Otmar Scherzer

Numerische Mathematik

Vorlesungsskriptum SS 2010

Computational Science Center
Universität Wien
Nordbergstr. 15 1090 Wien

Inhaltsverzeichnis

Dank und Literaturstellen

Diese Vorlesungsskriptum beruht auf dem Buch von Prof. Martin Hanke (Universität Mainz) [7].

Die Literatur zur numerischen Mathematik ist äußerst umfangreich. An dieser Stelle sei auf einige kürzlich erschienene oder wieder aufgelegte Bücher zur Numerischen Mathematik hingewiesen, ohne aber einen Anspruch auf Vollständigkeit zu erheben [10, 12, 11, 9, 3, 2, 6] .

Kapitel 1

Rundefehler, Kondition und Stabilität

1.1 Rundefehler

Eine Fehlerquelle bei der Implementierung jedes numerischen Algorithmus sind Daten - und **Rundefehler**.

Während der Rundefehler in jeder einzelnen **Elementaroperation** (Addition, Multiplikation, Standardfunktionen, ...) in der Regel vernachlässigt werden kann, kann es in einem Algorithmus zu einer Kumulation der Fehler führen, und sich problematisch auf das berechnete Ergebnis auswirken. Man spricht von einem **stabilen** Algorithmus, wenn keine problematische Fehlerverstärkung auftritt.

Zur genauen Untersuchung des Einflusses von Rundefehlern verwendet man ein Modell, dass jeder Elementaroperation auf dem Rechner die nächstgelegene Maschinenzahl zuordnet:

$$a(!\circ)b = \mathrm{Rnd}(a \circ b) \ .$$

Hierbei ist $\circ$ die mathematische Grundoperation und $(!\circ)$ die entsprechende Realisierung am Rechner. Die Operation $\mathrm{Rnd}(x)$ bezeichnet die Rundung von x zur nächstgelegenen Maschinenzahl.

Wir treffen eine (idealisierte) Annahme, dass die Rundung den **tatsächlichen** Wert mit einer bestimmten relativen Genauigkeit bestimmt,

$$\mathrm{Rnd}(x) = x(1+\varepsilon) \text{ mit } |\varepsilon| \leq \mathbf{eps} \ . \tag{1.1}$$

Hierbei ist **eps** die so genannte **Maschinengenauigkeit**, die wie folgt definiert ist:

$$\mathbf{eps} := \inf\{|x| : 1(!-)x \neq 1\} .$$

Der genaue Wert ist dabei vom Rechner abhängig. In der Regel ist **eps** eine negative Potenz von 2, also 2^{-d}. Insgesamt ergibt sich daher eine Modell für die Realisierung der Elementaroperationen am Rechner

$$a(!\circ)b = \mathrm{Rnd}(a \circ b) = (a \circ b)(1+\varepsilon) \text{ mit } |\varepsilon| \leq \mathbf{eps} . \tag{1.2}$$

Dieses Modell wird unrealistisch, wenn entweder das exakte oder das gerundete Ergebnis 0 wird, wie wir an folgendem Beispiel demonstrieren.

Beispiel 1.1. Sei $a = 5.5$ und $b = 5.5001$. Angenommen, die Maschinengenauigkeit ist **eps**$= 0.01$, dann gilt $a(!-)b = 0$ und somit kann (1.2), was in diesem Fall lautet

$$0 = a(!-)b = (a \circ b)(1+\varepsilon) = 0.0001(1+\varepsilon) ,$$

für kein ε gelten. In diesem Fall versagt also die getroffene Modellannahme (1.1) und jede darauf basierende Stabilitätsanalyse. Man spricht in diesem Fall von einem **underflow**. Ein guter Compiler wird bei einem underflow eine Warnung ausgeben.

Im Folgenden führen wir den Begriff der **Kondition** eines Problems ein: Sei F eine reellwertigen Funktion von n reellen Variablen x_i (zusammengefasst in einem Vektor $x \in \mathbb{R}^n$). Wir wollen die Auswertung der Funktion F an einem vorgegebenen Vektor x berechnen, d.h., wir wollen

$$y = F(x)$$

berechnen. Aufgrund von Datenfehlern oder fortgepflanzten Rundefehlern wird nun die Auswertung der Funktion F nicht an der Stelle x, sondern an der Stelle $\tilde{x} = x + \Delta x$ ausgewertet. Wie wirkt sich dieser Eingangsfehler auf die Auswertung der Funktion F aus?

Bezeichnen wir mit $\Delta y := F(x + \Delta x) - F(x)$ den Fehler im Ergebnis, so gilt, falls $F \in C^1(\mathbb{R}^n)$ ist nach dem Mittelwertsatz

$$\begin{aligned} \Delta y &= F(x_1 + \Delta x_1, x_2 + \Delta x_2, ..., x_n + \Delta x_n) - F(x_1, x_2, ..., x_n) \\ &= \sum_{i=1}^{n} \frac{\partial F}{\partial x_i}(\zeta)\Delta x_i , \end{aligned}$$

wobei ζ auf der Strecke zwischen x und $x + \Delta x$ liegt. Ist die Ableitung Lipschitz-stetig, dann gilt sogar

$$\Delta y = \sum_{i=1}^{n} \frac{\partial F}{\partial x_i}(x)\Delta x_i + O(\varepsilon^2)\,, \tag{1.3}$$

wobei $\varepsilon := \max_{i=1,\dots,n} |\Delta x_i|$ gesetzt wird.[1] Um sich die Analyse zu vereinfachen, vernachlässigt man den $O(\varepsilon^2)$-Term und verwendet $(\mathcal{K}_{\text{abs}})_{i=1,\dots,n} := \left|\frac{\partial F}{\partial x_i}(x)\right|$ als ein Maß für die Verstärkung des Fehlers.

Üblicherweise ist die relative Fehlerverstärkung von größerer Bedeutung als die absolute Fehlerverstärkung. Ein Maß für die relative Fehlerverstärkung ergibt sich aus (1.3) unter Vernachlässigung des $O(\varepsilon^2)$ Terms wie folgt:

$$\begin{aligned} \frac{\Delta y}{y} &= \sum_{i=1}^{n} \frac{\partial F}{\partial x_i}(x)\frac{\Delta x_i}{F(x)} \\ &= \sum_{i=1}^{n} \frac{\partial F}{\partial x_i}(x)\frac{x_i}{F(x)}\frac{\Delta x_i}{x_i}\,. \end{aligned} \tag{1.4}$$

Definition 1.2. $\mathcal{K}_{\text{abs}} = \left(\left|\frac{\partial F}{\partial x_i}(x)\right|\right)_{i=1,\dots,n}$ heißt der Vektor der **absoluten Konditionszahlen**. Der Vektor $\mathcal{K}_{\text{rel}} = \left(\left|\frac{\partial F}{\partial x_i}(x)\frac{x_i}{F(x)}\right|\right)_{i=1,\dots,n}$ heißt der Vektor der **relativen Konditionszahlen**.

Die Konditionszahlen beschreiben also die Verstärkung der absoluten bzw. relativen Eingangsdaten bei der Auswertung der Funktion F. Ein Problem heißt **schlecht konditioniert**, wenn einer der beiden Maxima der Konditionsvektoren signifikant größer als 1 ist. Ansonst nennt man das Problem **gut konditioniert**.

Eine Verallgemeinerung auf mehrdimensionale Funktionen $F : \mathbb{R}^n \to \mathbb{R}^m$ ist leicht möglich, wenn man die einzelnen Komponenten der Funktion betrachtet.

Anhand von zwei Beispielen erläutern wir den Konditionsbegriff:

Beispiel 1.3. Addition: Sei $F(x) = x_1 + x_2$, $x = [x_1, x_2]^t$. Dann gilt für die relativen Konditionszahlen

$$(\mathcal{K}_{\text{rel}})_i = \left|\frac{\partial F}{\partial x_i}(x)\frac{x_i}{x_1 + x_2}\right| = \left|\frac{x_i}{F(x)}\right|\,, \quad i = 1, 2\,.$$

[1] Wir verwenden die Notation $a_\varepsilon = O(\varepsilon)$, wenn für ein $\varepsilon_0 >$ eine von ε unabhängige Konstante $C > 0$ existiert, so dass für alle $\varepsilon \in (0, \varepsilon_0)$ die Ungleichung $|a_\varepsilon| \le C\varepsilon$ gilt.

Die relativen Konditionszahlen sind somit groß, wenn für $i = 1$ oder $i = 2$ der Betrag von $F(x) = x_1 + x_2$ sehr viel kleiner als der Betrag von x_i ist. Dieses Phänomen bezeichnet man als **Auslöschung**.

Beispielsweise ergibt sich für

$$\begin{aligned} x_1 = 1.000001\,, \quad x_2 &= -1\,, \\ \Delta x_1 = 0.001\,, \quad \Delta x_2 &= 0\,, \end{aligned}$$

$$x_1 + x_2 = 0.000001\,, \quad (x_1 + \Delta x_1) + (x_2 + \Delta x_2) = 0.001001\,.$$

Der absolute Fehler 0.001001 ist also fast gleich groß wie die Fehler in den Daten. Der relative Fehler (= 0.001001/0.000001) ist also um einen Faktor 10^6 größer.

Multiplikation: Wir betrachten die Multiplikation zweier Zahlen, $F(x) = ax$. Dabei sei a ein fester Parameter und x die einzige Eingangsgröße: In diesem Fall lautet die absolute Konditionszahl

$$\mathcal{K}_{\text{abs}} = |F'(x)| = a\,.$$

Die absolute Kondition ist daher schlecht, wenn $|a|$ sehr viel größer als 1 ist – in diesem Fall ergibt sich also eine starke absolute Fehlerverstärkung. Der relative Fehler bleibt allerdings gleich ($\mathcal{K}_{\text{rel}} = 1$).

Betrachten wir nun die Implementierung eines Algorithmus $!f$ als Realisierung einer reellen Funktion f. Im Verlauf des Algorithmus müssen auf jeden Fall Rundefehler mit einer relativen Genauigkeit **eps** in Kauf genommen werden, d.h. $\left|\frac{\Delta x}{x}\right| \leq \mathbf{eps}$. Man kann daher nicht erwarten, dass die Genauigkeit des Ergebnisses besser ist, was nach (1.4) bedeutet, dass

$$\left|\frac{!f(x) - f(x)}{f(x)}\right| \sim \left|f'(x)\frac{x}{f(x)}\right| \left|\frac{\Delta x}{x}\right| \leq C_V |\mathcal{K}_{\text{rel}}| \mathbf{eps}\,, \tag{1.5}$$

mit einer nicht zu großen Konstanten $C_V > 0$.

Diese Form der Stabilitätsanalyse nennt man **Vorwärtsanalyse** und der Algorithmus $!f$ heißt **vorwärts stabil**, wenn (1.5) erfüllt ist. Insbesondere impliziert die Annahme (1.2), dass die Grundoperationen ($!\circ$) vorwärts stabil sind. Die Definition (1.5) gilt natürlich auch, falls die Funktion f mehrdimensional ist.

Die Rückwärtsanalyse betrachten wir jetzt gleich für eine mehrdimenionale Funktion, d.h., $F : \mathbb{R}^n \to \mathbb{R}$. Wir verwenden dabei die Notation, dass die Multiplikation und Division von Vektoren komponentenweise ist. Bei der **Rückwärtsanalyse** interpretiert man die berechnete Näherung als exakte Lösung eines Problems mit gestörten Eingangsdaten, also $!F(x) = F(x+\Delta x)$ und untersucht die Größe $|\Delta x|$. Gibt es mehrere Urbilder $x + \Delta x$, so nimmt man traditionell eines mit betragskleinster Störung Δx. Gilt dann

$$\left\| \frac{\Delta x}{x} \right\| \leq C_R \mathbf{eps} \, ,$$

wobei $\|y\| = \sqrt{\sum_{i=1}^n y_i^2}$ die Euklidsche Norm ist, und C_R nicht zu groß ist, so nennen wir den Algorithmus $!F$ **rückwärts stabil**. Gibt es kein Urbild $x + \Delta x$, dann ist $!F$ **nicht** rückwärts stabil. Für einen rückwärts stabilen Algorithmus ergibt sich nach (1.4) und (1.6) mit $\tilde{x} = x + \Delta x$

$$\begin{aligned} \frac{|!F(x) - F(x)|}{|F(x)|} &= \frac{|F(\tilde{x}) - F(x)|}{|F(x)|} \\ &\leq \|\mathcal{K}_{\text{rel}}\| \left\| \frac{\tilde{x} - x}{x} \right\| \\ &\leq \|\mathcal{K}_{\text{rel}}\| C_R \mathbf{eps} \, . \end{aligned}$$

Folglich ist jeder rückwärts stabile Algorithmus auch vorwärts stabil, wenn man $C_R = C_V$ setzt. Die Umkehrung ist jedoch **nicht** richtig.

In der Folge werden wir Abschätzungen für den "relativen Fehler"

$$\frac{\|\Delta x\|}{\|x\|} \tag{1.6}$$

herleiten. Im Allgemeinen wird dieser Term als ein Maß für die Fehlerverstärkung von Algorithmen herangezogen und nicht $\left\| \frac{\Delta x}{x} \right\|$, da der Ausdruck (1.6) leichter zu handhaben ist. Im Fall, dass $\|\cdot\|$ die Euklidnorm ist, ist dieser Term stets kleiner als $\left\| \frac{\Delta x}{x} \right\|$.

1.2 Vektor und Matrixnormen

Wir betrachten sowohl reelle als auch komplexe Vektoren und Matrizen. In den nachfolgenden Betrachtungen ist es zumeist unerheblich, ob die zugehörigen Einträge reell oder komplex sind. Für diesen Fall schreiben wir der

Einfachheit $\mathbb{K}$ für den entsprechenden Zahlenkörper und meinen damit, dass die entsprechenden Resultate in gleicher Weise in $\mathbb{R}$ und $\mathbb{C}$ gelten.

Entsprechend bezeichnet $\mathbb{K}^n$ den Raum der n-dimensionalen Vektoren über $\mathbb{K}$,

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, \quad x_i \in \mathbb{K},$$

und $\mathbb{K}^{m\times n}$ den Raum der $m \times n$ Matrizen über $\mathbb{K}$,

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}, \quad a_{ij} \in \mathbb{K}\,.$$

Ist $x \in \mathbb{K}^n$, so unterscheiden wir zwischen

$$x^t = [x_1, x_2, ..., x_n] \in \mathbb{K}^{1\times n} \text{ und } x^* = [\overline{x}_1, \overline{x}_2, ..., \overline{x}_n] \in \mathbb{K}^{1\times n};$$

bei x^* sind die Einträge konjugiert komplex. Für $\mathbb{K} = \mathbb{R}$ stimmen x^* und x^t überein. A^t und A^* in $\mathbb{K}^{n\times m}$ sind entsprechend definiert.

Im Raum $\mathbb{K}^n$ greifen wir gelegentlich auf die **kartesische Basis** $\{e_1, ..., e_n\}$ zurück, wobei $e_i = [\delta_{ij}]_{j=1}^n$ den Vektor bezeichnet, der in der i-ten Komponente eine Eins und ansonsten nur Nulleinträge enthält.

$$\delta_{ij} = \begin{cases} 1, & i = j\,, \\ 0, & i \neq j\,, \end{cases}$$

ist das **Kronecker-Symbol**.

Definition 1.4. Sei $X = \mathbb{K}^n$ oder $X = \mathbb{K}^{m\times n}$. Eine Abbildung $\|\cdot\| : X \to \mathbb{R}_0^+$ heißt **Norm**, wenn die drei Eigenschaften

1. $\|x\| > 0$ für alle $x \in X\backslash\{0\}$,
2. $\|\alpha x\| = |\alpha|\|x\|$ für alle $x \in X$ und $\alpha \in \mathbb{K}$,
3. $\|x + y\| \leq \|x\| + \|y\|$ für alle $x, y \in X$

erfüllt sind.

Jede Norm in $X = \mathbb{K}^n$ oder $X = \mathbb{K}^{m\times n}$ induziert eine Metrik

$$d(x, y) = \|x - y\| \text{ für alle } x, y \in X\,,$$

und damit einen Konvergenzbegriff: Die Folge $\{x_k\} \subseteq X$ ist konvergent mit Grenzelement $x \in X$ genau dann, wenn $d(x_k, x)$ für $k \to \infty$ gegen Null konvergiert. Die Wahl dieses Konvergenzbegriffes ist in $X = \mathbb{K}^n$ oder $X = \mathbb{K}^{m\times n}$ **nicht** von der Wahl der Norm abhängig. Zum Beweis dieser Behauptung verwenden wir das folgende fundamentale Resultat aus der linearen Algebra:

Satz 1.5. *Alle Normen auf $\mathbb{K}^n$ und $\mathbb{K}^{m\times n}$ sind äquivalent, d.h., für je zwei Normen $\|\cdot\|_a$ und $\|\cdot\|_b$ auf X gibt es positive Konstanten $c, C > 0$ mit*

$$c\|x\|_a \le \|x\|_b \le C\|x\|_a \text{ für alle } x \in X.$$

Beweis. Siehe [4]. □

Korollar 1.6. *Sei $\{x_k\}$ eine Folge in $X = \mathbb{K}^n$ oder $X = \mathbb{K}^{m\times n}$. Dann existiert entweder ein $x \in X$ und x_k konvergiert gegen x für $k \to \infty$ oder die Folge ist nicht konvergent (in jeder beliebigen Norm).*

Folgt aus Satz 1.5.

Beispiel 1.7. Die am häufigsten verwendeten Normen in $X = \mathbb{K}^n$ sind die

1. **Betragssummennorm:** $\|x\|_1 := \sum_{i=1}^n |x_i|\,,$
2. **Euklidnorm:** $\|x\|_2 := \sqrt{\sum_{i=1}^n |x_i|^2} = \sqrt{x^*x}\,,$
3. **Maximumnorm:** $\|x\|_\infty := \max_{i=1,\dots,n} |x_i|\,.$

Häufigste verwendete Normen in $X = \mathbb{K}^{m\times n}$ sind die

1. **Spaltensummennorm:** $\|A\|_1 := \max_{1\le j\le n} \sum_{i=1}^m |a_{i,j}|$: Maximum über die Spalten der Summen der Beträge der Zeileneinträge.
2. **Zeilensummennorm:** $\|A\|_\infty := \max_{1\le i\le m} \sum_{j=1}^n |a_{i,j}|$: Maximum über die Zeilen der Summe der Beträge der Spalteneinträge.
3. **Frobeniusnorm:** $\|A\|_F := \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{i,j}|^2}\,.$

Der (quadratische) Vektorraum $\mathbb{K}^{n\times n}$ unterscheidet sich von den anderen Räumen dadurch, dass noch Multiplikation AB für $A, B \in \mathbb{K}^{n\times n}$ wohldefiniert ist.

Definition 1.8. Eine Matrixnorm $\|\cdot\|_M$ auf $\mathbb{K}^{n\times n}$ heißt **submultiplikativ**, falls

$$\|AB\|_M \leq \|A\|_M \|B\|_M \text{ für alle } A, B \in \mathbb{K}^{n\times n} .$$

Eine Matrixnorm $\|\cdot\|_M$ auf $\mathbb{K}^{n\times n}$ heißt **verträglich** mit der Vektornorm $\|\cdot\|$ auf $\mathbb{K}^n$, falls

$$\|Ax\|_M \leq \|A\|_M \|x\| \text{ für alle } A \in \mathbb{K}^{n\times n} \text{ und alle } x \in \mathbb{K}^n .$$

Beispiel 1.9. 1. $\|A\| := \max_{1\leq i,j\leq n} |a_{ij}|$ ist eine Matrixnorm auf $\mathbb{K}^{n\times n}$, aber nicht submultiplikativ. Für

$$A = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \text{ ist } A^2 = \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix} .$$

Aber es ist $\|A^2\| = 2 \not\leq \|A\|^2 = 1$.

Die Frobeniusnorm ist mit der Euklidnorm verträglich: Dazu verwenden wir zuerst die Cauchy-Schwarzsche Ungleichung und erhalten

$$|(Ax)_i|^2 = \left|\sum_{j=1}^{n} a_{ij}x_j\right|^2 \leq \sum_{j=1}^{n} |a_{ij}|^2 \sum_{j=1}^{n} |x_j|^2 = \sum_{j=1}^{n} |a_{ij}|^2 \|x\|_2^2 .$$

Daraus folgt:

$$\sum_{i=1}^{m} |(Ax)_i|^2 \leq \sum_{i=1}^{m}\sum_{j=1}^{n} |a_{ij}|^2 \|x\|_2^2 = \|A\|_F^2 \|x\|_2^2 .$$

Definition und Satz 1.10. *Sei $\|\cdot\|$ eine Vektornorm auf $\mathbb{K}^n$. Dann ist*

$$|||A||| := \sup_{x\neq 0} \frac{\|Ax\|}{\|x\|} = \max_{\|x\|=1} \|Ax\|$$

eine Norm auf $\mathbb{K}^{n\times n}$ – die durch $\|\cdot\|$ **induzierte Norm**.

Die Normeigenschaften sind dabei leicht nachgerechnet.

Beispiel 1.11. Sei $A \in \mathbb{K}^{n\times n}$ und $x \in \mathbb{K}^n$. Dann gilt

$$\begin{aligned}
\|Ax\|_1 &= \text{(Spaltensummennorm)} \\
&= \sum_{i=1}^{n} |(Ax)_i| \\
&= \sum_{i=1}^{n} \left| \sum_{j=1}^{n} a_{ij}x_j \right| \\
&\leq \sum_{i=1}^{n} \sum_{j=1}^{n} |a_{ij}||x_j| \\
&\leq \sum_{j=1}^{n} |x_j| \sum_{i=1}^{n} |a_{ij}| \\
&\leq \sum_{j=1}^{n} |x_j| \max_{j=1,\ldots,n} \sum_{i=1}^{n} |a_{ij}| \\
&= \|x\|_1 \|A\|_1 \, .
\end{aligned}$$

Also ist die Spaltensummennorm mit der Betragssummennorm verträglich; das bedeutet

$$\frac{\|Ax\|_1}{\|x\|_1} \leq \|A\|_1 \text{ für alle } x \neq 0 \, . \tag{1.7}$$

Wir wollen nun zeigen, dass $\|A\|_1$ die kleinstmögliche Schranke ist, so dass für alle x (1.7) gilt. Dazu werden wir ein $0 \neq x \in \mathbb{K}^n$ suchen, so dass in (1.7) Gleichheit gilt. Wir wählen nun den Spaltenindex j, für den

$$\|A\|_1 = \sum_{i=1}^{n} |a_{ij}|$$

gilt und setzen für x den j-ten kartesischen Basisvektor e_j. Dann gilt

$$\|A\|_1 = \|Ae_j\|_1 = \frac{\|Ae_j\|_1}{\|e_j\|_1} \, .$$

In ähnlicher Weise zeigt man, dass die Zeilensummennorm durch die Maximumnorm induziert wird.

Lemma 1.12. *Die durch $\|\cdot\|$ induzierte (Matrix–) Norm $|||\cdot|||$ ist submultiplikativ und ist mit der Ausgangsnorm verträglich. Ist $\|\cdot\|_M$ eine andere mit $\|\cdot\|$ verträgliche Norm, dann gilt $|||A||| \leq \|A\|_M$ für alle $A \in \mathbb{K}^{n\times n}$.*

Beweis. • Für $B \neq 0$ gilt

$$\begin{aligned}
|||AB||| &= \sup_{x\neq 0} \frac{\|ABx\|}{\|x\|} \\
&= \sup_{Bx\neq 0} \left(\frac{\|ABx\|}{\|Bx\|} \frac{\|Bx\|}{\|x\|} \right) \\
&\leq \sup_{Bx\neq 0} \frac{\|ABx\|}{\|Bx\|} \sup_{Bx\neq 0} \frac{\|Bx\|}{\|x\|} \\
&\leq \sup_{y\neq 0} \frac{\|Ay\|}{\|y\|} \sup_{x\neq 0} \frac{\|Bx\|}{\|x\|} \\
&= |||A||| \, |||B||| \ .
\end{aligned}$$

Folglich ist die induzierte Norm submultiplikativ.

- Die Verträglichkeit mit der Ausgangsnorm folgt unmittelbar aus der Definition: Demnach ist nämlich

$$|||A||| = \sup_{x\neq 0} \frac{\|Ax\|}{\|x\|} \geq \frac{\|Ax\|}{\|x\|} \text{ für jedes } x \neq 0 \,,$$

beziehungsweise $\|Ax\| \leq |||A||| \, \|x\|$.

- Sei $\|\cdot\|_M$ eine andere mit $\|\cdot\|$ verträgliche Norm. Nach Definition 1.10 gilt $|||A||| = \|Ax\|$ für ein gewisses $x \in \mathbb{K}^n$ mit $\|x\| = 1$, und aus der Verträglichkeit der zweiten Matrixnorm folgt daher

$$|||A||| = \|Ax\| \leq \|A\|_M \|x\| = \|A\|_M \ .$$

□

Die vermutlich wichtigste Norm in $\mathbb{K}^n$ ist die Euklidnorm. Wir werden uns nun mit der durch die Euklidnorm induzierten Matrixnorm in $\mathbb{K}^{m\times n}$ (muss nicht notwendigerweise eine quadratische Matrix sein) beschäftigen. Diese induzierte Norm heißt **Spektralnorm**

$$\begin{aligned}
\|A\|_2 &:= \max_{\|x\|_2=1} \|Ax\|_2 \\
&= \max_{\|x\|_2=1} \sqrt{(Ax)^*(Ax)} \\
&= \max_{\|x\|_2=1} \sqrt{x^* A^* A x} \ .
\end{aligned} \tag{1.8}$$

Die Spektralnorm ist nicht mit dem **Spektralradius** $\rho(A)$ einer quadratischen Matrix $A \in \mathbb{K}^{n\times n}$ zu verwechseln: Ist $\sigma(A)$ die Menge aller Eigenwerte von A, dann ist der Spektralradius gegeben durch

$$\rho(A) := \max\{|\lambda| : \lambda \in \sigma(A)\}\,, \tag{1.9}$$

also dem betragsgrößtem Eigenwert der Matrix A. Zwischen Spektralradius und Spektralnorm besteht folgender Zusammenhang:

Satz 1.13. *Für jede Matrix $A \in \mathbb{K}^{m\times n}$ ist $\|A\|_2 = \sqrt{\rho(A^*A)}$.*

Beweis. A^*A ist hermitesch und positiv semidefinit[2], denn $x^*(A^*A)x = \|Ax\|_2^2 \geq 0$. Demnach existiert eine Orthonormalbasis $\{x_1, ..., x_n\}$ von $\mathbb{K}^n$ aus Eigenvektoren von A^*A[3] mit zugehörigen Eigenwerten

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0\,.$$

Jeder Vektor $x \in \mathbb{K}^n$ mit $\|x\|_2 = 1$ lässt sich in dieser Basis entwickeln, d.h., es existieren $\zeta_i \in \mathbb{K}$, $i = 1, \cdots n$, so dass $x = \sum_{i=1}^n \zeta_i x_i$ und

$$\begin{aligned} 1 = x^*x &= \sum_{i=1}^n \overline{\zeta_i} x_i^* \sum_{j=1}^n \zeta_j x_j \\ &= \sum_{i,j=1}^n \overline{\zeta_i}\zeta_j x_i^* x_j \\ &= \sum_{i,j=1}^n \overline{\zeta_i}\zeta_j \delta_{ij} \\ &= \sum_{i=1}^n |\zeta_i|^2\,; \end{aligned}$$

[2]Eine Matrix $B \in K^{n\times n}$ heißt hermitesch, falls $B^* = B$. Sie heißt positiv semidefinit falls $\|Bx\|_2^2 \geq 0$ für alle $x \in \mathbb{K}^n$.

[3]Wir wollen Eigenwerte immer so verstehen, dass die Euklidnorm auf 1 normiert ist

$$\begin{aligned} x^*A^*Ax &= \sum_{i=1}^{n} \overline{\zeta_i} x_i^* A^* A \sum_{j=1}^{n} \zeta_j x_j \\ &= \sum_{i,j=1}^{n} \overline{\zeta_i} x_i^* \zeta_j \lambda_j x_j \\ &= \sum_{i,j=1}^{n} \overline{\zeta_i} \zeta_j \lambda_j \delta_{ij} \\ &= \sum_{i=1}^{n} |\zeta_i|^2 \lambda_i \\ &\leq \lambda_1 \sum_{i=1}^{n} |\zeta_i|^2 \\ &= \lambda_1 \ . \end{aligned}$$

Mit anderen Worten: Es gilt $\max_{\|x\|_2=1} x^*A^*Ax \leq \lambda_1$. Es gilt aber auch für $x = x_1$

$$x_1^*A^*Ax_1 = x_1^*\lambda_1 x_1 = \lambda_1$$

und daher $\max_{\|x\|_2=1} x^*A^*Ax = \lambda_1 = \rho(A^*A)$. Zusammen mit (1.8) folgt daher die Behauptung. □

Satz 1.13 erklärt, warum die durch die Euklidnorm induzierte Matrixnorm Spektralnorm genannt wird! Allerdings ist nicht das Spektrum von A entscheidend, sondern das von A^*A. Die Berechnung der Spektralnorm benötigt den größten Eigenwert von A^*A und ist deshalb viel aufwendiger als die Berechnung der Zeilen- bzw. Spaltensummennorm. Für viele Anwendungen ist aber eine Abschätzung des größten Eigenwertes von A^*A ausreichend. Ein solche Abschätzung beweisen wir nun.

Satz 1.14. *Für $A \in \mathbb{K}^{m\times n}$ gilt $\|A\|_2 \leq \sqrt{\|A\|_1 \|A\|_\infty}$.*

Beweis. Nach Satz 1.13 ist $\|A\|_2^2$ der größte Eigenwert von A^*A. Sei x_1 ein zugehöriger Eigenvektor (mit $\|x_1\|_2 = 1$) und $\hat{x}_1 = x_1/\|x_1\|_1$. Insbesondere gilt $\|\hat{x}_1\|_1 = 1$. Da die Spaltensummennorm durch die Betragssummennorm

induziert in $\mathbb{K}^m$ und $\mathbb{K}^n$ induziert wird, gilt

$$\begin{aligned}\|A\hat{x}_1\|_2^2 &= \|A^*A\hat{x}_1\|_1 \\ &\leq \|A^*\|_1\|A\hat{x}_1\|_1 \\ &\leq \|A^*\|_1\|A\|_1\|\hat{x}_1\|_1 \\ &= \|A^*\|_1\|A\|_1 \ .\end{aligned}$$

Da $\|A^*\|_1 = \|A\|_\infty$ gilt, folgt die Behauptung. □

Eine wichtige Anwendung von Matrxinormen ergibt sich bei der Bestimmung der Kondition des Problems. Ein lineares Gleichungssystem zu lösen sei $A \in \mathbb{K}^{n\times n}$ nichtsingulär und Δb ein Eingangsfehler, dann gilt

$$x = A^{-1}b \text{ und } x + \Delta x = A^{-1}(b + \Delta b) = A^{-1}b + A^{-1}\Delta b \ .$$

Also ist der Fehler in der Lösung

$$\Delta x = A^{-1}\Delta b \ .$$

Sind die Matrixnorm $\|\cdot\|_M$ und die Vektornorm $\|\cdot\|$ verträglich, dann gilt

$$\begin{aligned}\frac{\|\Delta x\|}{\|x\|} &= \frac{\|A^{-1}\Delta b\|}{\|x\|} \\ &\leq \|A^{-1}\|_M \frac{\|\Delta b\|}{\|b\|}\frac{\|Ax\|}{\|x\|} \\ &\leq \|A^{-1}\|_M\|A\|_M \frac{\|\Delta b\|}{\|b\|} \ .\end{aligned} \tag{1.10}$$

Definition 1.15. Der Faktor

$$\mathrm{cond}_M(A) := \|A^{-1}\|_M\|A\|_M$$

wird als **Kondition** der Matrix A bzgl. der Matrixnorm $\|\cdot\|_M$ bezeichnet.

Aus der Abschätzung (1.10) sieht man, dass die Kondition einer Matrix A eine skalare Operation ist (ähnlich zur relativen Konditionszahl einer skalaren Operation). Die Abschätzung (1.10) ist scharf, in dem Sinn, dass man immer einen Fehlervektor angeben kann, so dass in der Ungleichung Gleichheit gilt.

Kapitel 2

Eliminiationsalgorithmen

Der grundlegende Baustein aller numerischen Algorithmen ist die Lösung linearer Gleichungssysteme. Es gibt eine große Anzahl von verschiedenen Lösungsmethoden für lineare Gleichungssysteme. Eine kleine Anzahl solcher Verfahren wird in dieser Vorlesung behandelt.

2.1 Die LR-Zerlegung

Der wichtigste Algorithmus zur Lösung linearer Gleichungssysteme ist der **Gauß-Algorithmus**, welcher implizit eine Zerlegung einer Koeffizientenmatrix A in zwei Dreiecksmatrizen bewirkt. Diese, sogenannte LR–Zerlegung werden wir jetzt studieren.

Wir betrachten zunächst einen beliebigen Vektor $x = [x_1, ..., x_n]^t \in \mathbb{K}^n$ und nehmen an, dass $x_k \neq 0$ ist. Mit e_k bezeichnen wir den k-ten kartesischen Einheitsvektor in $\mathbb{K}^n$. Schließlich sei

$$L_k = I - l_k e_k^* \,, \tag{2.1}$$

mit $l_k = [0, ..., 0, l_{k+1,k}, ..., l_{n,k}]^t \in \mathbb{K}^n$ mit $l_{jk} = x_j/x_k$, $j = k+1, ..., n$. Somit

ist

$$L_k x = \begin{bmatrix} 1 & 0 & & & \cdots & 0 \\ 0 & \ddots & \ddots & & & \vdots \\ & \ddots & 1 & 0 & & \\ & & -l_{k+1,k} & 1 & \ddots & \\ \vdots & & \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & -l_{n,k} & \cdots & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_k \\ x_{k+1} \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} x_1 \\ \vdots \\ x_k \\ 0 \\ \vdots \\ 0 \end{bmatrix} .$$

Matrizen der Form L_k können also benutzt werden, um die unteren $n-k$ Einträge eines Spaltenvektors zu Null zu transformieren.

Sei $A = A_1 = [a_{ij}]_{ij}$ eine $n \times n$–Matrix und $x = [a_{i1}]_i \in \mathbb{K}^n$ die *erste* Spalte von A_1. Wenn $a_{11} \neq 0$ ist, dann gilt mit der Matrix $L_1 = I - l_1 e_1^*$

$$\begin{aligned} L_1 A_1 &= \begin{bmatrix} 1 & 0 & \cdots & & 0 \\ -l_{21} & 1 & 0 & \cdots & 0 \\ -l_{31} & 0 & 1 & & 0 \\ \vdots & & \ddots & \ddots & 0 \\ -l_{n1} & 0 & \cdots & 0 & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ a_{31} & a_{32} & \cdots & a_{3n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \\ &= \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ 0 & a_{32}^{(2)} & \cdots & a_{3n}^{(2)} \\ \vdots & \vdots & & \vdots \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} \end{bmatrix} =: A_2 . \end{aligned} \tag{2.2}$$

Dies entspricht dem ersten Schritt im Gauß-Algorithmus. Wenn $a_{22}^{(2)} \neq 0$ ist, wählen wir in einem zweiten Schritt ($k = 2$) für x die zweite Spalte von A_2, also $x = [a_{12}, a_{22}^{(2)}, ..., a_{n2}^{(2)}]^t$. Mit der zugehörigen Matrix $L_2 = I - l_2 e_2^*$ ergibt sich dann entsprechend $A_3 = L_2 A_2$, wobei:

$$L_2 = \begin{bmatrix} 1 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & -l_{32} & 0 & \cdots & \cdots & 0 \\ 0 & -l_{42} & 0 & 1 & 0 & \cdots \\ \vdots & \vdots & & \ddots & \ddots & 0 \\ 0 & -l_{n2} & & \cdots & \cdots & 1 \end{bmatrix} \quad A_3 = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2n}^{(2)} \\ 0 & 0 & a_{33}^{(3)} & \cdots & a_{3n}^{(2)} \\ 0 & 0 & a_{43}^{(3)} & \cdots & a_{4n}^{(2)} \\ \vdots & \vdots & & & \vdots \\ 0 & 0 & a_{n3}^{(3)} & \cdots & a_{nn}^{(2)} \end{bmatrix} .$$

Geht man auf diese Art und Weise weiter fort (immer vorausgesetzt, dass das **Pivotelement** $a_{ii}^{(i)}$ von Null verschieden ist), dann erhält man nach $(n-1)$ Transformationen schließlich eine obere Dreiecksmatrix $R := A_n$ und es gilt:

$$R = L_{n-1}A_{n-1} = L_{n-1}L_{n-2}\cdots L_1 A\ .$$

Mit anderen Worten: Es ist

$$A = LR \text{ mit } L = L_1^{-1}L_2^{-1}\cdots L_{n-1}^{-1}\ . \tag{2.3}$$

Die inversen Matrizen L_i^{-1} können explizit angegeben werden. Aus dem folgenden Resultat sieht man insbesondere, dass L tatsächlich eine Dreiecksmatrix ist.

Satz 2.1. *Es ist* $L_i^{-1} = I + l_i e_i^*$ *(man beachte* $L_i = I - l_i e_i^*$*) und* $L = I + l_1 e_1^* + \cdots + l_{n-1}e_{n-1}^*$.

Beweis. Es gilt

$$e_i^* l_j = [0,\cdots,0,1,0,\cdots,0]\begin{bmatrix} 0 \\ \vdots \\ 0 \\ l_{j+1,j} \\ \vdots \\ l_{n,j} \end{bmatrix} = \begin{cases} 0 & i \le j \\ l_{i,j} & i \ge j+1 \end{cases}\ . \tag{2.4}$$

Daraus folgt zunächst die erste Behauptung, denn

$$\begin{aligned}(I - l_i e_i^*)(I + l_i e_i^*) &= I - l_i e_i^* + l_i e_i^* - l_i e_i^* l_i e_i^* \\ &= (e_i^* l_i = 0 \text{ wegen } (2.4)) \\ &\quad I - l_i e_i^* + l_i e_i^* \\ &= I\ .\end{aligned}$$

Die spezielle Form von L ergibt sich induktiv: Dazu nehmen wir an, dass für ein $1 \le k < n$ gilt

$$L_1^{-1}\cdots L_k^{-1} = I + l_1 e_1^* + \cdots + l_k e_k^*\ .$$

Für $k = 1$ ist diese Gleichung wegen des ersten Teils des Lemmas, den wir schon bewiesen haben, erfüllt. Aus $L_{k+1}^{-1} = I + l_{k+1}e_{k+1}^*$ folgt dann

$$L_1^{-1}\cdots L_{k+1}^{-1} = (I + l_1 e_1^* + \cdots + l_k e_k^*)(I + l_{k+1}e_{k+1}^*)\ ,$$

und wegen (2.4) ergibt dies

$$\begin{aligned} L_1^{-1}\cdots L_{k+1}^{-1} =& I + l_1 e_1^* + \cdots + l_k e_k^* + l_{k+1} e_{k+1}^* + \sum_{i=1}^{k} l_i e_i^* l_{k+1} e_{k+1}^* \\ =& (e_i^* l_{k+1} = 0 \text{ für } i = 1, \cdots, k) \\ & I + l_1 e_1^* + \cdots + l_k e_k^* + l_{k+1} e_{k+1}^* \,. \end{aligned}$$

Damit ist die Induktionsbehauptung auch für $k+1$ erfüllt. □

Wird im Verlaufe des Gauß-Algorithmus ein Pivotelement $a_{i,i}^{(i)}$ Null, so bricht der Algorithmus zusammen. Sind hingegen alle Pivotelemente für $i = 1, \cdots, n$ von Null verschieden, dann haben wir insgesamt folgendes Resultat bewiesen:

Satz 2.2. *Falls kein Pivotelement Null wird, bestimmt der Gauß-Algorithmus eine LR-Zerlegung.*

$$A = LR = \begin{bmatrix} 1 & & & & \\ l_{21} & 1 & & & \\ l_{31} & l_{32} & 1 & & \\ \vdots & \vdots & \vdots & \vdots & \\ l_{n1} & l_{n2} & \cdots & l_{n,n-1} & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2n}^{(2)} \\ & & a_{33}^{(3)} & \cdots & a_{3n}^{(3)} \\ & 0 & \vdots & \vdots & \\ & & & & a_{nn}^{(n)} \end{bmatrix}$$

in eine linke untere und eine rechte obere Dreiecksmatrix.

Gleichungssysteme mit Dreiecksmatrizen können unmittelbar durch Vorwärts- bzw. Rückwärtssubstitution gelöst werden. Somit ermöglicht die LR-Zerlegung in einfacher Weise die Lösung eines linearen Gleichungssytems $Ax = b$.

Algorithmus 2.3. 1. Zerlege $A = LR$ mit dem Gauß-Algorithmus

2. Löse $Ax = LRx = b$ in zwei Schritten wie folgt:

   - Löse $Ly = b$ durch Vorwärtssubstitution
   - Löse $Rx = y$ durch Rückwärtssubstitution

   Da $Ax = L(Rx) = v$ ist x die gesuchte Lösung des Gleichungssystems $Ax = b$.

Im Folgenden berechnen wir den Aufwand der LR-Zerlegung. Aus (2.2) erkennt man, dass eine Matrix-Matrix-Multiplikation $A_{k+1} = L_k A_k$ genau $(n-k)^2$ Multiplikationen kostet. Dazu kommen noch $(n-k)$ Division um den k-ten Spaltenvektor l_{jk} ($[a_{jk}^{(k)}]/a_{kk}^{(k)}$) zu bestimmen. Insgesamt ergibt sich also ein Gesamtaufwand in der LR-Zerlegung von

$$\begin{aligned}\sum_{k=1}^{n-1}(n-k+1)(n-k) = \sum_{j=1}^{n-1}(j+1)j &= \frac{(n-1)n(2n-1)}{6} + \frac{n(n-1)}{2} \\ &= \frac{1}{3}n^3 + O(n^2)\end{aligned}$$

Multiplikationen und Divisionen.

Der Aufwand zur Berechnung der eigentlichen Lösung ist gegenüber dem Aufwand der Berechnung der LR-Zerlegung vernachlässigbar: Dazu ist zunächst für jeden Eintrag von L, der von Null und Eins verschieden ist, eine Multiplikation erforderlich. Die gleiche Anzahl von Multiplikationen werden bei der Rückwärtssubstitution mit R gebraucht, zuzüglich n Divisionen durch die Diagonalelemente. Insgesamt ergibt sich also ein Aufwand von Multiplikationen und Divisionen von

$$\begin{aligned}\frac{(n-1)^2}{2}(\text{Vorwärtssubstitution}) + \frac{(n-1)^2}{2} + n(\text{Rückwärtssubstitution}) \\ = n^2 + n - 1\end{aligned}$$

Multiplikationen bzw. Divisionen.

Der Aufwand der Additionen und Subtraktionen ist im Regelfall von der Rechenzeit her vernachlässigbar, und wird deshalb in den Aufwandsberechnungen im allgemeinen auch vernachlässigt.

Beispiel 2.4. Wir demonstrieren die Funktionsweise der LR-Zerlegung anhand eines einfachen Beispiels:

$$\begin{aligned} A &= \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 10 \end{bmatrix} \\ &= (\text{“}k = 1'') \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 4 & 7 \\ 0 & -3 & -6 \\ 0 & -6 & -11 \end{bmatrix} \\ &= (\text{“}k = 2'') \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 4 & 7 \\ 0 & -3 & -6 \\ 0 & 0 & 1 \end{bmatrix} \\ &= LR\,. \end{aligned}$$

Ist $b = [1, 1, 1]^t$ die rechte Seite des linearen Gleichungssystems, dann ergibt sich zunächst y aus $Ly = b$,

$$\begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \text{ also } y = \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix},$$

und dann ist schließlich x eine Lösung von $Rx = y$:

$$\begin{bmatrix} 1 & 4 & 7 \\ 0 & -3 & -6 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}, \text{ also } x = \begin{bmatrix} -1/3 \\ 1/3 \\ 0 \end{bmatrix}.$$

Leider sind die Voraussetzungen des Satzes 2.2 nicht immer erfüllt. Bei einer singulären Matrix $A \in \mathbb{K}^{n\times n}$ gilt

$$0 = \det A = (\det L)(\det R) = 1 \prod_{i=1}^{n} a_{ii}^{(i)},$$

und die rechte Seite wäre von Null verschieden, wenn alle Pivotelemente ungleich Null sind. Es gibt aber auch nichtsinguläre Matrizen, die keine LR-Faktorisierung besitzen:

Beispiel 2.5.

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \neq \begin{bmatrix} 1 & 0 \\ l_{21} & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} \\ 0 & r_{22} \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} \\ l_{21}r_{11} & l_{21}r_{12} + r_{22} \end{bmatrix} :$$

Durch Vergleich der Matrix A mit der Matrix auf der rechten Seite der Ungleichungskette ergibt sich zwangsläufig, dass $r_{11} = 0$, was im Widerspruch zur Bedingung $l_{21}r_{11} = 1$ steht. In diesem Beispiel bricht der Gauß-Algorithmus sofort zusammen, da das erste Pivotelement a_{11} Null ist.

Selbst wenn eine LR-Zerlegung existiert, können numerische Instabilitäten im Verlauf der Gauß-Elimination auftreten, wenn ein Pivotelement betragsmäßig sehr klein wird.

Beispiel 2.6. Wir betrachten das lineare 2×2 Gleichungssystem $Ax = b$ mit

$$A = \begin{bmatrix} 0.001 & -1 \\ 1 & 2 \end{bmatrix} \text{ und } b = \begin{bmatrix} -4 \\ 6 \end{bmatrix} .$$

Die Matrix A ist gut konditioniert: Mit einer expliziten Formel für die Inverse von A ergibt sich

$$A^{-1} = \begin{bmatrix} 1.996... & 0.998... \\ -0.998... & 0.001... \end{bmatrix}$$

und somit ist $\text{cond}_\infty = \|A\|_\infty \|A^{-1}\|_\infty \sim 3{*}3 = 9$ (also die Kondition bzgl. der Zeilensummennorm). Das bedeutet, dass wir gemäß (1.10) bei der Lösung des Gleichungssystems mit einer Verneunfachung des relativen Fehlers in der Maximumsnorm rechnen müssen. Für das obige Beispiel ergibt sich eine Lösung

$$x = A^{-1}b = \begin{bmatrix} -1.996... \\ 3.998... \end{bmatrix} .$$

Nehmen wir an, wir hätten einen Rechner mit einer Maschinengenauigkeit **eps** $= 0.005$. Dann ergibt der Gauß-Algorithmus zunächst die exakte LR-Zerlegung von A,

$$L^{-1}(!*)A = \begin{bmatrix} 1 & 0 \\ -1000 & 1 \end{bmatrix} (!*) \begin{bmatrix} 0.001 & -1 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 0.001 & -1 \\ 0 & 1000 \end{bmatrix} .$$

Beachte $r_{22} = 1000(!+)2$ ist gerundet.

Bei der Vorwärtssubstitution ergeben sich wieder Rundungsfehler.

$$y_1 = -4\,, \quad y_2 = 6(!-)(1000(!*)y_1) = 6(!+)4000 = 4010\,.$$

Dies führt dann zu folgender "Lösung" von $x = Ry$

$$x_2 = 4010(!/)1000 = 4.01 \text{ und } x_1 = 1000(!*)(-4(!+)x_2) = 10\,.$$

Der relative Fehler in der Lösung ist circa $12/4 = 3$ und der relative Fehler in den Daten ist $0.005/6 = 0.0008333....$ Also ist die relative Fehlerverstärkung um einen Faktor 3600, was nicht von der gleichen Größenordnung ist, wie sie durch die Kondition vorhergesagt ist. Also ist der Algorithmus **nicht** stabil.

Der Grund der Instabilität liegt in der Verwendung des kleinen Pivotelement durch das im Algorithmus durchdividiert werden muss.

Zur Stabilisierung der Gauß-Elimination bietet sich an, in jedem Eliminierungsschritt die i-te Zeile mit einer anderen Zeile zu vertauschen, um ein möglichst großes Pivotelement zu erhalten.

Beispiel 2.7. Wendet man diese Technik an, dann ergibt sich bei gleicher Rechengenauigkeit

$$\begin{bmatrix} 1 & 2 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -0.001 & 1 \end{bmatrix} (!*) \begin{bmatrix} 1 & 2 \\ 0.001 & -1 \end{bmatrix},$$

und anschließender Vorwärts- und Rückwärtssubstitution (man beachte, dass natürlich auch die Komponenten der rechten Seite $[b_1, b_2]^t$ vertauscht werden müssen)

$$y_1 = 6\,, \quad y_2 = -4(!-)0.006 = -4.01\,,$$

und

$$x_2 = 4.01\,, \quad x_1 = 6(!-)8.02 = -2.02\,.$$

Die Genauigkeit ist nun von der Größenordnung des zu erwartenden Fehlers.

Das Attribut "groß" für ein Pivotelement ist relativ zu verstehen. Multipliziert man bei der Matrix des obigen Beispiels A die oberste Zeile mit einem Faktor 10000, dann würden man mit dem gleichem Argument, wie im letzten Beispiel auf die Vertauschung verzichten. Das wäre aber ein Fehler, denn die Multiplikation der Pivotzeile ändert nichts an den Summanden im Eliminationsschritt.

Bewährt hat sich die **Spaltenpivotsuche** (**partial pivoting**), bei der im i-ten Teilschritt das Element $a_{k,i}^{(i)}$ $(i \leq k \leq n)$ als Pivot-Element gewählt wird, das **relativ** zur Betragsummennorm der jeweiligen Zeile am betragsgrößten ist, d.h., für das

$$\frac{\left|a_{k,i}^{(i)}\right|}{\sum_{l=i}^{n} |a_{k,l}^{(i)}|}$$

bezüglich k maximal wird.

Mit Spaltenpivotsuche wird die Matrixformulierung des Gauß-Algorithmus schwieriger. Werden vor dem i-ten Eliminationsschritt beispielsweise die i-te und die j-te Zeile vertauscht, dann kann dies auch mit einer zugehörigen Permutationsmatrix

$$P_i = \left[\begin{array}{ccc|ccc|ccc} 1 & & & & & & & & \\ & \ddots & & & & & & & \\ & & 1 & & & & & & \\ \hline & & & 0 & & 1 & & & \\ & & & & 1 & & & & \\ & & & & & \ddots & & & \\ & & & & & & 1 & & \\ & & & 1 & & & 0 & & \\ \hline & & & & & & & 1 & \\ & & & & & & & & \ddots \\ & & & & & & & & & 1 \end{array}\right] \tag{2.5}$$

beschrieben werden. In der obigen Matrix grenzen die eingezogenen Teile den Bereich zwischen i-ter und j-ter Zeile bzw. Spalte ab. Es gelten folgende Rechenregeln:

- Multiplikation einer Matrix A mit P_i von *links* (also P_iA) entspricht einer Vertauschung der i-ten und j-ten *Zeile* von A;
- Multiplikation einer Matrix A mit P_i von *rechts* entspricht einer Vertauschung der i-ten und j-ten *Spalte* von A.
- $P_i^2 = I$.

Werden also vor dem i-ten Eliminationsschritt die i-te und j-te Zeile von A vertauscht, dann bedeutet das, dass in dem Eliminationsschritt die Eliminationsmatrix L_i von links an P_iA_i heranmultipliziert wird, also

$$A_{i+1} = L_iP_iA_i \ . \tag{2.6}$$

Lemma 2.8. *Sei $k < i$, P_i wie in (2.5) und $L_k = I - l_ke_k^*$ (wie in (2.1)). Dann ist $P_iL_k = \tilde{L}_kP_i$, wobei $\tilde{L}_k$ bis auf eine Vertauschung von l_{ik} und l_{jk} wieder die Form (2.1) hat.*

Beweis. Aus $P_i^2 = I$ folgt

$$P_i L_k = P_i L_k P_i^2 = (P_i L_k P_i) P_i;$$

Es bleibt zu zeigen, dass $\tilde{L}_k := P_i L_k P_i$ die behauptete Gestalt hat. Aufgrund der oben angeführten Rechenregeln mit P_i folgt weiter:

$$\begin{aligned}
\tilde{L}_k &= (P_i L_k) P_i \\
&= \left[\begin{array}{ccc|ccc|c}
\ddots & & & & & & \\
& 1 & & & & & \\
& -l_{k+1,k} & 1 & & & & \\
& \vdots & & \ddots & & & \\
\hline
& -l_{j,k} & & 0 & & 1 & \\
& \vdots & & & 1 & & \\
& \vdots & & & & \ddots & \\
& \vdots & & & & & 1 \\
& -l_{ik} & & 1 & & 0 & \\
\hline
& \vdots & & & & & \ddots \\
& -l_{n,k} & & & & & 1
\end{array}\right] P_i \\
&= \left[\begin{array}{ccc|ccc|c}
\ddots & & & & & & \\
& 1 & & & & & \\
& -l_{k+1,k} & 1 & & & & \\
& \vdots & & \ddots & & & \\
\hline
& -l_{j,k} & & 1 & & 0 & \\
& \vdots & & & 1 & & \\
& \vdots & & & & \ddots & \\
& \vdots & & & & & 1 \\
& -l_{ik} & & 0 & & 1 & \\
\hline
& \vdots & & & & & \ddots \\
& -l_{n,k} & & & & & 1
\end{array}\right].
\end{aligned}$$

□

Mit diesen Hilfsmitteln können wir folgendes Resultat über den Gauß-Algorithmus mit Spaltenpivotsuche beweisen:

Satz 2.9. *Ist A nichtsingulär, dann definiert der Gauß-Algorithmus mit Spaltenpivotsuche eine Zerlegung der Matrix $PA = \tilde{L}R$, wobei $R = A_n$ eine rechte obere Dreiecksmatrix ist, für die gilt $A_{i+1} = L_i P_i A_i$, und P_i eine Permuationsmatrix ist. Die linke untere Dreiecksmatrix ergibt sich durch Vertauschen geeigneter Elemente in den Spalten der Matrix L aus Lemma 2.1.*

Beweis. Nehmen wir zunächst an, dass der Gauß-Eliminiationsalgorithmus mit Spaltenpivotsuche nicht zusammenbricht. Dann ergibt sich aus (2.6) ($A_{i+1} = L_i P_i A_i$) durch sukzessives Anwenden von Lemma 2.8:

$$\begin{aligned} R = A_n &= L_{n-1}P_{n-1}A_{n-1} = L_{n-1}P_{n-1}L_{n-2}P_{n-2}L_{n-3}P_{n-3}\cdots A \\ &= L_{n-1}\tilde{L}_{n-2}P_{n-1}P_{n-2}L_{n-3}P_{n-3}\cdots A \\ &= L_{n-1}\tilde{L}_{n-2}\tilde{L}_{n-3}P_{n-1}P_{n-2}P_{n-3}\cdots A\,. \end{aligned}$$

Hierbei bezeichnet $\tilde{L}_{n-3}$ die Matrix, die sich gemäß Lemma 2.8 aus L_{n-3} nach dem Vertauschen mit P_{n-2} *und* P_{n-1} ergibt. Bezeichne auch $\tilde{L}_{n-1} = L_{n-1}$, dann ergibt sich durch Auflösen der Rekursion

$$R = \tilde{L}_{n-1}\cdots\tilde{L}_1 P_{n-1}\cdots P_1 A = \tilde{L}_{n-1}\cdots\tilde{L}_1 PA\,.$$

Die Matrix $\tilde{L}$ aus der Formulierung des Satzes ist somit wie in Lemma 2.1 über den Gauß-Algorithmus ohne Pivotsuche das Produkt der Inversen von $\tilde{L}_1$ bis $\tilde{L}_{n-1}$, d.h., auch die Elemente von $\tilde{L}$ unterscheiden sich von den Elementen von L aus Lemma 2.1 lediglich durch Permutationen.

Zu klären bleibt schließlich noch, dass der Gauß-Algorithmus mit Spaltenpivotsuche nicht abbricht, also dass alle Pivotelemente nach der i-ten Spaltenpivotsuche von Null verschieden sind. Wäre etwa das Pivotelement nach dem i-ten Teilschritt tatsächlich Null, dann sind zwangsläufig wegen der Auswahlregel **alle** Elemente $a^{(i)}_{j,i}, j \geq i$, gleich Null, d.h.,

$$A_i = \left[\begin{array}{ccc|ccc} a_{1,1} & \cdots & & a_{1,i} & \cdots & a_{1,n} \\ & \ddots & & \vdots & & \vdots \\ \hline & & & 0 & a_{i,i+1} & \cdots & a_{i,n} \\ & & & \vdots & \vdots & \vdots & \vdots \\ & & & 0 & a_{n,i+1} & \cdots & a_{n,n} \end{array}\right].$$

Die Determinante des rechten unteren quadratischen Blocks ist demnach Null und daher ist auch die Determinante von A_i Null. Durch den Produktsatz

für Determinanten folgt daraus aber

$$\begin{aligned} 0 = \det(A_i) &= \det(L_{i-1}P_{i-1}\cdots L_1P_1A) \\ &= \left(\prod_{j=1}^{n}\det(L_j)\prod_{j=1}^{n}\det(P_j)\right)\det(A)\,. \end{aligned}$$

Da die Determinante einer unteren Dreiecksmatrix das Produkt der Diagonaleinträge ist, ist $\det(L_j) = 1$. Die Permutationsmatrizen P_j erfüllen $\det(P_j) = \pm 1$. Somit folgt aus obiger Gleichung, dass A singulär ist. Was im Widerspruch zu unserer Annahme steht. □

Für eine spezielle Klasse von Matrizen kann auf die Spaltenpivotsuche verzichtet werden, da ohnehin niemals Zeilen vertauscht werden. Eine solche Klasse ist die Klasse der strikt diagonalen Matrizen.

Definition 2.10. Eine Matrix A heißt **strikt diagonaldominant**, falls

$$|a_{ii}| > \sum_{j\neq i}|a_{ij}| \text{ für alle } i = 1,\cdots,n\,.$$

Satz 2.11. *Ist A strikt diagonaldominant, dann wählt die Pivotsuche in jedem Eliminiationsschritt das Diagonalelement a_{ii}^i als Pivotelement aus. Insbesondere existiert also eine LR-Zerlegung von A und A ist nicht singulär.*

Beweis. Betrachten wir zunächst die Auswahl des ersten Pivotelements. Da A strikt diagonaldominant ist, ist

$$\sum_{j=1}^{n}|a_{1,j}| = \sum_{j\neq 1}|a_{1,j}| + |a_{1,1}| < 2|a_{1,1}|\,.$$

Das bedeutet, dass

$$\frac{1}{2} < \frac{|a_{1,1}|}{\sum_{j=1}^{n}|a_{1,j}|}\,. \tag{2.7}$$

Analog sieht man, dass für $i \neq 1$

$$2|a_{i,1}| \leq |a_{i,1}| + |a_{i,i}| \leq \sum_{j=1}^{n}|a_{i,j}|\,. \tag{2.8}$$

Vergleicht man (2.7) und (2.8), so erkennt man, dass das $(1,1)$-Element als Pivotelement ausgewählt wird.

Der Beweis läuft nun induktiv durch, wenn wir zeigen können, dass die rechte untere $(n-1)^2$-Untermatrix von A_2 wieder strikt diagonaldominant ist. Dazu schreiben wir den ersten Eliminationsschritt in der Blockform

$$\left[\begin{array}{c|c} a_{11} & b^t \\ \hline a & B \end{array}\right] = \left[\begin{array}{c|c} 1 & 0 \\ \hline a/a_{11} & \begin{array}{ccc} 1 & & \\ & \ddots & \\ & & 1 \end{array} \end{array}\right] \left[\begin{array}{c|c} a_{11} & b^t \\ \hline 0 & B - \frac{1}{a_{11}} a b^t \end{array}\right] .$$

Die rechte Matrix ist A_2; der untere Block von A_2 ist also gegeben durch $B - \frac{1}{a_{11}} a b^t$, sodass das (i,j)-te Element von A_2 für $2 \le i, j \le n$ die folgende Form hat:

$$a_{i,j}^{(2)} = a_{i,j} - \frac{a_{i1} a_{1j}}{a_{11}} , \quad 2 \le i, j \le n .$$

Die Untermatrix von A_2 ist demnach genau dann strikt diagonaldominant, wenn

$$\left| a_{i,i} - \frac{a_{i,1} a_{1,i}}{a_{1,1}} \right| > \sum_{j=2, j\neq i}^{n} \left| a_{i,j} - \frac{a_{i,1} a_{1,j}}{a_{1,1}} \right| , \quad i = 2, \cdots n . \tag{2.9}$$

Wegen der strikten Diagonaldominanz von A gilt tatsächlich

$$\begin{aligned} \sum_{j=2, j\neq i}^{n} \left| a_{i,j} - \frac{a_{i,1} a_{1,j}}{a_{1,1}} \right| &\le \sum_{j=2, j\neq i}^{n} |a_{i,j}| + \left| \frac{a_{i,1}}{a_{1,1}} \right| \sum_{j=2, j\neq i}^{n} |a_{1,j}| \\ &< \sum_{j=2, j\neq i}^{n} |a_{i,j}| + |a_{i,1}| \frac{|a_{1,1}| - |a_{1,i}|}{|a_{1,1}|} \\ &< |a_{i,i}| - |a_{i,1}| + |a_{i,1}| - \frac{|a_{i,1} a_{1,i}|}{|a_{1,1}|} \\ &= |a_{i,i}| - \frac{|a_{i,1} a_{1,i}|}{|a_{1,1}|} \\ &\le \left| a_{i,i} - \frac{a_{i,1} a_{1,i}}{a_{1,1}} \right| . \end{aligned}$$

Somit ist A_2 ebenfalls strikt diagonaldominant.

Nach Satz 2.2 liefert der Gauß-Algorithmus eine Faktorisierung $A = LR$ mit nichtsingulären Matrizen L und R und damit ist auch A nicht singulär. □

Mit der Spaltenpivotsuche arbeitet der Gauß-Algorithmus in der Praxis sehr zuverlässig, obwohl immer noch Beispiele konstruiert werden können, bei denen selbst diese Pivotwahl versagt. In solchen Ausnahmefällen kann man statt dessen eine andere Pivotstrategie verfolgen, die so genannte **Totalpivotsuche** (**total pivoting**). Dabei wählt man vor dem i-ten Eliminiationsschritt aus dem gesamten rechten unteren Matrixblock (also aus den Indizes (j,k) mit $i \leq j,k \leq n$) das Element $a_{j,k}^{(i)}$ als Pivot-Element aus, das betragsmäßig am **größten** ist (**kein** relatives Kriterium). Das entsprechende Element (etwa $a_{j,k}^{(i)}$) wird an die (i,i)-te Position gebracht, indem wie zuvor die Zeilen j und i und zusätzlich noch die Spalten k und i vertauscht werden. Letzteres wird formal dadurch beschrieben, dass A mit einer Permutationsmatrix Q_i von rechts multipliziert wird (Q_i sieht wie die Permutationsmatrix in (2.5) aus, wobei k die Rolle von j übernimmt). Entsprechend zu (2.6), ergibt dies die Matrixtransformation

$$A_{i+1} = L_i P_i A_i Q_i \,,$$

und man erhält schließlich die LR-Zerlegung der Matrix PAQ mit $Q = Q_1 \cdots Q_{n-1}$.

Wird die Totalpivotsuche verwendet, dann entsprechen Spaltenvertauschungen Permutationen der Lösung x. Der Ergebnisvektor ist also nicht in der richtigen Reihenfolge.

Die Totalpivotsuche wird in der Praxis nur sehr selten eingesetzt, da die Suche nach dem betragsgrößten Element im i-ten Schritt einen Aufwand $O(i^2)$ entspricht. Der Gesamtaufwand der totalen Pivotsuche ist also gegenüber der eigentlichen Rechnung nicht mehr vernachlässigbar.

Hat man mit der LR-Zerlegung von A erst einmal eine Näherung x der Lösung $\hat{x}$ berechnet, dann gibt es eine relativ billige Möglichkeit die Näherung zu verbessern:

Algorithmus 2.12. (Nachiteration)

- Berechne das Residuum $r = b - Ax$ doppelt genau
- Löse $Lz = r$ mit der LR-Zerlegung: $Ly = r; Rr = y$. Da die LR Zerlegung schon bekannt ist, ist dieser Schritt von vernachlässigbarem Rechenaufwand.
- Ersetze x durch $x + z$.

Jede Nachiteration ist relativ billig, da die Vorwärts –, bzw. Rückwärtssubstitution mit nur n^2 Multiplikationen auskommt. Im Gegensatz zur LR-Zerlegung die $O(n^3)$ Operationen benötigt.

Wir fassen die wichtigsten Ergebnisse über den Gauß-Algorithmus in einer Tabelle zusammen

Verfahren	Aufwand
ohne Pivotierung	$\frac{1}{3}n^3 + O(n^2)$
mit Spaltenpivotierung	$\frac{1}{3}n^3 + O(n^2)$
mit Totalpivotierung	$\frac{1}{3}n^3 + O(\sum_{i=1}^{n} i^2)$

2.2 Die Cholesky-Zerlegung

Wir betrachten zunächst eine "Blockversion" der LR-Zerlegung. Dazu partitionieren wir eine gegebene Matrix $A \in \mathbb{K}^{n\times n}$ in die Form

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \text{ mit nichtsingulärem } A_{11} \in \mathbb{K}^{p\times p} .$$

Dabei ist $A_{12} \in \mathbb{K}^{p\times(n-p)}$, $A_{21} \in \mathbb{K}^{(n-p)\times p}$ und $A_{22} \in \mathbb{K}^{(n-p)\times(n-p)}$. Bei der Block-LR-Zerlegung von A gehen wir analog zum vorigen Abschnitt vor und faktorisieren

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} I & 0 \\ A_{21}A_{11}^{-1} & I \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ 0 & S \end{bmatrix}$$

mit

$$S = A_{22} - A_{21}A_{11}^{-1}A_{12} . \tag{2.10}$$

Definition 2.13. Die $(n-p)\times(n-p)$–Matrix S aus (2.10) heißt **Schurkomplement** von A bzgl. A_{11}.

Die Lösung eines linearen Gleichungssystems $Ax = b$ kann durch (Block)–Vorwärts– und Rückwärtssubstitution erfolgen: Dazu werden die Vektoren x und $b \in \mathbb{K}^n$ in ihren ersten p Komponenten $x_1, b_1 \in \mathbb{K}^p$ und die restlichen Komponenten $x_2, p_2 \in \mathbb{K}^{n-p}$ zerlegt, d.h. wir betrachten das System

$$\begin{bmatrix} I & 0 \\ A_{21}A_{11}^{-1} & I \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \text{ (Vorwärtssubstitution)}$$

$$\begin{bmatrix} A_{11} & A_{12} \\ 0 & S \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \text{ (Rückwärtssubstitution)) .}$$

Die Vorwärtssubstitution ergibt also Hilfsvektoren

$$\begin{aligned} y_1 &= b_1 \,, \\ y_2 &= b_2 - A_{21} A_{11}^{-1} b_1 \end{aligned}$$

aus denen dann durch anschließende Rücksubstitution das Ergebnis berechnet wird:

$$\begin{aligned} x_2 &= S^{-1} y_2 = S^{-1}(b_2 - A_{21} A_{11}^{-1} b_1) \,, \\ x_1 &= A_{11}^{-1}(b_1 - A_{12} x_2) \,. \end{aligned}$$

Letzteres ist allerdings nur möglich, wenn S nichtsingulär ist.

Lemma 2.14. *A sei hermitesch und positiv definit. Dann ist für jedes $1 \leq p \leq n$ die Submatrix A_{11} hermitesch und sowohl A_{11} wie S sind hermitesch und positiv definit.*

Beweis. Wegen

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = A = A^* = \begin{bmatrix} A_{11}^* & A_{21}^* \\ A_{12}^* & A_{22}^* \end{bmatrix}$$

ergibt sich

$$A_{11} = A_{11}^* \,, \quad A_{22} = A_{22}^* \text{ und } A_{12} = A_{21}^* \,.$$

Folglich ist A_{11} hermitesch und für einen beliebigen Vektor $x \in \mathbb{K}^p$ gilt

$$0 \leq \begin{bmatrix} x \\ 0 \end{bmatrix}^* A \begin{bmatrix} x \\ 0 \end{bmatrix} = \begin{bmatrix} x \\ 0 \end{bmatrix}^* \begin{bmatrix} A_{11} x \\ A_{21} x \end{bmatrix} = x^* A_{11} x \,,$$

wobei Gleichheit wegen der positiven Definitheit von A nur dann gelten kann, wenn $x = 0$. Also ist A_{11} ebenfalls positiv definit und A_{11}^{-1} existiert. S ist somit wohldefiniert mit

$$S^* = A_{22}^* - A_{12}^* A_{11}^{-*} A_{21}^* = A_{22} - A_{21} A_{11}^{-1} A_{12} = S \,.$$

Schließlich definieren wir für einen beliebigen Vektor $y \in \mathbb{K}^{n-p}$ den zu-

gehörigen Vektor $x = -A_{11}^{-1}A_{12}y \in \mathbb{K}^p$ und erhalten

$$\begin{aligned} 0 &\leq \begin{bmatrix} x \\ y \end{bmatrix}^* A \begin{bmatrix} x \\ y \end{bmatrix} \\ &= \begin{bmatrix} x \\ y \end{bmatrix}^* \begin{bmatrix} A_{11}x + A_{12}y \\ A_{21}x + A_{22}y \end{bmatrix} \\ &= \begin{bmatrix} x \\ y \end{bmatrix}^* \begin{bmatrix} -A_{12}y + A_{12}y \\ -A_{21}A_{11}^{-1}A_{12}y + A_{22}y \end{bmatrix} \\ &= \begin{bmatrix} x \\ y \end{bmatrix}^* \begin{bmatrix} 0 \\ Sy \end{bmatrix} \\ &= y^*Sy\,, \end{aligned}$$

wobei wiederum Gleichheit nur für $y = 0$ gelten kann. Damit ist S auch positiv definit. □

Beim Gauß-Algorithmus wird die Matrix A in das Produkt $A = LR$ einer linken unteren und einer rechten oberen Dreiecksmatrix zerlegt. Von besonderem Interesse ist der Fall $R = L^*$.

Definition 2.15. Eine Zerlegung $A = LL^*$ mit unterer Dreiecksmatrix L mit positiven Diagonaleinträgen heißt **Cholesky-Zerlegung** von A. [1]

Eine notwendige Bedingung für die Existenz einer Cholesky-Zerlegung gibt das folgende Resultat.

Proposition 2.16. *Hat A eine Cholesky-Zerlegung, dann ist A hermitesch und positiv definit.*

Beweis. Aus $A = LL^*$ folgt unmittelbar

$$A^* = (L^*)^*L^* = LL^* = A\,;$$

Also ist A hermitesch. Ferner gilt

$$x^*Ax = x^*LL^*x = (L^*x)^*L^*x = \|L^*x\|_2^2 \geq 0\,, \quad x \in \mathbb{K}^n\,.$$

Dabei gilt Gleichheit genau für $x = 0$, da L positive Diagonaleinträge hat und somit nicht singulär ist - dies folgt aus der Tatsache, dass die Determinante einer Dreiecksmatrix das Produkt der Diagonalelemente ist. Folglich ist A positiv definit. □

[1] Die Cholesky-Zerlegung soll eindeutig sein, was unter der Voraussetzung positiver Diagonaleinträge garantiert werden kann.

Tatsächlich ist diese Bedingung an A auch hinreichend.

Satz 2.17. *Ist A hermitesch und positiv definit, dann existiert eine Cholesky-Zerlegung von A.*

Beweis. Der Beweis wird induktiv über die Dimension der Matrix geführt, wobei für $n = 1$ die "Matrix" nur aus dem Element a_{11} besteht, das positiv sein muss, da die Matrix $A = a_{11}$ positiv definit ist. Also kann man für $n = 1$ einfach $L = [\sqrt{a_{11}}]$ setzen.

Sei nun die Behauptung für alle Matrizen der Dimension $n-1$ korrekt und A eine beliebige $n \times n$ Matrix. Dann partitionieren wir

$$A = \begin{bmatrix} a_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \text{ mit } A_{22} \in \mathbb{K}^{(n-1)\times(n-1)} \text{ und } A_{12} = A_{21}^* \, .$$

Nach Lemma 2.14 ist a_{11} positiv und das Schurkomplement $S = A_{22} - A_{21}A_{12}/a_{11} \in \mathbb{K}^{(n-1)\times(n-1)}$ von A bzgl. a_{11} ist hermitesch und positiv definit; daher existiert $l_{11} := \sqrt{a_{11}} > 0$ und aufgrund der Induktionsannahme hat S eine Cholesky-Zerlegung $S = L_S L_S^*$. Wir setzen

$$L = \begin{bmatrix} l_{11} & 0 \\ A_{21}/l_{11} & L_S \end{bmatrix}, \quad L^* = \begin{bmatrix} l_{11} & A_{12}/l_{11} \\ 0 & L_S^* \end{bmatrix},$$

und dann folgt

$$LL^* = \begin{bmatrix} l_{11} & 0 \\ A_{21}/l_{11} & L_S \end{bmatrix} \begin{bmatrix} l_{11} & A_{12}/l_{11} \\ 0 & L_S^* \end{bmatrix} = \begin{bmatrix} l_{11}^2 & A_{12} \\ A_{21} & B \end{bmatrix} .$$

mit

$$B = \frac{1}{l_{11}^2} A_{21}A_{12} + L_S L_S^* = \frac{1}{a_{11}} A_{21}A_{12} + S = A_{22} \, .$$

Also ist

$$LL^* = \begin{bmatrix} a_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = A$$

eine Cholesky-Zerlegung von A. □

Die Berechnung der Einträge von L erfolgt sukzessive durch zeilenweise Koeffizientenvergleich bei dem Produkt $A = LL^*$,

$$\begin{bmatrix} a_{11} & a_{12} & \vdots & a_{1n} \\ a_{21} & a_{22} & \vdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \vdots & a_{nn} \end{bmatrix} = \begin{bmatrix} l_{11} & & \vdots & 0 \\ l_{21} & l_{22} & \ddots & \\ \cdots & & \ddots & \\ l_{n1} & l_{n2} & \vdots & l_{nn} \end{bmatrix} \begin{bmatrix} \overline{l_{11}} & \overline{l_{21}} & \vdots & \overline{l_{n1}} \\ & \overline{l_{22}} & \vdots & \overline{l_{n2}} \\ & & \ddots & \\ 0 & & & \overline{l_{nn}} \end{bmatrix}$$

Auf diese Weise ergeben sich die Einträge von L in der folgenden Weise:

$$\begin{array}{ll} a_{11} = |l_{11}|^2 & l_{11} = \sqrt{a_{11}} \\ & \\ a_{21} = l_{21}\overline{l_{11}} & l_{21} = a_{21}/\overline{l_{11}} \\ a_{22} = |l_{21}|^2 + |l_{22}|^2 & l_{22} = \sqrt{a_{22} - |l_{21}|^2} \\ & \Rightarrow \\ a_{31} = l_{31}\overline{l_{11}} & l_{31} = a_{31}/\overline{l_{11}} \\ a_{32} = l_{31}\overline{l_{21}} + l_{32}\overline{l_{22}} & l_{32} = (a_{32} - l_{31}\overline{l_{21}})/\overline{l_{22}} \\ a_{33} = |l_{31}|^2 + |l_{32}|^2 + |l_{33}|^2 & l_{33} = \sqrt{a_{33} - |l_{31}|^2 + |l_{32}|^2} \\ \vdots & \vdots \end{array}$$

Die Lösbarkeit dieser (nichtlinearen) Gleichungen ist durch den Existenzbeweis (Satz 2.17) gewährleistet, d.h. alle Quadratwurzeln existieren und die resultierenden Diagonalelemente l_{ii} von L sind ungleich Null. Aus dem Algorithmus lässt sich nun sofort folgendes Resultat ableiten

Korollar 2.18. *Die Cholesky-Zerlegung einer hermiteschen positiv definiten Matrix A ist eindeutig bestimmt.*

Wie man aus dem Algorithmus sofort sieht, ist der Aufwand zur Berechnung von l_{ij} (mit $i \geq j$) maximal j Multiplikationen, Divisionen und Wurzeln (der Aufwand zur Berechnung der Additionen wird wieder vernachlässigt). Demnach ergibt sich ein Gesamtaufwand bei der Berechnung der Cholesky-Zerlegung von

$$\sum_{j=1}^{n}(n+1-j)j = \frac{n(n+1)^2}{2} - \frac{n(n+1)(2n+1)}{6} = \frac{1}{6}n^3 + O(n^2) \; .$$

Die Cholesky-Zerlegung kann genauso eingesetzt werden, wie die LR-Zerlegung. Sie hat den Vorteil, dass sie etwa nur halb so viel kostet. Darüberhinaus ist ein entscheidender Vorteil der Cholesky-Zerlegung, dass LL^* immer hermitesch und positiv definit ist, selbst wenn L aufgrund von Rundefehlern nur eine Näherung an den exakten Cholesky-Faktor sein sollte. Wird hingegen eine LR-Faktorisierung der hermiteschen, positiv definiten Matrix A berechnet, dann ist wegen der Rundefehler nicht gewährleistet, dass das Produkt LR hermitesch und positiv definit ist.

2.3 Die QR-Zerlegung

Bisher haben wir uns nur mit der Zerlegung von quadratischen Matrizen beschäftigt. Es gibt aber viele Anwendungen - auf die wir später noch zurückkommen werden – die eine Zerlegung von rechteckigen Matrizen erfordern. Ein solcher Algorithmus ist die QR-Zerlegung, mit dem wir uns im folgenden beschäftigen werden. Dazu brauchen wir einige Hilfsmittel.

Definition 2.19. Sei $v \in \mathbb{K}^r \backslash \{0\}$: Dann heißt die Matrix

$$P = I - \frac{2}{v^* v} v v^* \in \mathbb{K}^{r \times r}$$

Householder-Transformation.

Lemma 2.20. *Die Householder-Transformation P ist eine hermitesche, unitäre Matrix mit*

$$Pv = -v \text{ und } Pw = w \text{ für alle } w \in [v]^\perp .$$

Beweis. Aus der Definition von P folgt

$$P_{ij} = 1\delta_{ij} - \frac{2}{v^* v} v_i \overline{v_j} = 1\delta_{ij} - \overline{\frac{2}{v^* v} v_j \overline{v}_i} = \overline{P_{ji}} = P^*_{ij} .$$

Also ist P hermitesch. Außerdem ist P unitär, denn

$$P^* P = P^2 = I - \frac{4}{v^* v} v v^* + \frac{4}{(v^* v)^2} v (v^* v) v^* = I - \frac{4}{v^* v} v v^* + \frac{4}{v^* v} v v^* = I .$$

Schließlich ergibt sich für den Vektor v aus der Definition von P und für beliebiges $w \perp v$ (d.h. $v^* w = 0$)

$$\begin{aligned} Pv &= Iv - \frac{2}{v^* v} v (v^* v) = v - 2v = -v , \\ Pw &= Iw - \frac{2}{v^* v} v (v^* w) = w - 0 = w . \end{aligned}$$

□

Abbildung 2.1 illustriert Lemma 2.20, nach der eine Housholdertransformation eine Abbildungsmatrix einer geometrischen Spiegelung ist.

Insbesondere lassen diese Spiegelungen die Euklidnorm invariant, denn es gilt

$$\|Px\|_2^2 = (Px)^* Px = x^* P^* Px = x^* x = \|x\|_2^2 . \tag{2.11}$$

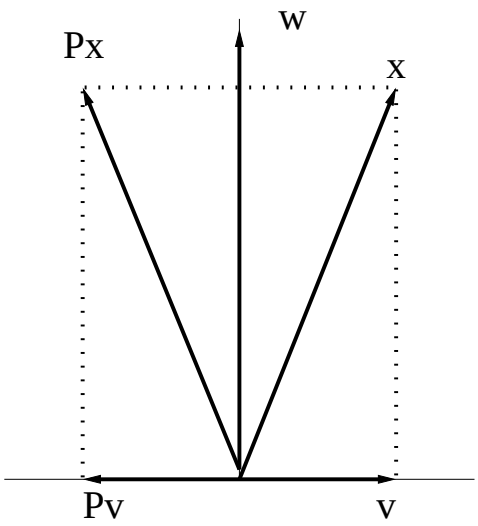

Abbildung 2.1: Householder-Transformationen sind Spiegelungen

Wir verwenden nun Householder-Transformationen, um – ähnlich zu den Gauß-Eliminiationsmatrizen L_k – eine (nicht notwendigerweise quadratische Matrix) auf "obere Dreiecksform" zu transformieren. Wie das bei einer rechteckigen Matrix zu verstehen ist, wird gleich klar werden.

Wir konstruieren zunächst eine Householder-Matrix P, die einen beliebigen Vektor $x \in \mathbb{K}^r \backslash \{0\}$ auf ein Vielfaches von $e_1 \in \mathbb{K}^r$ transformiert, vgl. Abbildung 2.2. Das heißt, wir wollen einen Vektor $v \in \mathbb{K}^r \backslash \{0\}$ finden, sodass gilt

$$Px = x - \frac{2}{v^* v} v(v^* x) = \zeta e_1 \text{ mit } |\zeta| = \|x\|_2 \,.$$

Für den Fall, dass x und e_1 linear abhängig sind, kann man einfach $v = 0$ und $P = I$ wählen. Andernfalls muss $v^* x \neq 0$ und wir setzen

$$v = \frac{1}{|\zeta|}(x + \alpha e_1) \,.$$

Bezeichnet x_1 die erste Komponente von x, dann ergibt dieser Ansatz

$$v^* x = |\zeta| + \frac{\alpha}{|\zeta|} x_1 \text{ und } v^* v = 1 + 2\frac{\alpha}{|\zeta|^2} x_1 + \frac{\alpha^2}{|\zeta|^2} \,.$$

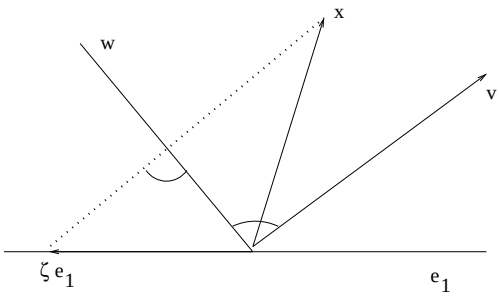

Abbildung 2.2: Spiegelung von x auf e_1

Daraus folgt

$$Px = x - \frac{2\zeta^2 + 2\alpha x_1}{\zeta^2 + 2\alpha x_1 + \alpha^2} x - \frac{2\alpha\zeta^2 + 2\alpha x_1}{\zeta^2 + 2\alpha x_1 + \alpha^2} e_1 \, .$$

Um die Forderung $Px = \zeta e_1$ zu erfüllen, wählen wir α so, dass sich die ersten beiden Terme auf der rechten Seite der obigen Darstellung von Px zu Null ergänzen. Also so, dass

$$2\zeta^2 + 2\alpha x_1 = \zeta^2 + 2\alpha x_1 + \alpha^2 \, .$$

Das ergibt $\alpha = \pm\zeta$ und $Px = -\alpha e_1$. Wir wählen das Vorzeichen von α so, dass es mit dem von x_1 übereinstimmt, also $\operatorname{sign}(x_1) = \operatorname{sign}(\alpha)$ – das hat numerische Vorteile, ist aber im Prinzip eine willkürliche Wahl. [2] Damit haben wir

$$v = \frac{1}{\|x\|_2} x + \operatorname{sign}(x_1) e_1 \, , \quad v^* v = 2 + 2\frac{|x_1|}{\|x\|_2} \, . \tag{2.12}$$

Das Ergebnis lautet dann

$$Px = -\operatorname{sign}(x_1)\|x\|_2 e_1 \, .$$

Satz 2.21. *Sei $A \in \mathbb{K}^{m\times n}$ mit $m \geq n$ und* $\operatorname{Rang}(A) = n$*. Dann existiert eine unitäre Matrix $Q \in \mathbb{K}^{m\times m}$ und einer obere Dreiecksmatrix $R \in \mathbb{K}^{m\times n}$ mit*

$$A = QR = Q \left[\begin{array}{ccc} r_{11} & \cdots & r_{1n} \\ & \ddots & \vdots \\ 0 & & r_{nn} \\ \hline 0 & \cdots & 0 \end{array}\right] \, .$$

[2] Bei uns ist $\operatorname{sign}(x) = 1$ falls $x > 0$, $\operatorname{sign}(x) = -1$ falls $x < 0$ und $\operatorname{sign}(0) = 0$.

Dabei sind $r_{11}, \cdots, r_{nn}$ jeweils von Null verschieden.

Beweis. Wir bestimmen die gesuchte Zerlegung, indem wir in jedem Schritt eine Householder-Transformation von links an A heranmultiplizieren, um sukzessive die Spalten 1 bis n von R zu erhalten. Dies ergibt dann die Darstellung

$$P_n \cdots P_1 A = R \tag{2.13}$$

mit Householder-Transformationen P_i, und daraus folgt dann die QR-Faktorisierung

$$A = QR \text{ mit } Q = P_1^* \cdots P_n^* = P_1 \cdots P_n \, .$$

Im ersten Schritt setzen wir $A_1 = A$ und für x die erste Spalte a_1 von A_1 und bestimmen die Householder-Transformation $P_1 \in \mathbb{K}^{m \times m}$ gemäß (2.12). Es folgt

$$P_1 a_1 = r_{11} e_1 \, , \quad r_{11} = \pm \|a_1\|_2 \neq 0 \, ,$$

beziehungsweise

$$P_1 A = \left[\begin{array}{c|ccc} r_{11} & * & * & * \\ \hline 0 & & A_2 & \end{array}\right] \text{ mit } A_2 \in \mathbb{K}^{(m-1)\times(m-1)} \, .$$

Nehmen wir an, dass wir nach i Schritten Householder-Matrizen $P_1, \cdots, P_i$ konstruiert haben mit

$$P_i \cdots P_1 A = \left[\begin{array}{ccc|c} r_{11} & \cdots & r_{1i} & \\ & \ddots & \vdots & R_i' \\ 0 & & r_{ii} & \\ \hline 0 & \cdots & 0 & A_{i+1} \end{array}\right] \tag{2.14}$$

wobei $R_i' \in \mathbb{K}^{i \times (n-i)}$ und $A_{i+1} \in \mathbb{K}^{(m-i)\times(n-i)}$. Da nach Voraussetzung A Rang n, also vollen Spaltenrang besitzt, hat auch A_{i+1} vollen Spaltenrang.

Im nächsten Schritt können wir daher für $x \in \mathbb{K}^{m-i}$ die erste Spalte a_{i+1} von A_{i+1} wählen und konstruieren die Householder-Matrix $P_{i+1}' \in \mathbb{K}^{(m-i)\times(m-i)}$ über einen Vektor $v' \in \mathbb{K}^{m-i}$ gemäß (2.12). Auf diese Weise ergibt sich

$$P_{i+1}' A_{i+1} = \left[\begin{array}{c|ccc} r_{i+1,i+1} & * & * & * \\ \hline 0 & & A_{i+2} & \end{array}\right]$$

mit $r_{i+1,i+1} = \pm \|a_{i+1}\|_2 \neq 0$ und $A_{i+2} \in \mathbb{K}^{(m-i-1)\times(m-i-1)}$. Somit folgt mit

$$P_{i+1} := \left[\begin{array}{c|c} I & 0 \\ \hline 0 & P_{i+1}' \end{array}\right]$$

$$P_{i+1}P_i\cdots P_1A = \left[\begin{array}{ccc|c|c} r_{11} & \cdots & r_{1i} & & \\ & \ddots & \vdots & R_i' & \\ 0 & & r_{ii} & & \\ \hline 0 & \cdots & 0 & r_{i+1,i+1} & * \ * \ * \\ \hline 0 & \cdots & 0 & 0 & A_{i+2} \end{array}\right] .$$

Man beachte, dass sich in jedem Schritt die ersten i Zeilen **nicht** verändern. P_{i+1} kann selbst wieder als Householder-Transformation mit einem Vektor $v \in \mathbb{K}^m$ der Form $v^t = [0v'^t]$ aufgefasst werden, also eines gestreckten Vektors. Durch vollständige Induktion erhalten wir nun die gewünschte Zerlegung (2.13). □

Bei der Implementierung dieses Verfahrens ist darauf zu achten, dass die Householder-Transformationen **niemals** explizit gebildet werden, denn sonst kostet die Berechnung von PA $O(m^2n)$ Multiplikationen. PA kann statt dessen mit nur $2mn$ Multiplikationen über die Darstellung

$$PA = A - \frac{2}{v^*v}vv^*A = A - \frac{2}{v^*v}vw^* , \quad w = A^*v ,$$

ausgerechnet werden. Man beachte, dass die Berechnung von A^*v und vw^* jeweils nur maximal $m * n$ Operationen benötigen.

Um P zu einem späteren Zeitpunkt weiterverwenden zu können, reicht es aus, den Vektor v abzuspeichern.

Wir fassen nun den Algorithmus der QR-Zerlegung zusammen:

Algorithmus 2.22. `for` $i = 0, \cdots, n-1$

- `Setzung:` a_{i+1} `sei die erste Spalte von` A_{i+1} `und` $a_{i+1,1}$ `ihre erste Komponente (vgl. (2.14));`
- `setze` $v = \frac{a_{i+1}}{\|a_{i+1}\|_2} + \operatorname{sign}(a_{i+1,1})e_i$`, (vgl. (2.12));`
- `setze` $\beta = \frac{2}{v^*v} = \left(1 + \frac{|a_{i+1,1}|}{\|a_{i+1}\|_2}\right)^{-1}$`, (vgl. (2.12));`
- `berechne` $w = \beta A_{i+1}^* v$`;`
- `ersetze` A_{i+1} `durch` $A_{i+1} - vw^*$`;`

`end for`

Der Aufwand beim i-ten Schleifenaufwand der QR-Zerlegung ist wie folgt:

Teilrechnung im i-ten Schritt	Anzahl der Multiplikationen
v gemäß (2.12)[3]:	$2(m-i+1)$
β :	1
w:	$(n-i+1)(m-i+1)$
A_{i+1}:	$(n-i+1)(m-i+1)$
$\sum \approx$	$2(n-i+2)(m-i+1)$

Damit ergibt sich insgesamt ein Aufwand von

$$\begin{aligned} 2\sum_{i=1}^{n}(n-i+2)(m-i+1) &= 2\sum_{j=2}^{n+1} j(m-n+j-1) \\ &= 2\sum_{i=2}^{n+1} i^2 + 2(m-n-1)\sum_{i=2}^{n+1} i \\ &= \frac{2}{3}n^3 + (m-n)n^2 + O(mn) \\ &= mn^2 - \frac{1}{3}n^2 + O(mn)\ . \end{aligned}$$

Die QR-Zerlegung kann alternativ zur LR-Zerlegung eingesetzt werden: In diesem Fall zerlegt man $A = QR$ und löst dann $QRx = b$ durch Rücksubstitution aus der Gleichung $Rx = Q^*b$ (die rechte Seite Q^*b kann wieder mit $O(n^2)$ Multiplikationen berechnet werden, indem man jede einzelne Householder-Transformation P_i mit einem Vektor b multipliziert). Dieses Verfahren ist allerdings um etwa einen Faktor 2 teurer als die Gauß-Elimination. Man beachte, dass die Zerlegung nur mehr auf eine Dreiecksmatrix führt.

Die QR-Zerlegung wird zur Lösung von linearen Ausgleichsproblemen verwendet, wie wir später sehen werden.

Der Algorithmus 2.22 bricht mit $v = 0$ in einem Schleifendurchlauf zusammen, wenn A keinen vollen Spaltenrang hat. Um so einen Abbruch zu vermeiden, müssen bei der QR-Zerlegung vor jedem Teilschritt die Spalten der Submatrix A_{i+1} aus (2.14) derart permutiert werden (ähnlich zur Pivotsuche bei der Gauß-Elimination), dass die Euklidnorm ihrer ersten Spalte a_{i+1} maximal wird. Das wird **Spaltenpivotsuche** genannt. Mit dieser Spaltenpivotsuche bricht Algorithmus 2.22 erst dann zusammen, wenn die gesamte Restmatrix A_{i+1} aus (2.14) die Nullmatrix wird; in diesem Moment hat man eine Faktorisierung

$$Q^*A\Pi = \begin{bmatrix} R_1 & R_2 \\ 0 & 0 \end{bmatrix}$$

berechnet, wobei $\Pi \in \mathbb{K}^{n\times n}$ eine Permutationsmatrix, $R_1 \in \mathbb{K}^{p\times p}$ eine obere Dreiecksmatrix und R_2 eine unbestimmte (in der Regel voll besetzte) $p \times (n-p)$ Matrix ist. Diese Faktorisierung reicht allerdings nur aus, um **eine** Lösung - im Regelfall aber keine ausgezeichnete Lösung - zu berechnen.

Die QR-Zerlegung gehört zu den stabilsten Algorithmen in der numerischen linearen Algebra. Der Grund liegt darin, dass die Orthogonalitätstransformationen wegen $\mathrm{cond}_2(Q) = 1$ keinerlei Fehlerverstärkung hervorrufen.

Kapitel 3

Iterationsverfahren

Wenn die Matrizen sehr groß sind, verbieten sich Eliminationsverfahren wegen ihres hohen Aufwands. Zudem sind die großen, in der Praxis auftretenden Systeme meist dünn besetzt, d.h. nur wenige (etwa 10 Einträge pro Zeile) sind ungleich Null. Typische Beispiele sind Steifigkeitsmatrizen, die bei der Lösung von partiellen Differentialgleichungen auftreten. Obwohl die Matrix eines solchen Problems wegen der Dünnbesetztheit noch gut in den Speicher passen mag, trifft dies für die Faktorisierung L und R nicht mehr zu. In solchen Fällen behilft man sich gerne mit Iterationsverfahren, die das Gleichungssystem zwar nicht exakt, aber hinreichend genau lösen.

Bevor wir konkrete Verfahren vorstellen, wiederholen wir ein fundamentales Resultat aus der Analysis, den Banachschen Fixpunktsatz:

Satz 3.1. *Sei $\Phi : \mathcal{K} \to \mathcal{K}$ eine (nichlineare) bzgl. $\|\cdot\|$ kontrahierende Selbstabbildung einer abgeschlossenen Teilmenge $\mathcal{K} \subseteq \mathbb{K}^n$, d.h.,*

$$\|\Phi(x) - \Phi(y)\| \leq q\|x - y\| \text{ für ein } q < 1 \text{ und alle } x, y \in \mathcal{K} .$$

Dann hat die Fixpunktgleichung $x = \Phi(x)$ genau eine Lösung $\hat{x} \in \mathcal{K}$, und die Iterationsfolge $\{x^{(k)}\}$ mit $x^{(0)} \in \mathcal{K}$ beliebig, $x^{(k+1)} = \Phi(x^{(k)})$ für $k = 0, 1, 2, \cdots$, konvergiert gegen $\hat{x}$ für $k \to \infty$. Darüberhinaus ist für $k \geq 1$

1. $\|x^{(k)} - \hat{x}\| \leq q\|x^{(k-1)} - \hat{x}\|$ *(Monotonie);*
2. $\|x^{(k)} - \hat{x}\| \leq \frac{q^k}{1-q}\|x^{(1)} - x^{(0)}\|$ *(a-priori Schranke);*
3. $\|x^{(k)} - \hat{x}\| \leq \frac{q}{1-q}\|x^{(k)} - x^{(k-1)}\|$ *(a-posteriori Schranke);*

Beweis. 1. Wir wählen einen beliebigen Startwert $x^{(0)} \in \mathcal{K}$ und betrachten die durch $x^{(k+1)} = \Phi(x^{(k)})$, $k = 0, 1, 2, \cdots$, definierte Iterationsreihenfolge. Aufgrund der Kontraktionseigenschaft von Φ gilt für ein beliebiges $k \in \mathbb{N}$, dass

$$\|x^{(k+1)} - x^{(k)}\| = \|\Phi(x^{(k)}) - \Phi(x^{(k-1)})\| \leq q\|x^{(k)} - x^{(k-1)}\| \,. \tag{3.1}$$

Damit ergibt sich induktiv

$$\|x^{(k+1)} - x^{(k)}\| \leq q^k \|x^{(1)} - x^{(0)}\| \,, \quad k \in \mathbb{N} \,. \tag{3.2}$$

Wir zeigen nun, dass die Folge $x^{(k)}$ eine Cauchy-Folge ist. Dazu wählen wir $m, l \in \mathbb{N}$ mit $l > m$ und erhalten aus (3.2)

$$\begin{aligned} \|x^{(l)} - x^{(m)}\| &\leq \|x^{(l)} - x^{(l-1)}\| + \|x^{(l-1)} - x^{(l-2)}\| + \cdots \\ &\quad + \|x^{(m+1)} - x^{(m)}\| \\ &\leq (q^{l-1} + q^{l-2} + \cdots + q^m)\|x^1 - x^{(0)}\| \\ &\leq q^m \frac{1}{1-q} \|x^{(1)} - x^{(0)}\| \,. \end{aligned} \tag{3.3}$$

Da q^m für $m \to \infty$ gegen Null konvergiert, wird der letzte Ausdruck mit hinreichend großem m kleiner als jede positive Zahl ε. Daher ist $\{x^{(k)}\}$ eine Cauchy-Folge mit Grenzwert x. Da alle Iterierten wegen der Selbstabbildungseigenschaft in der *abgeschlossenen* Menge $\mathcal{K}$ bleiben, gehört auch x zu $\mathcal{K}$.

2. Als nächstes wird gezeigt, dass x ein Fixpunkt von Φ ist. Dazu beachte man zunächst, dass Φ (Lipschitz)-stetig ist. Folglich kann man in der Rekursion $x^{(k+1)} = \Phi(x^{(k)})$ den Iterationsindex k gegen ∞ gehen lassen und während die linke Seite gegen x konvergiert, konvergiert die rechte Seite wegen der Stetigkeit von Φ gegen $\Phi(x)$. Also ist $x = \Phi(x)$, bzw. x ist ein Fixpunkt von Φ. Damit ist die Existenz eines Fixpunktes nachgewiesen.

3. Die Eindeutigkeit folgt aus der Kontraktionseigenschaft: Angenommen $\hat{x}$ und x seien zwei Fixpunkte von Φ in $\mathcal{K}$; dann folgt

$$\|x - \hat{x}\| = \|\Phi(x) - \Phi(\hat{x})\| \leq q\|x - \hat{x}\| \,,$$

und dies kann wegen der Voraussetzung $q < 1$ nur gelten, wenn $\|x - \hat{x}\| = 0$, also $x = \hat{x}$ ist. Mit anderen Worten: Φ hat in $\mathcal{K}$ nur einen

Fixpunkt $\hat{x}$ und die Iterationsreihenfolge $\{x^{(k)}\}$ konvergiert für jedes $x^{(0)}$ gegen $\hat{x}$.

4. Es verbleibt der Nachweis der drei Fehlerabschätzungen. Die erste Ungleichung folgt wie vorher die Eindeutigkeit:

$$\|x^{(k)} - \hat{x}\| = \|\Phi(x^{(k-1)}) - \Phi(\hat{x})\| \leq q\|x^{(k-1)} - \hat{x}\| \,.$$

Die zweite Ungleichung folgt aus (3.3): Demnach ist für $m > k$

$$\|x^{(m)} - x^{(k)}\| = q^k \frac{1}{1-q}\|x^{(1)} - x^{(0)}\| \,,$$

und die Behauptung ergibt sich durch Grenzübergang $m \to \infty$. Zum Beweis der dritten Ungleichung schätzen wir die linke Seite von (3.1) mit der Dreiecksungleichung ab und erhalten zusammen mit $q < 1$

$$\begin{aligned} \|x^{(k+1)} - x^{(k)}\| &\geq \|x^{(k)} - \hat{x}\| - \|x^{(k+1)} - \hat{x}\| \\ &\geq \|x^{(k)} - \hat{x}\| - q\|x^{(k+1)} - \hat{x}\| \\ &= (1-q)\|x^{(k)} - \hat{x}\| \,. \end{aligned} \tag{3.4}$$

Eingesetzt in (3.1) folgt daraus die Behauptung des Satzes.

□

Der Banachsche Fixpunktsatz lässt sich nun wie folgt zur Konstruktion konvergenter Iterationsverfahren zur Lösung nichtsingulärer linearer Gleichungssysteme $Ax = b$ mit $A \in \mathbb{K}^{n\times n}$ und $b \in \mathbb{K}^n$ verwenden: Man wählt eine additive Zerlegung von A,

$$A = M - N \,,$$

wobei M invertierbar sein soll und bringt die Gleichung $Ax = b$ auf "Fixpunktgestalt"

$$Mx = Nx + b \text{ bzw. } x = Tx + c \tag{3.5}$$

mit $T = M^{-1}N$ und $c = M^{-1}b$; die rechte Seite $Tx + c$ entspricht also der (hier affin linearen) Funktion $\Phi(x)$ aus Satz 3.1.

Algorithmus 3.2. (Allgemeines Iterationsprinzip für lineare Gleichungssysteme)

- `Wähle` $A = M - N$ `mit invertierbarem` M `und` $x^{(0)} \in \mathbb{K}^n$ `beliebig`

- `for` $k = 1, \cdots$ `löse`

$$Mx^{(k)} = Nx^{(k-1)} + b\,. \tag{3.6}$$

Es ist offensichtlich, dass ein solches Verfahren nur dann sinnvoll ist, wenn Gleichungssysteme mit der Matrix M erheblich einfacher zu lösen sind als Gleichungssysteme mit A, und wenn die Matrix-Vektor-Multiplikation mit N billig ist (etwa, wenn N dünnbesetzt ist). Zur Konvergenz dieses Verfahrens gibt der Banachsche Fixpunktsatz die folgende Auskunft:

Satz 3.3. *Ist $|||\cdot|||$ eine Matrixnorm, die mit der Vektornorm $\|\cdot\|$ verträglich ist, und ist*

$$\left|\left|\left|M^{-1}N\right|\right|\right| < 1\,,$$

dann konvergiert das Iterationsverfahren (3.6) für jedes $x^{(0)}$ gegen $A^{-1}b$.

Beweis. Wir setzen $\Phi(x) = Tx + c$ mit $T = M^{-1}N$ und $c = M^{-1}b$. Aus (3.5) ist offensichtlich, dass alle Lösungen von $Ax = b$ auch Fixpunkte der Fixpunktgleichung $x = \Phi(x)$ sind und umgekehrt. $\mathcal{K} = \mathbb{K}^n$ ist abgeschlossen und wegen der Linearität von T folgt

$$\|\Phi(x) - \Phi(z)\| = \|T(x - z)\| \le |||T|||\,\|x - z\|\,,$$

und damit ist auch die zweite Voraussetzung des Banachschen Fixpunktsatzes mit $q = |||M^{-1}N|||$ erfüllt. Also konvergiert die Folge $\{x^{(k)}\}$ aus (3.6) gegen den eindeutigen Fixpunkt $\hat{x} = T\hat{x} + c$, also die eindeutige Lösung $\hat{x} = A^{-1}b$ des linearen Gleichungssystems. □

Korollar 3.4. *Sei $A = M - N$ invertierbar und $T = M^{-1}N$. Dann konvergiert das Iterationsverfahren (3.6) genau dann für jedes $x^{(0)}$ gegen $\hat{x} = A^{-1}b$, wenn für den Spektralradius $\rho(T)$ von T die Ungleichung $\rho(T) < 1$ erfüllt ist.*

Beweis. Falls $\rho(T) < 1$ ist, dann existiert eine Vektornorm $\|\cdot\|_\varepsilon$ und eine dadurch induzierte Matrixnorm $|||\cdot|||_\varepsilon$ mit $q := |||T|||_\varepsilon \le \rho(T) + \varepsilon < 1$ (Übungsbeispiel!). Damit ergibt sich eine Beweisrichtung aus Satz 3.3.

Ist umgekehrt $\rho(T) \ge 1$, dann existiert ein Eigenwert λ von A mit $|\lambda| \ge 1$ und zugehörigem Eigenvektor $z \ne 0$. Wählt man $x^{(0)} = A^{-1}b + z$, dann ergibt sich

$$\begin{aligned} x^{(k)} - \hat{x} &= Tx^{(k-1)} + c - \hat{x} \\ &= M^{-1}(Nx^{(k-1)} + b - M\hat{x}) \\ &= M^{-1}(Nx^{(k-1)} - N\hat{x}) \\ &= T(x^{(k-1)} - \hat{x})\,. \end{aligned}$$

Daraus folgt mit Induktion

$$x^{(k)} - \hat{x} = T^k(x^{(0)} - \hat{x}) = T^k z = \lambda^k z \; . \tag{3.7}$$

Folglich ist $\|x^{(k)} - \hat{x}\| = |\lambda|^k \|z\| \geq \|z\|$ und $x^{(k)}$ konvergiert für $k \to \infty$ nicht gegen $A^{-1}b$. □

Dem Spektralradius von T kommt also bei der Iteration (3.6) eine besondere Bedeutung zu. Gemäß Korollar 3.4 entscheidet der Spektralradius über Konvergenz und Divergenz der Iterationsfolge. Darüberhinaus bestimmt er aber auch die asymptotische Konvergenzgeschwindigkeit:

Satz 3.5. *Unter den Voraussetzungen von Korollar 3.4 gilt*

$$\max_{x^{(0)}} \limsup_{k\to\infty} \|\hat{x} - x^{(k)}\|^{1/k} = \rho(T) \; .$$

Beweis. Wie in (3.7) im Beweis zu Korollar 3.4, sieht man anhand des Eigenpaars (λ, z) von T mit $z \neq 0$, dass

$$\begin{aligned} \max_{x^{(0)}} \limsup_{k\to\infty} \|x^{(k)} - \hat{x}\|^{1/k} &\geq \limsup_{k\to\infty} \|T^k z\|^{1/k} \\ &= \limsup_{k\to\infty} |\lambda| \|z\|^{1/k} \\ &= |\lambda| \; . \end{aligned} \tag{3.8}$$

Also ist $\rho(T)$ mit Sicherheit eine untere Schranke.

Unter Verwendung der Vektornorm $\|\cdot\|_\varepsilon$ und der induzierten Matrixnorm $|||\cdot|||_\varepsilon$ aus dem Beweis von Korollar 3.4 ergibt sich ferner

$$\|x^{(k)} - \hat{x}\|_\varepsilon = \|T^k(x^{(0)} - \hat{x})\|_\varepsilon \leq |||T|||_\varepsilon^k \, \|x^{(0)} - \hat{x}\|_\varepsilon \; .$$

Wegen der Äquivalenz aller Normen in $\mathbb{K}^n$ folgt daher für ein geeignetes $c_\varepsilon > 0$:

$$\|x^{(k)} - \hat{x}\|^{1/k} \leq (c_\varepsilon \|x^{(k)} - \hat{x}\|_\varepsilon)^{1/k} \leq |||T|||_\varepsilon \, (c_\varepsilon \|x^{(0)} - \hat{x}\|_\varepsilon)^{1/k} \to |||T|||_\varepsilon$$

für $k \to \infty$. Folglich ist wegen (3.8)

$$\rho(T) \leq \max_{x^{(0)}} \limsup_{k\to\infty} \|x^{(k)} - \hat{x}\|^{1/k} \leq |||T|||_\varepsilon \leq \rho(T) + \varepsilon \, ,$$

und da $\varepsilon > 0$ beliebig klein gewählt werden kann, folgt hieraus die Behauptung. □

Beachte, dass das Resultat unabhängig von der betrachteten Vektornorm gilt, und somit ist das *asymptotische* Konvergenzverhalten der Iteration unabhängig von der betrachteten Norm.

Definition 3.6. Ausgehend von Satz 3.5 nennt man die Zahl $\rho(T)$ auch **(asymptotischen) Konvergenzfaktor** der Iteration (3.6). Die Zahl $-\log_{10}\rho(T)$ gibt die **(asymptotische) Konvergenzrate** an.

Als Faustregel kann man annehmen, dass etwa je $1/r$ Iterationsschritte für jede zusätzliche signifikante Dezimalstelle benötigt werden. Dies ergibt sich aus folgenden überlegungen: Laut Satz 3.5 gilt

$$\|x^{(k)} - \hat{x}\| \approx \rho(T)^k \text{ und } \|x^{(k+r)} - \hat{x}\| \approx \rho(T)^{k+r} ;$$

Somit gilt

$$\frac{\|x^{(k+r)} - \hat{x}\|}{\|x^{(k)} - \hat{x}\|} \approx \rho(T)^r .$$

Ein signifikante Stelle mehr nach r Iterationsschritte bedeutet, dass

$$0.1 = \frac{\|x^{(k+r)} - \hat{x}\|}{\|x^{(k)} - \hat{x}\|} \approx \rho(T)^r ,$$

also

$$-1 = \log_{10}(0.1) = r \log_{10}(\rho(T)) .$$

Im folgenden studieren wir zwei spezielle Iterationsverfahren, die von großer Bedeutung bei der Lösung von linearen Gleichungssystemen sind:

3.1 Einzel- und Gesamtschrittverfahren

Das einfachste Beispiel eines Iterationsverfahrens zur Lösung eines linearen Gleichungssystems $Ax = b$ mit $A = [a_{ij}]_{ij} \in \mathbb{K}^{n\times n}$ und $b = [b_i]_i \in \mathbb{K}^n$ ist das **Gesamtschrittverfahren** (oder **Jacobi-Verfahren**). Bezeichnen wir mit $x^{(k)} = [x_j^{(k)}]_j \in \mathbb{K}^n$ die Iterierten dieses Verfahrens, dann lautet die Iterationsvorschrift wie folgt:

Algorithmus 3.7. (Gesamtschrittverfahren)
`Wähle` $x^{(0)} \in \mathbb{K}^n$ `beliebig.`
`for` $k = 0, 1, \cdots$

- `for` $i = 1, \cdots, n$

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j \neq i} a_{ij} x_j^{(k)} \right) \tag{3.9}$$

`end for`

`until stop`

Offensichtlich muss für die Durchführbarkeit dieser Iterationsvorschrift $a_{ii} \neq 0\,, i = 1, \cdots, n$ vorausgesetzt werden.

Anhand von (3.9) erkennt man, dass in jedem Iterationsschritt (für jedes k) genau eine Multiplikation oder Division mit jedem von Null verschiedenen Eintrag von A nötig ist. Also ist im Fall von dünnbesetzten Matrizen ein Iterationsschritt relativ billig.

Die Frage nach der Konvergenz werden wir auf Satz 3.3 zurückführen. Dazu zerlegen wir

$$A = D - L - R$$

in eine Diagonal- und eine strikte linke untere und eine strikte rechte obere Dreiecksmatrix. Dann können die n Gleichungen (3.9), $i = 1, \cdots, n$, als eine Vektorgleichung

$$x^{(k+1)} = D^{-1}(b + (L + R)x^{(k)}) \tag{3.10}$$

geschrieben werden. Somit entspricht das Gesamtschrittverfahren der Fixpunktiteration (3.6) mit $M = D$ und $N = L + R$. Die entsprechende Iterationsmatrix

$$\mathcal{J} = M^{-1}N = D^{-1}(L + R)$$

wird **Gesamtschrittoperator** genannt.

Beim **Einzelschritt-** oder **Gauß-Seidel-Verfahren** setzt man in (3.9) alle bereits berechneten Komponenenten von $x^{(k+1)}$ auf der rechten Seite ein:

Algorithmus 3.8. (Einzelschrittverfahren) `Wähle` $x^{(0)} \in \mathbb{K}^n$ `beliebig.` `for` $k = 0, 1, \cdots$

- `for` $i = 1, \cdots, n$

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j<i} a_{ij} x_j^{(k+1)} - \sum_{j>i} a_{ij} x_j^{(k)} \right) \tag{3.11}$$

`end for`

```
until stop
```

Der Aufwand ist somit der gleiche wie beim Gesamtschrittverfahren. Entsprechend zu (3.10) erhält man die Matrixformulierung des Einzelschrittverfahren, indem man alle Komponenten von $x^{(k+1)}$ in (3.11) auf die linke Seite bringt. Dann folgt:

$$a_{ii}x_i^{(k+1)} + \sum_{j<i} a_{ij}x_j^{(k+1)} = b_i - \sum_{j>i} a_{ij}x_j^{(k)} \quad i = 1, \cdots n \ .$$

Insbesondere ergibt sich $x^{(k+1)}$ durch Auflösen des Dreieckssystems

$$(D - L)x^{(k+1)} = b + Rx^{(k)} \ . \tag{3.12}$$

Wiederum haben wir eine Fixpunktiteration der Form (3.6), diesmal mit $M = D - L$ und $N = R$. $\mathcal{L} = (D - L)^{-1}R$ wird **Einzelschrittoperator** genannt.

Mit Hilfe des Banachschen Fixpunktsatzes können folgende Aussagen über Konvergenz von Einzel-und Gesamtschrittverfahren bewiesen werden.

Satz 3.9. *Ist A strikt diagonaldominant, dann konvergiert Gesamt- und Einzelschrittverfahren für jeden Startvektor $x^{(0)}$ gegen die eindeutige Lösung von $Ax = b$.*

Beweis. Wir wollen Satz 3.3 anwenden. Zunächst betrachten wir das Gesamtschrittverfahren. Aus der strikten Diagonaldominanz (vgl. Definition 2.10) von A folgt unmittelbar

$$\|\mathcal{J}\|_\infty = \|D^{-1}(L + R)\|_\infty = \frac{1}{|a_{ii}|} \max_{i=1\cdots n} \sum_{j\neq i} |a_{ij}| =: q < 1 \ .$$

Das bedeutet, die Voraussetzung von Satz 3.3 (Banachscher Fixpunktsatz) ist für die Zeilensummennorm erfüllt.

Für das Einzelschrittverfahren ist der Beweis etwas komplizierter. Wieder verwenden wir die Zeilensummennorm und müssen nun nachweisen, dass

$$\|\mathcal{L}\|_\infty = \max_{\|x\|_\infty=1} \|\mathcal{L}x\|_\infty < 1 \ .$$

Sei also $\|x\|_\infty = 1$ und q wie zuvor definiert. Setzt man $y := \mathcal{L}x$, dann ergeben sich entsprechend zu (3.11) mit $b := 0$, $x^{(k)} := x$ und $y := x^{(k+1)}$, die

Komponenten y_i von y:

$$y_i = \frac{1}{a_{ii}} \left(-\sum_{j<i} a_{ij} y_j - \sum_{i<j} a_{ij} x_j \right) . \tag{3.13}$$

Wir zeigen nun induktiv, dass $|y_j| \leq q < 1$ für alle $j < i$ gilt.

Für $i = 1$ gilt

$$|y_1| = \left| \frac{1}{a_{11}} \left(\sum_{j>i} a_{ij} x_j \right) \right| \leq \|x\|_\infty \frac{1}{|a_{ii}|} \sum_{j>i} |a_{ij}| \leq q < 1 .$$

Im Induktionsschritt "$i-1 \to i$ schätzen wir in (3.13) $|y_i|$ mit der Dreiecksungleichung wie folgt ab:

$$\begin{aligned} |y_i| &\leq \frac{1}{|a_{ii}|} \left(\sum_{j<i} |a_{ij}||y_j| + \sum_{j>i} |a_{ij}||x_j| \right) \\ &\leq \frac{1}{|a_{ii}|} \left(\sum_{j<i} |a_{ij}| q + \sum_{j>i} |a_{ij}| \|x\|_\infty \right) \\ &\leq \frac{1}{|a_{ii}|} \left(\sum_{j<i} |a_{ij}| + \sum_{j>i} |a_{ij}| \right) \\ &= q. \end{aligned}$$

Für $i = n$ ergibt die Induktionsaussage schließlich $\|y\|_\infty \leq q$, und somit $\|\mathcal{L}\|_\infty \leq q$.

□

Beispiel 3.10. Gegeben sei das lineare Gleichungssystem $Ax = b$ mit

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

Die Lösung lautet $x^t = [1, -1, -1]$.

A ist strikt diagonaldominant: Der Faktor q aus dem Beweis von Satz 3.9 ist $q = 1/2$. Für den Startvektor $x^{(0)} = [1, 1, 1]^t$ ist $\|x^{(0)} - \hat{x}\|_\infty = 2$ und es ergibt sich

- bei dem Gesamtschrittverfahren $x_{\mathcal{J}}^{(1)} = (0, 1/2, 0)^t$ mit Fehler $\|x_{\mathcal{J}}^{(1)} - \hat{x}\|_\infty = 1$;
- bei dem Einzelschrittverfahren $x_{\mathcal{L}}^{(1)} = (0, -3/4, -1/8)^t$ mit Fehler $\|x_{\mathcal{L}}^{(1)} - \hat{x}\|_\infty = 1$.

Bezüglich der Maximumnorm wird der Fehler also tatsächlich in beiden Fällen genau um den Faktor 2 reduziert.

In der Tat konvergiert das Einzelschrittverfahren häufig schneller als das Gesamtschrittverfahren – es lassen sich aber auch Beispiele konstruieren, für die das Gesamtschrittverfahren schneller konvergiert.

Für spezielle Matrizen A lassen sich derartige Vergleiche präzisieren. Wir studieren Matrizen der Form

$$A = \begin{bmatrix} I & -B^* \\ -B & I \end{bmatrix} \in \mathbb{K}^{n\times n}, \tag{3.14}$$

mit $B \in \mathbb{K}^{p\times q}$, $0 < p, q < n$. Im vorliegenden Fall ist $D = I$ und

$$L = \begin{bmatrix} 0 & 0 \\ B & 0 \end{bmatrix}, \quad R = \begin{bmatrix} 0 & B^* \\ 0 & 0 \end{bmatrix}.$$

Daher ergeben sich für Einzel-und Gesamtschrittverfahren die Iterationsmatrizen

$$\mathcal{L} = (D - L)^{-1}R = \begin{bmatrix} I & 0 \\ -B & I \end{bmatrix}^{-1} \begin{bmatrix} 0 & B^* \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & B^* \\ 0 & BB^* \end{bmatrix} \tag{3.15}$$

sowie

$$\mathcal{J} = \begin{bmatrix} 0 & B^* \\ B & 0 \end{bmatrix} \text{ und daher } \mathcal{J}^2 = \begin{bmatrix} B^*B & 0 \\ 0 & BB^* \end{bmatrix}. \tag{3.16}$$

Das folgende Lemma liefert uns nun ein Resultat, aus welchem wir schließen werden können, dass für unsere gegebene Problemklasse das Einzelschrittverfahren schneller konvergiert.

Lemma 3.11. *Es sei $A \in \mathbb{K}^{(p\times q)}$, $B \in \mathbb{K}^{(q\times p)}$ und $C \in \mathbb{K}^{(n\times n)}$. Dann gilt:*

1. *Bis auf den möglichen Eigenwert 0 sind die Spektren von $AB \in \mathbb{K}^{p\times p}$ und $BA \in \mathbb{K}^{q\times q}$ gleich;*
2. $\sigma(C^2) = \{\lambda^2 : \lambda \in \sigma(C)\}$.

Beweis. 1. Ist $0 \neq \lambda \in \sigma(AB)$, dann existiert ein Eigenvektor $u \neq 0$ mit $ABu = \lambda u$. Wegen $\lambda u \neq 0$ ist auch $v := Bu \neq 0$ und es gilt

$$BAv = B(ABu) = B(\lambda u) = \lambda Bu = \lambda v \ .$$

Folglich ist $\lambda \in \sigma(BA)$ und $\sigma(AB)\backslash\{0\} \subseteq \sigma(BA)$. Mit dem gleichen Argument sieht man auch, dass $\sigma(BA)\backslash\{0\} \subseteq \sigma(AB)$.

2. Ist $\lambda \in \sigma(C)$, dann existiert $x \neq 0$ mit $Cx = \lambda x$ und damit ergibt sich $C^2x = C(\lambda x) = \lambda Cx = \lambda^2 x$. Also ist $\lambda^2 \in \sigma(C^2)$. Ist umgekehrt $\mu \in \sigma(C^2)$ und sind $\mp\lambda$ die beiden (komplexen) Wurzeln von μ, dann gilt

$$0 = \det\,(C^2 - \mu I) = \det\,((C-\lambda I)(C+\lambda I)) = \det\,(C-\lambda I)\det\,(C+\lambda I) \ .$$

Daher ist entweder λ oder $-\lambda$ im Spektrum von C, was zu zeigen war. □

Dieses Resultat ermöglicht es uns, nun Einzel- und Gesamtschrittverfahren für die spezielle Klasse von Matrizen aus Beispiel 3.10 zu vergleichen.

Satz 3.12. *Hat A die Form (3.14), dann gilt* $\rho(\mathcal{L}) = \rho(\mathcal{J})^2$.

Beweis. Nach (3.15) gilt

$$\begin{aligned} \det\,(\mathcal{L} - \lambda\mathcal{I}) &= \det\,((D-L)^{-1}R - \lambda I) \\ &= \det \begin{bmatrix} -\lambda I & B^* \\ 0 & BB^* - \lambda I \end{bmatrix} \\ &= \det\,(-\lambda I)\det\,(BB^* - \lambda I) \ . \end{aligned}$$

Also ist $\sigma(\mathcal{L}) = \{0\} \cup \sigma(BB^*)$ und $\rho(\mathcal{L}) = \rho(BB^*)$ (ρ beschreibt den betragsgrößten Eigenwert).

Andererseits ist wegen (3.16)

$$\sigma(\mathcal{J}^2) = \sigma(BB^*) \text{ ggf. zuzüglich des Eigenwerts } 0 \ ,$$

so dass nach Lemma 3.11 die Gleichheitskette

$$\rho(\mathcal{J})^2 = \rho(\mathcal{J}^2) = \rho(BB^*) = \rho(\mathcal{L})$$

gültig ist. □

Satz 3.12 zusammen mit Korollar 3.4 und Satz 3.5 liefert folgende Aussagen:

1. Für Matrizen der Form (3.14) ist entweder $\rho(\mathcal{J}) < 1$ und sowohl das Gesamt- als auch das Einzelschrittverfahren zu konvergieren, oder es ist $\rho(\mathcal{J}) \geq 1$, dann können beide Verfahren divergieren.

2. Im konvergenten Fall braucht das Einzelschrittverfahren bei einer vorgegebenen Genauigkeit etwa die Hälfte der Iterationen wie das Gesamtschrittverfahren.

3.2 Das Verfahren der konjugierten Gradienten

Das vermutlich effizienteste Verfahren zur Lösung von linearen Gleichungssystemen $Ax = b$, deren Koeffizientenmatrix $A \in \mathbb{R}^{n\times n}$ hermitesch und positiv definit ist (der Einfachheit halber beschränken wir uns nur auf reelle (symmetrische) Matrizen. Das besagte Verfahren lässt sich nicht in das allgemeine Schema einer Fixpunktiteration einordnen.

Sei

$$\Phi(x) = \frac{1}{2}x^*Ax - x^*b : \mathbb{R}^n \to \mathbb{R} \,.$$

Setzen wir $\hat{x} = A^{-1}b$, dann ergibt eine einfache Rechnung, dass

$$\begin{aligned}
\Phi(x) - \Phi(\hat{x}) &= \frac{1}{2}x^*Ax - x^*b - \frac{1}{2}\hat{x}^*A\hat{x} + \hat{x}^*b \\
&= \frac{1}{2}(x-\hat{x})^*A(x-\hat{x}) + x^*A\hat{x} - \hat{x}^*Ax - x^*b + \hat{x}^*b \\
&= \frac{1}{2}(x-\hat{x})^*A(x-\hat{x}) + x^*b - \hat{x}^*b - x^*b + \hat{x}^*b \\
&= \frac{1}{2}(x-\hat{x})^*A(x-\hat{x}) \,.
\end{aligned}$$

Da A positiv definit ist, ist der letzte Ausdruck nichtnegativ und genau dann Null, wenn $x = \hat{x}$ ist. Mit anderen Worten: Das Funktional Φ hat ein eindeutiges Minimum an der Stelle $x = \hat{x}$.

Definition 3.13. Ist A hermitesch und positiv definit, dann definiert

$$\|x\|_A := \sqrt{x^*Ax}\,, \quad x \in \mathbb{R}^n \,,$$

eine Norm in $\mathbb{R}^n$, die so genannte **Energienorm**. Zu einer Energienorm gehört ein **Skalarprodukt** (dies induziert geometrische Begriffe wie z.B. **Orthogonalität**), nämlich

$$\langle x, y \rangle_A = x^* A y \,, \quad x, y \in \mathbb{R}^n \,.$$

Demnach ist die Abweichung des Funktionals Φ von seinem Minimum,

$$\Phi(x) - \Phi(\hat{x}) = \frac{1}{2}(x - \hat{x})^* A (x - \hat{x}) = \frac{1}{2}\|x - \hat{x}\|_A^2 \,, \tag{3.17}$$

ein gut geeignetes Fehlermaß für den Abstand zwischen x und $\hat{x}$. Geometrisch bedeutet (3.17), dass der Graph der Funktion Φ bezüglich der Energienorm ein kreisförmiges Paraboloid ist, dessen Mittelpunkt in $\hat{x}$ liegt.

Wir konstruieren nun ein Verfahren zur Approximation von $\hat{x}$, welches das Funktional Φ sukzessive minimiert. Ist $x^{(k)}$ die aktuelle Iterierte, dann kann man beispielsweise eine geeignete *Suchrichtung* $d^{(k)} \neq 0$ für den nächsten Schritt wählen und $x^{(k+1)}$ über den Ansatz

$$x^{(k+1)} = x^{(k)} + \alpha d^{(k)} \tag{3.18}$$

bestimmen. In Abhängigkeit von α ergibt sich

$$\Phi(x^{(k)} + \alpha d^{(k)}) = \Phi(x^{(k)}) + \alpha d^{(k)*} A x^{(k)} + \frac{1}{2}\alpha^2 d^{(k)*} A d^{(k)} - \alpha d^{(k)*} b \,. \tag{3.19}$$

Durch Differentiation nach α erhält man die Schrittweite, für die der Wert von Φ minimal wird, nämlich

$$\alpha_k = \frac{(b - A x^{(k)})^* d^{(k)}}{d^{(k)*} A d^{(k)}} =: -\frac{r^{(k)*} d^{(k)}}{d^{(k)*} A d^{(k)}} \,. \tag{3.20}$$

Dabei ist der Nenner ungleich Null, da A positiv definit ist.

Offen in diesem allgemeinen Schema ist noch die Wahl der Suchrichtung. Aus (3.19) berechnet man die Richtungsableitung von Φ in Richtung $d^{(k)}$ wie folgt:

$$\lim_{\alpha \to 0} \frac{\Phi(x^{(k)} + \alpha d^{(k)}) - \Phi(x^{(k)})}{\alpha} = d^{(k)*}(A x^{(k)} - b);$$

demnach ist

$$\operatorname{grad} \Phi(x) = A x - b \,.$$

Der negative Gradient gibt die Richtung des steilsten Abfalls von Φ an. Für unser Funktional Φ stimmt die Richtung zufälligerweise mit dem Residuum überein. Das Residuum muss jedoch nicht die bestmögliche Wahl sein.

Bei dem Verfahren der konjugierten Gradienten macht man den Ansatz

$$d^{(k+1)} = r^{(k+1)} + \beta_k d^{(k)} \text{ mit } \left\langle d^{(k+1)}, d^{(k)} \right\rangle_A = 0 \, . \tag{3.21}$$

Die Bedingung $\left\langle d^{(k+1)}, d^{(k)} \right\rangle_A = 0$ bedeutet, dass die beiden Suchrichtungen $d^{(k+1)}$ und $d^{(k)}$ zueinander orthogonal (bzgl. des inneren Produktes $< \cdot, \cdot >_A$) stehen, was den Namen "Verfahren der konjugierten Richtungen" (oder CG-Verfahren - conjugate gradient method) motiviert.

Diese resultierende Bedingung an β lautet

$$\beta_k = -\frac{r^{(k+1)*} A d^{(k)}}{d^{(k)*} A d^{(k)}} \, . \tag{3.22}$$

Wie bereits bemerkt, ist das Verfahren (3.20) und (3.22) nur dann wohldefiniert, wenn $d^{(k+1)}$ ungleich Null ist. Aus (3.21) sieht man, dass $d^{(k+1)}$ nur Null werden kann, wenn $d^{(k)}$ und $r^{(k+1)}$ linear abhängig sind.

Nun überlegen wir uns, dass das Verfahren wohldefiniert ist. Genauer, für alle Iterierten $x^{(k)} \neq \hat{x}$ ist $d^{(k)} \neq 0$.

Dazu bemerken wir zuerst, dass gemäß der Definition von α_k die Gerade $x^{(k)} + \alpha d^{(k)}$ die Niveaulinie zum Wert von $\Phi(x^{(k+1)})$ im Punkt $x^{(k+1)}$ berührt. Da wegen (3.20) das Minimum eindeutig ist, kann die Gerade die Niveaulinie nicht schneiden. Da $d^{(k)}$ tangential zur Niveaulinie von Φ mit Wert $\Phi(x^{(k+1)}$ ist, und $r^{(k+1)}$ eine Abstiegsrichtung ist, können $r^{(k+1)}$ und $d^{(k)}$ nur dann linear abhängig sein, wenn $r^{(k)} = 0$ ist, also $x^{(k)}$ die Lösung des linearen Gleichungssystems ist.

Wir weisen jetzt Eigenschaften des Verfahrens der konjugierten Gradienten nach.

Lemma 3.14. *Sei $x^{(0)}$ ein beliebiger Startvektor und $d^{(0)} = r^{(0)}$. Falls $x^{(k)} \neq \hat{x}$ ist für $k = 0, \cdots, m$, dann gilt*

1. $r^{(m)*} d^{(j)} = 0$, *für alle* $0 \leq j < m$,
2. $r^{(m)*} r^{(j)} = 0$, *für alle* $0 \leq j < m$,
3. $\left\langle d^{(m)}, d^{(j)} \right\rangle_A = 0$, *für alle* $0 \leq j < m$.

Beweis. Wir bemerken zunächst, dass $Ax^{(k+1)} = Ax^{(k)} + \alpha_k Ad^{(k)}$ ist; folglich gilt

$$r^{(k+1)} = Ax^{(k+1)} - b = Ax^{(k)} - b + \alpha_k Ad^{(k)} = r^{(k)} + \alpha_k Ad^{(k)}\,, \quad k \geq 0\,. \tag{3.23}$$

Daher bewirkt die Wahl von α_k gemäß (3.20), dass für $x^{(k)} \neq \hat{x}$

$$r^{(k+1)*} d^{(k)} = (r^{(k)} + \alpha_k Ad^{(k)})^* d^{(k)} = r^{(k)*} d^{(k)} + \alpha_k d^{(k)*} Ad^{(k)} = 0\,. \tag{3.24}$$

Nach dieser einführenden Bemerkung führen wir einen Induktionsbeweis:

$\underline{m = 1}$: Setzt man $k = 0$ in (3.24), dann gilt $r^{(1)*} d^{(0)} = 0$ und somit ergibt das gerade die Behauptung 1 für $m = 1$ und $j = 0$. Da $r^{(0)} = d^{(0)}$ folgt aus Behauptung 1 sofort auch Behauptung 2. Der dritte Teil ist trivial: $\left\langle d^{(1)}, d^{(0)} \right\rangle_A = 0$ wegen der Definition des Verfahrens.

$\underline{m \to m+1}$: Im Induktionsschritt nehmen wir an, dass *alle drei* Aussagen für ein m erfüllt sind. Zunächst wissen wir aus (3.24), dass $r^{(m+1)*} d^{(m)} = 0$. Aus (3.23) folgt zusammen mit den beiden Induktionsannahmen 1 und 3, dass

$$r^{(m+1)*} d^{(j)} = r^{(m)*} d^{(j)} + \alpha_m \left\langle d^{(m)}, d^{(j)} \right\rangle_A = 0 - 0\,, \text{ für } 0 \leq j < m\,. \tag{3.25}$$

Folglich gilt der erste Teil der Behauptung auch für $m + 1$.

Wegen (3.21) ist $r^{(j)} = d^{(j)} - \beta_{j-1} d^{(j-1)}$, $1 \leq j \leq m$, und $r^{(0)} = d^{(0)}$. Damit gilt wegen des ersten Teils des schon Bewiesenen auch für $1 \leq j \leq m$

$$r^{(m+1)*} r^{(j)} = r^{(m+1)*} d^{(j)} - \beta_{j-1} r^{(m+1)*} d^{(j-1)} = 0 - 0\,.$$

Zum Beweis des dritten Teils der Behauptung: $\left\langle d^{(m+1)}, d^{(m)} \right\rangle_A = 0$ folgt unmittelbar aus der Definition des Verfahrens, und somit haben wir die Behauptung für $j = m$ verifiziert. Für $j < m$ ergibt sich aus (3.21) und der Induktionsannahme, dass

$$\left\langle d^{(m+1)}, d^{(j)} \right\rangle_A = \left\langle r^{(m+1)}, d^{(j)} \right\rangle_A + \beta_m \left\langle d^{(m)}, d^{(j)} \right\rangle_A = r^{(m+1)*} Ad^{(j)}\,.$$

Ersetzt man nun $Ad^{(j)}$ gemäß (3.23) und (3.21), dann ergibt sich

$$\begin{aligned}
\alpha_j \left\langle d^{(m+1)}, d^{(j)} \right\rangle_A &= \alpha_j d^{(m+1)*} A d^{(j)} \\
&\underbrace{=}_{(3.21)} r^{(m+1)*}(r^{(j+1)} - r^{(j)}) + \beta_m d^{(m)*}(r^{(j+1)} - r^{(j)}) \\
&= 0 + \beta_m d^{(m)*}(r^{(j+1)} - r^{(j)}) \\
&\underbrace{=}_{(3.23)} \beta_m d^{(m)*} \alpha_j A d^{(j)} \\
&= \beta_m \alpha_j \left\langle d^{(m)}, d^{(j)} \right\rangle_A \\
&\underbrace{=}_{Induktionsannahme} 0 \,.
\end{aligned}$$

Wenn wir noch zeigen, dass $\alpha_j \neq 0$ für $0 \leq j < m$, dann ist die Behauptung vollständig bewiesen. Nehmen wir an, es gelte $\alpha_j = 0$: Wegen (3.20) ist das gleichbedeutend mit $r^{(j)*} d^{(j)} = 0$ und aus (3.21) und der Induktionsannahme folgt somit

$$\begin{aligned}
0 &= r^{(j)*}(r^{(j)} + \beta_{j-1} d^{(j-1)} \\
&= r^{(j)*} r^{(j)} + \beta_{j-1} r^{(j)*} d^{(j-1)} \\
&= \|r^{(j)}\|_2^2 \qquad \text{falls } 0 < j < m \,, \\
0 &= r^{(0)*} d^{(0)} = \|r^{(0)}\|_2^2 \text{ falls } j = 0 \,.
\end{aligned}$$

Da $x^{(j)} \neq \hat{x}$, muss $\|r^{(j)}\|_2 \neq 0$. Also erhalten wir einen Widerspruch zur Annahme $\alpha_j \neq 0$. Somit ist $\left\langle d^{(m+1)}, d^{(j)} \right\rangle_A = 0$ für alle $0 \leq j < m+1$. Damit ist der Induktionsschluss vollständig bewiesen.

□

Gemäß Lemma 3.14, Behauptung 3 sind alle Suchrichtungen paarweise A-konjugiert. Gemäß Lemma 3.14, Behauptung 2 sind alle Residuen linear unabhängig. Daher ergibt sich nach spätestens $n = \dim(A)$ Schritten $r^{(n)} = 0$, also $\hat{x} = x^{(n)}$.

Korollar 3.15. *Nach spätestens $n = \dim(A)$ Schritten findet das CG-Verfahren die exakte Lösung $\hat{x}$.*

In der Praxis ist diese Ergebnis nicht von allzu großer Bedeutung, da das CG-Verfahren in erster Linie eingesetzt wird und wesentlich weniger als n-Iterationsschritte benötigt. Zudem sind die Orthogonalitätseigenschaften aus

Lemma 3.14 mit zunehmender Iterationsdauer aufgrund von Rundefehlern nicht mehr exakt erfüllt, so dass das Korollar in der Praxis nicht relevant ist.

Das CG-Verfahren hat Optimalitätseigenschaften, die wir im folgenden Satz zusammenfassen:

Satz 3.16. *Sei $k = 1, 2, \cdots$ fix. Darüberhinaus sei $d^{(0)} = r^{(0)}$ und $x^{(k)} \neq \hat{x}$ die k-te Iterierte des CG-Verfahrens. Dann liegt $x^{(k)}$ in dem affinen Raum*

$$x^{(k)} \in x^{(0)} + \left[r^{(0)}, \cdots, r^{(k-1)}\right] = x^{(0)} + \left[r^{(0)}, Ar^{(0)}, \cdots, A^{(k-1)}r^{(0)}\right] \ . \quad (3.26)$$

Unter allen Elementen x dieser Menge minimiert $x^{(k)}$ die Zielfunktion Φ. Der Raum

$$\mathcal{K}_k(A, r^{(0)}) = \left[r^{(0)}, Ar^{(0)}, \cdots, A^{(k-1)}r^{(0)}\right]$$

wird **Krylov-Raum** *der Dimension k von A bezüglich $r^{(0)}$ genannt.*

Beweis. Wir beweisen zunächst induktiv, dass

$$d^{(j)} \in \left[r^{(0)}, \cdots, r^{(k-1)}\right] \ , \quad j = 0, \cdots, k-1 \ . \quad (3.27)$$

Für $j = 0$ ist $d^{(0)} = r^{(0)}$ und der Induktionsschluss ergibt sich für $1 = j < k$ aus (3.21): Für $0 \leq j \leq k-2$ folgt aus $d^{(j)} = r^{(j)} + \beta_{j-1}d^{(j-1)}$

$$d^{(j)} \in \left[r^{(j)}, d^{(j-1)}\right] \subseteq \left[r^{(j)}, r^{(0)}, \cdots, r^{(k-1)}\right] = \left[r^{(0)}, \cdots, r^{(k-1)}\right] \ .$$

und somit gilt zusammenfassend

$$\left[d^{(0)}, \cdots, d^{(k-1)}\right] \subseteq \left[r^{(0)}, \cdots, r^{(k-1)}\right] \ .$$

Solange $x^{(k)} \neq \hat{x}$ ist, folgt aus Lemma 3.14, dass jede der beiden Mengen $\{d^{(j)}\}_{j=0}^{k-1}$ und $\{r^{(j)}\}_{j=0}^{k-1}$ aus linear unabhängigen Vektoren besteht. Demnach ist

$$\left[d^{(0)}, \cdots, d^{(k-1)}\right] = \left[r^{(0)}, \cdots, r^{(k-1)}\right] \ . \quad (3.28)$$

Nun beweisen wir (3.26): Aus (3.18) folgt

$$x^{(k)} = x^{(0)} + \sum_{j=0}^{k-1} \alpha_j d^{(j)} \in x^{(0)} + \left[r^{(0)}, \cdots, r^{(k-1)}\right] \ .$$

Damit haben wir die erste Inklusion in (3.26) gezeigt. Nun zeigen wir induktiv, dass

$$r^{(j)} \in \left[r^{(0)}, \cdots, A^{k-1}r^{(0)}\right] \ , \quad j = 0, \cdots, k-1 \ . \quad (3.29)$$

Für $j = 0$ ist diese Behauptung sicher richtig. Beim Induktionsschluss von $j-1$ nach j gilt zunächst wegen (3.27) (man beachte, dass $j-1 \leq k-2$)

$$d^{(j-1)} \in \left[r^{(0)}, \cdots, r^{(k-2)}\right] .$$

Somit folgt aus der Induktionsannnahme die Beziehung

$$d^{(j-1)} \in \left[r^{(0)}, \cdots, r^{(k-2)}\right] \subseteq \left[r^{(0)}, \cdots, A^{k-2}r^{(0)}\right] .$$

Wiederum, unter Verwendung von $j-1 \leq k-2$, folgt aus (3.29) und (3.23) die gewünschte Inklusion

$$r^{(j)} = r^{(j-1)} + \alpha_{j-1}Ad^{(j-1)} \in \left[r^{(0)}, \cdots, A^{k-1}r^{(0)}\right] .$$

Demnach ist

$$\left[r^{(0)}, \cdots, r^{(k-1)}\right] \subseteq \left[r^{(0)}, Ar^{(0)}, \cdots, A^{k-1}r^{(0)}\right] ,$$

Da $r^{(0)}, \cdots, r^{(k-1)}$ linear unabhängig sind, hat die rechte Seite maximale Dimension k. Also stimmen die beiden Mengen überein. Damit gilt (3.26).

Zum Beweis der Minimialeigenschaft: Aus Korollar 3.15 folgt schließlich die Existenz eines Iterationsindex $m \leq n$, so dass

$$\hat{x} = x^{(m)} = x^{(0)} + \sum_{j=0}^{m-1} \alpha_j d^{(j)}$$

(m muss nicht unbedingt mit n übereinstimmen). Folglich ist

$$\hat{x} - x^{(k)} = \sum_{j=k}^{m-1} \alpha_j d^{(j)} .$$

Für ein beliebiges Element x von $x^{(0)} + \left[r^{(0)}, Ar^{(0)}, \cdots, A^{(k-1)}r^{(0)}\right]$ gilt wegen (3.28)

$$\hat{x} - x = \hat{x} - x^{(k)} + \sum_{j=0}^{k-1} \delta_j d^{(j)} ,$$

mit geeigneten $\delta_j \in \mathbb{R}$.

Da die Suchrichtungen nach Lemma 3.14 A-konjugiert sind, folgt daher aus dem Satz von Pythagoras

$$\begin{aligned}\Phi(x) - \Phi(\hat{x}) &= \left(\frac{1}{2}x^*Ax - x^*b\right) - \left(\frac{1}{2}\hat{x}^*A\hat{x} - \hat{x}^*b\right) \\ &= \frac{1}{2}\|x - \hat{x}\|_A^2 \\ &= \frac{1}{2}\|x^{(k)} - \hat{x}\|_A^2 + \frac{1}{2}\left\|\sum_{j=0}^{k-1}\delta_j d^{(j)}\right\|_A^2 \\ &= \Phi(x^{(k)}) - \phi(\hat{x}) + \frac{1}{2}\left\|\sum_{j=0}^{k-1}\delta_j d^{(j)}\right\|_A^2 .\end{aligned}$$

Demnach ist $\Phi(x) \geq \Phi(x^{(k)})$ mit Gleichheit genau dann wenn $x = x^{(k)}$. Also is $x^{(k)}$ das minimierende Element auf dem Krylov-Raum. □

Für die Implementierung des CG-Verfahrens in der Praxis verwendet man nicht die Gleichungen (3.20) und (3.22) für α_k und β_k, sondern die folgenden Darstellungen (3.30) und (3.31).

Zur Herleitung dieser Formeln verwenden wir, dass aufgrund von Lemma 3.14 und (3.21) gilt:

$$r^{(k)*}d^{(k)} = r^{(k)*}r^{(k)} + \beta_{k-1}r^{(k)*}d^{(k-1)} = r^{(k)*}r^{(k)} .$$

In (3.20) eingesetzt, ergibt sich

$$\alpha_k = -\frac{\|r^{(k)}\|_2^2}{d^{(k)*}Ad^{(k)}} . \tag{3.30}$$

Aus (3.23), (3.30) und Lemma 3.14 folgt

$$\begin{aligned}r^{(k+1)*}Ad^{(k)} &\underbrace{=}_{(3.23)} \frac{1}{\alpha_k}(r^{(k+1)*}r^{(k+1)} - r^{(k+1)*}r^{(k)}) \\ &\underbrace{=}_{\text{Lemma3.14}} \frac{1}{\alpha_k}\|r^{(k+1)}\|_2^2 \\ &= -\frac{\|r^{(k+1)}\|_2^2}{\|r^{(k)}\|_2^2}d^{(k)*}Ad^{(k)} .\end{aligned}$$

Anstelle von (3.22) kann man daher die Formel

$$\beta_k = \frac{\|r^{(k+1)}\|_2^2}{\|r^{(k)}\|_2^2} \tag{3.31}$$

verwenden. Damit lautet das CG-Verfahren in der zu implementierenden Form wie folgt:

Algorithmus 3.17. (Verfahren der konjugierten Gradienten)

`Wähle` $x^{(0)}$ `beliebig; setze` $r^{(0)} = Ax^{(0)} - b,\ d^{(0)} = r^{(0)}$.

`for` $k = 0, 1, 2, \cdots$

$$\begin{aligned}
\alpha_k &= -\frac{\|r^{(k)}\|_2^2}{d^{(k)*} A d^{(k)}} \\
x^{(k+1)} &= x^{(k)} + \alpha_k d^{(k)} \\
r^{(k+1)} &= r^{(k)} + \alpha_k A d^{(k)} \\
\beta_k &= \frac{\|r^{(k+1)}\|_2^2}{\|r^{(k)}\|_2^2} \\
d^{(k+1)} &= r^{(k+1)} + \beta_k d^{(k)}
\end{aligned}$$

`until stop`

Bemerkung 3.18. Die gezeigten Resultate gelten ausnahmslos auch für komplexe (hermitesche) Matrizen, also $\mathbb{K} = \mathbb{C}$. In diesem Fall muss allerdings das Funktional

$$\Phi(x) = \frac{1}{2} x^* A x - \Re(x^* b) = \frac{1}{2}\|x - \hat{x}\|_A^2 - \frac{1}{2}\hat{x}^* A \hat{x} \tag{3.32}$$

für $x \in \mathbb{C}^n$ betrachtet werden.

3.3 Vorkonditionierung

Die Kondition $\mathrm{cond}_2(A)$ der Koeffizientenmatrix A des zu lösenden linearen Gleichungssystems ist ein entscheidender Parameter für die Konvergenz des CG-Verfahrens. Dabei ist die Konvergenz umso langsamer, je schlechter konditioniert die Matrix A ist. Man kann sogar zeigen, dass

$$q \approx 1 - 2\,\mathrm{cond}_2^{-1/2}(A)\ .$$

Beim Vorkonditionieren versucht man deshalb das Gleichungssystem $Ax = b$ durch ein äquivalentes System

$$M^{-1}Ax = M^{-1}b$$

zu ersetzen, dessen Kondition besser ist.

Kapitel 4

Eigenwerte

In diesem Kapitel beschäftigen wir uns mit der Berechnung der Eigenwerte einer Matrix $A \in \mathbb{K}^{n\times n}$. Die Eigenwertgleichung an und für sich,

$$Ax = \lambda x \quad x \in \mathbb{K}^n, \backslash\{0\},\ \lambda \in \mathbb{C}\,,$$

ist *nichtlinear*: wir wissen aus der linearen Algebra, dass die Nullstellen des charakteristischen Polynoms

$$p(\lambda) = \det\,(A - \lambda I)$$

die Eigenwerte von A sind. Aus dieser Beziehung lässt sich sofort erkennen, dass die Eigenwertbestimmung ein nichtlineares Problem ist.

Zur Eigenwertberechnung kann man also nichtlineare Gleichungssystemlöser, wie etwa das *Newtonsche Verfahren* verwenden. Allerdings gibt es für die Eigenwertberechnung bessere Algorithmen (im Sinne von Stabilität und Konvergenzeigenschaften), die die spezielle Struktur eines Eigenwertproblems berücksichtigen. Solche Algorithmen werden wir im Folgenden behandeln.

4.1 Eigenwerteinschließung

Im diesem Teil diskutieren wir relativ leicht zu berechnende Abschätzungen für Eigenwerte.

Satz 4.1. (Satz von Gerschgorin) *Sei $A = [a_{ij}] \in \mathbb{K}^{n\times n}$ und λ ein beliebiger Eigenwert von A. Dann gilt*

$$\lambda \in \bigcup_{i=1}^{n} \mathcal{K}_i = \bigcup_{i=1}^{n} \left\{ \zeta \in \mathbb{K} : |\zeta - a_{ii}| \leq \sum_{j\neq i} |a_{ij}| \right\}\,. \tag{4.1}$$

Beweis. Sei $Ax = \lambda x$ mit $x = [x_i] \neq 0$. Dann existiert ein i mit $|x_j| \leq |x_i|$ für alle $j \neq i$. Bezeichnet $(Ax)_i$ die i-te Komponente von Ax, dann folgt

$$\lambda x_i = (Ax)_i = \sum_{j=1}^{n} a_{ij} x_j$$

und somit ist

$$|\lambda - a_{ii}| = \left| \sum_{j \neq i} a_{ij} \frac{x_j}{x_i} \right| \leq \sum_{j \neq i} |a_{ij}| \; .$$

Also ist $\lambda \in \mathcal{K}_i \subseteq \bigcup_{j=1}^{n} \mathcal{K}_j$. □

Für $\overline{\lambda} \in \sigma(A^*)$ gilt der Satz von Gerschgorin entsprechend, nämlich

$$\overline{\lambda} \in \bigcup_{i=1}^{n} \mathcal{K}_i = \bigcup_{i=1}^{n} \left\{ \zeta : |\zeta - \overline{a}_{ii}| \leq \sum_{j=1, j \neq i}^{n} |a_{ji}| \right\}$$

oder äquivalent

$$\lambda \in \bigcup_{i=1}^{n} \overline{\mathcal{K}}_i = \bigcup_{i=1}^{n} \left\{ \zeta : |\zeta - a_{ii}| \leq \sum_{j=1, j \neq i}^{n} |a_{ji}| \right\} \tag{4.2}$$

Weitere Einschließungssätze beruhen auf dem Konzept des Wertebereichs einer Matrix.

Definition 4.2. Unter dem **Wertebereich** einer Matrix $A \in \mathbb{K}^{n \times n}$ versteht man die Menge aller **Rayleigh-Quotienten** $\frac{x^* A x}{x^* x}$ mit $x \in \mathbb{C}^n \backslash \{0\}$,

$$\mathcal{W}(A) := \left\{ \zeta = \frac{x^* A x}{x^* x} : x \in \mathbb{C}^n \backslash \{0\} \right\} \subseteq \mathbb{C} \; .$$

In dieser Definition ist es wesentlich, dass x alle *komplexen* Vektoren durchläuft, selbst dann, wenn A eine reelle Matrix ist. Der Wertebereich beinhaltet insbesondere die Eigenwerte der Matrix. Weitere wichtige Eigenschaften sind im folgenden Lemma zusammengefasst.

Lemma 4.3. *1. $\mathcal{W}(A)$ ist zusammenhängend.*

2. Ist A hermitesch, dann ist $\mathcal{W}(A)$ das reelle Intervall $[\lambda_{\min}, \lambda_{\max}]$.

3. *Ist A* **schiefsymmetrisch**, *d.h., $A^* = -A$, dann ist $\mathcal{W}(A)$ ein rein imaginäres Intervall, nämlich die konvexe Hülle aller Eigenwerte von A.*

Beweis. 1. Liegen ζ_0 und ζ_1 im Wertebereich $\mathcal{W}(A)$, dann existieren $x_0, x_1 \in \mathbb{C}\backslash\{0\}$ mit

$$\zeta_0 = \frac{x_0^* A x_0}{x_0^* x_0}\,, \quad \zeta_1 = \frac{x_1^* A x_1}{x_1^* x_1}\,.$$

Nehmen wir an, dass $0 \neq \zeta_0 \neq \zeta_1 \neq 0$, dann sind x_0 und x_1 linear unabhängig (die zwei Vektoren x_0 und x_1 können nur dann linear abhängig sein, wenn x_0 ein Vielfaches von x_1 ist, was aber impliziert, dass $\zeta_0 = \zeta_1$ ist) und die Verbindungsgerade

$$[x_0, x_1] := \{x_t = x_0 + t(x_1 - x_0) : t \in [0,1]\}$$

enthält folglich nicht den Nullpunkt. Damit ist die Stetigkeit der Abbildung $t \to \frac{x_t^* A x_t}{x_t^* x_t}$ gewährleistet, so dass

$$\zeta_t := \frac{x_t^* A x_t}{x_t^* x_t}\,, \quad 0 \leq t \leq 1\,,$$

eine stetige Kurve in $\mathcal{W}(A)$ bildet, die ζ_0 mit ζ_1 verbindet.

2. Wir wählen eine Orthonormalbasis $\{x_i\}_{i=1}^n$ in $\mathbb{C}^n$ mit $Ax_i = \lambda x_i$, wobei die Eigenwerte von A absteigend sortiert sein sollen, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$. Für einen beliebigen Vektor $x = \sum_{i=1}^n \zeta_i x_i \in \mathbb{C}^n$, $\zeta_i \in \mathbb{C}$, gilt dann

$$x^* A x = \sum_{i,j=1}^n \overline{\zeta}_i \zeta_j x_i^* A x_j = \sum_{i,j=1}^n \overline{\zeta}_i \zeta_j \lambda_j x_i^* x_j = \sum_{i=1}^n \lambda_i |\zeta_i|^2\,,$$

und wegen der Anordnung der Eigenwerte folgt

$$\lambda_n \|x\|_2^2 = \lambda_n \sum_{i=1}^n |\zeta_i|^2 \leq \sum_{i=1}^n \lambda_i |\zeta_i|^2 \leq \lambda_1 \sum_{i=1}^n |\zeta_i|^2 = \lambda_1 \|x\|_2^2\,.$$

Damit ist zunächst gezeigt, dass $\mathcal{W}(A) \subseteq [\lambda_n, \lambda_1]$ gilt. Da $\mathcal{W}(A)$ nach dem ersten Teil dieses Satzes auch zusammenhängend ist, folgt die zweite Behauptung.

3. Wegen $A^* = -A$ ist iA hermitesch, denn

$$(iA)^* = \bar{i}A^* = -iA^* = iA \; .$$

Ferner ist

$$\mathcal{W}(A) = \mathcal{W}(-iiA) = -i\mathcal{W}(iA) = -i\left[\lambda_{\min}, \lambda_{\max}\right] \; .$$

Daher folgt die Behauptung aus dem zweiten Teil des Satzes.

□

Für jede beliebige Matrix $A \in \mathbb{K}^{n\times n}$ ist

$$A = \frac{A + A^*}{2} + \frac{A - A^*}{2}$$

eine Zerlegung in die hermitesche Matrix $\frac{A+A^*}{2}$ und die schiefsymmetrische Matrix $\frac{A-A^*}{2}$. Diese Zerlegung ist die Grundlage des folgenden Einschließungssatzes.

Satz 4.4. (Satz von Bendixon) *Das Spektrum von $A \in \mathbb{K}^{n\times n}$ ist in dem Rechteck*

$$\sigma(A) \subseteq \mathcal{R} := \mathcal{W}\left(\frac{A + A^*}{2}\right) + \mathcal{W}\left(\frac{A - A^*}{2}\right) \tag{4.3}$$

enthalten.

Beweis. Wir zeigen die stärkere Aussage $\mathcal{W}(A) \subseteq \mathcal{R}$. Für $x \in \mathbb{C}^n\backslash\{0\}$ gilt

$$\begin{aligned}
\frac{x^*Ax}{x^*x} &= \frac{x^*\left(\frac{A+A^*}{2} + \frac{A-A^*}{2}\right)x}{x^*x} \\
&= \frac{x^*\frac{A+A^*}{2}x}{x^*x} + \frac{x^*\frac{A-A^*}{2}x}{x^*x} \\
&\in \mathcal{W}\left(\frac{A + A^*}{2}\right) + \mathcal{W}\left(\frac{A - A^*}{2}\right) \; .
\end{aligned}$$

□

Beispiel 4.5. Wir wenden die Resultate aus den Sätzen von Gerschgorin und Bendixson auf die Matrix

$$A = \begin{bmatrix} 4 & 0 & -3 \\ 0 & -1 & 1 \\ -1 & 1 & 0 \end{bmatrix}$$

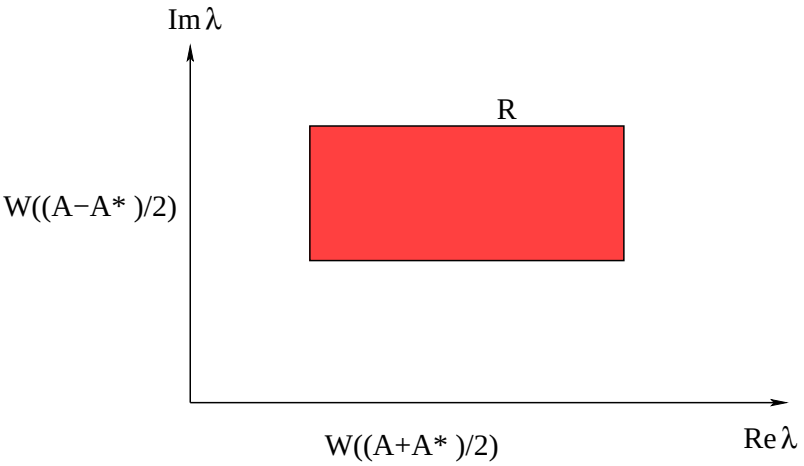

Abbildung 4.1: Satz von Bendixon

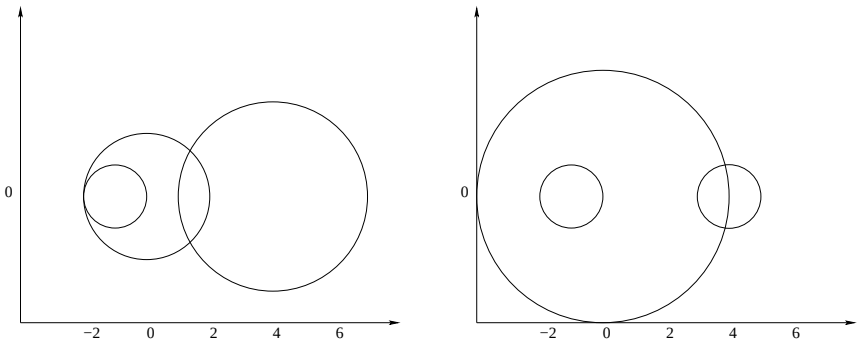

Abbildung 4.2: Gerschgorinkreise für A (links) und A^* rechts

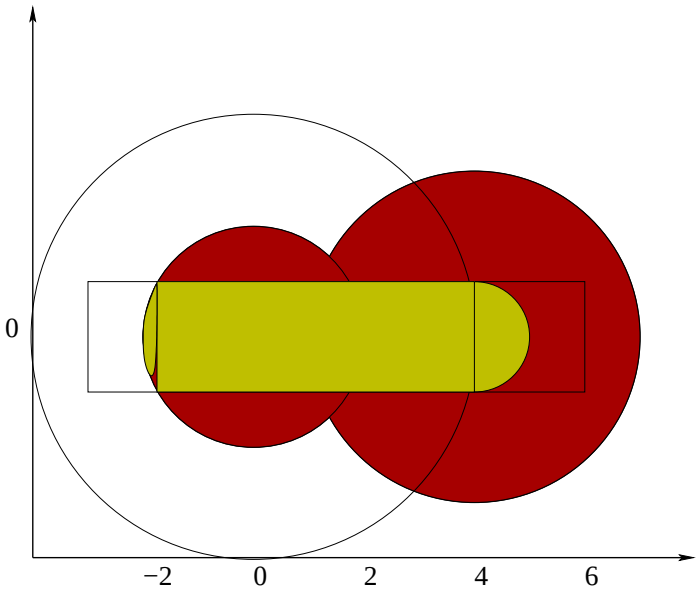

Abbildung 4.3: Alle Einschließungen gemeinsam

an. Abbildung 4.2 zeigt die Gerschgorinkreise für A und A^*. Für den Satz von Bendixon berechnen wir ferner den symmetrischen und den schiefsymmetrsichen Anteil von A,

$$H = \frac{A + A^t}{2} = \begin{bmatrix} 4 & 0 & 2 \\ 0 & -1 & 1 \\ 2 & 1 & 0 \end{bmatrix}, \quad S = \frac{A - A^t}{2} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}.$$

Die Spektren von H und S könnten beispielsweise mit Hilfe des Satzes von Gerschgorin eingeschlossen werden: Auf diese Weise erhält man das etwas gröbere Dreieck

$$\tilde{\mathcal{R}} = [-3, 6] \times [-i, i] \supset \mathcal{R} \supset \sigma(A) .$$

Folglich muss das Spektrum von A im Schnitt *aller* drei Einschlussmengen liegen. Dies ergibt die dunkelste (gelbe) Menge in Abbildung 4.3. Tatsächlich ist das Spektrum

$$\sigma(A) = \{-1.7878, 0.1198, 4.6679\} .$$

Für hermitesche Matrizen ist noch folgendes Resultat von großer Bedeutung:

Satz 4.6. (Maxmin Prinzip von Courant-Fischer) *Sei $A = A^* \in \mathbb{K}^{n\times n}$ und $\{z_1, \cdots, z_n\} \subseteq \mathbb{K}^n$ ein orthonormales System. Ferner seien $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_n$ die absteigend sortierten Eigenwerte von A mit zugehörigen Eigenvektoren x_i. Dann ist*

$$\min_{x\in[z_1,\cdots,z_k],x\neq 0} \frac{x^*Ax}{x^*x} \leq \lambda_k \tag{4.4}$$

und dabei gilt Gleichheit für $[z_1, \cdots, z_k] = [x_1, \cdots, x_k]$.

Beweis. Wir konstruieren zunächst einen nichttrivialen Vektor $x = \sum_{i=1}^k \zeta_i z_i \in [z_1, \cdots, z_k]$, der senkrecht auf die Eigenvektoren $x_1, \cdots, x_{k-1}$ von A steht. Dazu müssen die Koeffizienten $\zeta_1, \cdots, \zeta_k$ das homogene lineare Gleichungssystem

$$x^*x_j = \sum_{i=1}^{k} (z_i^*x_j)\overline{\zeta}_j = 0 \quad j = 1, \cdots, k-1\,, \tag{4.5}$$

lösen. Da dieses Gleichungssystem unterbestimmt ist ($k-1$ Gleichungen für k Koeffizienten), hat es eine nichttriviale Lösung $[\zeta_1, \cdots, \zeta_k] \neq 0$. Wird der zugehörige Vektor $x \neq 0$ in die Eigenbasis von A umentwickelt, $x = \sum_{i=1}^n \chi_i x_i$, dann folgt aus (4.5), dass

$$\begin{aligned}
x^*Ax &= x^* \sum_{i=1}^{n} \chi_i \lambda_i x_i \\
&= \sum_{i=1}^{n} \lambda_i \chi_i x^* x_i \\
&= (4.5) \sum_{i=k}^{n} \lambda_i \chi_i \sum_{j=1}^{n} \overline{\chi}_j x_j^* x_i \\
&= \sum_{i=k}^{n} \lambda_i |\chi_i|^2 \\
&\leq \lambda_k \sum_{i=k}^{n} |\chi_i|^2 \\
&\leq \lambda_k \|x\|_2^2\,.
\end{aligned}$$

Hieraus folgt die Ungleichung

$$\frac{x^*Ax}{x^*x} \leq \lambda_k$$

und die Behauptung (4.4) ist bewiesen.

Falls $[z_1, \cdots, z_k] = [x_1, \cdots, x_k]$ ist, dann kann jedes $x \in [z_1, \cdots, z_k]$ geschrieben werden als $x = \sum_{i=1}^{k} \chi_i x_i$ und es folgt

$$x^* A x = \sum_{i=1}^{k} \lambda_i \chi_i \sum_{j=1}^{k} \overline{\chi}_j x_j^* x_i = \sum_{i=1}^{k} \lambda_i |\chi_i|^2 \geq \lambda_k \sum_{i=1}^{k} |\chi_i|^2 = \lambda_k \|x\|_2^2 \, .$$

Somit ist

$$\min_{x \in [z_1, \cdots, z_k], x \neq 0} \frac{x^* A x}{x^* x} \geq \lambda_k \, ,$$

und demnach gilt in diesem Fall Gleichheit in (4.4). □

Unter den Voraussetzungen von Satz 4.6 lautet ein entsprechendes Maximumprinzip wie folgt

$$\max_{x \perp [z_1, \cdots, z_k], x \neq 0} \frac{x^* A x}{x^* x} \geq \lambda_{k+1} \, ,$$

mit Gleichheit für $[z_1, \cdots, z_k] = [x_1, \cdots, x_k]$.

4.2 Kondition des Eigenwertproblems

Wir betrachten die Matrix

$$A = \begin{bmatrix} 0 & \cdots & \cdots & 0 & -a_0 \\ 1 & 0 & & \ddots & -a_1 \\ & 1 & \ddots & \ddots & \ddots \\ & & \ddots & 0 & \ddots \\ 0 & & & 1 & -a_{n-1} \end{bmatrix} \tag{4.6}$$

und entwickeln die Determinante von $A - \lambda I$ nach der letzten Spalte. Dann ergibt sich das charakteristische Polynom von A aus

$$(-1)^n \det\,(A - \lambda I) = \lambda^n + a_{n-1}\lambda^{n-1} + \cdots + a_1 \lambda + a_0 =: p(\lambda) \, .$$

Ist umgekehrt p ein beliebiges vorgegebenes Polynom vom Grad n mit Koeffizienten $a_0, \cdots, a_{n-1}$, dann nennt man die Matrix A aus (4.6) die **Frobenius-Begleitmatrix** von p.

Das spezielle Polynom $p_0(\lambda) = (\lambda - a)^n$ hat eine n-fache Nullstelle $\hat{\lambda} = a$, während $p_\varepsilon(\lambda) = (\lambda - a)^n - \varepsilon$ mit $\varepsilon > 0$ die folgenden Nullstellen λ_k besitzt,

$$\lambda_k = a + \varepsilon^{1/n} e^{i2\pi k/n}, \quad k = 1, \cdots, n .$$

Das "gestörte" Polynom p_ε unterscheidet sich von p lediglich in den Koeffizienten vor λ^0 (und zwar gerade um ε). Obwohl sich also die entsprechenden Frobenius-Begleitmatrizen sowohl in der Zeilensummen-, Spaltensummen-, Spektral- und Frobeniusnorm nur um ε unterscheiden, haben die Eigenwerte der beiden Matrizen den Abstand

$$|\Delta\lambda| = |\lambda_k - \hat{\lambda}| = \varepsilon^{1/n}, \quad k = 1, \cdots, n .$$

Für $a \neq 0$ ist daher

$$\frac{\Delta\lambda}{\hat{\lambda}} = \frac{\varepsilon^{1/n}}{|a|} = c_\varepsilon \frac{\|\Delta A\|}{\|A\|} \text{ mit } c_\varepsilon = \frac{\|A\|}{|a|} \frac{\varepsilon^{1/n}}{\varepsilon} . \tag{4.7}$$

Da c_ε für $\varepsilon \to 0$ beliebig groß wird, ist die relative Konditionszahl des allgemeinen Eigenwertproblems im Allgemeinen unendlich groß. Man kann aber auch zeigen, dass die Eigenwerte stetig von den Einträgen der Matrix abhängen und der gefundene Exponent $1/n$ in (4.7) schlimmstmöglich ist.

Definition 4.7. Eine Matrix $A \in \mathbb{K}^{n\times n}$ heißt **diagonalisierbar**, falls eine Basis $\{x_i\}_{i=1}^n$ des $\mathbb{K}^n$ aus Eigenvektoren existiert. Mit $X = [x_1, \cdots, x_n] \in \mathbb{K}^{n\times n}$ gilt dann

$$A = XDX^{-1}, \tag{4.8}$$

wobei in der Diagonalmatrix D die Eigenwerte $\lambda_1, \cdots, \lambda_n$ auf der Diagonale stehen. Kann X zudem unitär gewählt werden, dann heißt A **normal**. Normale Matrizen lassen sich durch die Gleichung $AA^* = A^*A$ charakterisieren.

Hermitesche Matrizen sind also ein Spezialfall der normalen Matrizen.

In der Regel ist eine Matrix allerdings *nicht* diagonalisierbar. Ein typisches Beispiel ist die Matrix

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix},$$

deren charakteristisches Polynom $p(\lambda) = \lambda^2$ eine doppelte Nullstelle in $\lambda = 0$ hat; A hat jedoch nur den Eigenvektor $[1, 0]^t$.

Im folgenden bringen wir ohne Beweis einige Abschätzungen für die Fehler in den Eigenwerten bei Störungen in den Matrixeinträgen.

Satz 4.8. (Satz von Bauer und Fike) *$A \in \mathbb{K}^{n\times n}$ sei diagonalisierbar mit entsprechender Faktorisierung $A = XDX^{-1}$ wie in (4.8); ferner sei $F \in \mathbb{K}^{n\times n}$ und λ ein Eigenwert von $A + F$. Dann existiert ein Eigenwert $\hat{\lambda}$ von A mit*

$$|\lambda - \hat{\lambda}| \leq \mathrm{cond}(X)\|F\| \; .$$

Hierbei bezeichnet $\|\cdot\|$ wahlweise die Zeilensummen-, Spaltensummen- oder Spektralnorm und $\mathrm{cond}(X)$ die entsprechende Konditionszahl von X.

Korollar 4.9. *Ist $A \in \mathbb{K}^{n\times n}$ normal, $F \in \mathbb{K}^{n\times n}$ beliebig und λ ein Eigenwert von $A + F \in \mathbb{K}^{n\times n}$, dann existiert $\hat{\lambda} \in \sigma(A)$ mit*

$$|\lambda - \hat{\lambda}| \leq \|F\|_2 \; .$$

Beweis. Im betrachteten Fall ist die Eigenvektormatrix X eine unitäre Matrix. Folglich ist $\|X\|_2 = 1$ und $\|X^{-1}\|_2 = \|X^*\|_2 = 1$. Damit ist $\mathrm{cond}_2(X) = 1$ und somit folgt die Behauptung aus Satz 4.8. □

Satz 4.10. (Einschließungsatz basierend auf Bauer und Fike) *Sei $A \in \mathbb{K}^{n\times n}$ eine hermitsche Matrix und $x \in \mathbb{K}^n$ ein Vektor mit*

$$\|Ax - \lambda x\|_2 = \varepsilon\|x\|_2 \; .$$

Dann besitzt A einen Eigenwert $\hat{\lambda}$ mit

$$\|\hat{\lambda} - \lambda\| \leq \varepsilon \; .$$

Ist ferner $\hat{x}$ ein Eigenvektor zum Eigenwert $\hat{\lambda}$ und γ der kleinste Abstand von λ zu einem anderen Eigenwert von A, dann gilt für den eingeschlossenen Winkel θ zwischen x und $\hat{x}$ die Abschätzung

$$|\sin(\theta)| \leq \frac{\varepsilon}{\gamma} \; .$$

Satz 4.11. (Satz von Wielandt und Hoffman) **A und F seien hermitesch, $\hat{\lambda}_1 \geq \hat{\lambda}_2 \cdots \geq \hat{\lambda}_n$ und $\lambda_1 \geq \lambda_2 \cdots \geq \lambda_n$ seien die Eigenwerte von A, bzw. $A + F$. Dann ist**

$$\sum_{i=1}^{n} (\lambda_i - \hat{\lambda}_i)^2 \leq \|F\|_F^2 \; .$$

In den folgenden Unterkapiteln beschäftigen wir uns mit numerischen Methoden zur Berechnung von Eigenwerten und Eigenvektoren.

4.3 Potenzmethode

Das erste von uns betrachtete konstruktive Verfahren zur Berechnung einzelner Eigenwerte und Eigenvektoren ist die **Potenzmethode von v. Mises**.

Wir beschränken uns vorderhand auf reelle diagonalisierbare $n \times n$ Matrizen mit n betragsmäßig verschiedenen (und daher reellen) Eigenwerten λ_i, $i = 1, \cdots, n$, die nach der Größe ihres Betrages angeordnet sind:

$$|\lambda_1| > \cdots > |\lambda_n| \geq 0 \, .$$

Alle Ergebnisse können mit entsprechendem technischen Aufwand auf den allgemeinen Fall einer beliebigen Matrix $A \in \mathbb{K}^{n \times n}$ übertragen werden.

Ist $\| \cdot \|$ eine Vektornorm und sind $v_i, i = 1, \cdots, n$, mit $\|v_i\| = 1$ die zu λ_i gehörigen Eigenvektoren von A, dann kann jeder Vektor $x \in \mathbb{K}^n$ in diese Eigenvektorbasis entwickelt werden,

$$x = \sum_{i=1}^{n} \zeta_i v_i \, . \tag{4.9}$$

Demnach ist

$$A^k x = \sum_{i=1}^{n} \lambda_i^k \zeta_i v_i \, . \tag{4.10}$$

Das v. Mises-Verfahren beruht auf der asymptotischen Darstellung

$$A^k x \approx \lambda_1^k \zeta_1 v_1 \, , \quad k \to \infty \, .$$

Algorithmus 4.12. `Bestimme einen Näherungsvektor` $z^{(0)}$ `mit` $\|z^{(0)}\| = 1$

- `for` $k = 1, 2, \cdots$

$$\tilde{z}^{(k)} := A z^{(k-1)} \, , \quad z^{(k)} := \frac{\tilde{z}^{(k)}}{\|\tilde{z}^{(k)}\|} \, , \tag{4.11}$$

  `end for`

Die Iterierten der Potenzmethode haben folgende Eigenschaften:

Satz 4.13. *es gelte (4.9) mit* $\zeta_1 \neq 0$ *und* $q := \left|\frac{\lambda_2}{\lambda_1}\right| < 1$*, dann ist*

$$\|\tilde{z}^{(k)}\| = |\lambda_1| + O(q^k)\,, \quad k \to \infty\,. \tag{4.12}$$

Ferner gilt für $k \to \infty$*:*

$$\begin{aligned} \|z^{(k)} - \operatorname{sign}(\zeta_1)v_1\| &= O(q^k) \quad \text{für } \lambda_1 > 0\,, \\ \|(-1)^k z^{(k)} - \operatorname{sign}(\zeta_1)v_1\| &= O(q^k) \quad \text{für } \lambda_1 < 0\,. \end{aligned}$$

Beweis. Im Beweis verwenden wir die Identität

$$z^{(k)} = \frac{\tilde{z}^{(k)}}{\|\tilde{z}^{(k)}\|} = \frac{Az^{(k-1)}}{\|Az^{(k-1)}\|} = \frac{A\tilde{z}^{(k-1)}}{\|A\tilde{z}^{(k-1)}\|} = \frac{A^2 z^{(k-2)}}{\|A^2 z^{(k-2)}\|}\,.$$

Mit Induktion folgt somit

$$z^{(k)} = \frac{A^k z^{(0)}}{\|A^k z^{(0)}\|}\,.$$

Aus (4.10) folgt mit $x := z^{(k)}$

$$A^k x = \lambda_1^k \zeta_1 (v_1 + w^{(k)}) \text{ mit } w^{(k)} = \sum_{i=2}^{n} \left(\frac{\lambda_i}{\lambda_1}\right)^k \frac{\zeta_i}{\zeta_1} v_i$$

und

$$\|w^{(k)}\| \leq q^k \sum_{i=2}^{n} \left|\frac{\zeta_i}{\zeta_1}\right| = O(q^k)\,. \tag{4.13}$$

Damit ist

$$z^{(k)} = \frac{A^k x}{\|A^k x\|} = \operatorname{sign}(\lambda_1^k \zeta_1) \frac{v_1 + w^{(k)}}{\|v_1 + w^{(k)}\|} = \operatorname{sign}(\lambda_1^k \zeta_1) v_1 + e^{(k)} \tag{4.14}$$

mit

$$e^{(k)} = \frac{\operatorname{sign}(\lambda_1^k \zeta_1)}{\|v_1 + w^{(k)}\|} \left(w^{(k)} + (1 - \|v_1 + w^{(k)}\|)v_1\right)\,.$$

Nun ist $\|v_1\| - \|w^{(k)}\| \leq \|v_1 + w^{(k)}\| \leq \|v_1\| + \|w^{(k)}\|$ und $\|v_1\| = 1$, so dass aus (4.13) folgt, dass

$$\left|1 - \|v_1 + w^{(k)}\|\right| = \left|\|v_1\| - \|v_1 + w^{(k)}\|\right| \leq \|w^{(k)}\| = O(q^{(k)}) \quad k \to \infty\,.$$

Daraus folgt $\|e^{(k)}\| = O(q^{(k)})$ für $k \to \infty$, und es ist gezeigt, dass für $z^{(k)}$ und $\tilde{z}^{(k+1)}$ für $k \to \infty$ gilt

$$z^{(k)} = \operatorname{sign}(\lambda_1^k \zeta_1) v_1 + O(q^k) ,$$
$$\tilde{z}^{(k+1)} = \lambda_1 \operatorname{sign}(\lambda_1^k \zeta_1) v_1 + O(q^k) .$$

Daraus folgt unmittelbar die Behauptung. □

Bemerkung 4.14.

- Aus (4.12) schließt man bei der Iteration zunächst auf $|\lambda_1|$. Anhand des Vorzeichenverhaltens von $z^{(k)}$ erkennt man das Vorzeichen von λ_1: Alternieren die Vorzeichen von $z^{(k)}$, dann ist $\lambda_1 < 0$; ansonst ist $\lambda_1 > 0$.

- Die Normierung $\tilde{z}^{(k)} \to z^{(k)}$ in (4.11) ist sinnvoll, wenn auch nicht notwendig.

- Die Voraussetzung $\zeta_1 \neq 0$ kann natürlich nicht a–priori überprüft werden, da die Eigenvektoren nicht bekannt sind. In der Praxis erweist sich diese Restriktion nicht als einschränkend.

Die Potenzmethode von v. Mises kann in dieser Form nur verwendet werden, um λ_1 zu bestimmen. Zur Berechnung anderer Eigenwerte von A kann jedoch A geeignet transformiert werden.

Beispiel 4.15.

1. Ist $\lambda_n \neq 0$ und verwendet man A^{-1} statt A in (4.11), dann wird dies **inverse Iteration** genannt. A^{-1} hat die Eigenwerte λ_i^{-1} mit
$$|\lambda_n^{-1}| > |\lambda_{n-1}^{-1}| > \cdots > |\lambda_1^{-1}|$$
mit den gleichen Eigenvektoren wie A. Also approximiert die inverse Iteration $|\lambda_n^{-1}|$ und den zugehörigen Eigenvektor v_n.

2. Ist λ eine Näherung an einen Eigenwert von A, liegt aber selbst nicht im Spektrum $\sigma(A)$, dann ergibt (4.11) mit $(A - \lambda I)^{-1}$ anstellen von A die **gebrochene Iteration von Wielandt** . $(A - \lambda I)^{-1}$ hat die Eigenwerte $(\lambda_i - \lambda)^{-1}, i = 1, \cdots n$.

Kapitel 5

Nichtlineare Gleichungen

Nach den linearen Gleichungen untersuchen wir nun nichtlineare Gleichungen in einer oder mehreren reellen Variablen. Nichtlineare Gleichungen werden zumeist als Nullstellenaufgaben formuliert, d.h., gesucht ist die Nullstelle der Abbildung

$$f : \mathcal{D}(f) \subseteq \mathbb{R}^n \to \mathbb{R}^n \ .$$

Durch die Transformation $f(x) := g(x) - y$ kann jede nichtlineare Gleichung $g(x) = y$ unmittelbar in eine solche Nullstellenaufgabe übergeführt werden.

Da nichtlineare Gleichungen in der Regel nicht mehr geschlossen gelöst werden können, kommen nur iterative Algorithmen der Lösung in Frage. Häufig besteht dabei ein Iterationsschritt in der Lösung eines linearen Teilproblems.

5.1 Konvergenzordnung

Wir unterscheiden verschiedene Konvergenzbegriffe bei iterativen Lösungsverfahren.

Definition 5.1. Sei $\{\varepsilon_k\}_{k\in\mathbb{N}_0}$ eine positive reelle Nullfolge. Man sagt, dass die Konvergenz dieser Folge (mindestens) die Ordnung $p \geq 1$ hat, wenn ein $C > 0$ und ein $k_0 \in \mathbb{N}$ existiert, so dass

$$\varepsilon_{k+1} \leq C\varepsilon_k^p \text{ für } k \geq k_0 \ . \tag{5.1}$$

Für $p = 1$ wird vorausgesetzt, dass $C < 1$ ist.

Entsprechend wird die Konvergenzordnung einer konvergenten Folge $\{x^{(k)}\} \subseteq \mathbb{R}^n$ mit Grenzwert $\hat{x}$ über die Konvergenzordnung der Fehlerfolge

$$\varepsilon_k = \|x^{(k)} - \hat{x}\|$$

definiert. Da alle Normen in $\mathbb{R}^n$ äquivalent sind, spielt es bei der Definition der Konvergenzordnung keine Rolle, welche Norm verwendet wird.

Beispiel 5.2.

1. Der Fall $p = 1$ ist von besonderer Bedeutung und uns bereits im Zusammenhang mit dem Banachschen Fixpunktsatz (3.1) begegnet. Nach Satz 3.5 hat die Fixpunktiteration

$$x^{(k+1)} = Tx^{(k)} + c\,, \quad T \in \mathbb{R}^{n\times n},\ c, x^{(0)} \in \mathbb{R}^n\,, \tag{5.2}$$

   die Konvergenzordnung $p = 1$ (und nicht mehr), falls $0 < \rho(T) < 1$.

2. Der Fall $p = 2$ (quadratische Konvergenz) wird uns im Zusammenhang mit dem **Newton-Verfahren** begegnen.

3. p braucht nicht unbedingt eine ganze Zahl sein. Ein entsprechendes Beispiel ist das **Sekantenverfahren**.

4. Die Folge $\varepsilon_k := k^{-\nu}$, $\nu \in \mathbb{R}^+$, konvergiert gegen 0, erfüllt aber keine Ungleichung der Form (5.1). Man spricht in diesem Fall von **sublinearer Konvergenz** (langsamer als linear). Entsprechend heißt eine Nullfolge $\{\varepsilon_k\}$ **superlinear konvergent**, falls

$$\frac{\varepsilon_{k+1}}{\varepsilon_k} \to 0\,, \quad k \to \infty\,.$$

Bei nichtlinearen Verfahren ist es wichtig zwischen *lokaler* und *globaler* Konvergenz zu unterscheiden:

Definition 5.3. Ein Iterationsverfahren $x^{(k+1)} = \Phi(x^{(k)})$ mit einer Funktion $\Phi : \mathcal{D}(\Phi) \subseteq \mathbb{R}^n \to \mathbb{R}^n$ heißt **lokal konvergent** gegen $x \in \mathbb{R}^n$, falls eine Umgebung $\mathcal{U} \subseteq \mathbb{R}^n$ um $x \in \mathcal{U}$ existiert, so dass für Startvektoren $x^{(0)} \in \mathcal{U}$ die resultierende Folge $\{x^{(k)}\}$ gegen x konvergiert. Man spricht von **globaler Konvergenz**, falls $\mathcal{U} = \mathbb{R}^n$ ist.

Satz 5.4. *Sei $\emptyset \neq \mathcal{D}(\Phi)$ offen. Die Funktion $\Phi : \mathcal{D}(\Phi) \subseteq \mathbb{R}^n \to \mathbb{R}^n$ sei stetig differenzierbar mit Fixpunkt $\hat{x}$. Ferner sei $\|\cdot\|$ eine Norm in $\mathbb{R}^n$ und $|||\cdot|||$ eine verträgliche Matrixnorm mit $|||\nabla\Phi(x)||| < 1$ für alle $x \in \mathcal{D}(\Phi)$. Dann konvergiert das Iterationsverfahren*

$$x^{(k+1)} = \Phi(x^{(k)})\,, \quad k = 0, 1, 2, \cdots$$

(mindestens) lokal linear gegen $\hat{x}$.

Beweis. Wegen der Stetigkeit von $\nabla\Phi$ existiert eine abgeschlossene Kugel $\mathcal{U} \subseteq \mathcal{D}(\Phi)$ um $\hat{x}$ mit Radius $\rho > 0$ mit

$$|||\nabla\Phi(x)||| \leq q < 1 \text{ für alle } x \in \mathcal{U} .$$

Aus dem Mittelwertsatz in $\mathbb{R}^n$ folgt

$$\Phi(y) - \Phi(x) \leq \int_0^1 \nabla\Phi(x + t(y-x))(y-x)\, dt ,$$

folgt

$$\|\Phi(y) - \Phi(x)\| = \int_0^1 |||\nabla\Phi(x + t(y-x))|||\, \|y-x\|\, dt \leq q\|y-x\| .$$

Das bedeutet, dass Φ eine Kontraktion auf $\mathcal{U}$ ist. Für $y = \hat{x}$ sieht man ferner, dass Φ eine Selbstabbildung der Menge $\mathcal{U}$ ist, denn für $x \in \mathcal{U}$ ist

$$\|\Phi(x) - \hat{x}\| = \|\Phi(x) - \Phi(\hat{x})\| \leq q\|x - \hat{x}\| \leq q\rho < \rho ,$$

also ist auch $\Phi(x) \in \mathcal{U}$. Damit folgt die Behauptung des Satzes aus dem Banachschen Fixpunktsatz. □

Die Existenz des Fixpunktes braucht man nur für die Selbstabbildungseigenschaften.

Satz 5.5. *Sei $p = 2, 3, \ldots$. Die Funktion $\Phi : [a, b] \to \mathbb{R}$ sei $(p+1)$-mal stetig differenzierbar in (a, b), und es sei $\Phi(\hat{x}) = \hat{x}$ für ein $\hat{x} \in (a, b)$. Gilt*

$$0 = \Phi'(\hat{x}) = \cdots = \Phi^{(p-1)}(\hat{x}) \text{ und } \Phi^{(p)}(\hat{x}) \neq 0 , \tag{5.3}$$

dann konvergiert das Iterationsverfahren $x^{(k+1)} = \Phi(x^{(k)})$ lokal mit der Ordnung p gegen $\hat{x}$.

Beweis. Zuerst wenden wir Theorem 5.4 auf eine offene Umgebung $\mathcal{U}$ in (a, b) an, in der $\Phi'(x) \leq q < 1$ gilt. Dies ist wegen (5.3) möglich. Nach Theorem 5.4 ist damit die Folge $x^{(k)}$ konvergent. Da $\Phi^{(p+1)}$ beschränkt in $\mathcal{U}$ ist und $\Phi^{(p)}(\hat{x}) \neq 0$, existiert eine Umgebung $\mathcal{U}$ von $\hat{x}$ in (a, b), sodass für alle $x^{(k)}$ ab einem gewissen großen Index k_0 gilt

$$\left| \frac{\Phi^{(p+1)}(\xi)}{(p+1)!} (x^{(k)} - \hat{x})^{p+1} \right| < \frac{1}{2} \left| \frac{\Phi^{(p)}(\hat{x})}{p!} \right| , \tag{5.4}$$

wobei ξ ein beliebiger Punkt in $\mathcal{U}$ ist.

Durch Taylorentwicklung ergibt sich

$$\Phi(x^{(k)}) = \Phi(\hat{x}) + \sum_{i=1}^{p} \frac{\Phi^{(i)}(\hat{x})}{i!}(x^{(k)} - \hat{x})^i + \frac{\Phi^{(p+1)}(\zeta)}{(p+1)!}(x^{(k)} - \hat{x})^{p+1};$$

für ein ζ zwischen $\hat{x}$ und $x^{(k)}$. Da $\hat{x}$ ein Fixpunkt ist, ist nach Voraussetzung also

$$x^{(k+1)} = \Phi(x^{(k)}) = \hat{x} + \frac{\Phi^{(p)}(\hat{x})}{p!}(x^{(k)} - \hat{x})^p + \frac{\Phi^{(p+1)}(\zeta)}{(p+1)!}(x^{(k)} - \hat{x})^{p+1} .$$

Damit gilt wegen der Dreiecksungleichung

$$|x^{(k+1)} - \hat{x}| \geq \left| \frac{\Phi^{(p)}(\hat{x})}{p!}(x^{(k)} - \hat{x})^p \right| - \left| \frac{\Phi^{(p+1)}(\zeta)}{(p+1)!}(x^{(k)} - \hat{x})^{p+1} \right| .$$

Damit folgt aus (5.4)

$$\frac{1}{2}\frac{|\Phi^{(p)}(\hat{x})|}{p!}|x^{(k)} - \hat{x}|^p < |x^{(k+1)} - \hat{x}| < \frac{3}{2}\frac{|\Phi^{(p)}(\hat{x})|}{p!}|x^{(k)} - \hat{x}|^p .$$

Für $|x^{(k)} - \hat{x}|^{p-1} < p!/|3\Phi^{(p)}(\hat{x})|$ ist die rechte Seite kleiner als $|x^{(k)} - \hat{x}|/2$ und daher konvergiert die Iteration lokal gegen $\hat{x}$. Offensichtlich ist die Konvergenzordnung p. □

5.2 Nullstellenbestimmung reeller Funktionen

Im folgenden betrachten wir superlinear konvergente Iterationsverfahren zur Lösung der Nullstellengleichung

$$f(\hat{x}) = 0 . \tag{5.5}$$

Dabei beschränken wir uns zunächst auf den Fall einer reellwertigen Funktion f einer Variablen $x \in [a, b] \subseteq \mathbb{R}$: der mehrdimensionale Fall wird später behandelt.

5.2.1 Das Newton-Verfahren

Um die allgemeinen Ergebnisse dieses Kapitels nutzen zu können, bringen wir die Gleichung (5.5) zuerst in Fixpunktform. Denkbar ist etwa eine Gleichung der Form

$$x = x + q(x)f(x) =: \Phi(x)$$

mit einer glatten Funktion q, von der wir zunächst nur voraussetzen, dass $q(x)$ in einer Umgebung von $\hat{x}$ von Null verschieden ist. Mit Hinblick auf Satz 5.5 fordern wir zunächst, dass $\Phi'(x)$ verschwindet – wir wollen ein Verfahren mit hoher Konsistenzordnung generieren. Also wollen wir erreichen, dass

$$\Phi'(\hat{x}) = 1 + q'(\hat{x})f(\hat{x}) + q(\hat{x})f'(\hat{x}) = 0 \ .$$

Da $f(\hat{x}) = 0$, erhalten wir somit die Bedingung

$$\Phi'(\hat{x}) = 1 + q(\hat{x})f'(\hat{x}) = 0 \ ,$$

oder

$$q(\hat{x}) = -\frac{1}{f'(\hat{x})} \ .$$

Dies ist natürlich nur für $f'(\hat{x}) \neq 0$ möglich. Mit der Wahl $q = -1/f'$ resultiert das Newton-Verfahren.

Satz 5.6. *Sei $f \in C^3[a,b]$ und $\hat{x} \in (a,b)$ mit $f(\hat{x}) = 0$ und $f'(\hat{x}) \neq 0$. Dann konvergiert das* **Newton-Verfahren**

$$x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})} \tag{5.6}$$

lokal quadratisch gegen $\hat{x}$.

Beweis. Die Behauptung folgt aufgrund der Konstruktion sofort aus Satz 5.5 . □

Leider ist die Konvergenz des Newton-Verfahrens in der Regel nur lokal. Nur in Ausnahmefällen kann globale Konvergenz garantiert werden. Eine solche Ausnahme ist der Fall einer konvexen Funktion f.

Satz 5.7. *Sei $I \subseteq \mathbb{R}$ ein Intervall und $f : I \to \mathbb{R}$ monoton wachsend und konvex mit (eindeutiger) Nullstelle $\hat{x} \in I$. Dann konvergiert das Newton-Verfahren für alle $x^{(0)} \in I$ mit $x^{(0)} \geq \hat{x}$ monoton gegen $\hat{x}$.*

Beweis. Wir nehmen an, dass für ein $k \geq 0$ die aktuelle Iterierte $x^{(k)} \geq \hat{x}$ ist und beweisen die Induktionsbehauptung

$$\hat{x} \leq x^{(k+1)} \leq x^{(k)} \,. \tag{5.7}$$

Hierzu wenden wir die Konvexitätsbedingung an, dass für alle $u, v \in I$ und $0 \leq \alpha \leq 1$

$$f(\alpha u + (1-\alpha)v) \leq \alpha f(u) + (1-\alpha) f(v)$$

gilt auf $u = x^{(k+1)}$ und $v = x^{(k)}$. Dies ergibt

$$\alpha f(x^{(k+1)}) \geq f(\alpha x^{(k+1)} + (1-\alpha)x^{(k)}) - (1-\alpha) f(x^{(k)}) \,,$$

beziehungsweise

$$f(x^{(k+1)}) \geq \frac{f(x^{(k)} + \alpha(x^{(k+1)} - x^{(k)})) - f(x^{(k)})}{\alpha} + f(x^{(k)}) \,.$$

Dies gilt nach Voraussetzung für alle $\alpha \in (0, 1]$. Durch Grenzübergang $\alpha \to 0$ folgt somit

$$f(x^{(k+1)}) \geq f'(x^{(k)})(x^{(k+1)} - x^{(k)}) + f(x^{(k)}) \,.$$

Man beachte, dass eine konvexe Funktion differenzierbar ist, weshalb wir im Satz diese Eigenschaft nicht extra voraussetzen müssen. Die rechte Seite ist aufgrund der Newton Vorschrift (5.6) Null. Also ist $f(x^{(k+1)})$ nichtnegativ und wegen der Monotonie folglich $x^{(k+1)} \geq \hat{x}$. Da nach Voraussetzung und Induktionsvoraussetzung sowohl $f(x^{(k)})$ als auch $f'(x^{(k)})$ nichtnegativ sind, gilt wegen der Definition des Newton Verfahrens,

$$x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})},$$

dass $x^{(k+1)} \leq x^{(k)}$. Damit ist die Induktionsbehauptung (5.7) vollständig bewiesen. □

5.2.2 Das Sekantenverfahren

Der Aufwand bei der Implementierung des Newton-Verfahrens (5.6) steckt in der Auswertung von f und f'. In der Praxis ist die Funktion f oft nicht explizit bekannt oder um ein Vielfaches komplizierter als die Funktion f

selbst. Daher ersetzt man gelegentlich die Ableitung $f'(x^{(k)})$ in (5.6) durch den Differenzenquotienten, etwa

$$f'(x^{(k)}) \approx \frac{f(x^{(k)}) - f(x^{(k-1)})}{x^{(k)} - x^{(k-1)}} \ .$$

Auf diese Weise erhält man für $k \geq 1$ die Iterationsvorschrift des **Sekantenverfahrens**:

$$\begin{aligned} x^{(k+1)} &= x^{(k)} - \frac{x^{(k)} - x^{(k-1)}}{f(x^{(k)}) - f(x^{(k-1)})} f(x^{(k)} \\ &= \frac{x^{(k-1)} f(x^{(k)}) - x^{(k)} f(x^{(k-1)})}{f(x^{(k)}) - f(x^{(k-1)})} \ . \end{aligned} \tag{5.8}$$

Der Name Sekantenverfahren beruht auf der folgenden geometrischen Interpretation:

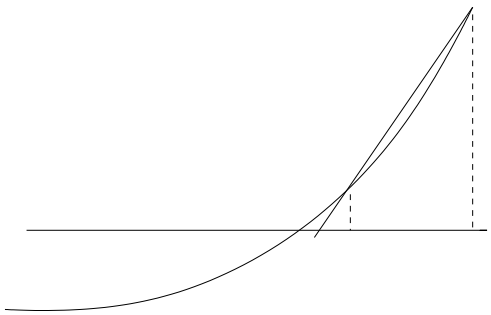

Abbildung 5.1: Geometrische Interpretation des Sekantenverfahrens

Satz 5.8. *f sei zweimal stetig differenzierbar in $[a, b]$ und es sei $f(\hat{x}) = 0$ für ein $\hat{x} \in (a, b)$ mit $f'(\hat{x}) \neq 0$ und $f''(\hat{x}) \neq 0$. Dann konvergiert das Sekantenverfahren lokal gegen $\hat{x}$ mit exakter Konvergenzordnung*

$$p = \frac{1}{2}(1 + \sqrt{5}) = 1.61803 \cdots$$

5.3 Das Newton–Verfahren in $\mathbb{R}^n$

Das Newton–Verfahren (5.6) lässt sich unmittelbar zur Nullstellenbestimmung einer Funktion $F : \mathcal{D}(F) \subseteq \mathbb{R}^n \to \mathbb{R}^n$ übertragen:

$$x^{(k+1)} = x^{(k)} - \nabla F(x^{(k)})^{-1} F(x^{(k)}) \ . \tag{5.9}$$

Hierbei ist $\nabla F(x)$ die *Jacobi-Matrix*

$$\nabla F(x) = \left[\frac{\partial F_i}{\partial x_j}\right]_{ij} \in \mathbb{R}^{n\times n} .$$

Dabei bezeichnet i die Zeilen und j die Spaltenposition.

Um die Wohldefiniertheit dieses Verfahrens sicherzustellen, müssen wir nun fordern, dass $\nabla F(\hat{x})$ nicht singulär ist; dies entspricht der Verallgemeinerung der Bedingung im eindimensionalen Fall, wo wir gefordert haben, dass $f'(\hat{x}) \neq 0$. Die Rekursion (5.9) ist äquivalent zu

$$F(x^{(k)}) + \nabla F(x^{(k)})(x^{(k+1)} - x^{(k)}) = 0 . \tag{5.10}$$

Also ist $x^{(k+1)}$ eine Nullstelle der Taylor–Linearisierung von F um $x^{(k)}$. In (5.10) wird die ursprüngliche *nichtlineare* Gleichung

$$F(x) = 0$$

durch die lokale *Linearisierung*

$$F(x^{(k)}) + \nabla F(x^{(k)})(x - x^{(k)}) = 0$$

ersetzt. Auch für die Implementierung des Newton–Verfahrens verwendet man zumeist die äquivalente Formulierung (5.10) anstelle von (5.9).

Algorithmus 5.9. Newton–Verfahren in $\mathbb{R}^n$:

- `Wähle` $x^{(0)} \in \mathcal{D}(F)$
- `for` $k = 1, 2, \cdots$
  `löse das lineare Gleichungssystem`

  $$\nabla F(x^{(k)})h^{(k)} = -F(x^{(k)})$$

  `Ersetze` $x^{(k+1)} = x^{(k)} + h^{(k)}$ `und überprüfe` $x^{(k+1)} \in \mathcal{D}(F)$
  `end for`

Für $n = 1$ ist dieser Algorithmus äquivalent zu der Iterationsvorschrift (5.6). In $\mathbb{R}^n$ verwendet man jenen linearen Gleichungssystemlöser, der die speziellen Eigenschaften der Jacobi–Matrizen am besten nutzt.

Wir beweisen nun einen Konvergenzsatz für Algorithmus 5.9.

Satz 5.10. *$\|\cdot\|$ und $|||\cdot|||$ seien ein Paar einer verträglichen Vektor- bzw. Matrixnorm. $F : \mathcal{D}(F) \subseteq \mathbb{R}^n \to \mathbb{R}^n$ habe eine Nullstelle $\hat{x}$ und sei stetig differenzierbar in einer Kugel $\mathcal{U} \subseteq \mathcal{D}(F)$ um $\hat{x}$. $\nabla F(x)$ sei invertierbar für alle $x \in \mathcal{U}$ mit*

$$\left|\left|\left|\nabla F(x)^{-1}(\nabla F(y) - \nabla F(x))\right|\right|\right| \leq L\|y - x\| \tag{5.11}$$

für alle $x, y \in \mathcal{U}$ mit einem festem $L > 0$. Dann gilt: Ist $x^{(0)} \in \mathcal{U}$ mit

$$\rho := \|\hat{x} - x^{(0)}\| < 2/L\,,$$

dann liegen alle Iterierten $x^{(k)}$ von (5.9) in der Kugel $\mathcal{U}_\rho(\hat{x})$ mit Radius ρ um $\hat{x}$ und konvergieren quadratisch gegen $\hat{x}$,

$$\|\hat{x} - x^{(k+1)}\| \leq \frac{L}{2}\|\hat{x} - x^{(k)}\|^2\,, \quad k = 0, 1, 2, \cdots \tag{5.12}$$

Beweis. Da $\hat{x}$ eine Nullstelle von F ist, gilt

$$\begin{aligned} x^{(k+1)} - \hat{x} &= x^{(k)} - \nabla F(x^{(k)})^{-1}F(x^{(k)}) - \hat{x} \\ &= x^{(k)} - \hat{x} - \nabla F(x^{(k)})^{-1}(F(x^{(k)}) - F(\hat{x})) \\ &= \nabla F(x^{(k)})^{-1}\left(F(\hat{x}) - F(x^{(k)}) - \nabla F(x^{(k)})(\hat{x} - x^{(k)})\right)\,. \end{aligned}$$

Aus dem Mittelwertsatz folgt

$$\begin{aligned} &x^{(k+1)} - \hat{x} \\ &= \nabla F(x^{(k)})^{-1}\left(\int_0^1 \nabla F(x^{(k)} + t(\hat{x} - x^{(k)}))(\hat{x} - x^{(k)})\,dt\; - \nabla F(x^{(k)})(\hat{x} - x^{(k)})\right) \\ &= \int_0^1 \nabla F(x^{(k)})^{-1}\left((\nabla F(x^{(k)} + t(\hat{x} - x^{(k)})) - \nabla F(x^{(k)}))(\hat{x} - x^{(k)})\right)\,dt\,. \end{aligned}$$

Folglich gilt

$$\begin{aligned} &\|x^{(k+1)} - \hat{x}\| \\ &\leq \int_0^1 \left|\left|\left|\nabla F(x^{(k)})^{-1}(\nabla F(x^{(k)} + t(\hat{x} - x^{(k)})) - \nabla F(x^{(k)}))\right|\right|\right| \|\hat{x} - x^{(k)}\|\,dt \\ &\leq L\|x^{(k)} - \hat{x}\|^2 \int_0^1 t\,dt \\ &= \frac{L}{2}\|x^{(k)} - \hat{x}\|^2\,, \end{aligned}$$

womit die quadratische Konvergenz (5.12) nachgewiesen ist. Gilt zudem $\|x^{(k)} - \hat{x}\| \leq \rho < 2/L$, dann ist

$$\|x^{(k+1)} - \hat{x}\| \leq \left(\frac{L}{2}\|x^{(k)} - \hat{x}\|\right) \|x^{(k)} - \hat{x}\| \leq \rho\frac{L}{2}\|x^{(k)} - \hat{x}\| < \|x^{(k)} - \hat{x}\| < \rho \; .$$

Damit haben wir gezeigt, dass $\|x^{(k)} - \hat{x}\|$ monoton fallend ist und $x^{(k)} \in \mathcal{U}_\rho(\hat{x})$. Damit ist insbesondere die Folge $(\|x^{(k)} - \hat{x}\|)_{k\in\mathbb{N}}$ konvergent. Bezeichnen wir den Grenzwert mit ε, so erfüllt er wegen (5.12) die Ungleichung

$$0 \leq \varepsilon \leq \frac{L}{2}\varepsilon^2 \leq \rho\frac{L}{2}\varepsilon \; .$$

Da $\rho\frac{L}{2} < 1$, muss $\varepsilon = 0$ sein. Also konvergiert $x^{(k)}$ gegen $\hat{x}$. □

Bemerkung 5.11.

1. In der Kugel $\mathcal{U}_\rho(\hat{x})$ kann es nur eine Nullstelle von F geben; ist nämlich $\tilde{x}$ eine zweite Nullstelle von F, dann konvergiert die Iteration (5.9) mit $x^{(0)} = \tilde{x}$ gegen $\tilde{x}$. Aus (5.12) folgt dann der Widerspruch zur Annahme, dass es zwei Nullstellen gibt.

2. (5.11) ist eine Art Lipschitz–Bedingung an $\nabla F(\cdot)$. Die Konstante ist dabei unabhängig von möglichen Transformationen

$$\tilde{F}(x) := AF(x) \text{ mit nichtsingulärem } A \in \mathbb{R}^{n\times n} \; .$$

3. Für $n = 1$ ist (5.11) erfüllt, falls ∇F Lipschitz-stetig ist. Dies ist eine deutlich schwächere Voraussetzung als in Satz 5.6, wo wir gefordert haben, dass $f \in C^3(a, b)$.

Da die Konvergenzaussage von Satz 5.10 nur lokal gültig ist, stellt sich die Frage, wie man in der Praxis entscheiden kann, ob das Verfahren mit dem verwendeten $x^{(0)}$ konvergiert oder nicht. Sehr häufig wird ein **Monotonietest** verwendet, wo geprüft wird, ob

$$\|\nabla F(x^{(k)})^{-1}F(x^{(k+1)})\| \leq \frac{1}{2}\|\nabla F(x^{(k)})^{-1}F(x^{(k)})\| \; . \tag{5.13}$$

Der Term $h^{(k)} = \nabla F(x^{(k)})^{-1}F(x^{(k)})$ wird in Algorithmus 5.9 berechnet, also ist für den Monotonietest die Berechnung der rechten Seite kein erheblicher

Mehraufwand. Hingegen bedeutet die Berechnung der rechten Seite im allgemeinen einen erheblichen Mehraufwand. Dieser ist aber vernachlässigbar, wenn man das lineare Gleichungssystem

$$\nabla F(x^{(k)})h^{(k)} = -F(x^{(k)})$$

mit einer $LR-$Zerlegung von $\nabla F(x^{(k)})$ gelöst wird. In diesem Fall ist der Aufwand zur Lösung des Gleichungssystems

$$\nabla F(x^{(k)})d^{(k)} = F(x^{(k+1)})$$

gegenüber der Bestimmung der LR-Zerlegung vernachlässigbar. Für den Monotonietest muss man schließlich noch die Norm von $d^{(k)}$ bestimmen.

Ist der Monotonietest über einige Iterationen hinweg nicht erfüllt, dann muss befürchtet werden, dass die Iteration nicht konvergiert. In diesem Fall muss die Newton Iteration mit einer besseren Startnäherung begonnen werden.

Kapitel 6

Numerische Quadratur

Gegenstand dieses Kapitels ist die numerische Approximation bestimmter Integrale

$$I[f] = \int_a^b f(x)\,dx\,,$$

die nicht in geschlossener Form durch Angabe einer Stammfunktion integriert werden können.

6.1 Trapezregel

Die einfachsten Approximationsformeln sind die *Mittelpunktsformel*

$$\int_a^b f(x)\,dx \approx (b-a) f\left(\frac{a+b}{2}\right) \tag{6.1}$$

und die *Trapezformel*

$$\int_a^b f(x)\,dx \approx \frac{b-a}{2} f(a) + \frac{b-a}{2} f(b)\,. \tag{6.2}$$

Natürlich gilt bei (6.1) und (6.2) in der Regel keine Gleichheit; die Formel ist nicht exakt. In der Praxis zerlegt man das Intervall $[a,b]$ in n gleich große Teilintervalle und wendet (6.1) und (6.2) auf jedes Teilintervall an. Bei der Mittelpunktsformel ergibt sich somit eine Riemann'sche Zwischensumme, während die Trapezformel auf die *(zusammengesetzte) Trapezregel* (oder

Trapezsumme) führt:

$$a = x_0 < x_1 < x_2 < \cdots < x_n = b\,, \qquad x_i = a + ih\,, h = \frac{b-a}{n}\,,$$
$$T_n[f] := \sum_{i=1}^{n} \frac{x_i - x_{i-1}}{2}(f(x_i) + f(x_{i-1})) = \frac{h}{2}f(a) + h\sum_{i=1}^{n-1} f(x_i) + \frac{h}{2}f(b)\,. \tag{6.3}$$

Aus der Definition des Riemann-Integral folgt unmittelbar, dass die entsprechend zusammengesetzten Regeln für $n \to \infty$ gegen $I[f]$ konvergieren. Unter Zusatzvoraussetzungen an f kann folgende Fehlerabschätzung bewiesen werden:

Satz 6.1. *Sei $f \in C^2[a,b]$. Dann gilt*

$$|I[f] - T_n[f]| \leq \frac{b-a}{12}\|f''\|_{[a,b]}h^2$$

mit $h = \frac{(b-a)}{n}$. Dabei bezeichnet $\|\cdot\|_{[a,b]}$ die Matrixnorm über dem Intervall $[a,b]$.

Beweis. Wir betrachten zunächst die Trapezformel, also $n = 1$. Dann gilt

$$T_1[f] = \frac{b-a}{2}f(a) + \frac{b-a}{2}f(b) = \int_a^b p(x)\,dx$$

mit

$$p(x) = f(a) + \frac{x-a}{b-a}(f(b) - f(a))\,.$$

Die letzte Identität sieht man indem man die Probe macht:

$$\begin{aligned}\int_a^b p(x)\,dx &= f(a)(b-a) + \frac{f(b)-f(a)}{b-a}\frac{1}{2}(b-a)^2 \\ &= f(a)(b-a) + \frac{b-a}{2}(f(b) - f(a)) \\ &= \frac{b-a}{2}(f(b) + f(a))\,.\end{aligned}$$

Also ist

$$|I[f] - T_1[f]| = \left|\int_a^b (f(x) - p(x))\,dx\right| \leq \int_a^b |f(x) - p(x)|\,dx\,.$$

Sei $g(x) := f(x) - p(x)$ und definiere für $a < t < b$

$$h_t(x) := g(x) - \frac{w(x)}{w(t)} g(t)\,, \qquad w(x) = (x-a)(x-b)\,.$$

h_t hat Nullstellen in $x = a$, $x = b$ (da $g(a) = g(b) = w(a) = w(b) = 0$), sowie in $x = t$. Also hat h_t mindestens einen Wendepunkt ζ in (a, b): Dort gilt

$$0 = h_t''(\zeta) = g''(\zeta) - \frac{g(t)}{w(t)} w''(\zeta) = g''(\zeta) - 2\frac{g(t)}{w(t)}\,,$$

bzw.

$$g(t) = \frac{1}{2} g''(\zeta)(t-a)(t-b)\,. \tag{6.4}$$

Da $g''(x) = f''(x)$ gilt folgt

$$\begin{aligned} |I[f] - T_1[f]| &\le \int_a^b |g(t)|\, dt \\ &\le \frac{1}{2} \|g''\|_{[a,b]} \int_a^b (t-a)(t-b)\, dt \\ &= \frac{1}{12} \|f''\|_{[a,b]} (b-a)^3\,. \end{aligned}$$

Angewendet auf (6.3) ergibt sich

$$\begin{aligned} |I[f] - T_n[f]| &\le \sum_{i=1}^n \left| \int_{x_{i-1}}^{x_i} f(x) dx - \frac{h}{2}(f(x_{i-1}) - f(x_i)) \right| \\ &\le \frac{1}{12} \sum_{i=1}^n \|f''\|_{[a,b]} h^3 \\ &\le \frac{b-a}{12} \|f''\|_{[a,b]} h^2\,. \end{aligned}$$

□

Im Folgenden betrachten wir weitere Methoden zur Approximation des *gewichteten* Integrals

$$I_{\hat{w}}[f] = \int_a^b f(x)\hat{w}(x)\, dx\,.$$

Dabei sei $\hat{w}(x)$ eine positive integrierbare Funktion über dem Intervall $I := [a, b]$ mit

$$\int_I \hat{w}(x)\, dx < \infty \, .$$

Zur Approximation von $I_{\hat{w}}[f]$ betrachten wir Ausdrücke der Form

$$Q[f] = \sum_{i=0}^{m} \hat{w}_i f(x_i)$$

mit *Knoten* x_i und *Gewichten* $\hat{w}_i$. Eine solche Näherung nennen wir *Quadraturformel*, wenn die Wahl von m, $\{x_i\}$ und $\{\hat{w}_i\}$ **fest** ist (wie etwa bei der Trapezregel (6.2). Unter dem zugehörigen *zusammengesetzten Quadraturverfahren* verstehen wir dann die Unterteilung von $[a, b]$ in n gleich große Teilintervalle, auf die jeweils eine Quadraturformel angewendet wird (Bezeichnung $Q_n[f]$).

Um die qualitativen Merkmale einer Quadraturformel bzw. eines zusammengesetzten Quadraturverfahrens beschreiben zu können, führen wir die Begriffe *Exaktheitsgrad* und *Konvergenzordnung* eines Quadraturverfahrens ein. Dabei bezeichnet Π_m den Vektorraum aller Polynome vom Grad höchstens m.

Definition 6.2.

1. Eine Quadraturformel $Q[f]$ hat **Exaktheitsgrad** q, falls
$$Q[p] = I_{\hat{w}}[p] \text{ für alle } p \in \Pi_q \, .$$

2. Ein zusammengesetztes Quadraturverfahren konvergiert gegen $I_{\hat{w}}[f]$ mit der Ordnung s, falls
$$|Q_n[f] - I_{\hat{w}}[f]| = O(n^{-s}) \, , \quad n \to \infty \, .$$

Beispiel 6.3. Sei $\hat{w} = 1$, dann hat die Trapezformel Exaktheitsgrad $q = 1$, und die zusammengesetzte Trapezformel konvergiert mit Ordnung $s = 2$ für alle $f \in C^2[a, b]$.

Kapitel 7

Gewöhnliche Differentialgleichungen

Wir werden im ersten Teil dieser Vorlesung Theorie und Numerik für Systeme von gewöhnlichen Differentialgleichungen der Form

$$y' = f(t, y)\ , t \in [0, T] \text{ mit der Anfangsbedingung } y(0) = y_0 \qquad (7.1)$$

behandeln. Dabei ist zu beachten, dass y eine vektorwertige Funktion sein kann. Wir sprechen in diesem Fall von einem *System erster Ordnung*.

7.1 Das Euler Verfahren

Obwohl dieses Verfahren heute in der Praxis kaum mehr verwendet wird, so kann man doch sehr eindringlich die Idee von Lösungsverfahren für gewöhnliche Differentialgleichungen schildern.

Das Euler-Verfahren beruht auf der Beobachtung, dass die Funktion f in jedem Punkt $(t, y) \in \Omega$ die Steigung der Lösungskurve definiert. Sobald eine Lösungskurve $(t, y(t))$ durch den Punkt (t, y) läuft, hat die Tangente an die Kurve in diesem Punkt die Steigung $f(t, y(t))$.

Das Euler-Verfahren (oder auch Polygonzugverfahren) macht sich diesen Sachverhalt wie folgt zunutze: Auf einem vorgegebenen Gitter

$$\Delta = \{0 = t_0 < t_1 < t_2 < ... < t_n\} \subseteq I$$

wird diejenige Gerade als lokale Approximation der Funktion $y(.)$ gewählt, dessen *rechtsseitige Ableitung* in dem jeweiligen Gitterknoten mit der vorgegebenen Steigung $f(t, y(t))$ übereinstimmt. Da durch y_0 und $f(0, y_0)$ am

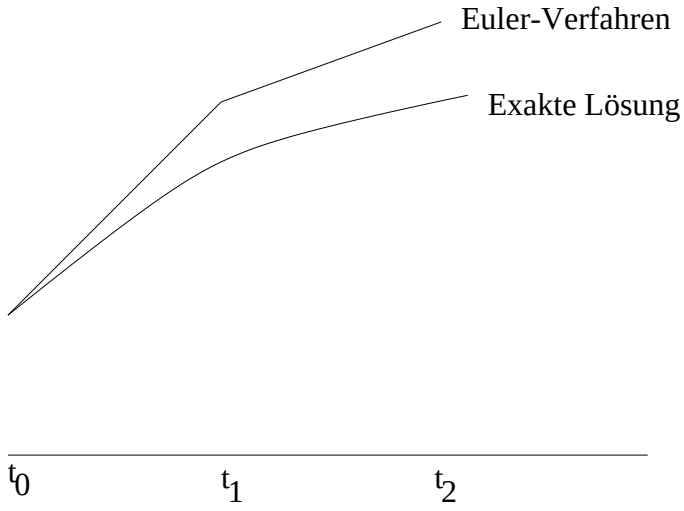

Abbildung 7.1: Schematische Darstellung des Euler-Verfahrens

linken Rand der Funktionswert und die Anfangssteigung der Geraden festgelegt sind, lassen sich die Koeffizienten $y_i \in \mathbb{R}^d$ der Geraden in **expliziter** Weise rekursiv bestimmen:

$$y_{i+1} = y_i + (t_{i+1} - t_i) f(t_i, y_i) \; .$$

Man beachte, dass das Euler-Verfahren auch dadurch erklärt werden kann, dass man die Zeitableitung durch einen *Vorwärtsdifferenzenquotienten* approximiert.

Beispiel 7.1. Wir studieren die einfache gewöhnliche Differentialgleichung

$$y' = y, \quad y(0) = 1 \; .$$

Die exakte Lösung ist $y(t) = \exp(t)$. Mit dem Euler-Verfahren ergibt sich in einem äquidistanten Gitter ($t_i = ih$) (man beachte $f(t, y) = y$)

$$\begin{aligned} y_0 &= 1 \, , \\ y_1 &= y_0 + hf(h, y_0) = 1 + h \, , \\ y_2 &= y_1 + hf(h, y_1) = 1 + h + h(1 + h) = (1 + h)^2 \, , \\ y_3 &= y_2 + hf(h, y_2) = (1 + h)^2 + h(1 + h)^2 = (1 + h)^3 \; . \end{aligned}$$

Mit Induktion kann man leicht zeigen, dass $y_n = (1 + h)^n$. Für $t_n = T$ bedeutet dies

$$y_n = (1 + T/n)^n \to \exp(T) = y(T) \, , \quad n \to \infty \; .$$

Klarerweise löst das Euler-Verfahren die gewöhnliche Differentialgleichung *nicht* exakt. Wir studieren im folgenden wie gut das Euler-Verfahren die Lösung der gewöhnlichen Differentialgleichung approximiert. Fehlerabschätzungen der nun folgenden Form sind von zentraler Bedeutung, um die Qualität von numerischen Verfahren zur Lösung von gewöhnlichen Differentialgleichungen zu beurteilen.

Für die folgende Fehlerabschätzung beschränken wir uns auf ein äquidistantes Gitter Δ mit konstanten Schrittweite $h = t_{i+1} - t_i$, $i = 0, ..., n$.

Satz 7.2. *Sei $I = [0, T]$ und $f : I \times \mathbb{R}^d \to \mathbb{R}^d$ stetig differenzierbar und bezüglich y global Lipschitz-stetig,*

$$\|f(t, y) - f(t, z)\|_2 \leq L\|y - z\|_2 \text{ für alle } t \in I \text{ und } y, z \in \mathbb{R}^d \, .$$

Ist nun y die eindeutige Lösung des Anfangswertproblems (7.1) und sind y_i, $i = 1, ..., n$, die berechneten Näherungen des Euler-Verfahren an den Gitterpunkten $t_i \in I$, dann gilt

$$\|y(t_i) - y_i\|_2 \leq \frac{(1 + Lh)^i - 1}{2L}\|y''\|_{[0,T]}h \leq \frac{\exp(Lt_i) - 1}{2L}\|y''\|_{[0,T]}h \, , \quad i = 0, ..., n \, .$$

Hierbei ist $\|y''\|_{[0,T]} = \max_{0 \leq t \leq T} \|y''(t)\|_2$ (man beachte $y''(t) \in \mathbb{R}^d$).

Beweis: Die Voraussetzung an f garantiert, dass y zweimal stetig differenzierbar ist, wie man aus folgendem heuristischen Argument sofort sieht:

$$y'' = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial y}y' = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial y}f \, .$$

Der Beweis dieses Satzes gliedert sich nun in drei Teile:

Lokaler Fehler: Nehmen wir zunächst an, dass für t_i das Euler-Verfahren auf dem Punkt $(t_i, y(t_i))$ der *exakten* Lösungskurve initiiert ist. Sei z_{i+1} die Approximation für $y(t_{i+1})$, die man mit dem Euler-Verfahren

berechnet, dann gilt

$$
\begin{array}{ll}
\|y(t_{i+1}) - z_{i+1}\|_2 & \\
\underbrace{=}_{\text{Def. des Eulerverfahrens}} & \|y(t_{i+1}) - (y(t_i) + hf(t_i, y(t_i)))\|_2 \\
\underbrace{=}_{\text{Def. der Differentialgleichung}} & \|y(t_{i+1}) - y(t_i) + hy'(t_i)\|_2 \\
\underbrace{=}_{\text{Hauptsatz der Integralrechnung}} & \left\| \int_{t_i}^{t_{i+1}} y'(\tau)\, d\tau - y'(t_i)h \right\|_2 \\
\underbrace{=}_{\text{Def. von } h} & \left\| \int_{t_i}^{t_{i+1}} y'(\tau) - y'(t_i)\, d\tau \right\|_2 \\
\underbrace{\leq}_{\text{Mittelwertsatz der Integralrechnung}} & \|y''\|_{[0,T]} \int_{t_i}^{t_{i+1}} (\tau - t_i)\, d\tau \\
\leq & \frac{1}{2}\|y''\|_{[0,T]} h^2 \ .
\end{array}
$$

Lokale Fehlerfortpflanzung: Tatsächlich ist das Euler-Verfahren nach i Schritten nicht auf der exakten Lösungskurve, sondern hat statt dessen eine Näherung y_i von $y(t_i)$ berechnet. Daher müssen wir untersuchen, wie sich der Fehler im i-ten Schritt, $y_i - y(t_i)$, auf das Resultat im $i+1$-Schritt auswirkt. Mit den Rechenfehlern des Euler-Verfahrens ergibt sich

$$y_{i+1} = y_i + hf(t_i, y_i) \text{ und } z_{i+1} = y(t_i) + hf(t_i, y(t_i)) \ .$$

Man beachte, dass der erste Term zur Berechnung von y_{i+1}, Fehler in $y(t_i)$ mit berücksichtigt. Damit folgt also

$$
\begin{array}{ll}
\|y_{i+1} - z_{i+1}\|_2 & \\
\leq & \|y_i - y(t_i)\|_2 + h\|f(t_i, y_i) - f(t_i, y(t_i))\|_2 \\
\leq & (1 + hL)\|y_i - y(t_i)\|_2 \ .
\end{array}
$$

Kummulierter Fehler: Jetzt leiten wir eine obere Schranke für die Norm des Gesamtfehlers ε_i nach i Zeitschritten her. Ziel und Aussage des Satzes ist es zu zeigen, dass

$$\varepsilon_i \leq \frac{(1 + Lh)^i - 1}{2L} \|y''\|_{[0,T]} h \,, \quad i = 0, ..., n \ . \tag{7.2}$$

Wir führen den Beweis dieser Behauptung mit Induktion: Für $i = 0$ gilt per Definition des Euler-Verfahrens $y(t_0) = y_0$ und $\varepsilon_i = 0$; damit

ist (7.2) in diesem Fall trivialerweise erfüllt. Aus den ersten beiden Beweisschritten ergibt sich nun

$$\begin{aligned} \|y(t_{i+1}) - y_{i+1}\|_2 &\leq \|z_{i+1} - y_{i+1}\|_2 + \|y(t_{i+1}) - z_{i+1}\|_2 \\ &\leq (1 + hL)\varepsilon_i + \frac{1}{2}\|y''\|_{[0,T]}h^2 \end{aligned} \tag{7.3}$$

Aus der Induktionsannahme folgt nun

$$\begin{aligned} &\|y(t_{i+1}) - y_{i+1}\|_2 \\ &\leq \quad \frac{1}{2L}\left((1 + hL)^{i+1} - 1 - hL + hL\right)\|y''\|_{[0,T]}h \\ &= \quad \frac{(1 + hL)^{i+1} - 1}{2L}\|y''\|_{[0,T]}h \ , \end{aligned} \tag{7.4}$$

was zu zeigen war. Wegen der elementaren Ungleichung $1 + hL \leq \exp(hL)$ und $t_i = ih \in (0, T]$ folgt auch der Rest der Behauptung.

□

Aus Satz 7.2 folgt, dass der Fehler des Euler-Verfahrens *linear* in h gegen Null konvergiert, falls das Gitter sukzessive verfeinert wird. Die Verfeinerung wird aber in dem Moment kritisch, in dem der Rundungsfehler die Größenordnung des lokalen Fehlers erreicht. Die folgende heuristische Überlegung mag das belegen: Nehmen wir an, im $(i + 1)$-ten Schritt kommt zu den bereits untersuchten Fehlern (lokaler Fehler und fortgepflanzter Fehler) noch ein additiver Rundungsfehler der Größenordnung ε (Maschinengenauigkeit) hinzu. Dann erhalten wir anstelle von (7.3) die Ungleichung

$$\varepsilon_{i+1} \leq (1 + hL)\varepsilon_i + \frac{1}{2L}\|y''\|_{[0,T]}h^2 + \varepsilon \ .$$

Induktiv ergibt sich entsprechend

$$\varepsilon_i \leq \frac{\exp(Lih) - 1}{2L}\left(\|y''\|_{[0,T]}h + 2\frac{\varepsilon}{h}\right) \qquad i = 0, ..., n \ . \tag{7.5}$$

Das bedeutet: Der Gesamtfehler des Euler-Verfahrens setzt sich aus einem (für $h \to 0$ konvergenten) Verfahrensfehler und einem (für $h \to 0$ divergentem) fortgepflanztem Rundungsfehler zusammen. Man sieht leicht, dass die Schranke auf der rechten Seite von (7.5) für $h \sim \sqrt{\varepsilon}$ ihren minimalen Wert von der Größenordnung $\sqrt{\varepsilon}$ annimmt.

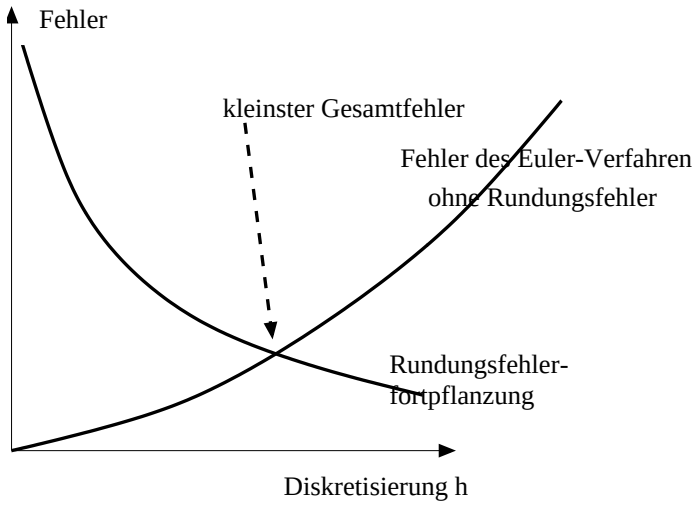

Abbildung 7.2: Gesamtfehler des Euler-Verfahrens

7.2 Das implizite Euler-Verfahren

In diesem Abschnitt verwenden wir als allgemeine Voraussetzung an f die schwache Lipschitz-Bedingung

$$\langle f(t,y) - f(t,z), y - z\rangle_2 \leq l\|y - z\|_2^2 \text{ für alle } (t,y),(t,z) \in \Omega\,. \tag{7.6}$$

Beim impliziten Euler approximiert man die exakte Lösung wieder lokal durch Geraden, aber im Unterschied zum Euler-Verfahren, fordert man nun, dass die *linksseitige Ableitung* der Geraden im Gitterknoten mit dem Wert von $f(t_i, y_i)$ übereinstimmt. Wie der Name des Verfahrens besagt, kann die Bestimmung der Geraden nun nicht mehr explizit erfolgen. Statt dessen ergibt sich y_{i+1} als Lösung des folgenden (im allgemeinen nichtlinearen) Gleichungssystems

$$y_{i+1} = y_i + hf(t_{i+1}, y_{i+1})\,. \tag{7.7}$$

Beispiel 7.3. Wir betrachten wiederum die einfache Differentialgleichung

$$y' = \lambda y\,, \qquad y(0) = 1\,,\ \text{mit } \lambda < 0\,.$$

Die exakte Lösung ist $y(t) = \exp(\lambda t)$. Das implizite Euler-Verfahren hat in diesem einfachen Fall die Form

$$y_{i+1} = y_i + h\lambda y_{i+1} \text{ oder äquivalent } y_{i+1} = \frac{1}{1 - h\lambda} y_i\,. \tag{7.8}$$

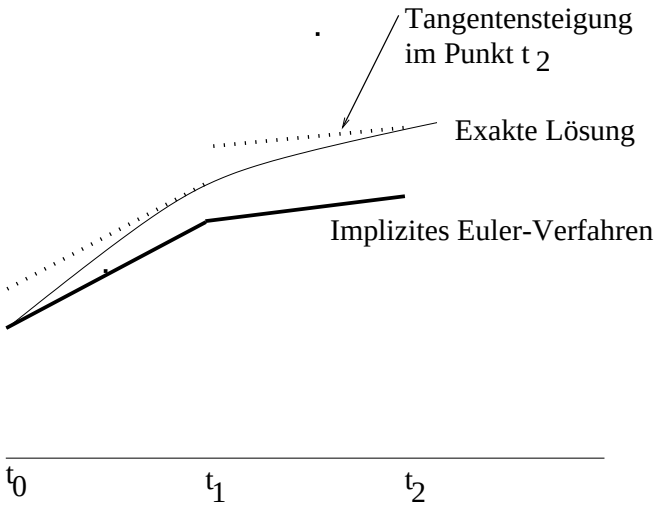

Abbildung 7.3: Schematische Darstellung des impliziten Euler-Verfahrens

Also gilt

$$y_n = \frac{1}{(1-h\lambda)^n} \, .$$

Mit $T = nh$ ergibt sich

$$y_n = \left(1 - \frac{T}{n}\lambda\right)^{-n} \to \exp(\lambda T) = y(T) \, , \quad n \to \infty \, .$$

Aus (7.8) erkennt man, dass die lokale Fehlerverstärkung in einem Zeitschritt durch (beachte $l = \lambda < 0$)

$$\kappa_{IE} = (1 - \lambda h)^{-1} \in (\exp(\lambda h), 1) = (\kappa, 1)$$

gegeben ist. Da diese Beziehung unabhängig von h gilt, ist das implizite Euler-Verfahren **immer gut konditioniert**. Im Gegensatz zum Euler-Verfahren, welches nur dann gut konditioniert ist, falls $|l|h = O(1)$ ist.

Dieses Beispiel zeigt das typische Konvergenzverhalten des impliziten Euler-Verfahren; es ist in vielen Problemen unabhängig von der Wahl von h konvergent. Die Nachteile des impliziten Euler-Verfahrens liegen auf der Hand, es ist die Lösbarkeit des nichtlinearen Gleichung in jedem Iterationsschritt.

Im nächsten Satz studieren wir die Lösbarkeit dieses Gleichungssystems.

Satz 7.4. *Sei $I = [0,T]$, und $f : I \times \mathbb{R}^d \to \mathbb{R}^d$ sei stetig differenzierbar und erfülle die schwache Lipschitz-Bedingung (7.6) für ein $l \in \mathbb{R}$. Dann existiert zu jedem $y \in \mathbb{R}^d$ und jedes $t \in (0,T]$ eine eindeutige Lösung Y der nichtlinearen Gleichung*

$$Y = y + hf(t,Y)\,, \tag{7.9}$$

vorausgesetzt, dass $hl < 1$ ist.

Beweis: Lösungen von (7.9) sind Nullstellen der Funktion

$$F(Y) := y + hf(t,Y) - Y\ .$$

Dabei erfüllt die Funktion F ebenfalls eine schwache Lipschitz-Bedingung:

$$\begin{aligned}\langle F(Y) - F(Z), Y - Z\rangle_2 &= h\,\langle f(t,Y) - f(t,Z), Y - Z\rangle_2 - \|Y - Z\|_2^2 \\ &\le -(1-hl)\|Y - Z\|_2^2\ .\end{aligned}$$

Aus $1 - hl > 0$ folgt unmittelbar, dass F höchstens eine Nullstelle Y haben kann, also die Eindeutigkeit der Lösung von $F(Y) = 0$. Nullstellen Y der Funktion F sind aber auch die stationären Lösungen $u(t)$ der Differentialgleichung

$$u'(\tau) = F(u(\tau))\ .$$

Beachte: τ ist eine künstlich eingeführte Zeit. [1]

Ist die Funktion f stetig differenzierbar, so ist auch F stetig differenzierbar und daher insbesondere lokal Lipschitz-stetig. Also existieren zu jedem $u_0, v_0 \in \mathbb{R}^d$ Lösungen u und v der Differentialgleichung (Satz von Picard-Lindelöf)

$$u' = F(u), u(0) = u_0 \quad v' = F(v), v(0) = v_0\ . \tag{7.10}$$

Ferner gilt für beliebiges t_0 die Ungleichung (Beweis als Hausübung)

$$\|u(t_0) - v(t_0)\|_2 \le \exp(-(1-hl)t_0)\|u_0 - v_0\|_2. \tag{7.11}$$

Da $q := \exp(-(1-hl)t_0) < 1$ gilt, ist die Abbildung $u_0 \to u(t_0)$ eine kontrahierende Selbstabbildung des $\mathbb{R}^d$, und nach dem Banachschen Fixpunktsatz existiert eine eindeutiger Fixpunkt dieser Abbildung, den wir mit Y bezeichnen. Man beachte, dass sich für jedes t_0 verschiedene Y ergeben.

[1] Existiert eine Funktion u mit den Eigenschaften $v = \lim_{t\to\infty} u(t)$ und $\lim_{t\to\infty} u'(t) = 0$, so heisst v stationäre Lösung.

Jede Lösung der Differentialgleichung (7.10) ist t_0 periodisch:

$$v'(\tau) = u'(\tau + t_0) = F(u(\tau + t_0)) = F(v(\tau)) \; .$$

Also sind u und v beides Lösungen von (7.10) mit gleichem Anfangswert $v(0) = u(t_0) = Y = u_0$. Aus der bereits gezeigten Eindeutigkeit folgt nun, dass $u(\tau) = v(\tau) = u(\tau + t_0)$.

Ebenso sieht man, dass für festes $t_1 > 0$ die Funktion $v_1(t) = u(t + t_1)$ eine Lösung der Differentialgleichung (7.10) mit Anfangswert $v_1(0) = u(t_1)$ ist. Aus der t_0 Peridodizität folgt auch die t_0 Periodizität von v_1. Also folgt aus (7.11):

$$\begin{aligned} \|u(t_1) - Y\|_2 &= \|v_1(0) - u(0)\|_2 \\ &= \|v_1(t_0) - u(t_0)\|_2 \\ &\leq \exp(-(1 - hl)t_0)\|v_1(0) - u(0)\|_2 \\ &= q\|u(t_1) - Y\|_2 \; . \end{aligned}$$

Da $q < 1$ ist, muss also $u(t_1) = Y$ sein. Da aber t_1 beliebig war, ergibt sich $u \equiv Y$ und damit ist Y die gesuchte Nullstelle von F. □

Die Bedingung $hl < 1$ ist insbesondere dann erfüllt, wenn l negativ ist. Damit gilt die Bedingung unabhängig von der Konstante L in einer *starken* Lipschitz-Bedingung an f. Im Fall einer positiven Konstante l ergeben sich Einschränkungen an die Schrittweite h.

Nun kommt der angekündigte Konvergenzsatz für das implizite Euler-Verfahren.

Satz 7.5. *Es gelten die Voraussetzungen des Satzes 7.4 an f. Darüberhinaus gelte für die Konstante l im Satz 7.4 hl ¡ 1. Dann gilt für das implizite Euler-Verfahren die Fehlerabschätzung*

$$\|y(t_i) - y_i\|_2 \leq \frac{1}{2l}\left(\left(\frac{1}{1 - lh}\right)^i - 1\right)\|y''\|_{[0,T]}h \, , \quad t_i = ih \in I \; . \tag{7.12}$$

Beweis: Der Aufbau des Beweis dieses Satzes ist ähnlich wie der Beweis zu Satz 7.2, und wird deshalb weggelassen. □

Aus diesem Satz folgt unmittelbar das folgende Korollar

Korollar 7.6. *Es gelten die Voraussetzungen von Satz 7.5 mit einem $l < 0$. Dann gilt für alle $t_i \in [0, T]$ die Abschätzung*

$$\|y(t_i) - y_i\|_2 \leq \frac{1}{2|l|} \|y''\|_{[0,T]} h \, .$$

Zur Implementierung des impliziten Euler-Verfahrens muss man das nichtlineare Gleichungssystem (7.9) effizient lösen. Typischerweise verwendet man ein Newton (-artiges) Verfahren

$$y_{i+1}^{(n+1)} = y_{i+1}^{(n)} - \left(I - h f_y(t_{i+1}, y_{i+1}^{(n)}) \right)^{-1} \left(y_{i+1}^{(n)} - y_i - h f(t_{i+1}, y_{i+1}^{(n)}) \right) \, ,$$
$$n = 0, 1, 2, ...$$

Die Berechnung der **Jacobi-Matrix** $J = f_y(t_{i+1}, y_{i+1}^{(n)})$ und der Inversen von $I - hJ$ sind meist sehr aufwendig. Deshalb verwendet man anstelle des Newton-Verfahrens zumeist **Quasi-Newton Verfahren**, bei der die Jacobi-Matrix geeignet approximiert wird.

7.3 Runge-Kutta Verfahren

Der erhebliche Nachteil der beiden Euler-Verfahren ist ihre langsame Konvergenz (in Abhängigkeit der Zeitdiskretisierung). Betrachtet man die Konvergenzbeweise zu den Sätzen 7.2 und 7.5, so sieht man, dass für diese langsame Konvergenz fast ausschließlich der lokale Fehler verantwortlich ist. Intuitiv lässt sich argumentieren, dass die Tangentensteigung an den Randpunkten des Intervalls $[t_i, t_{i+1}]$ die *Sekante* durch die optimalen Punkte $(t_i, y(t_i))$ und $(t_{i+1}, y(t_{i+1}))$ sehr schlecht approximiert. Es liegt daher nahe, zur Konvergenzverbesserung einen Ansatz der Form

$$y_{i+1} = y_i + h \sum_{j=1}^{s} b_j f(t_i + c_j h, n_j) \, , \quad \sum_{j=1}^{s} b_j = 1 \, , \tag{7.13}$$

zu wählen, mit Näherungen η_j für $y(t_i + c_j h)$; s nennt man dabei die **Stufenzahl** des Verfahrens.

Speziell bei den beiden Euler Verfahren ist jeweils $s = 1$ und $c_1 = 0$, $\eta_1 = y_i$ (explizites Euler-Verfahren), bzw., $c_1 = 1$, $\eta_1 = y_{i+1}$ (implizites Euler-Verfahren).

Da bei (7.13) jeweils ausgehend von $y_i \approx y(t_i)$ die nächste Näherung $y_{i+1} \approx y(t_{i+1})$ berechnet wird, spricht man bei Verfahren dieser Art von **Einschrittverfahren**. Im Gegensatz dazu verwenden **Mehrschrittverfahren** auch ältere Näherungen $y_{i-1}, ...,$ zur Berechnung von y_{i+1}.

Nehmen wir nun an, dass $y_i = y(t_i)$ auf der exakten Lösungskurve liegt und bestimmen davon ausgehend wie im ersten Schritt des Beweises von Satz 7.2 den lokalen Fehler des Verfahrens (7.13), dann ergibt sich mit dem Hauptsatz der Differentialrechnung

$$\begin{aligned} y(t_{i+1}) - y_{i+1} &\underbrace{=}_{\text{Annahme } y(t_i)=y_i} y(t_{i+1}) - y(t_i) - h\sum_{j=1}^{s} b_j f(t_i + c_j h, n_j) \\ &= \int_{t_i}^{t_{i+1}} y'(t)\,dt - h\sum_{j=1}^{s} b_j f(t_i + c_j h, n_j) \\ &\underbrace{=}_{Dgl.} \int_{t_i}^{t_{i+1}} f(t, y(t))\,dt \; - h\sum_{j=1}^{s} b_j f(t_i + c_j h, n_j)\ . \end{aligned}$$

Wir sehen daher, dass der lokale Fehler klein wird, falls die Summe $h\sum_{j=1}^{s} b_j f(t_i + c_j h, n_j)$ eine gute Approximation des entsprechenden Integrals $\int_{t_i}^{t_{i+1}} f(t, y(t))\,dt$ ist.

Daher liegt es nahe, **Quadraturformeln** zur Wahl der Parameter $\{b_j\}$, $\{c_j\}$ und $\{\eta_j\}$ heranzuziehen.

Beispiel 7.7. Mit der Mittelpunktsformel ergibt sich der Ansatz

$$y_{i+1} = y_i + hf(t_i + h/2, \eta_1)\ , \tag{7.14}$$

wobei idealerweise $\eta_1 = y(t_i + h/2)$ sein sollte; allerdings ist dieser Wert nicht bekannt. Eine Näherung kann jedoch leicht gefunden werden durch

$$\eta_1 = y(t_i) + \frac{h}{2}y'(t_i) \approx y_i + \frac{h}{2}f(t_i, y_i)\ .$$

Das ist das **Verfahren von Runge** aus dem Jahre 1895.

Ein andere Alternative ist die Trapezregel. Sie ergibt sich aus dem Ansatz

$$y_{i+1} = y_i + \frac{h}{2}f(t_i, y_i) + \frac{h}{2}f(t_{i+1}, \tilde{\eta}_1)\,,$$

wobei nun $\tilde{\eta}_i = y(t_i+h)$ sein sollte. Geht man wie beim Verfahren vom Runge vor und ersetzt man

$$\tilde{\eta}_1 = y_i + hy'(t_i)\,,$$

dann ergibt sich das **Verfahren von Heun**.

Wir studieren das Verfahren von Runge etwas genauer. Bei diesem Verfahren ergibt sich die Taylorentwicklung für hinreichend glattes f unter der Voraussetzung $y_i = y(t_i)$

$$\begin{aligned} y_{i+1} &= y_i + hf(t_i, y_i) + \frac{h^2}{2} f_t(t_i, y_i) + \frac{h^2}{2} f_y(t_i, y_i) f(t_i, y_i) + O(h^3)\,, \\ y(t_{i+1}) &= y(t_i) + hy'(t_i) + \frac{h^2}{2} y''(t_i) + O(h^3) \\ &= y_i + hf(t_i, y_i) + \frac{h^2}{2} \left(f_t(t_i, y_i) + f_y(t_i, y_i) f(t_i, y_i)\right) + O(h^3)\,. \end{aligned}$$

Damit gilt für den lokalen Fehler

$$\|y(t_{i+1}) - y_{i+1}\|_2 = O(h^3)\,.$$

Das Verfahren von Runge hat also einen kleineren lokalen Fehler als die beiden Euler-Verfahren.

Definition 7.8. Ein Einschrittverfahren hat die **(Konsistenz)-Ordnung** q, falls für jede Differentialgleichung $y' = f(t, y)$ mit $f \in C^{q+1}(I \times J)$ und jedes $t_i \in I$ für den lokalen Fehler gilt:

$$y_i = y(t_i) \in J \Rightarrow \|y_{i+1} - y(t_{i+1})\|_2 = O(h^{q+1})\,, \quad h \to 0\,.$$

Beachte: Die Ordnung ist q (und nicht $q+1$), obwohl die entsprechende h-Potenz $q+1$ ist! Wie wir später sehen werde, ist die Konvergenzordnung an einem festen Punkt $t_0 \in (0, T]$ bei einem Verfahren der Ordnung q $O(h^q)$ ist.

Beispiel 7.9. Die beiden Euler-Verfahren haben die Ordnung $q = 1$ und das Verfahren von Runge hat die Ordnung $q = 2$.

Wir studieren im folgenden den Zusammenhang zwischen Quadraturverfahren und dem Ansatz (7.13).

Satz 7.10. *Hat ein Einschrittverfahren der Form (7.13) die Ordnung q, dann hat die Quadraturformel*

$$Q[g] = \sum_{j=1}^{s} b_j g(c_j) \approx \int_0^1 g(x)\,dx$$

den Exaktheitsgrad $q-1$ – das bedeutet, dass Polynome bis zum Grad $q-1$ exakt integriert werden können.

Beweis: Für $0 \le n < q$ betrachten wir das "Anfangswertproblem"

$$y' = t^n\,, \quad y(0) = 0\,.$$

Dieses Problem hat die eindeutige Lösung $t^{n+1}/(n+1)$.

Sei $y_0 = 0$. Dann gilt nach der Definition 7.8 der Konsistenzordnung eines Einschrittverfahrens der Ordnung q die Abschätzung

$$|y(h) - y_1| = \left| \frac{1}{n+1} h^{n+1} - h \sum_{j=1}^{s} b_j (c_j h)^n \right| = O(h^{q+1})\,.$$

Diese Abschätzung gilt unabhängig von der Wahl von η_j in (7.13).

Dividiert man die obige Gleichheit durch h^{n+1}, so erhält man

$$\left| \frac{1}{n+1} - \sum_{j=1}^{s} b_j (c_j)^n \right| = O(h^{q-n}) = o(1)\,,$$

woraus durch Grenzübergang $h \to 0$ folgt

$$Q[t^n] = \sum_{j=1}^{s} b_j c_j^n = \frac{1}{n+1} = \int_0^1 t^n\,dt\,.$$

Damit ist die Quadraturformel $Q[.]$ exakt für alle Monome $t^n, n = 0, ..., q-1$, und damit für den ganzen Unterraum der Polynome vom maximalen Grad $q-1$. □

Aus diesem Satz ergibt sich unter Verwendung von allgemeinen Resultaten über Gauß-Quadraturformeln, dass ein s-stufiges Einschrittverfahren maximal die Ordnung $q = 2s$ haben kann.

Der Zusammenhang zwischen der Ordnung eines Einschrittverfahrens und dem Exaktheitsgrad einer Quadraturformel lässt sich gezielt weiterverfolgen, um Verfahren höherer Ordnung zu konstruieren. Das ist die Idee der **Runge-Kutta Verfahren**. Dabei gilt es allerdings, noch eine Regel für die Wahl der Koeffizienten η_j anzugeben. Wegen

$$\eta_j \approx y(t_i+c_jh) = y(t_i)+\int_{t_i}^{t_i+c_jh} y'(t)\,dt \;= y(t_i)+\int_{t_i}^{t_i+c_jh} f(t,y(t))\,dt \quad (7.15)$$

bietet sich hier wieder eine Quadraturformel an. Um zusätzliche Funktionsauswertungen $f(t,y)$ zu vermeiden, beschränken man sich dabei auf die gleichen Werte $f(t_i+c_jh,\eta_j)$, $j=1,...,s$, wie für die Berechnung von y_{i+1}. Das ergibt folgenden Ansatz:

$$\eta_j = y_i + h\sum_{k=1}^{s} a_{jk} f(t_i+c_kh,\eta_k)\,, \quad \sum_{k=1}^{s} a_{jk} = c_j\,. \qquad (7.16)$$

Falls $a_{jk}=0$ für $j\le k$ ist die Rechenvorschrift *explizit* und führt auf ein **explizites Runge-Kutta** Verfahren. Ansonsten ergibt sich ein **implizites Runge-Kutta** Verfahren. Die Bedingung $\sum_{k=1}^{s} a_{jk} = c_j$ ist aus Quadraturformelmethoden motiviert, wird aber in der Literatur nicht einheitlich verwendet. Es ist eine unwesentliche, aber sehr nützliche Voraussetzung.

Üblicherweise werden die Koeffizienten $\{a_{jk},b_j,c_j\}$ in einem quadratischen Tableau zusammengefasst (das so genannte **Runge-Kutta Abc**),

$$\begin{array}{c|c} \mathrm{c} & \mathrm{A} \\ \hline & \mathrm{b}^t \end{array} \;=\; \begin{array}{c|ccccc} c_1 & a_{1,1} & ... & ... & ... & a_{1,s} \\ c_2 & a_{2,1} & a_{2,2} & ... & ... & ... \\ c_3 & a_{3,1} & a_{3,2} & ... & ... & ... \\ ... & ... & ... & ... & ... & ... \\ c_s & a_{s,1} & ... & ... & a_{s,s-1} & a_{s,s} \\ \hline & b_1 & b_2 & ... & b_{s-1} & b_s \end{array}$$

wobei wir $A=[a_{j,k}]\in\mathbb{R}^{s\times s}$, $b=[b_1,...,b_s]^t\in\mathbb{R}^s$ und $c=[c_1,...,c_s]^t\in\mathbb{R}^s$ gesetzt haben. Wir sprechen ab nun von dem Runge-Kutta Verfahren (A,b,c).

Beispiel 7.11. Für das explizite und implizite Euler Verfahren ergeben sich folgende Tableaus:

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array} \qquad \begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}$$

Das Verfahren von Runge (hier in zwei Stufen aufgeschrieben)

$$\begin{aligned} \eta_1 &= y_i + \tfrac{h}{2} f(t_i, y_i) \\ y_{i+1} &= y_i + h f(t_i + h/2, \eta_1) \end{aligned}$$

passt auf den ersten Blick nicht in das Runge-Kutta Schema. Man beachte, dass in der Definition von η_1 auf der rechten Seite kein η auftaucht. Das heißt, um das Verfahren von Runge in ein Schema zu pressen, ergänzt man das Runge-Kutta Schema um den Index Null und setzt $c_0 = 0$ und $\eta_0 = y_i$, sodass

$$\eta_j = y_i + h \sum_{k=0}^{1} a_{jk} f(t_i + c_k h, \eta_k)\,, \quad \sum_{k=0}^{1} a_{jk} = c_j \text{ for } j = 0, 1\,. \tag{7.17}$$

Setzt man nun noch konkret $c_1 = 1/2$, so ergibt sich

$$\begin{aligned} \eta_0 &= y_i + h \textstyle\sum_{k=0}^{1} 0 \cdot f(t_i + c_k h, \eta_k) \\ \eta_1 &= y_i + \tfrac{h}{2} f(t_i + h/2, \eta_0) \\ y_{i+1} &= y_i + h f(t_i + h/2, \eta_1)\,. \end{aligned}$$

Diese drei Gleichungen werden mit folgendem Runge-Kutta Tableau beschrieben

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1/2 & 1/2 & 0 \\ \hline & 0 & 1 \end{array}$$

Wir konstruieren nun ein Verfahren der Ordnung 3 und leiten uns dazu Bedingungen für die Parameter des Runge-Kutta Schemas her.

Satz 7.12. *Runge-Kutta Verfahren haben mindestens die Ordnung 1. Ein Runge-Kutta Verfahren (7.13), (7.16) ist von zweiter Ordnung, wenn*

$$\sum_{j=1}^{s} b_j c_j = \frac{1}{2} \tag{7.18}$$

Es ist von dritter Ordnung, wenn zusätzlich

$$\sum_{j=1}^{s} b_j c_j^2 = \frac{1}{3} \text{ und } \sum_{j=1}^{s} b_j \sum_{k=1}^{s} a_{jk} c_k = \frac{1}{6}\,. \tag{7.19}$$

Man überprüft sofort, dass beim Verfahren von Runge (7.14) die Bedingung (7.18) erfüllt ist, nicht aber die beiden Bedingungen in (7.19).

Beispiel 7.13. Wann immer in der Literatur von *dem* Runge-Kutta Verfahren gesprochen wird, dann ist das folgende Verfahren von **Kutta** (1901) auf der Basis der Simpson-Formel gemeint. Die Koeffizienten dieses Verfahrens lauten:

$$c_1 = 0\,, \quad c_2 = 1/2\,, \quad c_3 = 1/2, \quad c_4 = 1$$

mit den Gewichten

$$b_1 = 1/6\,, \quad b_2 = 1/3\,, \quad b_3 = 1/3, \quad b_4 = 1/6\,.$$

Wegen des Exaktheitsgrad $q = 3$ der Simpson-Formel sind die Bedingungen (7.18) und die erste von (7.19) automatisch erfüllt. Berücksichtigt man, dass das Verfahren explizit ist, so reduziert sich die verbleibende Ordnungsbedingung in (7.19) zu

$$\frac{1}{6}a_{32} + \frac{1}{12}a_{42} + \frac{1}{12}a_{43} = \frac{1}{6}$$

so dass die Bestimmung der $\{a_{jk}\}$ in dieser Weise unterbestimmt ist. Erweitert man Satz 7.12, dann ergibt sich eine eindeutige Lösung aller Ordnungsbedingungen für ein explizites Verfahren vierter Ordnung, die im folgenden Tableau dargestellt ist:

0				
1/2	1/2			
1/2	0	1/2		
1	0	0	1	
	1/6	1/3	1/3	1/6

Literaturverzeichnis

[1] J.-P. Demailly. *Gewöhnliche Differentialgleichungen. Theoretische und numerische Aspekte*. Vieweg, Wiesbaden, 1994. Aus d. Franz. uebers. von Mathias Heckele.

[2] P. Deuflhard and A. Hohmann. *Numerische Mathematik I. Eine algorithmisch orientierte Einführung*. De Gruyter, Berlin, 1993. 2., überarb. Aufl.

[3] G. H. Golub and J. M. Ortega. *Wissenschaftliches Rechnen und Differentialgleichungen*. Berliner Studienreihe zur Mathematik. Heldermann Verlag, Berlin, 1995.

[4] G. H. Golub and Ch. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, 1996. third edition.

[5] R. D. Grigorieff. *Numerik gewöhnlicher Differentialgleichungen, Band 1: Einschrittverfahren*. Teubner, Stuttgart, 1972.

[6] G. Haemmerlin and K.-H. Hoffmann. *Numerische Mathematik*. Springer Verlag, Berlin, Heidelberg, New York, fourth edition, 1994.

[7] M. Hanke. *Grundlagen der Numerischen Mathematik und des Wissenschaftlichen Rechnens*. Teubner, Stuttgart, Leipzig, Wiesbaden, 2002.

[8] H. Heuser. *Gewöhnliche Differentialgleichungen*. Teubner, Stuttgart, second edition, 1991.

[9] N. J. Higham. *Accuracy and stability of numerical algorithms*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1996.

[10] A. Quarteroni, R. Sacco, and F. Saleri. *Numerical Mathematics*. Springer Verlag, Berlin, 2000.

[11] H. R. Schwarz. *Numerische Mathematik*. B. G. Teubner, Stuttgart, fourth edition, 1997. With a contribution by Jörg Waldvogel.

[12] J. Stoer. *Numerische Mathematik 1*. Springer Verlag, Berlin, 1999.

[13] W. Walter. *Gewöhnliche Differentialgleichungen*. Springer Verlag, Berlin, 6., berarb. u. erw. aufl. edition, 1996.

Abbildungsverzeichnis