

1 Lineare Gleichungssysteme I

1.1 Problemstellung

Seien $A \in \mathbb{C}^{m \times n}$ und $b \in \mathbb{C}^m$ gegeben und alle $x \in \mathbb{C}^n$ mit der Eigenschaft $Ax = b$ gesucht. $Ax = b$ heißt dann ein lineares Gleichungssystem mit m Gleichungen und n Unbekannten. Es heißt unterbestimmt, falls $m < n$, und überbestimmt, falls $m > n$. Es heißt homogen, falls $b = 0$, und inhomogen, falls $b \neq 0$.

Wann besitzt $Ax = b$ eine Lösung? Wann ist eine solche Lösung eindeutig? Aus der linearen Algebra wissen wir: $Ax = b$ ist genau dann lösbar, wenn $\text{Rang}(A) = \text{Rang}(A, b)$. Für den Spezialfall $m = n$ gilt: Die Aussagen $Ax = b$ ist eindeutig lösbar, $\text{Rang}(A) = n$, A ist regulär und $\det A \neq 0$ sind äquivalent.

Die Fragen für $m = n$ sind: Wie erkennt man, ob A regulär ist? Wie kann man dann die Lösung x bestimmen? Wie kann man bei singulärem A feststellen, ob $Ax = b$ eine Lösung hat oder nicht?

Im Fall $n = 2$ gilt

$$x = \left(\frac{a_{22}b_1 - a_{12}b_2}{a_{11}a_{22} - a_{12}a_{21}}, \frac{-a_{21}b_1 + a_{11}b_2}{a_{11}a_{22} - a_{12}a_{21}} \right)^T,$$

sofern $\det A = a_{11}a_{22} - a_{12}a_{21} \neq 0$. Was ist, wenn $\det A = 0$? Was ist, wenn $n > 2$?

1.2 Gestaffelte Gleichungssysteme, Dreiecksmatrizen

Das lineare Gleichungssystem

$$\begin{aligned} 3x_1 + x_2 + 2x_3 &= 66 \\ 2x_2 + 4x_3 &= 84 \\ 5x_3 &= 75 \end{aligned}$$

ist ein Beispiel für ein gestaffeltes Gleichungssystem. Wir lösen von unten nach oben: Die letzte Zeile liefert $x_3 = 25$, Einsetzen in die vorletzte Zeile $x_2 = 12$ und in die erste Zeile $x_1 = 8$.

In Matrixform ergibt sich

$$Ax = \begin{pmatrix} 3 & 1 & 2 \\ 0 & 2 & 4 \\ 0 & 0 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 66 \\ 84 \\ 75 \end{pmatrix} = b$$

mit $\det A = 3 \cdot 2 \cdot 5 = 30 \neq 0$.

Eine Matrix $A = (a_{ik}) \in \mathbb{C}^{m \times n}$ heißt rechte obere Dreiecksmatrix, falls $a_{ik} = 0$ für $i > k$, und linke untere Dreiecksmatrix, falls $a_{ik} = 0$ für $i < k$. Ein lineares Gleichungssystem mit Dreiecksmatrix heißt gestaffeltes Gleichungssystem.

Ein lineares Gleichungssystem mit rechter oberer Dreiecksmatrix A , d. h.

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\ a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{nn}x_n &= b_n, \end{aligned}$$

für das $a_{ii} \neq 0$ für $i = 1, \dots, n$ gilt, ist eindeutig lösbar, da $\det A = \prod_{i=1}^n a_{ii} \neq 0$. Ferner kann der Lösungsvektor x direkt durch Rückwärtseinsetzen/Rückwärtssubstitution

$$x_i = \frac{1}{a_{ii}} \left(b_i - \sum_{k=i+1}^n a_{ik} x_k \right), \quad i = n-1, \dots, 1$$

berechnet werden.

Bei linken unteren Dreiecksmatrizen funktioniert ein analoges Verfahren, das man Vorwärtseinsetzen/Vorwärtssubstitution nennt.

Wenn man eine reguläre Matrix A in ein Produkt zweier Dreiecksmatrizen zerlegen kann, dann ist das Gleichungssystem ganz einfach zu lösen. Sei also $A = LR$ mit regulärer linker unterer Dreiecksmatrix L und rechter oberer Dreiecksmatrix R , so löse $Ax = b$ folgendermaßen: Löse $Ly = b$ nach y durch Vorwärtseinsetzen und schließlich $Rx = y$ nach x durch Rückwärtseinsetzen. Dann gilt $b = Ly = LRx = Ax$.

1.3 Gauß-Elimination

Prinzip der Gauß-Elimination: Überführe ein lineares Gleichungssystem in ein Gleichungssystem mit Dreiecksmatrix mit Hilfe elementarer Umformungen.

Folgende elementare Umformungen lassen die Lösungsmenge von $Ax = b$ unverändert: Vertauschen zweier Gleichungen; Multiplikation einer Zeile mit $\lambda \neq 0$; Addition des q -fachen einer Zeile zu einer anderen Zeile ($q \in \mathbb{C}$); Vertauschen zweier Spalten in A , wenn die entsprechenden Komponenten in x mitvertauscht werden.

Beispiel:

$$\begin{array}{ccc|c} x_1 & x_2 & x_3 & b \\ \hline 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 1/2 \\ 1 & 3 & 9 & 1/3 \end{array} \Rightarrow \begin{array}{ccc|c} x_1 & x_2 & x_3 & b^{(1)} \\ \hline 1 & 1 & 1 & 1 \\ 0 & 1 & 3 & -1/2 \\ 0 & 2 & 8 & -2/3 \end{array} \Rightarrow \begin{array}{ccc|c} x_1 & x_2 & x_3 & b^{(2)} \\ \hline 1 & 1 & 1 & 1 \\ 0 & 1 & 3 & -1/2 \\ 0 & 0 & 2 & 1/3 \end{array}.$$

Damit folgt $x_3 = 1/6$, $x_2 = -1$ und $x_1 = 11/6$.

Allgemeiner Fall:

Schritt 1: Erzeuge Nullen unterhalb von a_{11} . Addiere dazu das $(-a_{i1}/a_{11})$ -fache der 1. Zeile zur i -ten Zeile, $i = 2, \dots, n$. (Ist $a_{11} = 0$, so suche ein j mit $a_{j1} \neq 0$ und vertausche die 1. Zeile mit der j -ten Zeile. Gibt es kein solches j , so ist A singulär, und ein Spaltentausch ist notwendig.) Es entsteht ein neues Gleichungssystem $A^{(1)}x = b^{(1)}$

$$\begin{array}{cccc|c} x_1 & x_2 & \cdots & x_n & b^{(1)} \\ \hline a_{11}^{(0)} & a_{12}^{(0)} & \cdots & a_{1n}^{(0)} & b_1^{(0)} \\ 0 & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} & b_2^{(1)} \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & a_{n2}^{(1)} & \cdots & a_{nn}^{(1)} & b_n^{(1)} \end{array}$$

mit der Restmatrix $(a_{ik}^{(1)})$ für $2 \leq i, k \leq n$.

Schritt 2: Wende Schritt 1 auf die entstandene $(n-1) \times (n-1)$ -Restmatrix an. Wiederhole ihn, bis nach insgesamt r Eliminationsschritten eine Restmatrix $(a_{ik}^{(r)})$ für $r+1 \leq i, k \leq n$ entstanden ist, so dass:

- Es ist $r = n - 1$: Dann ist A^{n-1} auf Dreiecksgestalt, also Algorithmus erfolgreich.
- Es ist $r < n - 1$ mit $a_{j,r+1}^{(r)} \neq 0$ für ein $j = r + 1, \dots, n$: Tausche zwei geeignete Zeilen und führe Schritt 1 aus.
- Es ist $r < n - 1$ mit $a_{j,r+1}^{(r)} = 0$ für alle $j = r + 1, \dots, n$: Tausche zwei geeignete Spalten (und evtl. Zeilen) und führe Schritt 1 aus. Ist die Restmatrix die Nullmatrix, siehe später.

Das Element $a_{r+1,r+1}^{(r)}$, welches ungleich 0 sein muss, heißt Pivotelement, analog Pivotzeile und -spalte. Mögliche Strategien zur Pivotwahl sind:

- Kanonische Pivotwahl: Keine Vertauschungen, daher Abbruch selbst bei regulärer Matrix möglich, z. B. bei $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$
- Spaltenpivotsuche: Bestimme als Pivotelement das betragsgrößte Element in der Spalte $r + 1$ der Restmatrix, daher eventuell Zeilentausch notwendig, aber Abbruch nur bei singulärer Matrix.
- Totalpivotsuche: Bestimme als Pivotelement das betragsgrößte Element in der Restmatrix, daher eventuell Zeilen- oder Spaltentausch notwendig, aber Abbruch nur, wenn die Restmatrix die Nullmatrix ist.

Wenn A regulär ist, dann kann die Gauß-Elimination ohne Spaltenvertauschungen durchgeführt werden. Sind dagegen vor einem Eliminationsschritt alle Einträge in der Pivotspalte 0, so ist A singulär.

Der Aufwand der Gauß-Elimination beträgt $n^3/3 + O(n^2)$ Multiplikationen und Divisionen. Wenn man zusätzlich die Einträge oberhalb der Diagonalen eliminiert, beträgt der Aufwand $n^3/2 + O(n^2)$ Multiplikationen und Divisionen.

1.4 Dreieckszerlegungen

Sei $A \in \mathbb{C}^{n \times n}$ eine Matrix, für die die Gauß-Elimination mit kanonischer Pivotwahl durchgeführt werden kann. Dann liefert die Gauß-Elimination eine Zerlegung $A = LR$ in eine linke untere Dreiecksmatrix L mit Einsdiagonale und eine rechte obere Dreiecksmatrix R . Ferner ist diese Zerlegung eindeutig bestimmt. Diese Zerlegung heißt LR -Zerlegung oder Dreieckszerlegung.

Lässt man Vertauschungen zu, so gilt: Ist A regulär, so existiert eine Permutationsmatrix P , so dass $PA = LR$. (L und R haben die Eigenschaften wie oben.) Ansonsten existieren Permutationsmatrizen P und Q , so dass $PAQ = LR$. P und Q sind im Allgemeinen nicht eindeutig bestimmt; L und R ebenfalls nicht.

Die Gauß-Elimination und die LR -Zerlegung (ob mit Permutationen oder ohne) haben denselben Aufwand. Die Gauß-Elimination arbeitet jedoch sofort mit der rechten Seite. Die LR -Zerlegung ist dann effizienter, wenn das Gleichungssystem für mehrere rechte Seiten gelöst werden muss.

Sei A regulär und $A = LR$. Dann kann R in das Produkt DU zerlegt werden, worin D eine Diagonalmatrix und U eine rechte obere Dreiecksmatrix (also wie R), aber zusätzlich mit Einsdiagonale ist. Diese Zerlegung heißt LDU -Zerlegung. Sie kann analog mit Spaltenpivotsuche für $PA = LR = LDU$ formuliert werden.

Sei $A = (a_{ik}) \in \mathbb{C}^{n \times n}$. Dann heißt $A_r := (a_{ik})_{1 \leq i, k \leq r} \in \mathbb{C}^{r \times r}$ die Hauptuntermatrix der Ordnung r von A und $\det A_r$ der Hauptminor der Ordnung r von A . Die Gauß-Elimination

ist genau dann mit kanonischer Pivotwahl für A durchführbar, wenn $\det A_r \neq 0$ für alle $r = 1, \dots, n-1$. Dieser Satz ist für die Numerik nicht besonders hilfreich, weil die Forderung $\det A_r \neq 0$ nicht effizient getestet werden kann. Außerdem ist die Spaltenpivotsuche auch bei solchen Matrizen numerisch „besser“.

1.5 Matrizen spezieller Struktur

Eine Matrix $A = (a_{ik}) \in \mathbb{C}^{n \times n}$ heißt strikt diagonaldominant, wenn

$$\sum_{k=1, k \neq i}^n |a_{ik}| < |a_{ii}| \quad \forall i = 1, \dots, n.$$

Dann ist A regulär, und die Gauß-Elimination kann mit kanonischer Pivotwahl durchgeführt werden, wobei alle Restmatrizen $A^{(r)}$ strikt diagonaldominant sind.

Eine symmetrische (hermitesche) Matrix $A = (a_{ik}) \in \mathbb{R}^{n \times n}$ ($A = (a_{ik}) \in \mathbb{C}^{n \times n}$) heißt positiv definit, wenn

$$x^T A x > 0 \quad \forall x \in \mathbb{R}^n \setminus \{0\} \quad (x^* A x > 0 \quad \forall x \in \mathbb{C}^n \setminus \{0\}).$$

Sie heißt positiv semidefinit, wenn auch Gleichheit herrschen darf. Positiv definite Matrizen sind regulär.

$A \in \mathbb{C}^{n \times n}$ ist genau dann positiv definit, wenn die Hauptuntermatrizen A_r für alle $r = 1, \dots, n$ positiv definit sind. Ferner folgt aus der positiven Definitheit, dass $a_{ii} > 0$ für alle $i = 1, \dots, n$ und dass die Restmatrizen $A^{(r)}$ für alle $r = 1, \dots, n-1$ positiv definit sind. Insbesondere ist der Gauß-Algorithmus mit kanonischer Pivotwahl durchführbar.

Auch wenn bei strikt diagonaldominanten und positiv definiten Matrizen die Gauß-Elimination mit kanonischer Pivotwahl möglich ist, heißt das nicht, dass sie numerisch günstig ist.

Für eine symmetrische (hermitesche) Matrix $A \in \mathbb{R}^{n \times n}$ ($A \in \mathbb{C}^{n \times n}$) gilt: A ist genau dann positiv definit, wenn sie eine LDU -Zerlegung mit $d_{ii} > 0$ für alle $i = 1, \dots, n$ besitzt. Insbesondere gilt dann $U = L^T$ ($U = L^*$). Außerdem ist A ist genau dann positiv definit, wenn $\det A_r > 0$ für alle $r = 1, \dots, n$, d. h. alle Hauptminoren sind positiv.

Für eine Matrix $A \in \mathbb{C}^{n \times n}$ gilt: A ist genau dann positiv definit, wenn eine rechte obere Dreiecksmatrix C mit positiven Diagonaleinträgen existiert, so dass $A = C^* C$. Diese Zerlegung ist eindeutig und heißt Cholesky-Zerlegung von A .

Existiert eine Cholesky-Zerlegung $A = C^* C$, so gilt für die LDU -Zerlegung $A = LDL^*$ mit positiven Diagonaleinträgen von D . Der Zusammenhang ist durch $C^* = L\sqrt{D}$ gegeben.

Die Cholesky-Zerlegung kann direkt berechnet werden: Aus $A = C^* C$ folgt

$$a_{ik} = \sum_{j=1}^n \bar{c}_{ji} c_{jk} \quad \forall i, k = 1, \dots, n.$$

Daraus folgen sofort die Formeln

$$c_{ii} = \left(a_{ii} - \sum_{j=1}^{i-1} |c_{ji}|^2 \right)^{1/2}, \quad c_{ik} = \frac{1}{c_{ii}} \left(a_{ik} - \sum_{j=1}^{k-1} \bar{c}_{ji} c_{jk} \right) \quad \forall i, k = 1, \dots, n.$$

Die Einträge von C können mit diesen Formeln in der Reihenfolge $c_{11}, c_{12}, \dots, c_{1n}, c_{22}, c_{23}, \dots, c_{2n}, \dots, c_{nn}$ bestimmt werden.

Sei $A \in \mathbb{R}^{n \times n}$ ($A \in \mathbb{C}^{n \times n}$) eine symmetrische (hermitesche) Matrix, für die die Gauß-Elimination mit kanonischer Pivotwahl möglich ist. Dann sind alle Restmatrizen symmetrisch (hermitesch), und in diesem Fall halbieren sich Speicher- und Zeitaufwand etwa.

Eine Matrix $A \in \mathbb{C}^{n \times n}$ heißt eine Dreibandmatrix, wenn $a_{ik} = 0$ für alle $i, k = 1, \dots, n$ mit $|i - k| > 1$. Für eine solche Matrix sei die Gauß-Elimination mit kanonischer Pivotwahl möglich. Dann alle Restmatrizen sowie die Faktoren L und R Dreibandmatrizen. Die inverse Matrix A^{-1} ist im Allgemeinen jedoch vollbesetzt.

1.6 Zusammenfassung

Wir betrachten ausschließlich lineare Gleichungssysteme mit quadratischer, regulärer Matrix über \mathbb{R} oder \mathbb{C} . Diese sind immer eindeutig lösbar.

Gleichungssysteme mit linker unterer oder rechter oberer Dreiecksmatrix heißen gestaffelt. Sie können leicht durch Vorwärts- oder Rückwärtseinsetzen gelöst werden. Die Idee der Dreieckszerlegungen ist es, das Lösen eines Gleichungssystems auf das Lösen zweier gestaffelter zurückzuführen.

Eine LR -Zerlegung von A nennen wir das Paar (L, R) mit $A = LR$, so dass L eine linke untere Dreiecksmatrix mit Einsdiagonale und R eine reguläre, rechte obere Dreiecksmatrix ist. Besitzt A eine LR -Zerlegung, so ist diese eindeutig bestimmt. Sie existiert genau dann nicht, wenn im Algorithmus ein Pivotelement verschwindet. In diesem Fall kann man eine Spaltenpivotsuche durchführen, die grundsätzlich numerisch besser ist.

Eine P - LR -Zerlegung von A nennen wir das Tupel (L, R, P) mit $PA = LR$, so dass L eine linke untere Dreiecksmatrix mit Einsdiagonale, R eine reguläre, rechte obere Dreiecksmatrix und P eine Permutationsmatrix ist. Sie existiert genau dann, wenn A regulär ist. Sie ist eindeutig, wenn man im Algorithmus festlegt, wie man bei gleich großen Pivotelementen verfährt.

Eine LDU - bzw. P - LDU -Zerlegung von A nennen wir das Tupel (L, D, U) bzw. (L, D, U, P) mit $A = LDU$ bzw. $PA = LDU$, so dass L eine linke untere Dreiecksmatrix mit Einsdiagonale, D eine reguläre Diagonalmatrix, R eine rechte obere Dreiecksmatrix mit Einsdiagonale und P eine Permutationsmatrix ist. Der Zusammenhang zur LR - bzw. P - LR -Zerlegung ist einfach durch $R = DU$ gegeben.

Ist A symmetrisch (hermitesch), so gilt $U = L^T$ ($U = L^*$) für die LDU -Zerlegung. Ist PA symmetrisch (hermitesch), so gilt $U = L^T$ ($U = L^*$) für die P - LDU -Zerlegung. Ist A sogar positiv definit, so existiert eine LR - und LDU -Zerlegung, es gilt $U = L^T$ ($U = L^*$) wie zuvor, und die Diagonaleinträge von D sind positiv (insbesondere reell). Genau dann existiert auch eine Cholesky-Zerlegung $A = C^*C$ mit einer rechten oberen Dreiecksmatrix mit positiven Diagonaleinträgen. Es gilt $C^* = L\sqrt{D}$. Die Cholesky-Zerlegung wird üblicherweise allerdings direkt mit den Formeln berechnet.

Hat A eine gewisse Bandstruktur, so besitzen auch die Dreiecksmatrizen L , R und U sowie der Cholesky-Faktor C diese Bandstruktur. Das ist nützlich zur Speicherung der Matrizen. Allerdings müssen die Faktoren nicht schwach besetzt sein, nur weil A schwach besetzt ist.

2 Fehlertheorie

2.1 Fehlerarten

- Problembedingte Fehler: Idealisierungsfehler: Das Modell ist bereits nur eine Näherung.
- Datenfehler: Die Eingabedaten sind durch Messfehler ungenau. Wie wirken sie auf die

Lösung? Das hängt von der Kondition des Problems ab.

- Durch numerisches Rechnen bedingte Fehler: Diskretisierungsfehler: Kontinuierliche werden durch diskrete Größen ersetzt. Abbruchfehler: Iterationen werden nach endlich vielen Schritten abgebrochen. Wie groß ist der Fehler? Rundungsfehler: Der Computer rechnet nur mit endlich vielen Stellen. Wie pflanzen sich die Fehler fort? Das hängt von der Stabilität ab.

Eine Faustregel ist: Das Ergebnis wird brauchbar sein, wenn das Problem gut konditioniert ist und der Algorithmus stabil ist. Je schlechter das Problem konditioniert ist, desto ausgefeilter muss der Algorithmus gewählt werden, um eine brauchbare Lösung zu liefern.

Ist das Problem schlecht konditioniert: Formuliere es in ein äquivalentes Problem um, das besser konditioniert ist. Ist der Algorithmus nicht hinreichend stabil: Formuliere die Rechenschritte um oder wähle einen anderen Algorithmus.

2.2 Absolute und relative Fehler, Normen

Seien $x \in \mathbb{C}$ eine exakte Größe und $x + \Delta x \in \mathbb{C}$ eine Näherung für x . Dann heißen $|\Delta x|$ der absolute Fehler der Näherung und $|\Delta x|/|x|$ der relative Fehler der Näherung (falls $x \neq 0$).

Für einen \mathbb{C} -Vektorraum X heißt eine Abbildung $\|\cdot\|: X \rightarrow [0, \infty)$ eine Norm auf X , wenn $\|x\| = 0 \iff x = 0$ (Definitheit), $\|\alpha x\| = |\alpha| \|x\|$ (Homogenität) und $\|x + y\| \leq \|x\| + \|y\|$ (Dreiecksungleichung) für alle $x, y \in X$ und $\alpha \in \mathbb{C}$ gilt.

Für $X = \mathbb{C}^n$ sind die Eins-Norm, die Euklid-Norm und die Maximumnorm gegeben durch

$$\|x\|_1 := \sum_{i=1}^n |x_i|, \quad \|x\|_2 := \left(\sum_{i=1}^n |x_i|^2 \right)^{1/2}, \quad \|x\|_\infty := \max\{|x_i| \mid i = 1, \dots, n\}.$$

Allgemeiner ist für jedes $p \in [1, \infty)$ die Abbildung $x \mapsto (\sum_{i=1}^n |x_i|^p)^{1/p}$ eine Norm.

Der Vektorraum $X = \mathbb{C}^{m \times n}$ besteht aus den $m \times n$ -Matrizen, die wir als lineare Abbildungen auffassen. Dann ist der Definitionsbereich $X_1 = \mathbb{C}^n$ und der Zielbereich $X_2 = \mathbb{C}^m$. Versehen wir X_1 und X_2 mit Normen $\|\cdot\|_{(1)}$ und $\|\cdot\|_{(2)}$, so definieren wir die von diesen Normen induzierte Matrixnorm (Abbildungsnorm) durch

$$\text{lub}_{(1,2)}(A) := \|A\|_{(1,2)} := \sup\{\|Ax\|_{(2)} \mid \|x\|_{(1)} = 1\} = \sup\left\{ \frac{\|Ax\|_{(2)}}{\|x\|_{(1)}} \mid x \neq 0 \right\}.$$

Seien $\|\cdot\|$, $\|\cdot\|_{(1)}$ und $\|\cdot\|_{(2)}$ jeweils beliebige Normen auf $\mathbb{C}^{m \times n}$, \mathbb{C}^n und \mathbb{C}^m . Dann heißt die Matrixnorm $\|\cdot\|$ verträglich mit dem Vektornormenpaar $(\|\cdot\|_{(1)}, \|\cdot\|_{(2)})$, falls

$$\|Ax\|_{(2)} \leq \|A\| \|x\|_{(1)} \quad \forall A \in \mathbb{C}^{m \times n} \quad \forall x \in \mathbb{C}^n.$$

Es gilt: Die vom Paar $(\|\cdot\|_{(1)}, \|\cdot\|_{(2)})$ induzierte Norm $\|\cdot\|_{(1,2)}$ ist verträglich mit diesem Paar.

Sei $\|\cdot\|$ eine beliebige Norm auf $\mathbb{C}^{m \times n}$. Sie heißt submultiplikativ, falls

$$\|AB\| \leq \|A\| \|B\| \quad \forall A \in \mathbb{C}^{m \times p} \quad \forall B \in \mathbb{C}^{p \times n}.$$

Es gilt: Jede induzierte Norm ist submultiplikativ.

Wir definieren durch

$$\|A\|_1 := \max_{k=1}^n \sum_{i=1}^m |a_{ik}|, \quad \|A\|_2 := \sqrt{\varrho(A^*A)}, \quad \|A\|_\infty := \max_{i=1}^m \sum_{k=1}^n |a_{ik}|$$

die Spaltensummennorm, Spektralnorm und Zeilensummennorm, wobei

$$\varrho(B) := \max\{|\lambda| \mid \lambda \in \mathbb{C} \text{ ist Eigenwert von } B\}$$

der Spektralradius der quadratischen Matrix B ist. Die drei Normen sind induziert von den Vektornormenpaaren $(\|\cdot\|_1, \|\cdot\|_1)$, $(\|\cdot\|_2, \|\cdot\|_2)$ und $(\|\cdot\|_\infty, \|\cdot\|_\infty)$, was die Schreibweisen rechtfertigt.

Normäquivalenz (vgl. AmV): Auf jedem endlich-dimensionalen Vektorraum X sind je zwei Normen $\|\cdot\|_{(1)}$ und $\|\cdot\|_{(2)}$ äquivalent, d. h. es existieren Konstanten $C_*, C^* > 0$ mit

$$C_* \|x\|_{(1)} \leq \|x\|_{(2)} \leq C^* \|x\|_{(1)} \quad \forall x \in X.$$

Damit folgt, dass die Normen $\|\cdot\|_1, \|\cdot\|_2$ und $\|\cdot\|_\infty$ auf dem \mathbb{C}^n äquivalent sind, und es gilt

$$\begin{aligned} \|x\|_2 \leq \|x\|_1 \leq \sqrt{n} \|x\|_2, & \quad \frac{1}{\sqrt{n}} \|x\|_1 \leq \|x\|_2 \leq \|x\|_1, & \quad \frac{1}{n} \|x\|_1 \leq \|x\|_\infty \leq \|x\|_1, \\ \|x\|_\infty \leq \|x\|_1 \leq n \|x\|_\infty, & \quad \|x\|_\infty \leq \|x\|_2 \leq \sqrt{n} \|x\|_\infty, & \quad \frac{1}{\sqrt{n}} \|x\|_2 \leq \|x\|_\infty \leq \|x\|_2. \end{aligned}$$

Alle Konstanten sind scharf.

Ebenso folgt, dass alle Normen auf $\mathbb{C}^{m \times n}$ äquivalent sind. Sind insbesondere $\|\cdot\|_{(1,1)}$ und $\|\cdot\|_{(2,2)}$ zwei Matrixnormen, die von den Vektornormenpaaren $(\|\cdot\|_{(1)}, \|\cdot\|_{(1)})$ und $(\|\cdot\|_{(2)}, \|\cdot\|_{(2)})$ induziert sind, für die $C_* \|x\|_{(1)} \leq \|x\|_{(2)} \leq C^* \|x\|_{(1)}$ scharf ist, so gilt

$$\|A\|_{(2,2)} = \sup_{x \neq 0} \frac{\|Ax\|_{(2)}}{\|x\|_{(2)}} \leq \sup_{x \neq 0} \frac{C^* \|Ax\|_{(1)}}{C_* \|x\|_{(1)}} = \frac{C^*}{C_*} \|A\|_{(1,1)}.$$

Auch diese Abschätzung ist scharf.

Seien $\|\cdot\|_{(1)}$ und $\|\cdot\|_{(2)}$ zwei beliebige Vektornormen, $\|\cdot\|_{(1,2)}$ ihre induzierte Matrixnorm und $\|\cdot\|$ irgendeine Matrixnorm, die mit dem Vektornormenpaar verträglich ist. Dann gilt

$$\|A\| \geq \|A\|_{(1,2)} \quad \forall A \in \mathbb{C}^{m \times n},$$

d. h. die induzierte Norm ist die kleinste unter allen verträglichen Normen.

Durch

$$\|A\|_F := \left(\sum_{i=1}^m \sum_{k=1}^n |a_{ik}|^2 \right)^{1/2} = \sqrt{\text{Spur}(A^*A)}$$

wird eine submultiplikative und mit der Euklidnorm verträgliche Matrixnorm definiert, die sog. Frobeniusnorm. Mit dem vorigen Satz gilt dann $\|A\|_2 \leq \|A\|_F$.

Störungssatz: Seien $A, B \in \mathbb{C}^{n \times n}$, A regulär, $\|\cdot\|$ eine submultiplikative Matrixnorm und $\|A^{-1}B\| < 1$. Dann ist $A + B$ regulär, und für die Inverse gilt

$$\|(A + B)^{-1}\| \leq \frac{\|A^{-1}\|}{1 - \|A^{-1}B\|}.$$

Hierbei ist B eine Störung von A , und wenn B hinreichend klein ist, ist auch die gestörte Matrix $A + B$ noch invertierbar. Die Menge der invertierbaren Matrizen ist also offen in $\mathbb{C}^{n \times n}$, was nicht von der Wahl der Norm abhängt.

Ist $A \in \mathbb{C}^{n \times n}$ mit $\varrho(A) < 1$, so ist $I - A$ regulär, und die Inverse kann als Neumannsche Reihe in der Form

$$(I - A)^{-1} = \sum_{k=0}^{\infty} A^k$$

geschrieben werden. Diese Reihe ist absolut konvergent.

2.3 Kondition eines Problems

Seien $f: [a, b] \rightarrow \mathbb{R}$ stetig differenzierbar und $x, x + \Delta x \in (a, b)$. Ist K ein kompaktes Teilintervall von (a, b) , das x und $x + \Delta x$ enthält, dann ist $f'|_K$ beschränkt. Es gilt also

$$|f(x + \Delta x)| \leq |f(x)| + |\max f'|_K |\Delta x|.$$

Problem für die Numerik: f' kann trotzdem beliebig groß sein.

Die Taylorformel motiviert sofort die Abschätzung $f(x + \Delta x) - f(x) \approx f'(x)\Delta x$, so dass wir definieren:

$$\begin{aligned} |\Delta x| &= \text{abs. Datenfehler}, & \frac{|\Delta x|}{|x|} &= \text{rel. Datenfehler}, \\ |f(x + \Delta x) - f(x)| &= \text{abs. Resultatsfehler}, & \frac{|f(x + \Delta x) - f(x)|}{|f(x)|} &= \text{rel. Resultatsfehler}, \\ |f'(x)| &= \text{abs. Verstärk.faktor}, & \frac{|xf'(x)|}{|f(x)|} &= \text{rel. Verstärk.faktor}. \end{aligned}$$

Die Zahlen $|f'(x)|$ bzw. $|xf'(x)/f(x)|$ sind ein Maß dafür, wie sich absolute bzw. relative Fehler in den Daten auf das Ergebnis auswirken. Sie heißen absolute bzw. relative Konditionszahlen der Aufgabe, $f(x)$ zu berechnen.

Für eine reguläre Matrix $A \in \mathbb{C}^{n \times n}$ heißt

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

die Kondition bezüglich $\|\cdot\|$. Wir schreiben $\text{cond}_1, \text{cond}_2$, usw., wenn die Kondition bezüglich der Norm $\|\cdot\|_1, \|\cdot\|_2$, usw. gemeint ist.

Es gelte $Ax = b$ mit regulärer Matrix $A \in \mathbb{C}^{n \times n}$ und rechter Seite $b \in \mathbb{C}^n \setminus \{0\}$. Beide Daten werden durch $\Delta A \in \mathbb{C}^{n \times n}$ und $\Delta b \in \mathbb{C}^n$ gestört, so dass $\text{cond}(A)\|\Delta A\|/\|A\| < 1$ ist. Dann ist das gestörte Gleichungssystem $(A + \Delta A)\xi = b + \Delta b$ eindeutig lösbar, und der relative Fehler der Lösung $\xi \in \mathbb{C}^{n \times n}$ kann gegen die relativen Datenfehler in der Form

$$\frac{\|\xi - x\|}{\|x\|} \leq \frac{\text{cond}(A)}{1 - \text{cond}(A)\frac{\|\Delta A\|}{\|A\|}} \left(\frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta b\|}{\|b\|} \right)$$

abgeschätzt werden. Wenn der Nenner fast 1 ist, dann entspricht $\text{cond}(A)$ der relativen Konditionszahl der Aufgabe, das Gleichungssystem $Ax = b$ zu lösen.

Die folgende Matrix $A_a \in \mathbb{R}^{2 \times 2}$ ist für alle $a > 0$ regulär, und es gilt

$$A_a := \begin{pmatrix} 1+a & 1 \\ 1 & 1 \end{pmatrix} \implies A_a^{-1} = \frac{1}{a} \begin{pmatrix} 1 & -1 \\ -1 & 1+a \end{pmatrix}.$$

Weiter ergibt sich $\|A_a\|_\infty = 2 + a$ und $\|A_a^{-1}\|_\infty = (2 + a)/a$, d.h. $\text{cond}_\infty(A_a) = (2 + a)^2/a$. Für $a \ll 1$ ist A_a fast singulär und entsprechend gilt $\text{cond}_\infty(A_a) \gg 1$. Für $\varepsilon > 0$ gilt weiter

$$\Delta A := \begin{pmatrix} \varepsilon & 0 \\ 0 & 0 \end{pmatrix} \implies \|\Delta A\|_\infty = \varepsilon \implies \text{cond}_\infty(A_a) \frac{\|\Delta A\|_\infty}{\|A_a\|_\infty} = \frac{(2 + a)\varepsilon}{a}.$$

Es gelte wieder $Ax = b$ mit regulärer Matrix $A \in \mathbb{C}^{n \times n}$ und rechter Seite $b \in \mathbb{C}^n \setminus \{0\}$. Für ein $\xi \in \mathbb{C}^n$ heißt $r := A\xi - b \in \mathbb{C}^n$ das Residuum zu ξ . Dann gilt

$$\frac{\|\xi - x\|}{\|x\|} \leq \text{cond}(A) \frac{\|r\|}{\|b\|}.$$

Ist ξ eine Näherung eines Lösungsalgorithmus, dann stellt ein kleines Residuum nur dann eine gute Näherung sicher, wenn $\text{cond}(A)$ nicht zu groß ist.

Seien nun $U \subset \mathbb{R}^n$ offen, $f: U \rightarrow \mathbb{R}^m$ stetig differenzierbar und $x_1, \dots, x_n \in \mathbb{R}$ gegebene Daten. Es sollen die m Ergebnisse $f_i(x_1, \dots, x_n)$ für $i = 1, \dots, m$ berechnet werden. Dann gilt mit der Taylorformel

$$\Delta y_i := f_i(x + \Delta x) - f_i(x) \approx \sum_{k=1}^n \frac{\partial f_i(x)}{\partial x_k} \Delta x_k \implies \Delta y \approx J_f(x) \Delta x.$$

Daraus folgen in erster Näherung die Abschätzungen

$$\begin{aligned} |\Delta y_i| &\leq \sum_{k=1}^n \underbrace{\left| \frac{\partial f_i(x)}{\partial x_k} \right|}_{=: \kappa_{ik}^{(abs)}(x)} |\Delta x_k|, & \|\Delta y\| &\leq \underbrace{\|J_f(x)\|}_{=: \kappa^{(abs)}(x)} \|\Delta x\|, \\ \frac{|\Delta y_i|}{|y_i|} &\leq \sum_{k=1}^n \underbrace{\left| \frac{\partial f_i(x)}{\partial x_k} \frac{x_k}{f_i(x)} \right|}_{=: \kappa_{ik}^{(rel)}(x)} \frac{|\Delta x_k|}{|x_k|}, & \frac{\|\Delta y\|}{\|y\|} &\leq \underbrace{\frac{\|J_f(x)\| \|x\|}{\|f(x)\|}}_{=: \kappa^{(rel)}(x)} \frac{\|\Delta x\|}{\|x\|}. \end{aligned}$$

Die Kappas heißen absolute und relative Konditionszahl und partielle Konditionszahlen der Aufgabe, $f(x)$ zu berechnen. Sie heißt gut konditioniert, wenn die Konditionszahl(en) in der Größenordnung von 1 liegt/en, ansonsten schlecht konditioniert.

Für die Addition zweier reeller Zahlen x_1 und x_2 gilt bezüglich $\|\cdot\|_1$

$$\kappa^{(rel)}(x) = \frac{|x_1| + |x_2|}{|x_1 + x_2|} \geq 1.$$

Es ist $\kappa^{(rel)}(x) = 1$, falls x_1 und x_2 das gleiche Vorzeichen haben. Die Addition ist ebenfalls gut konditioniert, wenn einer der Summanden betragsmäßig viel größer ist als der andere Summand. Aber:

Die Aufgabe, die Summe $x_1 + x_2$ zu berechnen, wenn $x_1 \approx -x_2$ gilt, ist extrem schlecht konditioniert. Es kommt zur katastrophalen Auslöschung führender Ziffern. Das kann sogar so weit gehen, dass keine einzige Ziffer des Ergebnisses mehr korrekt ist.

Dieses Problem gilt es immer zu vermeiden. Ein Beispiel ist die Funktion

$$f(t) = \frac{1}{1+2t} - \frac{1-t}{1+t}$$

für $0 < t \ll 1$. Für $t \rightarrow 0$ konvergieren beide Brüche gegen 1, und zwar beide von unten. Für $t = 10^{-6}$ erhalten wir auf 50 Ziffern genau

$$\begin{aligned} \frac{1}{1+2 \cdot 10^{-6}} &\doteq 0,99999800000399999200001599996800006399987200025600, \\ \frac{1-10^{-6}}{1+10^{-6}} &\doteq 0,99999800000199999800000199999800000199999800000200. \end{aligned}$$

Würde man nur mit 15 Stellen (ungefähr Matlab) rechnen, dann ergäbe sich

$$f(10^{-6}) \doteq 0,000000000002000,$$

also nur noch 4 (!) signifikante Stellen.

Multiplikation und Division sind gut konditioniert bezüglich relativer Fehler. Demnach ist die zum vorigen Beispiel analytisch äquivalente Funktionsvorschrift

$$f(t) = \frac{2t^2}{(1+t)(1+2t)}$$

numerisch stabil, und man erhält einen auf 15 Stellen genauen Wert:

$$f(10^{-6}) \doteq 1,99999400001400 \cdot 10^{-12}.$$

Ein zweites Beispiel sei das Lösen der quadratischen Gleichung $y^2 - x_1y + x_2 = 0$ nach y für gegebene x_1, x_2 . Bekanntlich lassen sich die Lösungen $y_{1;2}$ durch

$$y_{1;2} = f_{1;2}(x_1, x_2) = \frac{x_1}{2} \pm \sqrt{\left(\frac{x_1}{2}\right)^2 - x_2}$$

bestimmen. Die Jacobimatrix lässt sich zu

$$J_f(x) = \frac{1}{y_1 - y_2} \begin{pmatrix} y_1 & -1 \\ -y_2 & 1 \end{pmatrix}$$

berechnen, und deren Einträge werden groß, wenn $y_1 \approx y_2$. Es ergibt sich also: Das Nullstellenproblem ist schlecht konditioniert, wenn die Nullstellen nahe beieinander liegen.

Ist hingegen $|x_2| \ll |x_1|$, dann gilt

$$\left|\frac{x_1}{2}\right| = \sqrt{\left(\frac{x_1}{2}\right)^2} \approx \sqrt{\left(\frac{x_1}{2}\right)^2 - x_2},$$

d. h. bei einer der Formeln für y_1 oder y_2 tritt Auslöschung auf, d. h. das Ergebnis ist schlecht, obwohl das Problem gut konditioniert ist. Das bedeutet, der Algorithmus ist schlecht, d. h. nicht stabil.

Die Kondition hängt nur vom Problem ab und beschreibt die intrinsische Schwierigkeit. Auch wenn es gut konditioniert ist, kann es Algorithmen geben, die stabil sind, und solche, die es nicht sind.

2.4 Zahldarstellung, Maschinenzahlen, Gleitkommaarithmetik

Zu einer Zahl $x \in \mathbb{R}$ heißt

$$x = \pm \sum_{i=1}^{\infty} a_i 10^{K-i} = \pm 10^K \sum_{i=1}^{\infty} a_i 10^{-i}$$

eine exakte dezimale Gleitkomma-Darstellung von x . Die Zahl $B = 10$ heißt Basis der Darstellung, und $0, a_1 a_2 a_3 \dots$ mit $a_i \in \{0, \dots, 9\}$ und $K \in \mathbb{Z}$ heißen Mantisse und Exponent von x . Die Darstellung heißt normalisiert, falls $a_1 \neq 0$, und sie ist eindeutig bestimmt, wenn nicht $a_i = 9$ für fast alle $i \in \mathbb{N}$ gilt.

Andere gebräuchliche Basen sind $B = 2$ mit Ziffern $\{0, 1\}$ (Dual- oder Binärsystem), $B = 8$ mit Ziffern $\{0, 1, \dots, 7\}$ (Oktalsystem) und $B = 16$ mit Ziffern $\{0, 1, \dots, 9, A, B, \dots, F\}$ (Hexadezimalsystem).

Auf einem Computer sind nur endlich viele a_i speicherbar, d. h. a_1, \dots, a_M für eine gegebene Mantissenlänge $M \in \mathbb{N}$. Auch an den Exponenten K gibt es eine Einschränkung der Form $K_* \leq K \leq K^*$ mit $K_*, K^* \in \mathbb{Z}$.

Für $B, M \in \mathbb{N}$, $B \neq 1$ und $K_*, K^* \in \mathbb{Z}$ heißt

$$\mathcal{M}(B, M, K^*, K_*) := \left\{ \pm(0, a_1 a_2 \dots a_M)_B \cdot B^K \mid a_i \in \{0, 1, \dots, M-1\} \forall i = 1, \dots, M, a_1 \neq 0, K_* \leq K \leq K^*, K \in \mathbb{Z} \right\} \cup \{0\}$$

die Menge der (normalisierten) Maschinenzahlen. (Anmerkung: Natürlich werden im IEEE-Standard auch denormalisierte Darstellungen benutzt, um Zahlen kleiner als $(0,10\dots0)_B \cdot B^{K^*}$ zu speichern. Deshalb ist die definierte Menge \mathcal{M} nicht, wie im Skript behauptet, der volle Maschinenzahlenbereich. Wer es ganz genau wissen will, der schaue sich einfach <http://754r.ucbtest.org/standards/854.html> an.)

Eine „vernünftige“ Abbildung $\text{rd}: \mathbb{R} \rightarrow \mathcal{M}$ heißt Rundung auf die nächste Maschinenzahl. Eine solche vernünftige Rundung erfüllt die Fehlerschranken

$$|\text{rd}(x) - x| < \frac{1}{2} B^{K(x)-M} \quad \text{und} \quad \frac{|\text{rd}(x) - x|}{|x|} < \frac{1}{2} B^{1-M} =: \text{eps}.$$

eps heißt Maschinengenauigkeit, und für $B = 2$ gilt $\text{eps} = 2^{-M}$. Man berechne zum Testen mit Matlab $\log(\text{eps})/\log(2)$.

Wichtig ist, dass bei jeder Operation wie $+$, $-$, \cdot oder $:$ gerundet werden muss, weil das exakte Ergebnis i. a. keine Maschinenzahl ist. Das führt dazu, dass das Assoziativ- und das Distributivgesetz nicht mehr erfüllt sind. Sind a und b Maschinenzahlen, so bezeichnen wir mit $a \boxplus b$ die Maschinenzahl, die der Summe am nächsten liegt, also $\boxplus: \mathcal{M} \times \mathcal{M} \rightarrow \mathcal{M}$ mit $a \boxplus b := \text{rd}(a + b)$. Ebenso \boxminus, \boxdiv , usw.

Ferner gilt für gerades B , also auch für $B = 2$, die wichtige Beziehung

$$\text{eps} = \min\{a \in \mathcal{M} \mid 1 \boxplus a > 1\}.$$

2.5 Stabilität (Gutartigkeit) eines Algorithmus

Wir betrachten eine Rechenvorschrift (Algorithmus), die aus gegebenen Daten x_1, \dots, x_n die gesuchten Größen y_1, \dots, y_m berechnet, die durch Funktionen $y_i = f_i(x_1, \dots, x_n)$ gegeben sind. Durch die Rundung der Daten rechnet der Algorithmus jedoch mit $x + \Delta x$, wodurch ein verzerrtes Ergebnis $y + \Delta y$ berechnet wird. Dies führt zu einem unvermeidbaren relativen Fehler für y_i , der durch

$$\text{eps} \sum_{k=1}^n \kappa_{ik}^{(rel)}(x)$$

abgeschätzt werden kann.

Durch die Gleitkomma-Arithmetik berechnet der Computer allerdings $\boxed{f(x + \Delta x)}$ statt $f(x + \Delta x)$. Es entsteht also ein zusätzlicher Algorithmusfehler, und für den relativen Gesamtfehler folgt

$$\frac{\|f(x) - \boxed{f(x + \Delta x)}\|}{\|f(x)\|} \leq \frac{\|f(x) - f(x + \Delta x)\|}{\|f(x)\|} + \frac{\|f(x + \Delta x) - \boxed{f(x + \Delta x)}\|}{\|f(x + \Delta x)\|} + \frac{\|f(x) - f(x + \Delta x)\|}{\|f(x)\|} \cdot \frac{\|f(x + \Delta x) - \boxed{f(x + \Delta x)}\|}{\|f(x + \Delta x)\|}.$$

Darin ist der erste Summand der unvermeidbare relative Fehler, der zweite der relative Algorithmusfehler und der dritte das Produkt aus beiden.

Wir nennen einen Algorithmus stabil (gutartig), wenn der Algorithmusfehler nicht von größerer Ordnung als der unvermeidbare Fehler ist.

Auch wenn es zur Stabilität nicht viel mehr Allgemeines zu sagen gibt und der Abschnitt daher sehr kurz ausfällt, ist sie dennoch eines der wichtigsten Begriffe bei numerischen Algorithmen.

2.6 Zusammenfassung

Wichtig sind die Eins-, Euklid- und Maximumnorm auf dem \mathbb{C}^n und die von ihnen induzierten Matrixnormen, nämlich die Spaltensummen, Spektral- und Zeilensummennorm. Sie sind jeweils submultiplikativ und mit der Vektornorm verträglich.

Im Allgemeinen berechnet man ein y in Abhängigkeit von x_1, \dots, x_n . Wir betrachten das Problem als gut konditioniert, wenn bei exakter Rechnung kleine Veränderungen der x_i auch nur kleine Veränderungen des y mit sich bringen. Wir betrachten für $n = 2$ die Addition $y = x_1 + x_2$. Stören wir x_1 um ε , so erhalten wir

$$|(x_1 + \varepsilon + x_2) - (x_1 + x_2)| = |\varepsilon|,$$

d. h. der absolute Ausgangsfehler ist gleich dem absoluten Eingangsfehler – also ist das Problem absolut gut konditioniert. Für den relativen Fehler gilt jedoch

$$\left| \frac{(x_1 + \varepsilon + x_2) - (x_1 + x_2)}{x_1 + x_2} \right| = \left| \frac{\varepsilon}{x_1 + x_2} \right|,$$

und dieser kann beliebig viel größer als ε sein. Das Problem ist relativ schlecht konditioniert, wenn $x_1 + x_2 \approx 0$.

Noch einmal: Es ist unbedingt zu vermeiden, dass fast gleich große Zahlen (mit demselben Vorzeichen) voneinander subtrahiert werden. Allerdings tritt dieser Fall z. B. beim numerischen Differenzieren fast immer auf, so dass dieses Problem schlecht konditioniert ist.

Die Kondition von Gleichungssystemen hängt von der Matrix ab. In der Regel betrachtet man dazu die Kondition $\text{cond}(A) = \|A\| \|A^{-1}\|$ der Koeffizientenmatrix A .

Datenfehler entstehen vor Beginn eines Algorithmus, weil die Daten nicht exakt vorgegeben sind, und während des Rechnens, weil es nur endlich viele speicherbare Zahlen (Maschinenzahlen) gibt. Daher muss nach jeder Rechenoperation gerundet werden. Ein Algorithmus heißt stabil, wenn er der Kondition des Problems entsprechend verwertbare Ergebnisse liefert. Hat ein Problem die Kondition 10, dann hoffen wir, dass ein Algorithmus die Datenfehler nicht mehr als um den Faktor 100 verstärkt.

3 Interpolation und numerische Integration

Von einer Funktion, beispielsweise $f: [0, 10] \rightarrow \mathbb{R}$, seien nur die Funktionswerte $f(0), f(1), f(2), \dots, f(10)$ bekannt. Kann man Rückschlüsse auf den Verlauf der Funktion ziehen? Kann man die Funktion insbesondere näherungsweise zeichnen? Man kann die gegebenen Funktionswerte mit geeigneten Funktionen interpolieren (Polynome, Sinus und Cosinus, Splines, usw.). Kann man diese einfacheren Funktionen verwenden, um die ursprüngliche Funktion zu integrieren?

3.1 Polynominterpolation

Problemstellung: Gegeben seien $n + 1$ Stützstellen $t_0, \dots, t_n \in \mathbb{R}$ und $n + 1$ Stützwerte $y_0, \dots, y_n \in \mathbb{R}$. Gesucht ist ein Polynom $p_n \in \Pi_n$ (Vektorraum der Polynome vom Höchstgrad n) mit $p_n(t_i) = y_i$ für alle $i = 0, \dots, n$.

Schreibt man ein solches Polynom als $p_n(t) = \sum_{k=0}^n a_k t^k$, so entsteht aus den Forderungen $p_n(t_i) = y_i$ für alle $i = 0, \dots, n$ ein Gleichungssystem

$$\begin{pmatrix} 1 & t_0 & \cdots & t_0^n \\ 1 & t_1 & \cdots & t_1^n \\ \vdots & \vdots & & \vdots \\ 1 & t_n & \cdots & t_n^n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}.$$

Die Koeffizientenmatrix V_n ist eine Vandermonde-Matrix mit Determinante $\prod_{k>i}(t_k - t_i)$, d. h. die Matrix ist regulär, wenn die Stützstellen paarweise verschieden sind. In diesem Fall ist also die Polynominterpolationsaufgabe eindeutig lösbar, und das Polynom p_n heißt das Interpolationspolynom.

Der Grad von p_n kann natürlich auch kleiner als n sein. Ferner ist für jedes Polynom q auch $p_n(t) + q(t) \prod_{i=0}^n (t - t_i)$ ein interpolierendes Polynom, aber von größerem Grad als n . Das Lösen des Gleichungssystems ist allerdings kein guter numerischer Algorithmus, weil die Matrix schlecht konditioniert ist, wenn z. B. zwei Stützstellen nahe beieinander liegen.

Sind die $t_0, t_1, \dots, t_n \in \mathbb{R}$ paarweise verschieden, so definieren wir durch

$$\ell_{i,n}(t) := \prod_{k=0, k \neq i}^n \frac{t - t_k}{t_i - t_k} \quad \forall k = 0, \dots, n \quad \text{und} \quad \omega_{n+1}(t) := \prod_{i=0}^n (t - t_i)$$

das i -te Lagrange-Grundpolynom und das Knotenpolynom zu den Stützstellen t_0, t_1, \dots, t_n . Damit folgt:

- Es gilt die Darstellung

$$\ell_{i,n}(t) = \frac{1}{\omega'_{n+1}(t_i)} \frac{\omega_{n+1}(t)}{t - t_i}.$$

- $\ell_{i,n}$ ist ein Polynom vom Grad n mit $\ell_{i,n}(t_k) = \delta_{ik}$. Es löst also die Interpolationsaufgabe $\ell_{i,n}(t_i) = 1$ und $\ell_{i,n}(t_k) = 0$ für alle $k \neq i$.
- $\mathfrak{E}_n = (t^0, t^1, \dots, t^n)$ und $\mathfrak{L}_n = (\ell_{0,n}, \ell_{1,n}, \dots, \ell_{n,n})$ sind Basen des Π_n , und die Vandermonde-Matrix V_n wirkt als Basiswechsellmatrix, d. h. ist $p \in \Pi_n$ ein beliebiges Polynom, so gilt

$$V_n[p]_{\mathfrak{E}_n} = [p]_{\mathfrak{L}_n},$$

wobei $[p]_{\mathfrak{E}_n}$ und $[p]_{\mathfrak{L}_n}$ den Koordinatenvektor von p bezüglich \mathfrak{E}_n und \mathfrak{L}_n bezeichnen.

- Das Polynom

$$p_n(t) = \sum_{i=0}^n y_i \ell_{i,n}(t) = \omega_{n+1}(t) \sum_{i=0}^n \frac{y_i}{\omega'_{n+1}(t_i)} \frac{1}{t - t_i}$$

löst die Interpolationsaufgabe und heißt das Interpolationspolynom in Lagrangescher Darstellung.

- Es gilt

$$\sum_{i=0}^n \ell_{i,n}(t) = 1 \quad \forall t \in \mathbb{R},$$

denn links steht ein Interpolationspolynom in Lagrangescher Darstellung mit $y_0 = \dots = y_n = 1$, und rechts steht das Interpolationspolynom explizit, nämlich identisch 1.

Seien $C \neq 0$ und $c_{i,n} = C/\omega'_{n+1}(t_i)$ für $k = 0, \dots, n$. Dann gilt für das Interpolationspolynom die Formel

$$p_n(t) = \sum_{i=0}^n \frac{c_{i,n} y_i}{t - t_i} \bigg/ \sum_{i=0}^n \frac{c_{i,n}}{t - t_i},$$

welche seine baryzentrische Darstellung heißt.

Um einzusehen, dass dies tatsächlich ein Polynom vom Grad n liefert, berechnen wir mit den vorigen Eigenschaften

$$\begin{aligned} \sum_{i=0}^n \frac{c_{i,n}}{t - t_i} &= \sum_{i=0}^n \frac{C}{\omega'_{n+1}(t_i)(t - t_i)} = \frac{C}{\omega_{n+1}(t)} \sum_{i=0}^n \frac{\omega_{n+1}(t)}{\omega'_{n+1}(t_i)(t - t_i)} \\ &= \frac{C}{\omega_{n+1}(t)} \sum_{i=0}^n \ell_{i,n}(t) = \frac{C}{\omega_{n+1}(t)}. \end{aligned}$$

Dann folgt

$$p_n(t) = \frac{\omega_{n+1}(t)}{C} \sum_{i=0}^n \frac{c_{i,n} y_i}{t - t_i} = \sum_{i=0}^n y_i \frac{\omega_{n+1}(t)}{\omega'_{n+1}(t_i)(t - t_i)} = \sum_{i=0}^n y_i \ell_{i,n}(t),$$

was auch die Korrektheit zeigt.

Seien wie zuvor t_0, \dots, t_n paarweise verschiedene Stützstellen und y_0, \dots, y_n Stützwerte. Für $k = 0, \dots, n$ und $i = 0, \dots, n - k$ sei $p_{i,k}$ das Interpolationspolynom vom Grad k , das die Werte y_i, \dots, y_{i+k} auf den Stellen t_i, \dots, t_{i+k} interpoliert. Dann kann das Interpolationspolynom p_n mit dem Neville-Aitken-Algorithmus

$$\begin{aligned} p_{i,0}(t) &\equiv y_i && \text{für } i = 0, \dots, n, \\ p_{i,k}(t) &= \frac{(t - t_i)p_{i+1,k-1}(t) - (t - t_{i+k})p_{i,k-1}(t)}{t_{i+k} - t_i} && \text{für } k = 1, \dots, n, \quad i = 0, \dots, n - k, \\ p_n(t) &= p_{0,n}(t) \end{aligned}$$

berechnet werden.

Der Algorithmus wird lediglich dazu benutzt, das Polynom an einer festen Stelle $t \in \mathbb{R}$ auszurechnen, weil ansonsten Polynome rekursiv zu verarbeiten sind, was zu aufwendig ist. Allerdings muss die Auswertungsstelle t nicht in der konvexen Hülle der t_0, \dots, t_n liegen; man spricht dann von Extrapolation, weil das Interpolationspolynom nicht mehr zwischen zwei Stützstellen ausgewertet wird. Ein wichtiger solcher Spezialfall ist $t = 0$ mit $t_i > 0$ für alle $i = 0, \dots, n$.

Wir definieren eine weitere Basis $\mathfrak{N}_n = (\omega_0, \dots, \omega_n)$ von Π_n mit $\omega_k(t) = \prod_{i=0}^{k-1} (t - t_i)$; das leere Produkt ist $\omega_0(t) \equiv 1$. Die bezüglich dieser Basis entwickelte Darstellung

$$p_n(t) = \sum_{k=0}^n \gamma_k \omega_k(t)$$

heißt die Newtonsche Darstellung des Interpolationspolynoms. Wegen $\omega_k(t_i) = 0$ für $i < k$ entsteht ein lineares Gleichungssystem mit linker unterer Dreiecksmatrix

$$\begin{pmatrix} \omega_0(t_0) & & & \\ \omega_0(t_1) & \omega_1(t_1) & & \\ \vdots & \vdots & \ddots & \\ \omega_0(t_n) & \omega_1(t_n) & \cdots & \omega_n(t_n) \end{pmatrix} \begin{pmatrix} \gamma_0 \\ \gamma_1 \\ \vdots \\ \gamma_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix},$$

das mit Vorwärtseinsetzen gelöst werden kann.

Eine bessere Methode, die das Aufstellen der obigen Matrix umgeht, ist eine Neville-Aitken-artige Rekursion. Es gilt ähnlich wie oben

$$\begin{aligned} \gamma_{i,0} &= y_i && \text{für } i = 0, \dots, n, \\ \gamma_{i,k} &= \frac{\gamma_{i+1,k-1} - \gamma_{i,k-1}}{t_{i+k} - t_i} && \text{für } k = 1, \dots, n, \quad i = 0, \dots, n-k, \\ \gamma_k &= \gamma_{0,k} && \text{für } k = 0, \dots, n. \end{aligned}$$

Man nennt $\gamma_{i,k}$ die dividierte Differenz der Ordnung k zu t_i, \dots, t_{i+k} . Ist f eine stetige Funktion mit $y_i = f(t_i)$ für $i = 0, \dots, n$, so schreibt man auch $\gamma_{i,k} = f[t_i, \dots, t_{i+k}]$, und mit dieser Schreibweise gilt

$$\begin{aligned} f[t_i] &= y_i && \forall i = 0, \dots, n, \\ f[t_i, \dots, t_{i+k}] &= \frac{f[t_{i+1}, \dots, t_{i+k}] - f[t_i, \dots, t_{i+k-1}]}{t_{i+k} - t_i} && \forall k = 1, \dots, n \quad \forall i = 0, \dots, n-k, \\ p_n(t) &= \sum_{k=0}^n f[t_0, \dots, t_k] \omega_k(t). \end{aligned}$$

Weiter gilt für stetiges f die Summendarstellung

$$f[t_0, \dots, t_n] = \sum_{i=0}^n \frac{f(t_i)}{\omega'_{n+1}(t_i)},$$

aus der folgt, dass es auf die Reihenfolge der t_0, \dots, t_n nicht ankommt, und für n -mal stetig differenzierbares f gilt die Integraldarstellung

$$f[t_0, \dots, t_n] = \int_0^1 \int_0^{s_1} \cdots \int_0^{s_{n-1}} f^{(n)} \left(t_0 + \sum_{i=1}^n s_i (t_i - t_{i-1}) \right) ds_n \cdots ds_2 ds_1$$

woraus mit dem Mittelwertsatz

$$f[t_0, \dots, t_n] = \frac{f^{(n)}(\xi)}{n!} \quad \text{mit einem } \xi \in [\min\{t_0, \dots, t_n\}, \max\{t_0, \dots, t_n\}]$$

folgt.

Für $(n+1)$ -mal stetig differenzierbares f gilt für das Restglied der Interpolation

$$f(t) - p_n(t) = f[t_0, \dots, t_n, t] \prod_{i=0}^n (t - t_i) = \frac{f^{(n+1)}(\xi_t)}{(n+1)!} \omega_{n+1}(t)$$

mit einem $\xi_t \in [\min\{t_0, \dots, t_n, t\}, \max\{t_0, \dots, t_n, t\}]$.

Der Interpolationsfehler kann also durch

$$|f(t) - p_n(t)| \leq \frac{|\omega_{n+1}(t)|}{(n+1)!} \max_{\xi} |f^{(n+1)}(\xi)|$$

abgeschätzt werden, wobei wie üblich $\xi \in [\min\{t_0, \dots, t_n\}, \max\{t_0, \dots, t_n\}]$.

Die letzten Formeln, genauer die ab der Integraldarstellung für dividierte Differenzen, sind auch korrekt, wenn die Stützstellen nicht mehr paarweise verschieden sind. Vor allem gilt

$$f[\underbrace{t_0, \dots, t_0}_{n+1 \text{ mal}}] = \frac{f^{(n)}(t_0)}{n!}.$$

In diesem Extremfall $t_0 = t_1 = \dots = t_n$ ist p_n das um t_0 entwickelte Taylorpolynom

$$p_n(t) = \sum_{k=0}^n \frac{f^{(k)}(t_0)}{k!} (t - t_0)^k.$$

Hier gilt $p_n^{(k)}(t_0) = f^{(k)}(t_0)$ für $k = 0, \dots, n$, d. h. das Polynom stimmt zusätzlich in den Ableitungen an der Stelle t_0 mit f überein – statt nur im Funktionswert.

Weitere Ergebnisse: Das Interpolationspolynom konvergiert für wachsendes n i. a. nicht gegen die stetige Funktion f , nicht einmal punktweise. Statt dessen oszilliert es immer stärker und stimmt zwar in immer mehr Interpolationspunkten mit f überein, hat aber zwischen diesen vom Verhalten wenig mit f zu tun. Aber: Nach dem Weierstraßschen Approximationssatz lässt sich jede stetige Funktion $f: [a, b] \rightarrow \mathbb{R}$ gleichmäßig durch Polynome (sogar mit rationalen Koeffizienten) approximieren; das hat allerdings nichts mehr mit Interpolation zu tun. Der Vektorraum $\mathcal{C}^0([a, b], \mathbb{R})$ mit der Supremumsnorm ist damit separabel, weil eine abzählbare Teilmenge, nämlich der Vektorraum $\mathbb{Q}[x]$, dicht liegt.

Wir wollen nun noch einmal systematisch zusätzlich zu den Funktionswerten noch Ableitungen vorgeben. Seien also wieder t_0, \dots, t_n paarweise verschiedene Stützstellen. Für jedes $i = 0, \dots, n$ gebe nun $m_i + 1$ Werte $y_i^{(0)}, \dots, y_i^{(m_i)}$ vor und suche ein Polynom p_m mit

$$p_m^{(k)}(t_i) = y_i^{(k)} \quad \forall k = 0, \dots, m_i \quad \forall i = 0, \dots, n.$$

Dann gibt es genau ein Polynom $p_m \in \Pi_m$ mit $m = \sum_{i=0}^n (m_i + 1) - 1 = \sum_{i=0}^n m_i + n$, das die obige Hermite-Interpolationsaufgabe löst.

Wir betrachten den wichtigen Spezialfall $m_i = 1$ für $i = 0, \dots, n$, d. h. f und f' werden an den Stützstellen t_0, \dots, t_n interpoliert. Für $(2n+2)$ -mal stetig differenzierbares f gilt für das Restglied dieser Interpolation

$$f(t) - p_{2n+1}(t) = \frac{f^{(2n+2)}(\xi_t)}{(2n+2)!} \omega_{n+1}(t)^2$$

mit einem $\xi_t \in [\min\{t_0, \dots, t_n, t\}, \max\{t_0, \dots, t_n, t\}]$.

In erster Linie werden wir die Interpolation zur Konstruktion von Quadraturformeln benutzen. Analog werden sie auch zur numerischen Lösung von Differential- und Integralgleichungen verwendet. Schon erwähnt wurde, dass das Interpolationspolynom auch außerhalb der konvexen Hülle der t_0, \dots, t_n ausgewertet werden kann; man spricht dann von Extrapolation. Um verwertbare Ergebnisse zu erhalten, sollte der Auswertungspunkt nicht „zu weit weg“ von den t_0, \dots, t_n sein.

Ein abschließendes Beispiel bildet die numerische Differentiation. Für eine hinreichend glatte Funktion f wollen wir $f'(x)$ für ein festes x bestimmen, d. h. analytisch

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x-h)}{2h} =: \lim_{h \rightarrow 0} g(h).$$

Achtung: Numerisch tritt für kleines h immer Auslöschung im Zähler auf, denn es muss ja $f(x+h) - f(x-h) \rightarrow 0$ für $h \rightarrow 0$ gelten, d. h. das sture Auswerten von g für kleine h ist nicht stabil. (Zusätzlich ist numerische Differentiation i. a. noch schlecht konditioniert.) Man kann zeigen, dass eine Annahme der Form

$$g(h) = f'(x) + a_2 h^2 + a_4 h^4 + \dots$$

sinnvoll ist. Dann wertet man g in $h_0 > h_1 > \dots > h_n > 0$ aus und interpoliert g an den Stellen h_i^2 mit den berechneten Werten mit einem Interpolationspolynom p_n . Dann ist der extrapolierte Wert $p_n(0)$ eine i. a. wesentlich bessere Approximation an $f'(x)$ als $g(h)$ für sehr kleine h .

3.2 Das Hornerschema

Sei $p_n(t) = c_0 + c_1(t - t_0) + \dots + c_n(t - t_0) \cdots (t - t_{n-1})$ mit $c_n \neq 0$ ein Polynom vom Grad n in der Newton-Darstellung. Um es an einer Stelle τ auszuwerten, klammere es geschickt in

$$p_n(\tau) = c_0 + \left(c_1 + \cdots \left(\cdots \left(c_{n-1} + c_n(\tau - t_{n-1}) \right) (\tau - t_{n-2}) \cdots \right) \cdots (\tau - t_1) \right) (\tau - t_0)$$

um. Daraus entsteht sofort der Algorithmus für das einfache Hornerschema:

$$c'_n = c_n, \quad c'_k = c_k + c'_{k+1}(\tau - t_k) \quad \forall k = n-1, \dots, 0, \quad p_n(\tau) = c'_0.$$

Das Hornerschema erlaubt ferner eine effiziente Division mit Rest durch einen Linearfaktor $t - \tau$, denn es gilt

$$p_n(t) = (t - \tau)p_{n-1}(t) + p_n(\tau) \quad \text{mit} \quad p_{n-1}(t) := \sum_{k=1}^n c'_k \prod_{i=0}^{k-2} (t - t_i)$$

mit dem Quotienten $p_{n-1}(t)$ und dem Rest $p_n(\tau)$ (Polynom vom Grad 0). Im Fall, dass τ eine Nullstelle von p_n ist, verschwindet der Rest, und es steht eine echte Zerlegung da.

Formt man die Gleichung in

$$\frac{p_n(\tau) - p_n(t)}{\tau - t} = p_{n-1}(t)$$

um und lässt $t \rightarrow \tau$ gehen, so konvergiert die linke Seite gegen $p'_n(\tau)$ und die rechte gegen $p_{n-1}(\tau)$. Wendet man auf $p_{n-1}(t)$ noch einmal das Hornerschema an, so ergibt sich

$$\frac{p_n(\tau) - p_n(t)}{\tau - t} = p_{n-1}(t) = (t - \tau)p_{n-2}(t) + p_{n-1}(\tau),$$

und mit de l'Hôpital sieht man $p_{n-2}(\tau) = p''_n(\tau)/2$. Man kann bis zur n -ten Ableitung iterieren und erhält schließlich $p_0(\tau) = p_n^{(n)}(\tau)/n!$. Dadurch entsteht das vollständige Hornerschema

$$c_k^{(0)} = c_k,$$

$$\left. \begin{aligned} c_n^{(i+1)} &= c_n^{(i)}, \\ c_k^{(i+1)} &= c_k^{(i)} + c_{k+1}^{(i+1)}(\tau - t_{k-i}) \quad \forall k = n-1, \dots, i \end{aligned} \right\} \quad \forall i = 0, \dots, n,$$

$$\frac{p_n^{(i)}(\tau)}{i!} = c_i^{(i+1)}.$$

Das Polynom kann damit um eine Stelle τ entwickelt werden, denn mit der Darstellung als Taylorpolynom folgt

$$p_n(t) = \sum_{k=0}^n \frac{p_n^{(k)}(\tau)}{k!} (t - \tau)^k = \sum_{k=0}^n c_k^{(k+1)} (t - \tau)^k.$$

Im Spezialfall $t_0 = \dots = t_{n-1} = 0$ stimmt die anfängliche Newton-Darstellung mit der Standarddarstellung $p_n(t) = c_0 + c_1 t + \dots + c_n t^n$ überein. Dann kann p_n an der Stelle τ ausgewertet oder um τ unentwickelt werden.

Im Spezialfall $c_0 = \dots = c_{n-1} = 0$ und $c_n = 1$ ergibt sich $p_n(t) = \prod_{i=0}^{n-1} (t - t_i)$, und mit $\tau = 0$ erhalten wir $p_n(t) = \sum_{k=0}^n c_k^{(k+1)} t^k$. Damit sind die Koeffizienten Funktionen der Nullstellen t_0, \dots, t_{n-1} . Für quadratische Gleichungen $x^2 + px + q = (x - x_1)(x - x_2)$ liefert z. B. der Satz von Viëta $p = -(x_1 + x_2)$ und $q = x_1 x_2$.

Im Spezialfall $t_0 = \dots = t_{n-1} = -1$, $c_0 = \dots = c_{n-1} = 0$, $c_n = 1$ und $\tau = 0$ erhalten wir

$$\sum_{k=0}^n \binom{n}{k} t^k = (t + 1)^n = p_n(t) = \sum_{k=0}^n c_k^{(k+1)} t^k,$$

d. h. mit dem Horner Schema können effizient Binomialkoeffizienten ausgerechnet werden.

3.3 Splines

Die Polynominterpolation hat bei einer großen Stützstellenmenge das Problem, dass das Interpolationspolynom vor allem am Rand stark oszilliert. Eine bessere Idee ist es, zwischen je zwei Stellen ein Polynom von kleinerem Grad zu verwenden und diese an den Stellen hinreichend glatt zu verbinden.

Sei nun $\Delta = \{t_0, \dots, t_n\}$ mit $a = t_0 < t_1 < \dots < t_n = b$ eine Knotenmenge. Eine Funktion $s: [a, b] \rightarrow \mathbb{R}$ heißt ein kubischer Spline zur Knotenmenge Δ , falls s eine \mathcal{C}^2 -Funktion ist und die Einschränkung $s|_{I_i}$ von s auf ein Teilintervall $I_i = [t_{i-1}, t_i]$ für alle $i = 1, \dots, n$ ein Polynom vom Höchstgrad 3 ist. (Wer es genau wissen möchte: $\mathcal{C}^2([a, b], \mathbb{R})$ ist der Vektorraum aller Funktionen $f: [a, b] \rightarrow \mathbb{R}$, die auf (a, b) zweimal stetig differenzierbar sind und für die f , f' und f'' rechtsstetig in a und linksstetig in b fortgesetzt werden können.)

Ein kubischer Spline besteht also aus n Polynomen vom Höchstgrad 3, besitzt also $4n$ Freiheitsgrade. An den inneren $n - 1$ Knoten gibt es jedoch je drei Stetigkeitsbedingungen, wodurch $n + 3$ Freiheitsgrade übrig bleiben. Die Menge S_Δ aller kubischen Splines zu Δ bildet daher einen \mathbb{R} -Vektorraum mit $\dim S_\Delta = n + 3$. Eine mögliche Basis ist

$$\left\{ 1, t - t_0, (t - t_0)^2, (t - t_0)^3; (t - t_0)_+^3, \dots, (t - t_{n-1})_+^3 \right\},$$

worin $f_+ = \max\{f, 0\}$ für eine reellwertige Funktion f .

Wir geben nun $n + 1$ Stützwerte $y_0, \dots, y_n \in \mathbb{R}$ vor und fragen uns, ob es einen kubischen Spline $s \in S_\Delta$ mit $s(t_i) = y_i$ für alle $i = 0, \dots, n$ gibt. Man stellt fest, dass man noch immer $\dim S_\Delta - (n + 1) = 2$ Freiheitsgrade hat, d. h. die Lösungsmenge ist ein affiner Unterraum der Dimension 2. Es gibt drei geläufige Möglichkeiten, den Lösung eindeutig zu machen:

- Wir fordern zusätzlich $s''(a) = s''(b) = 0$, was zu einem natürlichen kubischen Spline führt.
- Wir fordern zusätzlich $s'(a) = q_0$ und $s'(b) = q_n$, was zu einem kubischen Hermite-Spline oder vollständigen kubischen Spline führt.
- Unter der Bedingung, dass ohnehin schon $y_0 = y_n$ gilt, fordern wir zusätzlich $s'(a) = s'(b)$ und $s''(a) = s''(b)$, was zu einem periodischen kubischen Spline führt.

Alle zusätzlichen Forderungen führen zu eindeutigen Lösungen.

Seien wie zuvor $a = t_0 < t_1 < \dots < t_n = b$, $y_0, \dots, y_n \in \mathbb{R}$, $s \in S_\Delta$ der interpolierende natürliche kubische Spline und $f \in \mathcal{C}^2([a, b], \mathbb{R})$ irgendeine interpolierende Funktion. Dann gilt für $f \neq s$ die Minimaleigenschaft

$$\int_a^b s''(t)^2 dt < \int_a^b f''(t)^2 dt,$$

d. h. der Spline minimiert in gewissem Sinn die Krümmung unter allen interpolierenden Funktionen.

Ist $s \in S_\Delta$ ein kubischer Spline, so ist s'' zumindest noch stetig und stimmt auf jedem Teilintervall I_i mit einem Polynom vom Höchstgrad 1 überein. (Insbesondere ist s'' stetig und fast überall differenzierbar, was eine mögliche Charakterisierung des Sobolew-Raums $H^1([a, b], \mathbb{R})$ ist.) Man kann s'' auch als Polygonzug bezeichnen. Er wird charakterisiert durch die Momente $M_i = s''(t_i)$ für $i = 0, \dots, n$, die im Folgenden bestimmt werden sollen.

Durch lineare Interpolation ergibt sich zunächst

$$s''|_{I_i}(t) = \frac{M_i(t - t_{i-1}) + M_{i-1}(t_i - t)}{t_i - t_{i-1}} \quad \forall i = 1, \dots, n.$$

Zweimalige Integration bezüglich t und Anpassen der beiden Integrationskonstanten, so dass $s|_{I_i}(t_{i-1}) = y_{i-1}$ und $s|_{I_i}(t_i) = y_i$ gilt, ergibt

$$s|_{I_i}(t) = \frac{t - t_{i-1}}{h_i} \left(y_i + \frac{M_i}{6} ((t - t_{i-1})^2 - h_i^2) \right) + \frac{t_i - t}{h_i} \left(y_{i-1} + \frac{M_{i-1}}{6} ((t_i - t)^2 - h_i^2) \right)$$

mit $h_i = t_i - t_{i-1}$ für $i = 1, \dots, n$. Differenzieren nach t ergibt

$$s'|_{I_i}(t) = \frac{y_i}{h_i} + \frac{M_i}{6h_i} (3(t - t_{i-1})^2 - h_i^2) - \frac{y_{i-1}}{h_i} - \frac{M_{i-1}}{6h_i} (3(t_i - t)^2 - h_i^2).$$

Die Forderung $s'|_{I_i}(t_i) = s'|_{I_{i+1}}(t_i)$ für $i = 1, \dots, n - 1$, was der Stetigkeitsbedingung von s' in t_i entspricht, ergibt

$$\begin{aligned} s'|_{I_i}(t_i) &= \frac{y_i - y_{i-1}}{h_i} + \frac{M_i h_i}{3} + \frac{M_{i-1} h_i}{6} = \frac{y_{i+1} - y_i}{h_{i+1}} - \frac{M_{i+1} h_{i+1}}{6} - \frac{M_i h_{i+1}}{3} = s'|_{I_{i+1}}(t_i) \\ \implies h_i M_{i-1} + 2(h_i + h_{i+1}) M_i + h_{i+1} M_{i+1} &= 6 \left(\frac{y_{i+1} - y_i}{h_{i+1}} - \frac{y_i - y_{i-1}}{h_i} \right). \end{aligned}$$

Für den natürlichen Spline gilt $M_0 = M_n = 0$, so dass die Momente M_1, \dots, M_{n-1} bestimmt werden müssen. Dazu verwenden wir die gerade hergeleiteten $n - 1$ Gleichungen und

schreiben sie als lineares Gleichungssystem

$$\begin{pmatrix} 2(h_1 + h_2) & h_2 & & & \\ h_2 & \ddots & \ddots & & \\ & \ddots & \ddots & h_{n-1} & \\ & & h_{n-1} & 2(h_{n-1} + h_n) & \\ & & & & \end{pmatrix} \begin{pmatrix} M_1 \\ \vdots \\ M_{n-1} \end{pmatrix} = 6 \begin{pmatrix} \frac{y_2 - y_1}{h_2} - \frac{y_1 - y_0}{h_1} \\ \vdots \\ \frac{y_n - y_{n-1}}{h_n} - \frac{y_{n-1} - y_{n-2}}{h_{n-1}} \end{pmatrix}$$

mit einer symmetrischen Dreibandmatrix. Dividiert man die i -te Zeile ($i = 1, \dots, n-1$) noch durch $h_i + h_{i+1}$, so erhalten wir

$$\begin{pmatrix} 2 & \lambda_1 & & & \\ \mu_2 & \ddots & \ddots & & \\ & \ddots & \ddots & \lambda_{n-2} & \\ & & \mu_{n-1} & 2 & \end{pmatrix} \begin{pmatrix} M_1 \\ \vdots \\ M_{n-1} \end{pmatrix} = 6 \begin{pmatrix} s[t_0, t_1, t_2] \\ \vdots \\ s[t_{n-2}, t_{n-1}, t_n] \end{pmatrix}$$

mit $\lambda_i = h_{i+1}/(h_i + h_{i+1})$ und $\mu_i = h_i/(h_i + h_{i+1}) = 1 - \lambda_i$ für $i = 1, \dots, n-1$.

Für den Hermite-Spline ist i. a. $M_0 \neq 0$ und $M_n \neq 0$, so dass $n+1$ Momente zu bestimmen sind. Die beiden zusätzlichen Bedingungen sind

$$s'|_{I_1}(t_0) = \frac{y_1}{h_1} - \frac{M_1 h_1}{6} - \frac{y_0}{h_1} - \frac{M_0 h_1}{3} \stackrel{!}{=} q_0,$$

$$s'|_{I_n}(t_n) = \frac{y_n}{h_n} + \frac{M_n h_n}{3} - \frac{y_{n-1}}{h_n} + \frac{M_{n-1} h_n}{6} \stackrel{!}{=} q_n.$$

Damit erhalten wir

$$\begin{pmatrix} 2h_1 & h_1 & & & & & \\ h_1 & 2(h_1 + h_2) & h_2 & & & & \\ & h_2 & \ddots & \ddots & & & \\ & & \ddots & \ddots & h_{n-1} & & \\ & & & h_{n-1} & 2(h_{n-1} + h_n) & h_n & \\ & & & & h_n & 2h_n & \end{pmatrix} \begin{pmatrix} M_0 \\ \vdots \\ M_n \end{pmatrix} = 6 \begin{pmatrix} \frac{y_1 - y_0}{h_1} - q_0 \\ \frac{y_2 - y_1}{h_2} - \frac{y_1 - y_0}{h_1} \\ \vdots \\ \frac{y_n - y_{n-1}}{h_n} - \frac{y_{n-1} - y_{n-2}}{h_{n-1}} \\ q_n - \frac{y_n - y_{n-1}}{h_n} \end{pmatrix}$$

mit einer symmetrischen Dreibandmatrix bzw. nach Division

$$\begin{pmatrix} 2 & 1 & & & & & \\ \mu_1 & 2 & \lambda_1 & & & & \\ & \mu_2 & \ddots & \ddots & & & \\ & & \ddots & \ddots & \lambda_{n-2} & & \\ & & & \mu_{n-1} & 2 & \lambda_{n-1} & \\ & & & & 1 & 2 & \end{pmatrix} \begin{pmatrix} M_0 \\ \vdots \\ M_n \end{pmatrix} = 6 \begin{pmatrix} s[t_0, t_0, t_1] \\ s[t_0, t_1, t_2] \\ \vdots \\ s[t_{n-2}, t_{n-1}, t_n] \\ s[t_{n-1}, t_n, t_n] \end{pmatrix}.$$

Für den periodischen Spline übernehmen wir die Matrix von zuvor ohne die erste und letzte Zeile und beachten, dass $M_0 = s''(t_0) = s''(t_n) = M_n$ gilt. Daher eliminieren wir M_0 aus dem Lösungsvektor und schreiben das h_1 bzw. μ_1 in die letzte Spalte der Matrix. Die Bedingung $s'(t_0) = s'(t_n)$ führt zu einer zusätzlichen Zeile. Damit erhalten wir

$$\begin{pmatrix} 2(h_1 + h_2) & h_2 & & & & & & h_1 \\ h_2 & \ddots & \ddots & & & & & \\ & \ddots & \ddots & & & & & \\ & & & h_{n-1} & & & & \\ h_1 & & h_{n-1} & 2(h_{n-1} + h_n) & & h_n & & \\ & & & h_n & & 2(h_n + h_1) & & \end{pmatrix} \begin{pmatrix} M_1 \\ \vdots \\ M_n \end{pmatrix} = 6 \begin{pmatrix} \frac{y_2 - y_1}{h_2} - \frac{y_1 - y_0}{h_1} \\ \vdots \\ \frac{y_n - y_{n-1}}{h_n} - \frac{y_{n-1} - y_{n-2}}{h_{n-1}} \\ \frac{y_1 - y_0}{h_1} - \frac{y_n - y_{n-1}}{h_n} \end{pmatrix}$$

mit einer symmetrischen dünnbesetzten Matrix bzw. nach Division

$$\begin{pmatrix} 2 & \lambda_1 & & & & & & \mu_1 \\ \mu_2 & \ddots & \ddots & & & & & \\ & \ddots & \ddots & & & & & \\ & & & \lambda_{n-2} & & & & \\ \lambda_n & & \mu_{n-1} & 2 & \lambda_{n-1} & & & \\ & & & \mu_n & 2 & & & \end{pmatrix} \begin{pmatrix} M_1 \\ \vdots \\ M_n \end{pmatrix} = 6 \begin{pmatrix} s[t_0, t_1, t_2] \\ \vdots \\ s[t_{n-2}, t_{n-1}, t_n] \\ s[t_{n-1}, t_n, t_0] \end{pmatrix}$$

mit $\lambda_n = h_1/(h_n + h_1)$ und $\mu_n = h_n/(h_n + h_1)$.

Alle Matrizen A (in der Version mit den λ_i und μ_i) sind regulär, und sie erfüllen $\|A\|_\infty = 3$ und $\|A^{-1}\|_\infty \leq 1$. Damit folgt $\text{cond}_\infty(A) \leq 3$, d.h. die Gleichungssysteme sind sehr gut konditioniert, und zwar unabhängig von der Anzahl und Lage der Knoten. Zur Berechnung der M_i sind nur $5n$ wesentliche Operationen durchzuführen, und die Gauß-Elimination ist mit kanonischer Pivotwahl möglich, weil die Matrix strikt diagonaldominant ist.

3.4 Quadratur (numerische Integration)

Für eine stetige Funktion $f: [a, b] \rightarrow \mathbb{R}$ ist eine Approximation an das Integral $I[f] = \int_a^b f(t) dt$ gesucht. Eine solche Quadraturformel hat in der Regel die Gestalt

$$Q_n[f] = \sum_{i=0}^n \alpha_{in} f(t_{in}),$$

worin die $\alpha_{in} \in \mathbb{R}$ Gewichte und die $t_{in} \in [a, b]$ Knoten oder Stützstellen heißen. Die Theorie der Quadratur beschäftigt sich nun im Wesentlichen damit, wie die Gewichte und Knoten geschickt gewählt werden können, so dass der Quadraturfehler klein wird, ohne dass zu viele Funktionsauswertungen benötigt werden.

Seien $f \in C([a, b], \mathbb{R})$, $a \leq t_{0n} < t_{1n} < \dots < t_{nn} \leq b$ und $p_n \in \Pi_n$ das eindeutig bestimmte Interpolationspolynom zu den Stützstellen t_{0n}, \dots, t_{nn} und Stützwerten $f(t_{0n}), \dots, f(t_{nn})$.

Dann kann eine Quadraturformel dadurch konstruiert werden, dass statt f das Polynom p_n integriert wird, und es gilt

$$Q_n[f] = \sum_{i=0}^n \left(\int_a^b \ell_{i,n}(t) dt \right) f(t_{in}),$$

d. h. die Gewichte sind gerade die Integrale über die Lagrange-Grundpolynome.

Wir definieren den Exaktheitsgrad einer beliebigen Quadraturformel $Q[f]$ als die natürliche Zahl $m \in \mathbb{N}$, so dass $Q[p] = I[p]$ für alle $p \in \Pi_m$ gilt, aber ein Polynom $p_{m+1} \in \Pi_{m+1}$ mit $Q[p_{m+1}] \neq I[p_{m+1}]$ existiert. Ist die Quadraturformel (wie bei $\sum_i \alpha_i f(t_i)$) linear, so muss für den Exaktheitsgrad nur eine Basis des Π_m getestet werden. Der Exaktheitsgrad einer Quadraturformel der Form $Q_n[f] = \sum_{i=0}^n \alpha_{in} f(t_{in})$ ist immer kleiner $2(n+1)$ und damit endlich, denn das Polynom $p(t) = \prod_{i=0}^n (t - t_{in})^2 \in \Pi_{2(n+1)}$ (Quadrat des Knotenpolynoms) wird nie exakt approximiert.

Die zuletzt hergeleitete Formel ist optimal im folgenden Sinn: Sind die Knoten t_{0n}, \dots, t_{nn} vorgegeben, so erreicht die Quadraturformel $Q_n[f] = \sum_{i=0}^n \alpha_{in} f(t_{in})$ genau dann den maximalen Exaktheitsgrad, wenn die Gewichte als $\alpha_{in} = \int_a^b \ell_{i,n}(t) dt$ gewählt werden. Dieser Exaktheitsgrad ist nach der Konstruktion über das Interpolationspolynom mindestens n .

Damit die Gewichte nicht vom Integrationsintervall abhängen, skaliert man für gewöhnlich $[a, b]$ z. B. auf $[0, 1]$ (oder auch $[-1, 1]$) um. Mit der Transformation $s_{in} = (t_{in} - a)/(b - a)$ und $s = (t - a)/(b - a)$ erhält man dann

$$\alpha_{in} = (b - a) \int_0^1 \prod_{k=0, k \neq i}^n \frac{s - s_{kn}}{s_{in} - s_{kn}} ds.$$

Für den Quadraturfehler bei $(n+1)$ -mal stetig differenzierbarem f gilt

$$\begin{aligned} |Q_n[f] - I[f]| &= \left| \int_a^b (p_n(t) - f(t)) dt \right| = \left| \int_a^b f[t_{0n}, \dots, t_{nn}] \omega_{n+1}(t) dt \right| \\ &\leq \frac{1}{(n+1)!} \|f^{(n+1)}\|_{[a,b]} \int_a^b |\omega_{n+1}(t)| dt = \frac{(b-a)^{n+2}}{(n+1)!} \|f^{(n+1)}\|_{[a,b]} \int_0^1 \prod_{i=0}^n |s - s_{in}| ds. \end{aligned}$$

Darin bezeichne $\|g\|_{[a,b]}$ die Supremumsnorm einer Funktion $g \in \mathcal{C}([a, b], \mathbb{R})$. Diese Fehlerformel wird meist wie folgt verwendet: Für eine bestimmte Quadraturformel sind die Knoten t_{0n}, \dots, t_{nn} bekannt. Der letzte Integralausdruck hängt aber ausschließlich von den (umskalierten) Knoten ab und kann daher unabhängig von a, b und f bestimmt werden. Die verbleibende Schwierigkeit besteht darin, eine gute Abschätzung für $\|f^{(n+1)}\|_{[a,b]}$ zu finden.

Eine wichtige Klasse von Quadraturformeln sind die abgeschlossenen Newton-Cotes-Formeln. Für jedes $n \in \mathbb{N}$ gibt es genau eine solche Formel, nämlich die Polynominterpolationsformel zum äquidistanten Gitter mit $t_{0n} = a$ und $t_{nn} = b$. Merke: Die Newton-Cotes-Formel n -ter Ordnung entsteht, indem das Interpolationspolynom auf dem äquidistanten Gitter mit $n+1$ Knoten, unter denen a und b sind, integriert wird.

Für $n = 1$ gilt $t_{01} = a$ und $t_{11} = b$, und man erhält die Trapezregel

$$T[f] = \frac{h}{2} (f(a) + f(b)) \quad \text{mit Fehler} \quad |T[f] - I[f]| \leq \frac{b-a}{12} \|f''\|_{[a,b]} h^2,$$

wobei $h = b - a$ ist. Für $n = 2$ gilt $t_{02} = a$, $t_{12} = (a + b)/2$ und $t_{22} = b$, und man erhält die Simpsonregel

$$S[f] = \frac{h}{3} \left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right) \quad \text{mit Fehler} \quad |S[f] - I[f]| \leq \frac{b-a}{180} \|f^{(4)}\|_{[a,b]} h^4,$$

wobei $h = (b - a)/2$ ist. Während beide Formeln für jedes stetige f definiert sind, kann die Fehlerabschätzung natürlich nur für C^2 - bzw. C^4 -Funktionen formuliert werden.

Diese Formeln sind nur für kleine Polynomgrade geeignet, und zwar aus zwei Gründen: Erstens konvergiert für wachsendes n die Folge der Newton-Cotes-Approximationen i. a. nicht gegen das Integral. Zweitens entstehen bei Polynomgraden $n \geq 8$ negative Gewichte, was numerisch unsinnig ist: Es kann nicht sinnvoll sein, das Integral einer positiven Funktion zu approximieren, indem in der Mittelung bestimmte Funktionswerte negativ gewichtet werden. Stichwort: Auslöschung.

Bei den halboffenen oder offenen Newton-Cotes-Formeln ist einer der beiden Randpunkte oder beide Randpunkte kein Knoten. Diese Formeln sind nicht weiter relevant, mit einer Ausnahme, nämlich der Mittelpunkregel

$$M[f] = hf\left(\frac{a+b}{2}\right) \quad \text{mit Fehler} \quad |M[f] - I[f]| \leq \frac{b-a}{24} \|f''\|_{[a,b]} h^2,$$

wobei $h = b - a$ ist.

$T[f]$, $S[f]$ und formal auch $M[f]$ entstehen durch Integration des Interpolationspolynoms vom Grad 1, 2 und formal 0 (durch einen Knoten $t_{00} = (a + b)/2$ festgelegt). Sie besitzen also auch mindestens diese Exaktheitsgrade, und tatsächlich gilt sogar: $T[f]$ und $M[f]$ haben Exaktheitsgrad 1, $S[f]$ hat Exaktheitsgrad 3. Dies lässt sich auch leicht daran erkennen, dass der Fehler für die entsprechenden Polynome verschwindet. Ferner sind die Fehlerschranken scharf, d. h. bestmöglich.

Der größere Exaktheitsgrad ist eine allgemeine Folgerung: Wird eine Quadraturformel $Q_n[f]$ durch Polynominterpolation mit geradem n konstruiert und werden die Knoten symmetrisch zu $(a + b)/2$ gewählt, dann hat die Quadraturformel automatisch mindestens Exaktheitsgrad $n + 1$ (statt nur n). Das ist leicht einzusehen, weil $I[p_k] = Q_n[p_k] = 0$ für alle $p_k(t) = (t - (a + b)/2)^k$ mit ungeradem $k \in \mathbb{N}$.

Statt den Polynomgrad zu erhöhen, was wie oben beschrieben i. a. nicht zum gewünschten Ziel führt, zerlegt man das Intervall $[a, b]$ in m gleichgroße Teilintervalle und wendet auf jedem Teilintervall eine Newton-Cotes-Formel an. Dabei entstehen sog. zusammengesetzte Quadraturverfahren, und zwar das Trapezverfahren

$$T_m[f] = \frac{h}{2} f(a) + h \sum_{i=1}^{m-1} f(a + ih) + \frac{h}{2} f(b) \\ \text{mit} \quad h = \frac{b-a}{m} \quad \text{und Fehler} \quad |T_m[f] - I[f]| \leq \frac{b-a}{12} \|f''\|_{[a,b]} h^2,$$

das Simpsonverfahren

$$S_m[f] = \frac{h}{3} f(a) + \frac{4h}{3} \sum_{i=1}^m f(a + (2i - 1)h) + \frac{2h}{3} \sum_{i=1}^{m-1} f(a + 2ih) + \frac{h}{3} f(b) \\ \text{mit} \quad h = \frac{b-a}{2m} \quad \text{und Fehler} \quad |S_m[f] - I[f]| \leq \frac{b-a}{180} \|f^{(4)}\|_{[a,b]} h^4$$

und das Mittelpunktverfahren

$$M_m[f] = h \sum_{i=1}^m f\left(a + \left(i - \frac{1}{2}\right)h\right)$$

mit $h = \frac{b-a}{m}$ und Fehler $|M_m[f] - I[f]| \leq \frac{b-a}{24} \|f''\|_{[a,b]} h^2$.

$T_m[f]$ benötigt $m + 1$ Funktionsauswertungen, $S_m[f]$ $2m + 1$ und $M_m[f]$ genau m .

Sei $Q_1[f], Q_2[f], Q_3[f], \dots$ ein beliebiges Quadraturverfahren, und die Anzahl der Funktionsauswertungen für $Q_m[f]$ sei durch $C'm$ mit einem $C' > 0$ nach oben abschätzbar. Gibt es dann für $\alpha > 0$ eine Konstante $C_\alpha > 0$, so dass

$$|Q_m[f] - I[f]| \leq C_\alpha m^{-\alpha}$$

für eine geeignete Funktionenklasse, so hat das Quadraturverfahren mindestens die Konsistenzordnung α . Das bedeutet: Verdoppelt man die Anzahl der Funktionsauswertungen, so reduziert sich der Quadraturfehler um den Faktor $2^{-\alpha}$.

Daraus ergibt sich sofort mit den angegebenen Fehlerformeln: Das Trapez- und Mittelpunktverfahren haben für \mathcal{C}^2 -Funktionen Konsistenzordnung 2, während das Simpsonverfahren für \mathcal{C}^4 -Funktionen Konsistenzordnung 4 besitzt. Erfüllt ein stetiges f nicht diese Voraussetzungen, so konvergieren alle drei Verfahren noch immer gegen das Integral, aber die Konvergenz wird i. a. deutlich langsamer sein. Ganz wesentlich ist, dass der Exaktheitsgrad des zusammengesetzten Verfahrens nicht größer ist als in der ursprünglichen Regel.

Die analytische Aussage $T_m[f] \rightarrow I[f]$ hilft numerisch wenig weiter, weil wir ja nicht beliebig kleine $h = (b-a)/m$ einsetzen können; das würde den Rechenaufwand in die Höhe treiben und Stabilitätsprobleme mit sich bringen. Statt dessen kann man $T_m[f]$ für einige h ausrechnen und dann eine Extrapolation auf $h = 0$ vornehmen. Dazu muss man das Verhalten des Quadraturfehlers $R(h) = T_m[f] - I[f]$ kennen, und genau dies leistet die Euler-Maclaurinsche Summenformel

$$T_m[f] = I[f] + \sum_{i=1}^k \sigma_i h^{2i} + \sigma_{k+1}(h) h^{2k+2} \quad \text{mit} \quad \sigma_{k+1}(h) = \frac{B_{2k+2}}{(2k+2)!} (b-a) f^{(2k+2)}(\xi_h)$$

und $\sigma_i = \frac{B_{2i}}{(2i)!} (f^{(2i-1)}(b) - f^{(2i-1)}(a)) \quad \forall i = 1, \dots, k$

mit einer Funktion $f \in \mathcal{C}^{2k+2}([a, b], \mathbb{R})$, den Bernoullizahlen B_ℓ und einem $\xi_h \in (a, b)$.

Das Trapezverfahren ist extrem leistungsstark für $(b-a)$ -periodische Funktionen. Dann gilt $\sigma_i = 0$ für alle $i = 1, \dots, k$, und der Quadraturfehler kann mit $|T_m[f] - I[f]| = O(h^{2k+2})$ abgeschätzt werden. Ist f sogar eine \mathcal{C}^∞ -Funktion, dann konvergiert $T_m[f] \rightarrow I[f]$ schneller als jede Potenz.

Wir geben nun natürliche Zahlen $m_0 < m_1 < \dots < m_n$ vor, woraus $h_0 > h_1 > \dots > h_n$ folgt, und berechnen die Trapezsummen $T_{m_i}[f]$ für $i = 0, \dots, n$. Dann wird der Neville-Aitken-Algorithmus in der Form

$$T_{i0} = T_{m_i}[f] \quad \forall i = 0, \dots, n,$$

$$T_{ik} = T_{i+1, k-1} + \frac{T_{i+1, k-1} - T_{i, k-1}}{(h_i/h_{i+k})^2 - 1} \quad \forall i = 0, \dots, n-k \quad \forall k = 1, \dots, n$$

Richardson-Extrapolation auf Schrittweite 0 genannt. In T_{0n} kommt nämlich das Polynom, das die Daten $(h_i^2, T_{m_i}[f])$ für $i = 0, \dots, n$ interpoliert, ausgewertet an der Stelle 0 zurück. Der Algorithmus heißt dann Romberg-Quadratur, und die T_{ik} bilden das Romberg-Tableau.

Durch die Romberg-Quadratur wird tatsächlich der Exaktheitsgrad größer als er bei der Trapezregel ist. Wir betrachten die Funktion $f: [0, 1] \rightarrow \mathbb{R}$ mit $f(x) = x^2$ und exaktem Integral $I[f] = 1/3$. Ferner wählen wir $m_0 = 1$ und $m_1 = 2$, d. h. $h_0 = 1$ und $h_1 = 1/2$, und berechnen

$$T_{m_0}[f] = h_0 \left(\frac{f(0)}{2} + \frac{f(1)}{2} \right) = \frac{1}{2}, \quad T_{m_1}[f] = h_1 \left(\frac{f(0)}{2} + f\left(\frac{1}{2}\right) + \frac{f(1)}{2} \right) = \frac{3}{8}.$$

Beide Näherungen sind nicht exakt, weil die Trapezformel nur Exaktheitsgrad 1 besitzt. Für das Romberg-Tableau gilt dann

$$T_{00} = \frac{1}{2}, \quad T_{10} = \frac{3}{8}, \quad T_{01} = T_{10} + \frac{T_{10} - T_{00}}{(h_0/h_1)^2 - 1} = \frac{1}{3},$$

d. h. das Ergebnis T_{10} der Romberg-Quadratur ist exakt.

Nun suchen wir Quadraturformeln, die einen höheren Exaktheitsgrad besitzen als die Newton-Cotes-Formeln. Wir wissen bereits, wie wir zu gegebenen Knoten die Gewichte optimal wählen müssen, und wir wissen auch, dass bei $n+1$ Knoten der Exaktheitsgrad höchstens $2n+1$ sein kann. Wie zuvor betrachten wir wieder $t_{0n} < t_{1n} < \dots < t_{nn}$, eine Funktion $f \in \mathcal{C}^{2n+2}([t_{0n}, t_{nn}], \mathbb{R})$ und definieren

$$p_{2n+1}(t) = \sum_{i=0}^n \left(f(t_{in})(1 - 2\ell'_{i,n}(t_{in})(t - t_{in})) + f'(t_{in})(t - t_{in}) \right) \ell_{i,n}(t)^2.$$

Dann erfüllt p_{2n+1} eine Hermite-Interpolation, d. h. es gilt

$$p_{2n+1}(t_{in}) = f(t_{in}), \quad p'_{2n+1}(t_{in}) = f'(t_{in}) \quad \forall i = 0, \dots, n,$$

$$p_{2n+1} \in \Pi_{2n+1} \quad \text{und} \quad f(t) - p_{2n+1}(t) = \frac{f^{(2n+2)}(\xi_t)}{(2n+2)!} \omega_{n+1}(t)^2.$$

Wir konstruieren nun eine Quadraturformel, eine sog. Hermite-Quadratur, indem wir statt f das Polynom p_{2n+1} integrieren, und erhalten

$$H_n[f] = \int_a^b p_{2n+1}(t) dt = \sum_{i=0}^n \alpha_{in} f(t_{in}) + \sum_{i=0}^n \beta_{in} f'(t_{in})$$

mit $\alpha_{in} = \int_a^b (1 - 2\ell'_{i,n}(t_{in})(t - t_{in})) \ell_{i,n}(t)^2 dt$ und $\beta_{in} = \int_a^b (t - t_{in}) \ell_{i,n}(t)^2 dt$.

Diese Quadraturformel ist i. a. unbrauchbar, weil man die Ableitung von f auswerten muss.

Das Ziel ist es nun, die Knoten t_{0n}, \dots, t_{nn} derart zu wählen, dass die β_{in} allesamt 0 werden. Wegen $\alpha_{in} = \int_a^b \ell_{i,n}(t)^2 dt - 2\ell'_{i,n}(t_{in})\beta_{in}$ sind dann alle Gewichte α_{in} positiv, und die Quadraturformel hat mit $n+1$ Knoten den maximal möglichen Exaktheitsgrad $2n+1$. Mit einer Darstellung für $\ell_{i,n}$ erhalten wir

$$\beta_{in} = \frac{1}{\omega'_{n+1}(t_{in})} \int_a^b \omega_{n+1}(t) \ell_{i,n}(t) dt \stackrel{!}{=} 0$$

und interpretieren dieses Ergebnis im Folgenden.

Zunächst stellen wir fest, dass auf dem Vektorraum $\mathcal{C}([a, b], \mathbb{R})$ durch

$$\langle f, g \rangle_{\mathcal{L}^2(a,b)} = \int_a^b f(x)g(x) dx \quad \forall f, g \in \mathcal{C}([a, b], \mathbb{R})$$

ein Skalarprodukt definiert ist, das sog. \mathcal{L}^2 -Skalarprodukt.

Insbesondere ist es ein Skalarprodukt auf dem Vektorraum $\mathbb{R}[x]$, und für jedes Skalarprodukt $\langle \cdot, \cdot \rangle$ auf diesem Polynomraum gilt: Für jede Folge c_0, c_1, c_2, \dots von 0 verschiedener reeller Zahlen gibt es genau eine Folge q_0, q_1, q_2, \dots von Polynomen, so dass q_k ein Polynom vom Grad k mit Leitkoeffizient c_k ist und $q_k \perp \Pi_{k-1}$ gilt, d. h. $\langle q_k, p \rangle = 0$ für alle $p \in \Pi_{k-1}$. Dann heißt q_k das k -te Orthogonalpolynom bezüglich $\langle \cdot, \cdot \rangle$, und es ist bis auf ein skalares Vielfaches eindeutig definiert. Das Polynom q_k besitzt k paarweise verschiedene reelle Nullstellen, die allesamt im Innern von $[a, b]$ liegen. Die Polynome können beispielsweise mit dem Gram-Schmidt-Verfahren bestimmt werden.

Die Forderung $\beta_{in} = 0$ für alle $i = 0, \dots, n$ ergibt nun, dass $\omega_{n+1} \in \Pi_{n+1}$ bezüglich des \mathcal{L}^2 -Skalarprodukts senkrecht auf jedem $\ell_{i,n} \in \Pi_n$ stehen muss. Da aber $\{\ell_{0,n}, \dots, \ell_{n,n}\}$ eine Basis von Π_n ist, muss ω_{n+1} auf dem ganzen Π_n senkrecht stehen, d. h. es muss bis auf eine Konstante mit dem $(n+1)$ -ten Orthogonalpolynom übereinstimmen. Wir nennen die Quadraturformel

$$G_n[f] = \sum_{i=0}^n \alpha_{in} f(t_{in}) \quad \text{mit} \quad \alpha_{in} = \int_a^b \ell_{i,n}(t) dt = \int_a^b \ell_{i,n}(t)^2 dt,$$

in der die Knoten t_{0n}, \dots, t_{nn} die $n+1$ paarweise verschiedenen Nullstellen des $(n+1)$ -ten Orthogonalpolynoms zu einem Skalarprodukt $\langle \cdot, \cdot \rangle$ auf $\mathcal{C}([a, b], \mathbb{R})$ sind, die Gauß-Quadratur zu $\langle \cdot, \cdot \rangle$. Diese besitzt als einzige den maximal möglichen Exaktheitsgrad, und alle Gewichte sind positiv.

Die Orthogonalpolynome bezüglich des $\mathcal{L}^2(-1, 1)$ -Skalarprodukts mit einer bestimmten Normierung heißen Legendre-Polynome P_k . Es gilt $P_0(x) = 1$ und $P_1(x) = x$, und die weiteren Polynome können mit der Rekursionsformel

$$(k+1)P_{k+1}(x) = (2k+1)xP_k(x) - kP_{k-1}(x) \quad \forall k \in \mathbb{N}$$

bestimmt werden. Die Gauß-Quadratur $G_n[f]$, die das Legendre-Polynom P_{n+1} benutzt, heißt Gauß-Legendre-Formel. Für $n = 0$ ergibt sich die Mittelpunkregel $G_0[f] = M[f] = 2f(0)$, für $n = 1$ und $n = 2$ erhalten wir

$$G_1[f] = f\left(-\frac{1}{\sqrt{3}}\right) + f\left(\frac{1}{\sqrt{3}}\right), \quad G_2[f] = \frac{5}{9}f\left(-\sqrt{\frac{3}{5}}\right) + \frac{8}{9}f(0) + \frac{5}{9}f\left(\sqrt{\frac{3}{5}}\right).$$

Seien nun $J \subseteq \mathbb{R}$ ein (möglicherweise unbeschränktes) Intervall und $w: J \rightarrow \mathbb{R}$ eine Funktion (Gewichtsfunktion), so dass $\langle \cdot, \cdot \rangle_w: \mathbb{R}[x] \times \mathbb{R}[x] \rightarrow \mathbb{R}$ gegeben durch

$$(p, q) \mapsto \langle p, q \rangle_w = \int_J p(x)q(x)w(x) dx \quad \forall p, q \in \mathbb{R}[x]$$

ein Skalarprodukt auf $\mathbb{R}[x]$ ist. (Notwendigerweise muss w z. B. Riemann-integrierbar und nicht-negativ sein.) Zu jedem Skalarprodukt $\langle \cdot, \cdot \rangle_w$ gehört dann ein System von Orthogonalpolynomen sowie eine Gauß-Quadratur. Wir geben drei Beispiele:

Für $J = (-1, 1)$ und $w(t) = (1 - t^2)^{-1/2}$ ergeben sich die Tschebyscheff-Polynome T_k . Es ist $T_0(x) = 1$ und $T_1(x) = x$, sie erfüllen die Rekursionsformel $T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x)$, und es gilt $T_k(x) = \cos(k \arccos x)$ für $x \in [-1, 1]$. Für $J = [0, \infty)$ und $w(t) = e^{-t}$ ergeben sich die Laguerre-Polynome L_k . Es ist $L_0(x) = 1$ und $L_1(x) = -x + 1$, und sie erfüllen die Rekursionsformel $(k + 1)L_{k+1}(x) = (2k + 1 - x)L_k(x) - kL_{k-1}(x)$. Für $J = \mathbb{R}$ und $w(t) = e^{-t^2}$ ergeben sich die Hermite-Polynome H_k . Es ist $H_0(x) = 1$ und $H_1(x) = 2x$, und sie erfüllen die Rekursionsformel $H_{k+1}(x) = 2xH_k(x) - 2kH_{k-1}(x)$.

3.5 Zusammenfassung

Bei der Interpolation werden gegebene Punkte durch geeignete Funktionen „verbunden“. Wir betrachten nur die Polynominterpolation und geben dazu $n + 1$ paarweise verschiedene reelle Stützstellen t_0, \dots, t_n vor. Für jedes $i = 0, \dots, n$ geben wir $m_i + 1$ Werte $y_i^{(0)}, \dots, y_i^{(m_i)}$ vor. Dann gibt es ein eindeutiges Polynom $p_m \in \Pi_m$ für $m = \sum_{i=0}^n m_i + n$, so dass

$$p_m^{(k)}(t_i) = y_i^{(k)} \quad \forall k = 0, \dots, m_i \quad \forall i = 0, \dots, n.$$

Für $m_i = 0$ für alle i erhalten wir die gewöhnliche Interpolationsaufgabe. Dann ist $m = n$, und p_n kann als

$$p_n(t) = \sum_{i=0}^n c_i t^i = \sum_{i=0}^n y_i l_{i,n}(t) = \sum_{i=0}^n \gamma_i \omega_i(t)$$

geschrieben werden. Die erste Darstellung ist bezüglich der Standardbasis, und zur Berechnung der c_i muss eine möglicherweise schlecht konditionierte Vandermonde-Matrix invertiert werden. Bei der Lagrange-Darstellung sind die Koeffizienten trivial, allerdings ist die Basis hässlich. Bei der letzten Darstellung mit der Newton-Basis können die Koeffizienten γ_i effizient mit einem Neville-Aitken-Algorithmus berechnet werden. Die letzte Darstellung ist insbesondere auch für den allgemeinen Fall ($m_i > 0$) gut geeignet.

Die Konvergenzeigenschaften der Interpolationspolynome mit wachsendem n beispielsweise gegen eine gegebene stetige Funktion sind schlecht. Deshalb verwendet man zur graphischen Darstellung häufig kubische Splines, um die Punkte zu verbinden. Auch hier wird ein Gleichungssystem für die Momente $s''(t_i)$ gelöst; dieses ist allerdings unabhängig von der Wahl der Stützstellen gut konditioniert.

Die Polynominterpolation ist grundlegend für die Konstruktion von Quadraturformeln, die Integrale stetiger Funktionen approximieren sollen. Führt man auf $[a, b]$ äquidistante Knoten $a = t_0 < \dots < t_n = b$ ein und ersetzt in $\int_a^b f(t) dt$ das f durch das Interpolationspolynom, so entsteht die n -te Newton-Cotes-Formel. Für $n = 1$ bzw. $n = 2$ nennt man sie Trapez- bzw. Simpsonregel.

Wegen der schlechten Konvergenzeigenschaft sind große n nicht geeignet; statt dessen unterteilt man $[a, b]$ zunächst in m Teilintervalle, um auf jedem Intervall eine Newton-Cotes-Formel zu verwenden. Dadurch entstehen Quadraturverfahren, die für $m \rightarrow \infty$ tatsächlich gegen das Integral konvergieren.

Benutzt man für paarweise disjunkte $t_0, \dots, t_n \in (a, b)$ die Hermite-Interpolation, d. h. $m_i = 1$, so entsteht eine Quadraturformel als Linearkombination von $f(t_0), \dots, f(t_n), f'(t_0), \dots, f'(t_n)$. Die Koeffizienten vor den Ableitungen können durch geeignete Wahl der t_0, \dots, t_n zu 0 gemacht werden. Dann entsteht die Gauß-Quadratur mit maximalem Exaktheitsgrad.

4 Nichtlineare Gleichungssysteme

Seien $D \subseteq \mathbb{R}^n$ und $F, G: D \rightarrow \mathbb{R}^n$ zwei Abbildungen. Dann interessieren wir uns für zwei Problemstellungen: Finde eine Nullstelle $\bar{x} \in D$ von F , d. h. $F(\bar{x}) = 0$ (Nullstellenproblem). Finde einen Fixpunkt $\bar{x} \in D$ von G , d. h. $G(\bar{x}) = \bar{x}$ (Fixpunktproblem). Wir werden sehen, dass sich die Probleme ineinander überführen lassen. Die auftretenden Fragestellungen beziehen sich auf die Existenz, die Eindeutigkeit sowie die Algorithmen zur numerischen Bestimmung solcher Nullstellen und Fixpunkte.

4.1 Fixpunktsätze und sukzessive Substitution

Seien $(X, \|\cdot\|_X)$ und $(Y, \|\cdot\|_Y)$ zwei normierte Räume und $D \subseteq X$. Eine Abbildung $G: D \rightarrow Y$ heißt Lipschitz-stetig (auf D), wenn eine Konstante $L > 0$, die Lipschitz-Konstante, derart existiert, dass

$$\|G(x_1) - G(x_2)\|_Y \leq L\|x_1 - x_2\|_X \quad \forall x_1, x_2 \in D$$

gilt. G heißt kontrahierend oder eine Kontraktion, wenn $L < 1$ möglich ist; L heißt dann auch Kontraktionskonstante.

Ist G Lipschitz-stetig, so ist G gleichmäßig stetig (wähle $\delta = \varepsilon/L > 0$ zu vorgegebenem $\varepsilon > 0$), d. h. insbesondere stetig. Bekanntermaßen ist die Wurzelfunktion $x \mapsto \sqrt{x}$ auf $[0, 1]$ stetig und auch gleichmäßig stetig, da $[0, 1]$ kompakt ist. Sie ist nicht Lipschitz-stetig auf $[0, 1]$, wohl aber auf jedem $[a, 1]$ mit $a > 0$. Sie ist kontrahierend auf jedem $[a, 1]$ mit $a > 1/4$.

Die Tatsache, ob eine Abbildung Lipschitz-stetig ist, ist unabhängig von der Wahl der Normen, wenn die Räume endlich-dimensional sind, wovon wir im Folgenden immer ausgehen; das liegt einfach an der Normäquivalenz. Der Wert von L jedoch hängt natürlich von der Norm ab, insbesondere die Fragestellung, ob $L < 1$ gewählt werden kann und die Abbildung daher kontraktiv ist.

Seien $(\mathbb{R}^n, \|\cdot\|_X)$ und $(\mathbb{R}^m, \|\cdot\|_Y)$ zwei normierte Räume und $D \subset \mathbb{R}^n$ konvex und kompakt (also abgeschlossen und beschränkt), und wir betrachten eine Abbildung $G \in \mathcal{C}^1(D, \mathbb{R}^m)$. Die Differenzierbarkeit von Abbildungen ist zunächst nur auf offenen Mengen definiert; wir definieren hier etwas salopp für abgeschlossenes $D \subseteq \mathbb{R}^n$

$$\mathcal{C}^1(D, \mathbb{R}^m) = \left\{ G \in \mathcal{C}^1(D^\circ, \mathbb{R}^m) \mid \partial_i G_k: D^\circ \rightarrow \mathbb{R}^m \text{ ist stetig fortsetzbar auf } \partial D \right\}.$$

Stetig fortsetzbar bedeutet: Es gibt eine stetige Funktion $F_{ik}: D \rightarrow \mathbb{R}$ mit $\partial_i G_k = F_{ik}|_{D^\circ}$.

(Wer es genauer wissen möchte: Diese Definition hinkt daran, dass D° leer sein kann: Man betrachte $\mathbb{Z} \times \mathbb{Z} \subset \mathbb{R}^2$ abgeschlossen oder auch die kompakte Einheitskreislinie $S^1 \subset \mathbb{R}^2$. Das Innere ist jeweils leer, und auf der ersten diskreten Menge möchte man sicherlich keine Differenzierbarkeit definieren. Die sinnvolle Forderung ist $D = \overline{D^\circ}$, d. h. die Menge ist der Abschluss ihrer inneren Punkte; dies ist bei den beiden Beispielen nicht der Fall. Sinnvollerweise definiert man also andersherum: Ist $\Omega \subseteq \mathbb{R}^n$ ein Gebiet (also offen und zusammenhängend), dann gehört eine Funktion $f: \Omega \rightarrow \mathbb{R}$ zu $\mathcal{C}^k(\Omega)$, wenn $f|_\Omega$ eine $\mathcal{C}^k(\Omega)$ -Funktion ist und alle Ableitungen von $f|_\Omega$ bis zur Ordnung k stetig auf Ω fortgesetzt werden können.)

Zurück zu $G \in \mathcal{C}^1(D, \mathbb{R}^m)$ mit $D \subset \mathbb{R}^n$ konvex und kompakt. Dann ist G Lipschitz-stetig, und für die Lipschitz-Konstante gilt

$$L = \max_{x \in D} \|G'(x)\|_{(X, Y)}.$$

Darin bezeichnet $G'(x) = J_G(x)$ die in Numerik häufig so geschriebene Ableitung (Jacobi-matrix) von G und $\|\cdot\|_{(X,Y)}$ die von den Normen $\|\cdot\|_X$ und $\|\cdot\|_Y$ induzierte Matrixnorm (lub-Norm). Der Beweis benutzt trivial den Schrankensatz aus AmV.

Seien $\|\cdot\|$ eine Norm auf dem \mathbb{R}^n , $D \subseteq \mathbb{R}^n$ und $G: D \rightarrow \mathbb{R}^n$. Ist dann D nicht-leer und abgeschlossen sowie G eine Selbstabbildung, d. h. $G(D) \subseteq D$, und eine Kontraktion bezüglich $\|\cdot\|$ mit Konstante $q < 1$, so gilt der Banachsche Fixpunktsatz, der sagt: G besitzt genau einen Fixpunkt $\bar{x} \in D$, also $G(\bar{x}) = \bar{x}$, und für jeden Startwert $x_0 \in D$ konvergiert die durch die Fixpunktiteration

$$x_{k+1} = G(x_k) \quad \forall k \in \mathbb{N}_0$$

definierte Folge $(x_k)_k$ in D gegen \bar{x} . Man spricht daher von globaler Konvergenz.

Es gelten die drei folgenden fundamental wichtigen Abschätzungen: Die Konvergenz der Folge $(x_k)_k$ ist monoton, d. h.

$$\|\bar{x} - x_k\| \leq q\|\bar{x} - x_{k-1}\| \quad \forall k \in \mathbb{N},$$

d. h. jedes Folgenglied liegt näher am Fixpunkt als das vorige. Die a-priori-Abschätzung

$$\|\bar{x} - x_k\| \leq \frac{q^k}{1-q}\|x_1 - x_0\| \quad \forall k \in \mathbb{N}$$

macht eine Aussage über den Fehler der k -ten Iterierten x_k , wenn nur die erste Iterierte x_1 berechnet wurde, während die a-posteriori-Abschätzung

$$\|\bar{x} - x_k\| \leq \frac{q}{1-q}\|x_k - x_{k-1}\| \quad \forall k \in \mathbb{N}$$

ihren Fehler schätzt, wenn sie schon berechnet wurde.

Als Beispiel seien $D = [0, 1]$ und $G: D \rightarrow \mathbb{R}$ gegeben durch $G(x) = x^3/10 + 1/2$. Dann ist D nicht-leer und abgeschlossen, und aus der Monotonie von G mit $G(0) = 1/2$ und $G(1) = 3/5$ folgt, dass G eine Selbstabbildung ist. Ferner gilt

$$|G(x) - G(y)| = \left| \frac{x^3}{10} + \frac{1}{2} - \frac{y^3}{10} - \frac{1}{2} \right| = \frac{|x^2 + xy + y^2|}{10}|x - y| \leq \frac{3}{10}|x - y|,$$

d. h. G ist eine Kontraktion mit Konstante $q = 3/10$. Man hätte auch $q = \sup_{x \in D} G'(x) = \sup_{x \in [0,1]} 3x^2/10 = 3/10$ rechnen können. Man erhält z. B. die Folge

$$x_0 = \frac{1}{2}, \quad x_1 = G(x_0) = \frac{41}{80}, \quad x_2 = G(x_1) = \frac{2\,628\,921}{5\,120\,000} \approx 0,513\,461\,133.$$

Die Funktion $G: [a, b] \rightarrow \mathbb{R}$ besitze einen Fixpunkt $\bar{x} \in (a, b)$ und sei in einer Umgebung von \bar{x} stetig differenzierbar. Ist dann $|G'(\bar{x})| < 1$, so gibt es ein $\varepsilon > 0$ mit der Eigenschaft: Die durch die Fixpunktiteration $x_k = G(x_{k-1})$ definierte Folge $(x_k)_k$ konvergiert für jeden Startpunkt $x_0 \in [\bar{x} - \varepsilon, \bar{x} + \varepsilon]$ gegen \bar{x} . Dieses Verhalten nennt man lokale Konvergenz.

Im gerade genannten Fall, d. h. wenn $|G'(\bar{x})| < 1$ gilt, heißt \bar{x} ein anziehender oder attraktiver Fixpunkt von G . Im entgegengesetzten Fall $|G'(\bar{x})| > 1$ nennt man \bar{x} abstoßend oder repulsiv.

Interessant ist das Verhalten der Fixpunktiteration für $G: [0, 1] \rightarrow \mathbb{R}$ mit $G(x) = ax(1-x)$, wobei $a \in \mathbb{R}$ ein Parameter ist. Man zeigt leicht, dass G genau dann eine Selbstabbildung ist, wenn $a \in [0, 4]$ gilt. Diese Gleichung wird logistische Gleichung oder Verhulst-Modell genannt. Die Fixpunktgleichung $G(\bar{x}) = \bar{x}$ hat die Lösungen $\bar{x} = 0$ und $\bar{x} = 1 - 1/a$, und

es gilt $G'(1 - 1/a) = 2 - a$. Für $a \geq 3$ oszilliert die Fixpunktiteration zwischen mehreren Häufungswerten, und es gibt auch Bereiche für a , in denen das Verhalten chaotisch ist.

Sei nun $\varrho(T)$ der Spektralradius einer Matrix $T \in \mathbb{R}^{n \times n}$. Dann gilt $\varrho(T) \leq \|T\|$ für jede Matrixnorm $\|\cdot\|$, die mit einer Vektornorm verträglich ist. Beweis: Sei $v \in \mathbb{C}^n$ ein normierter Eigenvektor zum Eigenwert $\lambda \in \mathbb{C}$ mit $|\lambda| = \varrho(T)$. Dann gilt

$$\varrho(T) = |\lambda| = |\lambda| \|v\| = \|\lambda v\| = \|Tv\| \leq \|T\| \|v\| = \|T\|.$$

Ferner gibt es im Fall $\varrho(T) < 1$ eine verträgliche Matrixnorm $\|\cdot\|$, so dass auch $\|T\| < 1$.

Die Abbildung $G: D \rightarrow \mathbb{R}^n$ mit $D \subseteq \mathbb{R}^n$ besitze einen Fixpunkt $\bar{x} \in D^\circ$ und sei in einer Umgebung von \bar{x} stetig differenzierbar. Ist dann $\varrho(G'(\bar{x})) < 1$, so gibt es ein $\varepsilon > 0$ mit der Eigenschaft: Die durch die Fixpunktiteration $x_k = G(x_{k-1})$ definierte Folge $(x_k)_k$ konvergiert für jeden Startpunkt $x_0 \in U_\varepsilon(\bar{x})$ gegen \bar{x} . Dies ist die lokale Konvergenz von Abbildungen in mehreren Variablen.

Wir haben nun genug über Voraussetzungen für die (in der Regel nur lokale) Konvergenz gegen einen Fixpunkt gesagt; numerisch ebenso wichtig ist die Konvergenzgeschwindigkeit. Wie verhält sich also der Fehler $\|x_k - \bar{x}\|$ für wachsendes k ? Dazu betrachten wir zunächst eine nicht-negative reelle Nullfolge $(a_k)_k$.

Wir nennen $p \geq 1$ die Mindestkonvergenzordnung von $(a_k)_k$, falls

$$\exists k_0 \in \mathbb{N} \quad \forall k \geq k_0 : \quad a_{k+1} \leq C a_k^p$$

mit einer Konstanten $C \in (0, 1)$ falls $p = 1$ bzw. $C > 0$ falls $p > 1$. Bei $p = 1$, $p = 2$ bzw. $p = 3$ sagen wir, die Folge konvergiere mindestens linear, quadratisch bzw. kubisch gegen 0.

Hat $(a_k)_k$ die Mindestkonvergenzordnung $p \geq 1$, dann definieren wir

$$Q_p := \limsup_{k \rightarrow \infty} \frac{a_{k+1}}{a_k^p} < \infty$$

und nennen dies die (asymptotische) Konvergenzrate von $(a_k)_k$. Gilt $Q_p = 0$, so sagen wir bei $p = 1$, $p = 2$ bzw. $p = 3$, die Folge konvergiere mindestens superlinear, superquadratisch bzw. superkubisch gegen 0.

Hat $(a_k)_k$ die Mindestkonvergenzordnung $p \geq 1$ mit $Q_p > 0$, so hat sie die Konvergenzordnung p . Bei $p = 1$, $p = 2$ bzw. $p = 3$ sagen wir, die Folge konvergiere linear, quadratisch bzw. kubisch gegen 0.

Wir werden dies auf ein Iterationsverfahren an und definieren: Eine Fixpunktiteration $x_k = G(x_{k-1})$ hat die Mindestkonvergenzordnung $p \geq 1$, falls alle Folgen $(\|x_k - \bar{x}\|)_k$, die mit einem Startwert x_0 gestartet wurden, für den $x_k \rightarrow \bar{x}$ gilt, die Mindestkonvergenzordnung p haben. Je größer p ist, desto schneller ist die Konvergenz. Bei gegebenem p ist die Konvergenz um so schneller, je kleiner Q_p ist. Damit erhalten wir sofort: Jede Iteration, die auf dem Banachschen Fixpunktsatz beruht, konvergiert mindestens lokal linear.

Allgemeiner gilt: Seien $p \in \mathbb{N}$, $G \in \mathcal{C}^p([a, b], \mathbb{R})$ und $\bar{x} \in (a, b)$ mit $G(\bar{x}) = \bar{x}$ ein Fixpunkt. Ist dann $|G'(\bar{x})| < 1$ falls $p = 1$ bzw. $G^{(i)}(\bar{x}) = 0$ für alle $i = 1, \dots, p-1$ und $G^{(p)}(\bar{x}) \neq 0$ falls $p \geq 2$, so hat das von G erzeugte Iterationsverfahren die lokale Konvergenzordnung p mit Konvergenzrate $Q_p = |G^{(p)}(\bar{x})|/p!$.

4.2 Newton-Verfahren und Varianten davon

Von einem gegebenen $F \in \mathcal{C}^1([a, b], \mathbb{R})$ suchen wir eine Nullstelle, also ein $\bar{x} \in [a, b]$ mit $F(\bar{x}) = 0$. Zu einem gegebenen $x_k \in [a, b]$ definieren wir x_{k+1} als Nullstelle der Tangente t_k

an F in x_k . Mit der Tangentengleichung $t_k(x) = F(x_k) + F'(x_k)(x - x_k)$ erhalten wir aus der Forderung $t_k(x_{k+1}) = 0$ das Newton-Verfahren

$$x_{k+1} = x_k - \frac{F(x_k)}{F'(x_k)} = G(x_k) \quad \forall k \in \mathbb{N}_0 \quad \text{mit} \quad G(x) = x - \frac{F(x)}{F'(x)}.$$

Darin ist x_0 eine geeignete Startnäherung.

Das Newton-Verfahren ist in der Praxis *das* Verfahren für nichtlineare Gleichungen. Dies ist in dem folgenden Satz begründet: Hat $F \in \mathcal{C}^2([a, b], \mathbb{R})$ eine Nullstelle $\bar{x} \in (a, b)$ und gilt $F'(x) \neq 0$ für alle $x \in [a, b]$, so konvergiert das Newton-Verfahren mindestens lokal quadratisch gegen \bar{x} , d. h. es gibt eine Konstante $C > 0$ und ein $\varepsilon > 0$, so dass

$$|x_{k+1} - \bar{x}| \leq C|x_k - \bar{x}|^2 \quad \forall x_k \in U_\varepsilon(\bar{x}) \cap [a, b].$$

Wird $F \in \mathcal{C}^2$ zu $F \in \mathcal{C}^1$ abgeschwächt, so konvergiert das Newton-Verfahren noch mindestens lokal superlinear.

Das Newton-Verfahren konvergiert sehr schnell, wenn die Startnäherung x_0 hinreichend gut war, und divergiert möglicherweise, wenn das x_0 ungünstig gewählt wurde. Zur Bestimmung von x_0 ist das Bisektionsverfahren nützlich: Seien $F \in \mathcal{C}([a, b], \mathbb{R})$ mit $F(a)F(b) < 0$; letzteres sichert die Existenz einer Nullstelle. Dann definieren wir $[a_0, b_0] = [a, b]$ und rekursiv

$$m_k = \frac{a_k + b_k}{2} \quad \text{und} \quad [a_{k+1}, b_{k+1}] = \begin{cases} [a_k, m_k] & \text{für } f(a_k)f(m_k) < 0, \\ [m_k, b_k] & \text{für } f(m_k)f(b_k) < 0 \end{cases} \quad \forall k \in \mathbb{N}_0.$$

Das Intervall wird also sukzessive halbiert und mit derjenigen Hälfte weitergearbeitet, die eine Nullstelle enthält.

Seien $F \in \mathcal{C}^1([a, b], \mathbb{R})$ mit einer Nullstelle $\bar{x} \in [a, b]$ und F' monoton, und sowohl der Startwert x_0 als auch die erste Iterierte $x_1 = x_0 - F(x_0)/F'(x_0)$ liege in $[a, b]$. Dann konvergiert das Newton-Verfahren ab der Iterierten x_1 monoton gegen \bar{x} , d. h. es gilt $\bar{x} \leq x_{k+1} \leq x_k$ oder $x_k \leq x_{k+1} \leq \bar{x}$ für alle $k \in \mathbb{N}$.

Definieren wir speziell $F(x) = x^m - a$ für $m \in \mathbb{N}$ und $a > 0$, so hat F eine einfache Nullstelle bei $\bar{x} = \sqrt[m]{a}$. Das auf diese Funktion angewandte Newton-Verfahren

$$x_{k+1} = \left(1 - \frac{1}{m}\right)x_k + \frac{a}{m}x_k^{1-m}$$

heißt dann Heron-Verfahren zur Approximation der m -ten Wurzel. Der Startwert kann auch problemlos komplex gewählt werden, um die anderen komplexen Nullstellen zu finden.

Seien nun $F \in \mathcal{C}^p([a, b], \mathbb{R})$ und $\bar{x} \in (a, b)$ eine p -fache Nullstelle von F , d. h. $F(\bar{x}) = F'(\bar{x}) = \dots = F^{(p-1)}(\bar{x}) = 0$ und $F^{(p)}(\bar{x}) \neq 0$. Dann konvergiert das Newton-Verfahren lokal linear gegen \bar{x} mit der Konvergenzrate $Q_1 = 1 - 1/p$. Für eine \mathcal{C}^{p+1} -Funktion F kann dieses Ergebnis verbessert werden, indem man zur Iteration

$$x_{k+1} = x_k - p \frac{F(x_k)}{F'(x_k)} \quad \forall k \in \mathbb{N}_0$$

übergeht. Dieses modifizierte Newton-Verfahren ist optimiert auf p -fache Nullstellen, denn es konvergiert wieder mindestens lokal quadratisch.

Das Auswerten der Ableitung kann verbilligt werden: Eine Möglichkeit ist, den Term $F'(x_k)$ durch $F'(x_0)$ zu ersetzen. Nach K Schritten wird dann mehrere Male $F'(x_K)$ benutzt, usw.

Eine andere Möglichkeit ist, die Ableitung in jedem Schritt durch einen Differenzenquotienten zu ersetzen. Dann erhält man das Sekantenverfahren

$$x_{k+1} = x_k - \frac{x_k \frac{F(x_k) - F(x_{k-1})}{x_k - x_{k-1}}}{F(x_k) - F(x_{k-1})} \quad \forall k \in \mathbb{N},$$

das zwei Startwerte x_0 und x_1 benötigt. Unter entsprechenden Voraussetzungen konvergiert es lokal mit der Ordnung des Goldenen Schnitts, also $(\sqrt{5} + 1)/2 \approx 1,62$.

4.3 Das Newton-Verfahren für Systeme nichtlinearer Gleichungen

Hier sei nun $D \subseteq \mathbb{R}^n$ und $F \in \mathcal{C}^1(D, \mathbb{R}^n)$. Wieder suchen wir eine Nullstelle, also ein $\bar{x} \in D$ mit $F(\bar{x}) = 0$. Die eindimensionale Formel motiviert sofort das mehrdimensionale Newton-Verfahren

$$x^{(k+1)} = x^{(k)} - F'(x^{(k)})^{-1} F(x^{(k)}) \quad \forall k \in \mathbb{N}_0,$$

worin $F'(x)$ die Ableitung (Jacobimatrix) von F in $x \in D$ und $x^{(0)} \in D$ ein Startwert ist. (Die Iterationsindizes schreiben wir hier oben in Klammern.)

Um die Matrixinversion zu vermeiden, löst man in der Regel zunächst das lineare Gleichungssystem

$$F'(x^{(k)}) \Delta x^{(k)} = -F(x^{(k)})$$

nach dem sog. Newton-Inkrement $\Delta x^{(k)} \in \mathbb{R}^n$ und aktualisiert dann

$$x^{(k+1)} = x^{(k)} + \Delta x^{(k)}.$$

Unter einer gewissen Lipschitz-artigen Bedingung an die Ableitung F' konvergiert auch das mehrdimensionale Newton-Verfahren mindestens lokal quadratisch gegen die Nullstelle.

4.4 Zusammenfassung

Seien $D \subset \mathbb{R}^n$ abgeschlossen und $G: D \rightarrow D$ eine Kontraktion mit Konstante $q < 1$ bezüglich einer Norm $\|\cdot\|$, d. h.

$$\|G(x_1) - G(x_2)\| \leq q \|x_1 - x_2\| \quad \forall x_1, x_2 \in D.$$

Dann sagt der Banachscher Fixpunktsatz, dass es genau ein $\bar{x} \in D$ mit $G(\bar{x}) = \bar{x}$ gibt und dass $G^k(x_0) \rightarrow \bar{x}$ für $k \rightarrow \infty$ für jeden Startpunkt $x_0 \in D$. Diese Konvergenz ist monoton, und die k -te Iterierte x_k erfüllt eine a-priori- und eine a-posteriori-Abschätzung.

Ist $G: [a, b] \rightarrow \mathbb{R}$ in einer Umgebung um einen Fixpunkt \bar{x} stetig differenzierbar, so ist \bar{x} anziehend, falls $|G'(\bar{x})| < 1$, und abstoßend, falls $|G'(\bar{x})| > 1$. Im ersten Fall liegt lokale Konvergenz vor. Für eine Abbildung $G: D \rightarrow D$ mit $D \subset \mathbb{R}^n$ heißt ein Fixpunkt \bar{x} anziehend, falls $\rho(G'(\bar{x})) < 1$ erfüllt ist.

$p \geq 1$ heißt Mindestkonvergenzordnung einer nicht-negativen Nullfolge $(a_k)_k$, falls $a_{k+1} \leq C a_k^p$ für hinreichend große k und einer Konstanten $C > 0$. (Bei $p = 1$ muss $C < 1$ sein.) $Q_p = \limsup a_{k+1}/a_k$ heißt die zugehörige Konvergenzrate. Damit kann die Konvergenz von $\|x_k - \hat{x}\|$ untersucht werden.

Das wichtigste Verfahren, um eine Nullstelle einer nichtlinearen Funktion F zu finden, ist das Newton-Verfahren. In einer Dimension lautet es

$$x_{k+1} = x_k - \frac{F(x_k)}{F'(x_k)}.$$

Bei einer einfachen Nullstelle konvergiert es unter schwachen Zusatzvoraussetzungen lokal quadratisch. Das ist sehr gut, problematisch hingegen kann das Auffinden einer geeigneten Startnäherung x_0 sein. In einer Dimension kann man dazu das Bisektionsverfahren verwenden.

Bei einer Abbildung $F: D \rightarrow \mathbb{R}^n$ lautet das Verfahren entsprechend

$$x^{(k+1)} = x^{(k)} - F'(x^{(k)})^{-1}F(x^{(k)}).$$

Es besitzt analoge Eigenschaften wie das in einer Dimension.

5 Lineare Gleichungssysteme II

5.1 Iterative Lösung linearer Gleichungssysteme

Wie in Kapitel 1 suchen wir eine Lösung $\bar{x} \in \mathbb{R}^n$ des linearen Gleichungssystems $Ax = b$ mit regulärer Matrix $A \in \mathbb{R}^{n \times n}$. Die Strategie ist nun aber: Konstruiere eine Abbildung $G: \mathbb{R}^n \rightarrow \mathbb{R}^n$ und spezifiziere die Menge der Startwerte $x^{(0)}$, so dass die durch die sukzessive Substitution $x^{(k+1)} = G(x^{(k)})$ definierte Folge $(x^{(k)})_k$ für $k \rightarrow \infty$ gegen \bar{x} konvergiert.

Wir wählen $G(x) = Tx + r$ mit $T \in \mathbb{R}^{n \times n}$ und $r \in \mathbb{R}^n$. Gilt dann $\rho(T) < 1$, so gibt es eine Norm mit $\|T\| < 1$, G ist damit eine Kontraktion, besitzt genau einen Fixpunkt, und die sukzessive Substitution konvergiert für jeden Startwert $x^{(0)}$ gegen diesen eindeutigen Fixpunkt. Die Konvergenz ist linear mit den Fehlerabschätzungen des Banachschen Fixpunktsatzes mit $q = \rho(T)$. Man beachte den fundamentalen Unterschied der globalen Konvergenz im Vergleich zur lokalen bei den nichtlinearen Gleichungen.

Selbstverständlich müssen wir T und r in Abhängigkeit von A und b so wählen, dass der Fixpunkt von G gerade \bar{x} ist. Es muss also gelten $\bar{x} = G(\bar{x}) = T\bar{x} + r$, d. h. $r = (I - T)\bar{x} = (I - T)A^{-1}b$; um r zu bestimmen, muss also die Matrix $(I - T)A^{-1}$ berechnet werden. Die andere Bedingung ist natürlich $\rho(T) < 1$. Beide Forderungen sind gewissermaßen gegenläufig, wie das folgende pathologische Beispiel demonstrieren soll: Wenn wir $T = 0$ wählen, dann ist $\rho(T) = 0 < 1$ bestmöglich erfüllt, allerdings gilt $(I - T)A^{-1} = A^{-1}$, d. h. zur Bestimmung von r muss A invertiert werden. Das ist aber sinnlos, weil dann das Gleichungssystem bereits gelöst wäre. Die Bedingung, dass r leicht berechnet werden kann, ist also schlechtestmöglich erfüllt.

Ein allgemeines Konzept ist die Zerlegung

$$A = \begin{pmatrix} a_{11} & & & \\ & \ddots & & \\ & & \ddots & \\ & & & a_{nn} \end{pmatrix} + \begin{pmatrix} 0 & & & \\ a_{21} & \ddots & & \\ \vdots & \ddots & \ddots & \\ a_{n1} & \cdots & a_{n,n-1} & 0 \end{pmatrix} + \begin{pmatrix} 0 & a_{12} & \cdots & a_{1n} \\ & \ddots & \ddots & \vdots \\ & & \ddots & a_{n-1,n} \\ & & & 0 \end{pmatrix} \\ =: D - E - F$$

in einen diagonalen, einen strikten linken unteren und einen strikten rechten oberen Anteil. Wir setzen im Folgenden voraus, dass D invertierbar ist, d. h. dass $a_{ii} \neq 0$ für $i = 1, \dots, n$.

Beim Gesamtschrittverfahren bzw. Jacobi-Verfahren schreiben wir

$$\begin{aligned} b = A\bar{x} &= (D - E - F)\bar{x} = D\bar{x} - (E + F)\bar{x} \implies D\bar{x} = (E + F)\bar{x} + b \\ \implies \bar{x} &= D^{-1}(E + F)\bar{x} + D^{-1}b = \mathcal{J}\bar{x} + r \implies x^{(k+1)} = \mathcal{J}x^{(k)} + r \end{aligned}$$

mit dem Gesamtschrittoperator $\mathcal{J} = D^{-1}(E + F)$ und $r = D^{-1}b$. Es gilt

$$\begin{aligned} (\mathcal{J})_{ij} &= \begin{cases} 0 & \text{falls } i = j, \\ -\frac{a_{ij}}{a_{ii}} & \text{sonst} \end{cases} \implies \|\mathcal{J}\|_1 = \max \left\{ \sum_{i=1, i \neq j}^n \frac{|a_{ij}|}{|a_{ii}|} \mid j = 1, \dots, n \right\} \\ &\implies \|\mathcal{J}\|_\infty = \max \left\{ \sum_{j=1, j \neq i}^n \frac{|a_{ij}|}{|a_{ii}|} \mid i = 1, \dots, n \right\}. \end{aligned}$$

Daraus folgt: $\|\mathcal{J}\|_1 < 1$ ist genau dann erfüllt, wenn A^T strikt diagonaldominant ist. Ähnlich ist $\|\mathcal{J}\|_\infty < 1$ ist genau dann erfüllt, wenn A strikt diagonaldominant ist.

In Komponentenschreibweise erhalten wir

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1, j \neq i}^n a_{ij} x_j^{(k)} \right), \quad i = 1, \dots, n.$$

Dies Formulierung motiviert die folgende Idee: Unter der Voraussetzung, dass das Gesamtschrittverfahren konvergiert, wird $x_i^{(k+1)}$ i. a. eine bessere Näherung für \bar{x}_i als $x_i^{(k)}$ sein. Verwendet man also zur Berechnung von $x_i^{(k+1)}$ für $j < i$ die neuen $x_j^{(k+1)}$ statt der $x_j^{(k)}$, so entsteht das komponentenweise Einzelschrittverfahren bzw. Gauß-Seidel-Verfahren

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij} x_j^{(k)} \right), \quad i = 1, \dots, n.$$

Auch wenn dies in der Regel in einer Schleife komponentenweise implementiert wird, ist die Matrixformulierung für die Theorie interessant. Es gilt

$$\begin{aligned} b &= A\bar{x} = (D - E - F)\bar{x} = (D - E)\bar{x} - F\bar{x} \implies (D - E)\bar{x} = F\bar{x} + b \\ \implies \bar{x} &= (D - E)^{-1}F\bar{x} + (D - E)^{-1}b = \mathcal{H}\bar{x} + r \implies x^{(k+1)} = \mathcal{H}x^{(k)} + r \end{aligned}$$

mit dem Einzelschrittoperator $\mathcal{H} = (D - E)^{-1}F$ und $r = (D - E)^{-1}b$.

Damit erhalten wir: Ist A strikt diagonaldominant, dann gilt für den Einzelschrittoperator \mathcal{H} und den Gesamtschrittoperator \mathcal{J} die Beziehung

$$\|\mathcal{H}\|_\infty \leq \|\mathcal{J}\|_\infty < 1,$$

d. h. beide Verfahren sind global konvergent. Häufig konvergiert das Einzelschrittverfahren tatsächlich schneller, allerdings wird die Konvergenzrate der Verfahren durch $\rho(\mathcal{H})$ und $\rho(\mathcal{J})$ bestimmt, zwischen denen keine allgemeine Ungleichung gilt.

Ein wesentlicher Vorteil des Gauß-Seidel-Verfahrens gegenüber dem Jacobi-Verfahren zeigt sich im folgenden Satz: Ist A positiv definit, so folgt $\rho(\mathcal{H}) < 1$, d. h. für das Gauß-Seidel-Verfahren liegt globale Konvergenz vor. Für das Jacobi-Verfahren gilt keine analoge Aussage.

Beide Verfahren können relaxiert werden, was die Konvergenzgeschwindigkeit erhöhen kann. Für ein $\omega \in [0, 1]$ setzt man

$$x_i^{(k+1)} = (1 - \omega)x_i^{(k)} + \frac{\omega}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij} x_j^{(k)} \right), \quad i = 1, \dots, n$$

für das relaxierte Einzelschrittverfahren und

$$x_i^{(k+1)} = (1 - \omega)x_i^{(k)} + \frac{\omega}{a_{ii}} \left(b_i - \sum_{j=1, j \neq i}^n a_{ij}x_j^{(k)} \right), \quad i = 1, \dots, n$$

für das relaxierte Gesamtschrittverfahren. Für $\omega = 1$ entsteht das ursprüngliche Verfahren, für $\omega = 0$ bleibt man konstant bei $x^{(0)}$ stehen (sinnlos).

Abschließend soll ein ganz anderer Zugang diskutiert werden, um $Ax = b$ zu lösen. Für reelle Daten A und b definieren wir zunächst das Funktional $f: \mathbb{R}^n \rightarrow \mathbb{R}$ durch

$$f(x) = \frac{1}{2}x^T Ax - b^T x.$$

Als Polynom zweiten Grades in x_1, \dots, x_n ist f beliebig oft stetig differenzierbar, und es gilt

$$\begin{aligned} \frac{\partial f(x)}{\partial x_i} &= \frac{\partial}{\partial x_i} \left(\frac{1}{2} \sum_{j,k=1}^n a_{jk}x_jx_k - \sum_{j=1}^n b_jx_j \right) = \frac{1}{2} \sum_{j,k=1}^n a_{jk}\delta_{ij}x_k + \frac{1}{2} \sum_{j,k=1}^n a_{jk}x_j\delta_{ik} - \sum_{j=1}^n b_j\delta_{ij} \\ &= \frac{1}{2} \sum_{k=1}^n a_{ik}x_k + \frac{1}{2} \sum_{j=1}^n a_{ji}x_j - b_i = \frac{1}{2}(Ax)_i + \frac{1}{2}(A^T x)_i - b_i, \end{aligned}$$

d. h.

$$\nabla f(x) = \frac{1}{2}(A + A^T)x - b.$$

Gilt nun $A = A^T$, d. h. ist A symmetrisch, so folgt daraus $\nabla f(x) = Ax - b$, und damit besitzt f genau einen kritischen Punkt, nämlich die eindeutige Lösung des Gleichungssystems. Die Hessematrix von f , also die Jacobimatrix von ∇f , ist in diesem Fall gerade A , d. h. f besitzt ein Extremum, falls A (positiv oder negativ) definit ist.

Ist also $A \in \mathbb{R}^{n \times n}$ symmetrisch und positiv definit, so kann das Gleichungssystem $Ax = b$ gelöst werden, indem das angegebene Funktional f minimiert wird. Die Idee, das Minimum von f zu finden, ist die folgende: Zu einem k -ten Iterationspunkt $x^{(k)}$ wird eine geeignete Richtung $p^{(k)}$ gewählt und eine Schrittweite t_k derart bestimmt, dass die Funktion $\varphi_k: \mathbb{R} \rightarrow \mathbb{R}$ mit $\varphi_k(t) = f(x^{(k)} + tp^{(k)})$ in t_k ihr Minimum annimmt. Dann ist $x^{(k+1)} = x^{(k)} + t_k p^{(k)}$ der neue Iterationspunkt, zu dem eine neue geeignete Richtung gewählt werden muss, die im günstigsten Fall linear unabhängig zu den vorherigen ist.

Um das t_k zu bestimmen, rechnen wir mit der Kettenregel

$$\begin{aligned} 0 &\stackrel{!}{=} \varphi_k'(t_k) = J_f(x^{(k)} + t_k p^{(k)}) J_{t \rightarrow x^{(k)} + t p^{(k)}}(t_k) = \left(A(x^{(k)} + t_k p^{(k)}) - b \right)^T p^{(k)} \\ &= (Ax^{(k)} - b)^T p^{(k)} + (Ap^{(k)})^T p^{(k)} t_k \implies t_k = -\frac{p^{(k)T} \nabla f(x^{(k)})}{p^{(k)T} Ap^{(k)}}. \end{aligned}$$

Man beachte, dass der Nenner nicht null werden kann, weil A positiv definit ist. Die Bestimmung der optimalen Suchrichtungen ist schwieriger. Eine Möglichkeit ist, $p^{(k)} = -\nabla f(x^{(k)})$ zu wählen, d. h. jedesmal den steilsten Abstieg. Dies führt auf das folgende Gradientenverfahren: Für einen gegebenen Startvektor $x^{(0)} \in \mathbb{R}^n$ iteriere über $k \in \mathbb{N}_0$ die Anweisungen

$$r^{(k)} = Ax^{(k)} - b, \quad p^{(k)} = -r^{(k)}, \quad t_k = -\frac{p^{(k)T} r^{(k)}}{p^{(k)T} Ap^{(k)}}, \quad x^{(k+1)} = x^{(k)} + t_k p^{(k)}.$$

Hier heißt $r^{(k)}$ das k -te Residuum. Natürlich kann in diesem Algorithmus entweder p oder r eliminiert werden; er dient lediglich der Vorbereitung für den nächsten Algorithmus.

Man rechnet leicht nach, dass das Residuum $r^{(k)}$ im k -ten Schritt senkrecht auf der Suchrichtung $p^{(k-1)}$ des vorherigen Schritt steht, und zwar

$$\begin{aligned} p^{(k-1)\text{T}} r^{(k)} &= p^{(k-1)\text{T}} (Ax^{(k)} - b) = p^{(k-1)\text{T}} (Ax^{(k-1)} + t_{k-1} Ap^{(k-1)} - b) \\ &= p^{(k-1)\text{T}} \left(r^{(k-1)} - \frac{p^{(k-1)\text{T}} r^{(k-1)}}{p^{(k-1)\text{T}} Ap^{(k-1)}} Ap^{(k-1)} \right) \\ &= p^{(k-1)\text{T}} r^{(k-1)} - \frac{p^{(k-1)\text{T}} r^{(k-1)}}{p^{(k-1)\text{T}} Ap^{(k-1)}} p^{(k-1)\text{T}} Ap^{(k-1)} = 0. \end{aligned}$$

Diese Eigenschaft ist unabhängig von der Art und Weise, wie $p^{(k-1)}$ gewählt wurde! Sie ist vielmehr das Ergebnis der Wahl von t_{k-1} .

Die neue Idee ist, die weiteren Suchrichtungen nach $p^{(0)} = -r^{(0)}$ derart zu wählen, dass $p^{(k+1)}$ eine Linearkombination von $r^{(k+1)}$ und $p^{(k)}$ ist. Die Linearkombination soll so geartet sein, dass sie $f(x^{(k+1)} + tp^{(k+1)})$ minimiert. Dazu definieren wir $\chi_k: \mathbb{R} \rightarrow \mathbb{R}$ durch

$$\chi_k(\beta) = f\left(x^{(k+1)} + t(-r^{(k+1)} + \beta p^{(k)})\right)$$

und minimieren diese. Wir erhalten wieder mit der Kettenregel

$$\begin{aligned} 0 &\stackrel{!}{=} \chi'_k(\beta_k) = J_f\left(x^{(k+1)} + t(-r^{(k+1)} + \beta_k p^{(k)})\right) J_{\beta \rightarrow x^{(k+1)} + t(-r^{(k+1)} + \beta p^{(k)})}(\beta_k) \\ &= \left(A\left(x^{(k+1)} + t(-r^{(k+1)} + \beta_k p^{(k)})\right) - b\right)^{\text{T}} t p^{(k)} \\ &= t\left(Ax^{(k+1)} - b - tAr^{(k+1)} + t\beta_k Ap^{(k)}\right)^{\text{T}} p^{(k)} = t^2\left(-Ar^{(k+1)} + \beta_k Ap^{(k)}\right)^{\text{T}} p^{(k)}, \end{aligned}$$

da $Ax^{(k+1)} - b = r^{(k+1)} \perp p^{(k)}$. Auflösen ergibt schließlich

$$\beta_k = \frac{p^{(k)\text{T}} Ar^{(k+1)}}{p^{(k)\text{T}} Ap^{(k)}}.$$

Für das so konstruierte $p^{(k+1)}$ gilt ferner

$$p^{(k)\text{T}} Ap^{(k+1)} = p^{(k)\text{T}} A\left(-r^{(k+1)} + \frac{p^{(k)\text{T}} Ar^{(k+1)}}{p^{(k)\text{T}} Ap^{(k)}} p^{(k)}\right) = 0.$$

Weil A symmetrisch und positiv definit ist, sind die Abbildungen

$$\langle u, v \rangle_A = u^{\text{T}} Av \quad \text{bzw.} \quad \|u\|_A = \sqrt{\langle u, u \rangle_A}$$

ein Skalarprodukt bzw. eine Norm. Wir nennen u und v zueinander A -orthogonal oder A -konjugiert, wenn $\langle u, v \rangle_A = 0$ gilt, und schreiben $u \perp_A v$.

Mit dieser Vorarbeit formulieren wir nun das CG-Verfahren zum Lösen eines Gleichungssystems mit symmetrischer und positiv definiten Matrix $A \in \mathbb{R}^{n \times n}$. Für einen gegebenen Startvektor $x^{(0)} \in \mathbb{R}^n$ setze

$$r^{(0)} = Ax^{(0)} - b, \quad p^{(0)} = -r^{(0)}$$

und iteriere dann über $k \in \mathbb{N}_0$ die Anweisungen

$$\begin{aligned} t_k &= \frac{\|r^{(k)}\|_2^2}{\|p^{(k)}\|_A^2}, & x^{(k+1)} &= x^{(k)} + t_k p^{(k)}, & r^{(k+1)} &= r^{(k)} + t_k A p^{(k)}, \\ \beta_k &= \frac{\|r^{(k+1)}\|_2^2}{\|r^{(k)}\|_2^2}, & p^{(k+1)} &= -r^{(k+1)} + \beta_k p^{(k)}, \end{aligned}$$

bis das Residuum verschwindet (oder hinreichend klein ist).

Im Folgenden müssen wir noch die von der Herleitung abweichenden Formeln für t_k , $r^{(k+1)}$ und β_k begründen. Für die erste gilt klarerweise $\|r^{(0)}\|^2 = -p^{(0)\text{T}} r^{(0)}$ nach Konstruktion, und für $k \in \mathbb{N}$ folgt

$$\|r^{(k)}\|_2^2 = r^{(k)\text{T}} r^{(k)} = (\beta_{k-1} p^{(k-1)} - p^{(k)})^{\text{T}} r^{(k)} = -p^{(k)\text{T}} r^{(k)}$$

wegen $p^{(k-1)} \perp r^{(k)}$. Für die zweite gilt

$$r^{(k+1)} = Ax^{(k+1)} - b = A(x^{(k)} + t_k p^{(k)}) - b = Ax^{(k)} - b + t_k A p^{(k)} = r^{(k)} + t_k A p^{(k)}.$$

Für die dritte zeigen wir zunächst $r^{(k)\text{T}} A p^{(k)} = -\|p^{(k)}\|_A^2$: Für $k = 0$ gilt dies nach Konstruktion, für $k \in \mathbb{N}$ folgt

$$r^{(k)\text{T}} A p^{(k)} = (\beta_{k-1} p^{(k-1)} - p^{(k)})^{\text{T}} A p^{(k)} = -\|p^{(k)}\|_A^2$$

wegen $p^{(k-1)} \perp_A p^{(k)}$. Damit ergibt sich

$$r^{(k)\text{T}} r^{(k+1)} = r^{(k)\text{T}} (r^{(k)} + t_k A p^{(k)}) = \|r^{(k)}\|_2^2 + \frac{\|r^{(k)}\|_2^2}{\|p^{(k)}\|_A^2} r^{(k)\text{T}} A p^{(k)} = 0,$$

d. h. $r^{(k)} \perp r^{(k+1)}$. Daraus folgt dann mit

$$\frac{r^{(k+1)\text{T}} r^{(k+1)}}{\|r^{(k)}\|_2^2} = \frac{(r^{(k)} + t_k A p^{(k)})^{\text{T}} r^{(k+1)}}{t_k \|p^{(k)}\|_A^2} = \frac{r^{(k)\text{T}} r^{(k+1)} + t_k p^{(k)\text{T}} A r^{(k+1)}}{t_k \|p^{(k)}\|_A^2} = \beta_k$$

die dritte Formel.

Mit einer geeigneten Induktion kann man noch die Orthogonalitäten

$$p^{(i)} \perp r^{(k)}, \quad r^{(i)} \perp r^{(k)}, \quad p^{(i)} \perp_A p^{(k)}$$

für $i < k$ zeigen. Insbesondere steht also $r^{(n)}$ orthogonal auf allen $r^{(0)}, \dots, r^{(n-1)}$, wovon sich der Name CG = conjugate gradients ableitet. Da diese aber auch paarweise aufeinander orthogonal stehen, folgt: Entweder ist ein $r^{(i)} = 0$ für $i \in \{0, \dots, n-1\}$, oder aber die $r^{(0)}, \dots, r^{(n-1)}$ bilden eine Orthogonalbasis des \mathbb{R}^n und $r^{(n)} = 0$. Ein verschwindendes Residuum bedeutet aber, dass das Minimum gefunden wurde. Das CG-Verfahren findet also nach höchstens n Schritten die Lösung des Gleichungssystems und ist daher streng genommen kein Iterationsverfahren. Bei exakter Arithmetik konvergiert es nicht gegen die Lösung, sondern liefert sie spätestens nach dem n -ten Schritt exakt.

5.2 Die QR-Zerlegung

Wir nennen die Vektoren $q_1, \dots, q_n \in \mathbb{C}^m$ ein Orthonormalsystem, wenn

$$\langle q_i, q_j \rangle = q_i^* q_j = \delta_{ij} \quad \forall 1 \leq i, j \leq n.$$

Für die Matrix $Q = (q_1, \dots, q_n)$ gilt dann $Q^*Q = I \in \mathbb{C}^{n \times n}$, und für $m = n$ heißt Q unitär. Die Menge der unitären Matrizen bildet bezüglich Produkt eine Gruppe, und für unitäres U gilt

$$\|U\|_2 = \sqrt{\varrho(U^*U)} = \sqrt{\varrho(I)} = 1,$$

also auch $\text{cond}_2(U) = 1$, und U ist eine Isometrie, d. h.

$$\|Ux\|_2 = \langle Ux, Ux \rangle = \langle x, U^*Ux \rangle = \langle x, x \rangle = \|x\|_2.$$

Für $m \geq n$ sei $A \in \mathbb{C}^{m \times n}$ mit maximalem Rang, d. h. $\text{Rang}(A) = n$. Das Paar (Q, R) mit $A = QR$ nennen wir eine QR-Zerlegung von A , wenn die Spalten von $Q \in \mathbb{C}^{m \times n}$ ein Orthonormalsystem bilden und $R \in \mathbb{C}^{n \times n}$ eine rechte obere Dreiecksmatrix mit positiver Diagonale ist. Unter den angegebenen Voraussetzungen existiert diese Zerlegung und ist eindeutig bestimmt. Gelegentlich wird auch $Q \in \mathbb{C}^{m \times m}$ unitär und $R \in \mathbb{C}^{m \times n}$ gefordert. Diese Version kann man leicht konstruieren, indem man die Spalten von Q zu einer Orthonormalbasis des \mathbb{C}^m ergänzt und bei R Nullzeilen anfügt.

Die Matrix $A^*A \in \mathbb{C}^{n \times n}$ ist hermitesch und wegen der Rangbedingung positiv definit, so dass eine Cholesky-Zerlegung $A^*A = C^*C$ existiert. Dann erfüllen $Q = AC^{-1}$ und $R = C$ die Forderungen an die QR-Zerlegung. Das Zerlegen von A^*A ist wegen $\text{cond}_2(A^*A) = \text{cond}_2(A)^2$ jedoch schlechter konditioniert als das Zerlegen von A . Daher andere Vorgehensweise:

Für einen normierten Vektor $v \in \mathbb{C}^m$ nennen wir die Matrix

$$S_v = I - 2vv^* \in \mathbb{C}^{m \times m}$$

die Householder-Transformation zu v . Sie ist hermitesch und unitär, d. h. $S_v = S_v^* = S_v^{-1}$. Ist (v, u_2, \dots, u_m) eine Orthonormalbasis von \mathbb{C}^m , so gilt $S_v v = -v$ und $S_v u_i = u_i$, d. h. S_v hat einmal den Eigenwert -1 und $(m-1)$ -mal den Eigenwert 1 . S_v beschreibt eine Spiegelung entlang v .

Das Householder-Verfahren spiegelt sukzessive die Spalten von A derart, dass die Einträge der k -ten Spalte unterhalb des k -ten Eintrags zu 0 werden. Zuerst wird also ein Vektor v_1 gewählt, so dass die erste Spalte von $S_{v_1}A$ ein Vielfaches von e_1 ist, danach ein Vektor v_2 , so dass die zweite Spalte von $S_{v_2}S_{v_1}A$ eine Linearkombination von v_1, v_2 ist. Nach n Schritten ergibt sich

$$S_{v_n} \cdots S_{v_1} A = R.$$

Dies können wir in $A = QR$ mit $Q = (S_{v_n} \cdots S_{v_1})^*$ umschreiben.

Dieser Householder-Algorithmus ist stabil, wenn man ihn geeignet implementiert. Ist z. B. $a_1 \in \mathbb{C}^m$ die erste Spalte von A , so müssen wir $v_1 \in \mathbb{C}^m$ so wählen, dass

$$S_{v_1} a_1 = (I - 2v_1 v_1^*) a_1 = a_1 - 2(v_1^* a_1) v_1 = \kappa_1 e_1$$

für ein $\kappa_1 \in \mathbb{C}$ gilt. Anwenden der Norm liefert $\kappa_1 = \zeta_1 \|a_1\|_2$ mit $|\zeta_1| = 1$. Nachrechnen ergibt, dass für $\zeta = \pm 1$

$$v_1 = \alpha_1 (a_1 \mp \|a_1\|_2 e_1)$$

mit einer Normierungskonstanten α_1 gewählt werden kann. Das Rechenzeichen in „ \mp “ ist nur für die erste Komponente interessant. Um Auslöschung bei $a_{11} \mp \|a_1\|_2$ zu vermeiden, wählen wir dasjenige Rechenzeichen, so dass a_{11} und $\mp \|a_1\|_2$ dasselbe Vorzeichen haben.

Im k -ten Schritt multiplizieren wir an die Matrix $S_{v_{k-1}} \cdots S_{v_1} A$ eine Householder-Transformation der Form

$$S_{v_k} = \begin{pmatrix} I_{k-1} & 0 \\ 0 & S_{\tilde{v}_k} \end{pmatrix} \quad \text{mit} \quad \tilde{v}_k \in \mathbb{C}^{m-k+1},$$

die nur noch auf der Restmatrix operiert.

Die QR -Zerlegung kann bei $m = n$ auch zum Lösen des Gleichungssystems $Ax = b$ benutzt werden. Aus $Rx = Q^*b$ kann durch Rückwärtssubstitution das x bestimmt werden. Q^*b berechnet man einfach während der Iteration mit. Der numerische Vorteil gegenüber der Gauß-Elimination ist, dass die unitären Transformationen die Kondition der Matrizen erhalten, denn $\text{cond}_2(R) = \text{cond}_2(S_{v_{n-1}} \cdots S_{v_1} A) = \text{cond}_2(A)$. Damit ist der Lösungsalgorithmus mit der QR -Zerlegung stabiler und kann auch für schlecht konditionierte Gleichungssysteme benutzt werden, bei denen die Gauß-Elimination versagt.

Ist hingegen $m > n$, so ist das Gleichungssystem $Ax = b$ überbestimmt und hat im Allgemeinen keine Lösung. Dennoch liefert die QR -Zerlegung ein „sinnvolles“ $\bar{x} \in \mathbb{R}^n$, nämlich dasjenige, das die Optimierungsaufgabe

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2$$

löst. Dieses ist durch $\bar{x} = R^{-1}Q_1^*b$ gegeben, wobei in Q_1^* und R_1 die ersten n Zeilen von Q^* und R stehen.

Ist $m < n$, so hat das Gleichungssystem $Ax = b$ „wahrscheinlich“ viele Lösungen, und man kann nach der Lösung mit kleinster Euklidnorm fragen, also nach der Lösung von

$$\min_{x \in \mathbb{R}^n} \|x\|_2 \quad \text{unter der Nebenbedingung} \quad Ax = b$$

fragen. Hat A maximalen Rang, also $\text{Rang}(A) = m$, so ist die Minimalstelle eindeutig durch $\bar{x} = A^*(AA^*)^{-1}b$ gegeben. Auch dies kann numerisch stabiler mit der QR -Zerlegung (diesmal von A^*) berechnet, und es gilt $\bar{x} = Q_1 R_1^{-*} b$, wobei in Q_1 und R_1^* die ersten m Spalten von Q und R^* stehen.

5.3 Singulärwerte und die Pseudoinverse einer Matrix

Nun seien $m, n \in \mathbb{N}$ und $A \in \mathbb{R}^{m \times n}$ völlig beliebig. Dann existieren unitäre Matrizen $U = (u_1, \dots, u_m) \in \mathbb{R}^{m \times m}$, $V = (v_1, \dots, v_n) \in \mathbb{R}^{n \times n}$ und eine Matrix $\Sigma \in \mathbb{R}^{m \times n}$, so dass $A = U \Sigma V^T$ und: Σ ist die Nullmatrix bis auf $(\Sigma)_{ii} = \sigma_i$ für $i = 1, \dots, p = \min\{m, n\}$ mit $\sigma_1 \geq \dots \geq \sigma_p \geq 0$. Diese Zerlegung heißt eine Singulärwertzerlegung von A , die $\sigma_1, \dots, \sigma_p$ heißen die Singulärwerte, und u_i bzw. v_i heißt ein i -ter Links- bzw. Rechts-Singulärvektor.

Dann gilt:

- $Av_i = \sigma_i u_i$ und $A^T u_i = \sigma_i v_i$ für alle $i = 1, \dots, p$.
- AA^T besitzt als Eigenwerte die σ_i^2 und $(m - p)$ -mal die 0.
- $A^T A$ besitzt als Eigenwerte die σ_i^2 und $(n - p)$ -mal die 0.

Für die Singulärwerte gelte $\sigma_1 \geq \dots \geq \sigma_r > \sigma_{r+1} = \dots = \sigma_p = 0$. Dann hat A Rang r , und es gilt weiter:

- Bild $A = \text{span}\{u_1, \dots, u_r\}$ und $(\text{Bild } A)^\perp = \text{span}\{u_{r+1}, \dots, u_m\}$.
- Kern $A = \text{span}\{v_{r+1}, \dots, v_n\}$ und $(\text{Kern } A)^\perp = \text{span}\{v_1, \dots, v_r\}$.
- $A = \sum_{i=1}^r \sigma_i u_i v_i^T$.
- $\|A\|_2^2 = \sigma_1^2$ und $\|A\|_F^2 = \sum_{i=1}^r \sigma_i^2$.

Auch mit der Singulärwertzerlegung kann das Minimierungsproblem $\min_{x \in \mathbb{R}^n} \|Ax - b\|_2$ gelöst werden, und zwar unabhängig davon, ob $m = n$, $m > n$ oder $m < n$. Nach etwas Rechenarbeit, die wir hier auslassen wollen, kann man zeigen, dass das minimierende $\bar{x} \in \mathbb{R}^n$ immer durch $\bar{x} = A^\dagger b$ gegeben ist. A^\dagger ist dabei eine sog. Pseudoinverse von A .

Zu jeder Matrix $A \in \mathbb{R}^{m \times n}$ gibt es genau eine Matrix $X \in \mathbb{R}^{n \times m}$, die die vier Moore-Penrose-Bedingungen

$$AXA = A, \quad XAX = X, \quad (AX)^T = AX, \quad (XA)^T = XA$$

erfüllt. Diese Lösung wird mit A^\dagger bezeichnet und heißt die Moore-Penrose-Pseudoinverse zu A . Sie ist eine Verallgemeinerung der gewöhnlichen Inversen, denn ist A quadratisch und regulär, so gilt $A^\dagger = A^{-1}$.

Diese Pseudoinverse kann mit der Singulärwertzerlegung bestimmt werden. Ist nämlich $A = U\Sigma V^T$, so gilt $A^\dagger = V\Sigma^\dagger U^T$. Darin entsteht Σ^\dagger , indem alle positiven Einträge von Σ durch ihre Kehrwerte ersetzt werden. Es gibt allerdings auch noch andere Verfahren, um die Pseudoinverse zu berechnen.

5.4 Zusammenfassung

Ein lineares Gleichungssystem $Ax = b$ mit eindeutiger Lösung \bar{x} kann ebenfalls als Fixpunktgleichung geschrieben werden. Wir zerlegen $A = D - E - F$ mit Diagonalmatrix D , strikter unterer Dreiecksmatrix E und strikter oberer Dreiecksmatrix F . Dann lautet das Jacobi-Verfahren $x^{(k+1)} = \mathcal{J}x + D^{-1}b$ mit Gesamtschrittoperator $\mathcal{J} = D^{-1}(E + F)$ und das Gauß-Seidel-Verfahren $x^{(k+1)} = \mathcal{H}x + (D - E)^{-1}b$ mit dem Einzelschrittoperator $\mathcal{H} = (D - E)^{-1}F$.

Damit D bzw. $D - E$ invertierbar ist, müssen die Diagonalelemente von A ungleich 0 sein. Strikt diagonaldominante Matrizen erfüllen diese Voraussetzung, und diese Eigenschaft ist sogar hinreichend für die Konvergenz beider Verfahren.

Das CG-Verfahren ist das wohl beste Verfahren für symmetrische positiv definite Matrizen. Mit einem beliebigen Vektor $x^{(0)}$ startend, minimiert es sukzessive das Funktional $f(x) = x^T Ax / 2 + b^T x$, dessen Minimalstelle die gesuchte Lösung ist. Sind $p^{(0)}, \dots, p^{(k-1)}$ die ersten k Suchrichtungen, die zueinander A -konjugiert sind, dann ist die k -te Iterierte $x^{(k)}$ die Minimalstelle von $f|_{K_k}$, wobei $K_k = x^{(0)} + \text{span}(p^{(0)}, \dots, p^{(k-1)})$ ein affiner Raum ist. Die Lösung wird nach spätestens n Schritten gefunden, da $K_n = \mathbb{R}^n$.

Der Vorteil des CG-Verfahrens ist neben der einfachen Implementation die Tatsache, dass die Matrix A nicht verändert wird. Die Matrix-Vektor-Produkte sind also insbesondere dann sehr billig, wenn A schwach besetzt ist.

Der Householder-Algorithmus bestimmt die QR -Zerlegung einer Matrix $A \in \mathbb{C}^{m \times n}$ mit Rang n , indem er unitäre Matrizen sukzessive an A heranmultipliziert. Dadurch entsteht die rechte obere Dreiecksmatrix R . Die Zerlegung besitzt drei Anwendungen: Ist $m = n$, dann kann das Gleichungssystem $Ax = b$ auch mit schlecht konditionierter Matrix A gelöst werden. Ist $m > n$, dann kann eine sog. Kleinste-Quadrate-Lösung \bar{x} von $Ax = b$ bestimmt werden,

die $\|Ax - b\|_2$ minimiert. Ferner kann die diejenige Lösung von $A^*x = b$ berechnet werden, die die Euklidnorm $\|x\|_2$ minimiert.

Eine Singulärwertzerlegung von $A \in \mathbb{R}^{m \times n}$ ist von der Form $A = U\Sigma V^T$, wobei U und V unitär sind und Σ eine Diagonalmatrix mit nichtnegativen Einträgen, den Singulärwerten von A , ist. Mit ihr kann eine Pseudoinverse $A^\dagger = V\Sigma^\dagger U^T$ definiert werden. Dann löst $\bar{x} = A^\dagger b$ das Minimierungsproblem $\min_{x \in \mathbb{R}^n} \|Ax - b\|_2$.