Numerik

Hinweis : Mit der Angabe „Buch“ ist das Buch „Numerische Mathematik I“ von Peter Deuflhard gemeint.

TODO :
Pivotstrategien : Beispiel aus dem Buch S.10/11 einfügen ?
Nachiteration Kapitel 1 : Beispiel rechnen für Genauigkeitsverbesserung

Grundlagen

Zeichen :
$\doteq$ „bis auf Terme niederer Ordnung“

Def. „hinreichende Bedingung“ , „notwendige Bedingung“ :

Bsp. : A : n ist durch 4 teilbar B : n ist durch 2 teilbar

$A \Rightarrow B$ $\Leftrightarrow$ A ist hinreichende Bedingung für B

$\Leftrightarrow$ B ist notwendige Bedingung für A

Def. überbestimmtes / unterbestimmtes Gleichungssystem :

Ein Gleichungssystem mit m Gleichungen und n Unbekannten heißt unterbestimmt wenn m < m, quadratisch, wenn M = n und überbestimmt, wenn m > n ist.

Def. Matrix : $A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$

Bei Operationen auf den Zeilen einer Matrix (z.B. Elimination) Multiplikation von links.
Bei Operationen auf den Spalten einer Matrix Multiplikation von rechts.

Def. reguläre Matrix : $A \in R^{(n \times n)}$ ist regulär $\Leftrightarrow$ $Rang(A) \neq 0$ $\Leftrightarrow$ $\det(A) \neq 0$

Def. singuläre Matrix : $A \in R^{(n \times n)}$ ist singulär $\Leftrightarrow$ $Rang(A) = 0$ $\Leftrightarrow$ $\det(A) = 0$

$\Leftrightarrow$ Gleichungssystem ist nicht eindeutig lösbar
(Eine nicht-singuläre Matrix nennt man reguläre Matrix.)
(Eine singuläre Matrix besitzt keine Inverse.)

Def. normale Matrix : Die Matrix A heißt normal $\Leftrightarrow$ $A\overline{A}^T = \overline{A}^T A$

Def. unipotente Matrix : Diagonalelemente gleich eins

Def. positiv definite Matrix :

Eine symmetrische Matrix $A = A^T$ ist genau dann positiv definit, falls

$\langle x, Ax \rangle > 0$ für alle $x \neq 0$ ($\Leftrightarrow$ alle Hauptunterdeterminanten sind positiv)

Def. strikt diagonaldominante Matrix :

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

Def. konjugierte (ähnliche) Matrizen :

Zwei quadratische Matrizen A, B heißen konjugiert (ähnlich), wenn eine reguläre Matrix T existiert mit $B = TAT^{-1}$.

Def. Jacobi Matrix (bzw. Funktionalmatrix) :

Für Funktionen $f : R^n \to R^m$ fasst man die partiellen Ableitungen in einer Matrix $J_f(x)$ zusammen.

Es gilt $J_f(x)[i, j] := \dfrac{\partial f_i(x)}{\partial x_j}$, d.h. $J_f(x) := \begin{pmatrix} \dfrac{\partial f_1}{\partial x_1} & \cdots & \dfrac{\partial f_1}{\partial x_n} \\ \vdots & & \vdots \\ \dfrac{\partial f_m}{\partial x_1} & \cdots & \dfrac{\partial f_m}{\partial x_n} \end{pmatrix}$

Für den Spezialfall $f : R^n \to R$ ergibt sich ein Zeilenvektor. Den zugehörigen transponierten (Spalten-)Vektor nennt man **Gradient** von f an der Stelle x $(grad\ f(x))$

$$grad\ f(x) := \begin{pmatrix} \dfrac{\partial f}{\partial x_1} \\ \vdots \\ \dfrac{\partial f}{\partial x_n} \end{pmatrix}.$$

Def. Hessenberg-Matrix :

$$A = \begin{bmatrix} * & \cdots & \cdots & \cdots & * \\ * & \ddots & & & \vdots \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & * & * \end{bmatrix}$$

Def. Vektornorm :

Eine Abbildung $\|.\| : R^n \to R$ heißt Vektornorm auf R^n, falls :

1.) Positivität

$\|x\| > 0$ für $x \in R^n$

$\|x\| = 0 \quad \Leftrightarrow \quad x = 0$

2.) Homogenität

$\|a \cdot x\| = |a| \cdot \|x\| \qquad (a \in R\ ,\ x \in R^n)$

3.) Dreiecksungleichung

$\|x + y\| \leq \|x\| + \|y\|$

Vektornorm Beispiele :

$\|x\|_2 := \sqrt{\sum_{i=1}^{n} x_i^2}$ - euklidsche Norm

$\|x\|_1 := \sum_{i=1}^{n} |x_i|$ - 1-Norm

$\|x\|_\infty := \max_{i=1,\dots,n} |x_i|$ - Maximumnorm

Def. Matrixnorm :

Definition wie bei Vektornorm

Durch $\|A\|_p = \sup_{x \neq 0} \frac{\|Ax\|_p}{\|x\|_p}$ $\left(A \in R^{n \times m}\right)$ ist eine mit der Vektornorm $\|.\|$ verträgliche Matrixnorm definiert, die der Vektornorm $\|.\|$ zugeordnete (induzierte) Matrixnorm.

Eine Matrixnorm heißt **submultiplikativ**, falls

$\|AB\| \leq \|A\| \cdot \|B\|$ (z.B. Frobeniusnorm)

Matrixnorm Beispiele :

- Zeilensummennorm $\|A\|_\infty = \max_{i=1,\dots,m} \sum_{j=1}^{n} |a_{ij}|$
- Spaltensummennorm $\|A\|_1 = \max_{j=1,\dots,nj} \sum_{i=1}^{m} |a_{ij}|$
- Spektralnorm $\|A\|_2 = \sqrt{r(A^T A)}$
- Frobeniusnorm $\|A\|_F = \sqrt{\left(\sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}^2\right)}$

Def. Spektralradius :

$r(A) := \max_{k=1,\dots,n} |l_k(A)|$ = betragsmäßig größter Eigenwert der Matrix

Def. Nullraum einer Matrix :

Für $A \in R^{(m \times n)}$ heißt der Lösungsraum des homogenen linearen Gleichungssystems $Ax = 0$ der Nullraum oder der Kern von A.

Def. Spektrum einer Matrix :

Sei A eine (n, n) - Matrix. Dann heißt die Menge aller Eigenwerte von A das Spektrum von A.

1. Lineare Gleichungssysteme

- LGS $Ax = b \left(x = A^{-1}b\right)$ lässt sich für $\det A \neq 0$ mit der Cramerschen Regel berechnen :

$$x_j = \frac{1}{\det A} \cdot \begin{pmatrix} a_{11} & \dots & a_{1,j-1} & b_1 & a_{1,j+1} & \dots & a_{1,n} \\ a_{21} & \dots & a_{2,j-1} & b_2 & a_{2,j+1} & \dots & a_{2,n} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n1} & \dots & a_{n,j-1} & b_n & a_{3,j+1} & \dots & a_{n,n} \end{pmatrix}$$

, d.h. ersetze j-te Spalte von A durch b

Def. Determinante (nach Leibniz)

$$\det A = \sum_{p \in S_n} \operatorname{sgn} p \cdot a_{1,p(1)} \cdot \dots \cdot a_{n,p(n)}$$

$\Rightarrow$ Aufwand zur Berechnung von $\det A$ beträgt $n \cdot n!$ Operationen.

Auch mit der rekursiven Bestimmung über Unterdeterminanten nach dem Laplaceschen Entwicklungssatz sind 2^n Operationen auszuführen.

gestaffeltes Gleichungssystem

$$\begin{aligned} r_{11}x_1 + r_{12}x_2 + \dots + r_{1n}x_n &= z_1 \\ r_{22}x_2 + \dots + r_{2n}x_n &= z_2 \\ \ddots \quad \vdots \quad & \vdots \\ r_{nn}x_n &= z_n \end{aligned}$$

,d.h. $Rx = z$

Auflösung durch Rückwärtssubstitution

$x_n = z_n / r_{nn}$

$x_{n-1} = (z_{n-1} - r_{n-1,n}x_n) / r_{n-1,n-1}$

usw.

Rechenaufwand :
für die i-te Zeile je n-i Additionen und Multiplikationen sowie eine Division

D.h. $\sum_{i=1}^{n} (i-1) = \frac{n(n-1)}{2} \doteq \frac{n^2}{2}$ Multiplikationen und ebenso viele Additionen.

1.1 Gaußsche Eliminationsmethode (LR-Zerlegung)

Ziel / Motivation : Umformung eines LGS in ein gestaffeltes LGS

$A \to A^{(1)} \to A^{(2)} \dots \to A^{(n)} =: R$, wobei

$$A^{(k)} = \begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \dots & \dots & \dots & a_{1n}^{(1)} \\ & a_{22}^{(2)} & \dots & \dots & \dots & a_{2n}^{(2)} \\ & & \ddots & & & \vdots \\ & & & a_{kk}^{(k)} & \dots & a_{kn}^{(k)} \\ & & & \vdots & & \vdots \\ & & & a_{nk}^{(k)} & \dots & a_{nn}^{(k)} \end{pmatrix}$$

mit einer $\left(n-k+1 \quad , \quad n-k+1\right)$ Restmatrix rechts unten und den Pivotelementen $a_{kk}^{(k)}$

Algorithmus LR-Zerlegung :

$l_{ik} := a_{ik}^{(k)} / a_{kk}^{(k)}$ für $i = k+1,...,n$

$a_{ij}^{(k+1)} := a_{ij}^{(k)} - l_{ik} \cdot a_{kj}^{(k)}$

$b_i^{(k+1)} := b_i^{(k)} - l_{ik} \cdot b_k^{(k)}$

Da jeder Eliminationsschritt eine lineare Operation auf den Zeilen von A ist, lässt sich der Übergang von $A^{(k)}$ und $b^{(k)}$ zu $A^{(k+1)}$ und $b^{(k+1)}$ als Multiplikation mit deiner Matrix L_k (Frobenius-Matrix) von links darstellen.

Def. Frobenius-Matrix

$$\begin{pmatrix} 1 & & & & & 0 \\ & 1 & & & & \\ & & \ddots & & & \\ & & & 1 & & \\ & & & b_{i+1,i} & \cdots & \\ & & & \vdots & 1 & \\ 0 & & & b_{n,i} & & 1 \end{pmatrix}$$

besondere Eigenschaft : L_k^{-1} entsteht aus L_k durch Vorzeichenwechsel der l_{ik}

$Ax = b \quad \Rightarrow \quad Rx = z$ mit $R = L^{-1}A$ und $z = L^{-1}b$

Algorithmus der Gauß-Elimination (mit Aufwandsangabe)

1. $A = LR$ - Dreieckszerlegung $\sum_{k=1}^{n-1} k^2 \doteq \frac{n^3}{3}$
2. $Lz = b$ - Vorwärtssubstitution $\sum_{k=1}^{n-1} k \doteq \frac{n^2}{2}$
3. $Rx = z$ - Rückwärtssubstitution $\sum_{k=1}^{n-1} k \doteq \frac{n^2}{2}$

Der Hauptaufwand besteht also in der LR-Zerlegung, die für verschiedene rechte Seiten $b_1,...,b_j$ aber nur einmal durchgeführt werden muss.

Beispiel für Gaußsches Eliminationsverfahren / LR-Zerlegung :

Gegeben : $Ax = b$ hier : $\begin{pmatrix} 1 & 4 \\ 2 & 2 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \end{pmatrix}$

1.) LR-Zerlegung

$$L = \begin{pmatrix} 1 & 0 \\ l_{21} & 1 \end{pmatrix}, \; l_{ik} := \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \qquad \Rightarrow l_{21} = \frac{a_{21}}{a_{11}} = 2 \qquad \Rightarrow L = \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix}$$

$$R = L^{-1}A \qquad \Rightarrow R = \begin{pmatrix} 1 & 0 \\ -2 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 4 \\ 2 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 4 \\ 0 & -6 \end{pmatrix}$$

2.) $Lz = b$ **- Vorwärtssubstitution**

$$\begin{array}{cc|l} 1 & 0 & 2 \\ 2 & 1 & 2 \end{array} \quad \begin{array}{l} \rightarrow z_1 = 2 \\ \rightarrow z_2 = 2 - 4 = -2 \end{array}$$

3.) $Rx = z$ **- Rückwärtssubstitution**

$$\begin{array}{cc|l} 1 & 4 & 2 \\ 0 & -6 & -2 \end{array} \quad \begin{array}{l} \rightarrow x_1 = 2 - 4 \cdot 1/3 = 2/3 \\ \rightarrow x_2 = 1/3 \end{array}$$

, Lsg. : $\begin{pmatrix} 1 & 4 \\ 2 & 2 \end{pmatrix} \begin{pmatrix} 2/3 \\ 1/3 \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \end{pmatrix}$

1.2 Pivot-Strategien und Nachiteration

Motivation :
Wie man bereits an dem einfachen Beispiel

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad , \det A = -1 \quad , a_{11} = 0$$

sieht, gibt es Fälle, bei denen die Dreieckszerlegung versagt, obwohl $\det A \neq 0$.
Eine Vertauschung der Zeilen führt jedoch zur denkbar einfachsten LR-Zerlegung

$$\overline{A} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = I = LR \qquad \text{mit} \quad L = R = I$$

- Bei der rechnerischen Realisierung des Gaußschen Eliminationsverfahrens können Schwierigkeiten nicht nur bei verschwindenden, sondern auch bei „zu kleinen“ Pivotelementen entstehen.

$\Rightarrow$ **Spaltenpivotstrategie**

Bei der Gauß-Elimination wählt man diejenige Zeile als Pivotzeile, die das betragsmäßig größte Element in der Pivotspalte besitzt.

Algorithmus Gauß-Elimination mit Spaltenpivotstrategie :

a) Wähle im Eliminationsschritt $A^{(k)} \rightarrow A^{(k+1)}$ ein $p \in \{k,...,n\}$, so dass

$$\left|a_{pk}^{(k)}\right| \geq \left|a_{jk}^{(k)}\right| \quad \text{für} \quad j = k,...,n$$

Die Zeile p soll die Pivotzeile werden.

b) Vertausche die Zeilen p und k

$$A^{(k)} \rightarrow \tilde{A}^{(k)} \qquad \text{mit} \quad \tilde{a}_{ij}^{(k)} = \begin{cases} a_{kj}^{(k)}, & falls\ i = p \\ a_{pj}^{(k)}, & falls\ i = k \\ a_{ij}^{(k)}, & sonst \end{cases}$$

Nun gilt

$$\left|\tilde{l}_{ik}\right| = \left|\frac{\tilde{a}_{ik}^{(k)}}{\tilde{a}_{kk}^{(k)}}\right| = \left|\frac{\tilde{a}_{ik}^{(k)}}{a_{pk}^{(k)}}\right| \leq 1 \qquad .$$

c) Führe den nächsten Eliminationsschritt angewandt auf $\tilde{A}^{(k)}$ aus ,
$\tilde{A}^{(k)} \rightarrow A^{(k+1)}$.

Bemerkung :

Anstelle der Spaltenpivotstrategie mit Zeilentausch kann man auch eine Zeilenpivotstrategie mit Spaltentausch durchführen. Beide Strategien erfordern im schlimmsten Fall $O(n^2)$ zusätzliche Operationen.
Kombiniert man beide Möglichkeiten und sucht die gesamte Restmatrix nach dem betragsgrößten Element ab, so benötigt man zusätzlich $O(n^3)$ Operationen. Diese teure vollständige Pivotsuche wird so gut wie nie angewandt.

Zur formalen Beschreibung der Dreieckszerlegung mit Spaltenpivotsuche werden im Folgenden Permutationsmatrizen $P \in Mat_n(R)$ verwendet. Jeder Permutation $p \in S_n$ aus der symmetrischen Gruppe wird die Matrix

$$P_p = \left[e_{p(1)} ... e_{p(n)}\right] \qquad \text{zugeordnet, wobei} \qquad e_j = \left(d_{1j},...,d_{nj}\right)^T \text{ der } j\text{-te Einheitsvektor ist.}$$

Eine Permutation p der Zeilen einer Matrix A lässt sich durch die Multiplikation von links mit P_p ausdrücken.
(Analog eine Permutation der Spalten durch Multiplikation von rechts.)

Die Zuordnung $p \to P_p$ ist ein Gruppenhomomorphismus $S_n \to O(n)$ der symmetrischen Gruppe S_n in die orthogonale Gruppe $O(n)$.
Insbesondere gilt : $P^{-1} = P^T$.

Die Determinante einer Permutationsmatrix ist das Vorzeichen der zugehörigen Permutation, d.h.

$$\det P_p = \operatorname{sgn} p \in \{\pm 1\},$$

also $+1$ falls p durch eine gerade, und -1, falls p durch eine ungerade Anzahl von Transpositionen erzeugt wird.

Der folgende Satz zeigt, dass die Dreieckszerlegung mit Spaltenpivotsuche theoretisch nur versagen kann, falls die Matrix A singulär ist.

Satz 1.1

Für jede invertierbare Matrix A existiert eine Permutationsmatrix P derart, dass eine Dreieckszerlegung

$$PA = LR$$

möglich ist. Dabei kann P so gewählt werden, dass alle Elemente von L betragsmäßig kleiner gleich 1 sind.

Beweis : Buch S.13/14

Bemerkung :

Es gilt : $\det A = \det(P) \cdot \det(LR) = \operatorname{sgn}(p_0) \cdot r_{11} \cdot ... \cdot r_{nn}$

Allerdings : $\det(a \cdot A) = a^n \cdot \det A$

D.h. über die ganze Klasse dieser Transformationen betrachtet bleibt von der Determinante nur die invariante Boolesche Größe $\det A = 0$ oder $\det A \neq 0$ übrig.

Die Pivotstrategie lässt sich beliebig abändern, indem man die verschiedenen Zeilen mit unterschiedlichen statt gleichen Skalaren multipliziert. Dies führt zur praktisch enorm wichtigen Frage der **Skalierung**.

Def. Zeilenskalierung : Multiplikation von A mit einer Diagonalmatrix von links

$$A \to D_Z A$$

Def. Spaltenskalierung : Multiplikation von A mit einer Diagonalmatrix von rechts

$$A \to A D_S$$

Mathematisch gesprochen ändert man mit der Skalierung die Länge der Basisvektoren des Bildraumes (Zeilenskalierung) bzw. Urbildraumes (Spaltenskalierung) der durch die Matrix A beschriebenen linearen Abbildung.
(physikalisch gesehen z.B. Änderung der Einheiten - z.B. von mm auf km)
Damit die Lösung des linearen Gleichungssystems $Ax = b$ nicht von dieser Wahl der Einheiten abhängt, muss das System in geeigneter Weise skaliert werden :

$$A \to \tilde{A} := D_Z A D_S \quad ,$$

wobei

$$D_Z = diag(s_1,...,s_n) \qquad \text{und} \qquad D_S = diag(t_1,...,t_n) \ .$$

Auf den ersten Blick vernünftig scheinen die folgenden Strategien :

a) <u>Äquilibrierung der Zeilen bezüglich einer Vektornorm</u>

Sei A^i die i-te Zeile von A, und sei vorausgesetzt, dass keine Zeile eine Nullzeile ist.
Setzt man dann $D_S := I$ und

$$s_i := \left\| A^i \right\|^{-1} \qquad \text{für } i = 1,...,n \quad ,$$

so haben alle Zeilen von $\tilde{A}$ die Norm 1.

b) <u>Äquilibrierung der Spalten</u>

Seien alle Spalten A_j verschieden von Null.
Setzt man dann $D_Z := I$ und

$$t_j := \left\| A_j \right\|^{-1} \qquad \text{für } j = 1,...,n \quad ,$$

so haben alle Spalten von $\tilde{A}$ die Norm 1.

Problem :

Die berechnete Lösung $\tilde{x}$ kann immer noch „ziemlich ungenau“ sein.
Man könnte die Lösung $\tilde{x}$ verwerfen und versuchen, mit einer höheren Maschinengenauigkeit eine „bessere“ Lösung zu berechnen.
Dabei würden jedoch alle Informationen verloren gehen, die man bei der Berechnung von $\tilde{x}$ gewonnen hat.

Dies wird bei der sogenannten **<u>iterativen Verbesserung</u>** oder **<u>Nachiteration</u>** vermieden, bei der das **<u>Residuum</u>**

$$r(y) := b - Ay = A(x - y)$$

explizit ausgewertet wird. Verschwände das Residuum, so wäre die exakte Lösung x gefunden.
Der absolute Fehler $\Delta x_0 := x - x_0$ von $x_0 := \tilde{x}$ genügt der Gleichung

$$A \cdot \Delta x_0 = r(x_0) \quad . \qquad \textbf{(*)}$$

Bei der Lösung dieser Korrekturgleichung (*) erhält man im Allgemeinen eine wiederum fehlerbehaftete Korrektur $\tilde{\Delta} x_0 \neq \Delta x_0$. Trotzdem erwartet man, dass die Näherungslösung

$$x_1 := x_0 + \tilde{\Delta} x_0$$

„besser“ ist als x_0.

Die Idee der Nachiteration besteht nun darin, diesen Prozess solange zu wiederholen, bis die Näherungslösung x_i „genau genug“ ist.
Dabei beachte man, das sich das Gleichungssystem (*) nur in der rechten Seite von dem Ursprünglichen unterscheidet, so das die Berechnung der Korrekturen Δx_i relativ wenig Aufwand erfordert.
(weiterführend in Kapitel 2.3.2 - „Beurteilung von Näherungslösungen“)

1.3 Cholesky Zerlegung (für spd-Matrizen)

Motivation : Bei Gleichungssystemen mit spd-Matrizen kann die Dreieckszerlegung stark vereinfacht werden.

Satz 1.2 :

Für jede spd-Matrix existiert eine eindeutig bestimmte Zerlegung der Form $A = LDL^T$,
wobei L eine unipotente untere Dreiecksmatrix und D eine positive Diagonalmatrix (der Pivotelemente) ist.

Kor. 1.3 :

Da $D = diag(d_i)$ positiv ist, existiert $D^{\frac{1}{2}} = diag(\sqrt{d_i})$ und daher die Cholesky-Zerlegung

$A = \overline{L}\overline{L}^T$,wobei $\overline{L}$ die untere Dreiecksmatrix $\overline{L} := LD^{\frac{1}{2}}$ ist.

Algorithmus der Cholesky Zerlegung

for $k := 1$ to n do

$$l_{kk} := \left(a_{kk} - \sum_{j=1}^{k-1} l_{kj}^2 \right)^{\frac{1}{2}}$$

for $i := k+1$ to n do

$$l_{ik} := \left(a_{ik} - \sum_{j=1}^{k-1} l_{ij} \cdot l_{kj} \right) / l_{kk}$$

endfor

endfor

Beispiel für Cholesky-Zerlegung :

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

1.) mit LR-Zerlegung $\left(A = LDL^T\right)$

$$A^{(1)} = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \quad , l_{ik} := \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \quad \Rightarrow l_{21} = \frac{1}{2} \quad \Rightarrow L = \begin{pmatrix} 1 & 0 \\ 1/2 & 1 \end{pmatrix}$$

$$a_{ij}^{(k+1)} := a_{ij}^{(k)} - l_{ik} \cdot a_{kj}^{(k)} \quad \Rightarrow a_{21}^{(2)} = 1 - \frac{1}{2} \cdot 2 = 0 \text{ und } \quad a_{22}^{(2)} = 2 - \frac{1}{2} \cdot 1 = \frac{3}{2}$$

$$\Rightarrow A^{(2)} = \begin{pmatrix} 2 & 1 \\ 0 & 3/2 \end{pmatrix} \Rightarrow D = \begin{pmatrix} 2 & 0 \\ 0 & 3/2 \end{pmatrix}$$

Probe : $LDL^T = \begin{pmatrix} 1 & 0 \\ 1/2 & 1 \end{pmatrix} \cdot \begin{pmatrix} 2 & 0 \\ 0 & 3/2 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1/2 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 1 & 3/2 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1/2 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} = A$

2.) direkt $\left(A = \overline{L} \cdot \overline{L}^T\right)$

$$l_{kk} := \left(a_{kk} - \sum_{j=1}^{k-1} l_{kj}^2 \right)^{\frac{1}{2}} \quad \Rightarrow l_{11} = \sqrt{\left(a_{11} - \sum_{j=1}^{0} l_{kj}^2 \right)} = \sqrt{a_{11}} = \sqrt{2}$$

$$\text{for } i := k+1 \text{ to } n \text{ do} \quad l_{ik} := \left(a_{ik} - \sum_{j=1}^{k-1} l_{ij} \cdot l_{kj} \right) / l_{kk} \qquad \Rightarrow l_{21} = \frac{a_{21} - \sum_{j=1}^{0} l_{ij} \cdot l_{kj}}{l_{11}} = \frac{1}{\sqrt{2}}$$

$$l_{kk} := \left(a_{kk} - \sum_{j=1}^{k-1} l_{kj}^2 \right)^{\frac{1}{2}} \quad \Rightarrow l_{22} = \sqrt{\left(a_{22} - \sum_{j=1}^{1} l_{kj}^2 \right)} = \sqrt{a_{22} - l_{21}^2} = \sqrt{2 - \frac{1}{2}} = \sqrt{\frac{3}{2}}$$

Probe : $\overline{L} \cdot \overline{L}^T = \begin{pmatrix} \sqrt{2} & 0 \\ \frac{1}{\sqrt{2}} & \sqrt{\frac{3}{2}} \end{pmatrix} \cdot \begin{pmatrix} \sqrt{2} & \frac{1}{\sqrt{2}} \\ 0 & \sqrt{\frac{3}{2}} \end{pmatrix} = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} = A$

2. Fehleranalyse

Es gibt zwei Arten von Fehlern :

- **Eingabefehler** : Maschinengenauigkeit , Messfehler $- f(\tilde{x}) - f(x)$ $\rightarrow$ Kondition eines Problems
  - relativer Fehler einer Gleitkommazahl d mit Mantissenlänge l : $\frac{|x - fl(x)|}{|x|} \le eps := \frac{d^{1-l}}{2}$
  - Messfehler : absoluter Fehler/Toleranz $d \cdot x$ $|x - \bar{x}| \le d \cdot x$ ($\bar{x}$ = „wahrer“ Wert)
- **Fehler im Algorithmus** : Rundungsfehler , Approximationsfehler $- \tilde{f}(x) - f(x)$ $\rightarrow$ Stabilität eines Algorithmus
  - Rundungsfehler : $a \hat{\circ} b = (a \circ b)(1+e)$ für ein $e = e(x,y)$ mit $|e| \le eps$
  - Approximationsfehler : Berechnung des Sinus durch Reihenentwicklung

Machtlos gegenüber Eingabefehlern.
Fehler im Algorithmus kann man durch Änderung des Verfahrens vermeiden bzw. verringern

Gleitpunktzahlen sind auf der reellen Achse nicht gleichverteilt !

Def. Maschinenepsilon $eps := b^{1-l}$: **(** b **- Basis** l **- Mantissenlänge)**

= kleinste Maschinenzahl, die zu 1 addiert einen von 1 verschiedenen Wert ergibt.

Def. absoluter / relativer Fehler :

Der absolute Fehler einer Größe mit Sollwert $\bar{x}$ und Istwert x ist $dx := \|x - \bar{x}\|$.

Ist $\bar{x} \neq 0$, so ist der relative Fehler definiert als $\frac{\|x - \bar{x}\|}{\|\bar{x}\|}$.

2.1 Kondition eines Problems

= Charakterisierung des Verhältnisses von Eingabe- und Resultatmenge

Die Eingabe x ist logisch ununterscheidbar von allen Eingaben $\tilde{x}$, die sich im Rahmen der vorgegebenen Genauigkeit befinden. Statt der exakten Eingabe x betrachtet man die Eingabemenge E aller dieser gestörten Eingaben $\tilde{x}$.

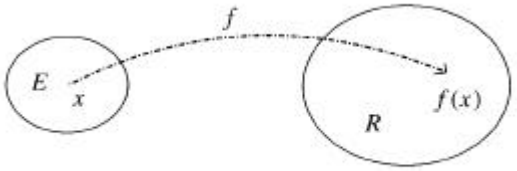


Abbildung 2.1. Eingabe- und Resultatmenge

Eine Maschinenzahl x repräsentiert demnach die Eingabemenge $E = \{\tilde{x} \in R : |\tilde{x} - x| \leq eps \cdot |x|\}$.

Wüsste man hingegen, dass die Eingabe x mit einer absoluten Toleranz $d \cdot x$ vorliegt, so wäre die Eingabemenge $E = \{\tilde{x} \in R : |\tilde{x} - x| \leq d \cdot x\}$.

Beispiel für die Kondition eines Problems : zeichnerische Bestimmung des Schnittpunktes zweier Geraden

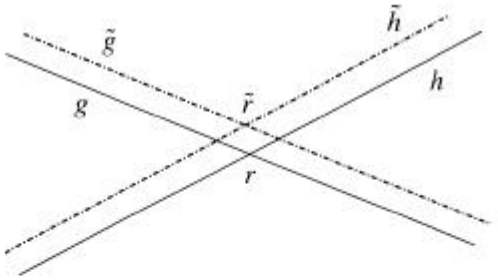


Abbildung 2.2. Schnittpunkt r zweier Geraden g, h (gut konditioniert)

Abbildung 2.3. Schleifender Schnitt (schlecht konditioniert)

Beispiel für die Kondition eines Problems : zeichnerische Bestimmung der Nullstellen einer Funktion

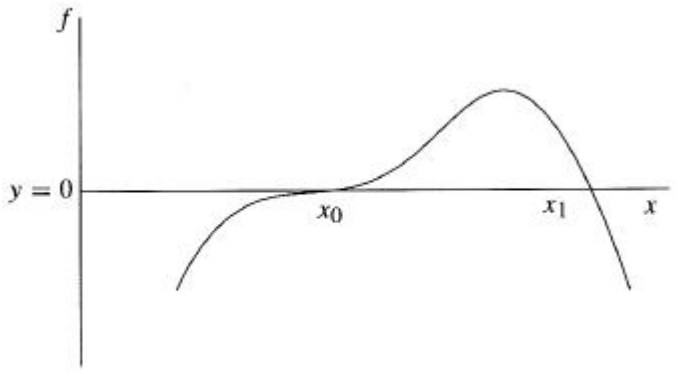


Abbildung 2.4. Schlecht konditionierte Nullstelle bei x_0, gut konditionierte bei x_1

2.1.1 Normweise Konditionsanalyse

Die **absolute normweise Kondition** des Problems (f, x) ist die kleinste Zahl $k_{abs} \geq 0$,
so daß $\|f(\tilde{x}) - f(x)\| \dot{\leq} k_{abs} \|\tilde{x} - x\|$ für $\tilde{x} \to x$
Das Problem ist schlecht bzw. unsachmäßig gestellt, wenn es keine solche Zahl gibt.

Die **relative normweise Kondition** des Problems (f, x) ist die kleinste Zahl $k_{rel} \geq 0$,
so daß $\frac{\|f(\tilde{x}) - f(x)\|}{\|f(x)\|} \dot{\leq} k_{rel} \frac{\|\tilde{x} - x\|}{\|x\|}$ für $\tilde{x} \to x$

D.h. k_{abs} beschreibt die Verstärkung des absoluten Fehlers und k_{rel} die Verstärkung des relativen Fehlers.
Ein Problem ist gut konditioniert, falls seine Kondition klein ist.
Als Orientierung : $k_{rel} = 1$ entspricht der reinen Rundung des Resultats.

Für differenzierbare f gilt aufgrund des Mittelwertsatzes :

- $k_{abs} = \|f'(x)\|$ und $k_{rel} = \|x\| \cdot \frac{\|f'(x)\|}{\|f(x)\|}$

  , wobei $\|f'(x)\|$ die Norm der Jacobi-Matrix $f'(x)$ in der zugeordneten Matrixnorm $\|A\| = \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|}$ ist.

Beispiel : Kondition der Addition (bzw. Subtraktion)

$f : R^2 \to R$ d.h. $f(x,y) := x + y$ $\Rightarrow f'(x,y) = \left(\frac{\partial f}{x} \quad \frac{\partial f}{y} \right) = (1 \quad 1)$

Wählt man auf R^2 die 1-Norm $\left\| (x \quad y)^T \right\| = |x| + |y|$ und auf R den Betrag, so folgt für die Ableitung in der zugeordneten Matrixnorm, dass $\| f'(x,y) \| = \|(1 \quad 1)\| = 1$.

$\Rightarrow k_{abs} = 1$ und $k_{rel} = \frac{|x| + |y|}{|x + y|}$

Hieraus folgt, dass die Subtraktion zweier annähernd gleicher Zahlen bezüglich der relativen Kondition schlecht konditioniert ist, da in diesem Fall gilt : $|x + y| << |x| + |y| \quad \Leftrightarrow \quad k_{rel} >> 1$.

$\Rightarrow$ **Vermeidbare Subtraktionen annähernd gleicher Zahlen vermeiden.**
Falls unvermeidbar, diese an den Anfang des Algorithmus stellen. (siehe Vorwärtsanalyse)

Beispiel für vermeidbare Auslöschung : Lösen einer quadratischen Gleichung

$f(x) := x^2 + px + q$ $x_{1,2} = -\frac{p}{2} \pm \sqrt{\frac{p^2}{4} - q}$

Liegt eine Nullstelle in der Nähe von Null, so tritt Auslöschung auf.
Nach dem Satz von Vieta jedoch ist q das Produkt der Nullstellen und somit lassen sich die beiden Lösungen stabil berechnen : $x_1 = -\frac{p}{2} + \mathrm{sgn}(p) \sqrt{\frac{p^2}{4} - q}$, $x_2 = \frac{q}{x_1}$

Beispiel : Kondition eines LGS Ax=b

Betrachtet man bei der Lösung des LGS nur den Vektor b als Eingabegröße, so wird das Problem von der Abbildung $f(b) := A^{-1}b$ beschrieben.

Diese ist bezüglich b linear, so daß $f'(b) = A^{-1}$.

$\Rightarrow \quad k_{abs} = \left\| A^{-1} \right\|$ und $k_{rel} = \|b\| \cdot \frac{\left\| A^{-1} \right\|}{\left\| A^{-1} b \right\|} = \frac{\|Ax\|}{\|x\|} \cdot \left\| A^{-1} \right\|$

Es lassen sich aber auch leicht Störungen in A berücksichtigen.
Die Abbildung $f(A) = A^{-1}b$ ist nichtlinear in A, aber differenzierbar.

Lemma 2.1 :

Die Abbildung $g : GL(n) \subset Mat_n(R) \to GL(n)$ mit $g(A) = A^{-1}$ ist differenzierbar, und es gilt

$g'(A) \cdot C = -A^{-1} \cdot C \cdot A^{-1}$ für alle $C \in Mat_n(R)$

Beweis : Buch S.35

Aus dem Lemma folgt für die Ableitung der Lösung $f(A) = A^{-1}b$ des LGS nach A, dass

$$f'(A) \cdot C = -A^{-1} \cdot C \cdot A^{-1}b = -A^{-1} \cdot C \cdot x \qquad \text{für } C \in Mat_n(R).$$

Hieraus folgen nun die Konditionen

$$k_{abs} = \|f'(A)\| = \sup_{\|C\|=1} \|A^{-1} \cdot C \cdot x\| \le \|A^{-1}\| \cdot \|x\|$$

$$k_{rel} = \frac{\|A\|}{\|x\|} \|f'(A)\| \le \|A\| \cdot \|A^{-1}\| \qquad \text{(Abschätzung siehe oben)}$$

Die Größe $k(A) := \|A\| \cdot \|A^{-1}\|$ wird **<u>Kondition der Matrix</u>** A genannt.

Sie beschreibt insbesondere die relative Kondition eines linearen Gleichungssystems $Ax = b$ für alle möglichen rechten Seiten $b \in R^n$.

Eine andere Darstellung für $k(A)$ ist $\qquad k(A) := \dfrac{\max_{\|x\|=1} \|Ax\|}{\min_{\|x\|=1} \|Ax\|} \in [0, \infty].$ (Buch Übung 2.12)

Vorteil : Auch für nicht invertierbare und rechteckige Matrizen wohldefiniert.

Aus dieser Darstellung ergeben sich sofort folgende drei Eigenschaften von $k(A)$:

1) $k(A) \ge 1$

2) $k(a \cdot A) = k(A)$ für alle $a \in R \quad , \quad a \neq 0$

3) $A \neq 0$ ist singulär $\Leftrightarrow$ $k(A) = \infty$

D.h. die Kondition $k(A)$ einer Matrix ist im Gegensatz zur Determinante $\det A$ invariant unter den skalaren Transformationen $A \to a \cdot A$.
Zusammen mit den Eigenschaften 1) und 3) kann diese Konditionszahl daher eher zur Charakterisierung der Lösbarkeit eines LGS herangezogen werden als die Determinante.

2.1.2 Komponentenweise Konditionsanalyse

Günstiger, weil alle Eingaben in einen Rechner mit einem relativen Fehler in den einzelnen Komponenten behaftet sind und so einige Phänomene bei der normweisen Betrachtung nicht erklärt werden können.
Auch nimmt die normweise Betrachtung z.B. bei LGS keine Rücksicht auf eventuell vorliegende Spezialstrukturen einer Matrix A, sondern analysiert das Verhalten bezüglich beliebiger Störungen dA, d.h. auch solcher, die diese Spezialstruktur nicht erhalten.

<u>Beispiel : Kondition einer Diagonalmatrix</u>

Die Lösung eines Gleichungssystems $Ax = b$ mit einer Diagonalmatrix $A = \begin{pmatrix} 1 & 0 \\ 0 & e \end{pmatrix}$, $A^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & e^{-1} \end{pmatrix}$

ist offensichtlich ein gut konditioniertes Problem, da die Gleichungen vollkommen unabhängig voneinander (entkoppelt) sind. (Dabei wird natürlich implizit vorausgesetzt, dass die Störungen die Diagonalgestalt erhalten.)

Die normweise Kondition $k_\infty(A) = \|A\|_\infty \cdot \|A^{-1}\|_\infty = 1 \cdot e^{-1} = \dfrac{1}{e}$ wird für kleine $e \le 1$ jedoch beliebig groß.

Erwartungsgemäß sollte die Kondition einer Diagonalmatrix, d.h. eines vollständig entkoppelten Gleichungssystems, stets 1 betragen – wie für eine skalare Gleichung.
$\Rightarrow$ Einführung der komponentenweisen Konditionsanalyse

Def. komponentenweise Kondition :

Die (im Urbild) komponentenweise Kondition des Problems (f, x) ist die kleinste Zahl $k_{rel} \geq 0$, so dass

$$\frac{\|f(\tilde{x}) - f(x)\|_\infty}{\|f(x)\|_\infty} \dot{\leq} k_{rel} \cdot \max_i \frac{|\tilde{x}_i - x_i|}{|x_i|} \qquad \text{für } \tilde{x} \to x.$$

Alternativ kann man die relative komponentenweise Kondition auch wiefolgt definieren :

$$\max_i \frac{|f_i(\tilde{x}) - f_i(x)|}{|f_i(x)|} \dot{\leq} k_{rel} \cdot \max_i \frac{|\tilde{x}_i - x_i|}{|x_i|} \qquad \text{für } \tilde{x} \to x.$$

Die Anwendung des Mittelwertsatzes $f(\tilde{x}) - f(x) = \int_{t=0}^{1} f'(x + t \cdot (\tilde{x} - x)) \cdot (\tilde{x} - x)\, dt$ liefert komponentenweise

$$|f(\tilde{x}) - f(x)| \leq \int_{t=0}^{1} |f'(x + t \cdot (\tilde{x} - x))| \cdot |(\tilde{x} - x)|\, dt \qquad \text{und daher}$$

- $$k_{rel} = \frac{\| \, |f'(x)| \cdot |x| \, \|_\infty}{\| \, f(x) \, \|_\infty}$$

Beispiel : Kondition der Multiplikation

Die Multiplikation zweier reeller Zahlen wird durch die Abbildung $f : R^2 \to R \quad , \quad (x, y)^T \mapsto f(x, y) = xy$ beschrieben. Sie ist differenzierbar mit $f'(x, y) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2} \right) = (y, x)$ und somit folgt

$$k_{rel} = \frac{\| \, |f'(x, y)| \cdot |(x, y)^T| \, \|_\infty}{\| \, f(x, y) \, \|_\infty} = \frac{|yx| + |xy|}{|xy|} = 2.$$

$\to$ Die Multiplikation ist daher noch gut konditioniert zu nennen.

Beispiel : Kondition des Skalarproduktes

$\langle x, y \rangle = \sum_{i=1}^{n} x_i y_i$ Es gilt, die Abbildung $f : R^{2n} \to R \quad , \quad (x, y) \mapsto \langle x, y \rangle$ an der Stelle (x, y) auszuwerten.

Da f differenzierbar ist mit $f'(x, y) = (y^T, x^T)$, folgt

$$k_{rel} = \frac{\| \, |(y^T, x^T)| \cdot |(x, y)| \, \|_\infty}{\|\langle x, y \rangle\|_\infty} = \frac{|y^T| \cdot |x| + |x^T| \cdot |y|}{|\langle x, y \rangle|} = 2 \frac{\langle \, |x|, |y| \, \rangle}{|\langle x, y \rangle|}$$

Beispiel : komponentenweise Kondition eines linearen Gleichungssystems (Skeelsche Kondition) $Ax = b$

- Betrachtet man nur den Vektor b als Eingabegröße, so wird das Problem von der Abbildung $f(b) = A^{-1}b$ beschrieben.

  $\Rightarrow f'(b) = A^{-1}$, da f bezüglich b linear

  $$\Rightarrow k_{rel} = \frac{\left\| \left| f'(b) \right| \cdot |b| \right\|_\infty}{\| f(b) \|_\infty} = \frac{\left\| \left| A^{-1} \right| \cdot |b| \right\|_\infty}{\| A^{-1} b \|_\infty} = \frac{\left\| \left| A^{-1} \right| \cdot |b| \right\|_\infty}{\| x \|_\infty}$$

  Mit dieser, von *Skeel* eingeführten, Zahl lassen sich die Störungen $\tilde{x} - x$, $\tilde{x} = A^{-1}\tilde{b}$ wiefolgt abschätzen :

  $$\frac{\|\tilde{x} - x\|_\infty}{\|x\|_\infty} \le k_{rel} \cdot e \qquad \text{für } |\tilde{b} - b| \le e \cdot |b|$$

- Betrachtet man Störungen in A, so gilt :

  Die Abbildung $f(A) = A^{-1}b$ ist differenzierbar mit $f'(A) \cdot C = -A^{-1}CA^{-1}b = -A^{-1}Cx$ für $C \in Mat_n(R)$.

  $$\Rightarrow k_{rel} = \frac{\left\| \left| f'(A) \right| \cdot |A| \right\|_\infty}{\| f(A) \|_\infty} = \frac{\left\| \left| A^{-1} \right| \cdot |A| \cdot |x| \right\|_\infty}{\|x\|_\infty}$$

Fässt man diese Ergebnisse zusammen und betrachtet Störungen in A und b, so addieren sich die beiden relativen Konditionen auf, und man erhält die **Kondition für das Gesamtproblem** :

$$k_{rel} = \frac{\left\| \left| A^{-1} \right| \cdot |A| \cdot |x| \quad + \quad \left| A^{-1} \right| \cdot |b| \right\|_\infty}{\|x\|_\infty} \le 2 \frac{\left\| \left| A^{-1} \right| \cdot |A| \cdot |x| \right\|_\infty}{\|x\|_\infty}$$

Setzt man für x den Vektor $e = (1,...,1)$ ein, so ergibt sich eine Charakterisierung der komponentenweisen Kondition von $Ax = b$ für alle möglichen rechten Seiten b :

$$\frac{1}{2} k_{rel} \le \frac{\left\| \left| A^{-1} \right| \cdot |A| \cdot |e| \right\|_\infty}{\|e\|_\infty} = \left\| \left| A^{-1} \right| \cdot |A| \right\|_\infty \quad \text{mit der sogenannten } \textbf{Skeelschen Kondition } k_C(A) := \left\| \left| A^{-1} \right| \cdot |A| \right\|_\infty$$

.Diese Kondition $k_C(A)$ erfüllt die gleichen drei Eigenschaften wie $k(A)$ (siehe früher)

und zusätzlich die eingangs geforderte Eigenschaft $k_C(D) = 1$ für jede Diagonalmatrix D.

Sie ist sogar invariant unter Zeilenskalierung, d.h. $k_C(D \cdot A) = k_C(A)$,

da $\left|(DA)^{-1}\right| \cdot |DA| = \left|A^{-1}\right| \cdot \left|D^{-1}\right| \cdot |D| \cdot |A| = \left|A^{-1}\right| \cdot |A|$.

2.2 Stabilitätskonzepte

Definition : $\tilde{f}(E) := \bigcup_{\Phi \in \tilde{f}} \Phi(E)$.

2.2.1 Vorwärtsanalyse

Analyse der Menge $\tilde{R} := \tilde{f}(E)$ aller durch Eingabefehler ***und*** Fehler im Algorithmus gestörten Resultate.

Ist $\tilde{R}$ von der gleichen Größenordnung wie R , so ist der Algorithmus stabil im Sinne der Vorwärtsanalyse.

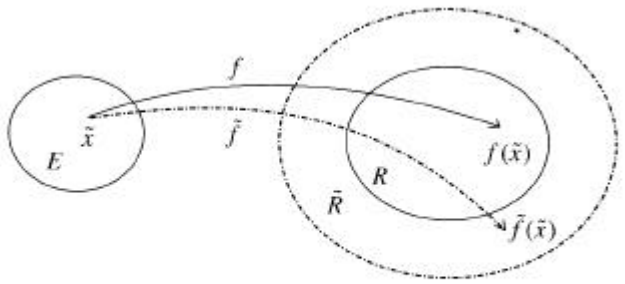


Abbildung 2.5. Eingabemenge und Resultatmengen bei der Vorwärtsanalyse

Der vom Algorithmus verursachte Fehler $\tilde{f}(x) - f(x)$ muss in Beziehung gesetzt werden zum unvermeidbaren Fehler $k_{rel} \cdot eps$. (eps = Eingabefehler)

Def. Stabilitätsindikator der normweisen Vorwärtsanalyse :

= Faktor um den der Algorithmus den unvermeidbaren Fehler verstärkt.

= kleinstmögliche Zahl $s \geq 0$, so dass für alle $\tilde{x} \in E$ gilt :

$$\frac{\left\| \tilde{f}(\tilde{x}) - f(\tilde{x}) \right\|}{\left\| f(\tilde{x}) \right\|} \dot{\leq} s \cdot k_{rel} \cdot eps \qquad \text{, für } eps \to 0 \quad ,$$

wobei $\tilde{f}$ im Folgenden die Gleitkommarealisierung eines Algorithmus zur Lösung des Problems (f, x) ist.

Analog:

Def. Stabilitätsindikator der komponentenweisen Vorwärtsanalyse :

= kleinstmögliche Zahl $s \geq 0$, so dass

$$\max_i \frac{\left| \tilde{f}_i(\tilde{x}) - f_i(\tilde{x}) \right|}{\left| f_i(\tilde{x}) \right|} \dot{\leq} s \cdot \bar{k}_{rel} \cdot eps \qquad \text{, wobei } \bar{k}_{rel} \text{ = komponentenweise relative Kondition von } (f, x)$$

- **Der Algorithmus $\tilde{f}$ heißt stabil im Sinne der Vorwärtsanalyse, falls s kleiner als die Anzahl der hintereinander ausgeführten Elementaroperationen ist.**

Lemma 2.2 :

Für die Elementaroperationen $\{+,-,*,/\}$ und ihre Gleitkommarealisierungen $\{\hat{+},\hat{-},\hat{*},\hat{/}\}$ gilt :

$$s \cdot k_{rel} \leq 1$$

Beweis :

$a \hat{\circ} b = (a \circ b)(1+e)$ für ein e mit $|e| \leq eps$. Hieraus folgt :

$$\frac{|a \hat{\circ} b - a \circ b|}{|a \circ b|} = \frac{|(a \circ b)(1+e) - a \circ b|}{|a \circ b|} = |e| \leq eps$$ **Q.e.d.**

Beispiel : Stabilität der Subtraktion

Im Fall der Auslöschung , $k >> 1$, ist der Stabilitätsindikator sehr klein , $s << 1$.

Die Subtraktion ist in diesem Fall also hervorragend stabil und bei totaler Auslöschung sogar fehlerfrei, $a \hat{-} b = a - b$.

Lemma 2.3 :

Sowohl im norm- als auch im komponentenweisen Konzept gilt für den Stabilitätsindikator s_f des zusammengesetzten Algorithmus $\tilde{f} = \tilde{h} \circ \tilde{g}$, dass

$$s_f k_f \leq s_h k_h + s_g k_g k_h .$$

Beweis : S.45

Schlußfolgerung :

- **Unvermeidbare Subtraktionen möglichst an den Anfang des Algorithmus stellen.**
  Denn da für jede Elementaroperation $s \cdot k_{rel} \leq 1$ gilt, droht Gefahr für die Stabilität der zusammengesetzten Abbildung $f = h \circ g$ nur, falls es sich bei der zweiten Abbildung h um eine Subtraktion handelt, so dass $k_h >> 1$.

Lemma 2.4 :

Sind die Funktionen g und h aus Lemma 2.3 skalar und differenzierbar, so gilt für den Stabilitätsindikator s_f des zusammengesetzten Algorithmus $\tilde{f} = \tilde{h} \circ \tilde{g}$, dass

$$s_f \leq \frac{s_h}{k_g} + s_g .$$

Beweis : S.47

Schlußfolgerung :

- Ist die Kondition k_g des ersten Teilproblems sehr klein, $k_g << 1$, so wird der Algorithmus also instabil.
  Eine kleine Kondition lässt sich auch als Informationsverlust interpretieren :
  Eine Änderung der Eingabe hat so gut wie keinen Einfluss auf das Resultat.
  Ein solcher Informationsverlust zu Anfang des Algorithmus hat daher Instabilität zur Folge.
  Desweiteren schlägt sich eine Instabilität zu Beginn des Algorithmus (großes s_g) voll auf den Gesamtalgorithmus durch.

TODO : Beispiel : cos mx S.47/48 ???

Beispiel : Summation von n reellen Zahlen (komponentenweise Vorwärtsanalyse)

Der einfachste Algorithmus für die Summe $s_n : R^n \to R \quad , \quad (x_1,...,x_n) \mapsto \sum_{i=1}^{n} x_i$ ist ihre rekursive Berechnung gemäß $s_n = s_{n-1} \circ a_n$ für $n > 2$ und $s_2 = a_2$, wobei $a_n : R^n \to R^{n-1} \quad , \quad (x_1,...,x_n) \mapsto (x_1 + x_2, x_3,...,x_n)$ die Addition der ersten beiden Komponenten bezeichnet.

Konditionszahl und Stabilitätsindikator von a_n stimmen mit denen der Addition zweier Zahlen überein, d.h. $k_{a_n} = k_+$ und $s_{a_n} = s_+$.

Mit den Bezeichnungen $k_j := k_{s_j}$ und $s_j := s_{s_j}$ gilt nach Lemma 2.3, dass

$$a_n k_n \quad \leq \quad s_{n-1} k_{n-1} + s_+ k_+ k_{n-1} = (s_{n-1} + s_+ k_+) \cdot k_{n-1} \quad \leq \quad (1 + s_{n-1}) \cdot k_{n-1} .$$

- Erinnerung : Konditionszahlen der Addition : $k_{abs} = 1$ und $k_{rel} = \frac{|x| + |y|}{|x + y|}$.

Für die Konditionen k_n gilt somit, dass $k_n = \frac{\sum_{i=1}^{n} |x_i|}{\left| \sum_{i=1}^{n} x_i \right|} \geq 1$ und $k_{n-1} = \frac{\sum_{i=3}^{n} |x_i| + |x_1 + x_2|}{\left| \sum_{i=1}^{n} x_i \right|} \leq k_n$,

und daher $s_n \leq 1 + s_{n-1}$. Da $s_2 = s_+ \leq \frac{1}{k_+} \leq 1$, folgt für den Stabilitätsindikator $s_n \leq n - 1$.

Bei den benötigten $n - 1$ Elementaroperationen ist der naive Algorithmus zur Summenbildung also numerisch stabil.

Beispiel : Ausführung des Skalarproduktes

Unterteilung der Berechnung des Skalarproduktes in die komponentenweise Multiplikation $p : R^n \times R^n \to R^n \quad , \quad ((x_i),(y_i)) \mapsto (x_i y_i)$ gefolgt von der im letzten Beispiel analysierten Summenbildung.

D.h. $f = s_n \circ p$.

Nach Lemma 2.3 gilt zusammen mit Lemma 2.2 und der Abschätzung des Stabilitätsindikators im letzten Beispiel, dass

$s_f k_f \leq (s_n + s_p k_p) \cdot k_n \leq (1 + s_n) \cdot k_n \leq n \cdot k_n$ und daher

$$s_f \leq n \frac{k_n}{k_f} = n \frac{\dfrac{\sum_{i=1}^{n} |x_i y_i|}{\left| \sum_{i=1}^{n} x_i y_i \right|}}{2 \dfrac{\sum_{i=1}^{n} |x_i y_i|}{\left| \sum_{i=1}^{n} x_i y_i \right|}} = \frac{n}{2}$$

Bei $2n - 1$ Elementaroperationen ist dieser Algorithmus für das Skalarprodukt also numerisch stabil.

2.2.2 Rückwärtsanalyse

Die durch den Algorithmus verursachten Fehler werden auf die Eingabegrößen zurückgespielt und so als zusätzliche Eingabefehler interpretiert.

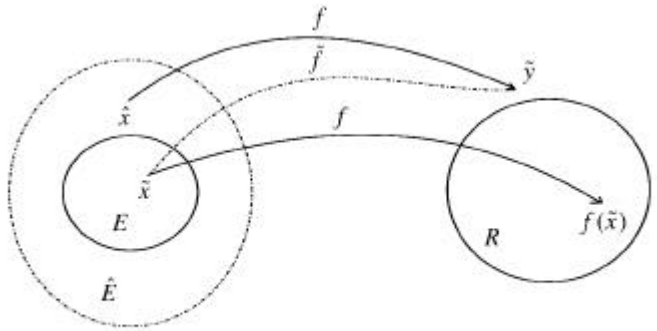


Abbildung 2.6. Eingabemengen und Resultatmenge bei der Rückwärtsanalyse

Die fehlerbehafteten Resultate $\tilde{y} = \tilde{f}(\tilde{x})$ werden als exakte Ergebnisse $\tilde{y} = f(\hat{x})$ zu gestörten Eingabegrößen $\hat{x}$ aufgefaßt.

Dies gelingt nur, wenn $\tilde{f}(E)$ im Bild von f liegt.

Ist dies nicht der Fall, so ist eine Rückwärtsanalyse nicht möglich, und der Algorithmus instabil im Sinne der Rückwärtsanalyse.

Für nicht injektive Abbildungen f wird es mehr als ein $\hat{x} \in f^{-1}(\tilde{y})$ geben.

Hier wählt man das Element $\hat{x}$ mit der geringsten Abweichung von der Eingabe $\tilde{x}$, d.h. $\|\hat{x} - \tilde{x}\| = \min$.

$$\Rightarrow \hat{E} := \left\{\hat{x} : f(\hat{x}) = \tilde{f}(\tilde{x}) \quad und \quad \|\hat{x} - \tilde{x}\| = \min \quad für\ ein\ \tilde{x} \in E\right\}$$

Das Verhältnis von $\hat{E}$ zur Eingabemenge E ist ein Maß für die Stabilität im Sinn der Rückwärtsanalyse.

Def. normweiser Rückwärtsfehler :

Der normweise Rückwärtsfehler des Algorithmus $\tilde{f}$ zur Lösung des Problems (f, x), ist die kleinste Zahl, für die für alle $\tilde{x} \in E$ ein $\hat{x}$ existiert, so dass

$$\frac{\|\hat{x} - \tilde{x}\|}{\|\tilde{x}\|} \dot{\leq} h \qquad \text{für } eps \to 0 .$$

Analog :

Def. komponentenweiser Rückwärtsfehler

= kleinste Zahl, für die für alle $\tilde{x} \in E$ ein $\hat{x}$ existiert, so dass

$$\max_i \frac{|\hat{x}_i - \tilde{x}_i|}{|\tilde{x}_i|} \dot{\leq} h \qquad \text{für } eps \to 0 .$$

Der Algorithmus heißt **stabil bezüglich des relativen Eingabefehlers** d , falls

$h < d$.

Für den durch Rundung verursachten Eingabefehler $d = eps$ wird definiert :

Def. Stabilitätsindikator der Rückwärtsanalyse :

$$s_R := \frac{h}{eps}$$

In der Definition taucht die Kondition des Problems nicht auf.

Die Rückwärtsanalyse benötigt im Gegensatz zur Vorwärtsanalyse keine vorhergehende Konditionsanalyse des Problems.
Auch sind die Ergebnisse leicht durch den Vergleich von Eingabefehler und Rückwärtsfehler zu interpretieren.

- **Aufgrund dieser Eigenschaften ist die Rückwärtsanalyse insbesondere bei komplexeren Algorithmen vorzuziehen.**

Lemma 2.5

$s \leq s_R$

Insbesondere folgt aus der Rückwärtsstabilität die Vorwärtsstabilität.

Beweis : S.50

Lemma 2.6

Gegeben sei die rekursive Vorschrift

$$\langle x, y \rangle := x_n y_n + \langle x^{(n-1)}, y^{(n-1)} \rangle$$

zur Auswertung des Skalarproduktes, wobei $x^{(n-1)} := (x_1, ..., x_{n-1})^T$ und $y^{(n-1)} := (y_1, ..., y_{n-1})^T$.

In dieser Form findet es sich meistens auf sequentiell arbeitenden Rechnern.
Hier nun die Frage nach der Stabilität bezüglich der Eingabe x , da diese später bei der Rückwärtsanalyse der Gauß-Elimination für die Analyse der Rückwärtssubstitution benötigt wird.

Die zugehörige Gleitpunktrealisierung des Skalarproduktes berechnet für $x, y \in R^n$ eine Lösung $\langle x, y \rangle_{fl}$, so dass

$$\langle x, y \rangle_{fl} = \langle \hat{x}, y \rangle \qquad \text{für ein } \hat{x} \in R^n \text{ mit} \qquad |x - \hat{x}| \dot{\leq} n \cdot eps \cdot |x|$$

, d.h. der relative komponentenweise Rückwärtsfehler beträgt $h \leq n \cdot eps$.

Das Skalarprodukt ist also (bei $2n-1$ Elementaroperationen) stabil im Sinne der Rückwärtsanalyse.

TODO : Beweis : S.51 ???

2.3 Anwendung auf lineare Gleichungssysteme (Konzepte der Kondition und Stabilität)

Nochmals die Fragestellung : Wann ist ein lineares Gleichungssystem $Ax = b$ lösbar ?
Im Unterschied zu Kapitel 1 , wo diese Frage auf der fiktiven Basis der reellen Zahlen beantwortet wurde,
wird hier nun die oben hergeleitete Fehlertheorie herangezogen.
Entsprechend wird die charakteristische Größe $\det A$ durch Konditionszahlen zu ersetzen sein.
Bei der normweisen Konditionsanalyse folgte

$$\frac{\|\tilde{x} - x\|}{\|x\|} \dot{\leq} k_{rel} \cdot d$$,wobei d der relative Eingabefehler von A (oder b) ist. **(*)**

Als Resultat für die relative Kondition folgte bei normweiser Betrachtung

$$k_{rel} = \|A^{-1}\| \frac{\|Ax\|}{\|x\|} \leq \|A^{-1}\| \cdot \|A\| = k(A)$$

und bei komponentenweiser Betrachtung

$$k_{rel} = \frac{\| \, |A^{-1}| \cdot |A| \cdot |x| \, \|_\infty}{\| \, x \, \|_\infty} \leq \| \, |A^{-1}| \cdot |A| \, \|_\infty = k_C(A) \; .$$

Desweiteren folgte für eine Matrix $A \in Mat_n(R)$, welche nicht die Nullmatrix ist, dass

$$\det A = 0 \quad \Leftrightarrow \quad k(A) = \infty$$

Falls $k(A) < \infty$, wäre das lineare Gleichungssystem also für jede rechte Seite im Prinzip eindeutig lösbar.

Andererseits lässt (*) nur dann ein numerisch brauchbares Resultat erwarten, falls $k_{rel} \cdot d$ hinreichend klein ist.

Darüber hinaus muss anstelle einer einzelnen Matrix A die Menge aller von A nicht zu unterscheidenden Matrizen, also z.B.

$$E := \left\{ \tilde{A} : \left\| \tilde{A} - A \right\| \le d \cdot \|A\| \right\}$$

betrachtet werden.

Es bietet sich deshalb an, eine Matrix A mit der relativen Genauigkeit d „fast singulär“ oder „numerisch singulär“ zu nennen, falls die zugehörige Eingabemenge mindestens eine (exakt) singuläre Matrix enthält.

Def. „fast singuläre“ Matrix :

Eine Matrix A heißt ***fast singulär*** oder ***numerisch singulär*** bezüglich der Kondition $k(A)$, falls

$d \cdot k(A) \ge 1$,wobei d die relative Genauigkeit der Matrix A ist.

Für die Rundungsfehler bei der Eingabe von A in einen Rechner wird z.B. angenommen, dass $d = eps$.

Beispiel :

$Ax = b_1$, $A = \begin{pmatrix} 1 & 1 \\ 0 & e \end{pmatrix}$, $b_1 = \begin{pmatrix} 2 \\ e \end{pmatrix}$, wobei $0 < e << 1$ die Eingabe darstellt.

Die Matrix A und die rechte Seite b_1 haben den gemeinsamen Eingabewert e ; sie sind also miteinander verkoppelt.

Daher ist die Kondition der Matrix

$$k(A) = \left\| A^{-1} \right\|_\infty \|A\|_\infty = \frac{1}{e} >> 1$$

nicht aussagekräftig für die Lösung des Problems (f_1, e) , $f_1(e) = A^{-1} b_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$.

Die Lösung ist sogar unabhängig von e , d.h. $f_1'(e) = 0$, und somit ist das Problem für alle e gut konditioniert.

Betrachtet man jedoch eine von e unabhängige rechte Seite $b_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, so folgt für die Lösung

$$f_2(e) = A^{-1} b_2 = x_2 = \begin{pmatrix} 1/e \\ 1/e \end{pmatrix}$$

und für die komponentenweisen Konditionen

$$k_{abs} = \| f_2'(e) \| = \frac{1}{e^2} \quad \text{und} \quad k_{rel} = \frac{\| \, |f_2'(e)| \cdot |e| \, \|}{\| \, f_2(e) \, \|} = 1 \, .$$

In diesem Fall ist nur noch die Richtung der Lösung gut konditioniert,was sich in der relativen Kondition widerspiegelt.

2.3.1 Rückwärtsanalyse der Gauß-Elimination

Wie gesehen, ist die Berechnung des Skalarproduktes in Gleitkommaarithmetik stabil im Sinne der Rückwärtsanalyse.
Die im ersten Kapitel besprochenen Algorithmen zur Lösung eines linearen Gleichungssystems erfordern im Einzelnen nur die Auswertung von Skalarprodukten bestimmter Zeilen und Spalten von Matrizen, so dass sich auf der Grundlage dieser Überlegungen eine Rückwärtsanalyse der Gaußschen Eliminationsmethode durchführen lässt.

Satz 2.7 (Rückwärtsanalyse der Gauß-Elimination)

Die Gleitkommarealisierung der Vorwärtssubstitution zur Lösung eines gestaffelten Gleichungssystems $Lx = b$ berechnet eine Lösung $\hat{x}$, so dass eine untere Dreiecksmatrix $\hat{L}$ existiert mit

$$\hat{L}\hat{x} = b \qquad \text{und} \quad \left|\hat{L} - L\right| \dot{\leq} n \cdot eps \cdot |L| \ ,$$

d.h. für den komponentenweisen relativen Rückwärtsfehler gilt $h \leq n \cdot eps$ und die Vorwärtssubstitution ist stabil im Sinne der Rückwärtsanalyse.

Beweis :

Der Algorithmus für die Vorwärtssubstitution lässt sich wie beim Skalarprodukt rekursiv formulieren :

$$l_{kk} x_k = b_k - \left\langle l^{(k-1)}, x^{(k-1)} \right\rangle \qquad \text{für } k = 1,..,n \text{ , wobei wieder gilt}$$

$$l^{(k-1)} := \left(l_{k,1},..., l_{k,k-1}\right)^T \text{ und } \qquad x^{(k-1)} := \left(x_1,..., x_{k-1}\right)^T$$

Realisiert in Gleitpunktarithmetik ergibt sich die Rekursion

$$l_{kk} \cdot (1+d_k) \cdot (1+e_k) \cdot \hat{x}_k = b_k - \left\langle l^{(k-1)}, \hat{x}^{(k-1)} \right\rangle_{fl}$$

, wobei die d_k und e_k mit $|d_k|, |e_k| \leq eps$ die relativen Fehler der Multiplikation bzw. Addition beschreiben.
Über die Gleitkommarealisierung des Skalarproduktes ist nach Lemma 2.6 bekannt, dass

$$\left\langle l^{(k-1)}, \hat{x}^{(k-1)} \right\rangle_{fl} = \left\langle \hat{l}^{(k-1)}, \hat{x}^{(k-1)} \right\rangle$$

für einen Vektor $\hat{l}^{(k-1)} = \left(\hat{l}_{k,1},..., \hat{l}_{k,k-1}\right)^T$ mit $\left|\hat{l}^{(k-1)} - l^{(k-1)}\right| \leq (k-1) \cdot eps \cdot \left|l^{(k-1)}\right|$.

Setzt man nun noch $\hat{l}_{kk} := l_{kk} \cdot (1+d_k) \cdot (1+e_k)$, so gilt wie behauptet

$$\hat{L}x = b \qquad \text{und} \quad \left|\hat{L} - L\right| \leq n \cdot eps \cdot |L| \ . \qquad \textbf{Q.e.d.}$$

Als erstes Resultat zur Rückwärtsanalyse der Gauß-Elimination wird nun die Qualität der LR-Zerlegung beurteilt :

Lemma 2.8 (Rückwärtsstabilität der LR-Zerlegung)

A besitze eine LR-Zerlegung. Dann berechnet die Gauß-Elimination $\hat{L}$ und $\hat{R}$, so dass

$$\hat{L}\hat{R} = \hat{A} \qquad \text{für eine Matrix } \hat{A} \text{ mit } \left|\hat{A} - A\right| \dot{\leq} n \cdot \left|\hat{L}\right| \cdot \left|\hat{R}\right| \cdot eps \ .$$

ohne Beweis

Satz 2.9 (Rückwärtsstabilität für Gauß-Algorithmus ohne Pivotisierung)

A besitze eine LR-Zerlegung. Dann berechnet das Gaußsche Eliminationsverfahren für das Gleichungssystem $Ax = b$ eine Lösung $\hat{x}$ mit

$$\hat{A}\hat{x} = b \ .$$

ohne Beweis

Daraus leitet sich nun die folgende Aussage über den normweisen Rückwärtsfehler der Gauß-Elimination mit Spaltenpivoting ab, welche auf *Wilkinson* zurückgeht :

Satz 2.10

Die Gauß-Elimination mit Spaltenpivoting für das Gleichungssystem $Ax = b$ berechnet ein $\hat{x}$, so dass

$$\hat{A}\hat{x} = b \qquad \text{für eine Matrix } \hat{A} \text{ mit } \frac{\left\|\hat{A} - A\right\|_\infty}{\|A\|_\infty} \dot{\leq} 2n^3 \cdot r_n(A) \cdot eps$$

, wobei $r_n(A) := \dfrac{a_{\max}}{\max\limits_{i,j}\left|a_{ij}\right|}$ und $a_{\max}$ der größte Betrag eines Elementes ist, welcher im Laufe der Elimination in den Restmatrizen $A^{(1)} = A$ bis $A^{(n)} = R$ auftritt.

Beweis :

Im Folgenden werden die im Laufe der Gauß-Elimination mit Spaltenpivoting $PA = LR$ berechneten Größen mit $\hat{P}, \hat{L}, \hat{R}, \hat{x}$ bezeichnet.

Dann besitzt $\hat{P}A$ eine LR-Zerlegung und nach Satz 2.9 existiert eine Matrix $\hat{A}$, so dass $\hat{A}\hat{x} = b$ und

$$\left|\hat{A} - A\right| \dot{\leq} 2n \cdot \hat{P}^T \cdot \left|\hat{L}\right| \cdot \left|\hat{R}\right| \cdot eps \ .$$

Wendet man hierauf die Supremumsnorm an, folgt wegen $\left\|\hat{P}\right\|_\infty = 1$, dass

$$\left\|\hat{A} - A\right\|_\infty \dot{\leq} 2n \cdot \left\|\hat{L}\right\|_\infty \cdot \left\|\hat{R}\right\|_\infty \cdot eps \ . \qquad \textbf{(*)}$$

Die Spaltenpivotstrategie sorgt dafür, dass alle Komponenten von $\hat{L}$ vom Betrag kleiner oder gleich eins sind, d.h.

$$\left\|\hat{L}\right\|_\infty \leq n \ .$$

Die Norm von $\hat{R}$ lässt sich abschätzen durch

$$\left\|\hat{R}\right\|_\infty \leq n \cdot \max_{i,j}\left|\hat{r}_{ij}\right| \leq n \cdot a_{\max} \ .$$

D.h. $\left\|\hat{A} - A\right\|_\infty \dot{\leq} 2n^3 \cdot a_{\max} \cdot eps$

Da $\max\limits_{i,j}\left|a_{ij}\right| \leq \|A\|_\infty$, folgt die Behauptung $\dfrac{\left\|\hat{A} - A\right\|_\infty}{\|A\|_\infty} \dot{\leq} 2n^3 \cdot \dfrac{a_{\max}}{\max\limits_{i,j}\left|a_{ij}\right|} \cdot eps$. **Q.e.d.**

Die Frage nach der Stabilität lässt sich nach Satz 2.10 nicht eindeutig beantworten. Es hängt offenbar von der Zahl $r_n(A)$ ab, ob sich die Matrix für die Gauß-Elimination eignet oder nicht. Im Allgemeinen lässt sich diese Größe nur durch

$$r_n(A) \leq 2^{n-1}$$

abschätzen, wobei diese Abschätzung scharf ist, da die Grenze für die von Wilkinson angegebene Matrix

$$A_W = \begin{pmatrix} 1 & & & 1 \\ -1 & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ -1 & \cdots & -1 & 1 \end{pmatrix}$$ angenommen wird.

Die Gauß-Elimination mit Spaltenpivoting ist also über die ganze Menge der invertierbaren Matrizen betrachtet nicht stabil. Für spezielle Klassen von Matrizen sieht die Situation jedoch wesentlich besser aus :

Art der Matrix	Spaltenpivoting	$r_n \leq$
invertierbar	Ja	2^{n-1}
obere Hessenberg-Matrix	Ja	n
A oder A^T strikt diagonaldominant	Überflüssig	2
tridiagonal	Ja	2
symmetrisch positiv definit	Nein	1
Statistik	Ja	$n^{2/3}$ (im Mittel)

So ist die Gauß-Elimination für spd-Matrizen (Cholesky-Verfahren) ebenso wie für strikt diagonaldominante Matrizen stabil. Gestützt von statistischen Überlegungen kann man sagen, dass die Gauß-Elimination „in der Regel“, d.h. für üblicherweise in der Praxis auftretende Matrizen, stabil ist.

2.3.2 Beurteilung von Näherungslösungen

Zu einem linearen Gleichungssystem $Ax = b$ liege die Näherungslösung $\tilde{x}$ vor.

Frage : Wie gut ist diese Lösung ?

$\Rightarrow$ Das Residuum $r(x) := b - Ax$ muss möglichst klein sein.

Problem : Die Norm $\|r(\tilde{x})\|$ kann durch eine Zeilenskalierung $Ax = b \quad \rightarrow \quad (D_Z A)x = D_Z b$ beliebig verändert werden, obwohl das Problem selbst dadurch nicht verändert wird.

Dies lässt sich auch an der Invarianz der Skeelschen Kondition $k_C(A)$ bezüglich Zeilenskalierung ablesen.

Das Residuum kann daher höchstens dann zur Beurteilung der Näherungslösung $\tilde{x}$ herangezogen werden, wenn es eine problemspezifische Bedeutung besitzt. Besser geeignet ist das Konzept des Rückwärtsfehlers.
Für eine Näherungslösung $\tilde{x}$ eines LGS lassen sich die normweisen und komponentenweisen Rückwärtsfehler direkt angeben.

Satz 2.11

Der normweise relative Rückwärtsfehler einer Näherungslösung $\tilde{x}$ eines linearen Gleichungssystems $f(A,b) = f^{-1}b$ bezüglich $\|(A,b)\| := \|A\| + \|b\|$ ist

$$h_N(\tilde{x}) = \frac{\|A\tilde{x} - b\|}{\|A\| \cdot \|\tilde{x}\| + \|b\|} .$$

Beweis : Fachliteratur

Satz 2.12 (Prager & Oettli)

Der komponentenweise relative Rückwärtsfehler einer Näherungslösung $\tilde{x}$ von $Ax = b$ ist

$$h_C(\tilde{x}) = \max_i \frac{|A\tilde{x} - b|_i}{(|A| \cdot |\tilde{x}| + b)_i} .$$

Beweis : Buch S.58

3. Lineare Ausgleichsprobleme

Verallgemeinerung des Lösungsbegiffes auf rechteckige (überbestimmte) Gleichungssysteme.

3.1 Gaußsche Methode der kleinsten Fehlerquadrate

3.1.1 Problemstellung :

Gegeben seien m Messpunkte (t_i, b_i) , die z.B. die Zustände b_i eines Objektes zu den Zeiten t_i beschreiben.
Es wird angenommen, dass diesen Messungen eine Gesetzmäßigkeit zugrunde liegt, so dass sich die Abhängigkeit von b von t durch eine Modellfunktion j mit $b(t) = j(t; x_1, ..., x_n)$ ausdrücken lässt, wobei in die Modellfunktion n unbekannte Parameter x_i eingehen.

Beispiel : Ohmsches Gesetz

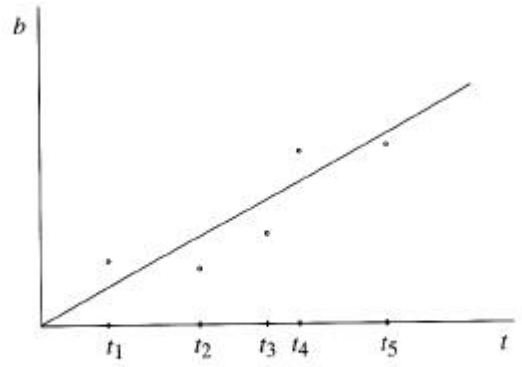


Abbildung 3.1. Lineare Ausgleichsrechnung beim Ohmschen Gesetz

Aufgabe : Eine Gerade durch den Nullpunkt legen, die dem Verlauf der Messungen „möglichst nahe“ kommt.
Die Messungen sind in der Regel fehlerbehaftet, und auch die Modellfunktionen stellen stets nur eine ungefähre Beschreibung der Wirklichkeit dar. (So gilt das Ohmsche Gesetz annähernd nur in einem mittleren Temperaturbereich.)
D.h. es wird gefordert, dass $b_i \approx j(t_i; x_1, ..., x_n)$ (für $i = 1, ..., m$). Nun gibt es mehrere Möglichkeiten, die einzelnen Abweichungen $\Delta_i := b_i - j(t_i; x_1, ..., x_n)$ zu gewichten.

Aufgrund wahrscheinlichkeitstheoretischer Überlegungen wählte Gauß die Quadrate Δ_i^2 aus.

D.h. $\Delta^2 := \sum_{i=1}^{m} \Delta_i^2 = \min$. **(*)**

Bemerkung : Äquivalenz zum Maximierungsproblem $e^{-\Delta^2} = \max$. Der Exponentialterm charakterisiert hierbei die Gaußsche Normalverteilung $f(x) = \frac{1}{s\sqrt{2p}} e^{-\frac{(x-m)^2}{2s^2}}$. $\rightarrow$ Maximum-Likelihood-Methode.

In der Form (*) sind die Fehler der einzelnen Messungen alle gleich gewichtet.
Normalerweise sind die Messungen (t_i, b_i) jedoch unterschiedlich genau. Zu jeder einzelnen Messung b_i gehört also eine absolute Messgenauigkeit bzw. Toleranz db_i .

$\Rightarrow$ Wichtung der einzelnen Fehler mit der Toleranz : $\sum_{i=1}^{m} \left(\frac{\Delta_i}{db_i} \right)^2 = \min$.

3.1.2 Normalgleichungen

Geometrisch gesprochen sucht man bei der Lösung des linearen Ausgleichsproblems nach einem Punkt $z = Ax$ aus dem Bildraum $R(A)$ von A, der den kleinsten Abstand zu dem gegebenen Punkt b hat.

Für $m = 2$ und $n = 1$ ist $R(A) \subset R^2$ entweder nur der Nullpunkt oder eine Gerade durch den Nullpunkt. (siehe Abbildung)

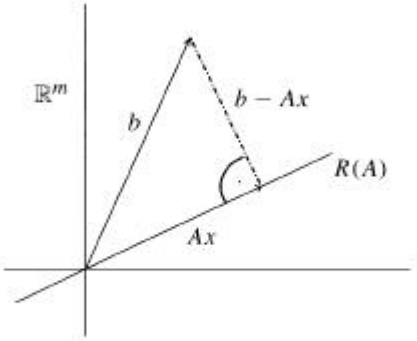


Abbildung 3.3. Projektion auf den Bildraum $R(A)$

Anschaulich ist klar, dass die Differenz $b - Ax$ gerade senkrecht auf dem Unterraum $R(A)$ stehen muss, damit der Abstand $\|b - Ax\|$ minimal ist.

Mit anderen Worten : Ax ist die **orthogonale** Projektion von b auf den Unterraum $R(A)$.

Satz 3.1

Sei V ein endlichdimensionaler euklidscher Vektorraum mit Skalarprodukt $\langle \cdot, \cdot \rangle$, $U \subset V$ ein Unterraum und

$$U^T = \{v \in V : \langle v, u \rangle = 0 \quad \textit{für alle } u \in U\}$$ sein **orthogonales Komplement** in V.

Dann gilt für alle $v \in V$ bezüglich der von der Skalarprodukt induzierten Norm $\|v\| = \sqrt{\langle v, v \rangle}$, dass

$$\|v - u\| = \min_{u' \in U} \|v - u'\| \quad \Leftrightarrow \quad v - u \quad \in U^T$$

Beweis : Buch S.70

Bemerkung 3.2

Damit ist die Lösung $u \in U$ von $\|v - u\| = \min$ eindeutig bestimmt und heißt **orthogonale Projektion** von v auf U.

Die Abbildung $P : V \to U$, $v \mapsto P \cdot v$ mit $\|v - Pv\| = \min_{u \in U} \|v - u\|$

ist linear und wird **orthogonale Projektion** von V auf U genannt.

Satz 3.3

Der Vektor $x \in R^n$ ist genau dann Lösung des linearen Ausgleichsproblems $\|b - Ax\| = \min$, falls er die sogenannten **Normalgleichungen** $A^T Ax = A^T b$ erfüllt.
Insbesondere ist das lineare Ausgleichsproblem genau dann eindeutig lösbar, wenn der Rang von A maximal ist, d.h. $Rang(A) = n$.

Beweis :

Satz 3.1 angewandt auf $V = R^m$ und $U = R(A)$ gilt

$$\|b - Ax\| = \min \Leftrightarrow \langle b - Ax, Ax' \rangle = 0 \quad \text{für alle } x' \in R^n$$
$$\Leftrightarrow \langle A^T(b - Ax), x' \rangle = 0 \quad \text{für alle } x' \in R^n$$
$$\Leftrightarrow A^T(b - Ax) = 0$$
$$\Leftrightarrow A^T Ax = A^T b$$

und daher die erste Aussage.
Der zweite Teil folgt nun aus der Tatsache, dass $A^T A$ genau dann invertierbar ist, wenn $Rang(A) = n$. **Q.e.d.**

Bemerkung 3.4

Geometrisch besagen die Normalgleichungen gerade, dass $b - Ax$ eine Normale auf $R(A) \subset R^m$ ist. Daher der Name.

3.1.3 Kondition des Projektionsproblems

Die relative Kondition des Projektionsproblems (P, b) bezüglich der Eingabe b hängt offenbar stark von dem Winkel J ab, den b mit dem Unterraum V einschließt.

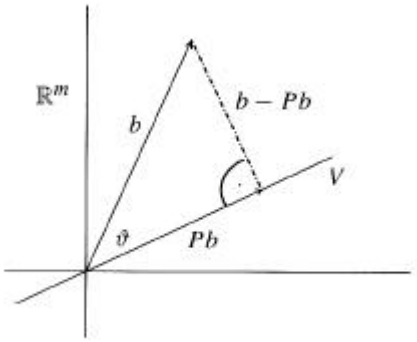


Abbildung 3.4. Projektion auf den Unterraum V

Ist der Winkel klein, d.h. $b \approx Pb$, so finden sich Störungen von b nahezu unverändert im Ergebnis Pb wieder. Andererseits äußert sich eine kleine Störung von b in großen relativen Schwankungen von Pb, wenn b fast senkrecht auf V steht.

Lemma 3.5

Sei $P : R^m \to V$ die orthogonale Projektion auf einen Unterraum V des R^n.

Für die Eingabe b bezeichne J den Winkel zwischen b und V, d.h. $\sin J = \frac{\|b - Pb\|_2}{\|b\|_2}$.

Dann gilt für die relative Kondition des Problems (P, b) bezüglich der euklidschen Norm, dass

$$k = \frac{1}{\cos J}\|P\|_2 .$$

Beweis :

Nach Pythagoras gilt $\|Pb\|^2 = \|b\|^2 - \|b - Pb\|^2$ und daher $\frac{\|Pb\|^2}{\|b\|^2} = 1 - \sin^2 J = \cos^2 J$.

Da P linear ist, ergibt sich daraus für die relative Kondition von (P, b) wie behauptet

$$k = \frac{\|b\|}{\|Pb\|}\|P'(b)\| = \frac{\|b\|}{\|Pb\|}\|P\| = \frac{1}{\sqrt{\frac{\|Pb\|^2}{\|b\|^2}}} = \frac{1}{\cos J}\|P\| .$$ **Q.e.d.**

Lemma 3.6

Für eine Matrix $A \in Mat_{m,n}(R)$ von maximalem Rang $p = n$ gilt

$$k_2(A^T A) = k_2(A)^2 .$$

Beweis :

Nach der Definition der Kondition einer rechteckigen Matrix gilt

$$k_2(A)^2 = \frac{\max_{\|x\|_2=1}\|Ax\|_2}{\min_{\|x\|_2=1}\|Ax\|_2} = \frac{\max_{\|x\|_2=1}\langle A^T Ax, x\rangle}{\min_{\|x\|_2=1}\langle A^T Ax, x\rangle} = \frac{l_{\max}(A^T A)}{l_{\min}(A^T A)} = k_2(A^T A)$$ **Q.e.d.**

Satz 3.7 (Kondition des linearen Ausgleichsproblems)

Sei $A \in Mat_{m,n}(R)$, $m \geq n$ eine Matrix von vollem Spaltenrang, $b \in R^n$, und x die (eindeutige) Lösung des Linearen Ausgleichsproblems $\|b - Ax\|_2 = \min$.

Es sei vorausgesetzt, dass $x \neq 0$ und J bezeichne den Winkel zwischen b und dem Bildraum $R(A)$ von A, d.h. $\sin J = \frac{\|b - Ax\|_2}{\|b\|_2} = \frac{\|r\|_2}{\|b\|_2}$ mit dem Residuum $r = b - Ax$. Dann gilt für die relative Kondition von x in der euklidschen Norm

- bezüglich Störungen in b

$$k \leq \frac{k_2(A)}{\cos J}$$

- bezüglich Störungen in A

$$k \leq k_2(A) + k_2(A)^2 \cdot \tan J \quad .$$

Beweis : Buch S.73/74

3.1.4 Lösung der Normalgleichungen

Geht man davon aus, dass das lineare Ausgleichsproblem eindeutig lösbar ist, d.h. $Rang(A) = n$, so ist $A^T A$ eine spd-Matrix. Daher bietet sich das Cholesky-Verfahren an. Für den Aufwand zur Lösung des linearen Ausgleichsproblems mit Hilfe der Normalgleichungen ergibt sich dann (Anzahl der Multiplikationen) :

- a) Berechnung von $A^T A$: $\sim \frac{1}{2} n^2 m$
- b) Cholesky-Zerlegung von $A^T A$: $\sim \frac{1}{6} n^3$

Für $m >> n$ überwiegt der Anteil von a), so das der Aufwand insgesamt

$\sim \frac{1}{2} n^2 m$ für $m >> n$ und $\sim \frac{2}{3} n^3$ für $m \approx n$ beträgt.

Die Lösung linearer Ausgleichsprobleme über die Normalgleichungen mit Hilfe der Cholesky-Zerlegung kann nur für große Residuen empfohlen werden, da bei kleinen Residuen der Übergang zu den Normalgleichungen eine Verschlechterung der Kondition bedeutet.

3.2 Orthogonalisierungsverfahren

Jeder Eliminationsprozess für lineare Gleichungssysteme, wie z.B. die Gauß-Elimination, lässt sich formal in der Form

$$A \xrightarrow{f_1} B_1 A \xrightarrow{f_2} B_2 B_1 A \xrightarrow{f_3} \ldots \xrightarrow{f_k} B_k \ldots B_1 A = R$$

darstellen, wobei die Matrizen B_j die Operationen auf der Matrix A beschreiben.

Bei der rekursiven Stabilitätsanalyse in Kapitel 2 folgte, dass die Stabilitätsindikatoren der Teilschritte eines Algorithmus verstärkt werden durch die Konditionen aller folgenden Teilschritte.

Beim oben beschriebenen Eliminationsprozess sind z.B. die Konditionen der Eliminationsmatrizen $B_j = L_j$ der Gaußschen Dreieckszerlegung nicht nach oben beschränkt, so dass es dabei zu Instabilitäten kommen kann.

Wählt man hingegen statt der L_j orthogonale Transformationen Q_j für die Elimination, so gilt bzgl. der euklidschen Norm, dass

$$k(Q_j) = \|Q_j\|_2 \cdot \|Q_j^{-1}\|_2 = \|Q_j\|_2 \cdot \|Q_j^T\|_2 = 1$$

\+ Diese sogenannten **Orthogonalisierungsverfahren** sind also auf jeden Fall stabil.
\- Leider ist diese Stabilität mit einem etwas höheren Aufwand verbunden als z.B. bei der Gauß-Elimination.
\+ Aufgrund der Invarianz der euklidschen Norm bezüglich orthogonaler Transformationen eignen sich Orthogonalisierungsverfahren zur Lösung linearer Ausgleichsprobleme.

Angenommen, man hätte die Matrix $A \in Mat_{m,n}(R)$ mit $m \geq n$ mit einer orthogonalen Matrix $Q \in O(m)$ auf obere Dreiecksgestalt $Q^T A = \begin{bmatrix} * & \ldots & * \\ & \ddots & \vdots \\ & & * \\ & & \end{bmatrix} = \begin{bmatrix} R \\ 0 \end{bmatrix}$ gebracht. ($R =$ obere Dreiecksmatrix)

Dann lässt sich die Lösung des linearen Ausgleichsproblems $\|b - Ax\| = \min$ als Alternative zur Lösung der Normalgleichungen auch wiefolgt bestimmen :

Satz 3.8

Sei $A \in Mat_{m,n}(R)$ mit $m \geq n$ und von maximalem Rang , $b \in R^m$, und $Q \in O(m)$ eine orthogonale Matrix mit $Q^T A = \begin{pmatrix} R \\ 0 \end{pmatrix}$ und $Q^T b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$, wobei $b_1 \in R^n$, $b_2 \in R^{m-n}$ und $R \in Mat_n(R)$ eine (invertierbare) obere Dreiecksmatrix ist. Dann ist $x = R^{-1} b_1$ die Lösung des linearen Ausgleichsproblems $\|b - Ax\| = \min$.

Beweis :

Da $Q \in O(m)$, gilt für alle $x \in R^m$

$$\|b - Ax\|^2 = \|Q^T (b - Ax)\|^2 = \left\| \begin{pmatrix} b_1 - Rx \\ b_2 \end{pmatrix} \right\|^2 = \|b_1 - Rx\|^2 + \|b_2\|^2 \geq \|b_2\|^2 .$$

Wegen $Rang(A) = Rang(R) = n$ ist R invertierbar. Der erste Summand $\|b_1 - Rx\|^2$ verschwindet daher genau für $x = R^{-1} b_1$. Das Residuum $r := b - Ax$ verschwindet im Allgemeinen nicht , und es ist $\|r\| = \|b_2\|$. **Q.e.d.**

Die in Frage kommenden orthogonalen Transformationen lassen sich für $m = 2$ leicht geometrisch ableiten, nämlich als Drehungen (Rotationen) bzw. Spiegelungen (Reflexionen).

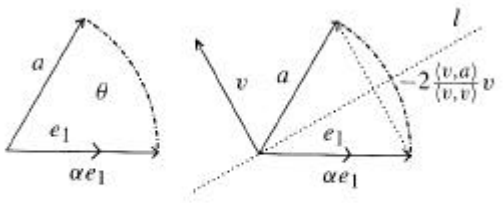


Abbildung 3.5. Drehung bzw. Spiegelung von a auf αe_1

Will man den Vektor $a \in R^2$ mit einer orthogonalen Transformation auf ein Vielfaches $a \cdot e_1$ des ersten Einheitsvektors abbilden, so folgt $a = \|a\|$.

- 1. Möglichkeit : a um den Winkel q auf $a \cdot e_1$ drehen , d.h.

$$a \mapsto a \cdot e_1 = Q \cdot a \qquad \text{mit } Q := \begin{pmatrix} \cos q & \sin q \\ -\sin q & \cos q \end{pmatrix}$$

- 2. Möglichkeit : a an der auf dem Vektor v senkrecht stehenden Geraden l spiegeln, d.h.

$$a \mapsto a \cdot e_1 = a - 2 \frac{\langle v, a \rangle}{\langle v, v \rangle} v$$, wobei v kollinear zur Differenz $a - a \cdot e_1$ ist.

3.2.1 Givens-Rotationen

Definition : Als Givens-Rotationen bezeichnet man Matrizen der Form

$$\Omega_{kl} := \begin{bmatrix} I & & & & \\ & c & & s & \leftarrow k \\ & & I & & \\ & -s & & c & \leftarrow l \\ & & & & I \end{bmatrix} \in Mat_m(R)$$

, wobei I jeweils die Einheitsmatrix der passenden Dimension ist und $c^2 + s^2 = 1$.
(Dabei sollen c und s an $\cos q$ und $\sin q$ erinnern.)

Geometrisch beschreibt die Matrix eine Drehung um den Winkel q in der $(k-l)-$Ebene.

- Wendet man Ω_{kl} auf einen Vektor $x \in R^m$ an, so folgt

$$x \mapsto y = \Omega_{kl} x \quad \text{mit} \quad y_i = (\Omega_{kl} x)_i = \begin{cases} cx_k + sx_l & , falls\ i = k \\ -sx_k + cx_l & , falls\ i = l \\ x_i & , falls\ i \neq k,l \end{cases} \qquad \textbf{(*)}$$

- Multipliziert man eine Matrix $A = [A_1,...,A_n] \in Mat_{m,n}(R)$ von links mit Ω_{kl}, so operiert die Givens-Rotation auf den Spalten, d.h.
  $\Omega_{kl} A = [\Omega_{kl} A_1,...,\Omega_{kl} A_n]$.
  Es werden daher wegen (*) nur die beiden Zeilen k und l der Matrix A verändert.
  Dies ist insbesondere wichtig, wenn man bei der Transformation möglichst Besetzungsstrukturen der Matrix erhalten möchte.

Wie sind nun die Koeffizienten c und s zu bestimmen, um eine Komponente x_l des Vektors x zu eliminieren ?

Da Ω_{kl} nur auf der $(k,l)-$Ebene operiert, genügt es, das Prinzip an dem Fall $m = 2$ zu erläutern.

Mit $x_k^2 + x_l^2 \neq 0$ und $c^2 + s^2 = 1$ gilt :

$$\begin{pmatrix} c & s \\ -s & c \end{pmatrix} \begin{pmatrix} x_k \\ x_l \end{pmatrix} = \begin{pmatrix} cx_k + sx_l \\ -sx_k + cx_l \end{pmatrix} = \begin{pmatrix} r \\ 0 \end{pmatrix}$$

$$\Rightarrow II.: x_k = \frac{cx_l}{s}$$

$$\Rightarrow in\ I.\ einsetzen: c^2 \frac{x_l}{s} + sx_l = r \quad \Rightarrow x_l(c^2 + s^2) = rs \quad \Rightarrow s = \frac{x_l}{r}$$

$$\Rightarrow in\ II.\ einsetzen: -\frac{x_l x_k}{r} + cx_l = 0 \quad \Rightarrow c = \frac{x_k}{r}$$

- für $|x_l| > |x_k|$: $\quad t := \dfrac{x_k}{x_l}$, $\quad s := \dfrac{1}{\sqrt{1+t^2}}$, $\quad c := s \cdot t$
- für $|x_l| \leq |x_k|$: $\quad t := \dfrac{x_l}{x_k}$, $\quad c := \dfrac{1}{\sqrt{1+t^2}}$, $\quad s := c \cdot t$

Damit vermeidet man zugleich Exponentenüberlauf.

Beispiel :
Spaltenweise werden nun die von Null verschiedenen Matrixkomponenten unterhalb der Diagonalen eliminiert.
(Die Indexpaare über den Pfeilen geben die Indizes der ausgeführten Givens-Rotation an.)

$$A = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \xrightarrow{(5,4)} \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ 0 & * & * & * \end{bmatrix} \xrightarrow{(4,3)} \ldots \xrightarrow{(2,1)} \begin{bmatrix} * & * & * & * \\ 0 & * & * & * \\ 0 & * & * & * \\ 0 & * & * & * \\ 0 & * & * & * \end{bmatrix} \qquad \xrightarrow{(5,4)} \begin{bmatrix} * & * & * & * \\ 0 & * & * & * \\ 0 & * & * & * \\ 0 & * & * & * \\ 0 & 0 & * & * \end{bmatrix} \xrightarrow{(4,3)} \ldots \xrightarrow{(5,4)} \begin{bmatrix} * & * & * & * \\ 0 & * & * & * \\ 0 & 0 & * & * \\ 0 & 0 & 0 & * \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Der Aufwand für die QR-Zerlegung einer vollbesetzten Ausgangsmatrix $A \in Mat_{m,n}$ beträgt :

- $m \approx n$ $:\sim \frac{n^2}{2}$ Quadratwurzeln und $\sim \frac{4n^3}{3}$ Multiplikationen
- $m >> n$ $:\sim mn$ Quadratwurzeln und $\sim 2mn^2$ Multiplikationen

Für $m = n$ erhält man somit eine Alternative zur Gaußschen Dreieckszerlegung aus Kapitel 1.

Die größere Stabilität muss jedoch mit einem erheblich höheren Aufwand, $\sim \frac{4n^3}{3}$ Multiplikationen gegenüber $\sim \frac{n^3}{3}$ bei Gauß, erkauft werden. Zu beachten ist aber, dass der Vergleich für dünnbesetzte Matrizen wesentlich günstiger ausfällt.
Praktisch verwendet man sogenannte schnelle Givens-Rotationen, welche die Auswertung der Quadratwurzeln vermeiden und eine QR-Zerlegung einer zeilenskalierten Matrix DA berechnen.

3.2.2 Householder-Reflexionen

Definition : Als Householder-Reflexionen bezeichnet man Matrizen $Q \in Mat_n(R)$ der Form

$$Q = I - 2\frac{vv^T}{v^T v} \qquad \text{mit } v \in R^n .$$

Diese Matrizen beschreiben die Reflexion an der auf v senkrecht stehenden Ebene. (siehe vorige Abbildung)
Insbesondere hängt Q nur von der Richtung von v ab.
Eigenschaften :

a) Q ist symmetrisch, d.h. $Q^T = Q$
b) Q ist orthogonal, d.h. $QQ^T = Q^T Q = I$
c) Q ist involutorisch, d.h. $Q^2 = I$

- Wendet man Q auf einen Vektor $y \in R^n$ an, so gilt

$$y \mapsto Q \cdot y = \left(I - 2\frac{vv^T}{v^T v} \right) y = y - 2\frac{\langle v, y \rangle}{\langle v, v \rangle} v \ .$$

Soll y auf ein Vielfaches $a \cdot e_1$ des ersten Einheitsvektors e_1 abgebildet werden, d.h.

$a \cdot e_1 = y - 2\frac{\langle v, y \rangle}{\langle v, v \rangle} v \in span(e_1)$, so folgt

$|a| = \|y\|_2$ und $v \in span(y - a \cdot e_1)$

Daher lässt sich Q bestimmen durch $v := y - a \cdot e_1$ mit $a = \pm \|y\|_2$.
Um Auslöschung bei der Berechnung von $v = (y_1 - a, y_2, ..., y_n)^T$ zu vermeiden, wählt man $a := -\mathrm{sgn}(y_1) \cdot \|y\|_2$.
Wegen $\langle v, v \rangle = \langle y - a \cdot e_1, y - a \cdot e_1 \rangle = \|y\|_2^2 - 2a \langle y, e_1 \rangle + a^2 = -2a(y_1 - a)$ lässt sich Qx für beliebige $x \in R^n$ am einfachsten berechnen durch

$$Qx = x - 2\frac{\langle v, x \rangle}{\langle v, v \rangle} v = x + \frac{\langle v, x \rangle}{a(x_1 - a)} v$$

- Transformiert man mit Hilfe der Householder-Reflexionen eine Matrix $A = [A_1, ..., A_n] \in Mat_{m,n}(R)$ in eine obere Dreiecksmatrix, so werden sukzessive die Elemente unterhalb der Diagonalen eliminiert.

Im ersten Schritt : $A \to A' := Q_1 A = \begin{bmatrix} a_1 & & & \\ 0 & & & \\ \vdots & A_2' & \cdots & A_n' \\ 0 & & & \end{bmatrix}$,

wobei $Q_1 = I - 2\frac{v_1 v_1^T}{v_1^T v_1}$ mit $v_1 := A_1 - a_1 e_1$ und $a_1 := -\mathrm{sgn}(a_{11}) \cdot \|A_1\|_2$.

Nach dem k-ten Schritt wurde somit die Ausgangsmatrix A bis auf eine Restmatrix $T^{(k+1)} \in Mat_{m-k,n-k}(R)$ auf obere Dreiecksgestalt gebracht.

$$A^{(k)} = \begin{bmatrix} * & \cdots & \cdots & \cdots & * \\ & \ddots & & & \vdots \\ & & * & \cdots & * \\ & & 0 & & \\ & & \vdots & T^{(k+1)} & \\ & & 0 & & \end{bmatrix}$$

Bildet man nun die orthogonale Matrix $Q_{k+1} = \left[\begin{array}{c|c} I_k & 0 \\ \hline 0 & \overline{Q}_{k+1} \end{array}\right]$,

wobei $\overline{Q}_{k+1} \in O(m-k)$ wie im ersten Schritt mit $T^{(k+1)}$ anstelle von A konstruiert wird, so lässt sich die nächste Spalte unterhalb der Diagonalen eliminieren.

Insgesamt erhält man so nach $p = \min(m-1, n)$ Schritten die obere Dreiecksmatrix

$$R = Q_p ... Q_1 A$$

und somit wegen $Q_i^2 = I$ die Zerlegung

$$A = QR \qquad \text{mit} \qquad Q = Q_1 ... Q_p \ .$$

Berechnet man nun also die Lösung des linearen Ausgleichsproblems $\|b - Ax\| = \min$ nach Satz 3.8 , indem man mit Hilfe der Householder-Reflexionen $Q_j \in O(m)$ die QR-Zerlegung der Matrix $A \in Mat_{m,n}(R)$ mit $m \geq n$ berechnet, so gelangt man zu folgendem Verfahren :

1) $A = QR$, QR-Zerlegung mit Householder-Reflexionen
2) $(b_1, b_2)^T = Q^T b$ mit $b_1 \in R^n$ und $b_2 \in R^{m-n}$, Transformation von b
3) $Rx = b_1$, Auflösung des gestaffelten Systems

Speicherschema :

Speichert man die Diagonalelemente $r_{ii} = a_i$ für $i = 1,..., p$ in einem separaten Vektor, so finden die Householder-Vektoren $v_1,...,v_p$ in der unteren Hälfte von A Platz.

Eine andere Möglichkeit besteht darin, die Householder-Vektoren so zu normieren, dass die erste Komponente $\langle v_i, e_i \rangle$ jeweils 1 ist und nicht abgespeichert werden muss.

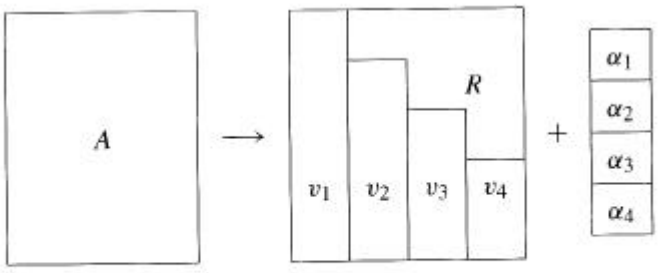


Abbildung 3.6. Speicheraufteilung bei der *QR*-Zerlegung mit Householder-Reflexionen für $m = 5$ und $n = 4$

Aufwand :

a) $m >> n$: $\sim 2n^2 m$ Multiplikationen

b) $m \approx n$: $\frac{2}{3} n^3$ Multiplikationen

Für $m \approx n$ benötigt man also in etwa den gleichen Aufwand wie bei dem Cholesky-Verfahren für Normalgleichungen.
Für $m >> n$ schneidet die QR-Zerlegung um Faktor 2 schlechter ab, hat aber die oben diskutierten Stabilitätsvorteile.

3.3 Verallgemeinerte Inverse

Die nach Satz 3.3 eindeutig bestimmte Lösung x des linearen Ausgleichsproblems $\|b - Ax\| = \min$ für $A \in Mat_{m,n}(R)$, $m \geq n$ und $Rang(A) = n$ wird formal mit $x = A^+ b$ bezeichnet.

Aus den Normalgleichungen folgt unter obigen Voraussetzungen, dass

$$A^T Ax = A^T b \Rightarrow A^T AA^+ b = A^T b \Rightarrow A^+ b = (A^T A)^{-1} A^T b$$
$$\Rightarrow A^+ = (A^T A)^{-1} A^T$$

Da $A^+ A = (A^T A)^{-1} A^T A = I$ gerade die Identität ergibt, nennt man A^+ auch die **Pseudoinverse** von A .

Bei beliebigen Matrizen $A \in Mat_{m,n}(R)$ ist die Lösung von $\|b - Ax\| = \min$ jedoch im Allgemeinen nicht mehr eindeutig bestimmt.

Es bezeichne $\overline{P} : R^m \to R(A) \subset R^m$ die orthogonale Projektion von R^m auf den Bildraum $R(A)$.

Dann bilden die Lösungen nach Satz 3.1 einen affinen Unterraum

$$L(b) := \{x \in R^n : \|b - Ax\| = \min\} = \{x \in R^n : Ax = \overline{P}b\} .$$

Um dennoch Eindeutigkeit zu erzwingen, wählt man die bezüglich der euklidschen Norm kleinste Lösung $x \in L(b)$, welche wieder mit $x = A^+ b$ bezeichnet werde.

Nun ist x gerade die orthogonale Projektion des Ursprungs $0 \in R^n$ auf den affinen Unterraum $L(b)$. (siehe Abbildung)

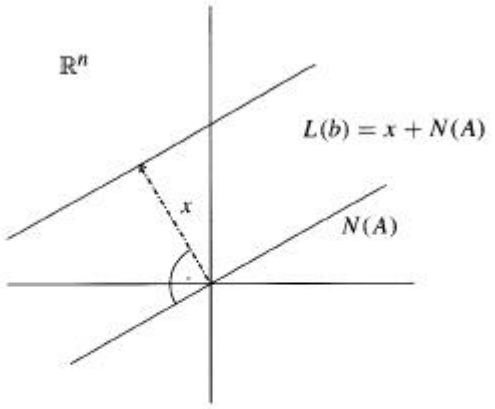


Abbildung 3.7. „Kleinste" Lösung des linearen Ausgleichsproblems als Projektion von 0 auf $L(b)$

Ist $\bar{x} \in L(b)$ irgendeine Lösung von $\|b - Ax\| = \min$, so erhält man alle Lösungen, indem man den Nullraum $N(A)$ von A um $\bar{x}$ verschiebt, d.h.

$$L(b) = \bar{x} + N(A) .$$

Daher muss die kleinste Lösung x senkrecht auf dem Nullraum $N(A)$ stehen, d.h. x ist der eindeutig bestimmte Vektor $x \in N(A)^{\perp}$ mit $\|b - Ax\| = \min$.

Def. Pseudoinverse :

Die Pseudoinverse einer Matrix $A \in Mat_{m,n}(R)$ ist die Matrix $A^+ \in Mat_{n,m}(R)$, so dass für alle $b \in R^m$ der Vektor $x = A^+ b$ die kleinste Lösung von $\|b - Ax\| = \min$ ist, d.h.

$$A^+ b \in N(A)^{\perp} \qquad \text{und} \qquad \|b - AA^+ b\| = \min .$$

Die Situation lässt sich durch das folgende kommutative Diagramm darstellen :

$$R^n \underset{A^+}{\overset{A}{\rightleftarrows}} R^m$$

$$P = A^+ A \quad \downarrow\uparrow i \qquad\qquad i \uparrow\downarrow \quad \overline{P} = AA^+$$

$$R(A^+) = N(A)^{\perp} \quad \cong \quad R(A)$$

Satz 3.9 (Penrose-Axiome)

Die Pseudoinverse $A^+ \in Mat_{n,m}(R)$ einer Matrix $A \in Mat_{m,n}(R)$ ist eindeutig charakterisiert durch folgende Eigenschaften :

1.) $(A^+A)^T = A^+A$
2.) $(AA^+)^T = AA^+$
3.) $A^+AA^+ = A^+$
4.) $AA^+A = A$

Beweis : Buch S.87

Bemerkung 3.10 :

Gilt nur ein Teil der Penrose-Axiome, so spricht man von einer **verallgemeinerten Inversen**.

Nun die Herleitung, wie sich die kleinste Lösung $x = A^+b$ für eine beliebige Matrix $A \in Mat_{n,m}(R)$ und $b \in R^m$ mit Hilfe der QR-Zerlegung berechnen lässt.

Sei $p := Rang(A) \le \min(m,n)$ der Rang der Matrix A.
Zur Vereinfachung wird auf Permutationen verzichtet und A durch orthogonale Transformationen $Q \in O(m)$ auf obere Dreiecksgestalt gebracht, d.h. $QA = \left[\begin{array}{c|c} R & S \\ \hline 0 & 0 \end{array}\right]$,

hier ist $R \in Mat_p(R)$ eine invertierbare obere Dreiecksmatrix und $S \in Mat_{p,n-p}(R)$.

Zerlegung der Vektoren x und Qb analog in

$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \quad \text{mit} \quad x_1 \in R^p \quad \text{und} \quad x_2 \in R^{n-p}$$

$$Qb = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \quad \text{mit} \quad b_1 \in R^p \quad \text{und} \quad b_2 \in R^{m-p}$$

Lemma 3.11

Mit den obigen Bezeichnungen ist x genau dann eine Lösung von $\|b - Ax\| = \min$, falls

$$x_1 = R^{-1}b_1 - R^{-1}Sx_2 \ .$$

Beweis :

Aufgrund der Invarianz der euklidschen Norm unter orthogonalen Transformationen gilt

$$\|b - Ax\|^2 = \|Qb - QAx\|^2 = \|Rx_1 + Sx_2 - b_1\|^2 + \|b_2\|^2 .$$

Der Ausdruck wird genau dann minimal, wenn $Rx_1 + Sx_2 - b_1 = 0$. **Q.e.d.**

Der Fall $p = Rang(A) = n$ entspricht dem bereits behandelten überbestimmten Gleichungssystem mit vollem Rang.
Die Matrix S verschwindet, und man erhält wie in Satz 3.8 die Lösung $x = x_1 = R^{-1}b_1$.
Für die Lösung im rangdefekten Fall $p < n$ gilt :

Lemma 3.12

Sei $p < n$, $V := R^{-1}S \in Mat_{p,n-p}(R)$ und $u := R^{-1}b_1 \in R^p$.

Dann ist die kleinste Lösung x von $\|b - Ax\| = \min$ gegeben durch $x = (x_1, x_2) \in R^p \times R^{n-p}$ mit

$$(I + V^T V) \cdot x_2 = V^T u \qquad \text{und} \qquad x_1 = u - Vx_2 .$$

Beweis : Buch S.88

Für die Berechnung von x_2 kann man die Cholesky-Zerlegung benutzen, da $I + V^T V$ eine spd-Matrix ist.

Zusammengefasst erhält man folgendes Verfahren zur Berechnung der kleinsten Lösung $x = A^+ b$ von $\|b - Ax\| = \min$:

Algorithmus Pseudoinverse über QR-Zerlegung :

1.) QR-Zerlegung von A mit $p = Rang(A)$, wobei $Q \in O(m)$, $R \in Mat_p(R)$ obere Dreiecksmatrix und $S \in Mat_{p,n-p}(R)$

2.) Berechnung von $V \in Mat_{p,n-p}(R)$ aus $R \cdot V = S$

3.) Cholesky-Zerlegung von $I + V^T V$ $\left(I + V^T V = LL^T\right)$, wobei $L \in Mat_{n-p}(R)$ untere Dreiecksmatrix

4.) $(b_1, b_2)^T := Qb$ mit $b_1 \in R^p$, $b_2 \in R^{m-p}$

5.) Berechnung von $u \in R^p$ aus $R \cdot u = b_1$

6.) Berechnung von $x_2 \in R^{n-p}$ aus $LL^T x_2 = V^T u$

7.) Setze $x_1 := u - Vx_2$

Dann ist $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = A^+ b$.

Bemerkung : Die Schritte 1.) bis 3.) müssen für verschiedene rechte Seiten b nur einmal durchgeführt werden.

4. Iterative Lösung linearer Gleichungssysteme

Die bisher beschriebenen sogenannten **direkten Verfahren zur Lösung eines LGS** (Gauß-Elimination, Cholesky-Zerlegung, QR-Zerlegung mit Householder- oder Givens-Transformationen) haben folgende Eigenschaften :

1.) Die Verfahren gehen von beliebigen (bei der Cholesky-Zerlegung symmetrischen) vollbesetzten Matrizen aus.

2.) Der Aufwand zur Lösung des Gleichungssystems liegt bei $O(n^3)$ (Multiplikationen) .

Dem gegenüber stehen jedoch in vielen Fällen große aber schwach besetzte Koeffizientenmatrizen (sparse matrix) .
Auch haben große Gleichungssysteme $(n > 10^4)$ häufig eine klare strukturierte Koeffizientenmatrix
(z.B. Bandmatrizen , Blockmatrizen) .

Die direkten Verfahren sind zur Behandlung derartiger Probleme ungeeignet; sie nutzen die spezielle Struktur nicht aus und dauern bei weitem zu lange.

Ansatz 1 :

Die Spezialstruktur der Matrix, insbesondere ihre Besetzungsstruktur (sparsity pattern) so weit wie möglich in direkten Verfahren ausnutzen.

Die Givens-Rotationen operieren jeweils nur auf zwei Zeilen (von links) oder Spalten (von rechts) einer Matrix und sind daher geeignet, eine Besetzungsstruktur weitgehend zu erhalten.
Die Householder-Transformationen dagegen sind dazu gänzlich ungeeignet. Sie zerstören bereits bei einem Schritt jedes Muster der Ausgangsmatrix.

Im Allgemeinen am schonendsten geht die Gauß-Elimination mit der Besetzungsstruktur von Matrizen um. Typischerweise wird dabei abwechselnd Spaltenpivotsuche mit evt. Zeilentausch und Zeilenpivotsuche mit evt. Spaltentausch ausgeführt, je nachdem, welche Strategie die meisten Nullelemente schont.

Ansatz 2 :

Iterative Verfahren zur Approximation der Lösung x.
Sinnvoll, da man in der Regel an der Lösung x nur bis auf eine vorgegebene Genauigkeit e interessiert ist, die von der Genauigkeit der Eingabedaten abhängt.

Ziel ist die Konstruktion einer Iterationsvorschrift $x_{k+1} = \Phi(x_0,...x_k)$, so dass

a) die Folge $\{x_k\}$ der Iterierten möglichst schnell gegen die Lösung x konvergiert,

b) x_{k+1} mit möglichst geringem Aufwand aus $x_0,..., x_k$ berechnet werden kann.

Bei der zweiten Forderung verlangt man meist, dass die Auswertung von Φ nicht wesentlich mehr Aufwand erfordert als eine einfache Matrix-Vektor-Multiplikation.
Desweiteren beträgt der Aufwand für dünnbesetzte Matrizen $O(n)$ und nicht $O(n^2)$ (wie bei vollbesetzten Matrizen), da häufig die Anzahl der von Null verschiedenen Elemente in einer Zeile unabhängig von der Dimension n des Problems ist.

4.1 Klassische Iterationsverfahren

Den meisten klassischen Iterationsverfahren liegt die Idee der Fixpunktiteration zugrunde.

Einschub Fixpunktiteration

Äquivalente Umformung der Gleichung $f(x) = 0$ in eine Fixpunktgleichung

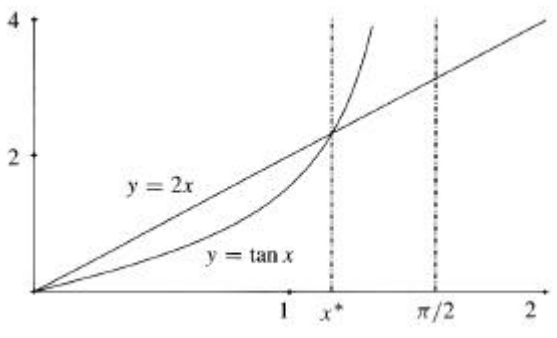
$$f(x) = x$$

und mit Hilfe der Iterationsvorschrift $x_{k+1} = f(x_k)$ mit $k = 0,1,...$

für einen gegebenen Startwert x_0 eine Folge $\{x_0, x_1,...\}$ konstruieren, welche gegen einen Fixpunkt x^* mit $f(x^*) = x^*$ konvergiert, der Lösung der Gleichung ist, d.h. $f(x^*) = 0$.

Beispiel Fixpunktiteration :

$$f(x) := 2x - \tan x = 0$$



Anhand der Abbildung $\left(x^* \approx 1.2\right)$ wählt man den Startwert $x_0 = 1.2$.
Umformen der Gleichung in eine Fixpunktgleichung :

$$x = \frac{1}{2}\tan x =: f_1(x_k) \qquad \text{oder} \qquad x = \arctan(2x) =: f_2(x_k)$$

Es ergeben sich die Zahlenwerte

k	$x_{k+1} = \frac{1}{2}\tan x_k$	$x_{k+1} = \arctan(2x_k)$
0	1.2	1.2
1	1.2860	1.1760
2	$1.70... > \frac{p}{2}$	1.1687
3		1.1665
4		1.1658
5		1.1656
6		1.1655
7		1.1655

Die erste Folge divergiert ($\tan x$ hat einen Pol bei $\frac{p}{2}$ und $x_2 > \frac{p}{2}$) , während die zweite konvergiert.
D.h. nicht jede naiv konstruierte Fixpunktiteration konvergiert.

Einschub Ende

Für ein Fixpunktverfahren $x_{k+1} = f(x_k)$ zur Lösung eines LGS $Ax = b$ muss eine Iterationsfunktion f derart konstruiert werden, dass sie genau einen Fixpunkt x^* besitzt und dieser gerade die exakte Lösung $x^* = x$ von $Ax = b$ ist.

$\Rightarrow$ Umformen der Gleichung $Ax = b$ in eine Fixpunktgleichung :

$$Ax = b \quad \Leftrightarrow \quad Q^{-1}(b - Ax) = 0$$
$$\Leftrightarrow \quad f(x) := \underbrace{(I - Q^{-1}A)}_{=:G} \cdot x + \underbrace{Q^{-1}b}_{=:c} = x \quad ,$$

wobei $Q \in GL(n)$ eine beliebige reguläre Matrix ist.

Natürlich muss dafür Sorge getragen werden, dass das Fixpunktverfahren $x_{k+1} = f(x_k) = Gx_k + c$ konvergiert

Def. : konsistentes Iterationsverfahren :

Ein Iterationsverfahren heißt konsistent, falls $x = A^{-1}b$ die Gleichung $x = Gx + c$ erfüllt.
Das GSV, das ESV und das SOR-Verfahren sind konsistent

Satz 4.1 (Notwendiges und hinreichendes Konvergenzkriterium)

Ein konsistentes Fixpunktverfahren / Iterationsverfahren $x_{k+1} = f(x_k) = Gx_k + c$ mit $G \in Mat_n(R)$ konvergiert genau dann für jeden Startwert $x_0 \in R^n$, wenn

$$r(G) < 1 \quad ,$$

wobei $r(G) = \max_j \left| l_j(G) \right|$ der **Spektralradius** der Iterationsmatrix G ist.

Beweis: Buch S.263

Da $r(G) \le \|G\|$ für jede zugeordnete Matrixnorm, ist $\|G\| < 1$ hinreichend für $r(G) < 1$.

In diesem Fall kann man die Fehler $x_k - x = G^k (x_0 - x)$ durch

$$\|x_k - x\| \le \|G\|^k \cdot \|x_0 - x\|$$

abschätzen.

Neben der Konvergenz wurde verlangt, dass sich $f(x) = Gx + c$ leicht berechnen lässt. Dazu muss die Matrix Q einfach zu invertieren sein.

Die am besten zu invertierende Matrix ist zweifellos $Q = I$.

$\Rightarrow$

Richardson-Verfahren

$$x_{k+1} = x_k - Ax_k + b \qquad (G = I - A)$$

Geht man von einer spd-Matrix A aus, so ergibt sich für den Spektralradius von G

$$r(G) = r(I - A) = \max\left(|1 - l_{\max}(A)|, |1 - l_{\min}(A)|\right).$$

Eine notwendige Bedingung für die Konvergenz der Richardson-Iteration ist somit

$$l_{\max}(A) < 2 .$$

Für sich genommen ist diese Iteration demnach nur selten verwendbar.

Die nächst komplizierteren Matrizen sind die Diagonalmatrizen, so dass als zweite Möglichkeit für Q die Diagonale D von

$$A = L + D + R$$

in Frage kommt, wobei $D = (a_{11}, ..., a_{nn})$ und

$$L := \begin{bmatrix} 0 & \cdots & \cdots & 0 \\ a_{21} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{n1} & \cdots & a_{n,n-1} & 0 \end{bmatrix} \quad , \quad R := \begin{bmatrix} 0 & a_{12} & \cdots & a_{11} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & a_{n-1,n} \\ 0 & \cdots & \cdots & 0 \end{bmatrix}$$

$\Rightarrow$

Jacobi-Verfahren (Gesamtschrittverfahren GSV)

$$x_{k+1} = (I - D^{-1}A) \cdot x_k + D^{-1}b \quad = -D^{-1}(L + R) \cdot x_k + D^{-1}b$$

Eine hinreichende Bedingung für seine Konvergenz ist die strikte Diagonaldominanz von A.

Algorithmus – GSV :

1.) Wähle Anfangsnäherung $x^{(0)} \in R^n$ und setze $k := 0$

2.) Iterationsschritt $x^{(k)} \mapsto x^{(k+1)}$

$$x_i^{(k+1)} := \frac{b_i - \sum_{j=1, i \ne j}^{n} a_{ij} x_j^{(k)}}{a_{ii}} \qquad i = 1, ..., n$$

$$k := k + 1$$

3.) if „Konvergenz“ then stop else goto 2.)

Abbruchkriterium : $\left\| x^{(k)} - A^{-1}b \right\| \leq TOL$ (Genauigkeitsschranke)

Problem: $A^{-1}b$ nicht bekannt

Idee : Schätze $\left\| x^{(k)} - A^{-1}b \right\|$ auf Grundlage der Konvergenzanalyse des Iterationsverfahrens

Satz 4.2 (Hinreichende Konvergenz des GSV)

Die Jacobi-Iteration $x_{k+1} = -D^{-1}(L+R)\cdot x_k + D^{-1}b$ konvergiert für jeden Startwert x_0 gegen die Lösung $x = A^{-1}b$, falls A strikt diagonaldominant ist, d.h.

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

Beweis :

Die Aussage folgt aus Satz 4.1 , da

$$r(D^{-1}(L+R)) \leq \left\| D^{-1}(L+R) \right\|_\infty = \max_i \sum_{i \neq j} \left| \frac{a_{ij}}{a_{ii}} \right| .$$ **Q.e.d.**

Nach den diagonalen folgen die gestaffelten Gleichungssysteme / Matrizen als nächstkomplexere.
Setzt man für Q die untere Dreieckshälfte $Q \coloneqq D + L$ an, so erhält man das

Gauß-Seidel-Verfahren (Einzelschrittverfahren ESV)

$$\begin{aligned} x_{k+1} &= (I - (D+L)^{-1}A)\cdot x_k + (D+L)^{-1}b \\ &= -(D+L)^{-1}R \cdot x_k + (D+L)^{-1}b \ . \end{aligned}$$

Satz 4.3 (Satz von Ostrowski und Reich)

Für symmetrische positiv definite Matrizen A gilt :

$r(G(w)) < 1 \quad \forall w \in]0,2[$, wobei $G(w)$ die Iterationsmatrix des SOR-Verfahrens ist.

Insbesondere konvergiert das ESV / Gauß-Seidel-Verfahren für jede spd-Matrix.

Algorithmus – ESV :

1.) Wähle Anfangsnäherung $x^{(0)} \in R^n$ und setze $k \coloneqq 0$

2.) Iterationsschritt $x^{(k)} \mapsto x^{(k+1)}$

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

$$k \coloneqq k+1$$

3.) if „Konvergenz“ then stop else goto 2.)

Zusammenfassung

Rechenaufwand : Beim GSV wie auch beim ESV pro Iterationsschritt $\leq n \cdot p$ Rechenoperationen, falls maximal p Nichtnullelemente pro Zeile von A

Speicherbedarf : GSV : 2 Vektoren der Länge n (x_alt , x_neu)

ESV : 1 Vektor der Länge n ($x_i^{(k+1)}$ wird auf den Platz von $x_i^{(k)}$ abgespeichert)

Achtung !!! : Durchführbarkeit nur für $a_{ii} \neq 0$, $i = 1,...,n$

Konvergenz : Konvergenz für beliebige Startvektoren $x^{(0)} \in R^n$, falls $\left\| G^{k+1} \right\| \to 0$ für eine verträgliche Matrixnorm

$$\left(\quad (x^{(k+1)} - x) = G(x^{(k)} - x) = G \cdot G(x^{(k-1)} - x) = ... = G^{k+1}(x^{(0)} - x) \quad \right)$$

hinreichende Bedingung : $\|G\| = q < 1 \quad \Rightarrow \left\| G^{k+1} \right\| \leq \|G\|^{k+1} = q^{k+1} \to 0$

notwendige Bedingung : $r(G) < 1$ für den Spektralradius.

Satz 4.4 (Zusammenhang GSV - ESV) – Satz von Stein und Rosenberg

Sei G_{GSV} die Iterationsmatrix des GSV und G_{ESV} die des ESV und alle Elemente von G_{GSV} größer Null.
Dann gilt :

$$0 < r(G_{GSV}) < 1 \quad \Rightarrow \quad 0 < r(G_{ESV}) < r(G_{GSV}) < 1$$
$$r(G_{GSV}) > 1 \quad \Rightarrow \quad 1 < r(G_{GSV}) < r(G_{ESV})$$

oder $r(G_{GSV}) = r(G_{ESV}) = 0$ oder $r(G_{GSV}) = r(G_{ESV}) = 1.$

D.h. wenn das GSV konvergiert, so konvergiert auch das ESV , und wenn das GSV divergiert, so auch das ESV. Desweiteren konvergiert das ESV schneller als das GSV.

Motivation : Konvergenzgeschwindigkeit verbessern

$\Rightarrow$

Relaxationsverfahren – SOR-Verfahren (successive over relaxation)

Das SOR-Verfahren unterscheidet sich nur wenig vom ESV. Zur Berechnung der jeweils folgenden Iterierten benutzt man die Idee des ESV, bildet aber anschließend noch einen gewichteten Mittelwert aus der alten und der neuen Komponente mit einem Parameter w , den man ziemlich frei wählen kann. Für $w = 1$ erhält man das alte ESV zurück.

$$\begin{aligned} x_{k+1} &= w \cdot (Gx_k + c) + (1-w) \cdot x_k \\ &= G_w x_k + w \cdot c \qquad \text{mit } G_w := w \cdot G + (1-w) \cdot I \ , \end{aligned}$$

wobei $w \in [0,1]$ ein sogenannter **Dämpfungsparameter** ist.

Auf diese Weise gewinnt man aus einer Fixpunktiteration $x_{k+1} = Gx_k + c$ eine ganze Schar von relaxierten Fixpunktiterationen mit der $w-$ abhängigen Iterationsfunktion

$$f_w(x) = w \cdot f(x) + (1-w) \cdot x = G_w x + w \cdot c \ .$$

Der Dämpfungsparameter w ist nun so zu wählen, dass $r(G_w)$ möglichst klein wird.

Tatsächlich ist es für die sogenannten **symmetrisierbaren Iterationsverfahren** sogar möglich, durch geeignete Wahl von w Konvergenz zu erzwingen, obwohl die Ausgangsiteration im Allgemeinen nicht konvergiert.

Def. symmetrisierbares Fixpunktverfahren :

Ein Fixpunktverfahren $x_{k+1} = Gx_k + c$ heißt symmetrisierbar, falls $I - G$ für jede spd-Matrix A ähnlich zu einer spd-Matrix ist, d.h. falls es eine reguläre Matrix $W \in GL(n)$ gibt, so dass

$$W \cdot (I - G) \cdot W^{-1}$$

eine spd-Matrix ist.

Relaxationsverfahren – Prof. Arnold

Idee : Berechne Näherung $\tilde{x}^{(k+1)}$ mit Iterationsverfahren und wähle

$$x^{(k+1)} := w \cdot \tilde{x}^{(k+1)} + (1-w) \cdot x^{(k)}$$

mit einem Parameter $w \in R$ so, dass $\left\| x^{(k+1)} - A^{-1}b \right\|$ „möglichst klein".

Wahl des Relaxationsparameters :

$w \in]0,1[$ ... under relaxation

$w > 1$ ... over relaxation

Hier Beschränkung auf Überrelaxation $(w > 1)$:

JOR (Relaxation des GSV / Jacobi-Verfahrens)

$$x_i^{(k+1)} := (1-w) \cdot x_i^{(k)} + w \frac{b_i - \sum_{j=1, i \neq j}^{n} a_{ij} x_j^{(k)}}{a_{ii}} = x_i^{(k)} + w \frac{b_i - \sum_{j=1}^{n} a_{ij} x_j^{(k)}}{a_{ii}} \qquad i = 1,...,n$$

SOR (Relaxation des ESV / Gauß-Seidel-Verfahrens)

$$x_i^{(k+1)} := x_i^{(k)} + w \frac{b_i - \sum_{j=1}^{n} a_{ij} x_j^{(k+1)} - \sum_{j=i}^{n} a_{ij} x_j^{(k)}}{a_{ii}} \qquad i = 1,...,n$$

Satz 4.5 (Satz von Kahan)

Für die Iterationsmatrix G_w des SOR-Verfahrens gilt :

$$r(G_w) \geq |w - 1|$$

Beweis : siehe Guido-Skript S.40

Def. zerlegbare Matrix (reduzibel) :

Eine Matrix $A \in R^{n \times n}$ heißt reduzibel / zerlegbar, wenn es eine Permutationsmatrix P gibt, so dass

$$P^T AP = \begin{pmatrix} C & D \\ 0 & F \end{pmatrix} \quad ,$$

wobei die Blockmatrizen C und F quadratisch sind.

Falls kein Nullblock erzeugt werden kann, heißt die Matrix A unzerlegbar (irreduzibel) .

Ordnet man A einem gerichteten Graphen $G(A)$ so zu, dass A n Knoten $K_1,...,K_n$ hat und das gilt :

$G(A)$ enthält die Kante $K_i \to K_j \quad \Leftrightarrow \quad a_{ij} \neq 0$,

dann folgt :

A ist genau dann unzerlegbar, wenn $G(A)$ zusammenhängend ist, d.h. wenn es zu beliebigen $i, j \in \{1,...,n\}$ einen gerichteten Weg von K_i nach K_j gibt.

Beispiel zerlegbare Matrix :

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

$G(A)$ ist nicht zusammenhängend, da kein gerichteter Weg von K_1 nach K_3 existiert.

A ist zerlegbar, da $P^T AP = \begin{pmatrix} 1 & 0 & 3 \\ 0 & 1 & -1 \\ 0 & 2 & 1 \end{pmatrix}$ für $P = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$

Def. konsistent geordnete Matrix :

Eine Matrix heißt konsistent geordnet, wenn die Eigenwerte von $-D^{-1}(a \cdot L + \frac{1}{a} R)$ für beliebiges $a \neq 0$ von a unabhängig sind.

Anriss Tschebyscheff-Beschleunigung :

Bei den bisherigen Fixpunktverfahren wird zur Berechnung von x_{k+1} nur die Information des letzten Iterationsschrittes herangezogen und die bereits berechneten Werte $x_0, \ldots, x_{k-1}$ nicht berücksichtigt.

Hier nun Verbesserung eines gegebenen Fixpunktverfahrens durch Konstruktion einer Linearkombination

$$y_k = \sum_{j=0}^{k} v_{kj} x_j$$

aus sämtlichen Werten $x_0, \ldots, x_k$.

Bei geeigneter Wahl der Koeffizienten v_{kj} soll die Ersatzfolge $\{y_0, y_1, \ldots\}$ schneller konvergieren als die Ausgangsfolge $\{x_0, x_1, \ldots\}$.

Bestimmung der Koeffizienten :

Falls $x_0 = \ldots = x_k = x$ bereits die Lösung ist, so muss auch $y_k = x$ gelten, woraus folgt, dass

$$\sum_{j=0}^{k} v_{kj} = 1 \ .$$

....weiter behandelt im Buch S.270-274 .....

4.2 Verfahren der konjugierten Gradienten (cg-Verfahren)

Motivation : Für LGS $Ax = b$ mit symmetrisch positiv definiter Koeffizientenmatrix A ein schnelles Verfahren finden.

Jede spd-Matrix A definiert in natürlicher Weise ein Skalarprodukt

$(x, y) := \langle x, Ay \rangle$ mit der zugeordneten Norm $\|y\|_A = \sqrt{(y, y)}$, der sogenannten Energienorm.

Nutzung eines problemangepassten Orthogonalitätsbegriffs :

Def. konjugierte Vektoren :

Sei A eine spd-Matrix. Dann heißen zwei Vektoren x und y konjugiert, wenn gilt :

$$x^T \cdot A \cdot y = 0$$

Ist A die Einheitsmatrix, so bleibt das normale Skalarprodukt.
Die zusätzlich zwischengeschobene Matrix A produziert ein etwas allgemeineres Skalarprodukt.
Dadurch, dass die Matrix A symmetrisch positiv definit ist, bleiben die Eigenschaften des Skalarproduktes erhalten.

Verfahren :

Der Grundgedanke der Methode der konjugierten Gradienten besteht darin, statt das vorgegebene Gleichungssystem

$$Ax = b$$

zu lösen, das folgende Funktional zu minimieren :

$$F(x) = \frac{1}{2} x^T Ax - b^T x \qquad \left(= \frac{1}{2} \langle x, Ax \rangle - \langle x, b \rangle \right)$$

Der Gradient dieses Funktionals lautet :

$$grad\ F(x) = Ax - b$$

Wurde für das Funktional ein Minimum bei x_0 errechnet, so verschwindet der Gradient an dieser Stelle :

$$grad\ F(x_0) = Ax_0 - b = 0 \quad ,$$

und damit ist x_0 die gesuchte Lösung des Gleichungssystems.

Beachte : Der Residuenvektor $r := Ax - b$ ist gerade gleich dem Gradienten des Funktionals F .

Man startet mit einem beliebigen Vektor $x^{(0)}$.
Um möglichst schnell zum Minimum zu gelangen, legt man die Relaxationsrichtung $p^{(0)}$ in Richtung des negativen Gradienten, den man leicht als Residuenvektor berechnen kann. Denn dort ist der stärkste Abstieg zu erwarten.

$$p^{(0)} = -r^{(0)} = -grad\ F(x^{(0)}) = -(Ax^{(0)} - b)$$

Für den damit zu erzielenden ersten Näherungsvektor macht man den Ansatz

$$x^{(1)} = x^{(0)} + s_0 \cdot p^{(0)}$$

Die Bedingung, dass das Funktional F minimal werden möge, führt zu

$$s_0 = \frac{(r^{(0)})^T \cdot r^{(0)}}{(p^{(0)})^T \cdot A \cdot p^{(0)}}$$

Im allgemeinen Schritt wählt man als Relaxationsrichtung eine Linearkombination des aktuellen Residuenvektors, also des jeweiligen Gradienten, und der vorhergehenden Relaxationsrichtung :

$$p^{(k)} := -r^{(k)} + b_k \cdot p^{(k-1)}$$

Hier bestimmt sich b_k aus der Forderung, dass $p^{(k)}$ und $p^{(k-1)}$ konjugiert zueinander sein mögen :

$$b_k := \frac{(r^{(k)})^T \cdot r^{(k)}}{(r^{(k-1)})^T \cdot r^{(k-1)}}$$

Anschließend bestimmt man

$$s_k := \frac{(r^{(k)})^T \cdot r^{(k)}}{(p^{(k)})^T \cdot A \cdot p^{(k)}}$$

und berechnet den neuen Näherungsvektor für die Lösung

$$x^{(k+1)} := x^{(k)} + s_k \cdot p^{(k)}$$

Algorithmus cg-Verfahren :

$x^{(0)}$ beliebig

$r^{(0)} := Ax^{(0)} - b$

$p^{(0)} := -r^{(0)}$

$k = 0$

if **accurate** then stop else (accurate = „$r^{(k)}$ hinreichend klein“)

begin

$$s_k := \frac{(r^{(k)})^T \cdot r^{(k)}}{(p^{(k)})^T \cdot A \cdot p^{(k)}} \qquad \left(= \frac{\langle r^{(k)}, r^{(k)} \rangle}{\langle p^{(k)}, A \cdot p^{(k)} \rangle} \right)$$

$$x^{(k+1)} := x^{(k)} + s_k \cdot p^{(k)}$$

$$k := k + 1$$

$$r^{(k)} := Ax^{(k)} - b$$

$$b_k := \frac{(r^{(k)})^T \cdot r^{(k)}}{(r^{(k-1)})^T \cdot r^{(k-1)}} \qquad \left(= \frac{\langle r^{(k)}, r^{(k)} \rangle}{\langle r^{(k-1)}, r^{(k-1)} \rangle} \right)$$

$$p^{(k)} := -r^{(k)} + b_k \cdot p^{(k-1)}$$

end

Algorithmus cg-Verfahren (Deuflhard S.277) :

$p_1 := r_0 := b - Ax_0$;

for $k := 1$ to $k_{\max}$ do

$$s_k := \frac{\langle r_{k-1}, r_{k-1} \rangle}{(p_k, p_k)} = \frac{\langle r_{k-1}, Br_{k-1} \rangle}{\langle p_k, Ap_k \rangle} \ ;$$

$x_k := x_{k-1} + s_k \cdot p_k$;

if accurate then exit ;

$r_k := r_{k-1} - s_k \cdot Ap_k$;

$$b_{k+1} := \frac{\langle r_k, r_k \rangle}{\langle r_{k-1}, r_{k-1} \rangle} \ ;$$

$p_{k+1} := r_k + b_{k+1} \cdot p_k$;

endfor

Bemerkungen :

- Pro Iterationsschritt wird tatsächlich im wesentlichen nur eine Matrix-Vektor-Multiplikation, nämlich $A \cdot p^{(k)}$ ausgeführt, womit das Verfahren den Forderungen bezüglich des Aufwands voll entspricht.
  (Zusätzlich werden noch Skalarprodukte und Additionen skalarer Vielfacher von Vektoren berechnet.)

- Für das Abbruchkriterium „accurate“ wäre $\left\| x - x^{(k)} \right\|$ hinreichend klein, was jedoch so nicht ausführbar ist.

  Deswegen lautet es in der Praxis „$\left\| r^{(k)} \right\|_2 = \left\| x - x^{(k)} \right\|_{A\,2}$ hinreichend klein“.

  Wie sich in Kapitel 2.3.2 - „Beurteilung von Näherungslösungen“ zeigte, ist die Residuennorm jedoch kein geeignetes Maß für die Konvergenz : Gerade bei schlecht konditionierten Systemen, d.h. $k(A) >> 1$, können sich die Iterierten drastisch verbessern, obwohl die Normen der Residuen wachsen. (siehe weiter unten : „Vorkonditionierung“)

- Die Relaxationsrichtungen bestehen aus paarweise konjugierten Richtungen, demzufolge sind die Residuenvektoren paarweise orthogonal. In einem n-dimensionalen Raum gibt es aber nur n orthogonale Vektoren. Die Methode endet also theoretisch nach n Schritten. In der Praxis läuft aber durch das Auftreten von Rundungsfehlern doch ein iterativer Prozess ab.

- Der aufwendigste Punkt ist die zweimalige Multiplikation von A mit $p^{(k)}$ bzw. $x^{(k)}$.
  Die zweite Multiplikation kann jedoch auf die erste zurückgeführt werden, denn es gilt :

$$\begin{aligned} r^{(k+1)} &= A \cdot x^{(k+1)} - b \\ &= A \cdot \left(x^{(k)} - s_k \cdot p^{(k)} \right) - b \\ &= r^{(k)} - s_k \cdot A \cdot p^{(k)} \end{aligned}$$

Betrachtungen zur Konvergenzgeschwindigkeit :

Satz 4.6

Der Approximationsfehler $x - x^{(k)}$ des cg-Verfahrens lässt sich in der Energienorm $\|y\|_A = \sqrt{\langle y, Ay \rangle}$ abschätzen durch

$$\left\| x - x^{(k)} \right\|_A \le 2 \left(\frac{\sqrt{k_2(A)} - 1}{\sqrt{k_2(A)} + 1} \right)^k \left\| x - x^{(0)} \right\|_A \quad ,$$

wobei $k_2(A)$ die Kondition von A bezüglich der euklidschen Norm ist.

Beweis : Buch S.278/279

Kor. 4.7

Um den Fehler in der Energienorm um einen Faktor e zu reduzieren, d.h.

$$\left\| x - x^{(k)} \right\|_A \le e \cdot \left\| x - x^{(0)} \right\|_A \quad ,$$

benötigt man höchstens k cg-Iterationen, wobei k die kleinste ganze Zahl ist mit

$$k \ge \frac{1}{2} \sqrt{k_2(A)} \cdot \ln\left(\frac{2}{e} \right)$$

Beweis : Buch S.279/280

4.3 Vorkonditionierung

Die Abschätzungen der Konvergenzgeschwindigkeit sowohl für die Tschebyscheff-Beschleunigung als auch für das cg-Verfahren hängen monoton von der Kondition $k_2(A)$ bezüglich der euklidschen Norm ab.

Ziel : Transformation des Problems $Ax = b$ derart, so dass die entstehende Matrix von möglichst kleiner Kondition ist.

Idee : Anstelle von $Ax = b$ mit einer spd-Matrix $A \in Mat_n(R)$ löst man das für jede invertierbare Matrix $B \in GL(n)$ äquivalente Problem

$$\overline{A}\overline{x} = b \qquad \text{mit} \quad \overline{A} := AB \quad \text{und} \quad \overline{x} := B^{-1}x \ .$$

Dabei ist darauf zu achten, dass die Symmetrie des Problems nicht zerstört wird, damit die Iterationsverfahren anwendbar bleiben. Ist B ebenfalls symmetrisch positiv definit, so ist die Matrix $\overline{A} = AB$ zwar nicht mehr bezüglich des euklidschen Skalarproduktes $\langle \cdot, \cdot \rangle$ **selbstadjungiert**, wohl aber bezüglich des von B induzierten Produktes

$$(\cdot,\cdot)_B := \langle \cdot, B\cdot \rangle \quad ,$$

da $(x, ABy)_B = \langle x, BABy \rangle = \langle ABx, By \rangle = (ABx, y)_B$.

Daher ist das cg-Verfahren wieder anwendbar, wenn man das Skalarprodukt geeignet überträgt :

$(\cdot,\cdot)_B$ übernimmt die Rolle des euklidschen Skalarproduktes $\langle \cdot, \cdot \rangle$ und das zugehörige „Energieprodukt“

$$(\cdot,\cdot)_{AB} = (AB\cdot,\cdot)_B = \langle AB\cdot, B\cdot \rangle$$

von $\overline{A} = AB$ die Rolle von $(\cdot,\cdot)$.

Daraus ergibt sich unmittelbar die folgende Iteration $\overline{x}_0, \overline{x}_1, \ldots$ zur Lösung von $\overline{A}\overline{x} = b$:

$p_1 := r_0 := b - AB\overline{x}_0$;

for $k := 1$ to $k_{\max}$ do

$$s_k := \frac{(r_{k-1}, r_{k-1})_B}{(p_k, p_k)_{AB}} = \frac{\langle r_{k-1}, Br_{k-1} \rangle}{\langle ABp_k, Bp_k \rangle} \ ;$$

$\overline{x}_k := \overline{x}_{k-1} + s_k \cdot p_k$;

if accurate then exit ;

$r_k := r_{k-1} - s_k \cdot ABp_k$;

$$b_{k+1} := \frac{(r_k, r_k)_B}{(r_{k-1}, r_{k-1})_B} = \frac{\langle r_k, Br_k \rangle}{\langle r_{k-1}, Br_{k-1} \rangle} \ ;$$

$p_{k+1} := r_k + b_{k+1} \cdot p_k$;

endfor

Da man an einer Iteration für die eigentliche Lösung $x = B\overline{x}$ interessiert ist, ersetzt man daher die Zeile für die $\overline{x}_k$ durch

$$x_k = x_{k-1} + s_k \cdot Bp_k \ .$$

Es fällt auf, dass nun die p_k nur in der letzten Zeile explizit auftauchen. Führt man aus diesem Grund die (A - orthogonalen) Vektoren $q_k := Bp_k$ ein, so ergibt sich folgende sparsame Version des vorkonditionierten cg-Verfahrens :

Algorithmus pcg-Verfahren (Deuflhard S.283) :
(preconditioned conjugate gradient method)

$r_0 := b - Ax_0$;

$q_1 := Br_0$;

for $k := 1$ to $k_{\max}$ do

$$s_k := \frac{(r_{k-1}, r_{k-1})_B}{(p_k, p_k)_{AB}} = \frac{\langle r_{k-1}, Br_{k-1} \rangle}{\langle Aq_k, q_k \rangle} ;$$

$x_k = x_{k-1} + s_k \cdot q_k$

if accurate then exit ;

$r_k := r_{k-1} - s_k \cdot Aq_k$;

$$b_{k+1} := \frac{(r_k, r_k)_B}{(r_{k-1}, r_{k-1})_B} = \frac{\langle r_k, Br_k \rangle}{\langle r_{k-1}, Br_{k-1} \rangle} ;$$

$q_{k+1} := Br_k + b_{k+1} \cdot q_k$;

endfor

Pro Iterationsschritt benötigt man jeweils eine Multiplikation mit der Matrix A (für Aq_k) bzw. mit B (für Br_k), also gegenüber dem ursprünglichen cg-Verfahren nur die Multiplikation mit B mehr.

(Konvergenzanalyse siehe Buch S.284 - S.286)

Beispiel

Eine sehr einfache, aber häufig schon wirkungsvolle Vorkonditionierung ist die Inverse $B := D^{-1}$ der Diagonale D von A, die sogenannte diagonale Vorkonditionierung.

Beispiel

Wendet man die Cholesky-Zerlegung $A = LL^T$ aus Kapitel 1 auf eine symmetrische dünnbesetzte Matrix an, so ist zu beobachten, dass außerhalb der Besetzungsstruktur von A meist nur „verhältnismäßig kleine“ Elemente l_{ij} entstehen. Dies führt zur Idee der unvollständigen Cholesky-Zerlegung (incomplete Cholesky decomposition , IC). Sie besteht darin, diese Elemente einfach wegzulassen.
Bezeichnet man mit

$P(A) := \{(i, j) : a_{ij} \neq 0\}$ die Indexmenge der nicht verschwindenden Elemente einer Matrix A ,

so konstruiert man also anstelle von L eine Matrix $\tilde{L}$ mit

$P(\tilde{L}) \subset P(A)$,

indem man wie bei der Cholesky-Zerlegung vorgeht und $\tilde{l}_{ij} := 0$ setzt für alle $(i, j) \notin P(A)$.

Dabei erwartet man, dass $A \approx \tilde{A} := \tilde{L}\tilde{L}^T$.

5. Lineare Eigenwertprobleme

Eigenwertproblem : Gegeben : quadratische Matrix A

Gesucht : alle Eigenwerte $l_i(A)$ mit zugehörigen Eigenvektoren x_i $\left(Ax_i = l_i x_i\right)$

oder aber die m betragsmäßig größten / kleinsten Eigenwerte

Häufigstes Eigenwertproblem : Ist A symmetrisch $\left(A = A^T\right)$, so gilt :

- $l_i \in R^n$
- Eigenvektoren zu verschiedenen Eigenwerten sind orthogonal
- Es gibt eine Orthonormalbasis des R^n aus Eigenvektoren von A.

Vorgriff :

Das Eigenwertproblem ist nur für normale Matrizen gut konditioniert, deren wichtigste Klasse die reel-symmetrischen Matrizen sind. Für allgemeine Matrizen ist dagegen das Problem der Singulärwertzerlegung gut konditioniert und praktisch enorm relevant.

Satz 5.1 (Satz von Gerschgorin)

Die Eigenwerte einer Matrix $A \in R^{n\times n}$ liegen in der Vereinigung $\bigcup_{i=1}^{n} K_i$ der Gerschgorin-Kreise

$$K_i := \left\{ z \in C : \ |z - a_{ii}| \le \sum_{j=1,\ j\neq i}^{n} |a_{ij}| \right\}$$

Beispiel : Gerschgorin-Kreise

$$A = \begin{pmatrix} 1 & 0.1 & -0.1 \\ 0 & 2 & 0.4 \\ -0.2 & 0 & 3 \end{pmatrix} \quad \Rightarrow \quad \begin{aligned} K_1 &= \{z : |z-1| \le 0.2\} \\ K_2 &= \{z : |z-2| \le 0.4\} \\ K_3 &= \{z : |z-3| \le 0.2\} \end{aligned}$$

5.1 Kondition des allgemeinen Eigenwertproblems

Lemma 5.2 (Stetigkeit des Eigenwertproblems)

Sei $l_0 \in C$ einfacher Eigenwert von $A \in Mat_n(C)$. Dann existiert eine stetig differenzierbare Abbildung

$$l : V \subset Mat_n(C) \to C \qquad , \qquad B \mapsto l(B)$$

von einer Umgebung V von A in $Mat_n(C)$, so dass $l(A) = l_0$

und $l(B)$ einfacher Eigenwert von B ist für alle $B \in V$.

Ist x_0 ein Eigenvektor von A zu l_0 und y_0 ein (adjungierter) Eigenvektor von $A^* := \overline{A}^T$ zum Eigenwert $\overline{l}_0$, d.h.

$$Ax_0 = l_0 x_0 \quad \text{und} \quad A^* y_0 = \overline{l}_0 y_0,$$

so gilt für die Ableitung von l an der Stelle A, dass

$$l'(A) \cdot C = \frac{\langle Cx_0, y_0 \rangle}{\langle x_0, y_0 \rangle} \qquad \text{für alle } C \in Mat_n(C).$$

Beweis : Buch S.134/135

Satz 5.3 (Kondition des Eigenwertproblems)

Die absolute Kondition der Bestimmung eines einfachen Eigenwertes l_0 einer Matrix $A \in Mat_n(C)$ bezüglich der 2-Norm ist

$$k_{abs} = \| l'(A) \| = \frac{\|x\| \cdot \|y\|}{|\langle x, y \rangle|} = \frac{1}{|\cos(\angle(x, y))|}$$

und die relative Kondition

$$k_{rel} = \frac{\|A\|}{|l_0|} \| l'(A) \| = \frac{\|A\|}{|l_0 \cdot \cos(\angle(x, y))|} \quad ,$$

wobei x ein Eigenvektor von A zum Eigenwert l_0 ist, d.h. $Ax = l_0 x$, und y ein adjungierter Eigenvektor, d.h. $A^* y = \overline{l}_0 y$.

Insbesondere ist das Eigenwertproblem für normale Matrizen gut konditioniert mit $k_{abs} = 1$.

5.2 Eigenwertberechnung

Motivation : Die Eigenwerte einer Matrix $A \in Mat_n(R)$ als Nullstellen des charakteristischen Polynoms $c_A(l) = \det(A - l \cdot I)$ zu berechnen ist allenfalls für $n = 2$ akzeptabel.

$\Rightarrow$ Entwicklung von Methoden, um die Eigenwerte und Eigenvektoren direkt zu bestimmen.

Einfachste Methode : direkte und inverse Vektoriteration

5.2.1 direkte Vektoriteration (power method)

Idee : Man iteriert die durch die Matrix $A \in Mat_n(R)$ gegebene Abbildung und definiert eine Folge $\{x_k\}_{k=0,1,...}$ für einen beliebigen Startvektor $x_0 \in R^n$ durch

$$x_{k+1} := Ax_k \qquad \text{für } k = 0,1,... \quad .$$

Ist ein einfacher Eigenwert l von A betragsmäßig echt größer als alle anderen Eigenwerte von A, so wird vermutet, dass sich l bei der Iteration gegenüber allen anderen Eigenwerten „durchsetzt“ und x_k gegen einen Eigenvektor von A zum Eigenwert l konvergiert.

Der Einfachheit halber Beschränkung auf symmetrische Matrizen :

Satz 5.4

Sei l_1 ein einfacher Eigenwert der symmetrischen Matrix $A \in Mat_n(R)$ und betragsmäßig echt größer als alle anderen Eigenwerte von A, d.h. $|l_1| > |l_2| \geq ... \geq |l_n|$.

Sei ferner $x_0 \in R^n$ ein Vektor, der nicht senkrecht auf dem Eigenraum von l_1 steht.

Dann konvergiert die Folge

$$y_k := \frac{x_k}{\|x_k\|} \qquad \text{mit} \qquad x_{k+1} := Ax_k$$

gegen einen normierten Eigenvektor von A zum Eigenwert l_1.

Beweis :

Sei $h_1,...,h_n$ eine Orthonormalbasis von Eigenvektoren von A mit $Ah_i = l_i h_i$.

Dann gilt $x_0 = \sum_{i=1}^{n} a_i h_i$ mit $a_1 = \langle x_0, h_1 \rangle \neq 0$.

Folglich ist

$$x_k = A^k x_0 = \sum_{i=1}^{n} a_i l_i^k h_i \quad = a_1 l_1^k \underbrace{\left(h_1 + \sum_{i=2}^{n} \frac{a_i}{a_1} \left(\frac{l_i}{l_1} \right)^k h_i \right)}_{=:z_k}.$$

Da $|l_i| < |l_1|$ für alle $i = 2,...,n$, gilt $\lim_{k\to\infty} z_k = h_1$ und daher

$$y_k = \frac{x_k}{\|x_k\|} = \frac{z_k}{\|z_k\|} \to \pm h_1 \qquad \text{für } k \to \infty. \qquad \textbf{Q.e.d.}$$

Nachteile der direkten Vektoriteration :

- Man erhält nur den Eigenvektor zum betragsgrößten Eigenwert von A.
- Die Konvergenzgeschwindigkeit hängt vom Quotienten $\left| \frac{l_2}{l_1} \right|$ ab.

  (D.h. liegen die Eigenwerte l_1 und l_2 betragsmäßig dicht zusammen, so konvergiert die direkte Vektoriteration nur sehr langsam.)

$\Rightarrow$

5.2.2 Inverse Iteration (inverse power method)

Angenommen, man hätte einen Schätzwert $\bar{l} \approx l_i$ eines beliebigen Eigenwertes l_i der Matrix A zur Verfügung, so dass

$$|\bar{l} - l_i| < |\bar{l} - l_j| \qquad \text{für alle } j \neq i. \qquad \textbf{(*)}$$

Dann ist $(\bar{l} - l_i)^{-1}$ der betragsgrößte Eigenwert der Matrix $(A - \bar{l} \cdot I)^{-1}$. Konsequenterweise konstruiert man deshalb die Vektoriteration für diese Matrix. Diese Idee liefert die Iterationsvorschrift

$$(A - \bar{l} \cdot I) \cdot x_{k+1} = x_k \qquad \text{für } k = 0,1,...$$

Man beachte, dass bei jedem Iterationsschritt ein lineares Gleichungssystem gelöst werden muss, jedoch nur für verschiedene rechte Seiten x_k. Die Matrix $A - \bar{l} \cdot I$ muss daher nur einmal zerlegt werden.

Nach Satz 5.4 konvergiert die Folge $y_k := \frac{x_k}{\|x_k\|}$ unter der Voraussetzung **(*)** für $k \to \infty$ gegen einen normierten Eigenvektor von A zum Eigenwert l_i, falls nicht gerade der Startvektor x_0 senkrecht auf dem Eigenvektor h_i zum Eigenwert l_i steht.

Der **Konvergenzfaktor** ist dabei

$$\max_{j \neq i} \left| \frac{l_i - \bar{l}}{l_j - \bar{l}} \right| < 1 \qquad .$$

Ist $\bar{l}$ eine besonders gute Schätzung von l_i, so gilt

$$\left| \frac{l_i - \bar{l}}{l_j - \bar{l}} \right| << 1 \qquad \text{für alle } j \neq i \qquad , \qquad \text{das Verfahren konvergiert in diesem Fall sehr schnell.}$$

Durch geeignete Wahl von $\bar{l}$ kann man also mit dieser Methode und einem nahezu beliebigen Startvektor x_0 einzelne Eigenwerte und Eigenvektoren herausgreifen.

Bemerkung :

Man beachte, dass die Matrix $A - \overline{l} \cdot I$ für „gut gewähltes“ $\overline{l} \approx l_i$ fast singulär ist.

Im vorliegenden Fall entstehen daraus jedoch keine numerischen Schwierigkeiten, da nur die Richtung des Eigenvektors gesucht ist, deren Berechnung gut konditioniert ist.

Beispiel Vektoriteration :

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

Eigenwerte : $l_1 = 1$, $l_2 = 2$

Geht man von einer Approximation $\overline{l} = 1 - e$ für l_1 mit $0 < e << 1$ aus, so ist die Matrix

$$A - \overline{l} \cdot I = \begin{pmatrix} -2+e & 3 \\ -2 & 3+e \end{pmatrix}$$ fast singulär und

$$\left(A - \overline{l} \cdot I\right)^{-1} = \frac{1}{e(e+1)} \begin{pmatrix} 3+e & -3 \\ 2 & -2+e \end{pmatrix}.$$

Da sich der Faktor $1/e(e+1)$ bei der Normierung herauskürzt, ist die Berechnung der Richtung einer Lösung x von $(A - \overline{l} \cdot I)x = b$ gut konditioniert.

Dies lässt sich auch an der relativen komponentenweisen Kondition

$$k_{rel} = \frac{\left\| \left| \left(A - \overline{l} \cdot I\right)^{-1} \right| \cdot |b| \right\|_\infty}{\|x\|_\infty}$$ bezüglich Störungen der rechten Seite ablesen.

Für $b := (1,0)^T$ ergibt sich z.B.

$$x = \left(A - \overline{l} \cdot I\right)^{-1} b = \left| \left(A - \overline{l} \cdot I\right)^{-1} \right| \cdot |b| = \frac{1}{e(e+1)} \begin{pmatrix} 3+e \\ 2 \end{pmatrix}$$ und daher (mit Kapitel 2 - LGS),

$$k_{rel} = k_C\left(\left(A - \overline{l} \cdot I\right)^{-1}, b\right) = 1 .$$

Tatsächlich wird in Programmen bei einer (echt) singulären Matrix $A - \overline{l} \cdot I$ ein Pivotelement $e = 0$ durch die relative Maschinengenauigkeit eps ersetzt und mit der nun fast singulären Matrix die inverse Vektoriteration durchgeführt.

5.2.3 QR-Algorithmus für symmetrische Eigenwertprobleme

Wie in Kapitel 5.1 gezeigt wurde, ist das Eigenwertproblem für symmetrische Matrizen gut konditioniert.

Motivation :
Effektive Methode entwickeln um sämtliche Eigenwerte einer reellen symmetrischen Matrix gleichzeitig zu berechnen.
Es ist bekannt, dass A nur reelle Eigenwerte $l_1, ..., l_n \in R$ besitzt und eine Orthonormalbasis $h_1, ..., h_n \in R^n$ aus Eigenvektoren $Ah_i = l_i h_i$ existiert, d.h.

$$Q^T A Q = \Lambda = diag(l_1, ..., l_n) \quad \text{mit } Q = [h_1, ..., h_n] \in O(n) . \qquad \textbf{(*)}$$

Die erste Idee wäre, Q direkt in endlich vielen Schritten zu bestimmen. Da die Eigenwerte die Wurzeln des charakteristischen Polynoms sind, hätte man damit auch ein endliches Verfahren zur Bestimmung der Nullstellen von Polynomen beliebigen Grades gefunden. Nach dem **Satz von Abel** jedoch existiert i.A. kein solches Verfahren. (basierend auf den Operationen +,-,*,/, $\sqrt{\ }$)

Die zweite Idee, die von (*) nahegelegt wird, ist, A durch eine Ähnlichkeitstransformation (Konjugation), z.B. mit orthogonalen Matrizen, der Diagonalgestalt näherzubringen, da die Eigenwerte invariant unter Ähnlichkeitstransformationen sind.
Versucht man, eine symmetrische Matrix A durch Konjugation mit Householder-Matrizen auf Diagonalgestalt zu bringen, so erweist sich dies schnell als unmöglich :

$$\begin{bmatrix} * & \dots & \dots & * \\ \vdots & & & \vdots \\ \vdots & & & \vdots \\ * & \dots & \dots & * \end{bmatrix} \xrightarrow{Q_1 \cdot} \begin{bmatrix} * & * & \dots & \dots & * \\ 0 & \vdots & & & \vdots \\ \vdots & \vdots & & & \vdots \\ 0 & * & \dots & \dots & * \end{bmatrix} \xrightarrow{\cdot Q_1^T} \begin{bmatrix} * & 0 & \dots & \dots & 0 \\ \vdots & * & & & * \\ \vdots & \vdots & & & \vdots \\ * & * & \dots & \dots & * \end{bmatrix}$$

Was die Multiplikation mit der Householder-Transformation von links geschafft hat, wird bei der Multiplikation von rechts wieder zerstört.
Anders sieht es aus, wenn man A nur auf **Tridiagonalgestalt** bringen will :

$$\begin{bmatrix} * & \dots & \dots & * \\ \vdots & & & \vdots \\ \vdots & & & \vdots \\ * & \dots & \dots & * \end{bmatrix} \xrightarrow{P_1 \cdot} \begin{bmatrix} * & * & \dots & \dots & * \\ * & \vdots & & & \vdots \\ 0 & \vdots & & & \vdots \\ \vdots & \vdots & & & \vdots \\ 0 & * & \dots & \dots & * \end{bmatrix} \xrightarrow{\cdot P_1^T} \begin{bmatrix} * & * & 0 & \dots & 0 \\ * & * & \dots & \dots & * \\ 0 & \vdots & & & \vdots \\ \vdots & \vdots & & & \vdots \\ 0 & * & \dots & \dots & * \end{bmatrix}$$

Lemma 5.5

Sei $A \in Mat_n(R)$ symmetrisch. Dann existiert eine orthogonale Matrix $P \in O(n)$, welche das Produkt von $n-2$ Householder-Reflexionen ist, so dass PAP^T Tridiagonalgestalt hat.

Beweis : Buch S.142

Somit wurde das Ausgangsproblem auf die Bestimmung der Eigenwerte einer symmetrischen Tridiagonalmatrix transformiert. Es existiert auch eine LR-Zerlegung, hier aber nur :

QR-Zerlegung : Diese existiert immer (keine Permutation nötig) und ist vor allen Dingen inhärent stabil.
Man definiert daher eine Folge $\{A_k\}_{k=1,2,\dots}$ von Matrizen durch

a) $A_1 = A$

b) $A_k = Q_k R_k$, QR-Zerlegung

c) $A_{k+1} = R_k Q_k$.

Lemma 5.6

Die Matrizen A_k haben folgende Eigenschaften :

1.) Die Matrizen A_k sind alle konjugiert zu A.

2.) Ist A symmetrisch, so auch alle A_k.

3.) Ist A symmetrisch und tridiagonal, so auch alle A_k.

Beweis : Buch S.142/143

Satz 5.7 (Konvergenzeigenschaften für den einfachen Fall, dass die Beträge der Eigenwerte paarweise verschieden sind)

Sei $A \in Mat_n(R)$ symmetrisch mit den Eigenwerten $l_1,...,l_n$, so dass $|l_1| > |l_2| > ... > |l_n| > 0$ und A_k, Q_k, R_k wie oben definiert. Dann gilt mit $A_k = \left(a_{ij}^{(k)}\right)$:

a) $\lim_{k\to\infty} Q_k = I$,

b) $\lim_{k\to\infty} R_k = \Lambda$,

c) $a_{ij}^{(k)} = O\left(\left|\frac{l_i}{l_j}\right|^k\right)$ für $i > j$.

Beweis : Buch S.144/145

Bemerkung :

Eine genauere Analyse zeigt, dass das Verfahren auch für mehrfache Eigenwerte $l_i = ... = l_j$ konvergiert.

Falls hingegen $l_i = -l_{i+1}$, so konvergiert das Verfahren nicht. Es bleiben 2×2 Blöcke stehen.

Liegen zwei Eigenwerte l_i, l_{i+1} betragsmäßig dicht beieinander, so konvergiert das Verfahren nur sehr langsam.
Dies kann mit Hilfe der sogenannten **Shift-Strategien** verbessert werden. Im Prinzip versucht man, die beiden Eigenwerte dichter an den Nullpunkt zu schieben und so den Quotienten $|l_{i+1} / l_i|$ zu verkleinern.

Dazu verwendet man für jeden Iterationsschritt k einen **Shift-Parameter** s_k und definiert die Folge $\{A_k\}$ durch :

a) $A_1 = A$

b) $A_k - s_k I = Q_k R_k$ QR-Zerlegung

c) $A_{k+1} = R_k Q_k + s_k I$

...detaillierter siehe Buch S. 145/146

Algorithmus QR-Algorithmus :

a) Reduziere das Problem auf Tridiagonalgestalt,
$A \to A_1 = PAP^T$, A_1 symmetrisch und tridiagonal , $P \in O(n)$.

b) Approximiere die Eigenwerte mit dem QR-Algorithmus mit Givens-Rotationen angewandt auf A_1 ,
$\Omega \cdot A_1 \cdot \Omega^T \approx \Lambda$, Ω Produkt aller Givens-Rotationen $\Omega_{ij}^{(k)}$.

c) Die Spalten von ΩP approximieren die Eigenvektoren von A :
$\Omega P \approx [h_1,...,h_n]$

Aufwand :

a) $\frac{4}{3}n^3$ Multiplikationen für die Transformation auf Tridiagonalgestalt,

b) $O(n^2)$ Multiplikationen für den QR-Algorithmus.

Für große n überwiegt daher der Aufwand für die Reduktion auf Tridiagonalgestalt.