Kapitel 1

Lineare Gleichungssysteme

1.1 Das Eliminationsverfahren

Sei K der Körper der komplexen oder der Körper der reellen Zahlen. A sei eine (n, n)-Matrix über K und b ein Vektor der Länge n. Wir betrachten das lineare Gleichungssystem

$$Ax = b \tag{1.1}$$

für den n-Vektor x. Aus den Anfangsvorlesungen ist bekannt, daß (1.1) genau dann für alle b eindeutig lösbar ist, wenn det $(A) \neq 0$. Es gibt auch eine explizite Formel für die Lösung x, nämlich die Cramer'sche Regel: Sei A_j die Matrix, die aus A durch Ersetzen der j-ten Spalte mit b entsteht. Dann gilt für die j-te Komponente x_j von x

$$x_j = \frac{\det(A_j)}{\det(A)} \quad , \qquad j = 1, \ldots, n\ . \tag{1.2}$$

Dies ist eine für den Mathematiker vollkommen befriedigende Lösung des Problems. Sie ist vollkommen einfach und explizit, und gilt immer dann, wenn das Problem sinnvoll ist, d.h. wenn det $(A) \neq 0$.

Für die Numerik ist (1.2) aber vollkommen nutzlos. Der Rechenaufwand zur Auswertung von (1.2) ist nämlich viel zu hoch. Die Berechnung von det (A) nach der Definition der Determinante als Summe von $n!$ Produkten mit je n Faktoren (wir werden später schnellere Methoden kennenlernen) erfordert $n!(n-1)$ Multiplikationen und $n! - 1$ Additionen/Subtraktionen.

Das gleiche gilt für det (A_j). Also benötigen wir zur Auswertung von (1.2)

$$(n+1)(n!(n-1) \ \text{Mult.} \ + \ (n!-1) \ \text{Add./Sub.}) \ + n \ \text{Divisionen} \qquad (1.3)$$

In Komplexitätsbetrachtungen dieser Art interessieren uns nur die Terme höchster Ordnung in n. Die anderen fassen wir durch das O-Symbol (O für Ordnung) zusammen. Wir schreiben $\mathrm{O}(g)$ für jeden Ausdruck, der sich in der Form

$$|\mathrm{O}(g)| \leq C|g| \,, \qquad g \to \infty$$

abschätzen läßt mit einer von g unabhängigen Konstanten C. Für (1.3) können wir dann schreiben

$$(n+2)! \ \text{Mult.} \ + \ \mathrm{O}((n+1)!) \ \text{ weitere Operationen} \,. \qquad (1.4)$$

Man interessiert sich dann nur für den ersten Term $(n+2)!$, nimmt also an, n sei so groß, daß dieser Term überwiegt. Für $n = 20$ ist $(n+2)! = 1.1 \ 10^{21}$, eine für die praktische Rechnung ganz absurde Zahl.

An dieser Stelle wollen wir gleich unsere Notation für Dezimalzahlen oder Gleitkommazahlen (*floating point*) erläutern. Mit 1.1 z.B. meinen wir eine reelle Zahl, welche nach Rundung auf 1 Stelle hinter dem Dezimalpunkt 1.1 ergibt. Die Schreibweise $x = 1.1$ meint also, daß $1.05 \leq x < 1.15$. Wir wollen diese Vereinbarung aber nicht allzu pedantisch verstehen, den Begriff der Rundung also nicht ganz präzisieren.

Wir werden nun das Eliminationsverfahren zur Lösung von (1.1) besprechen. Dieses wurde zwar in der Schule und in den Anfängervorlesungen schon behandelt. Es gibt aber spezielle numerische Aspekte, die uns zwingen, noch einmal ganz von vorne anzufangen. Wir schreiben dazu (1.1) ausführlich in der Form

$$\begin{array}{lcl} a_{1,1}x_1 + a_{1,2}x_2 & + \cdots + & a_{1,n}x_n = b_1 \,, \\ a_{2,1}x_1 + a_{2,2}x_2 & + \cdots + & a_{2,n}x_n = b_2 \,, \\ & \cdots & \\ a_{n,1}x_1 + a_{n,2}x_2 & + \cdots + & a_{n,n}x_n = b_n \,. \end{array} \qquad (1.5)$$

Sei det $(A) \neq 0$. Dann ist mindestens ein $a_{i,1} \neq 0$.

Eventuell nach einer Zeilenvertauschung können wir $a_{1,1} \neq 0$ annehmen. Wir eliminieren nun aus den Gleichungen 2 bis n die Variable x_1, indem wir zu den Zeilen von 2 bis n das $\ell_{i,1} = a_{i,1}/a_{11}$-fache der ersten Zeilen subtrahieren, $i = 2, \ldots, n$. Es entsteht dann das zu (1.5) äquivalente System

$$\begin{array}{rcrcll} a_{1,1}x_1 & + & a_{1,2}x_2 & + \cdots + & a_{1,n}x_n & = b_1 \\ & & (a_{2,2} - \ell_{2,1}a_{1,2})x_2 & + \cdots + & (a_{2,n} - \ell_{2,1}a_{1,n})x_n & = b_2 - \ell_{2,1}b_1 \\ & & & \cdots & & \\ & & (a_{n,2} - \ell_{n,1}a_{1,2})x_2 & + \cdots + & (a_{2,n} - \ell_{2,1}a_{1,n})x_n & = b_n - \ell_{2,1}b_1 \end{array} \tag{1.6}$$

Es enthält in den Zeilen $2, \ldots, n$ nur noch die Variablen $x_2, \ldots, x_n$. Diese Zeilen bilden also ein lineares System von $n-1$ Gleichungen in $n-1$ Unbekannten. Wir schreiben es als

$$\begin{array}{rcl} a^{(2)}_{2,2}x_2 & + \cdots + & a^{(2)}_{2,n}x_n = b^{(2)}_2 \\ & \cdots & \\ a^{(2)}_{n,2}x_2 & + \cdots + & a^{(2)}_{n,n}x_n = b^{(2)}_n \end{array} \tag{1.7}$$

$$\begin{array}{rcll} a^{(2)}_{i,k} & = & a_{i,k} - \ell_{i,1}a_{i,k} & , \quad i,k = 2, \ldots, n \quad , \\ b^{(2)}_i & = & b_i - \ell_{i,1}b_1 & , \quad i = 2, \ldots, n \quad , \\ \ell_{i,1} & = & a_{i,1}/a_{1,1} & , \quad i = 2, \ldots, n \quad . \end{array}$$

Ist (1.7) gelöst, so bekommt man eine Lösung von (1.6) (und damit von (1.5) oder (1.1)) durch

$$x_1 = (b_1 - a_{1,2}x_2 - \cdots - a_{1,n}x_n)/a_{11} \ .$$

Das lineare System (1.7) wird nun genauso behandelt wie (1.5). Man eliminiert x_2 aus den Gleichungen $3, \ldots, n$ - eventuell nach Zeilenvertauschung - und bekommt dann ein lineares System von $n-2$ Gleichungen für die $n-2$ Unbekannten $x_3, \ldots, n_n$. So geht man rekursiv vor, bis man bei einem System mit nur einer Unbekannten ankommt, das man direkt löst.

Wir wollen für das Eliminationsverfahren ein rekursives Programm schreiben:

elim (A, b, x, j, n)

/$\star$ Löst $\sum_{k=j}^{n} a_{i,k} x_k = b_i \quad , \quad i = j, \ldots, n$ $\star$/

{ Zeilenvertauschung () ;
for $i = j+1, \ldots, n$
{ $\ell = a_{i,j}/a_{j,j}$;
for $k = j+1, \ldots, n \quad a_{i,k} = a_{i,k} - \ell * a_{j,k}$;
$b_i = b_i - \ell * b_j$;
}
if $(j < n)$ elim $(A, b, x, j+1, n)$;
$x_j = b_j$;
for $k = j+1, \ldots, n \quad x_j = x_j - a_{j,k} * x_k$;
$x_j = x_j / a_{i,j}$;
}

Die Lösung des Systems (1.1) geschieht nun einfach durch den Aufruf

$$\text{elim}(A, b, x, 1, n) \ ;$$

Die in diesem Programm noch nicht spezifizierte Prozedur Zeilenvertauschung sucht unter den Elementen $a_{i,j}$, $i \geq j$ ein von Null verschiedenes, etwa $a_{i_0,j}$, und vertauscht dann die Zeilen j und i_0. Das Element $a_{i_0,j}$ heißt dann das Pivot, Zeile i_0 die Pivotzeile, und Spalte j die Pivotspalte. Also:

Zeilenvertauschung ();

{ Suche i_0;
for $k = j, \ldots, n$ swap $(a_{i_0,k}, a_{j,k})$;
swap (b_{i_0}, b_j);
}

In der Regel wird man mehrere i_0 zur Auswahl haben. Rein mathematisch gesehen ist es gleichgültig, welches wir nehmen. Für die Numerik ist es aber entscheidend, daß man aus den möglichen Indizes i denjenigen auswählt, für den $|a_{i,j}|$ maximal ist. Man bestimmt also i_0 so, daß $|a_{i_0,j}| \geq |a_{i,j}|, \quad i = j, \ldots, n$. Dieses i_0 ist im allgemeinen immer noch nicht eindeutig bestimmt, aber das spielt nun auch numerisch keine Rolle mehr. Wir sprechen von maximaler Spaltenpivotsuche.

Um zu verstehen, warum es gut ist, das Pivotelement möglichst groß zu wählen, müssen wir uns Gedanken über die Entstehung und Fortspflanzung von Rundungsfehlern machen.

Sei x eine reelle Zahl und $\tilde{x}$ eine Näherung für x. Dann nennen wir $|x - \tilde{x}|$ den absoluten und $|(x - \tilde{x})/x|$ den relativen Fehler von $\tilde{x}$, letzteres natürlich nur, wenn $x \neq 0$. Ist $\tilde{x}$ ein gerundeter Wert von x, so sprechen wir von den absoluten bzw. relativen Rundungsfehlern.

Aus einem kleinen Fehler am Anfang kann ein großer Fehler am Ende werden (so schon Aristoteles im 5. Buch der Methaphysik). Betrachten wir einmal folgende 4-stellige Rechnung

$$1.2547 - 1.2534 = 0.0013 \ .$$

Die beiden Operanden haben gemäß unserer Schreibweise je einen absoluten Fehler $\leq 5 \cdot 10^{-5}$. Daraus wird im Resultat ein absoluter Fehler von maximal 10^{-4}. Für die relativen Fehler bedeutet das ein gewaltiges Anwachsen. Aus $4 \cdot 10^{-5}$ in den Operanden wird $8 \cdot 10^{-2}$ im Resultat. Wir sehen, daß bei der Subtraktion nahezu gleichgroßer Zahlen der relative Fehler stark ansteigt. Wir sprechen von Auslöschung. Alle anderen Operationen, also Addition (von Zahlen gleichen Vorzeichens), Multiplikation und Division, führen zu keiner Verstärkung des relativen Fehlers. Auslöschung ist das Problem Nr. 1 der Numerik.

Glücklicherweise verursachen die meisten Auslöschungen keine Probleme. Bei der Analyse und dem Entwurf numerischer Algorithmen müssen Auslöschungen erkannt und - durch geeignete Gestaltung des Algorithmus - unschädlich gemacht werden. Dies wollen wir an Hand des Eliminationsverfahrens einmal vorführen.

Die einzige Rechenoperation, die beim Eliminationsverfahren ausgeführt wird, ist

$$a_{i,k} - \ell_{i,j} a_{j,k} \ , \quad \ell_{i,j} = a_{i,j}/a_{j,j} \ .$$

Wir wollen annehmen, daß alle Elemente von A die gleiche Größenordnung haben, etwa 1. Auslöschung bedeutet dann, daß ein sehr kleines $a_{i,k}$ auftritt. Dieses hat dann notwendigerweise großen relativen Fehler. Wir unterscheiden drei Fälle:

1. Das kleine Element $a_{i,k}$ steht nicht in Spalte j. Dann kann $a_{i,k} - \ell_{i,j} a_{j,k}$ trotz des großen relativen Fehlers von $a_{i,k}$ genau berechnet werden, da $a_{i,k}$ im Vergleich zu $\ell_{i,j} a_{j,k}$ sehr klein ist. Die Auslöschung ist harmlos.

2. Das Element $a_{i,j}$ in der j-ten Spalte unterhalb des Elementes $a_{j,j}$ fällt klein aus. Dann ist $\ell_{i,j}$ klein und hat großen relativen Fehler. $a_{i,k} - \ell_{i,j} a_{j,k}$ kann aber trotzdem genau berechnet werden, weil $\ell_{i,j} a_{j,k}$ klein gegenüber $a_{i,k}$ ist. Wieder ist die Auslöschung harmlos.

3. $a_{j,j}$ fällt klein aus. Jetzt wird $\ell_{i,j}$ groß und falsch. $a_{i,k} - \ell_{i,j} a_{j,k}$ kann nicht mehr genau berechnet werden. Die Auslöschung ist nicht harmlos.

Der Auslöschungseffekt ist also um so größer, je kleiner das Pivot $a_{j,j}$ ist. Im Prinzip könnte man jedes Element unterhalb und rechts von $a_{j,j}$ als Pivot nehmen. Dies würde Zeilen- und Spaltenvertauschungen erfordern. Die Praxis und auch die mathematische Theorie zeigen aber, daß es genügt, das betragsgrößte Element unterhalb von $a_{j,j}$ zu wählen. So kommt man zur maximalen Spaltenpivotsuche.

Spaltenpivotsuche ();

```
{   piv = |a_jj|;  i_0 = j;
    for i = j+1, ..., n
    if (|a_i,j| > piv)    {piv = |a_i,j|;  i_0 = i ; }
    if (piv  < ε)    {print ("Matrix singulär"); exit (1);}
    for k = j, ..., n    swap (a_j,k, a_i_0,k);
    swap (b_j, b_i_0);
}
```

Unser Programm für das Eliminationsverfahren enthält einen rekursiven Prozeduraufruf. Das ist zwar elegant, aber ineffizient. Auflösen der Rekursion führt zum endgültigen Programm.

```
elim (A, b, x, n)        /* Löst   Ax = b */
  for  j = 1, ..., n
    {  Spaltenpivotsuche () ;
       for   i = j + 1, ..., n
         {  ℓ = a_ij / a_jj  ;
            for  k = j + 1, ..., n   a_ik = a_ik − ℓ * a_jk  ;
            b_i = b_i − ℓ * b_j  ;
         }
    }
  for  j = n, ..., 1
    {  x_j = b_j;
       for  k = j + 1, ..., n     x_j = x_j − a_j,k * x_k;
       x_j = x_j / a_jj;
    }
```

Wir wollen die Anzahl der Rechenoperationen beim Eliminationsverfahren bestimmen. Da eine Multiplikation meist zusammen mit einer Add./Sub. auftritt, vereinbart man

1 flop (floating point operation) = 1 Mult./Div. + 1 Add./Sub. .

Sei K_j die Anzahl der flops für den j-ten Eliminationsschritt. Es ist

$$K_j = (n-j)^2 + \mathrm{O}(n-j) \; ,$$

also

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

Man vergleiche dies mit der Anzahl der Operationen (1.4) für die Cramer'sche Regel.

1.2 Die LR-Zerlegung

Wir wollen nun den Eliminationsprozess in einer anderen Form darstellen. Wir nennen die (n,n)-Matrix $L = (\ell_{i,k})$ linke Dreiecksmatrix, wenn $\ell_{i,k} = 0$ ist für $k > i$. Ebenso nennen wir die (n,n)-Matrix $R = (r_{i,k})$ rechte Dreiecksmatrix, wenn $r_{i,k} = 0$ ist für $i > k$. Die invertierbaren linken (und auch rechten) Dreiecksmatrizen bilden eine Gruppe bezüglich der Multiplikation.

Unter der LR-Zerlegung einer (n,n)-Matrix A versteht man die Berechnung von Dreiecksmatrizen L, R mit $A = LR$. Ist die LR-Zerlegung hergestellt, kann $Ax = b$ durch

$$Ly = b \quad , \quad Rx = y$$

ersetzt werden. Diese Systeme mit Dreiecksmatrizen können durch sogenanntes Vorwärts- bzw. Rückwärtseinsetzen gelöst werden:

$$\begin{array}{lcllcl}
y_1 & = & b_1/\ell_{1,1}\ , & x_n & = & y_n/r_{n,n}\ , \\
y_2 & = & (b_2 - \ell_{2,1}y_1)/\ell_{2,2}\ , & x_{n-1} & = & (y_{n-1} - r_{n-1,n}x_n)/r_{n,n}\ , \\
 & \vdots & & & \vdots & \\
y_n & = & (b_n - \ell_{n,1}y_1 - \cdots - \ell_{n,n-1}y_{n-1})\ell_{n,n}, & x_1 & = & (y_1 - r_{1,2}x_2 - \cdots - r_{1,n}x_n)/r_{1,1}\ .
\end{array}$$

Jeder dieser Prozesse erfordert $\frac{1}{2}n^2 +$ O (n) flops.

Zur Herstellung der LR-Zerlegung verwenden wir die Elementarmatrizen.

$$L_j = \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & -\ell_{j+1,j} & \ddots & \\ & & -\ell_{n,j} & & 1 \end{pmatrix} .$$

Sie weichen nur in der j-ten Spalte unterhalb des Diagonalelementes von der Einheitsmatrix ab und enthalten dort die Elemente $-\ell_{j+1,j}, \ldots, -\ell_{n,j}$. Elementarmatrizen genügen einigen einfachen Rechenregeln.

1) Anwendung auf einen Vektor:

$$L_j \begin{pmatrix} a_1 \\ \vdots \\ a_j \\ a_{j+1} \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} a_1 \\ \vdots \\ a_j \\ a_{j+1} - \ell_{j+1,j} a_j \\ \vdots \\ a_n - \ell_{n,j} a_j \end{pmatrix}$$

2) Anwendung auf eine (n, m)-Matrix A : $L_j A$ entsteht aus A durch Subtraktion des $\ell_{i,j}$-fachen der j-ten Zeile von der i-ten Zeile, $i = j+1, \ldots, n$. Dies ist genau die Operation, die wir beim j-ten Eliminationsschritt durchgeführt haben.

3) Aus 1) folgt sofort: L_j ist invertierbar, und L_j^{-1} entsteht aus L_j durch Streichen der Vorzeichen der $\ell_{i,j}$.

4) Das Produkt $L_j L_k$, $j < k$, berechnet sich durch "Überlagern" von L_j, L_k:

$$L_j L_k = \begin{pmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & -\ell_{j+1,j} & \ddots & & & \\ & & \vdots & & 1 & & \\ & & & & -\ell_{k+1,k} & \ddots & \\ & & -\ell_{n,j} & & -\ell_{n,k} & & 1 \end{pmatrix}$$

Neben den Elementarmatrizen benötigen wir noch Permutationsmatrizen. Sei $\{i_1, \ldots, i_n\}$ eine Permutation von $\{1, \ldots, n\}$ und sei e_i der i-te Einheitsvektor in K^n. Dann ist

$$P = \begin{pmatrix} e_{i_1}^* \\ \vdots \\ e_{i_n}^* \end{pmatrix}$$

die zur Permutation $\{i_1, \ldots, i_n\}$ gehörende Permutationsmatrix. Auch Permutationsmatrizen genügen einfachen Rechenregeln.

1) Anwendung auf einen Vektor:

$$P\begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} a_{i_1} \\ \vdots \\ a_{i_n} \end{pmatrix}$$

2) Anwendung auf eine (n,m)-Matrix $A : PA$ entsteht aus A durch Permutation der Zeilen entsprechend der Permutation $\{i_1, \ldots, i_n\}$.

3) AP^* entsteht aus A durch Permutation der Spalten gemäß der Permutation $\{i_1, \ldots, i_n\}$.

4) P ist invertierbar, und $P^{-1} = P^*$. Insbesondere ist $PP^* = P^*P = I$, also P unitär.

Zum Schluß wollen wir noch das Zusammenspiel von Elementar- und Permutationsmatrizen betrachten. Sei P eine Permutationsmatrix, welche nur Zeilen $> j$ vertauscht, d.h. $i_1 = 1, \ldots, i_j = j$. Dann gilt $PL_jP^* = L'_j$, wobei L'_j die gleiche Form wie L_j hat, aber mit (gemäß der Permutation) vertauschten Elementen $\ell'_{k,j} = \ell_{i_k,j},\ \ k > j$. Dies sieht man sofort, wenn man

$$L_j = I + \begin{pmatrix} 0 & & & & \\ & \ddots & & & \\ & & 0 & & \\ & & -\ell_{j+1,j} & \ddots & \\ & & -\ell_{n,j} & & 0 \end{pmatrix}$$

schreibt und die Wirkung von Links- bzw. Rechtsmultiplikationen mit P und P^* studiert. Es folgt die Vertauschungsrelation $PL_j = L'_jP$.

Wir können nun das Eliminationsverfahren durch rekursive Linksmultiplikation der Matrix (A, b) mit Permutations- und Elementarmatrizen darstellen. Sei P_j die Permutations- und Elementarmatrix, welche die Zeilenvertauschung (z.B. durch maximale Spaltenpivotsuche) vor dem j-ten Eliminationsschritt ausführt. P_j vertauscht also nur die Zeilen $j, \ldots, n$ untereinander. Sei L_j die Elementarmatrix mit $\ell_{i,j} = a_{i,j}/a_{i,j}$, wobei $a_{i,j}$ nun das (i,j)-Element vor Beginn der Elimination im j-ten Eliminationsschritt, also unmittelbar nach Anwendung von P_j ist. Das Eliminationsverfahren lautet dann

$$L_{n-1}P_{n-1} \cdots L_2P_2L_1P_1(A, b) = (R, y)\ .$$

Dabei ist R die rechte Dreiecksmatrix, die im Laufe des Eliminationsprozesses entsteht, und y die zugehörige rechte Seite. Das Produkt auf der linken Seite kann man durch die Vertauschungsregel $P_k L_j = L'_j P_k, \;\; k > j$, vereinfachen. Für $n = 4$ ist z.B.

$$\begin{aligned} L_3 P_3 L_2 P_2 L_1 P_1 &= L_3 P_3 L_2 L'_1 P_2 P_1 \\ &= L_3 L'_2 P_3 L'_1 P_2 P_1 \\ &= L_3 L'_2 L''_1 P_3 P_2 P_1 \\ &= L^{-1} P \;\;, \end{aligned}$$

$$L = (L''_1)^{-1} (L'_2)^{-1} L_3^{-1} \quad , \quad P = P_3 P_2 P_1 \; .$$

Nach unseren Rechenregeln ist L eine linke Dreiecksmatrix, die durch "Überlagern" der Elemente $\ell_{i,j}$, $i > j$, entsteht. P ist diejenige Permutationsmatrix, die alle während der Elimination aufgetretenen Zeilenvertauschungen ausführt.

Satz 1.2.1 *Zu jeder (n, n)-Matrix A gibt es eine Permutationsmatrix P, so daß PA eine LR-Zerlegung mit $\ell_{i,i} = 1, \;\; i = 1, \ldots, n$ hat.*

Bemerkungen:

1) Für die Gültigkeit von Satz 1 ist es unerheblich, ob A invertierbar ist oder nicht. Ist A nicht invertierbar, so kann man einmal kein von Null verschiedenes Pivotelement mehr finden. In diesem Fall kann man den Eliminationsschritt unterlassen, also $L_j = P_j = I$ setzen.

2) Wie das Beispiel $A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ zeigt, ist der Satz ohne die Permutation P falsch.

1.3 Fehlerabschätzung bei linearen Gleichungssystemen

Im letzten Paragraphen haben wir eine Methode zur Bestimmung der Lösung eines linearen Gleichungssystems kennengelernt. Wir werden nun die Abhängigkeit dieser Lösung von Störungen untersuchen. Hierzu zunächst ein

Beispiel:

Löse $Ax = b$ mit

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 0.99 \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Wir erhalten die Lösung $x = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$. Statt A, b seien nun nur die fehlerbehafteten Näherungen $\tilde{A}$, $\tilde{b}$ bekannt. Es gelte

$$\tilde{A} = \begin{pmatrix} 1.01 & 1.01 \\ 1 & 0.99 \end{pmatrix} \quad \tilde{b} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \Rightarrow \tilde{x} = \begin{pmatrix} 200/101 \\ -100/101 \end{pmatrix} .$$

Obwohl wir also einen Fehler von nur 1% in den Daten haben, bekommen wir einen Fehler von über 100% in der Lösung.

Wir werden versuchen, dies Phänomen zu erklären. Hierzu wiederholen wir einige Grundbegriffe der linearen Algebra.

Definition 1.3.1 *Eine Abbildung* $\| \cdot \| : V \to \mathbb{R}^{\geq 0}$ *eines* $\mathbb{C}$*-Vektorraums* V *in die nichtnegativen reellen Zahlen heißt Norm, falls für alle* x, $y \in V$, $\alpha \in \mathbb{C}$ *gilt*

1) $\|x\| = 0 \Leftrightarrow x = 0$

2) $\|\alpha x\| = |\alpha| \|x\|$

3) $\|x + y\| \leq \|x\| + \|y\|$ *(Dreiecksungleichung).*

Beispiele:

Sei $V = \mathbb{C}^n$. Wir benutzen

a) Euklidische Norm: $\|x\|_2 = \left(\sum\limits_{k=1}^{n} |x_k|^2\right)$

b) ∞-Norm: $\|x\|_\infty := \max\limits_k |x_k|$

c) T-Norm: $\|x\|_T := \|Tx\|_\infty$

Wir zeigen für $\|\cdot\|_T$ die Dreiecksungleichung:

$$\|x+y\|_T = \|Tx+Ty\|_\infty \leq \|Tx\|_\infty + \|Ty\|_\infty = \|x\|_T + \|y\|_T$$

Definition 1.3.2 *$\|\cdot\|_1$, $\|\cdot\|_2$ heißen äquivalent, falls es positive Konstanten c_1, c_2 gibt mit*

$$c_1\|\cdot\|_1 \leq \|\cdot\|_2 \quad \leq \quad c_2\|\cdot\|_1$$

Satz 1.3.1 *In endlich-dimensionalen Vektorräumen sind alle Normen äquivalent.*

Beweis: In jedem der angegebenen Lehrbücher.

Korollar 1.3.1 *Sei (x_n) Folge im Vektorraum V und V endlich-dimensional. Seien $\|\cdot\|_1$ und $\|\cdot\|_2$ Normen in V. Dann gilt:*

$$\begin{array}{rl} & x_n \textit{ konvergiert gegen } x \quad \textit{bezüglich Norm } \|\cdot\|_1 \\ \Leftrightarrow & x_n \textit{ konvergiert gegen } x \quad \textit{bezüglich Norm } \|\cdot\|_2. \end{array}$$

Beweis: $\|x_n - x\|_2 \leq c_2\|x_n - x\|_1 \overset{n\to\infty}{\to} 0$

$$\|x_n - x\|_1 \leq \tfrac{1}{c_1}\|x_n - x\|_2 \overset{n\to\infty}{\to} 0$$

Wir wollen nun spezielle Normen im Vektorraum der Matrizen definieren.

Definition 1.3.3 *Sei $\|\cdot\|$ eine Norm des $\mathbb{C}^n$. Dann heißt*

$$\|\cdot\| : \mathbb{C}^{(n,n)} \to \mathbb{R}^{\geq 0}\ , \qquad \|A\| := \sup_{x\in\mathbb{C}^n} \frac{\|Ax\|}{\|x\|}$$

die zugeordnete Matrizennorm.

Bemerkung: Es gilt

$$\|A\| = \sup_{x \in \mathbb{C}^n} \frac{\|Ax\|}{\|x\|} = \sup_{x \in \mathbb{C}^n} \|A \frac{x}{\|x\|}\| = \sup_{\|x\|=1,\, x \in \mathbb{C}^n} \|Ax\| \; .$$

Korollar 1.3.2 *$\| \cdot \|$ hat folgende Eigenschaften:*

1) $\| \cdot \|$ ist eine Vektorraumnorm im Vektorraum der Matrizen.

2) Sei λ Eigenvektor von A. Dann gilt $\|A\| \geq |\lambda|$.
<u>Beweis:</u> Sei x Eigenvektor von A zum Eigenwert λ, $\|x\| = 1$.
Dann ist $\|A\| \geq \|Ax\| = \|\lambda x\| = |\lambda| \|x\| = |\lambda|$.

3) $\|AB\| \leq \|A\| \|B\|$.
<u>Beweis:</u> Sei $B \neq 0$.

$$\|AB\| = \sup_{x \in \mathbb{R}^n} \frac{\|ABx\|}{\|x\|} = \sup_{x \in \mathbb{R}^n,\, Bx \neq 0} \frac{\|ABx\|}{\|Bx\|} \cdot \frac{\|Bx\|}{\|x\|} \leq \|A\| \cdot \|B\|.$$

Beispiele:

a) Unendlichnorm: Sei $\|x\|_\infty = 1,\ A \neq 0$.

$$\begin{aligned} \|Ax\|_\infty &= \max_i |\sum_{k=1}^n a_{ik} x_k| \\ &\leq \max_i \sum_{k=1}^n |a_{ik}||x_k| \\ &\leq \max_i \sum_{k=1}^n |a_{ik}|, \quad \text{also} \quad \|Ax\|_\infty \leq \max_i \sum_{k=1}^n |a_{ik}| \; . \end{aligned}$$

Das $\max_i \sum\limits_{k=1}^n |a_{ik}|$ werde für $i = j$ angenommen. Definiere

$$\tilde{x}_k := \begin{cases} \overline{a_{ik}}/|a_{ik}| & , \quad a_{ik} \neq 0 \\ 0 & \quad\ \text{sonst} \; . \end{cases}$$

Dann gilt $\|\tilde{x}\| = 1$, und

$$\|A\tilde{x}\|_\infty = \max_i |\sum_{k=1}^n a_{ik}x_k| \geq |\sum_{k=1}^n a_{jk}x_k| = \sum_{k=1}^n |a_{jk}| = \max_i \sum_{k=1}^n |a_{ik}| \; .$$

Also gilt $\|A\| = \max_i \sum\limits_{k=1}^n |a_{ik}|$.

b) Euklidische Norm: Es gilt $\|A\|_2 = \rho(A^*A)^{1/2}$, wobei $\rho(x)$ der Betrag des betragsmäßig größten Eigenwerts von X ist.
Beweis: Übungsaufgabe 6.

c) T–Norm:

$$\begin{aligned}\| \cdot \|_T : \|A\|_T &= \sup_{x\in\mathbb{R}^n} \frac{\|TAx\|_\infty}{\|Tx\|_\infty} = \sup_{Ty\in\mathbb{R}^n} \frac{\|TA_{-1}y\|_\infty}{\|TT^{-1} - y\|_\infty} \\ &= \sup_{y\in\mathbb{R}^n} \frac{\|TAT^{-1}y\|_\infty}{\|y\|_\infty} = \|TAT^{-1}\|_\infty \; .\end{aligned}$$

Wir haben gezeigt, daß $\rho(A) \leq A$ für jede Matrix A und daß für hermitesche Matrizen $\|A\|_2 = (\rho(A^*A))^{1/2} = \rho(A^2)^{1/2} = \rho(A)$. Man kann daher fragen: Gibt es für jede Matrix A eine (von A abhängige) Vektornorm $\| \cdot \|_A$, so daß $\|A\|_A = \rho(A)$? Die Antwort ist nein, aber es gilt der

Satz 1.3.2 *Zu jedem $A \in \mathbb{C}^{(n,n)}$ und jedem $\varepsilon > 0$ existiert eine Vektornorm $\| \cdot \|_{A,\varepsilon}$ auf dem $\mathbb{C}^n$, so daß für die zugeordnete Matrixnorm*

$$\|A\|_{A,\varepsilon} \leq \rho(A) + \varepsilon \; .$$

gilt.

Beweis: Sei

$$D = \operatorname{diag}(1, \varepsilon, \ldots, \varepsilon^{n-1}) = \begin{pmatrix} 1 & & & \mathbf{O} \\ & \varepsilon & & \\ & & \ddots & \\ \mathbf{O} & & & \varepsilon^{n-1} \end{pmatrix} .$$

Bei der Bildung von BD für eine Matrix B wird Spalte k von B mit ε^{n-1} multipliziert. Bei der Bildung von $D^{-1}B$ wird Zeile k von B mit ε^{-n+1} multipliziert.
Sei nun $J = P^{-1}AP$ die Jordan–Normalform von A. J hat die Form

$$\begin{pmatrix} \lambda_1 & \mu_1 & & \mathbf{O} \\ & \ddots & \ddots & \\ & & \ddots & \mu_{n-1} \\ \mathbf{O} & & & \lambda_n \end{pmatrix},$$

wobei die λ_n Eigenwerte von A sind und $\mu_1 \in \{1, 0\}$. Dann hat $C := D^{-1}JD$ die Form

$$\begin{pmatrix} \lambda_1 & \varepsilon\mu_1 & & \mathbf{O} \\ & \ddots & \ddots & \\ & & \ddots & \varepsilon\mu_{n-1} \\ \mathbf{O} & & & \lambda_n \end{pmatrix}.$$

Definiere $\|x\|_{A,b} := \|x\|_T$ mit $T := (PD)^{-1}$. Dann gilt:

$$\|A\|_T = \|D^{-1}P^{-1}APD\|_\infty = \|C\|_\infty \leq \rho(A) + \varepsilon .$$

Wir haben damit alle benötigten Hilfsmittel zur Verfügung gestellt. Zur Untersuchung der Fehlerabhängigkeit benutzen wir

Definition 1.3.4 *Sei $A \in \mathbb{C}^{(n,n)}$ invertierbar. Dann heißt $k(A) = \|A\|\|A^{-1}\|$ die Kondition von A bezüglich $\|\cdot\|$.*

Satz 1.3.3 *Sei $A \in \mathbb{C}^{n\times n}$ invertierbar, und $\Delta A \in \mathbb{C}^{(n,n)}$ eine Matrix mit $\|A^{-1}\|\|\Delta A\| < 1$. Dann gilt:*

a) *$(A + \Delta A)$ ist invertierbar,*
$\|(A + \Delta A)^{-1}\| \leq \frac{\|A^{-1}\|}{1-\|A^{-1}\|\|\Delta A\|}$.

b) *Sei $b \in \mathbb{C}^n\backslash\{0\}, \Delta b \in \mathbb{C}^n$. Seien x, Δx die Lösungen von $Ax = b$ und $(A + \Delta A)(x + \Delta x) = (b + \Delta b)$. Dann gilt*

$$\frac{\|\Delta x\|}{\|x\|} \leq \frac{k(A)}{1 - k(A)\frac{\|\Delta A\|}{\|A\|}} \{ \frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta b\|}{\|b\|} \} .$$

Beweis:

zu a) $A + \Delta A = A(I + A^{-1}\Delta A)$. Es gilt für $y \neq 0$:

$$\|(I + A^{-1}\Delta A)y\| \geq \|y\| - \|A^{-1}\Delta A y\| \geq \|y\|(1 - \|A^{-1}\|\|\Delta A\|) > 0.$$

Die zur Matrix $(I + A^{-1}\Delta A)$ gehörende lineare Abbildung ist also injektiv und damit ein Vektorraumisomorphismus als Abbildung von $\mathbb{C}^n$ nach $\mathbb{C}^n$. Deshalb sind $(I + A^{-1}\Delta A)$ und $A + \Delta A = A(I + A^{-1}\Delta A)$ invertierbar.

Setze nun $y := (I + A^{-1}\Delta A)^{-1}x$. Durch Einsetzen erhalten wir

$$\|x\| \geq \|(I + A^{-1}\Delta A)^{-1}x\|(1 - \|A^{-1}\|\|\Delta A\|) .$$

Für jedes $x \neq 0$ gilt damit

$$\|(I + A^{-1}\Delta A)^{-1}x\| \leq \frac{1}{1 - \|A^{-1}\|\|\Delta A\|} \cdot \|x\|$$

oder

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

Damit gilt:

$$\|(A + \Delta A)^{-1}\| = \|(I + A^{-1}\Delta A)^{-1}A^{-1} \leq \frac{\|A^{-1}\|}{1 - \|A^{-1}\|\|\Delta A\|}$$

zu b) Wir betrachten zunächst das Gleichungssystem

$$\begin{aligned} (A + \Delta A)(x + \Delta x) &= b + \Delta b \\ Ax &= b. \end{aligned}$$

Durch Subtraktion erhalten wir

$$(A + \Delta A)\Delta x = b - \Delta A \cdot x$$

und damit

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

Mit den Norm–Rechenregeln gilt:

$$\begin{aligned}
\frac{\|\Delta x\|}{\|x\|} &\leq \frac{1}{\|x\|}\|(A+\Delta A)^{-1}\|\|b - \Delta A \cdot x\| \\
&\leq \frac{\|A^{-1}\|}{1 - \|A^{-1}\|\|\Delta A\|}\left(\frac{\|\Delta b\|}{\|x\|} + \|\Delta A\|\right) \\
&= \frac{k(A)}{1 - k(A)\frac{\|\Delta A\|}{\|\Delta\|}}\left(\frac{\|\Delta b\|}{\|A\|\|x\|} + \frac{\|\Delta A\|}{\|A\|}\right) \\
&\leq \frac{k(A)}{1 - k(A)\frac{\|\Delta A\|}{\|A\|}}\left(\frac{\|\Delta b\|}{\|b\|} + \frac{\|\Delta A\|}{\|\Delta\|}\right)
\end{aligned}$$

Bei kleinem Fehler $\|\Delta A\|$ wird der relative Fehler von Matrix und Ergebnisvektor um den Faktor $k(A)$ erhöht (verstärkt).

Wir wollen nun unser einführendes Beispiel aufklären. Es galt:

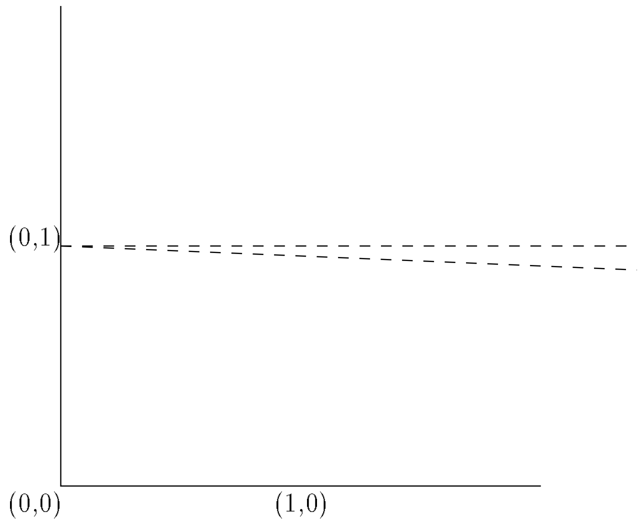
$$A = \begin{pmatrix} 1 & 1 \\ 1 & 0.99 \end{pmatrix}, \quad A^{-1} = \begin{pmatrix} 99 & -100 \\ -100 & 100 \end{pmatrix}.$$

Wir erhalten:

$$k(A)_\infty = 2 \cdot 200 = 400 .$$

Wir müssen also damit rechnen, daß ein gegebener Anfangsfehler in A und b sich um einen Faktor 400 verstärkt in x auswirkt.

Wir betrachten das Beispiel nun geometrisch. Die Lösung des Gleichungssystems läßt sich auch graphisch als Schnittpunkt von zwei Geraden bestimmen.



Grafische Darstellung des Modellproblems

Eine kleine Änderung von A oder b bewirkt eine große Änderung der Koordinaten des Schnittpunktes.

Wir wollen die Fehlerbetrachtungen auf die Rundungsfehler anwenden. Bei Lösung von $Ax = b$ entstehen zunächst bei der Eingabe auf dem Rechner Rundungsfehler $\Delta A = A - \tilde{A}$, $\Delta b = b - \tilde{b}$. Wir führen die Maschinengenauigkeit

$$\text{eps} = \max_{x \neq 0} \left| \frac{x - \tilde{x}}{x} \right| \tag{3.1}$$

ein, wo x alle reellen Zahlen durchläuft und $\tilde{x}$ die Rundung von x bedeutet. Bei m-stelliger dezimaler Gleitpunktarithmetik ist eps $\sim 10^{-m}$; der genaue Wert hängt von Zahldarstellung und Rundungsoperation ab. Der Exponentenbereich unserer Maschine wird als unendlich angenommen.

Es ist dann

$$\frac{\|\Delta A\|}{\|A\|} \sim \text{eps} \quad , \quad \frac{\|\Delta b\|}{\|b\|} \sim \text{eps} \; .$$

Die Anwendung des Satzes verlangt nun zunächst einmal

$$k(A)\, \text{eps} \; < 1 \; . \tag{3.2}$$

Ist dies der Fall, so folgt aus dem Satz für die Lösung $\tilde{x}$ von $\tilde{A}\tilde{x} = \tilde{b}$ mit

$\Delta x = x - \tilde{x}$

$$\frac{\|\Delta x\|}{\|x\|} \sim k(A)\mathrm{eps}\ . \tag{3.3}$$

Wir nennen dies den unvermeidbaren Fehler. Er ist nicht durch den verwendeten Algorithmus zur Lösung von $Ax = b$ bedingt, sondern allein durch die Maschinengenauigkeit und die Kondition von A bestimmt.

1.4 Die Cholesky-Zerlegung

Eine (n,n)-Matrix $A = (a_{ij})$ heißt hermitesch, wenn $A = A^*$ oder $a_{ij} = \overline{a}_{ji}$. Falls A hermitesch ist, ist $(x, Ay) = (Ax, y)$ reell, und A besitzt n reelle Eigenwerte und ein zugehöriges Orthogonalsystem von n Eigenvektoren. Ist darüber hinaus $(x, Ax) \geq 0$, so heißt A positiv semidefinit. Ist $(Ax, x) > 0$ für $x \neq 0$, so heißt A positiv definit. Ist A positiv definit, so offenbar auch alle Untermatrizen $(a_{ij})_{k \leq i,j \leq \ell}$ mit $k \leq \ell$; insbesondere sind also die Diagonalelemente positiv.

Wir wollen annehmen, die positive definite Matrix A besitze eine LR - Zerlegung. Sei also $A = LR$ und $\ell_{i,i} = 1$, $i = 1, \ldots, n$. Dann ist $A^* = R^* L^*$, also $A = R^* L^*$ eine weitere LR-Zerlegung von A. Sei D die Diagonale von R. Dann ist $A = R^*(D^*)^{-1} D^* L^*$ eine LR-Zerlegung von A, deren linker Faktor nur $1'$sen auf der Hauptdiagonalen hat. Nach der Eindeutigkeit der LR-Zerlegung (Aufgabe 5) ist also $L = R^*(D^*)^{-1}$ und $D^* L^* = R$. Es folgt $D = D^*$, also D reell (im Falle $K = \mathbb{C}$, sonst ist diese Aussage leer). Es ist also $R = DL^*$ mit einer reellen Diagonalmatrix D, d.h. R und L^* stimme bis auf eine reelle Diagonalmatrix überein. Nach geeigneter Fixierung der Diagonalen von L kann man also $R = L^*$ annehmen, und wir kommen zu

$$A = LL^* \, . \tag{4.1}$$

Dies nennt man die Choleskey-Zerlegung von A. Der folgende Satz gibt die genaue Eindeutigkeitsbedingung, sein Beweis einen bequemen Algorithmus.

Satz 1.4.1 *Sei A positiv definit. Dann gibt es genau eine linke Dreiecksmatrix L mit positiven Diagonalelementen, so daß $A = LL^*$.*

Beweis: $A = LL^*$ bedeutet elementweise geschrieben

$$\sum_{k=1}^{j} \ell_{i,k} \overline{\ell}_{j,k} = a_{i,j} \quad , \quad n \geq i \geq j \geq 1 \; ; \tag{4.2}$$

für $K = \mathbb{R}$ kann das „ - " - Zeichen natürlich wegfallen. Das nichtlineare Gleichungssystem (4.2), bestehend aus $n(n+1)/2$ Gleichungen für ebenso viele Unbekannte. Es läßt sich rekursiv auflösen. Die Gleichungen für $j = 1$ lauten

$$\ell_{i,1} \overline{\ell}_{1,1} = a_{i,1} \; , \quad i = 1, \ldots, n \; .$$

Für $i = 1$ ergibt sich $\ell_{1,1} = \sqrt{a_{1,1}}$. Dabei wurde $\ell_{1,1} > 0$ und $a_{1,1} > 0$ benutzt. Für die weiteren Elemente der 1. Spalte von L ergibt sich dann

$$\ell_{i,1} = a_{i,1}/\ell_{1,1} \ , \quad i = 2, \ldots, n \ .$$

Damit ist die erste Spalte von L bestimmt.

Die Gleichungen (4.2) für $j = 2$ lauten

$$\ell_{i,1}\overline{\ell}_{2,1} + \ell_{i,2}\overline{\ell}_{2,2} = a_{i,2} \ , \quad i = 2, \ldots, n \ .$$

Für $j = 2$ ergibt sich

$$\ell_{2,2} = \sqrt{a_{2,2} - |\ell_{2,1}|^2} \ .$$

Dabei wurde angenommen, daß der Radikand positiv ist, und daß $\ell_{2,2} > 0$. Die weiteren Elemente der 2. Spalte von L ergeben sich dann zu

$$\ell_{i,2} = (a_{i,2} - \ell_{i,1}\overline{\ell}_{2,2})/\ell_{2,2} \ , \quad i = 3, \ldots, n \ .$$

Damit ist auch die zweite Spalte von L bestimmt.

Wir wollen nun annehmen, Spalten $1, \ldots, j-1$ von L seien bereits bestimmt. Dann schreiben wir die Gleichung (4.2) für die j-te Spalte hin, also

$$\ell_{i,1}\overline{\ell}_{j,1} + \cdots + \ell_{i,j-1}\overline{\ell}_{j,j-1} + \ell_{i,j}\overline{\ell}_{j,j} = a_{i,j} \ , \quad i = j, \ldots, n \ .$$

Für $i = j$ ergibt sich aus $\ell_{j,j} > 0$ sofort

$$\ell_{j,j} = \sqrt{a_{j,j} - |\ell_{j,1}|^2 - \cdots - |\ell_{j,j-1}|^2} \ , \tag{4.3}$$

wenn nur der Radikand positiv ist. Die weiteren Elemente der j-ten Spalte sind dann

$$\ell_{i,j} = (a_{ij} - \ell_{i,1}\overline{\ell}_{j,1} - \cdots - \ell_{i,j-1}\overline{\ell}_{j,j-1})/\ell_{i,j} \ . \tag{4.4}$$

Die auf der rechten Seite von (4.3), (4.4) auftretenden Elemente von L stehen bis auf $\ell_{j,j}$ alle in den bereits berechneten Spalten von L, und $\ell_{j,j}$ ergibt sich aus (4.3). Wir sehen, daß alle Elemente von L berechnet werden können, wenn nur der Radikand in (4.3) immer positiv ausfällt. Dies wollen wir nun durch vollständige Induktion zeigen. Für $j = 1$ ist dies sicherlich richtig. Sei es richtig bis zu einem $j \geq 1$. Dann lassen sich die Spalten $1, \ldots, j-1$ von L berechnen. Diese bilden die $(n, j-1)$-Matrix L_j, und es gilt

$$L_j L_j^* = \begin{pmatrix} a_{1,1} & & & & & \\ & \ddots & & & & \\ a_{j-1,1} & \cdots & a_{j-1,j-1} & & & \\ a_{j,1} & \cdots & a_{j,j-1} & x_{jj} & & \\ \vdots & & & & \ddots & \\ a_{n,1} & \cdots & a_{n,j-1} & x_{n,j} & \cdots & x_{n,n} \end{pmatrix} . \tag{4.5}$$

Es wurde nur die linke Hälfte der Matrix notiert; die rechte ergibt such aus der Hermitizität. Die Elemente $x_{i,j}$ sind ohne Bedeutung mit Ausnahme von $x_{j,j}$, und dieses ist

$$x_{j,j} = |\ell_{j,1}|^2 + \cdots + |\ell_{j,j-1}|^2 .$$

Wir zeigen, daß $x_{j,j} < a_{j,j}$. Wäre nämlich $x_{j,j} \geq a_{j,j}$, so wäre die (j,j)-Matrix

$$\begin{pmatrix} a_{1,1} & & & \\ \vdots & \ddots & & \\ a_{j-1,1} & & a_{j-1,j-1} & \\ a_{j,1} & & a_{j-1,j} & x_{j,j} \end{pmatrix}$$

(wieder haben wir nur die linke Hälfte notiert) positiv definit, denn dies wäre ja schon für $x_{j,j} = a_{j,j}$ richtig, um so mehr also für $x_{j,j} > a_{j,j}$. Damit hätte aber die rechte Seite von (4.5) mindestens den Rang j, während die linke Seite als Produkt von Matrizen mit höchstens $j-1$ Zeilen bzw. Spalten höchstens den Rang $j-1$ haben kann. Es kann also $x_{j,j} \geq a_{j,j}$ nicht richtig sein. Damit ist $x_{j,j} < a_{j,j}$ für $j = 1, \ldots, n$, d.h. der Radikant in (4.3) ist immer positiv, und L läßt sich in eindeutiger Weise bestimmen.

Der Beweis führt sofort zu einem Programm für die Cholesky-Zerlegung.

```
Cholesky (A, n)   /*  Überschreibt den links unteren Teil von A mit
                      seiner Cholesky-Zerlegung L. Arbeitet nur auf
                      dem linken unteren Teil von A */

{ for  j = 1, ..., n
        for  i = j + 1, ..., n
          {  s = a_{i,j} - a_{i,1} conj(a_{j,1}) - ... - a_{i,j-1} conj(a_{j,j-1}) ;
             if (i = j)
                { if (s <= 0) { print ("Matrix nicht pos. def.") ;
                                exit (1) ;
                              }
                  else a_{j,j} = sqrt (s) ;
                }
             else  a_{i,j} = s/a_{j,j} ;
          }
}
```

Für die Anzahl der benötigten flops findet man

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

Dies entspricht dem Eliminationsverfahren für hermitesche Matrizen (vergleiche Übungsaufgabe 3). Das Gleichungssystem $Ax = b$ kann nach Berechnung von L gelöst werden durch Lösung der Systeme $Ly = b$, $L^*x = y$ wie nach der LR-Zerlegung.

1.5 Die QR-Zerlegung

Sei A eine (n,m)-Matrix, $n \geq m$, mit linear unabhängigen Spalten $a_1, \ldots, a_m \in K^n$. Bekanntlich kann man die Spalten von A orthogonalisieren, d.h. man kann ein Orthonormalsystem von Vektoren $q_1, \ldots, q_m$ finden, so daß

$$\mathrm{sp}(a_1, \ldots, a_j) = \mathrm{sp}(q_1, \ldots, q_j)\ , \quad j = 1, \ldots, m \tag{5.1}$$

gilt. Hierbei bedeutet $\mathrm{sp}(a, b, \ldots)$ den von $a, b, \ldots$ aufgenannten linearen Unterraum. Die Orthogonalisierung kann durch das Schmidtsche Verfahren gemacht werden. Wir setzen

$$q_1 = a_1/r_{1,1}\ , \quad r_{1,1} = \|a_1\|$$

mit der euklidischen Norm. Damit ist (5.1) für $j = 1$ erfüllt. Haben wir bereits orthonormale Vektoren $q_1, \ldots, q_{j-1}$ bestimmt, so setzen wir

$$\hat{q}_j = a_j - r_{1,j} q_1 - \cdots - r_{j-1,j} q_{j-1} \tag{5.2}$$

und bestimmen die $r_{i,j}$, $i = 1, \ldots, j-1$, so, daß $(\hat{q}_j, q_i) = 0$, $i = 1, \ldots, j-1$, also

$$r_{i,j} = (a_j, q_i)\ . \tag{5.3}$$

Dann setzen wir

$$q_j = \hat{q}_j / r_{j,j} \quad , \quad r_{j,j} = \|\hat{q}_j\|\ .$$

So bestimmen wir rekursiv $q_1, \ldots, q_m$. Die $r_{i,j}$ können nicht verschwinden, wenn $a_1, \ldots, a_m$ linear unabhängig sind. Offenbar gilt

$$a_j = r_{1,j} q_1 + \cdots + r_{j,j} q_j\ , \quad j = 1, \ldots, m\ .$$

Mit der (n,n)-Matrix $Q = (q_1, \ldots, q_m)$ und der rechten (m,m)-Dreiecksmatrix R, deren (i,j)-Element $r_{i,j}$ ist für $i \leq j$ lautet dies

$$A = QR\ . \tag{5.4}$$

Dies ist QR-Zerlegung von A. Nach Herleitung ist diese Zerlegung in eine orthonormale Matrix Q und eine rechte Dreiecksmatrix R eindeutig bestimmt, wenn man die Diagonalelemente von R positiv annimmt.

Numerisch ist das Schmidtsche Verfahren völlig ungeeignet. Nach (5.2), (5.3) ist nämlich

$$\|\hat{q}_j\| = \min_{q \in \mathrm{sp}(q_1,\ldots,q_{j-1})} \|a_j - q\| \ .$$

$\hat{q}_j$ wird also sehr klein ausfallen, insbesondere dann, wenn a_j fast linear abhängig von $a_1, \ldots, a_{j-1}$ ist. Bei der Berechnung von $\hat{q}_j$ treten also Auslöschungen auf, so daß auch q_j nicht genau berechnet werden kann.

Ein sehr stabiles Verfahren zur QR-Zerlegung ist das Householder-Verfahren. Wir beschreiben es für reelle Matrizen A. Es macht Gebrauch von Spiegelungen

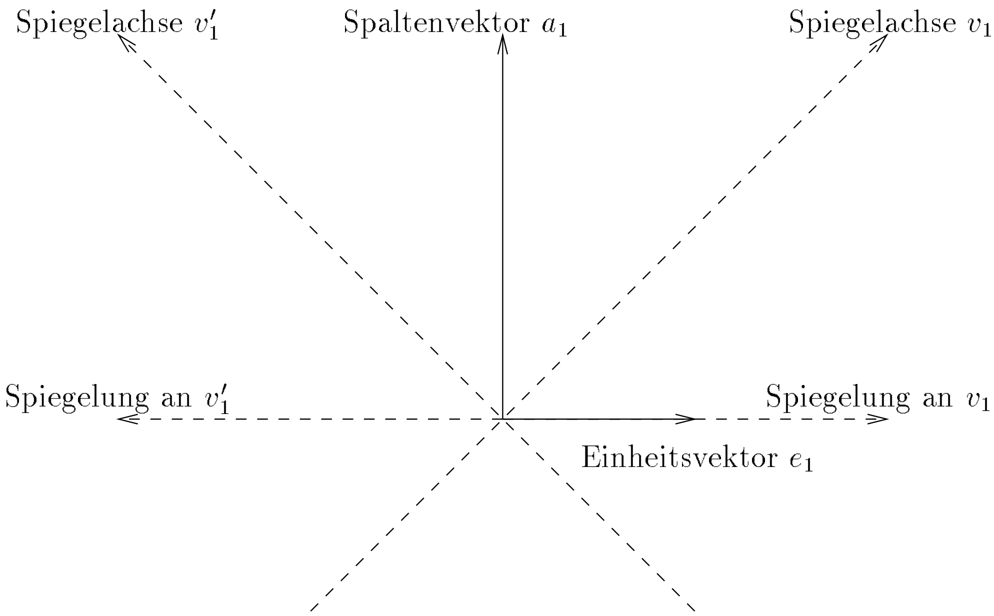
$$S = I - 2vv^* \quad , \quad \|v\| = 1$$

an der Hyperebene $v^\perp$. Dabei tritt das dyadische Produkt $(vv^*)_{ij} = v_i v_j$ auf. Es ist

$$\begin{aligned} S^2 &= (I - 2vv^*)(I - 2vv^*) &&= I - 4vv^* + 4vv^*vv^* \\ &= I - 4vv^* + 4vv^* &&= I \end{aligned}$$

und $S = S^*$. Also ist $SS^* = S^* = I$, d.h. S unitär.

Das Householder - Verfahren bestimmt Spiegelungen $S_1, \ldots, S_{m-1}$, so daß $S_j \cdots S_1 A$ in den Spalten $1, \ldots, j$ bereits rechte Dreiecksgestalt hat. Wir beschreiben ausführlich die Bestimmung von S_1. Die erste Spalte von $S_1 A$ lautet $S_1 a_1$. Wir müssen S_1 also so bestimmen, daß $S_1 a_1$ ein Vielfaches von e_1 ist. Dies kann man auf zwei Weisen erreichen, nämlich



durch Spiegelung an v_1, wo $v_1 = (a_1 + \alpha_1 e_1)/\beta_1$, $\beta_1 = \|a_1 + \alpha_1 e_1\|$ mit $\alpha_1 = \pm\|a_1\|$. Zur Vermeidung von Auslöschung wählt man $\alpha_1 = \|a_1\|$ sgn $(a_{1,1})$. Mit

$$\begin{array}{rclcl} \beta_1^2 & = & \|a_1\|^2 + \alpha_1^2 + 2\alpha a_{1,1} & = & 2\alpha_1(\alpha_1 + a_{1,1}) \\ 2v_1^* a_1 & = & \frac{2}{\beta_1}(a_1 + \alpha_1 e_1)^* a_1 & = & \frac{2}{\beta_1}(\|a_1\|^2 + \alpha_1 a_{1,1}) \\ & = & \frac{2}{\beta_1}\alpha_1(\alpha_1 + a_{1,1}) & = & \beta_1 \end{array}$$

erhalten wir, wie erwartet, für die erste Spalte von $S_1 A$

$$\begin{array}{rcl} S_1 a_1 & = & (I - 2v_1 v_1^*)a_1 = a_1 - 2v_1 v_1^* a_1 \\ & = & a_1 - \beta_1 v_1 = a_1 - (a_1 - \alpha_1 e_1) = -\alpha_1 e_1 \ . \end{array}$$

Die weiteren Spalten sind

$$S_1 a_k = a_k - 2v_1 v_1^* a_k \quad , \quad k = 2, \ldots, m \ .$$

$S_1 A$ hat die Gestalt

$$S_1 A = \left(\begin{array}{cccc} -\alpha_1 & r_{1,2} & \cdots & r_{1,m} \\ 0 & & & \\ \vdots & & A_2 & \\ 0 & & & \end{array} \right) ,$$

wobei A_2 eine $(n-1, n-1)$-Matrix ist. Wir eliminieren nun die Elemente unterhalb des $(2,2)$-Elements durch Linksmultiplikation mit einer Spiegelung der Form

$$S_2 = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & & & \\ \vdots & & I - 2v_2v_2^* & \\ 0 & & & \end{pmatrix},$$

wo nun I die $(n-1, n-1)$ Einheitsmatrix und $v_2 \in \mathbb{R}^{n-1}$, $\|v_2\| = 1$ bedeuten. v_2 berechnet sich aus der ersten Spalte von A_2 genau so, wie sich v_1 aus a_1 berechnete. Es wird dann

$$S_2S_1A = \begin{pmatrix} -\alpha_1 & r_{1,2} & \cdots & & r_{1,m} \\ 0 & -\alpha_2 & r_{2,3} & \cdots & r_{2,m} \\ \vdots & 0 & & & \\ & \vdots & & A_3 & \\ 0 & 0 & & & \end{pmatrix}$$

mit einer $(n-2, n-2)$-Matrix A_3. Nach m Schritten erhält man

$$S_m \cdots S_1 A = \begin{pmatrix} R \\ 0 \end{pmatrix}$$

mit der rechten Dreiecksmatrix

$$R = \begin{pmatrix} -\alpha_1 & r_{1,2} & \cdots & r_{1,m} \\ & \ddots & & \vdots \\ \mathbf{O} & & & r_{m-1,m} \\ & & & -\alpha_m \end{pmatrix}.$$

Es folgt

$$A = S_1 \cdots S_m \begin{pmatrix} R \\ 0 \end{pmatrix} = S \begin{pmatrix} R \\ 0 \end{pmatrix} = QR\,,$$

wobei A aus den Spalten $1, \ldots, m$ von S besteht. Damit haben wir die QR-Zerlegung von A gefunden.

Es ist nicht zweckmäßig, die Matrix S oder Q wirklich zu berechnen. Es ist nämlich sehr einfach, Sx alleine mit Hilfe der Vektoren $v_j \in \mathbb{R}^{n-j+1}$ zu berechnen. Es ist nämlich

$$S_j x = \begin{pmatrix} x_{j-1} \\ x_{n-j-1} - 2(v_j, x_{n-j-1})v_j \end{pmatrix} \quad , \quad x = \begin{pmatrix} x_{j-1} \\ x_{n-j+1} \end{pmatrix} ,$$

wobei x_{j-1}, x_{n-j+1} die Längen $j-1$ bzw. $n-j+1$ haben.

Dies verlangt nur $2(n-j-1) + \mathrm{O}(1)$ flops. Die sukzessive Berechnung von $Sx = S_1 \cdots S_m x$ erfordert daher etwa für $n = m$ nur $m^2 + \mathrm{O}(m)$ flops, also ebenso viele wie die Berechnung von Sx bei vorberechnetem S.

Ein Programm zur QR-Zerlegung berechnet also zweckmäßigerweise nicht Q, sondern die Spiegelungsvektoren $v_1, \ldots, v_m$.

qr_- dump (A, α, n, m)

/ $\star$ Führt die QR-Zerlegung der (m, n)-Matrix A durch. Nach Ablauf enthält A in und unterhalb der Diagonalen die Vektoren $v_1, \ldots, v_m$ und oberhalb der Diagonalen die Außerdiagonalelemente von R. Die Diagonalelemente von R werden auf den Vektor α geschrieben. $\star$ /

```
{  for  j = 1,...,m
   {    α_j = ||a_j||⋆ sgn (a_{j,j}); β = sqrt(2α_j ⋆ (α_j + a_{j,j}));
        a_{j,j} = (a_{j,j} + α_j)/β;
        for i = j+1,...,n   a_{i,j} = a_{i,j}/β;
   for  k = j+1,...,m
        { γ = 0;
        for i = j,...,n    γ = γ + a_{ik} ⋆ a_{ij};
        γ = 2 ⋆ γ;
        for i = j,...,n    a_{i,k} = a_{i,k} − γ ⋆ a_{i,j};
        }
   }
}
```

Dieses Programm wird noch ergänzt durch zwei weitere Programme, welche Sx bzw. S^*x bilden.

q_- mal $x(A, x, n, m)$

$/\star$ Überschreibt x mit Sx nach Aufruf von
$qr\_$ dump $(A, \alpha, n, m) \star /$

$\{$ for $j = m, \ldots, 1$
$\{\ \gamma = 2 \star (a_{j,j} \star x_j + \cdots + a_{n,j} \star x_n);$
for $i = j, \ldots, n \quad x_i = x_i - \gamma \star a_{i,j};$
$\}$
$\}$

q^*_- mal$_x(A, x, n, m)$

$/\star$ Überschreibt x mit S^*x nach Aufruf von
$qr\_$ dump $(A, \alpha, n, m) \star /$

$\{$ Wie oben, aber $j = 1, \ldots, m\}$

1.6 Unter- und überbestimmte lineare Systeme

Sei A eine (n,m)-Matrix über K. Das lineare Gleichungssystem $Ax = b$ heißt überbestimmt für $n > m$, unterbestimmt für $n < m$. Im ersten Fall ist $Ax = b$ in der Regel unlösbar, im zweiten Fall in der Regel nicht eindeutig lösbar. Wir wollen eine verallgemeinerte Lösung von $Ax = b$ definieren, welche immer eindeutig bestimmt ist, und Verfahren zu deren Berechnung angeben.

Wir bezeichnen mit ker (A) den Nullraum, mit range (A) den Wertebereich von A, also

$$\begin{aligned} \mathrm{ker}(A) &= \{x \in K^m : Ax = 0\} \ , \\ \mathrm{range}(A) &= \{y \in K^n : \exists\ x \in K^m \ \text{ mit } \ y = Ax\} \ . \end{aligned}$$

Weiter bezeichnen wir für lineare Unterräume U, V von K^n mit $U + V$ die Summen von Vektoren aus U, V. Ist $U \perp V$, so schreiben wir für $U + V$ auch $U \oplus V$.

Aus der linearen Algebra erinnert man, daß $Ax = b$ genau dann lösbar ist, wenn $b \perp$ ker (A^*). Wir wollen dies etwas anders formulieren.

Satz 1.6.1 *Für jede (n,m)-Matrix A gilt*

$$K^n = ker(A^*) \ \oplus \ \ range(A) \ .$$

Beweis: Zunächst ist ker (A^*) $\perp$ range (A). Ist nämlich $A^*x = 0$ und $y = Az$, so folgt

$$(x,y) = (x, Az) = (A^*x, z) = 0 \ ,$$

also $x \perp y$. Es bleibt zu zeigen, daß ker $(A^*)+$ range $(A) = K^n$ ist. Wäre dies nicht der Fall, so gäbe es ein $y \in K^n$ mit $y \neq 0$ und $y \perp$ ker (A^*), $y \perp$ range (A). Wegen $y \perp$ ker (A^*) wäre $Ax = y$ lösbar, also $y \in$ range (A). Dies steht im Widerspruch zu $y \perp$ range (A) und $y \neq 0$.

Als ersten Schritt zur Definition einer verallgemeinerten Lösung schwächen wir den Lösungsbegriff ab. Wir verlangen nicht mehr, daß $Ax - b = 0$ ist, sondern nur noch, daß $\|Ax - b\|$ möglichst klein ist. Dabei verwenden wir die euklidische Norm.

Satz 1.6.2 *$\|Ax-b\|$ nimmt genau dann für $x = x_0$ sein Minimum an, wenn $A^*Ax_0 = A^*b$.*

Beweis: Nach Satz 1 ist $b = b_1 + b_2$ mit $b_1 \in$ range (A) und $b_2 \in$ ker (A^*). Dann ist $Ax - b_1 \perp b_2$, und wir haben

$$\|Ax - b\|^2 = \|Ax - b_1 - b_2\|^2 = \|Ax - b_1\|^2 + \|b_2\|^2 \, .$$

$\|Ax - b\|$ ist also minimal genau dann, wenn $Ax - b_1 = 0$, und dies ist genau dann der Fall, wenn $A^*(Ax - b_1) = 0$.

□

Bemerkungen:

1) $A^*Ax = A^*b$ heißt System der Normalgleichungen. x^0 heißt Kleinste-Quadrate-Lösung. Man spricht auch von der (Gaußschen) Methode der kleinsten Quadrate.

2) Die Normalgleichungen sind immer lösbar. Denn es ist ker $(A) =$ ker(A^*A), also

$$A^*b \in \ \text{range}(A^*) \ \perp \ \ker(A)$$

und damit $A^*b \perp$ ker (A^*A).

Die zweite Bemerkung zeigt, daß die Kleinste-Quadrate-Lösung von $Ax = b$ immer existiert. Leider ist sie i. allg. nicht eindeutig.

Definition 1.6.1 *x^+ heißt verallgemeinerte (Moore-Penrose-) Lösung von $Ax = b$, wenn*

1) x^+ ist Kleinste-Quadrate-Lösung.

2) Unter allen Kleinste-Quadrate-Lösungen hat x^+ minimale Norm.

Satz 1.6.3 *x^+ ist eindeutig bestimmt. x^+ ist genau dann verallgemeinerte Lösung, wenn*

$$A^*Ax^+ = A^*b \, , \qquad x^+ \in \ range(A^*) \, .$$

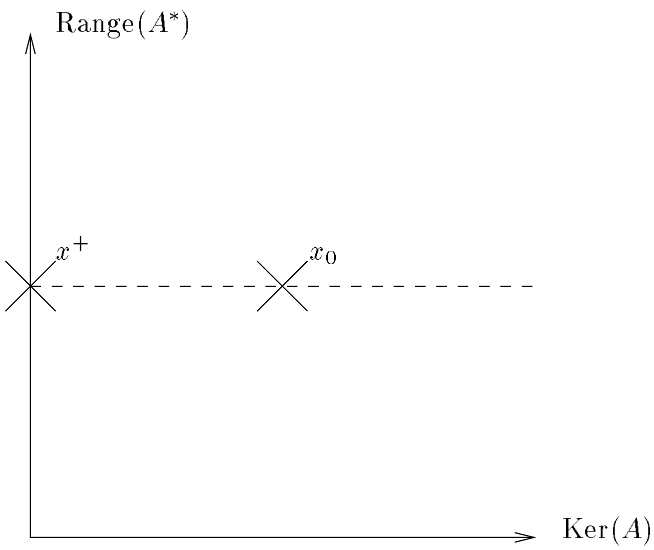
Beweis: Sei x_0 Kleinste-Quadrate-Lösung, also $A^*Ax_0 = A^*b$. Nach Satz 1 ist $x_0 = x_1 + x_2$ mit $x_1 \in$ range(A^*) , $x_2 \in$ ker(A). Auch x_1 erfüllt die Normalgleichungen und ist damit Kleinste-Quadrate-Lösung. Es gibt also immer eine Kleinste-Quadrate-Lösung x_1 in range (A^*). Jede weitere Kleinste-Quadrate-Lösung x ist dann von der Gestalt $x = x_1 + y$ mit $y \in$ ker $(A^*A) =$ ker (A). Wegen $x_1 \perp y$ ist

$$\|x\|^2 = \|x_1\|^2 + \|y\|^2 \ .$$

Die Kleinste-Quadrate-Lösung x^+ minimaler Norm erhält man also in eindeutiger Weise durch $y = 0$ oder $x^+ = x_1$.

□

Für $m = 2$ und $K = \mathbb{R}$ wird Satz 3 auch aus folgender Zeichnung klar:



Die gestrichelte Linie ist der affine Unterraum der Kleinste-Quadrate-Lösungen.

Die Zuordnung $b \to x^+$ ist offenbar linear. Also gibt es eine (m, n)-Matrix A^+ mit x^+b. A^+ heißt verallgemeinerte (Moore-Penrose-) Inverse von A. Ist $n = m$ und A invertierbar, so ist natürlich $A^+ = A^{-1}$. Hat A vollen Rang, so kann man A^+ aus Satz 3 leicht berechnen. Für $n \geq m$ sind dann nämlich die Normalgleichungen eindeutig lösbar, und man bekommt sofort

$$A^+ = (A^*A)^{-1}A^* \,.$$

Ist $n < m$, so ist $x^+ = A^*y$ und $A^*AA^*y = A^*b$, also $AA^*y = b$. Da jetzt AA^* invertierbar ist, folgt $y = (AA^*)^{-1}b$ und $x^+ = A^*(AA^*)^{-1}b$. Also ist in diesem Fall

$$A^+ = A^*(AA^*)^{-1} \,.$$

Die Bildung von Matrizen wie A^*A, AA^* ist nicht unproblematisch.

1) Wir betrachten das Beispiel

$$A = \begin{pmatrix} 1 & 1 \\ \varepsilon & 0 \\ 0 & \varepsilon \end{pmatrix} \,, \quad A^*A = \begin{pmatrix} 1+\varepsilon^2 & 1 \\ 1 & 1+\varepsilon^2 \end{pmatrix} \,.$$

   Ist ε so klein, daß $\varepsilon^2 <$ eps, so kann A^*A auf der Maschine nicht zuverlässig berechnet werden.

2) Sei $n = m$ und A invertierbar. Mit $\|A\| = (\rho(A^*A))^{1/2}$ erhalten wir dann

$$\|A^*A\| = (\rho((A^*A)^2)^{1/2} = \rho(A^*A) = \|A\|^2 \,,$$
$$\|(A^*A)^{-1}\| = \|A^{-1}\|^2$$

   und damit

$$k(A^*A) = k(A)^2 \,.$$

Ist also $k(A) \gg 1$, so ist $k(A^*A) \gg k(A)$. Die Kondition von A^*A ist dann viel schlechter als die von A.

Die Standard-Methode zur Berechnung der verallgemeinerten Lösung von $Ax = b$ ist die QR-Zerlegung. A besitze vollen Rang. Im überbestimmten Fall, also $n \geq m$, führen wir zunächst die QR-Zerlegung von A durch. Die Normalgleichungen lauten dann

$$A^*Ax = R^*Q^*QRx^+ = R^*Rx^+ = R^*Q^*b$$

oder, da R vollen Rang hat,

$$Rx^+ = Q^*b \,. \qquad (6.1)$$

x^+ berechnet sich nun durch Rückwärtseinsetzen. Für A^* bedeutet dies $A^* = R^{-1}Q^*$.

Im unterbestimmten Fall, also $n \le m$, beginnen wir mit der QR-Zerlegung von A^*. Die x^+ charakterisierenden Gleichungen

$$A^*Ax^+ = A^*b \,, \qquad x^+ = A^*y$$

schreiben sich dann als

$$QRR^*Q^*Qz = QRb \,, \quad x^+ = Qz \,.$$

Es ist $Q^*Q = I$, und QR hat maximalen Rang. Also folgt

$$R^*z = b \,, \qquad x^+ = Qz \,. \tag{6.2}$$

Jetzt kann z durch Vorwärtseinsetzen berechnet werden. A^+ ergibt sich zu $A^+ = Q(R^*)^{-1}$.

Im Gegensatz zu den Normalgleichungen haben (6.1), (6.2) vernünftige Kondition. Betrachten wir wieder den Fall $n = m$. Für $A = QR$ ist

$$\|A\| = \|QR\| = \max_{x \ne 0} \frac{\|QRx\|}{\|x\|} = \max_{x \ne 0} \frac{\|Rx\|}{\|x\|} = \|R\|$$

und entsprechend für A^{-1}. Also $k(A) = k(R)$, d.h. die Quadrierung der Kondition findet nicht statt.