R}^n, \quad G \in C^2(D). \tag{4.1.15}$$

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 nun

$$G(v(s)) \equiv 0 \quad \Rightarrow \quad G'(v(s))v'(s) \equiv 0. \tag{4.1.16}$$

Die Ableitung und Tangentenrichtung $v' \in \mathbb{R}^{n+1}$ ist somit ein Element des eindimensionalen Kerns bzw. 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 (4.1.16) bis auf das Vorzeichen (d.h. die Kurvenrichtung) eindeutig bestimmt. Für ein Verfahren zur Kurvenverfolgung steht also mit (4.1.16) ein *Prädiktor* in Richtung der Kurventangente als Ersatz für (4.1.8) zur Verfügung.

In Verallgemeinerung von (4.1.8) sollte aber 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 $y^{(0)}$ steht dann ungefähr senkrecht auf $v'(\hat{s})$, 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. Numerik I, Satz 4.6.2) mit minimaler Norm $\|d^{(0)}\|_2$ definiert,

$$d^{(0)} = -G'(y^{(0)})^{+} G(y^{(0)}),$$

$(G')^{+}$ ist dabei die *verallgemeinerte Inverse*. So ergibt sich das

Prädiktor-Korrektor-Verfahren für (4.1.15):

$$\text{Prädiktor-Schritt:} \quad \begin{cases} s_k := s_{k-1} + h_k, \\ y^{(0)}(s_k) := v(s_{k-1}) + h_k t_{k-1}, \quad \text{mit} \\ G'\Big(v(s_{k-1})\Big) t_{k-1} = 0, \ \|t_{k-1}\|_2 = 1, \end{cases} \tag{4.1.17}$$

$$\text{Korrektor-Schritt:} \quad \begin{cases} y^{(j+1)}(s_k) = y^{(j)}(s_k) - G'\Big(y^{(j)}(s_k)\Big)^{+} G(y^{(j)}(s_k)), \\ j = 0, 1, \dots \ . \end{cases} \tag{4.1.18}$$

Die Richtung der Tangente t_k in (4.1.17) wird am einfachsten durch $t_{k-1}^{\mathsf{T}} t_{k-2} > 0$ festgelegt. Zur Berechnung der Lösungen in (4.1.17) und (4.1.18) dient die *QR-Zerlegung* 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

$$0 = G't = (R_0^{\mathsf{T}}, 0) \underbrace{Q^{\mathsf{T}} t}_{=:\, r}, \ r \in \mathbb{R}^{n+1} \iff r_i = \pm\delta_{i,n+1} \ \Rightarrow \ t = \pm Q e^{(n+1)}.$$

Die Kleinste-Quadrate-Lösung zu $G'd = -G$ ergibt sich ebenfalls aus dieser QR-Zerlegung $d = -QE_{n,n+1}^{\mathsf{T}}(R_0^{\mathsf{T}})^{-1}G$. Auch hier kann i.a. (außer im Anfangspunkt) für den Prädiktorschritt (4.1.17) die QR-Zerlegung des letzten Korrektorschritts zu Stelle s_{k-1} benutzt werden.

Bei einer Schrittweitensteuerung wie in (4.1.14) für das Verfahren (4.1.17),(4.1.18) ist es zweckmäßig, außer der anfänglichen Kontraktion $\|y^{(2)} - y^{(1)}\|/\|y^{(1)} - y^{(0)}\| \overset{!}{\leq} q < 1$ auch den Winkel zwischen aufeinanderfolgenden Tangenten zu überwachen, $t_{k-1}^{\mathsf{T}} t_{k-2} \overset{!}{\geq} \tau > 0$.

Beispiel 4.1.3 Ein beliebtes Testbeispiel, dessen Schwierigkeit mit wachsender Dimension größer wird, ist

$$F(x) = 0, \quad F_i(x) := x_i - \exp\cos(i \sum_{j=1}^{n} x_j), \ i = 1, \dots, n.$$

Mit $G(x, \lambda) = \lambda F(x) + (1-\lambda)x$ ergibt sich dafür die Einbettung

$$G(x, \lambda) = 0, \quad G_i(x, \lambda) := x_i - \lambda \exp\cos(i \sum_{j=1}^{n} x_i), \ i = 1, \dots, n,$$

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