Technische Universität Ilmenau
Fakultät für Mathematik
und Naturwissenschaften
Institut für Mathematik

Postfach 10 0565
98684 Ilmenau
Germany
Tel.: 03677/692652
Fax: 03677/691241
Telex: 33 84 23 tuil d.
email: W.Neundorf@mathematik.tu-ilmenau.de

Preprint No. M */96

Manipulation von Matrizen I

Werner Neundorf

Teil I :

Grundlagen, Norm, Kondition und Skalierung

November 1996

‡MSC (1991): 65-01, 65-04, 65F35, 65F25, 68-04, 68Q25

Die Kondition einer Matrix A beeinflußt numerische Fehler, die in solchen Algorithmen wie Elimination und Dekomposition oder anderen Lösungsverfahren auftreten. Ein konvergentes Iterationsverfahren akkumuliert zwar nicht die Rundungsfehler wie eine direkte Methode, aber die Genauigkeit der iterativen Lösung sowie die Konvergenzrate sind abhängig von der Matrixkondition.
Um die numerischen Schwierigkeiten, verursacht durch eine schlechte Kondition der Matrix A, zu vermeiden, kann man hochgenaue Arithmetik nutzen, was natürlich Speicher- und Rechenkosten wachsen läßt. Jedoch gibt es einige andere a-priori "Heilmittel" für dieses Problem. Dazu zählen die Skalierung von A als eine Form der Konditionierung (Teil I) oder die Varianten der Faktorisierung, Transformation bzw. Orthogonalisierung der Matrix (Teil II).
Berücksichtigt man den riesigen Umfang von Publikationen auf diesem Gebiet, so kann der gegebene Überblick nicht erschöpfend sein. Jedoch enthält er subjektiv getroffen eine breite Auswahl von Matrixmanipulationen mit dem Ziel, die Kondition der Matrix zu verbessern oder ein befriedigendes Verhalten bei der Lösung von weiteren Problemen zu erreichen.

The conditioning of a matrix A may take influence on the numerical errors that occur during such algorithms like elimination process and decomposition method or in any other solution method. A convergent iterative procedure does not accumulate rounding errors in the same way as a direct method. Nevertheless, the accuracy of the iterative solution and the rate of convergence are still affected by conditioning of matrix A.
To compensate the numerical difficulties come from the bad condition of A, we may be forced to employ high-precision computations which will further increase the cost of storage and processing. However, there are some other initial "remidies" for this problem such as scaling of A - it's a form of conditioning (Part I) - or the matrix factorization, transformation resp. orthogonalization (Part II).
However due to the vast amount of publications in this area, this survey does not pretend to be exhaustive. Instead it presents a subjectiv but wide selection of matrix manipulations in order to achieve a better condition or a satisfactory behaviour, when solving further problems.

Key words: tutorial aspects, matrix norms, conditioning, scaling,
orthogonalization, factorization, analysis of algorithms, programs.

MSC (1991): 65-01, 65-04, 65F35, 65F25, 68-04, 68Q25

1 Einleitung

Viele Probleme benötigen die Handhabung von Matrizen bzw. die Lösung von linearen Gleichungssystemen. Ausgangspunkt dabei ist, daß man alle bzw. wichtige Informationen über die Matrix nutzt und diese auf ihre weitere Verarbeitung "vorbereitet". Zu solchen Maßnahmen gehören:

- Feststellung von Eigenschaften der Matrix in Bezug auf Symmetrie, (strenge) Regularität, Definitheit, Diagonaldominanz, Orthogonalität u.a.,
- Erkennen und Anwendung der Besetztheitsstruktur,
- Bandbreiten- und Profilreduzierung,
- (symmetrische) Zeilen/Spaltenpermutation,
- Elementeabgleich,
- Zerlegungs- und Transformationstechniken.

In dieser Arbeit wollen wir den Schwerpunkt auf folgende Problemklassen legen.

(1) Skalierung als eine Form der Verbesserung der Kondition der Matrix.

(2) Zelegungsmethoden der Form $A = BC$, $A = BCD$ oder ähnlich unter Einbeziehung von Aspekten, die sie numerisch gutartig machen. Damit ist natürlich formal unter zusätzlichen Bedingungen eine Transformation $C = B^{-1}A$ beschrieben.

(3) Transformationsmethoden der Form $A' = BAC$ möglichst mit Angabe der Transformationsmatrizen B und C, einschließlich der Betrachtung von Sonderfällen.
Ziel dabei ist es, daß die transformierte Matrix A' Eigenschaften besitzt, die ihre weitere Nutzung effizienter machen.

Dazu werden Lösungsalgorithmen bzw. implementierte Routinen in der Programmiersprache *Turbo Pascal* angegeben.
Einige ergänzende grundlegende Abschnitte sowie zahlreiche Beispiele sollen insgesamt das Verständnis für die Problematik unterstützen.

2 Ein Beispiel : Lösung eines Gleichungssystems

Die numerische Stabilität des Eliminationsverfahrens zur Lösung eines linearen GS kann empfindlich gestört werden durch Zeilen bzw. Spalten in der Matrix, die von stark unterschiedlicher Größenordnung sind. Nun ändert sich die Lösung des Systems bekanntlich nicht, wenn einzelne Gleichungen mit einem Faktor multipliziert werden. Die Multiplikation einer Spalte entspricht einer "Dimensionsumrechnung" der entsprechenden Variablen. Nutzt man beide Möglichkeiten, so bekommt man eine Matrix ausgeglichener Größenordnung. Diesen Vorgang nennt man ***Skalierung***.

Die Koeffizienten der Matrix A sind z.B. von annähernd gleicher Größenordnung, wenn die *Zeilen- und Spaltenbetragssummen* ungefähr übereinstimmen.

$$z_i = \sum_{j=1}^{n} |a_{ij}| \;\approx\; s_j = \sum_{i=1}^{n} |a_{ij}|, \quad i,j = 1(1)n.$$

Solche Matrizen heißen *äquilibriert* oder *normalisiert*. Nur für diese ist eine teilweise oder totale Pivotsuche sinnvoll. Ohne die Forderung nach "Gleichgewicht" könnte man jede Zeile durch Multiplikation mit einem hinreichend großen Faktor zur Pivotzeile machen, falls der Kandidat für das Pivotelement nur verschieden von Null ist.
Skalierung und Pivotstrategie können und sollten kombiniert werden. Das führt zu großer numerischer Stabilität bei relativ geringem zusätzlichen Aufwand. Dabei beschränkt man sich meist auf *Spaltenpivotisierung mit Zeilenvertauschung*, wählt aber das Pivotelement relativ zu einer Norm der entsprechenden Zeile der Ausgangsmatrix oder des Teiltableaus, ohne jedoch die Multiplikation mit den Skalierungsfaktoren tatsächlich durchzuführen (implizite Skalierung).
Führt man eine Skalierung explizit aus (beachte den Rechenaufwand dafür), sollte man nur Faktoren wählen, die Potenzen der Zahlenbasis des Rechners sind. So vermeidet man überflüssige zusätzliche Rundungsfehler.
Nachfolgendes Beispiel aus [28] zeigt anschaulich, daß eine Pivotstrategie alleine allgemein nicht ausreicht.

Man löse das lineare GS $Ax = b$ mit

$$A = \begin{pmatrix} 2.1 & 2512 & -2516 \\ -1.3 & 8.8 & -7.6 \\ 0.9 & -6.2 & 4.6 \end{pmatrix}, \quad b = (6.5, -5.3, 2.9)^T,$$

dessen exakte Lösung $x = (5,1,1)^T$ ist.
Als Pivotstrategie wird Spaltenpivotisierung mit Zeilenvertauschung (Kolonnenmaximumstrategie) gewählt. Die Gleichungen sind jedoch schon so angeordnet, daß diese zur Diagonalstrategie wird. Die Rechnungen werden mit 5-stelliger dezimaler Mantisse durchgeführt. Die Matrix $A = A^{(0)}$ wird durch ihre Zerlegungskomponenten L $(l_{ii} = 1)$ und U systematisch überschrieben, und nach 2 Schritten erhält man das transformierte System $Ux = c$.

$A^{(0)}$

0.21000E+1	0.25120E+4	-0.25160E+4	0.65000E+1
-0.13000E+1	0.88000E+1	-0.76000E+1	-0.53000E+1
0.90000E+0	-0.62000E+1	0.46000E+1	0.29000E+1

$A^{(1)}$

0.21000E+1	0.25120E+4	-0.25160E+4	0.65000E+1
-0.61905E+0	0.15639E+4	-0.15651E+4	-0.12762E+1
0.42857E+0	-0.10828E+4	0.10829E+4	0.11430E+0

$A^{(1)}$

0.21000E+1	0.25120E+4	-0.25160E+4	0.65000E+1
-0.61905E+0	**0.15639E+4**	-0.15651E+4	-0.12762E+1
0.42857E+0	-0.10828E+4	0.10829E+4	0.11430E+0

$A^{(2)}$

0.21000E+1	0.25120E+4	-0.25160E+4	0.65000E+1
-0.61905E+0	0.15639E+4	-0.15651E+4	-0.12762E+1
0.42857E+0	-0.69237E+0	0.70000E+0	0.76930E+0

Die Lösung von $Ux = c$ durch Rückwärtseinsetzen liefert

$x_3 = 1.0990,\ x_2 = 1.0990,\ x_1 = 5.1905.$

Die Abweichungen von der exakten Lösung entstehen durch die betragsmäßig großen Koeffizienten $a_{1j}^{(0)}$ im ersten Eliminationsschritt, womit bereits ein Informationsverlust durch Rundung eintritt. Zudem ist im zweiten Schritt bei der Berechnung von

$a_{33}^{(2)} = 1082.9 - 0.69237 * 1565.1 = 1082.9 - 1083.6|282 = 0.70000$

eine katastrophale Stellenauslöschung festzustellen.

Der Grund für das schlechte Ergebnis liegt darin, daß das Pivotelement des ersten Eliminationsschrittes klein ist im Vergleich zu Maximum der Beträge der übrigen Matrixelemente der ersten Zeile. Die Elemente a_{i1} und a_{1j} gehen in symmetrischer Weise in die Reduktionsformel für das erste Tableau ein

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

Eine einfache Maßnahme, die Situation zu verbessern, besteht darin, eine Zeilenskalierung vorzunehmen. Für die neuen Koeffizienten $\tilde{a}_{ij}$ gilt

$$\sum_{j=1}^{n} |\tilde{a}_{ij}| \ = 1, \quad i = 1(1)n.$$

Nach dieser Skalierung wird die Auswahl der Pivotelemente nach der Spaltenpivotstrategie günstig beeinflußt.

Explizite Skalierung mit Zeilenbetragssummen
bei 5-stelliger Rechengenauigkeit.

(1) Gleichungen nach 1. Skalierung

$\tilde{A}^{(0)}$

0.41749E-3	0.49939E-0	-0.50019E-0	0.12922E-2
-0.73446E-1	0.49718E-0	-0.42938E-0	-0.29944E-0
0.76923E-1	-0.52991E-0	0.39316E-0	0.24786E-0

Die Spaltenpivotisierung bestimmt $\tilde{a}_{31}$ zum Pivot, und man führt eine Zeilenvertauschung durch. Mit der neuen ersten Zeile wird die erste Spalte und anschließend das Resttableau berechnet.

(2) Nach 1. Eliminationsschritt

$\tilde{A}^{(1)}$

0.76923E-1	-0.52991E-0	0.39316E-0	0.24786E-0
-0.95480E-0	-0.87800E-2	-0.53990E-1	-0.62780E-1
0.54274E-2	0.50227E-0	-0.50232E-0	-0.53000E-4

Das Resttableau wollen wir wiederum zunächst skalieren, obwohl einfach zu sehen ist, daß das relative Pivot an der Stelle (3,2) sich befindet.

(3) Resttableau nach 2. Skalierung

$\tilde{A}^{(2)}$

0.76923E-1	-0.52991E-0	0.39316E-0	0.24786E-0
-0.95480E-0	-0.13988E-0	-0.86012E-0	-1.00020E-0
0.54274E-2	**0.49998E-0**	-0.50002E-0	-0.52758E-4

Man führt eine zweite Zeilenvertauschung durch und berechnet wiederum die restlichen Koeffizienten neu.

(4) Nach 2. Eliminationsschritt

$\tilde{A}^{(3)}$

0.76923E-1	-0.52991E-0	0.39316E-0	0.24786E-0
0.54274E-2	0.49998E-0	-0.50002E-0	-0.52758E-4
-0.95480E-0	-0.27977E-0	-1.00000E-0	-1.00020E-0

Das Rückwärtseinsetzen liefert mit

$$x_3 = 1.0002,\ x_2 = 1.0002,\ x_1 = 5.0003$$

recht gute Näherungswerte. Für die Koeffizienten der unteren Dreiecksmatrix (Quotienten) gilt $|l_{ik}| \leq 1$. Die Kombination von Pivotstrategie und Skalierungskonzept hat sich somit in diesem Beispiel bewährt.

Die Skalierung der Ausgangsgleichungen überträgt sich natürlich nicht auf die Gleichungen der reduzierten Systeme, so daß der für den ersten Schritt günstige Einfluß der Pivotwahl in späteren Eliminationsschritten verloren gehen kann. Deshalb haben wir hier das reduzierte System auch wieder skaliert. Praktisch tut man es aber nicht sowohl wegen der Vergrößerung des Rechenaufwandes als auch der Erzeugung zusätzlicher Rundungsfehler. Um dennoch das Konzept beizubehalten, wird eine *implizite Skalierung* vorgenommen, d.h. man "denkt sich" die Division durch die Zeilenbetragssummen.

Vor Ausführung des k-ten Eliminationsschrittes ermittelt man den Index l so, daß gilt

$$\frac{|a_{lk}^{(k-1)}|}{\sum_{j=k}^{n} |a_{lj}^{(k-1)}|} = \max_{k \leq i \leq n} \left\{ \frac{|a_{ik}^{(k-1)}|}{\sum_{j=k}^{n} |a_{ij}^{(k-1)}|} \right\} .$$

Ist $l > k$, erfolgt eine Vertauschung der Zeilen. Bei dieser Strategie brauchen die Koeffizienten $l_{ik},\ i > k,$ betragsmäßig nicht mehr durch Eins beschränkt zu sein.

Implizite Skalierung = relative Pivotwahl mit Zeilenbetragssummen
(relative Kolonnenmaximumstrategie) bei 5-stelliger Rechengenauigkeit.

Zur Verdeutlichung sind neben den Schemata die Summen der Beträge der Matrixelemente s_i sowie die für die Pivotwahl ausschlaggebenden Quotienten q_i aufgeführt.

$$s_i = \sum_{j=k}^{n} |a_{ij}^{(k-1)}|, \quad q_i = \frac{|a_{ik}^{(k-1)}|}{s_i}.$$

x_1	x_2	x_3		s_i	q_i
2.1	2512	-2516	6.5	5030.1	0.41749E-3
-1.3	8.8	-7.6	-5.3	17.7	0.73446E-1
0.9	-6.2	4.6	2.9	11.7	**0.76923E-1**

0.9	-6.2	4.6	2.9	-	-
-1.4444	-0.15530	-0.95580	1.1112	1.1111	0.13977
2.3333	**2526.5**	-2526.7	0.26660	5053.2	**0.49998**

0.9	-6.2	4.6	2.9
2.3333	2526.5	-2526.7	-0.26660
-1.4444	-0.61468E-4	-1.1111	-1.1112

Die Unbekannten berechnen sich daraus sukzessiv zu

$$x_3 = 1.0001,\ x_2 = 1.0001,\ x_1 = 5.0001.$$

Die Determinante der Matrix A ergibt sich nach

$det(A) = \prod_{i=1}^{3} u_{ii} = -2526.5$ im Vergleich mit dem exakten Wert -2526.504.

Das vorgestellte Konzept der Matrixzerlegung $PA = LU$ mit impliziter Skalierung fassen wir in algorithmischer Form zusammen und notieren es als TP-Routine. Diese Methode findet man in den meisten Routinen zur LU-Zerlegung der großen Softwarepakete. Für die Ausgangsmatrix und die Zahlenwerte der aufeinanderfolgenden Schemata kann ein und dasselbe Feld benutzt werden. Nach beendeter Zerlegung werden somit

$$a_{ij} = l_{ij},\ i > j,\ l_{ii} = 1, \text{ und } a_{ij} = u_{ij},\ i \le j,$$

bedeuten. Die Information über erfolgte Zeilenvertauschungen wird im Vektor ip aufgebaut. Daraus kann die Permutationsmatrix P abgeleitet werden. Die Toleranzen dienen zu Testen auf Singularität der Matrix.

```
procedure Zerlegung_Gauss_rel_Pivot2(n:integer; var a:matrix;
              eps,detmax:float; var det:float; var ip:vektor1;
              var t,stufe:integer);
 { Gauss-Zerlegung von  A'=P*A(n,n)=L*U  auf dem Platz
   bei relativer Spaltenpivotisierung und Zeilenvertauschung,
   sowie impliziter Skalierung
   ( --> Permutationsmatrix P auf der Basis des Permutationsvektors ip)

   det(A)  Determinante
                                                              n
   eps     Toleranz fuer Test auf Singularitaet |a[i,i]|<...,sum |a[i,j]|<...
                                                             j=i
   detmax  Toleranz fuer Test auf |det(A)|>...
   ip      Permutationsvektor der Zeilenvertauschung
   t       Indikator : 0..default
                       1..Abbruch mit eps
                       2..Abbruch mit detmax
   stufe   Stufe (m,m) bei vorzeitigem Abbruch
   Lit.: H.R.Schwarz. Numerische Mathematik. B.G.Teubner Stuttgart 1988. }

var i,j,l,k:integer;
    q,s,max:float;
begin
 det:=1;
 stufe:=0; t:=0;
 for i:=1 to n do ip[i]:=i;        { Permutationsvektor --> P }

 for k:=1 to n do                  { Schritte k=1,2,...,n }
  begin
   { relative Pivotwahl mit Kolonnenmaximumstrategie }
   max:=0;
   for i:=k to n do
    begin
     s:=0;
     for j:=k to n do s:=s+abs(a[i,j]);
     if s<eps then    { Matrix A bzw. Teiltableau hat "Fast-Null-Zeile" }
      begin stufe:=i; det:=0; t:=1; EXIT; end;
     q:=abs(a[i,k])/s;
     if q>max then begin max:=q; l:=i; end;
    end;
   { Test auf Singularitaet (Nullspalte) und grosses det(A) }
   if max<eps then begin stufe:=k; det:=0; t:=1; EXIT; end;
   max:=a[l,k];
   det:=det*max;
   if abs(det)>detmax then begin stufe:=k; t:=2; EXIT; end;

   { Zeilenvertauschung }
   if l>k then
    begin
     det:=-det;
     for j:=1 to n do
      begin s:=a[k,j]; a[k,j]:=a[l,j]; a[l,j]:=s; end;
     j:=ip[k]; ip[k]:=ip[l]; ip[l]:=j;
    end;

   { Bestimmung der Elemente des Resttableaus }
   for i:=k+1 to n do   { Zeilen }
    begin
     s:=a[i,k]/max;
     a[i,k]:=s;
     for j:=k+1 to n do a[i,j]:=a[i,j]-s*a[k,j];  { Spalten }
    end;
  end;
end;
```

Der Algorithmus ist eigentlich nach $n-1$ Schritten schon abgeschlossen. Der letzte Schritt ($k=n$) dient nur zur Kontrolle des Koeffizienten $a_{nn}=u_{nn}$ und der Determinantenberechnung.
Ein weiterer Programmiertrick ist, daß man die realtive Pivotwahl generell mit einem Vektor von Zeilennormen durchführt, der einmal vorab ermittelt wird. Eine solche Variante findet man in [22]. Dort werden die Zeilennormen $s_i=\sqrt{\sum_{j=1}^{n} a_{ij}^2}$ berechnet und ein sogenannter *Buchhaltervektor* ph mit den inversen Normen s_i^{-1} vorbesetzt. Bei Zeilentausch werden auch seine Komponenten entsprechend vertauscht. Der Aufwand verringert sich damit, ohne aber die Güte des Verfahrens entscheidend zu mindern.

```
procedure Zerlegung_Gauss_rel_Pivot1(n:integer; var a:matrix;
              eps,detmax:float; var det:float; var ip:vektor1;
              var t,stufe:integer);
 { Gauss-Zerlegung von  A'=P*A(n,n)=L*U  auf dem Platz
   bei relativer Spaltenpivotisierung und Zeilenvertauschung
   sowie impliziter Skalierung, Buchhaltervektor
   ( --> Permutationsmatrix P auf der Basis des Permutationsvektors ip)

   det(A)  Determinante
                                                          n
   eps     Toleranz fuer Test auf Singularitaet |a[i,i]|<...,sum a[i,j]^2<...
                                                         j=1
   detmax  Toleranz fuer Test auf |det(A)|>...
   ip      Permutationsvektor der Zeilenvertauschung
   t       Indikator : 0..default
                       1..Abbruch mit eps
                       2..Abbruch mit detmax
   stufe   Stufe (m,m) bei vorzeitigem Abbruch
   Lit.: N.Koeckler. Numerische Algorithmen in Softwaresystemen.
                     B.G.Teubner Stuttgart 1990.                 }

var i,j,l,k:integer;
    s,max,max1:float;
    ph:vektor;
begin
 det:=1;
 stufe:=0;
 t:=0;
 for i:=1 to n do
  begin
   ip[i]:=i;        { Permutationsvektor --> P }
   s:=0;
   for j:=1 to n do s:=s+sqr(a[i,j]);
   if s<eps then    { Matrix A hat "Fast-Null-Zeile" }
    begin stufe:=i; det:=0; t:=1; EXIT end;
   ph[i]:=1/sqrt(s);{ Vektor der Kehrwerte der Euklidischen Norm
                      der n Zeilenvektoren von A,
                      Buchhaltervektor                         }
  end;

 for k:=1 to n do   { Schritte k=1,2,...,n }
  begin
   { relative Pivotwahl mit Buchhaltervektor }
   max:=0;
   max1:=0;
   l:=k;
```

```
   for i:=k to n do
    begin
     s:=a[i,k];
     if abs(s*ph[i])>abs(max1) then
      begin max:=s; max1:=s*ph[i]; l:=i; end
    end;

   { Test auf Singularitaet und grosses det(A) }
   det:=det*max;
   if abs(max)<eps then begin stufe:=k; det:=0; t:=1; EXIT; end;
   if abs(det)>detmax then begin stufe:=k; t:=2; EXIT; end;

   { Zeilenvertauschung }
   if l>k then
    begin
     det:=-det;
     for j:=1 to n do
      begin s:=a[k,j]; a[k,j]:=a[l,j]; a[l,j]:=s; end;
     ph[l]:=ph[k];
     j:=ip[k]; ip[k]:=ip[l]; ip[l]:=j;
    end;

   { Bestimmung der Elemente des Resttableaus }
   for i:=k+1 to n do   { Zeilen }
    begin
     s:=a[i,k]/max;
     a[i,k]:=s;
     for j:=k+1 to n do a[i,j]:=a[i,j]-s*a[k,j];  { Spalten }
    end;
  end;
end;
```

Die Anwendung beider Prozeduren auf die obige Matrix mit dem Gleitkommaformat $float = single$ (das sind 7-8 Mantissenstellen) liefert das gleiche Ergebnis der Zerlegung. Dabei notieren wir nur die richtigen Stellen. Das heißt aber auch, daß zum Beispiel der Koeffizient l_{32} nur 5 gültige Dezimalziffern hat, denn der exakte Wert ist -7/113691 = -6.15703969531449E-5 .

Relative Pivotisierung ist also der Mindestaufwand, der allgemein in einen Zerlegungsalgorithmus zu investieren wäre.

0.8999999	-6.199999	4.5999999
2.333333	2526.466	-2526.7333
-1.4444444	-0.000061570	-1.111127

Später werden wir die wesentlichen Einflußgrößen für gute Pivotstrategien noch weiter konkretisieren.

3 Norm, Konditionszahl, skalierte Konditionszahl

Wir betrachten reelle Vektoren $x \in \mathbb{R}^n$ und reelle Matrizen $A, B \in \mathbb{R}^{n,n}$.

(1) Ausgewählte Vektornormen

$$\|x\|_1 = \sum_{j=1}^{n} |x_j| \qquad \text{(Betragssummennorm, Manhatten-Norm, } l_1\text{-Norm)}$$

$$\|x\|_2 = \sqrt{\sum_{j=1}^{n} |x_j|^2} \qquad \text{(Euklidische Norm, } l_2\text{-Norm)}$$

$$\|x\|_\infty = \max_{j=1(1)n} |x_j| \qquad \text{(Betragsmaximumnorm, Maximumnorm, Tschebyscheff-Norm, } l_\infty\text{-Norm)}$$

$$\|x\|_p = \left(\sum_{j=1}^{n} |x_j|^p \right)^{1/p} \qquad \text{(Hölder-Norm, } l_p\text{-Norm, } p = 1, 2, \infty \text{ Spezialfälle)}$$

$$\|x\|_{p,q} = \left(\sum_{j=1}^{n} q_j |x_j|^p \right)^{1/p} \qquad \text{(gewichtete } l_p\text{-Norm, } q_j > 0)$$

$$\|x\|_B = \sqrt{x^T B x} \qquad \text{(energetische Norm, } B = B^T > 0).$$

(2) *Induzierte* Matrixnorm (zugeordnete, natürliche Norm, Grenzennorm)
Durch die Vektornorm $\|x\|$ induzierte Matrixnorm ist

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

Sie ist mit der zugrundeliegenden Vektornorm zugleich kompatibel und unter allen mit dieser Vektornorm kompatiblen Matrixnormen die kleinste.
Die anschauliche Bedeutung der induzierten Matrixnorm ist die maximale Streckung, die ein Vektor x durch die Abbildung A erfährt.

(3) *Kompatible* Matrixnorm (passende, verträgliche, konsistente Norm)
Eine Matrixnorm $\|A\|_M$ heißt kompatibel zu einer Vektornorm $\|x\|_V$, wenn für alle A und x gilt

$$\|Ax\|_V \leq \|A\|_M \, \|x\|_V.$$

Ist die Abschätzung scharf, d.h. es gilt das Gleichheitszeichen für einen Nichtnullvektor, dann handelt es sich bei der kompatiblen Norm um eine induzierte.

(4) Eigenwert, Eigenvektor, Spektrum und Spektralradius einer Matrix
(reeller Fall)
Eine Zahl λ und ein Nichtnullvektor x, die der Gleichung $Ax = \lambda x$ genügen, heißen Eigenwert und dazugehöriger Eigenvektor der Matrix A.

Spektrum der Matrix : $\sigma(A) = \{\lambda : \lambda \text{ ist EW von } A\} = \{\lambda_1, \lambda_2, ..., \lambda_n\}$.

Spektralradius : $\rho(A) = \max |\lambda(A)| = \max_{1 \leq i \leq n} |\lambda_i(A)|$.

(5) Ausgewählte Matrixnormen

$$\begin{aligned}
\|A\|_\infty = \|A\|_Z &= \max_{i=1(1)n} \sum_{j=1}^{n} |a_{ij}| && \text{(Zeilensummennorm)} \\
\|A\|_1 = \|A\|_S &= \max_{j=1(1)n} \sum_{i=1}^{n} |a_{ij}| && \text{(Spaltensummennorm)} \\
\|A\|_F = \|A\|_E &= \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}^2} && \text{(Frobeniusnorm, Schursche Norm,} \\
&= \sqrt{Spur(A^T A)} && \text{Euklidische Norm)} \\
\|A\|_G = \|A\|_{max} &= n \max_{i,j=1(1)n} |a_{ij}| && \text{(Gesamtnorm, Maximumnorm)} \\
\|A\|_2 &= \sqrt{\max_{i=1(1)n} \mu_i}, && 0 \le \mu_i \in \sigma(A^T A)\ , \\
&= \sqrt{\rho(A^T A)} && \text{(Spektralnorm, Hilbertnorm).}
\end{aligned}$$

Falls $A = A^T$, dann $\|A\|_2 = \max\limits_{i=1(1)n} |\lambda_i(A)|$.

Fobenius- und Spektralnorm sind invariant unter Orthogonaltransformation. Mit einer orthogonalen Matrix Q $(Q^T Q = I)$ gilt nämlich $(QA)^T(QA) = A^T(Q^T Q)A = A^T A$.

(6) Kompatible und induzierte Matrixnormen

Vektornorm	Matrixnorm	
$\|x\|$	Kompatible	Induzierte
$\|x\|_1$	$\|A\|_1,\ \ \|A\|_G$	$\|A\|_1$
$\|x\|_2$	$\|A\|_2,\ \ \|A\|_G,\ \|A\|_F$	$\|A\|_2$
$\|x\|_\infty$	$\|A\|_\infty,\ \|A\|_G$	$\|A\|_\infty$

(7) Konditionszahl und Kondition (Sensitivität) einer Matrix

Für eine reguläre Matrix mit gegebener Norm sind die *(relative) Konditionszahl*

$$cond(A) = \kappa(A) = \|A\| \cdot \|A^{-1}\|$$

und die *absolute Konditionszahl* $acond(A) = \|A^{-1}\|$.

Es gilt $cond(A) \ge 1$. Die Matrix ist schlecht konditioniert, falls $cond(A) \gg 1$. Ist A singulär, definiert man $cond(A) = \infty$. Allgemein gilt

$$cond(A) = \max_{\|x\|=1} \|Ax\| / \min_{\|x\|=1} \|Ax\|.$$

Wenn die Matrix fast singulär ist, dann ist ein Eigenwert fast 0, während ein weiterer in der Größenordnung von $\sum_j |a_{ij}|$ liegt. Sind diese Werte groß, dann ist die Kondition von A i.a. schlecht. Das ist auch daran zu erkennen, daß für eine äquilibrierte Matrix A die Elemente der dazu inversen Matrix betragsmäßig groß werden.

Die Kondition einer Matrix kann man somit als normunabhängig betrachten. Die Konditionszahl ist nicht ohne größeren Aufwand berechenbar. Als grobe Näherung gilt bei einer Dreieckszerlegung $A = LR$ ($l_{ii} = 1$) die aus den Diagonalelementen r_{ii} gebildete Zahl

$$cond_N(A) = \max_{1 \le i,j \le n} |r_{ii}|/|r_{jj}|.$$

Als Begründung möge die Betrachtung der QR-Zerlegung von A mit Q Orthogonalmatrix, R rechte Dreiecksmatrix, und die Beziehung gelten

$$cond_2(A) = cond_2(QR) = cond_2(R) \approx \frac{\max|\lambda(R)|}{\min|\lambda(R)|} = \max_{i,j} \frac{|r_{ii}|}{|r_{jj}|}.$$

Für Diagonalmatrizen stimmt die Zahl $cond_N(A)$ mit den Konditionszahlen $cond_\infty(A)$, $cond_1(A)$ überein.

In [24, 25, 30] sind im Rahmen von Gleichungssystemlösern noch 3 Konditionsschätzungen angegeben, die ebenfalls auf der LR-Zerlegung von A basieren.

- *Hadamardsche* Konditionszahl

$$hcond(A) = \prod_{i=1}^{n} \frac{|r_{ii}|}{\|a_i\|_2} = \frac{|detA|}{\prod_{i=1}^{n} \|a_i\|_2} \le 1, \quad a_i \text{ Zeilenvektoren von } A.$$

  Ist der Wert sehr viel kleiner als 1, so ist die Matrix schlecht konditioniert.

- Konditionsschätzung nach *Forsythe/Moler*
  Dazu benötigt man die durch den Gaußalgorithmus ermittelte Lösung x_0 von $Ax_0 = b$ mit beliebiger rechter Seite (z.B. Einsvektor), das mit doppelter Genauigkeit berechnete Residuum $r_0 = b - Ax_0$ sowie die Lösung des Gleichungssystems für den Fehler $Az = r_0$ unter Verwendung der schon durchgeführten Dreieckszerlegung. Dann erhält man als Schätzung

$$fcond(A) = 2^t \frac{\|z\|_2}{\|x_0\|_2} \in (0, \infty)$$

  mit der Maschinengenauigkeit 2^{-t} (das ist beim TP-Gleitkommaformat *double* 2^{-52} ≈2.2E-16).
  Wird dieser Wert sehr groß im Vergleich zu 1, so kann man die Matrix als schlecht konditioniert betrachten. Werte um bzw. kleiner als 1 verweisen auf eine gute Kondition.

- Konditionsschätzung nach *Cline*
  Hier wird die Kondition $\|A\|_\infty \|A^{-1}\|_\infty$ abgeschätzt. Dazu braucht man die Zerlegung $PA = LR$ mit der Permutationsmatrix P.
  Für R^T müssen dann $x = (\pm 1, \pm 1, ..., \pm 1)^T$ und $y = R^{-T}x$ so bestimmt werden, daß $\|y\|_\infty$ oder $\|y\|_1$ möglichst groß wird.

Weiter ist durch Rückwärtselimination das GS $L^T z = y$ zu lösen. Somit erhält man die Näherung $K = \|z\|_\infty / \|x\|_\infty$ für $\|A^{-1}\|_\infty$. Noch besser ist jedoch der Schätzwert $K = \|z\|_2 / \|x\|_2$.
Der Schätzwert für die Kondition ergibt sich als $\|A\| K$.
Der größte Aufwand liegt hier in der Bestimmung der Vektoren x, y.

```
x[1]:=1; y[1]:=1/R[1,1];   { Komponenten x[1], y[1] fest }
                           { rechte Seite zunaechst = 0 }
for i:=2 to n do y[i]:=-R[1,i]*y[1]/R[i,i];
for k:=2 to n do
 begin
  v:=1/R[k,k];             { Beruecksichtigung von +-1 }
  x[k]:=y[k]-v;         { x[k..n] gleichzeitig als Hilfsvektor }
  y[k]:=y[k]+v;
  SMI:=Abs(x[k]); SPL:=Abs(y[k]);
  for i:=k+1 to n do
   begin
    v:=R[k,i]/R[i,i];
    x[i]:=y[i]-v*x[k];  y[i]:=y[i]-v*y[k];
    SMI:=SMI+Abs(x[i]); SPL:=SPL+Abs(y[i]);
   end;
  if SMI>SPL then begin
                   for i:=k to n do y[i]:=x[i];
                   x[k]:=-1;
                  end
             else x[k]:=1;
  end;
```

(8) Spektrale Konditionszahl

$$cond_2(A) = \begin{cases} \|A\|_2 \|A^{-1}\|_2 = \dfrac{\max\limits_{i=1(1)n} \sqrt{\lambda_i(A^T A)}}{\min\limits_{i=1(1)n} \sqrt{\lambda_i(A^T A)}}, & \text{falls } A \text{ regulär} \\ \dfrac{\max\limits_{i=1(1)n} |\lambda_i(A)|}{\min\limits_{i=1(1)n} |\lambda_i(A)|}, & \text{falls } A = A^T, \text{ regulär} \\ \dfrac{\max\limits_{i=1(1)n} \lambda_i(A)}{\min\limits_{i=1(1)n} \lambda_i(A)}, & \text{falls } A = A^T > 0. \end{cases}$$

Die Größen $\sigma_i = \sqrt{\lambda_i(A^T A)} \geq 0$ werden Singulärwerte der Matrix A genannt.

(9) Eigenschaften der Konditionszahlen

(a) $cond(AB) \leq cond(A)\ cond(B)$ für alle Matrixnormen,

(b) $cond(cA) = cond(A)$ für alle $c \in \mathbb{R}$,

(c) $cond_2(Q) = 1$ für Q orthogonal ($Q^T Q = I$),

(d) $cond_2(A) \leq cond_F(A) \leq cond_G(A) \leq n^2 cond_\infty(A)$,

(e) $n^{-1} cond_{1,F,\infty}(A) \leq cond_2(A) \leq n\ cond_{1,\infty}(A)$,

(f) $cond_2(QA) = cond_2(A)$ für Q orthogonal.

3.1 Berechnung der Inversen der Matrix

Die Genauigkeitsbetrachtungen brauchen die Kenntnis der Inversen von A oder wenigstens einer Näherungsinversen. Eine gute Abschätzung für $\|A^{-1}\|$ und somit für $cond(A)$ ist i.a. rechnerisch aufwendig. Hat man jedoch die Gaußzerlegung von A, so kann man folgende effiziente Implementation der Inversenberechnung anwenden.
Wir beschränken uns auf die Notation der Inversenberechnung auf der Basis der Gaußzerlegung $A = LR$ ohne Pivotstrategie im Fall einer streng regulären Matrix A (siehe [21]).
Man findet dort auch den allgemeinen Fall mittels der Gaußzerlegung einer regulären Matrix in der Form $PAQ = LR$ mit Totalpivotsuche sowie den Zeilen- bzw. Spaltenpermutationsmatrizen P und Q.

Algorithmus:

Seien $A = LR,\ L = (l_{ij}),\ l_{ii} = 1,\ R = (r_{ij})$ und $A^{-1} = (a_{ij}^{(-1)})$.
Rekursionsformeln

$$
\begin{aligned}
a_{nn}^{(-1)} &= \frac{1}{r_{nn}} \\
k &= n-1, n-2, ..., 1 \\
&\left.\begin{aligned}
a_{kj}^{(-1)} &= -\frac{1}{r_{kk}} \sum_{i=k+1}^{n} r_{ki}\ a_{ij}^{(-1)} \\
a_{jk}^{(-1)} &= -\sum_{i=k+1}^{n} a_{ji}^{(-1)}\ l_{ik} \\
a_{kk}^{(-1)} &= \frac{1}{r_{kk}} - \sum_{i=k+1}^{n} a_{ki}^{(-1)}\ l_{ik}.
\end{aligned}\right\} \ j = k+1, k+2, ..., n
\end{aligned}
$$

3.2 Bedeutung der Konditionszahl

Wenden wir uns kurz der Lösung des Gleichungssystems $Ax = b$ zu.
Die Konditionszahl $cond(A)$ der Koeffizientenmatrix ist der entscheidende Faktor, welche sowohl die Genauigkeit einer berechneten Näherung der Lösung als auch die Empfindlichkeit der Lösung gegenüber Änderungen in Matrix und rechter Seite beschreibt.

Fehlerabschätzungen

Es gibt eine Fülle von Varianten der Genauigkeitsbewertung von Ergebnissen der numerischen Rechnung. Sie beinhalten Abschätzungen für den Residuenvektor (Bildvektordifferenz) bzw. Forderungen an den absoluten oder relativen Fehler des Lösungsvektors. Letzteres passiert im Rahmen der sogenannten "Rückwärtsanalyse" (backward analysis), wo $x + \Delta x$ als als exakte Lösung des leicht gestörten Systems

$$(A + \Delta A)\ x' = b + \Delta b$$

betrachtet wird. Dazu benutzen wir kompatible Normen.

(1) Die Größe des *Residuenvektors* $r = A\tilde{x} - b$ der Näherung $\tilde{x}$ führt über die Gleichung $\Delta x = \tilde{x} - x = A^{-1}r$ (Δx wird auch Urbildfehler genannt) und Normabschätzungen auf die Beziehung

$$\|\Delta x\| \leq \|A^{-1}\|\|r\| = acond(A)\ \|r\| \leq \frac{\|C\|}{1 - \|CA - I\|}\|r\|,$$

wobei C genäherte Inverse von A mit $\|CA - I\| < 1$ ist und $S = CA - I$ die zu C gehörige Störmatrix (Restmatrix) sind.
Ist die Störmatrixnorm $\|S\| > 1$, so ist die Dreieckszerlegung zur Berechnung von C offenbar mit so großen Fehlern behaftet, daß diese mit numerisch stabileren Algorithmen und/oder erhöhter arithmetischer Genauigkeit notwendig ist.

(2) Mit obiger Beziehung und $\|b\| \leq \|A\|\|x\|$ erhält man aus (1) unmittelbar

$$\frac{\|\Delta x\|}{\|x\|} \leq \|A\|\|A^{-1}\|\frac{\|r\|}{\|b\|} = cond(A)\frac{\|r\|}{\|b\|}.$$

Beides sind Abschätzungen für den absoluten bzw. relativen Fehler. Sie bedeuten, daß neben einem kleinen Residuenvektor r die Elemente der inversen Matrix bzw. die Kondition der Matrix ausschlaggebend für den Fehler der Näherung $\tilde{x}$ sind. Kleines Residuum heißt also nicht automatisch kleiner Lösungsfehler.

Beispiel:

$$A(2,2) = \begin{pmatrix} 0.780 & 0.563 \\ 0.913 & 0.659 \end{pmatrix}, \quad b = \begin{pmatrix} 0.217 \\ 0.254 \end{pmatrix}, \quad x = \begin{pmatrix} 1 \\ -1 \end{pmatrix},$$

$$A^{-1} = \begin{pmatrix} 659000 & -563000 \\ -913000 & 780000 \end{pmatrix}, \quad cond_\infty(A) \approx 2.7 \cdot 10^6.$$

Betrachten wir zwei Näherungen mit den dazugehörigen Residuen, die deutlich ihren gegenläufigen Trend bei schlechter Matrixkondition widerspiegeln.
$\bar{x} = (0.999, -1.001)^T, \quad \bar{r} = (-0.001343, -0.001572)^T,$
$\hat{x} = (0.341, -0.087)^T, \quad \hat{r} = (0.000001, 0)^T.$

(3) Betrachtet man nur den Einfluß von Störungen der rechten Seite gemäß der Beziehung $A(x + \Delta x) = b + \Delta b$, so sind $A\Delta x = \Delta b$ und

$$\|\Delta x\| \leq \|A^{-1}\|\|\Delta b\| \leq \frac{\|C\|}{1 - \|S\|}\|\Delta b\|$$

mit der Näherungsinversen C und der Störmatrix $S = CA - I$.
Im Prinzip ist es eine andere Interpretation des Falls (1), wobei das Residuum r durch die Störung Δb ersetzt worden ist.

(4) Nimmt man die Näherungsinverse $C = (A + \Delta A)^{-1}$ und das gestörte Gleichungssystem der Form $(A + \Delta A)(x + \Delta x) = b$, so gelten

$$\Delta x = (CA - I)A^{-1}b = Sx, \quad \|\Delta x\| \leq \|S\|\|x\|.$$

Damit erhält man die Störmatrixnorm $\|S\|$ als relatives Fehlermaximum und eine weitere Fehlerkontrolle bei der Lösung von Gleichungssystemen.
Darüber hinaus treten die Störmatrix S und die Näherungsinverse C als erzeugende Matrix in der üblichen Nachiteration (das ist eine Fixpunktiteration)

$$x^{(m+1)} = x^{(m)} - C(Ax^{(m)} - b) = -Sx^{(m)} + Cb$$

für die Fehlerverbesserung auf, wo die Störmatrixnorm im wesentlichen die Kontraktionskonstante für die Konvergenz darstellt.

(5) Die allgemeine Abschätzung für die Empfindlichkeit (siehe [1, 28]) ist

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

bzw.

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

Die Konsequenzen dieser Abschätzungen sind zunächst, daß der Nenner der ersten Brüche der rechten Seite wohl definiert sein muß. Das heißt, das eine schlechte absolute oder relative Konditionszahl prinzipiell nur kleine Störungen der Matrix zuläßt. Dann haben wir die praktische Bedeutung in numerischen Berechnungen bei einer d-stelligen dezimalen Gleitkommaarithmetik.
Ist die Konditionszahl $cond(A) \approx 10^{\alpha}$ und $\nu = 5 \cdot 10^{-d}$, so ergibt sich mit

$$\nu \; cond(A) = 5 \cdot 10^{\alpha - d} \ll 1,$$
$$\|\Delta A\|/\|A\| \leq \nu, \; \|\Delta b\|/\|b\| \leq \nu,$$

aus der zweiten Beziehung von oben die qualitative Abschätzung

$$\frac{\|\Delta x\|}{\|x\|} \leq cond(A) \; 2\nu = 10^{-d+\alpha+1} = \varepsilon.$$

Bemerkungen:

1. Die Schätzung sagt, daß bei Lösung eines Gleichungssystems mit den obigen Annahmen auf Grund von Eingangsfehlern in der berechneten Lösung $\tilde{x}$ nur $d - \alpha - 1$ Dezimalstellen, bezogen auf die betragsgrößte Komponente, sicher sind. Eine pessimistische Aussage, die aber zutreffen kann.

2. Des weiteren spielt neben der Störmatrix auch die Größe $s = d - \alpha - 1$ bei einer Nachiteration des Gleichungssystems eine Rolle. Nur wenn $s > 0$ ist, wird mit jedem Nachiterationsschritt wenigstens eine Verbesserung der Näherungslösung um s Dezimalstellen eintreten und die Folge der Näherungslösungen konvergiert tatsächlich. Die Berechnung des Residuums mit höherer Genauigkeit geht also konform mit der Vergrößerung der Mantissenlänge d (damit $s > 0$).
3. Die Abschätzung $\|\Delta x\|_\infty / \|x\|_\infty \leq \varepsilon$ in der Maximumnorm garantiert nur, daß die betragsgroßen Komponenten von x einen durch ε beschränkten relativen Fehler haben. Der relative Fehler der betragskleinen Komponenten kann beliebig größer als ε sein. Feinere komponentenweise Abschätzungen der Form $|\Delta x_i|/|x_i| \leq \varepsilon_i$ sind wesentlich komplizierter.

(6) Abschließend noch die Betrachtung der Gaußelimination mit der Akkumulation von Rundungsfehlern in den n Schritten. Die Rundungsfehleranalyse (siehe [1]) zeigt wiederum den Einfluß der Matrixkondition. Der erzeugte Fehler $\delta x = \tilde{x} - x$ kann abgeschätzt werden gemäß

$$\|\delta x\| \ \leq \ \nu \ F(n) \ cond(A) \ \|x\|,$$

wobei $\nu = 10^{-d}$ (bzw. 2^{-t}) das Fehlerniveau (Genauigkeit der Gleitkommaarithmetik) und $F(n)$ die vom Verfahren und der Dimension n abhängige Kumulationskonstante bedeuten.
Für die Gaußelimination gilt grob

$$F(n) = \begin{cases} \mathcal{O}(n) & \text{ohne Pivotisierung bei} \\ & A \text{ diagonal dominant oder } A = A^T > 0 \\ \mathcal{O}(2^n) & \text{bei partieller Pivotisierung} \\ \mathcal{O}(n^{3/2}) & \text{bei vollständiger Pivotisierung} \end{cases} .$$

3.3 Skalierte Konditionszahl

A-posteriori-Fehlerschätzungen wie im vorhergehenden Abschnitt erlauben keine vorhersagende Gütebeurteilung von Pivotstrategien, da erst nach Berechnung der Lösung des Gleichungssystems die Fehlerkontrolle einsetzt. Es besteht aber die Möglichkeit, die Abhängigkeit der Lösung von kleinen Störungen der Koeffizientenmatrix vor der Rechnung zu untersuchen. Dabei tritt eine Konstante unabhängig von der Pivotwahl stets als Faktor auf. Dies ist die *skalierte Konditionszahl.*
Sie erlaubt die Bewertung des relativen Fehlers $\|\Delta x\|_\infty / \|x\|_\infty$ der berechneten Lösung unter Anwendung der im Punkt 3.2 (4) angegebenen Beziehung $\|\Delta x\| \leq \|S\|\|x\|$ mit Abschätzung der Störmatrixzeilensummennorm $\|S\|_\infty$, $S = CA - I$, und Einbeziehung der Näherungsinversen C auf der Basis einer Gaußzerlegung LU von A. Eine solche Genauigkeitsbeurteilung hat die allgemeine Gestalt

$$(*) \qquad \frac{\|\Delta x\|}{\|x\|} \leq c \ skal(A) \ \inf\{\varepsilon > 0 : \ |\Delta A| \leq \varepsilon |A|\},$$

wobei c eine positive Konstante,
$skal(A)$ die skalierte Konditionszahl und
$|\,.\,|$ ein noch näher zu definierendes Fehlermaß zur Beurteilung der Genauigkeit der in einer Gleitkommaarithmetik berechneten Gaußzerlegung bedeuten.

Die ***skalierte Konditionszahl*** der regulären Matrix A ist definiert gemäß

$$skal(A) \;=\; \inf_{D\in\mathcal{D}} \; cond(DA)$$

mit $\mathcal{D}$ - Menge der invertierbaren Diagonalmatrizen.

Zunächst soll nur die Frage beantwortet werden, ob für gewisse Matrixnormen die skalierte Konditionszahl durch eine spezielle Diagonalmatrix realisierbar ist. In [21] ist folgender Satz bewiesen.

Satz 3.1 :

Seien $A(n,n)$ eine reguläre Matrix mit den Zeilen $a_1, a_2, ..., a_n$ (das sind Nichtnullvektoren) und mit den Elementen $\delta_i = 1/\|a_i\|_1$ die Diagonalmatrix $D_0 = diag(\delta_1, \delta_2, ..., \delta_n)$ gegeben.

Dann gilt $skal_\infty(A) = cond_\infty(D_0A) = \|A^{-1}D_0^{-1}\|_\infty$.

Beweis:

1.Gemäß Definition gilt stets $skal(A) \le cond(D_0A)$.

2.Das Matrizenprodukt $A' = D_0A$ bedeutet eine Zeilenskalierung von A und liefert, wenn wir mit $a'_1, a'_2, ..., a'_n$ die Zeilen von A' bezeichnen, die Beziehungen $\|a'_i\|_1 = 1,\ i = 1,2,...n,$ und $\|A'\|_\infty = 1$.
Damit ist $cond_\infty(D_0A) = \|D_0A\|_\infty\|(D_0A)^{-1}\|_\infty = \|A^{-1}D_0^{-1}\|_\infty$.

3.Es genügt nun zu zeigen, daß für eine beliebige invertierbare Diagonalmatrix $D = diag(d_1, d_2, ..., d_n)$ die Ungleichung $cond_\infty(DA) \ge cond_\infty(D_0A)$ gilt.
Damit realisiert D_0 das Infimum.

Wegen $cond(c\,A) = cond(A)$ gilt $cond(DA) = cond(D_1A)$, mit $D_1 = D/\|DA\|$ und somit kann o.B.d.A. $\|DA\|_\infty = 1$ vorausgesetzt werden.

Dann ist aber $|d_i|\,\|a_i\|_1 = \|d_ia_i\|_1 \le 1 = |\delta_i|\,\|a_i\|_1$, also $|d_i| \le \delta_i$ für $i = 1,2,...,n$.

Hieraus folgt mit $A^{-1} = (a_{ij}^{(-1)})$

$$\begin{aligned} cond_\infty(DA) \;&=\; \|A^{-1}D^{-1}\|_\infty = \max_{1\le i\le n}\sum_{j=1}^{n}|a_{ij}^{(-1)}|\,|d_j|^{-1} \ge \sum_{j=1}^{n}|a_{ij}^{(-1)}|\,|\delta_j|^{-1} = \\ &=\; \|A^{-1}D_0^{-1}\|_\infty = cond_\infty(D_0A). \end{aligned}$$

□

Der Satz weist gleichzeitig darauf hin, daß die mittels der Norm $\|\,.\,\|_\infty$ definierte Kondition einer regulären Matrix minimal wird, wenn diese durch eine Diagonalmatrix mit Elementen gemäß der Betragssummen ihrer Zeilen skaliert wird.

3.4 Genauigkeitsbeurteilung der Lösung von $Ax=b$

In [29] erfolgte die Abschätzung $(*)$ mit einem komponentenweisen relativen Fehlermaß für $\Delta A = \hat{L}\hat{U} - A$ und A. Dabei ist $\hat{L}\hat{U}$ die in der Computerarithmetik berechnete Gaußzerlegung und $x + \Delta x$ die exakte Lösung von $\hat{L}\hat{U}x' = b$.
Das folgende Beispiel belegt jedoch, daß dieses Fehlermaß den Wert ∞ annehmen kann, während gleichzeitig der relative Fehler $\|\Delta x\|/\|x\|$ klein bleibt. Dazu genügt es, einen solchen Fall zu konstruieren, wo Komponenten von A Null sind und entsprechende Elemente von ΔA nicht verschwinden. Dies zeigt, daß diese komponentenweise Abschätzung grundsätzliche Nachteile für die Genauigkeitsbeurteilung bei computerberechneten Dreieckszerlegungen besitzt.

Beispiel:

Bei gerundeter Gleitpunktarithmetik der Mantissenlänge 2 zur Basis 10 liefert die Matrix

$$A(4,4) = \begin{pmatrix} 1 & 0.6 & 0.4 & 0 \\ 0.99 & 0 & 0 & 0 \\ 0.98 & 0 & 0 & 0.01 \\ 0 & 0 & -0.01 & 0 \end{pmatrix}$$

die Gaußzerlegung

$$\hat{L}\hat{U} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0.99 & 1 & 0 & 0 \\ 0.98 & 1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0.6 & 0.4 & 0 \\ 0 & -0.59 & -0.4 & 0 \\ 0 & 0 & 0.01 & 0.01 \\ 0 & 0 & 0 & 0.01 \end{pmatrix}.$$

Somit ist

$$\Delta A = \hat{L}\hat{U} - A = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0.004 & -0.004 & 0 \\ 0 & -0.002 & 0.002 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

und es gibt keine noch so große reelle Zahl $r > 0$, so daß $|\Delta A| \leq r|A|$.

Diese Unzulänglichkeit der komponentenweisen Abschätzung hat dazu geführt, daß in der Literatur meist eine Fehlerbeurteilung gemäß der Bedingung

$$(**) \qquad \frac{\|\Delta x\|}{\|x\|} \leq c\ skal(A)\ \inf\{\varepsilon > 0 :\ |\Delta A|_\infty \leq \varepsilon |A|_\infty\},$$

wobei mit $|\,.\,|_\infty$ der *maximale Absolutbetrag der Matrixelemente* bezeichnet wird.

Entsprechend könnte man dies auch in der Maximumnorm der Matrix $\|A\|_{max} = n|A|_\infty$ notieren.
Zwar ist die Abschätzung gut zugänglich, aber ihre Verwendung eher heuristisch als sachlich gestützt.

Anknüpfend an Abschnitt 3.2 (4) und Satz 3.1 wird deshalb in [21] mit der Störmatrix $S = CA - I$, $C = (\hat{L}\hat{U})^{-1}$, der Fehlermatrix $\Delta A = C^{-1} - A$, der Zeilensummennorm $\|A\|_\infty$ und der Betragssummennorm $\|x\|_1$ gearbeitet und dort folgende Abschätzung für $\|S\| = \|S\|_\infty$ hergeleitet. Entscheidend ist dabei die Berechnung einer möglichst guten Gaußzerlegung.

Satz 3.2 :

Falls $B = A^{-1}\,\Delta A$ und $b = \|B\| < 1$, gilt die Ungleichungskette

$$b(1-2b) \;\le\; (1-b)\,\|S\| \;\le\; b \;\le\; skal(A)\,\max_{1\le i\le n}\frac{\|\Delta a_i\|_1}{\|a_i\|_1}.$$

Beweis:

1.Wegen $\|B\| < 1$ ist $\|(I+B)^{-1}\| \le 1/(1-\|B\|)$ und

$$\begin{aligned}
|\|S\| - \|B\|| &\le \|S+B\| = \\
&= \|(A+\Delta A)^{-1}A - I + B\| = \|-(A+\Delta A)^{-1}\Delta A + B\| = \\
&= \|(A+\Delta A)^{-1}AA^{-1}\Delta A - B\| = \|(A^{-1}(A+\Delta A))^{-1}B - B\| = \\
&= \|((I+B)^{-1} - I)B\| = \|((I+B)^{-1} - (I+B)(I+B)^{-1})B\| = \\
&= \|(B(I+B)^{-1}B\| \le \frac{\|B\|^2}{1-\|B\|}.
\end{aligned}$$

Somit gilt

$$b\frac{1-2b}{1-b} = b - \frac{b^2}{1-b} \le \|S\| \le b + \frac{b^2}{1-b} = b\frac{1}{1-b}.$$

2.Nun zeigen wir den letzten Teil der Ungleichung.
Mit der Diagonalmatrix $D = diag(d_1, d_2, ..., d_n)$, $d_i = \|a_i\|_1$ gilt gemäß Satz 3.1 $skal(A) = cond(D^{-1}A) = \|A^{-1}D\|$.
Hieraus folgt

$$\|B\| = \|A^{-1}DD^{-1}\Delta A\| \le \|A^{-1}D\|\,\|D^{-1}\Delta A\| = skal(A)\,\max_{1\le i\le n}\frac{\|\Delta a_i\|_1}{\|a_i\|_1}.$$

□

Bemerkungen:

1. Die Schätzung von $\|S\|$ nach unten ist sehr grob, denn für $\|B\| \in (0.5, 1)$ ist die untere Grenze negativ und somit unbrauchbar.
2. Ein geschickterer Weg der Abschätzung von Teil 1 erhält man bei Annahme, daß B invertierbar ist, mittels $C = (A+\Delta A)^{-1}$, $S = CA - I = -C\Delta A$ sowie

$$\begin{aligned}
|\|S\| - \|B\|| &\le \|S+B\| = \\
&= \|C\Delta A - B\| = \|(C\Delta AB^{-1} - I)B\| = \\
&= \|(CA-I)B\| = \|SB\| \le \|S\|\,\|B\|.
\end{aligned}$$

Damit erhalten wir mit $s = \|S\|$ die Beziehung $|s - b| \leq sb$ und wie oben die Ungleichung $s \leq b/(1-b)$, aber von unten die neue Bedingung $s \geq b/(1+b)$, die eine bessere Grenze als $b(1-2b)/(1-b)$ darstellt.

3. Die Abschätzung zeigt, daß sich die Störmatrixnorm bis auf Glieder höherer Ordnung in $\|A^{-1}\,\Delta A\|$ auf die Berechnung von $\|A^{-1}\,\Delta A\|$ reduzieren läßt.
   Insbesondere liegt für $\|B\| \ll 1$ eine sehr gute Einschließung von $\|S\|$ vor.

4. Da die Norm $\|A^{-1}\,\Delta A\|$ noch nicht allzu praktikabel ist, wenden wir uns der Größe $skal(A) \max\limits_i \|\Delta a_i\|_1/\|a_i\|_1$ zu und versuchen, mehr über ihre Güte zu finden.

   Wenn die Matrix $Z = (z_{ij})$ ist, sei $|Z| = (|z_{ij}|)$.

   Weiter sei $A^{-1} = (a_{ij}^{(-1)})$, $\Delta A = (\Delta a_{ij})$, $D_1 = diag(\|\Delta a_1\|_1, \|\Delta a_2\|_1, ..., \|\Delta a_n\|_1)$.

   Dann kann man leicht nachrechnen die Gültigkeit der Beziehungen

$$\begin{aligned}
\|A^{-1}\,\Delta A\| &\leq \||A^{-1}|\,|\Delta A|\| = \|A^{-1}D_1\|, \\
\|A^{-1}D_1\| &= \max_{1\leq i\leq n} \sum_{j=1}^{n} |a_{ij}^{(-1)}|\,\|\Delta a_j\|_1 \;=\; \sum_{j=1}^{n} |a_{kj}^{(-1)}|\,\|\Delta a_j\|_1 = \\
&= \|\,(|a_{k1}^{(-1)}|\,\|\Delta a_1\|_1, |a_{k1}^{(-1)}|\,\|\Delta a_1\|, ..., |a_{kn}^{(-1)}|\,\|\Delta a_n\|_1)\,\|_1 = \\
&= \|a_k^{(-1)}D_1\| \quad \text{(die Zeile } a_k^{(-1)} \text{ realisiert } \|A^{-1}D_1\|), \\
\|A^{-1}\,\overline{\Delta A}\| &= \|A^{-1}D_1\| \quad \text{mit} \quad \overline{\Delta A} = (sign(a_{kj}^{(-1)}) \cdot \Delta a_{ij}).
\end{aligned}$$

   Das heißt, die Vorzeichensituation in ΔA kann so sein, daß die Gleichheit eintritt. Aber für

$$\lambda = \min_{1\leq i\leq n} \frac{\|\Delta a_i\|_1}{\|a_i\|_1} \quad \text{gilt elementweise} \quad \lambda \cdot D \leq D_1$$

   und daher

$$skal(A) \min_{1\leq i\leq n} \frac{\|\Delta a_i\|_1}{\|a_i\|_1} = \|A^{-1}D\|\,\lambda = \|A^{-1}\lambda D\| \leq \|A^{-1}D_1\|.$$

   Die Quotienten $\|\Delta a_i\|_1/\|a_i\|_1$ bestimmen somit neben $skal(A)$ wesentlich die Norm $\|A^{-1}\,\Delta A\|$ und damit die Störmatrixnorm $\|S\|$.

5. Ziel einer klugen Pivotstrategie bei der Gaußelimination, aber auch bei der LU-Zerlegung, sollte die Minimierung der Quotienten $\|\Delta a_i\|_1/\|a_i\|_1$ sein.

   In [21] sind die wesentlichen Einflußgrößen dafür beschrieben:

   a) Eine gute Pivotstrategie sollte das Längenwachstum der Zeilenvektoren von A während des Prozesses möglichst gering halten;
   b) Maximalpivots sollten nach Zeilennormierung ausgesucht werden.

   Die praktische Ausführung beinhaltet die Auswahl der Pivots nach impliziter Zeilenskalierung der Matrix (Division der Zeilen durch ihre Betragssummennorm) und Pivotstrategien angepaßt auf die jeweilige Problemklasse.

4 Skalierung

Theoretisch ist eine *Skalierung (Äquilibrierung)* der Matrix A durch eine *Äquivalenztransformation* $\hat{A} = D_1 A D_2$ mit zwei regulären Matrizen D_1, D_2 erreichbar. Zumeist wird das Problem insofern vereinfacht, daß die Transformationsmatrizen D_i diagonal sind oder zusätzlich noch eine von beiden die Einheitsmatrix ist.
Weiterhin soll die Skalierung die Kondition der Matrix verbessern. Da man im allgemeinen bei DA nur $cond(DA) \leq cond(D)cond(A)$ hat, aber wegen $cond(D) \geq 1$ mit $cond(DA) \geq cond(A)$ rechnen muß, wird sich die Kondition beim Übergang von A zu DA in der Regel verschlechtern. Deshalb ist man an Skalierungen (Transformationen) interessiert, unter denen die Kondition nicht schlechter wird.
Dies sind zum Beispiel Orthogonaltransformationen bei einer mittels Frobenius- oder Spektralnorm definierten Kondition. Eine Verbesserung tritt sogar ein, wenn D aus der Menge der Diagonalmatrizen mit speziellen Elementen ist und $cond_\infty(A)$ genommen wird.
Dann ist noch die praktische Seite, ob die Skalierung tatsächlich (explizit) ausgeführt wird oder nicht. Für die Lösung des Gleichungssystems $Ax = b$ heißt dies im ersten Fall die Transformation auf die neue Gestalt $D_1 A D_2 y = \hat{A}y = d = D_1 b$ mit $y = D_2^{-1}x$, wobei durch die *Zeilenskalierung* $A' = D_1 A$ und *Spaltenskalierung* $\hat{A} = A' D_2$ mit Diagonalmatrizen der wesentliche zusätzliche Rechenaufwand von je n^2 Multiplikationen sowie die Berechnung der Skalierungsfaktoren hinzukommt.
Im zweiten Fall der implizite Skalierung wird wie in Abschnitt 2 eine relative Spaltenpivotisierung mit Zeilenvertauschung durchgeführt. Mit **einem** Buchhaltervektor für die nicht äquilibrierte Matrix ist der Aufwand vergleichsweise gering, kann aber durch eine in jedem Teiltableau wiederholte Bestimmung der Skalierungsgrößen auch stark anwachsen.

4.1 Skalierungsvarianten

Im Zusammenhang mit der regulären Matrix $A = (a_{ij})$ und ihren Zeilen a_i sollen folgende Bezeichnungen gelten.
Zeilenbetragssummen

$$z_i = \|a_i\|_1 = \sum_{j=1}^{n} |a_{ij}| \; > \; 0, \quad i = 1(1)n,$$

Spaltenbetragssummen

$$s_j = \sum_{i=1}^{n} |a_{ij}| \; > \; 0, \quad j = 1(1)n,$$

Diagonalmatrizen

$$D_z = diag(z_1^{-1}, z_2^{-1}, ..., z_n^{-1}), \quad D_s = diag(s_1^{-1}, s_2^{-1}, ..., s_n^{-1}).$$

1. ***Einfache Normalisierung der Zeilenelemente***
Vor der Lösung des GS $Ax = b$ transformiere man Matrix und rechte Seite auf $A'x = b'$ gemäß

$$\left.\begin{array}{rcl} m_i & = & \max(|a_{i1}|, |a_{i2}|, ..., |a_{in}|, |b_i|) > 0 \\ a'_{ij} & = & a_{ij}/m_i, \quad j = 1, 2, ..., n \\ b'_i & = & b_i/m_i \end{array}\right\} \quad i = 1, 2, ..., n.$$

2. ***Zehnerpotenzskalierung***

$$D_1 = diag(10^{-k_1}, 10^{-k_2}, ..., 10^{-k_n}), \quad D_2 = diag(10^{-l_1}, 10^{-l_2}, ..., 10^{-l_n}).$$

Ganzzahlige Exponenten $k_i,\ l_j$ so bestimmen, daß

$$\sum_{i,j=1,\ a_{ij}\neq 0}^{n} (lg(|a_{ij}|) - k_i - l_j)^2 \to \min.$$

3. ***Erste Zeilenskalierung***

$$D_1 = D_z, \quad A' = D_1 A = \begin{pmatrix} \frac{a_{11}}{z_1} & \frac{a_{12}}{z_1} & \dots & \frac{a_{1n}}{z_1} \\ \frac{a_{21}}{z_2} & \frac{a_{22}}{z_2} & \dots & \frac{a_{2n}}{z_2} \\ \dots & \dots & \dots & \dots \\ \frac{a_{n1}}{z_n} & \frac{a_{n2}}{z_n} & \dots & \frac{a_{nn}}{z_n} \end{pmatrix}.$$

Damit $|a'_{ij}| \leq 1 \ \ \forall i, j, \quad \|a'_i\|_1 = \sum_{j=1}^{n} |a'_{ij}| = 1, \quad i = 1, 2, ..., n,$

$$\begin{array}{rcl} s'_j & = & \sum_{i=1}^{n} |a'_{ij}|, \quad \sum_{j=1}^{n} s'_j = \sum_{i=1}^{n}\sum_{j=1}^{n} |a'_{ij}| = n, \\ \|D_1\|_\infty & = & \dfrac{1}{\min\limits_i z_i} = \max\limits_i \dfrac{1}{z_i}, \\ \|D_1^{-1}\|_\infty & = & \max\limits_i z_i = \|A\|_\infty, \\ \|A'\|_\infty & = & \max\limits_i \|a'_i\|_1 = 1. \end{array}$$

Zusätzlicher Aufwand bei expliziter Ausführung: n^2 Multiplikationen.
Bei anschließender Lösung des GS mit Pivotstrategie können in Teiltableaus wieder große Elemente entstehen und Zeilenvertauschungen notwendig sein.

Nach Satz 3.1 wissen wir, daß mit einer beliebigen invertierbaren Diagonalmatrix D gilt $cond_\infty(D_1 A) \leq cond_\infty(DA)$.
Trotzdem wollen wir die Abschätzung noch einmal in einer etwas anderen Darstellung durchführen.

Wegen $A = D_1^{-1}A',\ D_1^{-1} = diag(z_1, z_2, ..., z_n),$ gelten

$$\begin{aligned}
\|D_1^{-1}A'\|_\infty &= \max_i \left\{ z_i \sum_{j=1}^{n} |a'_{ij}| \right\} = \max_i\{z_i \cdot 1\} = \max_i\{z_i\} \cdot 1 \\
&= \max_i\{z_i\}\ \max_i\{\|a'_i\|\} = \max_i\{z_i\}\ \|A'\|_\infty, \\
\|(D_1^{-1}A')^{-1}\|_\infty &= \|(A')^{-1}D_1\|_\infty, \quad (A')^{-1} = (\tilde{a}_{ij}) \\
&= \max_i \left\{ \sum_{j=1}^{n} |\tilde{a}_{ij}|\, z_j^{-1} \right\} \\
&\geq \max_i \left\{ \sum_{j=1}^{n} |\tilde{a}_{ij}|\ \min_j\{z_j^{-1}\} \right\} \\
&= \min_j\{z_j^{-1}\}\ \max_i \left\{ \sum_{j=1}^{n} |\tilde{a}_{ij}| \right\} \\
&= \min_j\{z_j^{-1}\}\ \|(A')^{-1}\|_\infty, \\
cond(A) &= cond(D_1^{-1}A') \\
&= \|D_1^{-1}A'\|_\infty\ \|(D_1^{-1}A')^{-1}\|_\infty \\
&\geq \max_i\{z_i\}\ \min_j\{z_j^{-1}\}\ \|A'\|_\infty\ \|(A')^{-1}\|_\infty \\
&= \max_i\{z_i\}\ (\max_j\{z_j\})^{-1}\ \|A'\|_\infty\ \|(A')^{-1}\|_\infty \\
&= cond(A').
\end{aligned}$$

4. ***Zweite Zeilenskalierung mit Vorzeichentest***
Beachtet man noch den Umstand, daß die entstehenden Diagonalelemente nicht negativ sein sollen, ergibt sich die Modifikation

$$D_2 = I,\ D_1 = diag(sign\ a_{ii})\ D_z = diag(\frac{sign\ a_{11}}{z_1}, \frac{sign\ a_{22}}{z_2}, ..., \frac{sign\ a_{nn}}{z_n}),$$

$$A' = D_1 A = \begin{pmatrix}
\frac{a_{11}\ sign\ a_{11}}{\sum_j |a_{1j}|} & \frac{a_{12}\ sign\ a_{11}}{\sum_j |a_{1j}|} & \dots & \frac{a_{1n}\ sign\ a_{11}}{\sum_j |a_{1j}|} \\
\frac{a_{21}\ sign\ a_{22}}{\sum_j |a_{2j}|} & \frac{a_{22}\ sign\ a_{22}}{\sum_j |a_{2j}|} & \dots & \frac{a_{2n}\ sign\ a_{22}}{\sum_j |a_{2j}|} \\
\dots & \dots & \dots & \dots \\
\frac{a_{n1}\ sign\ a_{nn}}{\sum_j |a_{nj}|} & \frac{a_{n2}\ sign\ a_{nn}}{\sum_j |a_{nj}|} & \dots & \frac{a_{nn}\ sign\ a_{nn}}{\sum_j |a_{nj}|}
\end{pmatrix}.$$

Die Einbeziehung des Vorzeichens ist sinnvoll und üblich. Für Norm und Kondition gelten die Aussagen des vorherigen Punktes.
Durch Zeilentausch kann zusätzlich $a'_{ii} > 0$ erreicht werden.

5. ***Dritte Zeilenskalierung***

Im Vergleich zur ersten Zeilenskalierung wird die Normierung modifiziert.

$$D_1 = \|A\|_\infty D_z = diag(d_i), \quad d_i = \frac{\|A\|_\infty}{\|a_i\|_1} = \frac{\|A\|_\infty}{z_i}, \quad \min_i d_i = 1,$$

$$A' = D_1 A = \|A\|_\infty \begin{pmatrix} \frac{a_{11}}{z_1} & \frac{a_{12}}{z_1} & \dots & \frac{a_{1n}}{z_1} \\ \frac{a_{21}}{z_2} & \frac{a_{22}}{z_2} & \dots & \frac{a_{2n}}{z_2} \\ \dots & \dots & \dots & \dots \\ \frac{a_{n1}}{z_n} & \frac{a_{n2}}{z_n} & \dots & \frac{a_{nn}}{z_n} \end{pmatrix}.$$

Damit ändert sich die Zeilensummennorm von A' im Vergleich zu A nicht und es gelten

$$\begin{aligned} \|A'\|_\infty &= \|A\|_\infty = \max_i z_i, \\ \|D_1\|_\infty &= \max_i d_i = \frac{\|A\|_\infty}{\min_i z_i} = \frac{\max_i z_i}{\min_i z_i} \geq 1, \\ \|D_1^{-1}\|_\infty &= \frac{\max_i z_i}{\|A\|_\infty} = 1. \end{aligned}$$

Analog lassen sich die Abschätzungen für die Kondition zeigen.

$$\begin{aligned} \|(A')^{-1}\|_\infty = \|A^{-1} D_1^{-1}\|_\infty &\leq \|A^{-1}\|_\infty \|D_1^{-1}\|_\infty = \|A^{-1}\|_\infty, \\ \|A^{-1}\|_\infty = \|(A')^{-1} D_1\|_\infty &\leq \|(A')^{-1}\|_\infty \|D_1\|_\infty = \\ &= \|(A')^{-1}\|_\infty \max_i d_i, \end{aligned}$$

$$cond_\infty(A') = \|A'\|_\infty \|(A')^{-1}\|_\infty \quad \begin{cases} \leq \|A\|_\infty \|A^{-1}\|_\infty = cond_\infty(A), \\ \geq \frac{\|A\|_\infty \|A^{-1}\|_\infty}{\max_i d_i} = \frac{cond_\infty(A)}{\max_i d_i}. \end{cases}$$

Entscheidend ist also das Verhältnis der Skalierungsfaktoren d_i zueinander. Damit noch einmal die Bestätigung.

Unter allen durch Zeilenskalierung aus einer Matrix hervorgehenden Matrizen hat jede zeilenäquilibrierte die kleinste Kondition in der Zeilensummennorm.

In Bezug auf Schranken wie in 3.2(5,6), die mit $cond_\infty(A)$ arbeiten, ist also eine Zeilenäquilibrierung der Matrix (des GS) optimal, deshalb auch Anwendung der skalierten Konditionszahl.

6. ***Jacobi-Normalisierung (Zeilenskalierung)***

$D = diag(d_i)$, $d_i = a_{ii} \neq 0$ (Voraussetzung), $D_2 = I$, $D_1 = D^{-1}$,

$$A' = D^{-1}A = \begin{pmatrix} 1 & \frac{a_{12}}{a_{11}} & \cdots & \frac{a_{1n}}{a_{11}} \\ \frac{a_{21}}{a_{22}} & 1 & \cdots & \frac{a_{2n}}{a_{22}} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{a_{n1}}{a_{nn}} & \frac{a_{n2}}{a_{nn}} & \cdots & 1 \end{pmatrix}, \quad a'_{ii} = 1 \ \ \forall i.$$

Anwendung von A' beim Gesamtschrittverfahren(GSV, Jacobi),

$$\begin{aligned} D^{-1}Ax &= D^{-1}b, \ \ A = D - B, \ B = D - A \\ D^{-1}(D-B)x &= D^{-1}b \\ x &= D^{-1}(Bx + b) \\ x &= x - D^{-1}(Ax - b), \ \ J = I - D^{-1}A = I - A' \\ x^{(m+1)} &= x^{(m)} - D^{-1}(Ax^{(m)} - b). \end{aligned}$$

Die Skalierungsmatrix D^{-1} tritt als einfacher Vorkonditionierer im Iterationsverfahren auf.
Das heißt aber auch, wenn man zuerst das GS explizit skaliert und dann für $A'x = b'$ das GSV

$$x^{(m+1)} = x^{(m)} - (D')^{-1}(A'x^{(m)} - b')$$

durchführt, dieses wegen $D' = I$ die gleiche Iterationsmatrix $J' = J = I - D^{-1}A$ enthält. Damit ändert sich prinzipiell nichts am Konvergenzverhalten des Verfahrens.

7. ***Zeilen- und Spaltenskalierung, Äquilibrierung***

Sei $A = A^T > 0$, damit $a_{ii} > 0$.

$$D_1 = D_2 = D_z^{1/2} = diag(\frac{1}{\sqrt{z_i}}) = diag(\frac{1}{\sqrt{a_{ii}}}), \ A' = D_1 A D_1 = A'^T > 0,$$

$$A' = \begin{pmatrix} 1 & \frac{a_{12}}{\sqrt{a_{11}a_{22}}} & \frac{a_{13}}{\sqrt{a_{11}a_{33}}} & \cdots & \frac{a_{1n}}{\sqrt{a_{11}a_{nn}}} \\ \frac{a_{21}}{\sqrt{a_{22}a_{11}}} & 1 & \frac{a_{23}}{\sqrt{a_{22}a_{33}}} & \cdots & \frac{a_{2n}}{\sqrt{a_{22}a_{nn}}} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{a_{n1}}{\sqrt{a_{nn}a_{11}}} & \frac{a_{n2}}{\sqrt{a_{nn}a_{22}}} & \frac{a_{n3}}{\sqrt{a_{nn}a_{33}}} & \cdots & 1 \end{pmatrix}.$$

Nachweis von $|a'_{ij}| < 1 \ \ \forall i \neq j$ erfolgt mittels spezieller Vektorpaare

$$\begin{aligned} x^T &: (1,1,0,0,...,0), \ \ (1,-1,0,0,...,0), \\ x^T &: (1,0,1,0,...,0), \ \ (1,0,-1,0,...,0) \text{ usw.} \end{aligned}$$

Damit ist $0 \leq \sum_{j=1}^{n} a'^{\,2}_{ij} \leq \gamma_i$ = Anzahl der Nichtnullelemente (NNE) in der i-ten Zeile.

Diese Äquilibrierung verändert nicht das für die Konvergenz entscheidende Spektrum der Iterationsmatrix bei den Verfahren JOR (extrapoliertes Jacobi-Verfahren) und SOR (Überrelaxationsverfahren). Der Nachweis für JOR ist wie folgt.

$$x^{(m+1)} = J_\omega x^{(m)} + c, \quad J_\omega = I - \omega D^{-1} A.$$

Nach Skalierung erhalten wir die Iterationsmatrix

$$\begin{aligned} J'_\omega &= I - \omega (D')^{-1} A', \quad A' = D_1 A D_1 \\ J'_\omega &= I - \omega (D_1 D D_1)^{-1} D_1 A D_1 \\ &= D_1^{-1} D_1 - \omega D_1^{-1} D^{-1} (D_1^{-1} D_1) A D_1 \\ &= D_1^{-1} (I - \omega D^{-1} A) D_1 \\ &= D_1^{-1} J_\omega D_1 \end{aligned}$$

und damit die Ähnlichkeit von J'_ω und J_ω.

8. ***Äquilibrierung mit noch zu bestimmender Diagonalmatrix***
Sei $A = A^T$.

$$A' = D_1 A D_2, \; D_1 = D_2 = diag(d_i), \; d_i > 0,$$

$$A' = \begin{pmatrix} d_1 & & & \\ & d_2 & & \\ & & ... & \\ & & & d_n \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} d_1 & & & \\ & d_2 & & \\ & & ... & \\ & & & d_n \end{pmatrix}.$$

Ziel: Euklidische Norm von allen Zeilen von A' ist 1 (genauso Spalten).

$$\sqrt{\sum_{j=1}^{n} a'^{\,2}_{ij}} = 1, \quad i = 1, 2, ..., n.$$

Wegen $a_{ij} = a_{ji}$ entstehen die identischen Gleichungsysteme

$$d_i \sqrt{\sum_{j=1}^{n} (a_{ij} d_j)^2} = 1, \quad i = 1, 2, ..., n,$$

$$\sqrt{\sum_{i=1}^{n} (d_i a_{ij})^2} \; d_j = 1, \quad j = 1, 2, ..., n.$$

Die Idee ist gut. Die Kondition des neuen GS wird nicht schlechter.
Aber für die Bestimmung der Größen $d_1, d_2, ..., d_n$ ist ein nichtlineares GS zu lösen (Aufwand!).
Die obige Zeilennorm ist nicht zu verwechseln mit der *Frobeniusnorm* der Matrix

$$\|A'\|_F = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} a'^{\,2}_{ij}} = \sqrt{n}.$$

9. ***"Optimale" Skalierung***

Sei $A = A^T$. Wir bilden die Matrix

$$A' = H^{-1}A(H^{-1})^T \quad \text{mit} \quad H \text{ regulär, aber einfache Gestalt}$$

Ziel : $cond(A') = 1 \quad bzw. \quad cond(A') \approx 1.$
Problem : Wahl von H.

(a) Weg über Eigenwerte von A
$A = A^T$ ist diagonalisierbar mittels Ähnlichkeitstransformation

$$A' = H^{-1}AH = \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ldots & \\ & & & \lambda_n \end{pmatrix}$$

System der orthonormalen EV $x^{(i)}$ bilden Spalten von H.

$$H = (x^{(1)}, x^{(2)}, ..., x^{(n)}), \quad H^{-1} = H^T = \begin{pmatrix} x^{(1)T} \\ x^{(2)T} \\ \ldots \\ x^{(n)T} \end{pmatrix}.$$

Aber Lösung eines EWP ist aufwendig!

(b) Andere Wahl von H
Es gilt

$$\begin{aligned} A' &= H^{-1}A(H^{-1})^T, \\ HA'H^{-1} &= A(HH^T)^{-1} = B. \end{aligned}$$

A' und $A(HH^T)^{-1}$ sind ähnliche Matrizen und haben damit die gleichen EW. Man wähle nun $(HH^T)^{-1}$ so, daß $A(HH^T)^{-1} = I$.
Damit wäre A' ähnlich zur Einheitsmatrix und $cond(A') = 1$. Was heißt aber $A(HH^T)^{-1} = I$? Das heißt $HH^T = A$, oder finde zumindest eine solche Matrix H, daß $HH^T \approx A$.
Damit ist man bei sogenannten Aufspaltungen (Zerlegungen) von A, ev. auch nur näherungsweise.
Für das GS bedeutet die Skalierung eine Umformung

$$\begin{aligned} Ax &= b \\ H^{-1}A(H^{-1})^T\hat{x} &= H^{-1}b \\ A'\hat{x} &= b' \quad \text{mit} \quad \hat{x} = H^T x\ , \quad b' = H^{-1}b. \end{aligned}$$

Der Ansatz erfordert also auch eine Transformation der gesuchten Lösung gemäß $x = (H^{-1})^T\hat{x}$.

Auswahlvarianten von H

- $A = A^T > 0$

$$H = H^T = D^{1/2} = diag(\sqrt{a_{ii}}), \quad \text{siehe Punkt 7}$$

- $A = D - E - E^T, \quad D > 0$

$$H = D^{1/2} - \omega E^T \quad \text{(Axelsson)}$$

$\omega > 0$ so wählen, daß (spektrale) Ähnlichkeit erreicht wird

$$HH^T = D - \omega(D^{1/2}E + E^T D^{1/2}) + \omega^2 E^T E \approx A$$

- $A = A^T > 0$

$H = C, \quad CC^T$ genäherte Cholesky-Zerlegung von A

$$HH^T = CC^T \approx A$$

Damit ist man eigentlich schon bei der Vorkonditionierung, die aber i.a. erst bei Iterationsverfahren richtig zum Einsatz und Tragen kommt.

10. ***Skalierungsprozedur***

Abschließend ein Algorithmus zum Ausbalancieren einer Matrix.
Es ist die Pascal-Version der Algol-Prozedur 'balance' aus [17], die zur Konditionsverbesserung der Matrix genutzt werden kann.

```
procedure Balance_Wilkinson_Reinsch
         (n:integer;               {Dimension der Matrix}
          var a:matrix;            {Matrix A(n,n)}
          var ip:vektor1;          {Permutationsvektor fuer
                                    Zeilentausch}
          var lowm,highm:integer); {untere und obere Grenze
                                    fuer Zeilen/Spaltentausch}
const Basis=2;     {Basis der Zahlendarstellung im Rechner}
label 1,2,3,4;
var   i,j,low,high:integer;
      c,r,s,f,Basis2:float;
      Ende:boolean;
procedure Tausch(x,y:float);
var h:float;
begin
 h:=x; x:=y; y:=h;
end;
procedure TauschVektor(j,m:integer);
var i:integer;
begin
 if j<>m then
  begin
   Tausch(ip[j],ip[m]);
   for i:=1 to high do Tausch(a[i,j],a[i,m]);
   for i:=low to n  do Tausch(a[j,i],a[m,i]);
  end;
```

```
end;
begin
 Basis2:=sqr(Basis);
 low:=1; high:=n;
 for i:=1 to n do ip[i]:=i;
 {Suchen von Grenzen fuer Zeilen/Spaltentausch,
  Zeilen mit Anfangsnullen}
1:for j:=high downto 1 do
   begin
    for i:=1 to high do if i<>j then if a[j,i]<>0 then goto 2;
    TauschVektor(j,high);
    Dec(high);
    goto 1;
   2: ;
   end;
3:for j:=low to high do
   begin
    for i:=low to high do if i<>j then if a[i,j]<>0 then goto 4;
    TauschVektor(j,low);
    Inc(low);
    goto 3;
   4: ;
   end;
 {Festlegung der Skalierung fuer Zeilen/Spalten
  in den berechneten Grenzen}
 repeat
  Ende:=true;
  for i:=low to high do
   begin
    c:=0; r:=0;
    for j:=low to high do if i<>j then
     begin
      c:=c+abs(a[j,i]);
      r:=r+abs(a[i,j]);
     end;
    f:=1; s:=c+r;
    while c*Basis<r  do begin f:=f*Basis; c:=c*Basis2; end;
    while c>=Basis*r do begin f:=f/Basis; c:=c/Basis2; end;
    if (c+r)/f<0.95*s then
     begin
      Ende:=false;
      for j:=low to n  do a[i,j]:=a[i,j]/f;  {Zeile  i durch f}
      for j:=1 to high do a[j,i]:=a[j,i]*f;  {Spalte i mal   f}
     end;
   end;
 until Ende;
 lowm:=low;
 highm:=high;
end;
```

Die Größen f, durch welche die i-te Zeile dividiert bzw. mit der die i-te Spalte multipliziert werden, sind in der Prozedur keine Ergebnisparameter. Sie werden jedoch gebraucht, wenn es um die Lösung eines GS geht, da die Spaltenmultiplikation proportional die Unbekannte x_i verändert.
Diese Balancierung manipuliert insbesondere die in einem "Nichtnullband" am Rande liegenden betragsgroßen Matrixelemente. Folgende Matrixbeispiele sind so gewählt, daß kein Zeilentausch durchgeführt wird.

Beispiele:

$$A_1 = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 10 & 0 \\ 0 & 1 & 3 \end{pmatrix}, \quad \text{keine Veränderung}$$

$$A_2 = \begin{pmatrix} 1 & 2 & 3 \\ 10 & 0 & 1 \\ 0 & 1 & 3 \end{pmatrix}, \qquad A_2' = \begin{pmatrix} 1 & 4 & 3 \\ 5 & 0 & 1/2 \\ 0 & 2 & 3 \end{pmatrix}$$

$$A_3 = \begin{pmatrix} 1 & 2 & 3 \\ 10 & 0 & 1 \\ 0 & 0 & 3 \end{pmatrix}, \qquad A_3' = \begin{pmatrix} 1 & 4 & 6 \\ 5 & 0 & 1 \\ 0 & 0 & 3 \end{pmatrix}$$

$$A_4 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 10 & 0 & 1 & 2 \\ 0 & 9 & 3 & 1 \\ 0 & 0 & 1 & 2 \end{pmatrix}, \quad A_4' = \begin{pmatrix} 1 & 2 & 3 & 2 \\ 10 & 0 & 1 & 1 \\ 0 & 9 & 3 & 1/2 \\ 0 & 0 & 2 & 2 \end{pmatrix}$$

$$A_5 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 10 & 0 & 1 & 2 \\ 0 & 0 & 3 & 1 \\ 0 & 0 & 1 & 2 \end{pmatrix}, \quad A_5' = \begin{pmatrix} 1 & 4 & 3/4 & 1 \\ 5 & 0 & 1/8 & 1/4 \\ 0 & 0 & 3 & 1 \\ 0 & 0 & 1 & 2 \end{pmatrix}$$

A	Konditionszahlen					
	$cond_\infty(A)$	$skal_\infty(A)$	$cond_2(A)$	$hcond(A)$	$fcond(A)$	$ccond(A)$
A_1	21	11	14.8	0.25	1.6	13.3
A_2	22	10.6	15.4	0.26	0.44	18.1
A_2'	8.4	6.7	6.2	0.34	0.37	9.4
A_3	11.4	5	7.5	0.53	0.80	9.4
A_3'	8.6	4.5	5.0	0.54	0.60	9.7
A_4	91	44	60.8	0.067	0.39	97.5
A_4'	68.6	37	49.3	0.069	2.89	64.1
A_5	20.8	9	12.1	0.25	0.67	17.6
A_5'	5.4	2.6	3.9	0.66	0.19	6.3

Tab.1. Verschiedene Konditionszahlen der Matrizen vor (A) und nach Ausbalancierung (A') (bei Kommastellen gerundet).

Man beachte, daß letztere 3 Konditionszahlen nur gewisse Abschätzungen darstellen (siehe $fcond$ für A_4, A_4'). Der positive Effekt der skalierten Konditionszahl, die oft in der Nähe der spektralen Kondition liegt, sowie der Ausbalancierung der Matrix sind deutlich zu erkennen.

4.2 Skalierung und Vorkonditionierung

Dabei geht man, wie in Punkt 4.1.(9b) angedeutet, oft von der Matrixzerlegung aus.

$$\boldsymbol{A = D\text{-}E\text{-}F = D(I\text{-}L\text{-}U)}$$

$\boldsymbol{I}$ = Einheitmatrix
$\boldsymbol{D = diag(A) = diag(a_{11}, a_{22}, \ldots, a_{nn})}$
$\boldsymbol{L = (l_{ij})}$ linke untere Dreiecksmatrix (Diagonale=0), $\boldsymbol{E = DL}$
$\boldsymbol{U = (u_{ij})}$ rechte obere Dreiecksmatrix (Diagonale=0), $\boldsymbol{F = DU}$.

Folgende Varianten können wir unterscheiden.

(1) Spezielle Skalierung als Form der Vorkonditionierung
Dies war Gegenstand der bisherigen Betrachtungen.
An die Skalierung schließt sich zumeist ein direktes Lösungsverfahren an, z.B. ein Eliminationsverfahren.
Bei Iterationsverfahren ist i.a. keine Verbesserung der Spektraleigenschaften der Iterationsmatrix zu erwarten, wenn sich die Skalierungsmatrix auch in Form der Vorkonditionierung zeigt.

(2) Spezielle Skalierung mit A^T
Diese geschieht mittels der Transformation

$$\begin{aligned} A^T Ax &= A^T b, \\ Bx &= c, \quad B = B^T > 0. \end{aligned}$$

Der Vorteil liegt in der Konvergenz des Einzelschrittverfahrens (Gauß-Seidel) für das GS $Bx = c$.
Erkauft wird dies mit dem hohen Aufwand von n^3 Operationen für eine explizite Ausführung der Matrixmultiplikation $A^T A$. Ein weiterer Nachteil ist, daß dabei die Kondition von A quadriert wird und damit sich die Konvergenzrate verschlechtert.

(3) Allgemeine Skalierung
Darunter verstehen wir eine Vorkonditionierung der Form

$$\begin{aligned} W^{-1}Ax &= W^{-1}b, \quad W \; regulär, \quad cond(W^{-1}A) \approx 1, \\ W &= HH^T, \; H \text{ linke untere Dreiecksmatrix (Diagonale} \neq 0). \end{aligned}$$

Den Aufwand kann man grob umreißen bei:

Direkten Verfahren
- notwendig : Lösung des GS,
- zusätzlich : Matrix * Matrix (n^3 Operationen).

Iterativen Verfahren
- notwendig : Matrix * Vektor (Ax),
- zusätzlich : Lösen von speziellen GS mit Koeffizientenmatrix in Dreiecksgestalt, z.B. $Hz = d$.

Insofern wird noch einmal die Bedeutung der allgemeinen Skalierung als Vorkonditionierung bei Iterationsververfahren unterstrichen.

Literatur

[1] KIELBASINSKI, A.; SCHWETLICK, H.: *Numerische lineare Algebra.* Mathematik für Naturwissenschaft und Technik Band 18, DVW, Berlin 1988.

[2] HACKBUSCH, W.: *Iterative Lösung großer schwach besetzter Gleichungssysteme.* Leitfäden der angewandten Mathematik und Mechanik Band 69. B.G. Teubner Stuttgart 1991.

[3] MAESS, G.: *Vorlesungen über numerische Mathematik.* Band 1, 2. Akademie-Verlag Berlin 1984, 1988.

[4] SCHWARZ, H.R.: 1.*Methode der finiten Elemente.* Leitfäden der angewandten Mathematik und Mechanik Band 47. B.G. Teubner Stuttgart 1991.

[5] ZLATEV, Z.: *Computational Methods for General Sparse Matrices.* Math. and Its Appl. Vol.65. Kluwer Academic Publishers London 1991.

[6] GUSTAVSON, F.: *A Survey of Some Sparse Matrix Theory and Techniques.* Jahrbuch Überblicke Mathematik. B.I.-Wissenschaftsverlag Mannheim 1981.

[7] BRUASET, A.M.: *A survey of preconditioned iterative methods.* Pitman Research Notes in Mathematics Series 328. Longman Scientific & Technical Essex, John Wiley & Sons, Inc., New York 1995.

[8] SCHWETLICK, H.; KRETZSCHMAR, H.: *Numerische Verfahren für Naturwissenschaftler und Ingenieure.* Fachbuchverlag Leipzig Köln 1991.

[9] SCHABACK, R; WERNER, H.: *Numerische Mathematik.* Springer-Verlag Berlin 1993.

[10] ÜBERHUBER, C.: *Computer-Numerik 1,2.* Springer-Verlag Berlin 1995.

[11] DEUFLHARD, P.; HOHMANN, A.: *Numerische Mathematik.* De Gruyter-Verlag Berlin New York 1991.

[12] RALSTON, A.: *A First Course in Numerical Analysis.* McGraw-Hill New York 1965.

[13] CHERKASOVA, M.P.: *Collected Problems in Numerical Methods.* Akademie-Verlag Berlin 1972.

[14] SCHWARZ, H.R.; RUTISHAUSER, H.; STIEFEL, E.: *Numerik symmetrischer Matrizen.* Leitfäden der angewandten Mathematik, Bd. 11. Stuttgart 1968, B.G. Teubner VG Leipzig 1969.

[15] JANKOWSKA, J.; JANKOWSKI, M.: *Przeglad metod i algorytmow numerycznych.* Band 1. WNT Warszawa 1981.

[16] HÄMMERLIN, G.; HOFFMANN, K.-H.: *Numerische Mathematik.* Grundwissen Mathematik 7. Springer-Verlag Berlin 1991.

[17] WILKINSON, J.H.; REINSCH, C.: *Linear Algebra.* Handbook for automatic computation, Vol. II. Grundlehren der mathematischen Wissenschaften in Einzeldarstellungen, Bd. 186. Berlin-Heidelberg-New York 1971.

[18] STOER, J.: *Numerische Mathematik I.* 5. Aufl. Springer-Verlag Berlin 1989.

[19] STOER, J.; BURLISCH, R.: *Einführung in die Numerische Mathematik II.* 3.Aufl. Springer-Verlag Berlin 1990.

[20] AXELSSON, O.: *Iterative Solution Methods.* Cambridge University Press 1994.

[21] DONNER, K.: *Skalierung von Matrizen und numerische Stabilität der Gauß-Elimination.* Preprint Universität Passau, MIP-9514 September 1995.

[22] KÖCKLER, N.: *Numerische Algorithmen in Softwaressystemen : unter besonderer Berücksichtigung der NAG-Bibliothek.* B.G. Teubner Stuttgart 1990.

[23] RICE, J.R.: *Numerical Methods, Software and Analysis.* 2nd Edition. Academic Press Inc. Boston 1993.

[24] ENGELN-MÜLLGES, G.; REUTTER, F.:
1.*Formelsammlung zur Numerischen Mathematik mit FORTRAN 77-Programmen.* Bibliogr. Institut Mannheim 1988.
2.*Formelsammlung zur Numerischen Mathematik mit Turbo Pascal-Programmen.* BI-Wissenschaftsverlag Mannheim 1991.

[25] ENGELN-MÜLLGES, G.; REUTTER, F.: *Numerik-Algorithmen mit ANSI C-Programmen.* (auch für Turbo Pascal, FORTRAN). BI-Wissenschaftsverlag Mannheim 1993.

[26] KOSE, K.; SCHRÖDER, R.; WIELICZEK, K.: *Numerik sehen und verstehen.* Ein kombiniertes Lehr- und Arbeitsbuch mit Visualisierungssoftware. Vieweg Braunschweig 1992.

[27] ZURMÜHL, R.; FALK, S.: *Matrizen und ihre Anwendungen.* Teil 2, Numerische Mehtoden. Springer-Verlag Berlin 1984.

[28] SCHWARZ, H.R.: *Numerische Mathematik.* B.G.Teubner Stuttgart 1988.

[29] BAUER, F.L.: *Optimally scaled matrices.* Numer. Mathematik 5(1963)73-87.

[30] DIETEL, J.: *Formelsammlung zu Numerischen Mathematik mit Turbo Pascal-Programmen* (TPNUM). Rechenzentrum der RWTH Aachen 1993.

Anschrift:

Dr. Werner Neundorf
Technische Universität Ilmenau, Institut für Mathematik
PF 10 0565
D - 98684 Ilmenau

e-mail : neundorf@mathematik.tu-ilmenau.de