

Einführung in die Numerische Mathematik
Teil II
SS 05

Prof. Dr. P. Spellucci

Revision 27.7.2005

Dieses Skriptum stellt den Inhalt der Vorlesung in einer sehr knappen, sicher nicht buchreifen Form dar. Es soll nicht das Studium der einschlägigen Lehrbücher ersetzen. Für Hinweise auf Fehler, unklare Formulierungen, wünschenswerte Ergänzungen etc. bin ich jederzeit dankbar. Man bedenke jedoch den Zeitrahmen der Veranstaltung, der lediglich 14 Doppelstunden umfasst, weshalb der eine oder andere Punkt wohl etwas zu kurz kommt oder auch einmal ganz wegfallen muss. Abschnitte, die im Kleindruck erscheinen, insbesondere eher technische Beweise, werden in der Vorlesung nicht vorgetragen. Sie sind aber für einen interessierten Leser zur Arbeitsvereinfachung hier aufgenommen worden. Diese Abschnitte sind durch eine Sequenz aus << und >> eingeklammert, um die Orientierung zu erleichtern. Das Gleiche gilt für die mit "ERG" gekennzeichneten Abschnitte bzw. Kapitel. Viele der in diesem Skript beschriebenen Verfahren können mit unserem interaktiven System NUMAWWW

<http://numawww.mathematik.tu-darmstadt.de:8081>

erprobt werden, ohne dabei selbst Programme erstellen zu müssen. Ebenso steht den Studierenden auf dem CIP-Pool MATLAB in der Version R12.1 zur Verfügung, das viele dieser Verfahren als fest implementierte Funktionen zur Verfügung stellt.

Basisliteratur:

1. J. Stoer, R. Bulirsch: Einführung in die Numerische Mathematik II Springer, Heidelberger Taschenbücher.
auch erhältlich in englischer Übersetzung:
Introduction to Numerical Analysis (als ein Band), Springer.
2. A. Quarteroni, R. Sacco, F. Saleri: Numerische Mathematik 1,2. Springer Lehrbuch.
auch erhältlich in englischer Übersetzung:
Quarteroni, Alfio; Sacco, Riccardo; Saleri, Fausto Numerical mathematics. New York, NY: Springer.
3. A. Bjoerck, G. Dahlquist: Numerische Methoden. Oldenbourg Verlag.
auch erhältlich in englischer Übersetzung
Numerical methods (mit Koautor Andersson)

Weitere Literatur wird am Ende jedes Kapitels angegeben.

Inhaltsverzeichnis

5	Das Matrizen–Eigenwertproblem	5
5.1	Lokalisierung von Eigenwerten. Die Sensitivität des Eigenwertproblems	7
5.2	Unitäre Ähnlichkeitstransformation auf obere Hessenberg-Form bzw. Tridiagonalform	18
5.3	Eigenwerte einer hermiteschen Tridiagonalmatrix	22
5.4	Bestimmung der Eigenvektoren einer hermiteschen Dreibandmatrix (ERG)	25
5.5	Direkte Iteration nach v. Mises und Verfahren von Wielandt	28
5.6	Das QR–Verfahren	35
5.7	Das Lanczos–Verfahren	48
5.8	Allgemeine Eigenwertprobleme	55
5.9	Die Singulärwertzerlegung (svd)	60
5.10	Zusammenfassung. Weiterführende Literatur	65
6	Iterative Lösung linearer Gleichungssysteme	67
6.0	Motivation	67
6.1	Allgemeine Ansätze zur Entwicklung von Iterationsverfahren Splittingverfahren	69
6.2	Konvergenzsätze für spezielle Matrizen	75
6.3	Blockiterationsverfahren	91
6.4	Das cg–Verfahren von Hestenes und Stiefel	92
6.5	Das Verfahren der generalisierten minimalen Residuen GMRES	104
6.6	Die Methode von Kaczmarz zur iterativen Lösung von linearen Gleichungssystemen	108
6.7	Ein spezielles Verfahren für große nichtlineare dünnbesetzte Systeme . . .	113
6.8	Zusammenfassung. Weiterführende Literatur	120

7	Rundungsfehleranalyse numerischer Algorithmen, (ERG)	121
7.1	Zahldarstellung	121
7.2	Fehlerfortpflanzungsgesetze der elementaren arithmetischen Verknüpfungen	126
7.3	Vorwärtsanalyse der Rundungsfehlereffekte	127
7.4	Das allgemeine Fehlerfortpflanzungsgesetz. Konditionszahlen eines mathematischen Problems.	129
7.5	Rückwurfanalyse	133
7.6	Intervallarithmetik	138
7.7	Weiterführende Literatur	139
8	Trigonometrische Interpolation ERG	141
8.1	Interpolation auf dem komplexen Einheitskreis	141
8.2	Trigonometrische Interpolation: (FFT Fast Fourier Transform)	144
9	Notation, Formeln	151
10	Zugang zu numerischer Software und anderer Information	155
10.1	Softwarebibliotheken	155
10.2	Information über Optimierungssoftware	157
10.3	Suchen nach software	157
10.4	Andere wichtige Quellen	157
10.5	Hilfe bei Fragen	158

Kapitel 5

Das Matrizen–Eigenwertproblem

In diesem Kapitel beschäftigen wir uns mit der Lokalisierung und numerischen Berechnung der reellen und komplexen Eigenwerte einer Matrix $A \in \mathbb{C}^{n \times n}$ (oder $\mathbb{R}^{n \times n}$) und der zugehörigen Eigenvektoren. Von naiven Standpunkt aus könnte man vermuten, daß es mit der Bestimmung der Nullstellen λ_i des charakteristischen Polynoms

$$p_n(\lambda; A) \stackrel{def}{=} \det(A - \lambda I)$$

und der Lösung der homogenen Gleichungssysteme

$$(A - \lambda_i I)x_i = 0$$

getan sei, also der Kombination eines skalaren Nullstellenproblems mit linearer Algebra. Der so formal beschriebene Bestimmungsweg:

- Berechnung der Koeffizienten von $p_n(\lambda; A)$
- Nullstellenbestimmung
- Lösung homogener Gleichungssysteme

erweist sich in der Praxis jedoch als völlig unbrauchbar, sowohl unter dem Gesichtspunkt des Rechenaufwandes als auch unter dem Gesichtspunkt der numerischen Stabilität. Zur Erläuterung des letzteren diene das folgende kleine Beispiel:

Die Matrix

$$A = \begin{pmatrix} 1000 & 1 \\ 1 & 1000 \end{pmatrix}$$

hat die Eigenwerte $\lambda_1 = 1001$ und $\lambda_2 = 999$. Ändert man A ab zu

$$\tilde{A} = \begin{pmatrix} 1000.001 & 1 \\ 1 & 1000 \end{pmatrix}$$

dann erhält man $\tilde{\lambda}_1 = 1001.00050\dots$, $\tilde{\lambda}_2 = 999.00050\dots$

Es ist $p_2(\lambda; A) = \lambda^2 - 2000\lambda + 999999$ und

Satz 5.1.1 Sei $A \in \mathbb{C}^{n \times n}$ und $\|\cdot\|$ eine einer Vektornorm zugeordnete Matrixnorm auf $\mathbb{C}^{n \times n}$. Dann gilt für jeden Eigenwert $\lambda(A)$

$$|\lambda(A)| \leq \rho(A) \leq \|A\|$$

□

Eine bereits wesentlich genauere Lokalisierung liefert häufig

Satz 5.1.2 Kreisesatz von Gerschgorin Sei $A \in \mathbb{C}^{n \times n}$ und

$$\mathcal{K}_i := \left\{ \lambda \in \mathbb{C} : |\lambda - \alpha_{ii}| \leq \sum_{\substack{j=1 \\ j \neq i}}^n |\alpha_{ij}| \right\}$$

$$\tilde{\mathcal{K}}_i := \left\{ \lambda \in \mathbb{C} : |\lambda - \alpha_{ii}| \leq \sum_{\substack{j=1 \\ j \neq i}}^n |\alpha_{ji}| \right\}$$

Ist dann $\lambda(A)$ ein Eigenwert von A , dann gilt:

$$\lambda(A) \in \left(\bigcup_{i=1}^n \mathcal{K}_i \right) \cap \left(\bigcup_{i=1}^n \tilde{\mathcal{K}}_i \right).$$

Beweis als einfache Übung. Hinweis: $Ax = \lambda x$, $x \neq 0$. Betrachte Zeile i mit $|x_i| = \|x\|_\infty$. □

Da die Matrizen A und $D^{-1}AD$ die gleichen Eigenwerte besitzen, kann man manchmal durch die Wahl geeigneter Transformationsmatrizen D (gewöhnlich beschränkt man sich auf Diagonalmatrizen) die Aussage von Satz 5.1.2 bedeutend verschärfen.

Beispiel 5.1.1 Sei

$$A = \begin{pmatrix} 1 & 10^{-3} & 10^{-4} \\ 10^{-3} & 2 & 10^{-3} \\ 10^{-4} & 10^{-3} & 3 \end{pmatrix}$$

Dann gilt nach Satz 5.1.2, weil A symmetrisch, d.h. $\lambda = \lambda(A)$ reell, für jeden Eigenwert von A

$$\lambda \in [1 - 0.0011, 1 + 0.0011] \cup [2 - 0.002, 2 + 0.002] \cup [3 + 0.0011, 3 + 0.0011].$$

Mit $D_1 := \text{diag}(1, 100, 10)$, $D_2 := \text{diag}(100, 1, 100)$, $D_3 := \text{diag}(10, 100, 1)$ erhält man nacheinander auch die folgenden Einschließungsmengen:

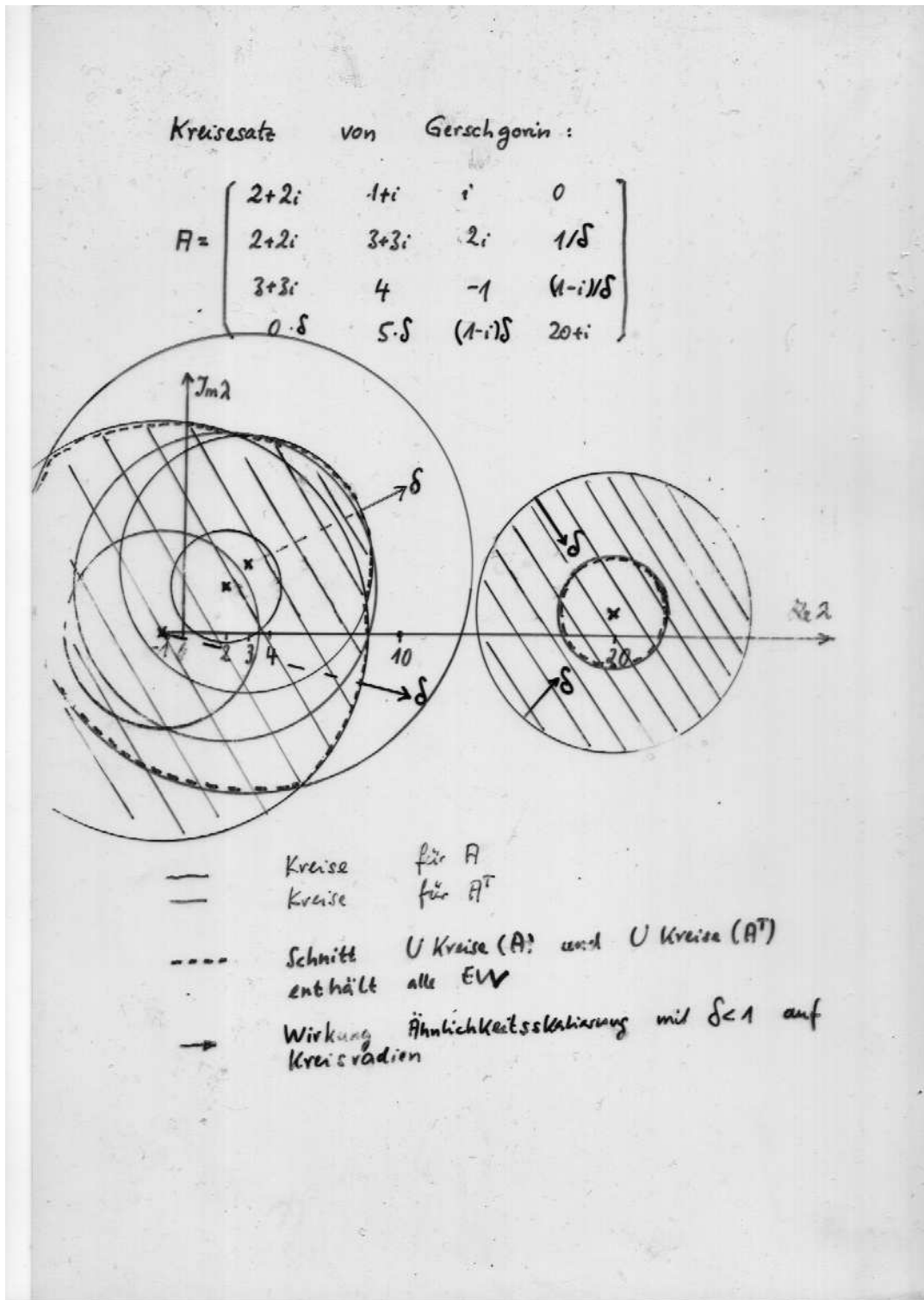
$$[1 - 2 \cdot 10^{-5}, 1 + 2 \cdot 10^{-5}] \cup [2 - 0.11, 2 + 0.11] \cup [3 - 0.0011, 3 + 0.0011]$$

$$[1 - 0.1001, 1 + 0.1001] \cup [2 - 2 \cdot 10^{-5}, 2 + 2 \cdot 10^{-5}] \cup [3 - 0.1001, 3 + 0.1001]$$

$$[1 - 0.0011, 1 + 0.0011] \cup [2 - 2 \cdot 10^{-2}, 2 + 2 \cdot 10^{-2}] \cup [3 - 2 \cdot 10^{-5}, 3 + 2 \cdot 10^{-5}]$$

und der Durchschnitt aller dieser Bereiche ist

$$[1 - 2 \cdot 10^{-5}, 1 + 2 \cdot 10^{-5}] \cup [2 - 2 \cdot 10^{-5}, 2 + 2 \cdot 10^{-5}] \cup [3 - 2 \cdot 10^{-5}, 3 + 2 \cdot 10^{-5}] \quad \square$$



Es gilt sogar die folgende Verschärfung von Satz 5.1.2:
Zusatz zu Satz 5.1.2

Ist $\{i_1, \dots, i_n\}$ Permutation von $\{1, \dots, n\}$ und

$$\left(\bigcup_{j=1}^m \mathcal{K}_{i_j}\right) \cap \mathcal{K}_{i_s} = \emptyset \quad s = m+1, \dots, n,$$

dann enthält $\bigcup_{j=1}^m \mathcal{K}_{i_j}$ genau m Eigenwerte von A (mit ihrer Vielfachheit gezählt), d.h. jede

Wegzusammenhangskomponente von $\bigcup_{i=1}^n \mathcal{K}_i$ enthält genauso viele Eigenwerte wie Kreise.

Beweis: Man setze

$$\begin{aligned} D &:= \text{diag}(\alpha_{11}, \dots, \alpha_{nn}) \\ B(\tau) &:= D + \tau(A - D) \quad 0 \leq \tau \leq 1, \end{aligned}$$

d.h. $B(0) = D$ und $B(1) = A$. Alle Eigenwerte von $B(\tau)$ liegen nach Satz 5.1.2 in

$$\bigcup_{i=1}^n \mathcal{K}_i(\tau); \quad \mathcal{K}_i(\tau) = \{z \in \mathbb{C} : |z - \alpha_{ii}| \leq \tau \sum_{\substack{j \neq i \\ j=1}}^n |\alpha_{ij}|\}$$

und die Aussage vom Zusatz zu Satz 5.1.2 gilt trivialerweise für $B(0)$. Die Eigenwerte von $B(\tau)$ hängen nach Satz 5.1.3 stetig von τ ab. Da aber $\bigcup_{j=1}^m \mathcal{K}_{i_j}(0)$ genau m Eigenwerte von $B(0)$ enthält und

$$\forall \tau \in [0, 1] : \left(\bigcup_{j=1}^m \mathcal{K}_{i_j}(\tau)\right) \cap \mathcal{K}_{i_s}(1) \subset \left(\bigcup_{j=1}^m \mathcal{K}_{i_j}(1)\right) \cap \mathcal{K}_{i_s}(1) = \emptyset$$

enthält auch $\bigcup_{j=1}^m \mathcal{K}_{i_j}(\tau)$ genau m Eigenwerte für $0 \leq \tau \leq 1$ □

Beispiel 5.1.1 Fortsetzung

Somit gilt für die drei Eigenwerte $\lambda_1, \lambda_2, \lambda_3$ von A bei geeigneter Numerierung:

$$i - 2 \cdot 10^{-5} \leq \lambda_i \leq i + 2 \cdot 10^{-5}, \quad i = 1, 2, 3$$

□

Für eine beliebige Matrix kennt man nur die folgende allgemeine Störungsaussage für die Eigenwerte, die keine Annahmen über das Eigenvektorsystem voraussetzt:

Satz 5.1.3 Seien $A, B \in \mathbb{C}^{n \times n}$, λ_i die Eigenwerte von A und λ'_i die Eigenwerte von B , $i = 1, \dots, n$ (jeweils mit ihrer Vielfachheit gezählt.)

Sei

$$\rho := \max\{|\alpha_{ij}|, |\beta_{ij}| : 1 \leq i, j \leq n\}$$

$$\delta := \frac{1}{n\rho} \sum_{i=1}^n \sum_{j=1}^n |\alpha_{ij} - \beta_{ij}| \quad .$$

Dann gibt es eine Numerierung der λ_i und λ'_i , so daß zu jedem λ_i ein λ'_i gehört mit

$$|\lambda_i - \lambda'_i| \leq 2(n+1)^2 \rho \sqrt[n]{\delta}$$

(d.h. die Eigenwerte einer Matrix sind Hölderstetige Funktionen der Matrixkoeffizienten vom Index $\frac{1}{n}$)

Beweis: siehe bei Ostrowski, A.M.: Solution of Equations in Euclidean and Banach Spaces, 3.ed., Acad. Press 1973, p.334-335 und 276-279. \square

Die Aussage dieses Satzes (Hölderstetigkeit der Eigenwerte mit Hölderindex $1/n$) kann nicht verbessert werden, wie folgendes Beispiel zeigt:

Beispiel 5.1.2

$$A(\varepsilon) = \begin{pmatrix} 1 & 1 & 0 & \dots & 0 \\ 0 & 1 & 1 & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 1 & 1 \\ \varepsilon & 0 & \dots & 0 & 1 \end{pmatrix}$$

hat für $\varepsilon = 0$ den n -fachen Eigenwert 1 und für $\varepsilon > 0$ die n paarweise verschiedenen Eigenwerte

$$\lambda_j = 1 - \varepsilon^{(1/n)} \exp(2\pi i(j-1)/n), \quad j = 1, \dots, n$$

wobei mit $\varepsilon^{(1/n)}$ der (reelle) Hauptwert gemeint ist.

Ist x ein Eigenvektor von A , dann kann man mit Hilfe von

$$Ax = \lambda x \Rightarrow \lambda = \frac{x^H Ax}{x^H x}$$

den zugehörigen Eigenwert berechnen. Hat man eine Eigenvektornäherung x (im folgenden Satz kann jeder beliebige Vektor $x \neq 0$, für den auch $Ax \neq 0$ ist, als eine solche Näherung dienen), dann kann man mit Hilfe dieses **Rayleighquotienten**

$$R(x; A) := \frac{x^H Ax}{x^H x}$$

eine zugehörige Eigenwertnäherung definieren, für die man ebenfalls eine Einschließungsaussage herleiten kann. Weil für $x \neq 0$ und $Ax = 0$ x ein Eigenvektor zum

Eigenwert null von A ist, schliessen wir diesen Fall jetzt aus:

Satz 5.1.4 $A \in \mathbb{C}^{n \times n}$ sei diagonalähnlich mit Eigenwerten $\lambda_1, \dots, \lambda_n$. Sei $x \in \mathbb{C}^n, x \neq 0$ und $Ax \neq 0$. Definiere

$$\lambda := R(x; A)$$

Dann gilt

(i) $\|Ax - \lambda x\|_2^2 \leq \|Ax - cx\|_2^2 \quad \forall c \in \mathbb{C}$

(ii) $\exists \lambda_j \neq 0, \quad \lambda_j$ Eigenwert von A und

$$\left| \frac{\lambda_j - \lambda}{\lambda_j} \right| \leq \frac{\|Ax - \lambda x\|_2}{\|Ax\|_2} \text{cond}_{\|\cdot\|_2}(U)$$

wobei $U = (u_1, \dots, u_n)$ ein vollständiges Eigenvektorsystem von A ist

(iii) Ist A normal, (d.h. $AA^H = A^H A$), dann $\exists \lambda_j \neq 0$ Eigenwert von A mit

$$\left| \frac{\lambda_j - \lambda}{\lambda_j} \right| \leq \frac{\|Ax - \lambda x\|_2}{\|Ax\|_2}$$

□

Beweis:

(i) o.B.d.A. $\|x\|_2 = 1$

$$\begin{aligned} \|Ax - cx\|_2^2 &= (x^H A^H - \bar{c}x^H)(Ax - cx) = x^H A^H Ax - \bar{c}x^H Ax - cx^H A^H x + |c|^2 \\ &= \|Ax\|_2^2 + |c - x^H Ax|^2 - |x^H Ax|^2 \geq \|Ax\|_2^2 - |x^H Ax|^2 \geq 0 \quad \text{mit "=" für } c = \lambda \end{aligned}$$

(Man beachte daß $|x^H Ax|^2 \leq \|Ax\|_2^2 \|x\|_2^2$ nach der Cauchy-Schwarzschen Ungleichung und $\|x\|_2^2 = 1$ nach Setzung.)

(ii) Sei $U^{-1}AU = \text{diag}(\lambda_1, \dots, \lambda_n) \stackrel{\text{def}}{=} \Lambda$; $y \stackrel{\text{def}}{=} U^{-1}x$

$$\begin{aligned}
\frac{\|Ax - \lambda x\|_2}{\|Ax\|_2} &= \frac{\|U(\Lambda - \lambda I)U^{-1}x\|_2}{\|U\Lambda U^{-1}x\|_2} \\
&\geq \frac{\|(\Lambda - \lambda I)y\|_2}{\|U^{-1}\|_2 \|U\|_2 \|\Lambda y\|_2} \\
&= \frac{1}{\|U\|_2 \|U^{-1}\|_2} \cdot \left(\frac{\sum_{i=1}^n |\lambda_i - \lambda|^2 |\eta_i|^2}{\sum_{i=1}^n |\lambda_i|^2 |\eta_i|^2} \right)^{1/2} \\
&= \frac{1}{\text{cond}_{\|\cdot\|_2}(U)} \left(\frac{|\lambda|^2 \sum_{\lambda_i=0} |\eta_i|^2 + \sum_{\lambda_i \neq 0} \left| \frac{\lambda_i - \lambda}{\lambda_i} \right|^2 |\eta_i \lambda_i|^2}{\sum_{\lambda_i \neq 0} |\lambda_i \eta_i|^2} \right)^{1/2} \\
&\geq \frac{1}{\text{cond}_{\|\cdot\|_2}(U)} \cdot \min_{i: \lambda_i \neq 0} \left| \frac{\lambda_i - \lambda}{\lambda_i} \right|
\end{aligned}$$

Man beachte, daß

$$\forall z : \|Uz\| \geq \frac{1}{\|U^{-1}\|} \|z\|.$$

(iii) Für normales A existiert ein unitäres vollständiges Eigenvektorsystem, d.h. es wird $\text{cond}_{\|\cdot\|_2}(U) = 1$

□

Der oben eingeführte Rayleighquotient ist also (im Sinne einer Einsetzprobe für ein Eigenwert–Eigenvektorpaar) eine optimale Schätzung für einen Eigenwert zu einer gegebenen Eigenvektornäherung.

Für hermitesche Matrizen, die in den Anwendungen eine besonders wichtige Rolle spielen, hat der Rayleighquotient viele schöne Eigenschaften, deren wichtigste hier angeführt seien.

Satz 5.1.5 Sei $A \in \mathbb{C}^{n \times n}$ hermitisch mit vollständigem unitärem Eigenvektorsystem $X = (x_1, \dots, x_n)$, $Ax_j = \lambda_j x_j \quad j = 1, \dots, n$. $\tilde{x} \in \mathbb{C}^n$ sei gegeben als Näherung für x_j mit

$$\tilde{x}^H \tilde{x} = 1, \quad \tilde{x} = x_j + \sum_{k=1}^n \epsilon_k x_k, \quad |\epsilon_k| \leq \epsilon \quad (\forall k)$$

Dann gilt

$$|R(\tilde{x}; A) - \lambda_j| \leq \sum_{\substack{i=1 \\ i \neq j}}^n |\lambda_i - \lambda_j| |\epsilon_i|^2 \leq 2\|A\|(n-1)\epsilon^2$$

d.h. der Fehler im Rayleighquotienten ist quadratisch klein in den Fehlern der Eigenvektornäherung. □

Beweis:

$$\begin{aligned}
R(\tilde{x}; A) &= (x_j + \sum_{k=1}^n \epsilon_k x_k)^H \underbrace{\sum_{i=1}^n \lambda_i x_i x_i^H}_A (x_j + \sum_{k=1}^n \epsilon_k x_k) \\
&= \sum_{i=1}^n \lambda_i \underbrace{((x_j + \sum_{k=1}^n \epsilon_k x_k)^H x_i)(x_i^H (x_j + \sum_{k=1}^n \epsilon_k x_k))}_{\delta_{ij} + \sum_{k=1}^n \bar{\epsilon}_k \delta_{ki}} \\
&\quad \underbrace{\hspace{10em}}_{|\delta_{ij} + \sum_{k=1}^n \epsilon_k \delta_{ki}|^2} \\
&= \sum_{\substack{i=1 \\ i \neq j}}^n \lambda_i |\epsilon_i|^2 + \lambda_j |1 + \epsilon_j|^2 \\
&= \sum_{\substack{i=1 \\ i \neq j}}^n (\lambda_i - \lambda_j) |\epsilon_i|^2 + \lambda_j \underbrace{(|1 + \epsilon_j|^2 + \sum_{\substack{i=1 \\ i \neq j}}^n |\epsilon_i|^2)}_{=\tilde{x}^H \tilde{x} = 1}
\end{aligned}$$

Betragsabschätzung, Anwendung der Dreiecksungleichung und der trivialen Schranke

$$|\lambda_i - \lambda_j| \leq 2\rho(A) \leq 2\|A\|$$

□

Eine vollständige Charakterisierung aller Eigenwerte einer hermiteschen Matrix durch eine Maximierungsaufgabe mit Nebenbedingungen liefert der

Satz 5.1.6 Courant'sches Minimax-Prinzip Sei $A \in \mathbb{C}^{n \times n}$ hermitisch. Die Eigenwerte von A seien (mit ihrer Vielfachheit gezählt) geordnet nach

$$\lambda_1 \geq \dots \geq \lambda_n .$$

\mathcal{V}_j bezeichne das System aller j -dimensionalen Teilräume von \mathbb{C}^n , $\mathcal{V}_0 = \{0\}$. Es gilt

$$\lambda_k = \min_{V \in \mathcal{V}_{k-1}} \max\{R(x; A) : x \neq 0, x^H v = 0 \forall v \in V\} = \min_{V: \dim(V)=k-1} \left\{ \max_{\substack{x: x \perp V \\ x \neq 0}} R(x; A) \right\}$$

$$\lambda_k = \max_{V \in \mathcal{V}_{n-k}} \min\{R(x; A) : x \neq 0, x^H v = 0 \forall v \in V\} = \max_{V: \dim(V)=n-k} \left\{ \min_{\substack{x: x \perp V \\ x \neq 0}} R(x; A) \right\}$$

□

<<

Beweis: Sei $U = (u_1, \dots, u_n)$ ein unitäres vollständiges Eigenvektorsystem von A , d.h. $Au_i = \lambda_i u_i$, $U^H U = I$. Ist $x \in \mathbb{C}^n$ beliebig, dann

$$x = \sum_{i=1}^n \gamma_i u_i \quad \text{mit } \gamma_i = u_i^H x$$

Somit

$$\begin{aligned}
 R(x; A) &= \frac{x^H U \Lambda U^H x}{x^H U^H U x} \\
 &= \sum_{i=1}^n \frac{|\gamma_i|^2}{\sum_{j=1}^n |\gamma_j|^2} \lambda_i \quad \begin{cases} \geq \lambda_k, & \text{falls } \gamma_{k+1} = \dots = \gamma_n = 0 \\ \leq \lambda_k, & \text{falls } \gamma_1 = \dots = \gamma_{k-1} = 0 \end{cases}
 \end{aligned}$$

Ist nun $V \in \mathcal{V}_{k-1}$, dann existiert $\tilde{x} \in \mathbb{C}^n$ mit folgenden Eigenschaften:

$$\tilde{x} \neq 0, \quad \tilde{x} = \sum_{i=1}^k \gamma_i u_i, \quad \tilde{x}^H v = 0 \quad \forall v \in V$$

Setze zum Beweis mit einer orthonormierten Basis v_1, \dots, v_{k-1} von V

$$g = \begin{pmatrix} \gamma_1 \\ \vdots \\ \gamma_k \end{pmatrix}, \quad g \neq 0 \quad \text{Lösung von} \quad \begin{pmatrix} v_1^H \\ \vdots \\ v_{k-1}^H \end{pmatrix} (u_1, \dots, u_k) g = 0$$

Also ist in diesem Falle

$$\max\{R(x; A) : x \neq 0, x^H v = 0 \quad \forall v \in V\} \geq \lambda_k$$

Gleichheit tritt hier ein für den Fall $V = \text{span}\{u_1, \dots, u_{k-1}\}$, d.h.

$$g = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \gamma_k \end{pmatrix} \quad \gamma_k \neq 0. \text{ Ist } V \in \mathcal{V}_{n-k}, \text{ dann existiert } \tilde{x} \in \mathbb{C}^n \text{ mit}$$

$$\tilde{x} \neq 0, \quad \tilde{x} = \sum_{j=k}^n \gamma_j u_j, \quad \tilde{x}^H v = 0 \quad \forall v \in V$$

Sei dazu $g = (\gamma_k, \dots, \gamma_n)^T \neq 0$ eine Lösung von $\begin{pmatrix} v_1^H \\ \vdots \\ v_{n-k}^H \end{pmatrix} (u_k, \dots, u_n) g = 0$ mit einer orthonormierten Basis v_1, \dots, v_{n-k} von \mathcal{V}_{n-k} , also

$$\min\{R(x; A) : x \neq 0, x^H v = 0 \quad \forall v \in V\} \leq \lambda_k$$

mit Gleichheit für $V = \text{span}\{u_{k+1}, \dots, u_n\}$ (dann ist $g = (\gamma_k, 0, \dots, 0)^T$) □

>>

Eine Folgerung ist:

Satz 5.1.7 Seien $A, B \in \mathbb{C}^{n \times n}$ beide hermitisch. $\lambda_i(A), \lambda_i(B)$ bezeichne die Eigenwerte von A und B und die Numerierung sei so vorgenommen, daß $\lambda_1(A) \geq \dots \geq \lambda_n(A), \quad \lambda_1(B) \geq \dots \geq \lambda_n(B)$. Dann gilt

$$\forall k \in \{1, \dots, n\} : \quad |\lambda_k(A) - \lambda_k(B)| \leq \rho(B - A) \tag{5.1}$$

□

Beweis: Setze $B := A + (B - A), \quad C := B - A$, verwende Satz 5.1.6 (Übung!) □

(5.1) stellt natürlich ein viel günstigeres Resultat dar als der Satz 5.1.3. Man beachte, daß (5.1) auch bei mehrfachen Eigenwerten gilt! Natürlich ist die Voraussetzung, daß beide Matrizen hermitisch sein sollen, sehr einschränkend. Die Eigenwerte einer diagonalähnlichen Matrix hängen wenigstens noch lipschitzstetig von den Matrixkoeffizienten ab. Dies besagt

Satz 5.1.8 Bauer-Fike-Theorem *Ist $A \in \mathbb{C}^{n \times n}$ diagonalähnlich, $U = (u_1, \dots, u_n)$ ein vollständiges Eigenvektorsystem von A , und ist $B \in \mathbb{C}^{n \times n}$ beliebig, dann gibt es zu jedem Eigenwert $\lambda_j(B)$ einen Eigenwert $\lambda_{i(j)}(A)$ von A mit*

$$|\lambda_{i(j)}(A) - \lambda_j(B)| \leq \text{cond}_{\|\cdot\|_\infty}(U) \|B - A\|_\infty \quad (5.2)$$

□

(Bem.: Diese Aussage gilt sogar für jede absolute Norm, d.i. eine Norm mit $\| |x| \| = \|x\| \quad \forall x \in \mathbb{C}^n$.)

Beweisskizze: Nichttrivialer Fall:

$\lambda(B) \neq \lambda_i(A) \quad i = 1, \dots, n, \quad \lambda(B)$ ein Eigenwert von B mit Eigenvektor $x \neq 0 \quad \Rightarrow$

$Bx - Ax = \lambda(B)x - Ax \quad \Rightarrow$

$x = (\lambda(B)I - A)^{-1}(B - A)x$, Normabschätzung, $I = UU^{-1}$,

$A = U\Lambda_A U^{-1}$ einsetzen.

$$\|(\lambda(B)I - \Lambda_A)^{-1}\|_\infty = \frac{1}{\min_i |\lambda(B) - \lambda_i(A)|}$$

□

Für nichtdiagonalähnliche Matrizen kann eine zu (5.1) bzw. (5.2) analoge Aussage nicht erwartet werden, siehe obiges Beispiel zur Hölderstetigkeit. Neben den bis jetzt abgeleiteten Eigenwertabschätzungen interessieren natürlich auch asymptotische Fehlerausagen für die Eigenwerte und Eigenvektoren.

Satz 5.1.9 Wilkinson Sei $A \in \mathbb{C}^{n \times n}$ diagonalähnlich, $X = (x_1, \dots, x_n)$ ein vollständiges Eigenvektorsystem von A , $Ax_i = \lambda_i x_i$ sowie $\|x_i\|_2 = 1 \quad i = 1, \dots, n$.

Ferner sei

$$X^{-1} =: Y = \begin{pmatrix} y_1^H \\ \vdots \\ y_n^H \end{pmatrix} \quad (\text{d.h. } y_i^H A = y_i^H \lambda_i \quad y_i \text{ sogenannter Linkseigenvektor zu } \lambda_i).$$

λ_j sei ein einfacher Eigenwert von A . Dann gilt: zu $F \in \mathbb{C}^{n \times n}$ mit $\|F\|_2$ hinreichend klein existiert ein einfacher Eigenwert μ_j von $A + F$ mit Eigenvektor z_j , $\|z_j\| = 1$ so daß

$$\begin{aligned} \mu_j &= \lambda_j + \frac{y_j^H F x_j}{\|y_j\|_2 \|x_j\|_2} \cdot \frac{\|y_j\|_2 \|x_j\|_2}{y_j^H x_j} + \mathcal{O}(\|F\|_2^2) \\ z_j &= x_j + \left(\sum_{\substack{i=1 \\ i \neq j}}^n \frac{y_i^H F x_j}{\|y_i\|_2 \|x_j\|_2} \cdot \frac{1}{\lambda_j - \lambda_i} \frac{\|y_i\|_2 \|x_i\|_2}{y_i^H x_i} x_i \right) + \mathcal{O}(\|F\|_2^2) \end{aligned}$$

□

Beweis: Man benutze den Hauptsatz über implizite Funktionen für das Problem $G(x, \lambda, \varepsilon) = 0$ mit $G(x_j, \lambda_j, 0) = 0$,

$$\begin{aligned} G(x, \lambda, \varepsilon) &= \begin{pmatrix} (A + \varepsilon F_0)x - \lambda x \\ \dots\dots\dots \\ x^T x - 1 \end{pmatrix} \\ F &= \varepsilon F_0 \end{aligned}$$

mit ε in einer geeigneten Nullumgebung und stelle damit die Lösung x , λ als Funktionen von ε dar. □

Entscheidender Fehlerverstärkungsfaktor für einen Eigenwert ist also $\|y_j\|_2 \|x_j\|_2 / |y_j^H x_j| (\gg 1 \text{ möglich bei nichtnormalen Matrizen})$, während bei einem Eigenvektor zusätzlich die **Separation** der Eigenwerte und alle Terme

$$\|y_i\|_2 \|x_i\|_2 / |y_i^H x_i| = \frac{1}{\cos(\angle(x_i, y_i))}$$

eine Rolle spielen. Nach den Setzungen des Satzes ist natürlich $|y_i^H x_i| = 1$, um aber die Tatsache hervorzuheben, daß im allgemeinen Rechts- und Linkseigenvektoren nicht orthogonal sind, bevorzugen wir diese Schreibweise, bei der der erste Term stets $\leq \|F\|$ ist, während die übrigen die Fehlerverstärkungs- bzw. Dämpfungsfaktoren darstellen.

5.2 Unitäre Ähnlichkeitstransformation auf obere Hessenberg-Form bzw. Tridiagonalform

Die Lösung des vollständigen Matrizen-Eigenwertproblems für eine vollbesetzte Matrix beginnt stets mit einer Ähnlichkeitstransformation auf “kondensierte” Form, mit dem Ziel, die Matrix möglichst “schmal” besetzt zu erhalten. Diese Transformation schleppt beim praktischen Rechnen unvermeidbare Rundungsfehler ein. Um noch brauchbar zu sein, dürfen jedoch die Eigenwerte dadurch nicht wesentlich stärker verfälscht werden als wenn die Ausgangsmatrix selbst im Rahmen der Rundungsgenauigkeit abgeändert würde. Aus diesem Grund kommt bei einer allgemeinen Matrix nur die Transformation auf Hessenberggestalt in Frage (was wir aber nicht beweisen wollen). Eine Transformation einer allgemeinen Matrix auf Tridiagonalgestalt ist bekannt, aber leider numerisch instabil.

In diesem Abschnitt geben wir eine **unitäre** Ähnlichkeitstransformation auf Hessenberggestalt an. Eine Ähnlichkeitstransformation mit Dreiecksmatrizen, wie sie bei der Gauss-Elimination benutzt werden, ist auch möglich, wir beschränken uns hier aber auf die Verwendung von Householdermatrizen, die numerisch besonders unempfindlich ist.

$$\square = A \rightarrow U_1 A U_1 \rightarrow \cdots U_{n-2} \cdots U_1 A U_1 U_2 \cdots U_{n-2} = \begin{array}{c} \diagdown \\ \square \\ \diagup \end{array}$$

Dabei sind die einzelnen U_i hermitisch und unitär. Ist A selbst hermitisch, dann wird

$$(U_{n-2} \cdots U_1 A U_1 U_2 \cdots U_{n-2})^H = U_{n-2} \cdots U_1 A U_1 U_2 \cdots U_{n-2}, \quad \text{hermitisch}$$

d.h. die **transformierte Matrix** erhält automatisch **Dreibandform!** Die Transformation verläuft in $n - 2$ Schritten. Sei nach $j - 1$ Schritten

$$A_j = U_{j-1} \cdots U_1 A U_1 U_2 \cdots U_{j-1} = \begin{pmatrix} \alpha_{11}^{(j)} & \alpha_{12}^{(j)} & \cdots & \alpha_{1,j-1}^{(j)} & \alpha_{1j}^{(j)} & \cdots & \alpha_{1n}^{(j)} \\ \alpha_{21}^{(j)} & \alpha_{22}^{(j)} & \cdots & \alpha_{2,j-1}^{(j)} & \alpha_{2j}^{(j)} & \cdots & \alpha_{2n}^{(j)} \\ 0 & \alpha_{32}^{(j)} & & \vdots & \vdots & & \vdots \\ \vdots & 0 & \ddots & \alpha_{j,j-1}^{(j)} & \alpha_{jj}^{(j)} & & \alpha_{j,n}^{(j)} \\ \vdots & \vdots & \ddots & 0 & \alpha_{j+1,j}^{(j)} & & \vdots \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & \cdots & \cdots & 0 & \alpha_{n,j}^{(j)} & \cdots & \alpha_{n,n}^{(j)} \end{pmatrix}$$

Dann wird

$$A_{j+1} = U_j A_j U_j$$

mit

$$U_j = \left(\begin{array}{c|c} I & O \\ \hline O & \hat{U}_j \end{array} \right) \quad \hat{U}_j = I - \beta_j \hat{w}_j \hat{w}_j^H$$

wobei wiederum \hat{U}_j so konstruiert ist, daß

$$\hat{U}_j \begin{pmatrix} \alpha_{j+1,j}^{(j)} \\ \vdots \\ \alpha_{n,j}^{(j)} \end{pmatrix} = -\exp(i\varphi_j) \sigma_j \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Es ergeben sich die Formeln (vergl. Kapitel 3, Householder-Transformation)

$$\hat{w}_j = \begin{pmatrix} \exp(i\varphi_j) (|\alpha_{j+1,j}^{(j)}| + \sigma_j) \\ \alpha_{j+2,j}^{(j)} \\ \vdots \\ \alpha_{n,j}^{(j)} \end{pmatrix} \quad \sigma_j = \left(\sum_{k=j+1}^n |\alpha_{k,j}^{(j)}|^2 \right)^{1/2}$$

$$\alpha_{j+1,j}^{(j)} = \exp(i\varphi_j) |\alpha_{j+1,j}^{(j)}| \quad (\text{def } \varphi_j)$$

$$\beta_j = \frac{2}{\hat{w}_j^H \hat{w}_j} .$$

Da die ersten j Spalten von U_j Einheitsspalten sind, ändert die Multiplikation von $U_j A_j$ mit U_j von rechts die eben neu erzeugten Nullen in Spalte j nicht. Die Multiplikation von A_j mit U_j von links ändert die ersten j Zeilen nicht. Damit ist die Transformation bereits vollständig hergeleitet.

Bei der praktischen Durchführung der Transformation nutzt man die spezielle Struktur von U_j aus. Sei

$$A_j = \left(\begin{array}{c|c} A_{11}^{(j)} & A_{12}^{(j)} \\ \hline A_{21}^{(j)} & A_{22}^{(j)} \end{array} \right)$$

Dann wird

$$A_{j+1} = \left(\begin{array}{c|c} A_{11}^{(j)} & A_{12}^{(j)} \hat{U}_j \\ \hline \hat{U}_j A_{21}^{(j)} & \hat{U}_j A_{22}^{(j)} \hat{U}_j \end{array} \right)$$

$$\begin{aligned} \hat{U}_j A_{21}^{(j)} &= A_{21}^{(j)} - \beta_j \hat{w}_j \hat{w}_j^H A_{21}^{(j)} = A_{21}^{(j)} - \hat{u}_j \hat{z}_j^H \\ A_{12}^{(j)} \hat{U}_j &= A_{12}^{(j)} - \beta_j A_{12}^{(j)} \hat{w}_j \hat{w}_j^H = A_{12}^{(j)} - \hat{y}_j \hat{u}_j^H \\ \hat{U}_j A_{22}^{(j)} \hat{U}_j &= (I - \beta_j \hat{w}_j \hat{w}_j^H) (A_{22}^{(j)} - \beta_j A_{22}^{(j)} \hat{w}_j \hat{w}_j^H) \\ &= A_{22}^{(j)} - \hat{u}_j (\hat{w}_j^H A_{22}^{(j)}) - (A_{22}^{(j)} \hat{w}_j) \hat{u}_j^H + (\hat{w}_j^H A_{22}^{(j)} \hat{w}_j) \hat{u}_j \hat{u}_j^H \\ &= A_{22}^{(j)} - \hat{u}_j (\hat{s}_j^H - \frac{\gamma_j}{2} \hat{u}_j^H) - (\hat{t}_j - \frac{\gamma_j}{2} \hat{u}_j) \hat{u}_j^H \end{aligned}$$

mit

$$\begin{aligned} \hat{u}_j &= \beta_j \hat{w}_j \\ \hat{t}_j &= A_{22}^{(j)} \hat{w}_j \\ \gamma_j &= \hat{w}_j^H \hat{t}_j \\ \hat{s}_j^H &= \hat{w}_j^H A_{22}^{(j)} \end{aligned}$$

und $\hat{z}_j^H = \hat{w}_j^H A_{21}^{(j)}$, $\hat{y}_j = A_{12}^{(j)} \hat{w}_j$.

Nach Voraussetzung ist jedoch

$$A_{21}^{(j)} = \begin{pmatrix} 0 & \cdots & 0 & \alpha_{j+1,j}^{(j)} \\ \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & \alpha_{n,j}^{(j)} \end{pmatrix}$$

so daß die explizite Berechnung von $\hat{U}_j A_{21}^{(j)}$ entfällt:

$$\hat{U}_j A_{21}^{(j)} = \begin{pmatrix} 0 & \cdots & 0 & -\exp(i\varphi_j)\sigma_j \\ \vdots & & \vdots & 0 \\ \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & 0 \end{pmatrix}$$

Im hermiteschen Fall beachte man ferner

$$\begin{aligned} A \text{ hermitisch: } \quad A_{12}^{(j)} \hat{U}_j &= (\hat{U}_j A_{21}^{(j)})^H \quad (\text{keine explizite Berechnung}) \\ \hat{t}_j &= \hat{s}_j \end{aligned}$$

wodurch sich der Gesamtrechenaufwand mehr als halbiert.

(A allgemein: $\frac{5}{3}n^3 + \mathcal{O}(n^2)$, A hermitisch: $\frac{2}{3}n^3 + \mathcal{O}(n^2)$ wesentliche Operationen) Wir haben somit konstruktiv gezeigt:

Satz 5.2.1 *Jede komplexe $n \times n$ -Matrix kann durch eine unitäre Ähnlichkeitstransformation aus $n - 2$ Householdermatrizen auf obere Hessenberggestalt transformiert werden. Ist die Ausgangsmatrix hermitisch, dann ist die resultierende Matrix hermitisch tridiagonal. \square*

Beispiel 5.2.1 *Wir transformieren die Matrix*

$$A = \begin{pmatrix} 1 & 4 & 3 \\ 4 & -3 & 9 \\ 3 & 9 & 3 \end{pmatrix}$$

auf Tridiagonalgestalt. Es genügt ein Schritt zur Transformation auf HESSENBERGgestalt, die wegen der Symmetrie von A mit der gewünschten Tridiagonalgestalt übereinstimmt. Dabei ist $j = 1$ und $n = 3$, außerdem $A_1 = A$.

Für U_1 werden β_1 und \hat{w}_1 benötigt, um $\begin{pmatrix} \alpha_{21} \\ \alpha_{31} \end{pmatrix} = \begin{pmatrix} 4 \\ 3 \end{pmatrix}$ zu transformieren. Mit

$$\sigma_1 = \sqrt{\sum_{k=j+1}^n |\alpha_{kj}^{(j)}|^2} = \sqrt{4^2 + 3^2} = \sqrt{25} = 5$$

ist

$$\beta_1 = \frac{1}{\sigma_1(\sigma_1 + |\alpha_{21}^{(1)}|)} = \frac{1}{5(5 + 4)} = \frac{1}{45}$$

und

$$\hat{w}_1 = \begin{pmatrix} \alpha_{21} \\ \alpha_{31} \end{pmatrix} + \sigma_1 \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 4 \\ 3 \end{pmatrix} + \begin{pmatrix} 5 \\ 0 \end{pmatrix} = \begin{pmatrix} 9 \\ 3 \end{pmatrix}.$$

Von der Transformation $A_2 = U_1 A U_1$ mit

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$$

$$A_2 = \begin{pmatrix} I & 0 \\ 0 & \hat{U}_1 \end{pmatrix} \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & \hat{U}_1 \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \hat{U}_1 \\ \hat{U}_1 A_{21} & \hat{U}_1 A_{22} \hat{U}_1 \end{pmatrix}$$

sind bereits $A_{11} = 1$ und $\hat{U}_1 A_{21} = -\sigma_1 \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} -5 \\ 0 \end{pmatrix}$ bekannt. Die Symmetrie von A liefert $A_{12} \hat{U}_1 = (-5, 0)$, so daß nur $\hat{U}_1 A_{22} \hat{U}_1$ zu bestimmen ist:

$$\begin{aligned} \hat{U}_1 A_{22} \hat{U}_1 &= (I - \beta_1 \hat{w}_1 \hat{w}_1^H) A_{22} (I - \beta_1 \hat{w}_1 \hat{w}_1^H) \\ &= A_{22} - \beta_1 \hat{w}_1 \hat{w}_1^H A_{22} - A_{22} \beta_1 \hat{w}_1 \hat{w}_1^H + (\beta_1 \hat{w}_1 \hat{w}_1^H) A_{22} (\beta_1 \hat{w}_1 \hat{w}_1^H) \\ &= \begin{pmatrix} -3 & 9 \\ 9 & 3 \end{pmatrix} - \frac{1}{45} \begin{pmatrix} 9 \\ 3 \end{pmatrix} (9, 3) \begin{pmatrix} -3 & 9 \\ 9 & 3 \end{pmatrix} \\ &\quad - \begin{pmatrix} -3 & 9 \\ 9 & 3 \end{pmatrix} \frac{1}{45} \begin{pmatrix} 9 \\ 3 \end{pmatrix} (9, 3) \\ &\quad + \frac{1}{45} \begin{pmatrix} 9 \\ 3 \end{pmatrix} (9, 3) \begin{pmatrix} -3 & 9 \\ 9 & 3 \end{pmatrix} \frac{1}{45} \begin{pmatrix} 9 \\ 3 \end{pmatrix} (9, 3) \\ &= \begin{pmatrix} -3 & 9 \\ 9 & 3 \end{pmatrix} - \frac{1}{45} \begin{pmatrix} 9 \\ 3 \end{pmatrix} (0, 90) - \frac{1}{45} \begin{pmatrix} 0 \\ 90 \end{pmatrix} (9, 3) \\ &\quad + \frac{270}{(45)^2} \begin{pmatrix} 9 \\ 3 \end{pmatrix} (9, 3) \\ &= \begin{pmatrix} -3 & 9 \\ 9 & 3 \end{pmatrix} - \frac{1}{45} \begin{pmatrix} 0 & 810 \\ 0 & 270 \end{pmatrix} - \frac{1}{45} \begin{pmatrix} 0 & 0 \\ 810 & 270 \end{pmatrix} + \frac{270}{(45)^2} \begin{pmatrix} 81 & 27 \\ 27 & 9 \end{pmatrix} \\ &= \begin{pmatrix} -3 & 9 \\ 9 & 3 \end{pmatrix} - \begin{pmatrix} 0 & 18 \\ 0 & 6 \end{pmatrix} - \begin{pmatrix} 0 & 0 \\ 18 & 6 \end{pmatrix} + \frac{2}{5} \begin{pmatrix} 27 & 9 \\ 9 & 3 \end{pmatrix} \\ &= \begin{pmatrix} -3 & -9 \\ -9 & -9 \end{pmatrix} + \frac{6}{5} \begin{pmatrix} 9 & 3 \\ 3 & 1 \end{pmatrix} \\ &= \frac{1}{5} \begin{pmatrix} 39 & -27 \\ -27 & -39 \end{pmatrix} \end{aligned}$$

Insgesamt ist somit

$$A_2 = \frac{1}{5} \begin{pmatrix} 5 & -25 & 0 \\ -25 & 39 & -27 \\ 0 & -27 & -39 \end{pmatrix}.$$

□

Eine entsprechende Transformation auf untere Hessenbergform ist genauso möglich. Man arbeitet von rechts die Zeilen ab.

5.3 Eigenwerte einer hermiteschen Tridiagonalmatrix

Sei

$$T = \begin{pmatrix} \alpha_1 & \beta_1 & & & \\ \gamma_1 & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \beta_{n-1} \\ & & & \gamma_{n-1} & \alpha_n \end{pmatrix} \quad \begin{array}{l} \gamma_i = \bar{\beta}_i \\ \alpha_i \in \mathbb{R} \end{array} \quad (5.3)$$

Wir setzen voraus: $\gamma_i \neq 0$, $i = 1, \dots, n-1$.

Andernfalls zerfällt die Tridiagonalmatrix in kleinere Tridiagonaluntermatrizen, deren Eigenwertproblem gesondert betrachtet werden kann.

Satz 5.3.1 *Ist $T \in \mathbb{C}^{n \times n}$ eine hermitesche Tridiagonalmatrix der Form (5.3) und $\gamma_i \neq 0$ für $i = 1, \dots, n-1$, dann hat T nur einfache reelle Eigenwerte.*

Beweis: *Eine hermitesche Matrix hat ein vollständiges Eigenvektorsystem. Es ist also für jeden Eigenwert geometrische Vielfachheit = algebraische Vielfachheit. Nach Voraussetzung enthält jedoch die Matrix $T - \lambda I$ für jedes λ eine invertierbare Untermatrix der Dimension $n-1$ (mit der Nebendiagonalen als Diagonale), also ist die geometrische Vielfachheit stets 1. \square*

Bemerkung 5.3.1 *Ist T eine reelle unsymmetrische Tridiagonalmatrix mit $\beta_i \gamma_i > 0$, $i = 1, \dots, n-1$, dann kann T durch eine Ähnlichkeitstransformation mit der Diagonalmatrix $D = \text{diag}(\delta_i)$, $\delta_1 := 1$, $\delta_{i+1} := \delta_i \sqrt{\beta_i / \gamma_i}$ $\hat{T} := DTD^{-1}$ in eine symmetrische Matrix \hat{T} überführen. Auch solche Matrizen haben also nur reelle einfache Eigenwerte. \square*

Zur Berechnung der Eigenwerte von T benutzen wir den

Satz 5.3.2 Trägheitssatz von Sylvester *Sei $A \in \mathbb{C}^{n \times n}$ hermitisch und $X \in \mathbb{C}^{n \times n}$ regulär. Dann haben $X^H A X$ und A gleichviele Eigenwerte > 0 , $= 0$, < 0 .
Beweis: siehe z.B. bei Falk-Zurmühl, Matrizen. \square*

Übertragen auf $A - \mu I$ heißt das:

$A - \mu I$ und $X^H(A - \mu I)X$ haben gleichviele Eigenwerte > 0 , $= 0$, < 0 ,

d.h. A hat entsprechend viele Eigenwerte

$> \mu$, $= \mu$, $< \mu$ ($\mu \in \mathbb{R}$). Dies Resultat soll auf T (5.3) angewendet werden mit einer Matrix

$$X \in \mathbb{C}^{n \times n} \quad \text{wo} \quad X^{-1} = \begin{pmatrix} 1 & \xi_1 & & & \\ & \ddots & \ddots & & \\ & & \ddots & \ddots & \\ & & & \ddots & \xi_{n-1} \\ & & & & 1 \end{pmatrix}$$

Dabei soll gelten

$$X^H(T - \mu I)X = Q = \text{diag}(q_1, \dots, q_n) \quad q_i \in \mathbb{R}$$

d.h.

$$T - \mu I = (X^H)^{-1}QX^{-1} = \begin{pmatrix} 1 & & & & \\ \bar{\xi}_1 & 1 & & & \\ & \bar{\xi}_2 & 1 & & \\ & & \ddots & \ddots & \\ & & & \bar{\xi}_{n-1} & 1 \end{pmatrix} \begin{pmatrix} q_1 & & & & \\ & q_2 & & & \\ & & q_3 & & \\ & & & \ddots & \\ & & & & q_n \end{pmatrix} \begin{pmatrix} 1 & \xi_1 & & & \\ & 1 & \xi_2 & & \\ & & 1 & \ddots & \\ & & & \ddots & \xi_{n-1} \\ & & & & 1 \end{pmatrix}$$

Die q_i sind also die Quotienten aufeinanderfolgender Hauptabschnitts - Unterdeterminanten von $T - \mu I$. Hieraus ergeben sich die Gleichungen

$$\begin{aligned} q_1 &= \alpha_1 - \mu, & q_1 \xi_1 &= \beta_1 \quad (\Rightarrow q_1 \bar{\xi}_1 = \gamma_1 = \bar{\beta}_1) \\ q_2 + q_1 |\xi_1|^2 &= \alpha_2 - \mu, & q_2 \xi_2 &= \beta_2 \end{aligned}$$

allgemein

$$\begin{aligned} q_k + q_{k-1} |\xi_{k-1}|^2 &= \alpha_k - \mu, \\ q_k \xi_k &= \beta_k, \quad k = 1, \dots, n \end{aligned}$$

mit der Initialisierung

$$\xi_0 = 0 \quad \text{und} \quad q_0 = 1, \quad \text{sowie} \quad \beta_n = 0.$$

Weil $\beta_k \neq 0$ für $k = 1, \dots, n-1$, existiert ξ_k für $q_k \neq 0$ $k = 1, \dots, n-1$, d.h.

$$\xi_k = \beta_k / q_k \quad k = 1, \dots, n-1$$

Wird ein $q_k = 0$, so ersetzt man q_k durch $\epsilon \ll 1$ (d.h. α_k durch $\alpha_k + \epsilon$). Wegen (5.1) ändert dies die Eigenwerte nur um ϵ . Man rechnet also stets gemäß

$$\boxed{q_k = \alpha_k - \mu - |\beta_{k-1}|^2 / q_{k-1} \quad k = 1, \dots, n, \quad q_0 := 1, \quad \beta_0 := 0} \quad (5.4)$$

Nach Satz 5.3.2 gilt für die Anzahlen

$$\#\{k : q_k < 0, \quad 1 \leq k \leq n\} = \#\{\lambda : \lambda \text{ Eigenwert von } T \text{ und } \lambda < \mu\}$$

Dieses Ergebnis kann man unmittelbar zu einer Intervallschachtelungsmethode zur Bestimmung jedes beliebigen Eigenwerts λ_j von T ausnutzen. Ausgehend von der Numerierung $\lambda_1 \leq \dots \leq \lambda_n$ und z.B. der trivialen Einschließung

$$[a_0, b_0] := [-\|T\|_\infty, \|T\|_\infty],$$

die alle Eigenwerte von T enthält, (Satz 5.1.1) setzt man für $s = 0, 1, \dots$

$$\begin{aligned}\mu_s &:= (a_s + b_s)/2 \\ m &:= \#\{q_k : q_k < 0, \text{ berechnet aus (5.4) mit } \mu = \mu_s\} \\ a_{s+1} &:= \begin{cases} a_s & \text{falls } m \geq j \\ \mu_s & \text{sonst} \end{cases} \\ b_{s+1} &:= \begin{cases} \mu_s & \text{falls } m \geq j \\ b_s & \text{sonst} \end{cases}\end{aligned}$$

Dann gilt $\lim_{s \rightarrow \infty} \mu_s = \lambda_j$

Dieses Verfahren, in der Literatur als Bisektionsverfahren bekannt, ist außerordentlich robust und sehr effizient.

Beispiel 5.3.1

$$T = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 3 & 2 & 0 \\ 0 & 2 & 5 & 3 \\ 0 & 0 & 3 & 7 \end{pmatrix}$$

$\lambda_2 \in [1, 2] =: [a_0, b_0]$

s	μ	q_1	q_2	q_3	q_4	m	
0	1.5	-0.500000	3.500000	2.357143	1.681818	1	$\Rightarrow \lambda_2 > 1.5$
1	1.75	-0.750000	2.583333	1.701613	-0.039100	2	$\Rightarrow \lambda_2 < 1.75$
2	1.625	-0.625000	2.975000	2.030462	0.942511	1	
3	1.6875	-0.687500	2.767045	1.866915	0.491712	1	
4	1.71875	-0.718750	2.672554	1.784555	0.237975	1	
5	1.734375	-0.734375	2.627327	1.743165	0.102603	1	
6	1.7421875	-0.742187	2.605181	1.722410	0.032577	1	
7	1.74609375	-0.746094	2.594220	1.712017	-0.003050	2	
8	1.744140625	-0.744141	2.599691	1.717215	0.014816	1	

□

Ist A allgemein nur hermitisch und besitzt A eine Zerlegung

$$A - \mu I = LDL^H \quad D = \text{diag}(\delta_i)$$

mit invertierbarem L , dann ist die Anzahl der negativen δ_i gleich der Anzahl der Eigenwerte von A , die kleiner sind als μ . Die Schwierigkeit in der Anwendung dieser Tatsache besteht bei allgemeinem A in der Tatsache, daß man die Elemente von L nicht aus der Rekursion für die δ eliminieren kann und die Zerlegung numerisch instabil wird, wenn μ in der Nähe eines Eigenwertes einer Hauptuntermatrix von A liegt.

5.4 Bestimmung der Eigenvektoren einer hermiteschen Dreibandmatrix (ERG)

Wir behandeln hier die Frage, wie man das fast singuläre (oder im Glücksfall sogar exakt singuläre) System

$$(A - \lambda I)x = 0, \quad x \neq 0$$

mit einer "guten" Eigenwertnäherung λ behandeln soll. Wir setzen im Folgenden voraus, T sei eine nichtzerfallende hermitesche Dreibandmatrix (d.h. $\gamma_i \neq 0 \quad i = 1, \dots, n-1$ (vgl. 5.3)) und μ eine bis auf Maschinengenauigkeit bestimmte Eigenwertnäherung für einen Eigenwert λ von T (z.B. mit dem Bisektionsverfahren bestimmt bis $\mu_s \leq a_s$ oder $\mu_s \geq b_s$ aufgrund von Rundungseffekten). Wir wissen, daß für beliebiges λ $\text{Rang}(T - \lambda I) \geq n - 1$ und daß keines der Subdiagonalelemente von T verschwindet, so daß die Dreieckszerlegung von $T - \mu I$ mit Zeilenvertauschung vollständig durchführbar ist. Bei Spaltenpivotsuche (diese ist hier unerlässlich) gilt für die Dreieckszerlegung mit der Permutationsmatrix P deshalb

$$P(T - \mu I) = L \cdot R$$

$|\rho_{jj}| \geq |\beta_j|, \quad j = 1, \dots, n-1$. Ist $\mu = \lambda_j$, dann wird bei rundungsfehlerfreier Rechnung $\rho_{nn} = 0$ und eine Lösung x von $Rx = 0$ mit $\xi_n = 1$ wird Eigenvektor von T . In der Praxis kann man aber keineswegs immer ein "kleines" ρ_{nn} beobachten, auch wenn μ eine sehr gute Eigenwertnäherung ist. Folgender Satz gibt Auskunft, wie man dennoch eine gute Eigenvektornäherung finden kann, wenn nur die Eigenwertnäherung brauchbar ist:

Satz 5.4.1 Sei T eine hermitesche $n \times n$ Dreibandmatrix, $P(T - \mu I) = LR$ eine mit Spaltenpivotsuche durchgeführte Dreieckszerlegung, μ eine Eigenwertnäherung für den Eigenwert λ_j von T mit

$$\mu = \lambda_j + \vartheta\delta, \quad \text{wo } |\vartheta| \leq 1, \quad \delta \text{ hinreichend klein.}$$

Alle Subdiagonalelemente von T seien von null verschieden. Dann existiert (mindestens) ein $i \in \{1, \dots, n\}$, so daß die Lösung x_i von

$$Rx_i = e_i \rho_{ii}, \quad x_i = (\xi_{i1}, \dots, \xi_{in})^T$$

(mit $\xi_{nn} := 1$ falls $i = n$ und $\rho_{nn} = 0$)

$$\frac{x_i}{\|x_i\|_2} = \alpha u_j + d \quad \text{mit} \quad \begin{aligned} \|d\|_2 &\leq \frac{n^3 \delta}{\min\{|\lambda_i - \lambda_j|, i \neq j\}} + \mathcal{O}(\delta^2) \\ |\alpha| &= 1 \end{aligned} \quad (5.5)$$

erfüllt, wobei (u_1, \dots, u_n) ein orthonormiertes Eigenvektorsystem von T bezeichnet und $Tu_i = \lambda_i u_i, \quad i = 1, \dots, n$

□

<<

Beweis: Setze $y_i := \rho_{ii} P L^T e_i, \quad U = (u_1, \dots, u_n)$.
Dann wird mit $T = U \Lambda U^H, \quad \Lambda = \text{diag}(\lambda_i)$

$$(T - \mu I)x_i = P^T L R x_i = \rho_{ii} P^T L e_i = y_i$$

Im Falle $\rho_{nn} = 0$ ist nichts mehr zu zeigen. Sei $\rho_{nn} \neq 0$ und

$$x'_i := \frac{1}{\rho_{ii}} x_i, \quad y'_i := \frac{1}{\rho_{ii}} y_i, \quad i = 1, \dots, n$$

und x'_i, y'_i nach den u_1, \dots, u_n entwickelt:

$$x'_i = \sum_{k=1}^n \tilde{\xi}_{ik} u_k, \quad y'_i = \sum_{k=1}^n \tilde{\eta}_{ik} u_k.$$

Dann besteht der Zusammenhang

$$(\lambda_k - \mu) \tilde{\xi}_{ik} = \tilde{\eta}_{ik} \quad i, k = 1, \dots, n.$$

Es gilt (da die Elemente von Le_i betragsmäßig kleinergleich 1 sind)

$$\begin{pmatrix} \tilde{\eta}_{i1} \\ \vdots \\ \tilde{\eta}_{in} \end{pmatrix} = U^H P^T L e_i \Rightarrow |\tilde{\eta}_{ik}| \leq \sqrt{n} \quad i, k = 1, \dots, n$$

und weil $\|Ue_j\|_\infty \geq 1/\sqrt{n}$ und $\|L^{-T}\|_\infty \leq n$

$$\begin{aligned} \max_{i \in \{1, \dots, n\}} |\tilde{\eta}_{ij}| &= \max_{i \in \{1, \dots, n\}} |e_j^T U^H P^T L e_i| \\ &= \|L^T P U e_j\|_\infty \geq \|P U e_j\|_\infty / \|L^{-T}\|_\infty \geq 1/n^{3/2} \end{aligned}$$

Nun ist aber mit

$$\begin{aligned} \sigma &:= \min\{|\lambda_i - \lambda_j| : i \neq j\} > 0 \\ |\tilde{\xi}_{ik}| &= \frac{|\tilde{\eta}_{ik}|}{|\lambda_k - \lambda_j - \delta\vartheta|} \leq \frac{\sqrt{n}}{\sigma - \delta} \quad \text{für } k \neq j, \quad \forall i \end{aligned}$$

(für δ hinreichend klein ist $\sigma - \delta > 0$), während

$$|\tilde{\xi}_{ij}| = \frac{|\tilde{\eta}_{ij}|}{|\delta\vartheta|} \geq \frac{1}{n^{3/2}\delta} \quad \text{für } i \text{ geeignet.}$$

Für dieses i wird

$$\begin{aligned} \|x'_i\|_2 &= \left(\sum_{k=1}^n |\tilde{\xi}_{ik}|^2 \right)^{1/2} = |\tilde{\xi}_{ij}| \left(1 + \sum_{\substack{k=1 \\ k \neq j}}^n \frac{|\tilde{\xi}_{ik}|^2}{|\tilde{\xi}_{ij}|^2} \right)^{1/2} \\ &= |\tilde{\xi}_{ij}| (1 + \vartheta_{ij}) \end{aligned}$$

mit

$$0 \leq \vartheta_{ij} \leq n^4 \delta^2 / (\sigma - \delta)^2.$$

(Man beachte dazu $1 \leq \sqrt{1 + \xi} \leq 1 + \xi$ für $0 \leq \xi \leq 1$.)

Somit

$$\begin{aligned} \frac{x_i}{\|x_i\|_2} &= \frac{x'_i}{\|x'_i\|_2} = \frac{\tilde{\xi}_{ij}}{|\tilde{\xi}_{ij}|(1 + \vartheta_{ij})} u_j + \sum_{\substack{k=1 \\ k \neq j}}^n \frac{\tilde{\xi}_{ik}}{|\tilde{\xi}_{ij}|(1 + \vartheta_{ij})} u_k \\ &= \underbrace{\frac{\tilde{\xi}_{ij}}{|\tilde{\xi}_{ij}|}}_{\alpha} u_j + d \end{aligned}$$

mit

$$\begin{aligned} \|d\|_2 &\leq 1 - \frac{1}{1 + \frac{n^4\delta^2}{(\sigma-\delta)^2}} + \frac{(n-1)\sqrt{nn^{3/2}}\delta}{\sigma-\delta} \\ &\leq \frac{n^3\delta}{\sigma} + \mathcal{O}(\delta^2) \end{aligned}$$

□

>>

Bemerkung 5.4.1 Aufgrund der Herleitung ist klar, daß der Faktor n^3 in der Abschätzung (5.5) vergleichsweise pessimistisch ist. Viel eher kann man hierfür den Faktor 1 annehmen. Man kann zeigen, daß $i = n$ geeignet ist, wenn mit der Partitionierung

$$R = \left(\begin{array}{c|c} R_{11} & r \\ \hline 0 & \rho_{nn} \end{array} \right)$$

$\|R_{11}^{-1}r\|_2$ "klein" (z.B. $\leq n$) ist. In der Praxis geht man so vor, daß man

$$Rx = \rho_{n,n}e_n$$

löst, dies als Startwert für die gebrochene Iteration (s.h.) benutzt und die Rechnung nach einem weiteren Schritt abbricht, nachdem zum erstenmal

$$\|x_i\|_\infty \geq \frac{1}{100n\varepsilon} |\rho_{n,n}| \quad (5.6)$$

galt. Der Faktor $\frac{1}{100n}$ ist dabei ein (etwas willkürlich) gewählter Sicherheitsfaktor (der wahre Fehler in μ ist ja unbekannt!). Falls der Test (5.6) nach drei Schritten nicht erfüllt ist, betrachtet man auch die Eigenwertnäherung μ als "zu schlecht". Man kann ferner zeigen, daß Abänderungen der Elemente von T in der Größenordnung $\varepsilon\|T\|_2$ Fehler in den Eigenvektoren von der Größenordnung

$$\frac{n\varepsilon\|T\|_2}{\min_{i \neq j} |\lambda_i - \lambda_j|}$$

zur Folge haben können (Satz 5.1.8). Satz 5.3.2 stellt also ein ganz ausgezeichnetes Resultat dar. Man kann weiter zeigen, daß auch die Rundungsfehler bei der Dreieckszerlegung von $P(T - \mu I)$ und bei der Lösung von (5.5) die Aussage des Satzes nicht wesentlich ändern. □

Bemerkung 5.4.2 Es besteht kein quantitativer Zusammenhang zwischen der Größe der Außerdiagonalelemente (d.h. $\min_i |\beta_i|$) und $\sigma = \min_{i \neq j} |\lambda_i - \lambda_j|$.

Wilkinson hat ein Beispiel einer 21×21 -Matrix angegeben mit $\beta_i = 1 \ \forall i$ und σ in der Größenordnung von 10^{-9} . Bei Matrizen mit solch "fast zusammenfallenden" Eigenwerten ist die Bestimmung eines einzelnen Eigenvektors ganz außerordentlich schwierig. □

5.5 Direkte Iteration nach v. Mises und Verfahren von Wielandt

Ziel der in diesem Abschnitt beschriebenen Verfahren ist es, zunächst eine Eigenvektornäherung zu finden. Wie man dazu dann eine Eigenwertnäherung bekommen kann, haben wir bereits in Satz 5.1.4 gesehen. Die (für die Praxis allerdings unbrauchbare) Grundversion der **direkten Iteration von v. Mises** lautet:

1. Wähle $x_0 \in \mathbb{C}^n$, $x_0 \neq 0$ geeignet
2. Für $k = 0, 1, 2, \dots$ setze

$$x_{k+1} = Ax_k$$

Satz 5.5.1 *Es sei $A \in \mathbb{C}^{n \times n}$ diagonalähnlich und $Au_i = \lambda_i u_i$, $U = (u_1, \dots, u_n)$ ein vollständiges Eigenvektorsystem von A . Ferner gelte*

$$x_0 = \sum_{i=1}^n \xi_i u_i \quad \text{mit } \xi_1 \neq 0$$

$$|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_n|$$

Dann gilt für die oben konstruierte Vektorfolge $\{x_k\}$

- (i) $x_k = \xi_1 \lambda_1^k (u_1 + \mathcal{O}(|\frac{\lambda_2}{\lambda_1}|^k))$
- (ii) $R(x_k; A) = \lambda_1 (1 + \mathcal{O}(|\frac{\lambda_2}{\lambda_1}|^k))$

□

Beweis: Man beachte, daß

$$\xi_1 = v_1^H x_0$$

gilt, wo v_1^H der biorthonormierte Linkseigenvektor zu λ_1 ist, also die erste Zeile von $(u_1, \dots, u_n)^{-1}$.

$$x_k = A^k x_0 = (u_1, \dots, u_n) \operatorname{diag}(\lambda_1^k, \dots, \lambda_n^k) (u_1, \dots, u_n)^{-1} (u_1, \dots, u_n) \begin{pmatrix} \xi_1 \\ \vdots \\ \xi_n \end{pmatrix}$$

$$= \sum_{i=1}^n \lambda_i^k \xi_i u_i = \xi_1 \lambda_1^k (u_1 + \underbrace{\sum_{i=2}^n \frac{\xi_i}{\xi_1} \left(\frac{\lambda_i}{\lambda_1}\right)^k u_i}_{|\cdot| \leq |\frac{\lambda_2}{\lambda_1}|^k})$$

$$\begin{aligned}
R(x_k; A) &= \frac{x_k^H x_{k+1}}{x_k^H x_k} \\
&= \frac{\bar{\xi}_1 \bar{\lambda}_1^k (u_1^H + \mathcal{O}(|\frac{\lambda_2}{\lambda_1}|^k)) \xi_1 \lambda_1^{k+1} (u_1 + \mathcal{O}(|\frac{\lambda_2}{\lambda_1}|^{k+1}))}{\bar{\xi}_1 \bar{\lambda}_1^k (u_1^H + \mathcal{O}(|\frac{\lambda_2}{\lambda_1}|^k)) \xi_1 \lambda_1^k (u_1 + \mathcal{O}(|\frac{\lambda_2}{\lambda_1}|^k))} \\
&= \lambda_1 (1 + \mathcal{O}(|\frac{\lambda_2}{\lambda_1}|^k))
\end{aligned}$$

□

Bemerkung 5.5.1

- (a) Für $|\lambda_1| \neq 1$ führt die praktische Durchführung der obigen “Grundversion” schnell zu Exponentenüber- oder -unterlauf. Man rechnet stattdessen mit Normierung nach

$$\begin{aligned}
\tilde{x}_{k+1} &:= Ax_k \\
\varrho_k &= x_k^H \tilde{x}_{k+1} \\
x_{k+1} &= \tilde{x}_{k+1} / \|\tilde{x}_{k+1}\|
\end{aligned}$$

mit $x_0 : \|x_0\| = 1$ geeignet gewählt. Es gilt dann

$$x_k = \vartheta_k (u_1 + \mathcal{O}(|\frac{\lambda_2}{\lambda_1}|^k)) \quad |\vartheta_k| = 1$$

und

$$\varrho_k = \lambda_1 (1 + \mathcal{O}(|\frac{\lambda_2}{\lambda_1}|^k))$$

und im hermiteschen Fall sogar

$$\varrho_k = \lambda_1 (1 + \mathcal{O}(|\frac{\lambda_2}{\lambda_1}|^{2k})) .$$

Normiert man x_k auf $(x_k)_j := 1$ für eine Komponente j mit $(u_1)_j \neq 0$, dann ist $\{x_k\}$ konvergent gegen ein Vielfaches von u_1 .

- (b) Satz 5.5.1 gilt entsprechend für $\lambda_1 = \lambda_2 = \dots = \lambda_r$,
 $|\lambda_1| > |\lambda_{r+1}| \geq \dots \geq |\lambda_n|$, falls $x_0 = \sum_{i=1}^n \xi_i u_i$ und $\sum_{i=1}^r |\xi_i| \neq 0$.

- (c) Die Voraussetzung $\xi_1 \neq 0$ bzw. $\sum_{i=1}^r |\xi_i| \neq 0$ wird in der Praxis durch eingeschleppte Rundungsfehler stets erfüllt, auch wenn x_0 ungeeignet war.

- (d) Man kann auf die Voraussetzung “A diagonalähnlich” verzichten. Dann tritt aber an die Stelle des Fehlerterms $\mathcal{O}(|\frac{\lambda_2}{\lambda_1}|^k)$, die ja wenigstens noch Konvergenz mit der Konvergenzgeschwindigkeit der geometrischen Reihe sicherstellt, ein Term $\mathcal{O}(\frac{1}{k})$, die Konvergenzgeschwindigkeit ist dann also sublinear.

- (e) Bei verschiedenen betragsgleichen und betragsdominanten Eigenwerten von A (z.B. betragsdominanter komplexer Eigenwert einer reellen Matrix!) tritt keine Konvergenz ein. Man kennt aber Verallgemeinerungen des Verfahrens auch auf diesen Fall. (simultane Vektoriteration, vgl. Literaturhinweis am Ende dieses Kapitels)

□

Beispiel 5.5.1 Wir betrachten die direkte Iteration für eine symmetrische 4×4 -Matrix mit den Eigenvektoren

$$\sqrt{\frac{2}{n+1}} \left(\sin\left(\frac{ij\pi}{n+1}\right) \right)_{i=1, \dots, n}$$

mit den folgenden Eigenwerten:

1. 10,5,1,10
2. 10,5,1,-10
3. 10,5,1,-9.99
4. 10,5,1,-9.9
5. 10,5,1,-9

Wir versuchen, den jeweils dominanten Eigenwert mit einer Genauigkeitsforderung 10^{-6} mit dem Startvektor $(1, 0, 0, 0)^T$ zu finden. Es ergibt sich folgendes:

Fall	λ_2/λ_1	Schritte	Bemerkung
1.	$5/10 = 0.5$	11	Doppelter Eigenwert
2.			Keine Konv.: zwei betragsgrößte EW
3.	$9.99/10 = 0.999$	7252	
4.	$9.9/10 = 0.99$	723	Faktor 10 zum Fall 3)
5.	$9/10 = 0.9$	70	Faktor 10 zum Fall 4)

Sowohl das Konvergenzverhalten ($\mathcal{O}(|\frac{\lambda_2}{\lambda_1}|^{2k})$) als auch die geforderten Voraussetzungen für das Verfahren lassen sich durch die numerischen Resultate voll bestätigen:

$$0.5^{22} = 2.3 \cdot 10^{-7}, \quad 0.999^{14504} = 4.99 \cdot 10^{-7}, \quad 0.99^{1446} = 4.88 \cdot 10^{-7}, \quad 0.9^{140} = 3.9 \cdot 10^{-7}.$$

Die Konvergenzgeschwindigkeit der direkten Iteration hängt entscheidend von dem Quotienten $|\lambda_2/\lambda_1| < 1$ ab. Seien $\lambda_1, \dots, \lambda_n$ die Eigenwerte von A . μ sei eine Eigenwertnäherung für λ_i und es gelte

$$0 < |\lambda_i - \mu| < |\lambda_j - \mu| \quad \forall j \in \{1, \dots, n\} \setminus \{i\}$$

Dann ist $A - \mu I$ regulär und für die Eigenwerte $\tau_k = \frac{1}{\lambda_k - \mu}$ von $(A - \mu I)^{-1}$ gilt

$$|\tau_i| > |\tau_j| \quad \forall j \in \{1, \dots, n\} \setminus \{i\}$$

Ferner wird $\max_{j \neq i} |\tau_j|/|\tau_i|$ um so kleiner, je besser die Eigenwertnäherung war. Die direkte Iteration für $(A - \mu I)^{-1}$ führt dann also zu schneller Konvergenz. Dies ist die dem Verfahren von Wielandt, der sogenannten “gebrochenen” oder “inversen” Iteration, zugrundeliegende Idee. Entscheidend für die praktische Brauchbarkeit des Verfahrens ist es, daß die Inverse $(A - \mu I)^{-1}$ nicht explizit gebildet zu werden braucht. Vielmehr löst man pro Schritt das lineare Gleichungssystem

$$\begin{aligned}(A - \mu I)\tilde{x}_{k+1} &= x_k \\ x_{k+1} &:= \tilde{x}_{k+1}/\|\tilde{x}_{k+1}\|\end{aligned}$$

was ohne großen Aufwand möglich ist, wenn man (ein für allemal) eine Dreieckszerlegung von $(A - \mu I)$ (zumindest mit Spaltenpivotstrategie!) berechnet hat: Mit

$$P(A - \mu I)Q = L \cdot R$$

rechnet man dann gemäß

$$\begin{aligned}z_k &:= Px_k \\ Lv_k &= z_k \\ Rw_k &= v_k \\ Q^T \tilde{x}_{k+1} &= w_k\end{aligned}$$

Man könnte mit der neu errechneten Eigenvektornäherung \tilde{x}_{k+1} auch eine neue Eigenwertnäherung definieren, eine neue Dreieckszerlegung berechnen usf. Wegen des hohen Rechenaufwandes lohnt sich dies aber in der Regel nicht.

Beispiel 5.5.2 *Gesucht ist der kleinste Eigenwert λ_3 der Matrix*

$$\begin{pmatrix} 30 & 2 & 0 \\ 2 & 20 & 1 \\ 0 & 1 & 10 \end{pmatrix}.$$

Nach dem Kreisesatz von Gerschgorin liegt der kleinste Eigenwert in einem Kreis vom Radius 1 um 10 und ist isoliert. Wir benutzen deshalb den Shift $\mu = 10$. Als Startvektor nehmen wir $x_0 = (0, 0, 1)^T$. Mit dem Shift $\mu = 10$ erhält man die Matrix

$$\tilde{A} = A - \mu I = A - 10I = \begin{pmatrix} 20 & 2 & 0 \\ 2 & 10 & 1 \\ 0 & 1 & 0 \end{pmatrix}.$$

Nun muß das Gleichungssystem $(A - \mu I)x_1 = \tilde{A}x_1 = x_0$ gelöst werden:

$$\begin{array}{ccc|ccc|ccc|ccc|ccc} 20 & 2 & 0 & 0 & 1 & 5 & 1/2 & 0 & 1 & 5 & 1/2 & +0 & 1 & 0 & 0 & -0.1 \\ 2 & 10 & 1 & 0 & \mapsto & 0 & 1 & 0 & 1 & \mapsto & 0 & 1 & 0 & +1 & \mapsto & 0 & 1 & 0 & +1 \\ 0 & 1 & 0 & 1 & 0 & -98 & -10 & 0 & 0 & 0 & 1 & -9.8 & 0 & 0 & 1 & -9.8 \end{array}$$

Damit lautet $x_1 = (-0.1, 1, -9.8)^T$. Mit dem RAYLEIGH-Quotienten erhält man zuerst eine Näherung für den Eigenwert $\sigma_i = \frac{1}{\lambda_i - \mu}$ von $\tilde{A}^{-1} = (A - \mu I)^{-1}$:

$$\sigma_3 \approx R(x_0; \tilde{A}^{-1}) = \frac{x_0^T \tilde{A}^{-1} x_0}{x_0^T x_0} = \frac{x_0^T x_1}{x_0^T x_0} = \frac{-9.8}{1} = -9.8$$

Wegen $\lambda_3 = \mu + \frac{1}{\sigma_3}$ liefert dies als neue Näherung für λ_3 :

$$\lambda_3 \approx \mu + \frac{1}{R(x_0; \tilde{A}^{-1})} = \mu + \frac{x_0^T x_0}{x_0^T x_1} = 10 + \frac{1}{-9.8} \approx 9.89796$$

<<

Bezüglich der Beschaffung eines geeigneten Startvektors gibt der folgende Satz Auskunft:

Satz 5.5.2 Sei $A \in \mathbb{C}^{n \times n}$ diagonalähnlich, $Au_i = \lambda_i u_i \quad i = 1, \dots, n$,
 $\|u_i\|_2 = 1 \quad \forall i$ worin $U = (u_1, \dots, u_n)$ ein vollständiges Eigenvektorsystem von A bezeichnet
und

$$P(A - \mu I)Q = LR$$

P, Q Permutationsmatrizen,

L untere Dreiecksmatrix mit Diagonale 1 und Elementen von Betrag ≤ 1

R obere Dreiecksmatrix

Ferner sei s definiert durch $|\rho_{ss}| = \min_i |\rho_{ii}|$ und $\hat{R} := \text{diag}(\rho_{11}^{-1}, \dots, \rho_{nn}^{-1})R$. Dann gilt, falls μ kein Eigenwert ist

$$\begin{aligned} \min_j |\lambda_j - \mu| &\leq \sqrt{n} |\rho_{ss}| \text{cond}_{\|\cdot\|_2}(U) \\ &\leq \min_j |\lambda_j - \mu| \|L^{-1}\|_2 \|\hat{R}^{-1}\|_2 \text{cond}_{\|\cdot\|_2}(U) \sqrt{n} \end{aligned}$$

Definiert man x_1 durch

$$RQ^T x_1 = \rho_{ss} e_s \quad (\hat{=} x_0 := \rho_{ss} P^T L e_s) \quad (5.7)$$

und den Index k durch $|\xi_k| = \max_i |\xi_i|$, wo

$$x_1 = \sum_{i=1}^n \xi_i u_i$$

dann gilt für den zugehörigen Eigenwert λ_k die Abschätzung

$$|\lambda_k - \mu| \leq n^{3/2} |\rho_{ss}| \text{cond}_{\|\cdot\|_2}(U)$$

(sind also die verschiedenen Eigenwerte von A hinreichend separiert im Vergleich zu $\min_j |\lambda_j - \mu|$, dann wird $|\lambda_k - \mu| = \min_j |\lambda_j - \mu|$ und somit x_1 eine zur inversen Iteration sehr gut geeignete Startnäherung.) □

Beweis: Ändert man den Wert ρ_{ss} in der Dreieckszerlegung ab in 0, dann ist dies äquivalent zur Abänderung von

$$A - \mu I \quad \text{in} \quad B := A - \mu I - \rho_{ss} P^T L e_s e_s^T Q^T$$

und B wird singular. Satz 5.1.8 liefert die Abschätzung

$$\begin{aligned} \left| \underbrace{0}_{\lambda(B)} - \underbrace{(\lambda_i - \mu)}_{\lambda(A - \mu I)} \right| &\leq \text{cond}_{\|\cdot\|_2}(U) |\rho_{ss}| \|P^T L e_s e_s^T Q\|_2 \\ &\leq \sqrt{n} |\rho_{ss}| \text{cond}_{\|\cdot\|_2}(U) \end{aligned}$$

für ein geeignetes i . Sei nun j_0 definiert durch $|\lambda_{j_0} - \mu| = \min_i |\lambda_i - \mu|$.

Es gilt wegen $\|u_{j_0}\|_2 = 1$

$$(A - \mu I)^{-1} u_{j_0} = \frac{1}{\lambda_{j_0} - \mu} u_{j_0}$$

$$\Rightarrow \frac{1}{|\lambda_{j_0} - \mu|} \leq \|(A - \mu I)^{-1}\|_2 = \|(P^T L R Q^T)^{-1}\|_2 \leq \|L^{-1}\|_2 \|\hat{R}^{-1}\|_2 \frac{1}{|\rho_{ss}|}$$

Somit

$$|\rho_{ss}| \leq \min_j |\lambda_j - \mu| \|L^{-1}\|_2 \|\hat{R}^{-1}\|_2$$

Nach Definition von x_1 und des Index k ist

$$1 \leq \|x_1\|_2 \leq n |\xi_k|$$

Weiterhin gilt

$$x_0 = \sum_{i=1}^n \xi_i (\lambda_i - \mu) u_i$$

$$|\xi_k| |\lambda_k - \mu| \leq \|U^{-1} x_0\|_2 = \|U^{-1} \rho_{ss} P^T L e_s\|_2 \leq |\rho_{ss}| \|U^{-1}\|_2 \sqrt{n}$$

$$|\lambda_k - \mu| \leq n^{3/2} |\rho_{ss}| \|U^{-1}\|_2 \leq n^{3/2} |\rho_{ss}| \text{cond}_{\|\cdot\|_2}(U)$$

□

Bemerkung 5.5.2 Wurde die Dreieckszerlegung mit vollständiger Pivotwahl durchgeführt, dann gilt für die Elemente von \hat{R} : $\hat{\rho}_{ii} = 1, |\hat{\rho}_{ij}| \leq 1$.

In diesem Fall kann $\|L^{-1}\|_2 \|\hat{R}^{-1}\|_2$ als eine Funktion von n allein abgeschätzt werden (Üb.). Ist $\mu = \lambda_j$ für ein j , dann wird $\rho_{ss} = 0$ und x_1 selbst wird zugehöriger Eigenvektor. Man erkennt, daß ρ_{ss} linear mit $\min_j |\lambda_j - \mu|$ gegen null geht. Dennoch treten bei obiger Bestimmung von x_1 keine numerischen Probleme auf. Man kann auch hier einen zu Satz 5.4.1 analogen Satz formulieren, d.h. ist $\mu - \lambda_j \approx f(n)\vartheta\varepsilon$, dann ist bereits $x_1 \approx u_j$ bis auf einen Fehler der Ordnung ε , bei dem als Fehlerverstärkungsfaktor allerdings (u.U. große) Terme analog Satz 5.1.8 auftreten. □

>>

Ist A eine nichtzerfallende obere Hessenbergmatrix, dann ist der erste Koordinateneinheitsvektor stets ein geeigneter Startvektor für das von Mises bzw. Wielandtverfahren, weil kein Linkseigenvektor von A die erste Komponente =0 haben kann, wie man leicht durch Widerspruchsbeweis verifiziert.

Das Wielandt–Verfahren wird in der Praxis benutzt, um zu bereits mit hoher Genauigkeit gefundenen Eigenwerten die entsprechenden Eigenvektoren zu bestimmen, allerdings meist in der rudimentären Form, nur x_1 gemäß Satz 5.5.2 zu bestimmen, und nur 1 bis 2 Iterationsschritte folgen zu lassen, und dies auch nur für Hessenbergmatrizen.

Bemerkung 5.5.3 Man kennt auch Verallgemeinerungen dieser Iterationsverfahren zur simultanen Bestimmung mehrerer Eigenvektoren. Dabei muss man in hermiteschen Fall dafür sorgen, daß das Eigenvektorsystem immer orthonormiert bleibt, bzw. im nicht-hermiteschen Fall, wo man dann auch das Linkseigenvektorsystem mitbestimmen muss, Links- und Rechteigenvektoren biorthonormiert. Im hermiteschen Fall lautet das Verfahren zur Bestimmung der p kleinsten Eigenwerte einer positiv definiten symmetrischen Matrix A ("RITZIT" nach Rutishauer): ($X^{(k)}$ ist hier eine $n \times p$ -Matrix)

$$\begin{aligned} AZ^{(k+1)} &= X^{(k)} \quad p \text{ simultane lineare Gleichungssysteme lösen} \\ Z^{(k+1)} &= Q_{k+1}R_{k+1} \quad \text{QR-Zerlegung} \\ R_{k+1}R_{k+1}^T &= V_{k+1}\Lambda_{k+1}^{-2}V_{k+1}^T \quad \text{vollständiges Eigenwertproblem der Dimension } p \text{ lösen} \\ X^{(k+1)} &= Q_{k+1}V_{k+1} \quad \text{Matrixmultiplikation} \end{aligned}$$

und dazu gilt der

Satz 5.5.3 Seien

$$0 < \lambda_1 \leq \dots < \lambda_p < \lambda_{p+1} \leq \dots$$

die Eigenwerte der hermiteschen Matrix A mit den orthonormierten Eigenvektoren x_i , $i = 1, \dots, p$ und $(X^{(0)})^T(x_1, \dots, x_p)$ sei invertierbar. Dann gibt es Konstanten γ_i , so daß

$$|\sin \angle(x_i^{(k)}, x_i)| \leq \gamma_i (\lambda_i / \lambda_{p+1})^k$$

□

Für den nichthermiteschen Fall gibt es eine Verallgemeinerung dieses Verfahrens von Stewart: Simultaneous iteration for computing invariant subspaces of non-Hermitian matrices. Numer. Math. 25, 123-136 (1976).

5.6 Das QR–Verfahren

Das im Folgenden beschriebene QR–Verfahren wird vor allem dann eingesetzt, wenn es gilt, alle Eigenwerte (und evtl. auch Eigenvektoren) einer Matrix zu bestimmen. Aus Aufwandsgründen führt man dieses Verfahren nur an Hessenberg– bzw. Tridiagonalmatrizen durch, d.h. eine allgemeine Matrix wird zunächst auf die entsprechende Gestalt transformiert. (Diese Matrixformen sind invariant gegenüber der im Algorithmus definierten Transformation; Übg.). In der folgenden Darstellung betrachten wir jedoch den allgemeinen Fall und gehen auf rechentechnische Besonderheiten in Bemerkungen ein. Wesentliche Hilfsmittel zum Verständnis des Verfahrens sind der Satz von Schur und das Verfahren von Wielandt, jetzt mit variablen Spektralverschiebungen μ_k . Zunächst zeigen wir

Satz 5.6.1 Satz von Schur Sei $A \in \mathbb{C}^{n \times n}$ beliebig. Dann existiert ein unitäres U , so daß

$$U^H A U = R = \text{obere Dreiecksmatrix}$$

(mit den Eigenwerten von A auf der Diagonalen von R)

□

Beweis: Sei x ein Eigenvektor von A zum Eigenwert λ . Wähle Q unitär mit

$$Q^H x = \|x\| e_1$$

z.B. als eine Householdermatrix, siehe Kapitel 4. Dann ist

$$Q^H A Q = \left(\begin{array}{c|c} \lambda_1 & a^H \\ \hline 0 & \tilde{A} \end{array} \right)$$

denn

$$Q^H A Q e_1 = Q^H A \frac{x}{\|x\|} = Q^H \lambda \frac{x}{\|x\|} = \lambda e_1.$$

\tilde{A} hat die übrigen Eigenwerte von A zu Eigenwerten. Wir wiederholen den Vorgang mit \tilde{A} und definieren die entsprechende Ähnlichkeitstransformation für A durch

$$\left(\begin{array}{c|c} 1 & 0 \\ \hline 0 & \tilde{Q} \end{array} \right)$$

usw. U ist dann Produkt aller dieser Matrizen. □

Beispiel 5.6.1 *Wir berechnen die Schurnormalform der Matrix*

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

Ein Eigenvektor von A ist $\frac{1}{3}(1, 2, 2)^T$. Mit dem gegebenen Eigenvektor wird die erste HOUSEHOLDER-Matrix bestimmt, als

$$U_1 = I - \beta_1 u_1 u_1^T$$

wobei

$$u_1 = \begin{pmatrix} \text{sign}(y_1)(|y_1| + \|y\|_2) \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} \frac{4}{3} \\ \frac{2}{3} \\ \frac{2}{3} \end{pmatrix}$$

und folglich $\beta_1 = \frac{2}{u_1^T u_1} = \frac{3}{4}$ ist. *Es wird natürlich vermieden, die Housholdermatrix explizit aufzustellen, vielmehr berechnet man die Matrix $A_1 = U_1 A U_1$ spaltenweise, bzw. zeilenweise.*

Die i -te Spalte der Matrix $\tilde{A}_1 = U_1 A$ ist dann gegeben durch

$$(\tilde{A}_1)_{\cdot, i} = (U_1 A)_{\cdot, i} = (I - \beta_1 u_1 u_1^T) A_{\cdot, i} = A_{\cdot, i} - \beta_1 (u_1^T A_{\cdot, i}) u_1.$$

Es ergibt sich

$$\tilde{A}_1 = \frac{1}{3} \begin{pmatrix} -6 & -13 & 7 \\ -6 & 1 & 2 \\ 0 & -1 & 1 \end{pmatrix}$$

Die j -te Zeile der Matrix $A_1 = \tilde{A}_1 U_1$ lautet

$$(A_1)_{j, \cdot} = (\tilde{A}_1 U_1)_{j, \cdot} = (\tilde{A}_1)_{j, \cdot} (I - \beta_1 u_1 u_1^T) = (\tilde{A}_1)_{j, \cdot} - \beta_1 ((\tilde{A}_1)_{j, \cdot} u_1) u_1^T.$$

Das ergibt

$$A_1 = \frac{1}{3} \begin{pmatrix} 6 & -7 & 13 \\ 0 & 4 & 5 \\ 0 & 1 & -1 \end{pmatrix}.$$

Im zweiten Schritt muß nun ein Eigenvektor der Restmatrix

$$\bar{A}_1 = \frac{1}{3} \begin{pmatrix} 4 & 5 \\ 1 & -1 \end{pmatrix}$$

bestimmt werden. Das charakteristische Polynom lautet

$$\left(\frac{4}{3} - \lambda\right)\left(-\frac{1}{3} - \lambda\right) - \frac{5}{9} = 0$$

und liefert die Eigenwerte

$$\lambda_2 = \frac{1}{2}(1 + \sqrt{5}) \quad \text{und} \quad \lambda_3 = \frac{1}{2}(1 - \sqrt{5}).$$

Der Eigenvektor zum Eigenwert λ_2 ist

$$v_2 = \begin{pmatrix} 1 \\ -\frac{5}{10} + \frac{3}{10}\sqrt{5} \end{pmatrix}.$$

Die zweite Householdermatrix U_2 wird dann gebildet durch

$$U_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \hat{U}_{11} & \hat{U}_{12} \\ 0 & \hat{U}_{21} & \hat{U}_{22} \end{pmatrix} \quad \text{mit} \quad \hat{U}_2 = \begin{pmatrix} \hat{U}_{11} & \hat{U}_{12} \\ \hat{U}_{21} & \hat{U}_{22} \end{pmatrix} = I - \beta_2 u_2 u_2^T,$$

wobei

$$u_2 = \begin{pmatrix} 2.014485 \\ 0.170820 \end{pmatrix} \quad \text{und} \quad \beta_2 = \frac{2}{u_2^T u_2} = 0.489317.$$

Nach dem selben Schema wie im ersten Schritt kann nun $A_2 = U_2 A_1 U_2$ berechnen ohne U_2 explizit aufzustellen. Es folgt

$$A_2 = \begin{pmatrix} 2 & 1.570365 & 4.664352 \\ 0 & 1.618034 & -1.333333 \\ 0 & 0 & -.618034 \end{pmatrix}.$$

Eine Umformulierung dieses Satzes ist

Satz 5.6.2 Sei $A \in \mathbb{C}^{n \times n}$ bel.; $A^H x = \bar{\lambda} x$, mit $x \neq 0$ und $Q \in \mathbb{C}^{n \times n}$ unitär mit $Qe_n = \alpha x$. Dann gilt

$$Q^H A^H Q e_n = \bar{\lambda} e_n$$

d.h.

$$Q^H A Q = \left(\begin{array}{ccc|c} & & & * \\ & \tilde{A} & & \vdots \\ & & & * \\ \hline 0 & \dots & 0 & \lambda \end{array} \right) \quad \text{mit } \tilde{A} \in \mathbb{C}^{(n-1) \times (n-1)}$$

□

Wir betrachten nun die Anwendung dieses Satzes im Zusammenhang mit ungenauen Eigenvektornäherungen

Wir nehmen nun zunächst einmal an, μ_0 sei eine “gute” Näherung für einen Eigenwert λ von A und daß die Anwendung des Wielandtverfahrens sinnvoll ist.

Sei eine QR -Zerlegung von $A - \mu_0 I$ gegeben.

$$A - \mu_0 I = Q_0 R_0$$

Mit $\mu_0 \rightarrow \lambda$ gilt $\rho_{ii}^{(0)} \rightarrow 0$ für mindestens ein i . Ist A eine obere Hessenbergmatrix mit nicht verschwindenden Subdiagonalelementen (in der Praxis allein interessierender Fall!), dann ist notwendig $i = n$ und deshalb wollen wir im Folgenden diesen Fall betrachten. Sei also $\rho_{nn}^{(0)}$ “sehr klein”. Es wird

$$(A^H - \bar{\mu}_0 I) Q_0 e_n = R_0^H e_n = \bar{\rho}_{nn}^{(0)} e_n \approx 0, \quad (|\bar{\rho}_{nn}^{(0)}| = |\rho_{nn}^{(0)}| \approx 0)$$

d.h. $x_1 := Q_0 e_n$ ist offenbar eine gute Eigenvektornäherung von $(A^H - \bar{\mu}_0 I)$ (d.h. Linkseigenvektornäherung von $A - \mu_0 I$) und geht aus dem Wielandtverfahren für A^H hervor mit der Verschiebung $\bar{\mu}_0$ und $x_0 := \bar{\rho}_{nn}^{(0)} e_n$.

Ferner wird

$$Q_0^H (A^H - \bar{\mu}_0 I) Q_0 e_n \approx 0 \quad \text{d.h.} \quad e_n^T Q_0^H A Q_0 \approx \mu_0 e_n^T$$

d.h. die letzte Zeile von

$$Q_0^H A Q_0 = Q_0^H (Q_0 R_0 + \mu_0 I) Q_0 = R_0 Q_0 + \mu_0 I$$

enthält also außerhalb der Diagonalen nur “kleine” Elemente, vgl. Satz 5.6.2. Da $x_1 = Q_0 e_n$ offenbar eine gute Eigenvektornäherung ist, ist

$$\bar{\mu}_1 := R(x_1; A^H) = e_n^T \underbrace{Q_0^H A^H Q_0}_{=: A_1^H} e_n = \bar{\alpha}_{nn}^{(1)}$$

d.h.

$$\mu_1 := \alpha_{nn}^{(1)} \quad \text{mit } A_1 := Q_0^H A Q_0 = R_0 Q_0 + \mu_0 I$$

eine eventuell bessere Eigenwertnäherung für λ . Wir wiederholen darum den Vorgang für

$$\begin{aligned} A_1 - \mu_1 I & \quad (= R_0 Q_0 + (\mu_0 - \mu_1)I) : \\ A_1 - \mu_1 I & = Q_1 R_1 \end{aligned}$$

Dann

$$\begin{aligned} (A_1^H - \bar{\mu}_1 I) Q_1 e_n & = \bar{\rho}_{nn}^{(1)} e_n \\ (Q_0^H A^H Q_0 - \bar{\mu}_1 I) Q_1 e_n & = \bar{\rho}_{nn}^{(1)} e_n \\ (A^H - \bar{\mu}_1 I) \underbrace{Q_0 Q_1 e_n}_{=: x_2} & = \bar{\rho}_{nn}^{(1)} x_1 \end{aligned}$$

d.h. $x_2 := Q_0 Q_1 e_n$ ist das Ergebnis eines weiteren Wielandtschrittes, jetzt mit einer anderen, eventuell besseren Spektralverschiebung. Dies legt folgende Grundversion des QR-Algorithmus nahe:

Sei $A_0 := (\alpha_{ij}^{(0)}) := A$. Wähle $0 < \delta < 1$ (z.B. $\delta = 10^{-1}$ ist sinnvoll)

Für $k = 0, 1, \dots$

1. Wähle μ_k geeignet, z.B.

$$\mu_k = \begin{cases} 0 & \text{falls } k = 0 \text{ oder } \sum_{j=1}^{n-1} |\alpha_{n,j}^{(k)}|^2 > \delta \sum_{j=1}^{n-1} |\alpha_{n,j}^{(0)}|^2 \\ \alpha_{n,n}^{(k)} & \text{sonst} \end{cases}$$

2. Berechne die QR-Zerlegung

$$A_k - \mu_k I = Q_k R_k$$

3. $A_{k+1} := R_k Q_k + \mu_k I$

Beispiel 5.6.2 Sei

$$A = \begin{pmatrix} 30 & 2 & 0 \\ 2 & 20 & 1 \\ 0 & 1 & 10 \end{pmatrix}$$

Wir schätzen zunächst den kleinsten Eigenwert von A zu 10 unter Benutzung des Kreisatzes. Wir betrachten zunächst die QR-Zerlegung einer oberen Hessenbergmatrix. In jedem Schritt erfolgt eine Multiplikation der Art:

$$P_i A = \begin{pmatrix} I & 0 & 0 \\ 0 & \tilde{P}_i & 0 \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ 0 & A_{22} & A_{23} \\ 0 & A_{32} & A_{33} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ 0 & \tilde{P}_i A_{22} & \tilde{P}_i A_{23} \\ 0 & A_{32} & A_{33} \end{pmatrix}$$

wobei \tilde{P}_i eine 2×2 Matrix ist, die in der Diagonalen ab der i -ten Position steht. Wir sehen, daß bei der Produktbildung nur die Zeilen i und $i+1$ von links mit \tilde{P}_i multipliziert

werden. Analog erkennt man, daß bei dem Produkt RQ die Spalten i und $i+1$ von rechts mit \tilde{P}_i multipliziert werden.

Im Folgenden beschreiben wir nur die wesentlichen Teile \tilde{P}_i der P_i

$$A - 10I = \begin{pmatrix} 20 & 2 & 0 \\ 2 & 10 & 1 \\ 0 & 1 & 0 \end{pmatrix} \Rightarrow \tilde{P}_1 = \nu \begin{pmatrix} 20 & 2 \\ 2 & -20 \end{pmatrix} \text{ mit } \nu = \frac{1}{\sqrt{404}}$$

Es folgt für $A_1 = P_1(A - 10I)$:

$$A_1 = \begin{pmatrix} 1/\nu & 60\nu & 2\nu \\ 0 & -196\nu & -20\nu \\ 0 & 1 & 0 \end{pmatrix} \Rightarrow \tilde{P}_2 = \mu \begin{pmatrix} -196\nu & 1 \\ 1 & 196\nu \end{pmatrix} \text{ mit } \mu = \frac{1}{\sqrt{1 + 196^2\nu^2}}$$

$$R = P_2A_1 = \begin{pmatrix} 1/\nu & 60\nu & 2\nu \\ 0 & 1/\mu & 3920\nu^2\mu \\ 0 & 0 & -20\nu\mu \end{pmatrix}$$

$$RP_1 = \begin{pmatrix} 20 + 120\nu^2 & 2 - 1200\nu^2 & 2\nu \\ 2\nu/\mu & -20\nu/\mu & 3920\nu^2\mu \\ 0 & 0 & -20\nu\mu \end{pmatrix} = \begin{pmatrix} 20.29703 & * & * \\ 0.97539 & -9.75386 & 0.98985 \\ 0 & 0 & -0.10151 \end{pmatrix}$$

Aus Symmetriegründen brauchen wir die Multiplikation nicht ganz auszuführen.

$$RP_1P_2 = \begin{pmatrix} 20.29703 & * & * \\ 0.97539 & 9.80395 & * \\ 0 & -0.01036 & -0.10098 \end{pmatrix}$$

$\Rightarrow \lambda_1 \in [9.88865, 9.90938]$. Dies ist nun bereits eine erhebliche Verbesserung

Zu diesem Verfahren gilt

Satz 5.6.3 Seien die Folgen $\{A_k\}$, $\{Q_k\}$, $\{R_k\}$ durch obigen Algorithmus definiert und μ_k kein Eigenwert von A ($\forall k$). Setze

$$\tilde{Q}_k := Q_0 \cdots Q_k, \quad \tilde{R}_k := R_k \cdots R_0$$

Dann gilt

$$(A - \mu_k I) \cdots (A - \mu_0 I) = \tilde{Q}_k \tilde{R}_k$$

(i) $\tilde{Q}_k e_n = (((A - \mu_k I) \cdots (A - \mu_0 I))^{-1})^H \underbrace{e_n}_{x_0} / \tau_k$

$$|\tau_k| = \|(((A - \mu_k I) \cdots (A - \mu_0 I))^{-1})^H e_n\|$$

(ii) $\alpha_{nn}^{(k)} = R(\tilde{Q}_{k-1} e_n; A) = \lambda_0 + e_n^T \tilde{Q}_{k-1}^H (A - \lambda_0 I) \tilde{Q}_{k-1} e_n$

wenn man $\tilde{Q}_k e_n$ als Eigenvektornäherung zum Eigenwert $\bar{\lambda}_0$ von A^H ansieht. □

Beweis:

$$\begin{aligned}
A_k &= R_{k-1}Q_{k-1} + \mu_{k-1}I = Q_{k-1}^H(A_{k-1} - \mu_{k-1}I)Q_{k-1} + \mu_{k-1}I \\
&= Q_{k-1}^H A_{k-1} Q_{k-1} = \cdots = Q_{k-1}^H \cdots Q_0^H A Q_0 \cdots Q_{k-1} \\
&= \tilde{Q}_{k-1}^H A \tilde{Q}_{k-1}.
\end{aligned}$$

Daher

$$\begin{aligned}
A - \mu_j I &= \tilde{Q}_{j-1} A_j \tilde{Q}_{j-1}^H - \mu_j I = \tilde{Q}_{j-1} (A_j - \mu_j I) \tilde{Q}_{j-1}^H \\
&= \tilde{Q}_{j-1} Q_j R_j \tilde{Q}_{j-1}^H = \tilde{Q}_j R_j \tilde{Q}_{j-1}^H \\
(A - \mu_k I) \cdots (A - \mu_0 I) &= \tilde{Q}_k R_k \tilde{Q}_{k-1}^H \tilde{Q}_{k-1} R_{k-1} \cdots \tilde{Q}_0 \tilde{R}_0 \\
&= \tilde{Q}_k R_k \cdots R_0 = \tilde{Q}_k \tilde{R}_k
\end{aligned}$$

also ist

$$((A - \mu_0 I)^{-1} \cdots (A - \mu_k I)^{-1})^H e_n = (\tilde{R}_k^{-1} \tilde{Q}_k^H)^H e_n = \tilde{Q}_k e_n / \bar{\rho}_{nn}^{(k)}$$

und wegen $\|\tilde{Q}_k e_n\| = 1$ folgt (i).

Sei λ_0 ein Eigenwert von A und

$$v_k := (A^H - \bar{\lambda}_0 I) \tilde{Q}_{k-1} e_n = (\tilde{Q}_{k-1} A_k^H \tilde{Q}_{k-1}^H - \bar{\lambda}_0 I) \tilde{Q}_{k-1} e_n$$

also

$$\begin{aligned}
e_n^T A_k &= \lambda_0 e_n^T + v_k^H \tilde{Q}_{k-1} \\
\alpha_{nn}^{(k)} &= e_n^T A_k e_n = \lambda_0 + v_k^H \tilde{Q}_{k-1} e_n \\
&= e_n^T \tilde{Q}_{k-1}^H (A - \lambda_0 I) \tilde{Q}_{k-1} e_n + \lambda_0 \\
&= R(\tilde{Q}_{k-1} e_n; A)
\end{aligned}$$

□

Es ist also jeweils $\tilde{Q}_k e_n$ Resultat des Wielandt–Verfahrens mit den variablen Verschiebungen $\bar{\mu}_0, \dots, \bar{\mu}_k$ (für A^H) und $\alpha_{nn}^{(k)}$ der zur letzten Eigenvektornäherung gehörende Rayleighquotient. Die Resultate von Satz 5.1.4 sowie Abschnitt 5 zeigen, daß dementsprechend die letzte Zeile von A_k außerhalb der Diagonalen außerordentlich schnell gegen null konvergieren wird, wenn nur μ_0 bereits hinreichend gute Eigenwertnäherung war. Gestartet wird das Verfahren mit Verschiebung 0, bis sich in der letzten Zeile die Konvergenz zu manifestieren beginnt (vgl. obige Steuerung). Dazu gilt folgender Konvergenzsatz:

Satz 5.6.4 *Es sei $A \in \mathbb{C}^{n \times n}$ und für die Eigenwerte λ_i von A gelte*

$$|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n| > 0$$

Ferner sei

$$YA = \Lambda Y, \quad \Lambda := \text{diag}(\lambda_1, \dots, \lambda_n)$$

(d.h. $Y = X^{-1}$, wo X ein vollständiges Eigenvektorsystem von A ist) und Y besitze eine Dreieckszerlegung

$$Y = L_Y R_Y, \quad L_Y = \begin{array}{|c} \diagdown \\ \hline \end{array}, \quad R_Y = \begin{array}{|c} \hline \diagup \\ \hline \end{array}, \quad \text{diag}(L_Y) = (1, \dots, 1)$$

Dann gilt für den QR–Algorithmus mit $\mu_k \equiv 0$

$$\alpha_{ij}^{(k)} \xrightarrow[k \rightarrow \infty]{} 0 \quad \forall i, j \text{ mit } i > j, \quad \alpha_{ii}^{(k)} \xrightarrow[k \rightarrow \infty]{} \lambda_i$$

Mit $q := \max_{i>j} \left| \frac{\lambda_i}{\lambda_j} \right|$ gilt genauer $\alpha_{ij}^{(k)} = \mathcal{O}(q^k)$ $i < j$, $\alpha_{ii}^{(k)} = \lambda_i + \mathcal{O}(q^k)$

□

<<

Beweis:

1. Sei $X := Y^{-1}$, also $A = X\Lambda Y$ und daher

$$A^k = X\Lambda^k Y = X\Lambda^k L_Y \Lambda^{-k} \Lambda^k R_Y$$

Sei

$$X\Lambda^k L_Y \Lambda^{-k} = U_k \hat{R}_k$$

eine QR–Zerlegung mit $\hat{\rho}_{ii}^{(k)} > 0$ $i = 1, \dots, n$ ¹

Dann wird

$$A^k = U_k \underbrace{\hat{R}_k \Lambda^k R_Y}_{=: R_k^*} = U_k R_k^*$$

eine QR–Zerlegung und andererseits ist

$$A^k = \tilde{Q}_{k-1} \tilde{R}_{k-1}$$

ebenfalls eine QR–Zerlegung. Also (die QR–Zerlegung ist eindeutig bis auf unitäre Diagonaltransformation)

$$\exists \Theta_k \in \mathbb{C}^{n \times n} : \quad |\Theta_k| = I, \quad \tilde{Q}_{k-1} = U_k \Theta_k, \quad \tilde{R}_{k-1} = \bar{\Theta}_k R_k^*$$

¹Diese Zerlegung ist eindeutig bestimmt und \hat{R}_k hängt reell differenzierbar von den Real– und Imaginärteilen der Matrix ab (Cholesky–Faktor)

2. Wir analysieren das Grenzverhalten von $\Lambda^k L_Y \Lambda^{-k}$.

Es ist mit $L_Y = (l_{ij})$

$$(\Lambda^k L_Y \Lambda^{-k})_{ij} = \begin{cases} 0 & i < j \\ 1 & i = j \\ \mathcal{O}(|\lambda_i/\lambda_j|^k) & i > j \end{cases} \quad (5.8)$$

also

$$\Lambda^k L_Y \Lambda^{-k} = I + \mathcal{O}(q^k)$$

Mit $X = Q_X R_X$ QR–Zerlegung mit $\rho_{ii}^{(x)} > 0$ folgt

$$X \Lambda^k L_Y \Lambda^{-k} = Q_X R_X (I + \mathcal{O}(q^k)) = U_k \hat{R}_k$$

und daher $\hat{R}_k = R_X + \mathcal{O}(q^k)$, $U_k = Q_X + \mathcal{O}(q^k)$

(denn der Cholesky–Faktor einer positiv definiten Matrix hängt reell differenzierbar von den $2n^2$ Real- und Imaginärteilen der Matrix ab),

3. Grenzverhalten von $\Theta_k A_k \bar{\Theta}_k$

$$\begin{aligned} \Theta_k A_k \bar{\Theta}_k &= \Theta_k \tilde{Q}_{k-1}^H A \tilde{Q}_{k-1} \bar{\Theta}_k = U_k^H A U_k \\ &= (Q_X + \mathcal{O}(q^k))^H A (Q_X + \mathcal{O}(q^k)) \\ &= (R_X X^{-1} + \mathcal{O}(q^k)) X \Lambda X^{-1} (X R_X^{-1} + \mathcal{O}(q^k)) \\ &\quad \text{(wegen } Q_X^H = Q_X^{-1}) \\ &= R_X \Lambda R_X^{-1} + \mathcal{O}(q^k) \end{aligned}$$

Die Diagonalelemente von $R_X \Lambda R_X^{-1}$ sind aber gerade $\lambda_i \Rightarrow$ Beh.

□

Bemerkung 5.6.1

a) Man kann auf die Voraussetzung der Existenz einer Dreieckszerlegung von Y verzichten. Es existiert stets eine Dreieckszerlegung

$$PY = L_Y R_Y, \quad P = \begin{pmatrix} e_{\pi_1}^T \\ \vdots \\ e_{\pi_n}^T \end{pmatrix}$$

mit $l_{ij} = 0$ falls $i > j$ und $\pi_i < \pi_j$

(zugehörige Pivotstrategie: erstes Element $\neq 0$ in der jeweiligen Spalte wird Pivot)

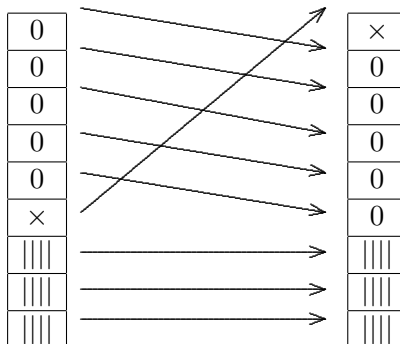


Abbildung 5.6.1

Setzt man dies entsprechend im Beweis von Satz 5.6.4 ein, dann ergibt sich $\alpha_{ii}^{(k)} \rightarrow \lambda_{\pi_i}$, die übrigen Aussagen bleiben unverändert.

- b) Durch eingeschleppte Rundungsfehler werden mehrfache Eigenwerte in der Praxis aufgelöst zu clustern dicht benachbarter Eigenwerte. Solche cluster werden durch Spektralverschiebung aufgelöst in Eigenwerte von unterschiedlichem Betrag, abgesehen den Fall konjugiert komplexer Eigenwerte und reeller Verschiebungen:

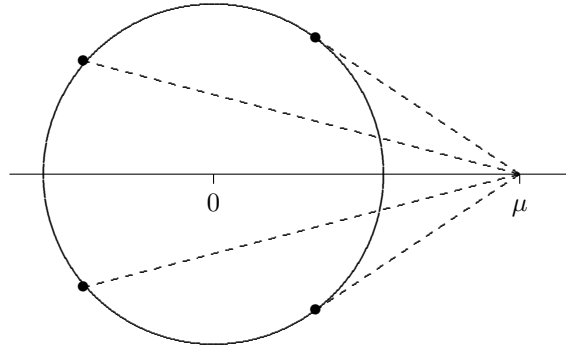


Abbildung 5.6.2

Bei einem solchen Eigenwertcluster nahe bei null sind die Quotienten $|\lambda_i/\lambda_j|$ stets deutlich von 1 verschieden.

- c) Sieht man den obigen Beweis noch einmal durch, so erkennt man, daß im Falle $|\lambda_1| \geq \dots \geq |\lambda_{n-2}| > |\lambda_{n-1}| = |\lambda_n|$ jedenfalls $\alpha_{n,j}^{(k)} \rightarrow 0$, $\alpha_{n-1,j} \rightarrow 0$ für $j = 1, \dots, n-2$. Man kann dann λ_n und λ_{n-1} aus der rechten unteren 2×2 Untermatrix approximieren. Für den Fall konjugiert komplexer Eigenwerte einer reellen Matrix gibt es eine in rein reeller Rechnung verlaufende “Doppelshift–Technik”, die auch dann für Konvergenzbeschleunigung sorgt.

□

>>

Aufgrund der Bemerkungen 5.6.1 a) – c) folgt, daß in der Praxis der QR –Algorithmus bei geeigneten Zusatzmaßnahmen für jede Matrix konvergiert. Wenn die Elemente der letzten Zeile außerhalb der Diagonale hinreichend klein geworden sind, beginnt man mit der Shift–Technik, wodurch sich die Konvergenz enorm beschleunigt.² Sind die Elemente der letzten Zeile praktisch zu null geworden, kann man die QR –Transformation der linken oberen $(n-1) \times (n-1)$ Untermatrix zur Konstruktion der QR –Transformation der vollen Matrix benutzen gemäß

$$\hat{Q} \in \mathbb{C}^{(n-1) \times (n-1)} \longmapsto \left(\begin{array}{c|c} \hat{Q} & 0 \\ \hline 0 & 1 \end{array} \right) \in \mathbb{C}^{n \times n}$$

Diese Technik benötigt man aber nur, wenn man die Eigenvektoren der Grenzdreiecksmatrix (und daraus alle Eigenvektoren von A) bestimmen will. Sonst kann man einfach die Dimension des Problems verkleinern.

²Die Größenordnung der Außerdiagonalelemente quadriert sich pro Schritt. Bei hermiteschen Matrizen ist die Konvergenz sogar normalerweise von dritter Ordnung.

Beispiel 5.6.3 Hier werden die Eigenwerte einer symmetrischen Matrix mit einem Eigenwertcluster bei 1000 berechnet. Das benutzte Verfahren ist eine Variante, das QL-Verfahren, bei dem eine untere Dreiecksmatrix aus einer unteren Hessenberg- bzw. Tridiagonalmatrix erzeugt wird. Das maßgebliche Element, das zu Null gemacht wird, ist also $a_{1,2}$. Die Eingabematrix wird also zunächst wie in Abschnitt 5.2 beschrieben auf Tridiagonalgestalt gebracht. Dies ist hier nicht wiedergegeben.

Matrix A :

```

row/column  1          2          3          4          5
  1  .6110000D+03  .1960000D+03  -.1920000D+03  .4070000D+03  -.8000000D+01
  2  .1960000D+03  .8990000D+03  .1130000D+03  -.1920000D+03  -.7100000D+02
  3  -.1920000D+03  .1130000D+03  .8990000D+03  .1960000D+03  .6100000D+02
  4  .4070000D+03  -.1920000D+03  .1960000D+03  .6110000D+03  .8000000D+01
  5  -.8000000D+01  -.7100000D+02  .6100000D+02  .8000000D+01  .4110000D+03
  6  -.5200000D+02  -.4300000D+02  .4900000D+02  .4400000D+02  -.5990000D+03
  7  -.4900000D+02  -.8000000D+01  .8000000D+01  .5900000D+02  .2080000D+03
  8  .2900000D+02  -.4400000D+02  .5200000D+02  -.2300000D+02  .2080000D+03

row/column  6          7          8
  1  -.5200000D+02  -.4900000D+02  .2900000D+02
  2  -.4300000D+02  -.8000000D+01  -.4400000D+02
  3  .4900000D+02  .8000000D+01  .5200000D+02
  4  .4400000D+02  .5900000D+02  -.2300000D+02
  5  -.5990000D+03  .2080000D+03  .2080000D+03
  6  .4110000D+03  .2080000D+03  .2080000D+03
  7  .2080000D+03  .9900000D+02  -.9110000D+03
  8  .2080000D+03  -.9110000D+03  .9900000D+02

```

Berechnete Eigenwerte und Fehler

```

lam[ 1] = -.10200490184300D+04  lam_exakt[ 1]-lam[ 1]  .9095D-12
lam[ 2] = -.14085954624932D-12  lam_exakt[ 2]-lam[ 2]  .1409D-12
lam[ 3] = .98048640721745D-01  lam_exakt[ 3]-lam[ 3]  -.1731D-12
lam[ 4] = .10000000000000D+04  lam_exakt[ 4]-lam[ 4]  .1137D-12
lam[ 5] = .10000000000000D+04  lam_exakt[ 5]-lam[ 5]  -.3411D-12
lam[ 6] = .10199019513593D+04  lam_exakt[ 6]-lam[ 6]  .0000D+00
lam[ 7] = .10200000000000D+04  lam_exakt[ 7]-lam[ 7]  -.5684D-12
lam[ 8] = .10200490184300D+04  lam_exakt[ 8]-lam[ 8]  -.2274D-12

```

Die Iteration ist im Folgenden dargestellt. k zählt die Iterationen pro Eigenwert und $n(k)$ gibt die laufende Dimension des Problems an, die sich mit jedem akzeptierten Eigenwert verringert. Man erkennt die durch die Shifts erzeugte ausserordentliche Konvergenzgeschwindigkeit.

Iterationsprotokoll :

```

  k  a(1,2)(k)  n(k)
  0  .5843D-06  8
  1  .3034D-13  8
Eigenwertnaeherung akzeptiert !
  0  -.1304D-05  7
  1  -.1621D-28  7
Eigenwertnaeherung akzeptiert !
  0  .1088D-02  6
  1  -.1030D-06  6
  2  .8160D-20  6
Eigenwertnaeherung akzeptiert !
  0  .3903D-02  5
  1  .2011D-12  5
  2  .2529D-51  5
Eigenwertnaeherung akzeptiert !
  0  -.1663D-10  4
  1  .1497D-47  4
Eigenwertnaeherung akzeptiert !
  0  .1877D-01  3

```


Für diese Shift–Technik gilt

Satz 5.6.5 *Es sei $A_0 = A$ eine nichtzerfallende hermitische Tridiagonalmatrix*

$$A_k = \begin{pmatrix} \alpha_1^{(k)} & \beta_1^{(k)} & & & & \\ \bar{\beta}_1^{(k)} & \alpha_2^{(k)} & \ddots & & & \\ & \ddots & \ddots & \ddots & & \\ & & & \ddots & \ddots & \\ & & & & \bar{\beta}_{n-1}^{(k)} & \beta_{n-1}^{(k)} \\ & & & & & \alpha_n^{(k)} \end{pmatrix}$$

werde mit dem QR–Algorithmus berechnet, wobei für alle k der Shift μ_k nach Wilkinson gewählt sei. Dann gilt:

$$\beta_{n-1}^{(k)} \rightarrow 0$$

(d.h. es tritt immer Konvergenz ein).

Falls auch $\beta_{n-2}^{(k)} \rightarrow 0$, $\beta_{n-3}^{(k)} \rightarrow 0$, $\alpha_{n-i}^{(k)} \rightarrow \lambda_{n-i}$, $i = 1, 2$, dann gilt sogar

$$\left| \frac{\beta_{n-1}^{(k+1)}}{(\beta_{n-1}^{(k)})^3 (\beta_{n-2}^{(k)})^2} \right| \rightarrow \frac{1}{|\lambda_{n-1} - \lambda_n|^3 |\lambda_{n-2} - \lambda_n|} = c > 0$$

(d.h. die Konvergenz ist schneller als kubisch). □

Man kann für beliebige hermitische Matrizen eine Übertragung der Wilkinson’schen Shift–Technik angeben, die zu globaler Konvergenz führt, vergleiche bei Parlett, The symmetric eigenproblem.

Im hermitischen Fall hat man also in der Verbindung von Householder– Transformationen auf Tridiagonalgestalt und QR–Algorithmus mit Wilkinson–Shift einen äußerst effizienten Algorithmus³, um alle Eigenwerte und Eigenvektoren zu bestimmen. Die Eigenwerte bilden sich im Laufe der Rechnung in der Diagonalen heraus, falls die obige explizite Shifttechnik angewendet wird; allerdings nicht in einer vorgebbaren Anordnung. Falls letzteres zwingend benötigt wird, muß man eine andere Shifttechnik benutzen. (Ratqr von F.L. Bauer)

Sobald bei der Bestimmung des l -ten Eigenwertes $|\beta_l^{(k)}|$ hinreichend klein geworden ist, etwa wie $|\beta_l^{(k)}| \leq \varepsilon \|A\|$, wird $\alpha_{l+1}^{(k)}$ als Eigenwert akzeptiert. Man braucht dann in den folgenden Schritten jeweils nur noch die linke obere Restmatrix der Dimension $n - l$ zu behandeln. Die Akkumulation aller angewandten unitären Ähnlichkeitstransformationen ergibt die Eigenvektormatrix.

³im Durchschnitt benötigt man zwei QR–Schritte pro Eigenwert.

5.7 Das Lanczos–Verfahren

Beim v. Mises-Verfahren, dem Wielandt-Verfahren und bei der simultanen Vektoriteration wird der k -te Iterationsschritt nur mit Hilfe der Information aus dem $(k - 1)$ -ten Iterationsschritt ausgeführt. Die Grundidee des Lanczos-Verfahrens ist es, die mit der Folge $x^{(0)}, x^{(1)}, \dots, x^{(k)}$ im v. Mises-Verfahren bzw. Wielandt-Verfahren gewonnene Information möglichst gut auszunutzen. Wir betrachten wieder den Fall eines symmetrischen Eigenwertproblems

$$Ax = \lambda x, \quad A = A^T \in \mathbb{R}^{n \times n}.$$

Die Eigenwerte von $Q_j^T A Q_j$ sollen als Näherungen für die Eigenwerte von A dienen. Dabei ist Q_j eine Orthonormalbasis des von $x^{(0)}, Ax^{(0)}, \dots, A^{j-1}x^{(0)}$ aufgespannten Raumes. Die erste Spalte von Q_j wird gleich $x^{(0)} / \|x^{(0)}\|_2$ gesetzt und allgemein gilt mit $X_j = (x^{(0)}, \dots, x^{(j-1)})$:

$$X_j = Q_j R_j \text{ mit einer oberen Dreiecksmatrix } R_j \text{ und } Q_j^T Q_j = I.$$

Die hohe Effizienz des Lanczos-Verfahrens ist dadurch bedingt, daß zum einen die größten bzw. kleinsten Eigenwerte von $Q_j^T A Q_j$ sehr schnell gegen die größten bzw. kleinsten Eigenwerte von A konvergieren (falls $x^{(0)}$ geeignet gewählt ist), zum anderen die Spalten von Q_j sukzessiv durch eine dreigliedrige Rekursion berechnet werden können und $Q_j^T A Q_j = T_j$ Tridiagonalgestalt erhält. Dies bedeutet, daß das Eigenwertproblem für T_j sehr effizient gelöst werden kann. Dabei treten nur Matrix-Vektorprodukte mit der Matrix A auf, sodaß auch Dünnbesetztheit dieser Matrix voll ausgenutzt werden kann. Das Verfahren wird ausschliesslich für hochdimensionale Probleme (Eigenwertprobleme diskretisierter Differentialgleichungen) mit Dimensionen bis 100000 und mehr angewendet. Man muss allerdings beachten, daß die Matrizen Q_j voll besetzt sind. Insofern stellt der verfügbare Hauptspeicher eine gewisse Grenze für die behandelbaren Probleme dar. Es gilt dazu

Satz 5.7.1 Sei A eine reelle symmetrische $n \times n$ -Matrix, $x^{(0)} \neq 0 \in \mathbb{R}^n$ beliebig und $x^{(i)} = A^i x^{(0)}$, $X_j = (x^{(0)}, \dots, x^{(j-1)})$, $Q_j = (q^{(1)}, \dots, q^{(j)})$, sowie für $i = 1, 2, \dots$

$$r^{(i+1)} = Aq^{(i)} - \alpha_i q^{(i)} - \beta_{i-1} q^{(i-1)}$$

mit

$$\begin{aligned}\alpha_i &= (q^{(i)})^T Aq^{(i)}, \\ \beta_i &= \|r^{(i+1)}\|_2,\end{aligned}$$

wo $q^{(1)} = x^{(0)} / \|x^{(0)}\|_2$, $q^{(0)} = 0$, $\beta_0 = 1$,

$$q^{(i+1)} = r^{(i+1)} / \beta_i \quad \text{für } \beta_i \neq 0.$$

Dann gilt: Der Algorithmus ist durchführbar, solange $\beta_i \neq 0$. In diesem Falle ist

$$Q_i^T A Q_i = \begin{bmatrix} \alpha_1 & \beta_1 & & & 0 \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \beta_{i-1} \\ 0 & & & \beta_{i-1} & \alpha_i \end{bmatrix} = T_i,$$

$Q_i^T Q_i = I$, $X_i = Q_i R_i$, R_i obere Dreiecksmatrix, d.h. die $q^{(j)}$ bilden eine Orthonormalbasis des von den $x^{(j)}$ aufgespannten Raumes. \square

<<

Beweis: Mit dem künstlich eingeführten Vektor $q^{(0)}$ gilt

$$\beta_{j-1} q^{(j-1)} + \alpha_j q^{(j)} + \beta_j q^{(j+1)} = Aq^{(j)}, \quad j = 1, \dots, i$$

und deshalb

$$Q_i T_i = A Q_i + \beta_i q^{(i+1)} e_i^T.$$

Wir zeigen nun zuerst, daß die $q^{(k)}$ paarweise orthogonal sind. Dann folgt bereits

$$Q_i^T A_i Q_i = T_i.$$

Die Orthogonalität der $q^{(k)}$ wird induktiv gezeigt. Wegen

$$(q^{(1)})^T q^{(2)} = (q^{(1)})^T Aq^{(1)} - \alpha_1 (q^{(1)})^T q^{(1)} = 0$$

haben wir eine Induktionsverankerung. Seien nun $q^{(1)}, \dots, q^{(k)}$ paarweise orthogonal und normiert. Dann wird für $r^{(k+1)} \neq 0$

$$\begin{aligned}(q^{(j)})^T q^{(k+1)} &= \frac{1}{\|r^{(k+1)}\|} (q^{(j)})^T (Aq^{(k)} - \alpha_k q^{(k)} - \beta_{k-1} q^{(k-1)}) \\ &= \frac{1}{\|r^{(k+1)}\|} \left((r^{(j+1)} + \alpha_j q^{(j)} + \beta_j q^{(j-1)})^T q^{(k)} - \alpha_k (q^{(j)})^T q^{(k)} - \beta_k (q^{(j)})^T q^{(k-1)} \right) \\ &= 0 \quad \text{für } j+1 < k, \text{ also für } j < k-1.\end{aligned}$$

Es bleiben die Fälle $j = k$ und $j = k - 1$. Nun ist wegen der Normierung der $q^{(j)}$ und der Definition von β_{k-1}

$$\begin{aligned} (q^{(k)})^T q^{(k+1)} &= \frac{1}{\|r^{(k+1)}\|} ((q^{(k)})^T A q^{(k)} - \alpha_k (q^{(k)})^T q^{(k)} - \beta_k (q^{(k)})^T (q^{(k-1)})) \\ &= 0 \text{ nach Definition von } \alpha_k \text{ und Induktionsvoraussetzung und} \\ (q^{(k-1)})^T q^{(k+1)} &= \frac{1}{\|r^{(k+1)}\|} ((q^{(k-1)})^T A q^{(k)} - \alpha_k (q^{(k-1)})^T q^{(k)} - \beta_{k-1}) \\ &= \frac{1}{\|r^{(k+1)}\|} (\|r^{(k)}\| (q^{(k)})^T q^{(k)} - \beta_{k-1}) \\ &= 0. \end{aligned}$$

Ferner ist

$$q^{(1)} = x^{(0)} / \|x^{(0)}\|.$$

Sei nun als Induktionsvoraussetzung

$$q^{(k)} \in \text{span}(x^{(0)}, Ax^{(0)}, \dots, A^{k-1}x^{(0)}).$$

Dann ist nach dem Bildungsgesetz für $q^{(k+1)}$

$$q^{(k+1)} \in \text{span}(x^{(0)}, Ax^{(0)}, \dots, A^{k-1}x^{(0)}) \cup \text{span}(Ax^{(0)}, A^2x^{(0)}, \dots, A^kx^{(0)})$$

und somit

$$Q_i = X_i \tilde{R}_i \text{ mit einer invertierbaren oberen Dreiecksmatrix } \tilde{R}_i$$

solange $\|r^{(i+1)}\| \neq 0$. □

>>

Spätestens für $i = n$ bricht das Verfahren (theoretisch) ab mit $r^{(n+1)} = 0$, d.h. $\beta_n = 0$. In diesem Fall wäre A durch eine orthonormale Ähnlichkeitstransformation auf Tridiagonalgestalt transformiert. Zu diesem Zweck ist das Verfahren aber ganz ungeeignet, weil aufgrund der Rundungsfehlereinflüsse in der Praxis die Matrix Q_j sehr schnell ihre Orthonormalität verliert. Dennoch bleibt die Tatsache gültig, daß für maßvoll kleines j die größten bzw. kleinsten Eigenwerte der tatsächlich berechneten Tridiagonalmatrix $\hat{T}_i = \text{tridiag}(\hat{\beta}_{i-1}, \hat{\alpha}_i, \hat{\beta}_i)$, wo $\hat{\alpha}_i, \hat{\beta}_i$ die berechneten Größen bezeichnen, die größten bzw. kleinsten Eigenwerte von A sehr gut approximieren, wenn $x^{(0)}$ geeignet gewählt ist. Selbstverständlich kann auch bei exakter Rechnung der Algorithmus in Abhängigkeit von $x^{(0)}$ vorzeitig abbrechen, z.B. wenn $x^{(0)}$ ein Eigenvektor von A ist, schon im ersten Schritt. Durch eine geeignete Umspeicherung während der Berechnung kann man den Algorithmus mit nur zwei Hilfsvektoren der Länge n durchführen, d.h. er ist auch nur sehr wenig speicheraufwendig.

Algorithmus:

$$v := \frac{x^{(0)}}{\|x^{(0)}\|} = q^{(1)},$$

$$u := 0,$$

$$\beta_0 := 1,$$

$$j := 0.$$

Solange $\beta_j \neq 0$:

$$\text{Für } i = 1, \dots, n : \{ \quad \gamma := u_i; \quad u_i := v_i/\beta_j; \quad v_i := -\gamma\beta_j. \}$$

$$\text{wenn } j \geq 1 \quad q^{(j+1)} := u; v := Au + v,$$

$$j := j + 1,$$

$$\alpha_j := u^T v,$$

$$v := v - \alpha_j u,$$

$$\beta_j := \|v\|_2.$$

Die Matrix A wird dabei niemals geändert. Man benötigt lediglich eine Routine für die Ausführung der Matrix-Vektormultiplikation Ax , wobei man die Besetzungsstruktur von A voll ausnutzen kann. Im Zusammenhang mit der Methode der Finiten Elemente genügt es z.B., die einzelnen Elementsteifigkeitsmatrizen vorliegen zu haben, anstelle der um Größenordnungen aufwendiger zu speichernden Gesamtsteifigkeitsmatrix, um diese Operation auszuführen.

Wenn man die Operation Au ersetzt durch die Gleichungslösung $Aw = u$, hat man das Lanczos-Verfahren in Verbindung mit der inversen Iteration.

Wie bereits erwähnt, dienen die Eigenwerte der aus den berechneten Werten α_i, β_i gebildeten Tridiagonalmatrix

$$T_j = \begin{bmatrix} \alpha_1 & \beta_1 & & & 0 \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \beta_{j-1} \\ 0 & & & \beta_{j-1} & \alpha_j \end{bmatrix}$$

für jeden Wert von j (d.h. für jeden weiteren Lanczos-Schritt) als Näherungen für einige Eigenwerte von A .

Für das Verfahren ist wesentlich, daß man Schätzungen für die Genauigkeit dieser Näherungen aus dem Eigenwertproblem von T_j selbst erhält, samt Näherungen für die dazugehörigen Eigenvektoren von A . Dies ist der Inhalt des folgenden Satzes.

Satz 5.7.2 Sei V_j eine orthonormierte Eigenvektor-Matrix von T_j :

$$V_j^T T_j V_j = \text{diag} [\Theta_1, \dots, \Theta_j],$$

und Y_j sei definiert durch

$$Y_j = Q_j V_j = [y_1, \dots, y_j].$$

Dann gilt

$$\|Ay_i - \Theta_i y_i\|_2 = |\beta_j| |v_{ji}|.$$

□

Ist also $v_{ji}\beta_j$ „klein“, dann bedeutet dies, daß Θ_i eine gute Eigenwertschätzung für A ist mit zugehöriger Eigenvektorschätzung y_i . Dies ist natürlich insbesondere dann der Fall, wenn β_j selbst sehr klein ist. Letzteres tritt allerdings in der Praxis selten auf. Dagegen wird $|v_{ji}|$ oft sehr schnell klein. Einen Hinweis auf die Konvergenzgeschwindigkeit der Eigenwertschätzungen liefert

Satz 5.7.3 Die reell-symmetrische Matrix A besitze die Eigenwerte $\lambda_1 > \dots > \lambda_n$ mit den zugehörigen orthonormierten Eigenvektoren z_1, \dots, z_n . $\Theta_{1,j} > \dots > \Theta_{j,j}$ seien die Eigenwerte von T_j . Ferner seien φ_1, ϱ_1 sowie φ_n, ϱ_n definiert durch

$$\begin{aligned} |\cos \varphi_1| &= |(q^{(1)})^T z_1|, & \varrho_1 &= (\lambda_1 - \lambda_2)/(\lambda_2 - \lambda_n), \\ |\cos \varphi_n| &= |(q^{(1)})^T z_n|, & \varrho_n &= (\lambda_{n-1} - \lambda_n)/(\lambda_1 - \lambda_{n-1}). \end{aligned}$$

und es gelte

$$x^{(0)} = \sum_{i=1}^n \gamma_i z_i \text{ mit } \gamma_1, \gamma_n \neq 0.$$

Dann gilt für $1 \leq j \leq n$

$$\begin{aligned} \lambda_1 &\geq \Theta_{1,j} \geq \lambda_1 - (\lambda_1 - \lambda_n)(\tan \varphi_1/p_{j-1}(1 + 2\varrho_1))^2, \\ \lambda_n &\leq \Theta_{j,j} \leq \lambda_n + (\lambda_1 - \lambda_n)(\tan \varphi_n/p_{j-1}(1 + 2\varrho_n))^2. \end{aligned}$$

Dabei ist p_j das Tschebyscheffpolynom erster Art von genauem Grad j mit der Normierung $p_j(1) = 1$. □

Man erkennt, daß im Fall gut separierter Eigenwerte und $|\tan \varphi_1|, |\tan \varphi_n|$ „klein“ (d.h. $x^{(0)}$ hat einen genügend großen Anteil in der Richtung von z_1 bzw. z_n), die Fehlerschranken sehr schnell klein werden, weil die Tschebyscheffpolynome außerhalb des Intervalls $[-1, 1]$ sehr schnell anwachsen.

Auswertungen dieser Schranken zeigen, daß das Lanczos-Verfahren bezüglich seiner Näherungsgüte der direkten Vektoriteration hoch überlegen ist.

Leider werden die theoretisch so günstigen Eigenschaften des Lanczos-Verfahrens durch die extreme Rundungsfehlerempfindlichkeit des Verfahrens stark nivelliert. Diese Rundungsfehlerempfindlichkeit zeigt sich darin, daß die tatsächlich berechneten Vektoren

$r^{(i)}$ sehr schnell ihre Orthogonalität verlieren. Die Orthogonalität ist aber für alle Aussagen über das Verfahren entscheidend.

Beispiel 5.7.1 *Es sollen die kleinsten Eigenwerte der 50×50 5-Bandmatrix*

$$A = \begin{bmatrix} 1 & -2 & 1 & \cdots & \cdots & \cdots & 0 \\ -2 & 5 & -4 & 1 & & & \vdots \\ 1 & -4 & 6 & -4 & 1 & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & 1 & -4 & 6 & -4 & 1 \\ \vdots & & & 1 & -4 & 5 & -2 \\ 0 & \cdots & \cdots & \cdots & 1 & -2 & 1 \end{bmatrix}$$

berechnet werden. Dieses Problem entsteht aus der Diskretisierung des Differentialgleichungseigenwertproblems

$$\begin{aligned} y^{(4)} &= \lambda y, \\ y''(0) = y'''(0) &= y''(1) = y'''(1) = 0. \end{aligned}$$

(Balkenbiegung eines beidseitig elastisch gelagerten Balkens).

Die 10 kleinsten Eigenwerte von A sind

$$\begin{aligned} \lambda_{50} &= 1.43894475 \cdot 10^{-5}, \\ \lambda_{49} &= 2.29794694 \cdot 10^{-4}, \\ \lambda_{48} &= 1.15966134 \cdot 10^{-3}, \\ \lambda_{47} &= 3.6489003 \cdot 10^{-3}, \\ \lambda_{46} &= 8.85782128 \cdot 10^{-3}, \\ \lambda_{45} &= 0.0182399992, \\ \lambda_{44} &= 0.0335144259, \\ \lambda_{43} &= 0.0566323917, \\ \lambda_{42} &= 0.0897396256, \\ \lambda_{41} &= 0.1351343. \end{aligned}$$

Es wurde das Lanczos-Verfahren in Verbindung mit der inversen Iteration verwendet. Als Startvektor diente dabei $x^{(0)} = [1, 2, 3, \dots]^T$. Schon im dritten Lanczosschritt ergeben sich die Näherungen

$$\begin{aligned} \theta_3 &= \underline{1.43899796} \cdot 10^{-5} && \text{für } \lambda_{50}, \\ \theta_2 &= \underline{2.40714104} \cdot 10^{-4} && \text{für } \lambda_{49}, \\ \theta_1 &= \underline{0.0154333215} && (\text{für } \lambda_{45}). \end{aligned}$$

Die korrekten Stellen sind unterstrichen.

Ferner ist

$$\|I - \hat{Q}_3^T \hat{Q}_3\|_E = 1.75 \cdot 10^{-8},^4$$

d.h. die ersten drei $\hat{q}^{(i)}$ erfüllen die Forderung der Orthonormalität im Rahmen der Rechengenauigkeit von 10 Stellen recht gut ($\hat{q}^{(i)}$ bezeichnet die tatsächlich berechneten Werte).

Im 5. Lanczos-Schritt haben wir

$$\begin{aligned}\Theta_5 &= \underline{1.43899751} \cdot 10^{-5} && \text{für } \lambda_{50}, \\ \Theta_4 &= \underline{2.29795209} \cdot 10^{-4} && \text{für } \lambda_{49}, \\ \Theta_3 &= \underline{1.16123194} \cdot 10^{-3} && \text{für } \lambda_{48}, \\ \Theta_2 &= 4.51726268 \cdot 10^{-3}, \\ \Theta_1 &= 0.125580791, \\ \|I - \hat{Q}_5^T \hat{Q}_5\|_E &= 3.75 \cdot 10^{-4}.\end{aligned}$$

Im 7. Lanczos-Schritt schließlich wird

$$\begin{aligned}\Theta_7 &= \underline{1.43899751} \cdot 10^{-5} && \text{für } \lambda_{50}, \\ \Theta_6 &= \underline{1.43901098} \cdot 10^{-5} && \text{für } \lambda_{50} \quad (!), \\ \Theta_5 &= \underline{2.29795195} \cdot 10^{-4} && \text{für } \lambda_{49}, \\ \Theta_4 &= \underline{1.15966716} \cdot 10^{-3} && \text{für } \lambda_{48}, \\ \Theta_3 &= \underline{3.68169396} \cdot 10^{-3} && \text{für } \lambda_{47}, \\ \Theta_2 &= \underline{0.0116532546} && \text{für } \lambda_{45}, \\ \Theta_1 &= 0.257072226, \\ \|I - \hat{Q}_7^T \hat{Q}_7\|_E &= 1.414 \quad (!),\end{aligned}$$

d.h. die Matrix \hat{Q}_7 ist nun auch nicht annäherungsweise orthonormal. Gleichzeitig tritt eine Näherung für λ_{50} als Doublette auf, obwohl λ_{50} ein einfacher, gut separierter Eigenwert von A ist.

Dies ist typisch für das Lanczos-Verfahren unter Rundungsfehlereinfluß. Das Erkennen solcher falschen Doubletten stellt eine besondere Schwierigkeit für die Anwendung des Verfahrens dar. Man erkennt auch, daß die Eigenwerte nicht alle systematisch angenähert werden, z.B. fehlt eine Näherung für λ_{46} , während eine für λ_{45} vorliegt. Dies liegt am Startvektor. \square

Den Verlust der Orthogonalität bei den $\hat{q}^{(i)}$ könnte man dadurch ausgleichen, daß man jedes berechnete $\hat{q}^{(i)}$ sofort bezüglich aller zuvor berechneten Vektoren $\hat{q}^{(1)}, \dots, \hat{q}^{(i-1)}$ orthogonalisiert. Dies würde aber den Rechen- und auch den Speicherzugriffsaufwand für das Verfahren (die $\hat{q}^{(i)}$ wird man bei großem n gewöhnlich auf einem Hintergrundspeicher halten) enorm erhöhen. Um das Entstehen falscher Doubletten zu vermeiden genügt es, $\hat{q}^{(i)}$ bzgl. der bereits mit hinreichender Genauigkeit gefundenen Eigenvektoren von A zu orthogonalisieren (sogenannte selektive Orthogonalisierung). Als „hinreichend“ genau definiert man dabei einen Defekt in der Einsetzprobe von der Größenordnung der halben Rechengenauigkeit, d.h.

$$\|Ay_i - \Theta_i y_i\|_2 \leq \sqrt{\varepsilon} \|A\|_E,$$

wobei nur die y_i getestet werden, für die

$$|\beta_j| |v_{ji}| \leq \sqrt{\varepsilon} \|A\|_E$$

im j -ten Lanczos-Schritt gilt. Natürlich muß man dazu in jedem Schritt das vollständige Eigenwert / Eigenvektor-Problem der Matrizen T_j lösen, was aber nur wenig aufwendig ist.

⁴Ist A eine beliebige Matrix, so ist $\|A\|_E := (\text{Sp}(AA^*))^{1/2}$

Beispiel 5.7.2 Wir betrachten die Aufgabenstellung von Beispiel 5.7.1, jetzt mit selektiver Orthogonalisierungs-Testgröße

$$\|Ay_i - \Theta_i y_i\| \leq 16 \cdot 10^{-5}.$$

Im 10-ten Lanczos-Schritt ist

$$\begin{aligned} \|I - \hat{Q}_{10}^T \hat{Q}_{10}\|_E &= 4.631 \cdot 10^{-4}, \\ \Theta_{10} &= \underline{1.43899752} \cdot 10^{-5}, \\ \Theta_9 &= \underline{2.29795196} \cdot 10^{-4}, \\ \Theta_8 &= \underline{1.15966203} \cdot 10^{-3}, \\ \Theta_7 &= \underline{3.64890097} \cdot 10^{-3}, \\ \Theta_6 &= \underline{8.85782268} \cdot 10^{-3}, \\ \Theta_5 &= \underline{0.018240746}, \\ \Theta_4 &= \underline{0.0337289387}, \\ \Theta_3 &= 0.0651717673, \\ \Theta_2 &= 0.185803438, \\ \Theta_1 &= 1.74281859. \end{aligned}$$

Mit einem Aufwand, der 10 Schritten der einfachen inversen Iteration im wesentlichen entspricht, sind bereits die sieben kleinsten Eigenwerte von A mit guter Genauigkeit gefunden. \square

5.8 Allgemeine Eigenwertprobleme

Neben dem speziellen Eigenwertproblem tritt häufig auch das allgemeine Eigenwertproblem

$$A x = \lambda B x, \quad x \neq 0, \quad (5.9)$$

auf, allerdings meistens für $A = A^H$, $B = B^H$, B positiv definit. In den Anwendungen treten auch noch allgemeinere Aufgaben auf, etwa

$$(K + \lambda C - \lambda^2 M) x = 0, \quad x \neq 0, \quad (5.10)$$

oder sogar

$$(A - M(\lambda)) x = 0, \quad x \neq 0,$$

mit einer von λ abhängenden Matrix $M(\lambda)$.

Wir wollen zunächst (5.9) im Falle allgemeiner komplexer $n \times n$ Matrizen A , B betrachten. Solange B invertierbar bleibt, kann (5.9) unmittelbar auf das spezielle Eigenwertproblem zurückgeführt werden:

$$A x = \lambda B x \quad \Leftrightarrow \quad B^{-1} A x = \lambda x.$$

Die explizite Durchführung dieser Transformation ist nur dann zu empfehlen, wenn die Dimension des Problems nicht groß und B nicht zu schlecht konditioniert ist. Wenn A

und B hermitisch sind und B positiv definit, wird man die Transformation mit Hilfe der Cholesky-Zerlegung von B bevorzugen, die die hermitische Struktur des Problems erhält:

$$\begin{aligned} Ax &= \lambda Bx, \quad B = LL^H \quad \Leftrightarrow \quad L^{-1}A(L^{-1})^H L^H x = \lambda L^H x \\ \Leftrightarrow \quad Cy &= \lambda y \quad \text{mit} \quad y = L^H x, \quad C = C^H = L^{-1}A(L^{-1})^H. \end{aligned}$$

Zunächst wollen wir uns kurz mit dem allgemeinen Problem (5.9) befassen. Eine hinreichende und notwendige Bedingung zur Existenz von $x \neq 0$ in (5.9) ist ersichtlich

$$\det(A - \lambda B) = 0, \quad (5.11)$$

und dies ist wiederum ein Polynom vom Höchstgrad n in λ . Somit ist jede komplexe Zahl λ Lösung von (5.11) oder es gibt höchstens n solcher Werte. Der erste Fall kann durchaus eintreten, wie das Beispiel

$$A = \begin{bmatrix} 2 & 4 \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 2 & 1 \\ 0 & 0 \end{bmatrix}$$

zeigt.

Wie beim speziellen Eigenwertproblem ist die Lösungsstruktur von (5.9) unmittelbar überschaubar, wenn A und B obere (oder untere) Dreiecksmatrizen sind. Gilt nämlich

$$a_{ij} = 0, \quad j < i \quad \text{und} \quad b_{ij} = 0, \quad j < i,$$

dann ist ersichtlich

$$\begin{aligned} \lambda \in \mathbb{C}, \quad \lambda \text{ beliebig, Lösung von (5.11), wenn } a_{ii} = b_{ii} = 0 \text{ für ein } i, \\ \lambda \in \{a_{ii}/b_{ii} : b_{ii} \neq 0\} \quad \text{sonst.} \end{aligned}$$

Nun ist mit unitärem Q und Z

$$\det(A - \lambda B) = 0 \quad \Leftrightarrow \quad \det(Q^H A Z - \lambda Q^H B Z) = 0.$$

Ferner gilt folgende Verallgemeinerung des Satzes von Schur:

Satz 5.8.1 Zu beliebigen $A, B \in \mathbb{C}^{n \times n}$ existieren unitäre Matrizen Q und Z , so daß

$$T = Q^H A Z \quad \text{und} \quad S = Q^H B Z \quad (5.12)$$

obere Dreiecksgestalt besitzen.

(Zum Beweis vergleiche etwa G.H. Golub, Ch. van Loan: Matrix Computations, J. Hopkins Press.) □

Der aus dem QR-Algorithmus hergeleitete *QZ-Algorithmus von Stewart und Moler* bestimmt iterativ eine Folge von unitären Transformationen, die die Transformation (5.12) annähert.

Im Folgenden beschränken wir uns auf den Fall reeller Matrizen A , B . Der QZ-Algorithmus beginnt mit einer vorbereitenden Transformation

$$A \mapsto \tilde{A} = Q^T A Z, \quad B \mapsto \tilde{B} = Q^T B Z,$$

so daß \tilde{A} eine obere Hessenbergmatrix und \tilde{B} eine obere Dreiecksmatrix wird. Diese vorbereitende Transformation besteht aus zwei Teilen: Zuerst wird B auf obere Dreiecksgestalt gebracht, etwa durch Householder-Transformationen, und die gleiche Transformation wird auf A angewendet. Dann werden in der Reihenfolge

$$(n, 1), (n-1, 1), \dots, (3, 1), (n, 2), \dots, (4, 2), \dots, (n, n-2)$$

jeweils durch eine Givenstransformation von links ein Element in A in null überführt und durch weitere Givenstransformationen von rechts ein dadurch in der Subdiagonale von B eingeführtes Element ungleich Null annulliert, ohne die Nullstruktur in A wieder zu zerstören, z.B.

$$A = \begin{bmatrix} 1 & -1 & 10 & -10 \\ 2 & 1 & 20 & 30 \\ 0 & 3 & 5 & 5 \\ 0 & 4 & -5 & 5 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 2 & 2 & -3 \\ 0 & 1 & 4 & 1 \\ 0 & 0 & 10 & -\frac{15}{4} \\ 0 & 0 & 0 & 5 \end{bmatrix},$$

$$\Omega_1 = \left[\begin{array}{cc|cc} 1 & 0 & & 0 \\ 0 & 1 & & \\ \hline & & 3/5 & 4/5 \\ 0 & & 4/5 & -3/5 \end{array} \right],$$

$$\Omega_1 A = \begin{bmatrix} 1 & -1 & 10 & -10 \\ 2 & 1 & 20 & 30 \\ 0 & 5 & -1 & 7 \\ 0 & 0 & 7 & 1 \end{bmatrix}, \quad \Omega_1 B = \begin{bmatrix} 1 & 2 & 2 & -3 \\ 0 & 1 & 4 & 1 \\ 0 & 0 & 6 & \frac{7}{4} \\ 0 & 0 & 8 & -6 \end{bmatrix},$$

$$\Omega_2 = \left[\begin{array}{cc|cc} 1 & 0 & & 0 \\ 0 & 1 & & \\ \hline & & 0.6 & 0.8 \\ 0 & & 0.8 & -0.6 \end{array} \right],$$

$$\Omega_1 A \Omega_2 = \begin{bmatrix} 1 & -1 & -2 & 14 \\ 2 & 1 & 36 & -2 \\ 0 & 5 & 5 & -5 \\ 0 & 0 & 5 & 5 \end{bmatrix}, \quad \Omega_1 B \Omega_2 = \begin{bmatrix} 1 & 2 & -1.2 & 3.4 \\ 0 & 1 & 3.2 & 2.6 \\ 0 & 0 & 5 & 3.75 \\ 0 & 0 & 0 & 10 \end{bmatrix}.$$

Der weitere Algorithmus geht von der Normalform

- A nichtzerfallende obere Hessenbergmatrix,
- B invertierbare Dreiecksmatrix

aus. Ist eines der Subdiagonalelemente von A null, zerfällt das allgemeine Eigenwertproblem in zwei solche kleinerer Dimension. Ist ein Diagonalelement von B null, kann man das Problem durch weitere Äquivalenztransformationen in eines mit einer zerfallenden Hessenbergmatrix A überführen. Der weitere Algorithmus ist im Prinzip der QR-Algorithmus für die Matrix AB^{-1} , d.h. für eine obere nichtzerfallende Hessenbergmatrix, da A obere nichtzerfallende Hessenbergmatrix und B^{-1} eine obere Dreiecksmatrix ist. Die Matrix AB^{-1} wird jedoch dabei nicht explizit gebildet, vielmehr arbeitet man stets mit orthonormalen Transformationen an A und B . Dies beruht auf den folgenden Zusammenhängen: Ist A Hessenbergmatrix und B invertierbare obere Dreiecksmatrix, so ist

$$Ax = \lambda Bx \Leftrightarrow AB^{-1}y = \lambda y \text{ mit } y = Bx .$$

AB^{-1} ist wieder eine obere Hessenbergmatrix. Nun gilt

Satz 5.8.2 *Ist A eine beliebige $n \times n$ Matrix und B eine nichtzerfallende obere Hessenbergmatrix sowie Q unitär, und*

$$B = Q^H A Q .$$

Dann ist B bis auf eine unitäre diagonale Ähnlichkeitstransformation und Q durch den entsprechenden diagonalen Faktor von rechts eindeutig bestimmt durch die erste Spalte von Q .

Im QZ-Algorithmus arbeitet man mit Matrizen A_k und B_k wie oben beschrieben. Man bestimmt nun den ersten Givens-Transformationsschritt für die QR-Zerlegung der Matrix

$$C_k = A_k B_k^{-1} - \mu_k I$$

Dies ist aber zugleich auch der erste QR-Zerlegungsschritt für

$$A_k - \mu_k B_k$$

(ist also auch möglich sogar für singuläres B_k). Dies legt die erste Spalte einer unitären Matrix \tilde{Q}_k fest. Man wendet diese Transformation nun auf A_k und B_k an und sodann weitere Givens-Transformationen von links und rechts, bis mit so definierten unitären Matrizen \tilde{Q}_k und Z_k wieder gilt

$$A_{k+1} \text{ obere Hessenbergmatrix und } B_{k+1} \text{ obere Dreiecksmatrix}$$

wo

$$A_{k+1} = \tilde{Q}_k A_k Z_k \quad B_{k+1} = \tilde{Q}_k B_k Z_k .$$

Dann ist nach obigem Satz \tilde{Q}_k bis auf unitäre Diagonaltransformation identisch mit dem Q_k aus einem QR-Schritt für C_k und somit gilt mit den obigen A_{k+1} und B_{k+1}

$$C_{k+1} = A_{k+1} B_{k+1}^{-1}$$

bis auf eine unitäre diagonale Ähnlichkeitstransformation, wo C_{k+1} aus C_k mit einem QR-Schritt hervorgeht. In der Praxis verwendet man im reellen Fall in der Regel Doppelschritte mit zwei Shifts aus der rechten unteren 2×2 Matrix von $A_k - \mu_k B_k$. Da der Algorithmus nur orthonormale Transformationen benutzt, ist er sehr rundungsfehlerstabil. Weil er in Verbindung mit der Doppelshifttechnik benutzt werden kann, ist er auch vergleichsweise effizient. Numerische Erfahrungen zeigen, daß man etwa $30n^3$ Operationen braucht, um alle Eigenwerte sowie die angenäherten Matrizen Q und Z aus (5.12) zu bestimmen.

Für ein hermitisches allgemeines Eigenwertproblem ist der QZ-Algorithmus nicht zu empfehlen, da er diese wichtige Eigenschaft des Ausgangsproblems zerstört. Ist die Dimension des Problems n klein, B positiv definit und nicht zu schlecht konditioniert, kann man die Transformation auf ein spezielles hermitisches Problem mittels der Cholesky-Zerlegung von B anwenden. Bei Problemen hoher Dimension mit schwach besetzten Matrizen A und B bietet sich die simultane Vektoriteration oder eine angepaßte Variante des Lanczos-Verfahrens an.

Für allgemeinere nichtlineare Eigenwertprobleme, etwa (5.10), benutzt man gerne folgende Verallgemeinerung der Wielandtiteration mit variablem Shift:

Gegeben sei ein nichtlineares Eigenwertproblem

$$(A - M(\lambda))x = 0, \quad x \neq 0, \quad (5.13)$$

mit $A = A^H$, $M(\lambda) = M^H(\lambda)$, $\frac{d}{d\lambda} M(\lambda) = M'(\lambda)$ positiv definit in einer Umgebung eines Eigenwertes λ_1 von (5.13). (λ_1 heißt Eigenwert der Aufgabe (5.13), falls $\det(A - M(\lambda_1)) = 0$ gilt).

In (5.9) etwa ist

$$M(\lambda) = \lambda B, \quad M'(\lambda) = B,$$

in (5.10)

$$M(\lambda) = -\lambda C + \lambda^2 K, \quad M'(\lambda) = -C + 2\lambda K,$$

d.h. die Voraussetzungen an $M(\lambda)$ bedeuten hier

$$\begin{aligned} B &= B^H && \text{positiv definit, } \lambda \text{ beliebig,} \\ C &= C^H && - C \text{ positiv semidefinit,} \\ K &= K^H && \text{positiv definit, } \lambda > 0. \end{aligned}$$

Mit $\lambda^{(0)} \neq \lambda_1$ und $u^{(0)}$ mit $(u^{(0)})^H M'(\lambda_1) u_1 \neq 0$, worin u_1 ein zu λ_1 gehörender Eigenvektor von (5.13) ist, iteriere man dann gemäß

$$\begin{aligned} (A - M(\lambda^{(\nu)})) w^{(\nu)} &= M'(\lambda^{(\nu)}) u^{(\nu)} \\ &\quad \text{(Gleichungssystem für } w^{(\nu)}) \\ \lambda^{(\nu+1)} &= \lambda^{(\nu)} + \frac{(u^{(\nu)})^H M'(\lambda^{(\nu)}) w^{(\nu)}}{(u^{(\nu)})^H M'(\lambda^{(\nu)}) u^{(\nu)}} \\ u^{(\nu+1)} &= w^{(\nu)} / \|w^{(\nu)}\| \end{aligned}$$

für $\nu = 0, 1, 2, \dots$

Angewandt auf ein spezielles hermitesches Eigenwertproblem ($M'(\lambda) = I$) ist dies gerade das Wielandtverfahren mit dem Rayleighquotienten als Shift. Man kann zeigen, daß dieses Verfahren lokal von zweiter Ordnung konvergiert, wenn λ_1 ein einfacher isolierter Eigenwert von (5.13) ist, $M(\lambda)$ in einer Umgebung von λ_1 dreimal stetig differenzierbar ist und $\lambda^{(0)}$, $u^{(0)}$ hinreichend gute Näherungen für λ_1 und u_1 sind (siehe z.B. bei Höhn, Habilitationsschrift).

Das Kernproblem bei diesem Algorithmus ist die Auflösung des Systems für $w^{(\nu)}$ mit der in der Regel indefiniten und sehr großen schwach besetzten Matrix $A - M(\lambda^{(\nu)})$. Wegen der Indefinitheit versagen hier die Standardmethoden zur iterativen Lösung dieses Systems.

5.9 Die Singulärwertzerlegung (svd)

Im Zusammenhang mit linearen Ausgleichsproblemen in Abschnitt 3.7 haben wir bereits von der sogenannten Singulärwertzerlegung einer allgemeinen rechteckigen Matrix Gebrauch gemacht:

$$A = U \begin{pmatrix} \Sigma \\ 0 \end{pmatrix} V^H \quad (5.14)$$

mit A als komplexer $m \times n$ -Matrix $m \geq n$, U als unitärer $m \times m$ -Matrix, V als unitärer $n \times n$ -Matrix, Σ als Diagonalmatrix mit nichtnegativen Diagonalelementen. 0 ist eine $(m - n) \times n$ -Nullmatrix. Diese Zerlegung ist in vielen Zusammenhängen von großem Nutzen. Anschaulich läßt sie die folgende Deutung zu:

Die durch die Matrix A beschriebene lineare Transformation des \mathbb{C}^n in den \mathbb{C}^m entsteht durch die Hintereinanderschaltung einer Drehspiegelung in \mathbb{C}^n (V^H), einer Achsenmaßstabsänderung (Σ), der kanonischen Einbettung in den größeren Raum \mathbb{C}^m (Anhängen von $m - n$ Nullen) und einer weiteren Drehspiegelung (U) im \mathbb{C}^m .

Beispiel 5.9.1 Für

$$A = \frac{1}{\sqrt{15}} \begin{bmatrix} 1 & 3 \\ 5 & 0 \\ 1 & 3 \end{bmatrix}$$

errechnet man

$$A A^H = \frac{1}{3} \begin{bmatrix} 2 & 1 & 2 \\ 1 & 5 & 1 \\ 2 & 1 & 2 \end{bmatrix}$$

mit den Eigenwerten

$$\lambda_1 = \sigma_1^2 = 2, \quad \lambda_2 = \sigma_2^2 = 1, \quad \lambda_3 = \sigma_3^2 = 0.$$

Die zugehörigen orthonormierten Eigenvektoren sind der Reihe nach

$$u_1 = \frac{1}{\sqrt{6}} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}, \quad u_2 = \frac{1}{\sqrt{3}} \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}, \quad u_3 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}.$$

Es ergibt sich weiter

$$\begin{aligned} A^H U &= \frac{1}{\sqrt{10}} \begin{bmatrix} 4 & -\sqrt{2} & 0 \\ 2 & 2\sqrt{2} & 0 \end{bmatrix} \\ &= (V\Sigma, \begin{pmatrix} 0 \\ 0 \end{pmatrix}) \end{aligned}$$

mit

$$\Sigma = \begin{bmatrix} \sqrt{2} & 0 \\ 0 & 1 \end{bmatrix}, \quad V = \frac{1}{\sqrt{5}} \begin{bmatrix} 2 & -1 \\ 1 & 2 \end{bmatrix}.$$

□

Bei der praktischen Durchführung der Zerlegung (5.14) ist es jedoch nicht sinnvoll, den Weg über das Eigenwertproblem von $A^H A$ oder $A A^H$ zu gehen, weil bei der Aufstellung dieser Matrizen durch eingeschleppte Rundungsfehler Information über die kleinsten Singulärwerte verloren geht. Man berechnet vielmehr die Zerlegung (5.14) auf iterativem Weg. Hierzu gibt es verschiedene Zugänge. Hier beschreiben wir die Lösung nach Golub, Kahan und Reinsch (realisiert in LAPACK und MATLAB). Es gibt zwei Varianten, hier die zum QR-Verfahren passende: Zuerst wird durch geeignete Householder-Transformationen A auf obere Bidiagonalgestalt gebracht:

$$J = U A W$$

mit

$$J = \left. \begin{array}{cccc} * & * & 0 & 0 & \cdots \\ 0 & * & * & 0 & \cdots \\ & \ddots & \ddots & & \\ 0 & & * & * & \\ 0 & & 0 & * & \\ \hline & & 0 & & \end{array} \right\} \begin{array}{l} n \\ m-n \end{array}$$

U und W entstehen dabei durch n bzw. $n - 2$ Householder-Transformationen, und zwar wird zuerst die erste Spalte von A durch eine Transformation von links in ein Vielfaches des ersten Koordinateneinheitsvektors überführt, danach die Elemente $(1, 3)$ bis $(1, n)$ der ersten Zeile in null durch eine Householder-Transformation von rechts, die die erste Spalte unberührt lässt. Daraufhin werden die Elemente der zweiten Spalte unterhalb des Diagonalelements in null überführt, und so fort gemäß dem Beispiel mit $m = 5, n = 3$:

$$A = \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \end{bmatrix} \longrightarrow P_1 A = \begin{bmatrix} * & * & * \\ 0 & * & * \\ 0 & * & * \\ 0 & * & * \\ 0 & * & * \end{bmatrix} \longrightarrow P_1 A \tilde{P}_1 = \begin{bmatrix} * & * & 0 \\ 0 & * & * \\ 0 & * & * \\ 0 & * & * \\ 0 & * & * \end{bmatrix} \longrightarrow$$

$$P_2 P_1 A \tilde{P}_1 = \begin{bmatrix} * & * & 0 \\ 0 & * & * \\ 0 & 0 & * \\ 0 & 0 & * \\ 0 & 0 & * \end{bmatrix} \longrightarrow P_3 P_2 P_1 A \tilde{P}_1 = \begin{bmatrix} * & * & 0 \\ 0 & * & * \\ 0 & 0 & * \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

U und W haben also die Form

$$\begin{aligned} U &= P_n \cdots P_1, \\ W &= \tilde{P}_1 \cdots \tilde{P}_{n-2}. \end{aligned}$$

Die Matrix $J^H J$ ist eine Tridiagonalmatrix und der QR-Algorithmus mit Wilkinson-Shift ist global konvergent. Er erhält die Tridiagonalstruktur. Man will aber die Tridiagonalmatrix nicht explizit bilden und transformieren, sondern immer weiter nur an J arbeiten. Dies ist tatsächlich möglich. Grundlage dafür ist folgende Beobachtung von Francis (Francis, J.: The QR transformation. A unitary analogue to the LR transformation. Comput. J. 4, 265–271 (1961,1962)): Sei T eine Tridiagonalmatrix und

$$T - \mu I = \Omega_1 \cdots \Omega_{n-1} R$$

R ist obere Dreiecksmatrix und Ω_i sind die benutzten Givensrotationen zur Annullierung der Subdiagonale von T . (Man beachte, daß Ω_1 der erste Faktor ist, der von links auf $T - \mu I$ angewendet wird.) Die nächste Matrix in der Folge ist dann bekanntlich

$$R\Omega_1 \cdots \Omega_{n-1} + \mu I \stackrel{\text{def}}{=} T_+$$

und Ω_1 ist so bestimmt, daß es von links auf T angewandt das Element (2,1) annulliert und von rechts angewandt das Element (1,2). Die erste Spalte des Produktes

$$\Omega_1 \cdots \Omega_{n-1}$$

stimmt mit der ersten Spalte von Ω_1 überein, da die übrigen Matrizen nur Spalten 2 bis n tangieren. Ist nun W eine beliebige unitäre Matrix, deren erste Spalte mit der von Ω_1 übereinstimmt, hat T kein Subdiagonalelement gleich null (wie wir immer voraussetzen) und ist Folgendes erfüllt:

$$\tilde{T} \stackrel{\text{def}}{=} W^H T W \text{ ist tridiagonal}$$

dann ist

$$\tilde{T} = D^H T_+ D$$

mit einer Diagonalmatrix aus Elementen vom Betrag eins, also identisch bis auf eine triviale Ähnlichkeitstransformation. Wir konstruieren nun eine zweiseitige unitäre Transformation von $J \hat{Q}_k$ (mit $\hat{Q}_j = Q_0 \cdots Q_j$ aus den vorausgegangenen Schritten), die mit Ω_1 von rechts beginnt und die Bidiagonalstruktur erhält. Diese besteht aus einer Wechselfolge von Givensrotationen von rechts und links. Die links auftretenden Operationen heben sich bei der Multiplikation mit der konjugiert komplexen wieder heraus. Es entsteht dann eine neue Tridiagonalmatrix, die genau der Bildung des obigen \tilde{T} entspricht. W ist dabei das Produkt der von rechts arbeitenden Givensrotationen und weil

wir mit Ω_1 beginnen, erfüllt es die Bedingungen von Francis. Somit erhalten wir durch das direkte Arbeiten an $J\hat{Q}_k$ ein Äquivalent zu einem QR-Schritt an der zugehörigen Tridiagonalmatrix und der bekannte Konvergenzsatz für das QR-Verfahren mit Wilkinsonshift ist anwendbar. Dies bedeutet hier, daß das Element $(n, n-1)$ der zugehörigen Tridiagonalmatrix entsprechend dem Element $(n-1, n)$ von J schnell gegen null konvergiert und ein erster Singulärwert gefunden wird. Dann erfolgt die Erniedrigung der Dimension usw. Schematisch sieht diese Transformationsfolge an $J\hat{Q}_k$ so aus (für den Fall $n=5$) (Hier steht wieder J für $J\hat{Q}_k$)

$$J \Omega_1 = \begin{bmatrix} * & * & 0 & 0 & 0 \\ + & * & * & 0 & 0 \\ 0 & 0 & * & * & 0 \\ 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & 0 & * \end{bmatrix} \longrightarrow$$

$$\tilde{\Omega}_1 J \Omega_1 = \begin{bmatrix} * & * & + & 0 & 0 \\ 0 & * & * & 0 & 0 \\ 0 & 0 & * & * & 0 \\ 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & 0 & * \end{bmatrix} \longrightarrow$$

$$\tilde{\Omega}_1 J \Omega_1 \Omega_2 = \begin{bmatrix} * & * & 0 & 0 & 0 \\ 0 & * & * & 0 & 0 \\ 0 & + & * & * & 0 \\ 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & 0 & * \end{bmatrix} \longrightarrow$$

$$\tilde{\Omega}_2 \tilde{\Omega}_1 J \Omega_1 \Omega_2 = \begin{bmatrix} * & * & 0 & 0 & 0 \\ 0 & * & * & + & 0 \\ 0 & 0 & * & * & 0 \\ 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & 0 & * \end{bmatrix} \longrightarrow$$

$$\tilde{\Omega}_2 \tilde{\Omega}_1 J \Omega_1 \Omega_2 \Omega_3 = \begin{bmatrix} * & * & 0 & 0 & 0 \\ 0 & * & * & 0 & 0 \\ 0 & 0 & * & * & 0 \\ 0 & 0 & + & * & * \\ 0 & 0 & 0 & 0 & * \end{bmatrix} \longrightarrow$$

$$\tilde{\Omega}_3 \tilde{\Omega}_2 \tilde{\Omega}_1 J \Omega_1 \Omega_2 \Omega_3 = \begin{bmatrix} * & * & 0 & 0 & 0 \\ 0 & * & * & 0 & 0 \\ 0 & 0 & * & * & + \\ 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & 0 & * \end{bmatrix} \longrightarrow$$

$$\tilde{\Omega}_3 \tilde{\Omega}_2 \tilde{\Omega}_1 J \Omega_1 \Omega_2 \Omega_3 \Omega_4 = \begin{bmatrix} * & * & 0 & 0 & 0 \\ 0 & * & * & 0 & 0 \\ 0 & 0 & * & * & 0 \\ 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & + & * \end{bmatrix} \longrightarrow$$

$$\tilde{\Omega}_4 \tilde{\Omega}_3 \tilde{\Omega}_2 \tilde{\Omega}_1 J \Omega_1 \Omega_2 \Omega_3 \Omega_4 = \begin{bmatrix} * & * & 0 & 0 & 0 \\ 0 & * & * & 0 & 0 \\ 0 & 0 & * & * & 0 \\ 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & 0 & * \end{bmatrix}.$$

In diesem Schema bedeuten „*“ Elemente der Bidiagonalstruktur und „+“ Elemente ungleich 0, die diese Struktur stören und durch die zusätzlichen Givenstransformationen wieder in null überführt werden. Wegen $\tilde{\Omega}_i^H \tilde{\Omega}_i = I$ ist die so implizit erhaltene Transformation von $J^H J$ (bzw. $\hat{Q}_k^H J^H J \hat{Q}_k$) äquivalent mit der Transformation, die ein Schritt des QR-Algorithmus an dieser Tridiagonalmatrix bewirken würde. Der Konvergenzsatz 5.6.5 überträgt sich entsprechend, d.h. im Grenzwert nähert dann J die Matrix $\begin{pmatrix} \Sigma \\ 0 \end{pmatrix}$ aus (5.14) an. Die Akkumulation aller Givens- bzw. Householder-Transformationen von links bzw. rechts entspricht dann der Matrix U bzw. V^H .

Als Konvergenzbedingung erhalten wir lediglich, daß die Bidiagonalmatrix nicht zerfällt, d.h. kein Subdiagonalelement ist null. Ist dies aber der Fall, kann man das Problem wieder in mehrere kleinere Probleme zerlegen.

Eine der wichtigsten Anwendungen der Singulärwertzerlegung liegt in der Lösung singulärer bzw. sehr schlecht konditionierter Ausgleichsaufgaben.

Ist die Zerlegung (5.14) gegeben und Σ invertierbar, dann lautet die Lösung der Aufgabe

$$\begin{aligned} \|A \tilde{x} - b\|_2 &= \min_x \|A x - b\|_2, \\ \tilde{x} &= V(\Sigma^{-1}, 0) U^H b. \end{aligned} \quad (5.15)$$

Ist jedoch Σ nicht invertierbar, d.h. hat A den Rang $r < n$, dann ist die Lösung von (5.15) nicht eindeutig bestimmt. Erst durch geeignete Zusatzbedingungen an x wird die Lösung der Minimumaufgabe wieder eindeutig. Üblich ist in diesem Zusammenhang die Forderung minimaler Länge auch für x , d.h.

$$\|\tilde{x}\|_2 = \min\{ \|y\|_2 : \|A y - b\|_2 \leq \|A x - b\|_2 \text{ für alle } x\}.$$

Die Lösung lautet dann

$$\tilde{x} = V(\Sigma^+, 0) U^H b$$

mit

$$\Sigma^+ = \text{diag}(\sigma_i^+) \quad \text{und} \quad \sigma_i^+ = \begin{cases} 1/\sigma_i & \text{falls } \sigma_i \neq 0 \\ 0 & \text{sonst.} \end{cases}$$

Bemerkung 5.9.1 Die Matrix

$$A^I = V(\Sigma^+, 0) U^H$$

heißt die Moore-Penrose-Pseudoinverse von A und stellt eine Verallgemeinerung des Begriffs „inverse Matrix“ auf nichtinvertierbare und nichtquadratische Matrizen dar. Man kann zeigen, daß A^I folgende vier Bedingungen erfüllt, durch die sie auch eindeutig bestimmt ist:

$$\begin{aligned} (A^I A) &= (A^I A)^H, \\ (A A^I) &= (A A^I)^H, \\ A^I A A^I &= A^I, \\ A A^I A &= A. \end{aligned}$$

□

5.10 Zusammenfassung. Weiterführende Literatur

Die Problemstellung eines Matrixeigenwertproblems tritt in der Praxis in zwei Varianten auf: als vollständiges Eigenwert/Eigenvektorproblem, wo es gilt, alle Eigenwerte und Eigenvektoren zu finden, und als partielles Problem, wo nur einige Eigenwerte mit Eigenvektoren gesucht sind, in den technischen Anwendungen in der Regel die kleinsten und in den Anwendungen in der Stochastik die grössten Eigenwerte. Das vollständige Eigenwertproblem tritt in der Regel nur bei kleineren Dimensionen auf (n maximal im Bereich von einigen hundert). Hier bietet sich mit dem QR-Verfahren ein universell einsetzbares Instrument an, daß bei geeigneter Implementierung quasi als "black box" nutzbar ist. Die Methoden zur Bestimmung einzelner Eigenwerte sehr grosser Matrizen, also das Lanczos-Verfahren und seine Verallgemeinerungen sowie die (hier nicht besprochene) simultane Vektoriteration erfordern dagegen eine sinnvolle Wahl der Startvektoren, was nur mit Detailkenntnissen der spezifischen Problemstellung gelingt.

Weiterführende Literatur, wo man auch die Beweise der hier nicht bewiesenen Sätze findet :

1. Bai, Z.; Demmel, J.; Dongarra, J.; Ruhe, A.; van der Vorst, H.: *Templates for the solution of algebraic eigenvalue problems* SIAM 2000 .
2. Golub, G.H.; van Loan, Ch.F. : *Matrix computations* 3rd ed. Baltimore: John Hopkins Univ. Press 1996.
3. Parlett, B.N. : *The symmetric eigenvalue problem* Reprint. Philadelphia, PA: SIAM, Society for Industrial and Applied Mathematics. 1998.
4. Wilkinson, J.H.: *The algebraic eigenvalue problem*. Oxford University Press 1965.
5. Zurmühl, Rudolf; Falk, Sigurd: *Matrizen und ihre Anwendungen*. Teil 1: 6. Aufl. 1992, Teil 2: 5. Aufl. 1986 Berlin: Springer.

Kapitel 6

Iterative Lösung linearer Gleichungssysteme

6.0 Motivation

Viele Probleme der Praxis führen auf die Aufgabe, außerordentlich große lineare Gleichungssysteme mit n im Bereich $10^4 - 10^6$ lösen zu müssen, wobei allerdings die Koeffizientenmatrix A eine sehr spezielle Struktur besitzt; jede Zeile besitzt nur wenige, z.B. 5, von null verschiedene Elemente in einer ganz speziellen Anordnung. In dieser Situation sind die direkten Verfahren aus Gründen des Speicherbedarfs und auch des Rechenaufwandes nicht mehr sinnvoll einsetzbar. Auch ist es häufig nicht einmal sinnvoll, die exakte Lösung eines solchen riesigen Systems zu bestimmen, weil diese selbst nur eine (häufig grobe) Approximation der Lösung eines anderen Problems ist. Dies ist z.B. immer der Fall bei den Gleichungssystemen, die bei der Diskretisierung von elliptischen Differentialoperatoren in zwei oder drei Raumdimensionen entstehen.

Die Diskretisierung der Poissongleichung mit Dirichletrandbedingungen auf dem Einheitsquadrat und der Gitterweite $h = 1/(N+1)$ führt bereits in zwei Raumdimensionen zu einem System mit N^2 Unbekannten und einer symmetrischen Blocktridiagonalmatrix (bei zeilenweiser oder spaltenweiser Numerierung) mit der Halbbandbreite $N+1$. Wenn man hierauf die Cholesky-Zerlegung anwenden würde, so benötigte man dafür unter Berücksichtigung der Bandstruktur $\frac{1}{2}N^4$ Rechenoperationen und etwa N^3 Speicherplätze, da innerhalb des Bandes bei der Zerlegung ein fast vollständiges "fill in" eintritt. Im üblichen Bereich $50 \leq N \leq 100$ ist dies schon sehr aufwendig. Völlig unmöglich wird dies im dreidimensionalen Fall, wo dann die Halbbandbreite N^2+1 beträgt, der Speicherbedarf also auf N^5 anwächst. Die nachfolgenden Abbildungen zeigen eine solche Matrix und ihren Choleskyfaktor (als obere Dreiecksmatrix)

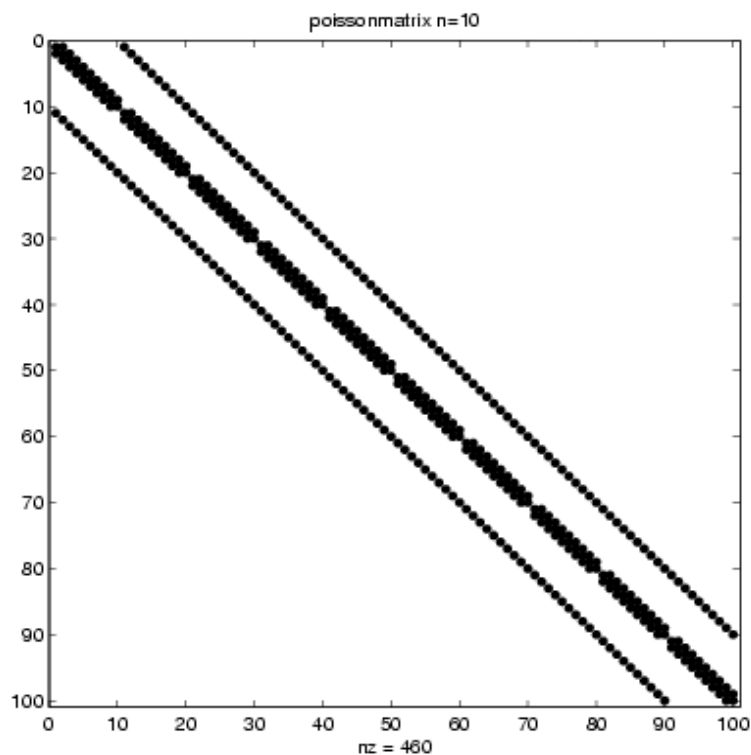


Abbildung 6.0.1

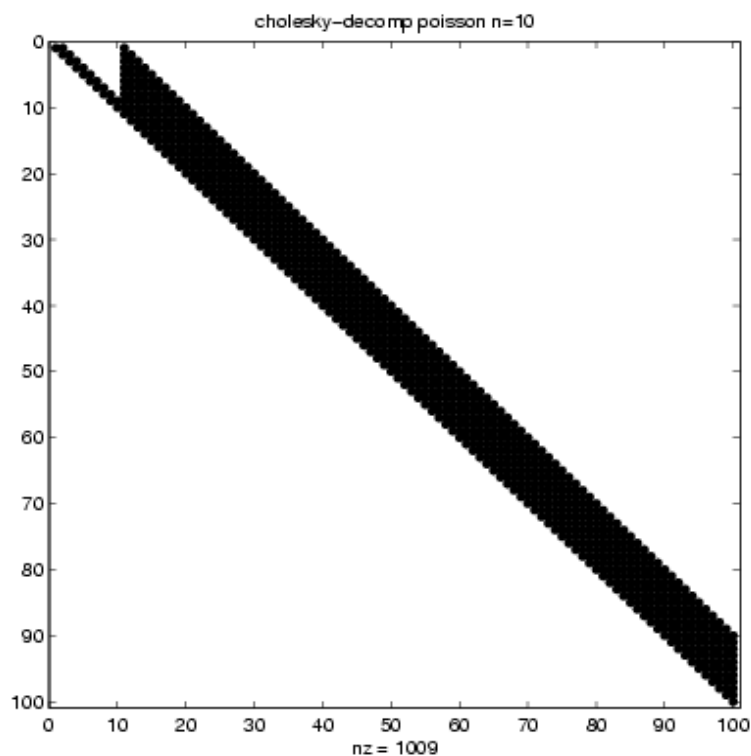


Abbildung 6.0.2

Es ist also in dieser Situation durchaus sinnvoll, nach einfach strukturierten iterativen Verfahren zu suchen, bei denen die Lösung x eines Gleichungssystems durch eine unendliche Folge $\{x_k\}$ angenähert wird, jeder einzelne Iterationsschritt $x_k \rightarrow x_{k+1}$ aber nur sehr wenig Rechenaufwand erfordert.

6.1. ALLGEMEINE ANSÄTZE ZUR ENTWICKLUNG VON ITERATIONSVERFAHRENSPLITTINGV

Im Folgenden benutzen wir als Notation Großbuchstaben für Matrizen, kleine lateinische Buchstaben für Vektoren und kleine griechische Buchstaben für Skalare. Für die Komponenten benutzen wir die entsprechenden Buchstaben, z.B. hat x die Komponenten ξ_1, \dots, ξ_n , die Matrix A hat die Spalten a_1, \dots, a_n und die Elemente $\alpha_{i,j}$. $\rho(A)$ bezeichnet den Spektralradius von A , λ bezeichnet einen Eigenwert. x_k bezeichnet eine Vektorfolge und $\xi_{i,k}$ ist dann die i -te Komponente von x_k .

6.1 Allgemeine Ansätze zur Entwicklung von Iterationsverfahren Splittingverfahren

Im Folgenden wird stets angenommen, daß das vorliegende Gleichungssystem $Ax^* = b$ tatsächlich lösbar ist, b also im Bildraum von A liegt. Eine Grundidee zur Gewinnung iterativer Ansätze zur Lösung linearer Gleichungssysteme $Ax^* = b$ ist die folgende: Man zerlegt A additiv

$$A = M + N$$

Damit ist

$$Ax^* = b \Leftrightarrow Mx^* = b - Nx^*$$

Mit einem Faktor $\omega \neq 0$ und einer beliebigen weiteren Matrix C schließlich

$$Mx^* = b - Nx^* \Leftrightarrow (\omega M + C)x^* = (C - \omega N)x^* + \omega b$$

C, M und ω werden so gewählt, daß $\omega M + C$ regulär ist und eine einfache Struktur hat, sodaß ein Gleichungssystem mit dieser Matrix viel einfacher zu lösen ist als eines mit der Matrix A . Ersetzt man nun auf der rechten Seite der letzten Gleichung x^* durch x_k und links x^* durch x_{k+1} , dann ergibt sich die Iterationsvorschrift

$$(\omega M + C)x_{k+1} = (C - \omega N)x_k + \omega b \quad \text{zu lösen nach } x_{k+1}$$

bzw.

$$x_{k+1} = (\omega M + C)^{-1}((C - \omega N)x_k + \omega b) \stackrel{\text{def}}{=} \Phi(x_k) \quad (6.1)$$

Diese letztere Form dient aber nur theoretischen Zwecken. Nach Konstruktion ist $x^* = \Phi(x^*)$ und daher

$$x_{k+1} - x^* = \underbrace{(\omega M + C)^{-1}(C - \omega N)}_{\stackrel{\text{def}}{=} B(\omega)}(x_k - x^*)$$

d.h. bei beliebig vorgegebener erster Näherung x_0

$$x_k - x^* = (B(\omega))^k(x_0 - x^*) .$$

Es gilt

Satz 6.1.1 Das Iterationsverfahren (6.1) ist genau dann für beliebige x_0 konvergent gegen x^* , wenn

$$\rho(B(\omega)) < 1$$

Beweis: Sei $\rho(B(\omega)) < 1$. ($I - B(\omega)$ ist regulär, daher ist $x^* = B(\omega)x^* + \omega(\omega M + C)^{-1}b$ eindeutig auflösbar nach x^* . Es existiert eine Vektornorm (siehe Einf. in die Num. Math. I) $\|\cdot\|$ auf \mathbb{C}^n sodaß die zugeordnete Matrixnorm

$$\|B(\omega)\| \leq \rho(B(\omega)) + \varepsilon < 1 \quad (\varepsilon > 0 \text{ bel. klein vorgegeben})$$

erfüllt. Somit ist

$$\|x_k - x^*\| \leq \|B(\omega)\|^k \|x_0 - x^*\|, \text{ d.h. } \lim_{k \rightarrow \infty} x_k = x^*$$

Sei umgekehrt nun $\rho(B(\omega)) \geq 1$. Dann existiert ein Eigenwert λ von $B(\omega)$ mit $|\lambda| \geq 1$. Sei $v \neq 0$ ein zugehöriger Eigenvektor. Dann gilt

$$(B(\omega))^k v = \lambda^k v$$

Setze $x_0 = x^* + v$. Dann

$$x_k - x^* = \lambda^k v, \text{ d.h. } \|x_k - x^*\| = |\lambda|^k \|v\| \geq \|v\| > 0 \quad \forall k$$

□

Bekanntlich existiert dann bei beliebig klein vorgegebenem $\varepsilon > 0$ eine Norm, so daß in dieser Norm der Abstand $\|x_k - x^*\|$ um den Faktor $\rho(B(\omega)) + \varepsilon$ pro Schritt verkleinert wird. Es muß also offensichtlich bei der Wahl von M, N, C und ω das Ziel bestehen, $\rho(B(\omega))$ möglichst klein zu machen. Bevor wir uns weiter mit der Theorie der Verfahren, insbesondere mit dem Verhalten von $\rho(B(\omega))$ bei einem wichtigen Sonderfall, befassen, sollen nun die für die Praxis wichtigsten Verfahren hergeleitet werden. Dabei gehen wir von folgender Standardzerlegung der Matrix A aus:

$$A = -L + D - U$$

$$A = \begin{array}{|ccc|} \hline & & -U \\ & D & \\ -L & & \hline \end{array} \quad \begin{array}{l} D \quad \text{Diagonalelemente von } A \\ -U \quad \text{Elemente über der Diagonalen} \\ -L \quad \text{Elemente unterhalb der Diagonalen} \end{array}$$

Dann ergibt

$$1. \quad M \stackrel{\text{def}}{=} D, \quad N \stackrel{\text{def}}{=} -L - U, \quad C = 0, \quad \omega = 1 \text{ das}$$

Gesamtschritt (Jacobi-) Verfahren

$$\xi_{i,k+1} = \xi_{i,k} + \frac{1}{\alpha_{ii}} \left(- \sum_{j=1}^n \alpha_{i,j} \xi_{j,k} + \beta_i \right) \quad \begin{array}{l} i = 1, \dots, n \\ k = 0, 1, 2, \dots \end{array}$$

$$x_{k+1} = D^{-1}((L + U)x_k + b)$$

6.1. ALLGEMEINE ANSÄTZE ZUR ENTWICKLUNG VON ITERATIONSVERFAHRENSPLITTINGV

2. $M \stackrel{\text{def}}{=} -L + D$, $N = -U$, $C = 0$, $\omega = 1$ das

Einzelschritt (Gauß–Seidel)–Verfahren

$$\xi_{i,k+1} = \xi_{i,k} + \frac{1}{\alpha_{ii}} \left(- \sum_{j=1}^{i-1} \alpha_{ij} \xi_{j,k+1} - \sum_{j=i}^n \alpha_{ij} \xi_{j,k} + \beta_i \right) \quad \begin{array}{l} i = 1, \dots, n \\ k = 0, 1, 2, \dots \end{array}$$

$$(-L + D)x_{k+1} = Ux_k + b$$

3. $M \stackrel{\text{def}}{=} -L + D$, $N = -U$, $C = (1 - \omega)D$, $\omega \neq 0$ das

SOR–Verfahren

(sukzessive Überrelaxation, ω : “Relaxationsparameter”)

(für $\omega = 1$ identisch mit dem Gauß–Seidel–Verfahren; der Begriff Überrelaxation wird nur für den Fall $\omega > 1$ benutzt, für $\omega < 1$ “Unterrelaxation”)

In Komponentenschreibweise lautet dieses Verfahren:

$$\xi_{i,k+1} = \frac{\omega}{\alpha_{ii}} \left(\beta_i - \sum_{j=1}^{i-1} \alpha_{ij} \xi_{j,k+1} - \sum_{j=i}^n \alpha_{ij} \xi_{j,k} \right) + \xi_{i,k}. \quad (6.2)$$

In Vektorschreibweise

$$(-\omega L + D)x_{k+1} = ((1 - \omega)D + \omega U)x_k + \omega b.$$

An (6.2) erkennt man leicht die Idee des Verfahrens:

Ausgehend von $(\xi_{1,k+1}, \dots, \xi_{i-1,k+1}, \xi_{i,k}, \dots, \xi_{n,k})^T$ als letzter (“bester”) Näherung für x^* bildet man zunächst $\xi_{i,k+1}^{GS}$ nach dem Gauß–Seidel–Verfahren und berechnet sodann $\xi_{i,k+1}$ durch Vergrößerung oder Verkleinerung der Gauß–Seidel–Korrektur um den Faktor ω . Man kann diese Verfahren auch als iterative Verfahren zur Lösung des Nullstellenproblems

$$F(x) = (F_1(x), \dots, F_n(x))^T = Ax - b = 0$$

deuten. Dann ergibt sich wegen $\alpha_{i,i} = \partial_i F_i(x)$ ein Zusammenhang mit einem komponentenweise durchgeführten Newtonverfahren: Das Jacobiverfahren lautet in dieser Schreibweise nämlich

$$\xi_{i,k+1} = \xi_{i,k} - F_i(x_k) / (\partial_i F_i(x_k))$$

und das SOR-Verfahren

$$\xi_{i,k+1} = \xi_{i,k} - \omega F_i(\xi_{1,k+1}, \dots, \xi_{i-1,k+1}, \xi_{i,k}, \dots, \xi_{n,k}) / (\partial_i F_i(x_k))$$

Man beachte, daß man hier in jedem Teilschritt alle neuen Residuen mit höchstens n Multiplikationen und Additionen berechnen kann, man also in jedem Teilschritt auch die Norm von F leicht berechnen kann.

Im Zweidimensionalen erlauben die Verfahren eine einfache graphische Deutung: die neue i -te Komponente ist so gewählt, daß die i -te Gleichung bei sonst unveränderten übrigen Komponenten exakt erfüllt ist im Falle $\omega = 1$. Für anderes ω muss man die dazu notwendige Änderung mit ω multiplizieren. Beim Gauß–Seidel–Verfahren bzw. SOR–Verfahren hat man schon den nächsten “Zwischenpunkt” und nach Durchlauf der n Gleichungen den nächsten Punkt. Beim Jacobi–Verfahren werden erst alle Korrekturen einzeln gebildet und dann gleichzeitig auf den alten Punkt angewendet.

Beispiel 6.1.1 Gegeben sei das lineare Gleichungssystem $Ax = b$ mit

$$A = \begin{pmatrix} 5 & -4 \\ 1 & -2 \end{pmatrix} \quad \text{und} \quad b = \begin{pmatrix} 9 \\ -1 \end{pmatrix}.$$

Mit dem Startvektor $x^{(0)} = (-6, -6)^T$ führen wir jeweils drei Schritte des Jacobi- und des Gauß-Seidel-Verfahrens aus. Die Zerlegung der Matrix A in $D - L - U$ ergibt für das **Jacobi-Verfahren** die Iterationsvorschrift

$$\begin{aligned} x^{(k+1)} &= D^{-1}(L + U)x^{(k)} + D^{-1}b \\ &= \begin{pmatrix} \frac{1}{5} & 0 \\ 0 & -\frac{1}{2} \end{pmatrix} \left(\begin{pmatrix} 0 & 0 \\ -1 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 4 \\ 0 & 0 \end{pmatrix} \right) x^{(k)} + \begin{pmatrix} \frac{1}{5} & 0 \\ 0 & -\frac{1}{2} \end{pmatrix} \begin{pmatrix} 9 \\ -1 \end{pmatrix} \\ &= \begin{pmatrix} 0 & \frac{4}{5} \\ \frac{1}{2} & 0 \end{pmatrix} x^{(k)} + \begin{pmatrix} \frac{9}{5} \\ \frac{1}{2} \end{pmatrix} \end{aligned}$$

Damit ergibt sich die Iterationsfolge

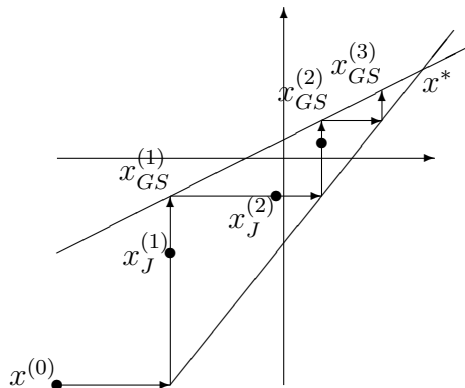
$$\begin{pmatrix} -6 \\ -6 \end{pmatrix}, \quad \begin{pmatrix} -3 \\ -2.5 \end{pmatrix}, \quad \begin{pmatrix} -\frac{1}{5} \\ -1 \end{pmatrix}, \quad \begin{pmatrix} \frac{1}{5} \\ \frac{2}{5} \end{pmatrix}, \quad \dots$$

Das **Gauß-Seidel-Verfahren** ist gegeben durch

$$\begin{aligned} x^{(k+1)} &= D^{-1}(Lx^{(k+1)} + Ux^{(k)} + b) \\ x_1^{(k+1)} &= \frac{1}{5}(9 + 4x_2^{(k)}) \\ x_2^{(k+1)} &= \frac{1}{2}(1 + x_1^{(k+1)}) \end{aligned}$$

Damit ergibt sich die Iterationsfolge

$$\begin{pmatrix} -6 \\ -6 \end{pmatrix}, \quad \begin{pmatrix} -3 \\ -1 \end{pmatrix}, \quad \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad \begin{pmatrix} \frac{13}{5} \\ \frac{9}{5} \end{pmatrix}, \quad \dots$$



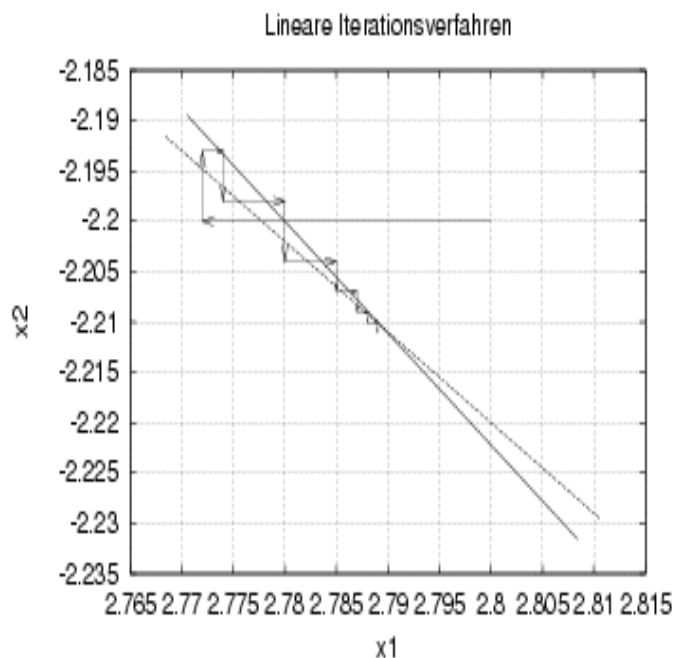
Beispiel 6.1.2 Hier folgt die Darstellung der Iteration für das SOR-Verfahren mit

$$A = \begin{pmatrix} 1.0 & 0.9 \\ 0.9 & 1 \end{pmatrix} \quad b = \begin{pmatrix} 0.8 \\ 0.3 \end{pmatrix} \quad \omega = 1.39286$$

Die Iterationsfolge ist

6.1. ALLGEMEINE ANSÄTZE ZUR ENTWICKLUNG VON ITERATIONSVERFAHRENSPLITTINGV

k	x(1)	x(2)	r
0	2.800000	-2.200000	0.2828427E-01
1	2.772143	-2.200000	0.9351765E-02
2	2.772143	-2.192936	0.2493744E-02
3	2.774232	-2.192936	0.3916963E-02
4	2.774232	-2.198330	0.4528347E-02
5	2.780172	-2.198330	0.4176263E-02
6	2.780172	-2.203658	0.3462886E-02
7	2.784518	-2.203658	0.2702011E-02
8	2.784518	-2.207012	0.2027286E-02
9	2.787015	-2.207012	0.1480038E-02
10	2.787015	-2.208825	0.1058975E-02
11	2.788307	-2.208825	0.7460880E-03
12	2.788307	-2.209732	0.5192620E-03
13	2.788936	-2.209732	0.3578318E-03
14	2.788936	-2.210165	0.2445744E-03
15	2.789232	-2.210165	0.1660158E-03
16	2.789232	-2.210365	0.1120303E-03
17	2.789366	-2.210365	0.7521769E-04
18	2.789366	-2.210455	0.5027890E-04
19	2.789427	-2.210455	0.3347840E-04
20	2.789427	-2.210495	0.2221513E-04
21	2.789453	-2.210495	0.1469601E-04
22	2.789453	-2.210513	0.9695118E-05
23	2.789465	-2.210513	0.6380080E-05
24	2.789465	-2.210521	0.4189082E-05
25	2.789470	-2.210521	0.2744849E-05
26	2.789470	-2.210524	0.1795149E-05
27	2.789472	-2.210524	0.1172012E-05
28	2.789472	-2.210525	0.7639617E-06



Dies sind also 14 Schritte, da auch die Zwischenwerte tabelliert sind. Zunächst scheint die Genauigkeit sich sogar (in der Maximumnorm) zu verschlechtern. Das Gauß-Seidel-Verfahren benötigt für die gleiche Genauigkeit bereits 37 Schritte. Für gewisse Matrizen kann das *SOR*-Verfahren die Konvergenz ganz erheblich beschleunigen, wenn ω optimal gewählt ist.

Bemerkung 6.1.1 Man kann versuchen, durch Verallgemeinerung der Verfahrensstruktur zu besseren (schnelleren) Verfahren zu gelangen. Mögliche Ansätze sind:

a.) Die Hintereinanderschaltung verschiedener Iterationsverfahren:

$$\begin{aligned} \text{z.B. } x_{k+1/2} &= \Phi_1(x_k) && \text{“Zwischenschritt”} \\ x_{k+1} &= \Phi_2(x_{k+1/2}) = \Phi_2(\Phi_1(x_k)) \end{aligned}$$

b.) Allgemeinere Zerlegung:

$$\begin{aligned} A &= M + N_1 + N_2, & C &= C_1 + C_2 \\ (\omega M + C)x_{k+1} &= (C_1 - \omega N_1)x_k + (C_2 - \omega N_2)x_{k-1} + \omega b \end{aligned}$$

Ein interessantes Beispiel für a.) ist das *SSOR*-Verfahren. Hier ist

$$\begin{aligned} \Phi_1(x) &= (-\omega L + D)^{-1}\{((1 - \omega)D + \omega U)x + \omega b\} && \text{(d.h. } \textit{SOR}) \\ \Phi_2(x) &= (D - \omega U)^{-1}\{((1 - \omega)D + \omega L)x + \omega b\} && \text{d.h. } M = D - U \\ &&& N = -L \\ &&& C = (1 - \omega)D \\ &&& \text{“SOR rückwärts”} \end{aligned}$$

d.h.

$$x_{k+1} = (D - \omega U)^{-1}((1 - \omega)D + \omega L)(-\omega L + D)^{-1}((1 - \omega)D + \omega U)x_k + \\ + \omega(D - \omega U)^{-1}(I + ((1 - \omega)D + \omega L)(-\omega L + D)^{-1})b$$

Ist A symmetrisch und positiv definit, dann $U = L^T$ und D positiv definit, so daß

$$\begin{aligned} & (D - \omega U)^{-1}((1 - \omega)D + \omega L)(-\omega L + D)^{-1}((1 - \omega)D + \omega U) = \\ & = D^{-1/2}(I - \omega \hat{L}^T)^{-1}((1 - \omega)I + \omega \hat{L})(-\omega \hat{L} + I)^{-1}((1 - \omega)I + \omega \hat{L}^T)D^{1/2} \\ & = \underbrace{((I - \omega \hat{L}^T)D^{1/2})^{-1}}_R ((2 - \omega)I - R^T)(R^T)^{-1}((2 - \omega)I - R)D^{1/2} \\ & = (RD^{1/2})^{-1}((2 - \omega)R^{-T} - I)((2 - \omega)R^{-1} - I)(RD^{1/2}) \\ & \quad (\text{mit } \hat{L} \stackrel{\text{def}}{=} D^{-1/2}LD^{-1/2}) \end{aligned}$$

d.h. die Iterationsmatrix ist in diesem Fall ähnlich zu einer symmetrischen, positiv semidefiniten Matrix, hat also nur reelle nichtnegative Eigenwerte, was die Möglichkeit zu einer weiteren Konvergenzbeschleunigung bietet. Eine solche Möglichkeit ist realisiert im sogenannten SSOR-SI Verfahren von Young. Siehe dazu den Text von Hageman und Young. Für sich alleine ist das SSOR-Verfahren gewöhnlich langsamer als das SOR-Verfahren ! \square

6.2 Konvergenzsätze für spezielle Matrizen

Im Folgenden geht es darum, einfach nachprüfbar hinreichende Bedingungen für die Konvergenz der einzelnen Verfahren zu entwickeln. Aufgrund der Vorüberlegungen in Satz 6.1.1 ist folgendes Ergebnis offensichtlich:

Satz 6.2.1 Das Gesamtschrittverfahren

$$x_{k+1} = D^{-1}(L + U)x_k + D^{-1}b$$

ist konvergent, falls ($A = D - L - U$)

$$|\alpha_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^n |\alpha_{ij}| \quad i = 1, \dots, n \quad \text{“starkes Zeilensummenkriterium”}^1$$

oder

$$|\alpha_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^n |\alpha_{ji}| \quad i = 1, \dots, n \quad \text{“starkes Spaltensummenkriterium”}$$

(Man nennt ein solches A streng diagonaldominant nach Zeilen bzw. Spalten.)

Beweis: Benutze $\|\cdot\|_\infty$ bzw. $\|\cdot\|_1$. Details als Übungsaufgabe \square

Für die Gleichungssysteme, die bei der Methode der finiten Differenzen entstehen, ist das starke Zeilensummenkriterium in der Regel nicht erfüllt, denn die diskretisierten Ableitungen bilden die Konstante noch korrekt auf null ab, d.h. das alle Zeilen, in denen keine randnahen Punkte auftreten, den Vektor $(1, \dots, 1)^T$ als Nullvektor besitzen. Hier hilft eine Verschärfung von Satz 6.2.1 weiter, für deren Formulierung wir erst noch einen neuen Begriff einführen müssen:

Definition 6.2.1 $A \in \mathbb{C}^{n \times n}$ heißt unzerlegbar (irreduzibel), falls **keine** Permutationsmatrix P existiert mit

$$P^T A P = \left(\begin{array}{c|c} \tilde{A}_{11} & \tilde{A}_{12} \\ \hline 0 & \tilde{A}_{22} \end{array} \right) \quad \tilde{A}_{ii} \text{ quadratisch.}$$

□

Definition 6.2.2 Unter dem einer Matrix $A \in \mathbb{C}^{n \times n}$ zugeordneten gerichteten Graphen $\mathcal{G}(A)$ versteht man folgende Konstruktion:

- $\mathcal{G}(A)$ besteht aus n "Knoten" P_1, \dots, P_n
- Eine gerichtete Kante $P_i \mapsto P_j$ verbindet P_i und P_j genau dann, wenn $\alpha_{ij} \neq 0$. (für $\alpha_{ii} \neq 0$ also $P_i \mapsto P_i$ mit sich selbst verbunden)
- Gerichtete Wege in $\mathcal{G}(A)$ sind aus gerichteten Kanten zusammengesetzt.
- Der gerichtete Graph $\mathcal{G}(A)$ heißt **zusammenhängend**, falls für jedes Knotenpaar (P_i, P_j) , $1 \leq i, j \leq n$, $i \neq j$, ein gerichteter Weg von P_i nach P_j besteht.

□

Dazu gilt folgender

Satz 6.2.2 Eine $n \times n$ Matrix A ist **irreduzibel** genau dann, wenn der zugeordnete gerichtete Graph $\mathcal{G}(A)$ zusammenhängend ist.

<<

Beweis: Sei $\mathcal{G}(A)$ zusammenhängend und $1 \leq i, j \leq n$, $i \neq j$ beliebig.

Nach Def. 6.2.2 existieren i_1, \dots, i_m mit

$$\alpha_{i, i_1} \times \alpha_{i_1, i_2} \times \dots \times \alpha_{i_m, j} \neq 0 \quad (6.3)$$

Angenommen, A sei reduzibel, d.h. $\alpha_{s,k} = 0$ für $s \in S$, $k \in K$, wobei $S \cap K = \emptyset$, $S \cup K = \{1, \dots, n\}$. Wähle $i \in S$ und $j \in K$. Mit $\alpha_{i, i_1} \neq 0$ folgt $i_1 \notin K$, d.h. $i_1 \in S$, d.h. $i_2 \in S$ usw., so daß die Konstruktion einer Kette (6.3) unmöglich wäre. Nun beweisen wir die Umkehrung. Sei A irreduzibel. Wähle $i \in \{1, \dots, n\}$ bel. und setze

$$\mathcal{I}_i \stackrel{\text{def}}{=} \{k : \exists \{i_1, \dots, i_m\} : \alpha_{i, i_1} \times \dots \times \alpha_{i_m, k} \neq 0\}$$

$\mathcal{I}_i \neq \emptyset$. Andernfalls $\alpha_{i1}, \dots, \alpha_{in} = 0$ (Widerspruch!)

Angenommen $\mathcal{I}_i \neq \{1, \dots, n\}$ und $l \in \{1, \dots, n\} \setminus \mathcal{I}_i$ bel. (Es entspricht also \mathcal{I}_i der Menge S und $\{1, \dots, n\} \setminus \mathcal{I}_i$ entspricht K).

Beh.: $\alpha_{j,l} = 0$ für $j \in \mathcal{I}_i$.

(Dies wäre ein Widerspruch zur angenommenen Irreduzibilität von A).

Wäre aber $\alpha_{j_0,l} \neq 0$ für ein $j_0 \in \mathcal{I}_i$, dann existiert $\{i_1, \dots, i_m\}$ mit

$$\alpha_{i_1,i_1} \times \alpha_{i_1,i_2} \times \dots \times \alpha_{i_m,j_0} \neq 0, \quad \text{und} \quad \alpha_{j_0,l} \neq 0$$

d.h. $l \in \mathcal{I}_i$ (Widerspruch!) D.h. $\mathcal{I}_i = \{1, \dots, n\}$, und da i beliebig war, ist $\mathcal{G}(A)$ zusammenhängend. \square

>>

Definition 6.2.3 $A \in \mathbb{C}^{n \times n}$ heißt **irreduzibel diagonaldominant**, falls A irreduzibel ist und

$$\forall i : |\alpha_{ii}| \geq \sum_{\substack{j=1 \\ j \neq i}}^n |\alpha_{ij}|, \quad \exists i_0 : |\alpha_{i_0 i_0}| > \sum_{\substack{j=1 \\ j \neq i_0}}^n |\alpha_{i_0 j}|$$

\square

Satz 6.2.3 Sei A streng- oder irreduzibel diagonaldominant. Dann ist A regulär und das Gesamtschrittverfahren konvergiert.

<<

Beweis: Der Beweis verläuft in folgenden Schritten:

α) Wir zeigen, daß das Gesamtschrittverfahren wohldefiniert ist

β) Wir zeigen, daß $\rho(D^{-1}(L+U)) < 1$

γ) Aus β) folgt, daß das Gesamtschrittverfahren bei **beliebigem** b gegen eine Lösung von $Ax = b$ konvergiert. Hieraus folgt die Regularität von A .

Zu α) Im Falle der strengen Diagonaldominanz ist die Behauptung klar. Sei A irreduzibel diagonaldominant, dann $\alpha_{ii} \neq 0 \quad \forall i$
(sonst $\alpha_{i_0 1} = \dots = \alpha_{i_0 n} = 0$ Widerspruch zu irreduzibel)

Zu β) Im Fall der strengen Diagonaldominanz folgt die Behauptung aus Satz 6.2.1

Im 2. Fall ist nach Vor. mit

$$B \stackrel{\text{def}}{=} D^{-1}(L+U) = (\beta_{ij})$$

$$\rho(B) \leq \|B\|_\infty \leq 1$$

Annahme: $\rho(B) = 1$, d.h. $\exists \lambda \in \mathbb{C}$

$$|\lambda| = 1 \quad (B - \lambda I)x = 0$$

d.h.

$$\sum_{j=1}^n \beta_{i,j} \xi_j = \xi_i \lambda$$

mit $x \neq 0$. $B - \lambda I$ ist irreduzibel diagonaldominant, da sie sich von $D^{-1}A$ nur in der Diagonale unterscheidet, wo 1 durch $-\lambda$ mit $|-\lambda| = 1$ ersetzt ist.

Annahme: mit $x = (\xi_1, \dots, \xi_n)^T$ sei $|\xi_1| = \dots = |\xi_n| = \xi \neq 0$. Dann

$$\left| \sum_{j=1}^n \beta_{ij} \xi_j \right| = \xi \leq \xi \sum_{j=1}^n |\beta_{ij}| \quad \forall i$$

im Widerspruch zur Annahme über A . (Man dividiere durch ξ). Daher ist

$$\mathcal{I} \stackrel{\text{def}}{=} \{j : |\xi_j| \geq |\xi_i| \quad \forall i; \quad |\xi_j| > |\xi_{i_0}| \text{ für ein } i_0 \in \{1, \dots, n\}\} \neq \emptyset.$$

Sei $j \in \mathcal{I}$. Dann gilt (beachte $\beta_{ii} = 0$)

$$\begin{aligned} 0 &= \left| \sum_{i=1}^n \beta_{ji} \xi_i - \lambda \xi_j \right| \Rightarrow 1 \leq \sum_{i=1}^n |\beta_{ji}| \underbrace{\frac{|\xi_i|}{|\xi_j|}}_{\otimes} \Rightarrow \\ &\quad \otimes = \leq 1 \text{ und } = 1 \text{ für } i \in \mathcal{I} \\ 1 &\leq \sum_{i \in \mathcal{I}} |\beta_{ji}| + \sum_{i \notin \mathcal{I}} |\beta_{ji}| \underbrace{\frac{|\xi_i|}{|\xi_j|}}_{< 1} < \sum_{i=1}^n |\beta_{ji}| \leq 1 \quad : \text{Widerspruch!} \end{aligned}$$

$$\text{falls } \sum_{i \notin \mathcal{I}} |\beta_{ji}| \neq 0. \Rightarrow j \in \mathcal{I} : \sum_{i \notin \mathcal{I}} |\beta_{ji}| = 0 \Rightarrow$$

$$(\beta_{ji} = \alpha_{ji}/\alpha_{jj}) \sum_{i \notin \mathcal{I}} |\alpha_{ji}| = 0 \text{ für } j \in \mathcal{I} \Rightarrow A \text{ reduzibel: Widerspruch! (Die entsprechende}$$

Permutationsmatrix P bildet die Spalten mit $i \in \mathcal{I}$ auf die $|\mathcal{I}|$ ersten Spalten und die übrigen Spalten auf die letzten $n - |\mathcal{I}|$ ab.) \square

>>

Man könnte meinen, daß aufgrund der Konstruktion des Verfahrens das Einzelschrittverfahren stets "besser konvergiert" als das Gesamtschrittverfahren. Dies gilt jedoch nicht allgemein. Es gibt Gegenbeispiele.

Beispiel 6.2.1 Sei

$$A = \begin{pmatrix} 2 & 1 & 4 \\ 1 & 2 & -4 \\ 4 & 4 & 2 \end{pmatrix}$$

Hier konvergiert das Gesamtschrittverfahren, nicht aber das Einzelschrittverfahren. Und mit

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

ist es umgekehrt.

Folgende Resultate sind bekannt:

Satz 6.2.4 Ist $A \in \mathbb{R}^{n \times n}$, $\alpha_{ij} \leq 0$ für $i \neq j$ und $\alpha_{ii} \neq 0$, $i = 1, \dots, n$

$$J \stackrel{\text{def}}{=} D^{-1}(L + U), \quad H \stackrel{\text{def}}{=} (-L + D)^{-1}U,$$

dann gilt genau eine der folgenden Beziehungen:

1. $\rho(H) = \rho(J) = 0$
2. $0 < \rho(H) < \rho(J) < 1$
3. $\rho(H) = \rho(J) = 1$
4. $\rho(H) > \rho(J) > 1$

Beweis: siehe Varga: Matrix iterative analysis: Stein–Rosenberg–Theorem. \square

Der Satz besagt, daß für Matrizen der angegebenen Art das Einzelschrittverfahren besser konvergiert als das Gesamtschrittverfahren, wenn überhaupt eines der Verfahren konvergiert. Insbesondere für die Matrix, die bei der Diskretisierung der Poissongleichung mit Dirichletranddaten mit finiten Differenzen auftritt, ist also wegen Satz 6.2.4 und Satz 6.2.3 sowohl das Gesamt– wie das Einzelschrittverfahren konvergent, und zwar das Einzelschrittverfahren schneller als das Gesamtschrittverfahren. Die Irreduzibilität dieser Matrix ist allerdings etwas mühsam nachzuweisen.

In der Praxis spielen vor allem folgende beiden Matrizentypen eine Rolle:

1. symmetrische, positiv definite Matrizen
2. M-Matrizen

Definition 6.2.4 $A \in \mathbb{R}^{n \times n}$ heißt M-Matrix, falls

1. $\alpha_{ij} \leq 0$ für $i \neq j$
2. A^{-1} existiert und $A^{-1} \geq 0$ komponentenweise.

A heißt L-Matrix, wenn

1. $\alpha_{ii} > 0$, $i = 1, \dots, n$
2. $\alpha_{ij} \leq 0$ für $i \neq j$.

\square

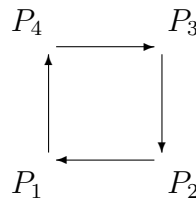
Satz 6.2.4 besagt u.a., daß für eine L-Matrix das Einzelschrittverfahren, wenn es konvergiert, stets schneller konvergiert als das Gesamtschrittverfahren, das dann aber auch konvergiert.

Beispiel 6.2.2 Im folgenden Beispiel werden an Hand von drei Matrizen die hier definierten speziellen Matrixeigenschaften noch einmal diskutiert:

1.

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

- Irreduzibilität: Der gerichtete Graph, der der Matrix A zugeordnet werden kann hat die Gestalt



Damit ist die Matrix irreduzibel, da es für je zwei beliebige Punkte P_1 und P_2 immer einen Weg von P_1 nach P_2 gibt.

- Die Matrix A ist **nicht** diagonaldominant, und somit auch **nicht** strikt oder irreduzibel diagonaldominant, da die Diagonalelemente der Matrix kleiner sind als die Summe der Einträge in der entsprechenden Zeile:

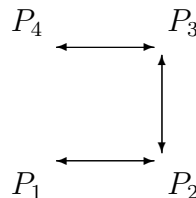
$$0 < 1$$

- A ist **weder** eine L - **noch** eine M -Matrix.

2.

$$B = \begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{pmatrix}$$

- Irreduzibilität: Der gerichtete Graph, der der Matrix B zugeordnet werden kann hat die Gestalt



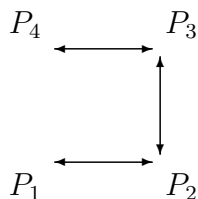
Damit ist die Matrix B irreduzibel.

- B ist diagonaldominant, wegen $2 \geq 1 + 1$ und $2 > 1$,
- und sie ist irreduzibel diagonaldominant.
- Aber die Matrix B ist **nicht** strikt diagonaldominant, da in der 2. Zeile nicht die echte Ungleichung gilt.
- B ist eine L -Matrix
- Folglich ist B eine M -Matrix.

3.

$$C = \begin{pmatrix} 2 & -2 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -2 & 2 \end{pmatrix}$$

- Irreduzibilität: Der gerichtete Graph, der der Matrix C zugeordnet werden kann hat die Gestalt



Damit ist die Matrix C irreduzibel.

- C ist diagonaldominant, wegen $2 \geq 1 + 1$ und $2 \geq 2$,
- aber sie ist **nicht** irreduzibel diagonaldominant, weil nie die Ungleichheit gilt.
- Aus dem gleichen Grund ist die Matrix C **nicht** strikt diagonaldominant.
- C ist eine L-Matrix
- C ist **keine** M-Matrix, da C singulär ist.

Der folgende Satz liefert eine Charakterisierung der M-Matrizen:

Satz 6.2.5 A ist M-Matrix $\Leftrightarrow A$ ist L-Matrix und $\rho(D^{-1}(L+U)) < 1$, wobei $A = -L + D - U$ die übliche Zerlegung von A ist.

<<

Beweis: Sei $\alpha_{ii} > 0$ für $i = 1, \dots, n$ $\alpha_{ij} \leq 0$ für $i \neq j$ und $\rho(D^{-1}(L+U)) < 1$.

Setze $J \stackrel{\text{def}}{=} D^{-1}(L+U)$. Dann ist auch $\rho(-J) < 1$ und daher konvergiert die Neumannsche Reihe (siehe Einf. Num. Math. I)

$$(I - J)^{-1} = \sum_{k=0}^{\infty} J^k \geq 0 \quad \text{da } J \geq 0$$

Nun ist aber $I - J = D^{-1}A$, d.h.

$$(I - J)^{-1} = A^{-1}D \geq 0 \Rightarrow \exists A^{-1} \geq 0$$

Sei umgekehrt A eine M-Matrix. Annahme: $\alpha_{ii} \leq 0$ für ein i . Dann wegen der Voraussetzung ($\alpha_{ij} \leq 0$ für $i \neq j$)

$$Ae_i \leq 0 \Rightarrow \underbrace{A^{-1}Ae_i}_{\geq 0} \leq 0 \Rightarrow e_i \leq 0 \quad \text{Widerspruch!}$$

Also gilt $\alpha_{ii} > 0$ für $i = 1, \dots, n$, d.h.

$$J \stackrel{\text{def}}{=} D^{-1}(L+U)$$

ist wohldefiniert und $J \geq 0$. Ferner existiert $A^{-1}D = (I - J)^{-1}$.
Sei λ ein Eigenwert von J mit Eigenvektor $x \neq 0$. Dann

$$\begin{aligned} |\lambda||x| &\leq J|x| \\ (I - J)|x| &\leq (1 - |\lambda|)|x| \end{aligned}$$

und weil $(I - J)^{-1} \geq 0$

$$|x| \leq \underbrace{(1 - |\lambda|)(I - J)^{-1}}_{y \neq 0, y \geq 0} \underbrace{|x|}_{\neq 0} \Rightarrow |\lambda| < 1, \text{ d.h. } \rho(J) < 1$$

□

>>

Das Gesamtschrittverfahren ist somit für jede M-Matrix konvergent und wegen Satz 6.2.4 somit auch das Einzelschrittverfahren.

Definition 6.2.5 Sei $A \in \mathbb{R}^{n \times n}$. $A = N - P$ heißt reguläre Aufspaltung von A , wenn $N \in \mathbb{R}^{n \times n}$ regulär ist und $N^{-1} \geq 0$, $P \geq 0$. □

Satz 6.2.6 Sei $A \in \mathbb{R}^{n \times n}$ und $A = N - P$ eine reguläre Aufspaltung von A . Dann gilt

$$\rho(N^{-1}P) < 1 \Leftrightarrow A^{-1} \geq 0.$$

<<

Beweis: “ \Rightarrow ” Trivial gilt $H \stackrel{\text{def}}{=} N^{-1}P \geq 0$. Wenn $\rho(H) < 1$, dann auch $\rho(-H) < 1$, d.h. $(I - H)^{-1}$ existiert und $(I - H)^{-1} \geq 0$ und daher $A^{-1} = (I - H)^{-1}N^{-1} \geq 0$.

“ \Leftarrow ” Sei $A^{-1} \geq 0$. Es ist $A^{-1} = (N - P)^{-1} = (I - N^{-1}P)^{-1}N^{-1}$.

Nun ist mit $H \stackrel{\text{def}}{=} N^{-1}P \ (\geq 0)$

$$\begin{aligned} 0 &\leq (I + H + \dots + H^m)N^{-1} = (I - H^{m+1})(I - H)^{-1}N^{-1} \\ &= (I - H^{m+1})A^{-1} \leq A^{-1} \end{aligned}$$

d.h. $\sum_{k=0}^m H^k$ ist konvergent und somit $\rho(H) < 1$ □

>>

Folgerung:

Satz 6.2.7 Ist A streng oder irreduzibel diagonaldominant und L-Matrix, dann ist A M-Matrix.

Beweis: $N \stackrel{\text{def}}{=} D$, $P = L + U$ ist reguläre Aufspaltung und nach Satz 6.2.3 $\rho(D^{-1}(L + U)) < 1$. Satz 6.2.6 liefert die Behauptung. \square

Wir wissen bereits, daß das Einzelschrittverfahren für M-Matrizen konvergiert. Es gilt sogar stärker

Satz 6.2.8 *Ist A eine M-Matrix, dann konvergiert das SOR-Verfahren im Bereich $0 < \omega \leq 1$.*

<<

Beweis: Wir zeigen, daß

$$\begin{aligned} A &= \frac{1}{\omega}(D - \omega L - (1 - \omega)D - \omega U) \\ &= \underbrace{\frac{1}{\omega}(D - \omega L)}_{=:N} - \underbrace{\frac{1}{\omega}((1 - \omega)D + \omega U)}_{=:P} \end{aligned}$$

eine reguläre Aufspaltung von A ist für $0 < \omega \leq 1$. Dazu ist nur noch zu zeigen $(D - \omega L)^{-1} \geq 0$. (Dann folgt mit Satz 6.2.6 die Behauptung.)

Nun ist aber

$$\begin{aligned} (D - \omega L)^{-1} &= (I - \underbrace{\omega D^{-1}L}_{\geq 0})^{-1}D^{-1} \\ &= (I + \omega D^{-1}L + \dots + \omega^{n-1}(D^{-1}L)^{n-1})D^{-1} \geq 0 \end{aligned}$$

\square

>>

Für die Praxis ist allerdings das Resultat von Satz 6.2.8 nicht sehr interessant, da man zeigen kann, daß für eine irreduzible M-Matrix $\rho(B(\omega)) = \rho((D - \omega L)^{-1}((1 - \omega)D + \omega U))$ in einem Bereich $0 \leq \omega \leq \bar{\omega}$ mit $\bar{\omega} > 1$ streng monoton fallend ist (vgl. Varga: Matrix iterative analysis). Interessant ist die Frage nach der Konvergenz des SOR-Verfahrens im Bereich $\omega > 1$, insbesondere die Frage nach der Existenz eines optimalen Relaxationsparameters ω_{opt} .

Im Folgenden sei stets mit $A = -L + D - U$ und

$$B(\omega) = (D - \omega L)^{-1}((1 - \omega)D + \omega U).$$

Zunächst gilt

Satz 6.2.9 *Es gilt stets $\rho(B(\omega)) \geq |\omega - 1|$*

Beweis:

$$\det(B(\omega) - \lambda I) = \det(\underbrace{(D - \omega L)^{-1}}_{(I - \omega D^{-1}L)^{-1}D^{-1}} ((1 - \omega)D + \omega U - \lambda(D - \omega L)))$$

man beachte, daß $(I - \omega D^{-1}L)$ eine untere Dreiecksmatrix mit Diagonale $(1, \dots, 1)$ ist und wende den Produktsatz für Determinanten an

$$= \det((1 - \omega - \lambda)I + \omega D^{-1}U + \lambda \omega D^{-1}L)$$

unter Setzung von $\lambda = 0 \Rightarrow$

$$\det(B(\omega)) = \prod_{i=1}^n \lambda_i(B(\omega)) = (1 - \omega)^n$$

\Rightarrow

$$\rho(B(\omega)) = \max_i |\lambda_i(B(\omega))| \geq |\omega - 1|$$

□

Somit ist (für reelles ω) nur das Intervall $0 < \omega < 2$ von Interesse. Wir wissen bereits, daß für M-Matrizen $\rho(B(\omega))$ in einem Intervall $0 < \omega \leq \bar{\omega}$ mit $\bar{\omega} \geq 1$ monoton fallend ist. Für positiv definite Matrizen erhält man die Aussage, daß man ausser $\omega \in]0, 2[$ keine weitere Einschränkung an die Wahl von ω hat, um Konvergenz des SOR-Verfahrens zu erhalten, allerdings hat man keine Aussage über die Wahl eines "guten" ω -Wertes.

Satz 6.2.10 *Ist $A \in \mathbb{R}^{n \times n}$ symmetrisch und positiv definit, dann gilt $\rho(B(\omega)) < 1$ für $0 < \omega < 2$.*

Beweis: Setze

$$\begin{aligned} f(x) &\stackrel{def}{=} \frac{1}{2}x^T A x - b^T x \\ &= -\frac{1}{2}b^T A^{-1}b + \frac{1}{2}(x - A^{-1}b)^T A(x - A^{-1}b) \geq -\frac{1}{2}b^T A^{-1}b. \end{aligned}$$

Es ist also $\nabla f(x) = Ax - b$ und $\nabla^2 f(x) = A$. Also ist $x^* = A^{-1}b$ die eindeutige Minimalstelle von f . Ferner definiere mit den Bezeichnungen von (6.2)

$$y_0 = x_0, \quad y_{kn+i} \stackrel{def}{=} \begin{pmatrix} \xi_{1,k+1} \\ \vdots \\ \xi_{i,k+1} \\ \xi_{i+1,k} \\ \vdots \\ \xi_{n,k} \end{pmatrix} \quad \begin{array}{l} 1 \leq i \leq n \\ k = 0, 1, 2, \dots \end{array}$$

$$r_j \stackrel{def}{=} Ay_j - b \quad j \in \mathbb{N}_0$$

$$\rho_{j,i} \stackrel{def}{=} e_i^T r_j$$

Dann gilt

$$y_{kn+i} - y_{kn+i-1} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \xi_{i,k+1} - \xi_{i,k} \\ 0 \\ \vdots \\ 0 \end{pmatrix} = -\frac{\omega}{\alpha_{ii}} \underbrace{(e_i^T r_{kn+i-1})}_{\rho_{kn+i-1,i}} e_i$$

und $x_k = y_{kn}$ für $k \in \mathbb{N}_0$

Nun wird

$$f(y_{kn+i}) - f(y_{kn+i-1}) = -\frac{1}{2\alpha_{ii}} \omega(2-\omega) \rho_{kn+i-1,i}^2$$

wie man durch Ausrechnen unmittelbar verifiziert. D.h. für $0 < \omega < 2$ ist die Folge $f(y_j)$ monoton nicht wachsend, nach unten beschränkt, folglich konvergent, d.h.

$$\lim_{k \rightarrow \infty} \rho_{kn+i-1,i} = 0 \quad \text{für } i = 1, \dots, n \quad (6.4)$$

und damit auch

$$\lim_{k \rightarrow \infty} y_{kn+i} - y_{kn+i-1} = 0$$

und wegen

$$r_{j+1} - r_j = A(y_{j+1} - y_j)$$

auch mit $j = kn + i - 1$

$$\lim_{k \rightarrow \infty} (r_{kn+i} - r_{kn+i-1}) = 0 \quad i = 1, \dots, n \quad (6.5)$$

Zu zeigen bleibt: $\lim_{k \rightarrow \infty} r_{kn} = 0$

Nun gilt für $k \geq k_0(\varepsilon)$ wegen (6.4) und (6.5)

$$\begin{aligned} |\rho_{j+1,i} - \rho_{j,i}| &\leq \varepsilon && \text{für } i = 1, \dots, n && \text{und } j \geq j_0 = nk_0 \\ |\rho_{kn+i-1,i}| &\leq \varepsilon && \text{für } k \geq k_0(\varepsilon) && \text{und } i = 1, \dots, n \end{aligned}$$

Also $|\rho_{kn,1}| \leq \varepsilon$

$$\begin{aligned} |\rho_{kn+1,2} - \rho_{kn,2}| &\leq \varepsilon && \text{und } |\rho_{kn+1,2}| &\leq \varepsilon &\Rightarrow & |\rho_{kn,2}| &\leq 2\varepsilon \\ |\rho_{kn+2,3} - \rho_{kn+1,3}| &\leq \varepsilon && \text{und } |\rho_{kn+2,3}| &\leq \varepsilon &\Rightarrow & |\rho_{kn+1,3}| &\leq 2\varepsilon \\ |\rho_{kn+1,3} - \rho_{kn,3}| &\leq \varepsilon && \text{und } |\rho_{kn+1,3}| &\leq 2\varepsilon &\Rightarrow & |\rho_{kn,3}| &\leq 3\varepsilon \end{aligned}$$

und schließlich

$$|\rho_{kn,n}| \leq n\varepsilon$$

d.h. $\|r_{kn}\|_\infty \leq n\varepsilon$ d.h. $r_{kn} = Ax_k - b \xrightarrow[k \rightarrow \infty]{} 0$, d.h. $x_k \rightarrow x^* = A^{-1}b$

bei bel. x_0 . Satz 6.1.1 \Rightarrow Beh. □

Bemerkung 6.2.1 *Ein ganz anders gearteter Beweis kann bei Stoer & Bulirsch, Bd. 2 nachgelesen werden.* □

Satz 6.2.10 gibt eine interessante anschauliche Deutung des Relaxationsverfahrens als Minimierungsverfahren für die dort konstruierte Funktion f , deren Gradient das “Residuum” $Ax - b$ ist. Die Flächen $f(x) = c$ im \mathbb{R}^n sind konzentrische Ellipsoide und beim Relaxationsverfahren wird der gemeinsame Mittelpunkt x^* (die eindeutige Minimalstelle von f auf \mathbb{R}^n) längs der zyklisch durchlaufenen Koordinatenrichtung mit monotonem Abstieg für f erreicht.

Aus diesem Beweis folgt auch offensichtlich, daß die Konvergenz des SOR-Verfahrens erhalten bleibt, wenn man in (6.2) ω durch ω_{ik} mit

$$0 < \min_i \liminf_k \omega_{i,k} \leq \max_i \limsup_k \omega_{i,k} < 2$$

ersetzt. (Satz 6.2.1 ist allerdings nicht anwendbar und die Iterationsmatrizen können dann nicht mehr in der bisher betrachteten Form geschrieben werden.) Für eine spezielle Matrizenklasse tritt an die Stelle der bisher erhaltenen qualitativen Aussagen ein quantitatives Resultat über das Verhalten von $\rho(B(\omega))$ als Funktion von ω :

Definition 6.2.6 Sei $A = D(I - \hat{L} - \hat{U})$ mit $\hat{L} \stackrel{def}{=} D^{-1}L$, $\hat{U} \stackrel{def}{=} D^{-1}U$ die übliche Aufspaltung von A in Diagonaleil, striktes unteres und striktes oberes Dreieck. A heißt **konsistent geordnet**, falls mit $\alpha, \beta \neq 0$ gilt:
 λ Eigenwert von $\alpha \hat{L} + \frac{1}{\alpha} \hat{U} \iff \lambda$ Eigenwert $\beta \hat{L} + \frac{1}{\beta} \hat{U}$, d.h. diese Eigenwerte sind von α unabhängig. □

Satz 6.2.11 Sei

$$A = \begin{pmatrix} D_1 & A_{12} & 0 & & & 0 \\ A_{21} & D_2 & A_{23} & \ddots & & \vdots \\ 0 & A_{32} & D_3 & A_{34} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & A_{n-1,n} \\ 0 & & & 0 & A_{n,n-1} & D_n \end{pmatrix}$$

d.h. blockweise tridiagonal mit regulären Diagonalmatrizen D_i als Diagonalblöcken. Dann ist A konsistent geordnet.

Beweis: Man benutze eine Ähnlichkeitstransformation mit einer Blockdiagonalmatrix mit der Struktur $\text{diag}(I, \alpha I, \dots, \alpha^{N-1}I)$. □

Die Matrix, die bei der Diskretisierung der Poissongleichung und zeilen- oder spaltenweiser Numerierung entsteht, hat nicht die in Satz 6.2.11. angegebene Form. Numeriert man allerdings die Werte $u_{i,j}^h$ in der Reihenfolge $(N, 1), (N, 2), (N - 1, 1), (N, 3), (N - 1, 2), (N - 2, 1)$ usw., dann nimmt die Koeffizientenmatrix diese Form an. Dennoch kann man hier direkt zeigen, daß diese Matrix konsistent geordnet ist (Übung). Es gilt nun

Satz 6.2.12 Sei A konsistent geordnet. Dann gilt mit

$$J \stackrel{\text{def}}{=} D^{-1}(L + U)$$

- a) λ Eigenwert von $J \Leftrightarrow -\lambda$ Eigenwert von J
 b) μ Eigenwert von $B(\omega) \Leftrightarrow \exists \lambda : \lambda$ Eigenwert von J und $\mu\lambda^2\omega^2 = (\mu + \omega - 1)^2$

<<

Beweis a) Sei $J(\alpha) \stackrel{\text{def}}{=} \alpha D^{-1}L + \frac{1}{\alpha} D^{-1}U$. Dann $J(-1) = -J(1)$ während nach Vor. $J(1)$ und $J(-1)$ die gleichen Eigenwerte haben.

Beweis b) “ \Rightarrow ” Nach Herleitung in Satz 6.2.9 ist für $\mu \neq 0$

$$\begin{aligned} \det(B(\omega) - \mu I) &= \det(\omega D^{-1}U + \mu\omega D^{-1}L + (1 - \omega - \mu)I) \\ &= \det \left(\omega\sqrt{\mu} \left(\underbrace{\frac{1}{\sqrt{\mu}} D^{-1}U + \sqrt{\mu} D^{-1}L}_{J(\sqrt{\mu})} \right) + (1 - \omega - \mu)I \right) \\ &= (\omega\sqrt{\mu})^n \det \left(J(\sqrt{\mu}) - \frac{\mu + \omega - 1}{\omega\sqrt{\mu}} I \right) \end{aligned}$$

d.h. $\mu \neq 0$ Eigenwert von $B(\omega) \Leftrightarrow \frac{\mu + \omega - 1}{\omega\sqrt{\mu}}$ Eigenwert von $J(\sqrt{\mu})$

und damit von $J(1)$. Ferner 0 Eigenwert von

$B(\omega) \Rightarrow \det(B(\omega)) = (1 - \omega)^n = 0 \Rightarrow \omega = 1 \Rightarrow$ b) trivial.

“ \Leftarrow ” Sei λ Eigenwert von J und $\mu\lambda^2\omega^2 = (\mu + \omega - 1)^2$.

Im Falle $\mu \neq 0$ $\lambda = \pm \frac{\mu + \omega - 1}{\omega\sqrt{\mu}}$, wegen a) o.B.d.A. $\lambda = \frac{\mu + \omega - 1}{\omega\sqrt{\mu}}$ d.h. λ Eigenwert von

$J(1)$, also auch von $J(\sqrt{\mu})$, also μ Eigenwert von $B(\omega)$. Ist dagegen $\mu = 0$, dann $\omega = 1$, aber $\det(B(1)) = 0$, d.h. 0 ist Eigenwert von $B(1)$ \square

>>

Folgerung:

Ist A konsistent geordnet, dann

$$\rho((D - L)^{-1}U) = \rho^2(D^{-1}(L + U)),$$

d.h. das **Einzelschrittverfahren konvergiert im wesentlichen doppelt so schnell wie das Gesamtschrittverfahren.** (Setze $\omega = 1$ in b): $\mu = \lambda^2!$)

Satz 6.2.13 *A sei konsistent geordnet. Die Eigenwerte λ_i von $J \stackrel{\text{def}}{=} D^{-1}(L + U)$ mögen $\lambda_i \in] - 1, 1[$ erfüllen, $i = 1, \dots, n$.*

Dann gilt mit

$$\omega_{\text{opt}} \stackrel{\text{def}}{=} \frac{2}{1 + \sqrt{1 - \hat{\rho}^2}}, \quad \hat{\rho} \stackrel{\text{def}}{=} \rho(J)$$

$$\rho(B(\omega)) = \begin{cases} \left(\frac{\hat{\rho}\omega}{2} + \frac{1}{2} \sqrt{\hat{\rho}^2 \omega^2 - 4(\omega - 1)} \right)^2 & \text{für } 0 \leq \omega \leq \omega_{\text{opt}} \\ \omega - 1 & \text{für } \omega_{\text{opt}} \leq \omega \leq 2 \end{cases} \quad (6.6)$$

In $[0, \omega_{\text{opt}}]$ ist $\rho(B(\omega))$ streng monoton fallend (und in $[\omega_{\text{opt}}, 2]$ streng monoton wachsend, letzteres jedoch ist trivial wegen (6.6)).

<<

Beweis: Auflösung der quadratischen Gleichung in Satz 6.2.12 liefert

$$\begin{aligned} \mu_i &= \frac{1}{2} \left(\lambda_i^2 \omega^2 - 2(\omega - 1) \pm \sqrt{(\lambda_i^2 \omega^2 - 2(\omega - 1))^2 - 4(\omega - 1)^2} \right) \\ &= \frac{1}{2} \left(\lambda_i^2 \omega^2 - 2(\omega - 1) \pm \sqrt{\lambda_i^4 \omega^4 - 4(\omega - 1)\lambda_i^2 \omega^2} \right) \\ &= \frac{1}{4} \lambda_i^2 \omega^2 + \frac{1}{4} \lambda_i^2 \omega^2 + 1 - \omega \pm \sqrt{\frac{\lambda_i^2 \omega^2}{4} + (1 - \omega)} \cdot 2 \frac{\lambda_i \omega}{2} \\ &= \left(\frac{\lambda_i \omega}{2} \pm \sqrt{\frac{\lambda_i^2 \omega^2}{4} + (1 - \omega)} \right)^2 \\ &= \frac{1}{4} \left(\lambda_i \omega \pm \sqrt{\lambda_i^2 \omega^2 + 4(1 - \omega)} \right)^2 \end{aligned}$$

Hierin ist der Radikant positiv, solange

$$0 \leq \omega \leq \frac{2}{1 + \sqrt{1 - \lambda_i^2}} =: \omega_i$$

Für $\omega \geq \omega_i$ gilt $|\mu_i| = \omega - 1$

(Beachte $z \in \mathbb{C} \Rightarrow |z|^2 = (\Re z)^2 + (\Im z)^2$).

ω_i ist ≥ 1 (beachte $0 \leq \lambda_i^2 < 1$) und monoton wachsend mit λ_i , dies liefert bereits den zweiten Teil von (6.6). Für $0 \leq \omega \leq \omega_i$ ergibt sich der betragsgrößere Wert μ_i aus

$$|\mu_i| = \frac{1}{4} \left(|\lambda_i| \omega + \sqrt{\lambda_i^2 \omega^2 + 4(1 - \omega)} \right)^2 \geq \omega - 1$$

und dieser Wert wächst monoton mit $|\lambda_i|$, dies liefert den ersten Teil von (6.6). Schließlich ergibt Differentiation nach ω die dritte Behauptung. \square

>>

Der Graph von $\rho(B(\omega))$ hat folgendes Aussehen

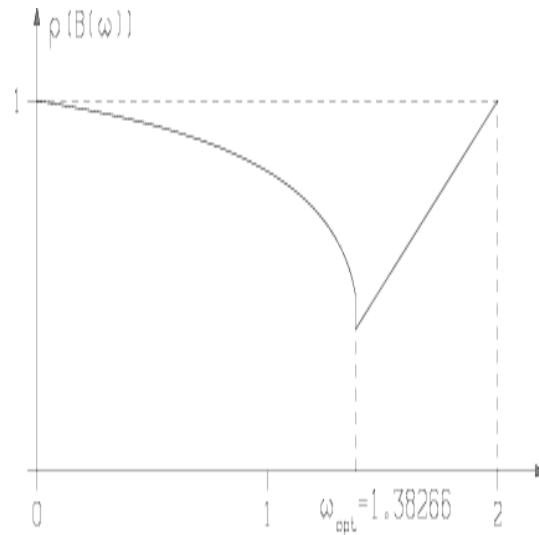


Abbildung 6.2.1

In der Abbildung ist $\rho(D^{-1}(L + U)) = 0.9$ angenommen.

Man erkennt den großen Gewinn an Konvergenzgeschwindigkeit bei der Wahl von $\omega = \omega_{opt}$. Allerdings sollte man in der Praxis ω stets überschätzen, wie man an der linksseitigen “senkrechten Tangente” erkennt. Dazu benötigt man eine obere Schranke für $\rho(D^{-1}(L + U))$. Zu deren Berechnung in der Praxis vgl. Satz 3.2.15 unten. Ohne Beweis sei noch folgendes Resultat von Vandergraft über den Zusammenhang zwischen Kondition von A und der Konvergenzgeschwindigkeit des SOR-Verfahrens bei optimal gewähltem Relaxationsparameter angegeben:

Satz 6.2.14 Sei A blockweise tridiagonal mit Diagonalblöcken I , symmetrisch und positiv definit. Dann gilt

$$\frac{\kappa - 1}{\kappa + 1} \geq \rho(B(\omega_{opt})) \geq \frac{\kappa - \beta}{\kappa + \beta}$$

$$\text{mit } \kappa \stackrel{def}{=} \text{cond}_{\|\cdot\|_2}^{1/2}(A) \quad \text{und} \quad \beta \stackrel{def}{=} \left(2 + \frac{4}{\kappa}\right)^{1/2}$$

□

Für eine schlecht konditionierte Matrix ist also SOR mit ω_{opt} immer noch sehr langsam, allerdings viel schneller als das Einzelschritt- und das Gesamtschrittverfahren. Unter den gleichen Bedingungen wie in Satz 6.2.14 gilt nämlich

$$\rho^2(\underbrace{D^{-1}(L + U)}_I) = \rho((D - L)^{-1}U)$$

und

$$\rho(D^{-1}(L + U)) = 1 - \lambda_{min}(A) = \lambda_{max}(A) - 1 = 1 - \lambda_{max}(A)/\text{cond}_{\|\cdot\|_2}(A)$$

mit

$$1 \leq \lambda_{max}(A) \leq 2$$

Beispiel 6.2.3

$$\text{cond}_{\|\cdot\|_2}(A) = 10000 \quad \kappa = 100$$

$$\rho_{\text{Gesamtschritt}} \geq 0.9998$$

$$\rho_{\text{Einzelschritt}} \geq 0.99986$$

$$\rho_{\text{SOR}_{\text{opt}}} \leq 1 - 2/101 = 0.980198$$

$$0.9998^{1000} = 0.8187$$

$$0.9996^{1000} = 0.67026$$

$$0.980198^{1000} = 2.059 \cdot 10^{-9}$$

} Fehlerfaktor nach
1000 Schritten

□

Es fragt sich nun, wie man den Spektralradius der Iterationsmatrix mit geringem Aufwand bestimmen kann. Von speziellen Startvektoren abgesehen kann man dies erreichen, wenn man $\|Ax_k - b\|^{1/k}$ betrachtet. Es gilt nämlich

Satz 6.2.15 Sei A regulär, $Ax^* = b \iff x^* = Gx^* + g$ $\rho(G) < 1$ und $x_{k+1} = Gx_k + g \quad (\forall k)$. Dann gilt für jede Norm

$$\rho(G) \geq \overline{\lim}_{k \rightarrow \infty} \|Ax_k - b\|^{1/k} \quad (6.7)$$

mit Gleichheit für fast alle x_0

Beweis: Sei T die Matrix, die G auf Jordannormalform ähnlichkeitstransformiert:

$$T^{-1}GT = J.$$

Wegen

$$x_k - x^* = G^k(x_0 - x^*)$$

ist

$$Ax_k - b = A(x_k - x^*) = ATJ^kT^{-1}(x_0 - x^*). \quad (6.8)$$

J ist eine Blockdiagonalmatrix mit Diagonalblöcken J_i , $i = 1, \dots, s$ und jedes J_i hat die Form

$$J_i = \lambda_i I + \hat{J}_i \quad \text{mit der Dimension } n_i$$

Dabei ist \hat{J}_i eine Matrix, bei der nur die Elemente auf der ersten Superdiagonalen ungleich null, und zwar gleich eins, sind. Deshalb ist

$$J^k = \text{blockdiag}(J_1^k, \dots, J_s^k)$$

und für $k \geq n_i - 1$

$$J_i^k = \sum_{j=0}^{n_i-1} \binom{k}{j} \lambda_i^{k-i} (\hat{J}_i)^j$$

weil $(\hat{J}_i)^j = O$ für $j > n_i - 1$. Wir klammern in (6.8) rechts $\rho(G)$ aus und erhalten

$$\|Ax_k - b\|^{1/k} \leq \rho(G) (\|A\| \|T\| \|T^{-1}(x_0 - x^*)\|)^{1/k} \|(J/\rho(G))^k\|^{1/k}.$$

Alle Elemente von $(J/\rho(G))^k$ sind aber kleiner oder gleich k^n und wegen

$$(C \cdot n \cdot k^n)^{1/k} \rightarrow 1 \quad \text{für } k \rightarrow \infty$$

folgt zunächst die behauptete Abschätzung nach unten. Wir bemerken nun, daß in $(J/\varrho(G))^k$ alle Diagonalblöcke gegen null gehen, die nicht zu Eigenwerten mit Betrag $\varrho(G)$ gehören. Wir setzen

$$T^{-1}(x_0 - x^*) \stackrel{def}{=} y = (y_1^T, \dots, y_s^T)^T$$

und nehmen an, daß mindestens ein y_i , das zu einem J_i mit betragsmaximalem Eigenwert gehört, nicht null ist. Fast alle x_0 führen zu solch einem y . Dann ist nach der gleichen Rechnung

$$T^{-1}A^{-1}(Ax_k - b) = ((J_1^k y_1)^T, \dots, (J_s^k y_s)^T)^T$$

und daher unter Ausnutzung der Äquivalenz der Normen und Übergang zur Maximumnorm mit $\|\cdot\| \geq c\|\cdot\|_\infty$

$$(\|T^{-1}A^{-1}\|)^{1/k} \|Ax_k - b\|^{1/k} \geq \varrho(G) \max\{\|(J_i/\varrho(G))^k y_i\|_\infty^{1/k}\} c^{1/k}$$

und Grenzübergang liefert nun die Behauptung mit Gleichheit. \square

6.3 Blockiterationsverfahren

Folgende Verallgemeinerung der bisher betrachteten Iterationsverfahren ist naheliegend: Man partitioniert Matrix, Lösungsvektor und rechte Seite in Blöcke bzw. Teilvektoren:

$$A = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & & \\ \vdots & \vdots & & \vdots \\ A_{n1} & \cdots & \cdots & A_{nn} \end{pmatrix} \quad x = \begin{pmatrix} x_1 \\ \vdots \\ \vdots \\ x_n \end{pmatrix} \quad b = \begin{pmatrix} b_1 \\ \vdots \\ \vdots \\ b_n \end{pmatrix}$$

und setzt nun in der Herleitung der Verfahren

$$D = \begin{pmatrix} A_{11} & & & \\ & \ddots & & \\ & & \ddots & \\ & & & A_{nn} \end{pmatrix} - L = \begin{pmatrix} 0 & & & \\ A_{21} & \ddots & & \\ & & \ddots & \\ A_{n1} & \cdots & A_{n,n-1} & 0 \end{pmatrix} \quad (6.9)$$

$$-U = \begin{pmatrix} 0 & A_{12} & \cdots & A_{1n} \\ & \ddots & & \\ & & \ddots & A_{n-1,n} \\ & & & 0 \end{pmatrix}$$

So lautet z.B. die Iterationsvorschrift für das Block-SOR-Verfahren

$$A_{ii} x_{i,k+1} = \omega(b_i - \sum_{j=1}^{i-1} A_{ij} x_{j,k+1} - \sum_{j=i}^n A_{ij} x_{j,k}) + A_{ii} x_{i,k}$$

$$i = 1, \dots, n, \quad k = 0, 1, 2, \dots$$

Diese Vorgehensweise erfordert pro Schritt die Auflösung von n “kleinen” Gleichungssystemen. Dies kann jedoch lohnend sein, wenn z.B. die A_{ii} einfach gebaut sind (z.B. Dreibandmatrizen wie sie bei Differenzenverfahren für elliptische Randwertprobleme bei zeilen- oder spaltenweiser Numerierung auftreten). Die Sätze 6.2.9 – 6.2.13 lassen sich auf solche Blockverfahren leicht übertragen z.B. gilt

Satz 6.3.1 *Sei A eine symmetrische Block-Tridiagonalmatrix und positiv definit. Dann konvergiert das Block-SOR-Verfahren für $0 < \omega < 2$. Der optimale Block-SOR-Parameter ist gegeben durch*

$$\omega_{\text{opt}} = \frac{2}{1 + \sqrt{1 - \mu^2}}, \quad \mu = \rho(D^{-1}(L + U))$$

mit D, L, U aus (6.9) und die Funktion $\rho((D - \omega L)^{-1}((1 - \omega)D + \omega U))$ hat die gleichen Eigenschaften wie in Satz 6.2.13 angegeben. \square

Oft ist das Block-SOR-Verfahren nicht aufwendiger als das gewöhnliche SOR-Verfahren, aber

$$(\rho_{\text{opt}})_{\text{Block-SOR}} \approx (\rho_{\text{opt}})_{\text{SOR}}^{\sqrt{2}} \quad \text{für } N \gg 1!$$

6.4 Das cg-Verfahren von Hestenes und Stiefel

Im Beweis von Satz 6.2.10 hatten wir gesehen, daß man das SOR-Verfahren zur Lösung von $Ax = b$ mit A symmetrisch positiv definit als Minimierungsverfahren für die Funktion

$$f(x) = \frac{1}{2}x^T Ax - b^T x \quad (\nabla f(x) = Ax - b)$$

auffassen kann, wobei man in zyklischer Reihenfolge entlang der Koordinatenrichtungen fortschreitet. Es ist schon vom Fall $n = 2$ her anschaulich klar, daß die Koordinatenachsen als Fortschreitrichtungen keineswegs besonders günstig sein müssen. Im Fall $n = 2$ stellen die Kurven $f(x) = c$ konzentrische Ellipsen dar. Ein schrittweise Minimierung von f entlang den Hauptachsen der Ellipsen würde die Minimalstelle von f (d.h. die Lösung von $Ax^* = b$) in zwei Schritten liefern und entsprechendes gilt auch für allgemeines n . Diese Hauptachsen sind ein **Spezialfall** sogenannter A -orthogonaler Richtungen. Sie sind zugleich auch orthogonal. Die Bestimmung der Hauptachsen, das sind die Eigenvektoren von A , wäre aber zur Lösung von $Ax = b$ ein unsinniger Aufwand. Wir zeigen im Folgenden, daß die Minimierung entlang sogenannter A -orthogonaler (auch A -konjugiert genannt) Richtungen ebenfalls zu einem finiten Verfahren führt. Abbildung 3.4.1 zeigt die Konstruktion eines A -orthogonalen p_1 zu gegebenem p_0 für $n = 2$.

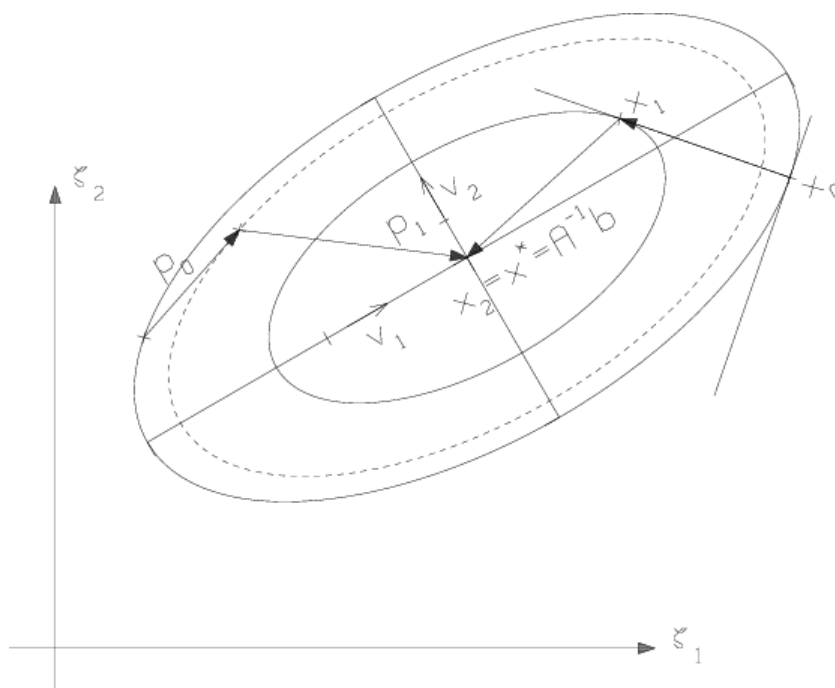


Abbildung 6.4.1

Definition 6.4.1 Sei $A \in \mathbb{R}^{n \times n}$ symmetrisch positiv definit. Ein System von Vektoren $\{p_0, \dots, p_{n-1}\}$ heißt A -orthogonal, falls

$$p_j^T A p_k = \kappa_j \delta_{jk} \quad \text{mit } \kappa_j > 0, \quad \delta_{jk} = \begin{cases} 0 & j \neq k \\ 1 & j = k \end{cases}$$

□

Satz 6.4.1 Sei $f(x) = \frac{1}{2}x^T A x - b^T x$, $A \in \mathbb{R}^{n \times n}$ symmetrisch positiv definit. $\{p_0, \dots, p_{n-1}\}$ sei ein System von A -orthogonalen Richtungen. $x_0 \in \mathbb{R}^n$ sei beliebig und

$$x_{k+1} = x_k - \sigma_k p_k, \quad \sigma_k = (p_k)^T (A x_k - b) / p_k^T A p_k \quad k = 0, 1, \dots, n - 1$$

Dann gilt

- (i) x_{k+1} minimiert $f(x_k - \sigma p_k)$ bzgl. σ
- (ii) $(A x_k - b)^T p_j = 0$, $j = 0, \dots, k - 1$, d.h. $x_k = \operatorname{argmin} \{f(x) : x \in x_0 + \operatorname{span} \{p_0, \dots, p_{k-1}\}\}$
- (iii) $x_n = x^* = A^{-1}b$

Beweis:

$$\begin{aligned} \text{(i)} \quad \frac{d}{d\sigma} f(x_k - \sigma p_k) = 0 &\Leftrightarrow -\nabla f(x_k - \sigma p_k)^T p_k = 0 \\ &\Leftrightarrow p_k^T (A(x_k - \sigma p_k) - b) = 0 \\ &\Leftrightarrow \sigma = p_k^T (A x_k - b) / p_k^T A p_k \end{aligned}$$

(ii) Wird induktiv gezeigt. $k = 1$ ist die Folge von (i).

Gelte $(Ax_k - b)^T p_j = 0$ für $j = 0, 1, \dots, k-1$

Zu zeigen: $(Ax_{k+1} - b)^T p_j = 0$ für $j = 0, \dots, k$

$j = k$ ist wiederum klar wegen (i).

Aber

$$\begin{aligned} (Ax_{k+1} - b)^T p_j &= (A(x_k - \sigma_k p_k) - b)^T p_j \\ &= (Ax_k - b - \sigma_k A p_k)^T p_j \\ &= (Ax_k - b)^T p_j - \sigma_k p_k^T A p_j = 0 \end{aligned}$$

nach Ind.-Vor. bzw. Def $\{p_j\}$ für $j < k$.

(iii) (ii) mit $k = n$ liefert $(Ax_n - b)^T \underbrace{(p_0, \dots, p_{n-1})}_{\text{reguläre Matrix}} = 0$, d.h. $Ax_n = b$

□

An der Konstruktion in Satz 6.4.1 erkennt man, daß man die Richtung p_k erst benötigt, wenn man x_k schon gefunden hat. So ist es naheliegend, sich A -orthogonale Richtungen p_j durch ein Orthogonalisierungsverfahren bzgl. des Skalarproduktes $(x, y) := x^T A y$ erst im Laufe der Rechnung zu verschaffen, ausgehend von

$$p_0 := Ax_0 - b = \nabla f(x_0)$$

mit Hilfe der Gram-Schmidt-Orthogonalisierung

$$p_k = \nabla f(x_k) + \sum_{j=0}^{k-1} \beta_{kj} p_j, \quad \begin{array}{l} \nabla f(x_k) \neq 0 \\ \text{sonst } x_k = x^* \end{array}$$

mit β_{kj} so gewählt, daß $p_j^T A p_k = 0$ für $j = 0, \dots, k-1$.

Wesentlich für die Praktikabilität des Verfahrens ist nun, daß sich herausstellt

$$\beta_{k,0} = \dots = \beta_{k,k-2} = 0, \quad \beta_{k,k-1} = \nabla f(x_k)^T \nabla f(x_k) / \nabla f(x_{k-1})^T \nabla f(x_{k-1}),$$

so daß sich ein außerordentlich niedriger Rechenaufwand und Speicherbedarf pro Schritt ergibt.

Satz 6.4.2 Es sei $f(x) = \frac{1}{2} x^T A x - b^T x$, $A \in \mathbb{R}^{n \times n}$ symm. pos. def., $x_0 \in \mathbb{R}^n$ sei beliebig und

$$x_{k+1} = x_k - \sigma_k p_k \quad \text{mit } p_k = \begin{cases} \nabla f(x_k) & \text{für } k = 0 \\ \nabla f(x_k) + \frac{\|\nabla f(x_k)\|_2^2}{\|\nabla f(x_{k-1})\|_2^2} p_{k-1} & \text{für } k > 0 \end{cases}$$

und $\sigma_k = \nabla f(x_k)^T p_k / p_k^T A p_k$

Dann gilt:

$\nabla f(x_j) \neq 0$ für $j = 0, \dots, k$

$\Rightarrow \{p_0, \dots, p_k\}$ sind A -orthogonal $\Rightarrow \exists N \leq n : x_N = A^{-1}b$.

Beweis: Sei im Folgenden $r_j = \nabla f(x_j) = Ax_j - b$ und $\beta_j := \frac{\|r_j\|_2^2}{\|r_{j-1}\|_2^2}$

Die Behauptung wird induktiv bewiesen.

Man beachte $-Ap_k = \frac{1}{\sigma_k}(r_{k+1} - r_k)$

$k = 0$: $r_0 \neq 0 \Leftrightarrow p_0 = r_0 \neq 0$

$k = 1$: Zu zeigen $r_1 \neq 0 \Rightarrow p_1^T Ap_0 = 0$, $p_1 \neq 0$.

Es ist

$$\begin{aligned} p_1 = r_1 + \beta_1 r_0 \Rightarrow -p_1^T Ap_0 &= p_1^T \frac{1}{\sigma_0}(r_1 - r_0) \\ &= \frac{1}{\sigma_0}(r_1^T + \beta_1 r_0^T)(r_1 - r_0) \\ &= \frac{1}{\sigma_0}(r_1^T r_1 - \underbrace{r_1^T r_0}_{=0} + \frac{r_1^T r_1}{r_0^T r_0} \underbrace{r_0^T r_1}_{=0} - r_1^T r_1) = 0. \end{aligned}$$

Wegen $r_1^T p_1 = r_1^T r_1 \neq 0$ ist $p_1 \neq 0$.

$k \rightarrow k+1$: Zu zeigen $r_{k+1} \neq 0$ und $\{p_0, \dots, p_k\}$ A -orthogonal \Rightarrow

$\{p_0, \dots, p_{k+1}\}$ A -orthogonal, d.h. $p_{k+1}^T Ap_j = 0$ für $j = 0, \dots, k$ und $p_{k+1} \neq 0$. Zunächst wegen $r_{k+1}^T p_k = 0$ $r_{k+1}^T p_{k+1} = r_{k+1}^T r_{k+1} \neq 0$, d.h. $p_{k+1} \neq 0$.

Für $j < k$ ergibt sich aus

$$\begin{aligned} -Ap_j &= \frac{1}{\sigma_j}(r_{j+1} - r_j) = \frac{1}{\sigma_j}(p_{j+1} - \beta_{j+1}p_j - p_j + \beta_j p_{j-1}) \\ -p_{k+1}^T Ap_j &= -(r_{k+1}^T + \beta_{k+1}p_k^T)Ap_j = -r_{k+1}^T Ap_j \\ &= \frac{1}{\sigma_j}(r_{k+1}^T p_{j+1} - \beta_{j+1}r_{k+1}^T p_j - r_{k+1}^T p_j + \beta_j r_{k+1}^T p_{j-1}) = 0 \end{aligned}$$

nach Satz 6.4.1(ii).

Für $j = k$ gilt

$$\begin{aligned} -p_{k+1}^T Ap_k &= (r_{k+1}^T + \beta_{k+1}p_k^T) \cdot \frac{1}{\sigma_k}(r_{k+1} - r_k) \\ &= \frac{1}{\sigma_k}(r_{k+1}^T r_{k+1} + \beta_{k+1} \underbrace{p_k^T r_{k+1}}_{=0} - r_{k+1}^T r_k - \beta_{k+1} r_k^T p_k) \\ &= \frac{1}{\sigma_k} \underbrace{(r_{k+1}^T r_{k+1} (1 - \frac{r_k^T p_k}{r_k^T r_k)})}_{=0} - r_{k+1}^T r_k \\ &= -\frac{1}{\sigma_k} \underbrace{r_{k+1}^T (p_k - \beta_k p_{k-1})}_{=0} = \frac{\beta_k}{\sigma_k} r_{k+1}^T p_{k-1} \\ &= \frac{\beta_k}{\sigma_k} (A(x_k - \sigma_k p_k) - b)^T p_{k-1} \\ &= \frac{\beta_k}{\sigma_k} (r_k^T p_{k-1} - \sigma_k p_k^T Ap_{k-1}) = 0 \end{aligned}$$

□

Beispiel 6.4.1 $n = 2$, $A = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}$, $b = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $x^{(0)} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

$$\begin{aligned} \Rightarrow p^{(0)} &= \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \quad \delta_0 = 1, \quad \sigma_0 = \frac{1}{2}, \quad x^{(1)} = \begin{pmatrix} \frac{1}{2} \\ 0 \end{pmatrix}, \\ Ax^{(1)} - b &= A(x^{(0)} - \sigma_0 \cdot p^{(0)}) - b = \underbrace{Ax^{(0)} - b}_{=r^{(0)}} - \sigma_0 \underbrace{Ap^{(0)}}_{\substack{\text{bereits für} \\ \sigma_0 \text{ berechnet}}} \\ Ax^{(1)} - b &= \begin{pmatrix} 0 \\ \frac{1}{2} \end{pmatrix} = r^{(1)} \\ p^{(1)} &= \begin{pmatrix} 0 \\ \frac{1}{2} \end{pmatrix} + \frac{1}{1} \begin{pmatrix} -1 \\ 0 \end{pmatrix} = \begin{pmatrix} -\frac{1}{4} \\ \frac{1}{2} \end{pmatrix} \\ Ap^{(1)} &= \begin{pmatrix} 0 \\ \frac{1}{4} \end{pmatrix} \\ p^{(1)T} Ap^{(1)} &= \frac{1}{8}, \quad \sigma_1 = 2 \\ x^{(2)} &= \begin{pmatrix} \frac{1}{2} \\ 0 \end{pmatrix} - 2 \begin{pmatrix} -\frac{1}{4} \\ \frac{1}{2} \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \end{aligned}$$

□

Dieses Verfahren liefert also, trotz seiner iterativen Struktur, bei exakter Rechnung die Lösung eines linearen Gleichungssystems mit positiv definiten Koeffizientenmatrix in höchstens n Schritten. Ist A dünnbesetzt, dann sind die einzelnen Rechenschritte auch nur sehr wenig aufwendig. Das Verfahren erweist sich allerdings als ziemlich empfindlich gegen Rundungsfehler, so daß man unter Rundungseinfluß nach n Schritten eine mehr oder weniger verfälschte Näherungslösung erhält. Man kann dann das Verfahren mit x_n neu starten oder auch einfach formal fortsetzen. Es mag dem Anfänger erstaunlich erscheinen, daß nun ein Verfahren wie SOR überhaupt mit dem cg-Verfahren konkurrieren kann. Dies hat damit zu tun, daß man bei den großen linearen Gleichungssystemen in der Praxis gar keine exakte Lösung braucht und eventuell die Rechnung schon nach weniger als n Schritten abbrechen möchte.

Während nun z.B. SOR pro Schritt eine ziemlich gleichmäßige Fehlerreduktion gewährleistet, ist das Verhalten des cg-Verfahrens in dieser Hinsicht etwas irregulär. Der folgende Satz gibt Auskunft über die pro Schritt zu erwartende Fehlerreduktion:

Satz 6.4.3 Sei $A \in \mathbb{R}^{n \times n}$ symmetrisch und positiv definit mit den Eigenwerten $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n > 0$, $b \in \mathbb{R}^n$ bel., $x^* := A^{-1}b$ und

$$E(x) := \frac{1}{2}(x - x^*)^T A(x - x^*).$$

Dann gilt für die mit dem cg-Verfahren gebildete Folge $\{x_k\}$

$$\begin{aligned} E(x_{k+1}) &= \frac{1}{2} \min_{P_k \in \Pi_k} (x_0 - x^*)^T A(I + AP_k(A))^2 (x_0 - x^*) \\ &\leq \min_{P_k \in \Pi_k} \max_i |1 + \lambda_i P_k(\lambda_i)|^2 E(x_0) \\ &\leq \left(\frac{\lambda_{k+1} - \lambda_n}{\lambda_{k+1} + \lambda_n} \right)^2 E(x_0) \quad k = 0, 1, \dots, n-1 \end{aligned}$$

Beweis: Wir zeigen zunächst, daß $p_j = P_j(A)r_0$ mit $P_j \in \Pi_j$ und $r_0 = Ax_0 - b$. Dazu gehen wir induktiv vor:

$$\begin{aligned} p_0 &= r_0 = P_0(A)r_0 \text{ mit } P_0(\tau) \equiv 1 \in \Pi_0 \\ p_j &= Ax_j - b + \beta_j p_{j-1} \\ &= A(x_0 - \sum_{i=0}^{j-1} \sigma_i p_i) - b + \beta_j p_{j-1} \\ &= \underbrace{Ax_0 - b}_{r_0} - \sum_{i=0}^{j-1} \sigma_i A p_i + \beta_j p_{j-1} = P_j(A)r_0 \\ \text{mit } P_j(\tau) &= 1 + \beta_j P_{j-1}(\tau) - \sum_{i=0}^{j-1} \sigma_i \tau \underbrace{P_i(\tau)}_{\in \Pi_{j-1}} \in \Pi_j \end{aligned}$$

nach Induktionsvoraussetzung. Somit gilt

$$\begin{aligned} x_j - x^* &= x_0 - x^* - \sum_{i=0}^{j-1} \sigma_i p_i = x_0 - x^* - \sum_{i=0}^{j-1} \sigma_i P_i(A)r_0 \\ &= (I - \sum_{i=0}^{j-1} \sigma_i P_i(A)A)(x_0 - x^*) \\ &= (I + AQ_{j-1}(A))(x_0 - x^*) \\ \text{mit } Q_{j-1}(\tau) &= - \sum_{i=0}^{j-1} \sigma_i P_i(\tau) \in \Pi_{j-1}. \end{aligned}$$

Sei $A = V^T \text{diag}(\lambda_i)V$ mit $V^T V = V V^T = I$.

Dann gilt mit $\Lambda := \text{diag}(\lambda_1, \dots, \lambda_n)$

$$V(x_j - x^*) = \underbrace{(I + \Lambda Q_{j-1}(\Lambda))}_{\text{Diagonalmatrix}} \underbrace{V(x_0 - x^*)}_{=:y}$$

und (wegen der Vertauschbarkeit von Diagonalmatrizen)

$$\begin{aligned} E(x_{k+1}) &= \frac{1}{2}(x_{k+1} - x^*)^T V^T \Lambda V (x_{k+1} - x^*) \\ &= \frac{1}{2} y^T (I + \Lambda Q_k(\Lambda))^2 \Lambda y \quad \text{mit } y = V(x_0 - x^*) = \begin{pmatrix} \eta_1 \\ \vdots \\ \eta_n \end{pmatrix} \\ &= \frac{1}{2} \sum_{i=1}^n \eta_i^2 \lambda_i (1 + \lambda_i Q_k(\lambda_i))^2 \end{aligned}$$

Wegen Satz 6.4.1 (ii) ist dies der kleinste Wert für $E(x)$, der mit der Konstruktion

$$p_j = F_j(A)r_0, \quad x_{j+1} = x_j - \tau_j p_j \quad j = 0, \dots, k, \quad F_j(A) \in \Pi_j$$

überhaupt erreicht werden kann, d.h. es gilt (beachte, daß die Abhängigkeit von den β 's und σ 's nur in Q_k steht)

$$\begin{aligned} E(x_{k+1}) &= \min_{F_k \in \Pi_k} \frac{1}{2} \sum_{i=1}^n \eta_i^2 \lambda_i (1 + \lambda_i F_k(\lambda_i))^2 \\ &\leq \min_{F_k \in \Pi_k} \max_i (1 + \lambda_i F_k(\lambda_i))^2 \cdot \underbrace{\frac{1}{2} \sum_{i=1}^n \eta_i^2 \lambda_i}_{=E(x_0)}. \end{aligned}$$

Wir konstruieren nun ein spezielles $F_k^* \in \Pi_k$ in der folgenden Weise:

$$F_k^*(\lambda) \stackrel{\text{def}}{=} \frac{1}{\lambda} \left(\frac{(-1)^{k+1}}{\lambda_1 \cdots \lambda_k \frac{\lambda_{k+1} + \lambda_n}{2}} (\lambda - \lambda_1)(\lambda - \lambda_2) \cdots (\lambda - \lambda_k) \left(\lambda - \frac{\lambda_{k+1} + \lambda_n}{2} \right) - 1 \right).$$

Es ist also

$$1 + \lambda F_k^*(\lambda) = 0 \quad \text{für } \lambda \in \{\lambda_1, \dots, \lambda_k\} \cup \left\{ \frac{\lambda_{k+1} + \lambda_n}{2} \right\}$$

Weil $1 + \lambda F_k^*(\lambda) \in \Pi_{k+1}$, hat dieses Polynom notwendig folgenden Verlauf:

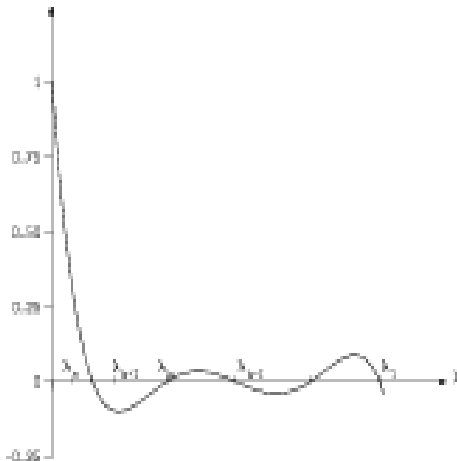


Abbildung 6.4.2

Bis zu einer Stelle $\tilde{\lambda} \in]\frac{\lambda_{k+1} + \lambda_n}{2}, \lambda_k[$ ist $1 + \lambda F_k^*(\lambda)$ monoton fallend und konvex, in $[\tilde{\lambda}, \lambda_k]$ monoton wachsend, betragsmäßig monoton fallend. $\tilde{\lambda}$ ist definiert durch $\tilde{\lambda} \in [\frac{\lambda_n + \lambda_{k+1}}{2}, \lambda_k]$ und $(F_k^*)'(\tilde{\lambda})\tilde{\lambda} + F_k^*(\tilde{\lambda}) = 0$.
Folglich gilt

$$\lambda \in [\lambda_n, \lambda_{k+1}] \Rightarrow |1 + \lambda F_k^*(\lambda)| \leq \left| 1 - \frac{2\lambda}{\lambda_{k+1} + \lambda_n} \right| \leq \frac{\lambda_{k+1} - \lambda_n}{\lambda_{k+1} + \lambda_n}$$

□

Falls also z.B.

$$\lambda_1 \gg \lambda_n \quad \text{und} \quad \lambda_i = \lambda_{i-1} - \varepsilon \quad i = 2, \dots, n-1 \quad \text{mit} \quad \varepsilon \ll \lambda_1 - \lambda_n,$$

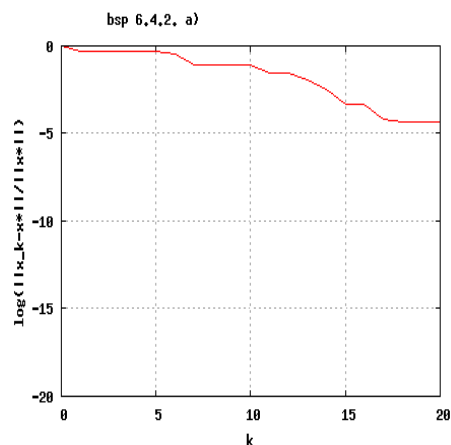
dann liefern die ersten $n-1$ Schritte nur eine sehr geringfügige Fehlerverkleinerung.

Beispiel 6.4.2 Wir betrachten bei einer symmetrisch positiv definiten Matrix mit der Eigenvektormatrix $(\sin(\frac{ij\pi}{n+1}))2/\sqrt{n+1}$ folgende Eigenwertverteilungen

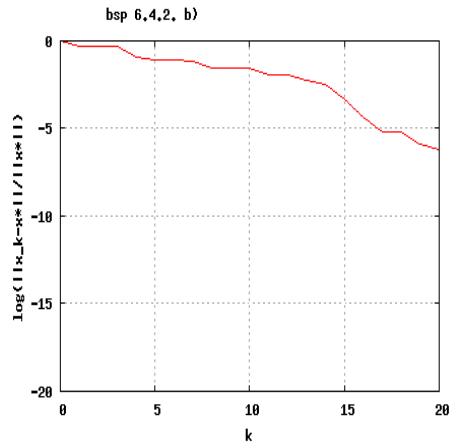
- a) $1, 2, 3, \dots, 10, 5500, 6000, 6500, 7000, \dots, 10000$.
- b) $1, 2, 3, \dots, 10, 9100, 9200, 9300, \dots, 10000$.
- c) $1, 2, 3, \dots, 10, 9910, 9920, 9930, \dots, 10000$.
- d) $1, 2, 3, \dots, 10, 9991, 9992, 9993, \dots, 10000$.
- e) $1, 2, 3, \dots, 19, 10000$.
- f) $1, 8200, 8300, 8400, \dots, 10000$.
- g) $1, 500, 1000, 1500, 2000, \dots, 4500, 5500, \dots, 10000$.

Die Grafiken zeigen jeweils in einer halblogarithmischen Darstellung die Grösse $\|x_k - x^*\|/\|x^*\|$. Theoretisch sollte der Fehler nach 20 Iterationen null sein. Wir erhalten

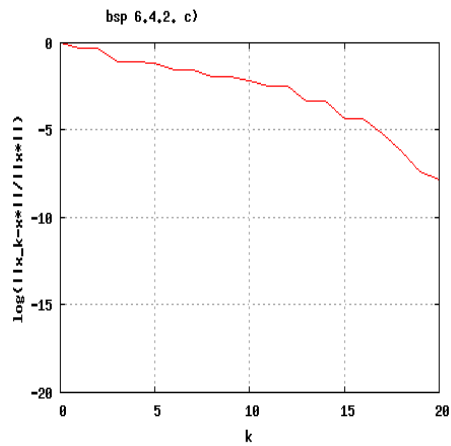
1. Resultat nach 20 Iterationen: skaliertes Fehler in x $1.0E-4$



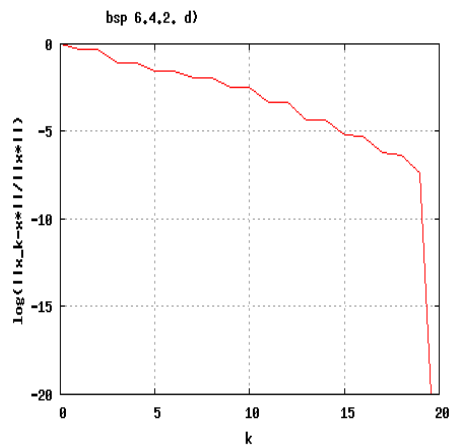
2. *Resultat nach 20 Iterationen: skaliertes Fehler in $x = 1.0E-6$*



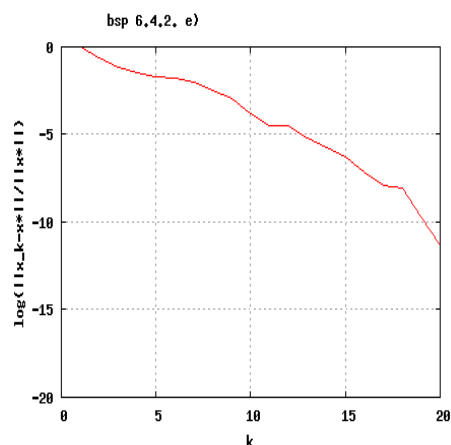
3. *Resultat nach 20 Iterationen: skaliertes Fehler in $x = 1.0E-7$*



4. *Resultat nach 20 Iterationen: skaliertes Fehler in $x = 1.0E-20$*



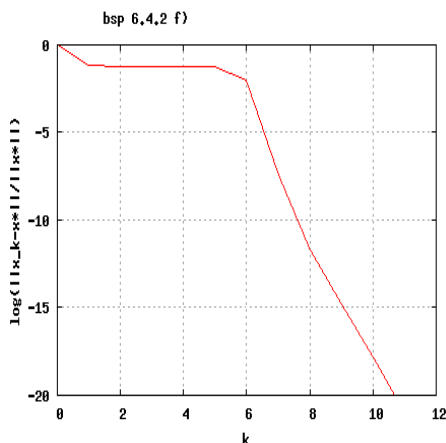
5. *Resultat nach 20 Iterationen: skaliertes Fehler in $x = 1.0E-12$*



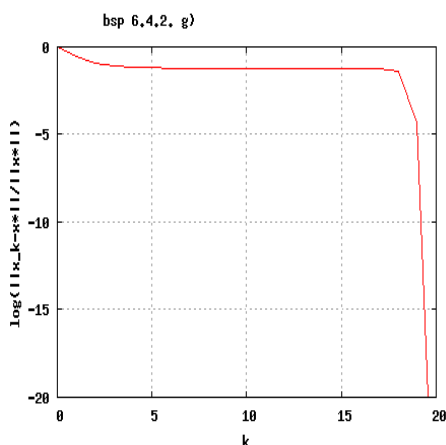
Man erkennt an den Beispielen a)-d), dass die Konvergenz um so besser ist, je dichter die Eigenwerte bei 1 bzw. 10000 zusammen liegen. Dieses Verhalten lässt sich mit Hilfe der Fehlerabschätzung dadurch erklären, dass ein Polynom $1 + \lambda P(\lambda)$, welches an den Stellen 1 und 10000 klein ist, wegen der Stetigkeit auch für Werte in der Nähe noch klein ist. Anders ausgedrückt lässt sich für Werte, die dicht zusammenliegen, besser ein Polynom finden, welches an diesen Stellen klein ist, als bei Werten, die weit auseinander liegen. Allerdings liefern die Daten e) ein schlechteres Ergebnis als bei a), obwohl die Abstände zwischen den Eigenwerten die gleichen sind. Dies liegt daran, dass hier die Spanne der kleinen Eigenwerte bei gleicher Konditionszahl grösser ist. Dies erkennt man, wenn man die Gestalt des Polynoms $Q(\lambda) := 1 + \lambda P(\lambda)$ genauer betrachtet. $P(\lambda)$ ist ein beliebiges Polynom k -ten Grades. Dann ist $Q(\lambda)$ ein Polynom $k + 1$ -ten Grades, welches die Bedingung $Q(0) = 1$ erfüllt. Bis auf diese Bedingung kann $Q(\lambda)$ beliebig gewählt werden. Wegen dieser Bedingung $Q(0) = 1$ werden Polynome $Q(\lambda)$ die für die Eigenwerte um 1 sehr klein sind an der Stelle 10000 von der Tendenz her sehr groß.

Es sind somit für die Konvergenz auch Eigenwerte, die gruppenweise dicht beieinander liegen, günstig (Sofern die Konditionszahl gleich bleibt.)

Diese Tendenz zeigen auch die nächsten beiden Beispiele.



7. Resultat nach 20 Iterationen: skalierter Fehler in $x = 1.0E-20$



Das Beispiel f) liefert bereits nach 12 Iterationen eine bis Maschinengenauigkeit genaue Lösung. Bei dieser Eigenwertverteilung liegen zwar die großen Eigenwerte relativ weit auseinander, allerdings ist 1 der einzige kleine Eigenwert.

Bemerkenswert ist im Fehlerdiagramm die Stufe um die zweite bis sechste Iteration, in der sich die Lösung kaum verbessert. Dies kommt daher, dass die Fehleranteile bezüglich der Eigenvektoren von A zu den großen Eigenwerten einen großen Anteil im Fehler liefern, dieser Anteil dann aber in den ersten Iterationsschritten schnell kleiner wird, bis der Anteil bezüglich der Eigenvektoren großer Eigenwerte sehr klein ist und der Anteil bezüglich der kleinen Eigenwerte langsam verringert werden muss. Das Beispiel g) zeigt ein ähnliches Verhalten, die Konvergenz ist allerdings langsamer, da die Eigenwerte hier weiter auseinander liegen.

Bemerkung 6.4.1 Für das reine Gradientenverfahren mit "optimaler" Schrittweite, (das in dieser Form auch als Richardsonverfahren bezeichnet wird, und für den Spezialfall $\text{diag}(A) = I$ mit dem Jacobiverfahren mit einem abgeänderten Schrittweitenfaktor übereinstimmt)

$$x_{k+1} = x_k - \sigma_k(Ax_k - b), \quad \sigma_k = \|Ax_k - b\|_2^2 / \|A^{\frac{1}{2}}(Ax_k - b)\|_2^2$$

beweist man unter den Vor. von Satz 6.4.3

$$E(x_{k+1}) \leq \left(\frac{\lambda_1 - \lambda_n}{\lambda_1 + \lambda_n} \right)^2 E(x_k)$$

und man kann zeigen, daß hier “im wesentlichen” sogar das Gleichheitszeichen gilt. Bei schlecht konditionierten Matrizen ($\lambda_1/\lambda_n \gg 1$) hat man also nur eine sehr geringe Fehlerreduktion pro Schritt, nämlich

$$\|A^{\frac{1}{2}}(x_{k+1} - x^*)\|_2 \leq \left(1 - \frac{2}{1 + \text{cond}_{\|\cdot\|_2}(A)}\right) \|A^{\frac{1}{2}}(x_k - x^*)\|_2$$

Für das SOR-Verfahren kann man dagegen zeigen, daß

$$\|A^{\frac{1}{2}}(x_{k+1} - x^*)\|_2 \leq \gamma(\omega) \|A^{\frac{1}{2}}(x_k - x^*)\|_2 \quad \text{mit} \quad 0 < \gamma(\omega) < 1$$

und $\gamma(\omega_{\text{opt}}) \leq 1 - \frac{2}{1 + \text{cond}_{\|\cdot\|_2}^{\frac{1}{2}}(A)}$ für $k \geq k_0$.

Es gibt also viele Eigenwertverteilungen (z.B. bei 2.0.), bei denen SOR mit $\omega \approx \omega_{\text{opt}}$ zunächst schneller konvergiert als das cg-Verfahren! In allen Fällen hängt die Konvergenzgeschwindigkeit entscheidend von der Kondition von A ab und es ist naheliegend zu versuchen, durch eine “einfache” Transformation

$$A \mapsto (\hat{L}^T)^{-1} A (\hat{L})^{-1},$$

\hat{L} “Näherung” für den Choleskyfaktor L von A , die Kondition von A zu verbessern. Man kann die Verfahren dabei so umschreiben, daß nur Gleichungssysteme mit \hat{L}, \hat{L}^T zusätzlich gelöst werden müssen, die transformierte Matrix selbst aber nie explizit auftritt, sogenannte **präkonditionierte cg-Verfahren**.

Man kann die Präkonditionierung auch dadurch vornehmen, daß man formal das Gleichungssystem mit einer Matrix von links multipliziert (ohne dies explizit auszuführen). Das Verfahren wird dann mit Originalvariablen (x) gerechnet:

$M =$ Präkonditionierungsmatrix (symmetrisch, positiv definit)

$$\begin{aligned} r_0 &= Ax_0 - b \\ Mp_0 &= r_0 \quad \text{ergibt } p_0 \\ \delta_0 &= r_0^T p_0 \end{aligned}$$

Für $k = 0, 1, \dots$

$$\begin{aligned} y_k &= Ap_k \\ \sigma_k &= \delta_k / y_k^T p_k \\ x_{k+1} &= x_k - \sigma_k p_k \\ r_{k+1} &= r_k - \sigma_k y_k \\ Mz_{k+1} &= r_{k+1} \quad \text{ergibt } z_{k+1} \\ \delta_{k+1} &= r_{k+1}^T z_{k+1} \\ \beta_{k+1} &= \delta_{k+1} / \delta_k \\ p_{k+1} &= z_{k+1} + \beta_{k+1} p_k \end{aligned}$$

²Mit $A = V\Lambda V^H$, $\Lambda = \text{diag}(\lambda_i)$, $\lambda_i > 0$ und $VV^H = V^H V = I$ ist $A^{\frac{1}{2}} = V\Lambda^{\frac{1}{2}}V^H = V \text{diag}(\lambda_i^{\frac{1}{2}})V^H$

6.5 Das Verfahren der generalisierten minimalen Residuen GMRES

In Verbindung mit geeigneten, problemabhängigen Prädiktionierern gehört das cg-Verfahren zu den erfolgreichsten für große Systeme mit symmetrischer, positiv definitiver Koeffizientenmatrix. Man hat deshalb intensiv nach Möglichkeiten gesucht, dieses Verfahren in geeigneter Weise auf Systeme mit nichtsymmetrischer Matrix zu übertragen. Eine zunächst naheliegende Idee, das Gleichungssystem

$$Ax = b$$

durch Multiplikation mit A^T in eines mit symmetrisch positiv definitiver Matrix zu überführen,

$$A^T Ax = A^T b$$

ist nicht empfehlenswert, wegen der damit verbundenen Verschlechterung (Quadrierung) der Kondition. Eine recht erfolgreiche Verallgemeinerung geht aus von der Eigenschaft (ii) in Satz 6.4.1:

$$x_{k+1} = \operatorname{argmin} \{f(x) : x \in x_0 + \operatorname{span} \{p_0, \dots, p_{k-1}\}\}.$$

Aufgrund der Rekursion für die p_j und

$$\begin{aligned} p_0 &= Ax_0 - b = r_0 \\ p_j &= Ax_j - b + \beta_{j,j}p_{j-1} \\ &= Ax_{j-1} - b - \sigma_{j-1}Ap_{j-1} + \beta_{j,j-1}p_{j-1} \end{aligned}$$

folgt, daß

$$\operatorname{span} \{p_0, \dots, p_{k-1}\} = \operatorname{span} \{r_0, Ar_0, \dots, A^{k-1}r_0\}.$$

Diesen Unterraum bezeichnet man als Krylow-Unterraum zu r_0 (er kommt auch in einem von Krylow angegebenen Verfahren zur Berechnung des charakteristischen Polynoms einer Matrix vor). Für das cg-Verfahren kann man die Optimalität somit auch ausdrücken durch

$$\begin{aligned} Ax_k - b &\perp \operatorname{span} \{r_0, Ar_0, \dots, A^{k-1}r_0\} \stackrel{\text{def}}{=} V_k \\ x_k &= x_0 + \hat{V}_k y_k, \quad \hat{V}_k = \text{Basis von } V_k \end{aligned}$$

Für eine allgemeine Matrix ist dieser Ansatz nicht notwendig wohldefiniert, weil $\hat{V}_k^H A \hat{V}_k$ singular sein kann. Jedoch ist

$$Ax_k - b \perp A \hat{V}_k \quad x_k = x_0 + \hat{V}_k y_k$$

6.5. DAS VERFAHREN DER GENERALISIERTEN MINIMALEN RESIDUEN GMRES105

immer wohldefiniert, da $\hat{V}_k^H A^H A \hat{V}_k$ positiv definit ist für Rang $(\hat{V}_k) = k$. Dies ist der Ansatz von GMRES (generalisierte minimale Residuen). Es ergibt sich die Bestimmungsgleichung

$$\hat{V}_k^H A^H (A(x_0 + \hat{V}_k y_k) - b) = 0,$$

d.h.

$$y_k = -(\hat{V}_k^H A^H A \hat{V}_k)^{-1} \hat{V}_k^H A^H r_0.$$

Diese Lösung kann man auch deuten als die von

$$\begin{aligned} \|A(x_0 + \hat{V}_k y_k) - b\|_2^2 &= \min_y \|A(x_0 + \hat{V}_k y) - b\|_2^2 \\ x_k &= x_0 + \hat{V}_k y_k. \end{aligned}$$

Dieses Ausgleichsproblem könnte man über eine QR-Zerlegung von $A\hat{V}_k$ lösen. Geschickter ist es jedoch, induktiv eine Orthogonalbasis Q_k von V_k aufzubauen, mit $r_0/\|r_0\|$ als erster Spalte. Also betrachtet man die Zusammenhänge

$$\begin{aligned} (r_0, \dots, A^{k-1}r_0) &= Q_k R_k, \\ (r_0, \dots, A^k r_0) = (r_0, A Q_k R_k) &= Q_{k+1} R_{k+1}, \\ r_0 = Q_k e_1^{(k)} \|r_0\| &= Q_{k+1} e_1^{(k+1)} \|r_0\|, \quad e_1^{(j)} \text{1. Einheitsvektor in } \mathbb{R}^j. \end{aligned}$$

Die Matrix R_k hat hier die Diagonalelemente $\varrho_{0,0}, \dots, \varrho_{k-1,k-1}$. Dann wird

$$\begin{aligned} Q_k^H (r_0, \dots, A^k r_0) &= (e_1 \|r_0\|, Q_k^H A Q_k R_k) = (I_k, 0) \begin{pmatrix} R_k & c_k \\ 0 & \varrho_{k,k} \end{pmatrix} \\ &= (R_k, c_k) = (\|r_0\| e_1^{(k)}, H_k) \end{aligned}$$

Dabei ist H_k die $k \times k$ Hessenbergmatrix, die entsteht, wenn man von R_{k+1} die erste Spalte und die letzte Zeile streicht. Also ist

$$Q_k^H A Q_k = H_k R_k^{-1} = \hat{H}_k \in \mathbb{C}^{k \times k}$$

selbst eine obere Hessenbergmatrix. Es gilt weiterhin

$$(r_0, A Q_k) = (r_0, A Q_k R_k) \left(\frac{1}{0} \mid \frac{0}{R_k^{-1}} \right) = Q_{k+1} R_{k+1} \left(\frac{1}{0} \mid \frac{0}{R_k^{-1}} \right)$$

Also

$$\begin{aligned} A Q_k &= Q_{k+1} R_{k+1} \left(\frac{1}{0} \mid \frac{0}{R_k^{-1}} \right) \begin{pmatrix} 0 \\ \vdots \\ I_k \end{pmatrix} \\ &= Q_{k+1} R_{k+1} \begin{pmatrix} 0 \\ R_k^{-1} \end{pmatrix} = Q_{k+1} \begin{pmatrix} \|r_0\| & \vdots & & & \\ 0 & \vdots & & H_k & \\ \vdots & \vdots & & & \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & \varrho_{k,k} \end{pmatrix} \begin{pmatrix} 0 \\ \vdots \\ R_k^{-1} \end{pmatrix} \\ &= Q_{k+1} \left(\frac{\hat{H}_k}{0 \dots 0 h_{k+1,k}} \right) \stackrel{\text{def}}{=} Q_{k+1} \tilde{H}_k. \end{aligned}$$

Setzen wir

$$Q_k = (q_0, \dots, q_{k-1})$$

so erhalten wir hieraus eine direkte Rekursion für die q_i , nämlich

$$Aq_{k-1} = \sum_{j=1}^{k+1} h_{j,k} q_{j-1}$$

also mit

$$\begin{aligned} h_{j,k} &\stackrel{\text{def}}{=} q_{j-1}^H Aq_{k-1} \quad j = 1, \dots, k \\ h_{k+1,k} &\stackrel{\text{def}}{=} \left\| \left(Aq_{k-1} - \sum_{j=1}^k h_{j,k} q_{j-1} \right) \right\| \\ q_k &= \left(Aq_{k-1} - \sum_{j=1}^k h_{j,k} q_{j-1} \right) / h_{k+1,k} . \end{aligned}$$

Diese Formel setzt natürlich $h_{k+1,k} \neq 0$ voraus. Dies ist nach obiger Herleitung genau dann verletzt, wenn $q_{k,k} = 0$, also wenn $A^k r_0$ von $r_0, \dots, A^{k-1} r_0$ linear abhängig ist. Dann ist

$$AQ_k = Q_k \hat{H}_k .$$

Mit dem Ansatz $x_k = x_0 + Q_k y_k$ wird wegen $AQ_k = Q_{k+1} \tilde{H}_k$ (s.o.)

$$\|AQ_k y_k + r_0\|_2 = \|Q_{k+1} \tilde{H}_k y_k + Q_{k+1} \|r_0\| e_1\|_2$$

und y_k errechnet sich somit aus der linearen Ausgleichsaufgabe

$$\|\tilde{H}_k y + \|r_0\| e_1\|_2 \stackrel{!}{=} \min_y$$

(man ergänze Q_{k+1} zu einem vollständigen Orthonormalsystem U in \mathbb{C}^n und multipliziere in $\|\cdot\|_2$ mit U^H).

Mit wachsendem k wächst \tilde{H}_k Schritt um Schritt und die Ausgleichsaufgabe der Dimension $(k+1) * k$ läßt sich wegen der Hessenberggestalt mit $\mathcal{O}(k^2)$ Operationen lösen. Gilt $h_{k+1,k} = 0$ zum erstenmal, dann haben wir

$$\|\hat{H}_k y + \|r_0\| e_1\|_2 = \min_y$$

zu lösen und wegen $\text{Rang}(\hat{H}_k) = k$ ist dies ein kompatibles System, die Lösung von $Ax = b$ damit durch x_k gegeben. Dieses Verfahren liefert also für invertierbares A immer nach spätestens n Schritten die Lösung des Gleichungssystems.

Der Aufwand in Schritt k ist $\mathcal{O}(nk)$ (i.w. für die Skalarprodukte $h_{j,k}$) und eine Matrixvektormultiplikation mit A . Man ist in der Praxis natürlich auch nicht daran interessiert, viele Schritte zu machen, zumal man alle q 's benötigt, um x zu berechnen, und dies sind

6.5. DAS VERFAHREN DER GENERALISIERTEN MINIMALEN RESIDUEN GMRES107

in der Regel voll besetzte Vektoren. Fehleraussagen für GMRES sind noch immer Gegenstand aktiver Forschung. Ein einfaches Resultat sei hier angegeben:

Satz 6.5.1 *Sei A invertierbar und diagonalähnlich:*

$$T^{-1}AT = \Lambda = \text{diag} (\lambda_1, \dots, \lambda_n).$$

Dann gilt

$$\|Ax_k - b\|_2 \leq \text{cond}_{\|\cdot\|_2}(T) \left(\min_{\substack{p \in \Pi_k \\ p(0)=1}} \max_{i=1, \dots, n} |p(\lambda_i)| \right) \|Ax_0 - b\|_2.$$

□

Ist A normal, (d.h. $AA^H = A^H A$), dann kann man $\text{cond}_{\|\cdot\|_2}(T) = 1$ nehmen und die Fehleraussage entspricht i.w. der beim cg-Verfahren. In anderen Fällen kann $\text{cond}_{\|\cdot\|_2}(T) \gg 1$ ausfallen. Der zweite Term hängt entscheidend von der Eigenwertverteilung ab und für beliebige Verteilungen liefert der Satz überhaupt keine brauchbaren Aussagen. Liegen alle Eigenwerte relativ dicht beieinander, dann ist die Situation günstig. Dies unterstreicht die Notwendigkeit der Konstruktion geeigneter Prädiktionierer. Außerdem weiß man, daß das Konvergenzverhalten auch stark von x_0 abhängt, was sich in obigem Satz aber nicht wiederfindet. In der Praxis wird das Verfahren (mit einem geeigneten Prädiktionierer) so angewendet, daß nach $k_0 \ll n$ Schritten mit $x_0 := x_{k_0}$ neu gestartet wird, so daß nur k_0 Vektoren q_i benötigt werden. Es gibt hier jedoch das Problem, daß u.U. Stagnation eintritt, d.h. der Fehler nimmt schliesslich nicht mehr ab.

Es gibt viele weitere Modifikationen des cg-Verfahrens für nicht positiv definite Systeme, von denen aber keines durchweg gute Resultate liefert. Dies ist ein Gebiet aktiver Forschung.

Beispiel 6.5.1 *Das folgende Beispiel demonstriert die Schwierigkeiten, auf die man bei allgemeinen Matrizen mit dem GMRES-Verfahren stossen kann. Es wird eine unsymmetrische Matrix der Dimension 100 mit zufälligen Einträgen aber vorgegebenen Singulärwerten generiert (eine Matrix vom Typ "randcolu" aus der Matrixgalerie von Higham, die man z.B. in MATLAB findet). Die Singulärwerte sind*

$$10 \frac{\sigma_i}{\sum_{j=1}^{100} \sigma_j^2} \quad \text{mit } \sigma_j = \exp((j-1)/9.9)$$

Diese Matrix hat die Konditionszahl $2.2026 \cdot 10^4$ und eine Eigenwertverteilung gemäß der folgenden Abbildung (Eigenwerte sind die Strahlenenden)

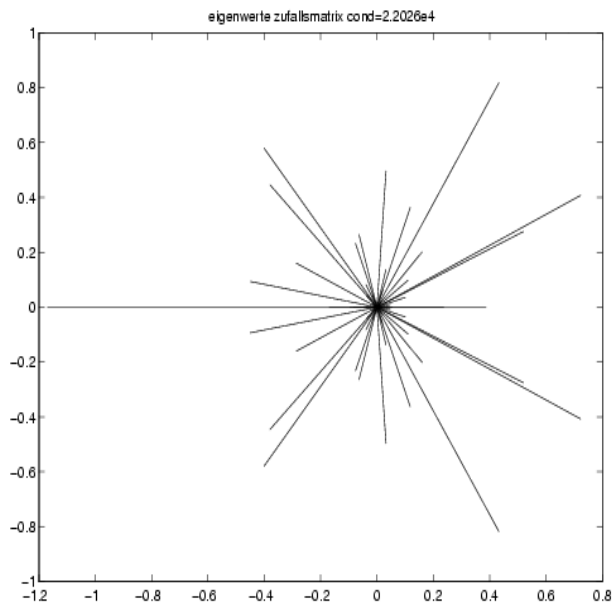


Abbildung 6.5.1

Wir wählen $x^* = (1, \dots, 1)^T$ und berechnen $b = Ax^*$. Es wird GMRES mit Restart nach jeweils 10 Schritten angewendet. Hier tritt Stagnation nach 20 Schritten auf, das durch $\|b\|_2$ skalierte Residuum verbleibt in der Grössenordnung von 0.7.

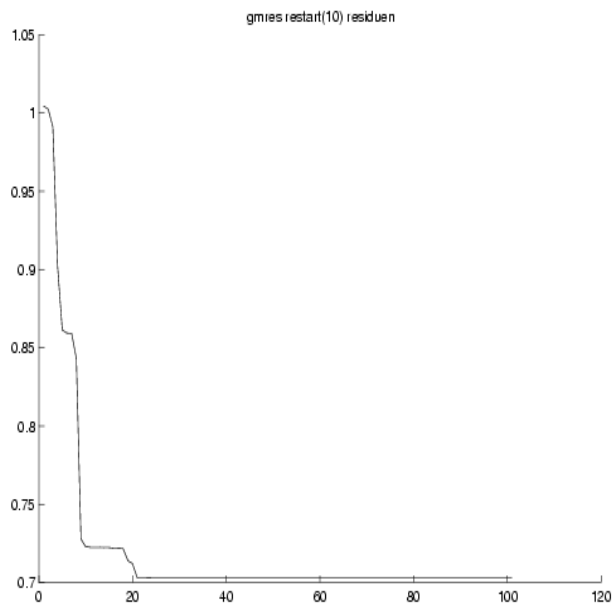


Abbildung 6.5.2

6.6 Die Methode von Kaczmarz zur iterativen Lösung von linearen Gleichungssystemen

Während die in 6.1-6.5 besprochenen Verfahren nur für jeweils spezielle Matrizen anwendbar waren, ist das folgende für beliebige, sogar singuläre und nichtquadratische Matrizen definiert. Seine Konvergenzgeschwindigkeit ist allerdings sehr gering.

Das Verfahren beruht auf der Idee, die Lösung x von $Ax = b$ für b im Bildraum von A zu charakterisieren als

$$x \in \bigcap_{i=1}^m H_i, \quad H_i = \{y \in \mathbb{C}^n : \tilde{a}_i^H y - \beta_i = 0\} \quad A \in \mathbb{C}^{m \times n},$$

d.h. H_i ist die Menge aller Punkte (Hyperebene), die die i -te Gleichung des Systems erfüllen. Dabei haben wir die Zeilen von A mit \tilde{a}_i^H bezeichnet, d.h.

$$A = \begin{pmatrix} \tilde{a}_1^H \\ \vdots \\ \tilde{a}_m^H \end{pmatrix} \quad A^H = (\tilde{a}_1, \dots, \tilde{a}_m).$$

Der Fehler, den das Einsetzen von x in die i -te Gleichung ergibt, ist also

$$\tilde{a}_i^H x - \beta_i$$

und es gilt

$$A^H e_i = \tilde{a}_i.$$

Von einem beliebigen x^0 ausgehend wird nun versucht, durch orthogonale Projektion von x^k auf eine der H_i den Fehler in x^k zu verkleinern. In der Originalmethode von Kaczmarz (optimales Kaczmarzverfahren) wird als $i = i(k)$ diejenige Hyperebene ausgewählt, die den größten Einsetzfehler ergibt:

$$|\tilde{a}_{i(k)}^H x - \beta_{i(k)}| = \max_{1 \leq j \leq m} |\tilde{a}_j^H x - \beta_j|.$$

Dies erfordert allerdings $\sum_{i=1}^m z_i$ Multiplikationen, wo z_i die Anzahl der Elemente $\neq 0$ in Zeile i von A ist, bei voll besetzter Matrix also mn .

x^{k+1} wird dann als orthogonale Projektion von x^k auf $H_{i(k)}$ berechnet, d.h.

$$x^{k+1} - x^k = \alpha \tilde{a}_{i(k)}$$

mit geeignetem α und

$$\tilde{a}_{i(k)}^H x^{k+1} - \beta_{i(k)} = 0$$

ergibt

$$\alpha = -(\tilde{a}_{i(k)}^H x^k - \beta_{i(k)}) / \|\tilde{a}_{i(k)}\|_2^2,$$

d.h.

$$x^{k+1} = x^k - \frac{e_{i(k)}^T (Ax^k - b)}{\|A^H e_{i(k)}\|_2^2} A^H e_{i(k)}.$$

Die Methode konvergiert sogar für singuläres und nichtquadratisches A gegen eine Lösung, und zwar linear. Sei dazu definiert

$$\begin{aligned} N(A) &\stackrel{def}{=} \{w \in \mathbb{C}^n : Aw = 0\} \quad \text{Nullraum von } A \\ N^\perp(A) &\stackrel{def}{=} \{w \in \mathbb{C}^n : w^T z = 0 \quad \forall z \in N(A)\} \\ \delta &\stackrel{def}{=} \min_{\substack{w \in N^\perp(A) \\ \|w\|_2 = 1}} \|Aw\|_2 \quad (> 0) \end{aligned}$$

und wie vorausgesetzt

$$b \in R(A),$$

$$R(A) = \{z : \exists x \in \mathbb{C}^n : z = Ax\}.$$

Sei

$$\hat{u} \in N^\perp(A) \quad \text{mit} \quad A\hat{u} = b.$$

(\hat{u} ist eindeutig bestimmt!)

Satz 6.6.1 Bezeichne $P_{N(A)}$ den orthogonalen Projektor auf $N(A)$ (in $\|\cdot\|_2$) und

$$x \stackrel{\text{def}}{=} \hat{u} + P_{N(A)}x^0.$$

Dann konvergiert das optimale Kaczmarz-Verfahren gegen x mit

$$\|x^{k+1} - x\|_2 \leq \left(1 - \frac{\delta^2}{n\gamma^2}\right)^{\frac{1}{2}} \|x^k - x\|_2$$

$$\gamma = \max_i \|\tilde{a}_i\|_2$$

□

<<

Beweis:

$$\|x^{k+1} - x\|_2^2 = \|x^k - x\|_2^2 - 2\Re \left(\frac{(e_{i(k)}^T (Ax^k - b))(x^k - x)^H A^H e_{i(k)}}{\|A^H e_{i(k)}\|_2^2} \right)$$

$$+ \frac{|e_{i(k)}^T (Ax^k - b)|^2 e_{i(k)}^T A A^H e_{i(k)}}{\|A^H e_{i(k)}\|_2^4}.$$

Es ist aber

$$e_{i(k)}^T A A^H e_{i(k)} = \|A^H e_{i(k)}\|_2^2$$

und mit

$$b = Ax$$

$$e_{i(k)}^T (Ax^k - b)(x^k - x)^H A^H e_{i(k)} = (x^k - x)^H A^H e_{i(k)} e_{i(k)}^T A (x^k - x) \in \mathbb{R}_+.$$

Somit

$$\|x^{k+1} - x\|_2^2 = \|x^k - x\|_2^2 - \frac{|e_{i(k)}^T (Ax^k - b)|^2}{\|A^H e_{i(k)}\|_2^2}.$$

Nach Definition von $i(k)$ ist

$$|e_{i(k)}^T (Ax^k - b)|^2 = \|Ax^k - b\|_\infty^2 \geq \frac{1}{n} \|Ax^k - b\|_2^2$$

somit

$$\begin{aligned} \|x^{k+1} - x\|_2^2 &\leq \|x^k - x\|_2^2 - \frac{1}{n} \cdot \frac{\|Ax^k - b\|_2^2}{\|A^H e_{i(k)}\|_2^2} \\ &\leq \|x^k - x\|_2^2 - \frac{1}{n\gamma^2} \|Ax^k - b\|_2^2. \end{aligned}$$

Nun gilt

$$(P_{N(A)} A^H e_i)^H P_{N(A)} A^H e_i = e_i^H A P_{N(A)}^2 A^H e_i = e_i^H A P_{N(A)} A^H e_i = 0.$$

Also

$$P_{N(A)} A^H e_i = 0 \quad i = 1, \dots, n,$$

somit

$$P_{N(A)} x^{k+1} = P_{N(A)} x^k \quad \forall k.$$

Somit

$$x^k - x = x^k - \hat{u} - P_{N(A)} x^0 = x^k - \hat{u} - P_{N(A)} x^k \in N^\perp(A)$$

d.h.

$$\begin{aligned} \|x^{k+1} - x\|_2^2 &\leq \left(1 - \frac{1}{n\gamma^2} \frac{\|A(x^k - x)\|_2^2}{\|x^k - x\|_2^2}\right) \|x^k - x\|_2^2 \\ &\leq \left(1 - \frac{\delta^2}{n\gamma^2}\right) \|x^k - x\|_2^2 \end{aligned}$$

□

>>

Im regulären Fall $m = n$, $\det(A) \neq 0$ ist

$$\frac{\delta}{\gamma} \approx \frac{1}{\text{cond}_{\|\cdot\|_2}(A)},$$

die Konvergenzgeschwindigkeit ist also bei schlechter Kondition außerordentlich langsam, weshalb man dieses Verfahren nur anwendet, wenn keine der in den anderen Iterationsverfahren benutzten Struktureigenschaften gegeben ist. Der Hauptaufwand eines Iterationsschrittes liegt hier in der Bestimmung von $i(k)$. Aber auch das *zyklische Iterationsverfahren* mit

$$i(k) = k \quad \forall k$$

ist konvergent aufgrund folgenden Satzes

Satz 6.6.2 Seien H_i , $i = 1, \dots, r$, Unterräume von \mathbb{C}^n und P_i die orthogonalen Projektionen von \mathbb{C}^n auf H_i . Ferner sei

$$D \stackrel{\text{def}}{=} \bigcap_{i=1}^r H_i \neq \emptyset.$$

$x^0 \in \mathbb{C}^n$ sei beliebig und

$$x^{k+1} = x^k + \lambda_k (P_{i(k)} x^k - x^k) \quad k = 0, 1, 2, \dots,$$

wo

$$\begin{aligned} 0 < \varepsilon_1 &\leq \lambda_k \leq \varepsilon_2 < 2 & \forall k \\ i(k) &= (k \bmod r) + 1. \end{aligned}$$

Dann gilt

$$\exists \lim_{k \rightarrow \infty} x^k = x^* \in D.$$

Dieser Satz ist ein Spezialfall des Satzes von Bregman, L.M.: Finding the common point of convex sets by the method of successive projection. Dokl. Akad. Nauk. SSSR 162(3), 487–490, (1965). \square

Über die Konvergenzgeschwindigkeit liegen für den Algorithmus mit zyklischen Projektionen allerdings keine Aussagen vor. Verbesserungen bieten sich an z.B. durch simultane Projektion auf den Durchschnitt mehrerer H_i u.s.w.

Beispiel 6.6.1 Gegeben sei das lineare Gleichungssystem $Ax = b$ mit

$$A = \begin{pmatrix} 0 & 2 \\ 1 & -1 \end{pmatrix} \quad \text{und} \quad b = \begin{pmatrix} 4 \\ 1 \end{pmatrix}.$$

Wir führen mit dem Startvektor $x^{(0)} = (-1, 0)^T$ vier Schritte des Verfahrens von Kaczmarz aus.

$k = 0 :$

$$Ax^0 - b = \begin{pmatrix} -4 \\ -2 \end{pmatrix} \Rightarrow i(0) = 1$$

und

$$A^T e_{i(0)} = \begin{pmatrix} 0 \\ 2 \end{pmatrix} \Rightarrow \|A^T e_{i(0)}\|_2^2 = 4.$$

Es ergibt sich

$$x^1 = \begin{pmatrix} -1 \\ 0 \end{pmatrix} - \frac{-4}{4} \begin{pmatrix} 0 \\ 2 \end{pmatrix} = \begin{pmatrix} -1 \\ 2 \end{pmatrix}.$$

$k = 1 :$

$$Ax^1 - b = \begin{pmatrix} 0 \\ -4 \end{pmatrix} \Rightarrow i(1) = 2$$

und

$$A^T e_{i(1)} = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \Rightarrow \|A^T e_{i(1)}\|_2^2 = 2.$$

Es ergibt sich

$$x^1 = \begin{pmatrix} -1 \\ 2 \end{pmatrix} - \frac{-4}{2} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

$k = 2$:

$$Ax^2 - b = \begin{pmatrix} -4 \\ 0 \end{pmatrix} \Rightarrow i(2) = 1$$

und

$$A^T e_{i(2)} = \begin{pmatrix} 0 \\ 2 \end{pmatrix} \Rightarrow \|A^T e_{i(2)}\|_2^2 = 4.$$

Es ergibt sich

$$x^3 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} - \frac{-4}{4} \begin{pmatrix} 0 \\ 2 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}.$$

$k = 3$:

$$Ax^3 - b = \begin{pmatrix} 0 \\ -2 \end{pmatrix} \Rightarrow i(3) = 2$$

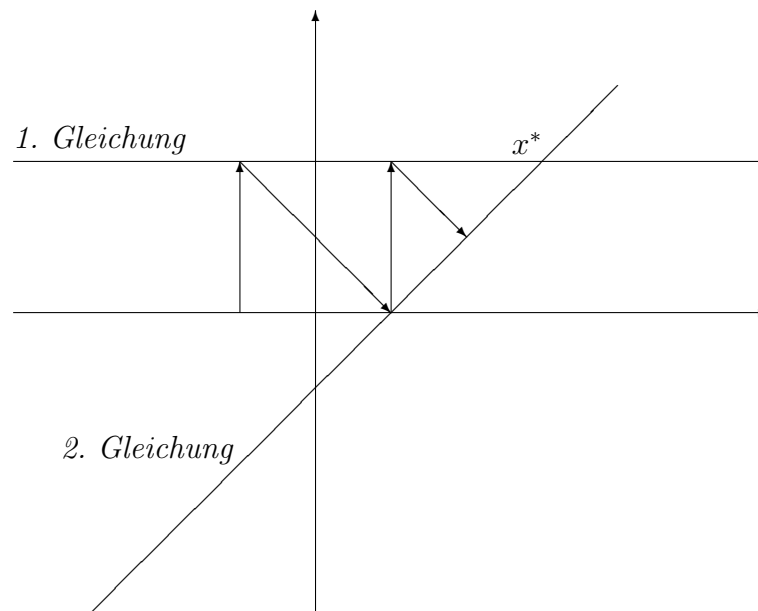
und

$$A^T e_{i(3)} = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \Rightarrow \|A^T e_{i(3)}\|_2^2 = 2.$$

Es ergibt sich

$$x^4 = \begin{pmatrix} 1 \\ 2 \end{pmatrix} - \frac{-2}{2} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}.$$

Die graphische Darstellung ergibt



6.7 Ein spezielles Verfahren für große nichtlineare dünnbesetzte Systeme

Bei der Diskretisierung nichtlinearer Differentialgleichungen entstehen nichtlineare Gleichungssysteme, zu deren Lösung das Newtonverfahren in Kombination mit einem direkten linearen Gleichungslöser aus Aufwandsgründen ebenso ungeeignet ist wie der

Gauß'sche Algorithmus für den entsprechenden linearen Fall. Eine Kombination der Ideen, die zur Konstruktion von Iterationsverfahren für lineare Gleichungssysteme führten, mit den Methoden zur Lösung skalarer oder niedrig dimensionaler nichtlinearer Gleichungen führt hier zum Erfolg. Von den zahlreichen möglichen Varianten (vgl. Ortega & Rheinboldt: Iterative solution of nonlinear equations in several variables, Acad. Press 1970) sei hier nur das SOR–Newton–Verfahren besprochen, das sich in der Praxis besonders bewährt hat. Die wesentliche Idee des SOR–Verfahrens für lineare Gleichungssysteme war die folgende: Ist $(\xi_{1,k+1}, \dots, \xi_{i-1,k+1}, \xi_{i,k}, \dots, \xi_{n,k})^T$ die laufende Näherung für x^* im i^{ten} Teilschritt des k^{ten} Iterationsschrittes, $i \in \{1, \dots, n\}$, dann ist $\hat{\xi}_{i,k+1}$ dadurch definiert, daß die i^{te} Gleichung bei sonst festgehaltenen Variablen durch Wahl der i^{ten} Variablen exakt erfüllt wurde. Dabei ist

$$\hat{y}_{kn+i} := (\xi_{1,k+1}, \dots, \xi_{i-1,k+1}, \hat{\xi}_{i,k+1}, \xi_{i+1,k}, \dots, \xi_{n,k})^T \quad \text{so, daß} \quad e_i^T (A\hat{y}_{kn+i} - b) = 0.$$

Schließlich wurde

$$\xi_{i,k+1} = \omega \hat{\xi}_{i,k+1} + (1 - \omega) \xi_{i,k}$$

gesetzt. Die Übertragung auf den nichtlinearen Fall geschieht dadurch, daß man $\hat{\xi}_{i,k+1}$ aus einem Schritt des Newtonverfahrens für die i^{te} Gleichung (nur in der i^{ten} Variablen) definiert:

$$\begin{aligned} \hat{\xi}_{i,k+1} &= \xi_{i,k} - \frac{f_i(\xi_{1,k+1}, \dots, \xi_{i-1,k+1}, \xi_{i,k}, \dots, \xi_{n,k})}{\frac{\partial}{\partial \xi_i} f_i(\xi_{1,k+1}, \dots, \xi_{i-1,k+1}, \xi_{i,k}, \dots, \xi_{n,k})} \\ \xi_{i,k+1} &= \omega \hat{\xi}_{i,k+1} + (1 - \omega) \xi_{i,k} \end{aligned}$$

Mit den Bezeichnungen

$$F = (f_1, \dots, f_n)^T, \quad y_{kn+i} = (\xi_{1,k+1}, \dots, \xi_{i,k+1}, \xi_{i+1,k}, \dots, \xi_{n,k})^T$$

ergibt dies

$$\begin{aligned} y_{kn+i} &= y_{kn+i-1} - \omega \frac{e_i^T F(y_{kn+i-1})}{e_i^T \mathcal{J}_F(y_{kn+i-1}) e_i} e_i & i &= 1, \dots, n \\ y_{kn} &\hat{=} x_k & k &= 0, 1, 2, \dots \end{aligned}$$

Für lineares F ist dies gerade das gewöhnliche SOR–Verfahren.

Wir beweisen für dieses Verfahren sowohl einen lokalen als auch einen globalen Konvergenzsatz:

Satz 6.7.1 Sei $\mathcal{D} \subset \mathbb{R}^n$ konvex, $F \in C^2(\mathcal{D})$, $x^* \in \mathcal{D}$, $F(x^*) = 0$. Ferner sei

$$\mathcal{J}_F(x^*) = -L^* + D^* - U^*$$

die Zerlegung in striktes unteres Dreieck, Diagonaleil und striktes oberes Dreieck. Falls D^* regulär ist und

$$\rho((D^* - \omega L^*)^{-1}((1 - \omega)D^* + \omega U^*)) < 1$$

(d.h. das SOR-Verfahren ist für den gegebenen ω -Wert bei einem linearen Gleichungssystem mit $\mathcal{J}_F(x^*)$ als Koeffizientenmatrix konvergent), dann konvergiert das SOR-Newton-Verfahren für hinreichend gute Startwerte. \square

<<

Beweis: Anwendung des Satzes von Ostrowski:

Zu zeigen ist $\rho(\mathcal{J}_\Phi(x^*)) < 1$. Die Iterationsfunktion Φ des SOR-Verfahrens wird implizit beschrieben durch

$$\begin{aligned} \varphi_1(x) &= \xi_1 - \omega \frac{f_1(\xi_1, \dots, \xi_n)}{\frac{\partial}{\partial \xi_1} f_1(\xi_1, \dots, \xi_n)} \\ \varphi_i(x) &= \xi_i - \omega \frac{f_i(\varphi_1(x), \dots, \varphi_{i-1}(x), \xi_i, \dots, \xi_n)}{\frac{\partial}{\partial \xi_i} f_i(\varphi_1(x), \dots, \varphi_{i-1}(x), \xi_i, \dots, \xi_n)} \\ & \quad i = 2, \dots, n \end{aligned}$$

Ist x^* Nullstelle von F , dann ersichtlich $y^* = \Phi(x^*) = x^*$. Erklärt man also die Funktion $G : \mathcal{D} \times \mathcal{D} \rightarrow \mathbb{R}^n$ durch

$$\begin{aligned} g_1(x, y) &= -\eta_1 + \xi_1 - \omega \frac{f_1(\xi_1, \dots, \xi_n)}{\frac{\partial}{\partial \xi_1} f_1(\xi_1, \dots, \xi_n)} \\ g_i(x, y) &= -\eta_i + \xi_i - \omega \frac{f_i(\eta_1, \dots, \eta_{i-1}, \xi_i, \dots, \xi_n)}{\frac{\partial}{\partial \xi_i} f_i(\eta_1, \dots, \eta_{i-1}, \xi_i, \dots, \xi_n)} \\ & \quad i = 2, \dots, n \end{aligned}$$

(mit $y = (\eta_1, \dots, \eta_n)^T$ und $x = (\xi_1, \dots, \xi_n)^T$) dann liefert der Hauptsatz über implizite Funktionen³, daß durch die Gleichung $G(x, y) = 0$ für $x \in U(x^*)$, $y \in V(x^*)$ y eindeutig erklärt ist als eine differenzierbare Funktion Φ von x , mit $x^* = \Phi(x^*)$ d.h. Φ ist durch (6.10) lokal wohldefiniert und die Iterationsfunktion Φ ist differenzierbar.

³Beachte: $G \in C^1(\mathcal{D} \times \mathcal{D})$, $G(x^*, x^*) = 0$, $G_y(x^*, x^*) = \begin{pmatrix} -1 & 0 & & \\ & \ddots & & \\ & & & -1 \end{pmatrix}$ regulär.

Wir haben mit $\delta_{ij} = \begin{cases} 0 & i \neq j \\ 1 & \text{sonst} \end{cases}$

$$\begin{aligned} \frac{\partial \varphi_i(x)}{\partial \xi_j} &= \delta_{ij} - \omega \frac{1}{\frac{\partial}{\partial \xi_i} f_i(\varphi_1(x), \dots, \varphi_{i-1}(x), \xi_i, \dots, \xi_n)} \cdot \\ &\quad \cdot \left(\sum_{k=1}^{i-1} \frac{\partial}{\partial \xi_k} f_i(\dots) \frac{\partial \varphi_k}{\partial \xi_j}(x) + \sum_{k=i}^n \frac{\partial}{\partial \xi_k} f_i(\dots) \delta_{kj} \right) - \\ &\quad - \omega f_i(\dots) \left(\frac{\partial}{\partial \xi_j} \frac{1}{\frac{\partial}{\partial \xi_i} f_i(\dots)} \right) \quad i, j \in \{1, \dots, n\} \end{aligned}$$

Setzt man x^* ein, hält $j_0 \in \{1, \dots, n\}$ fest und multipliziert man die i^{te} Gleichung mit $\frac{\partial}{\partial \xi_i} f_i(x^*)$ durch, dann ergibt sich durch Zusammenfassung der n Gleichungen (i läuft von 1 bis n)

$$\begin{aligned} (-\omega L + D^*) \underbrace{\begin{pmatrix} \frac{\partial \varphi_1}{\partial \xi_{j_0}}(x^*) \\ \vdots \\ \frac{\partial \varphi_n}{\partial \xi_{j_0}}(x^*) \end{pmatrix}}_{\mathcal{J}_{\Phi}(x^*) e_{j_0}} &= \begin{pmatrix} O \\ \vdots \\ \frac{\partial}{\partial \xi_{j_0}} f_{j_0}(x^*) \\ \vdots \\ O \end{pmatrix} - \omega \begin{pmatrix} \sum_{k=1}^n \frac{\partial}{\partial \xi_k} f_1(x^*) \delta_{k,j_0} \\ \vdots \\ \sum_{k=n}^n \frac{\partial}{\partial \xi_k} f_n(x^*) \delta_{k,j_0} \end{pmatrix} \\ &= \begin{pmatrix} O \\ \vdots \\ \frac{\partial}{\partial \xi_{j_0}} f_{j_0}(x^*) \\ \vdots \\ O \end{pmatrix} - \omega \begin{pmatrix} \frac{\partial}{\partial \xi_{j_0}} f_1(x^*) \\ \vdots \\ \frac{\partial}{\partial \xi_{j_0}} f_{j_0}(x^*) \\ \vdots \\ O \end{pmatrix} \\ &= ((1 - \omega)D^* + \omega U^*) e_{j_0} \end{aligned}$$

und weil j_0 eine beliebige Spaltennummer war

$$\mathcal{J}_{\Phi}(x^*) = (-\omega L^* + D^*)^{-1} ((1 - \omega)D^* + \omega U^*).$$

□

Lokal kann also ω gemäß der SOR–Theorie für den linearen Fall wählen, aber dies besagt natürlich nichts über die zulässigen ω -Werte, wenn man von der wahren Lösung noch weit entfernt ist. In einem für die Anwendungen wichtigen Spezialfall kann man jedoch die globale Konvergenz des SOR–Newton–Verfahrens beweisen. Dieser Fall (und der zugehörige Konvergenzbeweis) stellt die natürliche Verallgemeinerung des in Satz 6.2.10 betrachteten linearen Falles dar: (SOR-Verfahren für A symmetrisch positiv definit)

Satz 6.7.2 Sei $\mathcal{D} \subset \mathbb{R}^n$ konvex und offen,
 $f_0 : \mathcal{D} \rightarrow \mathbb{R}$ sei zweimal stetig differenzierbar auf \mathcal{D} , ein Niveaubereich

$$\mathcal{L}_0 \stackrel{\text{def}}{=} \{x \in \mathcal{D} : f_0(x) \leq f_0(z_0)\}$$

für ein $z_0 \in \mathcal{D}$ sei kompakt und die Eigenwerte der Hessematrix von f_0 dort gleichmässig nach unten beschränkt durch eine Konstante $\mu_1 > 0$. (Eine solche Funktion heißt gleichmässig konvex auf \mathcal{L}_0 .)

Dann gilt: $\exists \omega_0 > 0 : \forall \omega \in]0, \omega_0[$
 $\exists \lim_{k \rightarrow \infty} x_k = x^* \in \mathcal{L}_0$, wo $F(x^*) = 0$ mit

$$F(x) := \nabla f_0(x) \quad (\text{d.h. } \mathcal{J}_F(x) = \nabla^2 f_0(x))$$

und $\{x_k\}$ nach dem SOR-Newton-Verfahren bestimmt ist, falls nur $x_0 \in \mathcal{L}_0$. Auf die Kenntnis von ω_0 kann man verzichten, wenn man das Verfahren in folgender Weise modifiziert:

$$y_{kn+i} = y_{kn+i-1} - \omega 2^{-j} \frac{e_i^T F(y_{kn+i-1})}{e_i^T \mathcal{J}_F(y_{kn+i-1})} e_i, \quad j = j(i, k)$$

wo j definiert wird durch die Forderung $j \in \mathbb{N}_0$ **minimal**, so daß $y_{kn+i} \in \mathcal{D}$ und

$$(A) \quad f_0(y_{kn+i-1}) - f_0(y_{kn+i}) \geq 2^{-j-2} \omega \frac{(e_i^T F(y_{kn+i-1}))^2}{e_i^T \mathcal{J}_F(y_{kn+i-1})} e_i$$

mit $0 < \omega < 2$

□

<<

Beweis: Wir beweisen zuerst die zweite Behauptung und zeigen dabei, daß der Term 2^{-j} unabhängig von k, i und x_0 nach unten beschränkt ist durch eine Konstante > 0 . Wählt man dann ω_0 gleich dieser Konstanten, dann ist der erste Teil mitbewiesen. Setze mit $y_{kn+i-1} \in \mathcal{L}_0$

$$\sigma_{k,i} := e_i^T F(y_{kn+i-1}) / e_i^T \mathcal{J}_F(y_{kn+i-1}) e_i$$

(Man beachte, daß der Nenner $\geq \mu_1 > 0$ bleibt nach Vor.)

Es gilt dann

$$0 \leq |\sigma_{k,i}| \leq |e_i^T F(y_{kn+i-1})| / \mu_1 \leq \max_{x \in \mathcal{L}_0} \|F(x)\|_\infty / \mu_1 =: C_1$$

Da \mathcal{L}_0 beschränkt und abgeschlossen ist, $\subset \mathcal{D}$ nach Vor. und \mathcal{D} offen, existiert $\delta > 0$, so daß mit $x \in \mathcal{L}_0$ auch noch $x \pm \tau e_i \in \mathcal{D} \quad \forall \tau \in [0, \delta], \forall i$.

Also wird jedenfalls $y_{kn+i} \in \mathcal{D}$, falls $2^{-j} C_1 \leq \delta$. Sei j entsprechend gewählt. Dann gilt nach der Taylorformel

$$f_0(y_{kn+i-1}) - f_0(y_{kn+i-1} - \omega \sigma_{i,k} 2^{-j} e_i) =$$

$$= \omega \sigma_{i,k} 2^{-j} e_i^T \nabla f_0(y_{kn+i-1}) - \frac{1}{2} \omega^2 \sigma_{i,k}^2 2^{-2j} e_i^T \nabla^2 f_0(\tilde{y}_{kn+i-1}) e_i$$

wobei \tilde{y}_{kn+i-1} eine Zwischenstelle bezeichnet. Setzt man

$$M_2 \stackrel{\text{def}}{=} \max \left\{ e_i^T \nabla^2 f(z) e_i \quad : \quad \|z - x\|_\infty \leq \delta \quad \text{für ein } \begin{array}{l} x \in \mathcal{L}_0, \\ \forall i \in \{1, \dots, n\} \end{array} \right\}$$

so kann man abschätzen (wegen $\omega^2 < 2\omega$)

$$f_0(y_{kn+i-1}) - f_0(y_{kn+i-1} - \omega \sigma_{i,k} 2^{-j} e_i) \geq \omega 2^{-j} \frac{(e_i^T \nabla f_0(y_{kn+i-1}))^2}{e_i^T \nabla^2 f_0(y_{kn+i-1}) e_i} (1 - 2^{-j} \frac{M_2}{\mu_1})$$

Es gilt aber $1 - 2^{-j} \frac{M_2}{\mu_1} \geq 2^{-2}$ falls $2^{-j} \leq \frac{3}{4} \frac{\mu_1}{M_2}$, d.h. für den Term 2^{-j} gilt die globale untere Schranke

$$2^{-j} \geq \min\{1, \delta/(2C_1), \frac{3}{8} \mu_1/M_2\} =: C_2$$

und damit ist gezeigt: y_{kn+i} ist wohldefiniert,

$$\begin{aligned} f_0(y_{kn+i-1}) - f_0(y_{kn+i}) &\geq \omega C_2 (e_i^T \nabla f_0(y_{kn+i-1}))^2 / (4M_2) \quad \forall i, k \\ &\geq \omega C_2 \sigma_{k,i}^2 \mu_1^2 / (4M_2) \end{aligned}$$

d.h. auch $y_{kn+i} \in \mathcal{L}_0$. Dabei war noch $0 < \omega < 2$ frei wählbar.

(Zum Beweis von Teil 1 des Satzes hat man lediglich $\omega_0 := C_2$ zu setzen und die Abschätzungen entsprechend durchzuführen.) Da die Folge $\{y_{kn+i}\}$ beschränkt bleibt, besitzt jede Teilfolge einen Häufungswert. Wir zeigen nun noch, daß nur ein Häufungswert existiert und daß in diesem Häufungswert $\nabla f_0 = F$ verschwindet.

Aufgrund der Abstiegsabschätzung gilt $\sigma_{k,i} \rightarrow 0$ und da der Nenner in $\sigma_{k,i}$ gleichmäßig beschränkt ist durch M_2 bzw. μ_1 , gilt auch

$$\rho_{kn+i-1,i} := e_i^T \nabla f_0(y_{kn+i-1}) \xrightarrow[k \rightarrow \infty]{} 0, \quad i \in \{1, \dots, n\}$$

Sei im Folgenden

$$r_j := \nabla f_0(y_j), \quad \rho_{j,i} := e_i^T r_j$$

Wir zeigen zunächst

$$r_{k,n} \xrightarrow[k \rightarrow \infty]{} 0$$

Wegen $\sigma_{k,i} \rightarrow 0$ gilt

$$y_{kn+i} - y_{kn+i-1} \rightarrow 0 \quad \begin{array}{l} k \rightarrow \infty \\ i \in \{1, \dots, n\} \end{array} \quad (6.10)$$

und

$$\begin{aligned} \nabla f_0(y_{kn+i}) - \nabla f_0(y_{kn+i-1}) &= \\ &= -\omega 2^{-j} \sigma_{k,i} \int_0^1 \nabla^2 f(y_{kn+i-1} + t(y_{kn+i} - y_{kn+i-1})) dt e_i \rightarrow 0 \end{aligned}$$

d.h.

$$\rho_{j+1,i} - \rho_{j,i} \rightarrow 0 \quad j \rightarrow \infty, \quad i \in \{1, \dots, n\}$$

Wörtlich wie in Satz 6.2.10 folgt $r_{k,n} \rightarrow 0$ mit $k \rightarrow \infty$.

Für jeden Häufungspunkt x^* von $\{x_k\} = \{y_{kn}\}$ folgt aus der Stetigkeit von ∇f_0 , daß $\nabla f_0(x^*) = 0$.

Mit $\lim_{\substack{k \rightarrow \infty \\ k \in K}} x_k = x^*$ folgt auch $\lim_{\substack{k \rightarrow \infty \\ k \in K}} y_{kn+i-1} = x^*$, $i = 1, \dots, n$ wegen (6.10).

6.7. EIN SPEZIELLES VERFAHREN FÜR GROSE NICHTLINEARE DÜNNBESETZTE SYSTEME 119

Es bleibt zu zeigen, daß nur ein Häufungspunkt $x^* \in \mathcal{L}_0$ existieren kann.

Ann.: $\exists x^*, x^{**} \in \mathcal{L}_0, \quad x^* \neq x^{**}, \quad \nabla f(x^*) = \nabla f(x^{**}) = 0 \quad \Rightarrow$

$$\begin{aligned} 0 &= (x^* - x^{**})^T \underbrace{(\nabla f(x^*) - \nabla f(x^{**}))}_{=0} \\ &= (x^* - x^{**})^T \int_0^1 \nabla^2 f(x^{**} + t(x^* - x^{**})) dt (x^* - x^{**}) \\ &\geq \mu_1 \|x^* - x^{**}\|^2 > 0 \text{ Widerspruch!} \end{aligned}$$

□

>>

Bemerkung 6.7.1 Man beachte, daß zu gegebenem F mit $F : \mathcal{D} \rightarrow \mathbb{R}^n$,

\mathcal{D} konvex, beschränkt, $\mathcal{J}_F(x)$ stetig und symmetrisch und

$\lambda_i(\mathcal{J}_F(x)) \geq \mu_1 > 0 \quad \forall x \in \mathcal{D}, \quad \forall i \in \{1, \dots, n\}$, (λ_i Eigenwert von \mathcal{J}_F) und $F(x^*) = 0$ für ein $x^* \in \mathcal{D}$ stets ein f_0 (Stammfunktion) mit den oben geforderten Eigenschaften gehört. In der Praxis ist normalerweise f_0 das primär gegebene.

Die Aufgabenstellung ist dann die der Minimierung einer gleichmäßig konvexen Funktion $f_0 : \mathcal{D} \subset \mathbb{R}^n \rightarrow \mathbb{R}$. Das SOR-Verfahren wird nur gewählt, wenn $n \gg 1$ (etwa $n \geq 1000$) und wenn $\mathcal{J}_F = \nabla^2 f_0$ dünn besetzt ist)

Die Anwendung des Abstiegstests (A) erscheint dann unproblematisch. Tatsächlich wird dieser Test aber in der Praxis nie angewandt, und zwar aus Aufwandsgründen. Ein Teilschritt des SOR-Newton-Verfahrens erfordert nur die Berechnung **einer Komponente** von ∇f_0 und eines Diagonalelements von $\nabla^2 f_0$. Die Auswertung von (A) erfordert dagegen pro j -Wert eine f_0 -Auswertung, und der Aufwand hierfür ist normalerweise n -mal so groß! Da andererseits die globale Schranke ω_0 unbekannt ist, bzw. viel zu pessimistisch wäre (kleines ω bedeutet extrem langsame Konvergenz) behilft man sich in der Praxis so, daß man zuerst ω (abhängig von der Stärke der Nichtlinearität) "klein" wählt und sich bei der Annäherung an die Lösung dann im Rahmen der linearen SOR-Theorie vergrößern läßt. □

Bemerkung 6.7.2 Zur Minimierung einer hochdimensionalen gleichmäßig konvexen Funktion kann man auch mit gutem Erfolg ein präkonditioniertes cg-Verfahren benutzen. Gegenüber dem linearen Fall hat man nur minimale Änderungen vorzunehmen: nach je n Schritten (spätestens) erfolgt ein sogenannter "Restart" der Suchrichtung mit dem Gradienten und die Berechnung der exakten optimalen Schrittweite wird ersetzt durch eine angenäherte eindimensionale Minimierung der Funktion, wobei die Schrittweite einer Abstiegsforderung unterworfen wird wie im oben beschriebenen Fall des SOR-Newtonverfahrens:

$$f(x_k) - f(x_k - \sigma_k d_k) \geq \delta \sigma_k \nabla f(x_k)^T d_k$$

mit $0 < \delta < 0.5$. Dabei wählt man σ_k aus der Folge

$$\sigma = \frac{\nabla f(x_k)^T d_k}{2(f(x_k - d_k) - f(x_k) + \nabla f(x_k)^T d_k)} \cdot 2^{-j}, \quad j = 0, 1, \dots$$

Man kann zeigen, daß das so bestimmte σ_k gleichmässig nach unten beschränkt ist und daß die so bestimmte Folge $\{x_k\}$ n -Schritt quadratisch konvergiert, d.h.

$$\|x^{kn+n} - x^*\| \leq C \|x^{kn} - x^*\|^2$$

mit einer geeigneten Konstanten C . Für grosses n ist diese Aussage nicht sonderlich attraktiv. Aber in Verbindung mit einem guten Präkonditionierer erhält man tatsächlich schnelle Konvergenz.

6.8 Zusammenfassung. Weiterführende Literatur

Mit der Ausnahme des universell anwendbaren, dafür aber nur sehr langsam konvergenten Kaczmarc-Verfahrens sind alle üblichen Iterationsverfahren zur Lösung linearer Gleichungssysteme nur bei speziellen Matrizen anwendbar. Die einfachen Splittingverfahren werden in der Regel nur noch als Subprozesse in komplizierteren zusammengesetzten Verfahren benutzt (Multigrid-Verfahren, die im Rahmen von Diskretisierungsverfahren für partielle Differentialgleichungen besprochen werden). Das cg-Verfahren und GMRES (sowie die zahlreichen bekannten Varianten dieser Verfahren) können mit einem guten Präkonditionierer auch als eigenständige Löser dienen. Der Präkonditionierer muss allerdings problemabhängig gewählt werden, eine "black box" Methode dafür ist unbekannt. Allen Iterationsverfahren gemeinsam ist die starke Abhängigkeit von der Matrixkondition. In der Regel ist ihr Einsatz auch nur für sehr grosse und dünnbesetzte Matrizen sinnvoll, denn alle beinhalten die wiederholte Bildung von Matrix-Vektor-Produkten mit der Systemmatrix.

Weiterführende Literatur:

- Barrett, Richard; Berry, Michael; Chan, Tony F.; Demmel, James; Donato, June; Dongarra, Jack; Eijkhout, Victor; Pozo, Roldan; Romine, Charles; van der Vorst, Henk: *Templates for the solution of linear systems: building blocks for iterative methods*. Philadelphia, PA: SIAM, (1993).
- Greenbaum, Anne: *Iterative methods for solving linear systems* Frontiers in Applied Mathematics. 17. Philadelphia, PA: SIAM, (1997)
- Hageman, Louis A.; Young, David M.: *Applied iterative methods*. Unabridged republication of the 1981 original. Mineola, NY: Dover Publications. (2004)
- Saad, Yousef: *Iterative methods for sparse linear systems*. 2nd ed. Philadelphia, PA: SIAM (2003).
- van der Vorst, Henk A.: *Iterative Krylov methods for large linear systems*. Cambridge: Cambridge University Press. (2003).
- Varga, Richard S.: *Matrix iterative analysis. 2nd revised and expanded ed.* Berlin: Springer (2000).

Kapitel 7

Rundungsfehleranalyse numerischer Algorithmen, (ERG)

Wir haben bereits mehrfach feststellen können, daß ein Algorithmus im Falle des Rechnens mit endlicher, fester Stellenzahl, wie es den tatsächlichen Gegebenheiten in der Praxis entspricht, sich wesentlich von seinem abstrakten Ideal (in exakter reeller Rechnung) unterscheiden kann. In diesem Kapitel wollen wir kurz Methoden beschreiben, die es erlauben, den Effekt der Rundungsfehler mathematisch zu erfassen.

7.1 Zahldarstellung

Sei B eine natürliche Zahl ≥ 2 . Dann kann bekanntlich $x \in \mathbb{R}$ dargestellt werden in der Form

$$x = \pm \left(\sum_{i=1}^{\infty} \xi_i B^{-i} \right) B^e \quad \text{mit} \quad \xi_1 \neq 0 \text{ für } x \neq 0 \quad (7.1)$$

und

$$\left. \begin{array}{l} \xi_i \in \{0, \dots, B-1\} \\ e \in \mathbb{Z} \end{array} \right\} \begin{array}{l} \text{“Ziffer”} \\ \text{“Exponent”} \end{array} \text{ von } x$$

Für $x \neq 0$ und $\xi_1 \neq 0$ heißen ξ_1, \dots, ξ_t die ersten t signifikanten Ziffern von x . Diese Darstellung ist eindeutig bis auf das Auftreten einer unendlichen Folge von Ziffern $B-1$. Die im wissenschaftlich-technischen Rechnen verwendeten sogenannten *Gleitpunkt-Zahlen* erhält man aus der Darstellung (7.1) durch Einschränkung der Parameter-Zahl, d.h. es wird nun nur noch zugelassen

$$x = 0$$

bzw.

$$x = \pm \left(\sum_{i=1}^t \xi_i B^{-i} \right) B^e \quad \text{mit} \quad \xi_1 \neq 0 \text{ und} \\ e_{\min} \leq e \leq e_{\max}, \quad e \in \mathbb{Z}.$$

Auf vielen neueren Rechnern gilt:

$$B = 2, \quad t \in \{24, 53\}, \quad e_{\max} = 127, 1023, \quad e_{\min} = -e_{\max}$$

Intern wird die Zahl x dann als Tripel

$$(VZ, c, \xi_1 \dots \xi_t) = (\text{Vorzeichen, Charakteristik, Mantisse})$$

mit

$$c = e - e_{\min}$$

dargestellt und für negative Zahlen gilt eine Komplementdarstellung. c heisst ‘‘Charakteristik’’ und ist also stets positiv. Bei $B = 2$ kann man die Speicherung von $\xi_1 (= 1)$ für $x \neq 0$ noch einsparen. Dies ist in der sogenannten IEEE-Arithmetik der Fall. Dort wird allerdings die Mantissendarstellung als $\sum_{i=0}^{t-1} \xi_{i+1} 2^{-i}$ interpretiert. Das entspricht einer Verschiebung von e um 1.

Beispiel 7.1.1 $B = 16, t = 6, e_{\max} = 63, e_{\min} = -64$

$$\left. \begin{array}{l} \begin{array}{|c|c|c|c|c|c|c|} \hline 4 & 1 & 1 & F & 0 & 0 & 0 & 0 \\ \hline \end{array} \\ \underbrace{\hspace{1.5cm}} \quad | \quad \xi_1 \quad \xi_2 \quad \xi_3 \quad \xi_4 \quad \xi_5 \quad \xi_6 \\ \\ \begin{array}{|c|c|c|c|c|} \hline 0 & 100 & 0001 \\ \hline \end{array} \\ \downarrow \quad \quad \quad \underbrace{\hspace{1.5cm}}_{64+1} \quad \Rightarrow \quad e = 1 \\ + \end{array} \right\} x = \left(\frac{1}{16} + \frac{15}{256} \right) 16^1 = 1.9375$$

□

Die Menge der so darstellbaren Zahlen bezeichnen wir mit \mathbb{M} (=Maschinenzahlen).

Beliebige reelle Zahlen muß man beim praktischen Rechnen durch Maschinenzahlen approximieren. Diese Approximation nennt man *Rundung*.

Üblich sind *Rundung durch Abschneiden* (die Ziffern ξ_{t+1} und folgende werden vernachlässigt) und ‘‘symmetrische’’ *Rundung* (für B gerade)

$$x = \pm \left(\sum_{i=1}^{\infty} \xi_i B^{-i} \right) B^e \xrightarrow{\text{rd}} \tilde{x} = \pm \left(\sum_{i=1}^t \xi_i B^{-i} \right) B^e$$

$$x = \pm \left(\sum_{i=1}^{\infty} \xi_i B^{-i} \right) B^e \xrightarrow{\text{rd}} \tilde{x} = \begin{cases} \pm \left(\sum_{i=1}^t \xi_i B^{-i} \right) B^e & \text{falls } \xi_{t+1} < B/2 \\ \pm \left(\sum_{i=1}^t \xi_i B^{-i} + B^{-t} \right) B^e & \text{falls } \xi_{t+1} \geq B/2. \end{cases}$$

Dabei wurde für e vorausgesetzt, daß $e_{\max} \geq e \geq e_{\min}$.

Für $e < e_{\min}$ setzt man gewöhnlich $\tilde{x} = 0$ (‘‘Exponentenunterlauf’’), während für $e > e_{\max}$ \tilde{x} gewöhnlich undefiniert bleibt (‘‘Exponentenüberlauf’’) (+INF in IEEE).

Exponentenunterlauf ist nicht völlig unkritisch, wenn man z.B. bedenkt, daß dann $\exp(-x)$ für $x > 709$ gewöhnlich null gesetzt wird, während man sich theoretisch auf $\exp(-x) > 0$ ($\forall x$) berufen kann. Die Aussage des folgenden Satzes beweist man leicht durch übliche Abschätzungstechniken:

Satz 7.1.1 Falls bei der Abbildung $x \mapsto \tilde{x} := \text{rd}(x)$ weder Exponentenunter- noch -überlauf eintritt, dann gilt für $x \neq 0$

$$\left| \frac{x - \tilde{x}}{x} \right| \leq c \cdot B \cdot B^{-t} =: \varepsilon \quad \text{mit} \quad c = \begin{cases} \frac{1}{2} & \text{beim symmetrischen Runden} \\ 1 & \text{beim Abschneiden.} \end{cases}$$

□

Die in Satz 7.1.1 auftretende Größe

$$\frac{x - \tilde{x}}{x}$$

nennt man den *relativen Fehler* der Approximation \tilde{x} an x und die Aussage von Satz 7.1.1 ist also die, daß der relative Fehler der Rundung $\mathbb{R} \rightarrow \mathbb{M}$ normalerweise stets gleichmäßig klein ist.

Die auf der rechten Seite der Ungleichung definierten Größe ε nennt man die *relative Maschinengenauigkeit* der durch \mathbb{M} beschriebenen Zahldarstellung.

Um die Effekte von Exponentenüberlauf und -unterlauf nicht diskutieren zu müssen, machen wir im folgenden die *vereinfachende Annahme*

$$e_{\max} = -e_{\min} = \infty.$$

Die arithmetische Verknüpfung t -stelliger Zahlen mittels $+, -, *, /$ ergibt gewöhnlich reelle Zahlen, zu deren exakter Darstellung mehr als t Stellen benötigt werden. Bei der Realisierung der Arithmetik auf einem Rechner muß man deshalb geeignete "Ersatzoperationen" benutzen, bei denen das exakte Ergebnis in geeigneter Weise durch eine Maschinenzahl approximiert wird. Diese Ersatzoperationen symbolisieren wir durch

$$gl(\cdot \# \cdot) \quad \# \in \{+, -, *, /\}.$$

Eine naheliegende Forderung an die Konstruktion von gl ist

$$\forall x, y \in \mathbb{M} : \quad gl(x \# y) = \text{rd}(x \# y) \quad \# \in \{+, -, *, /\}$$

Diese Forderung hat die einschneidende Konsequenz, daß intern das Zwischenresultat $x \# y$ auf $t + 2$ Stellen exakt berechnet werden muß und ist deshalb bei den meisten Rechnern nur näherungsweise erfüllt.

Beispiel 7.1.2 $B = 10$, $t = 4$, symmetrische Rundung

$$x = +.1000 * 10^1, \quad y = -.9045 * 10^{-1}$$

$$\begin{array}{r} x + y = \quad .1000 | \quad *10^1 \\ \quad - .0090 | 45 \quad *10^1 \\ \hline \quad + .0909 | 55 \end{array}$$

$$\text{rd}(x + y) = \quad +.9096 \quad *10^0$$

$$x = .1236, \quad y = .3579$$

$$x * y = .0442|36|44$$

$$\text{rd}(x * y) = .4424 * 10^{-1}. \quad \square$$

Aus Satz 7.1.1 und der Modellvorstellung ergibt sich mit einer kleinen Zusatzbetrachtung

Satz 7.1.2 Falls kein Exponentenüber- oder -unterlauf eintritt, dann gilt unter der Modellvorstellung

$$\begin{aligned} \text{rd}(x) &= x(1 + \eta) = \frac{x}{1 + \eta'} & |\eta|, |\eta'| &\leq \varepsilon \\ g\ell(x \# y) &= (x \# y)(1 + \tilde{\eta}) = \frac{x \# y}{1 + \eta^*} & |\tilde{\eta}|, |\eta^*| &\leq \varepsilon \end{aligned}$$

$\varepsilon =$ relative Maschinengenauigkeit. □

Die Größen η, η' usw hängen natürlich von den Variablen x, y und der Operation $\#$ ab, sind also bei verschiedenen Operationen verschieden, haben jedoch betragsmäßig alle die gemeinsame Schranke ε .

Nach Satz 7.1.2 liegt jedes Einzelresultat der Maschinearithmetik nahe beim wahren reellen Resultat (im Sinne eines kleinen relativen Fehlers). Dennoch gehorcht die Maschinearithmetik nicht den üblichen arithmetischen Gesetzen, was schwerwiegende Folgen haben kann.

Beispiel 7.1.3

a) $g\ell(x + y) = x \not\approx y = 0 \quad B = 10, t = 4, \quad x = .2234 * 10^0, y = .4999 * 10^{-4}$

b) $g\ell(\underbrace{a + g\ell(b + c)}_I) \neq \underbrace{g\ell(g\ell(a + b) + c)}_{II}$ im allgemeinen

$$B = 10, t = 8, \quad a = .23371258 * 10^{-4}, \quad b = .33678429 * 10^2, \quad c = -.33677811 * 10^2$$

$$I = .64137126 * 10^{-3}, \quad II = .64100000 * 10^{-3}$$

c) $B = 10, t = 5, \quad a = .234165 * 10^0, \quad \text{rd}(a) = -.23417 * 10^0$
 $b = -.234094 * 10^0, \quad \text{rd}(b) = -.23409 * 10^0$
 $g\ell(\text{rd}(a) + \text{rd}(b)) = .80000 * 10^{-4} = \text{rd}(a) + \text{rd}(b)$ fehlerfrei
 aber: $a + b = .71000 * 10^{-4}$, d.h.

$$\left| \frac{g\ell(\text{rd}(a) + \text{rd}(b)) - (a + b)}{(a + b)} \right| = 0.1267 \quad \text{d.h.} \approx 13\% \text{ Fehler!}$$

□

Ein abschreckendes Beispiel für die Auswirkung von Rundungsfehlern ist die (durch Integraltafeln nahelegte) Rekursion zur Bestimmung von Integralen der Form

$$y_n \stackrel{\text{def}}{=} \int_0^1 x^n f(x) dx$$

hier mit $f(x) = \frac{1}{x+5}$. Es gilt in diesem Fall die Rekursion

$$y_n = -5y_{n-1} + \frac{1}{n}, \quad n = 1, \dots$$

mit $y_n \in [0, 0.2]$ monoton fallend in n und dem Anfangswert

$$y_0 = \ln(1.2).$$

Wendet man diese Rekursion mit IEEE-Arithmetik mit 53 Binärstellen an im „Vorwärtsmodus“, dann erhält man das katastrophale Resultat

```

Y( 0)= .1823215567939546D+00
Y( 1)= .8839221603022717D-01
Y( 2)= .5803891984886412D-01
Y( 3)= .4313873408901275D-01
Y( 4)= .3430632955493624D-01
Y( 5)= .2846835222531885D-01
Y( 6)= .2432490554007241D-01
Y( 7)= .2123261515678079D-01
Y( 8)= .1883692421609604D-01
Y( 9)= .1692649003063093D-01
Y(10)= .1536754984684535D-01
Y(11)= .1407134167486415D-01
Y(12)= .1297662495901257D-01
Y(13)= .1203995212801408D-01
Y(14)= .1122881078850103D-01
Y(15)= .1052261272416149D-01
Y(16)= .9886936379192523D-02
Y(17)= .9388847515802088D-02
Y(18)= .8611317976545116D-02
Y(19)= .9574989064642839D-02
Y(20)= .2125054676785809D-02
Y(21)= .3699377423511857D-01
Y(22)= -.1395143257210474D+00
Y(23)= .7410498894748021D+00
Y(24)= -.3663582780707344D+01
Y(25)= .1835791390353672D+02
Y(26)= -.9175110797922206D+02
Y(27)= .4587925769331473D+03
Y(28)= -.2293927170380022D+04
Y(29)= .1146967033465873D+05
Y(30)= -.5734831833996033D+05 \ .

```

Setzt man dagegen einfach

$$y_{31} = 0$$

und rechnet von da „rückwärts“ y_0 aus, so erhält man

```

Y(30)= .6451612903225807D-02
Y(29)= .5376344086021506D-02
Y(28)= .5821282906933630D-02
Y(27)= .5978600561470417D-02
Y(26)= .6211687295113324D-02
Y(25)= .6449970233285027D-02
Y(24)= .6710005953342995D-02
Y(23)= .6991332142664734D-02
Y(22)= .7297385745380097D-02
Y(21)= .7631431941833073D-02
Y(20)= .7997523135442909D-02
Y(19)= .8400495372911417D-02
Y(18)= .8846216714891401D-02
Y(17)= .9341867768132831D-02
Y(16)= .9896332328726375D-02

```

Y(15) = .1052073353425472D-01
 Y(14) = .1122918662648239D-01
 Y(13) = .1203987696041781D-01
 Y(12) = .1297663999253182D-01
 Y(11) = .1407133866816030D-01
 Y(10) = .1536755044818612D-01
 Y(9) = .1692648991036278D-01
 Y(8) = .1883692424014967D-01
 Y(7) = .2123261515197007D-01
 Y(6) = .2432490554103456D-01
 Y(5) = .2846835222512642D-01
 Y(4) = .3430632955497472D-01
 Y(3) = .4313873408900505D-01
 Y(2) = .5803891984886566D-01
 Y(1) = .8839221603022688D-01
 Y(0) = .1823215567939546D+00 \ ,

also y_0 in voller Genauigkeit.

7.2 Fehlerfortpflanzungsgesetze der elementaren arithmetischen Verknüpfungen

In diesem Abschnitt behandeln wir die Frage, wie sich in den Eingangsdaten einer arithmetischen Operation vorhandene Fehler auf die absolute und relative Genauigkeit des Resultats auswirken. \tilde{x} und \tilde{y} seien also "Näherungen" für x, y . Es seien $z := x \# y$, $\tilde{z} := \tilde{x} \# \tilde{y}$. Hier wird angenommen, daß die Rechenoperationen $\tilde{x} \# \tilde{y}$ exakt ausgeführt werden!

Sei $\tilde{x} = x + \Delta x$, $\tilde{y} = y + \Delta y$.

1. Addition/Subtraktion:

$$\tilde{x} \pm \tilde{y} = (x + \Delta x) \pm (y + \Delta y) = x \pm y + \Delta x \pm \Delta y = z + \Delta z$$

$$\Delta z = \Delta x \pm \Delta y \Rightarrow |\Delta z| \leq |\Delta x| + |\Delta y|$$

$$\frac{\Delta z}{z} = \frac{\Delta x}{x \pm y} \pm \frac{\Delta y}{x \pm y} = \frac{x}{x \pm y} \frac{\Delta x}{x} \pm \frac{y}{x \pm y} \frac{\Delta y}{y}$$

$$\underbrace{\left| \frac{\Delta z}{z} \right|}_{\text{relativer Fehler } z} \leq \underbrace{\frac{|x|}{|x \pm y|}}_{*} \underbrace{\left| \frac{\Delta x}{x} \right|}_{\text{relativer Fehler } x} + \underbrace{\frac{|y|}{|x \pm y|}}_{*} \underbrace{\left| \frac{\Delta y}{y} \right|}_{\text{relativer Fehler } y}$$

* Fehlerverstärkungs- bzw -dämpfungsfaktoren

Kritische Situation: $|x \pm y| \ll |x| + |y|$ "Auslöschung führender Ziffern"

Beispiel 7.2.1 (siehe Beispiel 7.1.3c)

Setze $x = a$, $\tilde{x} = \text{rd}(a)$, $y = b$, $\tilde{y} = \text{rd}(b)$

$$\left| \frac{\text{rd}(a) + \text{rd}(b) - (a + b)}{(a + b)} \right| \leq \frac{.234165}{.71 * 10^{-4}} * 5 * 10^{-5} + \frac{.234094}{.71 * 10^{-4}} * 5 * 10^{-5} = .321$$

Abschätzung realistisch!

□

Durch geeignete Umformung arithmetischer Ausdrücke kann man manchmal die Auswirkungen der ‐Auslöschung‐ vermeiden. Typisches Beispiel:

$$\sqrt{x^2 + 1} - x = \frac{1}{x + \sqrt{x^2 + 1}}$$

In der linken Formel wirken sich für $x \gg 1$ Fehler bei der Berechnung der Wurzel katastrophal aus, bei der rechten nicht.

2. Multiplikation:

$$\begin{aligned} \tilde{z} &= \tilde{x}\tilde{y} = (x + \Delta x)(y + \Delta y) = xy + x\Delta y + y\Delta x + \Delta x\Delta y = z + \Delta z \\ \frac{|\Delta z|}{|z|} &= \frac{|x\Delta y + y\Delta x + \Delta x\Delta y|}{|xy|} \\ &\leq \frac{|x\Delta y|}{|xy|} + \frac{|y\Delta x|}{|xy|} + \frac{|\Delta x\Delta y|}{|xy|} \\ &= \frac{|\Delta x|}{|x|} + \frac{|\Delta y|}{|y|} + \underbrace{\frac{|\Delta x|}{|x|} \frac{|\Delta y|}{|y|}}_{\text{normalerweise vernachlässigbar}} \end{aligned}$$

Bei der Multiplikation addieren sich die Schranken der relativen Fehler (bis auf Glieder 2. Ordnung)

Division: (Vor.: $\tilde{y} \neq 0, y \neq 0$)

$$\begin{aligned} \tilde{z} - z &= \frac{x + \Delta x}{y + \Delta y} - \frac{x}{y} = \frac{x}{\underbrace{y}_z} \frac{\left(1 + \frac{\Delta x}{x}\right)}{\left(1 + \frac{\Delta y}{y}\right)} - \frac{x}{\underbrace{y}_z} = z \left(\frac{1 + \frac{\Delta x}{x}}{1 + \frac{\Delta y}{y}} - \underbrace{1}_1 \right) \\ \Rightarrow \frac{|\tilde{z} - z|}{|z|} &= \frac{\left| \frac{\Delta x}{x} - \frac{\Delta y}{y} \right|}{\left| 1 + \frac{\Delta y}{y} \right|} \leq \left(\frac{|\Delta x|}{|x|} + \frac{|\Delta y|}{|y|} \right) \underbrace{\left(1 + \frac{|\Delta y|}{|y|} + \left(\frac{|\Delta y|}{|y|} \right)^2 + \dots \right)}_{\text{geometrische Reihe!}} \end{aligned}$$

Bei der Division addieren sich die Schranken der relativen Fehler (bis auf Glieder zweiter und höherer Ordnung in diesen Fehlern).

Division und Multiplikation sind also bezüglich der Fortpflanzung der relativen Fehler ‐harmlose‐ Operationen, nicht aber Addition und Subtraktion.

7.3 Vorwärtsanalyse der Rundungsfehlereffekte

Jeder numerische Algorithmus besteht aus einer endlichen Folge elementarer arithmetischer Verknüpfungen. Aus einem Vektor $x^{(0)}$ von Eingangsdaten entsteht über eine Folge von Zwischenergebnissen $x^{(i)}$ ein Vektor y von Ausgangsdaten. (Die einzelnen Vektoren können verschieden lang sein). In exakter reeller Arithmetik wird der Übergang $x^{(i)} \mapsto x^{(i+1)}$ beschrieben durch eine vektorwertige Funktion Φ_i , wobei jede Komponentenfunktion entweder die identische Abbildung einer Komponente von $x^{(i)}$ oder

arithmetische Verknüpfung zweier Komponenten von $x^{(i)}$ oder einer Konstanten mit einer Komponente von $x^{(i)}$ ist. (Konstante kann man auch den Eingangsdaten zurechnen). Im ganzen hat man

$$y = F(x) = \Phi_N(\Phi_{N-1}(\dots(\Phi_1(x))\dots))$$

Bei gerundeter Rechnung hat man nun nicht die Abbildung Φ_i zur Verfügung, sondern nur die Ersatzoperation $\Phi_{i,\varepsilon}$, wobei diese Ersatzoperation dadurch entsteht, daß man “ $\#$ ” durch “ $g\ell(\cdot\#)$ ” ersetzt. Unter Ausnutzung der Resultate aus den Abschnitten 7.1 und 7.2 hat man

$$\frac{|g\ell(\tilde{x} \pm \tilde{y}) - (x \pm y)|}{|x \pm y|} \leq \frac{|x|}{|x \pm y|} \frac{|\Delta x|}{|x|} + \frac{|y|}{|x \pm y|} \frac{|\Delta y|}{|y|} + \varepsilon + \text{Terme höherer Ordnung}$$

$$\frac{|g\ell(\tilde{x}\# \tilde{y}) - (x\#y)|}{|x\#y|} \leq \frac{|\Delta x|}{|x|} + \frac{|\Delta y|}{|y|} + \varepsilon + \text{Terme höherer Ordnung.}$$

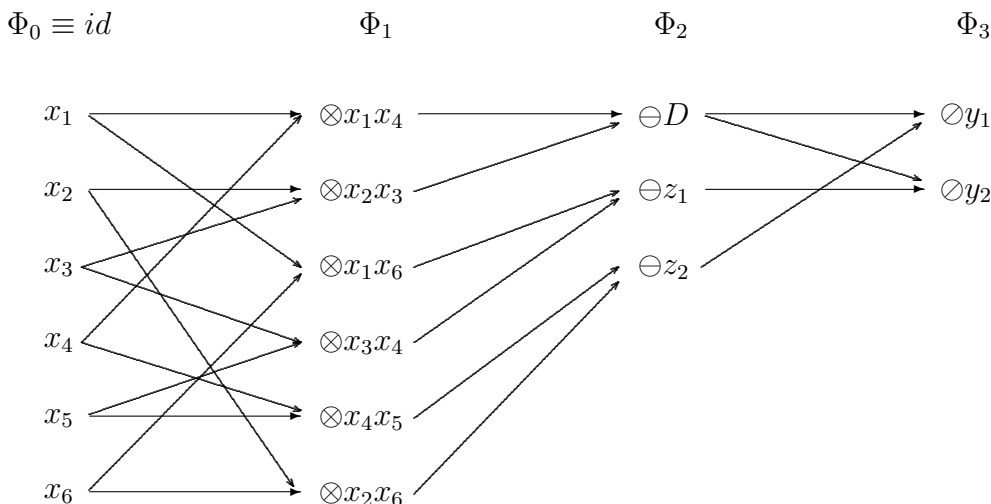
Mit

$$\tilde{y} := \Phi_{N,\varepsilon}(\Phi_{N-1,\varepsilon}(\dots(\Phi_{1,\varepsilon}(\text{rd}(x)))\dots))$$

erhält man unter Ausnutzung dieser Gesetze eine Darstellung für den relativen Fehler in den einzelnen Komponenten von \tilde{y} .

Beispiel 7.3.1 Wir betrachten die Lösung eines 2×2 -Gleichungssystems nach der Cramer’schen Regel. Eingangsdaten sind die 4 Koeffizienten der Matrix und die beiden Komponenten der rechten Seite. Resultat y sind die beiden Komponenten der Lösung:

$$\begin{pmatrix} x_1 & x_2 \\ x_3 & x_4 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} x_5 \\ x_6 \end{pmatrix} \quad \begin{array}{l} y_1 = \frac{x_5x_4 - x_6x_2}{x_1x_4 - x_2x_3} \quad \left. \begin{array}{l} \} z_2 \\ \} D \end{array} \right. \\ y_2 = \frac{x_1x_6 - x_3x_5}{x_1x_4 - x_2x_3} \quad \left. \begin{array}{l} \} z_1 \\ \} D \end{array} \right. \end{array}$$



Es wird nun ersetzt: $id(\cdot)$ durch $rd(\cdot)$, $x\#y$ durch $g\ell(x\#y)$. Dann erhält man unter Ausnutzung der obenstehenden Gesetze folgendes Diagramm für die maximalen Beträge der relativen Fehler:

7.4. DAS ALLGEMEINE FEHLERFORTPFLANZUNGSGESETZ. KONDITIONSZAHLEN EINES MAT

$$\left. \begin{array}{l} \varepsilon \\ \varepsilon \\ \varepsilon \\ \varepsilon \\ \varepsilon \\ \varepsilon \end{array} \right\}^* \begin{array}{l} 3\varepsilon \\ 3\varepsilon \\ 3\varepsilon \\ 3\varepsilon \\ 3\varepsilon \\ 3\varepsilon \end{array} \left. \begin{array}{l} (|x_2x_3| + |x_1x_4|) \frac{3\varepsilon}{|D|} + \varepsilon =: f_1 \\ (|x_1x_6| + |x_3x_5|) \frac{3\varepsilon}{|z_1|} + \varepsilon =: f_2 \\ (|x_4x_5| + |x_6x_2|) \frac{3\varepsilon}{|z_2|} + \varepsilon =: f_3 \end{array} \right\}^{***} \quad \begin{array}{l} \left| \frac{\Delta y_{1,\varepsilon}}{y_1} \right| \leq f_3 + f_1 + \varepsilon \\ \left| \frac{\Delta y_{2,\varepsilon}}{y_2} \right| \leq f_2 + f_1 + \varepsilon \end{array}$$

* Rundung x_1, \dots, x_6

** Fehlerfortpflanzung der relativen Fehler bei Multiplikation + neue hinzukommende Rundung

*** Fehlerfortpflanzung bei Subtraktion + neue Rundung

mit

$$\begin{aligned} \left| \frac{\Delta y_{1,\varepsilon}}{y_1} \right| &\leq 3\varepsilon \left(1 + \frac{|x_4x_5| + |x_6x_2|}{|D||y_1|} + \frac{|x_2x_3| + |x_1x_4|}{|D|} \right) \\ \left| \frac{\Delta y_{2,\varepsilon}}{y_2} \right| &\leq 3\varepsilon \left(1 + \frac{|x_1x_6| + |x_3x_5|}{|D||y_2|} + \frac{|x_2x_3| + |x_1x_4|}{|D|} \right) \end{aligned}$$

Terme höherer Ordnung in ε wurden dabei vernachlässigt. Man erkennt, daß die so berechnete Lösung ungenau wird, falls bei einer der drei Subtraktionen starke Auslöschung auftritt. \square

Im Prinzip kann man so die Fehlerfortpflanzung der relativen (oder auch der absoluten) Fehler in jedem Algorithmus vollständig verfolgen, die Auswertung der Abschätzungen wird allerdings u.U. recht schwierig. (Eine gründliche Darstellung dieses Themenkreises findet sich bei Stummel & Hainer, *Praktische Mathematik*, 2. Auflage, Teubner-Verlag).

7.4 Das allgemeine Fehlerfortpflanzungsgesetz. Konditionszahlen eines mathematischen Problems.

Es stellt sich uns nun die Frage, eine wie hohe Genauigkeit man überhaupt sinnvollerweise im Resultat einer numerischen Berechnung erwarten darf. Dazu müssen wir beachten, daß zumindest Eingangs- und Ausgangsdaten der Berechnung gerundete Zahlen sind, d.h. man hat bestenfalls

$$\tilde{y} = \text{rd}(F(\text{rd}(x))) = (I + \varepsilon D_1)F((I + \varepsilon D_2)x) \tag{7.2}$$

wobei D_1, D_2 zwei Diagonalmatrizen sind, deren Diagonalelemente betragsmäßig ≤ 1 sind. Man nennt einen Algorithmus zur Berechnung von F gutartig (auf \mathcal{D}), wenn das

berechnete Resultat y_ε darstellbar ist als

$$y_\varepsilon = (I + \varepsilon D_1)F((I + \varepsilon D_2)x) \quad \text{mit } \|D_1\|, \|D_2\| \leq c(n, m)$$

für alle $x \in \mathcal{D}$ mit $c(n, m)$ unabh. von x . Allgemeiner fragen wir zunächst nach der Größe der Differenz

$$y - \tilde{y} = F(x) - F(\tilde{x})$$

wenn $x - \tilde{x}$ (oder eine Schranke dafür) bekannt ist. Die Antwort dieser Frage liefert natürlich der Taylorsche Satz. Hierbei spielt die spezielle Form der Differenz $x - \tilde{x}$ überhaupt noch keine Rolle.

Satz 7.4.1 *Es sei $F = \begin{pmatrix} f_1 \\ \vdots \\ f_m \end{pmatrix} : \mathcal{D} \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$, $\mathcal{D} \neq \emptyset$ offen, konvex. Es gelte:*

$\forall x, y \in \mathcal{D}, \forall i \in \{1, \dots, n\}, \forall j \in \{1, \dots, m\} : \exists \frac{\partial f_j}{\partial x_i}(x)$ und

$$\left| \underbrace{\frac{\partial f_j}{\partial x_i}(x)}_* - \underbrace{\frac{\partial f_j}{\partial x_i}(y)}_* \right| \leq L \sum_{k=1}^n \underbrace{|x_k - y_k|}_{**} \quad 1$$

* partielle Ableitung der j -ten Komponentenfunktion nach der i -ten Variablen

** Komponenten von x bzw y

Dann gilt für $x, \tilde{x} \in \mathcal{D}$

$$\underbrace{f_j(\tilde{x}) - f_j(x)}_{\Delta f_j} = \sum_{i=1}^n \left(\frac{\partial f_j}{\partial x_i}(x) \right) \underbrace{(\tilde{x}_i - x_i)}_{\Delta x_i} + R_j$$

$$\text{wobei } |R_j| \leq L \left(\sum_{k=1}^n |\tilde{x}_k - x_k| \right)^2$$

(7.3)

□

Beispiel 7.4.1 $n = 2, m = 1, F = (f_1), f_1(x_1, x_2) = x_1/x_2$.

$$\frac{\partial f_1}{\partial x_1} = \frac{1}{x_2} \quad \frac{\partial f_1}{\partial x_2} = -\frac{x_1}{x_2^2}$$

($\mathcal{D} = \{(x_1, x_2) : \delta_1 > |x_1|, x_2 > \delta_2 > 0\}$). Bestimmung von $L : x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \in \mathcal{D}$

$$\frac{\partial^2 f}{\partial x_1^2} = 0 \quad \frac{\partial^2 f_1}{\partial x_1 \partial x_2} = -\frac{1}{x_2^2} \quad \frac{\partial^2 f_1}{\partial x_2^2} = \frac{2x_1}{x_2^3}$$

7.4. DAS ALLGEMEINE FEHLERFORTPFLANZUNGSGESETZ. KONDITIONSZAHLEN EINES MAT

$$\mathcal{G} = \bar{\mathcal{D}} = \{(x_1, x_2) : \delta_1 \geq |x_1|, x_2 \geq \delta_2 > 0\}$$

$$\begin{aligned} L &= \max \left\{ \left| \frac{1}{x_2^2} \right|, \frac{2|x_1|}{|x_2^3|} : \delta_1 \geq |x_1|, \delta_2 \leq x_2 \right\} \\ &= \max \left\{ \frac{1}{\delta_2^2}, \frac{2\delta_1}{\delta_2^3} \right\} \leq 2 \frac{\max\{1, \delta_1\}}{(\min\{1, \delta_2\})^3} \end{aligned}$$

Also

$$\frac{\tilde{x}_1}{\tilde{x}_2} - \frac{x_1}{x_2} = \frac{1}{x_2}(\tilde{x}_1 - x_1) - \frac{x_1}{x_2^2}(\tilde{x}_2 - x_2) + \text{Term zweiter Ordnung in } \tilde{x}_i - x_i$$

Division durch x_1/x_2 liefert mit $z := x_1/x_2$, $\tilde{z} := \tilde{x}_1/\tilde{x}_2$

$$\frac{z - \tilde{z}}{z} = \frac{\tilde{x}_1 - x_1}{x_1} - \frac{\tilde{x}_2 - x_2}{x_2} + \text{Term zweiter Ordnung in } \tilde{x}_i - x_i$$

d.h. das in 3.2 bereits hergeleitete Fehlerfortpflanzungsgesetz der Division. \square

Wir vernachlässigen R_j in (7.3) und schreiben

$$f_j(\tilde{x}) - f_j(x) \approx \sum_{i=1}^n \frac{\partial f_j}{\partial x_i}(x) (\tilde{x}_i - x_i)$$

Übergang zu den Beträgen liefert

$$|f_j(\tilde{x}) - f_j(x)| \leq \sum_{i=1}^n \underbrace{\left| \frac{\partial f_j}{\partial x_i}(x) \right|}_* \underbrace{|\tilde{x}_i - x_i|}_{**}$$

- * Dämpfung des Fehlereinflusses für $|\cdot| \leq 1$
Fehlerverstärkung für $|\cdot| > 1$
Gibt Empfindlichkeit von f_j gegenüber Fehlern in x_i an;
- ** **absolute** Fehler in \tilde{x}_i bzgl. x_i

Falls $f_j(x) \neq 0$ und $x_i \neq 0 \quad i = 1, \dots, n$, kann man zu **relativen** Fehlern übergehen:

$$\frac{|f_j(\tilde{x}) - f_j(x)|}{|f_j(x)|} \leq \sum_{i=1}^n \overbrace{\left| \frac{\partial f_j}{\partial x_i}(x) \right| \frac{x_i}{f_j(x)}}^* \frac{|\tilde{x}_i - x_i|}{|x_i|}$$

Definition 7.4.1 Die Zahlen

$$\left| \frac{\partial f_j}{\partial x_i}(x) \right| \quad \text{bzw.} \quad \left| \frac{\partial f_j}{\partial x_i}(x) \frac{x_i}{f_j(x)} \right| \quad i = 1, \dots, n, \quad j = 1, \dots, m$$

nennen wir die **Einzel-Konditionszahlen** von F bzgl. des absoluten bzw. des relativen Fehlers an der Stelle x . (Sie geben an, wie sich ein einzelner Fehler in einer Komponente des Arguments auf eine Komponentenfunktion auswirkt.) Ferner heie

$$C_{F,\text{abs}}(x) := n \max_{i,j} \left| \frac{\partial f_j}{\partial x_i}(x) \right|, \quad C_{F,\text{rel}}(x) := n \max_{i,j} \left| \frac{\partial f_j}{\partial x_i}(x) \frac{x_i}{f_j(x)} \right|$$

die (Gesamt-)Konditionszahl von F bzgl. des absoluten bzw. des relativen Fehlers in x . □

Satz 7.4.2 Sei $F = \begin{pmatrix} f_1 \\ \vdots \\ f_m \end{pmatrix}$ auf $\mathcal{D} \subset \mathbb{R}^n$, ($\mathcal{D} \neq \emptyset$ offen, konvex) partiell differenzierbar und die partiellen Ableitungen von F seien Lipschitzstetig auf \mathcal{D} . Dann gilt fur $\tilde{x}, x \in \mathcal{D}$

$$\max_j |f_j(x) - f_j(\tilde{x})| \leq C_{F,\text{abs}}(x) \max_k |\tilde{x}_k - x_k| + \text{Terme zweiter Ordnung}$$

und, falls $f_j(x) \neq 0$ fur $j = 1, \dots, m$ und $x_k \neq 0$, $k = 1, \dots, n$

$$\max_j \frac{|f_j(x) - f_j(\tilde{x})|}{|f_j(x)|} \leq C_{F,\text{rel}}(x) \max_k \frac{|\tilde{x}_k - x_k|}{|x_k|} + \text{Terme zweiter Ordnung}$$

□

Definition 7.4.2 Das Problem, zu gegebenem x $y = F(x)$ zu berechnen heit **gut konditioniert** bzgl. des absoluten bzw. des relativen Fehlers, falls $C_{F,\text{abs}}(x)$ bzw. $C_{F,\text{rel}}(x)$ in der Groenordnung von 1 liegt und **schlecht konditioniert**, falls $C_{F,\text{abs}}(x)$ bzw. $C_{F,\text{rel}}(x) \gg 1$. □

Diese Definition ist natrlich etwas vage, aber im konkreten Einzelfall lsst sich durchaus entscheiden, was unter ‘‘in der Groenordnung von 1’’ zu verstehen ist.

Beispiel 7.4.2 Wie Beispiel 7.1.3. Es wird

$$\begin{aligned} \frac{\partial y_1}{\partial x_5} &= \frac{x_4}{D}, & \frac{\partial y_1}{\partial x_6} &= -\frac{x_2}{D}, & \frac{\partial y_1}{\partial x_4} &= \frac{x_5}{D} - y_1 \frac{x_1}{D}, \\ \frac{\partial y_1}{\partial x_2} &= -\frac{x_6}{D} + y_1 \frac{x_3}{D}, & \frac{\partial y_1}{\partial x_1} &= -y_1 \frac{x_4}{D}, & \frac{\partial y_1}{\partial x_3} &= y_1 \frac{x_2}{D}, \\ \frac{\partial y_2}{\partial x_5} &= -\frac{x_3}{D}, & \frac{\partial y_2}{\partial x_6} &= \frac{x_1}{D}, & \frac{\partial y_2}{\partial x_4} &= \frac{x_6}{D} - y_2 \frac{x_4}{D}, \\ \frac{\partial y_2}{\partial x_3} &= -\frac{x_5}{D} + y_2 \frac{x_2}{D}, & \frac{\partial y_2}{\partial x_2} &= y_2 \frac{x_3}{D}, & \frac{\partial y_2}{\partial x_1} &= -y_2 \frac{x_1}{D}, \end{aligned}$$

d.h.

$$C_{F,\text{rel}}(x) = 6 \max \left\{ \frac{|x_4 x_5|}{|D| |y_1|} + \frac{|x_1 x_4|}{|D|}, \frac{|x_2 x_6|}{|D| |y_1|} + \frac{|x_3 x_2|}{|D|}, \frac{|x_3 x_5|}{|D| |y_2|} + \frac{|x_2 x_3|}{|D|}, \frac{|x_1 x_6|}{|D| |y_2|} + \frac{|x_1 x_4|}{|D|} \right\}.$$

Das Problem der Gleichungsauflösung bei 2×2 Gleichungssystemen ist also schlecht konditioniert, (wenn man für *beide* Lösungskomponenten den relativen Fehler zugrunde legt), falls

$$|D| \ll \max |x_i|$$

oder

$$\min\{|y_1|, |y_2|\} \ll \max\{|x_5|, |x_6|\}$$

□

Gehen wir nun davon aus, daß die einzigen Rundungsfehler in der Rundung der Eingangs- und Ausgangsdaten bestehen, dann erhalten wir als Schranke für den relativen Fehler

$$\max_j \frac{|\tilde{y}_j - y_j|}{|y_j|} \leq (1 + C_{F,\text{rel}}(x))\varepsilon + \mathcal{O}(\varepsilon^2) \quad (7.4)$$

(wobei natürlich vorausgesetzt ist, daß keiner der Werte $x_i, y_j = 0$ ist.)

Entsprechende Betrachtungen kann man für den absoluten Fehler anstellen. Man kann nun nicht erwarten, daß das Ergebnis irgendeines numerischen Algorithmus wesentlich genauer ist, als die Schranke (7.4) angibt. (Eine gewisse Überschätzung liegt im Faktor n in der Definition von $C_{F,\text{rel}}$.)

7.5 Rückwurfanalyse²

In Abschnitt 7.3 haben wir bereits eine Methode kennengelernt, um den Gesamteffekt aller Rundungsfehler abschätzen zu können. Die Gutartigkeit des Algorithmus²

²Ausführliche Darstellung z.B. bei Wilkinson, "Rundungsfehler", Springer HTB 44

entscheidet man dann aufgrund eines Vergleichs mit dem Resultat einer Konditionsanalyse (siehe Abschnitt 7.4). Die Rückwurfanalyse bietet dagegen in vielen Fällen die Möglichkeit, die Gutartigkeit eines Verfahrens direkt nachzuweisen. Hierbei wird unter systematischer Ausnutzung der Modellvorstellung für $gl(x \# y)$ das berechnete Resultat als exaktes Resultat für gestörte Eingangsdaten dargestellt. Man nimmt bei dieser Betrachtung o.B.d.A. an, daß die Eingangsdaten Maschinenzahlen sind, weil der Einfluß der Rundung der Eingangsdaten ohnehin allein durch die Kondition des Problems (unabhängig vom Algorithmus) gegeben ist.

Beispiel 7.5.1 Gauß'scher Algorithmus für ein 2×2 -Gleichungssystem

$$\begin{pmatrix} x_1 & x_3 \\ x_2 & x_4 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} x_5 \\ x_6 \end{pmatrix} \quad \text{o.B.d.A.} \quad x_i \in \mathbb{M}, \quad i = 1, \dots, 6$$

Fall $|x_1| \geq |x_2|$ und $x_1x_4 - x_2x_3 \neq 0$. (d.h. kein Zeilentausch notwendig, eindeutige Lösung)

Algorithmus:	Maschinenalgorithmus:
$\zeta_1 := x_2/x_1$	$\zeta_{1,\varepsilon} := gl(x_2/x_1) = x_2(1 + \eta_1)/x_1$
$\zeta_2 := \zeta_1 x_3$	$\zeta_{2,\varepsilon} := gl(\zeta_{1,\varepsilon} x_3) = x_2(1 + \eta_1)x_3(1 + \eta_2)/x_1$
$\zeta_3 := x_4 - \zeta_2$	$\zeta_{3,\varepsilon} := gl(x_4 - \zeta_{2,\varepsilon}) = (x_4 - \zeta_{2,\varepsilon})(1 + \eta_3)$ $= \left(x_4 - x_3 \frac{x_2}{x_1} (1 + \eta_1)(1 + \eta_2) \right) (1 + \eta_3)$
$\zeta_4 := \zeta_1 x_5$	$\zeta_{4,\varepsilon} := gl(\zeta_{1,\varepsilon} x_5) = x_5 \frac{x_2}{x_1} (1 + \eta_1)(1 + \eta_4)$
$\zeta_5 := x_6 - \zeta_4$	$\zeta_{5,\varepsilon} := gl(x_6 - \zeta_{4,\varepsilon}) = (x_6 - \zeta_{4,\varepsilon}) / (1 + \eta'_5)$
$y_2 := \zeta_5 / \zeta_3$	$y_{2,\varepsilon} := gl(\zeta_{5,\varepsilon} / \zeta_{3,\varepsilon}) = \zeta_{5,\varepsilon} / (\zeta_{3,\varepsilon} (1 + \eta'_6))$ $= \frac{x_6 - x_5 \frac{x_2}{x_1} (1 + \eta_1)(1 + \eta_4)}{\left(x_4 - x_3 \frac{x_2}{x_1} (1 + \eta_1)(1 + \eta_2) \right) (1 + \eta_3)(1 + \eta'_5)(1 + \eta'_6)}$
$\zeta_6 := x_3 y_2$	$\zeta_{6,\varepsilon} := gl(x_3 - y_{2,\varepsilon}) = x_3(1 + \eta_7)y_{2,\varepsilon}$
$\zeta_7 := x_5 - \zeta_6$	$\zeta_{7,\varepsilon} := gl(x_5 - \zeta_{6,\varepsilon}) = (x_5 - \zeta_{6,\varepsilon}) / (1 + \eta'_8)$
$y_1 := \zeta_7 / x_1$	$y_{1,\varepsilon} := gl(\zeta_{7,\varepsilon} / x_1) = \zeta_{7,\varepsilon} / (x_1(1 + \eta'_9))$ $= \frac{x_5 - x_3(1 + \eta_7)y_{2,\varepsilon}}{x_1(1 + \eta'_9)(1 + \eta'_8)}$

Also wird

$$\underbrace{\{x_1(1 + \eta'_9)(1 + \eta'_8)\}}_{=:x_{1,\varepsilon}} y_{1,\varepsilon} + \underbrace{\{x_3(1 + \eta_7)\}}_{=:x_{3,\varepsilon}} y_{2,\varepsilon} = x_5$$

und mit

$$\begin{aligned} x_{1,\varepsilon} &= x_1(1 + \eta'_9)(1 + \eta'_8) \\ x_{2,\varepsilon} &= x_2(1 + \eta_1)(1 + \eta_4)(1 + \eta'_9)(1 + \eta'_8) \\ x_{3,\varepsilon} &= x_3(1 + \eta_7) \end{aligned}$$

wird

$$y_{2,\varepsilon} = \frac{x_6 - x_5 \frac{x_{2,\varepsilon}}{x_{1,\varepsilon}}}{\underbrace{\left(x_4 - x_{3,\varepsilon} \frac{x_{2,\varepsilon}}{x_{1,\varepsilon}} \cdot \frac{(1 + \eta_2)}{(1 + \eta_4)} \right)}_{N(\varepsilon)} (1 + \eta_3)(1 + \eta'_5)(1 + \eta'_6)}$$

d.h.

$$y_{2,\varepsilon} = \frac{x_6 - x_5 \frac{x_{2,\varepsilon}}{x_{1,\varepsilon}}}{N(\varepsilon)}$$

mit

$$\begin{aligned} N(\varepsilon) &= x_4(1 + \eta_3)(1 + \eta'_5)(1 + \eta'_6) - x_{3,\varepsilon} \frac{x_{2,\varepsilon}}{x_{1,\varepsilon}} \frac{(1 + \eta_2)}{(1 + \eta_4)} (1 + \eta_3)(1 + \eta'_5)(1 + \eta'_6) \\ &= x_{4,\varepsilon} - x_{3,\varepsilon} \frac{x_{2,\varepsilon}}{x_{1,\varepsilon}} \end{aligned}$$

wo

$$\begin{aligned} x_{4,\varepsilon} &= x_4 + x_4 \left\{ (1 + \eta_3)(1 + \eta'_5)(1 + \eta'_6) - 1 \right\} \\ &\quad - x_{3,\varepsilon} \frac{x_{1,\varepsilon}}{x_{2,\varepsilon}} \left\{ \frac{(1 + \eta_2)}{(1 + \eta_4)} (1 + \eta_3)(1 + \eta'_5)(1 + \eta'_6) - 1 \right\} \end{aligned}$$

d.h. letztlich

$$x_{2,\varepsilon} y_{1,\varepsilon} + x_{4,\varepsilon} y_{2,\varepsilon} = x_6$$

d.h. die berechneten Werte $y_{1,\varepsilon}, y_{2,\varepsilon}$ sind die exakten Lösungen des Gleichungssystems

$$\begin{pmatrix} x_{1,\varepsilon} & x_{3,\varepsilon} \\ x_{2,\varepsilon} & x_{4,\varepsilon} \end{pmatrix} \begin{pmatrix} y_{1,\varepsilon} \\ y_{2,\varepsilon} \end{pmatrix} = \begin{pmatrix} x_5 \\ x_6 \end{pmatrix}$$

Dabei gilt

$$\begin{aligned} |x_{1,\varepsilon} - x_1| &\leq |x_1|(2\varepsilon + \mathcal{O}(\varepsilon^2)) \leq 2\varepsilon \|x\|_\infty + \mathcal{O}(\varepsilon^2) \\ |x_{2,\varepsilon} - x_2| &\leq |x_2|(4\varepsilon + \mathcal{O}(\varepsilon^2)) \leq 4\varepsilon \|x\|_\infty + \mathcal{O}(\varepsilon^2) \\ |x_{3,\varepsilon} - x_3| &\leq \varepsilon |x_3| \leq \varepsilon \|x\|_\infty \end{aligned}$$

und weil

$$\begin{aligned} |x_{3,\varepsilon}| &\leq |x_3|(1 + \mathcal{O}(\varepsilon)) \\ \frac{|x_{2,\varepsilon}|}{|x_{1,\varepsilon}|} &\leq \frac{|x_2|(1 + \mathcal{O}(\varepsilon))}{|x_1|(1 + \mathcal{O}(\varepsilon))} \leq 1 + \mathcal{O}(\varepsilon) \end{aligned}$$

schließlich

$$\begin{aligned} |x_{4,\varepsilon} - x_4| &\leq |x_4|(3\varepsilon + \mathcal{O}(\varepsilon^2)) + |x_3|(5\varepsilon + \mathcal{O}(\varepsilon^2)) \\ &\leq 8\varepsilon \|x\|_\infty + \mathcal{O}(\varepsilon^2) \end{aligned}$$

d.h. das Verfahren ist "gutartig", es erfüllt i.w. die Bedingung (7.2) in Abschnitt 7.4. Natürlich setzt diese Überlegung voraus, daß der Maschinentalgorithmus durchführbar ist, d.h. daß

$$|N(\varepsilon)| > 0$$

136 KAPITEL 7. RUNDUNGSFEHLERANALYSE NUMERISCHER ALGORITHMEN, (ERG)

(und daß kein Exponentenüber/ unterlauf eintritt. Vom letzteren Fall sieht man aber gewöhnlich ab, d.h. man benutzt die idealisierte Modellvorstellung). Aber

$$\begin{aligned}
 |N(\varepsilon)| &= \left| x_4 - x_3 \frac{x_2}{x_1} + (x_{4,\varepsilon} - x_4) + \left(x_3 \frac{x_2}{x_1} - x_{3,\varepsilon} \frac{x_{2,\varepsilon}}{x_{1,\varepsilon}} \right) \right| \\
 &\geq \left| \frac{x_4 x_1 - x_3 x_2}{x_1} \right| - \overbrace{(8\varepsilon \|x\|_\infty + \mathcal{O}(\varepsilon^2))} & - \underbrace{\left| (x_3 - x_{3,\varepsilon}) \frac{x_2}{x_1} \right|}_{\vartheta_2 \varepsilon \|x\|_\infty} - \underbrace{\left| x_{3,\varepsilon} \left(\frac{x_2}{x_1} - \frac{x_{2,\varepsilon}}{x_{1,\varepsilon}} \right) \right|}_{|\cdot| \leq 1} \\
 &\geq \left| \frac{x_4 x_1 - x_3 x_2}{x_1} \right| - (9\varepsilon \|x\|_\infty + \mathcal{O}(\varepsilon^2)) - \underbrace{\underbrace{|x_{3,\varepsilon}|}_{\leq |x_3|(1+\varepsilon)} \underbrace{\left| \frac{x_2}{x_1} \right|}_{\leq 1}}_{\leq 6\varepsilon + \mathcal{O}(\varepsilon^2)} \underbrace{\left| 1 - \frac{1 + 4\varepsilon + \mathcal{O}(\varepsilon^2)}{1 - 2\varepsilon + \mathcal{O}(\varepsilon^2)} \right|}_{\leq 6\varepsilon + \mathcal{O}(\varepsilon^2)} \\
 &\geq \left| \frac{x_4 x_1 - x_3 x_2}{x_1} \right| - (15\varepsilon \|x\|_\infty + \mathcal{O}(\varepsilon^2)).
 \end{aligned}$$

Falls wir also als \mathcal{D} die Menge

$$|x_i| \leq \Gamma, \quad |x_4 x_1 - x_3 x_2| \geq \gamma, \quad |x_1| \geq |x_2|$$

benutzen, wird

$$|N(\varepsilon)| \geq \frac{\gamma}{\Gamma} - 15\varepsilon \Gamma + \mathcal{O}(\varepsilon^2) \geq \frac{\gamma}{2\Gamma} + \mathcal{O}(\varepsilon^2) > 0$$

falls $\varepsilon \leq \varepsilon_0$ und ε_0 so gewählt ist, daß

$$\varepsilon_0 \leq \frac{\gamma}{30\Gamma^2}$$

(Man muß dann noch prüfen, daß der $\mathcal{O}(\varepsilon^2)$ -Term wirklich vernachlässigbar klein ist, was aber hier der Fall ist.) \square

Beispiel 7.5.2 Polynomauswertung nach dem Horner Schema,

$$p_3(x) = ((a_3 x + a_2)x + a_1)x + a_0$$

o.B.d.A. $x, a_0, a_1, a_2, a_3 \in \mathbb{M}$.

Algorithmus:	Maschinenalgorithmus
$p_3 := a_3$	$p_{3,\varepsilon} := a_3$
$p_2 := p_3 * x + a_2$	$p_{2,\varepsilon} := gl(gl(p_{3,\varepsilon} * x) + a_2) = (a_3 * (1 + \eta_1) + a_2)(1 + \eta_2)$
$p_1 := p_2 * x + a_1$	$p_{1,\varepsilon} := gl(gl(p_{2,\varepsilon} * x) + a_1) = (p_{2,\varepsilon} * x(1 + \eta_3) + a_1)(1 + \eta_4)$
$p_0 := p_1 * x + a_0$	$p_{0,\varepsilon} := gl(gl(p_{1,\varepsilon} * x) + a_0) = (p_{1,\varepsilon} * x(1 + \eta_5) + a_0)(1 + \eta_6)$

Im ganzen:

$$\begin{aligned}
 p_{0,\varepsilon} &= \overbrace{a_3(1 + \eta_1)(1 + \eta_2)(1 + \eta_3)(1 + \eta_4)(1 + \eta_5)(1 + \eta_6)}^{a_{3,\varepsilon}} x^3 \\
 &\quad + \overbrace{a_2(1 + \eta_2)(1 + \eta_3)(1 + \eta_4)(1 + \eta_5)(1 + \eta_6)}^{a_{2,\varepsilon}} x_2 \\
 &\quad + \underbrace{a_1(1 + \eta_4)(1 + \eta_5)(1 + \eta_6)}_{a_{1,\varepsilon}} x + \underbrace{a_0(1 + \eta_6)}_{a_{0,\varepsilon}}
 \end{aligned}$$

d.h. der Algorithmus ist im Sinne von 7.4 gutartig, wenn man ihn als Abbildung

$$\begin{pmatrix} x \\ a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} \rightarrow p_3(x)$$

interpretiert. □

Bemerkung 7.5.1 Die Gutartigkeit eines Algorithmus besagt nicht, daß das berechnete Resultat im üblichen Sinne "genau" ist. Wenn nämlich die Kondition der Problemstellung schlecht ist, dann ist das Resultat unvermeidlich stark verfälscht. □

Die explizite Mitführung der Klammerausdrücke $(1 + \eta_1) \cdots (1 + \eta_n)$ bei der Rückwurfanalyse ist sehr lästig. Zur formalen Vereinfachung ist folgendes Ergebnis nützlich, das insbesondere auch erlaubt, die oben bereits aufgetretenen $\mathcal{O}(\varepsilon^2)$ -Terme besser zu beherrschen.

Satz 7.5.1 Es sei $n\varepsilon \leq 0.1$. Ferner gelte $|\eta_i| \leq \varepsilon$, $i = 1, \dots, n$. Dann ist

$$1 - 1.06n\varepsilon \leq \prod_{i=1}^n (1 + \eta_i) \leq 1 + 1.06n\varepsilon$$

d.h.

$$\prod_{i=1}^n (1 + \eta_i) = 1 + 1.06n\varepsilon\vartheta \quad \text{mit} \quad |\vartheta| \leq 1, \quad \vartheta \text{ abhängig von } \eta_1, \dots, \eta_n.$$

□

Beweis: Wegen $0 < 1 - \varepsilon \leq 1 + \eta_i \leq 1 + \varepsilon$ $i = 1, \dots, n$ ist

$$(1 - \varepsilon)^n \leq \prod_{i=1}^n (1 + \eta_i) \leq (1 + \varepsilon)^n.$$

Aber

$$(1 \pm \varepsilon)^n = \sum_{i=0}^n \binom{n}{i} (\pm\varepsilon)^i = 1 \pm n\varepsilon + \sum_{i=2}^n \binom{n}{i} \varepsilon^i (\pm 1)^i.$$

Ferner

$$\left| \sum_{i=2}^n \binom{n}{i} \varepsilon^i (\pm 1)^i \right| \leq \varepsilon^2 \frac{n^2}{2} \sum_{i=0}^{\infty} (n\varepsilon)^i \leq n\varepsilon \cdot \frac{1}{2} n\varepsilon \frac{1}{1-n\varepsilon} \leq n\varepsilon \cdot 0.05/0.9 \leq 0.06n\varepsilon.$$

□

7.6 Intervallararithmetik

Die Intervallararithmetik stellt einen Versuch zur Automatisierung der Kontrolle des Einflusses von Daten- und Rundungsfehlern auf das Ergebnis einer numerischen Rechnung dar. Falls

$$\tilde{x} = x + \Delta x, \quad (\text{Eingangsdatenfehler } |\Delta x| \leq d)$$

dann wird \tilde{x} ersetzt durch das Intervall $[\tilde{x} - d, \tilde{x} + d] = I$, und dies wird für alle Eingangsdaten durchgeführt. Die arithmetische Verknüpfung zweier Intervalle I_1, I_2 wird definiert durch

$$I_3 = I_1 \# I_2 := [\min_{x_i \in I_i} x_1 \# x_2, \max_{x_i \in I_i} x_1 \# x_2].$$

Bei der Ausführung von $\#$ als $g\ell(\cdot, \cdot)$ werden zusätzlich die Intervallgrenzen so verkleinert bzw. vergrößert, daß das berechnete Intervall \tilde{I}_3 (mit Grenzen in \mathbb{M}) $\tilde{I}_3 \supset I_3$ erfüllt. Dann erhält das (die) Ergebnisintervall(e) garantiert die exakte Lösung, d.h. man erhält automatisch Näherung + Fehlerschranke. In gewissen Spezialfällen und mit speziellen Algorithmen erhält man so sehr schöne Resultate. Unkritisch angewandt führt die Methode jedoch oft zu groben Überschätzungen.

Beispiel 7.6.1 (ohne Rundungsfehler):

Gleichungssystem 2×2 mit Cramerscher Regel

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \quad \begin{aligned} x_1 &= \frac{b_1 a_{22} - b_2 a_{12}}{a_{11} a_{22} - a_{21} a_{12}} \\ x_2 &= \frac{b_2 a_{11} - b_1 a_{21}}{a_{11} a_{22} - a_{21} a_{12}} \end{aligned}$$

$$a_{11} \in [1, 2], \quad a_{12} \in [2, 3], \quad b_1 \in [3, 4], \quad a_{21} \in [2, 4], \quad a_{22} \in [-1, 1], \quad b_2 \in [0, 1],$$

\Rightarrow

$$a_{11} a_{22} - a_{21} a_{12} \in [1, 2] [-1, 1] - [2, 4] [2, 3] = [-2, 2] - [4, 12] = [-14, -2]$$

$$\begin{aligned} x_1 &\in ([3, 4] [-1, 1] - [0, 1] [2, 3]) / [-14, -2] \\ &= ([-4, 4] - [0, 3]) / [-14, -2] = [-7, 4] / [-14, -2] = [-2, 3.5] \\ x_2 &\in ([1, 2] [0, 1] - [2, 4] [3, 4]) / [-14, -2] \\ &= ([0, 2] - [6, 16]) / [-14, -2] = [-16, -4] / [-14, -2] = [\frac{2}{7}, 8]. \end{aligned}$$

□

Beispiel 7.6.2 Schlecht konditioniertes Gleichungssystem, Intervallararithmetik mit Rundung, Gauß-Algorithmus ohne Pivotwahl, da Koeffizientenmatrix positiv definit:

$$\begin{pmatrix} 0.500 & -0.333 \\ -0.333 & 0.251 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0.137 \\ -0.100 \end{pmatrix}$$

3-stellige dezimale Rechnung:

$$-0.333 \mapsto [-0.334, -0.332]$$

$$0.251 \mapsto [0.250, 0.252]$$

$$-0.100 \mapsto [-0.101, -0.099]$$

$$0.5 \mapsto [0.499, 0.501]$$

$$0.137 \mapsto [0.136, 0.138].$$

$$[-0.334, -0.332]/[0.499, 0.501] = [-.66933\dots, -.66267\dots]$$

$$= [-.670, -.662]$$

$$[-.670, -.662] * [-0.334, -0.332] = [.219784, .22378]$$

$$= [.219, .224]$$

$$[-.670, -.662] * [0.136, 0.138] = [-.9246 \cdot 10^{-1}, -.90032 \cdot 10^{-1}]$$

$$= [-.925 \cdot 10^{-1}, -.900 \cdot 10^{-1}]$$

$$[.250, .252] - [.219, .224] = [.260 \cdot 10^{-1}, .330 \cdot 10^{-1}]$$

$$[-.101, -.990 \cdot 10^{-1}] - [-.925 \cdot 10^{-1}, -.900 \cdot 10^{-1}] = [-.11 \cdot 10^{-1}, -.9 \cdot 10^{-2}]$$

$$[-.11 \cdot 10^{-1}, -.9 \cdot 10^{-2}]/[.260 \cdot 10^{-1}, .330 \cdot 10^{-1}] = [-.42307\dots, -.2727\dots]$$

$$= \underline{[-.424, -.272]} = [x_2]$$

$$[-0.334, -0.332] * [x_2] = [.90304 \cdot 10^{-1}, .141616]$$

$$= [.903 \cdot 10^{-1}, .142]$$

$$[0.136, 0.138] - [.903 \cdot 10^{-1}, .142] = [-.600 \cdot 10^{-2}, .477 \cdot 10^{-1}]$$

$$[-.600 \cdot 10^{-2}, .477 \cdot 10^{-1}]/[.499, .501] = [-.12024 \cdot 10^{-1}, .9559\dots \cdot 10^{-1}]$$

$$= \underline{[-.121 \cdot 10^{-1}, .956 \cdot 10^{-1}]} = [x_1]$$

Wahre Lösung (10-stellig berechnet):

$$x_1 = 7.4396003 \cdot 10^{-2}$$

$$x_2 = -.2997057\dots$$

die Einschließung ist also korrekt. An der Tatsache, daß die Intervallbreite der Intervalle $[x_1], [x_2]$ in der gleichen Größenordnung liegt wie der maximale Betrag eines Elementes dieser Intervalle, erkennt man hier sofort, "daß etwas nicht stimmen kann" (Hier: $\text{cond}_{\|\cdot\|}(A)\varepsilon = 0.2374\dots$, d.h. man kann bei 3-stelliger Rechnung nichts Vernünftiges erwarten). \square

7.7 Weiterführende Literatur

- Alefeld, Götz; Herzberger, Jürgen: *Einführung in die Intervallrechnung* Mannheim - Wien - Zürich: Bibliographisches Institut, B.I.-Wissenschaftsverlag (1974).

- Higham, Nicholas J.: *Accuracy and stability of numerical algorithms*. 2nd ed. Philadelphia, PA: SIAM. xxx, 680 p. (2002)
- Jaulin, Luc; Kieffer, Michel; Didrit, Olivier; Walter, Eric: *Applied interval analysis. With examples in parameter and state estimation, robust control and robotics*. London: Springer. (2001)
- Overton, M.L.: *Numerical computing with IEEE floating point arithmetic* SIAM 2001.
- Stummel, Friedrich; Hainer, Karl: *Praktische Mathematik. 2., bearb. u. erweiterte Aufl.* B. G. Teubner. (1982).
- Wilkinson, J.H.: *Rundungsfehler (German translation of Rounding errors in algebraic processes)* Berlin-Heidelberg-New York: Springer-Verlag. (1969).

Kapitel 8

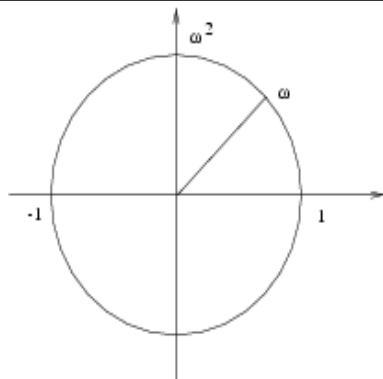
Trigonometrische Interpolation ERG

In Kapitel 1 haben wir gesehen, daß man ein Polynom gleichwertig durch $n + 1$ Koeffizienten einer beliebigen Basisdarstellung in Π_n oder durch $n + 1$ Wertepaare (Stützstellen/Stützwerte) darstellen kann. Die Transformation einer Darstellung in die andere ist dabei ein $\mathcal{O}(n^2)$ -Prozeß. Wir werden nun zeigen, daß man in einem Spezialfall den Aufwand auf $\mathcal{O}(n \log_2 n)$ reduzieren kann.

Im Folgenden bezeichnet ι die imaginäre Einheit $\sqrt{-1}$

8.1 Interpolation auf dem komplexen Einheitskreis

Definition 8.1.1 ω heißt primitive Einheitswurzel der Ordnung k , ($k \in \mathbb{N}$) falls $\omega^k = 1$ und $\omega^j \neq 1$, $1 \leq j < k$. □



ω primitive Einheitswurzel der Ordnung 8

ω^2 primitive Einheitswurzel der Ordnung 4

$$e^{\iota\pi/4} = e^{\iota 2\pi/8}$$

Satz 8.1.1 Sei ω eine primitive Einheitswurzel der Ordnung $n + 1$. Dann gilt

$$\sum_{s=0}^n \omega^{s\alpha} = \begin{cases} n + 1 & \alpha \equiv 0 \pmod{n + 1} \\ 0 & \text{sonst.} \end{cases}$$

□

Beweis: $\alpha \equiv 0 \pmod{n + 1} \Rightarrow \omega^{s\alpha} = 1 \Rightarrow \sum \dots = n + 1$.

Sei $\alpha \not\equiv 0 \pmod{n + 1}$. Setze $z := \omega^\alpha$. ($z \neq 1$). Dann $z^{n+1} = 1$

$$\sum_{s=0}^n \omega^{s\alpha} = \sum_{s=0}^n z^s = \frac{z^{n+1} - 1}{z - 1} = 0.$$

□

Wir betrachten nun die Aufgabenstellung der Transformation einer Koeffizientendarstellung eines Polynoms p in die Wertedarstellung sowie die zugehörige Rücktransformation:

$$\begin{aligned} T &: (a_0, \dots, a_n) \mapsto ((x_0, y_0), \dots, (x_n, y_n)) \\ T^{-1} &: ((x_0, y_0), \dots, (x_n, y_n)) \mapsto (a_0, \dots, a_n) \end{aligned}$$

wobei $x_k = \omega^k$ und ω eine primitive Einheitswurzel der Ordnung $n + 1$ ist.

Zur formalen Vereinfachung sei $n + 1 = 2^r$, $n = 2^r - 1$. Wir beschreiben zunächst die Transformation T :

Es ist

$$\begin{aligned} p(x) &= \sum_{v=0}^n a_v x^v = \sum_{\nu=0}^{\frac{n-1}{2}} a_{2\nu} x^{2\nu} + x \sum_{\nu=0}^{\frac{n-1}{2}} a_{2\nu+1} x^{2\nu} \\ &= p_1(y) + x p_2(y) \quad \text{mit} \quad y = x^2. \end{aligned}$$

Dabei gilt wegen $\frac{n-1}{2} = \frac{2^r-2}{2}$

$$\partial p_1 = \partial p_2 = 2^{r-1} - 1$$

d.h. für p_1 und p_2 kann dieselbe Zerlegungstechnik wiederholt werden. Die Auswertung von p_1 und p_2 ist zu leisten an den Stellen $y_k = x_k^2$, **d.h. in Wirklichkeit auf der Hälfte der Punkte**, z.B. bei

$$\begin{aligned} n = 7 \quad \omega &= \frac{1}{\sqrt{2}}(1 + i) \\ y_0 = 1, \quad y_1 = \omega^2, \quad y_2 = \omega^4, \quad y_3 = \omega^6, \\ y_4 = \omega^8 = 1 = y_0, \quad y_5 = \omega^{10} = \omega^2 = y_1, \quad y_6 = \omega^{12} = \omega^4 = y_2, \quad y_7 = \omega^{14} = \omega^6 = y_3. \end{aligned}$$

Sei

$$A(r) := \text{Aufwand zur Berechnung der Transformation } T \text{ mit } n = 2^r - 1.$$

Dann gilt ersichtlich

$$A(r) = 2A(r-1) + \frac{3}{2}2^r. \quad (8.1)$$

Der additive Teil $\frac{3}{2}2^r$ stammt von den $n+1$ Additionen $p_1(y_i) + x_i p_2(y_i)$ und den $(n+1)/2$ Multiplikationen $x_i p_2(y_i)$ (wegen $x_{i+2^{r-1}} = -x_i$ kann man die Hälfte der Multiplikationen einsparen). Aber

$$A(1) = 0$$

d.h.

$$A(r) = \frac{3}{2}2^r r = \frac{3}{2}(n+1) \log_2(n+1) = \mathcal{O}(n \log_2 n)$$

Wir beschreiben nun die inverse Transformation:

$$(a_0, \dots, a_n) \begin{pmatrix} \overbrace{1 & 1 & 1 & \cdots & \cdots & 1}^{\text{“van der Monde-Matrix”}} \\ \vdots & \omega & \omega^2 & \omega^3 & \cdots & \omega^n \\ \vdots & \omega^2 & \omega^4 & \omega^6 & \cdots & \omega^{2n} \\ \vdots & \omega^3 & \omega^6 & & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \omega^n & \omega^{2n} & \cdots & \cdots & \omega^{n^2} \end{pmatrix} = (y_0, \dots, y_n) \left. \vphantom{\begin{pmatrix} \overbrace{1 & 1 & 1 & \cdots & \cdots & 1}^{\text{“van der Monde-Matrix”}} \\ \vdots & \omega & \omega^2 & \omega^3 & \cdots & \omega^n \\ \vdots & \omega^2 & \omega^4 & \omega^6 & \cdots & \omega^{2n} \\ \vdots & \omega^3 & \omega^6 & & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \omega^n & \omega^{2n} & \cdots & \cdots & \omega^{n^2} \end{pmatrix}} \right\} (\omega^{kj}) \quad 0 \leq k, j \leq n$$

d.h. die Berechnung von a_0, \dots, a_n aus $((x_0, y_0), \dots, (x_n, y_n))$ läuft auf die Matrix-Vektormultiplikation

$$(a_0, \dots, a_n) = (y_0, \dots, y_n) V^{-1}$$

hinaus, wobei V die obige Vandermonde-Matrix ist. In diesem Spezialfall ist allerdings die Angabe von V^{-1} äußerst einfach:

Behauptung:

$$V^{-1} = (\tilde{v}_{kj}) = (\omega^{-kj}/(n+1)) \quad 0 \leq k, j \leq n$$

Beweis:

$$V^{-1}V = \left(\frac{1}{(n+1)} \sum_{s=0}^n \omega^{-ks} \omega^{js} \right) = \left(\frac{1}{(n+1)} \sum_{s=0}^n \omega^{(j-k)s} \right) = I$$

wegen Satz 8.1.1 und $k-j \neq 0 \pmod{n+1}$ falls $k \neq j$ und $0 \leq k, j \leq n$. Also wird

$$\begin{aligned} (a_0, \dots, a_n) &= \frac{1}{(n+1)} \left(\sum_{k=0}^n y_k \omega^{-kj} \right)_{j=0}^n \\ &= \frac{1}{(n+1)} \left(\sum_{k=0}^n y_k z_j^k \right)_{j=0}^n \end{aligned}$$

mit $z_j = \omega^{-j} = \left(\frac{1}{\omega}\right)^j$. Mit ω ist aber auch $\frac{1}{\omega}$ eine primitive Einheitswurzel der Ordnung $n+1$, also kann auch die Rücktransformation in $\mathcal{O}(n \log_2 n)$ Operationen geleistet werden.

Die Rekursion (8.1) drückt sich aus in einer rekursiven Faktorisierung der Matrix V . Wir bezeichnen V genauer mit $V(n+1)$ (wir behandeln hier den Fall $n+1=2^r$):

$$P_h(2^r)V(2^r) = \begin{pmatrix} V(2^{r-1}) & O \\ O & V(2^{r-1}) \end{pmatrix} \begin{pmatrix} I_{2^{r-1}} & I_{2^{r-1}} \\ D_{2^{r-1}} & -D_{2^{r-1}} \end{pmatrix}$$

worin $P_h(2^r)$ eine Permutationsmatrix ist, die die Zeilen $1, \dots, 2^r$ in die Reihenfolge $1, 3, \dots, 2^r - 1, 2, 4, \dots, 2^r$ permutiert und

$$D_{2^{r-1}} = \text{diag}(1, \omega, \dots, \omega^{2^{r-1}-1}) \quad \text{mit } \omega = \exp(i\pi/2^{r-1}).$$

8.2 Trigonometrische Interpolation: (FFT Fast Fourier Transform)

Aufgabe: Gegeben sind Wertepaare $\left(\frac{2\pi k}{n+1}, y_k\right)$, $k = 0, \dots, n$, $\omega := e^{i2\pi/(n+1)}$ primitive Einheitswurzel der Ordnung $n+1$, $n = 2^r - 1$, d.h. $n+1 = 2^r$.
Gesucht: Die zugehörige diskrete Fourierdarstellung, d.h.

$$f(x) = \frac{a_0}{2} + \sum_{j=1}^{m-1} \{a_j \cos(jx) + b_j \sin(jx)\} + a_m \cos(mx), \quad m = 2^{r-1}$$

$$2m \quad \text{Unbekannte, } m = 2^{r-1} \quad 2m = 2^r = n+1$$

$$2m \quad \text{Daten } y_0, \dots, y_n$$

mit

$$f(x_j) = y_j \quad j = 0, \dots, n.$$

Wir führen diese Aufgabe nun auf die obige komplexe Polynomauswertung zurück.

Man setze mit $b_0 = 0$, $b_m = 0$, $i = \sqrt{-1}$ und $z \stackrel{\text{def}}{=} \exp(ix)$ (d.h. $z(x_1) = \omega = e^{\frac{i2\pi}{n+1}}$ ist $(n+1)$ -te primitive Einheitswurzel)

$$\begin{aligned} c_j &= \frac{1}{2}(a_j - ib_j) \\ c_{-j} &= \frac{1}{2}(a_j + ib_j), \quad j = 0, \dots, m. \end{aligned}$$

8.2. TRIGONOMETRISCHE INTERPOLATION: (FFT FAST FOURIER TRANSFORM) 145

Damit wird für $x \in \{\frac{2\pi k}{n+1}\}$ wegen $\omega^{2m} = \omega^{n+1} = 1$

$$\begin{aligned} f(x) &= \frac{1}{2}a_0 + \sum_{j=1}^m \{a_j \cos jx + b_j \sin jx\} \\ &= c_0 + \sum_{j=1}^m \{(c_j + c_{-j}) \cos jx + i(c_j - c_{-j}) \sin jx\} \\ &= \sum_{j=-m}^m c_j e^{ijx} = \sum_{j=-m}^m c_j z^j \\ &= z^{-m} \sum_{j=0}^{2m-1} \tilde{c}_j z^j \\ \text{mit } \tilde{c}_j &= c_{j-m} \quad j = 1, \dots, 2m-1, \quad \tilde{c}_0 = c_{-m} + c_m = a_m \end{aligned}$$

d.h. die Bestimmung der a_j, b_j ist zurückgeführt auf die Bestimmung der komplexen Koeffizienten \tilde{c}_j des Polynoms vom Grad n mit $p_n(\omega^j) = y_j \omega^{mj} \quad j = 0, \dots, 2m-1, \omega^{mj} = (-1)^j$, und kann mit obigem Algorithmus in $\mathcal{O}(n \log_2 n)$ Operationen geleistet werden.

(Man beachte, daß $c_{-m} = c_m$, so daß nur $2m$ Unbekannte auftreten).

Wir haben bereits oben gesehen, daß diese Interpolationsaufgabe durch die Matrix-Vektormultiplikation

$$\begin{pmatrix} \tilde{c}_0 \\ \tilde{c}_1 \\ \dots \\ \dots \\ \tilde{c}_n \end{pmatrix} = \frac{1}{n+1} \left(\left(\frac{1}{\omega} \right)^{ij} \right) \begin{pmatrix} \tilde{y}_0 \\ \tilde{y}_1 \\ \dots \\ \dots \\ \tilde{y}_n \end{pmatrix}$$

mit $\tilde{y}_j = (-1)^j y_j$ geleistet wird, wobei $w = \frac{1}{\omega}$ wieder eine komplexe Einheitswurzel der Ordnung $n+1 = 2^r = 2m$ ist. In den Matrixkomponenten laufen i und j von 0 bis n und i repräsentiert den Zeilen- und j den Spaltenindex. Wir lassen den Faktor $\frac{1}{n+1}$ fort und setzen also

$$d_i = (n+1)\tilde{c}_i.$$

Dann haben wir

$$\begin{pmatrix} d_0 \\ d_1 \\ \dots \\ \dots \\ d_n \end{pmatrix} = \left(\left(\frac{1}{\omega} \right)^{ij} \right) \begin{pmatrix} \tilde{y}_0 \\ \tilde{y}_1 \\ \dots \\ \dots \\ \tilde{y}_n \end{pmatrix}.$$

Wir führen nun den Reduktionsschritt auf zwei unabhängige Gleichungen der halben Dimension vor. Dazu betrachten wir die d_i mit geraden und ungeraden Indizes getrennt.

Für $i = 0, \dots, m-1$ ist

$$\begin{aligned}
 d_{2i} &= \sum_{j=0}^{2m-1} \left(\frac{1}{\omega}\right)^{2ij} y_j \\
 &= \sum_{j=0}^{m-1} \left(\frac{1}{\omega}\right)^{2ij} y_j + \sum_{j=m}^{2m-1} \left(\frac{1}{\omega}\right)^{2ij} y_j \\
 &= \sum_{j=0}^{m-1} \left(\left(\frac{1}{\omega}\right)^2\right)^{ij} y_j + \sum_{j=0}^{m-1} \left(\frac{1}{\omega}\right)^{2i(j+m)} y_{j+m} \\
 &= \sum_{j=0}^{m-1} \left(\left(\frac{1}{\omega}\right)^2\right)^{ij} (y_j + y_{m+j})
 \end{aligned}$$

weil $\left(\frac{1}{\omega}\right)^{2im} = \left(\left(\frac{1}{\omega}\right)^{2m}\right)^i = 1$. Analog wird für $i = 0, \dots, m-1$

$$\begin{aligned}
 d_{2i+1} &= \sum_{j=0}^{2m-1} \left(\frac{1}{\omega}\right)^{(2i+1)j} y_j \\
 &= \sum_{j=0}^{m-1} \left(\frac{1}{\omega}\right)^{2ij} \left(\frac{1}{\omega}\right)^j y_j + \sum_{j=m}^{2m-1} \left(\frac{1}{\omega}\right)^{2ij} \left(\frac{1}{\omega}\right)^j y_j \\
 &= \sum_{j=0}^{m-1} \left(\left(\frac{1}{\omega}\right)^2\right)^{ij} \left(\frac{1}{\omega}\right)^j y_j + \sum_{j=0}^{m-1} \left(\frac{1}{\omega}\right)^{2i(j+m)} \left(\frac{1}{\omega}\right)^{j+m} y_{j+m} \\
 &= \sum_{j=0}^{m-1} \left(\left(\frac{1}{\omega}\right)^2\right)^{ij} \left(\frac{1}{\omega}\right)^j (y_j - y_{m+j})
 \end{aligned}$$

weil $\left(\frac{1}{\omega}\right)^m = -1$. Setzen wir also mit $w = \left(\frac{1}{\omega}\right)$

$$\begin{aligned}
 d^{(1,1)} &= (d_0, d_2, \dots, d_{2m-2})^T, \\
 y^{(1,1)} &= (y_0 + y_m, \dots, y_{m-1} + y_{2m-1})^T, \\
 d^{(1,2)} &= (d_1, d_3, \dots, d_{2m-1})^T, \\
 y^{(1,2)} &= (y_0 - y_m, w(y_1 - y_{m+1}), \dots, w^{m-1}(y_{m-1} - y_{2m-1}))^T,
 \end{aligned}$$

dann sind $d^{(1,1)}$ und $y^{(1,1)}$ sowie $d^{(1,2)}$ und $y^{(1,2)}$ durch identische Gleichungen wie oben d und y miteinander verknüpft, wobei nur die Dimension sich halbiert hat und $\left(\frac{1}{\omega}\right)$ durch $\left(\frac{1}{\omega}\right)^2$ ersetzt ist. Nach $r = \log_2 n$ solchen Reduktionsschritten ist man dann auf skalare Gleichungen der Form

$$d^{(r-1,j)} = y^{(r-1,j)}, \quad j = 0, \dots, 2m-1$$

gelangt, also trivialen Identitäten. Die einzigen Rechenoperationen, die dabei zu leisten sind, sind pro Schritt die Umrechnung der y 's, also die Additionen bzw. Subtraktionen

8.2. TRIGONOMETRISCHE INTERPOLATION: (FFT FAST FOURIER TRANSFORM)147

und die Multiplikationen mit $\left(\frac{1}{\omega}\right)^j$. Man kann alle diese Operationen in ein und demselben Vektor ablaufen lassen. Wenn man dann die geradzahigen Komponenten in die erste (Teil-)Hälfte und die ungeradzahigen jeweils die zweite ablegt, dann erhält man schliesslich die ursprünglich gesuchten d_j in einer permutierten Form. Die Originalanordnung erhält man durch den sogenannten „bitreversal“-Algorithmus. Numeriert man die Komponenten von 0 bis $2m - 1$, so sind die Binärdarstellungen dieser Indizes gerade

$$000000 \text{ (} r \text{ bits) bis } 111111 \text{ (} r \text{ bits .)}$$

Nun werden diese Indizes umgerechnet, indem man die Bitdarstellung ‘umgekehrt’ abliest, also das höchstwertigste Bit als Bitposition null interpretiert usw. d.h. z.B. 00110 wird zum Index 12, und die entsprechende Vektorkomponente wird dann der umgerechneten Position im (endgültigen) Vektor d zugewiesen.

Beispiel 8.2.1

$$r = 2, n = 3, m = 2 .$$

Wir haben also zunächst mit der Einheitswurzel der Ordnung 4 ($\omega = i$) zu rechnen. Sei der y -Vektor $y = (3, 2, 1, 4)^T$. Wegen der obigen Umrechnung in die komplexe Darstellung muß dieser Vektor mit ω^{mj} , $j = 0, \dots, 2m - 1$, also wegen $m = 2$ mit $1, -1, 1, -1$ durchmultipliziert werden. D.h. wir setzen in die obige Rekursion zunächst den Vektor $(3, -2, 1, -4)^T$ ein. Rekursion: (wegen $\omega = i$ ist $\left(\frac{1}{\omega}\right) = -i$)

$$\begin{pmatrix} 3 \\ -2 \\ 1 \\ -4 \end{pmatrix} \longrightarrow \begin{pmatrix} 4 \\ -6 \\ 2 \\ -2i \end{pmatrix}$$

Im zweiten Reduktionsschritt werden die Teilvektoren $(4, -6)^T$ und $(2, -2i)^T$ getrennt behandelt:

$$\begin{pmatrix} 4 \\ -6 \\ 2 \\ -2i \end{pmatrix} \longrightarrow \begin{pmatrix} -2 \\ 10 \\ 2 - 2i \\ 2 + 2i \end{pmatrix}$$

Jetzt kommt die „Bitreversion“

$$\begin{pmatrix} -2 \\ 10 \\ 2 - 2i \\ 2 + 2i \end{pmatrix} \longrightarrow \begin{pmatrix} -2 \\ 2 - 2i \\ 10 \\ 2 + 2i \end{pmatrix}$$

Um die richtigen Koeffizienten c_j zu bekommen, müssen wir noch durch $n + 1 = 4$ teilen:

$$\begin{pmatrix} -2 \\ 2 - 2i \\ 10 \\ 2 + 2i \end{pmatrix} \longrightarrow \begin{pmatrix} -\frac{1}{2} \\ \frac{1}{2} - \frac{1}{2}i \\ 2.5 \\ \frac{1}{2} + \frac{1}{2}i \end{pmatrix} .$$

Dies ist in obiger Notation der Koeffizientenvektor $(\tilde{c}_0, \dots, \tilde{c}_3)^T$ mit $\tilde{c}_j = c_{j-2}$ und laut Setzung $b_2 = 0$ ist $c_2 = c_{-2}$. Und jetzt folgt noch die Umrechnung der komplexen c 's in die reellen Koeffizienten nach obiger Formel:

$$\begin{aligned} a_0/2 &= c_0 = 2.5, \\ a_1 &= c_1 + c_{-1} = 1, \\ a_2 &= -\frac{1}{2} \\ b_0 &= 0, \\ b_1 &= i(c_1 - c_{-1}) = -1. \end{aligned}$$

Also erhalten wir als Interpolierende

$$2.5 + \cos(x) - \sin(x) - \frac{1}{2} \cos(2x)$$

und Einsetzen der Argumente

$$0, \pi/2, \pi, 3\pi/2$$

liefert erwartungsgemäss auch die Wertetabelle

$$3, 2, 1, 4.$$

Und noch ein grösseres Beispiel:

Beispiel 8.2.2 Beispiel 2 $n + 1 = 8, r = 3$.

y	\tilde{y}	Schritt 1	Schritt 2	Schritt 3	Division	Bitreversion
-1	-1	1	1	-1	$-\frac{1}{8}$	$\tilde{c}_0 = a_4$
1	-1	0	-2	3	$\frac{3}{8}$	\tilde{c}_4
0	0	0	1	$1 - 2i$	$(1 - 2i)/8$	\tilde{c}_2
1	-1	-2	$-2i$	$1 + 2i$	$(1 + 2i)/8$	\tilde{c}_6
2	2	-3	-3	$-3 - (1 - i)\sqrt{2}$	$(-3 - (1 - i)\sqrt{2})/8$	\tilde{c}_1
-1	1	$-(1 - i)\sqrt{2}$	$-(1 - i)\sqrt{2}$	$-3 + (1 - i)\sqrt{2}$	$(-3 + (1 - i)\sqrt{2})/8$	\tilde{c}_5
0	0	0	-3	$-3 + (1 - i)i\sqrt{2}$	$(-3 + (1 - i)i\sqrt{2})/8$	\tilde{c}_3
1	-1	0	$-(1 - i)(-i)\sqrt{2}$	$-3 - (1 - i)i\sqrt{2}$	$(-3 - (1 - i)i\sqrt{2})/8$	\tilde{c}_7

$$\begin{aligned} c_3 &= \tilde{c}_7 \\ c_2 &= \tilde{c}_6 \\ c_1 &= \tilde{c}_5 \\ c_0 &= \tilde{c}_4 = \frac{3}{8} = \frac{a_0}{2} \\ c_{-1} &= \tilde{c}_3 \\ c_{-2} &= \tilde{c}_2 \\ c_{-3} &= \tilde{c}_1 \end{aligned}$$

8.2. TRIGONOMETRISCHE INTERPOLATION: (FFT FAST FOURIER TRANSFORM)149

$$a_3 = c_3 + c_{-3} = \frac{-6 - 2\sqrt{2}}{8}$$

$$a_2 = c_2 + c_{-2} = \frac{1}{4}$$

$$a_1 = c_1 + c_{-1} = \frac{-6 + 2\sqrt{2}}{8}$$

$$b_3 = i(c_3 - c_{-3}) = \frac{1}{2\sqrt{2}}$$

$$b_2 = i(c_2 - c_{-2}) = -\frac{1}{2}$$

$$b_1 = i(c_1 - c_{-1}) = \frac{1}{2\sqrt{2}}$$

$$\begin{aligned} f(x) = & \frac{3}{8} + \frac{-6 + 2\sqrt{2}}{8} \cos(x) + \frac{1}{2\sqrt{2}} \sin(x) + \frac{1}{4} \cos(2x) - \frac{1}{2} \sin(2x) \\ & + \frac{-6 - 2\sqrt{2}}{8} \cos(3x) + \frac{1}{2\sqrt{2}} \sin(3x) - \frac{1}{8} \cos(4x) \end{aligned}$$

Weiterführende Literatur, insbesondere die Behandlung des allgemeinen Falles für n :

- W.L. Briggs, V.E. Hanson: *An Owners Manual for the discrete Fourier transform.* SIAM 1995.

Kapitel 9

Notation, Formeln

$$\begin{aligned} \|x\| &= (x^T x)^{1/2} && \text{euklidische Vektornorm, Länge von } x && (l_2\text{-Norm}) \\ \left\{ \begin{array}{l} \text{Andere gebräuchliche Längenmaße:} \\ \|x\|_\infty = \max_i |x_i| && \text{Maximumnorm} && (l_\infty\text{-Norm}) \\ \|x\|_1 = \sum_{i=1}^n |x_i| && \text{Betragssummennorm} && (l_1\text{-Norm}) \end{array} \right\} \end{aligned}$$

1. Ist f eine vektorwertige Funktion von n Veränderlichen x ,
 $f : \mathcal{D} \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$, so bezeichnet

$$\mathcal{J}_f(x) = \begin{pmatrix} \frac{\partial}{\partial x_1} f_1 & , \dots , & \frac{\partial}{\partial x_n} f_1 \\ \vdots & & \vdots \\ \frac{\partial}{\partial x_1} f_m & , \dots , & \frac{\partial}{\partial x_n} f_m \end{pmatrix}$$

die **Jacobimatrix** von f (Funktionalmatrix).

Jacobimatrix: Zeilennummer $\hat{=}$ Funktionsnummer
 Spaltennummer $\hat{=}$ Variablennummer

2. Der **Gradient** ist stets die transponierte Jacobimatrix:

$$\nabla f(x) = (\mathcal{J}_f(x))^T$$

Gradient: Zeilennummer $\hat{=}$ Variablennummer
 Spaltennummer $\hat{=}$ Funktionsnummer

Der Gradient einer skalaren Funktion ist also hier ein Spaltenvektor.

3. Ist G eine Funktion mehrerer Vektorvariablen, auch verschiedener Dimension, dann bezeichnet

$$\partial_i G(\dots)$$

die Jacobimatrix bezüglich des i -ten Teilvektors.

4. Für eine skalare Funktion $f : \mathcal{D} \subset \mathbb{R}^n \rightarrow \mathbb{R}$ bezeichnet

$$\nabla^2 f(x) = \left(\frac{\partial^2}{\partial x_i \partial x_j} f(x) \right) = \left((\nabla \nabla^T) f \right)(x)$$

die **Hessematrix** von f .

Für vektorwertige Funktionen kommt diese Konstruktion nur im Zusammenhang mit der Taylorentwicklung vor, d.h. als Vektor

$$\begin{bmatrix} d^T \nabla^2 f_1(x) d \\ \vdots \\ d^T \nabla^2 f_m(x) d \end{bmatrix} =: \underbrace{d^T \left(\nabla^2 f(x) \right) d}_{\text{symbolisch, nur für } m=1 \text{ echte Matrix-Vektor-Notation}}$$

mit einem Inkrementvektor $d \in \mathbb{R}^n$

Intervall im \mathbb{R}^n :

$$[x^0, x^0 + d] = \{x^0 + td, \quad 0 \leq t \leq 1\}$$

Hilfsmittel

Mittelwertsätze

$f \in C^1(\mathcal{D})$

$$\begin{aligned} f(x^0 + d) &= f(x^0) + \nabla f(x^0)^T d + o(\|d\|) \quad *) \\ f(x^0 + d) &= f(x^0) + \nabla f(x^0 + \vartheta d)^T d \quad \text{falls } f \text{ skalar, } 0 < \vartheta < 1 \\ f(x^0 + d) &= f(x^0) + \left(\underbrace{\int_0^1 \nabla f(x^0 + td)^T dt}_{\text{Integral ist komponentenweise zu nehmen}} \right) d \end{aligned}$$

Taylorentwicklung

$f \in C^2(\mathcal{D}), x^0 \in \mathcal{D}$

$$\begin{aligned} f(x^0 + d) &= f(x^0) + \nabla f(x^0)^T d + \frac{1}{2} d^T \nabla^2 f(x^0) d + o(\|d\|^2) \quad *) \\ f(x^0 + d) &= f(x^0) + \nabla f(x^0)^T d + \frac{1}{2} d^T \nabla^2 f(x^0 + \vartheta d) d \quad \text{falls } f \text{ skalar} \\ f(x^0 + d) &= f(x^0) + \nabla f(x^0)^T d + d^T \left(\int_0^1 (1-t) \nabla^2 f(x^0 + td) dt \right) d \end{aligned}$$

*) $o(\cdot)$ Landau-Symbol (klein-o)

$o(1)$ bezeichnet eine Größe, die bei einem (in der Regel implizit definierten) Grenzübergang gegen null geht. $o(\|d\|^k)$ bezeichnet eine Größe, die schneller gegen null geht als $\|d\|^k$, d.h.

$$o(\|d\|^k) / \|d\|^k \rightarrow 0 \quad \text{für } d \rightarrow 0.$$

$\mathcal{O}(h^n)$ bezeichnet eine Grösse mit

$$\mathcal{O}(h^n) \leq Ch^n$$

für einen definierten Grenzübergang von h , hier in der Regel $h \rightarrow 0$. $\mathcal{O}(1)$ eine beschränkte Grösse usw.

Taylorformel allgemeiner: Ist f eine k -mal stetig partiell ableitbare Funktion des Vektors x dann gilt

$$\begin{aligned} f(x+h) &= f(x) + \sum_{i=1}^n \left(\frac{\partial}{\partial x_i} f \right)(x) h_i + \\ &\quad \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \left(\frac{\partial^2}{\partial x_i \partial x_j} \right) f(x) h_i h_j + \\ &\quad \dots \dots \\ &\quad \frac{1}{k!} \sum_{i_1=1}^n \dots \sum_{i_k=1}^n \left(\frac{\partial^k}{\partial x_{i_1} \dots \partial x_{i_k}} \right) f(x + \theta_{f,h}) \prod_{j=1}^k h_{i_j} \end{aligned}$$

Ist f ein Vektorfeld, dann muss diese Taylorformel komponentenweise auf die einzelnen Komponentenfunktionen angewendet werden. Den letzten Summanden könnte man mit $\mathcal{O}(\|h\|^k)$ abkürzend angeben.

Formeln, Rechnen mit $\frac{d}{d\sigma}$, ∇ bei Vektorfunktionen:

$$\frac{d}{d\sigma} f(x - \sigma d)|_{\sigma=0} = -(\nabla f(x))^T d$$

$$\frac{d^2}{(d\sigma)^2} f(x - \sigma d)|_{\sigma=0} = d^T \nabla^2 f(x) d$$

$$\nabla(f(x)g(x)) = g(x)\nabla f(x) + f(x)\nabla g(x)$$

$$\nabla(f(g(x))) = \nabla g(x)\nabla f(y)|_{y=g(x)}$$

Insbesondere für: $f(y) = y^T y$ ergeben sich

$$\nabla(\|g(x)\|^2) = 2(\nabla g(x))g(x)$$

$$\nabla^2(\|g(x)\|^2) = 2(\nabla g(x))(\nabla g(x))^T + 2 \sum_{i=1}^m g_i(x) \nabla^2 g_i(x) \text{ für } g: \mathbb{R}^n \rightarrow \mathbb{R}^m$$

Differentiation einer inversen Matrix nach einem Parameter:

$$\frac{d}{d\sigma} (A(\sigma))^{-1} = -(A(\sigma))^{-1} \left(\frac{d}{d\sigma} A(\sigma) \right) (A(\sigma))^{-1}$$

Kapitel 10

Zugang zu numerischer Software und anderer Information

Don't reinvent the wheel! Für die Standardaufgaben der Numerischen Mathematik gibt es inzwischen public domain Programme sehr guter Qualität, sodass es oft nur notwendig ist, mehrere solcher Module zusammenzufügen, um ein spezifisches Problem zu lösen. Hier wird eine Liste der wichtigsten Informationsquellen angegeben.

10.1 Softwarebibliotheken

In der Regel findet man im Netz bereits vorgefertigte Softwarelösungen, die meisten davon für akademischen Gebrauch kostenfrei: Die bei weitem grösste und wichtigste Quelle ist die

NETLIB

Dies ist eine Sammlung von Programmbibliotheken in f77, f90, c, c++ für alle numerischen Anwendungen:

<http://www.netlib.org/>

Man kann nach Stichworten suchen ("search") und bekommt auch Informationen aus dem NaNet (Numerical Analysis Net)

Die Bibliotheken findet man unter "browse repository".

Die wichtigsten Bibliotheken sind:

1. amos, specfunc, cephes: spezielle Funktionen
2. ellpack : elliptische Randwertprobleme
3. fftpack : schnelle diskrete Fouriertransformation

156 KAPITEL 10. ZUGANG ZU NUMERISCHER SOFTWARE UND ANDERER INFORMATION

4. fitpack , dierckx: Spline Approximation und Interpolation
5. lapack, clapack, lapack90
die gesamte numerische Lineare Algebra (voll besetzte und Band-Matrizen) inklusive Eigenwertprobleme und lineare Ausgleichsrechnung in sehr guter Qualität
6. linpack , eispack : die Vorläufer von lapack. Einige der Verfahren aus diesen Bibliotheken wurden jedoch nicht in lapack übernommen.
7. lanz, lanczos : Eigenwerte/Eigenvektoren grosser dünn besetzter symmetrischer Matrizen
8. pdes/cwa : hyperbolische Erhaltungsgleichungen
9. templates : Iterationsverfahren für lineare Gleichungssysteme.
10. toms: Transactions on mathematical software. Sammlung von Algorithmen für verschiedene Aufgaben, sehr gute Qualität. u.a. automatische Differentiation, Arithmetik beliebiger Genauigkeit, mehrere Optimierungscodes, Nullstellenbestimmung, cubpack (Kubatur), partielle Differentialgleichungen
11. linalg: Iterative Verfahren für lineare Systeme, sonstige lineare Algebra
12. quadpack: Quadratur (bestimmte Integrale, 1-dimensional)
13. ode, odepack: numerische Integration von gewöhnlichen Differentialgleichungen, auch Randwertaufgaben.
14. fishpack: Lösung der Helmholtzgleichung mit Differenzenverfahren
15. opt,minpack1: Optimierungssoftware (nur ein kleiner Teil, s.u.)
16. slatec: eine eigenständige Bibliothek mit vielen wichtigen Lösern, u.a. ein Simplexverfahren für grosse dünn besetzte Probleme.

Daneben

<http://elib.zib.de/>

Dort gibt es auch Bibliotheken, teilweise mit guten Eigenentwicklungen der Gruppe um P. Deuffhard (die Codelib, mit Codes für das gedämpfte Newton- und Gauss-Newtonverfahren, dem System Kaskade zur Lösung elliptischer Gleichungen etc), sowie sonstige weitere Verweise. Die Programme aus Hairer-Norsett-Wanner (Integration gewöhnlicher DGLen I,II) findet man bei

<http://www.unige.ch/math/folks/haier>

10.2 Information über Optimierungssoftware

unter

<http://plato.la.asu.edu/guide.html>

Dort findet man eine vollständige Liste von frei verfügbarer Software für fast alle Bereiche der Optimierung und viele weitere Verweise.

10.3 Suchen nach software

Wenn der Name des Programmmoduls bekannt ist, kann man mit

xarchie

suchen, sonst benutzt man sinnvollerweise zuerst den Dienst

<http://math.nist.gov/HotGAMS/>

Dort öffnet sich ein Suchmenü, wo man nach Problemklassen geordnet durch einen Entscheidungsbaum geführt wird bis zu einer Liste verfügbarer Software (auch in den kommerziellen Bibliotheken IMSL und NAG). Falls der code als public domain vorliegt, wird er bei "Anklicken" sofort geliefert.

10.4 Andere wichtige Quellen

Wichtig für Ingenieursanwendungen: Die Finite-Element-Resources web-page von Ian MacPhedran:

http://www.engr.usask.ca/~macphed/finite/fe_resources/fe_resources.html

Dort gibt es viele Links, auch zu freiem FEM-Code, u.a. das Felt - System .

Ebenso:

<http://www.dealii.org>

mit C++code fuer adaptive finite Element-Berechnungen in 1D, 2D und 3D. Die Lösung grosser, auch unsymmetrischer Eigenwertprobleme leistet ARPACK

<http://www.caam.rice.edu/software/ARPACK>

Software für C++ findet man unter

<http://oonumerics.org/oon/>

Software vielfältiger Form für die schnelle Fouriertransformation findet man ausser in der Netlib auch unter

<http://www.fftw.org>

10.5 Hilfe bei Fragen

Hat man Fragen, z.B. nach Software, Literatur oder auch zu spezifischen mathematischen Fragestellungen, kann man in einer der Newsgroups eine Anfrage plazieren. Häufig bekommt man sehr schnell qualifizierte Hinweise. Zugang zu Newsgroups z.B. über

xrn

. mit "subscribe" . Die wichtigsten News-Groups sind hier

sci.math.num-analysis
sci.op-research

Es gibt natürlich auch im Bereich Informatik bzw. Software und Ingenieurwissenschaften eine Fülle solcher News-groups.

Im xrn-Menu kann man mit "post" ein Anfrage abschicken und dabei die Zielgruppe frei wählen.

Index

- Ähnlichkeitstransformation, 18
- Courant'sches Minimax-Prinzip, 14
- A-konjugiert, 92
- A-orthogonal, 92
- allgemeines Eigenwertproblem, 55
- Bauer-Fike, 16
- Bitreversion, 147
- Eigenwerte, tridiagonal, 22
- Einheitswurzel, primitive, 141
- Einzelkonditionszahlen, 132
- Einzelschritt-Verfahren, 71, 78, 79, 82, 87
- Einzelsensitivität von Eigenwerten, 17
- Exponentenüberlauf, 123
- Exponentenunterlauf, 123
- Fehlerfortpflanzung, 126
- Gesamtkonditionszahl, 132
- Gesamtschritt-Verfahren, 70, 78, 79
- Givens-Rotation, 46
- Gleitpunktzahl, 121
- GMRES, 105
- Golub und Kahan, 63
- Graph, zugeordneter gerichteter, 76
- Graph, zusammenhängender, 76
- Hessenberg-Form, 18
- IEEE-Darstellung, 122
- Interpolation, komplexe, auf dem Einheitskreis, 142
- Interpolation, trigonometrische, 144
- Intervallarithmetik, 138
- irreduzibel, 76
- irreduzibel diagonaldominant, 77
- Iteration v. Mises, 28
- Jacobi-Verfahren, 70
- Kaczmarz-Verfahren, 108
- Konditionszahlen, 132
- konsistent geordnet, 86
- Kreisesatz von Gerschgorin, 8
- Krylow-Unterraum, 104
- L-Matrix, 82
- Lanczos-Verfahrens, 48
- Lokalisierung von Eigenwerten, 7
- Maschinenarithmetik, 123
- Maschinengenauigkeit, 123
- Maschinenzahl, 122
- Minimierungsverfahren, 92
- Moore-Penrose-Pseudoinverse, 65
- Norm, absolute, 16
- Poisson-Gleichung, 67
- präkonditionierte cg-Verfahren, 103
- QR-Verfahren, 35
- QZ-Algorithmus, 56
- Rückwurfanalyse, 133
- Rayleighquotienten, 11
- reguläre Aufspaltung, 82
- Residuen, minimale, generalisierte, 105
- Rundung, 122
- Schur, Transformation, 35
- Sensitivität des Eigenwertproblems, 7
- SOR-Verfahren, 71
- starkes Spaltensummenkriterium, 75
- starkes Zeilensummenkriterium, 75
- Stein-Rosenberg-Theorem, 79
- Sylvester, 22

Trägheitssatz, 22

Tridiagonalform, 18

Tridiagonalmatrix, 22

Verallgemeinerung der Wielandtiteration, 59

Verfahren von Wielandt, 28

Vorwärtsanalyse, 128

Wielandtverfahren, 31

Wilkinson'schen Shift-Technik, 47