

Computing 6, 28—34 (1970)

© by Springer-Verlag 1970

**ALGOL-60 Algorithmen zur Auflösung linearer
Gleichungssysteme mit Fehlererfassung**

Von

G. Alefeld und J. Herzberger, Karlsruhe

(Eingegangen am 23. Juli 1969)

Zusammenfassung — Summary

ALGOL-60 Algorithmen zur Auflösung linearer Gleichungssysteme mit Fehlererfassung. In der vorliegenden Arbeit werden zwei Algorithmen zur Behandlung von linearen Gleichungssystemen mit Intervallkoeffizienten in ALGOL-60 angegeben. Damit ist es möglich auf Rechenanlagen mit ALGOL-60 Compilern solche Aufgaben unter Einschluß sämtlicher Rundungsfehler durchzuführen. Zum Verständnis der angegebenen Prozeduren ist einleitend eine knapp gehaltene Darstellung der programmierten Verfahren vorangestellt.

ALGOL-60 Algorithms for Computation of Systems of Linear Equations with Error Bounds. Two ALGOL-60 algorithms are presented to solve systems of linear equations whose coefficients are known to vary in given intervals. So it is possible to handle with such systems on an arbitrary computer possessing an ALGOL-60 compiler. Simultaneously all rounding errors are included. For understanding the programs a brief introduction is given.

1. Theoretische Grundlagen

Gleichungssysteme mit Intervallkoeffizienten treten auf, wenn die Ausgangsdaten eines Problems z. B. durch Meßungenauigkeiten nur in Intervalle eingeschlossen werden können. Darüber hinaus liegt dieselbe Aufgabestellung im allgemeinen auch vor, wenn man exakt gegebene Daten unter Erfassung der Rundungsfehler auf einer Rechenanlage darstellen will. Die Bestimmung von Einschließungsmengen für die Lösungen solcher Gleichungssysteme kann vorteilhaft mit Hilfe der Intervallarithmetik vorgenommen werden.

Es bezeichne zunächst $I(\mathbb{R})$ die Menge der reellen, abgeschlossenen und beschränkten Intervalle. Für Elemente $[x^1, x^2]$, $[y^1, y^2]$ aus $I(\mathbb{R})$ sind durch:

$$[x^1, x^2] * [y^1, y^2] := \{x * y : x^1 \leq x \leq x^2, y^1 \leq y \leq y^2\}, * \in \{+, -, \cdot, / \}$$

vier Verknüpfungen definiert, welche sich als Verknüpfungen der Intervallgrenzen einfach darstellen lassen. Matrizen und Vektoren, deren Elemente bzw. Komponenten jetzt Intervalle sind, bezeichnen wir mit $\mathfrak{A} = (A_{ij})$, $\mathfrak{B} = (B_{ij})$, ... bzw. $\mathfrak{a} = (A_i)$, $\mathfrak{b} = (B_i)$, ...; sind die Elemente dagegen reelle Zahlen dann wählen wir die Bezeichnungen $\mathfrak{A} = (a_{ij})$, $\mathfrak{B} = (b_{ij})$, ...

bzw. $\mathfrak{a} = (a_i)$, $\mathfrak{b} = (b_i)$, ... Verknüpfungen zwischen Intervallmatrizen sind wie üblich definiert z. B.

$$\mathfrak{A} \cdot \mathfrak{a} := \sum_j A_{ij} \cdot A_j \text{ usw.}$$

Da die Menge $I(\mathbb{R})$ mit den so eingeführten Verknüpfungen keinen Körper bildet führt z. B. die Anwendung des GAUSSSchen Algorithmus in fast allen Fällen zu praktisch wertlosen Ergebnissen. Die Auflösung eines Gleichungssystems mit Intervallkoeffizienten ist bei Verwendung der Intervallarithmetik eine nichtlineare Aufgabe.

Gegeben sei zunächst ein System der Gestalt:

$$\mathfrak{x} = \mathfrak{B} \cdot \mathfrak{x} + \mathfrak{b} \quad (1)$$

mit einer Intervallmatrix $\mathfrak{B} = (B_{ij})$ und einem Intervallvektor $\mathfrak{b} = (b_i)$. Kann man einen Vektor \mathfrak{x}^* so finden, daß er die Gleichung (1) erfüllt, dann gilt für eine Lösung \mathfrak{y}^* jedes Gleichungssystems der Art:

$$\mathfrak{y} = \mathfrak{B} \cdot \mathfrak{y} + \mathfrak{b} \quad (2)$$

mit $b_{ij} \in B_{ij}$ und $b_i \in B_i$ (oder kurz: $\mathfrak{B} \in \mathfrak{B}$ und $\mathfrak{b} \in \mathfrak{b}$) die Aussage:

$$\mathfrak{y}^* \in \mathfrak{x}^*.$$

Das bedeutet nun praktisch, daß \mathfrak{x}^* die Menge der Lösungen von der Schar der aus (1) zu bildenden linearen Gleichungssysteme der Gestalt (2) enthält. Zur Bestimmung des Vektors \mathfrak{x}^* sind verschiedene Iterationsverfahren betrachtet worden:

a) das Gesamtschrittverfahren: $\mathfrak{x}_{m+1} = \mathfrak{B} \cdot \mathfrak{x}_m + \mathfrak{b}$,

b) das Einzelschrittverfahren: $\mathfrak{x}_{m+1} = \mathfrak{L} \cdot \mathfrak{x}_{m+1} + (\mathfrak{D} + \mathfrak{R}) \cdot \mathfrak{x}_m + \mathfrak{b}$

(wobei $\mathfrak{B} = \mathfrak{L} + \mathfrak{D} + \mathfrak{R}$ die Zerlegung von \mathfrak{B} in eine strenge untere Dreiecksmatrix \mathfrak{L} , eine Diagonalmatrix \mathfrak{D} und eine strenge obere Dreiecksmatrix \mathfrak{R} ist).

Definiert man für ein Intervall $A = [a^1, a^2]$ aus $I(\mathbb{R})$ den Betrag $|A|$ zu $|A| := \max\{|a^1|, |a^2|\}$, dann gilt folgende Aussage:

Gesamt- bzw. Einzelschrittverfahren konvergieren für beliebige Anfangsintervallvektoren genau dann gegen das eindeutige Grenzelement \mathfrak{x}^* , wenn die Spektralradien $\rho((|B_{ij}|))$ bzw. $\rho((|\delta_{ij} - |L_{ij}||^{-1} \cdot (|D_{ij}| + |R_{ij}|))$ kleiner als 1 sind.

Nach bekannten Sätzen aus der Theorie der nichtnegativen Matrizen sind deshalb entweder beide Verfahren konvergent oder divergent. Im weiteren werde nun stets die Konvergenz vorausgesetzt.

Wählt man als Ausgangsvektor der Iteration einen Intervallvektor \mathfrak{x}_0 mit:

$$\mathfrak{x}^* \subset \mathfrak{x}_0 \quad (3)$$

(wobei die Inklusion in jeder Komponente gelten soll) dann gilt für alle Iterierten \mathfrak{x}_m bei beiden Verfahren:

$$\mathfrak{x}^* \subset \mathfrak{x}_m, \text{ für } m \geq 0.$$

Bezeichnen unter Voraussetzung (3) \tilde{x}_m die Iterierten des Gesamtschrittverfahrens a) und $\tilde{\tilde{x}}_m$ diejenigen des Einzelschrittverfahrens b) dann konvergiert das Einzelschrittverfahren im folgenden Sinne besser als das Gesamtschrittverfahren:

Für alle $m \geq 0$ gilt die Beziehung $\tilde{\tilde{x}}_m \subset \tilde{x}_m$.

Eine weitere Konvergenzbeschleunigung in diesem Sinne kann unter der Voraussetzung (3) erreicht werden durch die Vorschrift:

$$b)^* \tilde{x}_{m+1} = (\mathcal{Q} \cdot \tilde{x}_{m+1} + (\mathcal{D} + \mathcal{R}) \cdot \tilde{x}_m + \mathfrak{b}) \cap \tilde{x}_m.$$

Für die Iterierten \hat{x}_m nach der Vorschrift b)* gilt dann: $\hat{x}_m \subset \tilde{\tilde{x}}_m$.

Viele praktische Probleme z. B. die Behandlung von Randwertaufgaben mit Differenzenmethoden sind in der Form (1) mit einer Matrix \mathfrak{B} , die die Konvergenzbedingung erfüllt gegeben.

Ist ein System in der Form

$$\mathcal{A} \cdot \mathfrak{x} = \mathfrak{f} \quad (4)$$

mit einer Intervallmatrix \mathcal{A} und einem Intervallvektor \mathfrak{f} gegeben, so wird die Bestimmung eines Vektors \mathfrak{x}^* mit der Eigenschaft:

$$\{\mathcal{A}^{-1} \cdot \mathfrak{f} : \mathcal{A} \in \mathcal{A}, \mathfrak{f} \in \mathfrak{f}\} \subset \mathfrak{x}^*$$

auf den eben behandelten Fall (1) zurückgeführt:

Ist $\mathcal{A} \in \mathcal{A}$ und existiert \mathcal{A}^{-1} und besitzt

$$\mathfrak{x} = \mathfrak{B} \cdot \mathfrak{x} + \mathfrak{b}$$

mit $\mathfrak{B} = (\mathcal{E} - \mathcal{A}^{-1} \cdot \mathcal{A})$, $\mathfrak{b} = \mathcal{A}^{-1} \cdot \mathfrak{f}$ einen eindeutig bestimmten Fixpunkt $\bar{\mathfrak{x}}$ dann liefert $\bar{\mathfrak{x}}$ das Gewünschte.

Bei den hier hergeleiteten Aussagen wurde vorausgesetzt, daß die vorkommenden Intervallverknüpfungen im Bereich der reellen Intervalle $I(\mathbb{R})$ exakt auszuführen sind. Sämtliche Eigenschaften und Aussagen, insbesondere die angeführten Inklusionen, bleiben auch dann noch richtig wenn man sich auf den Bereich der Maschinenintervalle $I(\mathbb{R}_M)$ beschränkt und, wie im folgenden geschehen, dazu eine geeignet gewählte Maschinenintervallarithmetik (siehe [2, 3]) einführt.

Weitere Einzelheiten, der hier zusammenfassend dargestellten Ergebnisse, finden sich in den im Literaturverzeichnis angeführten Arbeiten.

2. ALGOL-60-Programme

2.1

procedure ITERATION 1 (*mat, vek, n, exit*) result: (*x*);
comment Für die Maschinenintervalloperationen werden die in [3] beschriebenen Prozeduren LOW (A), ADD (A, B, C), SUB (A, B, C), MUL (A, B, C) und DIV (A, B, C) benützt. Das Programm behandelt ein System der Form (1) mit n Unbekannten nach der Vorschrift b)*. Den Elementen $B_{ij} = [b_{ij}^1, b_{ij}^2]$ der Matrix \mathfrak{B} entsprechen die Feldelemente $mat[i, j, 1]$ und $mat[i, j, 2]$ des Feldes $mat[1:n, 1:n, 1:2]$,

die Feldelemente $vek [i, 1]$ und $vek [i, 2]$ des Feldes $vek [1:n, 1:2]$ bezeichnen die Komponenten von b und entsprechend $x [i, 1]$ und $x [i, 2]$ des Feldes $x [1:n, 1:2]$ diejenigen von x . Bei Nichterfüllung der Konvergenzkriterien bewirkt ein Sprung nach *exit* den Abbruch des Programms.;

```

value mat, vek, n; array mat, vek, x; integer n; label exit;
begin integer i, j; real max, hmax, norm, hnorm; boolean bv;
  array y [1:n], ik, hmat, hx [1:2];
  real procedure UP (a); value a; real a; UP := - LOW (- a);
  real procedure BETRAG (ugr, ogr); real ugr, ogr;
    if ABS (ugr) less ABS (ogr) then BETRAG := ABS (ogr) else
      BETRAG := ABS (ugr);
  max := 0;
  comment Abschätzung nach der Zeilensummennorm;
  for i := 1 step 1 until n do
    begin y [i] := 0;
      for j := 1 step 1 until n do
        y [i] := UP (y [i] + BETRAG (mat [i, j, 1], mat [i, j, 2]));
        if y [i] notless 1 then goto m 1
      end;
  comment Bestimmung eines Anfangsvektors bei erfülltem Zeilen-
  summenkriterium.;
  for i := 1 step 1 until n do
    begin hmax := 0;
      for j := 1 step 1 until n do
        hmax := UP (hmax + UP (BETRAG (mat [i, j, 1], mat [i, j, 2]) ×
        BETRAG (vek [j, 1], vek [j, 2])));
        hmax := UP (hmax/LOW (1 - y [i]));
        if max less hmax then max := hmax
      end; goto m 2;
  comment Abschätzung nach der Spaltensummennorm;
  m 1: norm := 0;
  for j := 1 step 1 until n do
    begin hnorm := 0;
      for i := 1 step 1 until n do
        hnorm := UP (hnorm + BETRAG (mat [i, j, 1], mat [i, j, 2]));
        if hnorm notless 1 then goto exit;
        if hnorm greater norm then norm := hnorm
      end;
  comment Bestimmung eines Anfangsvektors bei erfülltem Spalten-
  summenkriterium.;
  for i := 1 step 1 until n do
    for j := 1 step 1 until n do
      max := UP (max + UP (BETRAG (mat [i, j, 1], mat [i, j, 2]) ×
      BETRAG (vek [j, 1], vek [j, 2])));
      max := UP (max/LOW (1 - norm));

```

```

m 2: for i := 1 step 1 until n do
    begin x [i, 1] := LOW (vek [i, 1] - max);
          x [i, 2] := UP (vek [i, 2] + max) end;
comment Beginn der Iteration.;
m 3: bv := true;
for i := 1 step 1 until n do
begin ik [1] := vek [i, 1]; ik [2] := vek [i, 2];
  for j := 1 step 1 until n do
  begin hmat [1] := mat [i, j, 1]; hmat [2] := mat [i, j, 2];
        hx [1] := x [j, 1]; hx [2] := x [j, 2];
        MUL (hmat, hx, hx); ADD (hx, ik, ik)
  end;
  hx [1] := x [i, 1]; hx [2] := x [i, 2];
  if ik [1] less hx [1] then ik [1] := hx [1];
  if ik [2] greater hx [2] then ik [2] := hx [2];
  if bv then begin if ik [1] notequal x [i, 1] or ik [2] notequal
x [i, 2] then bv := false end;
  x [i, 1] := ik [1]; x [i, 2] := ik [2]
end;
if not bv then goto m 3
end;
end;

```

2.2

procedure ITERATION 2 (*mat, vek, n, exit*) result: (*x*);
comment Für die Maschinenintervalloperationen werden die in [3] beschriebenen Prozeduren LOW (A), ADD (A, B, C), SUB (A, B, C), MUL (A, B, C) und DIV (A, B, C) benützt. Das Programm behandelt ein System der Form (4) mit n Unbekannten. Den Elementen $A_{ij} = [a_{ij}^1, a_{ij}^2]$ der Matrix \mathfrak{A} entsprechen die Feldelemente $mat [i, j, 1]$ und $mat [i, j, 2]$ des Feldes $mat [1:n, 1:n, 1:2]$, die Feldelemente $vek [i, 1]$ und $vek [i, 2]$ des Feldes $vek [1:n, 1:2]$ bezeichnen die Komponenten von \mathfrak{k} und entsprechend $x [i, 1]$ und $x [i, 2]$ des Feldes $x [1:n, 1:2]$ diejenigen von \mathfrak{x} . Bei Nichterfüllung der Konvergenzkriterien bewirkt ein Sprung nach *exit* den Abbruch des Programms.;

```

value mat, vek, n; array mat, vek, x; integer n; label exit;
begin integer i, j, k; array hg 1, hg 2, hg 3 [1:2], hv [1:n, 1:2],
  pmat [1:n, 1:n];
procedure MATINV (pmat);
comment Bibliotheksprogramm zur Invertierung einer  $n \times n$ -
  Punktmatrix pmat. Die derart berechnete angenäherte Inverse
  wird auf demselben Feld pmat geliefert.;
array pmat; code;
procedure ITERATION 1 (mat, vek, n, exit, x);
comment Hierbei handelt es sich um die obige Prozedur gleichen
  Namens.;

```

```

value mat, vek, n; array mat, vek, x; integer n; label exit;
code;
for i := 1 step 1 until n do
for j := 1 step 1 until n do pmat [i, j] := (mat [i, j, 1] +
mat [i, j, 2])/2;
MATINV (pmat);
for i := 1 step 1 until n do
begin for j := 1 step 1 until n do
begin hg 1 [1] := hg 1 [2] := 0;
for k := 1 step 1 until n do
begin hg 2 [1] := hg 2 [2] := pmat [j, k];
hg 3 [1] := mat [k, i, 1]; hg 3 [2] := mat [k, i, 2];
MUL (hg 2, hg 3, hg 3); SUB (hg 1, hg 3, hg 1)
end;
hv [j, 1] := hg 1 [1]; hv [j, 2] := hg 1 [2]
end;
for k := 1 step 1 until n do
begin mat [k, i, 1] := hv [k, 1]; mat [k, i, 2] := hv [k, 2] end
end;
for i := 1 step 1 until n do
begin mat [i, i, 1] := LOW (1 + mat [i, i, 1]);
mat [i, i, 2] := - LOW (- 1 - mat [i, i, 2])
end;
for i := 1 step 1 until n do
begin hg 1 [1] := hg 1 [2] := 0;
for j := 1 step 1 until n do
begin hg 2 [1] := hg 2 [2] := pmat [i, j];
hg 3 [1] := vek [j, 1]; hg 3 [2] := vek [j, 2];
MUL (hg 2, hg 3, hg 3); ADD (hg 1, hg 3, hg 1)
end;
hv [i, 1] := hg 1 [1]; hv [i, 2] := hg 1 [2]
end;
ITERATION 1 (mat, hv, n, exit, x)
end;

```

Literatur

- [1] ALEFELD, G.: Intervallrechnung über den komplexen Zahlen und einige Anwendungen. Diss. Universität Karlsruhe. 1968.
- [2] APOSTOLATOS, N., und U. KULISCH: Approximation der erweiterten Intervallarithmetik durch die einfache Maschinenintervallarithmetik. *Comp.* **2**, 181–194 (1967).
- [3] CHRIST, H.: Realisierung einer Maschinenintervallarithmetik mit beliebigen ALGOL-60 Compilern. *Elektron. Rechenanlagen* **10**, 217–222 (1968).
- [4] HANSEN, E.: Interval arithmetic in matrix computations P. 1. J. S. I. A. M., series B, *Numerical Analysis*, **1**, 2 (1965).
- [5] HANSEN, E., and R. SMITH: Interval arithmetic in matrix computations P. 2. J. S. I. A. M., series B, *Numerical Analysis*, **4**, 1 (1967).
- [6] HERZBERGER, J.: Definition und Eigenschaften allgemeiner Intervallräume. *ZAMM-Sonderheft* (1969).

- [7] ULLRICH, Ch.: Algorithmen zur intervallmässigen Behandlung von linearen Gleichungssystemen. Diplomarbeit, Universität Karlsruhe. 1969.
- [8] KULISCH, U.: Grundzüge der Intervallrechnung. Erschienen in Überblicke, 2, Bibliograph. Institut Mannheim.
- [9] MAYER, O.: Algebraische und metrische Strukturen in der Intervallrechnung und einige Anwendungen. Comp. 5, 144–162 (1970).
- [10] MOORE, R. E.: Interval Analysis. Englewood Cliffs: Prentice-Hall. 1966.
- [11] KLATTE, R.: Anwendungen einer Verallgemeinerung der Modifikation des NEWTON-Verfahrens. Diplomarbeit, Universität Karlsruhe. 1969.

*Dr. Götz Alefeld und Dr. Jürgen Herzberger
Institut für Angewandte Mathematik
und Rechenzentrum
der Universität Karlsruhe
Englerstraße 2, D-75 Karlsruhe
Bundesrepublik Deutschland*