INHALTSVERZEICHNIS

Seite

TEIL 7 NUMERIK PARTIELLER DIFFERENTIALGLEICHUNGEN

FEHLERANALYSE

## § 1 Zahldarstellungen und Rundungsfehler

Die d-näre Darstellung von $x \in \mathbb{R}$ ist

$$x = \pm \sum_{k=\ell}^{-\infty} a_k \cdot d^k \qquad 0 \leq a_k < d \quad .$$

Schreibweise:

$$x = \pm\, a_\ell\, a_{\ell-1} \cdots a_0 \bullet a_{-1} \cdots \quad .$$

Der Punkt bestimmt die Lage von $a_0$ .

Beispiele:

1) d = 10 (Dezimale Darstellung)

π = 3.1415 92653 58979 32384 62643 ...

1 = 1.00 ... = 0.99 ... (nicht eindeutig)

2) d = 2 (Duale Darstellung)

5 = L O L

3.2 = LL . $\overline{\text{OO LL}}$ (Strich bedeutet Periode )

3) d = 16 (Hexadezimale oder Sedizemale Darstellung)

29 = 1 D

Problem: Die Rechenmaschine kann nur endlich viele Zahlen darstellen.

Beschreibung der Rechnerarithmetik (einer d-nären normalisierten Gleitkommamaschine mit Mantissenlänge m):

I. Menge der Maschinenzahlen A

$$A = \{a : a = \pm 0.\, a_1 \ldots a_m\, d^{\pm b_\ell \ldots b_0} \text{ mit } a_1 \neq 0 \text{ oder } a = 0\}$$

Der Wert der Zahlen $\neq 0$ ist

$$\pm\, a_1\, d^{-1} + \ldots + a_m\, d^{-m} \cdot d^{\pm(b_\ell d^\ell + \ldots + b_0 d^0)} \quad .$$

Bezeichnungen:

$$\pm\, 0.\underbrace{a_1 \ldots a_m}_{\text{heißt Mantisse}}\, d\underbrace{\pm\, b_\ell \ldots d_0\, b_0}_{\text{heißt Exponent}}$$

Beispiele: $d = 10$ , $m = 4$

| Maschinenzahl | Wert |
|---|---|
| $0.3142_{10}1$ | 3.142 . |

Keine Maschinenzahlen sind:

| | |
|---|---|
| $0.31415_{10}1$ | (Mantisse zu lang) |
| $0.0314_{10}1$ | (nicht normalisiert) |

## II. Rundung

**Definition 1.1:** Eine Abbildung

$$rd : \mathbb{R} \to A \qquad \text{heißt Rundung} \Leftrightarrow$$

$$|rd(x) - x| \leq \min_{a \in A} |a - x|$$

**Beispiel:** $d = 10$

$$x = 0 \quad rd(x) = 0$$

$$x \neq 0 \quad x = \pm 0.a_1 a_2 \ldots \; 10^b \;, \quad a_1 \neq 0$$

$$rd(x) = \pm \tilde{a} \cdot 10^b$$

$$\tilde{a} = \begin{cases} 0.a_1 \ldots a_m & \text{falls} \quad a_{m+1} \leq 4 \\ 0.a_1 \ldots a_m + 10^{-m} \ldots a_{m+1} \geq 5 \end{cases}$$

**Beispiele:** $m = 4$

$$\pi = 3.14159 \ldots \qquad rd(\pi) = 0.3142_{10}1$$

$$\sqrt{57} = 7.5498 \ldots \qquad rd(\sqrt{57}) = 0.7550_{10}1$$

$$x = 0.12535 \ldots \qquad rd(x) = 0.1254_{10}0$$

$$x = 0.1253499 \ldots \qquad rd(x) = 0.1253_{10}0$$

**Definition 1.2:**

$x \in \mathbb{R}^1$ , $\tilde{x}$ sei eine Näherung (Approximation) von $x$ .

$|x - \tilde{x}|$ heißt absoluter Fehler von $\tilde{x}$

$\left|\frac{x - \tilde{x}}{x}\right|$ heißt relativer Fehler von $\tilde{x}$ $(x \neq 0)$

Definition 1.3: $eps = \frac{1}{2} d^{-m+1}$

heißt die Maschinengenauigkeit einer d-nären normierten Gleitkommamaschine mit Mantissenlänge $m$ .

Satz 1.1: Sei $eps < 1$, $rd$ eine Rundung

Dann gilt:

(i) $\left|\frac{rd(x)-x}{x}\right| \leq eps$

(ii) $\left|\frac{rd(x)-x}{rd(x)}\right| \leq eps$

(iii) $\exists\ \rho \in \mathbb{R}$ mit $rd(x) = \rho x$ und

$|\ln \rho| \leq eps$

Beweis:

(i) $x = \pm\ 0.\ a_1 \ldots a_{m+1}\ d^b$

$|rd(x) - x| = \min_{a \in A} |a - x| \leq \frac{d}{2} d^{-m-1} d^b$

$|x| \geq d^{-1} d^b$ (wegen Normalisierung) $d^{-1} = 0,1$

$\Rightarrow \left|\frac{rd(x)-x}{x}\right| \leq \frac{1}{2} d^{-m} d^b / d^{-1} d^b = \frac{1}{2} d^{-m+1}$

(ii) wie (i)

(iii) Nach (i), (ii) existieren Zahlen $\varepsilon_i$ $i = 1,2$ mit $|\varepsilon_i| \leq$ eps, so daß

$$\frac{rd(x)-x}{x} = \varepsilon_1 \quad , \quad \frac{rd(x)-x}{rd(x)} = \varepsilon_2 \quad \text{gilt .}$$

$$rd\,x = (1+\varepsilon_1)x \qquad rd(x) = \frac{x}{1-\varepsilon_2}$$

Es gilt: $1 + \varepsilon \leq e^{\varepsilon} \quad \forall\, \varepsilon \in \mathbb{R}$

$$\Rightarrow \quad (1+\varepsilon_1) \leq e^{\varepsilon_1} \leq e^{eps}$$

$$(1-\varepsilon_2)^{-1} \geq e^{-\varepsilon_2} \geq e^{-eps}$$

$$\Rightarrow \quad \exists\, \rho \in \mathbb{R} \quad \text{mit} \quad rd(x) = \rho x \quad \text{mit} \quad e^{-eps} \leq \rho \leq e^{eps}$$

$\Rightarrow$ Behauptung.

## III. Maschinenoperationen

+, -, /, * ist nicht abgeschlossen in A.

Beispiel: d = 10, m = 2

$$1/0.9 = 1.\overline{11} \notin A$$

$$1.1 * 1.1 = 1.21 \notin A$$

$$0.13 + 0.0071 = 0.1371 \notin A$$

Statt der reellen Operationen sind Maschinenoperationen $\oplus$ , $\ominus$ , $\circledast$ , $\oslash$ erklärt:

$$x \oplus y = rd(x + y)$$

$$x \ominus y = rd(x - y)$$

$$x \circledast y = rd(x * y)$$

$$x \oslash y = rd(x / y) \quad .$$

Satz 1.2: Der relative Fehler der Maschinenoperationen ist $\leq$ eps. Es gilt

$$x \oplus y = \rho(x + y) \qquad |\ln \rho| \leq eps \quad .$$

Beweis: klar

Dies gilt jedoch nicht für die Nacheinanderausführung von Maschinenoperationen.

Beispiele: d = 10, m = 2

Reelle Operationen:

$$0.75 + 0.055 - 0.80 = 0.005$$

Maschinenoperationen:

$$\underbrace{\underbrace{(0.75 \oplus 0.055)}_{rd\ (0.805)}}_{0.81} - 0.80 = 0.01$$

der relative Fehler ist 1!

In anderer Reihenfolge der Operationen erhält man

$$0.75 \oplus (0.055 \ominus 0.80) = 0 \quad .$$

Dieses Resultat ist nicht besser. Man sieht außerdem, daß die Maschinenaddition nicht assoziativ ist.

Man sieht, daß Vorsicht am Platze ist. In den beiden folgenden Paragraphen werden wir sehen, wie man trotz der Unvollkommenheit der Maschinenarithmetik zu zuverlässigen Resultaten kommen kann.

Abweichungen der reellen Rechner von unserer idealisierten Maschine:

1) Unterschiedliche Rundung.

2) Unterschiedliche Normalisierung.

3) Endlicher Exponentialbereich. Für die kurze Arithmetik der IBM/370 ist z.B. $-64 \leq b \leq 63$.

   Beispiel: $d = 10$ , $m = 2$ , $b = -99,\ldots,99$

   Überlauf $0.13_{10}52 \odot 0.10_{10}60 = 0.13_{10}111 \notin A$

   $0.13_{10}1 \oslash 0.26_{10}-99 = 0.50_{10}100 \notin A$

   Unterlauf $0.12_{10}-99 \ominus 0.11_{10}-99 = 0.10_{10}-100 \notin A$

   Gute Maschinen brechen bei Unterlauf ab oder erstatten Meldung. Andere rechnen unnormalisiert weiter oder setzen das Resultat 0. Dann stimmt Satz 1.1 nicht mehr!

4) Andere Zahldarstellungen.

(a) Festkommarithmetik, z.B.

$$a \in A \Leftrightarrow a = \pm\, 0.a_1 \ldots a_m \,.$$

(b) Residuenarithmetik.

Seien $k_1,\ldots,k_m$ paarweise teilerfremd und $k = k_1 \cdots k_m$. Seien $a_i$ die Reste von $a$ bei Division durch $k_i$. Für $0 \leq a < k$ ist $a$ eindeutig durch diese Reste bestimmt: $a = (a_1,\ldots,a_m)$. Es gilt dann

$$ab = (a_1b_1,\ldots,a_mb_m) \quad ,$$

$$a+b = (a_1 + b_1,\ldots,a_m + b_m) \quad ,$$

wobei $a_ib_i$, $a_i + b_i$ modulo $k_i$ berechnet werden.

(c) Intervallarithmetik.

Hier rechnet man mit Zahlenpaaren $[\underline{a}\,,\,\overline{a}]$, welche das Intervall $\{x : \underline{a} \leq x \leq \overline{a}\}$ darstellen. Man erklärt dann

$$[\underline{a}\,,\,\overline{a}] + [\underline{b}\,,\,\overline{b}] = \{x = a+b : \underline{a} \leq a \leq \overline{a}\,,\ \underline{b} \leq b \leq \overline{b}\}$$

und entsprechend für die anderen Operationen.

## § 2 Kondition und Gutartigkeit

Aufgabe: Auswertung von $f(x) = y$, $x \in D$, $f : \mathbb{R}^p \to \mathbb{R}^q$

Was geschieht, wenn für $x$ nur Näherungen $\tilde{x} = x + \Delta x$ zur Verfügung stehen? $\tilde{y} = f(\tilde{x}) = y + \Delta y$

Wie groß ist der Fehler in $\tilde{y}$ ? Antwort gibt:

Satz 2.1: Sei $D$ konvex, $f \in C^1(D)$ .
1x stetig diff'bar über D
Dann gilt:

$$\left| \frac{\Delta y_i}{y_i} \right| \leq \sum_{j=1}^{p} \max_{z \in D} \left| \frac{\partial f_i(z)}{\partial x_j} \right| \cdot \left| \frac{x_j}{f_i(x)} \right| \cdot \left| \frac{\Delta x_j}{x_j} \right|$$

Beweis:

$$\Delta y_i = f_i(x + \Delta x) - f_i(x)$$

Wir betrachten nun die Hilfsfunktion $g$,

$$g(t) = f_i(x + t\Delta x) - f_i(x)$$

$$\Rightarrow \quad g(1) = \Delta y_i \ , \quad g(0) = 0$$

Mittelwertsatz:

$$\Rightarrow \quad |g(1) - g(0)| \leq 1 \cdot \max_{t \in [0,1]} |g'(t)| \qquad \text{mit}$$

$$g'(t) = \sum_{j=1}^{p} \frac{\partial f_i}{\partial x_j} (x + t\Delta x) \cdot \Delta x_j$$

$$\Rightarrow \quad |\Delta y_i| \leq \sum_{j=1}^{p} \max_{z \in D} \left| \frac{\partial f_i}{\partial x_j}(z) \right| |\Delta x_j|$$

$\Rightarrow$ Behauptung

Definition 2.1: Die Zahlen

$$k_{ij}(x) = \left| \frac{\partial f_i(x)}{\partial x_j} \; \frac{x_j}{f_i(x)} \right|$$

heißen Verstärkungsfaktoren. Für die Auswertung von $f(x)$ gilt dann näherunsweise

$$\left| \frac{\Delta y_i}{y_i} \right| \leq \sum_{j=1}^{p} k_{ij}(x) \left| \frac{\Delta x_j}{x_j} \right|$$

Definition 2.2: Die Aufgabe "Berechne $f(x)$" heißt gut konditioniert, falls alle $k_{ij}(x)$ die Größenordnung 1 haben. Andernfalls heißt die Aufgabe schlecht konditioniert. Es treten dann Verstärkungsfaktoren auf, welche viel größer sind als 1 ($k_{ij} >> 1$).

Beispiele:

1) $y_1 = x_1 + x_2$ $\quad k_{11} = \left| 1 \, \frac{x_1}{y_1} \right| = \left| \frac{x_1}{x_1 + x_2} \right|$

$$k_{12} = \left| \frac{x_2}{x_1 + x_2} \right|$$

Diese Aufgabe ist schlecht konditioniert, falls $\frac{x_1}{x_2} \approx -1$.

Merke: Subtraktion nahezu gleichgroßer Zahlen ist schlecht konditioniert!

$$1.31 - 1.25 = 0.06$$
$$1.32 - 1.24 = 0.08$$

rel. Fehler 0,8% 30%

Man spricht von Auslöschung.

2) $y_1 = x_1 x_2$ $\qquad k_{11} = k_{12} = 1$

3) $y_1 = x_1/x_2$ $\qquad k_{11} = k_{12} = 1$

4) $y_1 = x_1^a$ $\qquad k_{11} = a$

5) $y^2 - x_1 y - x_2 = 0$

$$y_1 = \frac{x_1}{2} + \sqrt{d} \quad , \quad d = x_1^2/4 + x_2$$

$$y_2 = \frac{x_1}{2} - \sqrt{d}$$

$$k_{11} = \left| \left(\frac{1}{2} + \frac{1}{2}\,\frac{x_1}{2}\, d^{-1/2}\right) \frac{x_1}{y_1} \right| = \frac{1}{2}\, d^{-1/2} \left| \left(d^{1/2} + \frac{x_1}{2}\right) \frac{x_1}{y_1} \right|$$

$$= \left| \frac{x_1}{2\sqrt{d}} \right| = k_{21}$$

$$k_{12} = \frac{|y_2|}{2\sqrt{d}} \qquad k_{22} = \frac{|y_1|}{2\sqrt{d}}$$

Das Lösen von quadratischen Gleichungen ist schlecht konditioniert, falls $\sqrt{d} << |x_1|, |y_1|, |y_2|$.

Die Auswertung von f geschieht durch einen Algorithmus, der aus den Daten über Zwischenergebnisse die Endergebnisse erzeugt.

| | |
|---|---|
| $x_1 , \ldots , x_p$ | Daten |
| $z_1 \qquad\qquad z_p$ | Zwischenergebnisse |
| $z_{p+1} = g_{p+1}\ (z_1,\ldots,z_p)$ | " " |
| $z_{p+2} = g_{p+2}\ (z_1,\ldots,z_{p+1})$ | " " |
| $z_m = g_m\ (z_1,\ldots,z_{m-1})$ | " " |
| $y_1 = z_{m-q+1} , \ldots , y_q = z_m$ | Endergebnisse |

Die $g_i$ seien direkt auf der Maschine auswertbar.

Der Zusammenhang der Zwischenresultate wird durch den Graphen des Algorithmus beschrieben. Der Graph ist durch seine Knoten und Kanten definiert: Die Knoten sind die Zwischenresultate $z_j$. Es existiert eine Kante von $z_k$ nach $z_j$, falls $z_k$ bei der Berechnung von $z_j$ benötigt wird.

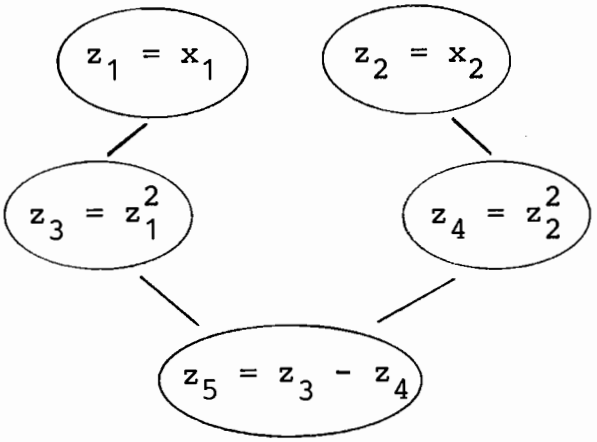
Beispiele: $y_1 = x_1^2 - x_2^2$

Algorithmus 1: $y_1 = x_1^2 - x_2^2$

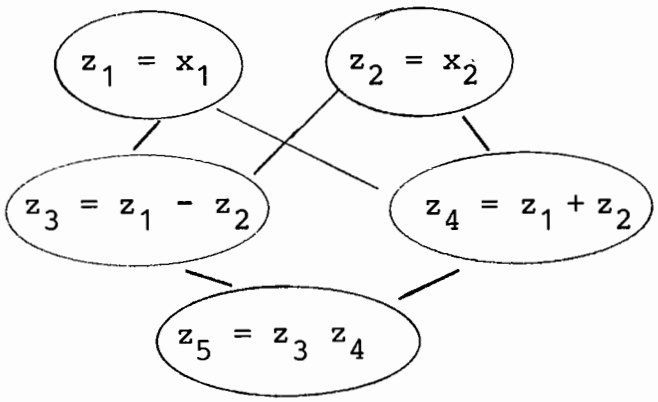
Algorithmus 2: $y_1 = (x_1 - x_2)(x_1 + x_2)$

Graphen der Algorithmen:

Algorithmus 1:




Algorithmus 2:



Bei der Durchführung eines Algorithmus treten 2 Fehler auf:

1) Der unvermeidbare Fehler.

   Er entsteht nur durch die Rundung der Daten und ist unabhängig vom Algorithmus.

Eine Abschätzung nach Satz 2.1 liefert:

Unvermeidbarer Fehler $\sum_{i=1}^{q} \left| \frac{\Delta y_i}{y_i} \right| \leq \sum_{i=1}^{q} \sum_{j=1}^{p} k_{ij}(x) \cdot eps$

2) Der Algorithmusfehler, auch kumulative Rundungsfehler.

   Man verfolge die Geschichte eines jeden Rundungsfehlers, der an irgendeinem Knoten entsteht, bis zum Endresultat.

Eine Abschätzung nach Satz 2.1 liefert:

Sei $\varepsilon_k$ der relative Fehler in $z_k$, $\varepsilon_i$ der in $z_i$, $z_j = g(z_i, z_k)$

$$a_{ji} := \left| \frac{\partial g}{\partial z_i} \frac{z_i}{z_j} \right| .$$

Für den rel. Fehler $\varepsilon_j$ des Zwischenresultates $z_j = g(z_i, z_k)$ gilt:

$$\varepsilon_j \sim (a_{ji}\, \varepsilon_i + a_{jk}\, \varepsilon_k) + eps .$$

Der Algorithmusfehler wird durch folgende Schritte berechnet:

1) Man schreibe die Verstärkungsfaktoren

$$\left| \frac{\partial z_j}{\partial z_i} \; \frac{z_i}{z_j} \right| \quad \text{an alle Kanten des Graphen ,}$$

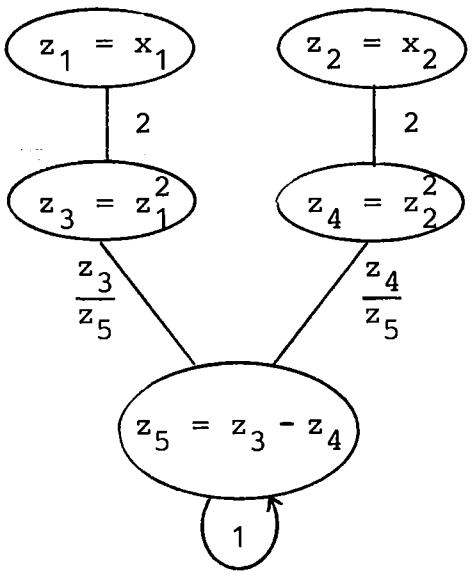
und ergänze den Graphen durch Kanten mit Verstärkungsfaktoren 1, welche Endresultate mit sich selber verbinden.

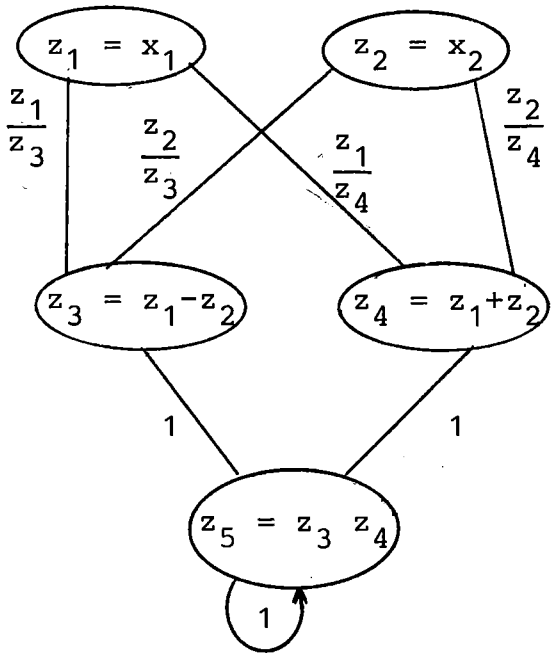
2) Für jeden Knoten finde man die Produkte der Verstärkungsfaktoren entlang eines jeden zu einem Endresultat führenden Weges.

3) Man summiere über die Produkte und multipliziere das Ergebnis mit eps .

<u>Beispiele:</u> $y_1 = x_1^2 - x_2^2$

Algorithmus 1: $y_1 = x_1^2 - x_2^2$ Algorithmus 2: $y_1 = (x_1-x_2)(x_1+x_2)$



Algorithmusfehler des Algorithmus 1:

$$\left(2\left|\frac{z_3}{z_5}\right| + \left|\frac{z_3}{z_5}\right| + 1 + 2\left|\frac{z_4}{z_5}\right| + \left|\frac{z_4}{z_5}\right|\right) \text{eps}$$

$$= \left(3\,\frac{x_1^2 + x_2^2}{|x_1^2 - x_2^2|} + 1\right) \text{eps}$$

Algorithmusfehler des Algorithmus 2:

$$\left(\left|\frac{z_1}{z_3}\right| + \left|\frac{z_1}{z_4}\right| + 1 + 1 + 1 + \left|\frac{z_2}{z_4}\right| + \left|\frac{z_2}{z_3}\right|\right) \text{eps}$$

$$= \left(\left(|x_1| + |x_2|\right)\left(\frac{1}{|x_1 - x_2|} + \frac{1}{|x_1 + x_2|}\right) + 3\right) \text{eps}$$

Definition 2.3: Ein Algorithmus heißt gutartig, wenn der Algorithmusfehler nicht "wesentlich" größer ist als der unvermeidbare Fehler.

Bemerkung: Ein gutartiger Algorithmus produziert nicht notwendig kleine relative Fehler!

Beispiel 1: $y_1 = x_1^2 - x_2^2$

Unvermeidbarer Fehler: $\left(\frac{2(x_1^2 + x_2^2)}{|x_1^2 - x_2^2|}\right)$ eps ,

Algorithmusfehler von Algorithmus 1:

$$\left(\frac{3(x_1^2 + x_2^2)}{|x_1^2 - x_2^2|} + 1\right) \text{eps}$$

⇒ Algorithmus 1 ist gutartig.

Auch Algorithmus 2 ist gutartig.

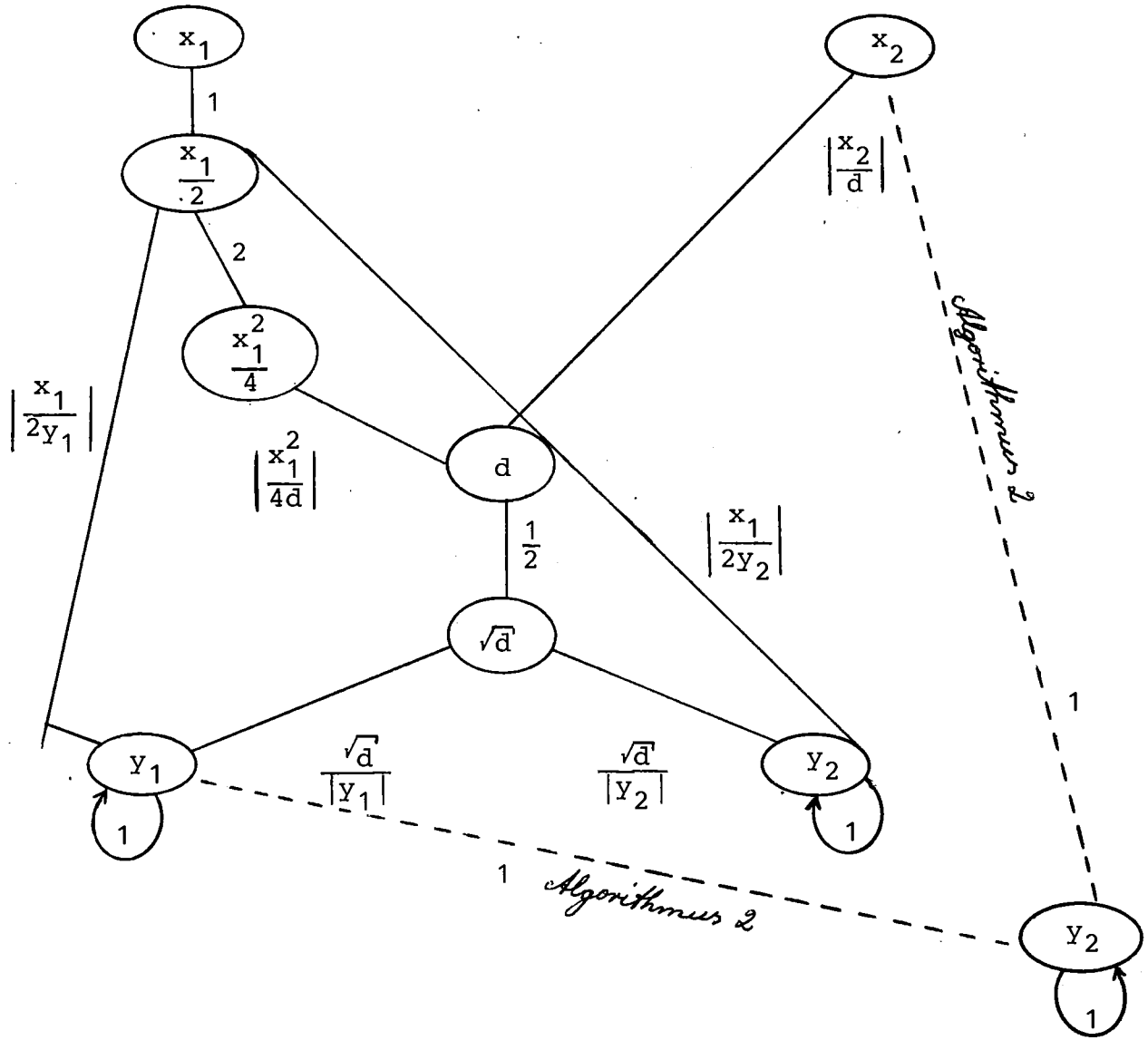
2) $y^2 - x_1 y - x_2 = 0$ ,

Unvermeidbarer Fehler: $d^{-\frac{1}{2}} (|x_1| + \frac{1}{2}|y_1| + \frac{1}{2}|y_2|)$ eps

Algorithmus 1: $d = \frac{x_1^2}{4} + x_2 > 0$

$$y_1 = \frac{x_1}{2} + \sqrt{d}$$

$$y_2 = \frac{x_1}{2} - \sqrt{d}$$



Algorithmusfehler:

$$\Big( \Big| 1 \cdot 2 \cdot \frac{x_1^2}{4d} \cdot \frac{1}{2} \cdot \frac{\sqrt{d}}{y_1} \Big| + \Big| 2 \, \frac{x_1^2}{4d} \;\; \frac{1}{2} \, \frac{\sqrt{d}}{y_1} \Big| + \Big| \frac{x_1^2}{4d} \;\; \frac{1}{2} \, \frac{\sqrt{d}}{y_1} \Big|$$

$$+ \Big| \frac{1}{2} \, \frac{\sqrt{d}}{y_1} \Big| + \Big| \frac{\sqrt{d}}{y_1} \Big| + 1 + \;\; 1 \Big| \frac{x_1}{2y_1} \Big| + \Big| \frac{x_1}{2y_1} \Big| \;\; +$$

$$\Big| \frac{x_2}{d} \cdot \frac{1}{2} \cdot \frac{\sqrt{d}}{y_1} \Big| + \text{entsprechenden Ausdruck mit } y_2 \text{ anstelle von } y_1 \Big) \text{ eps}$$

$$= \frac{\text{eps}}{|y_1|\sqrt{d}} \Big( \frac{x_1^2}{4} + \frac{x_1^2}{4} + \frac{x_1^2}{8} + \frac{1}{2} d + d + |y_1|\sqrt{d} + |x_1|\sqrt{d} + \frac{|x_2|}{2} \Big) + \text{Ausdruck in } y_2$$

$$= \frac{\text{eps}}{\sqrt{d}} \Big( \frac{1}{|y_1|} + \frac{1}{|y_2|} \Big) \Big( \frac{5}{8} x_1^2 + \frac{3}{2} d + \sqrt{d} \, |x_1| + \frac{|x_2|}{2} \Big) + 2 \text{ eps}$$

Der Algorithmus 1 ist nicht gutartig, falls $y_1$ oder $y_2 \approx 0$ sind, da dann $(\frac{1}{|y_1|} + \frac{1}{|y_2|})$ sehr groß wird.

Die Auslöschung bei $\frac{x_1}{2} - \sqrt{d}$ wird in Algorithmus 2 vermieden:

$$y_1 = \frac{x_1}{2} + \sqrt{d} \;, \quad y_2 = - \frac{x_2}{y_1}$$

Algorithmusfehler von Algorithmus 2:

$$\frac{2 \text{ eps}}{\sqrt{d}|y_1|} \Big( \frac{5}{8} x_1^2 + \frac{3}{2} d + \sqrt{d} \, |x_1| + \frac{|x_2|}{2} \Big) + \text{eps} \;\; +$$

$$(1 + 1 + 1) \text{ eps}$$

Dieser Algorithmus ist nicht gutartig für $y_1 \sim 0$.

Algorithmus 3 ist gutartig für alle $y_1, y_2$:

Falls $x_1 \geq 0 : y_1 = \frac{x_1}{2} + \sqrt{d}$ , $y_2 = -\frac{x_2}{y_1}$

$x_1 \leq 0 : y_2 = \frac{x_1}{2} - \sqrt{d}$ , $y_1 = -\frac{x_2}{y_2}$

Zahlenbeispiel:

Die kurze Gleitkommaarithmetik der IBM/370-Rechner verwendet 6 Hexadezimalziffern, also $eps = \frac{1}{2}\, 16^{1-6} \sim 5 \cdot 10^{-7}$. Wir erwarten also einen relativen Rundungsfehler der Größenordnung $5 \cdot 10^{-7}$, d.h. 6 richtige Dezimalen.

Lösen wir in dieser Arithmetik

$$y^2 - 2y - 0.0002 = 0 \quad ,$$

so ist auf 6 Stellen

$$y_1 = 2.00010 \quad , \quad y_2 = -\ 9.99950\ 10^{-5} \quad .$$

Algorithmus 1 liefert

$$y_1 = 2.00010 \quad , \quad y_2 = -\ 1.00135\ 10^{-4} \quad ,$$

und Algorithmus 2

$$y_1 = 2.00010 \quad , \quad y_2 = -\ 9.99949 \cdot 10^{-5} \quad .$$

Der Fehler bei Algorithmus 1 ist also $10^{-3}$, der bei Algorithmus 2 $10^{-6}$.

3) $y^2 - 2.4y + 1.4 = 0$ also $x_1 = 2.4,\ x_2 = 1.4$

Exaktes Ergebnis: $y_1 = 1.4$ , $y_2 = 1.0$ , $d = 0.04$

Was liefert Algorithmus 2 auf einer Maschine mit $d = 10,\ m = 2$ ?

$$d = 1.2 \overset{(2)}{\ominus} 1.4 = 0$$

$$\tilde{y}_1 = 1.2 \quad , \quad \tilde{y}_2 = 1.2$$

Dies liegt an der schlechten Kondition der Aufgabe.

4) $y_1 = 0.75 + 0.055 - 0.80 = x_1 + x_2 + x_3 = 0.005$

$$k_{ij} = \left| \frac{x_j}{y_1} \right| = 150,\ 11,\ 160$$

Unvermeidbarer Fehler: $321 \cdot \text{eps} = 321 \cdot 5_{10}{}^{-2} \approx 15$

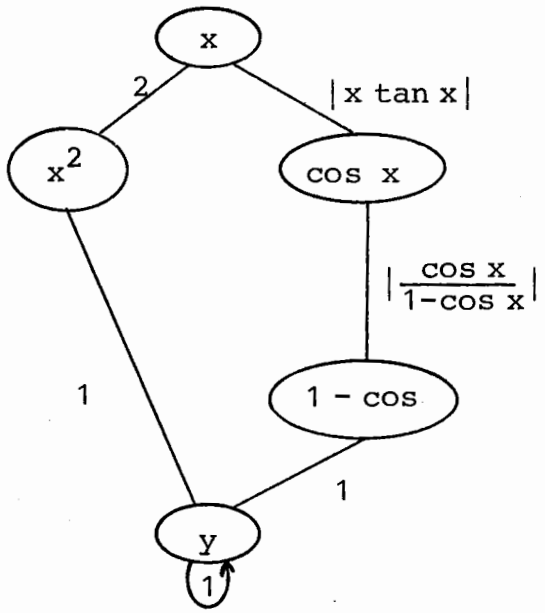
5) $$y = \frac{1 - \cos x}{x^2} \quad , \quad |x| \leq 1$$

$$= 2\left(\frac{\sin \frac{x}{2}}{x}\right)^2$$

$$k = \left| \frac{x \cos \frac{x}{2}}{\sin \frac{x}{2}} \right| \leq \frac{1}{\sin \frac{1}{2}}$$ also gut konditioniert.

Für diese Aufgabe untersuchen wir 3 Algorithmen.

Algorithmus 1:




Der kritische Teil des Algorithmusfehlers ist:

$$\left| \frac{\cos x}{1 - \cos x} \right| \cdot 1 >> k \quad ,$$

also ist der Algorithmus 1 nicht gutartig.

Algorithmus 2:

$$y = \begin{cases} 2 \left( \frac{\sin \frac{x}{2}}{x} \right)^2 & , \quad x = 0 \\ \frac{1}{2} & , \quad x = 0 \end{cases}$$

Algorithmus 3:

Man verwende $\cos x = 1 - \frac{1}{2} x^2 + \frac{1}{24} x^4 + \ldots$ für $|x| \leq \delta$

Algorithmen 2 und 3 sind gutartig (Bei Algorithmus 2 wird natürlich angenommen, daß auf der Maschine eine Funktion $\overset{o}{\sin}$ mit

$$\overset{o}{\sin} x = \rho \sin x \quad , \qquad |\ell n \, \rho| \leq eps$$

zur Verfügung steht.)

## § 3 Die Rückwärtsanalyse des Rundungsfehlers

Wilkinson: Kann das Maschinenresultat als Resultat einer reellen Arithmetik auf gestörten Daten interpretiert werden?

Beispiel: $0.75 + 0.055 - 0.80 = 0.005$

$(0.75 \oplus 0.055) \ominus 0.80 = 0.01$

$0.755 + 0.055 - 0.80 = 0.01$

↑ kleine Störung

Definition 3.1: Sei $y_i = f_i(x_1,\ldots,x_p)$

Die Maschinenresultate eines Algorithmus seien $\tilde{y}_i$ .

Existiert $R \in \mathbb{R}$ (nicht zu groß ), so daß gilt:

$$\tilde{y}_i = f_i(\tilde{x}_1^i, \ldots, \tilde{x}_p^i) \quad \text{mit} \quad \left|\frac{\tilde{x}_j^i - x_j}{x_j}\right| \leq R\,\text{eps} \ ,$$

dann heißt der Algorithmus rückwärts stabil.

Satz 3.1: Ein rückwärts stabiler Algorithmus ist gutartig.

Beweis: Algorithmusfehler =

$$\sum_{i=1}^{q} \left|\frac{y_i - \tilde{y}_i}{y_i}\right| = \sum_{i=1}^{q} \left|\frac{f_i(x) - f_i(\tilde{x})}{f_i(x)}\right|$$

$$\leq \sum_{i=1}^{q} \frac{1}{|f_i(x)|} \sum_{j=1}^{p} |x_j - \tilde{x}_j| \operatorname*{Max}_{y \in D} \left| \frac{\partial f_i}{\partial x_j}(y) \right|$$

$$\sim \sum_{i=1}^{q} \sum_{j=1}^{p} k_{ij}(x) \left| \frac{x_j - \tilde{x}_j}{x_j} \right|$$

$$\leq R \sum_{i=1}^{q} \sum_{j=1}^{p} k_{ij}(x) \; eps$$

= R * unvermeidbarer Fehler.

Beispiele zur Rückwärtsanalyse:

1) $y = x_1 + \ldots + x_n \quad , \quad x_i \in A$

Algorithmus: (n = 4) $y = ((x_1 + x_2) + x_3) + x_4$

$$\tilde{y} = ((x_1 \oplus x_2) \oplus x_3) \oplus x_4$$

(Satz 1.1 iii: $x_1 \oplus x_2 = (x_1 + x_2)\rho \quad , \; |\ell n\, \rho| \leq eps$ )

$$\tilde{y} = (((x_1 + x_2)\rho_1 + x_3)\rho_2 + x_4)\rho_3$$

$$= x_1\rho_1\rho_2\rho_3 + x_2\rho_1\rho_2\rho_3 + x_3\rho_2\rho_3 + x_4\rho_3$$

$$= \tilde{x}_1 + \tilde{x}_2 + \tilde{x}_3 + \tilde{x}_4$$

Schreibweise: $\rho_1 \ldots \rho_n = \rho^n \quad , \quad |\ell n\, \rho_i| \leq eps$

Rechenregeln: 1) $|\ell n\, \rho^n| \leq n \; eps$

2) $\rho^p \rho^q = \rho^{p+q}$

3) $\frac{\rho^p}{\rho^q} = \rho^{p\bar{+}q}$ , speziell $\rho = \frac{1}{\rho}$

4) $\rho - \rho \neq 0$

5) $|\rho^n - 1| \leq n\,\text{eps} + \underbrace{\text{Ausdruck in } (\text{eps})^2}_{O(\text{eps})^2}$

Beweis: $-n\,\text{eps} \leq \ell n\, \rho^n \leq n\,\text{eps}$

$\Leftrightarrow \quad e^{-n\,\text{eps}} \leq \rho^n \leq e^{n\,\text{eps}}$

$\Leftrightarrow \quad 1 - n\,\text{eps} + O(\text{eps}^2) \leq \rho^n \leq 1 + n\,\text{eps} + O(\text{eps}^2)$

$$\tilde{y} = (((x_1 + x_2)\rho + x_3)\rho + x_4)\rho$$

$$= x_1\,\rho^3 + x_2\,\rho^3 + x_3\,\rho^2 + x_4\rho$$

Für beliebiges $n$ gilt:

$$\tilde{y} = \sum_{i=1}^{n} x_i\,\rho^{n-i+1} = \sum_{i=1}^{n} \tilde{x}_i$$

mit $\left|\frac{\tilde{x}_i - x_i}{x_i}\right| \leq |\rho^{n-i+1} - 1| \leq (n - i+1)\,\text{eps} + \ldots$

Satz 3.2: Seien $a, \ell_1, \ldots, \ell_n$ , $r_1, \ldots, r_{n-1} \in A$, $r_n$ sei das Maschinenresultat für

$$\left( a - \sum_{k=1}^{n-1} \ell_k\, r_k \right) \Big/ \ell_n$$

durch den Algorithmus (n = 4)

$$(((a - \ell_1 r_1) - \ell_2 r_2) - \ell_3 r_3) / \ell_4$$

Dann gilt:

$$a = \sum_{k=1}^{n} \ell_k r_k \rho^k \quad .$$

Beweis: (n = 4)

$$r_4 = (((a-\ell_1 r_1 \rho)\rho - \ell_2 r_2 \rho)\rho - \ell_3 r_3 \rho)\rho \; / \ell_4 \rho$$

$$\Leftrightarrow \quad \rho^2 r_4 \ell_4 + \rho \ell_3 r_3 = (a - \ell_1 r_1 \rho)\rho - \ell_2 r_2 \rho)\rho$$

$$\vdots$$

$$\Leftrightarrow \quad \rho^4 r_4 \ell_4 + \rho^3 \ell_3 r_3 + \rho^2 \ell_2 r_2 + \rho \ell_1 r_1 = a$$

Beispiel: Komplexe Multiplikation

$$a = a_1 + i\,a_2 \qquad b = b_1 + i\,b_2$$

$$\underbrace{a\,b = a_1 b_1 - a_2 b_2}_{y_1} + i \underbrace{(a_1 b_2 + a_2 b_1)}_{y_2}$$

Schnelle komplexe Multiplikation:

$$x_1 = a_1 + a_2 \quad , \quad x_2 = b_1 + b_2 \quad , \quad x_3 = b_1 - b_2$$

$$z_1 = x_1 b_1 \quad , \quad z_2 = a_2 x_2 \quad , \quad z_3 = a_1 x_3$$

$$y_1 = z_1 - z_2 \qquad y_2 = z_1 - z_3$$

Rückwärtsabschätzung:

$$y_1 = x_1 b_1 - a_2 x_2$$

$$= (a_1 + a_2) b_1 - a_2 (b_1 + b_2)$$

$$\tilde{y}_1 = ((a_1 + a_2)\rho\ b_1 \rho - a_2 \rho (b_1 + b_2)\rho)\rho$$

$$= a_1 b_1 \rho^3 - a_2 b_2 \rho^3 + (\rho^3 - \rho^3) a_2 b_1$$

$$= \tilde{a}_1 b_1 - \tilde{a}_2 b_2 + (\rho^3 - \rho^3) a_2 b_1$$

Der schnelle Algorithmus ist offenbar nicht rückwärts stabil.

Zahlenbeispiel: $m = 3$ , $d = 10$

$$a_1 = 1.5_{10}{}^{-3} \qquad b_1 = 1.01$$

$$a_2 = 1.01 \qquad b_2 = 1.0_{10}{}^{-3}$$

$$y_1 = 5.05_{10}{}^{-4}$$

$$\tilde{y}_1 (\text{Standard}) = 5.1_{10}{}^{-4}$$

$$\tilde{y}_1 (\text{schnell}) = 0 \quad .$$

PRAKTISCHE LINEARE ALGEBRA

## § 4 Die L-R - Zerlegung

Sei $A = (a_{ik})$ eine (n,n) - Matrix

A heißt linke Dreiecksmatrix falls $a_{ik} = 0,\ k > i$

A " rechte " " " $a_{ik} = 0,\ k < i$

Das Ziel dieses Paragraphen ist die Darstellung von $A = L\,R$, wobei L eine linke Dreiecksmatrix und R eine rechte Dreiecksmatrix ist.

Hierfür werden Elementarmatrizen $L_j$ benötigt.

$$L_j = \begin{pmatrix} & & & \downarrow j\text{-te Spalte} & & \\ 1 & & & & & \\ & \ddots & & & & \\ & & \ddots & & & \\ & & & 1 & & \leftarrow j\text{-te Zeile} \\ & & & -\ell_{j+1,j} & \ddots & \\ & & & \vdots & & \ddots \\ & & & -\ell_{n,j} & & 1 \end{pmatrix}$$

Rechenregeln für Elementarmatrizen:

1) Sei $A = \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix}$ mit $a_i = (a_{i1},\ldots,a_{in})$ .

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

Wirkung = Subtraktion des $\ell_{i,j}$ - fachen von $a_j$ von Zeile $a_i$, $i > j$.

2) $L_j^{-1} =$

$$\begin{pmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ & & \ell_{j+1,j} & \ddots \\ & & \vdots & \ddots \\ & & \ell_{n,j} & 1 \end{pmatrix}$$

Macht Anwendung von $L_j$ rückgängig.

3) $L_j \cdot L_k =$ $j < k$

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

j-te Spalte ↓ — k-te Spalte ↓ — ← j-te Zeile — ← k-te Zeile

Entsteht durch „Überlagerung" von $L_j$, $L_k$ im links unteren Dreieck.

Weiter brauchen wir Permutationsmatrizen.

Seien $e_i = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \leftarrow$ i-te Zeile die kanonischen Einheitsvektoren.

$(i_1,\ldots,i_n)$ eine Permutation von $(1,\ldots,n)$

$P = \begin{pmatrix} e_{i_1}^T \\ \cdot \\ \cdot \\ \cdot \\ e_{i_n}^T \end{pmatrix}$ ist die Permutationsmatrix für die obige Permutation, d.h.

$$P \begin{pmatrix} 1 \\ \vdots \\ n \end{pmatrix} = \begin{pmatrix} i_1 \\ \vdots \\ i_n \end{pmatrix}$$

Rechenregeln für Permutationsmatrizen:

1) $PA = \begin{pmatrix} a_{i_1} \\ \cdot \\ \cdot \\ \cdot \\ a_{i_n} \end{pmatrix}$ Anwendung der Permutation auf die Zeilen.

2) $P^T P = I$ , $P^T = (e_{i_1},\ldots,e_{i_n})$

3) Sei $A = (a_1,\ldots,a_n)$

$$AP^T = (a_{i_1},\ldots,a_{i_n})$$

Anwendung der Permutation auf die Spalten.

$$= (Ae_{i_1},\ldots,Ae_{i_n})$$

4) Sei $P$ eine Permutationsmatrix, welche nur Zeilen $> j$ vertauscht. Dann ist $PL_j = L_j' P$, wobei $L_j'$ aus $L_j$ durch Vertauschen der nichttrivialen Elemente in Spalte $j$ (gemäß der Permutation von $P$) hervorgeht.

Zum Beweis schreiben wir

$$L_j = I + (0,\cdots,\underset{\substack{\uparrow\\ j\text{-te Spalte}}}{\ell},\cdots,0) \quad , \quad \ell = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ -\ell_{j+1,j} \\ \vdots \\ -\ell_{n,j} \end{pmatrix} \leftarrow j\text{-te Zeile}$$

$$\Rightarrow PL_j = P + (0,\cdots,\underset{\substack{\uparrow\\ j\text{-te Spalte}}}{\ell'},\cdots,0), \ \ell' = P\ell = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ -\ell'_{j+1,j} \\ \vdots \\ -\ell'_{n,j} \end{pmatrix} \leftarrow j\text{-te Zeile}$$

Rechtsmultiplikation mit $P^T$ vertauscht nur Spalten $> j$.

$$\Rightarrow PL_jP^T = PP^T + (0,\cdots,\ell',0,\cdots,0)$$

$$= I + (0,\cdots,\ell',0,\cdots,0)$$

$$= L_j$$

$\Rightarrow$ Beh.

Nun haben wir alle Hilfsmittel zur Berechnung der L-R-Zerlegung zur Verfügung.

Sei

$$A^{(1)} = \begin{pmatrix} a_{11}^{(1)} & \cdots & a_{1n}^{(1)} \\ \vdots & & \vdots \\ a_{n1}^{(1)} & \cdots & a_{nn}^{(1)} \end{pmatrix} = A$$

1. Schritt:

$$L_1 A^{(1)} = \begin{pmatrix} a_{11}^{(1)} & & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} \cdots & a_{2n}^{(2)} \\ \vdots & \vdots & \vdots \\ 0 & a_{n2}^{(2)} \cdots & a_{nn}^{(2)} \end{pmatrix}$$

mit

$$\ell_{i1} = \frac{a_{i1}^{(1)}}{a_{11}^{(1)}} \qquad i=2,\ldots,n$$

$$a_{11}^{(1)} \neq 0$$

$$a_{i,k}^{(2)} = a_{i,k}^{(1)} - \ell_{i1} \cdot a_{1,k}$$

$$i,k=2,\ldots,n$$

2. Schritt:

$$L_2 L_1 A = \begin{pmatrix} a_{11}^{(1)} & & & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ \vdots & 0 & a_{33}^{(3)} \cdots & a_{3n}^{(3)} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & a_{n3}^{(3)} \cdots & a_{nn}^{(3)} \end{pmatrix}$$

mit

$$\ell_{i2} = a_{i2}^{(2)} \Big/ a_{22}^{(2)} \qquad i = 3,\ldots,n \qquad a_{22}^{(2)} \neq 0$$

$$a_{i,k}^{(3)} = a_{i,k}^{(2)} - \ell_{i2}\, a_{2k}^{(2)} \qquad i,k = 3,\ldots,n$$

Vor dem j-ten Schritt:

$$L_{j-1}\cdots L_1 A = \begin{pmatrix} a_{11}^{(1)} & \cdots & & a_{1n}^{(1)} \\ & \ddots & & \vdots \\ & & a_{jj}^{(j)} & \cdots & a_{jn}^{(j)} \\ & \mathcal{O} & \vdots & & \vdots \\ & & a_{nj}^{(j)} & \cdots & a_{nn}^{(j)} \end{pmatrix}$$

j-ter Schritt:

$$L_j L_{j-1} \cdots L_1 A = \begin{pmatrix} a_{11}^{(1)} & & & & a_{1n}^{(1)} \\ & \ddots & & & \\ & & a_{jj}^{(j)} & \cdots & a_{jn}^{(j)} \\ & & 0 & a_{j+1,j+1}^{(j+1)} \cdots & a_{j+1,n}^{(j+1)} \\ & & \vdots & \vdots & \vdots \\ & & 0 & a_{n,j+1}^{(j+1)} \cdots & a_{nn}^{(j+1)} \end{pmatrix}$$

mit

$$\ell_{ij} = a_{ij}^{(j)} \Big/ a_{jj}^{(j)} \qquad i = j+1,\ldots,n \qquad a_{jj}^{(j)} \neq 0$$

$$a_{ik}^{(j+1)} = a_{ik}^{(j)} - \ell_{ij}\, a_{j,k}^{(j)} \qquad i,k=j+1,\ldots,n$$

Nach $n - 1$ Schritten:

$$L_{n-1} \cdots L_1 A = \begin{pmatrix} a_{11}^{(1)} & \cdots & a_{1n}^{(1)} \\ & \ddots & \vdots \\ & \mathbf{0} & \vdots \\ & & a_{nn}^{(n)} \end{pmatrix} = R$$

Nun invertieren wir die $L_j$ nacheinander und erhalten:

$$A = L_1^{-1} \cdots L_{n-1}^{-1} \cdot R = L\,R \quad ,$$

mit

$$L = \begin{pmatrix} 1 & & \\ \ell_{2,1} & \mathbf{0} & \\ \vdots & \ddots & \\ \ell_{n,1} & \cdots \ell_{n,n-1} & 1 \end{pmatrix}$$

Bis jetzt haben vorausgesetzt, daß das "Pivot" $a_{jj}^{(j)} \neq 0$.

Was tut man, falls für ein $j$ $a_{jj}^{(j)} = 0$ ?

$$L_{j-1} \cdots L_1 A = \begin{pmatrix} a_{11}^{(1)} & & & & a_{1n}^{(1)} \\ & \ddots & & & \\ & & a_{jj}^{(j)} & \cdots & a_{jn}^{(j)} \\ & & \vdots & & \vdots \\ & & a_{nj}^{(j)} & \cdots & a_{nn}^{(j)} \end{pmatrix}$$

1. Fall $a_{ij}^{(j)} = 0$ für $i = j,\ldots,n$ , dann $L_j = I$ Spalte = 0

2. Fall $\exists\ i \geq j \quad a_{ij}^{(j)} \neq 0$ , dann

multiplizieren wir von links mit $P_j$ .

$P_j$ ist die Permutationsmatrix zur Permutation

$$(i_1,\ldots,i_n) = (1\ ,\ \ldots\ ,\ \underset{\substack{\uparrow\\ j\text{-te Stelle}}}{i}\ ,\ \ldots\ ,\ \underset{\substack{\uparrow\\ i\text{-te Stelle}}}{j}\ ,\ \ldots\ ,\ n)$$

Wir erhalten also im allgemeinen Fall

$$L_{n-1}\ P_{n-1} \cdot \cdot \cdot L_2\ P_2\ L_1\ P_1\ A = R \quad .$$

Mit Hilfe der Vertauschungsrelation 4) von Permutations- und Elementarmatrizen führt dies auch zu einer LR - Zerlegung.

Sei der Einfachheit wegen $n = 4$.

$$L_3\ P_3\ L_2\ P_2\ L_1\ P_1\ A = R$$

Durch Anwendung 4) erhalten wir:

$$\Leftrightarrow \quad L_3\ P_3\ L_2\ L_1'\ P_2\ P_1\ A = R$$

$$\Leftrightarrow \quad L_3\ L_2'\ L_1''\ \underbrace{P_3\ P_2\ P_1}_{P}\ A = R$$

$$\Leftrightarrow \quad PA = \underbrace{L_1''^{-1}\ L_2'^{-1}\ L_3^{-1}}_{L}\ R$$

Wir fassen das Ergebnis zusammen in:

Satz 4.1: (über die L-R-Zerlegung)

Zu jeder (n,n)-Matrix A gibt es eine Permutationsmatrix P, eine linke Dreiecksmatrix L und eine rechte Dreiecksmatrix R, so daß

$$PA = L\,R$$

Wir schreiben nun ein"Programm" L R , das die L-R-Zerlegung durchführt.

```
Programm L R ;

comment    Berechnet die L-R-Zerlegung von  A = (a_ij)
           R  wird auf  A  überschrieben;

begin
for j=1 to n-1 do
    if  ∃ i≥j : a_ij≠0 then
        begin
        suche a_ij≠0; vertausche die Zeilen i,j;
        merke die Vertauschung;
        for i=j+1 to n do
            begin
            ℓ_ij=a_ij/a_jj;
            for k=j+1 to n do
                a_ik=a_ik-ℓ_ij a_jk;
            end i;
        end j;
end.
```

Die LR-Zerlegung von $A$ läßt sich auch direkt, d.h. ohne Bilden der Zwischenmatrix $A^{(j)}$ berechnen. Dieses Verfahren von Crout hatte für das Rechnen mit Bleistift und Papier praktische Bedeutung. Wir werden es nur benutzen, um die Rückwärtsanalyse bequem durchführen zu können.

Die Elemente $a_{ik}^{(j)}$, $\ell_{ij}$ können rekursiv folgendermaßen berechnet werden:

$$a_{ik}^{(j)} = a_{ik}^{(j-1)} - \ell_{i,j-1}\, a_{j-1,k}^{(j-1)} \qquad i,k \geq j$$

$$\ell_{ij} = a_{ij}^{(j)}/a_{jj}^{(j)} \qquad i \geq j$$

Wir lösen die Rekursionsformel für die $a_{ik}^{(j)}$ auf.

$$\begin{aligned} a_{ik}^{(j)} &= a_{ik}^{(j-1)} - \ell_{i,j-1}\, a_{j-1,k}^{(j-1)} \\ &= a_{ik}^{(j-2)} - \ell_{i,j-2}\, a_{j-2,k}^{(j-2)} - \ell_{i,j-1}\, a_{j-1,k}^{(j-1)} \\ &= \ldots \\ &= a_{ik}^{(1)} - \ell_{i1}\, a_{1k}^{(1)} - \ldots - \ell_{i,j-1}\, a_{j-1,k}^{(j-1)} \\ &= a_{ik} - \ell_{i1}\, r_{1k} - \ldots - \ell_{i,j-1}\, r_{j-1,k} \end{aligned}$$

Für $k \geq i = j$ bekommen wir

$$r_{ik} = (a_{ik} - \ell_{i1}\, r_{1k} - \ldots - \ell_{i,i-1}\, r_{i-1,k})/\ell_{ii}$$

Für $i \geq k = j$ ergibt sich aus der Formel für $\ell_{ij}$:

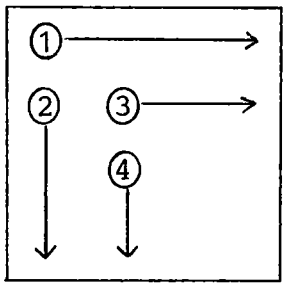
$$\ell_{ik} = a_{ik}^{(k)} / a_{kk}^{k}$$

$$= (a_{ik} - \ell_{i1}\, r_{1k} - \ldots - \ell_{i,k-1}\, r_{k-1,k}) / r_{kk}$$

Die Elemente der Matrix

$$\begin{matrix} r_{11} & \cdot & \cdot & \cdot & r_{1\,n} \\ \ell_{21} & & & & \cdot \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ \cdot & & & & \\ \ell_{n,1} & \cdots & \ell_{n-1,n} & & r_{n\,n} \end{matrix}$$

können also folgendermaßen berechnet werden:




1 Berechnung der 1. Zeile $r_{1k} = a_{1k}$ $k=1,\ldots,n$

2 Berechnung des Rests der 1. Spalte $\ell_{i1} = a_{i1}/r_{11}$ $i=2,\ldots,n$

3 Berechnung des Rests der 2. Zeile $r_{2k} = a_{2k} - \ell_{21}\, r_{1k}$ $r=2,\ldots,n$

4 Berechnung des Rests der 2. Spalte $\ell_{i2} = (a_{i2} - \ell_{i1}\, r_{12})/r_{22}$ $i=3,\ldots,n$

usw.

Satz 4.2: A sei eine (n,n)-Matrix von Maschinenzahlen. Das Programm LR in Maschinenarithmetik angewandt auf A liefert Matrizen $\tilde{L}$, $\tilde{R}$, $\tilde{P}$, so daß

$$\tilde{L}\,\tilde{R} = \tilde{P}\,A + F \qquad F = (f_{ik})$$

mit

$$|f_{ik}| \leq |\rho^m - 1| \sum_{j=1}^{m} |\tilde{\ell}_{ij}\,\tilde{r}_{jk}| \quad , \quad m = \text{Min } (i,k)$$

Beweis: Sei $A' = \tilde{P}A$. Dann läuft die Ausführung des LR - Programms auf den Crout'schen Algorithmus hinaus. Auf der Maschine werden also nach Satz 3.2 Größen $\tilde{r}_{ik}$, $\tilde{\ell}_{ik}$ berechnet mit

$$a'_{ik} = \sum_{j=1}^{m} \tilde{\ell}_{ij}\,\tilde{r}_{jk}\,\rho^j \quad , \quad m = \text{Min } (i,k) \, ,$$

$$= \sum_{j=1}^{m} \tilde{\ell}_{ij}\,\tilde{r}_{jk} + f_{ik} \quad ,$$

$$f_{ik} = \sum_{j=1}^{m} \tilde{\ell}_{ij}\,\tilde{r}_{jk}(\rho^j - 1) \quad . \qquad \square$$

Dieser Satz liefert eine Rückwärtsabschätzung. Der Algorithmus ist rückwärts stabil, wenn

$$\sum_{j=1}^{m} |\tilde{\ell}_{ij}\,\tilde{r}_{jk}|$$

nicht zu groß wird. Die $\tilde{\ell}_{ij}$ können wir kontrollieren: Wählen wir die Vertauschung in Spalte j so, daß nach der Vertauschung $|a_{jj}^{(j)}| \geq |a_{ij}^{(j)}|$, $i \geq j$ gilt (d.h. man vertauscht Zeile j mit einer Zeile i mit maximalem $|a_{ij}^{(j)}|$, $i \geq j$ ("Maximale Spaltenpivotsuche")), so gilt $|\tilde{\ell}_{ij}| \leq \rho$, und wir bekommen

$$|f_{ik}| \leq |\rho^m - 1|\rho \sum_{j=1}^{m} |\tilde{r}_{jk}| \quad .$$

Die $\tilde{r}_{jk}$ können zwar im Prinzip sehr groß werden, tun dies aber glücklicherweise meistens nicht. In jedem Fall kann man sie während der Rechnung leicht kontrollieren. In der Regel wird die Rundung monoton sein, d.h. aus $x \leq 1$ folgt $rd(x) \leq 1$. Dann gilt sogar $|\tilde{\ell}_{ij}| \leq 1$, und der Faktor $\rho$ entfällt.

## § 5 Lineare Gleichungssysteme.

$Ax = b$ $A$ sei eine $(n,m)$-Matrix.

Es sind nun folgende Fälle möglich:

1) $n = m$ , Rang $(A) = n$ $\Rightarrow$ $Ax = b$ ist eindeutig lösbar.

2) $n > m$ $Ax = b$ heißt überbestimmt.

in § 8 .

3) $n < m$ $Ax = b$ heißt unterbestimmt.

Wir beschäftigen uns in diesem Paragraphen nur mit Fall 1.

Wir bilden für $PA$ die L-R-Zerlegung und wenden $P$ auf das LGS an

$$PAx = Pb = b'$$

$$\Leftrightarrow \quad LRx = b'$$

$$\Leftrightarrow \quad Ly = b' \text{ und } Rx = y$$

Diese Gleichungssysteme sind sehr leicht lösbar.

Berechnung von $y$. Es gibt zwei Möglichkeiten:

a) Vorwärtseinsetzen: Nach der LR-Zerlegung bilde man

$$y_1 = b'_1$$

$$\text{for } i=2 \text{ to } n \text{ do } y_i = b'_i - \sum_{j=1}^{i-1} \ell_{ij}\, y_j$$

Die $\ell_{ik}$ speichert man zweckmäßigerweise unterhalb der Diagonalen von $A$.

b) $\underbrace{L_{n-1}\, P_{n-1} \cdots L_1\, P_1}_{L^{-1}\, P} A = R$ (siehe § 4) .

Damit gilt

$$Ly = Pb \qquad y = L^{-1}\, Pb = L_{n-1}\, P_{n-1} \cdots L_1\, P_1\, b$$

Man erhält also y, indem man die Operationen der LR-Zerlegung auf b (anstelle auf die Zeilen von A) anwendet. Dies kann simultan mit der LR-Zerlegung geschehen, wenn man das Programm LR auf die erweiterte Matrix (A,b) anwendet.

Berechnung von x.

Rückwärtseinsetzen:

$$x_n = y_n / r_{nn}$$

for j=n-1 downto 1 do

$$x_j = \left( y_j - \sum_{k=j+1}^{n} r_{jk}\, x_k \right) / r_{jj} \; ;$$

L R + Vorwärtseinsetzen + Rückwärtseinsetzen heißt (Gauß'sches) Eliminationsverfahren.

Satz 5.1: Das Gauß'sche Eliminationsverfahren mit Spaltenpivotsuche läßt sich genau dann durchführen, wenn $\det A \neq 0$, und liefert die eindeutig bestimmte Lösung von $Ax = b$ .

Beweis:

$$\det(A) = \det(L)\,\det(R) = r_{11} \cdots r_{nn} \neq 0 \quad . \qquad \blacksquare$$

Satz 5.2: A, b seien eine Matrix und ein Vektor von Maschinenzahlen. Für die auf der Maschine mit dem Eliminationsverfahren berechneten $\tilde{P}, \tilde{L}, \tilde{R}, \tilde{x}$ gilt

$$(\tilde{P}\,A + E)\tilde{x} = \tilde{P}\,b \,, \qquad \text{mit}$$

$$|e_{ik}| \leq 2|\rho^{n+1} - 1| \sum_{j=1}^{m} |\tilde{\ell}_{ij}\,\tilde{r}_{jk}| \,, \quad m = \text{Min}(i,k)$$

Beweis: Nach Satz 3.2 gilt

$$\sum_{j=1}^{i} \rho^{j}\,\tilde{\ell}_{ij}\,\tilde{y}_j = b'_i \quad \text{und} \quad \sum_{k=j}^{n} \tilde{r}_{jk}\,\tilde{x}_k\,\rho^{n-k+1} = \tilde{y}_j$$

$$\Rightarrow \quad \sum_{j} \sum_{k} \tilde{\ell}_{ij}\,\rho^{j-k+n+1}\,\tilde{x}_k\,\tilde{r}_{jk} = b'_i$$

$$\Rightarrow \quad \sum_{j,k} \tilde{\ell}_{ij}\,\tilde{r}_{jk}\,\tilde{x}_k - \sum_{j,k} (\rho^{j-k+n+1} - 1)\tilde{\ell}_{ij}\,\tilde{r}_{jk}\,\tilde{x}_k = b'_i$$

Nach Satz 4.2 gilt

$$\sum_{j} \tilde{\ell}_{ij}\,\tilde{r}_{jk} = a'_{ik} + f_{ik}$$

mit $\quad |f_{ik}| \leq |\rho^{n}-1| \sum_{j} |\tilde{\ell}_{ij}\,\tilde{r}_{jk}| \,,$

das heißt wir haben

$$e_{ik} = f_{ik} - \sum_{j=1}^{i} (\rho^{j-k+n-1} - 1)\tilde{\ell}_{ij}\,\tilde{r}_{jk}$$

$$\Rightarrow \qquad |e_{ik}| \le \left( |\rho^{n}-1| + |\rho^{n+1} - 1| \right) \sum |\tilde{\ell}_{ij}\,\tilde{r}_{ij}| \quad .$$

## § 6 Fehlerabschätzung bei linearen Gleichungssystemen

Wir betrachten das lineare Gleichungssystem $Ax = b$ und interessieren uns für folgende Fragen:

1. Wie wirken sich Fehler in $A$, $b$ auf $x$ aus?
2. Wie wirken sich die während der Lösung von $Ax = b$ auftretenden Rundungsfehler auf $x$ aus?

Wir werden als Fehlermaße Normen verwenden.

Definition 6.1: Eine (Vektor-) Norm in $\mathbb{C}^n$ ist eine Abbildung $x \to \|x\| \in \mathbb{R}^1$ mit folgenden Eigenschaften:

1. Definitheit: $\|x\| \geq 0$, $\|x\| = 0$ nur für $x = 0$.
2. Homogenität: $\|\alpha x\| = |\alpha| \, \|x\|$, $\alpha \in \mathbb{C}^1$
3. $\Delta$-Ungleichung: $\|x+y\| \leq \|x\| + \|y\|$.

Beispiele: Einige häufig benutzte Normen sind:

die Maximum-Norm $\|x\|_\infty = \operatorname{Max} |x_j|$,

die euklidische Norm $\|x\|_2 = \left( \sum_{j=1}^{n} x_j \overline{x}_j \right)^{1/2}$.

Satz 6.1: Seien $\|\cdot\|$, $\|\cdot\|'$ Normen in $\mathbb{C}^n$. Dann gibt es eine Konstante $C(n) \geq 1$, so daß für alle $x \in \mathbb{C}^n$

$$\frac{1}{C(n)} \|x\|' \leq \|x\| \leq C(n) \|x\|' .$$

Der Beweis für diesen Satz findet sich in allen angegebenen Lehrbüchern. Der Satz hat folgende Konsequenz: Ist $\|\cdot\|$ irgendeine

Norm, so sind folgende Aussagen gleichwertig:

(a) $\|x^{(k)} - x\| \to 0 \ , \quad k \to \infty$

(b) $x_i^{(k)} \to x_i \ , \quad k \to \infty \ , \quad i = 1,\ldots,n \ .$

Die komponentenweise Konvergenz in $\mathbb{C}^n$ ist also äquivalent zur Norm-Konvergenz.

Definition 6.2: Sei $A$ eine $(n,n)$-Matrix und $\|\cdot\|$ eine Vektor-Norm in $\mathbb{C}^n$. Wir setzen

$$\|A\| = \text{Max}\{\|Ax\| : \|x\| \leq 1\} = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|} \ .$$

Die Abbildung $A \to \|A\|$ heißt die der Vektor-Norm $x \to \|x\|$ zugeordnete Matrix-Norm.

Beispiele:

1) Die $\|\cdot\|_\infty$ zugeordnete Matrix-Norm ist die Zeilensummennorm

$$\|A\|_\infty = \max_i \sum_j |a_{ij}|$$

Denn es ist

$$\|A\|_\infty = \text{Max}\{\|Ax\|_\infty : \|x\|_\infty \leq 1\}$$

$$= \text{Max}\{\max_i |\sum_j a_{ij} x_j| : |x_j| \leq 1 \ , \quad j = 1,\ldots,n\}$$

$$\leq \max_i \sum_j |a_{ij}| \ .$$

Dieses Maximum werde für $i = i_o$ angenommen. Mit $x_j = \text{sgn } a_{i_o,j}$ gilt dann

$$\max_i \sum_j |a_{ij}| = \sum_j |a_{i_o j}| = \sum_j a_{i_o j} x_j = (Ax)_{i_o}$$

$$\leq \max \{ \|Ax\|_\infty : \|x\|_\infty \leq 1 \} = \|A\|_\infty .$$

2) Die $\|\cdot\|_2$ zugeordnete Matrix-Norm ist die Spektralnorm

$$\|A\|_2 = \left(\rho(A^*A)\right)^{1/2} , \quad A^* = \bar{A}^T ,$$

wobei $\rho(B)$ den Spektralradius von $B$ bedeutet:

$$\rho(B) = \max \{ |\lambda| : \lambda \text{ Eigenwert von } B \} .$$

(vgl. § 9).

Satz 6.2: Matrixnormen haben folgende leicht zu verifizierende Eigenschaften:

1) $A \to \|A\|$ ist eine Vektornorm in $\mathbb{C}^{n^2}$.

2) $\|AB\| \leq \|A\| \, \|B\|$, also auch $\|A^k\| \leq \|A\|^k$.

3) $\|I\| = 1$ für die Einheitsmatrix $I$.

4) $\|A\| = \text{Inf} \{K : \|Ax\| \leq K\|x\| \text{ für alle } x \}$.

Satz 6.3: Sei $B$ eine Matrix mit $\|B\| < 1$. Dann existiert $(I-B)^{-1}$, und es gilt:

1) $$(I-B)^{-1} = \sum_{k=0}^{\infty} B^k$$ (Neumann'sche Reihe)

2) $$\|(I-B)^{-1}\| \leq (1-\|B\|)^{-1} .$$

Beweis: Es gilt

$$\| \sum_{k=0}^{\infty} B^k \| \leq \sum_{k=0}^{\infty} \|B^k\| \leq \sum_{k=0}^{\infty} \|B\|^k = \frac{1}{1-\|B\|} < \infty .$$

Also konvergiert die Reihe komponentenweise im $\mathbb{C}^{n^2}$, und es ist

$$(I-B) \sum_{k=0}^{\infty} B^k = \sum_{k=0}^{\infty} B^k - \sum_{k=0}^{\infty} B^{k+1} = I .$$

■

Definition 6.3: Sei $A$ invertierbar. Als Kondition von $A$ (bezüglich einer Vektornorm $\|\cdot\|$) bezeichnen wir die Zahl

$$k(A) = \|A\| \, \|A^{-1}\| .$$

Satz 6.4: Sei $A$ invertierbar und $\Delta A$ eine Matrix mit

$$q = \frac{\|\Delta A\|}{\|A\|} k(A) < 1 .$$

Dann ist auch $A + \Delta A$ invertierbar, und es gilt

$$\|(A + \Delta A)^{-1}\| \leq \|A^{-1}\| / (1-q) .$$

Beweis: Es ist

$$A + \Delta A = A(I + A^{-1}\Delta A) = A(I + B) ,$$

$$\|B\| = \|A^{-1}\Delta A\| \leq \|A^{-1}\| \, \|\Delta A\| = \|A^{-1}\| \, \|A\| \frac{\|\Delta A\|}{\|A\|}$$

$$= q < 1 .$$

Also existiert nach Satz 6.3 die Inverse von $I+B$, und wir haben

$$\|(A+\Delta A)^{-1}\| = \|(I+B)^{-1}A^{-1}\| \leq \|A^{-1}\|/(1-q) \quad .$$

Satz 6.5: Sei $A$ invertierbar, und sei $x$ Lösung von $Ax = b$. Seien $\Delta A$, $\Delta b$ Störungen von $A$, $b$, und sei

$$q = k(A)\frac{\|\Delta A\|}{\|A\|} < 1 \quad .$$

Dann ist auch das gestörte System $(A+\Delta A)\tilde{x} = b+\Delta b$ eindeutig lösbar, und es gilt

$$\frac{\|\tilde{x}-x\|}{\|x\|} \leq \frac{k(A)}{1-q}\left\{\frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta b\|}{\|b\|}\right\}$$

Beweis: Nach Satz 6.4 ist $A+\Delta A$ invertierbar, das gestörte System also eindeutig lösbar. Schreiben wir $\tilde{x}=x+\Delta x$, so lautet dieses

$$(A+\Delta A)(x+\Delta x) = b+\Delta b \quad .$$

Subtraktion von $Ax = b$ und Umordnung ergibt

$$(A+\Delta A)\Delta x = \Delta b - \Delta A x \quad ,$$

nach Satz 6.4 also

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

oder

$$\frac{\|\Delta x\|}{\|x\|} \leq \frac{\|A^{-1}\|}{1-q} \left( \frac{\|\Delta b\|}{\|x\|} + \|\Delta A\| \right)$$

$$= \frac{\|A^{-1}\|}{1-q} \left( \frac{\|\Delta b\|}{\|b\|} \frac{\|b\|}{\|x\|} + \frac{\|\Delta A\|}{\|A\|} \|A\| \right) .$$

Mit $\|b\| / \|x\| = \|Ax\| / \|x\| \leq \|A\|$ folgt die Behauptung.

Dieser Satz sagt aus, daß $k(A)$ die Bedeutung eines Verstärkungsfaktors hat, wenn man die Fehler in der Norm mißt. $k(A)$ ist ein bequemeres (allerdings auch gröberes) Maß für die Kondition der Aufgabe $Ax = b$ als die Verstärkungsfaktoren aus § 2. Wir nennen die Aufgabe $Ax = b$ daher schlecht konditioniert, wenn $k(A) >> 1$.

Beispiel: Die Auswirkung schlechter Kondition kann man an folgendem Zahlenbeispiel sehen:

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 0.99 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad x = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$\Delta A = \begin{pmatrix} 0.01 & 0.01 \\ 0 & 0 \end{pmatrix}, \quad \Delta b = 0, \quad \tilde{x} = \begin{pmatrix} 200/101 \\ -100/101 \end{pmatrix} .$$

Obwohl der Fehler in $A$ bei 1% liegt, haben $x$, $\tilde{x}$ nichts mehr miteinander zu tun.

Zur Erklärung berechnen wir

$$\|A\|_\infty = 2,\ A^{-1} = \begin{pmatrix} 99 & -100 \\ -100 & 100 \end{pmatrix},\ \|A^{-1}\|_\infty = 200 \quad .$$

Es ist also $k(A) = 400$ und $q = 4$, so daß Satz 6.5 nicht anwendbar ist.

Geometrisch kann man dieses Beispiel leicht verstehen: Das System $Ax = b$ besteht aus den Gleichungen $a_i^T x = b_i$, $i = 1,2$, wobei die Vektoren $a_1$, $a_2$ beinahe die gleiche Richtung haben. $x$ ist also bestimmt als Schnittpunkt der beiden Geraden $a_i^T x = b_i$, $i = 1,2$, welche sich unter ganz kleinem Winkel schneiden. Damit genügt schon eine kleine Veränderung einer der beiden Geraden, um den Schnittpunkt ganz woanders hinfallen zu lassen.

Wir wollen nun eine Vorwärtsabschätzung für den Rundungsfehler beim Eliminationsverfahren angeben, wobei wir uns auf maximale Spaltenpivotsuche beschränken. Dann folgt aus Satz 5.2, daß die Maschine das System $(A+E)\tilde{x} = b$ löst mit

$$\|E\|_\infty \le 2|\rho^{n+1} - 1|\rho \sum_j \sum_k |\tilde{r}_{kj}|$$

$$\le 2|\rho^{n+1} - 1|\rho n \|\tilde{R}\|_\infty \quad .$$

Für $\tilde{x}$ gilt also nach Satz 6.5: Ist

$$q = \frac{\|E\|_\infty}{\|A\|_\infty} k(A) \le 2|\rho^{n+1} - 1|\rho n \frac{\|\tilde{R}\|_\infty}{\|A\|_\infty} k(A) < 1 \quad ,$$

so gilt

$$\|x - \tilde{x}\|_\infty / \|x\|_\infty \leq q/(1-q) \quad .$$

Wir wollen diese Abschätzung diskutieren. Sie ist praktisch nur von Bedeutung, wenn $q << 1$. Bis auf Terme der Ordnung $eps^2$ hat man

$$q = 2\, n(n+2)\, eps\, \frac{\|\tilde{R}\|_\infty}{\|A\|_\infty}\, k(A) \quad .$$

Während $\|\tilde{R}\|_\infty$ theoretisch viel größer als $\|A\|_\infty$ sein kann, stellt sich praktisch regelmäßig heraus, daß $\|\tilde{R}\|_\infty / \|A\|_\infty$ in der Nähe von 1 liegt. Auf jeden Fall läßt sich dieser Quotient leicht während der Rechnung berechnen ($\tilde{R}$ ist ja das Maschinenresultat!).

## § 7 Die Q-R Zerlegung

Definition: $Q$ sei eine $(n,m)$ komplexe Matrix $(n \geq m)$

$Q^* = \overline{Q}^T$ . | $Q$ heißt orthonormal, falls

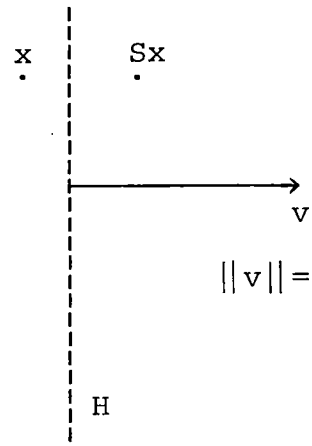
$Q^* Q = I$

Ist $Q$ orthonormal und $n = m$, dann heißt $Q$ unitär.

Beispiele:

1) Permutationsmatrizen.

2) Spiegelungen. $S$ sei die Spiegelung an der Hyperebene $H = v^{\perp}$




$Sx = x - 2(v^* x)v$

$= (I - 2 v v^*)x$

Die Matrix $vv^* = ((v_i v_k^*))$ ist dyadisches Produkt von $v$ und $v^*$

Es gilt $S = S^*$ und $S^2 = I$ , also ist $S$ unitär.

Wir wollen eine beliebige $(n,m)$-Matrix $A$ als Produkt $A = Q R$ schreiben, wobei $Q$ eine orthonormale Matrix und $R$ eine rechte Dreiecksmatrix ist.

Satz 7.1: Zu jeder (n,m)-Matrix A mit $n \geq m$ gibt es eine orthonormale (n,m) - Matrix Q und eine rechte Dreiecksmatrix R mit $A = Q R$.

Beweis: Schreibt man $A = (a_1,\ldots,a_m)$, $Q = (q_1,\ldots,q_m)$, so lautet $A = Q R$

$$a_1 = q_1 r_{11} ,$$
$$\cdots$$
$$a_j = q_1 r_{1j} + q_2 r_{2j} + \cdots + q_j r_{jj} ,$$
$$\cdots$$

Ist $a_1 = 0$, so wählen wir $q_1$ mit $\|q_1\| = 1$ beliebig und $r_{11} = 0$. Andernfalls wählen wir $r_{11} = \|a_1\|$ und $q_1 = a_1 / r_{11}$.

Seien $q_1,\ldots,q_{j-1}$ schon bestimmt. Dann wählen wir

$$r_{ij} = (a_j,q_i) \quad , \quad i = 1,\ldots,j-1$$

und setzen

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

Nach Wahl der $r_{ij}$ ist $\hat{q}_j$ orthogonal zu $q_1,\ldots q_{j-1}$. Ist $\hat{q}_j = 0$, so wählen wir für $q_j$ einen beliebigen zu $q_1,\ldots,q_{j-1}$ orthogonalen Vektor und setzen $r_{jj} = 0$. Andernfalls wählen wir $r_{jj} = \|\hat{q}_j\|$ und $\hat{q}_j = q_j / r_{jj}$. ∎

Das im Beweis benutzte Verfahren ist konstruktiv und kann im Prinzip zur Berechnung der QR - Zerlegung benutzt werden. Es treten aber Ungenauigkeiten auf, wenn $\|\hat{q}_j\| \ll \|a_j\|$. Dann treten bei der Berechnung von $\hat{q}_j$ Auslöschungen auf, so daß $\hat{q}_j$ großen relativen Fehler hat. Dann stimmt zwar die j-te Spalte von $A = QR$ noch, aber $q_j$ ist nicht mehr sehr gut orthogonal

zu $q_1,\ldots,q_{j-1}$, d.h. Q ist nicht mehr gut orthonormal.

Die praktische Berechnung der QR - Zerlegung geschieht durch das Householder-Verfahren.

Wir beschreiben es für reelles $A = (a_1,\ldots,a_m)$ .

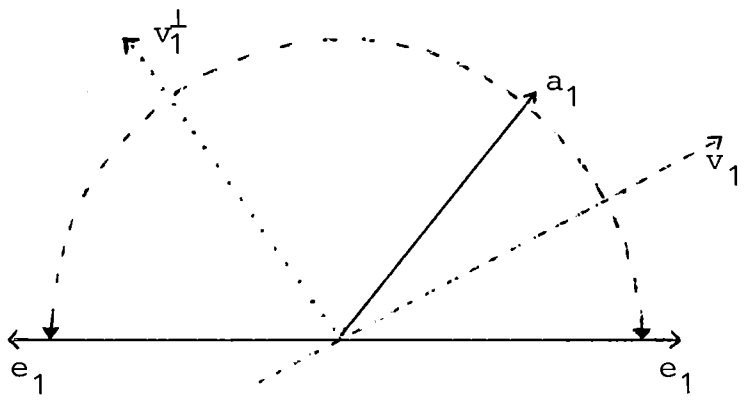
Die Idee des Householder-Verfahrens ist:

Finde Spiegelungen $S_1,\ldots,S_m$ , so daß

$$S_{j-1}\cdots S_1 A = \begin{pmatrix} a_{11}^{(2)} & & & a_{1m}^{(2)} \\ 0 & a_{22}^{(3)} & & a_{2m}^{(3)} \\ \vdots & \ddots & & \\ & 0 & a_{jj}^{(j+1)} \cdots & a_{jm}^{(j+1)} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & a_{nj}^{(j+1)} & a_{nm}^{(j+1)} \end{pmatrix}$$

1. Schritt des Householder-Verfahrens:

   Finde $S_1 = I - 2v_1 v_1^*$, so daß $S_1 a_1$ ein Vielfaches von $e_1$ wird.




   $v_1$ wird folgendermaßen berechnet:

Wir setzen zunächst

$$\hat{v}_1 = a_1 + \alpha_1 e_1 \quad , \quad \alpha_1 = \|a_1\| \operatorname{sgn}(a_{11}) \quad .$$

Das Vorzeichen von $\alpha_1$ ist so gewählt, daß bei der Berechnung der ersten Komponente von $\hat{v}_1$ keine Auslöschung auftritt. Ist $\hat{v}_1 = 0$ (dies kann nur für $a_1 = 0$ passieren), so setzen wir $v_1 = 0$, also $S_1 = I$ (d.h. der erste Schritt unterbleibt). Andernfalls setzen wir $\beta_1 = \|\hat{v}_1\|$ und $v_1 = \hat{v}_1 / \beta_1$. Aufgrund der Zeichnung ist klär, daß $S_1 a_1 = -\alpha_1 e_1$ sein muß. Dies bestätigt man leicht mit Hilfe von

$$\beta_1^2 = \|a_1 + \alpha_1 e_1\|^2 = \|a_1\|^2 + 2\alpha_1 a_{11} + \alpha_1^2$$

$$= \alpha_1^2 + 2\alpha_1 a_{11} + \alpha_1^2 = 2\alpha_1(\alpha_1 + a_{11}) \quad .$$

Die weiteren Spalten von $S_1 A$ ergeben sich als

$$S_1 a_k = a_k - 2(v_1, a_k) v_1 \quad , \quad k = 2,\ldots,m \; .$$

Das Überschreiben der Spalten $2,\ldots,m$ der Matrix $A$ mit den entsprechenden Spalten von $S_1 A$ sieht also folgendermaßen aus:

$$\alpha_1 = (a_{11}^2 + \cdots + a_{n1}^2)^{1/2} \operatorname{sgn}(a_{11}) \quad ,$$

$$\beta_1 = 2\alpha_1(\alpha_1 + a_{11}) \quad ,$$

$$v_{i1} = \begin{cases} (a_{11} + \alpha_1)/\beta_1 & , \quad i = 1, \\ a_{i1} / \beta_1 & , \quad i = 2,\ldots,n \quad , \end{cases}$$

$$a_{ik} = a_{ik} - 2(v_1, a_k) v_{i1} \quad , \quad i = 1,\ldots,n, \quad k = 2,\ldots,m \quad .$$

Der zweite Schritt verläuft genau so wie der erste, wobei aber nur die rechts untere $(n-1,m-1)$ - Teilmatrix von $S_1 A$ behandelt wird. Man setzt also

$$S_2 = \left(\begin{array}{c|c} 1 & 0 \\ \hline 0 & I-2v_2v_2^* \end{array}\right), \quad v_2 \in \mathbb{R}^{n-1}.$$

Nach $m$ Schritten des Householder-Verfahrens haben wir folgenden Zustand:

$$\underbrace{S_m \cdots S_1}_{Q^*} A = \begin{pmatrix} -\alpha_1 & a_{12}^{(2)} & \cdots & a_{1m}^{(2)} \\ & -\alpha_2 & a_{23}^{(3)} & a_{2m}^{(3)} \\ & & \ddots & \vdots \\ & 0 & & a_{m-1,m}^{(m)} \\ & & & -\alpha_m \\ 0 & & 0 & 0 \end{pmatrix} = \begin{pmatrix} R \\ 0 \end{pmatrix}.$$

$R$ ist eine rechte $(m,m)$ Dreiecksmatrix und

$Q$ ist eine unitäre $(n,n)$ - Matrix.

Die QR - Zerlegung bekommt man nun in der Form

$$A = Q\begin{pmatrix} R \\ O \end{pmatrix} = Q_m\, R$$

$Q_m$ enthält die ersten $m$ Spalten von $Q$ .

Da die $S_i$ symmetrisch sind gilt:

$$Q = S_1 \ldots S_m$$

Wir schreiben nun ein Programm QR, welches die QR - Zerlegung einer (n,n) - Matrix A herstellt. Nach Ablauf enthält A oberhalb der Diagonalen die Matrix R (ohne Diagonale), unterhalb und auf der Diagonalen die Vektoren $v_1,\ldots,v_m$ . Die Diagonale von R wird auf den Vektor $\alpha$ geschrieben.

```
for j=1 to m do
    begin
    α_j=( Σ_{i=j}^{n} (a_ij)^2 )^{1/2} sign a_jj;
    if α_j≠0 then                          bei α_j = 0 wird nichts getan
        begin
        β = 1/sqrt(2α_j(α_j+a_jj));
        a_jj=(a_jj+α_j)*β;
        for i=j+1 to n do a_ij=β * a_ij;    } v_j wird in Spalte j in
                                              die Zeilen j,...,n
                                              geschrieben
        for k=j+1 to m do
            begin                           } k-te Spalte von S_{j-1}...S_1 A
            γ=2 Σ_{i=j}^{n} a_ij a_ik;        wird mit entsprechender
            for i=j to n do a_ik=a_ik-γa_ij   Spalte von S_j...S_1 A
                                              überschrieben
            end k;
        end if;
    end.
```

Wir wollen nun die Matrix $Q$ auf einen Vektor $x$ anwenden. Dies tun wir, indem wir die $S_j$ nacheinander auf $x$ anwenden.

$$S_j = \left(\begin{array}{c|c} I_{j-1} & 0 \\ \hline 0 & I - 2\, v_j\, v_j^* \end{array}\right) \qquad x = \begin{pmatrix} x_{j-1} \\ \vdots \\ x_{n-j+1} \end{pmatrix}$$

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

Zur Bildung von $S_j\, x$ brauchen wir $2(n-j) + O(1)$ Rechenoperationen.

Um nun $Qx = S_1 \ldots S_m\, x$ zu bilden, brauchen wir

$$2 \sum_{j=1}^{m} (n-j) \sim 2nm - m^2 = m(2n - m) \leq n^2$$

Rechenoperationen. Diese Berechnung von $Qx$ ist also nicht aufwendiger als die Anwendung der $(n,n)$-Matrix $Q$ auf $x$ in der üblichen Weise. Man verzichtet daher auf die Berechnung von $Q$ und begnügt sich mit den Faktoren $S_1, \ldots, S_m$ bzw. mit den Vektoren $v_1, \ldots, v_m$.

## § 8 UNTER- UND ÜBERBESTIMMTE SYSTEME

Sei $A$ eine (komplexe) $(n,m)$-Matrix, $b \in \mathbb{C}^n$. Für $n \neq m$ hat $Ax = b$ in der Regel keine eindeutig bestimmte Lösung. Wir führen eine verallgemeinerte Lösung ein und beschreiben deren Berechnung.

Zunächst einige Bezeichnungen. Sei $\ker(A) = \{ x \in \mathbb{C}^m : Ax = O\}$, $\text{range}(A) = \{y \in \mathbb{C}^n : \exists\, x \text{ mit } y = Ax\}$. Für Teilmengen $U, V, W \subset \mathbb{C}^n$ schreiben wir $W = U \oplus V$ (orthogonale Summe), wenn $U \perp V$ und $W = U + V$.

SATZ 8.1: Für jede komplexe $(n,m)$ - Matrix gilt

$$\mathbb{C}^n = \ker(A^*) \oplus \text{range}(A) \quad .$$

BEWEIS: Ist $A^*x = O$ und $y = Az$, so ist

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

also $\ker(A^*) \perp \text{range}(A)$. Wäre $\ker(A^*) + \text{range}(A)$ echt enthalten in $\mathbb{C}^n$, so gäbe es $y \in \mathbb{C}^n$ mit $y \perp \ker(A^*)$, $\perp \text{range}(A)$. Aus $y \perp \ker(A^*)$ folgt aber nach dem Alternativsatz der linearen Algebra, daß $y = Ax$ lösbar, also $y \in \text{range}(A)$ ist. Wegen $y \perp \text{range}(A)$ ist $y = O$. ∎

Für $n \neq m$ können zwei Fälle auftreten. $Ax = b$ kann unlösbar sein (typisch für $n > m$: überbestimmt), oder es hat viele Lösungen (typisch für $n < m$: unterbestimmt). Man schwächt

zunächst den Begriff der Lösbarkeit ab, indem man anstelle von $Ax - b = 0$ nur noch verlangt, daß $\|Ax - b\|$ möglichst klein ist. Wir betrachten nur den Fall der euklidischen Norm.

SATZ 8.2: $x_o$ minimiert $\|Ax - b\|_2$ genau dann, wenn

$$A^*A\,x_o = A^*b \quad .$$

BEWEIS: Nach Satz 8.1 gilt $b = b_1 + b_2$ mit $b_1 \in \text{range}(A)$, $b_2 \in \ker(A^*)$. Dann ist $Ax - b_1 \perp b_2$ und damit

$$\|Ax-b\|_2^2 = \|(Ax-b_1) - b_2\|_2^2 = \|Ax-b_1\|^2 + \|b_2\|^2 \quad .$$

$\|Ax-b\|_2$ ist also minimal genau dann, wenn $Ax = b_1$, oder $Ax - b = -b_2$, d.h. wenn $Ax - b \in \ker(A^*)$ oder $A^*(Ax - b) = 0$.

■

BEMERKUNGEN:

1) $A^*Ax = A^*b$ heißt das Normalgleichungssystem für $Ax = b$. Die Berechnung von $x_o$ aus Satz 8.2 heißt "Methode der kleinsten Quadrate", $x_o$ Kleinste-Quadrate-Lösung.

2) Die Normalgleichungen sind immer lösbar. Denn

$$A^*b \perp \ker(A^*A) \quad ,$$

weil $\ker(A) = \ker(A^*A)$ und wegen Satz 8.1.

Eine Kleinste-Quadrate-Lösung $x_o$ existiert zwar immer, braucht aber nicht eindeutig zu sein. Um auch noch Eindeutigkeit zu erzwingen, wählt man unter allen Kleinste -

Quadrate - Lösungen diejenigen mit kleinster Norm. Diese nennt man die (Moore - Penrose) verallgemeinerte Lösung $x_{MP}$ von $Ax = b$.

SATZ 8.3: Die verallgemeinerte Lösung $x_{MP}$ ist eindeutig bestimmt durch

$$A^*Ax_{MP} = A^*b \quad , \quad x_{MP} \in \text{range}(A^*) \quad .$$

BEWEIS: Sei $x_o$ eine beliebige Kleinste - Quadrate - Lösung, also $A^*Ax_o = A^*b$ nach Satz 8.2. Nach Satz 8.1 ist $x_o = x_1 + x_2$ mit $x_1 \in \text{range}(A^*)$ und $Ax_2 = 0$. Offenbar ist auch $x_1$ Kleinste-Quadrate-Lösung. Es gibt also stets Kleinste-Quadrate-Lösungen in range $(A^*)$. Ist $x_1$ eine solche, so läßt sich jede weitere Kleinste-Quadrate-Lösung $x$ in der Form

$$x = x_1 + y$$

mit $A^*Ay = 0$ oder $Ay = 0$ schreiben. Wegen $x_1 \perp y$ ist

$$\|x\|^2 = \|x_1\|_2^2 + \|y\|_2^2 \geq \|x_1\|_2^2 \quad .$$

Also ist $x_{MP} = x_1$ das eindeutige Minimum von $\|x\|$ unter allen Lösungen von $A^*Ax = A^*b$.

■

Die Zuordnung $b \to x_{MP}$ ist offenbar linear und wird durch eine Matrix $A^+$ vermittelt, d.h.

$$x = A^+b \quad .$$

$A^+$ heißt die (Moore-Penrose) verallgemeinerte Inverse von $A$. Hat $A$ vollen Rang und ist $n \geq m$, so sind die Normalgleichungen eindeutig lösbar, und es gilt

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

Für die Berechnung verallgemeinerter Lösungen nehmen wir an, daß $A$ maximalen Rang hat. Wir betrachten zunächst den überbestimmten Fall, also $n \geq m$. Dann sind die Normalgleichungen $A^*Ax = A^*b$ eindeutig lösbar. Man könnte also $A^*A$ bilden und dann die Normalgleichungen durch das Eliminationsverfahren lösen. Das ist aber aus zwei Gründen nicht empfehlenswert.

(a) Die Berechnung von $A^*A$ ist nicht unproblematisch.

BEISPIEL:

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

Ist $\varepsilon < \sqrt{eps}$, so ist das Maschinenresultat für $A^*A$ die Einheitsmatrix.

(b) Die Kondition der Normalgleichungen ist unter Umständen sehr schlecht. Im Falle $n = m$ ist etwa

$$\|A\|_2 = (\rho(A^*A))^{1/2}$$

$$\|A^*A\|_2 = (\rho((A^*A)^2))^{1/2} = \rho(A^*A) = \|A\|_2^2 \quad ,$$

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

Ist also $k(A) >> 1$, so ist dies erst recht der Fall für $k(A^*A)$.

Aus diesen Gründen möchte man die verallgemeinerte Lösung ohne Bilden von $A^*A$ berechnen. Dies kann durch die QR - Zerlegung geschehen. Mit $A = QR$ wird aus den Normalgleichungen

$$R^*Q^*Q\,R\,x_{MP} = R^*Q^*b \quad .$$

Wegen $Q^*Q = I$ lautet dies nach Kürzen eines Faktors $R^*$

$$Rx_{MP} = Q^*b \quad .$$

Dies können wir auch in der Form $A^+ = R^{-1}Q^*$ schreiben.

Da $A$ maximalen Rang hat, ist $R$ nichtsingulär. Für $n = m$ gilt

$$\|A\|_2 = (\rho(A^*A))^{1/2} = (\rho(R^*R))^{1/2} = \|R\|_2$$

und entsprechend für $A^{-1}$, $R^{-1}$. Also $k(A) = k(R)$, so daß keine Quadrierung der Kondition auftritt.

Im unterbestimmten Fall, also $n \leq m$, verwendet man die QR - Zerlegung von $A^*$, also $A^* = QR$. Die Beziehungen von Satz 8.3 nehmen dann die Form

$$Q\,R\,R^*\,Q^*\,x_{MP} = Q\,R\,b \quad , \quad x_{MP} = Q\,R\,y$$

an. Linksmultiplikation mit $R^{-1}\,Q^*$ ergibt

$$R^*R\,y = b \quad .$$

Man setzt nun $z = Ry$ und löst dann

$$R^* z = b \quad , \quad x_{MP} = Q z \quad .$$

Anders ausgedrückt: $A^+ = Q(R^*)^{-1}$

## § 9 Eigenwertprobleme bei Matrizen

Eigenwertprobleme sind neben den linearen Gleichungssystemen die zweite Grundaufgabe der numerischen linearen Algebra. Wir wollen in diesem Abschnitt zunächst einige Tatsachen zusammenstellen.

Definition 9.1: Sei A eine komplexe (n,n)-Matrix. $\lambda \in \mathbb{C}$ heißt Eigenwert (EW) von A, wenn es $x \in \mathbb{C}^n$, $x \neq 0$ gibt mit $Ax = \lambda x$. x heißt dann Eigenvektor (EV) von A zum EW $\lambda$.

Als notwendige und hinreichende Bedingung dafür, daß $\lambda$ EW von A ist, hat man also

$$\varphi(\lambda) = \det(\lambda I - A) = 0.$$

$\varphi(\lambda)$ heißt "charakteristisches Polynom von A". $\varphi(\lambda)$ ist ein Polynom genau vom Grade n in $\lambda$:

$$\varphi(\lambda) = \lambda^n - \left(\sum_{i=1}^{n} a_{ii}\right)\lambda^{n-1} + \ldots + (-1)^n \det(A).$$

Definition 9.2: Jedem EW $\lambda$ von A ordnen wir zwei Vielfachheiten zu:

Seine algebraische Vielfachheit $\sigma(\lambda)$ = Vielfachheit von x als Nullstelle von $\varphi(\lambda)$.

Seine geometrische Vielfachheit $\rho(\lambda)$ = Anzahl der linear unabhängigen EV zu $\lambda$.

Sind also $\lambda_1, \ldots \lambda_m$ die verschiedenen EW von A und sind $\sigma_k = \sigma(\lambda_k)$ ihre algebraischen Vielfachheiten, so gilt

$$\varphi(\lambda) = \prod_{k=1}^{m} (\lambda - \lambda_k)^{\sigma_k} \quad , \quad \sum_{k=1}^{m} \sigma_k = n \; .$$

Für die geometrischen Vielfachheiten $\rho_k = \rho(\lambda_k)$ gilt nun

$$\sum_{k=1}^{m} \rho_k \leq n.$$

Definition 9.3: Die (n,n)-Matrizen A,B heißen ähnlich, wenn es eine nichtsinguläre (n,n)-Matrix X gibt mit

$$A = XBX^{-1}.$$

Satz 9.1: Seien A,B ähnlich. Dann haben A,B die gleichen Eigenwerte mit übereinstimmenden algebraischen und geometrischen Vielfachheiten.

Beweis: Sei $A = XBX^{-1}$. Dann ist

$$\begin{aligned} \det(\lambda I - A) &= \det(X(\lambda I - B)X^{-1}) \\ &= \det(X)\det(\lambda I - B)\det(X^{-1}) \\ &= \det(\lambda I - B). \end{aligned}$$

Die charakteristischen Polynome stimmen also überein, also auch die Eigenwerte samt ihrer algebraischen Vielfachheiten. Ist nun $\lambda$ ein EW von A der geometrischen Vielfachheit $\rho$, so gibt es $\rho$ EV $x_1,\ldots,x_\rho$ zu $\lambda$, also

$$Ax_k = \lambda x_k \ , \ k = 1,\ldots,\rho.$$

Mit $y_k = X^{-1}x_k$ gilt

$$\begin{aligned} B\,y_k &= X^{-1}AX\,X^{-1}x_k = X^{-1}Ax_k = \lambda X^{-1}x_k \\ &= \lambda y_k \ , \end{aligned}$$

also sind $Y_1,\ldots,Y_\rho$ l.u. EV von B zu $\lambda$. Die geometrische Vielfachheit von $\lambda$ als EW von A ist also nicht größer als die geometrische Vielfachheit von $\lambda$ als EW von B. Da die Voraussetzungen in A,B symmetrisch sind, müssen die geometrischen Vielfachheiten also übereinstimmen.

Beispiele:

1) $A = \begin{pmatrix} d_1 & & 0 \\ & \ddots & \\ 0 & & d_n \end{pmatrix}$. Dann ist $\det(\lambda I - A) = \prod_{k=1}^{n} (\lambda - d_k)$, also

$\lambda_k = d_k$ mit EV $e_k$ = k-tem Einheitsvektor. Offenbar ist

$\rho(\lambda_k) = \sigma(\lambda_k)$ , $k = 1,\ldots,n$.

2) $A = XDX^{-1}$ mit $D = \begin{pmatrix} d_1 & & \\ & \ddots & \\ & & d_n \end{pmatrix}$. Nach Satz 9.1 und Beispiel 1) ist $\lambda_k = d_k$ , $\rho(\lambda_k) = \sigma(\lambda_k)$ , $k=1,\ldots,n$. Dem Beweis von Satz 9.1 und Beispiel 1) entnimmt man die EV $x_k = Xe_k$ = k-te Spalte von X. Matrizen dieser Art, welche also ähnlich zu einer Diagonalmatrix sind, nennt man diagonalisierbar.

3) $J(\mu) = \begin{pmatrix} \mu & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & \mu \end{pmatrix}$ Es ist $\det(\lambda I - J(\mu)) = (\lambda-\mu)^n$, also ist $\lambda=\mu$ der einzige EW von $J(\mu)$, und er hat die algebraische Vielfachheit n. Ist x ein EV zum EW $\mu$ von $J(\mu)$,

$$(J(\mu) - \mu I)\, x = \begin{pmatrix} 0 & 1 & & \\ & 0 & 1 & \\ & & \ddots & \ddots \\ & & & 1 \\ & & & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} x_2 \\ x_3 \\ \vdots \\ x_n \\ 0 \end{pmatrix} = 0 ,$$

und der einzige l.u. EV ist $x=e_1$. Also ist die geometrische Vielfachheit von $\mu$ verschieden von seiner algebraischen Vielfachheit, nämlich 1.

Bis auf Ähnlichkeiten sind die Matrizen $J(\mu)$ bereits die allgemeinsten Matrizen, soweit das Eigenwertproblem betroffen ist. Es gilt nämlich der

Satz 9.2: (Jordan'sche Normalform). Jede komplexe (n,n)-Matrix ist ähnlich einer Matrix

$$J = \begin{pmatrix} J_1 & & \\ & \ddots & \\ & & J_r \end{pmatrix} , \quad J_\ell = \begin{pmatrix} \lambda_\ell & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_\ell \end{pmatrix} .$$

Die $J_\ell$ sind bis auf die Reihenfolge eindeutig festgelegt.

Beweis: Siehe etwa Smirnow, Anhang zu Teil III.1.

Bemerkungen zu Satz 9.2:

1) Jedes $\lambda_\ell$, $\ell=1,\ldots,n$ ist EW.

2) $\rho(\lambda)$ = Anzahl der $I_j$ mit $\lambda$ auf der Hauptdiagonalen. Die $\lambda_\ell$ sind also die nach ihren geometrischen Vielfachheiten gezählten EW.

3) $\sigma(\lambda)$ = Summe der Längen sämtlicher $I_j$ mit $\lambda$ auf der Hauptdiagonalen, denn

$$\det(\lambda I - J) = \prod_{\ell=1}^{r} \det(\lambda I - J_\ell) = \prod_{=1}^{r} (\lambda - \lambda_\ell)^{\nu_\ell},$$

$$\nu_\ell = \text{Länge}(J_\ell) .$$

Beispiel:

$$J = \begin{pmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & 2 & 1 & & & \\ & & & 2 & & & \\ & & & & 2 & & \\ & & & & & i & 1 \\ & & & & & & i \end{pmatrix} \qquad \begin{array}{ccc} \lambda & \rho(\lambda) & \sigma(\lambda) \\ 1 & 2 & 2 \\ 2 & 2 & 3 \\ i & 1 & 3 \end{array}$$

Sei nun A eine Matrix mit Jordan'scher Normalform J, also

$$A = XJX^{-1}, \quad J = \begin{pmatrix} J_1 & & \\ & \ddots & \\ & & J_r \end{pmatrix}$$

mit einer geeigneten Matrix X. Wir wollen uns die Bedeutung von X klarmachen. Sei $\nu_\ell$ die Länge von $J_\ell$. Wir spalten X auf entsprechend der Aufspaltung von J:

$$X = (X_1, \ldots, X_r)$$

mit $(n, \nu_\ell)$-Matrix $X_\ell$. Dann haben wir

$$AX_\ell = X_\ell J_\ell \,, \ell = 1, \ldots, r .$$

Wir untersuchen diese Gleichung für ein $\ell$ und setzen $\nu = \nu_\ell$ , $\lambda = \lambda_\ell$ . Seien $x_1,\ldots,x_\nu$ die Spalten von $X_\ell$, also $X_\ell = (x_1,\ldots,x_\nu)$ . Dann lautet die Gleichung

$$(AX_1,\ldots,AX_\nu) = (\lambda x_1, \lambda x_2+x_1, \ldots, x_\nu+x_{\nu-1})$$

oder

$$AX_1 = \lambda x_1$$

$$AX_1 = \lambda x_i + x_{i-1} , \quad i = 2,\ldots,\nu-1$$

$x_1$ ist also EV zum EW $\lambda_\ell$, wie wir schon aus Beispiel 3 und Satz 9.1 wissen. Für die weiteren Vektoren $x_i$ gilt

$$(A-\lambda I)\, x_i = x_{i-1}, \; i = 2,\ldots,\nu ,$$

also

$$(A-\lambda I)^{i-1} x_i = x_1, \quad (A-\lambda)^i x_i = 0 \quad .$$

Definition 9.3: Ein Vektor mit $(A-\lambda I)^{i-1} x \neq 0$, $(A-\lambda I)^i x = 0$ heißt Hauptvektor (HV) der Stufe i von A zum EW $\lambda$ . Der von allen Hauptvektoren zu einem EW $\lambda$ von A aufgespannte Teilraum heißt invarianter Unterraum von A zum EW $\lambda$ .

Bemerkung:

1. HVen der Stufe 1 sind gerade die EVen, der von ihnen aufgespannte Teilraum heißt Eigenraum zu $\lambda$.
2. Die algebraische Vielfachheit eines EW ist gleich der Dimension des zugehörigen invarianten Unterraumes.
3. Eine komplexe (n,n)-Matrix besitzt n l.u. Hauptvektoren, etwa die Spalten einer Matrix X, welche sie auf Jordan-Form transformiert.

Definition 9.4: Eine (n,n)-Matrix A heißt hermitesch, wenn $A = A^*$ mit $A^* = \overline{A}^{\cdot T}$.

Beispiel: Folgende Matrizen sind hermitesch:

$$\begin{pmatrix} 1 & i \\ -i & 2 \end{pmatrix} \quad , \quad \begin{pmatrix} 1 & 2 \\ 2 & 3 \end{pmatrix} \quad .$$

Insbesondere sind reell-symmetrische Matrizen hermitesch.

Satz 9.3: Sei A hermitesch. Dann sind alle EWe von A reell, und A besitzt n l.u. EVen.

Bemerkung: Die Jordan'sche Normalform einer hermiteschen Matrix ist also eine reelle Diagonalmatrix.

Beweis:

1) Realität der EWe. Ist $Ax = \lambda x$, so ist $(x,Ax) = \lambda(x,x)$ und $(x,x) > 0$ genügt es zu zeigen, daß $(x,Ax)$ reell ist. Dies folgt aus

$$\overline{(x,Ax)} = \overline{(A^* x,x)} = \overline{(Ax,x)} = (x,Ax)$$

2) Es genügt zu zeigen, daß keine HVen der Stufe 2 auftreten können. Ist x ein solcher, so ist

$$((A-\lambda I)x,(A-\lambda I)x) = (x, (A-\lambda I)^2 x) = 0 .$$

also $(A-\lambda I)x = 0$, im Widerspruch zu $(A-\lambda I)\, x \neq 0$.

## § 10 DIE POTENZMETHODE UND DAS LR-VERFAHREN

Sei $A$ eine komplexe $(n,n)$-Matrix. Wir wollen die Eigenwerte $\lambda_i$ von $A$ berechnen. Das einfachste Verfahren ist die Potenzmethode. Ausgehend von einem Vektor $x^{(0)}$ bildet sie der Reihe nach die Vektoren

$$x^{(k+1)} = Ax^{(k)} = A^{k+1} x^{(0)} \quad , \quad k = 0,1,\ldots \quad .$$

Wir analysieren die Potenzmethode zunächst in dem Fall

$$|\lambda_1| > |\lambda_2| > \ldots > |\lambda_n| \quad .$$

Dann hat $A$ $n$ l.u. Eigenvektoren $x_1,\ldots,x_n$, und es gilt

$$x^{(0)} = \sum_{i=1}^{n} c_i x_i \quad ,$$

$$x^{(k)} = A^k x^{(0)} = \sum_{i=1}^{n} c_i A^k x_i = \sum_{i=1}^{n} c_i \lambda_i^k x_i$$

$$= \lambda_1^k (c_1 x_1 + r_k) \quad ,$$

$$r_k = \sum_{i=2}^{n} c_i \left( \frac{\lambda_i}{\lambda_1} \right)^k x_i \quad .$$

Offenbar geht $r_k \to 0$ mit $k \to \infty$, und zwar gilt

$$\|r_k\| = o\left( \left( \frac{\lambda_2}{\lambda_1} \right)^k \right) \quad .$$

Zur Berechnung von $\lambda_1$ wählt man einen komplexen Vektor $c$ und bildet

$$(x^{(k)},c) = \lambda_1^k (c_1 (x_1,c) + (r_k,c)) \quad .$$

Ist $c_1(x_1,c) \neq 0$, so gilt für $k \to \infty$

$$\frac{(x^{(k+1)},c)}{(x^{(k)},c)} = \lambda_1 \frac{c_1(x_1,c) + (r_{k+1},c)}{c_1(x_1,c) + (r_k,c)} \to \lambda_1 \quad .$$

Genauer gilt

$$\frac{(x^{(k+1)},c)}{(x^{(k)},c)} = \lambda_1 + O\left(\left(\frac{\lambda_2}{\lambda_1}\right)^k\right) \quad ,$$

d.h. die Konvergenzgeschwindigkeit hängt von $\left|\frac{\lambda_2}{\lambda_1}\right|$ ab.

Einen Eigenvektor $x_1$ zu $\lambda_1$ bekommt man als Grenzwert der Folge $x^{(k)} / x_j^{(k)}$ für geeignet gewähltes $j$ ($j$-te Komponente von $x_1$ nicht 0 !).

BEISPIEL:

$$A = \begin{pmatrix} 90 & 231 & 70 \\ 110 & 336 & 110 \\ 70 & 231 & 90 \end{pmatrix}, \quad x^{(0)},x^{(1)},x^{(2)} = \begin{matrix} 1 & 391 & 190756 \\ 1 & 556 & 272836 \\ 1 & 391 & 190756 \end{matrix}$$

Mit $c = (1,1,1)^T$ erhält man

$$\frac{(x^{(1)},c)}{(x^{(0)},c)} = 446 \quad , \quad \frac{(x^{(2)},c)}{(x^{(1)},c)} = 489.05 \quad .$$

Für $j = 2$ lauten die normierten Vektoren $x^{(k)} / x_2^{(k)}$:

| | | |
|---|---|---|
| 1 | 0.703237 | 0.699160 |
| 1 | 1 | 1 |
| 1 | 0.703237 | 0.699160 |

Die exakten Werte sind

$$\lambda_1 = 490 \quad , \quad \lambda_2 = 20 \quad , \quad x_1 = \begin{pmatrix} 0.7 \\ 1 \\ 0.7 \end{pmatrix} \quad .$$

Aus dem kleinen Verhältnis $\frac{\lambda_2}{\lambda_1} = 0.04$ erklärt sich die schnelle Konvergenz.

Zur Berechnung der weiteren Eigenwerte bildet man die Matrix $T = (A - \mu I)^{-1}$. Diese hat die Eigenwerte $(\lambda_i - \mu)^{-1}$ mit den Eigenvektoren $x_i$. Zur Berechnung von $\lambda_2$ wählt man $\mu$ so, daß

$$|\lambda_2 - \mu| < |\lambda_i - \mu| \quad , \quad i \neq 2 \quad .$$

Dann ist $(\lambda_2 - \mu)^{-1}$ betragsgrößter Eigenwert von T. Diesen kann man nach der Potenzmethode berechnen. Zur Bildung von

$$x^{(k+1)} = T\, x^{(k)} \quad ,$$

$$(A - \mu) x^{(k+1)} = x^{(k)}$$

muß man bei jedem Schritt ein Gleichungssystem mit ein und derselben Matrix lösen. Man braucht also die LR-Zerlegung nur einmal durchzuführen.

Dieses Verfahren heißt "inverse Potenzmethode" oder Wielandt - -Iteration.

Sei nun A eine beliebige Matrix mit Jordan'scher Normalform J, also

$$A = X J X^{-1} \quad , \quad J = \begin{pmatrix} J_1 & & \\ & \ddots & \\ & & J_r \end{pmatrix} \quad , \quad J_\ell = \begin{pmatrix} \lambda_\ell & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_\ell \end{pmatrix} .$$

Die Eigenwerte $\lambda_\ell$ sind also nach geometrischer Vielfachheit gezählt und nach abnehmenden Beträgen geordnet:

$|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_r|$

Seien $x^{(k)} = Ax^{(k-1)}$ die Vektoren der Potenzmethode. Mit $x^{(k)} = Xy^{(k)}$ wird $y^{(k)} = J y^{(k-1)}$. Spalten wir $y^{(k)}$ auf in Teilvektoren $y^{(k)}$ der Länge $\nu_\ell$, so entsteht

$$y_\ell^{(k)} = J_\ell y_\ell^{(k-1)} \quad , \quad y_\ell^{(k)} = J_\ell^k y_\ell^{(0)} \quad , \quad \ell = 1,\ldots,r \quad , \quad y^{(k)} = \begin{pmatrix} y_1^{(k)} \\ \vdots \\ y_r^{(k)} \end{pmatrix} .$$

Zur Berechnung von $J_\ell^k$ setzen wir

$$J_\ell = \lambda_\ell I + N_\ell \quad , \quad N_\ell = \begin{pmatrix} 0 & 1 & & 0 \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ 0 & & & 0 \end{pmatrix} .$$

Wegen $N_\ell^{\nu_\ell} = 0$ wird dann für $k \geq \nu_\ell$

$$\begin{aligned} J_\ell^k = (\lambda_\ell I + N_\ell)^k &= \sum_{\nu=0}^{\nu_\ell - 1} \binom{k}{\nu} \lambda_\ell^{k-\nu} N_\ell^\nu \\ &= \lambda^k \sum_{\nu=0}^{\nu_\ell - 1} \binom{k}{\nu} \lambda_\ell^{-\nu} N_\ell^\nu \\ &= \lambda_\ell^k M_{\ell k} \end{aligned}$$

mit einem Polynom $M_{\ell k}$ vom Grade $< \nu_\ell$ in $k$. Damit haben wir

$$y_\ell^{(k)} = \lambda_\ell^k M_{\ell k} y_\ell^{(0)} \quad , \quad \ell = 1,\ldots,r \quad .$$

Um nun wieder zu den $x^{(k)}$ zurückzukommen, zerlegen wir $X = (X_1,\ldots,X_r)$ mit $(n,\nu_\ell)$ - Matrizen $X_\ell$ und haben dann

$$x^{(k)} = X\, y^{(k)} = \sum_{\ell=1}^{r} X_\ell\, y_\ell^{(k)} = \sum_{\ell=1}^{r} \lambda_\ell^k X_\ell M_{\ell k} y_\ell^{(0)} \quad .$$

Diese Darstellung von $x^{(k)}$ ist der Ersatz für die oben benutzte Entwicklung nach Eigenvektoren, in welche sie für $\nu_\ell = 1$, $\ell = 1,\ldots,r$ übergeht.

Wir untersuchen die Potenzmethode nun für verschiedene Fälle.

<u>Fall 1:</u> Es gibt einen Eigenwert $\lambda_1$ maximalen Betrags, und es ist $\rho(\lambda_1) = \sigma(\lambda_1) = \rho$.

Es ist also

$$\lambda_1 = \lambda_2 = \cdots = \lambda_\rho \quad , \quad |\lambda_\rho| > |\lambda_{\rho+1}| \geq \cdots \geq |\lambda_r| \quad ,$$

$$x^{(k)} = \sum_{\ell=1}^{\rho} \lambda_\ell^k M_{\ell k} y_\ell^{(0)} + \sum_{\ell=\rho+1}^{r} \lambda_\ell^k M_{\ell k} y_\ell^{(0)} \quad .$$

Für $\ell = 1,\ldots,\rho$ ist $\nu_\ell = 1$ und $M_{\ell k}$ die $(1,1)$ - Matrix 1. Also wird

$$x^{(k)} = \lambda_1^k \left\{ \sum_{\ell=1}^{\rho} X_\ell\, y_\ell^{(0)} + \sum_{\ell=\rho+1}^{r} \left( \frac{\lambda_\ell}{\lambda_1} \right)^k X_\ell M_{\ell k} y_\ell^{(0)} \right\}$$

$$= \lambda_1^k \{ x_1 + r_k \} \quad .$$

Hier ist $x_1$ ein Eigenvektor zum Eigenwert $\lambda_1$, und $r_k$ hat die Größenordnung

$$\left(\frac{\lambda_{\rho+1}}{\lambda_1}\right)^k k^{\nu-1} \to 0 \quad , \quad k \to \infty \quad ,$$

wo $\nu = \text{Max } \nu_\ell$. Damit hat m an in diesem Fall die gleichen Verhältnisse wie im oben diskutierten Fall. Die Konvergenz ist im wesentlichen $(\lambda_{\rho+1} / \lambda_\rho)^k$; der Faktor $k^{\nu-1}$ wächst so langsam, daß er keine Rolle spielt.

Fall 2: Es gibt einen Eigenwert $\lambda_1$ maximalen Betrags, aber es ist $\rho(\lambda_1) < \sigma(\lambda_1)$.

Wir betrachten den einfachsten Spezialfall $\rho(\lambda_1) = 1$, $\sigma(\lambda_1) = 2$. Es ist dann

$$\begin{aligned} x^{(k)} &= \lambda_1^k X_1 M_{1k} y_1^{(0)} + \sum_{\ell=2}^{r} \lambda_\ell^k X_\ell M_{\ell k} y_\ell^{(0)} \\ &= \lambda_1^k \left\{ X_1 M_{1k} y_1^{(0)} + \sum_{\ell=2}^{r} \left(\frac{\lambda_\ell}{\lambda_1}\right)^k X_\ell M_{\ell k} y_\ell^{(0)} \right\} \\ &= \lambda_1^k \left\{ X_1 M_{1k} y_1^{(0)} + r_k \right\} \quad . \end{aligned}$$

Ähnlich wie oben ist $r_k$ von der Größenordnung

$$r_k = \left(\frac{\lambda_2}{\lambda_1}\right)^k k^{\nu-1} \to 0 \quad , \quad k \to \infty$$

mit $\nu = \underset{\ell>1}{\text{Max}}\, \nu_\ell$. $M_{1k}$ ist ein Polynom vom Grade 1 in k, d.h.

$$X_1 M_{1k} y_1^{(0)} = a + k\,b$$

mit geeigneten Vektoren a, b. Bildet man nun $(x^k, c)$ und bildet die Quotienten zur Berechnung von $\lambda_1$, so entsteht

$$\frac{(x^{(k+1)},c)}{(x^{(k)},c)} = \lambda_1 \frac{(a,c) + (k+1)(b,c) + (r_{k+1},c)}{(a,c) + k(b,c) + (r_k,c)}$$

$$= \lambda_1 \left(1 + O\left(\frac{1}{k}\right)\right)$$

für $k \to \infty$. Man hat also auch in diesem Fall Konvergenz gegen $\lambda_1$, aber sehr langsame.

Fall 3: Es gibt verschiedene betragsmaximale Eigenwerte.

Wir behandeln wieder den einfachsten Spezialfall

$$|\lambda_1| = |\lambda_2| > |\lambda_3| \geq \ldots \geq |\lambda_r| \quad , \quad \lambda_1 \neq \lambda_2$$

mit $\sigma(\lambda_1) = \sigma(\lambda_2) = 1$. Es ist dann

$$x^{(k)} = \lambda_1^k X_1 y_1^{(0)} + \lambda_2^k X_2 y_2^{(0)} + \sum_{\ell=3}^{r} \lambda_\ell^k X_\ell M_{\ell k} y_\ell^{(0)}$$

$$= \lambda_1^k \left\{ X_1 y_1^{(0)} + \left(\frac{\lambda_2}{\lambda_1}\right)^k X_2 y_2^{(0)} + \sum_{\ell=3}^{r} \left(\frac{\lambda_\ell}{\lambda_1}\right)^k X_\ell M_{\ell k} y_\ell^{(0)} \right\}$$

$$= \lambda_1^k \left\{ X_1 y_1^{(0)} + \left(\frac{\lambda_2}{\lambda_1}\right)^k X_2 y_2^{(0)} + r_k \right\} \quad .$$

Wie in den früheren Fällen geht $r_k \to 0$ mit $k \to \infty$, und zwar (beinahe) geometrisch. Setzen wir

$$\frac{\lambda_2}{\lambda_1} = e^{i\alpha} \quad , \quad 0 < \alpha < 2\pi \quad ,$$

so ist

$$\left(\frac{\lambda_2}{\lambda_1}\right)^k = e^{i\alpha k} = \cos \alpha k + i \sin \alpha k \quad .$$

Der Vektor

$$x_1 \, y_1^{(0)} + \left(\frac{\lambda_2}{\lambda_1}\right)^k x_2 \, y_2^{(0)}$$

ist also (i. allg., d.h. für $y_2^{(0)} \neq 0$) nicht konvergent, vielmehr oszillierend. In diesem Fall haben wir also keine Konvergenz.

Zusammenfassend haben wir den

SATZ 10.1: Die Potenzmethode konvergiert, wenn es genau einen betragsgrößten Eigenwert gibt. Stimmen für diesen die algebraische und geometrische Vielfachheit überein, so ist die Konvergenz geometrisch. Gibt es verschiedene betragsgleiche Eigenwerte, so ist die Potenzmethode nicht konvergent.

Wir wollen uns nun überlegen, wie wir alle Eigenwerte einer Matrix durch die Potenzmethode berechnen können. Im Prinzip kann das - wie oben besprochen - durch die inverse Potenzmethode geschehen. Wir werden aber eine sehr viel elegantere Methode finden.

Betrachten wir wieder den Fall $n$ betragsmäßig verschiedener Eigenwerte, also $|\lambda_1| > |\lambda_2| > \ldots > |\lambda_n|$, und es gibt $n$ l.u. Eigenvektoren $x_1,\ldots,x_n$. Wenden wir die Potenzmethode auf $n$ Startvektoren $x_1^{(0)},\ldots,x_n^{(0)}$ gleichzeitig an, also

$$X_{k+1} = A \, X_k \quad , \qquad X_k = \left\{ x_1^{(k)},\ldots,x_n^{(k)} \right\} \quad ,$$

so passiert nicht viel Interessantes: Alle Spalten von $X_k$ werden von $\lambda_1^k \, x_1$ dominiert. Um dies zu vermeiden, gehen wir raffinierter vor. In der ersten Spalte machen wir die ganz normale Potenzmethode, normieren $x_1^{(1)}$ allerdings so, daß die erste Komponente 1 ist:

$$r_{11}\, x_1^{(1)} = A\, x_1^{(0)}$$

In der zweiten Spalte wollen wir aber möglichst keine Anteile von $x_1$ haben und subtrahieren daher ein geeignetes Vielfaches von $x_1^{(1)}$:

$$r_{22}\, x_2^{(1)} = A\, x_2^{(0)} - r_{12}\, x_1^{(1)} \quad .$$

$r_{12}$ bestimmen wir so, daß die erste Komponente von $x_2^{(1)}$ verschwindet. Danach wird $r_{22}$ so bestimmt, daß die zweite Komponente von $x_2^{(1)}$ 1 wird.

Entsprechend geht man in der Spalte $j$ vor: Man möchte, daß $x_j^{(1)}$ möglichst keine Anteile von $x_1,\ldots,x_{j-1}$ hat und subtrahiert dazu Vielfache von $x_1^{(1)},\ldots,x_{j-1}^{(1)}$ so, daß die ersten $j-1$ Komponenten von $x_j^{(1)}$ verschwinden. Anschließend wird $x_j^{(1)}$ so normiert, daß die $j$-te Komponente 1 ist:

$$r_{jj}\, x_j^{(1)} = A\, x_j^{(0)} - r_{1j}\, x_1^{(1)} - r_{2j}\, x_2^{(1)} - \ldots - r_{j,j-1} x_{j-1}^{(1)},$$

$$j = 1,\ldots,n \quad .$$

Faßt man die $r_{ij}$ zu der rechten Dreiecksmatrix $R_0$ zusammen, so lautet dies

$$X_1\, R_0 = A\, X_0 \quad .$$

$X_1$ ist eine linke Dreiecksmatrix mit Hauptdiagonale 1. Wir haben hier also die LR-Zerlegung von $A\, X_0$ vorliegen.

Die Potenzmethode läuft nun folgendermaßen: Sei $X_0 = I$. Ist $X_0$ berechnet, so bilde man die LR-Zerlegung

$$X_{k+1} R_k = A X_k$$

von $A X_k$, wo also $X_{k+1}$ der linke Faktor ist.

Aufgrund der Herleitung erwarten wir, daß mit $k \to \infty$

$$\begin{aligned} x_1^{(k)} &\to r_{11} x_1 \\ x_2^{(k)} &\to r_{12} x_1 + r_{22} x_2 \\ &\vdots \\ x_n^{(k)} &\to r_{1n} x_1 + r_{2n} x_2 + \ldots + r_{nn} x_n \end{aligned}$$

mit geeigneten Zahlen $r_{ij}$. Anders ausgedrückt:

$$X_k \to X R \quad , \qquad X = (x_1, \ldots, x_n) \quad .$$

Nach einer Idee von Rutishauser kann man die Rechnung sehr elegant durchführen: Man setze

$$L_k = X_k^{-1} X_{k+1} \quad , \qquad A_k = X_k^{-1} A X_k \quad .$$

Dann sind die $L_k$ linke Dreiecksmatrizen mit Diagonale 1, und die $A_k$ sind alle ähnlich zu $A$. Es gilt weiter

$$A_k = X_k^{-1} A X_k = (L_k X_{k+1}^{-1}) A X_k = L_k R_k \quad ,$$

$$A_{k+1} = (X_{k+1}^{-1} A) X_{k+1} = (R_k X_k^{-1}) X_{k+1} = R_k L_k \, .$$

Schließlich erwarten wir noch, daß mit $k \to \infty$

$$A_k = X_k^{-1} A X_k \to R^{-1} X^{-1} A X R = R^{-1} J R \quad ,$$

wobei $J$ die Diagonalmatrix mit den Eigenwerten $\lambda_\ell$ auf der Diagonalen ist. Damit sind wir beim LR - Verfahren angelangt:

1) $A_0 = A$ .

2) Ist $A_k$ berechnet, so zerlege man

$$A_k = L_k R_k$$

und setze

$$A_{k+1} = R_k L_k \quad .$$

Wir halten noch einmal fest:

Alle Matrizen $A_k$ sind ähnlich zu $A$, insbesondere gilt:

$$A_{k+1} = R_k L_k = R_k A_k R_k^{-1} = R_k R_{k-1} A_{k-1} R_{k-1}^{-1} R_k^{-1}$$

$$= \ldots = R_k \cdot \ldots \cdot R_0 A R_0^{-1} \cdot \ldots \cdot R_k^{-1} = (R_k \cdot \ldots \cdot R_0) A (R_k \cdot \ldots \cdot R_0)^{-1},$$

ebenso:

$$A_{k+1} = R_k L_k = L_k^{-1} A_k L_k = \ldots = (L_0 \cdot \ldots \cdot L_k)^{-1} A (L_0 \cdot \ldots \cdot L_k)$$

$$(*) \qquad \Leftrightarrow L_0 \cdot \ldots \cdot L_k A_{k+1} = A L_0 \cdot \ldots \cdot L_k$$

Damit läßt sich leicht eine später benötigte LR-Zerlegung für $A^k$ angeben, es gilt:

$$A^k = (L_0 \cdot \ldots \cdot L_{k-1})(R_{k-1} \cdot \ldots \cdot R_0) \quad ,$$

denn:

$$A^1 = A = L_o \cdot R_o$$

und:

$$L_o \cdot \ldots \cdot L_{k-2} \underbrace{L_{k-1}R_{k-1}}_{=A_{k-1}} R_{k-2} \cdot \ldots \cdot R_o \overset{(*)}{=} A(L_o \cdot \ldots \cdot L_{k-2})(R_{k-2} \cdot \ldots \cdot R_o)$$

, womit die Darstellung für $A^k$ durch Induktion bewiesen ist.

Wir wollen nun folgendes Resultat über das LR-Verfahren beweisen:

SATZ 10.2: Sei $A$ eine $(n,n)$ - Matrix. Es gelte

i) $\forall A_i \quad \exists L_i, R_i$ mit $A_i = L_i R_i$, d.h. das LR-Verfahren sei durchführbar.

ii) Die Eigenwerte von $A$ sind betragsmäßig getrennt, d.h. es gilt:

$$|\lambda_1| > |\lambda_2| > \ldots > |\lambda_n|$$

Damit ist $A$ diagonalisierbar, die entsprechende Zerlegung sei:

$$A = X \cdot J \cdot X^{-1} \quad , \quad J = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$$

iii) Für $X$ und $X^{-1}$ existieren LR-Zerlegungen:

$$X = L_+ R_+ \quad , \quad X^{-1} = L_- R_- \quad ,$$

$$(L_+)_{ii} = (L_-)_{ii} = 1 \quad , \quad i = 1, \ldots, n \; .$$

Dann gilt

$$\lim_{k\to\infty} A_k = \lim_{k\to\infty} R_k = \begin{pmatrix} \lambda_1 & & * \\ & \ddots & \\ 0 & & \lambda_n \end{pmatrix} \quad , \quad \lim_{k\to\infty} L_k = I \quad .$$

BEWEIS:

Idee: Durch den Vergleich zwei verschiedener LR-Zerlegungen von $A^k$ zeigen wir, daß die Subdiagonalelemente von $L_k$ (linear) gegen Null konvergieren.

Wir vereinfachen den Beweis, indem wir zusätzlich $\lambda_n \neq 0$ fordern.

Wir haben

$$\begin{aligned} & A = X \cdot J \cdot X^{-1} \\ \Rightarrow \quad & A^k = X \cdot J^k \cdot X^{-1} \\ & = L_+ R_+ J^k L_- R_- \\ & = L_+ R_+ (J^k L_- J^{-k}) J^k R_- \quad . \end{aligned}$$

Sei $L_- = (\ell_{ij})_{ij}$, dann gilt

$$(J^k L_- J^{-k})_{ij} = \underbrace{\left(\frac{\lambda_i}{\lambda_j}\right)^k}_{c_{ij}} \ell_{ij} \quad , \text{ wobei } \quad \ell_{ij} = \begin{cases} 1 \, , & i = j \\ 0 \, , & i < j \end{cases} ,$$

also gilt nach Voraussetzung ii) für $i > j$: $|c_{ij}| < 1$ und damit

$$J^k L_- J^{-k} = I + F_k$$

mit $\lim_{k\to\infty} F_k = 0$, $F_k$ - untere $\Delta$ - Matrix.

Also:

$$A^k = L_+ R_+ (I + F_k) J^k R_-$$

$$= L_+ (I + \underbrace{R_+ F_k R_+^{-1}}_{=: G_k}) R_+ J^k R_-$$

Da $F_k \xrightarrow{k} 0$, folgt: $G_k \xrightarrow{k} 0$, also: $I + G_k \xrightarrow{k} I$

Demnach existiert ein $K \in \mathbb{N}$, s.d. $\forall\, k \geq K$ gilt:

$$\exists\ \tilde{L}_k, \tilde{R}_k \quad \text{s.d.} \quad I + G_k = \tilde{L}_k \cdot \tilde{R}_k$$

(D.h. es existiert eine LR-Zerlegung von $I + G_k$.)

Infolge $G_k \xrightarrow{k} 0$ gilt:

$$\tilde{L}_k, \tilde{R}_k \xrightarrow{k} I$$

Damit haben wir

$$A^k = \underbrace{L_+ \tilde{L}_k}_{\substack{\text{untere} \\ \Delta\text{-Matrix}}} \quad \underbrace{\tilde{R}_k R_+ J^k R_-}_{\substack{\text{obere} \\ \Delta\text{-Matrix}}} .$$

Andererseits (s.o.):

$$A^k = L_0 \cdot \ldots \cdot L_{k-1} \cdot R_{k-1} \cdot \ldots \cdot R_0$$

Aus der Eindeutigkeit der LR-Zerlegung folgt damit:

$$L_0 \cdot \ldots \cdot L_{k-1} = L_+ \tilde{L}_k$$

$$R_{k-1} \cdot \ldots \cdot R_0 = \tilde{R}_k R_+ J^k R_-$$

Nun gilt:

$$L_k = (L_0 \cdot \ldots \cdot L_{k-1})^{-1}(L_0 \cdot \ldots \cdot L_k) = \tilde{L}_k^{-1} L_+^{-1} L_+ \tilde{L}_{k+1}$$

$$= \tilde{L}_k^{-1} \tilde{L}_k \xrightarrow{k} I$$

$$R_k = (R_k \cdot \ldots \cdot R_0)(R_{k-1} \cdot \ldots \cdot R_0)^{-1} = \tilde{R}_{k+1} R_+ J^{k+1} R_-^{-1} J^{-k} R_+^{-1} \tilde{R}_k^{-1}$$

$$= \tilde{R}_{k+1} R_+ J R_+^{-1} \tilde{R}_k^{-1} \xrightarrow{k} R_+ J R_+^{-1}$$

$$\Rightarrow A_k = L_k R_k \xrightarrow{k} R_+ J R_+^{-1}$$

$R_+ J R_+^{-1}$ ist eine obere Dreiecksmatrix, und mit $R_+ = (r_{ij})_{i,j}$, $R_+^{-1} = (\rho_{ij})_{i,j}$ gilt:

$$(R_+ J R_+^{-1})_{ii} = \sum_{k=1}^{n} r_{ik} \lambda_k \rho_{ki} = r_{ii} \lambda_i \rho_{ii} = \lambda_i$$

Damit ist Satz 10.2 bewiesen.

■

Wir wollen an dieser Stelle gleich auf die Schwächen des LR - Verfahrens hinweisen:

1) Die Voraussetzungen i) und ii) in Satz 10.2 sind notwendig, d.h. das Verfahren

   - bricht zusammen, falls $A_k$ keine LR-Zerlegung besitzt.
   - konvergiert i. allg. nicht, falls $X$ oder $X^{-1}$ keine LR-Zerlegung besitzt.

2) Das Verhältnis: "Konvergenzgeschwindigkeit zu Rechenaufwand pro Iterationsschritt ($\sim \frac{2}{3} n^3$)" ist ungünstig, das Verfahren insgesamt also aufwendig.

   Insbesondere ist die Konvergenz sehr langsam, wenn $|\lambda_{i+1} / \lambda_i| \approx 1$ ist.

3) Das Verfahren bricht bei schlecht konditionierten Matrizen zusammen.

## § 11 DAS QR - VERFAHREN

Wir hatten mit dem LR-Verfahren ein erstes Werkzeug zur Bestimmung aller Eigenwerte einer Matrix kennengelernt, jedoch gleich auf dessen Nachteile hinweisen müssen.

Wir wollen nun das in der Praxis am besten bewährte Verfahren, das QR-Verfahren von Francis (1960) vorstellen.

$$\text{für } k=0,1,2,\ldots \quad \begin{cases} \text{Setze} & A_0 = A \\ \text{zerlege} & A_k = Q_k \cdot R_k \\ \text{berechne} & A_{k+1} = R_k \cdot Q_k \end{cases},$$

wobei die $Q_k$ unitäre Matrizen und die $R_k$ rechte $\Delta$ - Matrizen sind.

D.h. das QR-Verfahren besitzt die gleiche elegante Form wie das LR-Verfahren, nur daß wir laufend (aufwendigere) QR anstelle von LR - Zerlegungen berechnen müssen. Nach Satz 7.1 haben wir damit ein stets ausführbares Iterationsschema: Zu einer Matrix $A_k$ läßt sich stets die QR-Zerlegung finden.

Über die Konvergenz des Verfahrens gibt der folgende Satz Auskunft:

SATZ 11.1: Sei $A$ eine $(n,n)$ - Matrix mit betragsmäßig getrennten Eigenwerten $|\lambda_1| > |\lambda_2| > \ldots > |\lambda_n|$.

Dann gilt für die iterierten Matrizen des QR - Verfahrens:

$$(A_k)_{ij} \xrightarrow{k\to\infty} \begin{cases} 0 & , \quad i > j \\ \lambda_i & , \quad i = j \end{cases}$$

Wir geben nur eine kurze

BEWEISSKIZZE: Analog zum Beweis von Satz 10.2 arbeitet man mit dem Vergleich zweier QR-Zerlegungen von $A^k$.

Aus $A_{k+1} = R_k Q_k = Q_k A_k Q_k^*$ folgt wie beim LR - Verfahren:

$$A_{k+1} = (Q_0 \cdot \ldots \cdot Q_k)^* A_0 (Q_0 \cdot \ldots \cdot Q_k) \quad \text{sowie}$$

$$A^k = (Q_0 \cdot \ldots \cdot Q_{k-1})(R_{k-1} \cdot \ldots \cdot R_0) \quad ,$$

womit die erste Q-R-Zerlegung von $A^k$ berechnet ist.

Die zweite Zerlegung bekommt man wieder über die Identität $A^k = X J^k X^{-1}$, $J = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$, indem man fast wörtlich wie im Beweis zu Satz 10.2 vorgeht.
Man nimmt dabei an, daß $X^{-1}$ eine LR - Zerlegung besitzt, ist dies nicht der Fall, so erscheinen die Eigenwerte i. allg. nicht mehr betragsmäßig geordnet, auch wird der Beweis schwieriger. Die restliche Argumentation verläuft analog zum Beweis für das LR - Verfahren.

■

Programm für den QR - Algorithmus

Wir nehmen an, daß wir schon ein Unterprogramm für die Q-R-Zerlegung nach Householder besitzen:

House: Proc (A, Vek)

Der Aufruf erfolge mit A - die zu zerlegende (n,n)-Matrix

Vek - ein beliebiger (n) - Vektor .

Das Unterprogramm berechne die QR-Zerlegung und überschreibe die Eingangsdaten wie folgt:

$$A \leftarrow \begin{pmatrix} v_1 & r_{12} & r_{13} & \cdots & r_{14} \\ & v_2 & r_{23} & & r_{24} \\ & & \ddots & & \\ & & & & r_{n-1,n} \\ & & & v_{n-1} & r_{n,n} \end{pmatrix}, \qquad Vek \leftarrow \begin{pmatrix} -r_{11} \\ -r_{22} \\ \vdots \\ -r_{nn} \end{pmatrix},$$

wobei $r_{ij}$ die Elemente der oberen $\Delta$-Matrix $R$ und $v_1,\ldots,v_{n-1}$ die nichttrivialen Anteile der Transformationsvektoren des Householderalgorithmus sind.

D.h.

$$R = \overbrace{S_{n-1} \cdot \ldots \cdot S_1}^{Q^*} A$$

$$\text{mit} \quad S_j = I - 2\tilde{v}_j \tilde{v}_j^* \,, \quad \tilde{v}_j = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ v_j \end{pmatrix} \begin{matrix} \} \\ \} \\ \} \end{matrix} \begin{matrix} \\ (j-1)\text{-führende Nullen} \\ \\ n-j+1 \end{matrix} .$$

Wegen $\|v_j\| = 1$ folgt $S_j^2 = I$, so daß gilt

$$R \cdot Q = R \cdot S_1 \cdot \ldots \cdot S_{n-1} \quad .$$

Bei der Berechnung von $R \cdot Q$ wollen wir natürlich nicht $n-1$

Matrizenprodukte bilden, sondern uns die Struktur der $S_j$ zunutze machen:

Sei $B$ eine $(n,n)$-Matrix mit Zeilenvektoren $b_1^T,\ldots,b_n^T$, also

$$B = \begin{pmatrix} b_1^T \\ \vdots \\ b_n^T \end{pmatrix} \Rightarrow B \cdot S_j = \begin{pmatrix} b_1^T \\ \vdots \\ b_n^T \end{pmatrix} \cdot (I - 2\tilde{v}_j \tilde{v}_j^*)$$

$$= \begin{pmatrix} b_1^T - 2(b_1^T \cdot \tilde{v}_j)v_j^* \\ \vdots \\ b_n^T - 2(b_n^T \cdot v_j)v_j^* \end{pmatrix} .$$

D.h. wir müssen für die Rechtsmultiplikation mit $S_j$ im wesentlichen $n$ Skalarprodukte und $n$ Vektoradditionen der "Länge" $n-j+1$ berechnen.

Der Rechenaufwand für die Berechnung einer Zeile von $R \cdot Q$ ist durch

$$2 \sum_{j=1}^{n-1} (n-j+1) = n^2 + O(n)$$

R.O. gegeben. Daraus folgt ein Gesamtaufwand für die Berechnung des Produktes $R \cdot Q$ von $n^3 + O(n^2)$ Rechenoperationen.

Nun können wir ein einfaches Programm zum Q-R-Verfahren angeben. Der Einfachheit halber geben wir uns eine obere Schranke $K_{max}$ von Iterationen vor, die wir stets durchführen wollen. Es gibt natürlich bessere Abbruchkriterien für die Iteration, wie:

"Quadratsumme der Subdiagonalelemente $\leq \varepsilon_1$"

oder "Änderung des Diagonalvektors in der Norm $\leq \varepsilon_2$"

$\varepsilon_1$, $\varepsilon_2$ passend.

```
/* Alle Eigenwerte von A mit dem QR - Verfahren */
Do K=1 to Kmax;
   Call House(A,Diag);          /* QR-Zerlegung */
   Do L=1 to N;                 /* Wir schreiben die obere */
      Do I=1 to L-1;            /* Dreiecksmatrix in das */
         R(L,I)=0;              /* zweidimensionale Feld R */
      End;
      R(L,L)=-Diag(L);
      Do I=L+1 to N;
         R(L,I)=A(L,I);
      End;
   End;                         /* Ende Abschreiben */
   Do J=1 to N-1;
      /* Berechnung von R · Q = R · S1 · ... · Sj */
      /* mit Überschreiben auf R                  */
      Do I=1 to N;
         /* N Zeilenvektoren von R */
         Skalpro = 0
         Do L=J to N;
            Skalpro = Skalpro + R(I,L)*A(L,J);
            /* (I-te Zeile von R)*(K-te Spalte von A) */
         End;
         Skalpro = 2 * Skalpro;
         Do L=J to N
            R(I,L)=R(I,L)-Skalpro * A(L,J);
         End;
      End;
   End;          /* Ende J, in R steht */
   A=R           /* nun das Produkt R · Q */
End;             /* Ende K */
Do I=1 to N;
   Ew(I)=A(I,I);
End;
```

BEMERKUNG: Das Verfahren ist in dieser Primitivform wenig effizient. Gängige Verbesserungen sind:

- Anfängliche Transformation von $A$ auf Hessenbergform, man zeigt leicht, daß dann alle iterierten Matrizen $A_k$ ebenfalls Hessenbergmatrizen sind.

- Konvergenzbeschleunigende Shift-Techniken, darunter versteht man die Spektralsverschiebungen:

$$A_k - s_k I = Q_k R_k$$

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

Es ist allerdings ein diffiziles Problem, eine günstige Wahl für $s_k$ zu finden.

Wir wollen noch kurz vermerken, was passiert, wenn die Voraussetzung der betragsmäßig getrennten Eigenwerte nicht erfüllt ist,

a) gilt für ein $\lambda_i$ : $\sigma(\lambda_i) = \rho(\lambda_i) > 1$, so ändert sich nichts.

b) ist $\sigma(\lambda_i) > 1$ und $\sigma(\lambda_i) > \rho(\lambda_i)$, so konvergiert das Verfahren extrem langsam (Besserung durch shifts möglich).

c) sind mehrere Eigenwerte betragsgleich, d.h. gilt:

$$|\lambda_1| > |\lambda_2| > \ldots > |\lambda_t| = |\lambda_{t+1}| = \ldots = |\lambda_s| > |\lambda_{s+1}| > \ldots > |\lambda_n|$$

gilt (wie auch beim LR-Verfahren) eine etwas modifizierte Konvergenzaussage:

$$A_k \xrightarrow{k} \begin{pmatrix} \lambda_1 & & & & \\ & \ddots & & & \\ & & \lambda_{t-1} & & * \\ & & & \boxed{A_k^{s,t}} & \\ 0 & & & & \lambda_{s+1} \\ & & & & & \ddots \\ & & & & & & \lambda_n \end{pmatrix}$$

Die Elemente von $A_k^{s,t}$ konvergieren i. allg. nicht, jedoch konvergieren die Eigenwerte von $A_k^{s,t}$ gegen $\lambda_t,\ldots,\lambda_s$. Dieser Fall kann bei reellen, nichtsymmetrischen Matrizen auftreten, die Eigenwerte sind dann konjugiert zueinander.

## § 12 FEHLERABSCHÄTZUNG BEI EIGENWERTPROBLEMEN

SATZ 12.1 (Gerschgorin): Sei $A$ (n,n)-Matrix und seien

$$r_i = \sum_{j \neq i} |a_{ij}|$$

$$K_i = \{z \in C : |z - a_{ii}| \leq r_i\}$$

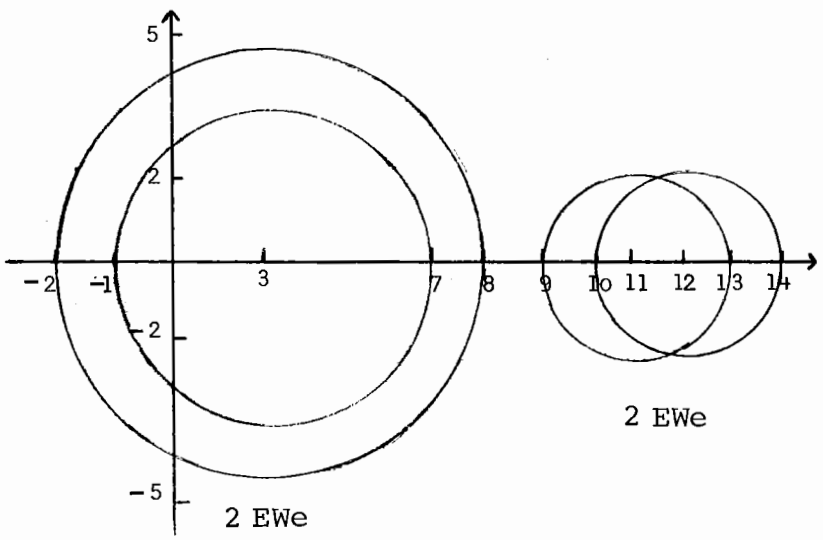
Für die "Gerschgorin-Kreise" $K_i$ gilt dann:

a) Alle Eigenwerte von $A$ sind in $\bigcup_{i=1}^{n} K_i$ enthalten.

b) m der Kreise $K_i$ seien punktfremd mit den restlichen $K_j$. Dann haben die in der Vereinigung dieser $K_i$ liegenden EWe zusammen genau die algebraische Vielfachheit m.

Beispiel:

$$A = \begin{pmatrix} 3 & 2 & 1 & -2 \\ 1 & 11 & 0 & 1 \\ -1 & 0 & 12 & -1 \\ -3 & 1 & 0 & 3 \end{pmatrix} \qquad \begin{aligned} r_1 &= 5 \\ r_2 &= 2 \\ r_3 &= 2 \\ r_4 &= 4 \end{aligned}$$



Beweis:

Zu a): Sei $\lambda$ EW und $x$ EV von $A$ mit $\|x\|_\infty = 1 = |x_{i_o}|$

$$Ax = \lambda x \Leftrightarrow \sum_{j=1}^{n} a_{ij} x_j = \lambda x_i \quad , \quad 1 \leq i \leq n$$

$$\Rightarrow (a_{ii} - \lambda) x_i = - \sum_{j \neq i} a_{ij} x_j$$

$$\Rightarrow |a_{ii} - \lambda||x_i| \leq \sum_{j \neq i} |a_{ij}||x_j| \leq$$

$$\leq \sum_{j \neq i} |a_{ij}| \|x\|_\infty = r_i$$

$$i = i_o \Rightarrow |a_{i_o i_o} - \lambda| \leq r_{i_o} \Rightarrow \lambda \in K_{i_o} \Rightarrow \text{a)}$$

Zu b): Sei für $0 \leq t \leq 1$ $A(t) = D + (A-D)t$

$$\text{mit} \quad D = \begin{pmatrix} a_{11} & & 0 \\ & \ddots & \\ 0 & & a_{nn} \end{pmatrix}$$

Es gilt: $A(0) = D$ und $A(1) = A$

Lemma: Seien $\lambda_i(t)$ die EWe von $A(t)$. Dann sind die $\lambda_i(t)$ stetige Funktionen von $t$.

Ist $\lambda_i(t_o)$ algebraisch p-facher EW, so hat $\lambda_i(t)$ bei $t_o$ eine höchstens p-fache Verzweigung (d.h. es gibt Umgebungen $U$ von $t_o$ und $V$ von $\lambda_i(t_o)$, so daß für $t \in U$ $A(t)$ Eigenwerte in $V$ besitzt, die zusammen genau die algebraische Vielfachheit $p$ haben).

$A(t)$ hat die Gerschgorin-Kreise $K_i(t) = \{z \in \mathbb{C} : |z-a_{ii}| \leq t \cdot r_i\}$ Für $t = o$ ist b) offensichtlich erfüllt. Ebenso für hinreichend kleines $t$, so daß die Gerschgorin-Kreise für verschiedene $a_{ii}$ punktfremd sind, da aus dem Lemma folgt, daß die Eigenwerte nicht aus einem Kreis in einen punktfremden anderen Kreis "springen" können. Dieses Argument gilt auch, wenn bei wachsendem $t$ einige der Kreise sich berühren oder überlappen und führt auf die Behauptung b).

Der Satz von Gerschgorin ist ein Beispiel für einen sogenannten Lokalisierungssatz. Sätze dieser Art sind für die Praxis sehr wertvoll, da sie ohne Rechnung einen Überblick über die ungefähre Lage der Eigenwerte geben. So interessiert in der Elektrotechnik oft nur das Vorzeichen von $\mathrm{Re}\ \lambda$.

SATZ 12.2: Sei $A$ Hermite'sch und seien $\lambda, x$ Näherungen für einen EW von $A$ mit zugehörigem EV.

Dann gibt es einen EW $\lambda_k$ von $A$ mit

$$|\lambda_k - \lambda| \leq \frac{\|d\|_2}{\|x\|_2}$$

BEWEIS: Sei $\{x_1,\ldots,x_n\}$ das Orthonormalsystem der EVen von A.

$$x \in \mathbb{C}^n \Rightarrow x = \sum_{i=1}^{n} c_i x_i\ ,\ c_i = x_i^* x \quad ,\quad \|x\|_2^2 = \sum_{i=1}^{n} |c_i|^2$$

$$d = Ax-\lambda x = \sum_{i=1}^{n} c_i(\lambda_i - \lambda)x_i$$

$$\|d\|_2^2 = \sum_{i=1}^{n} |c_i|^2 |\lambda_i-\lambda|^2 \geq \sum_{i=1}^{n} |c_i|^2 \underbrace{\min_{1\leq i\leq n} |\lambda_i-\lambda|^2}_{|\lambda_k-\lambda|^2} = \|x\|_2^2 |\lambda_k-\lambda|^2$$

SATZ 12.3: Seien $A$, $B$ Hermite'sche $(n,n)$-Matrizen. Dann kann man die Eigenwerte $\lambda_i$ von $A$ und die Eigenwerte $\mu_i$ von $B$ so anordnen, daß

$$|\lambda_i - \mu_i| \leq \|A - B\|_2$$

ist.

BEWEIS: Sei $\lambda_i$ Eigenwert zum Eigenvektor $x_i$ von $A$. Dann ist

$$d = Bx_i - \lambda_i x_i = Ax_i - \lambda_i x_i + (B - A)x_i$$
$$= (B - A)x_i$$

und damit

$$\|d\| \leq \|B-A\|_2 \|x_i\|_2 \quad .$$

Nach Satz 12.2 gibt es einen Eigenwert $\mu_i$ von $B$ mit

$$|\lambda_i - \mu_i| \leq \frac{\|d\|_2}{\|x_i\|_2} \leq \|B - A\|_2 \quad . \qquad \blacksquare$$

SATZ 12.4: Sei $A$ eine $(n,n)$-Matrix mit Eigenwerten $\lambda_1,\ldots,\lambda_r$, und sei $\nu$ die maximale Länge der Jordan-Kästchen von $A$. Sei $X$ eine Matrix, welche $A$ auf Jordan'sche Normalform bringt. Sei $A_\varepsilon = A + \varepsilon F$, $0 \leq \varepsilon \leq 1$. Dann liegen sämtliche Eigenwerte von $A_\varepsilon$ in der Vereinigung der Kreise

$$K_\ell = \{z \in \mathbb{C} : |z - \lambda_\ell| \leq \varepsilon^{1/\nu}(1 + k(X)) \|F\|_\infty\} \quad .$$

BEWEIS: Es ist $A = X J X^{-1}$. Also haben $A + \varepsilon F$, $J + \varepsilon G$ mit

$G = X^{-1} F X$ die gleichen Eigenwerte. Wir behandeln zwei Fälle.

1) $J$ besteht aus genau einem Jordan-Kästchen der Länge $\nu = n$. Sei $D$ die Diagonalmatrix mit der Diagonale $1, \varepsilon^{1/\nu}, \ldots, \varepsilon^{(\nu-1)/\nu}$. Dann ist

$$J = \begin{pmatrix} \lambda & 1 & & \\ & \lambda & 1 & \\ & & \ddots & \\ & & & \lambda \end{pmatrix} \quad , \quad D^{-1} J D = \begin{pmatrix} \lambda & \varepsilon^{1/\nu} & & \\ & \lambda & \varepsilon^{1/\nu} & \\ & & \ddots & \\ & & & \lambda \end{pmatrix} .$$

Die Matrix $D^{-1}(J + \varepsilon G)D$ hat die gleichen Eigenwerte wie $A + \varepsilon F$. Wir wenden den Satz von Gerschgorin auf $D^{-1}(J + \varepsilon G)D$ an. Die Gerschgorin-Kreise haben Mittelpunkt $\lambda + \varepsilon g_{ii}$ und Radius

$$r_i \leq \varepsilon^{1/\nu} + \varepsilon \, \varepsilon^{(1-\nu)/\nu} \sum_{\substack{j=1 \\ j \neq i}}^{n} |g_{ij}|$$

$$= \varepsilon^{1/\nu} \left( 1 + \sum_{\substack{j=1 \\ j \neq i}}^{n} |g_{ij}| \right) .$$

Jeder Eigenwert $\mu$ von $A_\varepsilon$ liegt in einem dieser Kreise, also

$$|\lambda + \varepsilon g_{ii} - \mu| \leq r_i$$

für ein $i$. Es folgt

$$|\lambda - \mu| \leq \varepsilon |g_{ii}| + r_i$$

$$\leq \varepsilon^{1/\nu} (1 + \sum_{j=1}^{n} |g_{ij}|)$$

$$\leq \varepsilon^{1/\nu} (1 + \|G\|_\infty)$$

$$\leq \varepsilon^{1/\nu}(1 + k(X)\|F\|_\infty) \quad .$$

■

2) $J$ bestehe aus mehreren Jordan-Kästchen $J_1,\ldots,J_r$. Dann setzt man $D$ aus Diagonalmatrix $D_1,\ldots,D_r$ zusammen und hat dann

$$D^{-1}(J + G)D = \begin{pmatrix} D_1^{-1} J_1 D_1 & & \\ & \ddots & \\ & & D_r^{-1} J_r D_r \end{pmatrix} + \varepsilon D^{-1} G D \quad .$$

Anwendung von 1) ergibt die Behauptung.

■

LÖSUNG VON GLEICHUNGEN

## § 13 Existenz von Lösungen

Beispiele:

1) $f(x) = x^2 - 2px - q = 0$

1. Fall $d = p^2 + q > 0$ : 2 Nullstellen

2. Fall $d = 0$ 1 Nullstelle

3. Fall $d < 0$ Keine reelle Nullstelle

2) $f(x) = x^3 + 3px - 2q = 0$

allg. Form: $y^3 + \beta y^2 + \gamma y + \delta = 0$
Subst: $x = y + \beta/3$ (Tschirnhausen-Trafo)
dann ergibt sich die reduzierte Form,
wobei $p, q$ aus $\beta, \gamma, \delta$ berechenbar sind.

Für die Nullstellen gelten die "Cardanischen Formeln":
Tartaglia (1500 - 1557)

$$\bar{x}_1 = u + v \; ; \; \bar{x}_2 = \varepsilon_1 u + \varepsilon_2 v \; ; \; \bar{x}_3 = \varepsilon_2 u + \varepsilon_1 v$$

mit:

$$u = (q + \sqrt{d})^{1/3} \; ; \; v = (q - \sqrt{d})^{1/3} \; ; \; d = p^3 + q^2$$

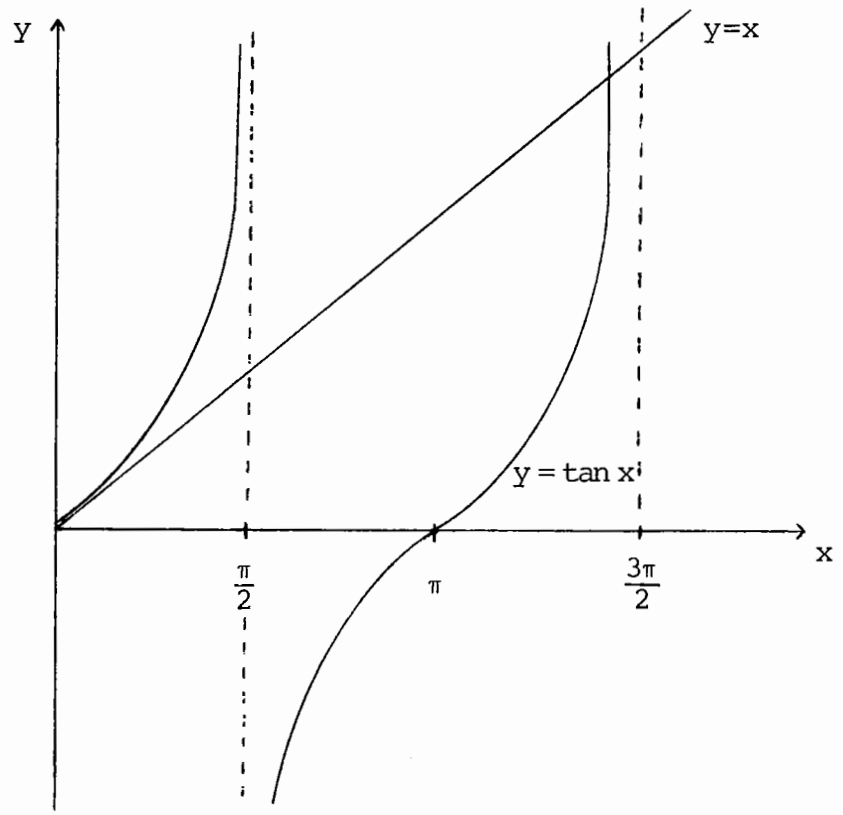
$$\varepsilon_{1/2} = -\frac{1}{2}(1 \pm i\sqrt{3})$$

Das Problem ist hier zwar analytisch gelöst, die Formeln helfen aber in der Praxis nicht viel. So gilt z.B. für $|p| << |q|$: $\sqrt{d} \approx |q|$, so daß bei der Berechnung von u oder v Auslöschung auftritt.

3) $f(x) = x - \tan x = 0$

Dieses Problem tritt bei der Berechnung der Schwingungen eines Balkens auf.

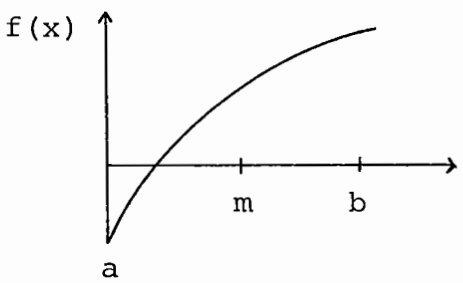
Einen Überblick über die Lösungen gibt die folgende Skizze:




$\bar{x}_o = 0$. In den Intervallen $(k\pi\ ,\ (k+\frac{1}{2})\pi)$ liegen Nullstellen $\bar{x}_k$, $k = 1,2,\ldots$ . Mit $\bar{x}_k$ ist auch $-\bar{x}_k$ Nullstelle.

Berechnung von Nullstellen:

Primitiv-Verfahren: Intervallhalbierung (Bisektion)




$f(a)\cdot f(b) < 0$, f stetig in [a,b] ⇒ f hat Nullstellen im Intervall [a,b]

Die Funktion wird in $m = \frac{1}{2}(a+b)$ ausgewertet, dann wählt man das Teilintervall, in dem eine Nullstelle liegen muß.

Das Verfahren ist vergleichsweise aufwendig und daher nur für sehr einfache Probleme geeignet.

Zur Verallgemeinerung auf höhere Dimensionen benötigen wir den Begriff des Fixpunktes:

Definition 13.1: $g : D \to D$ ; $x \in D$ heißt Fixpunkt von $g$, falls $g(x) = x$ .

Bemerkung: Wir untersuchen ab jetzt stets Gleichungen der Form $f(x) = x$. Jede Aufgabe $g(x) = 0$ läßt sich trivialerweise in der Form $g(x) + x = x$ als ein solches Fixpunktproblem formulieren.

Satz 13.1 (Brouwer): Sei $D \subseteq \mathbb{R}^n$ konvex und kompakt sowie $g : D \to D$ stetig. Dann hat $g$ in $D$ einen Fixpunkt.

Beweis: $n = 1$ , $D = [a,b]$

Wenn $g(a) = a$ oder $g(b) = b$, so ist bereits ein Fixpunkt gegeben. Also $g(a) > a$ und $g(b) < b$. Dann hat aber $f(x) = g(x) - x$ nach dem Zwischenwertsatz eine Nullstelle in $(a,b)$ und $g$ somit einen Fixpunkt.

Der Beweis für $n > 1$ findet sich in: E. Burger, Einführung in die Theorie der Spiele, 2. Aufl., S. 162-165, de Gruyter.

oder auch: Harro Heuser, Lehrbuch der Analysis II: 593-604

Beispiele:

1) $$x_1 - \sin(x_2 + e^{x_1}) = 0$$
$$x_2 - \cos(x_1 - e^{x_2}) = 0$$

Setze $g(x) = \begin{pmatrix} \sin(x_2 + e^{x_1}) \\ \cos(x_1 - e^{x_2}) \end{pmatrix}$, $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$

$D = \{x \in \mathbb{R}^2 : \|x\|_\infty \leq 1\}$ ist konvex und kompakt.

$g$ ist stetig in $D$ und es ist $g(D) \subseteq D$, also hat $g$ einen Fixpunkt $\bar{x} = \begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \end{pmatrix}$ in $D$, der das Gleichungssystem löst.

2) Typische Anwendung des Brouwer'schen Satzes in der VWL sind Gleichgewichtsprobleme: Ein Markt bestehe aus $n$ Gütern mit Preisen $x_1 \dots x_n \geq 0$ und aus $m$ Nachfragern. Bestehen die Preise $x = (x_1 \dots x_n)$, so kauft der $\ell$-te Nachfrager die Menge $f_i^\ell(x)$ von Gut $i$. Die totale Nachfrage nach Gut $i$ bei Preisen $x$ ist dann $f_i(x) = \sum_{\ell=1}^{m} f_i^\ell(x)$.

Man sagt, der Markt sei im Gleichgewicht, wenn

$$f_i(x) \leq 0 \quad , \quad i=1,\dots,n \quad , \quad f_i(x) = 0 \quad \text{falls} \quad x_i > 0.$$

Wir machen nun die Voraussetzungen:

1) Die Nachfrage hängt nur von den relativen Preisen ab, d.h. $f_i^\ell(tx) = f_i^\ell(x)$ für $t > 0$.

2) Der Markt ist abgeschlossen, d.h.

$$\sum_{i=1}^{n} x_i f_i^{\ell}(x) = 0 \quad , \quad \ell = 1,\ldots,m \quad .$$

3) Die $f_i^{\ell}$ sind stetig auf $S = \{x_i \geq 0 : \sum_{i=1}^{n} x_i = 1\}$.

SATZ 13.2: Unter den Voraussetzungen 1), 2), 3) gibt es einen Preisvektor $x$, bei welchem der Markt in Gleichgewicht ist.

BEWEIS:

$$g : S \to S \quad \text{mit} \quad g_i(x) = \frac{x_i + \text{Max}\,(0, f_i(x))}{1 + \sum_{k=1}^{n} \text{Max}\,(0, f_k(x))}$$

Nach dem Satz von Brouwer gibt es ein $\bar{x} \in S$ mit $g(\bar{x}) = \bar{x}$. Um nachzuweisen, daß $\bar{x}$ ein Gleichgewichtspunkt ist, bleibt zu zeigen: $\text{Max}\,(0, f_i(\bar{x})) = 0$; $i = 1,\ldots,n$. Dies ist eine einfache Übungsaufgabe.

## § 14 Iterationsverfahren

Iterationsverfahren gehören zu den wichtigsten Hilfsmitteln der praktischen Mathematik. Sie berechnen eine Folge von Näherungen, welche gegen die gesuchte Lösung konvergieren. Grundlage der Iterationsverfahren ist der Fixpunktsatz für kontrahierende Abbildungen.

Definition 14.1: Sei $D \subseteq R^n$ und $g : D \to R^n$ eine Abbildung. g heißt kontrahierend in D (bez. $\|\cdot\|$), falls es ein q mit $0 \leq q < 1$ gibt mit

$$\forall x, y \in D \quad \|g(x)-g(y)\| \leq q\|x-y\|$$

Beispiele: 1) Sei $D = [a,b]$ und g in D stetig differenzierbar mit $q = \operatorname{Max}_D |g'(x)| < 1$. Dann ist g in D kontrahierend (bez. Betrag als Norm).

2) Sei $D \subseteq R^n$ konvex und $g : D \to R^n$ in D stetig differenzierbar. Sei

$$g' = \begin{pmatrix} \frac{\partial g_1}{\partial x_1} & \cdots & \frac{\partial g_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial g_n}{\partial x_1} & \cdots & \frac{\partial g_n}{\partial x_n} \end{pmatrix}$$

die Funktionalmatrix von g. Sei $q = \operatorname{Max}_D \|g'(x)\|_\infty < 1$. Dann ist g in D kontrahierend (bez. $\|\cdot\|_\infty$).

Beweis: Sei $\varphi(\lambda) = g(\lambda x+(1-\lambda)y)$. Nach der Kettenregel ist

$$\varphi'(\lambda) = g'(\lambda x+(1-\lambda)y)(x-y)$$

also

$$\begin{aligned} \|g(x)-g(y)\|_\infty &= \|\varphi(1)-\varphi(0)\|_\infty \\ &\leq \operatorname{Max}_{[0,1]} \|\varphi'(\lambda)\|_\infty \\ &\leq \operatorname{Max}_{[0,1]} \|g'(\lambda x+(1-\lambda)y)(x-y)\| \\ &\leq \operatorname{Max}_D \|g'(x)\|_\infty \, \|x-y\|_\infty \\ &= q\|x-y\|_\infty \end{aligned}$$

Satz 14.1: (Kontraktionssatz, Fixpunktsatz von Banach):

Sei $D \subseteq R^n$ abgeschlossen und $g : D \to D$ in $D$ kontrahierend. Dann hat $g$ in $D$ genau einen Fixpunkt $\bar{x}$. Das Iterationsverfahren

$$x^{(k+1)} = g(x^{(k)}), \quad k = 0,1,\ldots$$

konvergiert für jede Wahl von $x^{(o)} \in D$ gegen $\bar{x}$, und es gilt

$$\|x^{(k)} - \bar{x}\| \leq \frac{q^k}{1-q} \|x^{(1)} - x^{(o)}\|.$$

Beweis: 1) Es ist

$$\|x^{(k+1)} - x^{(k)}\| \leq q \|x^{(k)} - x^{(k-1)}\|$$

$$\leq q^2 \|x^{(k-1)} - x^{(k-2)}\|$$

$$\leq q^k \|x^{(1)} - x^{(0)}\|,$$

also für $\ell > j$

$$\|x^{(\ell)} - x^{(j)}\| = \Big\| \sum_{k=j}^{\ell-1} (x^{(k+1)} - x^{(k)}) \Big\|$$

$$\leq \sum_{k=j}^{\ell-1} \|x^{(k+1)} - x^{(k)}\|$$

$$\leq \sum_{k=j}^{\ell-1} q^k \|x^{(1)} - x^{(o)}\|$$

$$= q^j (1+q+\ldots q^{\ell-1-j}) \|x^{(1)} - x^{(o)}\|$$

$$(*) \qquad \leq \frac{q^j}{1-q} \|x^{(1)} - x^{(o)}\|,$$

denn es gilt die Formel für die geometrische Reihe

$$\sum_{j=0}^{\infty} q^j = \frac{1}{1-q} \quad \text{für } |q| < 1.$$

Also hat man $\|x^{(\ell)} - x^{(j)}\| \to 0$ für $\ell, j \to \infty$. Die Folge $x^{(k)}$ ist also eine Cauchy-Folge, d. h.

$$\lim_{k\to\infty} x^{(k)} = \bar{x}$$

und $\bar{x} \in D$ weil $D$ abgeschlossen ist. Da $g$ stetig ist, gilt

$$\bar{x} \leftarrow x^{(k+1)} = g(x^{(k)}) \rightarrow g(\bar{x}),$$

d. h. $\bar{x} = g(\bar{x})$.

2) Eindeutigkeit: Seien $\bar{x}, \bar{\bar{x}}$ Fixpunkte von g in D. Dann ist

$$\|\bar{x}-\bar{\bar{x}}\| = \|g(\bar{x})-g(\bar{\bar{x}})\| \leq q\|\bar{x}-\bar{\bar{x}}\|$$

und aus $q < 1$ folgt $\bar{x} = \bar{\bar{x}}$.

3) Fehlerabschätzung: Läßt man in (*) $\ell \to \infty$ streben, so erhält man

$$\|\bar{x}-x^{(j)}\| \leq \frac{q^j}{1-q}\|x^{(1)}-x^{(0)}\| .$$

Beispiel: Wir versuchen, die Lösung von $x = \tan x$ in $[\pi, 3\pi/2]$ durch Iteration nach Satz 1$\beta_q$.1 zu berechnen und iterieren gemäß

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

Wir haben $g(x) = \tan x$, und es ist $g'(x) = \frac{1}{\cos^2 x} \geq 1$. g ist also in $[\pi, \frac{3\pi}{2}]$ nicht kontrahierend. Wir müssen unsere Gleichung erst in eine geeignete Form bringen. Dazu schreiben wir für $x \in [\pi, 3\pi/2]$

$$\begin{aligned} x = \tan x &\iff x = \tan(x-\pi) \\ &\iff \arctan x = x-\pi \\ &\iff x = \pi+\arctan x. \end{aligned}$$

Nun setzen wir

$$D = [\pi, \tfrac{3\pi}{2}] , \quad g(x) = \pi+\arctan x.$$

Offenbar ist $g(D) \subsetneq D$ und $g'(x) = \frac{1}{1+x^2}$, also g kontrahierend in D mit $q = \frac{1}{1+\pi^2} = 0.092 < 1$. Wir können also Satz 1$\beta_q$.1 anwenden und $\bar{x}$ berechnen:

| k | $x^{(k)}$ | $\frac{q^k}{1-q}\|x^{(1)}-x^{(0)}\|$ | $\bar{x}-x^{(k)}$ |
|---|---|---|---|
| 0 | 3.1416 | - | - |
| 1 | 4.4042 | 0.13 | 0.0892 |
| 2 | 4.4891 | 0.012 | 0.0843 |
| 3 | 4.4932 | 0.0011 | 0.0002 |
| 4 | 4.4934 | - | - |

Welche Fixpunkte kann man mit dem Iterationsverfahren berechnen? Der folgende Satz gibt darüber Auskunft:

Satz 14.2: (Lokaler Konvergenzsatz): $g : R^n \to R^n$ besitze einen Fixpunkt $\bar{x}$, und g sei kontrahierend in einer Umgebung von $\bar{x}$. Dann gibt es eine Umgebung $U(\bar{x})$, so daß das Iterationsverfahren $x^{(k+1)} = g(x^{(k)})$ für jedes $x^{(0)} \in U(\bar{x})$ gegen $\bar{x}$ konvergiert.

Beweis: Wir können annehmen, daß g in $D = \{x \in R^n : \|x-\bar{x}\| \leq r\}$, $r > 0$, kontrahiert sei mit der Konstanten q. Dann ist $g(D) \subseteq D$, denn für $x \in D$ ist

$$\|g(x)-\bar{x}\| = \|g(x)-g(\bar{x})\| \leq q\|x-\bar{x}\| \leq qr < r,$$

also auch $g(x) \in D$. Die Behauptung folgt nun aus Satz 13.1.

Wir wollen nun noch ein Maß für die Konvergenzgeschwindigkeit einführen:

Definition 14.2: Sei $x^{(k)}$ eine Folge in $R^n$ mit $\lim_k x^{(k)} = \bar{x}$.

Diese Konvergenz heißt von der Ordnung p (p=1: lineare Konvergenz, p=2 : quadratische Konvergenz), falls es eine Konstante C gibt mit

$$\|\bar{x}-x^{(k+1)}\| \leq C\|\bar{x}-x^{(k)}\|^p, \quad k = 0,1,\ldots .$$

Für p=1 wird $C < 1$ verlangt.

Satz 14.3: Die Konvergenz im Satz 13.2 ist linear.

Beweis: Es ist

$$\|\bar{x}-x^{(k+1)}\| = \|g(\bar{x})-g(x^{(k)})\| \leq q\|\bar{x}-x^{(k)}\| .$$

## § 15 Das Newton-Verfahren

Wir suchen eine systematische Methode, ein vorgegebenes Problem der Form:

$$\text{Gegeben: } f : \mathbb{R}^n \to \mathbb{R}^n \ ;$$
$$\text{gesucht: } \bar{x} \in \mathbb{R}^n \quad \text{mit} \quad f(\bar{x}) = 0$$

so zu formulieren, daß die Voraussetzungen des lokalen Konvergenzsatzes (Satz 15.2) automatisch erfüllt sind. Die Gleichung muß also als Fixpunktproblem formuliert werden, so daß das Iterationsverfahren konvergiert.

Das einfachste ist,

$$g(x) = x + f(x)$$

zu setzen. Dies ergibt zwar ein Fixpunktproblem, die Konvergenz des Iterationsverfahrens ist aber nicht gesichert. Sei daher:

$$g(x) = x + T(x)f(x)$$

mit einer geeignet zu wählenden, nichtsingulären (n,n)-Matrix T. $f(\bar{x}) = 0$ gilt genau dann, wenn $\bar{x}$ Fixpunkt von $g$ ist. Nach Satz 15.2 ist das Verhalten von $\|g'(x)\|_\infty$ für die Konvergenz des Iterationsverfahrens entscheidend. Wir berechnen die Ableitungen von

$$g_i(x) = x_i + \sum_{j=1}^{n} t_{ij}(x) f_j(x)$$

$$\frac{\partial g_i(x)}{\partial x_k} = \delta_{ik} + \sum_{j=1}^{n} \left( \frac{\partial t_{ij}(x)}{\partial x_k} f_j(x) + t_{ij}(x) \frac{\partial f_j(x)}{\partial x_k} \right)$$

(Dabei ist $e_{ik}$ das (i,k)-Element der Einheitsmatrix.)

Durch geeignete Wahl von $T$ wollen wir $\|g'(x)\|_\infty$ in der Nähe von $\bar{x}$ klein halten. In $\bar{x}$ gilt:

$$g'(\bar{x}) = I + T(\bar{x}) f'(\bar{x})$$

Wählen wir nun

$$T(\bar{x}) = - (f'(\bar{x}))^{-1} ,$$

so ist $g'(\bar{x}) = 0$, $\|g'(\bar{x})\|_\infty$ also klein in der Nähe von $\bar{x}$.

Da wir $\bar{x}$ nicht kennen, setzen wir

$$T(x) = - (f'(x))^{-1}$$

für alle $x$ und haben dann

$$g(x) = x - (f'(x))^{-1} f(x)$$

Dies führt auf das folgende Iterationsverfahren:

$$x^{k+1} = x^k - (f'(x^k))^{-1} f(x^k) \quad \text{"Newton-Verfahren"}$$

Beispiel: $f(x) = x^2 - 2$, d.h. Berechnung von $\bar{x} = \sqrt{2}$

$$g(x) = x - \frac{f(x)}{f'(x)} = x - \frac{x^2 - 2}{2x} = \frac{1}{2}\left(x + \frac{2}{x}\right)$$

$$x^{k+1} = \frac{1}{2}\left(x^k + \frac{2}{x^k}\right)$$

| k | $x_k$ | Anzahl der korrekten Dezimalen (erste falsche Dez. unterstrichen) |
|---|---|---|
| 0 | 1 | 1 |
| 1 | 1.5 | 1 |
| 2 | 1.417 | 3 |
| 3 | 1.414216 | 6 |
| 4 | 1.414213562 | 10 |

Bemerkung:

Praktisch benutzt man in höheren Dimensionen das Newton-Verfahren in der Form

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

So muß anstatt der Invertierung von $f'(x^k)$ ($n^3$ Operationen) nur noch ein lineares Gleichungssystem gelöst werden ($\frac{1}{3} n^3$ Operationen).

$f'(x^k)^{-1} f(x^k)$ wird als Lösung des Gleichungssystems

$$f'(x^k) \cdot X = f(x^k)$$

dann d. L-R-Zerlegung von $f'(x^k)$ in $\frac{1}{3}n^3$ R.O. berechnet.

Eine andere Herleitung des Newton-Verfahrens ist die auch ursprünglich von Newton verwendete Herleitung durch Linearisierung:

Nach dem Satz von Taylor gilt

$$f(x) = f(x^0) + f'(x^0)(x - x^0) + \varepsilon(x - x^0)$$

mit

$$\|\varepsilon(x - x^0)\| \leq c \cdot \|x - x_0\|^2$$

Für $\bar{x}$ gilt:

$$0 = f(x^0) + f'(x^0)(\bar{x} - x^0) + \varepsilon(\bar{x} - x^0)$$

Durch Vernachlässigung der in $\varepsilon$ zusammengefaßten Glieder höherer Ordnung ergibt sich die Gleichung

$$0 = f(x_0) + f'(x_0)(x^1 - x^0) \Leftrightarrow x^1 = x^0 - (f'(x^0))^{-1} f(x^0),$$

also das Newton-Verfahren.

Satz 15.1: $f : \mathbb{R}^n \to \mathbb{R}^n$ besitze die Nullstelle $\bar{x}$ und sei in einer Umgebung von $\bar{x}$ zweimal stetig differenzierbar. $f'(\bar{x})$ sei nichtsingulär.

Liegt $x^0$ hinreichend nahe bei $\bar{x}$, so konvergiert das Newton-Verfahren quadratisch gegen $\bar{x}$.

Beweis: Es gilt $g'(\bar{x}) = 0$, d.h. in einer Umgebung von $\bar{x}$ gilt $\|g'(x)\|_\infty \leq \frac{1}{2}$. Nach Satz 14.2 konvergiert die Folge $x^{k+1} = g(x^k)$ gegen $\bar{x}$, falls $x^0$ hinreichend nahe bei $\bar{x}$ gewählt wurde.

Zum Nachweis der quadratischen Konvergenz bilden wir

$$x^{k+1} - \bar{x} = x^k - \bar{x} - \left(f'(x^k)\right)^{-1} \left(f(x^k) - \underbrace{f(\bar{x})}_{=0}\right)$$

Der Satz von Taylor liefert:

$$f(\bar{x}) - f(x^k) = f'(x^k)(\bar{x} - x^k) + \varepsilon(\bar{x} - x^k)$$

mit

$$\|\varepsilon(\bar{x} - x^k)\| \leq M\|\bar{x} - x^k\|^2$$

Damit folgt:

$$\begin{aligned} x^{k+1} - \bar{x} &= x^k - \bar{x} + \left(f'(x^k)\right)^{-1}\left(f'(x^k)(\bar{x}-x^k) + \varepsilon(\bar{x} - x^k)\right) \\ &= x^k - \bar{x} + \bar{x} - x^k + \left(\left(f'(x^k)\right)^{-1} \varepsilon(\bar{x} - x^k)\right. \\ &= \left(f'(x^k)\right)^{-1} \varepsilon(\bar{x} - x^k) \end{aligned}$$

Da $f'(\bar{x})$ nichtsingulär ist, bleibt $f'(x)$ in einer Umgebung von $\bar{x}$ beschränkt. Für hinreichend große $k$ gilt dann:

$$\|f'(x^k)\|^{-1} \leq N < \infty$$

und damit

$$\|x^{k+1} - \bar{x}\| \leq N \cdot M \|x^k - \bar{x}\|^2 \quad . \qquad \blacksquare$$

Was passiert bei singulärem $f'(\bar{x})$ ? Wir betrachten den Fall $n = 1$: Sei $f'(\bar{x}) = 0$ aber $f'(x) \neq 0$ in einer Umgebung $V(\bar{x}) \setminus \{\bar{x}\}$.

Sei $f(x) = (x - \bar{x})^2 p(x)$ mit $p(\bar{x}) \neq 0$ .

Mit $f'(x) = 2(x-\bar{x})p(x) + (x-\bar{x})^2 p'(x)$

und $f''(x) = 2p(x) + 4(x-\bar{x})p'(x) + (x-\bar{x})^2 p''(x)$

erhalten wir:

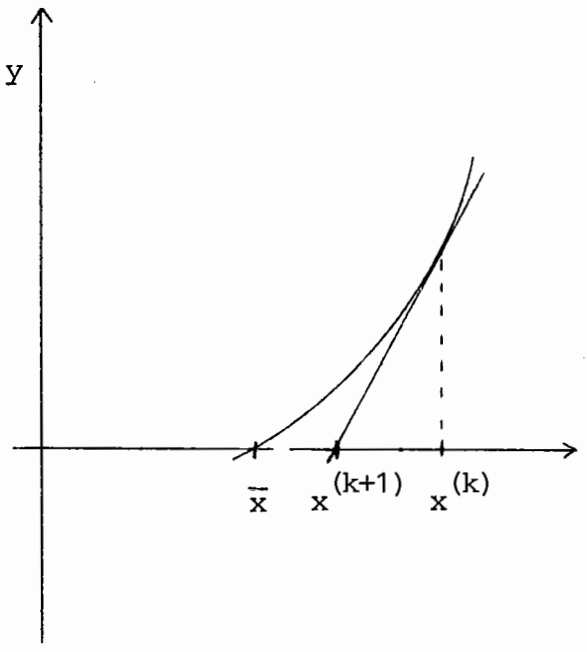
$$g'(x) = \frac{d}{dx}\left(x - \frac{f(x)}{f'(x)}\right) = \frac{f(x)f''(x)}{f'^2(x)} = \frac{(x-\bar{x})^2 p(x) \cdot 2p(x)(1+O(x-\bar{x}))}{4(x-\bar{x})^2 (p(x)+O(x-\bar{x}))^2}$$

$$= \frac{1}{2}(1 + O(x-\bar{x}))$$

Mit $g'(\bar{x}) = \frac{1}{2}$ folgt nach Satz 14.2 lineare Konvergenz. Für $f'(\bar{x}) = 0$ geht also die quadratische Konvergenz verloren.

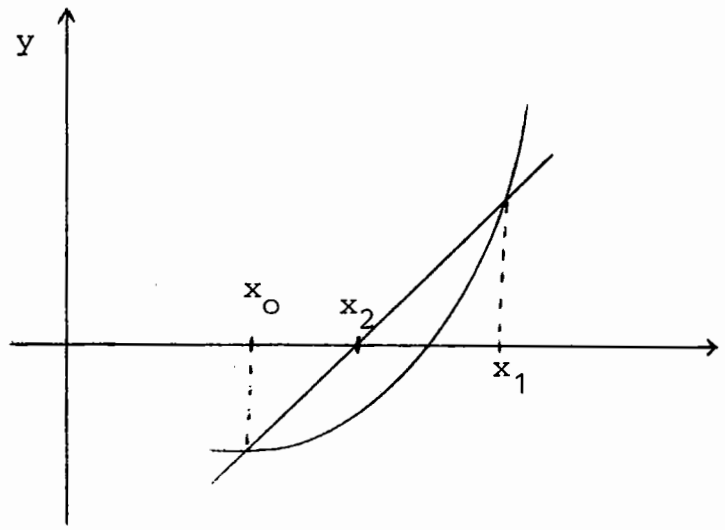
Geometrische Interpretation des Newton-Verfahrens:

Für $n = 1$ läßt sich das Newton-Verfahren geometrisch veranschaulichen:




$x^{(k+1)}$ ist die Nullstelle der Tangente $y = f(x^{(k)}) + f'(x^{(k)})(x - x^{(k)})$ an $y = f(x)$ in $x^{(k)}$.

Eine Vereinfachung des Newton-Verfahrens ist das Sekantenverfahren ("Regula falsi"):




Die Startwerte $x^{(0)}$ und $x^{(1)}$ werden willkürlich gewählt. Die Sekante wird durch

$$y_{(x)} = f(x^{(1)}) + (x - x^{(1)}) \frac{f(x^{(1)}) - f(x^{(0)})}{x^{(1)} - x^{(0)}}$$

beschrieben. Für die Nullstelle $x^{(2)}$ gilt:

$$x^{(2)} - x^{(1)} = - \left( \frac{f(x^{(1)}) - f(x^{(0)})}{x^{(1)} - x^{(0)}} \right)^{-1} f(x^{(1)})$$

Gegenüber dem Newton-Verfahren ist die Ableitung also durch einen Differenzenquotienten ersetzt worden. Man erhält die Iteration:

$$x^{(k+1)} = x^{(k)} - \frac{x^{(k)} - x^{(k-1)}}{f(x^{(k)}) - f(x^{(k-1)})} f(x^{(k)})$$

## § 16 Iterationsverfahren für lineare Gleichungssysteme

Das Standardverfahren zur Lösung linearer Gleichungssysteme ist das Eliminationsverfahren. Bei sehr großen Systemen wird der Rechenaufwand jedoch zu groß, hier zieht man Fixpunktverfahren vor.

Sei $Ax = b$ zu lösen. Wir zerlegen die (n,n)-Matrix $A$ in

$A = D + L + R$ mit:

$$D = \begin{pmatrix} a_{11} & & 0 \\ & \ddots & \\ 0 & & a_{nn} \end{pmatrix}, \quad L = \begin{pmatrix} 0 & & 0 \\ a_{21} & \ddots & \\ \vdots & \ddots & \ddots \\ a_{n1} & \cdots & a_{n,n-1} \quad 0 \end{pmatrix},$$

$$R = \begin{pmatrix} 0 & a_{12} & \cdots & a_{1n} \\ & \ddots & & \vdots \\ & & \ddots & a_{n-1,n} \\ 0 & & & 0 \end{pmatrix}$$

Von den vielen Iterationsverfahren, die zur Lösung von $(D + L + R)x = b$ entwickelt wurden, behandeln wir vier:

1) GS-Verfahren (Gesamtschritt- oder Jacobiverfahren)

   Es beruht auf der Iteration

$Dx^{k+1} + (L + R)x^k = b$ , $k=0,1,2,...$ mit einem Startvektor $x^o$.

Für jeden Iterationsschritt benötigt man $n^2$ Rechenoperationen und $n$ Divisionen. Werden weniger als $\frac{1}{3}n$ Schritte benötigt, ist das Verfahren schneller als das Eliminationsverfahren.

2) ES-Verfahren (Einzelschritt- oder Gauß-Seidel-Verfahren)

Mit der Iteration:

$(D + L)x^{k+1} + R\,x^k = b$ , $k=0,1,2,...$; $x^o$ Startvektor

Der Rechenaufwand ist genauso hoch wie beim GS-Verfahren. Der Unterschied zum GS-Verfahren besteht im Einsparen von Speicherplatz, wie die folgenden kleinen Programme zeigen:

```
procedure GS (x, k max);

comment   Führt k_max Schritte des GS-Verfahrens mit dem
          Startvektor x durch und schreibt x^k_max auf x

for k = 1 to k_max do
begin
x^o = x;
for i = 1 to n do x_i = (b_i - Σ_{j≠i} a_ij x^o_j)/a_ii;
end;
```

Man benötigt zusätzlichen Speicherplatz für den Vektor $x^0$. Dies wird beim ES-Verfahren vermieden:

```
procedure ES (x, k_max);
comment  Führt k_max Schritte des ES-Verfahrens mit dem
         Startvektor x durch und schreibt x^(k_max) auf x.
for k = 1 to k_max do
for i = 1 to n do x_i = (b_i - Σ_{j≠i} a_ij)/a_ii;
```

Betrachten wir die ersten Schritte des Verfahrens:

$$(D+L)x^{k+1} + Rx^k = b$$

$$a_{11}\, x_1^{k+1} + \sum_{j=2}^{n} a_{1j}\, x_j^k = b_1$$

$$a_{22}\, x_2^{k+1} + a_{21}\, x_1^{k+1} - \sum_{j=3}^{n} a_{2j}\, x_j^k = b_2 \qquad \text{u.s.w.}$$

Zur Berechnung der i-ten Komponente von $x^{k+1}$ benutzt das ES-Verfahren also bereits die vorher berechneten $x_1^{k+1}, \ldots, x_{i-1}^{k+1}$. Die Komponenten $x_1^k, \ldots, x_i^k$ werden nicht mehr benötigt und können daher überschrieben werden.

Diese Einsparung von $n$ Speicherplätzen fällt natürlich nur ins Gewicht, wenn die Matrix $A$, die $n^2$ Speicherplätze benötigt, nicht abgespeichert wird.

Beispiel: Dirichlet-Problem:

Gesucht ist eine in $Q = [0,1]^2$ stetige und im Innern zweimal stetig differenzierbare Funktion $u$ mit

$$-\Delta u = f \quad \text{in} \quad Q \qquad , \qquad \Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \quad .$$

$$u = 0 \quad \text{auf} \quad \text{Rand} \quad (Q)$$

Durch Diskretisierung entsteht ein lineares Gleichungssystem. Man betrachtet nur noch die Gitterpunkte $h\binom{i}{j}$, $i,j=0,\ldots,n$, $h = 1/n$ und ersetzt dort die Differentialgleichung durch

$$-4u_{ij} - (u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}) = h^2 f_{ij} \quad .$$

Dies ist ein lineares Gleichungssystem von $(n-1)^2$ Gleichungen für die $(n-1)^2$ Unbekannten $u_{ij}$, $0 < i,j < n$. Für kleine $h$ hofft man, daß $u_{ij}$ eine gute Näherung für $u$ im Gitterpunkt $h\binom{i}{j}$ ist.

Ein Schritt des Einzelschrittverfahrens lautet:

```
for i=1 to n-1 do
for j=1 to n-1 do
u_ij = (h^2 f_ij + u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1})/4;
```

Wir wollen nun die Konvergenz der Iterationsverfahren untersuchen. Zur Vorbereitung dient der folgende

Satz 16.1: Sei $B$ eine $(n,n)$-Matrix und

$$\rho(B) = \text{Max}\{|\lambda| : \lambda \text{ EW von } B\}$$

Dann gilt: $\rho(B) = \text{Inf}\{\|B\| : \|\cdot\| \text{ Norm}\}$

Beweis: Wir zeigen:

1) $\forall \|\cdot\| : \rho(B) \leq \|B\|$

2) $\forall \varepsilon > 0 \exists \|\cdot\| : \|B\| \leq \rho(B) + \varepsilon$

Zu 1): Sei $x^1$ EV von $B$ zum EW $\lambda$ mit $|\lambda| = \rho(B)$ und $\|x^1\| = 1$

Dann gilt: $\|B\| = \sup_{x \neq 0} \frac{\|Bx\|}{\|x\|} \geq \frac{\|Bx_1\|}{\|x_1\|} = \rho(B)$

Zu 2): Sei $J = X^{-1} B X$ die Jordan'sche Normalform von $B$.

Wir schreiben $J$ in der Form:

$$J = \begin{pmatrix} \lambda_1 & \theta_1 & & \\ & \ddots & \ddots & \\ & & \ddots & \theta_{n-1} \\ & & & \lambda_n \end{pmatrix}$$

wobei die $\lambda_i$ die EWe von $B$ und die $\theta_i$ aus $\{0,1\}$ sind.

Mit $\varepsilon > 0$ und

$$E = \begin{pmatrix} \varepsilon & & & 0 \\ & \varepsilon^2 & & \\ & & \ddots & \\ 0 & & & \varepsilon^n \end{pmatrix}$$ gilt:

$$E^{-1} J E = \begin{pmatrix} \lambda_1 & \varepsilon\theta_1 & & 0 \\ & \lambda_2 & \varepsilon\theta_2 & \\ & & \ddots & \\ & & & \varepsilon\theta_n \\ 0 & & & \lambda_n \end{pmatrix}$$

Wir definieren nun die Norm:

$$\|x\| = \|E^{-1} X^{-1} x\|_\infty$$

und berechnen damit

$$\|B\| = \max_{x\neq 0} \frac{\|Bx\|}{\|x\|} = \max_{x\neq 0} \frac{\|E^{-1} X^{-1} X J X^{-1} x\|_\infty}{\|\underbrace{E^{-1} X^{-1} x}_{y}\|_\infty}$$

$$= \max_{y\neq 0} \frac{\|E^{-1} J E y\|_\infty}{\|y\|_\infty} = \|E^{-1} J E\|_\infty$$

$$= \max_{i=1}^{n} (|\lambda_i| + \varepsilon\theta_i) \leq \rho(B) + \varepsilon$$

Satz 16.2: Das Iterationsverfahren

$$x^{k+1} = B x^k + c$$

konvergiert genau dann für jede Wahl von $x^0$ und $c$, wenn $\rho(B) < 1$. In diesem Fall ist $\bar{x} = \lim_{k\to\infty} x^k$ die eindeutige Lösung von $x = Bx + c$.

Beweis:

1) Das Verfahren konvergiere für beliebige $x^0$, $c$.

Setze $x^0 = c = x_1$ mit $Bx_1 = \lambda x_1$ und $|\lambda| = \rho(B)$

Dann gilt:

$$x^1 = Bx^0 + c = (\lambda + 1)x_1$$

$$x^k = (\lambda^k + \lambda^{k-1} + \ldots + \lambda + 1)x_1$$

Da die Folge $x^k$, $k=1,2,\ldots$ konvergiert, gilt $|\lambda| < 1$ und damit auch $\rho(B) < 1$.

2) Sei $\rho(B) < 1$. Nach Satz 16.1 gibt es eine Norm $\|\cdot\|$ mit $\|B\| < 1$. Damit ist

$g(x) := Bx + c$ kontrahierend in $\mathbb{C}^n$

Der Kontraktionssatz liefert dann die Behauptung.

Bemerkung zur Konvergenzgeschwindigkeit:

Es gilt: $e^k := x^k - \bar{x} = B(x^{k-1} - \bar{x}) = Be^{k-1} = B^k e^0$

Mit $B = XJX^{-1}$ gilt:

$$B^k = XJ^kX^{-1}$$

Nach § 9 gilt:

$$J^k = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix}^k M_k \quad ,$$

wobei $M_k$ ein Polynom in $k$ vom Grad $\leq \nu$ = maximale Länge der Jordan-Kästchen ist. Damit gilt:

$$\|B^k\| \leq k(X) \; (\rho(B))^k \; P_\nu(k) \xrightarrow[k\to\infty]{} 0$$

wobei $P_\nu$ wiederum ein Polynom vom Grad $\leq \nu$ ist.
Bis auf dieses Polynom wird die Konvergenzgeschwindigkeit von $(\rho(B))^k$ bestimmt.

Eine Konvergenzbeschleunigung erreicht man beim GS- bzw. ES-Verfahren durch die Einführung eines sogenannten Relaxationsfaktors $\omega$.

GS-Verfahren:

Anstelle der Iteration

$$x^{k+1} = D^{-1}(b - (L+R)x^k)$$

beim normalen GS-Verfahren, benutzt man

$$x^{k+1} = \omega D^{-1}(b - (L+R)x^k) + (1-\omega)x^k$$

Für $\omega = 1$ erhält man das alte GS-Verfahren zurück. Bei dieser Wahl von $\omega$ erwartet man, daß die neue Näherung deutlich besser als die vorhergehende ist. Wählt man $\omega < 1$ "traut man der neuen Näherung nicht", im Extremfall $\omega = 0$

erhält man $x^{k+1} = x^k$ .

Allgemein spricht man bei

$\omega < 1$ von Unterrelaxation

$\omega > 1$ von Überrelaxation

Für das ES-Verfahren erhalten wir:

$$x^{k+1} = \omega D^{-1} (b - L x^{k+1} - R x^k) + (1 - \omega) x^k$$

Hierauf beruht das SOR-Verfahren (Successive Overrelaxation), das in den 50er Jahren die Lösungsverfahren für lineare Gleichungssysteme revolutionierte und es ermöglichte, anstelle von Systemen mit $\sim$ 100 Variablen solche mit $\sim$ 1000 Variablen zu lösen.

Alle von uns betrachteten Iterationsverfahren haben die Form

$$x^{k+1} = B x^k + c \ .$$

Beim SOR-Verfahren ist beispielsweise

$$B = (D + \omega L)^{-1} ((1 - \omega) D - \omega R)$$

$$c = \omega (D + \omega L)^{-1} b$$

Zur Untersuchung der Konvergenz des SOR-Verfahrens können wir daher Satz 16.2 heranziehen.

Satz 16.3: Das SOR-Verfahren konvergiert bei beliebigem $x^0$ und $b$, falls $A$ positiv definit und $0 < \omega < 2$ ist.

Beweis: Zu zeigen: $\rho(B) < 1$ .

Sei x EV von B zum EW $\lambda$ mit $|\lambda| = \rho(B)$

Dann gilt:

$$((1-\omega)D - \omega R)x = \lambda(D + \omega L)x$$

woraus folgt:

$$(1-\omega)(Dx, x) - \omega(Rx, x) = \lambda[(Dx, x) + \omega(Lx, x)]$$

Wir führen die Abkürzungen

$$(Dx, x) = d \ , \ (Lx, x) = \ell$$

ein und beachten, daß wegen A pos. definit $R = L^*$ und damit $(Rx, x) = \overline{\ell}$ gilt. Damit ergibt sich:

$$(1-\omega)d - \omega\,\overline{\ell} = \lambda(d + \omega\,\ell)$$

$$\Rightarrow \quad \lambda = \frac{(1-\omega)d - \omega\overline{\ell}}{d + \omega\,\ell}$$

Sei $\ell = \alpha + i\,\beta$ , $\alpha, \beta \in \mathbb{R}$ . Dann gilt:

$$|\lambda|^2 = \frac{[(1-\omega)d - \omega\,\alpha]^2 + \omega^2\,\beta^2}{(d+\omega\alpha)^2 + \omega^2\,\beta^2}$$

$$|\lambda| < 1 \quad \Leftrightarrow \quad |(1-\omega)d - \omega\,\alpha| < |d + \omega\,\alpha|$$

$$\Leftrightarrow \quad |1 - \omega - \omega\,\alpha'| < |1 + \omega\,\alpha'|$$

$$\text{mit} \quad \alpha' = \frac{\alpha}{d}$$

Weiter gilt:

$$0 < (Ax, x) = d + \ell + \overline{\ell} = d + 2\alpha$$

$$\Rightarrow \alpha' = \frac{\alpha}{d} > -\frac{1}{2}$$

Mit $0 < \omega < 2$ gilt dann:

$$|1 + \omega\,\alpha'| = 1 + \omega\,\alpha'$$

und damit

$$|\lambda|^2 < 1 \Leftrightarrow |1 - \omega - \omega\,\alpha'| < 1 + \omega\,\alpha'$$

$$\Leftrightarrow \underbrace{-1 - \omega\,\alpha' < 1 - \omega - \omega\,\alpha'}\ < 1 + \omega\,\alpha'$$

$$\Leftrightarrow \omega < 2 \quad \wedge \quad -1 - \alpha' < \alpha'$$

$$\Leftrightarrow \omega < 2 \quad \wedge \quad \alpha' > -\frac{1}{2}$$

Damit gilt $\rho(B) < 1$ und der Satz ist bewiesen.

In der Praxis liegt das Problem darin, $\omega$ so zu bestimmen, daß $\rho(B)$ möglichst klein wird. Dies wird in der Vorlesung Praktische Mathematik II behandelt.

# INTERPOLATION UND APPROXIMATION

## § 17 Polynominterpolation

Definition: $\mathcal{P}_n$ bezeichne die Menge aller Polynome vom Grad $\leq n$:

$$\mathcal{P}_n := \left\{ \sum_{k=0}^{n} a_k x^k \,,\; a_k \in \mathbb{C} \right\}$$

Als Polynominterpolation bezeichnet man die folgende Aufgabe:

Gegeben: $x_0,\ldots,x_n \;,\; y_0,\ldots,y_n \in \mathbb{C}$

Gesucht: $P \in \mathcal{P}_n$ mit $P(x_j) = y_j \quad , j=0,\ldots,n$

Satz 17.1: $P$ ist eindeutig bestimmt, falls die $x_j$ paarweise verschieden sind.

Beweis: Mit

$$P(x_j) = \sum_{k=0}^{n} a_k x_j^k = y_j \qquad , j=0,\ldots,n$$

erhalten wir ein lineares Gleichungssystem für die Koeffizienten $a_0,\ldots,a_n$ von $P$. Dieses besitzt genau dann eine eindeutig bestimmte Lösung, wenn das homogene System

$$P(x_j) = \sum_{k=0}^{n} a_k x_j^k = 0 \qquad , j=0,\ldots,n$$

nur trivial lösbar ist. Dies ist aber offensichtlich der Fall,

da ein Polynom vom Grade $\leq n$ nur dann $n+1$ verschiedene Nullstellen haben kann, wenn es identisch verschwindet.

Bemerkung: Um das gesuchte Polynom zu berechnen, könnte man nun das Gleichungssystem lösen, beispielsweise mit der Cramer-'schen Regel. Dies führt auf die "Vandermond'sche Determinante":

$$V(x_o,\ldots,x_n) = \begin{vmatrix} 1 & \ldots & 1 \\ x_o & \ldots & x_n \\ x_o^2 & \ldots & x_n^2 \\ \vdots & & \vdots \\ x_o^n & \ldots & x_n^n \end{vmatrix} = \prod_{j>k} (x_j - x_k)$$

Für die Koeffizienten $a_k$ gilt dann:

$$a_k = \frac{\begin{vmatrix} 1 & y_o & 1 \\ x_o & y_1 & x_n \\ & \vdots & \\ x_o^n & y_n & x_n^n \end{vmatrix}}{V(x_o,\ldots,x_n)} \quad \text{(k-te Spalte)}$$

Zur praktischen Berechnung von $P$ stellen wir drei weniger aufwendige Methoden vor:

1) Die Form von Lagrange

(Mehr von theoretischer Bedeutung)

Man setzt

$$\omega_j(x) = \prod_{\substack{i=o \\ i \neq j}}^{n} \frac{x - x_i}{x_j - x_i}$$

Es gilt dann:

1) $\omega_j \in \mathcal{P}_n$

2) $\omega_j(x_k) = \begin{cases} 1 & k = j \\ 0 & \text{sonst} \end{cases}$

Damit ist das gesuchte Polynom:

$$P(x) = \sum_{j=o}^{n} y_j \, \omega_j(x)$$

<u>Beispiel:</u> $n = 2$

| $j$ | $x_j$ | $y_j$ |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 1 | 3 |
| 2 | 3 | 2 |

$$\omega_o(x) = \frac{(x-x_1)(x-x_2)}{(x_o-x_1)(x_o-x_2)} = \frac{1}{3}(x-1)(x-3)$$

$$\omega_1(x) = \frac{(x-x_o)(x-x_2)}{(x_1-x_o)(x_1-x_2)} = -\frac{1}{2}x(x-3)$$

$$\omega_2(x) = \frac{(x-x_o)(x-x_1)}{(x_2-x_o)(x_2-x_1)} = \frac{1}{6}x(x-1)$$

$$P(x) = \frac{1}{3}(x-1)(x-3) - \frac{3}{2}x(x-3) + \frac{1}{3}x(x-1)$$

$$= -\frac{5}{6}x^2 + \frac{17}{6}x + 1$$

In der Praxis möchte man häufig ein vorhandenes Interpolationspolynom durch Hinzunahme weiterer Stützstellen verbessern. Die Lagrange-Methode ist hierfür nicht geeignet, da man bei neu hinzukommenden Stützstellen mit der Rechnung von vorn beginnen muß. Dies ist bei den zwei folgenden Methoden nicht der Fall.

2) Die Rekursionsformel von Neville

Bezeichnung: Bei gegebenen $x_o,\ldots,x_n$, $y_o,\ldots,y_n$ bezeichnet $P_{i,i+1,\ldots,k}$ dasjenige Polynom aus $\mathcal{P}_{k-i}$, das an den Stellen $x_i,x_{i+1},\ldots,x_k$ die Werte $y_i, y_{i+1},\ldots, y_k$ annimmt.

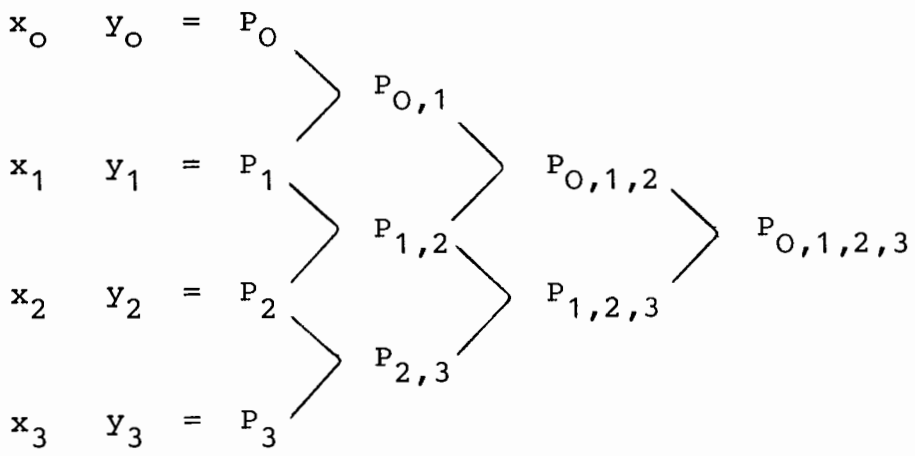
Für paarweise verschiedene $x_i,x_{i+1},\ldots,x_k$ ist $P_{i,\ldots,k}$ nach Satz 17.1 eindeutig bestimmt.

Satz 17.2: Seien $x_i,x_{i+1},\ldots,x_k$ paarweise verschieden. Dann gilt:

$$P_{i,\ldots,k}(x) = \frac{1}{x_k-x_i}\left((x-x_i)P_{i+1,\ldots,k}(x) + (x_k-x)P_{i,\ldots,k-1}(x)\right)$$

Beweis: Auf der rechten Seite steht ein Polynom vom Grad $\leq k-i$, das an den Stellen $x_i,\ldots,x_k$ die Werte $y_i,\ldots,y_k$ annimmt.

Diese Formel erlaubt die rekursive Berechnung des Interpolationspolynoms nach dem folgenden Schema (n = 3):




Bei Hinzunahme von weiteren Stützstellen wird das Schema einfach erweitert, ohne daß die bereits berechneten Ergebnisse ungültig werden.

Die Neville'sche Rekursion eignet sich mehr zur Auswertung des Interpolationspolynoms an einer bestimmten Stelle als zur Berechnung seiner Koeffizienten. Hierzu benutzt man die folgende Methode:

3) Newton'sche Form

Für $P = P_{o,\ldots,n}$ machen wir den folgenden Ansatz:

$$P_{o,\ldots,n}(x) = A_o + A_1(x-x_o) + A_2(x-x_o)(x-x_1) + \ldots$$

$$+ A_n(x-x_o) \cdot \ldots \cdot (x - x_{n-1})$$

Dies führt auf ein einfaches Gleichungssystem für die $A_i$:

$$P_{o,\ldots,n}(x_o) = A_o = y_o$$

$$P_{o,\ldots,n}(x_1) = A_o + A_1(x_1 - x_o) = y_1$$

$$P_{o,\ldots,n}(x_2) = A_o + A_1(x_2 - x_o) + A_2(x_2-x_o)(x_2-x_1) = y_2$$

$$\vdots$$

$$P_{o,\ldots,n}(x_n) = A_o + \ldots + A_n(x_n-x_o)\cdot\ldots\cdot(x_n-x_{n-1}) = y_n$$

Die $A_i$ können hieraus durch Vorwärtseinsetzen bestimmt werden, wir wollen jedoch explizite Formeln herleiten. Hierzu definieren wir rekursiv die sog. "Dividierten Differenzen" $[y_i \ldots y_k]$:

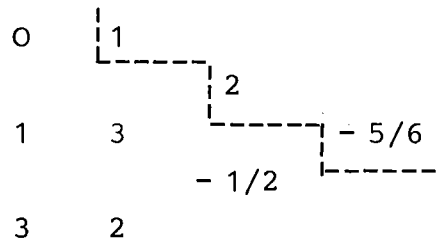
$$[y_i] = y_i \;,\; [y_i \ldots y_k] = \frac{1}{x_k - x_i}\left( [y_{i+1} \ldots y_k] - [y_i \ldots y_{k-1}] \right)$$

<u>Beispiel:</u> $[y_o\, y_1] = \dfrac{y_1 - y_o}{x_1 - x_o}$

Zur bequemen Berechnung der Dividierten Differenzen dient das sog. Differenzenschema:

$$\begin{array}{lllll}
x_o & [y_o] & & & \\
 & & [y_o\, y_1] & & \\
x_1 & [y_1] & & [y_o\, y_1\, y_2] & \\
 & & [y_1\, y_2] & & [y_o\, y_1\, y_2\, y_3] \\
x_2 & [y_2] & & [y_1\, y_2\, y_3] & \\
 & & [y_2\, y_3] & & \\
x_3 & [y_3] & & &
\end{array}$$

Beispiel: Für das bei der Lagrange-Form benutzte Beispiel lautet das Differenzenschema:




Satz 17.3: Für die Koeffizienten $A_0,\ldots,A_n$ der Newton'schen Form des Interpolationspolynoms gilt:

$$A_i = [y_0 \ldots y_i] \quad , \quad i = 0,\ldots,n \quad .$$

Beweis: Wir zeigen:

$$P_{i,\ldots,k}(x) = [y_i] + [y_i\, y_{i+1}]\,(x-x_i) + [y_i\, y_{i+1}\, y_{i+2}]\,(x-x_i)(x-x_{i+1})$$

$$+ \ldots + [y_i \ldots y_k](x-x_i)\cdot\ldots\cdot(x-x_{k-1})$$

durch Induktion nach $m = k - i$

$m = 0$: $P_i(x) = [y_i] = y_i$

Die Behauptung sei richtig für ein $m \geq 0$:

$$P_{i,\ldots,k}(x) = [y_i \ldots y_k]x^{k-i} + Q_1$$

$$P_{i+1,\ldots,k+1}(x) = [y_{i+1} \ldots y_{k+1}]x^{k-i} + Q_2$$

mit $Q_1, Q_2 \in \mathcal{P}_{k-i-1}$ .

Aus Satz 17.2 folgt:

$$P_{i,\ldots,k+1}(x) = \frac{1}{x_{k+1}-x_i}\Big((x-x_i)P_{i+1,\ldots,k+1}(x) + (x_{k+1}-x)P_{i,\ldots,k}(x)\Big)$$

$$= \frac{1}{x_{k+1}-x_i}\Big((x-x_i)\,[y_{i+1}\cdots y_{k+1}] + (x_{k+1}-x)[y_i\cdots y_k]\Big)x^{k-i} + R_1$$

$$= \frac{1}{x_{k+1}-x_i}\Big([y_{i+1}\cdots y_{k+1}] - [y_i\cdots y_k]\Big)x^{k-i+1} + R_2$$

$$= [y_i\cdots y_{k+1}]x^{k-i+1} + R_2$$

$$= [y_i\cdots y_{k+1}](x-x_i)(x-x_{i+1})\cdot\ldots\cdot(x-x_k) + R_3$$

mit $R_1, R_2, R_3 \in \mathcal{P}_{k-i}$

Für $R_3$ muß gelten:

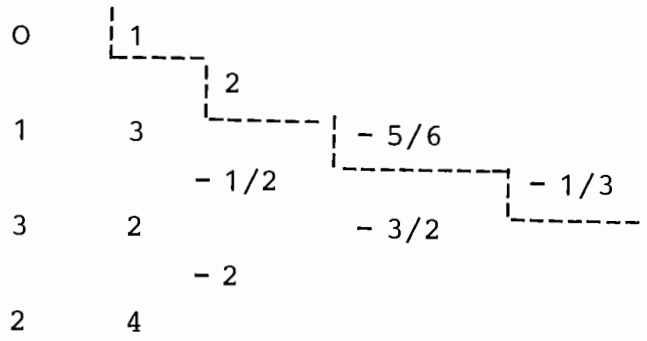
$$R_3(x_j) = P_{i,\ldots,k+1}(x_j) = y_j \qquad \text{für } j = i,\ldots,k$$

Damit ist $R_3 = P_{i,\ldots,k}$ und außerdem nach der Induktionsvor. von Newton'scher Form, womit die Behauptung bewiesen ist.

<u>Beispiel:</u> Das Newton'sche Interpolationspolynom für das obige Beispiel kann direkt der ersten Zeile des Differenzenschemas entnommen werden:

$$P(x) = 1 + 2x - 5/6\ x(x-1)$$

Will man Stützpunkte hinzunehmen, läßt sich das Differenzenschema leicht erweitern. Zu den Stützpunkten unseres Beispiels nehmen wir noch $x_3 = 2$ , $y_3 = 4$ hinzu und erhalten:



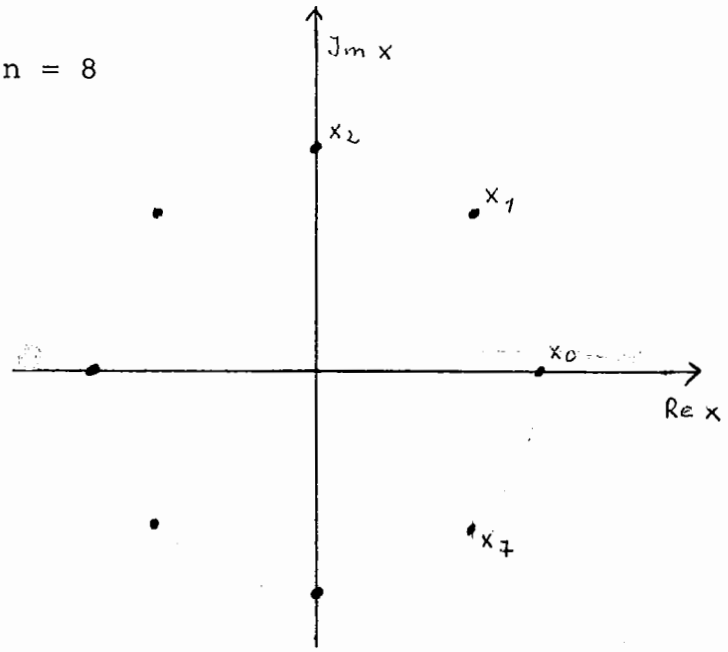

$$P(x) = 1 + 2x - \frac{5}{6}x(x-1) - \frac{1}{3}x(x-1)(x-3)$$

## § 18 Trigonometrische Interpolation

Wir betrachten in diesem Paragraphen einen wichtigen Spezialfall der Polynominterpolation, bei dem die Stützstellen in regelmäßigen Abständen auf dem komplexen Einheitskreis liegen:

$$x_j = e^{it_j} = \cos t_j + i \sin t_j \; ; \; t_j = 2\pi j/n \; ; \quad j=0,\ldots,n-1$$

Beispiel: n = 8




Nach Satz 17.1 gibt es ein eindeutig bestimmtes Polynom $P$ aus $\mathcal{P}_{n-1}$ mit $P(x_j) = y_j \; ; \; j=0,\ldots,n-1$

Die Koeffizienten von $P$ bezeichnen wir mit $\hat{y}_k$:

$$P(x) = \sum_{k=0}^{n-1} \hat{y}_k \, x^k$$

Sei $y = (y_0,\ldots,y_{n-1})^T$ und $\hat{y} = (\hat{y}_0,\ldots,\hat{y}_{n-1})^T$. Dann können wir die Interpolationsaufgabe

$$y_j = \sum_{k=0}^{n-1} \hat{y}_k \, x_j^k \qquad ; \quad j = 0,\ldots,n-1$$

in die Form

$$y = \begin{pmatrix} 1 & x_o^{=1} & \dots & x_o^{n-1} \\ 1 & x_1 & \dots & x_1^{n-1} \\ \vdots & & & \\ 1 & x_{n-1} & \dots & x_{n-1}^{n-1} \end{pmatrix} \hat{y} = W\hat{y}$$

umschreiben. Die Inversion der Matrix $W$ ist sehr einfach:

Satz 18.1: Es gilt: $W W^* = n I$

Beweis: $(W W^*)_{k\ell} = \sum_{j=o}^{n-1} x_k^j \, \bar{x}_\ell^j \qquad ; \quad k,\ell = o,\dots,n-1$

$$= \sum_{j=o}^{n-1} e^{i(t_k - t_\ell)j}$$

$$= \sum_{j=o}^{n-1} e^{2\pi i(k-\ell)j/n}$$

$$= \sum_{j=o}^{n-1} q^j \qquad \text{mit} \quad q = e^{2\pi i(k-\ell)/n}$$

$$= \begin{cases} \frac{q^n - 1}{q - 1} & q \neq 1 \\ n & q = 1 \end{cases}$$

$$= \begin{cases} o & k \neq \ell \\ n & k = \ell \end{cases}$$

Damit haben wir gleichzeitig die Orthogonalität der trigonometrischen Funktionen gezeigt:

$$\frac{1}{n}\sum_{j=o}^{n-1} e^{2\pi ijk/n} = \begin{cases} 1 & k = 0 \ , \ \pm n, \ \pm 2n \ \ldots \\ 0 & \text{sonst} \end{cases}$$

Wir folgern:

$$W^{-1} = \frac{1}{n} W^*$$

und damit

$$\hat{y} = \frac{1}{n} W^* y$$

In Komponenten:

$$\hat{y}_k = \frac{1}{n}\sum_{j=o}^{n-1} e^{-2\pi ijk/n} y_j \qquad (1)$$

$$y_j = \sum_{k=o}^{n-1} e^{2\pi ijk/n} \hat{y}_k \qquad (2)$$

(1) heißt diskrete Fouriertransformation der Länge $n$ ,
(2) heißt dementsprechend inverse diskrete Fouriertransformation der Länge $n$ . Beide werden in der praktischen Mathematik sehr häufig angewandt. Man programmiert jedoch nicht nach den Formeln (1), (2), was jeweils $n^2$ komplexe Rechenoperationen beanspruchen würde, sondern benutzt erheblich schnellere Algorithmen. Einen davon werden wir weiter unten kennenlernen.

Zunächst stellen wir jedoch die Beziehung zum Titel dieses Paragraphen her und drücken $P(x)$ durch trigonometrische Funktionen aus:

Wir setzen:

$$a_k = \frac{2}{n} \sum_{j=o}^{n-1} y_j \cos(2\pi kj/n)$$

$$b_k = \frac{2}{n} \sum_{j=o}^{n-1} y_j \sin(2\pi kj/n)$$

Dann ist

$$\hat{y}_k = \frac{1}{n} \sum_{j=o}^{n-1} y_j \, e^{-2\pi ijk/n}$$

$$= \frac{1}{n} \sum_{j=o}^{n-1} y_j \, (\cos(2\pi jk/n) - i \sin(2\pi jk/n))$$

$$= \frac{1}{2} \, (a_k - i\, b_k)$$

Man beachte, daß dies keine Zerlegung in Real- und Imaginärteil von $\hat{y}_k$ darstellt, da die $a_k$ und $b_k$ nicht notwendig reell sind. Es gilt:

$a_{n-k} = a_k$ und $b_{n-k} = - b_k$ , also ist

$$\hat{y}_{n-k} = \frac{1}{2} \, (a_k + ib_k)$$

Sei $n$ nun ungerade, also $n = 2m + 1$. Dann erhalten wir

$$P(x_j) = y_j = \sum_{k=o}^{n-1} \hat{y}_k \, e^{2\pi ijk/n}$$

$$= \hat{y}_o + \sum_{k=1}^{m} \hat{y}_k \, e^{2\pi ijk/n} + \sum_{k=m+1}^{n-1} \hat{y}_k \, e^{2\pi ijk/n}$$

$$= \hat{y}_o + \sum_{k=1}^{m} \hat{y}_k \, e^{2\pi ijk/n} + \sum_{k=1}^{m} \hat{y}_{n-k'} \, e^{2\pi ij(n-k')/n} \qquad (k' = n-k)$$

$$= \hat{y}_o + \sum_{k=1}^{m} \left( \hat{y}_k \, e^{2\pi ijk/n} + \hat{y}_{n-k} \, e^{-2\pi ijk/n} \right)$$

$$= \frac{a_o}{2} + \frac{1}{2} \sum_{k=1}^{m} \{ (a_k - i\, b_k)(\cos(2\pi jk/n) + i \sin(2\pi jk/n))$$

$$+ (a_k + i\, b_k)(\cos(2\pi jk/n) - i \sin(2\pi jk/n)) \}$$

$$\Rightarrow \quad P(x_j) = y_j = \frac{a_o}{2} + \sum_{k=1}^{m} (a_k \cos(2\pi jk/n) + b_k \sin(2\pi jk/n)) \qquad (*)$$

Für gerades $n$ , $n = 2m$, erhält man analog:

$$P(x_j) = y_j = \frac{a_o}{2} + \sum_{k=1}^{m-1} (a_k \cos(2\pi jk/n) + b_k \sin(2\pi jk/n))$$

$$+ \frac{a_m}{2} \cos(2\pi jm/n) \qquad (**)$$

wenn man beachtet, daß $b_m$ verschwindet.

Wir bezeichnen die Menge

$$\mathcal{T}_m = \left\{ \frac{a_o}{2} + \sum_{k=1}^{m} (a_k \cos kt + b_k \sin kt) \; ; \; a_k, \, b_k \in \mathbb{C} \right\}$$

als die Menge der trigonometrischen Polynome vom Grade $m$ .

Damit haben wir die folgende Aussage über die trigonometrische Interpolation gewonnen:

Satz 18.2: Gesucht sei $T \in \mathcal{T}_m$ mit

$$T(t_j) = y_j \quad , \quad j = 0,\ldots,n-1 \ ,$$

$$t_j = 2\pi j/n \ .$$

Für $n = 2m+1$ ist die Aufgabe eindeutig lösbar und die Lösung ist durch (*) gegeben.

Für $n = 2m$ gibt es genau eine Lösung mit $b_m = 0$, die durch (**) gegeben ist.

Wir betrachten nun einen effizienten Algorithmus zur Berechnung von $\hat{y}$, die schnelle Fouriertransformation (FFT) von Cooley - Tukey.

Sei $n$ gerade, $n = 2m$. Dann gilt:

$$\hat{y}_k = \sum_{j=0}^{n-1} y_j q^{jk} \quad \text{mit} \quad q = e^{-2\pi i/n}$$

Es gilt: $q^n = 1$ und $q^m = q^{n/2} = e^{-\pi i} = -1$

Die Idee ist nun, die Summe nach geraden und ungeraden Indizes zu zerlegen:

$$\hat{y}_k = \sum_{\substack{\ell=0 \\ j=2\ell}}^{m-1} (q^2)^{\ell k} y_{2\ell} + \sum_{\substack{\ell=0 \\ j=2\ell+1}}^{m-1} q^{(2\ell+1)k} y_{2\ell+1}$$

$$= \sum_{\ell=0}^{m-1} (q^2)^{\ell k} y_{2\ell} + q^k \sum_{\ell=0}^{m-1} (q^2)^{\ell k} y_{2\ell+1}$$

$$= g_k + q^k u_k$$

Wir sehen nun, daß sich $g_k$ und $u_k$ als Fouriertransformationen der Länge $m = n/2$ berechnen, da gilt:

$$q^2 = e^{-2\pi i/(n/2)} .$$

Weiter erhalten wir:

$$g_{k+m} = g_k$$

$$u_{k+m} = u_k$$

und damit

$$\left.\begin{aligned} \hat{y}_k &= g_k + q^k u_k \\ g_k + q^{k+m} u_k &= \hat{y}_{k+m} = g_k - q^k u_k \end{aligned}\right\} \quad k = 0,\ldots,m-1$$

Sei nun $M_p$ die Anzahl der komplexen Multiplikationen und $A_p$ die Anzahl der komplexen Additionen, die für eine schnelle Fouriertransformation der Länge $n = 2^p$ benötigt werden. Wenn wir die Berechnung von $q^k$ vernachlässigen, erhalten wir:

$$p = 0 \Rightarrow M_p = A_p = 0$$

$$M_{p+1} = 2 \cdot M_p + 2^p$$

$$A_{p+1} = 2 \cdot A_p + 2^{p+1}$$

$$\Rightarrow \quad M_p = \frac{1}{2}\, p\, 2^p = \frac{1}{2}\,(\log_2 n)\cdot n$$

$$A_p = p\, 2^p = (\log_2 n)\cdot n$$

Damit gilt der

Satz 18.3: Die Fouriertransformation der Länge $n = 2^p$ kann durch $\frac{1}{2}\, n(\log_2 n)$ komplexe Multiplikationen und $n\,(\log_2 n)$ komplexe Additionen berechnet werden.

Programm zur schnellen Fouriertransformation:

```
procedure FFT (y,n);

comment   berechnet ŷ der Länge n und schreibt ŷ
          auf y , n = 2^p;

if n > 1  then
          begin m=n/2;
          for ℓ=0 to m-1 do g[ℓ]=y[2ℓ]; u[ℓ]=y[2ℓ+1];
          FFT (g,m); FFT (u,m);
          for k=0 to m-1 do  u[k]=q^k*u[k];
                             y[k]=g[k]+u[k];
                             y[k+m]=g[k]-u[k];
          end FFT;
```

Die FFT nach Cooley-Tukey ist ein typisches Beispiel für das "divide and conquer" - Prinzip der Informatik:

(a) Zerlege das Problem in Teilprobleme

(b) Löse die Teilprobleme

(c) Setze die Lösungen der Teilprobleme zur Lösung des ganzen Problems zusammen.

Beispiele: $n = 1$

$$\hat{y}_o = y_o$$

$n = 2$

$$\hat{y}_o = y_o + y_1$$

$$\hat{y}_1 = y_o - y_1$$

$n = 4$

$$\hat{y}_o = g_o + u_o \qquad q = e^{-2\pi i/4} = -i$$

$$\hat{y}_1 = g_1 - i\,u_1$$

$$\hat{y}_2 = g_o - u_o$$

$$\hat{y}_3 = g_1 + i\,u_1$$

$$g_o = y_o + y_2 \qquad u_o = y_1 + y_3$$

$$g_1 = y_o - y_2 \qquad u_1 = y_1 - y_3$$

## § 19 Der Interpolationsfehler

Seien in einem Intervall $[a,b]$ $n+1$ paarweise verschiedene Stützstellen $x_0,\ldots,x_n$ und eine Funktion $f \in C^{n+1}[a,b]$ gegeben. Wir wollen $f(x)$ für $x \neq x_i$ approximieren.

Betrachten wir das eindeutig bestimmte Polynom $p \in \mathcal{P}_n$ mit $p(x_j) = f(x_j)$ , $j = 0,\ldots,n$ . Der folgende Satz macht eine Aussage über den Fehler, der bei einer Approximation von $f$ durch $p$ auftritt:

Satz 19.1: Zu jedem $x \in [a,b]$ existiert ein $\tilde{x} \in [a,b]$, so daß gilt:

$$f(x) - p(x) = w(x) \frac{f^{(n+1)}(\tilde{x})}{(n+1)!}$$

$$\text{mit:} \quad w(x) = \prod_{j=0}^{n} (x - x_j)$$

Beweis: Sei $\bar{x} \in [a,b]$ beliebig gewählt mit $\bar{x} \neq x_j$, $j=0,\ldots,n$. Wir setzen

$$F(x) = f(x) - p(x) - K w(x) \quad \text{mit einer Konstanten } K .$$

$$\Rightarrow \quad F(x_j) = 0 \; ; \; j = 0,\ldots,n$$

Wir wählen $K$ nun so, daß auch $F(\bar{x})$ verschwindet, also

$$K = \left( \frac{f - p}{w} \right)(\bar{x})$$

Damit hat $F$ in $[a,b]$ mindestens die $n+2$ Nullstellen $\bar{x}, x_0, \ldots, x_n$. Aus dem Satz von Rolle folgt, daß dann $F^{(n+1)}$ mindestens eine Nullstelle $\tilde{x}$ in $[a,b]$ besitzt.

Aus der Beziehung

$$F^{(n+1)}(x) = f^{(n+1)}(x) - K(n+1)!$$

folgt

$$K = \left( \frac{f - p}{w} \right) (\bar{x}) = \frac{f^{(n+1)}(\tilde{x})}{(n+1)!}$$

Also:

$$f(\bar{x}) - p(\bar{x}) = w(\bar{x}) \frac{f^{(n+1)}(\tilde{x})}{(n+1)!}$$

Diese Beziehung ist offenbar auch für $\bar{x} = x_j$, $j=0,\ldots,n$ erfüllt.

Für den Interpolationsfehler erhalten wir die Abschätzung

$$|f(x)-p(x)| \leq |w(x)| \max_{x\in[a,b]} \frac{|f^{(n+1)}(x)|}{(n+1)!}$$

Wir betrachten zuerst den Fall gleichmäßig verteilter Stützstellen:

1) Sei $x_j = a + jh$ mit der „Schrittweite" $h = \frac{b-a}{n}$

Für $x = a + \theta h$, $0 \leq \theta \leq n$ gilt:

$$w(x) = h^{n+1} \prod_{j=0}^{n} (\theta - j) \qquad \text{und}$$

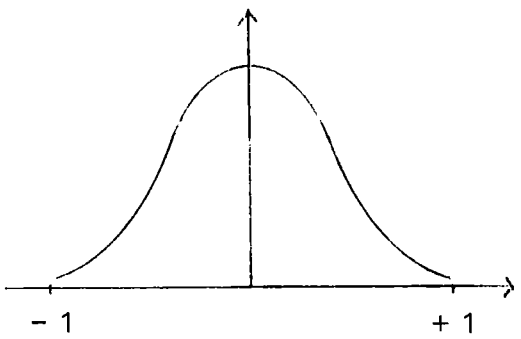
$$|f(x) - p(x)| \leq \frac{h^{n+1}}{(n+1)!} \left( \prod_{j=0}^{n} |\theta - j| \right) \operatorname{Max}_{x \in [a,b]} |f^{(n+1)}(x)|$$

(a) Für $b - a \to 0$ bei festem $n$ erhalten wir

$$f(x) - p(x) = O(h^{n+1})$$

(b) Der Fall $n \to \infty$ bei fester Intervallänge $b - a$ führt i.a. zu keiner Konvergenz. Wir betrachten hierzu das Beispiel von Runge:

$$f(x) = (1 + 25\,x^2)^{-1} \qquad \text{in} \quad [-1,1]$$




$f$ wird an den Stellen $x_j = -1 + \frac{2j}{n}$, $j=0,\ldots,n$ durch ein Polynom vom Grad $n$ interpoliert. Die folgende Tabelle zeigt, daß der Interpolationsfehler für große $n$ stark anwächst.

| n | $\max_{x\in[-1,1]} \lvert f(x)-p(x)\rvert$ |
|---|---|
| 1 | 0,96 |
| 5 | 0,43 |
| 13 | 1,07 |
| 19 | 8,57 |

2) Wir wählen die Stützstellen $x_0,\ldots,x_n$ nun so, daß $\max_{[a,b]} |w(x)|$ möglichst klein ist.

Für $[a,b] = [-1,1]$ erhalten wir

$$w(x) = 2^{-n}\, T_{n+1}(x)$$

wobei für $x \in [-1,1]$ die "Tschebyscheff-Polynome" $T_n$ wie folgt definiert sind:

$$T_n(x) = \cos n t \quad , \quad x = \cos t \quad (t = \arccos x) \quad , \quad 0 \le t \le \pi$$

Es gilt: $T_0(x) = 1$ und $T_1(x) = x$ , sowie für $n \ge 1$:

$$T_{n+1}(x) = \cos(n+1)t = \cos n t \cdot \cos t - \sin n t \cdot \sin t$$

$$T_{n-1}(x) = \cos(n-1)t = \cos n t \cdot \cos t + \sin n t \cdot \sin t$$

Addition der Gleichungen ergibt:

$$T_{n+1}(x) + T_{n-1}(x) = 2\,x\,T_n(x)$$

oder

$$T_{n+1} = 2\,x\,T_n - T_{n-1}$$

Da $T_o$ und $T_1$ Polynome sind, sind es auch alle anderen $T_n$. Die Rekursion zeigt weiter, daß $T_n$ für $n \geq 1$ die Form

$$T_n(x) = 2^{n-1}\,x^n + \text{Polynom} \in \mathcal{P}_{n-1}$$

haben muß.

$w(x) = 2^{-n}\,T_{n+1}(x)$ hat daher den Höchstkoeffizienten 1 und es gilt:

$$|w(x)| \leq 2^{-n} \quad \text{in} \quad [-1,1]$$

Die Nullstellen von $w$ sind unsere neuen Stützstellen:

$$\begin{aligned} w(x) = 0 &\Leftrightarrow T_{n+1}(x) = 0 \\ &\Leftrightarrow \cos(n+1)t = 0 \\ &\Leftrightarrow t = \frac{(j+\frac{1}{2})\pi}{n+1} \qquad j = 0,\ldots,n \\ &\Rightarrow x_j = \cos\frac{(j+\frac{1}{2})\pi}{n+1} \qquad j = 0,\ldots,n \end{aligned}$$

Bei dieser Wahl der Stützstellen ergibt sich die Fehlerabschätzung

$$|f(x) - p(x)| \leq \frac{|\text{Max } f^{(n+1)}(x)|}{2^n\,(n+1)!}$$

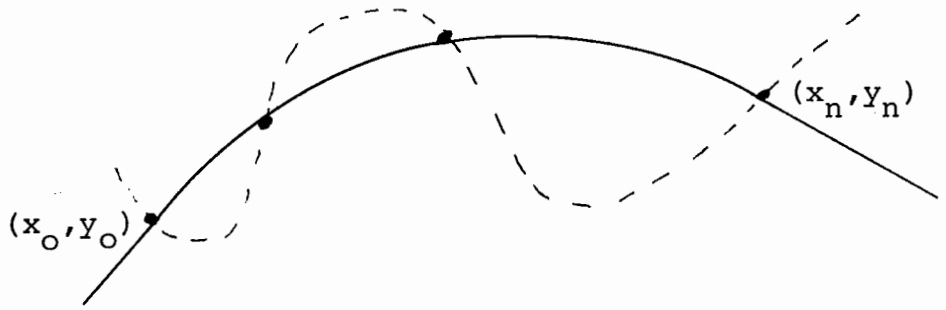
Für das obige Beispiel ergeben sich folgende Werte:

| n | $\underset{[-1,1]}{\text{Max}} \lvert f(x) - p(x)\rvert$ |
|---|---|
| 1 | 0.93 |
| 5 | 0.56 |
| 13 | 0.12 |
| 19 | 0.04 |

Die Verbesserung ist erheblich, die Approximation aber immer noch unbefriedigend.

## § 20 Spline - Interpolation

Wir suchen eine möglichst glatte Kurve, die wir durch die vorgegebenen Punkte $(x_j,y_j)$ der Ebene legen wollen. Dieses Problem tritt im Schiffsbau auf, wo der Verlauf des Rumpfes möglichst glatt an die Querstrebungen angepaßt werden muß. Die Schiffsbauer benutzen hierzu eine Straklatte (engl.: spline), d.h. einen dünnen elastischen Stab, den sie mit Gewichten zwingen, durch die vorgebenen Punkte zu gehen:




Der ungefähre Verlauf des Interpolationspolynoms ist gestrichelt gezeichnet. Man sieht sofort, daß dies keine Lösung unseres Problems sein kann. Für die gesuchte Kurve $s(x)$ fordern wir die folgenden Eigenschaften:

(a) $\quad s(x_j) = y_j \qquad j = 0,\ldots,n$

(b) $\quad s \in C^2 \;,\; s \in C^\infty(x_j,x_{j+1}) \qquad ,\; j = 0,\ldots,n-1$

(c) $\quad s$ linear außerhalb von $[x_o,x_n]$

Die gesuchte Funktion soll außerdem minimale "Gesamtkrümmung" haben. Für kleine Auslenkungen ist die Krümmung von $s$ proportional zu $s''$.

Das führt auf die Forderung

(d) Für alle $t$, die (a), (b) und (c) erfüllen, soll gelten:

$$\int_{x_o}^{x_n} (s''(x))^2 \, dx \leq \int_{x_o}^{x_n} (t''(x))^2 \, dx$$

Satz 20.1.: $s$ erfülle die Bedingungen (a) bis (d). Dann stimmt $s$ in jedem Intervall $[x_j, x_{j+1}]$ mit einem Polynom vom Grade $\leq 3$ überein.

Beweis: Wir betrachten den Vektorraum

$$H = \left\{ s \in C^2(\mathbb{R}^1) : s \text{ linear außerhalb von } [x_o, x_n] \right\},$$
$$s \in C^\infty(x_j, x_{j+1}), \quad j = 0, \ldots, n-1$$

auf dem wir mit

$$(s,t)_H = \int_{x_o}^{x_n} s'' \, t'' \, dx \quad \text{und} \quad \|s\|_H^2 = (s,s)_H$$

ein Skalarprodukt und eine Norm definieren.

Es gilt:

$$\|s\|_H = 0 \Leftrightarrow s'' = 0 \text{ in } [x_o, x_n] \Leftrightarrow s \text{ linear in } \mathbb{R}$$

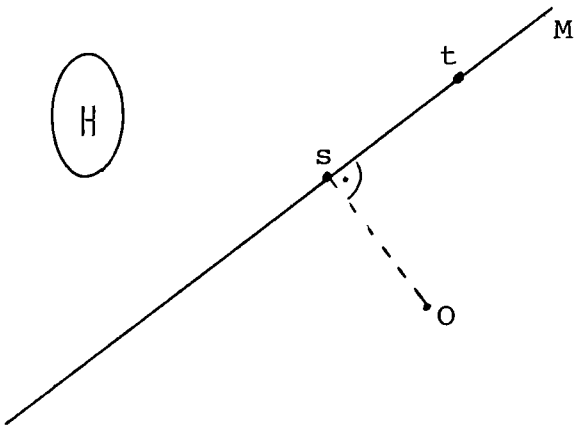
Das Skalarprodukt ermöglicht uns, auf $H$ den Begriff "Orthogonalität" einzuführen.

$$s \perp t \Leftrightarrow (s,t)_H = 0$$

Aus der Menge

$$M = \left\{ s \in H:\ s(x_j) = y_j \quad ; \quad j = 0,\ldots,n \right\}$$

suchen wir nun das Element $s$ mit minimaler Norm:




Wie die Skizze zeigt, gilt für das gesuchte $s$ :

$$\forall\ t \in M:\ s \perp s - t \ , \text{ also}$$

$$\forall\ t \in M:\ (s, s - t)_H = 0$$

$$\Leftrightarrow \quad \int_{x_0}^{x_n} s''(s - t)''\, dx = 0$$

Wir wählen nun $t = s$ außerhalb eines beliebig herausgegriffenen Teilintervalls $[x_i, x_{i+1}]$ und erhalten:

$$\int_{x_i}^{x_{i+1}} s''(s - t)''\, dx = 0$$

Wir wollen nun partiell integrieren. Dies ist möglich, weil $s$ in $[x_j, x_{j+1}]$ glatt ist.

Aus Stetigkeitsgründen muß gelten:

$$(s-t)^{(\nu)}(x_i) = (s-t)^{(\nu)}(x_{i+1}) = 0 \quad \nu = 0,1,2$$

Dann gilt:

$$0 = \int_{x_i}^{x_{i+1}} s''(s-t)''dx = -\int_{x_i}^{x_{i+1}} s'''(s-t)'dx = \int_{x_i}^{x_{i+1}} s^{(4)}(s-t)dx$$

woraus

$$s^{(4)} = 0 \quad \text{in} \quad [x_i, x_{i+1}]$$

folgt, was äquivalent zur Behauptung des Satzes ist.

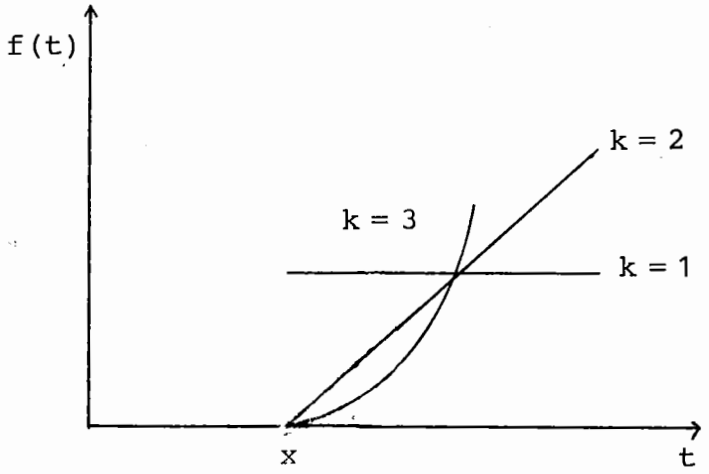
Definition 20.1: Sei eine Unterteilung $x_o < x_1 < \ldots < x_n$ gegeben. Eine Funktion $s$ heißt Spline der Ordnung $k$ (zur Unterteilung $x_o,\ldots,x_n$), falls $s \in C^{k-2}[x_o,x_n]$ und $s$ in jedem Intervall $[x_i,x_{i+1}]$ ein Polynom vom Grade $k-1$ ist.

Beispiele: Bei $k = 4$ spricht man von einem kubischen Spline. Ist außerdem Bedingung (c) erfüllt, $s$ also linear außerhalb von $[x_o,x_n]$, heißt $s$ natürlicher kubischer Spline.

Entsprechend spricht man bei $k = 3$ von einem quadratischem Spline. Für $k = 2$ ergibt sich ein Polygonzug (linearer Spline).

Wir wollen nun spezielle Splines definieren, die sog. B-Splines (B für Basis): Sei für ein beliebiges aber festes $x \in \mathbb{R}^1$

$$f(t) = (t-x)_+^{k-1} = \begin{cases} (t-x)^{k-1} & t > x \\ 0 & t \leq x \end{cases}$$




Es gilt:

$$f^{(k-2)}(t) = \begin{cases} (k-1)!(t-x) & t > x \\ 0 & t < x \end{cases}$$

$$\lim_{t \downarrow x} f^{(k-2)}(t) = 0$$

Also ist $f \in C^{k-2}(\mathbb{R}^1)$.

Definition 20.2: $B_{i,k}(x) = (x_{i+k} - x_i)\,[f_i,\ldots,f_{i+k}]$ mit $f_j = f(x_j)$ heißt B-Spline der Ordnung $k$.

Satz 20.2: $B_{i,k}$ ist Spline der Ordnung $k$ zur Unterteilung $x_0,\ldots,x_n$, welcher außerhalb $[x_i, x_{i+k}]$ verschwindet.

Beweis:

(a) $f_j = (x_j - x)_+^{k-1}$ ist offenbar in jedem Teilintervall $[x_\ell , x_{\ell+1}]$, $\ell = i,\ldots,i+k$ ein Polynom vom Grade $k-1$ in $x$. Dies gilt damit auch für $B_{i,k}$, das ja eine Linearkombination der $f_j$ ist.

(b) Da die $f_j$ alle k-2 mal stetig nach $x$ differenzierbar sind, gilt dies auch für $B_{i,k}$.

(c) Für $x \le x_i \le t$ ist $f(t) = (t-x)_+^{k-1} = (t-x)^{k-1}$ ein Polynom vom Grade $k-1$ in $x$. $[f_i,\ldots,f_{i+k}]$ ist dann die Dividierte Differenz der Ordnung $k$ eines Polynoms vom Grade $k-1$. Diese ist aber Null.

Lemma: Die Dividierte Differenz k-ter Ordnung eines Polynoms vom Grade $< k$ verschwindet.

Beweis: Sei $f \in \mathcal{P}_{k-1}$. Das Newton'sche Interpolationspolynom k-ter Ordnung für die Stützstellen $x_i,\ldots,x_{i+k}$ und die Stützwerte $f_j = f(x_j)$ ist

$$P(t) = [f_i] + [f_i,f_{i+1}](t-x_i) + \ldots + [f_i,\ldots,f_{i+k}](t-x_i)\cdot\ldots\cdot(t-x_{i+k-1}).$$

Eine weitere Lösung des Interpolationsproblems ist $f$ selbst. Also $f = P$ und damit $[f_i,\ldots,f_{i+k}] = 0$.

Damit gilt:

$$B_{i,k}(x) = 0 \quad \text{für} \quad x \le x_i$$

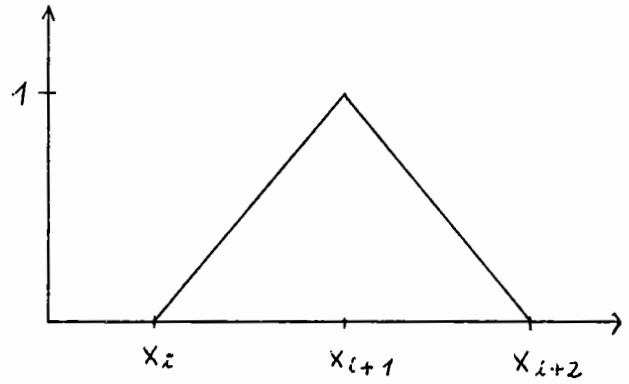
Für $x \ge x_{i+k}$, also $x > x_j$, $j=i,\ldots,i+k$, verschwinden

die $f_j$ und damit auch $[f_i,\ldots,f_{i+k}]$ . Damit gilt auch

$$B_{i,k}(x) = 0 \quad \text{für} \quad x \geq x_{i+k}$$

Beispiele: $k = 2$

$B_{i,2}$ hat die folgende Gestalt:




Wir rechnen dies für das Intervall $[x_i,x_{i+1}]$ nach:

$$B_{i,2}(x) = (x_{i+2} - x_i)\,[f_i,f_{i+1},f_{i+2}]$$

$$\left.\begin{aligned} f_i &= (x_i - x)_+ = 0 \\ f_{i+1} &= (x_{i+1}-x)_+ = (x_{i+1}-x) \\ f_{i+2} &= (x_{i+2}-x)_+ = (x_{i+2}-x) \end{aligned}\right\} \quad x \in [x_i,x_{i+1}]$$

Das Differenzenschema lautet:

$x_i$ 0

$\frac{x_{i+1} - x}{x_{i+1} - x_i}$

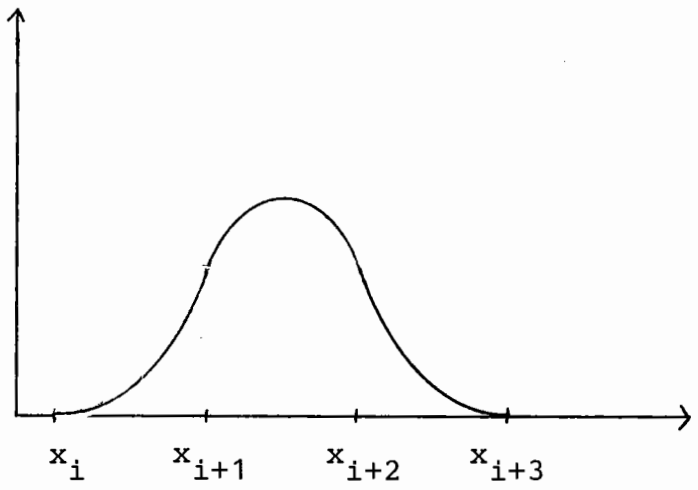
$x_{i+1}$ $(x_{i+1}-x)$ $\qquad\qquad \frac{1 - \frac{x_{i+1} - x}{x_{i+1} - x_i}}{x_{i+2} - x_i}$

$\underbrace{\frac{x_{i+2} - x_{i+1}}{x_{i+2} - x_{i+1}}}_{1}$

$x_{i+2}$ $(x_{i+2}-x)$

$$B_{i,2} = 1 - \frac{x_{i+1} - x}{x_{i+1} - x_i} \quad , \quad x \in [x_i , x_{i+1}]$$

Bei $k = 3$ erwarten wir, daß $B_{i,3}(x)$ die folgende Form hat:




Wir wollen jetzt eine bequemere Methode zur Berechnung der $B_{i,k}$ angeben. Dazu benötigen wir zunächst einmal die Leibniz'sche Formel.

Satz 20.3: Es seien $f_j, g_j, h_j$ mit $f_j = g_j h_j$, $j = i,\ldots,i+k$ gegeben. Dann gilt:

$$[f_i \dots f_{i+k}] = \sum_{r=i}^{i+k} [g_i \dots g_r]\,[h_r \dots h_{i+k}]$$

Bemerkung: Der Name "Leibniz'sche Formel" stammt von der bis auf einen in der Summe fehlenden Faktor $\binom{k}{r-i}$ vorhandenen formalen Ähnlichkeit zur Produktregel der Differentiation.

Beweis: Seien $f$, $g$, $h$ Interpolationspolynome aus $\mathcal{P}_k$ mit $f(x_j) = f_j$, $g(x_j) = g_j$, $h(x_j) = h_j$, $j = i,\dots,i+k$.

Wir schreiben sie in der Newton'schen Form:

$$g(x) = \sum_{r=i}^{i+k} [g_i \dots g_r]\,(x-x_i)\cdot\dots\cdot(x-x_{r-1})$$

Zur Berechnung von $h(x)$ denken wir uns die Stützstellen in der Reihenfolge $x_{i+k},\dots,x_i$ und beachten, daß $[h_{i+k} \dots h_s] = [h_s \dots h_{i+k}]$:

$$h(x) = \sum_{s=i}^{i+k} [h_s \dots h_{i+k}]\,(x-x_{i+k})\cdot\dots\cdot(x-x_{s+1})$$

Dann gilt:

$$(gh)(x) = \sum_{r,s=i}^{i+k} [g_i \dots g_r]\,[h_s \dots h_{i+k}]\,(x-x_i)\cdot\dots\cdot(x-x_{r-1})\cdot (x-x_{s+1})\cdot\dots\cdot(x-x_{i+k})$$

$$= \sum_{r\le s} \dots + \sum_{r>s} \dots$$

Die Summe mit $r > s$ verschwindet offenbar für $x = x_i,\dots,x_{i+k}$. Die Summe mit $r \le s$ ergibt ein Polynom vom Grad $\le k$, das an

den Stellen $x_j$ die Werte $f_j$ annimmt und daher mit $f$ übereinstimmen muß. Der Vergleich der Höchstkoeffizienten ergibt die Leibniz'sche Formel:

$$\sum_{r=s} [g_i \dots g_r]\,[h_s \dots h_{i+k}] = [f_i \dots f_{i+k}] \,.$$

■

Satz 20.4: Für $k \geq 2$ und $i = 0,\dots,n-k$ gilt:

$$B_{i,k}(x) = \frac{x - x_i}{x_{i+k-1} - x_i} B_{i,k-1}(x) + \frac{x_{i+k} - x}{x_{i+k} - x_{i+1}} B_{i+1,k-1}(x)$$

Beweis: Nach Definition ist

$$B_{i,k}(x) = (x_{i+k} - x_i)\,[f_i \dots f_{i+k}] \quad \text{mit} \quad f_i = (x_i - x)_+^{k-1}$$

Sei $g_i = (x_i - x)_+^{k-2}$ und $h_i = x_i - x$. Dann ist

$$f_i = g_i\, h_i$$

und

$$B_{i,k}(x) = (x_{i+k} - x_i) \sum_{r=i}^{i+k} [g_i \dots g_r]\,[h_r \dots h_{i+k}]$$

Da $h(t) = (t - x)$ ein Polynom ersten Grades in $t$ ist, verschwindet $[h_r \dots h_{i+k}]$ für $i + k - r > 1$. Also gilt nach der Leibniz'schen Formel

$$B_{i,k}(x) = (x_{i+k} - x_i)\,\Big\{[g_i \dots g_{i+k}]\,[h_{i+k}] + [g_i \dots g_{i+k-1}][h_{i+k-1}\; h_{i+k}]\Big\}$$

$$= (x_{i+k} - x_i) \left\{ [g_i \cdots g_{i+k}] \; (x_{i+k} - x) + [g_i \cdots g_{i+k-1}] \right\}$$

$$= (x_{i+k} - x_i) \left\{ \frac{[g_{i+1} \cdots g_{i+k}] - [g_i \cdots g_{i+k-1}]}{x_{i+k} - x_i} \; (x_{i+k} - x) + \right.$$

$$\left. + [g_i \cdots g_{i+k-1}] \right\}$$

$$= [g_{i+1} \cdots g_{i+k}] \; (x_{i+k} - x) + [g_i \cdots g_{i+k-1}] \; (x - x_i)$$

$$= \frac{B_{i+1,k-1}(x)}{x_{i+k} - x_{i+1}} \; (x_{i+k} - x) + \frac{B_{i,k-1}(x)}{x_{i+k-1} - x_i} \; (x - x_i) \quad .$$

■

Bemerkungen:

1) Zusammen mit

$$B_{i,1}(x) = \begin{cases} 1 \, , & x_i \leq x \leq x_{i+1} \\ 0 \, , & \text{sonst} \end{cases} \, ,$$

ermöglicht Satz 20.4 eine rekursive Berechnung der $B_{i,k}$.

2) Die rekursive Berechnung der $B_{i,k}$ ist gutartig, da nur Linearkombinationen positiver Zahlen gebildet werden.

Für die Ableitung der B-Splines ergibt sich nun ein einfacher Ausdruck:

Satz 20.5: Für $k \geq 2$ gilt:

$$B'_{i,k}(x) = (k-1) \left\{ \frac{B_{i,k-1}(x)}{x_{i+k-1} - x_i} - \frac{B_{i+1,k-1}}{x_{i+k} - x_{i+1}} \right\}$$

Beweis:

$$B_{i,k}(x) = (x_{i+k} - x_i) \quad [f_i \ldots f_{i+k}]$$

$$f_i = (x_i - x)_+^{k-1} \quad \Rightarrow \quad f_i' = -(k-1)(x_i - x)_+^{k-2}$$

$[f_i \ldots f_{i+k}]$ ist eine Linearkombination in den $f_j$ mit von $x$ unabhängigen Koeffizienten. Daher ist

$$B_{i,k}'(x) = (x_{i+k} - x_i) \quad [f_i' \ldots f_{i+k}']$$

$$= (x_{i+k} - x_i) \quad \frac{[f_{i+1}' \ldots f_{i+k}'] - [f_i' \ldots f_{i+k-1}']}{x_{i+k} - x_i}$$

$$= (k-1) \left( \frac{B_{i,k-1}(x)}{x_{i+k-1} - x_i} - \frac{B_{i+1,k}(x)}{x_{i+k} - x_{i+1}} \right)$$

■

Folgerung: Für eine Linearkombination

$$s(x) = \sum_{i=o}^{n-k} a_i \, B_{i,k}(x) \quad \text{gilt:}$$

$$s'(x) = \sum_{i=o}^{n-k} a_i \, B_{i,k}'(x)$$

$$= (k-1) \sum_{i=o}^{n-k} a_i \left( \frac{B_{i,k-1}(x)}{x_{i+k-1} - x_i} - \frac{B_{i+1,k-1}(x)}{x_{i+k} - x_{i+1}} \right)$$

$$= (k-1) \left\{ \sum_{i=o}^{n-k} a_i \, \frac{B_{i,k-1}(x)}{x_{i+k-1} - x_i} - \sum_{i'=1}^{n-k+1} a_{i'-1} \, \frac{B_{i',k-1}(x)}{x_{i'+k-1} - x_{i'}} \right\}$$

$$= (k-1) \sum_{i=o}^{n-k+1} \frac{a_i - a_{i-1}}{x_{i+k-1} - x_i} \, B_{i,k-1}(x)$$

$$\text{mit} \quad a_{-1} = a_{n-k+1} = 0$$

Sehen wir nun, wie man die B-Splines zur Lösung einer Interpolationsaufgabe benutzt:

Wir haben eine Unterteilung $x_o,\ldots,x_n$ und wollen an den Stellen $t_o,\ldots,t_{n-k}$ mit einem Spline

$$s(t) = \sum_{i=o}^{n-k} a_i B_{i,k}(t)$$

interpolieren. Für die Koeffizienten $a_i$ ergibt sich das Gleichungssystem:

$$s(t_j) = \sum_{i=o}^{n-k} a_i B_{i,k}(t_j) = y_j \quad , \quad j = o,\ldots,n-k \ .$$

Wenn $t_j$ zwischen $x_j$ und $x_{j+k}$ liegt, ist das System eindeutig lösbar:

Satz 20.6: Ist $x_i < t_j < x_{i+k}$ , $i = o,\ldots,n-k$, so ist die Matrix

$$B = \begin{pmatrix} B_{o,k}(t_o) & \cdots & B_{n-k,k}(t_o) \\ \vdots & & \vdots \\ B_{o,k}(t_{n-k}) & \cdots & B_{n-k,k}(t_{n-k}) \end{pmatrix}$$

invertierbar.

Beweis: Wir betrachten zunächst den Fall $k = 2$. Die Matrix ist dann tridiagonal ($n = 5$, Index $k$ unterdrückt):

$$\begin{pmatrix} B_0(t_0) & B_1(t_0) & & & \\ B_0(t_1) & B_1(t_1) & B_2(t_1) & & \\ & B_1(t_2) & B_2(t_3) & B_3(t_2) & \\ & & B_2(t_3) & B_3(t_3) & \end{pmatrix}$$

Verschwindet hier eines der Elemente links oder rechts der Diagonalen, dann zerfällt das System in kleinere. Wären alle diese Elemente $\neq 0$, so müßte inbesondere

$$B_1(t_0) \neq 0, \text{ d.h. } x_1 < t_0 < x_2$$

$$B_0(t_1) \neq 0, \text{ d.h. } x_1 < t_0 < x_2$$

sein. Dann wäre aber $B_2(t_1) = 0$. Für $n \geq 4$ zerfällt also jedes System notwendig in kleinere. Es genügt also, $n = 3$ zu betrachten, und dieser Fall ist trivial.

Sei nun die Behauptung richtig bis zu einer Ordnung $< k > 2$. Eine typische Stelle der Matrix für die Ordnung $k$ sieht dann so aus (Index $k$ unterdrückt):

$$\begin{pmatrix} & & \cdot & & \\ & & \cdot & \cdot & \\ & & \cdot & & \cdot \\ & B_j(t_{j-1}) & & B_{j+1}(t_{j-1}) & \\ \cdots & B_j(t_j) & & B_{j+1}(t_j) & \cdots \\ & \cdot & \cdot & & \\ & \cdot & \cdot & & \\ & \cdot & \cdot & & \cdot \end{pmatrix}$$

Wäre hier $B_{j+1}(t_j) = 0$, d.h. $x_j \leq t_j \leq x_{j+1}$, so wären erst recht alle Elemente rechts und oberhalb dieses Elementes (durch Punkt angedeutet) 0. Das gleiche träfe zu auf die Elemente links unterhalb $B_j(t_j)$, falls $B_j(t_j) = 0$, d.h. $t_j \geq x_{j+k-1}$. In beiden Fällen zerfiele die Matrix in kleinere, und wir könnten uns auf den (trivialen) Fall $n = k$ beschränken. Wir nehmen also an, daß

$$(*) \qquad x_{j+1} < t_j < x_{j+k-1} \quad , \qquad j = 0,\ldots,n-k \quad .$$

Wäre $B\ \alpha = 0$, $\alpha = (\alpha_0,\ldots,\alpha_{n-k})^T \neq 0$, so wäre

$$s = \sum_{j=0}^{n-k} \alpha_j B_{j,k}$$

eine Funktion $\in C^{n-k}$ mit $s(t_j) = 0$, $j = 0,\ldots,n-k$, $s(x_0) = 0$, $s(x_n) = 0$. Nach dem Satz von Rolle hätte dann $s'$ $n-k+2$ Nullstellen

$$T_{-1} \in [x_0, t_0] \ , \quad T_j \in [t_j, t_{j+1}] \ , \quad j=0,\ldots,n-k-1 \ , \quad T_{n-k} \in [t_{n-k}, x_n].$$

Wegen (*) muß

$$x_0 < T_{-1} < x_{k-1} \ , \quad x_{j+1} < T_j < x_{j+(k-1)+1} \ , \quad j=0,\ldots,n-k+1 \ , \quad x_{n-k} < T_{n-k} < x_n$$

sein. Damit haben wir in den $T_j$ $n-(k-1)+1$ Nullstellen von $s'$ gefunden, welche auch noch die Voraussetzung des Satzes für $k-1$ erfüllen. Wegen Satz 20.5 ist $s'$ eine Linearkombination der $B_{j,k-1}$, $j = 0,\ldots,n-(k-1)$. Nach Induktionsannahme folgt also $s' = 0$ und damit $s = 0$. Also ist $B$ nicht singulär.

■

## § 21 Approximation in normierten Räumen

Eine vorgegebene Funktion $f \in C[a,b]$ soll durch eine Funktion $u^*$ aus einem Unterraum $U \subset C[a,b]$ approximiert werden. Wir untersuchen in diesem Paragraphen, ob es eine optimale Wahl für $u^*$ gibt.

Wir führen zunächst auf $C[a,b]$ eine Norm $\|\cdot\|$ ein.

Für $1 \leq p < \infty$ ist beispielsweise

$$\|f\|_p = \left( \int_a^b |f(x)|^p \, dx \right)^{1/p}$$

eine Norm. Den wichtigsten Fall erhält man für $p \to \infty$:

$$\|f\|_\infty = \lim_{p\to\infty} \|f\|_p = \max_{[a,b]} |f(x)|$$

Wir legen uns im folgenden jedoch auf keine spezielle Norm fest. Sei $U$ der von den linear unabhängigen Funktionen $u_o, \ldots, u_n \in C[a,b]$ aufgespannte Unterraum:

$$U = \left\{ \sum_{k=o}^{n} a_k u_k \;,\; a_k \in \mathbb{R}^1 \right\}$$

und sei

$$\varepsilon(f) := \inf_{u\in U} \|f - u\| \quad .$$

Dann heißt unsere Aufgabe:

Finde $u^* \in U$ mit $\|f - u^*\| = \varepsilon(f)$ .

Satz 21.1: Das Approximationsproblem ist immer lösbar.

Beweis: Für $a := (a_o,\ldots,a_n)^T \in \mathbb{R}^{n+1}$ sei

$$F(a) = \|f - \sum_{k=o}^{n} a_k u_k\|$$

F ist offenbar stetig. Wir zeigen, daß $F(a) > \varepsilon(f)$ außerhalb einer kompakten Menge M gilt. Daraus folgt dann, daß die Funktion F ihr Minimum $\varepsilon(f)$ in M annimmt.

Sei $M := \left\{a \in \mathbb{R}^{n+1} : F(a) \leq F(0)\right\}$

$$\Rightarrow \quad F(a) > F(0) \geq \varepsilon(f) \quad \text{für} \quad a \notin M$$

$\mathbb{R}^{n+1} \setminus M$ ist offen, da es Urbild der offenen Menge $\left\{x \in \mathbb{R} : F(x) > F(0)\right\}$ ist. Daher ist M abgeschlossen.

Die Dreiecksungleichung ergibt:

$$F(a) \geq \|\sum_{k=o}^{n} a_k u_k\| - \|f\|$$

$\|\sum_{k=o}^{n} a_k u_k\|$ definiert eine Norm in $\mathbb{R}^{n+1}$. Da alle Normen in $\mathbb{R}^{n+1}$ äquivalent sind, gilt:

$$\exists \; c \in \mathbb{R}^1 : \|a\| \leq c \cdot \|\sum_{k=o}^{n} a_k u_k\| \quad \forall \; a \in \mathbb{R}^{n+1}$$

Damit zeigt man leicht, daß M auch beschränkt und somit kompakt ist.

$F(a) \geq c\|a\| - \|f\| \quad \Rightarrow a \in M: \; c\|a\| - \|f\| \leq F(a) \leq F(0)$

$\Rightarrow \|a\| \leq \frac{1}{c}\left(F(0) + \|f\|\right)$

Also existiert ein $a^* \in \mathbb{R}^{n+1}$ mit $F(a^*) = \varepsilon(f)$

$u^* = \sum_{k=o}^{n} a_k^* u_k$ ist dann Lösung des Approximationsproblems. ∎

Definition: Eine Norm heißt strikt, wenn gilt:

$$\|f + g\| = \|f\| + \|g\| \Rightarrow f, g \text{ linear abhängig}$$

Beispiele: $\| \|_2$ ist strikt, $\| \|_\infty$ hingegen nicht. Letzteres sieht man an dem Beispiel $f = 1$, $g = x$ in $C[0,1]$.

Satz 21.2: Das Approximationsproblem für strikte Normen ist eindeutig lösbar.

Beweis: Seien $u_1^*$ und $u_2^*$ Lösungen und $u^* := \frac{1}{2}(u_1^* + u_2^*)$. Dann gilt:

$$\|f - u^*\| \leq \frac{1}{2}\|f - u_1^*\| + \frac{1}{2}\|f - u_2^*\| = \varepsilon(f)$$

$u^*$ ist also ebenfalls eine Lösung und in der Ungleichung gilt Gleichheit. Dann folgt:

$$\exists\ \alpha, \beta \in \mathbb{R}^1 : \frac{\alpha}{2}(f - u_1^*) = \frac{\beta}{2}(f - u_2^*)$$

$$\Rightarrow \quad (\alpha - \beta)f = \alpha u_1^* - \beta u_2^*$$

$$\alpha = \beta \Rightarrow u_1^* = u_2^*$$

$$\alpha \neq \beta \Rightarrow f \in U \Rightarrow \varepsilon(f) = 0$$

$$\Rightarrow \|f - u_1^*\| = \|f - u_2^*\| = 0$$

$$\Rightarrow u_1^* = u_2^* = f$$

∎

## § 22 Tschebyscheff-Approximation

Wir betrachten im folgenden die Approximationsaufgabe mit der Norm

$$\|f\| = \max_{[a,b]} |f(x)|$$

Definition 22.1: Seien $u_0,\ldots,u_n \in C[a,b]$ linear unabhängig. $U = \langle u_0,\ldots,u_n \rangle$ heißt unisolvent, wenn für jede Unterteilung $a \leq x_0 < x_1 < \ldots < x_n \leq b$ und für jede Wahl von $y_j$, $j = 0,\ldots,n$ genau ein $u^* \in U$ existiert mit $u^*(x_j) = y_j$, $j = 0,\ldots,n$.

Bemerkung: U unisolvent $\Leftrightarrow \det(u_k(x_j)) \neq 0$ für jede Unterteilung.

$\Leftrightarrow$ Jede Funktion $u \in U$ mit mehr als n Nullstellen verschwindet identisch.

Beispiele:

1) $U = \mathcal{P}_n$.

2) $u_k(x) = e^{\lambda k x}$, $k = 0,\ldots,n$, $\lambda \neq 0 \in \mathbb{R}^1$.

$\det(u_k(x_j)) = (e^{\lambda x_j})^k = (y_j)^k$ ist gerade die Vandermonde-Determinante von $(y_0,\ldots,y_n)$.

3) $U = \mathcal{T}_m$, $n = 2m+1$ für $0 \leq a < b < 2\pi$.

4) $\langle 1, x^2 \rangle$ ist in $[-1,+1]$ nicht unisolvent.

5) $\langle B_{0,k},\ldots,B_{n-k,k} \rangle$ in $[x_0,x_n]$ nicht unisolvent.

Satz 22.1: Sei U unisolvent, $u \in U$ und $a \leq x_0 < x_1 < \ldots < x_{n+1} \leq b$. $(f-u)(x_j)$ habe alternierende Vorzeichen, d.h.

$$\exists\ \sigma\ ,\ |\sigma| = 1 : \operatorname{sgn}[(f-u)(x_j)] = (-1)^j\sigma\ ,\quad j=0,\ldots,n+1 \quad .$$

Dann gilt:

$$\min_{j=0}^{n+1} |(f-u)(x_j)| \leq \varepsilon(f) \leq \|f-u\| \quad .$$

Beweis:

Annahme: $\min_{j=0}^{n+1} |(f-u)(x_j)| > \varepsilon(f)$ .

$$\Rightarrow \quad \exists\ v \in U : \min_{j=0}^{n+1} |(f-u)(x_j)| > \|f-v\| =$$

$$= \max_{[a,b]} |(f-v)(x)| \geq \max_{j=0}^{n+1} (f-v)(x_j) \quad .$$

$$\Rightarrow \quad |(f-u)(x_j)| > |(f-v)(x_j)| \ ,\quad j = 0,\ldots,n+1 \quad .$$

Aus

$$|(f-u)(x_j)| = (\operatorname{sgn}(f-u)(x_j))(f-u)(x_j) = \sigma(-1)^j(f-u)(x_j)$$

folgt:

$$\sigma(-1)^j(f-u)(x_j) > |(f-v)(x_j)| \geq \sigma(-1)^j(f-v)(x_j) \ ,\quad j=0,\ldots,n+1$$

$$\Rightarrow \quad \sigma(-1)^j(v-u)(x_j) > 0 \ ,\quad j = 0,\ldots,n+1 \quad .$$

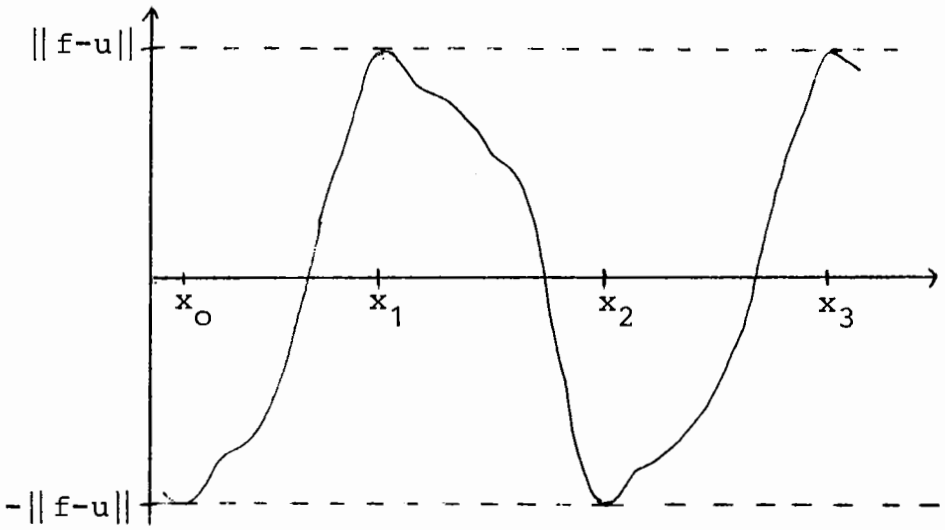
$v - u$ hat also mindestens $n+1$ Nullstellen. Da $U$ unisolvent ist, folgt $v = u$ und damit ein Widerspruch. ■

Definition 22.2: $x_0 < x_1 < \ldots < x_{n+1}$ heißt Alternante (der Länge n+2)

zu $f-u$, falls $f-u$ an den Stellen $x_j$ mit alternierendem Vorzeichen seinen Maximalwert annimmt, d.h.

$$\exists\ \sigma\ ,\ |\sigma| = 1 : (f-u)(x_j) = \sigma(-1)^j \|f-u\|$$

Beispiel: $n = 2$




Satz 22.2: Sei $U$ unisolvent und $u \in U$. $f-u$ besitze eine Alternante der Länge $n+2$. Dann ist $u$ Lösung des Approximationsproblems, d.h. $\|f-u\| = \varepsilon(f)$.

Beweis: Nach Satz 22.1 gilt:

$$\|f-u\| = \min_{j=o}^{n+1} |(f-u)(x_j)| \leq \varepsilon(f) \leq \|f-u\|$$

Bemerkung: Es gilt auch die umgekehrte Behauptung, d.h. zu jeder Lösung gibt es eine Alternante. Der Beweis ist schwieriger (vgl. G. Meinardus, Approximation von Funktionen und ihre numerische Behandlung).

Satz 22.3: Sei $U$ unisolvent und $a \leq x_o < \ldots < x_{n+1} \leq b$. Dann gibt es genau ein $u \in U$ und $d \in \mathbb{R}^1$ mit

$$u(x_j) = f(x_j) - d\,(-1)^j \quad , \qquad j = 0,\ldots,n+1$$

und es gilt:

$$|d| \leq \varepsilon(f) \leq \|f - u\|$$

Beweis:

$$(f-u)(x_j) = d(-1)^j \quad , \qquad j = 0,\ldots,n+1$$

$$\Rightarrow \quad |d| = \min_{j=0}^{n+1} |(f-u)(x_j)| \leq \varepsilon(f) \quad \text{nach Satz 22.1.}$$

Für $d$ und die Koeffizienten $a_k$ von $u = \sum_{k=0}^{n} a_k u_k$ erhält man die Gleichungen

$$\sum_{k=0}^{n} a_k u_k(x_j) + d(-1)^j = f(x_j) \;, \; j = 0,\ldots,n+1$$

Das zugehörige homogene System lautet

$$u(x_j) = \sum_{k=0}^{n} a_k u_k(x_j) = -d(-1)^j \;, \; j = 0,\ldots,n+1$$

$u$ hat also mindestens $n+1$ Nullstellen.

Da $U$ unisolvent ist, müssen die $a_k$ und damit auch $d$ verschwinden, das homogene System ist also nur trivial lösbar.

Satz 22.3 bildet die Grundlage des Remes-Algorithmus zur Lösung der Approximationsaufgabe:

Sei $f \in C[a,b]$ und $U$ unisolvent. Gesucht wird ein $u^* \in U$ mit

$$\|f - u^*\| \leq \|f - u\|, \quad \forall\, u \in U \quad .$$

Wir konstruieren eine Folge von Unterteilungen

$$X^{(k)} = (x_o^{(k)}, \ldots, x_{n+1}^{(k)}) \quad , \quad a \leq x_o^{(k)} < \ldots < x_{n+1}^{(k)} \leq b$$

und finden die nach Satz 22.3 eindeutig bestimmten $u^{(k)}$, $d^{(k)}$ mit

$$u^{(k)}(x_j) = f(x_j^{(k)}) - d^{(k)}(-1)^j \quad , \quad j = 0,\ldots,n+1 \ ,$$

$$|d^{(k)}| \leq \varepsilon(f) \leq \|f - u^{(k)}\| =: \varepsilon^{(k)} \quad .$$

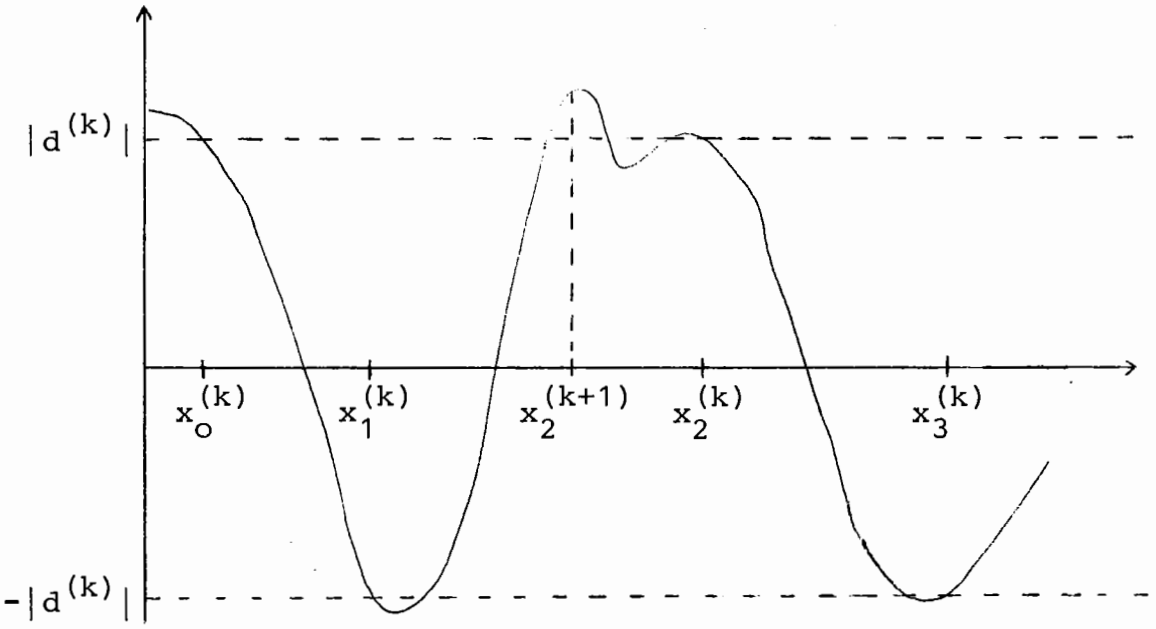
Die Unterteilung $X^{(0)}$ kann beliebig gewählt werden, es ist jedoch ein wichtiges praktisches Problem, ein geeignetes $X^{(0)}$ zu finden.

Sei $X^{(k)}$ berechnet. Falls $|d^{(k)}| = \varepsilon^{(k)}$, so ist $\|f-u^{(k)}\| = \varepsilon(f)$ und das Verfahren kann abgebrochen werden. Für $|d^{(k)}| < \varepsilon^{(k)}$ wird $X^{(k+1)}$ so gewählt, daß die folgenden Bedingungen erfüllt sind:

(a) $(f-u^{(k)})(x_j^{(k+1)})$ hat alternierendes Vorzeichen

(b) $|(f-u^{(k)})(x_j^{(k+1)})| \geq |d^{(k)}| \qquad j = 0,\ldots,n+1$

(c) $|(f-u^{(k)})(x_j^{(k+1)})| > |d^{(k)}|$ für mindestens ein $j$




In dem skizzierten Beispiel sind die Bedingungen (a) bis (c) erfüllt, wenn $x_2^{(k+1)}$ wie eingezeichnet gewählt wird und die übrigen Unterteilungspunkte nicht verändert werden.

Satz 22.4: Sei U unisolvent und seien die Bedingungen (a), (b), (c) erfüllt. Dann gilt

$$|d^{(k+1)}| > |d^{(k)}|$$

Beweis: Wir entwickeln $u^{(k+1)}$ nach der Basis $\{u_o,\ldots,u_n\}$ von U :

$$u^{(k+1)} = \sum_{i=o}^{n} a_i^{(k+1)} u_i$$

Für die Koeffizienten $a_i^{(k+1)}$ und für $d^{(k+1)}$ gelten die Gleichungen

$$\sum_{i=o}^{n} a_i^{(k+1)} u_i(x_j^{(k+1)}) + d^{(k+1)} (-1)^j = f(x_j^{(k+1)}), \quad j=o,\ldots,n+1$$

Nach der Cramer'schen Regel gilt für $d^{(k+1)}$:

$$d^{(k+1)} = \frac{\begin{vmatrix} u_o(x_o^{(k+1)}) & \ldots & u_n(x_o^{(k+1)}) & f(x_o^{(k+1)}) - u^{(k)}(x_o^{(k+1)}) \\ \vdots & & \vdots & \vdots \\ u_o(x_{n+1}^{(k+1)}) & \ldots & u_n(x_{n+1}^{(k+1)}) & f(x_{n+1}^{(k+1)}) - u^{(k)}(x_{n+1}^{(k+1)}) \end{vmatrix}}{\begin{vmatrix} u_o(x_o^{(k+1)}) & \ldots & u_n(x_o^{(k+1)}) & 1 \\ \vdots & & \vdots & \vdots \\ u_o(x_j^{(k+1)}) & \ldots & u_n(x_j^{(k+1)}) & (-1)^j \\ \vdots & & \vdots & \vdots \\ u_o(x_{n+1}^{(k+1)}) & \ldots & u_n(x_{n+1}^{(k+1)}) & (-1)^{n+1} \end{vmatrix}}$$

Dabei haben wir im Zähler von der letzten Spalte $(u^{(k)}(x_o^{(k+1)}), \ldots, u^{(k)}(x_{n+1}^{(k+1)}))^T$ subtrahiert. Dies ändert den Wert der Determinante jedoch nicht, da der subtrahierte Vektor eine Linearkombination der ersten $n+1$ Spalten ist.

Wir entwickeln beide Determinanten nach der letzten Spalte:

$$d^{(k+1)} = \frac{\sum_{j=o}^{n+1} (f-u^k)(x_j^{(k+1)}) (-1)^{n+1+j} D_j}{\sum_{j=o}^{n+1} (-1)^j (-1)^{n+1+j} D_j}$$

$D_j$ entsteht aus beiden Determinanten durch Streichen der j-ten Zeile und der letzten Spalte.

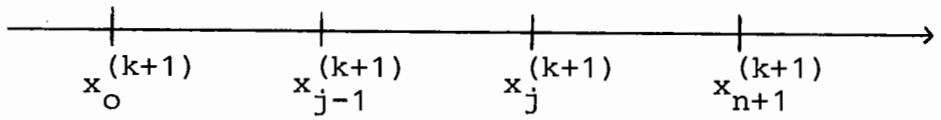
Sei $\mu_j = D_j / \left( \sum_{i=o}^{n+1} D_i \right)$. Dann gilt:

$$d^{(k+1)} = \sum_{j=o}^{n+1} (-1)^j \, (f - u^{(k)})(x_j^{(k+1)}) \, \mu_j \quad .$$

Wir zeigen nun, daß die $\mu_j$ alle streng positiv sind:

Da U unisolvent, gilt $D_j \neq O$ , $j = O,\ldots,n+1$ .

Betrachten wir die Unterteilung $X^{(k+1)}$:




$D_j$ hängt stetig von den $x_i^{(k+1)}$ , $i \neq j$ ab. Denken wir uns nun $x_{j-1}^{(k+1)}$ in Richtung von $x_j^{(k+1)}$ verschoben, so nähert sich $D_j$ dem Wert von $D_{j-1}$ stetig an, darf aber für keinen Wert von $x_{j-1}^{(k+1)}$ verschwinden. Daher müssen $D_j$ und $D_{j-1}$ gleiches Vorzeichen haben, woraus folgt, daß alle $D_i$ dasselbe Vorzeichen haben, die $\mu_i$ also streng positiv sind.

Aus der Voraussetzung (a) folgt, daß auch das Vorzeichen von $(-1)^j (f - u^k)(x_j^{(k+1)})$ von $j$ unabhängig ist. Damit gilt:

$$|d^{(k+1)}| = \sum_{j=o}^{n+1} |(f - u^{(k)})(x_j^{(k+1)})| \mu_j \overset{(b),(c)}{>} |d^{(k)}| \sum_{j=o}^{n+1} \mu_j = |d^{(k)}|$$

da $\sum_{j=0}^{n+1} \mu_j = 1$ .

Wir betrachten nun den Fall der Approximation mit Polynomen n-ten Grades und untersuchen, wie gut bereits das Interpolationspolynom approximiert.

Satz 22.5: Sei $U = \mathcal{P}_n$ und $u^* \in U$ mit $\|f - u^*\| = \varepsilon(f)$. Für $u_n \in U$ gelte $u_n(x_j) = f(x_j)$ , $j = 0,\ldots,n$ , wobei die $x_j$ aus $[a,b]$ und paarweise verschieden seien.

Dann gilt:

$$\|f - u_n\| \leq V_n \cdot \varepsilon(f)$$

mit $V_n = 1 + \| \sum_{j=0}^{n} |\omega_j(x)| \|$ , $\omega_j(x) = \prod_{\substack{i=0 \\ i \neq j}}^{n} \frac{x - x_i}{x_j - x_i}$

Beweis: Sei $L_n$ der Operator, der eine Funktion in ihr Interpolationspolynom zur Unterteilung $x_0,\ldots,x_n$ überführt. Wir benutzen die Lagrange'sche Form des Interpolationspolynoms und erhalten $u_n = L_n f = \sum_{j=0}^{n} f(x_j)\, \omega_j$. Es folgt:

$$\begin{aligned} f - u_n &= f - u^* + u^* - u_n \\ &= f - u^* + L_n u^* - L_n f \qquad (\text{da } u^* = L_n u^*) \\ &= f - u^* + L_n(u^* - f) \qquad (\text{da } L_n \text{ linear ist}) \end{aligned}$$

$$\Rightarrow |(f-u_n)(x)| \leq |(f-u^*)(x)| + \sum_{j=0}^{n} |(u^*-f)(x_j)| \; |\omega_j(x)|$$

$$\leq \|f - u^*\| \left(1 + \sum_{j=0}^{n} |\omega_j(x)|\right)$$

$$\Rightarrow \quad \|f - u_n\| \leq (1 + \| \sum_{j=0}^{n} |\omega_j(x)| \|) \, \varepsilon(f)$$

Beispiel: $[a,b] = [-1,+1]$, $x_j$ seien die Nullstellen des Tschebyscheff - Polynoms $T_{n+1}(x) = \cos(n+1)t$ , $x = \cos t$ :

$$x_j = \cos\left(\frac{j+\frac{1}{2}}{n+1}\pi\right) \quad , \quad j = 0,\ldots,n \quad .$$

Durch numerisches Rechnen erhält man die folgenden Werte für $V_n$ :

| n | $V_n$ |
|---|---|
| 4 | 3.0 |
| 10 | 3.5 |
| 100 | 4.9 |

Im für die Praxis interessanten Bereich $n \leq 10$ vergrößert man den Fehler nur um einen Faktor kleiner gleich 3.5 , wenn man mit dem Interpolationspolynom an Tschebyscheffstellen approximiert.

Als Startnäherung $X^{(0)}$ empfehlen sich daher die Maxima (Minima) von $f - u_n$ , wobei $u_n(x_j) = f(x_j)$, $j = 0,\ldots,n$ mit den Nullstellen $x_j$ von $T_{n+1}$ .

Mit dem "Alternantensatz" 22.2 können wir die Minimalitätseigenschaften der Tschebyscheff-Polynome, die wir im Paragraphen über den Interpolationsfehler benutzt haben, beweisen:

Sei $[a,b] = [-1,+1]$. Wir suchen ein Polynom $u \in \mathcal{P}_{n+1}$ mit Höchstkoeffizient 1 und $\|u\|_\infty$ minimal.

Es gilt:

$u = x^{n+1} - p$ mit $p \in \mathcal{P}_n$ und $\|x^{n+1} - p\|$ minimal.

$p$ ist also die bestmögliche Approximation von $x^{n+1}$ durch ein Polynom n-ten Grades.

<u>Behauptung:</u> $x^{n+1} - p = 2^{-n} T_{n+1}(x)$

Da $2^{-n} T_{n+1}$ den Höchstkoeffizient 1 hat, ist $p \in \mathcal{P}_n$.

Noch zu zeigen: Es gibt eine Alternante der Länge $n + 2$.

Sei $x_j = \cos\left(\frac{j\pi}{n+1}\right)$, $j = 0,\ldots,n+1$. Dann ist

$$T_{n+1}(x_j) = \cos\left((n+1)\frac{j\pi}{n+1}\right) = \cos(j\pi) = (-1)^j,$$

$$(x^{n+1} - p)(x_j) = (-1)^j 2^{-n} = (-1)^j \|x^{n+1} - p\| .$$

Also bilden die $x_j$ tatsächlich eine Alternante.

## § 23 Approximation nach Gauß

Sei $H$ ein unitärer Raum. Wir benutzen die durch das innere Produkt gegebene Norm:

$$\|f\| = (f, f)^{1/2}$$

Seien $u_o, \ldots, u_n \in H$ linear unabhängig und sei $U = \langle u_o, \ldots, u_n \rangle$. Die Approximationsaufgabe lautet dann:

Für gegebenes $f \in H$ wird $u^* \in U$ gesucht mit

$$\|f - u^*\| = \inf_{u \in U} \|f - u\|$$

Satz 23.1: $u^*$ existiert und ist eindeutig bestimmt.

Beweis: Die Existenz von $u^*$ folgt aus Satz 23.1. Die durch das innere Produkt induzierte Norm ist strikt, da für das innere Produkt die Cauchy-Schwartz'sche Ungleichung gilt. Aus Satz 23.2 folgt dann die Eindeutigkeit.

Satz 23.2: Es gilt $u^* = \sum_{i=o}^{n} a_i^* u_i$, wobei die Koeffizienten $a_i^*$ durch die Normalgleichungen

$$(*) \qquad \sum_{i=o}^{n} a_i^* (u_i, u_k) = (f, u_k) \qquad k = o, \ldots, n$$

bestimmt sind.

Beweis: Wir formen die Normalgleichungen etwas um:

$$(u_k , \sum_{i=o}^{n} a_i^* u_i - f) = 0 \qquad k = 0,\ldots,n$$

$\sum_{i=o}^{n} a_i^* u_i - f$ ist also orthogonal zu U. Für ein beliebiges $u = \sum_{i=o}^{n} a_i u_i$ gilt:

$$\|f-u\|^2 = \|f - \sum_{i=o}^{n} a_i u_i\|^2 = \|f - \underbrace{\sum_{i=o}^{n} a_i^* u_i}_{\in U^\perp} + \underbrace{\sum_{i=o}^{n} (a_i^* - a_i) u_i}_{\in U}\|^2$$

$$= \|f - \sum_{i=o}^{n} a_i^* u_i\|^2 + \|\sum_{i=o}^{n} (a_i^* - a_i) u_i\|^2$$

$$\geq \|f - \sum_{i=o}^{n} a_i^* u_i\|^2 = \|f - u^*\|^2$$

Beispiele:

1) $H = \mathbb{C}^p$

Die Basisvektoren $u_o,\ldots,u_n$ von U bilden die Matrix $A = (u_o,\ldots,u_n)$. Für $f \in \mathbb{C}^p$ führt die Aufgabe $\|f - Ax\|$ zu minimieren auf ein überbestimmtes lineares Gleichungssystem, siehe § 8 .

2) $H = C[-1,1]$, $U = \langle 1, t, t^2 \rangle$ , $f(t) = e^t$

$$\|f\| = \left( \int_{-1}^{1} |f(t)|^2 dt \right)^{1/2}$$

$$(u_i,u_k) = \int_{-1}^{1} u_i u_k dt = \int_{-1}^{1} t^{i+k} dt = \frac{1 + (-1)^{i+k}}{i+k+1}$$

$$(u_o , f) = \int_{-1}^{1} 1 \cdot e^t \, dt = e - 1/e$$

$$(u_1 , f) = \int_{-1}^{1} t\, e^t \, dt = 2/e$$

$$(u_2 , f) = \int_{-1}^{1} t^2 e^t \, dt = e - 5/e$$

Die Normalgleichungen lauten damit

$$\begin{pmatrix} 2 & 0 & \frac{2}{3} \\ 0 & \frac{2}{3} & 0 \\ \frac{2}{3} & 0 & \frac{2}{5} \end{pmatrix} \begin{pmatrix} a_o^* \\ a_1^* \\ a_2^* \end{pmatrix} = \begin{pmatrix} e - 1/e \\ 2/e \\ e - 5/e \end{pmatrix}$$

und führen auf $a_o^* = 0.99629$

$a_1^* = 1.10364$

$a_2^* = 0.53672$

Es gilt: $\max_{t\in[-1,1]} |(e^t - a_o^* - a_1^* t - a_2^* t^2| = 0.082$

3) Die Normalgleichungen vereinfachen sich bedeutend, wenn $u_o,\ldots,u_n$ ein Orthonormalsystem bilden:

$$(u_i , u_k) = \begin{cases} 1 & i = k \\ 0 & \text{sonst} \end{cases}$$

Dann gilt einfach

$$a_k^* = (f , u_k)$$

und

$$u^* = \sum_{i=0}^{n} (f, u_i)\, u_i$$

Die $a_k^*$ heißen in diesem Fall verallgemeinerte Fourier-Koeffizienten von $f$ bezüglich $u_0, \ldots, u_n$.

4) $H = C[0, 2\pi]$

$$(f,g) = \int_0^{2\pi} f\, \bar{g}\, dx$$

$$u_k = \frac{1}{\sqrt{2\pi}}\, e^{ikx} \qquad k = 0, \pm 1, \ldots, \pm m, \quad n = 2m$$

$$\int_0^{2\pi} u_k\, u_\ell\, dx = \frac{1}{2\pi} \int_0^{2\pi} e^{i(k-\ell)x}\, dx = \begin{cases} 1 & k = \ell \\ 0 & \text{sonst} \end{cases}$$

Hier sind die $a_k^* = (f, u_k)$ die normalen Fourier-Koeffizienten.

$$u^* = \sum_{k=0}^{n} (f, u_k) u_k = \frac{1}{2\pi} \sum_{k=0}^{n} \left( \int_0^{2\pi} f\, e^{-ikt}\, dt \right) e^{ikx}$$

heißt endliche Fourier-Reihe von $f$. Für $n \to \infty$ konvergiert $u^*$ in vielen Fällen gegen $f$.

5) $\left\{u_0, \ldots, u_{2m}\right\} = \left\{ \frac{1}{\sqrt{2\pi}}, \frac{1}{\sqrt{\pi}} \cos x, \frac{1}{\sqrt{\pi}} \sin x, \ldots, \frac{1}{\sqrt{\pi}} \cos mx, \frac{1}{\sqrt{\pi}} \sin mx \right\}$

$$u^*(x) = \frac{a_0}{2} + \sum_{k=1}^{m} (a_k \cos kx + b_k \sin kx)$$

heißt ebenfalls Fourier-Reihe. Bei reellem $f$ gilt

$$\left.\begin{matrix} a_k \\ \\ b_k \end{matrix}\right\} = \frac{1}{\pi} \int_0^{2\pi} \left\{\begin{matrix} \cos k\,x \\ \\ \sin k\,x \end{matrix}\right\} f(x)\,dx$$

Satz 23.3: Seien $u_0,\ldots,u_n$ linear unabhängig. Dann gibt es ein Orthonormalsystem $v_0,\ldots,v_n$ mit

$$< u_0,\ldots,u_m > = < v_0,\ldots,v_m > , \quad m = 0,\ldots,n \quad .$$

Beweis: Wie Satz 7.1 durch das E. Schmidt'sche Orthonormalisierungsverfahren.

Sei nun $H = C[a,b]$ und $(f,g) = \int_a^b w\,f\,g\,dx$ mit einer Funktion w, die "Gewichtsfunktion", welche stetig und in $(a,b)$ positiv ist. Ist $u_0 = 1$, $u_1 = x,\ldots,u_n = x^n$, so nennt man die $v_m$ "orthogonale Polynome" (zu $(a,b)$ und w).

Satz 23.4: $v_k$ hat in $(a,b)$ genau $k$ einfache Nullstellen.

Beweis: $x_1,\ldots,x_m$ seien die Zeichenwechsel von $v_k$ in $(a,b)$. Wir zeigen, daß $m = k$ gilt:

Sei $q = \sum_{i=1}^{m} (x - x_i) \in \mathcal{P}_m$

$$(q,v_k) = \int_a^b w\, q\, v_k\, dx = 0 \quad \text{falls} \quad m < k$$

$q \cdot v_k$ hat konstantes Vorzeichen

$\Rightarrow$ $q \cdot v_k \equiv 0$ in $(a,b)$ Widerspruch! ∎

Bemerkung: Die Nullstellen von $v_k$ trennen die Nullstellen von $v_k + 1$.

Satz 23.5: $v_{k+1} = (\alpha_k x - \beta_k) v_k - \gamma_k v_{k-1}$

Beweis:

Ansatz: $\tilde{v}_{k+1} = (x - \beta_k) v_k - \gamma_k v_{k-1}$

Die Konstanten werden so bestimmt, daß $v_{k+1}$ orthogonal zu $v_0,\ldots,v_k$ wird.

Für $\ell \leq k - 2$ gilt:

$$(\tilde{v}_{k+1}, v_\ell) = (xv_k, v_\ell) - \beta_k (v_k, v_\ell) - \gamma_k (v_{k-1}, v_\ell) = 0$$

$$\Leftrightarrow \quad (x\, v_k\, , v_\ell) = 0$$

$$\Leftrightarrow \quad (v_k, x\, v_\ell) = 0$$

Dies ist automatisch erfüllt, da $x\, v_\ell \in \mathcal{P}_{k-1}$.

∎

Für $\ell = k-1$ und $\ell = k$ erhält man die Gleichungen

$$(x\, v_k, v_{k-1}) - \gamma_k = 0$$

$$(x\, v_k, v_k) - \beta_k = 0$$

Es gibt also $\beta_k$, $\gamma_k$, so daß $\tilde{v}_{k+1} \perp \langle v_o, \ldots, v_k \rangle$.

$v_{k+1}$ entsteht aus $\tilde{v}_{k+1}$ durch Normieren.

■

Für die Werte der Koeffizienten $\alpha_k$, $\beta_k$, $\gamma_k$ gibt es Tabellen.

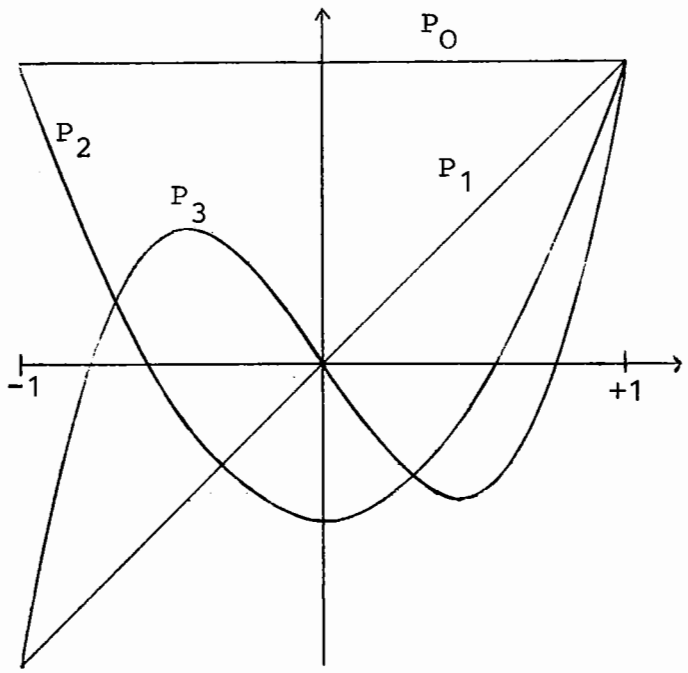
Beispiele:

$[a,b] = [-1,+1]$. Für $w = 1$ erhält man $v_k = c_k P_k$ mit den Legendre-Polynomen $P_k$:

$$P_o = 1 \qquad P_3 = \frac{1}{2}(5x^3 - 3x)$$

$$P_1 = x \qquad P_4 = \frac{1}{8}(35x^4 - 30x^2 + 3)$$

$$P_2 = \frac{1}{2}(3x^2 - 1)$$



Die Skizze zeigt, daß die Nullstellen von $P_{k+1}$ durch diejenigen von $P_k$ getrennt werden.

Bei einer anderen Wahl von [a,b] und w erhält man andere Orthogonalsysteme:

| [a,b] | w | Bezeichnung |
|---|---|---|
| [-1,+1] | 1 | $P_k$ Legendre-Pol. |
| [-1,+1] | $(1-x^2)^{-1/2}$ | $T_k$ Tschebyscheff-Pol. 1. Art |
| [-1,+1] | $(1-x^2)^{1/2}$ | $U_k$ Tschebyscheff-Pol. 2. Art |
| [-1,+1] | $(1-x)^{\alpha}(1+x)^{\beta}$ | $P_k^{(\alpha,\beta)}$ Jacobi-Polynome |
| $(-\infty,+\infty)$ | $e^{-x^2/2}$ | $H_k$ Hermite'sche Pol. |
| $(0,\infty)$ | $e^{-x}$ | $L_k$ Laguerre'sche Pol. |

# NUMERISCHE INTEGRATION UND DIFFERENTIATION

## § 24 Die Formeln von Newton-Cotes

Sei $f \in C[a,b]$. Wir wollen das Integral $I = \int_a^b f(x)dx$ numerisch berechnen und dabei nur Auswertungen von $f$ benutzen. Solche linearen Integrationsformeln haben die allgemeine Form:

$$I \simeq \sum_{k=o}^{n} A_k f(x_k)$$

mit $x_k \in [a,b]$ , $A_k \in \mathbb{R}^1$ . Die $x_k$ heißen Stützstellen, die von $f$ unabhängigen $A_k$ heißen Gewichte.

Wir betrachten nun die geschlossenen Newton-Cotes Formeln, d.h. die Randpunkte des Intervalls sind Stützstellen:

$$x_k = a + k \cdot h \quad , \quad h = \frac{b-a}{n} \qquad\qquad k = o,\ldots,n$$

$$I_n = \sum_{k=o}^{n} A_k f(x_k)$$

Eine Möglichkeit, die Gewichte $A_k$ zu bestimmen, ist, die Funktion $f$ durch ihr Interpolationspolynom $p$ zu ersetzen, das an ihrer Stelle integriert wird:

In der Form von Lagrange lautet $p$:

$$p(x) = \sum_{k=o}^{n} f(x_k)\, \omega_k(x)$$

$$\text{mit} \quad \omega_k(x) = \prod_{\substack{\ell=o \\ \ell \neq k}}^{n} \frac{x - x_\ell}{x_k - x_\ell}$$

$$\Rightarrow \quad I_n = \int_a^b p(x)\,dx = \sum_{k=o}^{n} \int_a^b \omega_k(x)\,dx \; f(x_k)$$

$$\Rightarrow \quad A_k = \int_a^b \omega_k(x)\,dx = \int_a^b \prod_{\substack{\ell=o \\ \ell \neq k}}^{n} \frac{x - x_\ell}{x_k - x_\ell}\,dx$$

Wir substituieren $x = h\,t + a$ und erhalten:

$$A_k = h \int_o^n \prod_{\substack{\ell=o \\ \ell \neq k}}^{n} \frac{t - \ell}{k - \ell}\,dt = h\,a_k$$

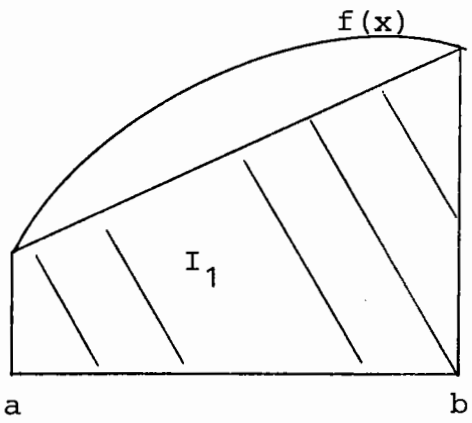
Für $n = 1$ erhalten wir damit die „Trapezregel":

$$a_o = \int_o^1 \frac{t - 1}{-1}\,dt = \frac{1}{2}$$

$$a_1 = \int_o^1 t\,dt = \frac{1}{2}$$

$$I_1 = \frac{h}{2}\,(f(a) + f(b)) = \frac{b-a}{2}\,(f(a) + f(b))$$

Das Integral wird also durch die Fläche eines Trapezes genähert:




Für $n = 2$ ergibt sich die trotz ihrer Einfachheit relativ genaue Simpson'sche Regel:

$$a_o = \frac{1}{3} \qquad a_1 = \frac{4}{3} \qquad a_2 = \frac{1}{3}$$

$$I_2 = \frac{h}{3}\,(f(a) + 4\,f\left(\frac{b+a}{2}\right) + f(b))$$

Die folgende Tabelle enthält die Koeffizienten $a_j$ für $n \leq 4$:

| n | $a_o$ | $a_1$ | $a_2$ | $a_3$ | $a_4$ | | Bezeichnung |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | | | | $\cdot \frac{1}{2}$ | Trapezregel |
| 2 | 1 | 4 | 1 | | | $\cdot \frac{1}{3}$ | Simpson-Regel |
| 3 | 1 | 3 | 3 | 1 | | $\cdot \frac{3}{8}$ | Newton'sche $\frac{3}{8}$ - Regel |
| 4 | 7 | 32 | 12 | 32 | 7 | $\cdot \frac{2}{45}$ | Milne - Regel |

Für $n \geq 8$ können negative Gewichte auftreten, was aus Rundungsfehlergründen nicht gut ist. Wie wir später sehen werden, kann man Formeln höherer Genaugkeit konstruieren, indem man die oben angegebenen Regeln auf Teilintervalle anwendet.

<u>Beispiel:</u> $I = \int_0^1 e^x\,dx = e - 1 = 1.7183$

$$I_1 = \frac{1}{2}\,(1 + e) = 1.8591$$

$$I_2 = \frac{1}{6}\,(1 + 4e^{\frac{1}{2}} + e) = 1.7189$$

$$I_3 = \frac{1}{8}\,(1 + 3e^{\frac{1}{3}} + 3e^{\frac{2}{3}} + e) = 1.7185$$

Man sieht, daß der Übergang von $I_1$ nach $I_2$ einen großen Gewinn an Genauigkeit ergibt.

Der folgende Satz gibt eine Abschätzung für den Fehler $|I-I_n|$:

Satz 24.1:

i) $f \in C^{n+1}[a,b] \Rightarrow |I-I_n| \leq h^{n+2} c_n \operatorname{Max}_{[a,b]} |f^{(n+1)}(x)|$

mit $c_n = \frac{1}{(n+1)!} \int_0^n \prod_{k=0}^{n} |t-k| dt$

ii) $n$ gerade und $f \in C^{n+2}[a,b] \Rightarrow$

$$|I - I_n| \leq h^{n+3} c_n^* \operatorname{Max}_{[a,b]} |f^{(n+2)}(x)|$$

$$\text{mit} \quad c_n^* = \frac{n}{2} c_n$$

Bemerkung: Bei geradem $n$ gewinnt man durch den Übergang zu $n+1$ keine Potenz von $h$. Die Potenzen von $h$ sind optimal, die Konstanten $c_n$, $c_n^*$ könnten verbessert werden.

Beweis:

i) Es gilt:

$$I - I_n = \int_a^b (f-p)(x) dx$$

und nach Satz 19.1:

$$(f-p)(x) = \frac{w(x)}{(n+1)!} f^{(n+1)}(\xi)$$

$$\text{mit} \quad w(x) = \prod_{k=0}^{n} (x - x_k)$$

$$\Rightarrow |I - I_n| \leq \frac{1}{(n+1)!} \int_a^b |w(x)| dx \operatorname{Max}_{[a,b]} |f^{(n+1)}(x)|$$

$c_n$ ergibt sich aus der Berechnung von $\int_a^b |w(x)|dx$:

$$\int_a^b w(x)\,dx = \int_a^b \prod_{k=o}^{n} |x - x_k|\,dx \qquad \begin{matrix} x = ht+a \\ = \end{matrix}$$

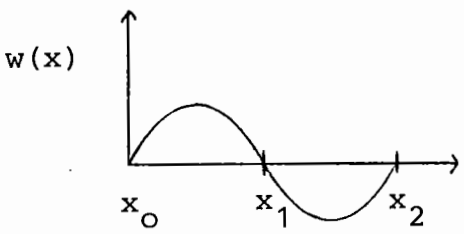
$$= h^{n+2} \int_o^n \prod_{k=o}^{n} |t - k|\,dt$$

$$\Rightarrow \quad |I - I_n| \leq h^{n+2} \frac{1}{(n+1)!} \int_o^n \prod_{k=o}^{n} |t - k|\,dt \quad \max_{[a,b]} |f^{(n+1)}(x)|$$

ii) Für gerades $n$ ist $w$ schiefsymmetrisch bezüglich der Intervallmitte $c = \frac{a+b}{2}$, es gilt also

$$\int_a^b w(x)\,dx = 0$$

Sei z.B. $n = 2$:




Damit hat man

$$\int_a^b (f - p)(x)\,dx = \frac{1}{(n+1)!} \int_a^b w(x) f^{(n+1)}(\xi)\,dx$$

$$= \frac{1}{(n+1)!} \int_a^b w(x) \left\{ f^{(n+1)}(c) + (\xi - c) f^{(n+2)}(\eta) \right\} dx$$

$$= \frac{1}{(n+1)!} \int_a^b w(x)(\xi - c) f^{(n+2)}(\eta)\,dx$$

Wegen $|\xi - c| \leq \frac{b-a}{2} = \frac{nh}{2}$ gilt:

$$\left|\int_a^b (f-p)(x)\,dx\right| \leq \frac{1}{(n+1)!} \int_a^b |w(x)|\,dx \cdot \frac{nh}{2} \max_{[a,b]} |f^{(n+2)}(x)|$$

$$= h^{n+2} c_n \frac{nh}{2} \max_{[a,b]} |f^{(n+2)}(x)|$$

$$= h^{n+3} c_n^* \max_{[a,b]} |f^{(n+2)}(x)|$$

Da das Maximum hoher Ableitungen von $f$ sehr schwer zu bestimmen ist, sind diese Formeln zur praktischen Abschätzung des Fehlers unbrauchbar. Ihr Nutzen liegt in der Information, mit welcher Potenz von $h$ der Fehler abfällt und daß er vom Maximum einer höheren Ableitung abhängt.

Wir konstruieren nun Formeln höherer Genaugkeit. Gegeben sei wieder eine äquidistante Unterteilung $x_k = a + kh$, $h = \frac{b-a}{n}$, $k = 0,\ldots,n$

Wir integrieren stückweise mit der Trapezregel:

$$\int_{x_k}^{x_{k+1}} f(x)\,dx \simeq \frac{h}{2}\,(f(x_k) + f(x_{k+1}))$$

$$\Rightarrow \quad I = \int_{x_o}^{x_n} f(x)\,dx = \sum_{k=o}^{n-1} \int_{x_k}^{x_{k+1}} f(x)\,dx$$

$$\simeq \frac{h}{2} \sum_{k=o}^{n-1} (f(x_k) + f(x_{k+1}))$$

$$= \frac{h}{2}\,(f_o + 2f_1 + 2f_2 + \ldots + 2f_{n-1} + f_n) =$$

$$= T_h$$

mit der Abkürzung $f_k := f(x_k)$.

Diese Formel heißt zusammengesetzte Trapezregel.

Für den Fehler gilt:

$$|I - T_h| = \sum_{k=o}^{n-1} \left| \int_{x_k}^{x_{k+1}} f(x)dx - \frac{h}{2} (f(x_k) + f(x_{k+1})) \right|$$

Nach Satz 24.1 gilt mit $c_1 = \frac{1}{2} \int_0^1 t(1-t)dt = \frac{1}{12}$ :

$$|I - T_h| \leq \frac{1}{12} \sum_{k=o}^{n-1} h^3 \max_{[x_k, x_{k+1}]} |f''(x)|$$

$$\leq \frac{n}{12} h^3 \max_{[a,b]} |f''(x)|$$

$$= \frac{b-a}{12} h^2 \max_{[a,b]} |f''(x)|$$

$$= O(h^2)$$

Für gerades $n$ bilden wir nun analog die zusammengesetzte Simpson-Regel, indem wir die Simpson-Regel auf jeweils zwei aufeinanderfolgende Teilintervalle anwenden und anschließend summieren:

$$I = \int_a^b f(x)dx = \sum_{k=o}^{\frac{n}{2}-1} \int_{x_{2k}}^{x_{2k+2}} f(x)\,dx$$

$$\simeq \frac{h}{3} (f_o + 4f_1 + f_2 + f_2 + 4f_3 + f_4 + \dots)$$

$$= \frac{h}{3} (f_o + 4f_1 + 2f_2 + 4f_3 + 2f_4 + \dots + 2f_{n-2} + 4f_{n-1} + f_n)$$

$$= S_h$$

Die Fehlerabschätzung liefert:

$$|I - S_h| \leq \frac{n}{2} \, c_2^* \, h^5 \underset{[a,b]}{\text{Max}} |f^{(4)}(x)|$$

$$= \frac{b-a}{2} \, c_2^* \, h^4 \underset{[a,b]}{\text{Max}} |f^{(4)}(x)|$$

$$= O(h^4)$$

Beispiele zur zusammengesetzten Trapezregel:

1) $I = \int_0^1 e^x \, dx$

| h | $T_h$ | $I - T_h$ | Quotient aufeinanderfolgender Fehler |
|---|---|---|---|
| 1 | 1.8591 | -0.1409 | - |
| 1/2 | 1.7539 | -0.0357 | 3.95 |
| 1/4 | 1.7272 | -0.0089 | 4.01 |

Der Fehlerquotientzeigt den Abfall mit $h^2$ : Bei Halbierung von $h$ geht der Fehler auf ungefähr ein Viertel zurück.

2) $I = \int_0^1 \sqrt{x} \, dx$

| h | $T_h$ | $I - T_h$ | Quotient aufeinanderfolgender Fehler |
|---|---|---|---|
| 1 | 0.5000 | 0.1667 | - |
| 1/2 | 0.6036 | 0.0631 | 2.67 |
| 1/4 | 0.6433 | 0.0234 | 2.70 |

Hier fällt der Fehler nicht mit $h^2$ ab. Unsere Abschätzung ist nicht anwendbar, da $\sqrt{x} \notin C^2[0,1]$.

## § 25 Das Romberg - Verfahren

Das Romberg-Verfahren beruht auf der Trapezregel. Durch Berechnen von Integrationsformeln mit verschiedener Schrittweite $h$ lassen sich Formeln konstruieren, deren Fehler mit einer hohen Potenz von $h$ abfällt.

Für das Integral

$$I = \int_a^b f(x)dx$$

ergibt die Trapezregel:

$$T_1(h) = \frac{h}{2}\,(f_o + 2f_1 + \ldots + 2f_{n-1} + f_n)$$

mit $f_i := f(x_i)$ , $x_i = a + i\,h$ , $i = 0,\ldots,n$ , $h = \frac{b-a}{n}$ .

Wir entwickeln nun $T_1(h)$ nach Potenzen von $h$ :

Satz 25.1: Sei $f \in C^{2m+2}[a,b]$. Dann gilt die Euler-McLaurin'sche Summenformel:

$$T_1(h) = I + c_1 h^2 + c_2 h^4 + \ldots + c_m h^{2m} + O(h^{2m+2})$$

mit $c_k = (-1)^{k+1} \frac{B_k}{(2k)!}\,(f^{(2k-1)}(b) - f^{(2k-1)}(a))$

und den Bernoulli'schen Zahlen $B_k$:

$$B_1 = \frac{1}{6} \quad , \quad B_2 = \frac{1}{30} \quad , \quad B_3 = \frac{1}{42} \quad , \ldots$$

Beweis: Siehe Stoer I, S. ~~105~~ 115.

Folgerung: Sei $f \in C^{2m+2}(\mathbb{R}^1)$ $2\pi$-periodisch. Dann gilt

$$\int_a^{a+2\pi} f(x)dx = T_1(h) + O(h^{2m+2})$$

Wenn $f$ sogar $\in C^{\infty}(\mathbb{R}^1)$, so gilt

$$\int_a^{a+2\pi} f(x)dx - T_1(h) \to 0 \text{ schneller als jede Potenz von } h .$$

Periodische Funktionen lassen sich also über die volle Periode außerordentlich gut mit der Trapezregel integrieren. Diese vereinfacht sich in diesem Fall zu

$$T_1(h) = h \sum_{j=o}^{n-1} f_j \quad .$$

Wir konstruieren nun Formeln hoher Genauigkeit für beliebige Funktionen. Wir bilden dazu

$$T_1(h) = I + c_1 h^2 + c_2 h^4 + \ldots + c_m h^{2m} + O(h^{2m+2})$$

$$T_1(\tfrac{h}{2}) = I + c_1 2^{-2} h^2 + c_2 2^{-4} h^4 + \ldots + c_m 2^{-2m} h^{2m} + O(h^{2m+2})$$

$$\Rightarrow 2^2 T_1(\tfrac{h}{2}) - T_1(h) = (2^2-1)I + c_2(2^{-2}-1)h^4 + \ldots + c_m(2^{2-2m}-1)h^{2m} + O(h^{2m+2})$$

$$\Rightarrow I = \frac{1}{2^2-1}\left(2^2 T_1(\tfrac{h}{2}) - T_1(h)\right) - c_2 \frac{2^{-2}-1}{2^2-1} h^4 - \ldots - c_m \frac{2^{2-2m}-1}{2^2-1} h^{2m} + O(h^{2m+2})$$

Mit

$$T_2(h) := \frac{1}{2^2-1}\left(2^2 T_1(\tfrac{h}{2}) - T_1(h)\right)$$

erhalten wir

$$I = T_2(h) + c_2' h^4 + \ldots + c_m' h^{2m} + O(h^{2m+2})$$

Durch Wiederholen dieses Verfahrens erhalten wir Formeln höherer Ordnung:

Sei $T_k(h)$ eine Formel der Ordnung $h^{2k}$, d.h.

$$T_k(h) = I + c_k h^{2k} + c_{k+1} h^{2k+2} + \ldots + c_m h^{2m} + O(h^{2m+2})$$

$$T_k(\tfrac{h}{2}) = I + c_k 2^{-2k} h^{2k} + \ldots + c_m 2^{-2m} h^{2m} + O(h^{2m+2})$$

$$\Rightarrow \quad 2^{2k} T_k(\tfrac{h}{2}) - T_k(h) =$$

$$= (2^{2k} - 1)I + c_{k+1}(2^{-2} - 1)h^{2k+2} + \ldots + c_m(2^{2k-2m})h^{2m} + O(h^{2m+2})$$

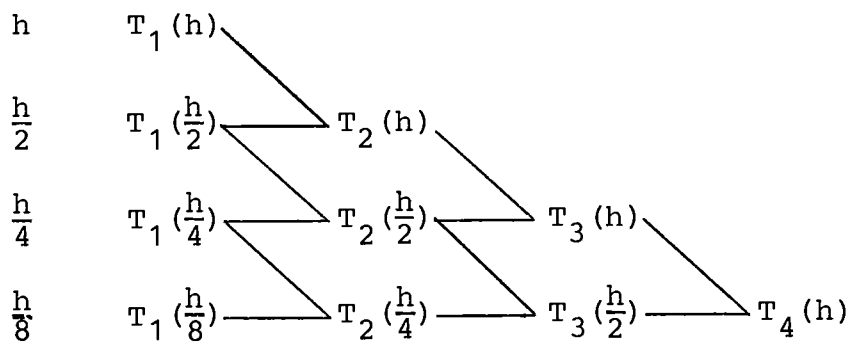
Mit

$$T_{k+1}(h) := \frac{1}{2^{2k}-1} (2^{2k} T_k(\tfrac{h}{2}) - T_k(h))$$

gilt

$$I = T_{k+1}(h) + O(h^{2k+2})$$

Diese Konstruktion von Formeln höherer Ordnung läßt sich im sog. Romberg-Schema darstellen:



Die Rekursion schreibt sich am besten in der Form

$$T_{k+1}(h) = T_k(\tfrac{h}{2}) + \frac{1}{2^{2k}-1}\,(T_k(\tfrac{h}{2}) - T_k(h))$$

Da $T_k(\frac{h}{2})$ und $T_k(h)$ für große $k$ und kleines $h$ jeweils gute Näherungen für $I$ sind, tritt beim Bilden der Differenz Auslöschung auf. Es lohnt also nicht, über $k = 6$ hinauszugehen.

Bei der Berechnung von $T_1(\frac{h}{2})$ wird man die schon in $T_1(h)$ benötigten Funktionswerte mitbenutzen: Mit $f_{i+1/2} = f(x_i + \frac{h}{2})$ ist

$$T_1(\tfrac{h}{2}) = \frac{h}{4}(f_o + 2f_{1/2} + 2f_1 + \ldots + 2f_{n-1/2} + f_n)$$

$$= \frac{h}{4}(f_o + 2f_1 + 2f_2 + \ldots + 2f_{n-1} + f_n)$$

$$+ \frac{h}{2}\,(f_{1/2} + f_{3/2} + \ldots + f_{n-1/2})$$

$$= \underbrace{\tfrac{1}{2}\,T_1(h)}_{\text{bereits bekannt}} + \frac{h}{2}\,(\underbrace{f_{1/2} + f_{3/2} + \ldots + f_{n-1/2}}_{\text{neu zu berechnen}})$$

<u>Beispiel:</u> $I = \int_0^1 e^x\,dx = 1.718281828$

| $h$ | $T_1$ | $T_2$ | $T_3$ | $T_4$ |
|---|---|---|---|---|
| 1 | 1.859140914 | | | |
| $\frac{1}{2}$ | 1.753931092 | 1.718861151 | | |
| $\frac{1}{4}$ | 1.727221904 | 1.718318841 | 1.718282687 | |
| $\frac{1}{8}$ | 1.720518592 | 1.718284155 | 1.718281842 | 1.718281829 |

Die Romberg-Integration wird vor allem in Programmen zur automatischen Integration benutzt. Sie sind etwa folgendermaßen aufgebaut:

Eingabe: Gewünschte relative Genauigkeit $\varepsilon$, Intervallgrenzen $a, b$, ein Unterprogramm zur Auswertung von $f$, maximale Anzahl $n$ der Auswertungen von $f$

Ausgabe: Eine Näherung $\tilde{I}$ für das Integral $I$ mit $|I - \tilde{I}| \leq \varepsilon |I|$, Zuverlässigkeitsindex

Verfahren:

1) Man berechnet etwa 4-6 Spalten des Romberg-Schemas für $h = b - a,\ (b-a)/2, \ldots$

2) Prüfung der Genauigkeit in Spalte $k$. Es gilt

$$T_k(h) = I + c_k h^{2k} + O(h^{2k+2})$$

$$T_k(\tfrac{h}{2}) = I + c_k 2^{-2k} h^{2k} + O(h^{2k+2})$$

$$\Rightarrow\ T_k(\tfrac{h}{2}) - T_k(h) = c_k (2^{-2k}-1) h^{2k} + O(h^{2k+2})$$

$$\Rightarrow\ c_k h^{2k} = \underbrace{\frac{1}{2^{-2k}-1}\,(T_k(\tfrac{h}{2}) - T_k(h))}_{\varepsilon_k(\frac{h}{2})} + O(h^{2k+2})$$

Man prüft, ob $\varepsilon_k(\frac{h}{4}) \sim 2^{-2k}\, \varepsilon_k(\frac{h}{2})$. Falls die Abweichung unter 10% liegt, wird $\varepsilon_k$ als Fehler akzeptiert.

3) Man führt das Romberg-Schema so lange fort, bis in der Spalte ganz rechts $|\varepsilon_k(\frac{h}{2})| \leq \varepsilon |T_k(\frac{h}{2})|$ gilt.

Gute Programme haben außerdem folgende Eigenschaften:

1) Falls die zu integrierende Funktion nicht analytisch ist, erkennen sie dies und wenden ein anderes Verfahren an.

2) Hat f beispielsweise einen Pol, können sie den Grad des Pols schätzen.

3) Teilintervalle, in denen die Funktion leicht zu integrieren ist, werden erkannt und abgespalten.

## § 26 Integration nach Gauß

Wir suchen eine Integrationsformel für das Integral

$$I = \int_a^b w(x) f(x) dx$$

mit einer in $(a,b)$ streng positiven Gewichtsfunktion $w$. Eine Integrationsformel der Form

$$G_n f = \sum_{j=1}^{n} A_j f(x_j)$$

hat die $2n$ freien Parameter $A_j$ und $x_j$. Die Formeln von Newton-Cotes integrieren ein Polynom n-ten Grades exakt. Wir wollen nun fordern, daß

$$G_n f = I \quad \text{für} \quad f \in \mathcal{P}_{2n-1}$$

Dies ergibt gerade $2n$ Bedingungen für die $2n$ Parameter. Der folgende Satz zeigt, daß diese Forderung maximal ist:

Satz 26.1: Es gibt keine Formel $G_n$, die in $\mathcal{P}_{2n}$ exakt ist.

Beweis: Annahme: $G_n f = \int_a^b w f dx \quad \forall f \in \mathcal{P}_{2n}$

Sei $f = \prod_{j=1}^{n} (x - x_j)^2 \in \mathcal{P}_{2n}$

$\Rightarrow \quad G_n f = 0 \neq \int_a^b w f dx > 0$ Widerspruch!

Zur Konstruktion einer in $\mathcal{P}_{2n-1}$ exakten Formel $G_n$ benutzen wir die nach dem Schmidt'schen Verfahren konstruierten orthonormalen Polynome $p_n$:

$$\int_a^b w\, p_n\, p_m\, dx = \begin{cases} 1 & m = n \\ 0 & m \neq n \end{cases}$$

Wir wissen: $p_n$ hat in $(a,b)$ genau $n$ einfache Nullstellen.

Satz 26.2: Es gibt Formeln $G_n$, welche auf $\mathcal{P}_{2n-1}$, exakt sind. Die $x_j$ sind die Nullstellen von $p_n$ und es gilt

$$A_j = \int_a^b w(x) \prod_{\substack{i=1 \\ i\neq j}}^{n} \left( \frac{x - x_i}{x_j - x_i} \right)^2 dx > 0$$

Beweis:

Sei $G_n$ die Newton-Cotes Formel zu den Nullstellen $x_1,\ldots,x_n$ von $p_n$, also

$$G_n f = \sum_{j=1}^{n} \int_a^b w \prod_{\substack{i=1 \\ i\neq j}}^{n} \frac{x - x_i}{x_j - x_i}\, dx\, f(x_j)$$

$G_n$ ist offenbar exakt in $\mathcal{P}_{n-1}$, da ja gerade das Interpolationspolynom vom Grad $n-1$ integriert wird. Ist $f \in \mathcal{P}_{2n-1}$, so schreiben wir:

$$f = q\, p_n + r \qquad \text{mit } q,\ r \in \mathcal{P}_{n-1}$$

$$\Rightarrow \int_a^b w f\, dx = \underbrace{\int_a^b w q p_n\, dx}_{=0 \text{ da } q\in\mathcal{P}_{n-1}} + \int_a^b w r\, dx$$

$$= G_n r \qquad \text{da } r \in \mathcal{P}_{n-1}$$

$$= G_n r + \underbrace{G_n(q\, p_n)}_{=0 \text{ da } p_n(x_j)=0,\ j=1,\ldots,n}$$

$$= G_n(r + q\, p_n)$$

$$= G_n f$$

Also ist $G_n$ exakt in $\mathcal{P}_{2n-1}$ und wir müssen nur noch die Formel für die Gewichte bestätigen. Mit

$$w_j = \prod_{\substack{i=1 \\ i \neq j}}^{n} \frac{x - x_i}{x_j - x_i}$$

ist $w_j^2 \in \mathcal{P}_{2n-2}$, also

$$\int_a^b w\, w_j^2\, dx = G_n(w_j^2) = \sum_{k=1}^{n} A_k\, w_j^2(x_k) =$$

$$= \sum_{k=1}^{n} A_k\, \delta_{jk} = A_j \quad .$$

Beispiel: $[a,b] = [-1,+1]$, $w = 1$.

Die $x_j$ sind die Nullstellen der Legendre-Polynome $P_n$.

| n | $x_1$ | $x_2$ | $x_3$ | $A_1$ | $A_2$ | $A_3$ |
|---|---|---|---|---|---|---|
| 1 | 0 | | | 2 | | |
| 2 | $-\sqrt{\frac{1}{3}}$ | $+\sqrt{\frac{1}{3}}$ | | 1 | 1 | |
| 3 | $-\sqrt{\frac{3}{5}}$ | 0 | $\sqrt{\frac{3}{5}}$ | $\frac{5}{9}$ | $\frac{8}{9}$ | $\frac{5}{9}$ |

$$I = \int_{-1}^{1} e^x\, dx = 2.350402$$

Die Simpson-Regel liefert:

$$I_2 = 2.362054$$

Dagegen ist mit gleich vielen Funktionswertungen

$$G_3 = 2.350337$$

Die Gauß-Formeln sind erheblich genauer als die Formeln von Newton-Cotes. Man kann aus ihnen aber keine einfache Vorschrift zur Konstruktion von Formeln höherer Genauigkeit, wie etwa das Romberg-Verfahren, ableiten. Daher sind sie numerisch ohne Bedeutung.

## § 27 Numerische Differentiation

Eine effektive aber aufwendige Methode zur numerischen Differentiation ist das Differenzieren des Splines, der $f$ an den Stützstellen $x_j$ interpoliert. Wir wollen nur die Methode des Taylor-Abgleichs vorstellen, die zu denselben Formeln führt, wie die Differentiation des Interpolationspolynoms.

Sei $f \in C^{p+1}(\mathbb{R}^1)$. Wir interessieren uns für eine Näherung von $f^{(k)}(0)$, die mit Hilfe der Werte $f(x_j)$ an Stützstellen $x_j$ berechnet wird. Der Satz von Taylor liefert

$$f(ih) = \sum_{\ell=0}^{p} f^{(\ell)}(0) \frac{(ih)^\ell}{\ell!} + O(h^{p+1})$$

Wir bilden die Linearkombination

$$\sum_{i=-q}^{q} \alpha_i f(ih) = \sum_{\ell=0}^{p} f^{(\ell)}(0) \frac{1}{\ell!} h^\ell \sum_{i=-q}^{q} i^\ell \alpha_i + O(h^{p+1})$$

Wir wählen die $\alpha_i$ nun so, daß

$$\sum_{i=-q}^{q} i^\ell \alpha_i = \begin{cases} 1 & \ell = k < p \\ 0 & \text{sonst} \end{cases}, \qquad \ell = 0,\dots,p$$

und erhalten

$$\frac{1}{k!} h^k f^{(k)}(0) = \sum_{i=-q}^{q} \alpha_i f(ih) + O(h^{p+1})$$

Für $2q+1 = p+1$ führt das Gleichungssystem für die $\alpha_i$ auf eine Vandermonde-Matrix, ist also lösbar. Für $2q+1 > p+1$ setzt man einige $\alpha_i = 0$, bis man wieder eine Vandermonde-

Matrix erhält. Die Lösung dieses Gleichungssystems heißt Taylor-Abgleich.

Wir haben also mit

$$D_h^{(k)} f = \frac{k!}{h^k} \sum_{i=-q}^{q} \alpha_i f(ih) = f^{(k)}(0) + O(h^{p+1-k})$$

eine Differentiationsformel der Ordnung $h^{p+1-k}$.

Beispiele: $k = 1$

$$p = 1,\ q = 1,\ \alpha_{-1} = 0,\ \alpha_0 = -1,\ \alpha_1 = +1$$

$$D_h^{(1)} f = \frac{f(h)-f(0)}{h}, \quad f'(0) = D_h^{(1)} f + O(h)$$

$$p = 2,\ q = 1,\ \alpha_{-1} = -\frac{1}{2},\ \alpha_0 = 0,\ \alpha_1 = \frac{1}{2}$$

$$D_h^{(1)} f = \frac{f(h)-f(-h)}{2h} \qquad f \in C^3 \Rightarrow f'(0) = D_h^{(1)} f + O(h^2)$$

$k = 2$

$$p = 3,\ q = 2,\ \alpha_{-2} = 0,\ \alpha_{-1} = \frac{1}{2},\ \alpha_0 = -1,\ \alpha_1 = \frac{1}{2},\ \alpha_2 = 0$$

$$D_h^{(2)} f = \frac{f(h)-2f(0)+f(-h)}{h^2}, \quad f''(0) = D_h^{(2)} f + O(h^2)$$

## § 28 DER FEHLER BEI INTEGRATION UND DIFFERENTIATION

Sei $If = \int_a^b f(x)dx$ zu berechnen. Steht anstelle von $f$ nur eine Näherung $\tilde{f}$ mit relativem Fehler $\varepsilon$ (also $|(f-\tilde{f})(x)| \leq \varepsilon|f(x)|$) zur Verfügung, so kann nur die Näherung $I\tilde{f}$ für $If$ berechnet werden. Es gilt

$$(28.1) \quad |If-I\tilde{f}| \leq \int_a^b |(f-\tilde{f})(x)|dx \leq \varepsilon I|f| \quad .$$

Falls $I|f|$, $|If|$ die gleiche Größenordnung haben (z.B. falls $f \geq 0$), so haben $|If-I\tilde{f}|$, $\varepsilon|If|$ die gleiche Größenordnung, also $I\tilde{f}$ einen relativen Fehler der Größenordnung $\varepsilon$. Die Integration ist in diesem Fall also eine gut konditionierte Aufgabe.

Ist aber $I|f|$ viel größer als $|If|$ (z.B. wenn $f$ eine stark oszillierende Funktion ist), dann ist der relative Fehler von $I\tilde{f}$ viel größer als $\varepsilon$. Die Integration ist dann schlecht konditioniert.

Wir untersuchen nun, ob die Berechnung von $f$ durch eine Integrationsformel der Ordnung $p$

$$I_h f = \sum_{j=1}^n A_j f(x_j) \quad , \quad |I_h f - If| \leq c\,h^p$$

ein gutartiger Algorithmus ist. Es gilt

$$\begin{aligned} |I_h\tilde{f} - If| &\leq |I_h(\tilde{f} - f)| + |(I_h - I)f| \\ &\leq \varepsilon \sum_{j=1}^n |A_j||f(x_j)| + c\,h^p \quad . \end{aligned}$$

Sind nun alle Gewichte $A_j$ positiv, so ist

$$\sum_{j=1}^{n} |A_j||f(x_j)| = \sum_{j=1}^{n} A_j|f(x_j)| = I_h|f| \sim I|f|$$

und damit näherungsweise

$$(28.2) \quad |I_h\tilde{f} - If| \leq \varepsilon I|f| + c\,h^p \quad .$$

Vergleich mit (28.1) zeigt, daß hier $\varepsilon I|f|$ der unvermeidliche, $c\,h^p$ der Algorithmusfehler ist. (Dabei haben wir den bei der Bildung der j-Summe entstehenden Rundungsfehler nicht berücksichtigt; er ist praktisch ohne jede Bedeutung). Ist also die Schrittweite $h$ hinreichend klein, etwa

$$(28.3) \qquad c\,h^p \leq \varepsilon I|f| \quad ,$$

so ist der Algorithmus gutartig. Dies ist nicht der Fall bei negativen Gewichten, denn dann kann

$$\varepsilon \sum |A_j||f(x_j)| >> \varepsilon I|f|$$

sein.

Nun zur Differentiation! Im Prinzip können $\tilde{f}^{(k)}(0)$ (falls dies überhaupt Sinn hat) und $f^{(k)}(0)$ beliebig verschieden sein. Wenden wir trotzdem eine Differentiationsformel der Ordnung $p+1-k$

$$D_h^{(k)} f = \frac{k!}{h^k} \sum_{i=-q}^{q} \alpha_i f(ih), \quad |D_h^{(k)} f - f^{(k)}(0)| \leq c\,h^{p+1-k}$$

an, so wird

$$|D_h^{(k)}\tilde{f} - f^{(k)}(0)| \leq |D_h^{(k)}(\tilde{f} - f)| + |D_h^{(k)}f - f^{(k)}(0)|$$

$$\leq \varepsilon h^{-k} a + c h^{p+1-k} \quad , \tag{28.4}$$

$$a = k! \sum_{i=-q}^{q} |\alpha_i||f(ih)| \quad .$$

Hier kann man nun $h$ nicht gegen Null gehen lassen, weil dabei der Fehler über alle Grenzen wächst. Es gibt hier ein optimales $h = h_o$, bei welchem der Fehler minimal wird. Größenordnungsmäßig kann man $h_o$ aus der Bedingung ("balancing terms")

$$\varepsilon h^{-k} a = c h^{p+1-k}$$

bestimmen zu $h_o = O(\varepsilon^{1/(1+p)})$. Für $h = h_o$ ist dann

$$D_h^{(k)}\tilde{f} - f^{(k)}(0) = O(\varepsilon^{1-k/(p+1)}) \quad .$$

Ein Fehler $\varepsilon$ in $f$ führt also - bei einer Formel der Ordnung $p$ und optimaler Wahl von $h$ - zu einem Fehler $\varepsilon^{\alpha}$, $\alpha = 1 - \frac{k}{p+1} < 1$ in $f^{(k)}$. Dies bedeutet

1) Numerische Differentiation führt zu einem Genauigkeitsverlust.

2) Dieser Verlust ist klein (d.h. $\alpha \sim 1$) falls, $p + 1 >> k$. Insbesondere hängt er von der Ordnung der verwendeten Differentiationsordnung ab.

Beispiel: Berechnung von $f'(0)$ $(k = 1)$

| Formel | p | $h_0$ | $\varepsilon$ |
|---|---|---|---|
| $\frac{1}{h}(f(h)-f(0))$ | 1 | $\varepsilon^{1/2}$ | $\varepsilon^{1/2}$ |
| $\frac{1}{2h}(f(h)-f(-h))$ | 2 | $\varepsilon^{1/3}$ | $\varepsilon^{2/3}$ |
| | 4 | $\varepsilon^{1/5}$ | $\varepsilon^{4/5}$ |
| | 6 | $\varepsilon^{1/7}$ | $\varepsilon^{7/8}$ |

GEWÖHNLICHE DIFFERENTIALGLEICHUNGEN

## § 29 Anfangswertaufgaben gewöhnlicher Differentialgleichungen

Es soll eine ganz kurze Einführung in die Theorie der Anfangswertaufgabe (AWA) gewöhnlicher Differentialgleichungen gegeben werden. Für eine gründliche Behandlung kann man etwa das Buch W. Walter, Gewöhnliche Differentialgleichungen, Springer 1972 (Heidelberger Taschenbuch) konsultieren.

Sei $D \subseteq \mathbb{R}^2$ ein Gebiet und $f \in C(D)$. Die Gleichung

$$y' = f(x,y)$$

heißt gewöhnliche Differentialgleichung 1. Ordnung. Eine Lösung dieser Differentialgleichung ist eine Funktion $y \in C^1$, so daß

$$y'(x) = f(x,y(x))$$

gilt. Die AWA besteht darin, eine solche Lösung zu finden, welche auch noch durch den Punkt $(x_o,y_o)$ geht, d.h.

$$y(x_o) = y_o \quad .$$

BEISPIELE:

1) Bevölkerungswachstum.

Sei $p(t)$ die Größe der Bevölkerung zur Zeit $t$, $g(t,p)$ ihre Geburtsrate, $s(t,p)$ ihre Sterberate. $p_o$ sei die Größe der Bevölkerung zur Zeit $t_o$.

Die Funktion $p$ löst offenbar die AWA

$$\frac{\dot{p}}{p} = g(t,p) - s(t,p) \quad , \qquad p(t_o) = p_o \quad .$$

2) Lineare Differentialgleichung.

$$y' = p(x)y + q(x) \quad , \qquad p, q \in C(a,b) \quad .$$

Die Lösung läßt sich explizit angeben. Man betrachtet zunächst die homogene Differentialgleichung $(q = 0)$

$$y' = p(x)y \quad .$$

Unter der Annahme $y(x) \neq 0$ erhält man der Reihe nach

$$\frac{y'}{y} = p(x), \quad \frac{d}{dx} \ln y = p(x) \ , \quad \ln y = \int_{x_0}^{x} p(t)dt + \ln y_0$$

$$y = y_0 \, e^{\int_{x_0}^{x} p(t)dt} \quad .$$

Die Lösung hängt also von einem freien Parameter $y_0$ ab, welcher offenbar gerade $y(x_0)$ ist und durch die Anfangsbedingung festgelegt wird.

Für die inhomogene Gleichung $(q \neq 0)$ macht man nun den Ansatz

$$y = c(x)y_H$$

mit einer Lösung $y_H$ der homogenen Gleichung. Es folgt

$$y' = c(x)y_H' + c'(x)y_H = c(x)p(x)y_H + c'(x)y_H =$$

$$p(x)y + c'(x)y_H \ .$$

$y$ ist also Lösung von $y' = p(x)y + q(x)$, wenn $c'(x)y_H = q(x)$ gilt. Mit

$$y_H = e^{\int_{x_0}^{x} p(x)dt}$$

ergibt sich

$$c'(x) = q(x)\, e^{-\int_{x_o}^{x} p(x)dt}, \quad c(x) = y_o + \int_{x_o}^{x} q(t)\, e^{-\int_{x_o}^{x} p(s)ds}\, dt$$

mit einer Konstanten $c_o$, und weiter

$$y = \left( y_o + \int_{x_o}^{x} q(t)\, e^{\int_{x_o}^{t} p(s)ds}\, dt \right) e^{\int_{x_o}^{x} p(t)dt}$$

$$= y_o\, e^{\int_{x_o}^{x} p(t)dt} + \int_{x_o}^{x} q(t)\, e^{\int_{t}^{x} p(s)ds}\, dt \quad .$$

Der erste Term ist eine Lösung der homogenen Gleichung mit Anfangswert $y_o$, der zweite die Lösung der inhomogenen Gleichung mit Anfangswert 0.

3) $y' = 1 + y^2$, $y(0) = 0$ hat die Lösung $y = \tan x$. Als stetig differenzierbare Funktion existiert diese nur für $|x| < \pi$ (obwohl $f(x,y) = 1 + y^2$ in $C^\infty(\mathbb{R}^2)$ ist).

4) $y' = y^{1/3}$, $y(0) = 0$ hat für jedes $c \geq 0$ die Lösung

$$y(x) = \begin{cases} \left(\frac{2}{3}(x-c)\right)^{3/2} & , \quad x \geq c \\ 0 & \text{sonst} \end{cases} .$$

Die Lösung einer AWA braucht also nicht eindeutig zu sein.

Zur Formulierung eines Existenz- und Eindeutigkeitssatzes benötigen wir folgende

DEFINITION: $f \in C(D)$ erfüllt in $D$ eine lokale Lipschitz - Bedingung, wenn es zu jedem $(x_o,y_o) \in D$ eine Umgebung $U$ und eine Zahl $L$ gibt mit

$$|f(x,y) - f(x,\bar{y})| \leq L|y - \bar{y}| , \quad \forall\ (x,y), (x,\bar{y}) \in U .$$

Diese Bedingung ist z.B. erfüllt, falls $f \in C^1(D)$.

SATZ 29.1: $f$ erfülle in $D$ eine lokale Lipschitz - Bedingung und $(x_o,y_o) \in D$. Dann gibt es eine Lösung $y$ der AWA

$$y' = f(x,y) \quad , \quad y(x_o) = y_o$$

mit folgender Eigenschaft: Jede weitere Lösung der AWA ist eine Restriktion von $y$.

Dabei heißt eine Funktion $\bar{y} : \bar{I} \to \mathbb{R}$ eine Restriktion von $y : I \to R$, wenn $\bar{I} \subseteq I$ und $y, \bar{y}$ auf $\bar{I}$ übereinstimmen.

Die im Satz genannte Eigenschaft von $y$ bedeutet einmal, daß die Lösungskurve dem Rand von $D$ beliebig nahe kommt, zum anderen, daß die Lösung eindeutig ist.

## § 30 Einschrittverfahren für Anfangswertaufgaben

Die AWA

$$(30.1) \qquad y' = f(x,y) \quad , \quad y(x_o) = y_o$$

besitze in einer abgeschlossenen Umgebung $U$ von $x_o$ eine eindeutig bestimmte Lösung $y$. Wir wollen $y$ auf dem Gitter $I_h : x_k = x_o + k\,h$, $k = 0,1,\ldots$ berechnen. Dazu ersetzen wir (30.1) durch die Differenzengleichung

$$(30.2) \qquad \frac{1}{h}\,(y_{k+1} - y_k) = f_h(x_k\,,\,y_k) \quad , \quad k = 0,1,\ldots$$

mit dem Startwert $y_o$ aus (30.1). Die "Schrittfunktion" $f_h$ wird so gewählt, daß $y_k$ eine Approximation für $y(x_k)$ ist. Wegen

$$y(x_{k+1}) - y(x_k) = \int_{x_k}^{x_{k+1}} f(x,y(x))\,dx$$

muß dazu

$$(30.3) \qquad h f_h(x_k,y_k) \sim \int_{x_k}^{x_{k+1}} f(x,y(x))\,dx$$

sein. (30.2) heißt Einschrittverfahren.

Die einfachste Art, (30.3) zu erfüllen, ist

$$f_h(x_k,y_k) = f(x_k,y_k) \quad .$$

Das so entstehende Einschrittverfahren

$$y_{k+1} = y_k + h\, f(x_k, y_k)$$

heißt Verfahren von Euler oder auch Polygonzugverfahren.

BEISPIEL: $y' = 1 + y^2$, $y(0) = 0$. Euler-Verfahren mit $h = 0.1$:

| k | $x_k$ | $y_k$ | $y(x_k)$ |
|---|---|---|---|
| 0 | 0.0 | 0.0000 | 0.0000 |
| 1 | 0.1 | 0.1000 | 0.1003 |
| 2 | 0.2 | 0.2010 | 0.2027 |
| 3 | 0.3 | 0.3050 | 0.3093 |
| 4 | 0.4 | 0.4143 | 0.4228 |
| 5 | 0.5 | 0.5315 | 0.5463 |

Den "lokalen Diskretisierungsfehler" oder "Abschneidefehler" eines Einschrittverfahrens bekommt man, wenn man die exakte Lösung von (30.1) in (30.2) einsetzt:

$$T_h(x_{k+1}) = \frac{1}{h}\,(y(x_{k+1}) - y(x_k)) - f_h(x_k, y(x_k)), \quad x_{k+1} \in U\ .$$

DEFINITION 30.1: Das Einschrittverfahren (30.2) heißt konsistent (mit (30.1)), falls

$$\lim_{h \to 0} \max_{x_k \in U} |T_h(x_k)| = 0 \quad .$$

Es heißt konsistent von der Ordnung $p$, falls für $h \to 0$

$$\max_{x_k \in U} |T_h(x_k)| = \mathcal{O}(h^p) \quad .$$

BEISPIELE:

1) Euler-Verfahren.

Für $y' = f(x,y)$ ist

$$T_h(x_{k+1}) = \frac{1}{h}\,(y(x_{k+1}) - y(x_k)) - f(x_k, y(x_k))$$

$$= y'(x_k) + \frac{h}{2}\,y''(\tilde{x}_k) - f(x_k, y(x_k)),\quad \tilde{x}_k \in (x_k, x_{k+1})$$

$$= \frac{h}{2}\,y''(\tilde{x}_k) \quad .$$

Also: Ist $y \in C^2(U)$, so ist das Euler-Verfahren konsistent mit der Ordnung 1.

2) Verbessertes Euler-Verfahren.

Nach der Trapezregel ist

$$(30.4)\qquad \int_{x_k}^{x_{k+1}} f(x, y(x))dx = \frac{h}{2}\left\{f(x_k, y(x_k)) + f(x_{k+1}, y_{k+1}))\right\} + O(h^3)$$

für $f \in C^2$ (also $y \in C^3$). Weiter ist für $y' = f(x,y)$

$$y(x_{k+1}) = y(x_k) + h\,y'(x_k) + \sigma(h^2)$$

$$= y(x_k) + h\,f(x_k, y(x_k)) + \sigma(h^2) \quad .$$

Setzt man dies in (30.4) ein, so entsteht

$$\frac{1}{h}\int_{x_k}^{x_{k+1}} f(x,y(x))dx = \frac{1}{2}\left\{f(x_k, y(x_k)) + f(x_{k+1}, y(x_k) + h\,f(x_k, y(x_k)))\right\}$$

$$+ \sigma(h^2) \quad .$$

Mit der Schrittfunktion

$$f_h(x,y) = \frac{1}{2}\{ f(x,y) + f(x+h, y+h\, f(x,y))\}$$

hat man also

$$T_h(x_{k+1}) = \frac{1}{h}\,(y(x_{k+1}) - y(x_k)) - f_h(x_k, y(x_k))$$

$$= \frac{1}{h}\int_{x_k}^{x_{k+1}} y'(x)dx - \frac{1}{h}\int_{x_k}^{x_{k+1}} f(x,y(x))dx + \mathcal{O}(h^2)$$

$$= \mathcal{O}(h^2) \quad .$$

Also hat man Konsistenz von der Ordnung 2.

<u>BEISPIEL:</u> $y' = 1 + y^2$, $y'(0) = 0$, $h = 0.1$. Verbessertes Euler - Verfahren:

| k | $x_k$ | $y_k$ | $y(x_k)$ |
|---|---|---|---|
| 0 | 0.0 | 0.0000 | 0.0000 |
| 1 | 0.1 | 0.1005 | 0.1003 |
| 2 | 0.2 | 0.2030 | 0.2027 |
| 3 | 0.3 | 0.3098 | 0.3093 |
| 4 | 0.4 | 0.4234 | 0.4228 |
| 5 | 0.5 | 0.5470 | 0.5463 |

Die Verbesserung gegenüber dem (einfachen) Euler-Verfahren ist offenkundig.

Wir wollen nun systematisch Verfahren hoher Konsistenzordnung herleiten. Eine vielbenutzte Klasse solcher Verfahren sind die

Runge - Kutta - Verfahren. Das Verfahren m-ter Stufe lautet

$$y_{k+1} = y_k + h(\gamma_1 f_1 + \ldots + \gamma_m f_m)(x_k, y_k) \quad ,$$

$$f_1(x,y) = f(x,y)$$

$$f_2(x,y) = f(x + \alpha_2 h \, , \, y + h\,\beta_{21} f_1(x,y))$$

$$\vdots$$

$$f_m(x,y) = f(x + \alpha_m h \, , \, y + h[\beta_{m1} f_1 + \ldots + \beta_{m,m-1} f_{m-1}](x,y)) \quad .$$

Man stellt alle Koeffizienten in dem Schema

$$\begin{array}{c|ccc} 0 & & & \\ \alpha_2 & \beta_{21} & & \\ \vdots & & & \\ \alpha_m & \beta_{m1} & \cdots & \beta_{m\,m-1} \\ \hline & \gamma_1 & \cdots & \gamma_m \end{array}$$

zusammen. Man nimmt übrigens immer $\alpha_k = \sum_{\ell=1}^{k-1} \beta_{k\ell}$ an.

BEISPIELE:

$m = 1$

$$\begin{array}{c|c} 0 & \\ \hline & 1 \end{array}$$

$$y_{k+1} = y_k + h\, f(x_k, y_k)$$

Euler, $p = 1$

$m = 2$

$$\begin{array}{c|cc} 0 & & \\ 1 & 1 & \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

$$y_{k+1} = y_k + \frac{h}{2}\,(f(x_k,y_k) + f(x_k + h,\, y_k + h\, f(x_k,y_k)))$$

Verbessertes Euler, $p = 2$

$m = 4$

| | | | | |
|---|---|---|---|---|
| 0 | | | | |
| $\frac{1}{2}$ | $\frac{1}{2}$ | | | |
| $\frac{1}{2}$ | 0 | $\frac{1}{2}$ | | |
| 1 | 0 | 0 | | 1 |
| | $\frac{1}{6}$ | $\frac{1}{3}$ | $\frac{1}{3}$ | $\frac{1}{6}$ |

$$y_{k+1} = y_k + \frac{h}{6}(f_1 + 2f_2 + 2f_3 + f_4)$$

$$f_1 = f(x_k, y_k)$$

$$f_2 = f(x_k + \frac{h}{2}, y_k + \frac{h}{2} f_1)$$

$$f_3 = f(x_k + \frac{h}{2}, y_k + \frac{h}{2} f_2)$$

$$f_4 = f(x_k + h, y_k + h f_3)$$

(Standard) Runge-Kutta, $p = 4$.

$m = 3$

| | | | |
|---|---|---|---|
| 0 | | | |
| $\frac{1}{2}$ | $\frac{1}{2}$ | | |
| 1 | -1 | 2 | |
| | $\frac{1}{6}$ | $\frac{4}{6}$ | $\frac{1}{6}$ |

$$y_{k+1} = y_k + \frac{h}{6}(f_1 + 4f_2 + f_3)$$

$$f_1 = f(x_k, y_k)$$

$$f_2 = f(x_k + \frac{h}{2}, y_k + \frac{h}{2} f_1)$$

$$f_3 = f(x_k + h, y_k - h f_1 + 2h f_2)$$

$p = 3$

Herleitung des 3-stufigen Verfahrens (d.h. m=3) der Konsistenzordnung 3 (d.h. p=3) für $f \in C^3$:

$$f_1 = f(x,y) = f = y'$$

$$f_2 = f(x+\alpha_2 h\ ,\ y+\beta_{21} h f_1) \ = \ f(x+\alpha_2 h\ ,\ y+\alpha_2 h f)$$

$$= \underbrace{f}_{y'} + h\alpha_2 \underbrace{(f_x+ff_y)}_{y''} + \frac{1}{2}(\alpha_2 h)^2 \underbrace{(f_{xx}+2f\,f_{xy}+f^2 f_{yy})}_{D} + \mathcal{O}(h^3)$$

$$f_3 = f(x+\alpha_3 h, y+\beta_{31} h f_1+\beta_{32} h f_2) = f(x+\alpha_3 h, y+h\alpha_3 f+h^2\beta_{32}\alpha_2 y'') + \mathcal{O}(h^3)$$

$$= f+h\alpha_3 f_x+(h\alpha_3 f+h^2\beta_{32}\alpha_2 y'')f_y + \frac{1}{2}(\alpha_3 h)^2 f_{xx} + (\alpha_3 h)^2 f\, f_{xy} + \frac{1}{2}(h\alpha_3)^2 f^2 f_{yy} + \mathcal{O}(h^3)$$

$$= \underbrace{f}_{y'} + h\alpha_3 \underbrace{(f_x+ff_y)}_{y''} + h^2(\beta_{32}\,{}_2 y'' f_y + \frac{1}{2}\alpha_3^2 \underbrace{(f_{xx} + 2f\,f_{xy} + f^2 f_{yy})}_{D}) + \mathcal{O}(h3)$$

Lokaler Diskretisierungsfehler:

$$T_h(x+h) = \frac{1}{h}(y(x+h) - y(x)) - (\gamma_1 f_1 + \gamma_2 f_2 + \gamma_3 f_3)(x,y(x))$$

$$= y'(x) + \frac{h}{2} y''(x) + \frac{h^2}{6} y'''(x) - (\gamma_1 + \gamma_2 + \gamma_3)y' - h(\gamma_2\alpha_2 + \gamma_3\alpha_3)y''(x)$$

$$- h^2\left(\frac{1}{2}\alpha_2^2 D\gamma_2 + (\beta_{32}\alpha_2 y'' f_y + \frac{1}{2}\alpha_3^2 D)\gamma_3\right) + \mathcal{O}(h^3)$$

Taylor-Abgleich:

Faktor von $h^0$ : $1 = \gamma_1 + \gamma_2 + \gamma_3$

" $h^1$ : $\frac{1}{2} = \gamma_2\alpha_2 + \gamma_3\alpha_3$

" $h^2$ : $\frac{1}{6} y''' = \frac{1}{2}\alpha_2^2 D\gamma_2 + \beta_{32}\alpha_2\gamma_3 y'' f_y + \frac{1}{2}\alpha_3^2 D\gamma_3$

Zur Umformung der letzten Gleichung berechnen wir

$$y''' = D + y'' f_y \quad .$$

Die Gleichung ist also erfüllt, wenn die beiden Beziehungen

$$\frac{1}{6} = \frac{1}{2} \alpha_2^2 \gamma_2 + \frac{1}{2} \alpha_3^2 \gamma_3 \quad , \qquad \frac{1}{2} = \beta_{32} \alpha_2 \gamma_3$$

erfüllt sind.

Wir haben also 4 Gleichungen für die Koeffizienten $\gamma_1$, $\gamma_2$, $\gamma_3$, $\alpha_2$, $\alpha_3$, $\beta_{32}$ gefunden. Eine Lösung ist

$$\gamma_1 = \gamma_3 = \frac{1}{6} \; , \quad \gamma_2 = \frac{4}{6} \; , \quad \alpha_2 = \frac{1}{2} \; , \quad \alpha_3 = 1 \; , \quad \beta_{32} = 2 \quad .$$

Dies ergibt das in der Tabelle angegebene Verfahren der Stufe 3.

## § 31 Konvergenz von Einschrittverfahren

Vom lokalen Diskretisierungsfehler zu unterscheiden ist der globale Diskretisierungsfehler

$$D_h(x_k) = y_k - y(x_k) \quad .$$

DEFINITION 31.1: Das Einschrittverfahren heißt (gegen die AWA) konvergent, falls

$$\lim_{h \to 0} \max_{x_k \in U} |D_h(x_k)| = 0 \; .$$

Es heißt konvergent von der Ordnung $p$, falls

$$\max_{x_k \in U} |D_h(x_k)| = \mathcal{O}(h^p) \quad .$$

LEMMA 31.1: Seien $q$, $d_k$, $a_k$ nichtnegative Zahlen mit

$$d_{k+1} \leq q\, d_k + a_k \quad , \qquad k = 0,1,\ldots \quad .$$

Dann gilt

$$d_k \leq q^k d_o + \sum_{j=0}^{k-1} q^{k-j-1} a_j \quad .$$

BEWEIS: Durch vollständige Induktion.

SATZ 31.1: $f_h$ erfülle in einer Umgebung der Kurve $(x,y(x))_{x \in U}$ eine Lipschitz - Bedingung, d.h. es gebe von $h, x$ unabhängige Zahlen $d > 0$, $L > 0$ mit

$$|f_h(x,z_1) - f_h(x,z_2)| \leq L|z_1 - z_2| \quad ,$$

$$\text{falls} \quad |y(x) - z_i| \leq d \ , \quad i = 1,2 \quad , \quad x \in U \quad .$$

Dann gilt: Solange $|y(x_k) - y_k| \leq d$ ist, besteht die Abschätzung

$$|y(x_k) - y_k| \leq \frac{1}{L}\left( e^{L|x_k - x_o|} - 1 \right) \max_{j=1}^{k} |T_h(x_j)| \quad .$$

BEWEIS: Aufgrund der Definition des lokalen Diskretisierungsfehlers haben wir

$$y(x_{k+1}) - y(x_k) = h\, f_h(x_k, y(x_k)) + h\, T_h(x_{k+1}) \quad .$$

Das Verfahren lautet

$$y_{k+1} - y_k = h\, f_h(x_k, y_k) \quad .$$

Subtraktion der beiden Beziehungen ergibt mit $d_k = y(x_k) - y_k$

$$d_{k+1} - d_k = h(f_h(x_k, y(x_k)) - f_h(x_k, y_k)) + h\, T_h(x_{k+1}) .$$

Solange $|d_k| \leq d$ ist, folgt aus der Lipschitz-Bedingung

$$|d_{k+1} - d_k| \leq h\, L\, |d_k| + h|T_h(x_{k+1})|$$

oder

$$|d_{k+1}| \leq (1 + h\, L)|d_k| + h|T_h(x_{k+1})| \quad .$$

Das Lemma ergibt nun unmittelbar

$$|d_k| \leq h \sum_{j=0}^{k-1} (1 + h L)^{k-j-1} |T_h(x_{j+1})|$$

$$\leq h \sum_{j=0}^{k-1} (1 + h L)^{k-j-1} \max_{j=1}^{k} |T_h(x_j)| \quad .$$

Die behauptete Abschätzung ergibt sich durch Aufsummieren der geometrischen Reihe.

■

SATZ 31.2: Das Einzelschrittverfahren sei konsistent (von der Ordnung p) und $f_h$ erfülle die Voraussetzung von Satz 31.1. Dann ist das Verfahren konvergent (von der Ordnung p ).

BEWEIS: Man wähle $h_o$ so klein, daß für alle $x_k \in U$

$$\frac{1}{L} \left( e^{L(x_k - x_o)} - 1 \right) \max_{j=1}^{k} |T_h(x_j)| \leq d \quad .$$

Dann gilt die Abschätzung von Satz 31.1 für alle $x_h \in U$ , und die Konvergenz (von der Ordnung p) folgt.

## § 32 Mehrschrittverfahren

Ein lineares m-Schrittverfahren hat die Form

$$\sum_{\nu=0}^{m} \alpha_\nu \, y_{k+\nu} = h \sum_{\nu=0}^{m} \beta_\nu \, f(x_{k+\nu}, y_{k+\nu}) \quad , \quad k = 0,1,\ldots$$

$$y_k = \bar{y}_k \quad , \quad k = 0,\ldots,m-1 \quad .$$

Es benötigt also $m$ Startwerte $\bar{y}_0,\ldots,\bar{y}_{m-1}$. Diese können etwa durch ein Einschrittverfahren gewonnen werden. Gegenüber den Einschrittverfahren besitzen Mehrschrittverfahren den Vorteil, daß sie pro Schritt nur eine Funktionsauswertung (nämlich $f_\nu(x_{k+\nu},y_{k+\nu})$, $\nu$ der größte Index mit $\beta_\nu \neq 0$ ) benötigen.
Ein Beispiel eines 2-Schrittverfahrens ist die Mittelpunktsregel

$$y_{k+2} - y_k = 2h \, f(x_{k+1},y_{k+1}) \quad .$$

Ein Einschrittverfahren heißt explizit, wenn $\beta_m = 0$. Dann läßt sich $y_{k+m}$ unmittelbar durch $y_{k+m-1},\ldots,y_k$ ausdrücken. Ist $\beta_m \neq 0$ , so tritt $y_{k+m}$ auch auf der rechten Seite auf und man muß $y_{k+m}$ durch Lösen einer nichtlinearen Gleichung berechnen. Dies kann iterativ in der Form

$$\alpha_m \, y_{k+m}^{(t+1)} + \sum_{\nu=0}^{m-1} \alpha_\nu \, y_{k+\nu} = h \, \beta_m \, f(x_{k+m},y_{k+m}^{(t)}) + h \sum_{\nu=0}^{m-1} \beta_\nu \, f(x_{k+\nu},y_{k+\nu})$$

geschehen. Wegen

$$\frac{\partial\, y_{k+m}^{(t+1)}}{\partial\, y_{k+m}^{(t)}} = h\,\frac{\beta_m}{\alpha_m}\; f_y(x_{k+m}, y_{k+m}^{(t)})$$

gewinnt man bei jedem Iterationsschritt einen Faktor $\sigma(h)$ an Genauigkeit. Die Iteration konvergiert für kleine $h$ also sehr schnell. Den Startwert $y_{m+k}^{(0)}$ kann man etwa durch ein explizites Verfahren berechnen. Man kombiniert also ein explizites mit einem impliziten Verfahren. In diesem Zusammenhang heißt das explizite Verfahren Prädiktor, das implizite Verfahren Korrektor, und man spricht von Prädiktor - Korrektor - Verfahren.

Im Folgenden werden wir die rückwärtsgenommenen Differenzen

$$\begin{aligned}
\nabla y_k &= y_k - y_{k-1} \\
\nabla^2 y_k &= \nabla y_k - \nabla y_{k-1} = y_k - 2y_{k-1} + y_{k-2} \\
&\vdots \\
\nabla^q y_k &= \nabla\nabla^{q-1} y_k
\end{aligned}$$

benutzen.

LEMMA 32.1: Es gilt für $q \geq 0$

$$\nabla^q y_k = \sum_{\nu=0}^{q} (-1)^\nu \binom{q}{\nu} y_{k-\nu} \quad , \quad y_{k-q} = \sum_{\nu=0}^{q} (-1)^\nu \binom{q}{\nu} \nabla^\nu y_k \quad .$$

BEWEIS: Wir definieren auf dem linearen Raum der Folgen $y = (y_k)_{k=-\infty,+\infty}$ den linearen Operator

$$(Ty)_k = y_{k-1} \quad .$$

Die binomische Formel ergibt

$$(I - T)^q = \sum_{\nu=0}^{q} \binom{q}{\nu} (-1)^\nu T^\nu \quad .$$

Wegen $I - T = \nabla$ , $(T^\nu y)_k = y_{k-\nu}$ ist dies die erste Formel. Die zweite bekommen wir ganz entsprechend aus

$$T^q = (I - \nabla)^q = \sum_{\nu=0}^{q} \binom{q}{\nu} (-\nabla)^\nu \quad .$$

LEMMA 32.2: Das Polynom $p$ vom Grade $\leqq q$ mit $p(x_{k-\ell}) = y_{k-\ell}$ $\ell = 0,\ldots,q$ ist

$$p(x) = \sum_{\nu=0}^{q} (-1)^\nu \binom{-s}{\nu} \nabla^\nu y_k \quad , \quad s = \frac{x-x_k}{h} \quad .$$

Hier sind die Binomialkoeffizienten für reelle $s$ durch

$$\binom{s}{\nu} = \frac{1}{\nu!} s(s-1) \ldots (s - (\nu - 1)) \quad .$$

erklärt.

BEWEIS: $p$ ist ein Polynom vom Grade $\leqq q$, denn es ist

$$\binom{-s}{\nu} = \frac{1}{\nu!} (-s - \nu + 1) \ldots (-s) \in \mathcal{P}_\nu \quad .$$

Weiter gilt für $0 \leqq \mu \leqq q$

$$\begin{aligned} p(x_{k-\mu}) &= \sum_{\nu=0}^{q} (-1)^\nu \binom{\mu}{\nu} \nabla^\nu y_k \\ &= \sum_{\nu=0}^{\mu} (-1)^\nu \binom{\mu}{\nu} \nabla^\nu y_k \\ &= y_{k-\mu} \end{aligned}$$

nach Lemma 32.1. ∎

Zur Aufstellung konkreter Mehrschrittverfahren gibt es grundsätzlich zwei Möglichkeiten:

(a) Integration.

Aus der Differentialgleichung folgt durch Integration

$$y(x_{k+m}) - y(x_{k+\ell}) = \int_{x_{k+\ell}}^{x_{k+m}} f(x,y(x))dx \quad .$$

Man ersetzt nun $f(x,y(x))$ durch das Interpolationspolynom $p$ vom Grade $m$ an den Stellen $x_k,\ldots,x_{k+m}$ (implizites Verfahren) oder vom Grade $m-1$ an den Stellen $x_k,\ldots,x_{k+m-1}$ (explizite Verfahren) und setzt

$$y_{k+m} - y_{k+\ell} = \int_{x_{k+\ell}}^{x_{k+m}} p(x)dx \quad .$$

Als Stützwerte werden bei der Interpolation die Zahlen $f(x_j,y_j)$ genommen. Es ist dann $p(x)$ eine lineare Funktion der Zahlen $f(x_j,y_j)$.

Die verschiedenen Verfahren unterscheiden sich durch ihr Integrationsintervall $(x_{k+\ell},x_{k+m})$ und durch die Stützstellen von $p$. Wir betrachten folgende Möglichkeiten:

| Intervall / Stützstellen | $(x_{k+m-1}, x_{k+m})$ | $(x_{k+m-2}, x_{k+m})$ | |
|---|---|---|---|
| $x_k, \ldots, x_{k+m-1}$ | Adams-Bashforth | Nyström | explizit |
| $x_k, \ldots, x_m$ | Adams-Moulton | Milne-Simpson | implizit |

Adams-Bashforth:

$$y_{k+m} - y_{k+m-1} = \int_{x_{k+m-1}}^{x_{k+m}} p(x)dx \ , \quad p(x) = \sum_{\nu=0}^{m-1} (-1)^\nu \binom{-s}{\nu} \nabla^\nu f_{k+m-1}$$

$$= h(\gamma_0 f_{k+m-1} + \gamma_1 \nabla^1 f_{k+m-1} + \ldots + \gamma_{m-1} \nabla^{m-1} f_{k+m-1})$$

Dabei haben wir $f_k$ für $f(x_k, y_k)$ geschrieben.

$$\gamma_\nu = \frac{1}{h} \int_{x_{k+m-1}}^{x_{k+m}} (-1)^\nu \binom{-s}{\nu} dx \qquad s = \frac{x - x_{k+m-1}}{h} \qquad \text{(Substitution)}$$

$$= (-1)^\nu \int_0^1 \binom{-s}{\nu} ds$$

Adams-Moulton:

$$y_{k+m} - y_{k+m-1} = \int_{x_{k+m-1}}^{x_{k+m}} p(x)dx \ , \quad p(x) = \sum_{\nu=0}^{m} (-1)^\nu \binom{-s}{\nu} \nabla^\nu f_{k+m}$$

$$= h(\gamma_0 f_{k+m} + \gamma_1 \nabla^1 f_{k+m} + \ldots + \gamma_m \nabla^m f_{k+m}) \ ,$$

$$\gamma_\nu = \frac{1}{h} (-1)^\nu \int_{x_{k+m-1}}^{x_{k+m}} \binom{-s}{\nu} dx \ , \quad s = \frac{x - x_{k+m}}{h}$$

$$= (-1)^\nu \int_{-1}^{0} \binom{-s}{\nu} ds$$

Nyström:

$$y_{k+m}-y_{k+m-1} = h(\gamma_o f_{k+m-1}+\gamma_1 \nabla^1 f_{k+m-1}+\ldots+\gamma_{m-1}\nabla^{m-1} f_{k+m-1}) ,$$

$$\gamma_\nu = (-1)^\nu \int_{-1}^{+1} \binom{-s}{\nu} ds$$

Milne-Simpson:

$$y_{k+m}-y_{k+m-2} = h(\gamma_o f_{k+m} + \gamma_1 \nabla^1 f_{k+m}+\ldots+\gamma_m \nabla^m f_{k+m})$$

$$\gamma_\nu = (-1)^\nu \int_{-1}^{0} \binom{-s}{\nu} ds$$

Die $\gamma_\nu$ sind in Tabellen erfaßt (siehe z.B. Henrici, S. 191):

| $\nu$ | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| Adams-Bashforth | 1 | $\frac{1}{2}$ | $\frac{5}{12}$ | $\frac{3}{8}$ | $\frac{251}{720}$ |
| Adams-Moulton | 1 | $-\frac{1}{2}$ | $-\frac{1}{12}$ | $-\frac{1}{24}$ | $-\frac{19}{720}$ |
| Nyström | 2 | 0 | $\frac{1}{3}$ | $\frac{1}{3}$ | $\frac{29}{90}$ |
| Milne-Simpson | 2 | -2 | $\frac{1}{3}$ | 0 | $-\frac{1}{90}$ |

Als Beispiel für den Gebrauch dieser Tabelle betrachten wir das 2-Schritt-Nyström-Verfahren. Mit $\gamma_o = 2$, $\gamma_1 = 0$ aus der entsprechenden Zeile der Tabelle ergibt sich mit $m = 2$

$$y_{k+2} - y_k = 2hf_{k+1} ,$$

also gerade die Mittelpunktsregel.

(b) Differentiation.

In der Differentialgleichung ersetzt man die Ableitung in einem Punkt $x_{k+\ell}$ durch die Ableitung des Interpolationspolynoms vom Grade m mit Stützstellen $x_k,\ldots,x_{k+m}$ und Stützwerten $y_k,\ldots,y_{k+m}$:

$$p'(x_{k+\ell}) = f(x_{k+\ell},y_{k+\ell}) \quad ,$$

$$p(x) = \sum_{\nu=0}^{m} (-1)^\nu \binom{-s}{\nu} \nabla^\nu y_{k+m} \quad , \quad s = \frac{x-x_{k+m}}{h} \quad .$$

Dies ergibt ein Verfahren der Form

$$\sum_{\nu=1}^{m} \gamma_{\nu,m-\ell} \nabla^\nu y_{k+m} = h f_{k+\ell} \quad ,$$

$$\gamma_{\nu,m-\ell} = h \frac{d}{dx} (-1)^\nu \binom{-s}{\nu} \Big|_{x=x_{k+\ell}}$$

$$= (-1)^\nu \frac{d}{ds} \binom{-s}{\nu} \Big|_{s=\ell-m} \quad .$$

Offenbar ist $\gamma_{o,m-\ell} = 0$. Einige $\gamma_{\nu,r}$ finden sich in folgender Tabelle:

| r \ ν | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 0 | 1 | $\frac{1}{2}$ | $\frac{1}{3}$ | $\frac{1}{4}$ |
| 1 | 1 | $-\frac{1}{2}$ | $-\frac{1}{6}$ | $-\frac{1}{12}$ |
| 2 | 1 | $-\frac{3}{2}$ | $\frac{1}{3}$ | $\frac{1}{12}$ |

Das 2-Schritt-Verfahren mit $\ell = 1$ lautet zum Beispiel

$$\nabla^1 y_{k+2} - \frac{1}{2} \nabla^2 y_{k+2} = h\, f_{k+1} \qquad \text{oder}$$

$$y_{k+2} - y_k = 2h f_{k+1} \quad ,$$

also wieder die Mittelpunktsregel.

## § 33 Konvergenz von Mehrschrittverfahren

Den lokalen Diskretisierungsfehler eines Mehrschrittverfahrens erklärt man wie beim Einschrittverfahren durch

$$T_h(x_{k+m}) = \frac{1}{h} \sum_{\nu=0}^{m} \alpha_\nu \, y(x_{k+\nu}) - \sum_{\nu=0}^{m} \beta_\nu \, f(x_{k+\nu}, y(x_{k+\nu}))$$

$$= \frac{1}{h} \sum_{\nu=0}^{m} \alpha_\nu \, y(x_{k+\nu}) - \sum_{\nu=0}^{m} \beta_\nu \, y'(x_{k+\nu})$$

für die exakte Lösung y. Die Definition der Konsistenz erfolgt dann wörtlich wie beim Einschrittverfahren.

BEISPIEL:

1) Um die Konsistenzordnung des Adams-Bashforth-Verfahrens zu bestimmen, erinnern wir uns an die Herleitung des Verfahrens. Danach ist

$$T_h(x_{k+m}) = \frac{1}{h} \left(y(x_{k+m}) - y(x_{k+m-1})\right) - \frac{1}{h} \int_{x_{k+m-1}}^{x_{k+m}} p(x)dx \quad ,$$

wo p das Interpolationspolynom vom Grade m-1 der Funktion f(x,y(x)) an den Stützstellen $x_k, \ldots, x_{k+m-1}$ ist. Nach § 19 ist für $f \in C^m$

$$p - f = \mathcal{O}(h^m) \quad ,$$

so daß wir

$$T_h(x_{k+m}) = \frac{1}{h} \int_{x_{k+m-1}}^{x_{k+m}} (f(x,y(x)) - p(x))dx = \mathcal{O}(h^m)$$

erhalten. Die Konsistenzordnung ist also (mindestens) m. Ebenso sieht man, daß die Konsistenzordnung des Nyström -

Verfahrens m ist, während die Konsistenzordnung der beiden impliziten Verfahren Adams-Moulton und Milne-Simpson m+1 ist.

2) $y_{k+2}-(1+a)y_{k+1}+ay_k = \frac{h}{2}((3-a)f_{k+1}-(1+a)f_k)$ .

Es ist für $y \in C^4$

$$y(x_{k+\nu}) = y(x_k) + \nu h y'(x_k) + \frac{1}{2}\nu^2h^2y''(x_k) + \frac{1}{6}\nu^3h^3y'''(x_k)+O(h^4).$$

$$y'(x_{k+\nu}) = y'(x_k)+ \nu h y''(x_k) + \frac{1}{2}\nu^2h^2y'''(x_k) + \sigma(h^3)$$

und damit

$$\begin{aligned} T_h(x_{k+m}) &= h\left(\frac{4}{2} - \frac{1}{2}(1+a) - \frac{1}{2}(3-a)\right)y''(x_k) \\ &\quad + h^2\left(\frac{8}{6} - \frac{1}{6}(1+a) - \frac{1}{4}(3-a)\right)y'''(x_k) + \sigma(h^3) \\ &= h^2\left(\frac{5}{12} + \frac{a}{12}\right)y'''(x_k) + \sigma(h^3) \quad . \end{aligned}$$

Wir haben also Konsistenzordnung p= 2 für $a \neq -5$ , sonst p=3.

Wir kommen nun zu einem wichtigen Begriff. Numerische Experimente mit dem im letzten Beispiel angegebenen Verfahren ergeben befriedigende Resultate für a=0 (d.h. Adams-Bashforth mit m=2), aber unbrauchbare für a = -5. Die Konsistenzordnung kann also für die Konvergenz nicht, wie bei den Einschrittverfahren, das einzig Maßgebende sein. Wir werden

sehen, daß bei Mehrschrittverfahren neben der Konsistenz die Stabilität notwendig für Konsistenz ist.

Sei für $\lambda \in \mathbb{C}$

$$\rho(\lambda) = \sum_{\nu=0}^{m} \alpha_\nu \lambda^\nu , \qquad \sigma(\lambda) = \sum_{\nu=0}^{m} \beta_\nu \lambda^\nu .$$

Die Eigenschaften dieser beiden Polynome werden sich als für die Eigenschaften des Mehrschrittverfahrens wichtig erweisen.

DEFINITION 33.1: Ein Mehrschrittverfahren heißt stabil, wenn für die Nullstellen $\lambda$ von $\rho$ folgende Bedingung erfüllt ist

a) $|\lambda| \leq 1$ b) Ist $|\lambda| = 1$, so ist $\lambda$ einfach.

BEISPIELE:

1) Adams-Bashforth.
Es ist $\rho(\lambda) = \lambda^{m-1}(\lambda-1)$. Nullstellen sind $\lambda = 0$ ((m-1)-fach) und $\lambda = 1$ (einfach). Also ist das Verfahren stabil.

2) Nyström.
Es ist $\rho(\lambda) = \lambda^{m-2}(\lambda^2-1)$. Nullstellen sind $\lambda = 0$ ((m-2)--fach) und $\lambda = \pm 1$ (jeweils einfach). Also ist das Verfahren stabil.

3) Das oben genannte Verfahren mit $\rho(\lambda) = \lambda^2 - (1+a)\lambda + a = (\lambda-a)(\lambda-1)$. Das Verfahren ist stabil genau dann, wenn $|a| \leq 1$, aber $a \neq 1$ ist.

DEFINITION 33.2: Ein Mehrschrittverfahren heißt konvergent, wenn für alle Startwerte $\bar{y}_k$ mit $\lim_{h\to 0} |y(x_k) - y_k| = 0$ , $k=0,\ldots,m-1$

$$\lim_{h\to 0} \max_{x_k \in U} |y(x_k) - y_k| = 0$$

gilt. Es heißt konvergent von der Ordnung $p$, wenn aus

$y(x_k) - \bar{y}_k = \mathcal{O}(h^p)$, $k=0,\ldots,m-1$

$$\max_{x_k \in U} |y(x_k) - y_k| = \mathcal{O}(h^p)$$

folgt.

Wir wollen zeigen, daß Konvergenz gleichbedeutend ist mit Konsistenz und Stabilität. Dazu benötigen wir einige einfache Tatsachen über Differenzengleichungen.

Unter einer linearen Differenzengleichung mit konstanten Koeffizienten versteht man eine Gleichung der Form

$$\sum_{\nu=0}^{m} \alpha_\nu z_{k+\nu} = c_k \quad , \quad k=0,1,\ldots \quad .$$

Die Gleichung heißt homogen, falls $c_k = 0$, andernfalls inhomogen. Für die Lösung der homogenen Gleichung macht man den Ansatz $z_k = \lambda^k$. Dies ist eine Lösung, wenn

$$\sum_{\nu=0}^{m} \alpha_\nu \lambda^{k+\nu} = \lambda^k \rho(\lambda) = 0 \quad , \quad k=0,1,\ldots \quad ,$$

d.h. wenn $\rho(\lambda) = 0$ ist. Ist $\lambda$ eine zweifache Nullstelle von $\rho$, so ist $\rho'(\lambda) = 0$ und damit auch

$$\sum_{\nu=0}^{m} \alpha_\nu (k+\nu)\lambda^{k+\nu} = \lambda^k \left\{ k \sum_{\nu=0}^{m} \alpha_\nu \lambda^\nu + \sum_{\nu=0}^{m} \alpha_\nu \nu \lambda^\nu \right\}$$

$$= \lambda^k \{k\, \rho(\lambda) + \rho'(\lambda)\} = 0 \quad ,$$

d.h. es ist auch $z_k = k\lambda^k$ Lösung. Genau so sieht man, daß im Falle einer r-fachen Wurzel $\lambda$ die Folgen $z_k = \lambda^k$, $z_k = k\lambda^k, \ldots, z_k = k^{r-1}\lambda^k$ Lösungen sind. Damit hat man aber auch schon alle Lösungen der homogenen Differenzengleichungen gefunden.

SATZ 33.1: Seien $\lambda_1, \ldots, \lambda_n$ die Nullstellen von $\rho$ mit den Vielfachheiten $r_1, \ldots, r_n$. Dann sind

$$\lambda_j^k \quad , \quad k\lambda_j^k, \ldots, k^{r_j-1}\lambda_j^k \qquad , \quad j=1,\ldots,n$$

$m = r_1 + \ldots + r_n$ Lösungen der homogenen Differenzengleichung. Jede weitere Lösung $z_k$ ist eine Linearkombination dieser Lösungen, d.h. es gibt mit Konstanten $a_{jr}$

$$z_k = \sum_{j=1}^{n} \sum_{r=0}^{r_j-1} a_{jr}\, k^r\, \lambda_j^k \qquad .$$

Die $a_{rj}$ sind eindeutig bestimmt.

BEWEIS: Sei $\alpha_m \neq 0$. Die Differenzengleichung kann dann in der Form

$$D_{k+1} = AD_k \quad , \quad D_k = \begin{bmatrix} d_k \\ \cdot \\ \cdot \\ \cdot \\ d_{k+m-1} \end{bmatrix} \quad , \quad A = \begin{bmatrix} 0 & 1 & 0 & \cdots & & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ & \cdot & & \cdot & & \\ & & \cdot & & \cdot & \\ & & & \cdot & & \\ 0 & & & & 0 & 1 \\ -\alpha_0 & -\alpha_1 & \cdot & \cdot & \cdot & -\alpha_{m-1} \end{bmatrix}$$

geschrieben werden.

Dann ist also $D_k = A^k D_0$. Ist $J = X^{-1} A X$ die Jordan'sche Normalform, so ist

$$D_k = X^{-1} J^k X D_0 \quad .$$

In unserem Fall haben die Jordan-Kästchen $J_1,\ldots,J_n$ zu den Eigenwerten $\lambda_1,\ldots,\lambda_n$ die Dimensionen $r_1,\ldots,r_n$. Die Potenzen von $J_\ell$ haben wir schon in §16 ausgerechnet; wir fanden

$$J^k = \lambda_\ell^k (A_0 + kA_1 + \ldots + k^{r_\ell - 1} A_{r_\ell - 1})$$

mit gewissen Matrizen $A_j$, die noch von $\lambda_\ell$ abhängen. $D_k$ ist also wirklich als Linearkombination der genannten Ausdrücke darstellbar.

∎

Als wichtige Folgerung aus Satz 33.1 haben wir:

SATZ 33.2: Ein Mehrschrittverfahren ist genau dann stabil, wenn alle Lösungen der Differenzengleichung

$$\sum_{\nu=0}^{m} \alpha_\nu z_{k+\nu} = 0 \qquad , \quad k = 0,1,\ldots$$

für $k \to \infty$ beschränkt bleiben.

SATZ 33.3: Sei $A$ eine $(m,m)$-Matrix und $\rho(A)$ ihr Spektralradius. Alle Eigenwerte von $A$ mit Betrag $\rho(A)$ seien algebraisch einfach. Dann gibt es eine Vektornorm $\| \ \|$, so daß $\|A\| = \rho(A)$.

BEWEIS: Seien $\lambda_1,\dots,\lambda_r$ die Eigenwerte von $A$ mit $|\lambda_i| = \rho(A)$. Dann gibt es eine Matrix $X$, so daß

$$X^{-1} A X = \left(\begin{array}{ccc|c} \lambda_1 & & & \\ & \ddots & & O \\ & & \lambda_r & \\ \hline & O & & B \end{array}\right) ,$$

wo die $(m-r,m-r)$-Matrix $B$ nur noch die Eigenwerte mit Betrag $< \rho(A)$ hat. Nach Satz 16.1 gibt es eine Norm $\| \ \|_{m-r}$ in $\mathbb{C}^{m-r}$ mit $\|B\|_{m-r} \leq \rho(A)$. Führen wir nun in $\mathbb{C}^m$ die Norm

$$\left\| \begin{pmatrix} x_r \\ x_{m-r} \end{pmatrix} \right\| = \operatorname{Max} \{ \|x_r\|_r, \|x_{m-r}\| \}$$

ein, so leistet die Norm $\|X^{-1}x\|$ das Gewünschte. ∎

SATZ 33.4: $f$ erfülle in einer Umgebung der Kurve $(x, y(x))_{x\in U}$ eine Lipschitz-Bedingung, d.h. es gebe von $x$ unabhängige Zahlen $d > 0$, $L > 0$ mit

$$|f(x,z_1) - f(x,z_2)| \leq L|z_1 - z_2|$$

falls $|z_i - y(x)| \leq d$, $i=1,2$. Das Mehrschrittverfahren sei stabil. Dann gibt es Konstanten $C_1, C_2$, $h_0 > 0$, so daß für $h < h_0$

$$|y(x_k)-y_k| \leq c_1 e^{c_2(x_k-x_0)} \left\{ \operatorname*{Max}_{k=0}^{m-1} |y(x_k)-y_k| + \operatorname*{Max}_{j=m}^{k} |T_h(x_j)| \right\} ,$$

solange $|y(x_k)-y_k| \leq d$.

BEWEIS: Nach Definition des lokalen Diskretisierungsfehlers ist

$$\sum_{\nu=0}^{m} \alpha_\nu y(x_{k+\nu}) = h \sum_{\nu=0}^{m} \beta_\nu f(x_{k+\nu}, y(x_{k+\nu})) + h\, T_h(x_{k+m}) \quad ,$$

und das Verfahren lautet

$$\sum_{\nu=0}^{m} \beta_\nu y_{k+\nu} = h \sum_{\nu=0}^{m} \beta_\nu f(x_{k+\nu}, y_{k+\nu}) \quad .$$

Substraktion ergibt mit $d_k = y(x_k) - y_k$

$$\sum_{\nu=0}^{m} \alpha_\nu d_{k+\nu} = h \sum_{\nu=0}^{m} \beta_\nu (f(x_{k+\nu}, y(x_{k+\nu})) - f(x_{k+\nu}, y_{k+\nu}) + h\, T_h(x_{k+m})$$

$$= h\, c_k \quad .$$

Mit Hilfe der Matrizen und Vektoren ($\alpha_m = 1$)

$$A = \begin{bmatrix} 0 & 1 & 0 & & & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ \vdots & & & & & \\ 0 & & \dots & & 0 & 1 \\ -\alpha_0 & -\alpha_1 & \dots & & & -\alpha_{m-1} \end{bmatrix}, \quad D_k = \begin{bmatrix} d_k \\ \vdots \\ d_{k+m-1} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

können wir dies auch in der Form

$$D_{k+1} = A\, D_k + h\, c_k\, B$$

schreiben. A hat $\rho(\lambda)$ als charakteristisches Polynom. Nach Satz 33.3 gibt es also eine Vektornorm $\| \ \|$, so daß $\|A\| \leq 1$. Es folgt

$$\|D_{k+1}\| \leq \|D_k\| + h|c_k|\|B\| \quad .$$

Solange $|y_k - y(x_k)| \leq d$ gilt, haben wir

$$|c_k| \leq L \sum_{\nu=0}^{m} |\beta_\nu||d_{k+\nu}| + |T_h(x_{k+m})| \leq K(\|D_k\| + \|D_{k+1}\|) + T_h(x_{k+m})$$

mit einer geeigneten Konstanten K. Für $h\|B\|\,\|K\| < 1$ ist also

$$\|D_{k+1}\| \leq q\|D_k\| + ha_k \quad ,$$

$$q = (1+h\|B\|K) \,/\, (1-h\|B\|K) \;, \quad a_k = |T_h(x_{k+m})| \,/\, (1-h\|B\|K) \quad .$$

Nach Lemma 31.1 folgt

$$\|D_k\| \leq q^k\|D_0\| + h \sum_{j=0}^{k-1} q^{k-j-1} a_j \quad .$$

Nun benutzen wir die Ungleichung

$$\frac{1+x}{1-x} \leq 1 + 4x \quad , \quad |x| \leq \frac{1}{2} \quad .$$

Dann wird für $h\|B\|K \leq 2$ $\quad q \leq 1 + 4\,h\|B\|K$ und damit

$$q^k \leq \left( 1+4\;B\;K\;\frac{x_k - x_0}{K} \right)^k \leq e^{4\|B\|K(x_k - x_0)} \quad .$$

Für $D_k$ ergibt sich nun durch Aufsummieren der geometrischen Reihe

$$\|D_k\| \leq q^k\|D_0\| + h\,\frac{q^k - 1}{q-1} \max_{j=0}^{k-1} a_j$$

$$\leq e^{4\|B\|K(x_k - x_0)} \left( \|D_0\| + \frac{1}{4\|B\|K} \max_{j=0}^{k-1} |T_h(x_{j+m})| \right) .$$

Da in $\mathbb{R}^m$ alle Normen äquivalent sind, folgt die behauptete Ungleichung.

Hieraus folgt wie in § 31 sofort

SATZ 33.5: f erfülle eine lokale Lipschitzbedingung. Ist das Mehrschrittverfahren stabil und konsistent (von der Ordnung p), so ist das Verfahren konvergent (von der Ordnung p).

Daß Stabilität notwendig ist für Konvergenz, folgt leicht aus dem Verhalten der Lösungen von Differenzengleichungen:

SATZ 33.6: Ist ein Mehrschrittverfahren konvergent für die AWA $y' = 0$, $y(0) = 0$, so ist es stabil.

BEWEIS: Sei $y$ eine Wurzel von $\rho$ der Vielfachheit $r$. Wir geben Anfangswerte

$$\bar{y}_k = k^{r-1}\lambda^k h \quad , \quad k=0,\ldots,m-1 \quad .$$

Das Verfahren lautet

$$\sum_{\nu=0}^{m} \alpha_\nu y_{k+\nu} = o \ , \quad k=0,1,\ldots \ , \quad y_k = \bar{y}_k \ , \quad k=0,\ldots,m-1 \quad .$$

Nach Satz 33.1 ist

$$y_k = k^{r-1}\lambda^k h \quad , \quad k=0,1,\ldots \quad .$$

Wir lassen nun $h \to 0$ und $k \to \infty$ so streben, daß $x_k = hk = \bar{x} > 0$. Dann muß wegen der vorausgesetzten Konvergenz $y_k \to 0$ streben. Also folgt

$$\lim_{k\to\infty} \left( \frac{\bar{x}}{h} \right)^{r-1} \lambda^{\bar{x}/h} h = 0 \quad .$$

Dies ist nur möglich, wenn $|\lambda| \leq 1$ und $r=1$ für $|\lambda| = 1$.

■

## § 34 Konsistenz und Stabilität bei Mehrschrittverfahren

Vom vorhergehenden Paragraphen ist es klar, daß man nur mit stabilen Verfahren arbeiten kann. Aus Effizienzgründen möchte man Verfahren möglichst hoher Konsistenzordnung verwenden. Unglücklicherweise beschränkt die Forderung nach Stabilität die an und für sich mögliche Konsistenzordnung.

SATZ 34.1: Ein Mehrschrittverfahren ist genau dann konsistent für alle $y \in C^2$, wenn $\rho(1) = 0$, $\rho'(1) - \sigma(1) = 0$. Es ist genau dann konsistent von der Ordnung $p$ für alle $y \in C^{p+1}$, wenn

$$\varphi(\lambda) = \frac{\rho(\lambda)}{\ell n\ \lambda} - \sigma(\lambda)$$

bei $\lambda = 1$ eine Nullstelle der Ordnung $p$ hat.

BEWEIS: Für $y \in C^{p+1}$ ist

$$y(x_{k+\nu}) = y(x_k) + \nu h y'(x_k) + \ldots + \frac{(\nu h)^p}{p!} y^{(p)}(x_k) + \mathcal{O}(h^{p+1})$$

$$y'(x_{k+\nu}) = y'(x_k) + \nu h y''(x_k) + \ldots + \frac{(\nu h)^{p-1}}{(p-1)!} y^{(p)}(x_k) + \mathcal{O}(h^p) .$$

Dies ergibt sich für den lokalen Diskretisierungsfehler

$$T_h(x_{k+m}) = \frac{1}{h} \sum_{\nu=0}^{m} \alpha_\nu y(x_{k+\nu}) - \sum_{\nu=0}^{m} \beta_\nu y'(x_{k+\nu})$$

$$= \frac{1}{h} C_0 y(x_k) + C_1 y'(x_k) + \ldots + h^{p-1} C_p y^{(p)}(x_k) + \mathcal{O}(h^p),$$

$$C_0 = \sum_{\nu=0}^{m} \alpha_\nu \quad ,$$

$$C_1 = \sum_{\nu=0}^{m} \nu\alpha_\nu - \sum_{\nu=0}^{m} \beta_\nu \quad ,$$

$$\vdots$$

$$C_p = \frac{1}{p!} \sum_{\nu=0}^{m} \nu^p \alpha_\nu - \frac{1}{(p-1)!} \sum_{\nu=0}^{m} \nu^{p-1} \beta_\nu \quad .$$

Sei nun

$$\chi(z) = \varphi(e^z) = \frac{1}{z} \sum_{\nu=0}^{m} \alpha_\nu e^{\nu z} - \sum_{\nu=0}^{m} \beta_\nu e^{\nu z} \quad .$$

Die Potenzreihe um $z = 0$ für $e^{\nu z}$ ergibt für $z \to 0$

$$\chi(z) = \frac{1}{z} \sum_{\nu=0}^{m} \alpha_\nu \sum_{\mu=0}^{p} \frac{(\nu z)^\mu}{\mu!} - \sum_{\nu=0}^{m} \beta_\nu \sum_{\mu=0}^{p-1} \frac{(\nu z)^\mu}{\mu!} + \mathcal{O}(z^p)$$

$$= \frac{1}{z} C_0 + C_1 + \ldots + z^{p-1} C_p + \mathcal{O}(z^p) \quad .$$

Nun gilt: $\varphi$ hat p-fache Nullstelle bei $\lambda = 1$

$\Leftrightarrow$ $\chi$ - " - $z = 0$

$\Leftrightarrow$ $C_0 = C_1 = \ldots = C_{p-1} = 0$

$\Leftrightarrow$ $T_h(x_{k+m}) = \mathcal{O}(h^p)$ für alle $y \in C^{p+1}$ .

Damit ist die zweite Aussage des Satzes bewiesen. Die erste folgt ähnlich wegen $C_0 = \rho(1)$, $C_1 = \rho'(1) - \sigma(1)$ .

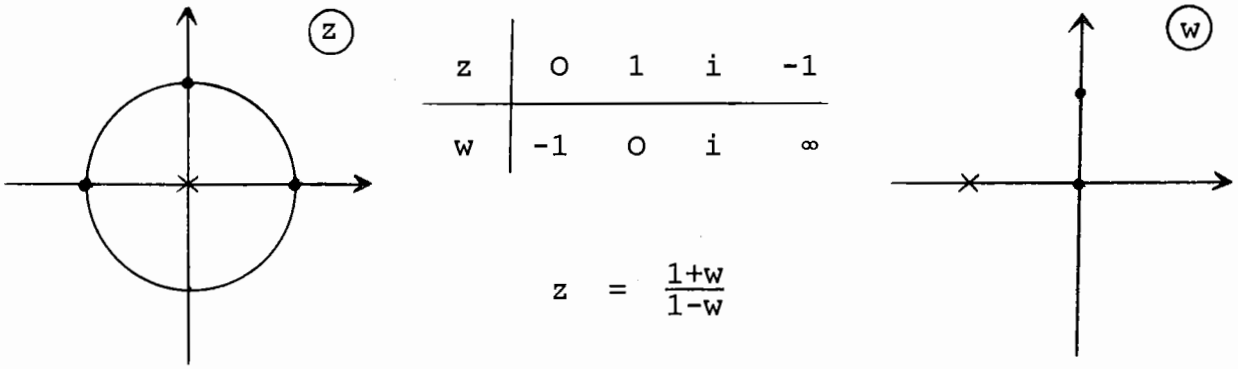
■

Nach dem Satz stellt Konsistenz der Ordnung p p+1 Bedingungen an die 2m+1 (nach Normierung etwa auf $\alpha_m = 1$) Koeffizienten eines m-Schrittverfahrens. Man erwartet also, daß man die Konsistenzordnung 2m erzielen kann. Dies ist auch der Fall, aber leider nutzlos, wie man an dem folgenden Satz sieht.

SATZ 34.2: Ist ein m-Schrittverfahren stabil, so ist seine Konsistenzordnung höchstens m+1 für m ungerade und m+2 für m gerade.

BEWEIS: Zunächst einige Vorbemerkungen.

(i) Die gebrochene lineare Transformation $w = \frac{z-1}{z+1}$ bildet den Einheitskreis der z-Ebene auf die linke Halbebene der w-Ebene ab:




Denn linear gebrochene Abbildungen bilden Kreise auf Kreise ab. Da ein Kreis durch 3 Punkte eindeutig bestimmt ist, geht der Einheitskreis mit den Punkten 1, i, -1 in die imaginäre Achse mit den Punkten O, i, ∞ über. Das Innere des Einheitskreises muß dabei in die linke Halbebene übergehen, weil O in -1 übergeht.

(ii) Die Koeffizienten eines reellen Polynoms, dessen Wurzeln nur Realteile $\leqq$ O haben, haben alle das gleiche Vorzeichen.

Denn ist $r$ ein solches Polynom und sind $x_\mu$ die reellen und $x_\nu \pm iy_\nu$ die konjugiert komplexen Wurzeln, so ist

$$r(z) = a \prod_\mu (z-x_\mu) \prod_\nu ((z-x_\nu)^2 + y_\nu^2)$$

und die Behauptung folgt durch Ausmultiplizieren.

(iii) Die Koeffizienten $C_{2\nu}$ in

$$\frac{z}{\ln \frac{1+z}{1-z}} = C_0 + C_2 z^2 + C_4 z^4 + \ldots$$

sind negativ für $\nu > 0$ (siehe Henrici, S. 233).

Nun zum Beweis des Satzes! Seien $\rho$, $\sigma$ die Polynome eines stabilen konsistenten Verfahrens. Wir setzen

$$r(w) = \left(\frac{1-w}{2}\right)^m \rho\left(\frac{1+w}{1-w}\right), \quad s(w) = \left(\frac{1-w}{2}\right)^m \sigma\left(\frac{1+w}{1-w}\right).$$

Dann hat nach (i) und wegen der Stabilität $r$ bei $w = 0$ eine einfache Nullstelle und sonst nur Nullstellen mit negativem Realteil. Nach (ii) ist r also von der Form

$$r(w) = a_1 w + a_2 w^2 + \ldots + a_m w^m$$

mit $a_1 \neq 0$, und $a_\ell$ hat das Vorzeichen von $a_1$, $\ell=2,\ldots,m$. Sei nun weiter

$$f(w) = \left(\frac{1-w}{2}\right)^m \varphi\left(\frac{1+w}{1-w}\right), \quad \varphi(z) = \frac{\rho(z)}{\ln z} - \sigma(z).$$

Nach Satz 34.1 ist die Ordnung $p$ der Nullstelle von $f$ bei 0 gleich der Konsistenzordnung des Verfahrens. Offenbar ist

$$f(w) = \frac{r(w)}{\ell n \frac{1+w}{1-w}} - s(w)$$

$$= b_0 + b_1 + \ldots + b_{p-1}w^{p-1} + \ldots \quad -s(w) \quad .$$

Da $s$ ein Polynom vom Grade $m$ ist, kann $f$ nur dann eine Nullstelle der Ordnung $p$ bei $0$ haben, wenn

$$b_{m+1} = b_{m+2} = \ldots = b_{p-1} = 0$$

ist. Für $m+1 > p-1$ ist diese Bedingung leer.

Wir berechnen nun die $b_\nu$. Es ist nach (iii)

$$b_0 + b_1 w + \ldots = \frac{w}{\ell n \frac{1+w}{1-w}} \quad \frac{r(w)}{w}$$

$$= (C_0+C_2w^2+C_4w^4+\ldots- )(a_1+a_2w+\ldots+a_m\, w^{m-1})$$

mit $C_{2\nu} < 0$ , $\nu > 0$. Ausmultiplikation und Koeffizientenvergleich für die geraden Potenzen ergibt

$$b_{2\nu} = C_0a_{2\nu+1} + C_2a_{2\nu-1} + \ldots + C_2\, a_1 \quad ,$$

wobei wir $a_\nu = 0$ für $\nu > m$ gesetzt haben. Nun unterscheiden wir zwei Fälle

(a) $m$ ungerade. Wir setzen $2\nu = m+1$ und bekommen

$$b_{m+1} = C_0a_{m+2} + C_2a_m + \ldots + C_{m+1}a_1 \quad .$$

Es ist $a_{m+2} = 0$, $C_{2\nu} < 0$, die $a_\ell$ haben alle das gleiche Vorzeichen, und $a_1 \neq 0$. Also folgt $b_{m+1} \neq 0$, d.h. es muß $p-1 < m+1$ oder $p \leqq m+1$ sein.

(b) m gerade. Wir setzen $2\nu = m+2$ und bekommen

$$b_{m+2} = c_0 a_{m+3} + c_2 a_{m+1} + c_4 a_{m-1} + \ldots + c_{m+2} a_1 \quad .$$

Wie oben folgt $b_{m+2} \neq 0$, d.h. es muß $p-1 < m+2$ oder $p \leq m+2$ sein. ∎

DEFINITION: Ein m-Schrittverfahren heißt optimal, wenn seine Konsistenzordnung m+1 ist für m ungerade und m+2 für m gerade.

BEISPIELE:

1) Die Verfahren von Adams-Moulton und Milne - Simpson haben die Konsistenzordnung m+1 und sind daher für ungerades m optimal.

2) Das Milne-Simpson-Verfahren für m=2, d.h.

$$y_{k+2} - y_k = h(2 f_{k+2} - 2 \nabla f_{k+2} + \frac{1}{3} \nabla^2 f_{k+2})$$

ist identisch zu dem Verfahren für m=3. Es hat also die Konsistenzordnung 3+1 = 4 und ist daher optimal. Dagegen hat die Mittelpunktsregel

$$y_{k+2} - y_k = 2h f_{k+1}$$

nur die Konsistenzordnung 2 und ist also nicht optimal.

## § 35 Extrapolationsverfahren

Wir wollen die dem Romberg-Verfahren zugrunde liegende Idee der Extrapolation auf die Lösung von Differentialgleichungen übertragen. Dazu schreiben wir für den an der Stelle $x$ mit irgendeinem Verfahren mit der Schrittweite $h$ berechneten Näherungswert $y(x,h)$. Gilt nun eine asymptotische Entwicklung der Form

$$(35.1)\quad y(x,h) = y(x) + h^p e_p(x) + \ldots + h^q e_q(x) + \mathcal{O}(h^{q+1})$$

mit von $h$ unabhängigen Funktionen $e_\ell$, so kann man ähnlich wie beim Romberg-Verfahren vorgehen: Man wiederholt die Rechnung mit kleinerer Schrittweite, etwa $\frac{h}{2}$, und hat dann

$$(35.2)\quad y(x,\tfrac{h}{2}) = y(x) + 2^{-p}h^p e_p(x) + \ldots + 2^{-q}h^q e_q(x) + \mathcal{O}(h^{q+1}) .$$

Nun wird der Term der Ordnung $h^p$ eliminiert:

$$2^p y(x,\tfrac{h}{2}) - y(x,h) = (2^p-1)y(x) + (2^{p-1}-1)h^{p+1}e_p(x) + \ldots + (2^{p-q}-1)h^q e_q(x) + \mathcal{O}(h^{q+1}) .$$

In

$$y^1(x,h) = \frac{1}{2^p-1}\,(2^p y(x,\tfrac{h}{2}) - y(x,h)) = y(x,\tfrac{h}{2}) + \frac{1}{2^p-1}\,(y(x,\tfrac{h}{2}) - y(x,h))$$

hat man also eine neue Formel mit der Ordnung $p+1$, und für diese gilt

$$y^1(x,h) = y(x) + h^{p+1}e^1_{p+1}(x) + \ldots + h^q e^1_q(x) + \mathcal{O}(h^{q+1})$$

mit neuen, ebenfalls von $h$ unabhängigen Funktionen $e^1_\ell$ .

Entsprechend konstruiert man Formeln $y^2, y^3, \ldots$ der Ordnungen $p+2, p+3, \ldots$ .

Eine andere Anwendung von (35.1) betrifft die Schätzung des Fehlers. Subtraktion von (35.1), (35.2) ergibt

$$y(x,h)-y(x,\tfrac{h}{2}) = h^p(1-2^{-p})e_p(x) + \mathcal{O}(h^{p+1}),$$

also

$$y(x,\tfrac{h}{2})-y(x) = 2^{-p}h^p e_p(x) + \mathcal{O}(h^{p+1})$$

$$= \frac{1}{2^p-1}\,(y(x,h)-y(x,\tfrac{h}{2})) + \mathcal{O}(h^{p+1}) \quad .$$

Für hinreichend kleine $h$ ist also

$$y(x,\tfrac{h}{2})-y(x) \;\sim\; \frac{1}{2^p-1}\,(y(x,h)-y(x,\tfrac{h}{2})) \quad .$$

Das Bestehen von (35.1) ist für Einschrittverfahren unter allgemeinen Voraussetzungen der Fall (siehe Bulirsch-Stoer II, 7.2.3), für Mehrschrittverfahren im allgemeinen aber nicht.

BEISPIEL: Die Mittelpunktsregel für die AWA $y' = -y$, $y(0) = 1$ lautet

$$y_{k+2} = y_k - 2hy_{k+1} \quad , \quad k = 0,1,\ldots \quad ,$$

und wir nehmen die Startwerte $y_0 = 1$, $y_1 = 1-h$ $(=y(h) + O(h^2))$. Nach Satz 33.1 ist

$$y_k = c_1\,\lambda_1^k + c_2\,\lambda_2^k$$

mit den Wurzeln

$$\lambda_1 = \sqrt{1+h^2}\left(1 - \frac{h}{\sqrt{1+h^2}}\right), \quad \lambda_2 = -\sqrt{1+h^2}\left(1 + \frac{h}{\sqrt{1+h^2}}\right)$$

von $\rho(\lambda) = \lambda^2 + 2h\lambda - 1$ und mit Konstanten $C_1$, $C_2$ mit

$$\begin{aligned} C_1 + C_2 &= 1 \\ C_1\lambda_1 + C_2\lambda_2 &= 1-h \end{aligned} \quad .$$

Sei nun $x > 0$ fest und $hk = x$. Dann ist

$$y(x,h) = C_1(h)\lambda_1(h)^{x/h} + C_2(h)\lambda_2(h)^{x/h} \quad ,$$

wobei wir jetzt die Abhängigkeit von $C_i$, $\lambda_i$ von $h$ explizit gemacht haben. Offenbar sind die Funktionen $\lambda_i(h)$ um $h=0$ in konvergente Potenzreihen nach $h$ entwickelbar. Das gleiche gilt dann auch für $C_i(h)$. Auch $\lambda_1(h)^{x/h}$ läßt eine solche Entwicklung zu, denn es ist für $h \to 0$

$$\begin{aligned} \ell n\ \lambda_1(h)^{x/h} &= \frac{x}{h}\,\ell n\ \lambda_1(h) = \frac{x}{h}\,\ell n\,(1+a_1h + a_2h^2 + \dots) \\ &= \frac{x}{h}\,(b_1h + h_2h^2 + \dots) = x(b_1+b_2h + \dots) \quad . \end{aligned}$$

Für $\lambda_2(h)^{x/h}$ kann dies aber wegen des Faktors $(-1)^{x/h}$ nicht zutreffen: Der Ausdruck oszilliert für $h \to 0$!

Die Herleitung eines Mehrschrittverfahrens mit (35.1) ist daher keineswegs einfach. Das bekannteste Verfahren dieser Art stammt von Gragg: Sei $n$ gerade.

$$y_0 = y(x_0)$$

$$y_1 = y_0 + h\,f(x_0,y_0) \qquad \text{(Euler)}$$

$$y_{k+2}-y_k = 2h\, f(x_{k+1},y_{k+1}), \qquad k=0,\ldots,n-1 \quad \text{(Mittelpunktsregel)}$$

$$y_n = \frac{1}{4}\, y_{n+1} + \frac{1}{2}\, y_n + \frac{1}{4}\, y_{n-1} \qquad \text{(Glättung)} \quad .$$

Man kann dann zeigen: Für hinreichend glattes $y$ gilt

$$y_n = y(x_n) + e_1(x_n)h^2 + e_2(x_n)h^4 + \ldots$$

mit von $h$, $n$ unabhängigen Funktionen $e_\ell$. Dies führt natürlich zu (31.1), wobei sogar nur gerade Potenzen von $h$ auftreten.

Die praktische Anwendung des Gragg-Verfahrens kann im einfachsten Fall etwa folgendermaßen geschehen: Man schreibt ein Unterprogramm

Gragg (x,y,h,tol,f) ,

welches folgendes leistet: Man setzt der Reihe nach

$$n = 2,4,\ldots$$
$$h = \frac{h}{2}\ ,\ \frac{h}{4}\ ,\ldots$$

und berechnet für $y(x+h)$ mit dem Gragg-Verfahren Näherungen

$T_1(h)$, $T_1(\frac{h}{2})$ , ... , und zwar mit

$T_1(h) = y_2$ mit Schrittweite $h$ ,

$T_2(h) = y_4$ mit Schrittweite $h/2$ usw. .

Man erstellt etwa 4 - 6 Spalten des Romberg-Schemas und prüft, ob die Fehlerschätzung in der rechten Spalte zuverlässig ist und $<$ tol ausfällt. Ist dies der Fall, so wird $x \to x+h$ ,

y → y(x+h) gesetzt. h bekommt unter Umständen auch einen neuen Wert: Müssen viele Zeilen des Romberg-Schemas berechnet werden, so wird h verkleinert. Tritt dagegen schon in den ersten Spalten des Romberg-Schemas ein Fehler < tol auf, so wird h vergrößert.

## § 36 Systeme von Differentialgleichungen und Differentialgleichungen höherer Ordnung

Wir betrachten nun die AWA für Systeme von n Differentialgleichungen 1. Ordnung

$$y_1' = f_1(x,y_1,\ldots,y_n) \quad , \quad y_1(x_0) = y_{10} \quad ,$$

$$y_2' = f_2(x,y_1,\ldots,y_n) \quad , \quad y_2(x_0) = y_{20} \quad ,$$

$$\vdots$$

$$y_n' = f_n(x,y_1,\ldots,y_n) \quad , \quad y_n(x_0) = y_{n0} \quad .$$

Führt man die Vektoren

$$y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \quad , \quad f(x,y) = \begin{pmatrix} f_1(x,y_1,\ldots,y_n) \\ \vdots \\ f_n(x,y_1,\ldots,y_n) \end{pmatrix} \quad , \quad y_0 = \begin{pmatrix} y_{10} \\ \vdots \\ y_{n0} \end{pmatrix}$$

ein, so kann man dafür

$$y' = f(x,y) \quad , \quad y(x_0) = y_0 \quad .$$

schreiben. Damit übertragen sich alle Aussagen und Verfahren für skalare Differentialgleichungen unmittelbar auf Systeme. Zum Beispiel lautet die Lipschitzbedingung für Systeme

$$\|f(x,y) - f(x,\bar{y})\| \leq L\|y - \bar{y}\|$$

mit einer Vektornorm $\|\cdot\|$. Das Euler-Verfahren lautet

$$y_{k+1} = y_k + hf(x_k,y_k) \quad , \qquad k = 0,1,\ldots \quad ,$$

wobei jetzt $y_k$ eine Näherung für den Vektor $y(x_k) = (y_1(x_k),\ldots, y_n(x_k))^T$ bedeutet.

Die AWA für Differentialgleichungen n-ter Ordnung lautet

$$y^{(n)} = f(x,y,y',\ldots,y^{(n-1)}) \quad ,$$

$$y^{(i)}(x_0) = y_0^{(i)} \quad , \quad i = 0,1,\ldots,n-1 \; .$$

Setzt man

$$y_1 = y \quad , \; y_2 = y',\ldots,y_n = y^{(n-1)} \quad ,$$

so entsteht ein AWA für ein System 1. Ordnung:

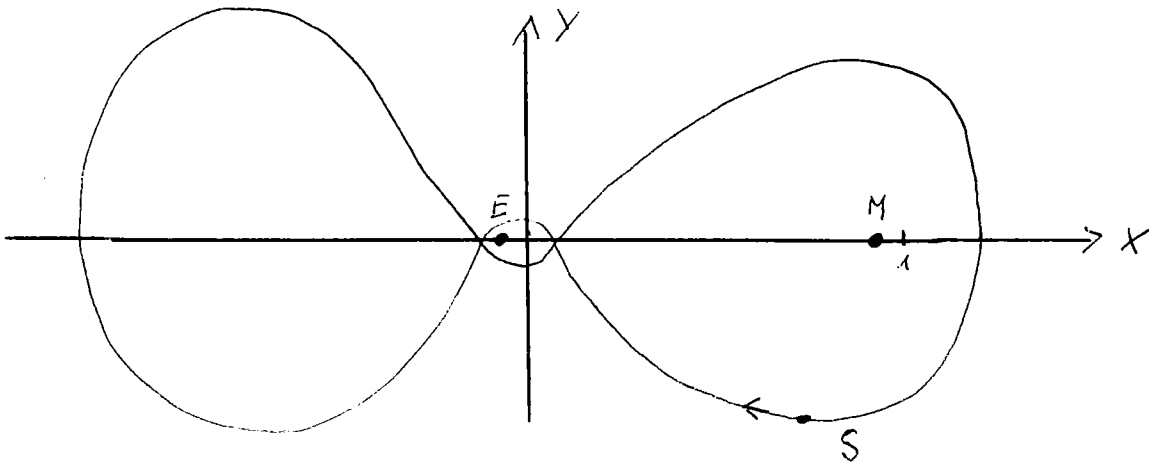
$$\begin{aligned}
y_1' &= y_2 \; , & y_1(x_0) &= y_0^{(0)} \; , \\
y_2' &= y_3 \; , & y_2(x_0) &= y_0' \; , \\
&\vdots & &\vdots \\
y_{n-1}' &= y_n \; , & y_{n-1}(x_0) &= y_0^{(n-2)} , \\
y_n' &= f(x,y_1,y_2,\ldots,y_n) \; , & y_n(x_0) &= y_0^{(n-1)} .
\end{aligned}$$

Entsprechend kann man Systeme höherer Ordnung auf Systeme 1. Ordnung zurückführen, indem man die Ableitungen als neue Unbekannte einführt. Als Beispiel betrachten wir folgendes "Restringierte 3-Körper-Problem" (siehe etwa Siegel, Vorlesungen über Himmelsmechanik, S. 105, Springer 1956), welches die Bewegung eines Satelliten im Schwerefeld von

von Erde und Mond in einer gemeinsamen Ebene beschreibt. Rechnet man im Schwerpunktsystem von Erde und Mond, so hat man folgende Konstellation:

| Körper | Masse | Koordination |
|---|---|---|
| Mond (M) | $\mu$ | $(1 - \mu , 0)$ |
| Erde (E) | $1 - \mu$ | $(- \mu , 0)$ |
| Satellit (S) | 0 | $(x(t),y(t))$ |

Hier ist $\mu$ $(0 < \mu < 1)$ die relative Mondmasse $\mu = 1/82.45$. Die Bewegung erfolgt in der x-y-Ebene, $t$ ist die Zeit:




$x,y$ erfüllen folgendes System 2. Ordnung:

$$\ddot{x} = 2\dot{y} + x + F_x(x,y)$$

$$\ddot{y} = -2\dot{x} + y + F_y(x,y)$$

$$\dot{x} = \frac{d}{dt} x \quad \text{usw.}$$

$$F(x,y) = \frac{1 - \mu}{((x+\mu)^2+y^2)^{1/2}} + \frac{\mu}{((x+\mu-1)^2+y^2)^{1/2}}$$

Dazu kommen noch Anfangswerte für $x,\dot{x},y,\dot{y}$. Setzen wir

$$y_1 = x \quad , \quad y_2 = y \quad , \quad y_3 = \dot{x} \quad , \quad y_4 = \dot{y} \quad ,$$

so bekommt man das System 1. Ordnung

$$\begin{aligned}
\dot{y}_1 &= y_3 \\
\dot{y}_2 &= y_4 \\
\dot{y}_3 &= 2y_4 + y_1 + F_x(y_1,y_2) \\
\dot{y}_4 &= -2y_3 + y_2 + F_y(y_1,y_2)
\end{aligned}$$

$$y_1(0) = x(0),\ y_2(0) = y(0),\ y_3(0) = \dot{x}(0),\ y_4(0) = \dot{y}(0) \quad .$$

## § 37 Randwertprobleme gewöhnlicher Differentialgleichungen

Bisher haben wir die Eindeutigkeit der Lösung des Systems $y' = f(x,y)$ dadurch erzwungen, daß wir $y(x_0)$ vorgeschrieben haben. Allgemeiner kann man Nebenbedingungen an verschiedenen Punkten stellen, etwa in der Form

$$y' = f(x,y) \quad , \qquad a \leq x \leq b$$

$$g(y(a),y(b)) = 0 \qquad . \tag{37.1}$$

Man spricht dann von einer Randwertaufgabe.

Ein besonders einfacher Fall ist die lineare Randwertaufgabe 2. Ordnung

$$y'' + p(x)y' + q(x) = f(x) \quad , \qquad a \leq x \leq b$$

$$y(a) = \alpha \ , \quad y(b) = \beta \quad , \tag{37.2}$$

welche man natürlich in (37.1) überführen kann.

Eine einfache und effiziente Methode zur Lösung von (37.2) ist das Differenzen-Verfahren. Sei $x_i = a + hi$, $h = (b-a)/n$, $i=0,\ldots,n$. In jedem Punkt $x_i$ ersetzen wir die Ableitungen durch geeignete Differenzenquotienten. Für $y \in C^4$ gilt

$$y''(x_i) = \frac{y(x_{i+1})-2y(x_i)+y(x_{i-1})}{h^2} + \mathcal{O}(h^2) \quad ,$$

$$y'(x_i) = \frac{y(x_{i+1}) - y(x_{i-1})}{2h} + \mathcal{O}(h^2) \quad ,$$

vgl. § 27. Es ist daher naheliegend, Näherungen $y_i$ für $y(x_i)$ als Lösung des Systems

$$\frac{y_{i+1}-2y_i+y_{i-1}}{h^2} + p(x_i)\,\frac{y_{i+1}-y_{i-1}}{2h} + q(x_i)y_i = f(x_i),\quad i=1,\ldots,n-1, \tag{37.3}$$

$$y_0 = \alpha\ , \qquad y_n = \beta$$

zu berechnen. Dies ist ein lineares System mit Tridiagonalmatrix.

SATZ 37.1: (37.2) sei eindeutig lösbar. Dann gibt es Konstanten $h_0$, $C > 0$, so daß für $h < h_0$ auch (37.3) eindeutig lösbar ist, und es gilt für $y \in C^4$

$$\max_{i=0}^{n} |y(x_i) - y_i| \leq Ch^2 \quad .$$

BEWEIS: Sei

$$Ly = y'' + p(x)y' + q(x)y,$$

$$L_h y_i = \frac{y_{i+1}-2y_i+y_{i-1}}{h^2} + p(x_i)\,\frac{y_{i+1}-y_{i-1}}{2h} + q(x_i)y_i \quad , \quad i=1,\ldots,n-1\ .$$

Dann ist für $y \in C^4$ der "lokale Diskretisierungsfehler"

$$L_h y(x_i) - Ly(x_i) = T_h(x_i) = \mathcal{O}(h^2) \quad .$$

Wegen $Ly = f$ folgt

$$L_h y(x_i) = f(x_i) + T_h(x_i)\ ,$$

und das Verfahren lautet

$$L_h y_i = f(x_i) \quad .$$

Durch Subtraktion findet man für die Fehler $d_i = y(x_i) - y_i$

$$L_h d_i = T_h(x_i) \quad , \quad i = 1,\ldots,n-1 \quad .$$

Mit den Vektoren und Matrizen

$$d_h = \begin{pmatrix} d_1 \\ \vdots \\ d_{n-1} \end{pmatrix} , \; T_h = \begin{pmatrix} T_h(x_1) \\ \vdots \\ T_h(x_{n-1}) \end{pmatrix} , \; A_h = \begin{pmatrix} a_1 & b_1 & & & \\ c_2 & a_2 & b_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & b_{n-2} \\ & & & c_{n-1} & a_{n-1} \end{pmatrix} ,$$

$$a_i = -\frac{2}{h^2} + q(x_i), \quad b_i = \frac{1}{h^2} + \frac{p(x_i)}{2h} \quad , \quad c_i = \frac{1}{h^2} - \frac{p(x_i)}{2h}$$

schreibt sich dies

$$A_h \, d_h = T_h \quad .$$

Ist $A_h$ invertierbar, so folgt

$$\|d_h\|_\infty \leq \|A_h^{-1}\|_\infty \, \|T_h\|_\infty \quad .$$

Kann man also für $h < h_0$ mit geeigneten $h_0 > 0$

(37.4) $\|A_h^{-1}\|_\infty \leq C$ (unabhängig von h)

zeigen, so folgt

$$\|d_h\|_\infty \leq C\|T_h\|_\infty = \mathcal{O}(h^2)$$

und der Satz ist bewiesen. (37.4) ist die Stabilitätsbedingung. Sie ist nicht leicht zu beweisen. Wir verweisen dazu auf die Literatur, insbesondere auf Stetter.

■

Allgemeiner kann man nichtlineare Randwertaufgaben

$$y'' = f(x,y,y') \quad , \quad a \leq x \leq b$$

(37.5)

$$y(a) = \alpha \quad , \quad y(b) = \beta$$

betrachten. Zur Lösung kann man ebenso vorgehen wie bei der linearen Aufgabe (37.2) und erhält dann anstelle des linearen Gleichungssystems (37.3) ein nichtlineares Gleichungssystem, das man etwa mit Hilfe des Newton-Verfahrens (vgl. § 15) lösen kann. Eine Alternative ist das "Schießverfahren": Man löst die AWA

$$y'' = f(x,y,y') \quad ,$$

$$y(a) = \alpha \quad , \quad y'(a) = s$$

und versucht $s$ so zu bestimmen, daß $y(b) = \beta$. Bezeichnet man die Lösung der AWA mit $y(x,s)$, so löst man also die nichtlineare Gleichung $y(b,s) = \beta$. Dazu kann man irgend eine der Methoden aus Teil III verwenden. Die Berechnung von $y(b,s)$ erfolgt dabei durch irgend ein Verfahren zur Lösung der AWA.

<u>BEISPIEL:</u> $y'' = \frac{3}{2} y^2$ , $y(0) = 4$, $y(1) = 1$. Wir schreiben die AWA mit den Anfangswerten $y(0) = 4$, $y'(0) = s$ als System 1. Ordnung und lösen dieses für verschiedene $s$ mit Hilfe des Gragg'schen Verfahrens, wie es in der IMSL - Routine DREBS implementiert ist. Die $s$-Werte werden nach dem Prinzip der Intervallhalbierung (vgl. § 13) gewählt:

| s | y(1,s) |
|---|---|
| - 5 | 12.057576 |
| -10 | - 2.400837 |
| - 7.5 | 2.223303 |
| - 8.725 | - 0.477253 |
| - 7.80625 | 1.452413 |
| - 7.959375 | 1.092695 |
| - 8.0359375 | 0.918937 |
| - 7.99765625 | 1.005317 |

Im Rahmen einer 3-stelligen Rechnung findet man also $s = -8.00$; die dazugehörige Lösung $y(x,s)$ ist eine Näherung für die gesuchte Lösung der Randwertaufgabe. Man findet übrigens noch eine weitere Lösung mit $s \sim -35.8$.

Es ist vielleicht interessant, diese bequeme Lösung von (37.5) mit der analytischen Methode zu vergleichen. Für die einfache Gleichung $y'' = f(y)$ findet man der Reihe nach formal

$$y'y'' = y'f(y) \quad ,$$

$$\frac{1}{2}\frac{d}{dx}y'^2 = \frac{d}{dx}F(y) \quad , \qquad F(y) = \int^{y} f(z)\,dz \quad ,$$

$$\frac{1}{2}y'^2 = F(y) + c \quad ,$$

$$y' = \pm\sqrt{2F(y)+c} \quad ,$$

$$\frac{dy}{\pm\sqrt{2F(y)+c}} = dx \quad ,$$

$$\pm\int_{\alpha}^{y(x)} \frac{dz}{\sqrt{2F(z)+c}} = x - a \quad .$$

Diese Gleichung muß man nach $y(x)$ auflösen und dann $c$ so bestimmen, daß $y(b) = \beta$ wird. Man sieht, daß sogar in diesem einfachen Fall die analytische Lösung viel komplizierter ist als das direkte numerische Verfahren.

# NUMERIK PARTIELLER DIFFERENTIALGLEICHUNGEN

## § 38 Anfangswertaufgaben partieller Differentialgleichungen

Treten in einer Differentialgleichung Ableitungen nach mehr als einer Variablen auf, so spricht man von partieller Differentialgleichung. Bei den Anfangswertaufgaben spielt eine dieser Variablen die Rolle der Zeit. Wie bezeichnen sie daher mit $t$ .

Wir betrachten die partielle Differentialgleichung

$$u_t = Au \qquad \text{in} \quad [a,b] \times [0,\infty)$$

für die vektorwertige Funktion $u = (u_1,\ldots,u_m)^T$, $u_i = u_i(x,t)$. $A$ ist der Differentialausdruck r-ter Ordnung

$$Au = \sum_{\rho=0}^{r} A_\rho(x) \frac{\partial^\rho}{\partial x^\rho} u$$

der Ordnung $r$ mit (m,m)-Matrizen $A_\rho(x)$. Zur eindeutigen Festlegung von $u$ gehört noch eine Anfangsbedingung

$$u(x,0) = u_0(x) \quad , \qquad a \leq x \leq b$$

und unter Umständen, in Abhängigkeit von $A$, Randbedingungen am Rande von $[a,b]$, also Bedingungen für die Funktion

$$u(a,t) \quad , \quad u(b,t) \quad , \qquad t \geq 0 \qquad .$$

Dies ist die Anfangswertaufgabe (AWA) oder auch Anfangsrandwertaufgabe.

BEISPIELE: 1) $u_t = u_x$ , $x \in \mathbb{R}^1$ .

Die Gleichung verlangt, daß $u$ entlang der Geraden $t+x = C$ konstant ist. Also ist

$$u(x,t) = u(x+t,0) = u_0(x+t)$$

die Lösung der AWA.

2) $u_t = Du_{xx}$ , $0 \leq x \leq 1$. Dies ist die Wärmeleitungs- oder Diffusionsgleichung. Sie beschreibt die Temperatur $u(x,t)$ eines Stabes der Länge 1 an der Stelle $x$ zur Zeit $t$; $u_0(x)$ ist demgemäß die Temperatur zur Zeit 0. Als Randbedingungen kommen z.B. in Frage

$$u(0,t),\ u(1,t) = 0 \qquad \text{(Enden gekühlt)}$$

$$u_x(0,t),\ u_x(1,t) = 0 \qquad \text{(Enden wärmeisoliert)} .$$

Die exakte Lösung für die ersten Randbedingungen ist

$$u(x,t) = \sum_{\ell=1}^{\infty} \hat{u}_\ell \sin(\ell\pi x)\, e^{-\pi^2\ell^2 t^2 D} ,$$

$$\hat{u}_\ell = 2 \int_0^1 u_0(x) \sin \ell\pi x \, dx ,$$

vgl. § 23.

3) $u_{tt} = c^2 u_{xx}$ , $0 \leq x \leq 1$. Dies ist die Wellengleichung. Sie beschreibt die Auslenkung $u(x,t)$ zur Zeit $t$ an der Stelle $x$ einer schwingenden Saite der Länge 1. Zur eindeutigen Festlegung von $u$ braucht man demgemäß die Auslenkung zur Zeit 0 sowie die Geschwindigkeit zur Zeit 0, also

$$u(x,0) = u_0(x) \quad , \qquad u_t(x,0) = u_1(x) \qquad .$$

Als Randbedingung tritt etwa auf

$$u(0,t) = u(1,t) = 0 \quad \text{(Enden fest eingespannt)} \; .$$

Die exakte Lösung unter diesen Randbedingungen ist

$$u(x,t) = \sum_{\ell=0}^{\infty} (\hat{u}_{0\ell} \cos(c\ell\pi t) + \hat{u}_{1\ell} \sin(c\ell\pi t)) \sin \ell\pi x \; ,$$

$$\hat{u}_0 = 2 \int_0^1 u_0(x) \sin(\ell\pi x)dx \quad ,$$

$$\hat{u}_1 = \frac{2}{c\ell\pi} \int_0^1 u_1(x) \sin(\ell\pi x)dx \; , \quad \ell > 0 \qquad .$$

Man kann dieses Problem in unser Schema einordnen, wenn man $v_1 = u_x$ , $v_2 = u_t$ setzt. Man erhält dann das System

$$\begin{pmatrix} v_1 \\ v_2 \end{pmatrix}_t = \begin{pmatrix} 0 & 1 \\ c^2 & 0 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}_x \quad , \quad \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} (x,0) = \begin{pmatrix} u_0(x) \\ u_1(x) \end{pmatrix} \quad .$$

4) Allgemeiner betrachten wir das hyperbolische System 1. Ordnung

$$u_t + Au_x = 0 \quad , \qquad u = \begin{pmatrix} u_1 \\ \vdots \\ u_m \end{pmatrix}$$

mit einer konstanten (m,m)-Matrix A, welche m verschieden reelle Eigenwerte $\lambda_1,\ldots,\lambda_m$ hat. Dann ist $A = Y^{-1}JY$ mit

$$J = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_m \end{pmatrix}, \quad Y = \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix}, \quad y_i(A - \lambda_i I) = 0 \quad .$$

Mit $v = Yu$ entsteht

$$v_t + Jv_x = 0 \quad .$$

Dies ist ein zerfallendes System von $m$ Differentialgleichungen

$$\frac{\partial}{\partial t} v_i + \lambda_i \frac{\partial}{\partial x} v_i = 0 \quad .$$

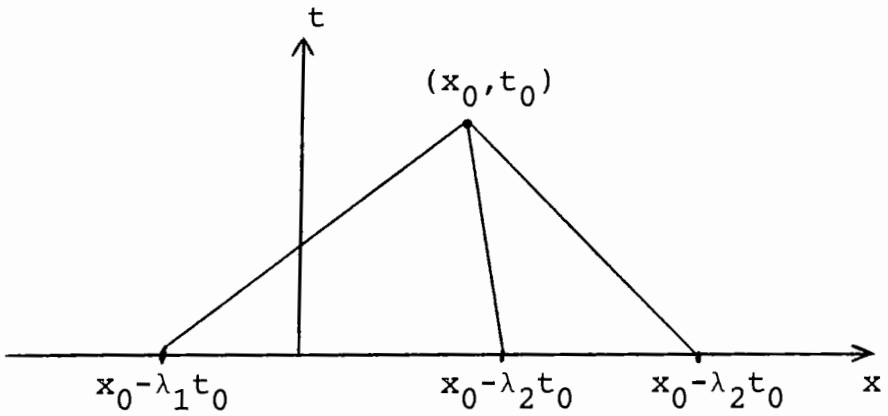
Entlang der Geraden $x = \lambda_i t + C$ ist $v_i$ konstant, denn es ist

$$\frac{d}{dt} v_i(x,t) = \frac{\partial}{\partial t} v_i(x,t) + \lambda_i \frac{\partial}{\partial x} v_i(x,t) = 0 \quad .$$

Die Lösung der Anfangswertaufgabe ist nun wie in Beispiel 1 möglich: Für jeden Punkt $(x_0,t_0)$ der x-t-Ebene bestimmt man die Gerade $x = \lambda_i t_0 + C$, welche durch diesen Punkt geht (es ist diejenige mit $x_0 = \lambda_i t_0 + C$) und sucht deren Schnittpunkt mit $t = 0$ (dies ist in $x = C = x_0 - \lambda_i t_0$). Dann ist

$$v_i(x_0,t_0) = v_i(x_0 - \lambda_i t_0, 0)$$

und dies ist aus der Anfangsbedingung bekannt.

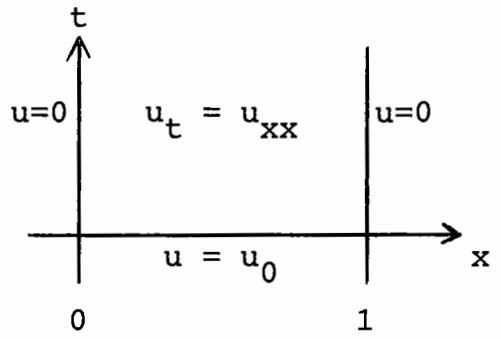


Die Geraden $x = \lambda_i t + c$ nennt man Charakteristiken. Man sieht, daß $u(x_0,t_0)$ nur von den Anfangswerten $u_0(x_0 - \lambda_i t_0)$, $i=1,\ldots,m$ abhängt. Man nennt daher das Intervall $[\mathrm{Min}(x_0 - \lambda_i t_0), \mathrm{Max}(x_0 - \lambda_i t_o)]$ das Abhängigkeitsgebiet von $(x_0,t_0)$.

Im Beispiel haben wir $A = -\begin{pmatrix} 0 & 1 \\ c^2 & 0 \end{pmatrix}$ und damit $\lambda_{1,2} = \pm c$. Die Charakteristiken sind also Geraden mit der Steigung $\pm \frac{1}{c}$. Das Abhängigkeitsgebiet von $(x_0,t_0)$ ist $[x_0 - ct_0, x_0 + ct_0]$. Eine Störung, die zur Zeit 0 bei $x_1$ ist, hat also zur Zeit $t_0 = (x_1-x_0)/c$ den Punkt $x_0$ erreicht. $c$ hat also die Bedeutung einer Ausbreitungsgeschwindigkeit.

## § 39 Einfachste Differenzenverfahren

Wir beginnen mit der AWA für die Wärmeleitungsgleichung.




Wir führen ein Gitter

$$t_\ell = \ell\Delta t \quad , \quad \ell = 0,1,\ldots \quad , \; x_k = kh \; , \; k=0,\ldots,n \; , \; h = \frac{1}{n}$$

ein und suchen für $u(x_k,t_\ell)$ eine Näherung $u_{k,\ell}$, welche die der Differentialgleichung analoge Differenzengleichung

$$\frac{1}{\Delta t}\,(u_{k,\ell+1} - u_{k,\ell}) = \frac{1}{h^2}\,(u_{k+1,\ell} - 2u_{k,\ell} + u_{k-1,\ell}) \quad ,$$

$$k = 1,\ldots,n-1 \quad , \quad \ell = 0,1,\ldots$$

erfüllt. Dazu kommen noch die Anfangs- und Randbedingungen

$$u_{k0} = u_0(x_k) \quad , \quad k = 0,\ldots,n$$

$$u_{0\ell} = u_{n\ell} = 0 \quad , \quad \ell = 1,2,\ldots \quad .$$

Die Differenzengleichungen können nach $u_{k,\ell+1}$ aufgelöst werden. Mit $\lambda = \Delta t/h^2$ gilt

$$u_{k,\ell+1} = \lambda(u_{k+1,\ell} + u_{k-1,\ell}) + (1 - 2\lambda)u_{k,\ell} \quad .$$

Sind also die Werte für die Zeit $t_\ell$ bekannt, so kann man sie für die Zeit $t_{\ell+1}$ berechnen. Für $t_0$ sind sie durch die Anfangsbedingungen gegeben. Die Rechenvorschrift kann durch folgenden Differenzenstern beschrieben werden:

| | | | | |
|---|---|---|---|---|
| $\ell+1$ | | 0 | | 0 neu zu berechnen |
| $\ell$ | . | . | . | . schon berechnet |
| | $\lambda$ | $1-2\lambda$ | $\lambda$ | |
| | $k-1$ | $k$ | $k+1$ | |

Als Beispiel führen wir die Rechnung durch für die Anfangswerte

$$u_{k,0} = \begin{cases} 1 & , \quad k = K, \; K+1 \\ 0 & , \quad \text{sonst} \end{cases} .$$

Dies entspricht einem Stab, der zur Zeit 0 in $[x_K, x_{K+1}]$ erhitzt und sonst überall kalt ist. Die Rechnung muß also die zeitliche Entwicklung eines solchen "hot spot" zeigen.

(a) $\lambda = \frac{1}{2}$ , d.h. $u_{k,\ell+1} = \frac{1}{2}(u_{k+1,\ell} + u_{k-1,\ell})$ .

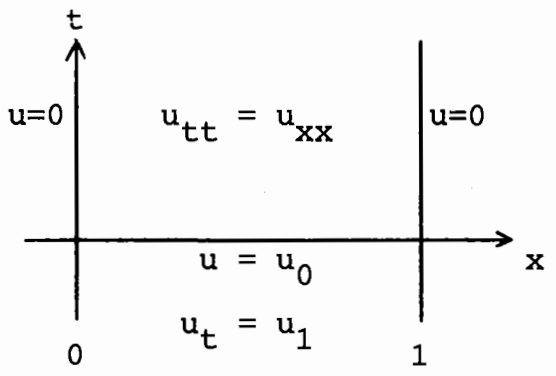
| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $\ell = 3$ | $\frac{1}{8}$ | $\frac{1}{8}$ | $\frac{3}{8}$ | $\frac{3}{8}$ | $\frac{3}{8}$ | $\frac{3}{8}$ | $\frac{1}{8}$ | $\frac{1}{8}$ |
| $\ell = 2$ | | $\frac{1}{4}$ | $\frac{1}{4}$ | $\frac{1}{2}$ | $\frac{1}{2}$ | $\frac{1}{4}$ | $\frac{1}{4}$ | |
| $\ell = 1$ | | | $\frac{1}{2}$ | $\frac{1}{2}$ | $\frac{1}{2}$ | $\frac{1}{2}$ | | |
| $\ell = 0$ | | | | 1 | 1 | | | |
| | | | | K | K+1 | | | |

(b) $\lambda = 1$ , d.h. $u_{k,\ell+1} = u_{k+1,\ell} + u_{k-1,\ell} - u_{k,\ell}$

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $\ell = 3$ | 1 | -2 | 3 | -1 | -1 | 3 | -2 | 1 |
| $\ell = 2$ | | 1 | -1 | 1 | 1 | -1 | 1 | |
| $\ell = 1$ | | | 1 | 0 | 0 | 1 | | |
| $\ell = 0$ | | | | 1 | 1 | | | |
| | | | | K | K+1 | | | |

Während (a) plausibel erscheint, ist (b) offenbar Unsinn. Wir sehen, daß der Erfolg der Rechnung ganz entscheidend von $\lambda$ abhängt.

Als weiteres Beispiel betrachten wir




Die Differentialgleichung wird im Punkt $(x_k, t_\ell)$ durch die Differenzengleichung

$$\frac{1}{(\Delta t)^2}\,(u_{k,\ell+1} - 2u_{k,\ell} + u_{k,\ell-1}) = \frac{1}{(\Delta x)^2}\,(u_{k+1,\ell} - 2u_{k,\ell} + u_{k-1,\ell})$$

ersetzt. Der Fehler dieser Diskretisierung ist $\mathcal{O}((\Delta t)^2 + (\Delta x)^2)$. Um diese Fehlerordnung auch bei der Diskretisierung von $u_t = u_1$ zu haben, führt man ein Zeitniveau $t_{-1}$ ein und kann dann

$$\frac{1}{\Delta t}\,(u_{k,1} - u_{k,-1}) = u_1(x_k) \quad ,$$

$$u_{k,0} = u_0(x_k)$$

setzen. Die Differenzengleichung wird dann für $\ell = 0,1,\ldots$ benutzt. Man kann sie nach $u_{k,\ell+1}$ auflösen und erhält

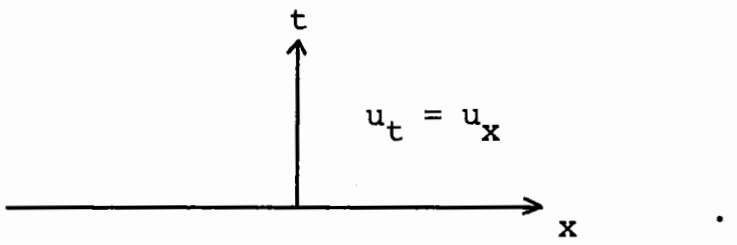
$$u_{k,\ell+1} = (u_{k+1,\ell} + u_{k-1,\ell}) + 2(1 - \lambda)u_{k,\ell} - u_{k,\ell-1} \quad .$$

Dies entspricht dem Differenzenstern

$$\begin{array}{ccccc} & & \circ & & \\ & & & & \\ \cdot & & \cdot & & \cdot \\ \lambda & & 2(1-\lambda) & & \lambda \\ & & & & \\ & & \cdot\,-1 & & \end{array}$$

Das Zeitniveau $-1$ wird in der Gleichung für $\ell = 0$ durch die Anfangsbedingung eliminiert, die entstehende Gleichung kann nach $u_{k,1}$ aufgelöst werden.

Schließlich betrachten wir noch




.

Es sind hier drei Differenzenverfahren gleichermaßen natürlich:

(a) $\frac{1}{\Delta t}(u_{k,\ell+1} - u_{k,\ell}) = \frac{1}{h}(u_{k,\ell} - u_{k-1,\ell})$

(b) $\frac{1}{\Delta t}(u_{k,\ell+1} - u_{k,\ell}) = \frac{1}{h}(u_{k+1,\ell} - u_{k,\ell})$

(c) $\frac{1}{\Delta t}(u_{k,\ell+1} - u_{k,\ell}) = \frac{1}{2h}(u_{k+1,\ell} - u_{k-1,\ell})$

Auflösen nach $u_{k,\ell+1}$ ergibt mit $\lambda = \Delta t/\Delta x$

(a) $u_{k,\ell+1} = (1+\lambda)u_{k,\ell} - \lambda u_{k-1,\ell}$

(b) $u_{k,\ell+1} = (1-\lambda)u_{k,\ell} + \lambda u_{k+1,\ell}$

(c) $u_{k,\ell+1} = u_{k,\ell} + \frac{\lambda}{2}(u_{k+1,\ell} - u_{k-1,\ell})$ .

Wir werden sehen, daß sich diese Verfahren vollkommen unterschiedlich verhalten.

## § 40 Stabilität

Wir gehen aus von der AWA (vgl. § 38)

$$u_t = Au \quad , \quad u(x,0) = u_0(x) \quad , \quad a \leq x \leq b \quad .$$

A sei ein Differentialoperator mit konstanten Koeffizienten, d.h. $A_\rho$ hängt nicht von $x$ ab. Dazu kommen noch, wie in § 38 besprochen, Randbedingungen bei $x = a,b$.

Auf dem Gitter $(t_\ell, x_k)$ betrachten wir das Differenzenverfahren

$$\frac{1}{\Delta t}\,(u_{k,\ell+1} - u_{k,\ell}) = \sum_\nu B_\nu(h) u_{k+\nu,\ell} \quad .$$

Wir führen die Vektoren und Matrizen

$$U_\ell = \begin{pmatrix} u_{0,\ell} \\ \cdot \\ \cdot \\ \cdot \\ u_{n,\ell} \end{pmatrix}, \quad C(\Delta t) = I + \Delta t \begin{pmatrix} B_0 & B_1 & \cdots & & \\ B_{-1} & B_0 & & B_1 & \cdots \\ & & \cdots & B_{-1} & B_0 \end{pmatrix}(h), \quad h = g(\Delta t)$$

ein. Es entsteht

$$U_{\ell+1} = C(\Delta t) U_\ell \quad .$$

DEFINITION 40.1: Das Verfahren heißt stabil, wenn es für alle $T > 0$ eine Konstante $M(T)$ gibt, so daß für $\ell\Delta t \leq T$

$$\| (C(\Delta t))^\ell \|_\infty \leq M(T) \quad .$$

BEISPIELE:

1) Für das einfachste Differenzenverfahren bei der Wärmeleitungsgleichung ist

$$C(\Delta t) = \begin{pmatrix} 1-2\lambda & & & & \\ \lambda & 1-2\lambda & \lambda & & \\ & \ddots & \ddots & \ddots & \\ & & \lambda & 1-2\lambda & \end{pmatrix}, \quad \|C(\Delta t)\|_\infty = |1-2\lambda| + 2\lambda \leqq 1 \Leftrightarrow \lambda \leqq 1/2.$$

Das Verfahren ist also für $\lambda \leqq 1/2$ stabil.

2) Für die Verfahren (a), (b), (c) für $u_t = u_x$ gilt der Reihe nach

$$\|C(\Delta t)\|_\infty = 1 + 2\lambda \quad , \quad |1 - \lambda| + \lambda \quad , \quad 1 + |\lambda| \quad .$$

Es ist also (b) stabil für $\lambda \leqq 1$.

Setzen wir für ganzes $m$

$$u_{k,\ell} = e^{ihkm}\, c_\ell \quad ,$$

so wird

$$\begin{aligned} u_{k,\ell+1} &= u_{k,\ell} + \Delta t \sum_\nu B_\nu(h) u_{k+\nu,\ell} \\ &= e^{ihkm} \left( I + \Delta t \sum_\nu B_\nu(h) e^{ih\nu m} \right) c_\ell \\ &= e^{ihkm}\, c_{\ell+1} \quad , \quad c_{\ell+1} = G(\Delta t, m)\, c_\ell \quad . \end{aligned}$$

Die Matrix

$$G(\Delta t, m) = I + \Delta t \sum_\nu B_\nu(h) e^{ih\nu m}$$

heißt Amplifikationsmatrix des Verfahrens.

BEISPIELE:

1) Wir bestimmen die Amplifikationsmatrix (in diesem Fall besser Amplifikationsfaktor) des einfachsten Differenzenverfahrens für die Wärmeleitungsgleichung:

$$u_{k,\ell+1} = (\lambda(e^{i(k+1)hm} + e^{-i(k+1)hm}) + (1-2\lambda)e^{ikhm})c_\ell$$

$$= (\lambda(e^{ihm} + e^{-ihm}) + 1 - 2\lambda)e^{ikhm}\, c_\ell$$

Klammerung ?

$$c_{\ell+1} = (2\lambda(\cos hm - 1) + 1)c_\ell \quad .$$

Also ist $G(\Delta t, m) = 2\lambda(\cos hm - 1) + 1$.

2) In Beispiel 3) aus § 38 hatten wir die Wellengleichung in das System ($c = 1$)

$$v_t = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} v_x \quad , \quad v(x,0) = v_0(x)$$

umgeschrieben. Für dieses wählen wir die Diskretisierung

$$(40.1)\quad \frac{1}{\Delta t}(v_{k,\ell+1} - v_{k,\ell}) = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \frac{1}{h}(v_{k+1,\ell} - v_{k,\ell}) + \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \frac{1}{h}(v_{k,\ell+1} - v_{k-1,\ell+1}) \, .$$

Sei $u_{k,\ell}$ nach dem einfachsten Differenzenverfahren aus § 39 berechnet, und sei

$$v^1_{k,\ell} = \frac{1}{\Delta t}(u_{k,\ell} - u_{k,\ell-1}), \quad v^2_{k,\ell} = \frac{1}{h}(u_{k,\ell} - u_{k-1,\ell}) \quad .$$

Dann ist

$$\frac{1}{\Delta t}\,(v^1_{k,\ell-1}-v^1_{k,\ell}) = \frac{1}{(\Delta t)^2}\,(u_{k,\ell+1} - 2u_{k,\ell} + u_{k,\ell-1})$$

$$= \frac{1}{h^2}\,(u_{k+1,\ell} - 2u_{k,\ell} + u_{k-1,\ell})$$

$$= \frac{1}{h}\,(v^2_{k+1,\ell} - v^2_{k,\ell}) \quad ,$$

$$\frac{1}{\Delta t}\,(v^2_{k,\ell+1}-v^2_{k,\ell}) = \frac{1}{h\Delta t}\,(u_{k,\ell+1} - u_{k-1,\ell+1} - u_{k,\ell} + u_{k-1,\ell})$$

$$= \frac{1}{h}\,(v^1_{k,\ell+1} - v^1_{k-1,\ell+1}) \quad .$$

Dies bedeutet, daß $v_{k,\ell} = (v^1_{k,\ell}, v^2_{k,\ell})^T$ gerade (40.1) löst.

Mit $\lambda = \Delta t/\Delta x$ schreibt sich (40.1)

$$v_{k,\ell+1} = v_{k,\ell} + \lambda \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} (v_{k+1,\ell} - v_{k,\ell}) + \lambda \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} (v_{k,\ell+1} - v_{k-1,\ell+1}).$$

Zur Berechnung der Amplifikationsmatrix setzen wir $v_{k,\ell} = e^{ihmk} c_\ell$.
Es folgt

$$e^{ihkm}\, c_{\ell+1} = e^{ihkm}\, c_\ell + \lambda \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} (e^{ihm} - 1) e^{ihkm} c_\ell$$

$$+ \lambda \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} (1 - e^{-ihm}) e^{ihkm} c_{\ell+1} \quad .$$

Auflösen nach $c_{\ell+1}$ ergibt mit $a = \lambda(e^{ihm} - 1)$

$$c_{\ell+1} = \begin{pmatrix} 1 & 0 \\ \bar{a} & 1 \end{pmatrix}^{-1} \begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & a \\ -\bar{a} & 1-|a|^2 \end{pmatrix} c_\ell$$

Damit haben wir die Amplifikationsmatrix gefunden:

$$G(\Delta t,m) = \begin{pmatrix} 1 & a \\ -\bar{a} & 1-|a|^2 \end{pmatrix} \quad .$$

SATZ 40.1 (v. Neumann-Bedingung): Für die Eigenwerte $\mu_i(\Delta t,m)$ der Amplifikationsmatrix eines stabilen Differenzenverfahrens gilt

$$|\mu_i(\Delta t,m)| \leq 1 + K\Delta t$$

mit einer von $\Delta t,m$ unabhängigen Konstanten $K$.

BEWEIS: Sei das Verfahren stabil, d.h.

$$\|C^\ell(\Delta t)U\|_\infty \leq M(T)\|U\|_\infty \quad , \quad \Delta t\cdot\ell \leq T \quad .$$

Setzen wir nun

$$U = \begin{pmatrix} u_1 \\ \cdot \\ \cdot \\ \cdot \\ u_n \end{pmatrix} \quad , \quad u_k = e^{ihmk}c \quad ,$$

so wird

$$C^\ell(\Delta t)U = \begin{pmatrix} G^\ell(\Delta t,m)u_1 \\ \cdot \\ \cdot \\ \cdot \\ G^\ell(\Delta t,m)u_n \end{pmatrix} \quad .$$

Also muß für alle $m$

$$\|G^\ell(\Delta t,m)\|_\infty \leq M(T) \quad , \quad \Delta t\cdot\ell \leq T$$

sein. Sei $\rho(\Delta t,m) = \text{Max } |\mu_i(\Delta t,m)|$ der Spektralradius von $G(\Delta t,m)$. Nach Satz 16.1 ist

$$\rho^{\ell}(\Delta t,m) \leq \|G^{\ell}(\Delta t,m)\|_{\infty} \leq M(T) \quad , \quad \Delta t\cdot\ell \leq T \quad .$$

Es folgt

$$\rho(\Delta t,m) \leq M(T)^{1/\ell} \leq M(T)^{\Delta t/T} \leq 1 + K\Delta t \quad .$$

■

BEISPIELE:

1) Für das einfachste Differenzenverfahren der Wärmeleitungsgleichung haben wir gefunden

$$G(\Delta t,m) = 2\ (\cos hm - 1) + 1 \quad .$$

Für den einzigen Eigenwert $\mu_1 = G(\Delta t,m)$ ist daher $1-2\lambda \leq \mu_1 \leq 1$, und dieses Intervall wird bei beliebigen $h,m$ ganz ausgeschöpft. Für $\lambda > \frac{1}{2}$ ist daher die v. Neumann - Bedingung nicht erfüllt und das Verfahren also instabil. Für $\lambda \leq \frac{1}{2}$ ist die v. Neumann - Bedingung erfüllt. Das sagt aber zunächst nichts, weil die v. Neumann - Bedingung ja nur notwendig ist.

2) Für das der Wellengleichung äquivalente System haben wir

$$G(\Delta t,m) = \begin{pmatrix} 1 & a \\ -\bar{a} & 1-|a|^2 \end{pmatrix} \quad , \quad a = \lambda(e^{ihm} - 1)$$

erhalten. Die Eigenwerte $\mu$ von $G(\Delta t,m)$ sind Lösungen von

$$\mu^2 + (\alpha - 2)\mu + 1 = 0$$

$$\alpha = |a|^2 = \lambda^2 ((\cos hm - 1)^2 + (\sin hm)^2)$$

$$= 2\lambda^2 (1 - \cos hm) \quad .$$

Also ist $0 \leqq \alpha \leqq 4\lambda^2$, und für beliebige $h$, $m$ wird jeder Punkt dieses Intervalls erreicht. Die Lösungen $\mu_{1,2}$ der quadratischen Gleichung sind

$$\mu_{1,2} = 1 - \frac{\alpha}{2} \pm \sqrt{\alpha(\frac{\alpha}{4} - 1)} \quad .$$

Für $\alpha > 4$ ist $\mu_2 < 1 - \frac{\alpha}{2}$, und $|\mu_2| \leqq 1 + K\Delta t$ ist nicht möglich. Die v. Neumann'sche Stabilitätsbedingung ist also für $\lambda > 1$ nicht erfüllt, das Verfahren also instabil. Für $\alpha \leqq 4$ sind $\mu_1$, $\mu_2$ konjugiert komplex, und wegen $\mu_1\mu_2 = 1$ muß $|\mu_1| = |\mu_2| = 1$ sein. Also ist für $\lambda \leqq 1$ die v. Neumann - Bedingung erfüllt. Daraus folgt natürlich nichts, weil die v. Neumann - Bedingung ja nur notwendig ist.

---

3) Für die Differenzenverfahren (a), (b) zu $u_t = u_x$ ist

$$G(\Delta t,m) = 1 + \lambda e^{ihm} \quad \text{bzw.} \quad (1 - \lambda) + \lambda e^{ihm} \quad ,$$

also

$$|\mu_1|^2 = (1 + \lambda(1 - \cos hm))^2 + \lambda^2(\sin hm)^2 \quad \text{bzw.}$$

$$|\mu_1|^2 = (1 + \lambda(\cos hm - 1))^2 + \lambda^2(\sin hm)^2 \quad .$$

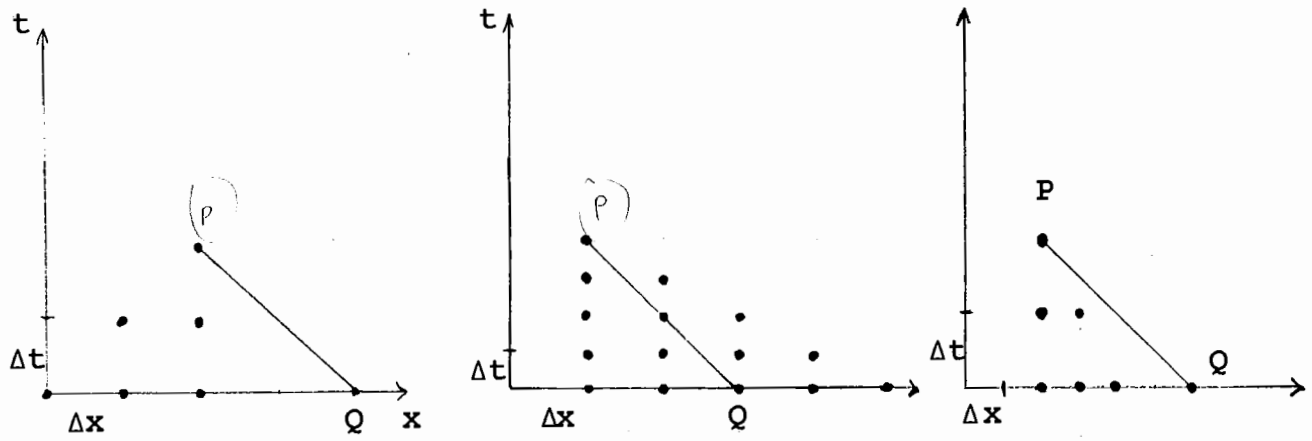
Im ersten Fall sieht man sofort, daß die v. Neumann - Bedingung für kein $\lambda > 0$ erfüllt ist. Verfahren (a) ist also instabil.

Für Verfahren (b) ist

$$|\mu_1|^2 = 1 + 2\lambda(\lambda-1)(1 - \cos hm) \leq 1$$

für $\lambda \leq 1$. Also ist die v. Neumann - Bedingung erfüllt.

Wir benutzen das letzte Beispiel, um (heuristisch) eine neue notwendige Stabilitätsbedingung herzuleiten. Wir betrachten dazu die Abhängigkeitsgebiete für $u_t = u_x$, Verfahren (a), (b):




(a) $\lambda = 1$ (b) $\lambda < 1$ (b) $\lambda > 1$

Der Wert von u in P hängt nur von Q ab. Beim Verfahren (a) hängt aber der Wert in P von dem in Q überhaupt nicht ab: Ändert man den Anfangswert in Q , so liefert (a) bei P immer

den gleichen Wert. (a) kann also nicht sinnvoll sein. Das gleiche gilt für (b) im Falle $\lambda > 1$. Im Falle $\lambda < 1$ gehört aber $Q$ zum Abhängigkeitsbereich von (b).

Wir kommen so zu einer weiteren notwendigen Bedingung, der Courant - Friedrichs - Lewy - Bedingung:

> Das Abhängigkeitsgebiet für die Differentialgleichung muß im Abhängigkeitsgebiet des Differenzenverfahrens enthalten sein.

## § 41 Konsistenz und Konvergenz bei AWAen

Wie bei gewöhnlichen Differentialgleichungen wird der lokale Diskretisierungsfehler durch Einsetzen der exakten Lösung in das Verfahren berechnet. In den Bezeichnungen von § 40 wird

$$T_{\Delta t}(x_k,t_{\ell+1}) = \frac{1}{\Delta t}\,(u(x_k,t_{\ell+1})-u(x_k,t_\ell) \;-\; \sum_\nu B_\nu(h)u(x_k,t_\ell)$$

der lokale Diskretisierungsfehler. Wie immer unterstellen wir $h = g(\Delta t)$. Konsistenz (-ordnung) und Konvergenz (-ordnung) wird nun genau wie bei gewöhnlichen Differentialgleichungen erklärt.

BEISPIELE:

1) Einfachstes Differenzenverfahren für $u_t = u_{xx}$. Mit $\lambda = \Delta t/h^2$ wird

$$T_{\Delta t}(x_k,t_\ell) = \mathcal{O}(\Delta t + h^2) = \mathcal{O}(\Delta t) \qquad , \quad u \in C^4 \quad .$$

Also ist das Verfahren konsistent von der Ordnung 1.

2) Einfachstes Differenzenverfahren für $u_{tt} = u_{xx}$. Mit $\lambda = \Delta t/\Delta x$ wird

$$T_{\Delta t}(x_k,t_\ell) = \mathcal{O}((\Delta t)^2 + h^2) = \mathcal{O}((\Delta t)^2) \;, \quad u \in C^4 \quad .$$

Also ist das Verfahren konsistent von der Ordnung 2.

SATZ 41.1: Das Verfahren sei stabil und konsistent (von der Ordnung p). Dann gibt es für $T > 0$ eine Konstante $M(T)$, so daß für $\Delta t\ell \leq T$ die Abschätzung

$$\operatorname{Max}_k \|u(x_k,t_\ell)-u_{k,\ell}\| \leq M(T)\left\{ \operatorname{Max}_k \|u(x_k,0)-u_{k,0}\| + T \operatorname{Max}_{k,j\leq\ell} \|T_t(x_k,t_j)\| \right\}$$

gilt $(\|\cdot\| = \|\cdot\|_\infty)$.

BEWEIS: Wir setzen

$$U_\ell = \begin{pmatrix} u_{1\ell} \\ \vdots \\ u_{n\ell} \end{pmatrix}, \quad U(t_\ell) = \begin{pmatrix} u(x_1,t_\ell) \\ \vdots \\ u(x_n,t_\ell) \end{pmatrix}, \quad T_{\Delta t}(t_\ell) = \begin{pmatrix} T_{\Delta t}(x_1,t_\ell) \\ \vdots \\ T_{\Delta t}(x_n,t_\ell) \end{pmatrix} .$$

Dann lautet das Verfahren

$$U_{\ell+1} = C(\Delta t)U_\ell$$

mit der Matrix $C(\Delta t)$ aus § 40. Weiter gilt

$$U(t_{\ell+1}) = C(\Delta t)U(t_\ell) + \Delta t\, T_{\Delta t}(t_{\ell+1}) \quad .$$

Subtraktion ergibt für $D_\ell = U(t_\ell) - U_\ell$

$$D_{\ell+1} = C(\Delta t)D_\ell + \Delta t\, T_{\Delta t}(t_{\ell+1}) \quad .$$

Ähnlich wie in Lemma 31.1 erhalten wir

$$D_\ell = C(\Delta t)^\ell D_0 + \Delta t \sum_{j=0}^{\ell-1} (C(\Delta t)^{\ell-j-1}\, T_{\Delta t}(t_{j+1}) \quad .$$

Nun benutzen wir die Stabilität: Für $\ell\Delta t \leq T$ ist mit der $\infty$-Norm

$$\|D_\ell\| \leq M(T)\|D_0\| + \Delta t\, M(T) \sum_{j=0}^{\ell-1} \|T_{\Delta t}(t_{j+1})\|$$

$$\leq M(T)\, \{\|D_0\| + T \operatorname{Max}_{j\leq\ell} \|T_{\Delta t}(t_j)\|\} \quad .$$

Dies ist die Behauptung ∎ .

Wie bei gewöhnlichen Differentialgleichungen könnten wir jetzt einen Konvergenzsatz formulieren. Wie er lautet ist jetzt wohl klar.

## § 42 Randwertaufgaben partieller Differentialgleichungen

Als Prototyp behandeln wir die Poisson'sche Differentialgleichung

$$-\Delta u = f \quad \text{in} \quad \Omega \subseteq \mathbb{R}^2 \ ,$$

$$\Delta = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} \ .$$

Dazu können noch Randbedingungen, z.B.

$$u = g \quad \text{auf} \quad \partial\Omega \quad \text{(Dirichlet)}$$

oder $\frac{\partial u}{\partial \nu} = g$ auf $\partial\Omega$ (Neumann, $\nu$ äußere Normale) .

Das Dirichlet-Problem ist immer lösbar. Z.B. ist für $\Omega = (0,1)^2$, $g = 0$

$$u(x_1,x_2) = \sum_{k,\ell=1}^{\infty} C_{k\ell} \sin k\pi x_1 \sin \ell\pi x_2$$

$$C_{k\ell} = \frac{4}{\pi^2(k^2+\ell^2)} \int_0^1 \int_0^1 f(x_1,x_2) \sin kx_1 \sin \ell x_2 \, dx_1 \, dx_2 .$$

Das Neumann - Problem ist nur unter der Verträglichkeitsbedingung

$$\int_\Omega f \, dx + \int_{\partial\Omega} g \, ds = 0$$

lösbar. Ist nämlich $u$ eine Lösung, so gilt nach dem Gauß'schen Integralsatz

$$\int_\Omega f \, dx = -\int_\Omega \Delta u \cdot 1 dx = -\int_{\partial\Omega} \frac{\partial u}{\partial \nu} \, ds = -\int_{\partial\Omega} g \, ds \ .$$

BEISPIELE:

1) Wirbelfreie Strömung.

$v(x,t)$ Geschwindigkeit, $\rho(x,t)$ Dichte.

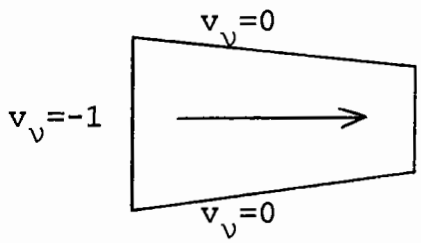
Kontinuitätsgleichung:

$$\dot{\rho} + \operatorname{div}(\rho v) = 0$$

Falls Strömung inkompressibel ($\rho$ unabhängig von x), wirbelfrei ($v = \nabla u$), stationär ($\dot{\rho} = o$, $\dot{v} = 0$):

$$\operatorname{div} \nabla u = \Delta u = 0 \quad .$$

Strömung durch ein sich verjüngendes Rohr:




Eingangsquerschnitt
= 2 × Ausgangsquerschnitt

In $u = \nabla v$ ausgedrückt erhält man das Neumann-Problem mit erfüllter Verträglichkeitsbedingung.

2) Potential $u$ des elektrischen Feldes $E = -\nabla u$ einer Ladungsverteilung $\rho$:

$$-\Delta u = 4\pi\rho$$

Exakte Lösung im freien Raum:

$$u(x) = \int \frac{\rho(y)dy}{|x-y|} \quad .$$

Falls $\rho$ in einem Gebiet $\Omega$ liegt, das von einem Leiter $\partial\Omega$ umgeben ist: $u = 0$ auf $\partial\Omega$. Lösung des Dirichtlet-Problems:

$$u(x) = 4\pi \int_{\Omega} G(x,y)\rho(y)dy \quad , \quad G \text{ Green'sche Funktion} \quad .$$

3) Stationäre Wärmeleitung:

$$0 = u_t = D\Delta u \quad \text{in} \quad \Omega \times [0,\infty]$$

Rand gekühlt: Dirichlet Rand isoliert: Neumann

4) Brown'sche Bewegung. Ein Teilchen starte bei $x$ und führe eine Irrfahrt durch. Wie groß ist die Wahrscheinlichkeit $w(x)$, daß das Teilchen auf dem Randstück $T \subseteq \partial\Omega$ landet?

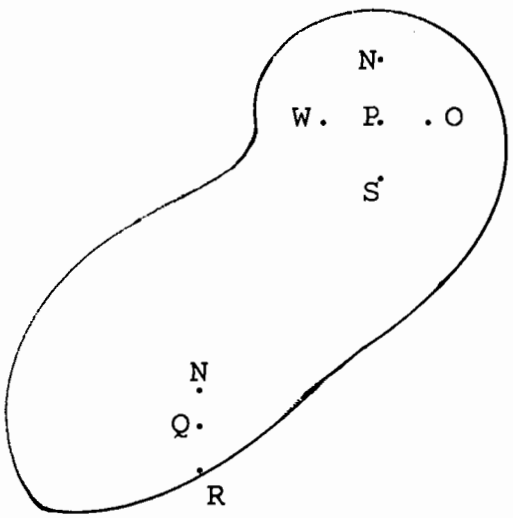
$$-\Delta w(x) = 0 \quad , \quad w(x) = \begin{cases} 1 & , \quad x \in T \\ 0 & \quad \text{sonst} \end{cases} \quad .$$

## § 43 Das einfachste Differenzenverfahren für Randwertaufgaben

Wir betrachten das Dirichlet - Problem

$$-\Delta u = f \quad \text{in} \quad \Omega \quad , \qquad u = g \quad \text{auf} \quad \partial\Omega$$

mit einer kompakten Menge $\Omega \subseteq \mathbb{R}^2$. Wir überdecken $\Omega$ mit einem




quadratischen Gitter der Schrittweite h. Ein Gitterpunkt P heißt innerer Gitterpunkt, wenn er samt seiner Nachbarn W, N, O, S in $\Omega$ liegt. Ein Gitterpunkt Q heißt randnah, wenn Q in $\Omega$, aber einer seiner Nachbarn nicht in $\Omega$ liegt. Wir wollen annehmen, $\Omega$ und das Gitter seien so beschaffen, daß es zu jedem randnahen Gitterpunkt Q einen Randpunkt R gibt mit Abstand $d \equiv h$ von Q, so daß der R gegenüberliegende Punkt (von Q ausgesehen, in Abbildung N ) innere Gitterpunkt ist.

Für jeden inneren Gitterpunkt P schreiben wir nun als diskretes Analogon für die Differentialgleichung

$$4\,u_P - u_N - u_S - u_O - u_W = h^2 f(P)$$

hin. Treten dabei Gitterpunkte auf, die auf dem Rand von $\Omega$ liegen, so wird der entsprechende Wert von $g$ eingesetzt. In jedem randnahen Punkt $Q$ approximieren wir $u(Q)$ durch lineare Interpolation zwischen $N$, $R$, also

$$u(Q) \sim \frac{1}{h+d}\,(hu(R) + du(N)) \quad .$$

Dies führt zu

$$u_Q - \frac{d}{h+d}\,u_N = \frac{h}{h+d}\,g(R) \quad .$$

BEISPIEL: $\Omega = [0,4] \times [0,5]$, $h = 1$, zeilenweise Numerierung der Gitterpunkte, Randwerte $g = 0$. Wir bekommen ein System $Au = f$ mit folgender Matrix $A$ :

| 1 | 2 | 3 | 4 |
|---|---|---|---|
| 5 | 6 | 7 | 8 |
| 9 | 10 | 11 | 12 |

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 4 | -1 | | | -1 | | | | | | | |
| 2 | -1 | 4 | -1 | | | -1 | | | | | | |
| 3 | | -1 | 4 | -1 | | | -1 | | | | | |
| 4 | | | -1 | 4 | | | | -1 | | | | |
| 5 | -1 | | | | 4 | -1 | | | -1 | | | |
| 6 | | -1 | | | -1 | 4 | -1 | | | -1 | | |
| 7 | | | -1 | | | -1 | 4 | -1 | | | -1 | |
| 8 | | | | -1 | | | -1 | 4 | | | | -1 |
| 9 | | | | | -1 | | | | 4 | -1 | | |
| 10 | | | | | | -1 | | | -1 | 4 | -1 | |
| 11 | | | | | | | -1 | | | -1 | 4 | -1 |
| 12 | | | | | | | | -1 | | | -1 | 4 |

Zur Lösung des linearen Gleichungssystems kommen neben den direkten Verfahren (Elimination unter Ausnutzung der Bandstruktur) vor allem

iterative Verfahren in Frage, z.B. SOR (vgl. § 16). Dieses lautet für das System $Ax = b$, $A = D + L + R$,

$$Dx^{k+1} = \omega(b - Lx^{k+1} - Rx^{k}) + (1-\omega)Dx^{k} \quad .$$

Ein Schritt dieses Verfahrens wird durch die Schleife

$$\text{do} \quad i=1 \text{ to } n;$$
$$x_i = \omega(b_i - \sum_{j \neq i} a_{ij} x_j) / a_{ii} + (1-\omega)x_i \ ;$$
$$\text{end};$$

realisiert. In unserem Fall benötigt ein Schritt also nur etwa $5n$ Rechenoperationen.

## § 44 Optimale Relaxationsparameter für SOR

Wir wissen aus § 16, daß das SOR - Verfahren für positiv definite Matrizen konvergiert, falls $0 < \omega < 2$ ist. Wir wollen jetzt $\omega$ für die in § 43 Gleichungssysteme so bestimmen, daß die Konvergenz möglichst schnell ist, d.h. der Spektralradius $\rho(C_\omega)$ der SOR-Matrix

$$C_\omega = (D + \omega L)^{-1}((1-\omega)D - \omega R) , \qquad A = D + L + r$$

soll möglichst klein sein.

DEFINITION 44.1: Die (n,n) - Matrix A hat die Eigenschaft A, falls sich $\{1,\ldots,n\}$ so in zwei disjunkte Teilmengen S, T zerlegen läßt, daß gilt:

$$a_{ij} \neq 0,\ i \neq j \ \Rightarrow\ i,j \text{ nicht beide in S oder beide in T.}$$

Ordnet man die Zeilen und Spalten von A so, daß zuerst die mit Nummern in S , dann die mit Nummern in T kommen, so hat A also folgende Gestalt:

| | S | T |
|---|---|---|
| S | $D_1$ | $F_1$ |
| T | $E_1$ | $D_2$ |

, $D_1$, $D_2$ quadratisch und diagonal.

Eine solche Matrix heißt konsistent geordnet.

BEISPIEL: Die Matrix A aus § 43 hat offenbar die Eigenschaft A: Man braucht die Mengen S, T nur gemäß

| | | | |
|---|---|---|---|
| s | t | s | t |
| t | s | t | s |
| s | t | s | t |

zu wählen. Es gilt dann:

$i \neq j$ und $a_{ij} \neq 0$ $\Rightarrow$ Punkte i,j benachbart $\Rightarrow$ i,j können nicht beide zu S und nicht beide zu T gehören.

Numerieren wir so, daß zunächst alle Punkte aus S, dann alle aus T an die Reihe kommen, also

| | | | |
|---|---|---|---|
| 1 | 7 | 2 | 8 |
| 9 | 3 | 10 | 4 |
| 5 | 11 | 6 | 12 |

so nimmt die Matrix die Gestalt

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 4 | | | | | | -1 | | -1 | | | |
| 2 | | 4 | | | | | -1 | -1 | | -1 | | |
| 3 | | | 4 | | | | -1 | | -1 | -1 | -1 | |
| 4 | | | | 4 | | | | -1 | | -1 | | -1 |
| 5 | | | | | 4 | | | | -1 | | -1 | |
| 6 | | | | | | 4 | | | | -1 | -1 | -1 |
| 7 | -1 | -1 | -1 | | | | 4 | | | | | |
| 8 | | -1 | | -1 | | | | 4 | | | | |
| 9 | -1 | | -1 | | -1 | | | | 4 | | | |
| 10 | | -1 | -1 | -1 | | -1 | | | | 4 | | |
| 11 | | | -1 | | -1 | -1 | | | | | 4 | |
| 12 | | | | -1 | | -1 | | | | | | 4 |

an und ist konsistent geordnet.

SATZ 44.1: A sei konsistent geordnet, $C_\omega$, $\omega \neq 0$, die zu A gehörige SOR - Matrix, und B die Matrix des Gesamtschrittverfahrens für A. Dann ist $\mu \neq 0$ Eigenwert von $C_\omega$ genau dann, wenn

$$\mu\,\lambda^2\,\omega^2 = (1 - \omega - \mu)^2$$

ist mit einem Eigenwert $\lambda$ von B.

BEWEIS: Nach §16 ist mit $A = D + L + R$

$$B = -D^{-1}(L + R), \quad C_\omega = (D+\omega L)^{-1}((1-\omega)D - \omega R) \quad .$$

$\lambda$ ist also Eigenwert von B genau dann, wenn

$$P(\lambda) = \det(L + R + \lambda D) = 0 \quad .$$

Ebenso ist $\mu$ Eigenwert von $C_\omega$ genau dann, wenn

$$Q(\mu) = \det\left( \frac{1-\omega-\mu}{\omega} D - R - \mu L \right) = 0 \quad .$$

Ist $A$ konsistent geordnet, so ist

$$P(\lambda) = \det\begin{pmatrix} \lambda D_1 & F_1 \\ E_1 & \lambda D_2 \end{pmatrix} \quad , \quad Q(\mu) = \det\begin{pmatrix} \nu D_1 & -F_1 \\ -\mu E_1 & \nu D_2 \end{pmatrix}, \quad \nu = \frac{1-\omega-\mu}{\omega} \quad .$$

Es gilt

$$P(-\lambda) = \det\begin{pmatrix} -\lambda D_1 & F_1 \\ E_1 & -\lambda D_2 \end{pmatrix} = \det\begin{pmatrix} -I & O \\ O & I \end{pmatrix}\begin{pmatrix} \lambda D_1 & F_1 \\ E_1 & \lambda D_2 \end{pmatrix}\begin{pmatrix} I & O \\ O & -I \end{pmatrix} = (-1)^n P(\lambda).$$

Weiter hat man

$$\begin{pmatrix} I & O \\ O & \mu^{-1/2} I \end{pmatrix}\begin{pmatrix} \nu D_1 & -F_1 \\ -\nu E_1 & \nu D_2 \end{pmatrix}\begin{pmatrix} \mu^{-1/2} I & O \\ O & I \end{pmatrix} = - \begin{pmatrix} -\mu^{-1/2}\nu D_1 & F_1 \\ E_1 & -\mu^{-1/2}\nu D_2 \end{pmatrix},$$

also

$$\mu^{-n/2} Q(\mu) = (-1)^n P(-\mu^{-1/2}\nu) = P(\mu^{-1/2}\nu) \quad .$$

Sei nun $\mu \neq 0$ ein Eigenwert von $C$ . Dann gibt es einen Eigenwert $\lambda$ von $B$ mit $\lambda = \mu^{-1/2}\nu$ oder $\mu\,\lambda^2\,\omega^2 = (1-\omega-\mu)^2$. Ist

umgekehrt für ein $\mu$ diese Beziehung mit einem Eigenwert $\lambda$ von $B$ erfüllt, so ist $\pm\lambda = \mu^{-1/2}\nu$. Da mit $\lambda$ auch $-\lambda$ Eigenwert von $B$ ist, folgt also $Q(\mu) = 0$. ∎

SATZ 44.2: Sei $A$ konsistent geordnet und positiv definit. Es sei $\lambda_1$ der größte Eigenwert von $B$. Dann nimmt $\rho(C_\omega)$ sein Minimum für

$$\omega = \omega_{opt} = \frac{2}{1+\sqrt{1-\lambda_1^2}}$$

an, und es gilt

$$\rho(C_{\omega_{opt}}) = \omega_{opt} - 1 \quad .$$

BEWEIS: Zunächst folgt aus der positiven Definitheit von $A$, daß alle Eigenwerte von $B$ reell und $\leq 1$ sind. Nach Satz 44.1 ist $\mu$ genau dann Eigenwert von $C_\omega$, wenn mit einem Eigenwert $\lambda$ von $B$

$$\mu\lambda^2\omega^2 = (1 - \omega - \mu)^2$$

gilt. Diese quadratische Gleichung hat die beiden Lösungen

$$\mu_{1,2}(\omega,\lambda) = 1 - \omega + \frac{\lambda^2\omega^2}{2} \pm |\lambda|\omega\sqrt{d} \quad , \quad d = 1 - \omega + \frac{\lambda^2\omega^2}{4} .$$

Ist $d \leq 0$, so ist $|\mu_1| = |\mu_2| = |1-\omega|$, wie man unmittelbar an der quadratischen Gleichung sieht. Für $d \geq 0$ entnimmt man der Figur 1

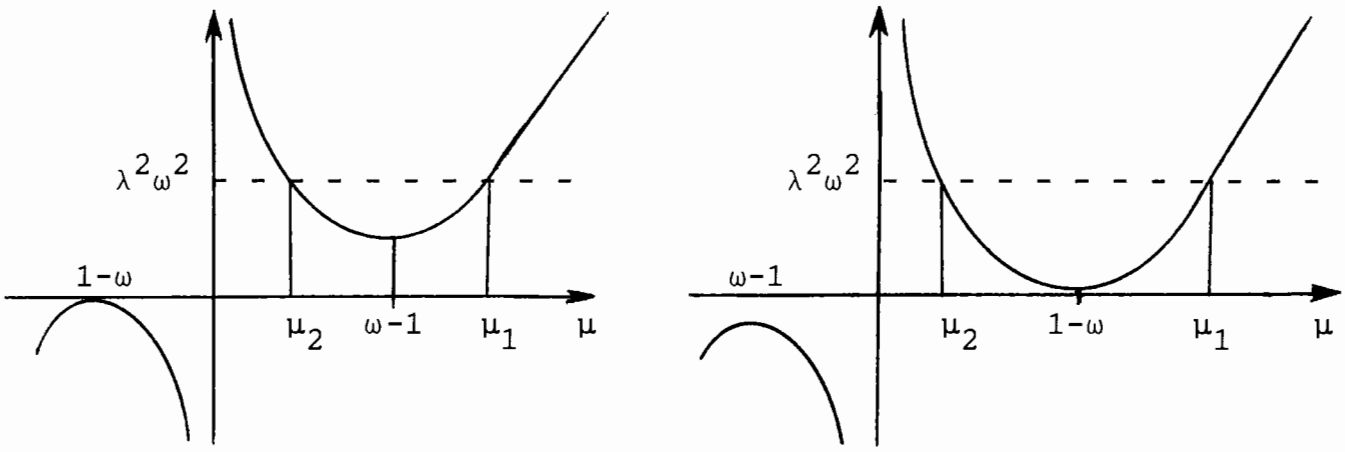



FIG. 1: Die Funktion $\frac{1}{\mu}(1- -\mu)^2$ für $\omega>1$ (links) und $\omega<1$ (rechts).

$$0 \leqq \mu_2(\omega,\lambda) \leqq |\omega-1| \leqq \mu_1(\omega,\lambda)$$

und daß $\mu_1$ eine monotone Funktion von $\lambda$ ist. Insgesamt erhalten wir also

$$\rho(C_\omega) = \text{Max}\ \{|\omega-1|,\quad \mu_1(\omega,\lambda_1)\} \quad .$$

Diese Funktion von $\omega$ ist in Fig. 2 gezeichnet. Für $\omega_{opt}$ gilt offenbar

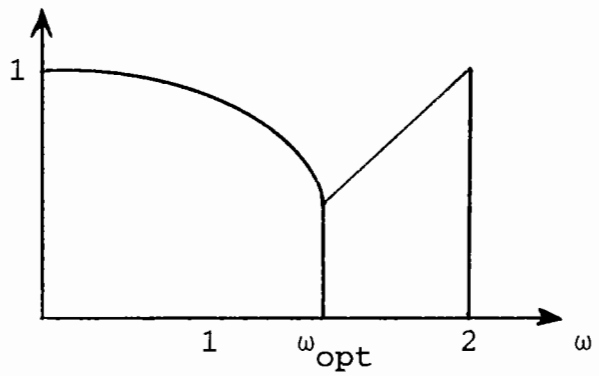



FIG. 2: Graph von $\rho(C_\omega)$

$\mu_1(\omega,\lambda_1) = |\omega-1|$, also $d \leqq 0$ und $d \geqq 0$, mithin $d = 0$. Diese quadratische Gleichung für $\omega$ hat die Lösungen

$$\omega_{1,2} = \frac{2}{1 \pm \sqrt{1-\lambda_1^2}} \quad ,$$

wovon nur die mit „ + " in Frage kommt, weil $\omega < 2$ sein muß. Also ist

$$\omega_{opt} = \frac{2}{1+\sqrt{1-\lambda_1^2}} \quad , \quad \rho(C_{\omega_{opt}}) = \omega_{opt} - 1 \quad .$$

■

FOLGERUNGEN AUS DEM BEWEIS:

1) Für $\omega > \omega_{opt}$ sind die betragsgrößten Eigenwerte von $C_\omega$ konjugiert komplex. Für $\omega < \omega_{opt}$ ist der betragsgrößte Eigenwert von $C_\omega$ reell und positiv.

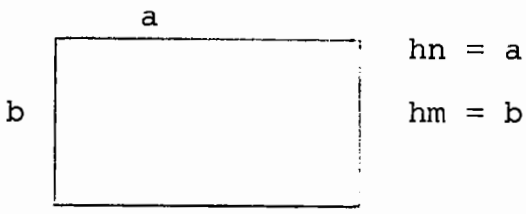
2) Da $\mu_1$ bei $\omega_{opt}$ eine senkrechte Tangente hat, ist Unterschätzung von $\omega_{opt}$ schlimmer als Überschätzung.

3) Für $\omega = 1$ erhält man

$$\rho(C_1) = \mu_1(1,\lambda_1) = \lambda_1^2 = (\rho(B))^2 \quad .$$

Dies bedeutet, daß das Einschrittverfahren doppelt so schnell konvergiert wie das Gesamtschrittverfahren.

Wir berechnen nun $\lambda_1$ für die Matrix A des Differenzenverfahrens für ein Rechteck.




Die Eigenwerte $\Lambda$ von A erfüllen

$$2u_{k,\ell}-u_{k-1,\ell}-u_{k+1,\ell} + 2u_{k,\ell}-u_{k,\ell-1}-u_{k,\ell+1} = h^2\Lambda u_{k,\ell} \quad ,$$

$$k,\ell = 1,\ldots,n-1, \quad u_{k,\ell} = 0 \quad \text{für} \quad k = 0,n \quad \text{und für} \quad \ell = 0,m.$$

Wir versuchen den Ansatz

$$u_{k,\ell} = \sin \alpha k\pi/n \sin \beta\ell\pi/m \quad , \ \alpha=1,\ldots,n-1 \ , \quad \beta=1,\ldots,m-1 \ .$$

Es ist nach dem Additionstheorem

$$\sin \alpha(k-1)\pi/n + \sin \alpha (k+1)\pi/n = 2 \sin \alpha k\pi/n \cos \alpha\pi/n$$

und damit

$$2u_{k,\ell} - u_{k-1,\ell} - u_{k+1,\ell} = 2(1 - \cos \alpha\pi/n)u_{k,\ell} \quad .$$

Zusammen mit der entsprechenden Beziehung für die zweiten Differenzen in $\ell$-Richtung findet man die Eigenwerte

$$h^2\Lambda_{\alpha,\beta} = 2(1 - \cos \alpha\pi/n) + 2(1 - \cos \beta\pi/m)$$

von A. Die Eigenwerte $\lambda$ von $B = -D^{-1}(A-D) = -\frac{h^2}{4}\left(A - \frac{4}{h^2} I\right)$ sind dann

$$\lambda_{\alpha,\beta} = -\frac{h^2}{4}\left(\Lambda_{\alpha,\beta} - \frac{4}{h^2} I\right) = \frac{1}{2}\left(\cos \alpha\pi/n + \cos \beta\pi/m\right) \quad .$$

Den maximalen Eigenwert $\lambda_1$ erhält man für $\alpha = \beta = 1$:

$$\lambda_1 = \frac{1}{2} (\cos \pi/n + \cos \pi/m) \quad .$$

Damit ist auch $\omega_{opt}$, $\rho(C_{\omega_{opt}})$ bestimmt. Für $a = b = \pi$ wird

$$\lambda_1 = \cos h \ , \quad \omega_{opt} = \frac{2}{1+\sin h} \ , \quad \rho(C_{\omega_{opt}}) = \frac{1-\sin h}{1+\sin h} \quad .$$

Wir vergleichen nun einige Verfahren für kleine h. Die letzte Spalte enthält die Anzahl der Iterationsschritte, welche zur Gewinnung eines Faktors $e^{-1}$ an Genauigkeit notwendig sind (d.h. $1/\ell n(1/\rho)$).

| Verfahren | Spektralradius | # Schritte für $e^{-1}$ |
|---|---|---|
| Gesamtschritt | $1 - h^2/2$ | $2h^{-2}$ |
| Einzelschritt | $1 - h^2$ | $h^{-2}$ |
| SOR, $\omega = \omega_{opt}$ | $1 - 2h$ | $\frac{1}{2} h^{-1}$ |

Für das Beispiel aus Aufgabe 42 (n = m = 10) erhält man z.B. folgende mittlere Fehler:

| Iterationen | $\omega = 1$ | $\omega = 1.53$ | $\omega = 1.53$, konsistent geordnet |
|---|---|---|---|
| 1 | 36 | 93 | 52 |
| 11 | 12 | 3.3 | 1.7 |
| 21 | 4.3 | 0.010 | 0.0042 |

Man beachte, daß zur Anwendung der Theorie die Reihenfolge der Gleichungen und die Numerierung der Unbekannten eine Rolle spielt.

Im allgemeinen ist $\lambda_1$ und damit $\omega_{opt}$ nicht bekannt. Zur Bestimmung von $\omega_{opt}$ hat man, neben Ausprobieren, folgende Möglichkeiten.

1) Berechne $\lambda_1$ durch einige Schritte der Potenzmethode für das Eigenwertproblem $Ax = (\lambda+1)Dx$ und wende dann Satz 44.2 an.

2) Berechne $\mu_1 = \mu_1(\omega, \mu_1)$ durch die Potenzmethode für $C_\omega$. Für $\omega < \omega_{opt}$ wissen wir, daß der betragsgrößte Eigenwert $\mu_1$ von $C_\omega$ reell und positiv ist, und

$$\lambda_1 = \frac{\omega + \mu_1 - 1}{\omega\sqrt{\mu_1}} \quad .$$

$\mu_1$ kann durch die Potenzmethode für $C_\omega$ berechnet werden. Dazu kann man die bei der SOR-Iteration auftretende Vektorfolge $x^k$ verwenden, etwa in der Form

$$\mu_1 \sim \frac{x_{i_0}^{k+1} - x_{i_0}^{k}}{x_{i_0}^{k} - x_{i_0}^{k-1}} \quad .$$

Tritt hier Konvergenz ein, so ist dies vermutlich ein Zeichen dafür, daß $\omega > \omega_{opt}$ und damit die betragsgrößten Eigenwerte von $C_\omega$ konjugiert komplex sind.

Für das Beispiel aus Aufgabe 42 erhält man z.B. für $\omega = 1$:

| Iteration | $\mu_1$ | $\lambda_1$ | $\omega_{opt}$ |
|---|---|---|---|
| 1 | 2.000 | - | - |
| 5 | 0.947 | 973 | 1.626 |
| 10 | 909 | 954 | 1.537 |
| 15 | 905 | 951 | 1.528 |

Der letzte Wert von $\omega$ stimmt in allen angegebenen Stellen.