Einführung in die Numerische Mathematik, Höhere Numerische Mathematik

F. Natterer

Institut für Numerische
und instrumentelle Mathematik

WS 2004/05, Di/Fr 13-15, M 4
und SS 2005, Di/Fr 11-13, M4

Inhaltsverzeichnis

Numerik : Zahlenmäßige Lösung mathematischer Probleme, wie
Lineare Gleichungssysteme
Nichtlineare Gleichungen/Nullstellen
Darstellung von Funktionen
Eigenwertprobleme
Integrale
mit Hilfe von Computern Sicherheit, Effizienz, Einfachheit.

Vorkenntnisse : Anfängervorlesungen
Höhere Programmiersprache (C, FORTRAN, MATLAB)

Eigentümlichkeiten der Numerik:

1. Hat den Charakter eines Handwerks: Üben, üben, üben!

2. Umgang mit Ungenauigkeiten: Rundungsfehler, Datenfehler.
3. Effizienz und Praktikabilität.
4. Benutzt Hilfsmittel aus vielen mathematischen Disziplinen.

Ausblick

Typische Fragestellungen der Numerik:

1. Nullstellen

$$f : \mathbf{R} \to \mathbf{R}\,.$$

Suche $\overline{x} \in \mathbf{R}$ mit $f(\overline{x}) = 0$.

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

Allgemeiner:

$$\begin{aligned} f(x) &= x^n + a_{n-1}x^{n-1} + \cdots + a_0 \ ? \\ f(x) &= e^x - \sin x \ ? \end{aligned}$$

Iterative Verfahren:

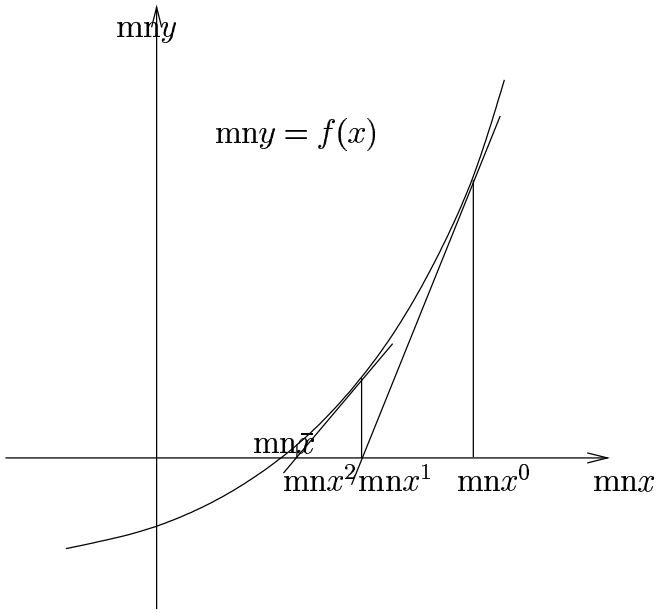


Abbildung 1: *Newton-Verfahren*

Ausgangsnäherung x_0. Berechne Folge (x_k) von Näherungen nach folgendem Rezept (Algorithmus):

Sei x berechnet. Berechne Tangente an $y = f(x)$ in $(x_k, f(x_k))$. Nehme als x_{k+1} die Nullstellen der Tangente.

$$\begin{aligned} y &= f(x_k) + f'(x_k)(x - x_k) \\ 0 &= f(x_k) + f'(x_k)(x_{k+1} - x_k) \\ x_{k+1} &= x_k - \frac{f(x_k)}{f'(x_k)} \quad \text{Newton-Verfahren} \end{aligned}$$

Beispiel:

$$\begin{aligned} f(x) &= x^2 - a\,, \quad \overline{x}_{1,2} = \pm\sqrt{a} \quad (a > 0) \\ x_{k+1} &= x_k - \frac{x_k^2 - a}{2x_k} = \frac{1}{2}\left(x_k + \frac{a}{x_k}\right) \end{aligned}$$

Numerisches Beispiel: $a = 2$, $\overline{x}_1 = 1.414213\ldots$

k	x_k	Anzahl der korrekten Dezimale
0	1	1
1	1.5	1
2	1.417	3
3	1.414216	6
4	1.414213562	10

Fragen:

Konvergenz $x \to \overline{x}$, für welche x_0?
Wie schnell ist die Konvergenz?
Übertragung auf Systeme von Gleichungen?

2. Integration

$$f : \mathbf{R}^1 \to \mathbf{R}^1 \,.$$

Berechne $\int_a^b g(x)dx$.

Stammfunktion F von f bekannt:

$$\int_a^b f(x)dx = F(b) - F(a)\,.$$

Annahme: Routine zur Berechnung von $f(x)$ für jedes x steht zur Verfügung.

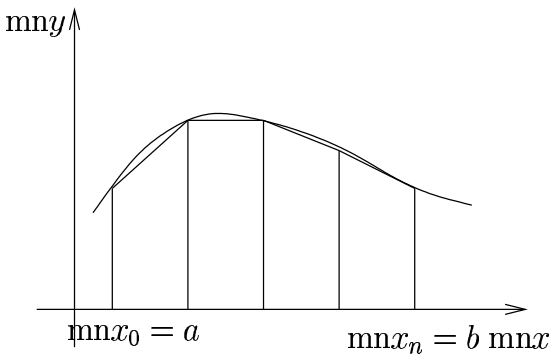


Abbildung 2: *Trapezregel*

$x_i = a + ih$, $i = 0, \ldots, n$, $h = (b-a)/n$

Approximiere das Integral durch die Fläche der Trapeze:

$$\begin{aligned} T_h &= \frac{h}{2}\left(f(x_0)+f(x_1)\right)+\frac{h}{2}\left(f(x_1)+f(x_2)\right)+\cdots+\frac{h}{2}\left(f(x_{n-1})+f(x_n)\right) \\ &= h\left(\frac{1}{2}f_0+f_1+\cdots f_{n-1}+\frac{1}{2}f_n\right)\ , \quad f_i=f(x_i)\ . \end{aligned}$$

Beispiel:

$$\begin{aligned} \int_0^1 e^x dx = e-1 &= 1.71828\ldots \\ T_1 = 1\cdot\left(\frac{1}{2}+\frac{1}{2}e\right) &= 1.859 \\ T_{\frac{1}{2}} = \frac{1}{2}\left(\frac{1}{2}+e^{1/2}+\frac{1}{2}e\right) &= 1.754 \\ T_{1/4} = \frac{1}{4}\left(\frac{1}{2}+e^{1/4}+e^{1/2}+e^{3/4}+\frac{1}{2}e\right) &= 1.727 \end{aligned}$$

Fragen:

Konvergenz für $h \to 0$?
Wie schnell ist die Konvergenz?
Formeln mit besserer Konvergenz?

3. Lineare Gleichungssysteme

$A(n,n)$-Matrix reeller Zahlen, $b \in \mathbf{R}^n$.

Gesucht: x mit $Ax = b$.
Cramer'sche Regel: A_j entstehe aus A durch Ersetzen der j-ten Spalte mit b. Dann gilt

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

Anzahl der Rechenoperationen:

$\det(A)$: Summe von $n!$ Produkten mit je n Faktoren, also
$n!(n-1)$ Multiplikationen und
$n!-1$ Additionen/Subtraktionen, d.h.
$(n+1)!$ Multiplikationen $+\mathrm{O}(n!)$ weitere Operationen.

(Wir werden später sehr viel effizientere Methoden zur Berechnung von Determinanten kennenlernen.) Das gleiche gilt für $\det(A_j)$. Also benötigen wir

$(n+2)!$ Multiplikationen + $\mathrm{O}((n+1)!)$ weitere Operationen.

Wir werden ein Verfahren angeben, das mit $\frac{1}{3}n^3 + O(n^2)$ Operationen auskommt.

Beispiel: $n = 20$:

$$\begin{aligned} (n+2)! &= 1.1\ {}_{10}{}^{21} \\ \frac{1}{3}n^3 &= 2.7\ {}_{10}{}^{3} \end{aligned}$$

Auswirkung von Fehlern:

$$\begin{aligned} x_1 + x_2 &= 1 \\ x_1 + 0.99x_2 &= 1 \end{aligned} \quad \text{Lösung} : \quad \begin{aligned} x_1 &= 1 \\ x_2 &= 0 \end{aligned}$$

$$\begin{aligned} 1.01x_1 + 1.01x_2 &= 1 \\ x_1 + 0.99x_2 &= 1 \end{aligned} \quad \text{Lösung} : \quad \begin{aligned} x_1 &= \tfrac{200}{101} \sim 2 \\ x_2 &= -\tfrac{100}{101} \sim -1 \end{aligned}$$

Dieser Fehler hat nichts mit Rundung zu tun.

4. Diskrete Fourier-Transformation

5. Über- und unterbestimmte lineare Systeme

6. Eigenwertprobleme

Kapitel 1

Fehler beim numerischen Rechnen

1.1 Absoluter und relativer Fehler

$x \in \mathbb{R}$, $\tilde{x}$ Näherung für x.

Absoluter Fehler : $\Delta x = \tilde{x} - x$

Relativer Fehler : $\Delta x/x = (\tilde{x} - x)/x \quad (x \neq 0)$

In der Numerik ist der relative Fehler der wichtigere. Er wird in Prozent angegeben.

Fehlerfortpflanzung:
Auszuwerten sei $y = f(x)$, $f : \mathbf{R}^m \to \mathbf{R}^n$.

Wie hängt der Fehler der y_i mit dem der x_j zusammen?

Satz 1.1.1 *Sei D offen und konvex, und sei $f \in C^1(D)$. Dann gilt für $x, x + \Delta x \in D$*

$$|\Delta y_i| \leq \sum_{j=1}^{m} \left| \sup_{z \in D} \frac{\partial f_i}{\partial x_j}(z) \right| |\Delta x_j| \,, \quad i = 1, \ldots, n \,.$$

Beweis: Mittelwertsatz der Differentialrechnung.

□

Folgerung: Für die relativen Fehler nahe bei x gilt ungefähr

$$\begin{aligned} \left| \frac{\Delta y_i}{y_i} \right| &\leq \sum_{j=1}^{m} k_{ij} \left| \frac{\Delta x_j}{x_j} \right| \,, \\ k_{ij}(x) &= \left| x_j \frac{\partial f_i}{\partial x_j}(x)/f_i(x) \right| \,. \end{aligned}$$

Die k_{ij} heißen Verstärkungsfaktoren.

Beispiele:

1) $y_1 = x_1 x_2$, $\quad k_{1j} = 1$

2) $y_1 = x_1 / x_2$, $\quad k_{1j} = 1$

3) $y_1 = x_1^a$, $\quad k_{11} = |a|$

4) $y_1 = x_1 + x_2$, $\quad k_{1j} = \left|\frac{x_i}{x_1+x_2}\right|$.

Hier sind die Verstärkungsfaktoren groß, falls $x_1 + x_2$ klein im Vergleich zu x_1, x_2. Zahlenbeispiel:

$$\begin{array}{rcl} 1.14 - 1.03 & = & 0.11 \\ 1.15 - 1.02 & = & 0.13 \\ \text{Fehler} \sim 1\% & & \text{Fehler} \sim 20\% \end{array}$$

5) $y^2 - x_1 y - x_2 = 0$, also
$y_1 = \frac{x_1}{2} + \sqrt{d}$, $y_2 = \frac{x_2}{2} - \sqrt{d}$, $d = x_1^2/4 + x_2$.
$k_{i1} = \left|\frac{x_i}{2\sqrt{d}}\right|$, $k_{12} = \left|\frac{y_2}{2\sqrt{d}}\right|$, $k_{22} = \left|\frac{y_1}{2\sqrt{d}}\right|$.

Die Verstärkungsfaktoren sind groß, falls $\sqrt{d}$ klein im Vergleich zu x_1, y_1 oder y_2.

1.2 Rechnerarithmetik

Wir beschreiben eine fiktive Rechnerarithmetik, die wir dezimale normalisierte Gleitkommaarithmetik mit Mantissenlänge m nennen. Existierende Rechner verfügen über eine Arithmetik, die unserer fiktiven sehr nahe kommt. Eine gewisse Standardisierung ist mit dem IEEE 754 Standard versucht worden.

1. Menge A der Maschinenzahlen. Dies ist die Menge der Zahlen der Form

$$\begin{array}{l} \pm 0.a_1 \ldots a_m 10^b \\ 0 \le a_i < 10 \text{ , } b \text{ ganz} \\ \text{entweder } a_1 \neq 0 \text{ oder alle } a_i = 0 \text{ (Normalisierung)} \end{array}$$

b heißt Exponent, $\pm 0.a_1 \ldots a_m$ Mantisse. Der Wert einer solchen Maschinenzahl ist natürlich

$$\pm \sum_{i=1}^{m} a_i 10^{b-i}$$

Beispiele: $m = 4$

$0.3142_{10}1$ ist Maschinenzahl mit Wert 3.142.
$0.31425_{10}1$, $0.0314_{10}1$ sind keine Maschinenzahlen.

2. Rundung. Eine Abbildung $rd : \mathbb{R} \to A$ heißt Rundung, falls

$$|rd(x) - x| \leq |a - x|$$

für alle $a \in A$.

Beispiel: Für $x = 0$ sei $rd(x) = 0$. Für $x \neq 0$ sei $x = \pm 0.a_1 a_2 \ldots \ \ 10^b$ mit $a_1 \neq 0$ und $rd(x) = \pm \tilde{a} 10^b$, wobei

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

Für $m = 4$ ist z.B.

$$\begin{array}{rclrcl} \pi & = & 3.1415926\ldots & rd(\pi) & = & 0.3142_{10}1 \\ \sqrt{57} & = & 7.5498\ldots & rd(\sqrt{57}) & = & 0.7550_{10}1 \\ x & = & 0.12535 & rd(x) & = & 0.1254_{10}0 \\ x & = & 0.1253499\ldots & rd(x) & = & 0.1253_{10}0 \end{array}$$

Diese Rundung ist also nicht eindeutig, rd also im mathematischen Sinne keine Abbildung. Dies ist aber leicht zu beheben.

Die Zahl

$$\text{eps} = 5 \cdot 10^{-m}$$

heißt Maschinengenauigkeit.

Satz 1.2.1 *Für jede Rundung rd gilt für* $x \neq 0$

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

Beweis: Sei $x = \pm 0.a_1 \ldots a_m \ldots \cdot 10^b$, $a_1 \neq 0$. Offenbar ist

$$\begin{array}{rcl} |rd(x) - x| & \leq & 5 \cdot \ 10^{-m-1} 10^b \, , \\ |x| & \geq & 10^{b-1} \, . \end{array}$$

Division ergibt die Behauptung.

Durch folgendes Programm kann man die Maschinengenauigkeit ungefähr berechnen:

$x = 1;$
while $(1 + x > 1)$ $x = x/2;$
$eps = x;$

□

3. Maschinenoperationen

Die reellen Operationen $+$, $-$, $/$, $\cdot$ können in A im allgemeinen nicht ausgeführt werden.

Beispiele: $m = 2$.

$$\begin{array}{lll} 1/0.9 = 1.111\ldots & \notin & A \\ 1.1 \cdot 1.1 = 1.21 & \notin & A \\ 0.13 + 0.0071 = 0.1371 & \notin & A \end{array}$$

Man definiert Maschinenoperationen $\oplus$, $\ominus$, $\oslash$, $\odot$ durch

$$x \oplus y = rd(x + y)\ , \quad x \ominus y = rd(x - y) \ \text{ usw.}$$

Satz 1.2.2 *Der relative Fehler der Maschinenoperationen ist* $\leq$ *eps.*

Beweis: Klar nach Satz 1.2.1.

□

Das Problem der Rundungsfehleranalyse liegt darin, daß Satz 1.2.2 für mehr als zwei Operanden nicht mehr gilt.

Beispiel: $m = 2$.

$$0.75 + 0.055 - 0.80 = 0.005$$

In Maschinenarithmetik wird daraus

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

oder

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

In jedem Fall bekommt man einen relativen Fehler von 100%! Überdies sieht man: Maschinenoperationen sind nicht assoziativ!

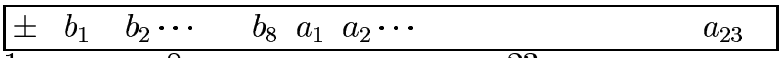
In dem Beispiel sind zwei Dinge passiert:

1. Es ist zunächst ein kleiner Fehler entstanden.
2. Dieser Fehler wird durch eine nachfolgende Addition/Subtraktion verstärkt.

Die durch Subtraktion nahezu gleich großer Zahlen verursachte Verstärkung eines Fehlers nennt man Auslöschung (= Auslöschung korrekter Dezimalstellen). Vermeidung von Auslöschung ist ein fundamentales Ziel der Numerik.

Wir haben einen unendlichen Exponentenbereich angenommen. Reale Maschinen haben natürlich einen endlichen Exponentenbereich. Dadurch kann es zu einem overflow (Exponent zu groß) oder zu einen underflow (Exponent zu klein) kommen. Die Maschine muß darauf irgendwie reagieren (exception handling).

Einige Angaben zu IEEE 754 single precision Gleitkommaarithmetik:



Der dargestellte Wert (Dualsystem) ist
$\pm\ 1.\, a_1\, a_2 \cdots a_{23} \cdot\ \ 2^{b-127}\qquad ,\quad b = b_1 \cdots b_8.$

Die Maschinengenauigkeit ergibt sich zu
$eps = 2^{-24} = 5.96 \cdot 10^{-8}$

Exception handling:

$1/0$	∞
overflow	∞
$0/0, 0 \cdot \infty, \sqrt{-1}$	NaN (not a number)
underflow	

Kapitel 2

Lineare Gleichungssysteme

2.1 Das Gauß'sche Eliminationsverfahren

A reelle (n,n)-Matrix, $b \in \mathbb{R}^n$. Gesucht ist $x \in \mathbb{R}^n$ mit $Ax = b$. Eindeutig lösbar falls $\det(A) \neq 0$. Ausführlich:

$$\begin{aligned} a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,n}x_n &= b_1 \\ a_{2,1}x_1 + a_{2,2}x_2 + \cdots + a_{2,n}x_n &= b_2 \\ &\cdots \\ a_{n,1}x_1 + a_{n,2}x_2 + \cdots + a_{n,2}x_n &= b_n \end{aligned}$$

Elimination von x_1 aus den Gleichungen $2,\ldots,n$: Sei $a_{1,1} \neq 0$. Multipliziere die 1-te Gleichung mit $\ell_{i,1} = a_{i,1}/a_{1,1}$ und subtrahiere die entstandene Gleichung von der i-ten, $i = 2,\ldots,n$:

$$(a_{i,2} - \ell_{i,1}a_{1,2})x_2 + \cdots + (a_{i,n} - \ell_{i,1}a_{n,1})x_n = b_i - \ell_{i,1}b_1$$

Dies ist ein System von $n-1$ Gleichungen in $x_2,\ldots,x_n$. Wir schreiben es als

$$\begin{aligned} a^{(2)}_{2,2}x_2 + a^{(2)}_{2,3}x_3 + \cdots + a^{(2)}_{2,n}x_n &= b^{(2)}_2 \\ a^{(2)}_{3,2}x_2 + a^{(2)}_{3,3}x_3 + \cdots + a^{(2)}_{3,n}x_n &= b^{(2)}_3 \\ &\cdots \\ a^{(2)}_{n,2}x_2 + a^{(2)}_{n,3}x_3 + \cdots + a^{(2)}_{n,n}x_n &= b^{(2)}_n \end{aligned}$$

$$a^{(2)}_{i,j} = a^{(1)}_{i,j} - \ell_{i,1}a^{(1)}_{1,j}\ , \quad b^{(2)}_i = b^{(1)}_i - \ell_{i,1}b^{(1)}_1\ .$$

Zur Vereinheitlichung haben wir $a^{(1)}_{i,j} = a_{i,j}$, $b^{(1)}_i = b_i$ gesetzt.

Elimination von x_2 aus den Gleichungen $3,\ldots,n$: Sei $a^{(2)}_{2,2} \neq 0$. Mit $\ell_{i,2} = a^{(2)}_{i,2}/a^{(2)}_{2,2}$ erhalten wir wie oben für $i = 3,\ldots,n$:

$$\left(a^{(2)}_{i,3} - \ell_{i,2}a^{(2)}_{2,3}\right)x_3 + \cdots + \left(a^{(2)}_{i,n} - \ell_{i,2}a^{(2)}_{2,n}\right)x_n = b^{(2)}_i - \ell_{i,2}b^{(2)}_2$$

Dies ist ein System von $n-3$ Gleichungen in $x_3, \ldots, x_n$. Wir schreiben es als

$$\begin{array}{rcl} a_{3,3}^{(3)} x_3 + a_{3,4}^{(3)} x_4 + \cdots + a_{3,n}^{(3)} x_n &=& b_3^{(3)} \\ a_{4,3}^{(3)} x_3 + a_{4,4}^{(3)} x_4 + \cdots + a_{4,n}^{(3)} x_n &=& b_4^{(3)} \\ & \cdots & \\ a_{n,3}^{(3)} x_2 + a_{n,4}^{(3)} x_4 + \cdots + a_{n,n}^{(3)} x_n &=& b_n^{(3)} \end{array}$$

$$a_{i,j}^{(3)} = a_{i,j}^{(2)} - \ell_{i,2} a_{2,j}^{(2)} \,, \quad b_i^{(3)} = b_i^{(2)} - \ell_{i,2} b_2^{(2)} \,.$$

So macht man weiter, bis man bei einer Gleichung in der Unbekannten x_n ankommt.

$$a_{n,n}^{(n)} x_n = b_n^{(n)}$$

$$a_{n,n}^{(n)} = a_{n,n}^{(n-1)} - \ell_{n,n-1} a_{n-1,n}^{(n-1)} \,, \quad b_n^{(n)} = b_n^{(n-1)} - \ell_{n,n-1} b_{n-1}^{(n-1)} \,,$$

$$\ell_{n,n-1} = a_{n,n-1}^{(n-1)} / a_{n-1,n-1}^{(n-1)} \,.$$

Damit ist der Elminationsprozeß abgeschlossen. Es folgt das Rückwärtseinsetzen:

$$\begin{array}{lcl} x_n &=& b_n^{(n)} / a_{n,n}^{(n)}, \\ x_{n-1} &=& \left(b_{n-1}^{(n-1)} - a_{n-1,n}^{(n-1)} \, x_n \right) / a_{n-1,n-1}^{(n-1)} \\ & \cdots & \\ x_2 &=& \left(b_2^{(2)} - a_{2,3}^{(2)} x_3 - \cdots - a_{2,n}^{(2)} x_n \right) / a_{2,2}^{(2)} \\ x_1 &=& \left(b_1^{(1)} - a_{1,2}^{(1)} x_2 - \cdots - a_{1,n}^{(1)} x_n \right) / a_{1,1}^{(1)} \end{array}$$

Dies ist das vollständige Eliminationsverfahren. Es läßt sich durchführen, wenn die "Pivotelemente" $a_{i,i}^{(i)} \neq 0$ sind.

Pseudocode für das Eliminationsverfahren:

elim (A, b, x, n)

```
{  for  j = 1,...,n
   {    for  i = j+1,...,n
        {  ℓ = a_{i,j} / a_{j,j};
           for  k = j+1,...,n  a_{ik} = a_{ik} - ℓ * a_{j,k};
           b_i = b_i - ℓ * b_j;
        }
   }
   for  j = n,...,1
   {    x_j = b_j ;
        for  k = j+1,...,n  x_j = x_j - a_{j,k} * x_k;
        x_j = x_j / a_{j,j};
   }
}
```

Anzahl der Rechenoperationen (1 flop = 1 Mult./Div. + 1 Add./Sub.):
K_j = Anzahl der flops für j-ten Eliminationsschritt:

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

Ganzer Eliminationsprozeß:

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

Rückwärtseinsetzen: $\frac{1}{2}n^2 + \mathrm{O}(n)$.

Satz 2.1.1 *Das Gauß'sche Eliminationsverfahren kann in* $\frac{1}{3}n^3 + O(n^2)$ *flops durchgeführt werden.*

Rundungsfehler:
Die einzige rundungsfehlerrelevante Operation ist $a_{i,k} - \ell_{i,j}a_{j,k}$, $\ell_{i,j} = a_{i,j}/a_{j,j}$. Wir nehmen an, daß zu Beginn der Rechnung alle $a_{i,k}$ Größenordnung 1 haben. Auslöschung tritt dann genau dann auf, wenn ein $a_{i,k}$ klein wird. Wir unterscheiden drei Fälle.

1) Für das kleine a_{ik} ist $k > j$. Dann kann $a_{i,k} - \ell_{i,j}a_{j,k}$ trotz des großen relativen Fehlers von $a_{i,k}$ noch genau berechnet werden. Die Auslöschung ist harmlos.

2) Für das kleine Element $a_{i,k}$ ist $k = j$ und $i > j$. Dann ist $\ell_{i,j}$ klein und hat großen relativen Fehler. $a_{i,k} - \ell_{i,j}a_{j,k}$ kann aber trotzdem genau berechnet werden. Die Auslöschung ist harmlos.

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

Um Fall 3) zu vermeiden, vertauscht man vor dem j-ten Eliminationsschritt die Zeile j mit der Zeile $i \geq j$, für welche $|a_{i,j}/a_{j,j}|$ möglichst groß wird. Man spricht von maximaler Spaltenpivotsuche.

2.2 Die LR-Zerlegung

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

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

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

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

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

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

Zur Herstellung der LR-Zerlegung verwenden wir die Elementarmatrizen.

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

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

1) Anwendung auf einen Vektor:

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

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

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

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

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

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

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

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

1) Anwendung auf einen Vektor:

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

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

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

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

Zum Schluß wollen wir noch das Zusammenspiel von Elementar- und Permutationsmatrizen betrachten. Sei P eine Permutationsmatrix, welche nur Zeilen $>j$ vertauscht, d.h. $i_1=1,\ldots,i_j=j$. Dann gilt $PL_jP^*=L_j'$, wobei

L'_j die gleiche Form wie L_j hat, aber mit (gemäß der Permutation) vertauschten Elementen $\ell'_{k,j} = \ell_{i_k,j},\ k > j$. Dies sieht man sofort, wenn man

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

schreibt und die Wirkung von Links- bzw. Rechtsmultiplikationen mit P und P^* studiert. Es folgt die Vertauschungsrelation $PL_j = L'_j P$. Wir können nun das Eliminationsverfahren durch rekursive Linksmultiplikation der Matrix (A, b) mit Permutations- und Elementarmatrizen darstellen. Sei P_j die Permutationsmatrix, welche die Zeilenvertauschung vor dem j-ten Eliminationsschritt ausführt. P_j vertauscht also nur die Zeilen $j, \ldots, n$ untereinander. Sei L_j die Elementarmatrix mit $\ell_{i,j} = a_{i,j}/a_{j,j}$, wobei $a_{i,j}$ nun das (i,j)-Element vor Beginn der Elimination im j-ten Eliminationsschritt, also unmittelbar nach Anwendung von P_j ist. Das Eliminationsverfahren lautet dann

$$L_{n-1}P_{n-1} \cdots L_2 P_2 L_1 P_1 (A, b) = (R, y)\ .$$

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

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

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

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

Satz 2.2.1 *Zu jeder (n,n)-Matrix A gibt es eine Permutationsmatrix P, so daß PA eine LR-Zerlegung mit $\ell_{i,i} = 1,\ i = 1, \ldots, n$ hat.*

Bemerkungen:

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

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

3) Die LR-Zerlegung von A kann mit Hilfe eines leicht abgeänderten Programms elim hergestellt werden. Die Matrix L erhält man, indem man die ℓ's als $\ell_{i,j}$ speichert. R ist die durch die erste j-Schleife umgeformte Matrix A. P erhält man als Produkt aller ausgeführten Zeilenvertauschungen.

4) Hat man mehrere Systeme $Ax = b^{(i)}$, $i = 1, \ldots, m$ mit ein und derselben Matrix A zu lösen, braucht man die LR-Zerlegung nur einmal auszuführen. Danach kann man jedes System durch $n^2 + \mathrm{O}(n)$ flops lösen.

2.3 Die Cholesky-Zerlegung

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

Wir wollen annehmen, die positive definite Matrix A besitze eine LR - Zerlegung. Sei also $A = LR$ und $\ell_{i,i} = 1$, $i = 1, \ldots, n$. Dann ist $A^* = R^* L^*$, also $A = R^* L^*$ eine weitere LR-Zerlegung von A. Sei D die Diagonale von R. Dann ist $A = R^*(D^*)^{-1} D^* L^*$ eine LR-Zerlegung von A, deren linker Faktor nur 1′sen auf der Hauptdiagonalen hat. Nach der Eindeutigkeit der LR-Zerlegung (Aufgabe 10) ist also $L = R^*(D^*)^{-1}$ und $D^* L^* = R$. Also ist $A = L D^* L^*$. Da A positiv definit ist müssen die nichttrivialen Elemente von D positiv sein. Nach geeigneter Fixierung der Diagonalen von L kann man also $R = L^*$ annehmen, und wir kommen zu

$$A = LL^* \,. \tag{3.1}$$

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

Satz 2.3.1 *Sei A positiv definit. Dann gibt es genau eine linke Dreiecksmatrix L mit positiven Diagonalelementen, so daß $A = LL^*$.*

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

$$\sum_{k=1}^{j} \ell_{i,k} \overline{\ell}_{j,k} = a_{i,j} \quad , \quad n \geq i \geq j \geq 1 \,; \tag{3.2}$$

für $K = \mathbf{R}$ kann das „ - " - Zeichen natürlich wegfallen. Das nichtlineare Gleichungssystem (3.2), bestehend aus $n(n+1)/2$ Gleichungen für ebenso viele Unbekannte, läßt sich rekursiv auflösen. Die Gleichungen für $j = 1$ lauten

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

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

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

Damit ist die erste Spalte von L bestimmt.

Die Gleichungen (3.2) für $j = 2$ lauten

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

Für $i = 2$ ergibt sich

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

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

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

Damit ist auch die zweite Spalte von L bestimmt.

Wir wollen nun annehmen, Spalten $1, \ldots, j-1$ von L seien bereits bestimmt. Dann schreiben wir die Gleichung (3.2) für die j-te Spalte hin, also

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

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

$$\ell_{j,j} = \sqrt{a_{j,j} - |\ell_{j,1}|^2 - \cdots - |\ell_{j,j-1}|^2} \ , \tag{3.3}$$

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

$$\ell_{i,j} = (a_{ij} - \ell_{i,1}\overline{\ell}_{j,1} - \cdots - \ell_{i,j-1}\overline{\ell}_{j,j-1})/\ell_{j,j} \ . \tag{3.4}$$

Die auf der rechten Seite von (3.3), (3.4) auftretenden Elemente von L stehen bis auf $\ell_{j,j}$ alle in den bereits berechneten Spalten von L, und $\ell_{j,j}$ ergibt sich aus (3.3). Wir sehen, daß alle Elemente von L berechnet werden können, wenn nur der Radikand in (3.3) immer positiv ausfällt. Dies wollen wir nun durch vollständige Induktion zeigen. Für $j = 1$ ist dies richtig, weil $a_{1,1} > 0$. Sei es richtig bis zu $j - 1$. Dann lassen sich die Spalten $1, \ldots, j-1$ von L berechnen. Diese bilden die $(n, j-1)$-Matrix L_j, und es gilt

$$L_j L_j^* = \begin{pmatrix} a_{1,1} & & & & & \\ & \ddots & & & & \\ a_{j-1,1} & \cdots & a_{j-1,j-1} & & & \\ a_{j,1} & \cdots & a_{j,j-1} & x_{jj} & & \\ \vdots & & & & \ddots & \\ a_{n,1} & \cdots & a_{n,j-1} & x_{n,j} & \cdots & x_{n,n} \end{pmatrix} . \tag{3.5}$$

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

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

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

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

(wieder haben wir nur die linke Hälfte notiert) positiv definit, denn dies wäre ja schon für $x_{j,j} = a_{j,j}$ richtig, um so mehr also für $x_{j,j} > a_{j,j}$. Damit hätte aber die rechte Seite von (3.5) mindestens den Rang j, während die linke Seite als Produkt von Matrizen mit höchstens $j-1$ Zeilen bzw. Spalten höchstens den Rang $j-1$ haben kann. Es kann also $x_{j,j} \geq a_{j,j}$ nicht richtig sein. Damit ist $x_{j,j} < a_{j,j}$ für $j = 1, \ldots, n$, d.h. der Radikand in (3.3) ist immer positiv, und L läßt sich in eindeutiger Weise bestimmen.

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

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

{ for  j = 1, ..., n
       for   i = j, ..., n
          {  s = a_{i,j} − a_{i,1} * ā_{j,1} − ··· − a_{i,j−1} * ā_{j,j−1}  ;
             if  (i = j)
                 { if (s ≤ 0)  { print ("Matrix nicht pos. def.")   ;
                                    exit (1)   ;
                               }
                   else a_{j,j} = sqrt (s) ;
                 }
             else  a_{i,j} = s/a_{j,j}  ;
          }
}
```

Für die Anzahl der benötigten flops findet man

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

Dies entspricht dem Eliminationsverfahren für hermitesche Matrizen (vergleiche Übungsaufgabe 7). Das Gleichungssystem $Ax = b$ kann nach Berechnung von L gelöst werden durch Lösung der Systeme $Ly = b$, $L^*x = y$ wie nach der LR-Zerlegung.

2.4 Die QR-Zerlegung

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

$$\mathrm{sp}(a_1, \ldots, a_j) = \mathrm{sp}(q_1, \ldots, q_j)\,, \quad j = 1, \ldots, m \tag{4.1}$$

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

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

mit der euklidischen Norm. Damit ist (4.1) für $j = 1$ erfüllt. Haben wir bereits orthonormale Vektoren $q_1, \ldots, q_{j-1}$ bestimmt, so setzen wir

$$\hat{q}_j = a_j - r_{1,j}q_1 - \cdots - r_{j-1,j}q_{j-1} \tag{4.2}$$

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

$$r_{i,j} = (a_j, q_i) \; . \tag{4.3}$$

Dann setzen wir

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

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

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

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

$$A = QR \; . \tag{4.4}$$

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

Numerisch ist das Schmidtsche Verfahren völlig ungeeignet. Nach (4.2), (4.3) ist nämlich

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

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

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

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

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

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

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

Das Householder - Verfahren bestimmt Spiegelungen $S_1, \ldots, S_m$, so daß $S_j \cdots S_1 A$ in den Spalten $1, \ldots, j$ bereits rechte Dreiecksgestalt hat. Wir beschreiben ausführlich die Bestimmung von S_1. Die erste Spalte von $S_1 A$ lautet $S_1 a_1$. Wir müssen S_1 also so bestimmen, daß $S_1 a_1$ ein Vielfaches von e_1 ist. Dies kann man auf zwei Weisen erreichen, nämlich
durch Spiegelung an v_1, wo $v_1 = (a_1 + \alpha_1 e_1)/\beta_1$, $\beta_1 = \|a_1 + \alpha_1 e_1\|$ mit $\alpha_1 = \pm \|a_1\|$. Zur Vermeidung von Auslöschung wählt man $\alpha_1 = \|a_1\| \; \mathrm{sgn}\, (a_{1,1})$. Mit

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

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

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

Die weiteren Spalten sind

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

$S_1 A$ hat die Gestalt

$$S_1 A = \begin{pmatrix} -\alpha_1 & r_{1,2} & \cdots & r_{1,m} \\ 0 & & & \\ \vdots & & A_2 & \\ 0 & & & \end{pmatrix} ,$$

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

$$S_2 = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & & & \\ \vdots & & I - 2v_2 v_2^* & \\ 0 & & & \end{pmatrix} ,$$

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

$$S_2 S_1 A = \begin{pmatrix} -\alpha_1 & r_{1,2} & \cdots & & r_{1,m} \\ 0 & -\alpha_2 & r_{2,3} & \cdots & r_{2,m} \\ \vdots & 0 & & & \\ & \vdots & & A_3 & \\ 0 & 0 & & & \end{pmatrix}$$

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

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

mit der rechten Dreiecksmatrix

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

Es folgt

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

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

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

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

wobei x^{j-1}, x^{n-j+1} die Längen $j-1$ bzw. $n-j+1$ haben. Dies verlangt nur $2(n-j-1)+\mathrm{O}(1)$ flops. Die sukzessive Berechnung von $Sx = S_1 \cdots S_m x$ erfordert daher etwa für $n=m$ nur $m^2+\mathrm{O}(m)$ flops, also ebenso viele wie die Berechnung von Sx bei vorberechnetem S. Ein Programm zur QR-Zerlegung berechnet also zweckmäßigerweise nicht Q, sondern die Spiegelungsvektoren $v_1,\ldots,v_m$.

$qr\_$ dcmp (A,α,n,m)

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

{ for $j=1,\ldots,m$
{ $\alpha_j = \|a_j\| *$ sgn $(a_{j,j})$; $\beta = \sqrt{2\alpha_j * (\alpha_j + a_{j,j})}$;
$a_{j,j} = (a_{j,j}+\alpha_j)/\beta$;
for $i = j+1,\ldots,n$ $a_{i,j} = a_{i,j}/\beta$;
for $k = j+1,\ldots,m$
{ $\gamma = 2 * (a_{j,k} * a_{j,j} + \cdots + a_{n,k} * a_{n,j})$;
for $i = j,\ldots,n$ $a_{i,k} = a_{i,k} - \gamma * a_{i,j}$;
}
}
}

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

$q\_$ mal $\_x(A,x,n,m)$
/* Überschreibt x mit Sx nach Aufruf von $qr\_$ dcmp (A,α,n,m) * /
{ for $j=m,\ldots,1$
{ $\gamma = 2 * (a_{j,j} * x_j + \cdots + a_{n,j} * x_n)$;
for $i = j,\ldots,n$ $x_i = x_i - \gamma * a_{i,j}$;
}
}
$q^*\_$ mal$_x(A,x,n,m)$
/* Überschreibt x mit S^*x nach Aufruf von $qr\_$ dcmp (A,α,n,m) * /
{ Wie oben, aber $j=1,\ldots,m$}

2.5 Fehlerabschätzung bei linearen Gleichungssystemen

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

Beispiel:

Löse $Ax = b$ mit

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

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

$$\tilde{A} = \begin{pmatrix} 1.01 & 1.01 \\ 1 & 0.99 \end{pmatrix} \quad , \quad \tilde{b} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} .$$

Die Lösung $\tilde{x}$ von $\tilde{A}\tilde{x} = \tilde{b}$ ist

$$\tilde{x} = \begin{pmatrix} 200/101 \\ -100/1001 \end{pmatrix} .$$

Obwohl wir also einen Fehler von nur 0.1% in den Daten haben, bekommen wir einen Fehler von über 100% in der Lösung.

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

Definition 2.5.1 *Eine Abbildung $\| \cdot \| : V \to \mathbb{R}^{\geq 0}$ eines $\mathbb{C}$-Vektorraums V in die nichtnegativen reellen Zahlen heißt Norm, falls für alle x, $y \in V$, $\alpha \in \mathbb{C}$ gilt*

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

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

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

Beispiele:

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

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

b) ∞-Norm: $\quad ||x||_\infty := \max_k |x_k|$

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

d) Sei $||\cdot||$ eine Norm und T eine nichtsinguläre (n,n)-Matrix. Dann ist auch

$$||x||_T = ||Tx||$$

eine Norm.

Wir wollen nun spezielle Normen im Vektorraum der Matrizen definieren.

Definition 2.5.2 *Sei $||\cdot||$ eine Norm in $\mathbb{C}^n$. Dann heißt*

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

die zugeordnete Matrixnorm.

Bemerkungen:
1) Es gilt $||A|| = \sup_{x\in\mathbb{C}^n} \frac{||Ax||}{||x||} = \sup_{x\in\mathbb{C}^n} ||A\frac{x}{||x||}|| = \sup_{||x||=1,\, x\in\mathbb{C}^n} ||Ax||$.

2) $||\cdot||$ ist eine Norm im Vektorraum der Matrizen.

3) Sei λ Eigenvektor von A. Dann gilt $||A|| \geq |\lambda|$.

4) $||AB|| \leq ||A||||B||$.

Beispiele:

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

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

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

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

Dann gilt $||\tilde{x}||_\infty = 1$, und es ist

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

Also gilt $||A|| = \max_i \sum_{k=1}^n |a_{ik}|$. Für $A = 0$ ist dies klar.

b) Euklidische Norm: Es gilt $||A||_2 = \rho(A^*A)^{1/2}$, wobei $\rho(X)$ der Spektralradius von X, d. h. der Betrag des betragsmäßig größten Eigenwerts von X ist. Insbesondere ist für hermite'sche Matrizen (d.h. $A = A^*$) $||A||_2 = \rho(A)$.

c) 1-Norm: $\quad ||A||_1 = \max_j \sum_i |a_{ij}|$.

d) T–Norm:

$$\begin{aligned} ||A||_T &= \sup_{x \in \mathbb{C}^n} \frac{||TAx||}{||Tx||} = \sup_{Ty \in \mathbb{C}^n} \frac{||TAT^{-1}y||}{||TT^{-1}y||} \\ &= \sup_{y \in \mathbb{C}^n} \frac{||TAT^{-1}y||}{||y||} = ||TAT^{-1}|| \, . \end{aligned}$$

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

Satz 2.5.1 *Zu jeder Matrix $A \in \mathbb{C}^{(n,n)}$ und jedem $\varepsilon > 0$ existiert eine Norm $|| \cdot ||_{A,\varepsilon}$ auf dem $\mathbb{C}^n$, so daß für die zugeordnete Matrixnorm*

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

gilt.

Beweis: Sei

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

Bei der Bildung von BD für eine Matrix B wird Spalte k von B mit ε^{k-1} multipliziert. Bei der Bildung von $D^{-1}B$ wird Zeile k von B mit ε^{-k+1} multipliziert.

Sei nun $J = P^{-1}AP$ die Jordansche Normalform von A. J hat die Form

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

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

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

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

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

□

Definition 2.5.3 *Sei $A \in \mathbb{C}^{(n,n)}$ invertierbar. Dann heißt $k(A) = ||A||||A^{-1}||$ die Kondition von A bezüglich $||\cdot||$.*

Satz 2.5.2 *Sei $A \in \mathbb{C}^{n,n}$ invertierbar, und $\Delta A \in \mathbb{C}^{(n,n)}$ eine Matrix mit $\frac{||\Delta A||}{||A||}k(A) < 1$. Dann gilt:*

a) *$(A + \Delta A)$ ist invertierbar,*

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

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

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

Beweis:

zu a) Es gilt für $y \neq 0$

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

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

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

Für jedes x gilt damit

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

oder

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

Damit gilt

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

zu b) Wir betrachten zunächst die Gleichungssysteme

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

Durch Subtraktion erhalten wir

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

und damit

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

Mit den Norm–Rechenregeln gilt

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

□

Wir können den Satz so interpretieren:

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

Wir wollen nun unser einführendes Beispiel aufklären. Es galt:

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

Wir erhalten:

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

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

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

Mit dere Maschinengenauigkeit eps ist dann

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

Die Anwendung des Satzes verlangt nun zunächst einmal

$$k(A)\ \text{eps}\ < 1 . \tag{5.1}$$

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

$$\frac{||\Delta x||}{||x||} \sim k(A)\ \text{eps} . \tag{5.2}$$

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

2.6 Unter- und überbestimmte lineare Systeme

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

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

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

Weiter bezeichnen wir für lineare Unterräume U, V von K^n mit $U + V$ die Summen von Vektoren aus U, V. Ist $U \perp V$, so schreiben wir für $U+V$ auch $U \oplus V$. Für später notieren wir, daß $ker(A) = ker(A^*A)$.

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

Satz 2.6.1 *Für jede (n,m)-Matrix A gilt*

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

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

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

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

□

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

Satz 2.6.2 *$||Ax - b||$ nimmt sein Minimum auf K^m an. Es nimmt genau dann für $x = x_0$ sein Minimum an, wenn $A^*Ax_0 = A^*b$.*

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

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

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

□

Bemerkungen:

1) $A^*Ax = A^*b$ heißt System der Normalgleichungen. x^0 heißt Kleinste-Quadrate-Lösung. Man spricht auch von der (Gaußschen) Methode der kleinsten Quadrate.
2) Die Normalgleichungen sind immer lösbar. Dies folgt natürlich aus dem eben bewiesenen Satz. Man kann es aber auch direkt bestätigen. Es ist ja $ker(A) = ker(A^*A)$, also

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

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

Die Kleinste-Quadrate-Lösung von $Ax = b$ existiert also immer. Leider ist sie i. allg. nicht eindeutig.

Definition 2.6.1 *x^+ heißt verallgemeinerte (Moore-Penrose-) Lösung von $Ax = b$, wenn*
1) x^+ ist Kleinste-Quadrate-Lösung.
2) Unter allen Kleinste-Quadrate-Lösungen hat x^+ minimale Norm.

Satz 2.6.3 *x^+ existiert und ist eindeutig bestimmt. x^+ ist genau dann verallgemeinerte Lösung, wenn*

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

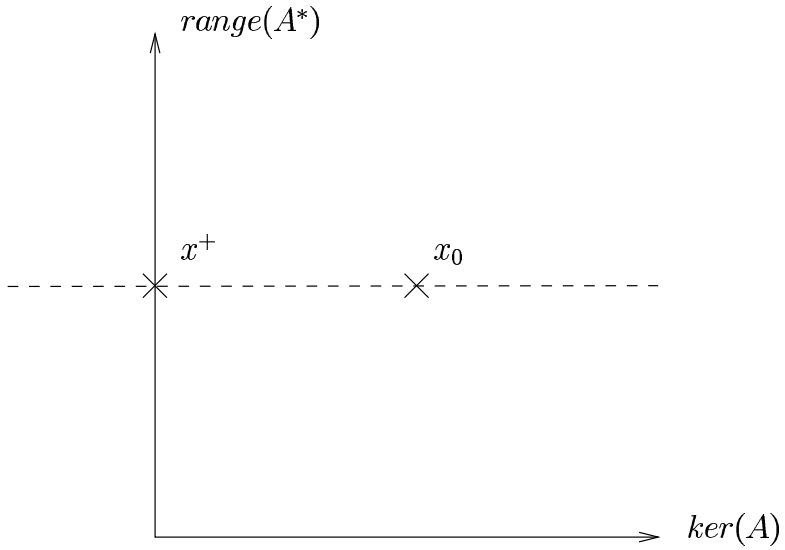
Beweis: Sei x_0 Kleinste-Quadrate-Lösung, also $A^*Ax_0 = A^*b$. Nach Satz 1, angewandt auf A^*, ist $x_0 = x_1 + x_2$ mit $x_1 \in range(A^*)$, $x_2 \in ker(A)$. Auch x_1 erfüllt die Normalgleichungen und ist damit Kleinste-Quadrate-Lösung. Es gibt also immer eine Kleinste-Quadrate-Lösung x_1 in $range(A^*)$.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

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

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

□

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



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

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

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

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

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

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

1) Wir betrachten das Beispiel

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

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

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

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

und damit

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

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

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

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

oder, da R^* vollen Rang hat,

$$Rx^+ = Q^*b \, . \tag{6.1}$$

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

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

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

schreiben sich dann als

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

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

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

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

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

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

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

Kapitel 3

Nichtlineare Gleichungen

3.1 Existenz von Lösungen

Sei $f : K^n \to K^n$ eine Abbildung. Gesucht ist ein $\overline{x} \in K^n$ mit $f(\overline{x}) = 0$.

Beispiele:

1) $f(x) = x^2 - 2px - q$ für $K = \mathbb{C}$ oder $K = \mathbb{R}$. Sei $d = p^2 + q$. Im komplexen Fall gibt es zwei Lösungen für $d \neq 0$, eine Lösung für $d = 0$. Im reellen Fall gibt es für $d > 0$ zwei Lösungen, für $d = 0$ eine, für $d < 0$ überhaupt keine. Die Berechnung der Lösung $\overline{x}$ kann durch die Formel

$$x_{1,2} = p \pm \sqrt{d}$$

erfolgen. Die Auswertung dieser Formel ist aber im Hinblick auf Rundungsfehler keineswegs harmlos. Ist etwa $p > 0$, $|q| \ll p$, so ist $\sqrt{d} \sim p$, und bei der Berechnung von x_2 tritt Auslöschung auf. In diesem Fall ist es besser, x_2 nach der Formel $x_2 = -q/x_1$ zu berechnen.

2) $f(x) = x^3 + 3px - 2q$ für $K = \mathbb{C}$. Die Berechnung der - maximal drei - Lösungen kann durch die Cardani'schen Formeln erfolgen:

$$\begin{array}{rclcrclcrcl} x_1 & = & u + v & , & x_2 & = & \varepsilon_1 u + \varepsilon_2 v & , & x_3 & = & \varepsilon_2 u + \varepsilon_1 v \quad , \\ u & = & (q + \sqrt{d})^{1/3} & , & v & = & (q - \sqrt{d})^{1/3} & , & d & = & p^3 + q^2 \quad , \end{array}$$

$$\varepsilon_{1,2} = -\frac{1}{2}(1 \pm i\sqrt{3}) \; .$$

Auch hier tritt unter Umständen, z. B. für $|p| \ll |q|$, Auslöschung auf.

3) $f(x) = x - \tan x$, $K = \mathbb{R}$. Ein Blick auf den Graphen von $\tan x$ zeigt, daß es in jedem Intervall $((k - \frac{1}{2})\pi, (k + \frac{1}{2})\pi)$, $k \in \mathbb{Z}$, genau eine Lösung gibt.

Ein primitives Verfahren zur Lösung von Gleichungen in $\mathbb{R}^1$ ist die Intervallhalbierung. Sei $f : \mathbb{R} \to \mathbb{R}$ stetig und $f(a)f(b) < 0$, $a < b$. Dann liegt in (a, b) sicher eine Nullstelle von f. Zu ihrer Berechnung verwenden wir den Algorithmus

$f_a = f(a)$; $f_b = f(b)$;
while $(b - a \geq \varepsilon)$
{ $c = (b + a)/2$;
$f_c = f(c)$;
if $(f_a f_c < 0)$ $\{b = c; f_b = f_c;\}$
else $\{a = c; f_a = f_c;\}$
}

Nach $n + 2$ Funktionsauswertungen hat er das Intervall (a, b), in dem eine Lösung liegt, um den Faktor 2^n verkürzt. Für einfache Funktionen f ist dies akzeptabel. Wir werden natürlich effizientere Methoden kennenlernen.

Das theoretische Hilfsmittel zur Lösung nichtlinearer Gleichungen sind Fixpunktsätze. $x \in \mathbb{R}^n$ heißt Fixpunkt einer Abbildung $g : \mathbb{R}^n \to \mathbb{R}^n$, falls $g(x) = x$. Jede Aufgabe $f(x) = 0$ läßt sich in der Form $g(x) = x$ schreiben. Man braucht ja nur $g(x) = f(x) + x$ zu setzen.

Satz 3.1.1 *(Fixpunktsatz von Brouwer): Sei $D \subseteq \mathbb{R}^n$ konvex und kompakt, $g : D \to D$ stetig. Dann besitzt g in D einen Fixpunkt.*

Beweis: Der Beweis für $n \geq 1$ findet sich in E. Burger, Einführung in der Theorie der Spiele, 2. Auflage, S. 162-165. Für $n = 1$ ist der Satz trivial: Sei $D = [a, b]$. Ist $g(a) = a$ oder $g(b) = b$, so besitzt g einen Fixpunkt. Andernfalls ist $g(a) > a$, $g(b) < b$. Die Funktion $f(x) = x - g(x)$ hat dann in $[a, b]$ einen Zeichenwechsel und damit eine Nullstelle, und diese ist Fixpunkt von g.

□

Beispiel:

In $\mathbb{R}^2$ betrachten wir

$$g\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \sin(x_2 + e^{x_1}) \\ \cos(x_1 - e^{x_2}) \end{pmatrix}$$

Die Menge $D = \{x \in \mathbb{R}^2 : \|x\|_\infty \leq 1\}$ ist konvex und kompakt, und g ist dort stetig. Offenbar ist $g(D) \subseteq D$. Also hat g in D einen Fixpunkt.

3.2 Iterationsverfahren

Iterationsverfahren gehören zu den wichtigsten Hilfsmitteln der Numerischen Mathematik. Sie berechnen mittels einer - oft sehr einfachen - Rekursionsformel eine Folge von Näherungen, welche gegen die gesuchte Lösung konvergiert. Grundlage vieler Iterationsverfahren ist der Fixpunktsatz für kontrahierende Abbildungen.

Definition 3.2.1 *Sei $D \subseteq K^n$ und $g : D \to K^n$ eine Abbildung. g heißt kontrahierend in D (bezüglich der Norm $\|\cdot\|$ in K^n), wenn es eine Konstante $q < 1$ gibt mit*

$$\|g(x) - g(y)\| \leq q\|x - y\|$$

für alle $x, y \in D$. q heißt Lipschitzkonstante von g in D.

Satz 3.2.1 *Sei $D \subseteq \mathbb{R}^n$ konvex und $g : D \to \mathbb{R}^n$ differenzierbar. Sei*

$$q = \sup_{x \in D} \|g'(x)\| < 1 \, .$$

Dann ist g in D (bezüglich $\|\cdot\|$) kontrahierend. Dabei ist g' die Jacobi-Matrix zu g, also

$$g' = \begin{pmatrix} \frac{\partial g_1}{\partial x_1} & , \ldots , & \frac{\partial g_1}{\partial x_n} \\ \vdots & & \\ \frac{\partial g_n}{\partial x_1} & , \ldots , & \frac{\partial g_n}{\partial x_n} \end{pmatrix} \quad \textit{für} \quad g = \begin{pmatrix} g_1 \\ \vdots \\ g_n \end{pmatrix} .$$

Beweis: Sei $f(t) = g(tx + (1-t)y)$, $0 \leq t \leq 1$. Nach der Kettenregel ist

$$f'(t) = g'(tx + (1-t)y)(x - y) \, ,$$

also

$$\begin{aligned} \|g(x) - g(y)\| &= \|f(1) - f(0)\| = \| \int_0^1 f'(t)dt \| \\ &\leq \sup_{0 \leq t \leq 1} \|f'(t)\| \\ &= \sup_{0 \leq t \leq 1} \|g'(tx + (1-t)y)(x - y)\| \\ &\leq q\|x - y\| \, , \end{aligned}$$

weil D konvex ist.

□

Satz 3.2.2 *(Kontraktionssatz, Fixpunktsatz von Banach): Sei $D \subseteq K^n$ abgeschlossen und $g : D \to D$ kontrahierend. Dann hat g in D genau einen Fixpunkt $\overline{x}$. Das Iterationsverfahren*

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

konvergiert für jede Wahl von $x^0 \in D$ gegen $\overline{x}$, und es gilt mit der Lipschitz-Konstanten q von g in D

$$\|x^k - \overline{x}\| \leq \frac{q^k}{1-q} \|x^1 - x^0\| \,.$$

Beweis:

1) Existenz von $\overline{x}$.

Es ist $x^k \in D$ falls $x^0 \in D$, und

$$\|x^{k+1} - x^k\| \leq q\|x^k - x^{k-1}\| \leq q^2\|x^{k-1} - x^{k-2}\| \leq \ldots \leq q^k\|x^1 - x^0\| \,,$$

also für $\ell > j$

$$\begin{aligned}
\|x^\ell - x^j\| &= \|\sum_{k=j}^{\ell-1}(x^{k+1} - x^k)\| \\
&\leq \sum_{k=j}^{\ell-1} \|x^{k+1} - x^k\| \\
&\leq \sum_{k=j}^{\ell-1} q^k \|x^1 - x^0\| \\
&= q^j(1 + \cdots + q^{\ell-1-j})\|x^1 - x^0\| \\
&\leq \frac{q^j}{1-q} \|x^1 - x^0\| \qquad (2.1)
\end{aligned}$$

wegen der Formel für die geometrische Reihe. Also gilt $x^\ell - x^j \to 0$ für ℓ, $j \to \infty$, d.h. (x^k) ist eine Cauchy-Folge und damit konvergent gegen ein $\overline{x} \in \mathbf{R}^n$. Da D abgeschlossen ist, ist sogar $\overline{x} \in D$, und wegen der Stetigkeit von g ist

$$\overline{x} = \lim_{k\to\infty} x^k = \lim_{k\to\infty} g(x^{k-1}) = g(\overline{x}) \,,$$

also $\overline{x}$ Fixpunkt von g.

2) Eindeutigkeit

Wäre $\tilde{x}$ ein weiterer Fixpunkt von g in D, so hätten wir

$$\|\overline{x} - \tilde{x}\| = \|g(\overline{x}) - g(\tilde{x})\| \leq q\|\overline{x} - \tilde{x}\| \,.$$

Wegen $q < 1$ folgte $\|\overline{x} - \tilde{x}\| = 0$.

3) Fehlerabschätzung

Lassen wir in (2.1) $\ell \to \infty$ streben, so folgt

$$\|\overline{x} - x^j\| \leq \frac{q^j}{1-q}\|x^1 - x^0\| \,.$$

□

Beispiel: Wir wollen die Lösung von $x = \tan x$ in $[\frac{\pi}{2}, \frac{3\pi}{2}]$ durch Iteration berechnen und versuchen zunächst

$$x^{k+1} = \tan x^k \,.$$

Es ist $g(x) = \tan x$ und damit $g'(x) = 1/\cos^2 x \geq 1$. g ist also nicht kontrahierend. Wir müssen unsere Gleichung erst in geeignete Form bringen. Dazu schreiben wir für $x \in [\frac{\pi}{2}, \frac{3\pi}{2}]$ für $x = \tan x$

$$x = \tan(x - \pi) \quad \text{oder} \quad \arctan x = x - \pi \quad \text{oder} \quad x = \pi + \arctan x$$

und setzen $g(x) = \pi + \arctan x$. Dann ist $g'(x) = 1/(1 + x^2)$. In $D = [\frac{\pi}{2}, \frac{3\pi}{2}]$ ist dann $|g'(x)| \leq 1/(1 + \pi^2/4) < 1$, und g bildet D in sich ab. Nach (2.1) ist g in D kontrahierend mit $q_1 = 1/(1 + \pi^2/4) = 0.2884$. Also hat g in D genau einen Fixpunkt, und das Iterationsverfahren

$$x^{k+1} = \pi + \arctan x^k$$

konvergiert gegen diesen. Mit $x^0 = \pi$ erhalten wir

k	x^k	$\frac{q_1^k}{1-q_1}\lvert x^1 - x^0\rvert$	$\frac{q_2^k}{1-q_2}\lvert x^1 - x^0\rvert$	$\overline{x} - x^k$
0	3.1416	–	–	1.3518
1	4.4042	0.5117	0.1279	0.0892
2	4.4891	0.1476	0.0118	0.0043
3	4.4932	0.0426	0.0011	0.0002
4	4.4934	0.0122	0.0001	0.0000

Wir sehen, daß die Fehlerabschätzung viel zu pessimistisch ist. Man kann auch $D = [\pi, \frac{3\pi}{2}]$ wählen mit $q_2 = 1/(1 + \pi^2) = 0.092$ und bekommt dann bessere Abschätzungen.

Welche Fixpunkte von g sind durch das Iterationsverfahren $x^{k+1} = g(x^k)$ berechenbar? - Durch Skizzen im $\mathbb{R}^1$ kommt man zu der Vermutung, daß dies diejenigen mit $|g'(\overline{x})| < 1$ sind. Dies ist der Inhalt des nächsten Satzes.

Satz 3.2.3 *(Lokaler Konvergenzsatz): Die Abbildung $g : K^n \to K^n$ besitze einen Fixpunkt $\overline{x}$, und es gebe eine Umgebung von $\overline{x}$, in der g kontrahierend ist. Dann gibt es eine Umgebung $U(\overline{x})$, so daß das Iterationsverfahren $x^{k+1} = g(x^k)$ für jedes $x^0 \in U(\overline{x})$ gegen $\overline{x}$ konvergiert.*

Beweis: Wir können annehmen, daß g in $D = \{x \in K^n : \|x - \overline{x}\| \leq r\}$ mit einem $r > 0$ kontrahierend ist, und zwar mit der Lipschitz-Konstanten $q < 1$. Dann ist für $x \in D$

$$\|g(x) - \overline{x}\| = \|g(x) - g(\overline{x})\| \leq q\|x - \overline{x}\| < qr \ ,$$

also auch $g(x) \in D$. g bildet also die abgeschlossene Menge D in sich ab und ist dort kontrahierend. Nach Satz 2.2 konvergiert das Iterationsverfahren für $x^0 \in D$.

□

Bemerkungen:

1) Sei $K = \mathbf{R}$. Die Bedingung des Satzes 2.3 ist erfüllt, wenn g in $\overline{x}$ stetig differenzierbar ist und $\rho(g'(\overline{x})) < 1$ ist. Dann gibt es nämlich nach Satz I.3.2 eine Norm $\| \cdot \|$ in $\mathbf{R}^n$ mit $\|g'(\overline{x})\| < 1$ für $x = \overline{x}$. Wegen der Stetigkeit von g' gilt dies dann auch in einer konvexen Umgebung von $\overline{x}$. Nach Satz 2.1 ist g in dieser Umgebung kontrahierend. Fixpunkte $\bar{x}$ mit $\varrho(g'(\bar{x}) < 1$ nennt man anziehend.

2) Für die Konvergenzgeschwindigkeit des Iterationsverfahrens ist offenbar die Lipschitz-Konstante q entscheidend:

$$\|x^{k+1} - \overline{x}\| = \|g(x^k) - g(\overline{x})\| \leq q\|x^k - \overline{x}\| \ .$$

Der Fehler vermindert sich also in jedem Schritt (mindestens) um den Faktor $q < 1$. Wir sprechen von linearer Konvergenz. Nach Satz 2.3 und Bemerkung 1 ist für die Konvergenzgeschwindigkeit die Zahl $\rho(g'(\overline{x}))$ entscheidend. Gilt hingegen für ein Iterationsverfahren

$$\|x^{k+1} - \overline{x}\| \leq C\|x^k - \overline{x}\|^p$$

mit einer Zahl $p > 1$, so sprechen wir von Konvergenz (mindestens) der Ordnung p. Für $p = 2$ sprechen wir von quadratischer, für $p = 3$ von kubischer Konvergenz. Quadratische Konvergenz ist dadurch gekennzeichnet, daß sich die Anzahl der korrekten Dezimalen von x^k bei jedem Schritt verdoppelt.

3.3 Das Newton-Verfahren

Zu lösen sei das nichtlineare System $f(x) = 0$, $f : D \to \mathbf{R}^n$, $D \subseteq \mathbf{R}^n$. Die Konvergenzgeschwindigkeit des Iterationsverfahrens $x^{k+1} = g(x^k)$ wird nach Bemerkung 2 aus dem vorigen Paragraphen durch die Zahl $\rho(g'(\overline{x}))$ bestimmt. Bei der Umwandlung einer Gleichung der Form $f(x) = 0$ in eine Fixpunktgleichung $x = g(x)$ versuchen wir daher $g'(\overline{x}) = 0$ zu erreichen. Wir machen für g den Ansatz

$$g(x) = x + A(x)f(x)$$

mit einer invertierbaren Matrix $A = (a_{i,j})$, d.h.

$$g_i(x) = x_i + \sum_{j=1}^{n} a_{i,j}(x) f_j(x) \; .$$

Wir berechnen die Jacobi-Matrix $g'(x)$. Es ist

$$\frac{\partial g_i}{\partial x_\ell} = \delta_{i,\ell} + \sum_{j=1}^{n} \frac{\partial a_{i,j}}{\partial x_\ell} f_j + \sum_{j=1}^{n} a_{i,j} \frac{\partial f_j}{\partial x_\ell} \; .$$

Für $x = \overline{x}$ ist $f(\overline{x}) = 0$, und wir können diese Gleichungen zusammenfassen zu

$$g'(\overline{x}) = I + A(\overline{x}) f'(\overline{x}) \; .$$

Um $g'(\overline{x}) = 0$ zu erreichen, brauchen wir also nur

$$A(x) = -(f'(x))^{-1}$$

zu wählen. Damit wird

$$g(x) = x - (f'(x))^{-1} f(x) \; , \tag{3.1}$$

und das Iterationsverfahren zur Lösung von $f(x) = 0$ lautet

$$x^{k+1} = x^k - (f'(x^k))^{-1} f(x^k) \; . \tag{3.2}$$

In dieser Form verlangt das Verfahren die Inversion von $f'(x^k)$. Dies ist für große n unzweckmäßig. Man schreibt daher

$$f'(x^k)(x^{k+1} - x^k) + f(x^k) = 0 \; . \tag{3.3}$$

Dies ist das Newton-Verfahren zur Lösung von $f(x) = 0$. Man hätte es auch einfacher durch Linearisierung herleiten können. Ist x^k eine Näherung für die gesuchte Lösung $\overline{x}$, so hat man für $f \in C^2(D)$

$$f(x) = f(x^k) + f'(x^k)(x - x^k) + \; \mathrm{O}\left(\|x - x^k\|^2\right) \; .$$

Man vernachlässigt nun $O(\|x - x^k\|^2)$ und nimmt als neue Näherung x^{k+1} für $\overline{x}$ die Lösung von

$$0 = f(x^k) + f'(x^k)(x - x^k) .$$

Dies ist genau (3.3). Im $\mathbf{R}^1$ ersetzt das Newton-Verfahren also die Kurve $y = f(x)$ durch die Tangente in x^k und berechnet x^{k+1} als Nullstelle der Tangente.

Beispiel: Wir lösen in $\mathbf{R}^1$ die Gleichung $x^2 - 2 = 0$, berechnen also $\overline{x} = \sqrt{2}$. Es ist

$$f(x) = x^2 - 2 , \quad q(x) = x - \frac{f(x)}{f'(x)} = x - \frac{x^2 - 2}{2x} = \frac{1}{2}(x + \frac{2}{x}) .$$

Mit $x^0 = 1$ erhalten wir

k	x_k	Anzahl der korrekten Dezimalen
0	1	1
1	1.5	1
2	1.41	3
3	1.41421	6
4	1.414213562375	12

Satz 3.3.1 *Die Abbildung $f : \mathbf{R}^n \to \mathbf{R}^n$ besitze die Nullstelle $\overline{x}$, sei in einer Umgebung von $\overline{x}$ zweimal stetig differenzierbar, und $f'(\overline{x})$ sei invertierbar. Dann gibt es eine Umgebung D von $\overline{x}$, so daß das Newton-Verfahren für alle $x^0 \in D$ quadratisch gegen $\overline{x}$ konvergiert.*

Beweis: Sei $g(x) = x - (f'(x))^{-1} f(x)$. Wegen $g'(\overline{x}) = 0$ gibt es eine Umgebung D von $\overline{x}$ mit $\|g'(x)\| \leq 1/2$ für $x \in D$. Nach Satz 2.3 konvergiert das Iterationsverfahren $x^{k+1} = g(x^k)$, also das Newton-Verfahren, gegen $\overline{x}$, wenn nur $x^0 \in D$.

Zum Nachweis der quadratischen Konvergenz bilden wir

$$x^{k+1} - \overline{x} = x^k - \overline{x} - (f'(x^k))^{-1}(f(x^k) - f(\overline{x})) .$$

Der Satz von Taylor liefert

$$f(x^k) - f(\overline{x}) = f'(x^k)(x^k - \overline{x}) + \varepsilon(\overline{x} - x^k) ,$$

$$\|\varepsilon(\overline{x} - x^k)\| \leq M\|\overline{x} - x^k\|^2$$

mit einer Konstanten M, welche die zweiten Ableitungen von f enthält. Dazu setzen wir D so klein voraus, daß $f \in C^2(D)$. Es folgt

$$\begin{aligned} x^{k+1} - \overline{x} &= x^k - \overline{x} - (x^k - \overline{x}) - (f'(x^k))^{-1}\varepsilon(\overline{x} - x^k) \\ &= -f'(x^k)^{-1}\varepsilon(\overline{x} - x^k) . \end{aligned}$$

Sei nun N eine Konstante mit $||(f'(x^k))^{-1}|| \leq N$. Dann gilt

$$||x^{k+1} - \overline{x}|| \leq NM||x^k - \overline{x}||^2 ,$$

und dies bedeutet quadratische Konvergenz.

□

Was passiert, wenn $f'(\overline{x})$ nicht invertierbar ist? - Wir betrachten den Fall $n = 1$. Sei also $f(\overline{x}) = 0,\ f \in C^2,\ f'(\overline{x}) = 0$, aber $f''(\overline{x}) \neq 0$. Dann ist

$$f(x) = (x - \overline{x})^2 p(x) , \quad p(\overline{x}) \neq 0$$

mit $p \in C^2$. Wir berechnen

$$f'(x) = 2(x-\overline{x})p(x)+(x-\overline{x})^2 p'(x), f''(x) = 2p(x)+4(x-\overline{x})p'(x)+(x-\overline{x})^2 p''(x)$$

und erhalten für $g(x) = x - f(x)/f'(x)$

$$\begin{aligned} g'(x) &= 1 - \frac{f'(x)}{f'(x)} + \frac{f(x)f''(x)}{f'^2(x)} = \frac{(x-\overline{x})^2 p(x)(2p(x) + \mathrm{O}(x-\overline{x}))}{4(x-\overline{x})^2(p(x) + \mathrm{O}(x-\overline{x}))^2} \\ &= \frac{1}{2}(1 + \mathrm{O}(x-\overline{x})) . \end{aligned}$$

Also ist $g'(\overline{x}) = \frac{1}{2}$, und nach Satz 2.3 folgt lineare Konvergenz. Für $f'(\overline{x}) = 0$ konvergiert das Newton-Verfahren also immer noch, aber die quadratische Konvergenz geht verloren.

Der Rechenaufwand für das Newton-Verfahren wird im wesentlichen durch die Berechnung von $f'(x)$ bestimmt. Es gibt verschiedene Varianten des Newton-Verfahrens, die diesen Aufwand reduzieren.

1. Das vereinfachte Newton-Verfahren

Hier ersetzt man einfach $(f'(x^k))^{-1}$ durch $(f'(x^0))^{-1}$, iteriert also gemäß

$$f'(x^0)(x^{k+1} - x^k) = -f(x^k) .$$

Man hat immer noch lokale Konvergenz, aber die quadratische Konvergenz geht verloren.

2. Das Sekanten-Verfahren ("Regula falsi") $(n = 1)$

Hier ersetzt man die Tangente des Newton-Verfahrens durch die Sekante in den Punkten x^k, x^{k-1}, also

$$y = f(x^k) + (x - x^k)\,\frac{f(x^k) - f(x^{k-1})}{x^k - x^{k-1}} ,$$

und nimmt als neue Näherung x^{k+1} die Nullstelle dieser Sekante, also

$$x^{k+1} = x^k - \left(\frac{f(x^k) - f(x^{k-1})}{x^k - x^{k-1}} \right)^{-1} f(x^k) \; .$$

Mit anderen Worten: Man ersetzt $f'(x^k)$ durch den Differenzenquotienten in x^k, x^{k-1}. Dies ist auch in $\mathbb{C}$ sinnvoll.

Satz 3.3.2 *Sei $f : \mathbb{R} \to \mathbb{R}$ in einer Umgebung der Nullstelle $\overline{x}$ zweimal stetig differenzierbar, und sei $f'(\overline{x}) \neq 0$. Dann gibt es eine Umgebung D von $\overline{x}$, so daß das Sekanten-Verfahren für jede Wahl von x^0, x^1 in D gegen $\overline{x}$ konvergiert, und zwar gilt $\|\overline{x} - x^k\| \leq c_k$, wobei $c_k \to 0$ mit der Konvergenzordnung $(1+\sqrt{5})/2 = 1.618$.*

Beweis:

(a) Es gibt eine Konstante C, so daß

$$|x^{k+1} - \overline{x}| \leq C|x^k - \overline{x}||x^{k-1} - \overline{x}| \; , \quad k = 1, 2, \ldots \; .$$

Hierzu schreiben wir

$$\begin{aligned} x^{k+1} - \overline{x} &= x^k - \overline{x} - \frac{f(x^k)}{\frac{f(x^k)-f(x^{k-1})}{x^k - x^{k-1}}} \\ &= (x^k - \overline{x}) \, \frac{\dfrac{f(x^k) - f(x^{k-1})}{x^k - x^{k-1}} - \dfrac{f(x^k) - f(\overline{x})}{x^k - \overline{x}}}{\dfrac{f(x^k) - f(x^{k-1})}{x^k - x^{k-1}}} \\ &= (x^k - \overline{x}) \, \frac{\int_0^1 \{f'(x^{k-1} + t(x^k - x^{k-1})) - f'(\overline{x} + t(x^k - \overline{x}))\} dt}{f'(\xi^k)} \end{aligned}$$

mit ξ^k zwischen x^k und x^{k-1}. In einer Umgebung D von $\overline{x}$ ist

$$|f'(x)| \geq m > 0 \; , \quad |f''(x)| \leq M$$

und daher

$$|x^{k+1} - \overline{x}| \leq |x^k - \overline{x}| \, \frac{M|x^{k-1} - \overline{x}|}{m} \; ,$$

falls x_k, $x_{k-1} \in D$. Ist $D = [\overline{x} - \varepsilon, \overline{x} + \varepsilon]$ und x^0, $x^1 \in D$, so folgt also $|x^2 - \overline{x}| \leq \varepsilon^2 \frac{M}{m}$, und für $\varepsilon \frac{M}{m} \leq 1$ ist auch $x^2 \in D$. Damit bleiben alle x^k in D, und (a) ist gezeigt.

(b) Wir setzen $e^{f_k} = C|x^k - \overline{x}|$. Dann wird

$$f_{k+1} \leq f_k + f_{k-1} \, .$$

Sei $F_0 = F_1 = 1$ und $F_{k+1} = F_k + F_{k-1}$. Ist $|f_0|, |f_1| \leq 1$, so gilt offenbar $f_k \leq F_k$, $k = 0, 1, \ldots$. Die F_k kann man leicht berechnen. Mit $\tau = \frac{1}{2}(1 + \sqrt{5})$ ist

$$F_k = c_1 \tau^k + c_2 (-\tau)^{-k} \, ,$$

$$c_1 = \frac{\tau^2}{1 + \tau^2} \quad , \quad c_2 = \frac{1}{1 + \tau^2} \, .$$

Wir setzen nun $\varepsilon_k = e^{F_k}$. Dann ist

$$\frac{\varepsilon_k}{\varepsilon_{k-1}^{\tau}} = e^{F_k - \tau F_{k-1}} = e^{c_2((-\tau)^{-k} - \tau(-\tau)^{-k+1})}$$

und dies strebt wegen $\tau > 1$ für $k \to \infty$ gegen 1. Also gilt $\varepsilon_k \leq c\varepsilon_{k-1}^{\tau}$ für ein geeignetes $c > 0$, und

$$|x^k - \overline{x}| = \frac{1}{C} e^{f_k} \leq \frac{1}{C} e^{F_k} = \frac{1}{C} \varepsilon_k \, .$$

Die Behauptung des Satzes folgt mit $c_k = \varepsilon_k / C$.

□

Bemerkung: Die Zahlen F_k heißen Fibonacci-Zahlen (Leonardo Fibonacci, 1175 - 1226).

3.4 Iterationsverfahren für lineare Gleichungssysteme

Das Standardverfahren zur Lösung linearer Gleichungssysteme ist das Eliminationsverfahren. Bei sehr großen Systemen mit spezieller Struktur können iterative Verfahren aber günstiger sein. Dies trifft insbesondere zu auf lineare Systeme, deren Matrizen nur sehr wenige von Null verschiedene Elemente haben (dünnbesetzte Matrizen).

Sei $Ax = b$ zu lösen. Sei $A = D + L + R$ mit

$$D = \begin{pmatrix} a_{1,1} & & \mathbf{O} \\ & \ddots & \\ \mathbf{O} & & a_{n,n} \end{pmatrix}, \; L = \begin{pmatrix} 0 & & & \mathbf{O} \\ a_{1,1} & \ddots & & \\ \vdots & & & \\ a_{n,1} & \cdots & a_{n,n-1} & 0 \end{pmatrix}, \; R = \begin{pmatrix} 0 & a_{1,2} & \cdots & a_{1,n} \\ & \ddots & & \vdots \\ & & & a_{n-1,n} \\ \mathbf{O} & & & 0 \end{pmatrix}.$$

Das Gesamtschritt- oder Jacobi-Verfahren zur Lösung von $(D+L+R)x = b$ lautet

$$Dx^{k+1} + Lx^k + Rx^k = b \,,$$

während das Einzelschritt- oder Gauß-Seidel-Verfahren gemäß

$$(D + L)x^{k+1} + Rx^k = b$$

iteriert. Elementweise lauten Gesamt- bzw. Einzelschrittverfahren

$$\begin{aligned} x_i^{k+1} &= \left(b_i - \sum_{j \neq i} a_{i,j} x_j^k \right) / a_{i,i} \,, \\ x_i^{k+1} &= \left(b_i - \sum_{j=1}^{i-1} a_{i,j} x_j^{k+1} - \sum_{j=1+1}^{n} a_{i,j} x_j^k \right) / a_{i,i} \,. \end{aligned}$$

Der rechentechnische Unterschied zwischen den beiden Verfahren zeigt sich am deutlichsten bei der Programmierung

```
GS(x, n)    /* Führt einen Gesamtschritt durch */
 {  for i = 1, ..., n
      x_i^1 = (b_i − Σ_{j≠i} a_{i,j} x_j)/a_{i,i};
    x = x^1;
 }
ES(x, n)    /* Führt einen Einzelschritt durch */
 {  for i = 1, ..., n
      x_i = (b_i − Σ_{j≠i} a_{i,j} x_j)/a_{i,i};
 }
```

Die Iterationsvorschrift des Einzelschrittverfahrens wird also durch sukzessives Überschreiben ausgeführt. Man braucht also nicht zwei Vektoren x, x^1

zu halten, sondern kommt mit einem aus. Da x häufig sehr groß ist, ist dies von Vorteil. Darüber hinaus werden wir sehen, daß das Einzelschrittverfahren häufig schneller konvergiert als das Gesamtschrittverfahren.

Beispiel: Wir wollen das Dirichlet-Problem

$$-\Delta u = f \quad \text{in } \Omega\,, \quad u = 0 \quad \text{auf } \partial\Omega$$

für das Quadrat $\Omega = (0,1)^2$ in $\mathbb{R}^2$ lösen. Hier ist $\Delta = \partial^2/\partial x^2 + \partial^2/\partial y^2$ der Laplace-Operator. Man überdeckt Ω durch ein Gitter mit Schrittweite $h = \frac{1}{n}$ und ersetzt die Differentialgleichung im Gitterpunkt $\begin{pmatrix} i \\ j \end{pmatrix} h$ durch

$$4u_{i,j} - (u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}) = h^2 f_{i,j}\,, \quad i,j = 1,\ldots,n-1\,.$$

Dies ist ein lineares System von $(n-1)^2$ Gleichungen in ebensovielen Unbekannten $u_{i,j}$; die $u_{i,j}$ mit $i,j \in \{0,n\}$ werden Null gesetzt. Ist h hinreichend klein, so hofft man, daß $u_{i,j}$ eine gute Näherung für $u\left(\begin{pmatrix} i \\ j \end{pmatrix} h\right)$ ist.

Ein Einzelschritt lautet nun einfach

$$\begin{array}{lcl} \text{for } i & = & 1,\ldots,n-1 \quad \text{for } j = 1,\ldots,n-1 \\ u_{i,j} & = & (h^2 f_{i,j} + u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1})/4\,; \end{array}$$

Beide Verfahren kann man beschleunigen durch Einführung eines "Relaxationsparameters" ω. Für das Gesamtschrittverfahren setzt man

$$x_i^{k+1} = (1-\omega)x_i^k + \omega(b_i - \sum_{j\neq i} a_{i,j}x_j^k)/a_{i,i}\,,$$

und für das Einzelschrittverfahren

$$x_i^{k+1} = (1-\omega)x_i^k + \omega\left(b_i - \sum_{j=1}^{i-1} a_{i,j}x_j^{k+1} - \sum_{j=i+1}^{n} a_{i,j}x_j^k\right)/a_{i,i}\,.$$

Man bildet also gewichtete Summen der k-ten und der $k+1$-ten Näherung. Das letzte Verfahren heißt SOR (successive overrelaxation) und spielt eine wichtige Rolle. Für die Programmierung von SOR gilt das über das Einzelschrittverfahren Gesagte.

In Matrizenschreibweise haben alle diese Verfahren die Form

$$x^{k+1} = Bx^k + c\,.$$

Dabei ist für das Gesamtschrittverfahren

$$B = -D^{-1}(L+R)\,, \quad c = D^{-1}b\,,$$

für das Einzelschrittverfahren

$$B = -(D+L)^{-1}R\,, \quad c = (D+L)^{-1}b$$

und für das SOR-Verfahren

$$B = (D+\omega L)^{-1}((1-\omega)D - \omega R)\,, \quad c = \omega(D+\omega L)^{-1}b\,.$$

Alle unsere Konvergenzaussagen werden auf folgendem Satz beruhen:

Satz 3.4.1 *Das Iterationsverfahren*

$$x^{k+1} = Bx^k + c$$

konvergiert genau dann für jede Wahl von x^0 und c, wenn $\rho(B) < 1$. In diesem Fall konvergiert es gegen die eindeutig bestimmte Lösung $\overline{x}$ von $\overline{x} = B\overline{x} + c$.

Beweis: Ist $\rho(B) < 1$, so gibt es nach Satz 2.5.1 eine Norm $\|\cdot\|$, so daß $\|B\| < 1$. Damit ist $g(x) = Bx + c$ kontrahierend im ganzen Raum. Konvergenz gegen $\overline{x}$ folgt dann aus Satz 2.2.

Sei umgekehrt das Verfahren für alle x^0, c konvergent. Wir wählen für x^0, c den Eigenvektor y von B mit Eigenwert λ. Dann ist

$$x^k = (1 + \lambda + \cdots + \lambda^k)y\,.$$

Dies ist nur konvergent für $|\lambda| < 1$. Also ist $\rho(B) < 1$.

□

Definition 3.4.1 *Eine (n,n)-Matrix A über K erfüllt das starke Zeilensummenkriterium, wenn $|a_{i,i}| > \sum\limits_{j\neq i} |a_{i,j}|$, $i = 1,\ldots,n$. Sie erfüllt das schwache Zeilensummenkriterium, wenn diese Ungleichungen mit mindestens einer Ausnahme im schwachen Sinne (d.h. mit $\geq$ an Stelle von $>$) gelten.*

A heißt reduzibel, wenn man $\{1,\ldots,n\}$ so in zwei disjunkte nichtleere Mengen I, J aufteilen kann, daß $a_{i,j} = 0$ für $(i,j) \in I \times J$. Andernfalls heißt A irreduzibel.

Ist A reduzibel, so kann man A durch Zeilen- und Spaltenvertauschungen in die Form

$$A = \begin{pmatrix} A_1 & 0 \\ A_3 & A_2 \end{pmatrix}$$

bringen. Das Gleichungssystem $Ax = b$ zerfällt dann in zwei kleinere Systeme mit den Matrizen A_1, A_2.

Satz 3.4.2 *Für die (n,n)-Matrix A sei eine der folgenden Bedingungen erfüllt:*

(a) A erfülle das starke Zeilensummenkriterium.

(b) A erfülle das schwache Zeilensummenkriterium und sei irreduzibel.

Dann konvergiert das Gesamtschrittverfahren zur Lösung von $Ax = b$ für jedes b und x^0 gegen die eindeutig bestimmte Lösung $\overline{x}$ von $A\overline{x} = b$.

Beweis: Wir zeigen, daß $\rho(D^{-1}(L+R)) < 1$, wo $A = D + L + R$. Unter der Voraussetzung (a) ist

$$||D^{-1}(L+R)||_\infty = \max_{i=1}^{n} \frac{1}{|a_{i,i}|} \sum_{j \neq i} |a_{i,j}| < 1 \,,$$

also in der Tat $\rho(D^{-1}(L+R)) < 1$. Unter der Voraussetzung (b) folgt $\rho(D^{-1}(L+R)) \leq 1$. Es genügt daher zu zeigen, daß $D^{-1}(L+R)$ keinen Eigenwert λ mit $|\lambda| = 1$ hat. Wäre λ ein solcher und x ein zugehöriger Eigenvektor mit $||x||_\infty = 1$, so hätten wir für alle $i \in I = \{i : |x_i| = 1\}$

$$1 = |x_i| = |\lambda||x_i| = |(D^{-1}(L+R)x)_i| = \frac{1}{|a_{i,i}|} \left| \sum_{j \neq i} a_{i,j} x_j \right| \leq 1 \,,$$

also

$$\left| \sum_{j \neq i} a_{i,j} x_j \right| = |a_{i,i}| \,, \quad i \in I \,.$$

Wegen des schwachen Zeilensummenkriteriums kann $J = \{1, \ldots, n\} \setminus I$ nicht leer sein. Da A irreduzibel ist, gibt es $(i_0, j_0) \in I \times J$ mit $a_{i_0,j_0} \neq 0$. Dann ist

$$\begin{aligned} 1 = |\lambda||x_{i_0}| &= |(D^{-1}(L+R)x)_{i_0}| = \frac{1}{|a_{i_0,i_0}|} \left| \sum_{j \neq i_0} a_{i_0,j} x_j \right| \\ &< \frac{1}{|a_{i_0,i_0}|} \sum_{j \neq i_0} |a_{i_0,j}| \leq 1 \,. \end{aligned}$$

Dieser Widerspruch zeigt, daß es keinen solchen Eigenwert λ geben kann.

□

Satz 3.4.3 *Die (n,n)-Matrix A sei positiv definit. Dann konvergiert das SOR-Verfahren zur Lösung von $Ax = b$ für jede Wahl von x^0 und b, wenn $0 < \omega < 2$.*

Beweis: Wir haben zu zeigen, daß für die SOR-Matrix $B_\omega = (D+\omega L)^{-1}((1-\omega)D - \omega R)$ der Spektralradius $\rho(B_\omega) < 1$ ist für $0 < \omega < 2$. Sei x ein Eigenvektor von B_ω zum Eigenwert λ mit $|\lambda| = \rho(B_\omega)$, also

$$((1-\omega)D - \omega R)x = \lambda(D + \omega L)x \; .$$

Inneres Produkt mit x ergibt

$$(1-\omega)(Dx,x) - \omega(Rx,x) = \lambda((Dx,x) + \omega(Lx,x)) \; .$$

Wir setzen $d = (Dx,x)$, $\ell = (Lx,x)$. Weil A hermitesch ist, gilt $(Rx,x) = (L^*x,x) = (x,Lx) = \overline{\ell}$, also

$$(1-\omega)d - \omega\overline{\ell} = \lambda(d + \omega\ell)$$

oder

$$\lambda = \frac{(1-\omega)d - \omega\overline{\ell}}{d + \omega\ell} \; .$$

Mit $\ell = \alpha + i\beta$, α, $\beta \in \mathbf{R}$ bekommen wir, weil $d > 0$,

$$|\lambda|^2 = \frac{((1-\omega)d - \omega\alpha)^2 + \omega^2\beta^2}{(d + \omega\alpha)^2 + \omega^2\beta^2} \; .$$

Es ist also genau dann $|\lambda| < 1$, wenn

$$|(1-\omega)d - \omega\alpha| < |d + \omega\alpha| \; .$$

Mit $\alpha' = \alpha/d$ lautet dies

$$|1 - \omega - \omega\alpha'| < |1 + \omega\alpha'| \; . \tag{4.1}$$

Da A positiv definit ist, gilt

$$0 < (Ax,x) = d + \ell + \overline{\ell} = d + 2\alpha = d(1 + 2\alpha') \; ,$$

also $\alpha' > -1/2$. Für $0 < \omega < 2$ können wir die Betragsstriche bei $|1 + \omega\alpha'|$ in (4.1) weglassen und erhalten

$$|1 - \omega - \omega\alpha'| < 1 + \omega\alpha'$$

als Bedingung für $|\lambda| < 1$. Dies ist aber für $0 < \omega < 2$ richtig.

□

Die Bedingung an ω kann man nicht abschwächen. Es gilt nämlich

Satz 3.4.4 *Sei B_ω die SOR-Matrix für eine beliebige Matrix mit nichtverschwindenden Diagonalelementen. Dann gilt*

$$\rho(B_\omega) \geq |\omega - 1| \; .$$

Beweis: Es ist

$$B_\omega = (I + \omega D^{-1}L)^{-1}((1-\omega)I - \omega D^{-1}R) \; .$$

Nach dem Multiplikationssatz für Determinanten ist also $\det(B_\omega) = (1-\omega)^n$. Sind $\lambda_1, \ldots, \lambda_n$ die Eigenwerte von B_ω (nicht notwendig verschieden), so gilt

$$\rho(B_\omega)^n \geq |\lambda_1 \cdots \lambda_n| = |\det(B_\omega)| = |1-\omega|^n \; ,$$

und daraus folgt die Behauptung.

□

Kapitel 4

Eigenwertprobleme

4.1 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 4.1.1 *Sei A eine komplexe (n,n)-Matrix. $\lambda \in \mathbb{C}$ heißt Eigenwert von A, wenn es $x \in \mathbb{C}^n$, $x \neq 0$ gibt mit $Ax = \lambda x$. x heißt dann Eigenvektor von A zum Eigenwert λ.*

Als notwendige und hinreichende Bedingung dafür, daß λ Eigenwert 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 λ:

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

Definition 4.1.2 *Jedem Eigenwert λ 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 Eigenvektor zu λ.

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

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

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

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

Definition 4.1.3 *Die (n,n)-Matrizen A, B heißen ähnlich, wenn es eine nichtsinguläre (n,n)-Matrix x gibt mit*

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

Satz 4.1.1 *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 λ ein Eigenwert von A der geometrischen Vielfachheit ρ, so gibt es ρ l.u. Eigenvektoren $x_1, \ldots, x_\rho$ zu λ, also

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

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

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

also sind $y_1, \ldots, y_\rho$ l.u. Eigenvektoren von B zu λ. Die geometrische Vielfachheit von λ als Eigenwert von A ist also nicht größer als die geometrische Vielfachheit von λ als Eigenwert von B. Da die Voraussetzungen in A, B symmetrisch sind, müssen die geometrischen Vielfachheiten 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 Eigenvektor $e_k = k$-tem Einheitsvektor. Offenbar ist

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

2) $A = XDX^{-1}$ mit $D = \begin{pmatrix} d_1 & & \\ & \ddots & \\ & & d_n \end{pmatrix}$. Nach Satz 6.1.1 und Beispiel 1) ist $\lambda_k = d_k$, $\rho(\lambda_k) = \sigma(\lambda_k)$, $k = 1, \ldots, n$. Dem Beweis von Satz 6.1.1 und Beispiel 1) entnimmt man die Eigenvektoren $x_k = Xe_k = k$-te Spalte von X. Matrizen dieser Art, welche also ähnlich zu einer Diagonalmatrix sind, nennt man diagonalisierbar.

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

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

und der einzige l.u. Eigenvektor ist $x = e_1$. Also ist die geometrische Vielfachheit von μ für $n > 1$ 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 4.1.2 *(Jordan'sche Normalform). Jede komplexe (n, n)-Matrix ist ähnlich einer Matrix*

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

Die J_ℓ sind bis auf die Reihenfolge eindeutig bestimmt.

Beweis: Siehe etwa F. Lorenz, Lineare Algebra II, Kap. IX, §4.

Bemerkungen zu Satz 1.2:

1) Jedes λ_ℓ, $\ell = 1, \ldots, r$ ist Eigenwert.

2) $\rho(\lambda) =$ Anzahl der J_ℓ mit λ auf der Hauptdiagonalen. Die λ_ℓ sind also die nach ihrer geometrischen Vielfachheit gezählten Eigenwerte.

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

$$\begin{aligned} \det\ (\lambda I - J) &= \prod_{\ell=1}^{r} \det\,(\lambda I - J_\ell) = \prod_{\ell=1}^{r} (\lambda - \lambda_\ell)^{\nu_\ell}\,, \\ \nu_\ell &= \text{Länge}\ (J_\ell)\,. \end{aligned}$$

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 ν_ℓ die Länge von J_ℓ. Wir spalten X auf entsprechend der Aufspaltung von J:

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

mit (n, ν_ℓ)-Matrizen X_ℓ. Dann haben wir

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

Insbesondere ist der von den Spalten von X_ℓ aufgespannte Teilraum ein invarianter Unterraum von A. Wir untersuchen diese Gleichung für ein ℓ und setzen $\nu = \nu_\ell$, $\lambda = \lambda_\ell$. Seien $x_1, \ldots, x_\nu$ die Spalten von X_ℓ, also $X_\ell = (x_1, \ldots, x_\nu)$. Dann lautet die Gleichung

$$(Ax_1, \ldots, Ax_\nu) = (\lambda x_1, \lambda x_2 + x_1, \ldots, \lambda x_\nu + x_{\nu-1})$$

oder

$$\begin{aligned} Ax_1 &= \lambda x_1 \quad\ , \\ Ax_i &= \lambda x_i + x_{i-1}\,, \quad i = 2, \ldots, \nu\,. \end{aligned}$$

x_1 ist also Eigenvektor zum Eigenwert λ, wie wir schon aus Beispiel 3 und Satz 6.1.1 wissen. Für die weiteren Vektoren x_i gilt

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

also

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

Definition 4.1.4 *Ein Vektor mit* $(A-\lambda I)^{i-1}x \neq 0$, $(A-\lambda I)^i x = 0$ *heißt Hauptvektor der Stufe* i *von* A *zum Eigenwert* λ. *Der von allen Hauptvektoren zu einem Eigenwert* λ *von* A *aufgespannte Teilraum heißt invarianter Unterraum von* A *zum Eigenwert* λ.

Bemerkungen:

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

Definition 4.1.5 *Eine* (n,n)*-Matrix* A *heißt hermitesch, wenn* $A = A^*$ *mit* $A^* = \overline{A}^T$.

Satz 4.1.3 *Sei* A *hermitesch. Dann sind alle Eigenwerte von* A *reell.* A *besitzt* n *l.u. paarweise orthonormale Eigenvektoren.*

Beweis:

1) Realität der Eigenwerte. 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) Orthonormalität der Eigenvektoren.

   Sind x_1, x_2 Eigenvektoren zu den verschiedenen Eigenwerten λ_1, λ_2, so gilt
$$0 = (Ax_1, x_2) - (Ax_2, x_1) = (\lambda_1 - \lambda_2)(x_1, x_2) \ ,$$
   also sind x_1, x_2 orthogonal. Sind $x_1, \ldots, x_2$ die Eigenvektoren zu dem Eigenwert λ, so kann man diese orthonormalisieren.

3) Es genügt zu zeigen, daß keine Hauptvektoren 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 $(A-\lambda I)x \neq 0$.

□

Bemerkung: Die Jordan'sche Normalform einer hermiteschen Matrix ist also eine reelle Diagonalmatrix. Die Matrix X kann unitär gewählt werden, also $XX^* = I$. A heißt positiv definit, wenn alle Eigenwerte > 0 sind. $(Ax, x) = 1$ stellt dann ein Ellipsoid dar mit Halbachsen $1/\sqrt{\lambda_\ell}$ in Richtung des ℓ-ten Eigenvektors.

4.2 Die Potenzmethode

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

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

Wir analysieren die Potenzmethode zunächst in dem Fall

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

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

$$\begin{aligned} x^{(0)} &= \sum_{i=1}^{n} c_i x_i \quad , \\ x^{(k)} &= A^k x^{(0)} = \sum_{i=1}^{n} c_i A^k x_i = \sum_{i=1}^{n} c_i \lambda_i^k x_i \\ &= \lambda_1^k (c_1 x_1 + r_k) \ , \\ r_k &= \sum_{i=2}^{n} c_i \left(\frac{\lambda_i}{\lambda_1}\right)^k x_i \ . \end{aligned}$$

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

$$\|r_k\| = 0\left(\left(\frac{\lambda_2}{\lambda_1}\right)^k\right) \quad .$$

Zur Berechnung von λ_1 wählt man einen komplexen Vektor d und bildet

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

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

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

Genauer gilt

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

d.h. die Konvergenzgeschwindigkeit entspricht der von $(\lambda_2/\lambda_1)^k$. Wir sprechen von geometrischer Konvergenz.

Einen Eigenvektor x_1 zu λ_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:

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

Mit $d = (1,1,1)^T$ erhält man

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

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

$$\begin{matrix} 1 & 0.703237 & 0.699160 \\ 1 & 1 & 1 \\ 1 & 0.703237 & 0.699160 \end{matrix}$$

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 λ_2 wählt man μ 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

$$\begin{aligned} x^{(k+1)} &= Tx^{(k)} \; , \\ (A - \mu I)x^{(k+1)} &= x^{(k)} \end{aligned}$$

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

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

Die Eigenwerte λ_ℓ sind also nach geometrischer Vielfachheit gezählt (d.h. hat λ_ℓ die geometrische Vielfachheit ρ_ℓ, so tritt λ_ℓ ρ_ℓ-mal auf) 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 $y^{(k)} = Jy^{(k-1)}$. Spalten wir $y^{(k)}$ auf in Teilvektoren $y_\ell^{(k)}$ der Länge ν_ℓ, so entsteht

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

Zur Berechnung von J_ℓ^k setzen wir

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

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

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

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

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

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

$$x^{(k)} = Xy^{(k)} = \sum_{\ell=1}^{r} X_\ell y_\ell^{(k)} = \sum_{\ell=1}^{r} \lambda_\ell^k X_\ell M_{\ell k} y_\ell^{(0)} \;.$$

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.

Fall 1: Es gibt einen Eigenwert maximalen Betrages, und dessen algebraische und geometrische Vielfachheit stimmen überein.

Der gemeinsame Wert dieser Vielfachheiten sei ϱ. Dann ist

$$\begin{aligned} \lambda_1 &= \lambda_2 = \cdots = \lambda_\rho\ ,\ \ |\lambda_\rho| > |\lambda_{\rho+1}| \geq \cdots \geq |\lambda_r|\ , \\ x^{(k)} &= \sum_{\ell=1}^{\rho} \lambda_\ell^k X_\ell M_{\ell k} y_\ell^{(0)} + \sum_{\ell=\rho+1}^{r} \lambda_\ell^k X_\ell M_{\ell k} y_\ell^{(0)}\ . \end{aligned}$$

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

$$\begin{aligned} x^{(k)} &= \lambda_1^k \left\{ \sum_{\ell=1}^{\rho} X_\ell y_\ell^{(0)} + \sum_{\ell=\rho+1}^{r} \left(\frac{\lambda_\ell}{\lambda_1}\right)^k X_\ell M_{\ell k} y_\ell^{(0)} \right\} \\ &= \lambda_1^k \{x_1 + r_k\}\ . \end{aligned}$$

Hier ist x_1 ein Eigenvektor zum Eigenwert λ_1, und r_k hat die Größenordnung

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

wo $\nu = \underset{\ell > \rho}{\text{Max}}\ \nu_\ell$. Damit hat man 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 numerisch kaum bemerkt wird.

Fall 2: Es gibt einen Eigenwert maximalen Betrages, aber seine geometrische Vielfachheit ist kleiner als die algebraische.

Wir betrachten den einfachsten Spezialfall

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

mit $\rho(\lambda_1) = 1$, $\sigma(\lambda_1) = 2$. Es ist dann

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

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.

$$X_1 M_{1k} y_1^{(0)} = a + kb$$

mit geeigneten Vektoren $a, b \in \mathbb{C}^2$. Bildet man nun (x^k, d) und die Quotienten zur Berechnung von λ_1, so entsteht

$$\begin{aligned}\frac{(x^{(k+1)}, d)}{(x^{(k)}, d)} &= \lambda_1 \frac{(a,d) + (k+1)(b,d) + (r_{k+1}, d)}{(a,d) + k(b,d) + (r_k, d)} \\ &= \lambda_1 \left(1 + \mathrm{O}\,(\frac{1}{k})\right)\end{aligned}$$

für $k \to \infty$, wenn nur $(b, d) \neq 0$. Man hat also auch in diesem Fall Konvergenz gegen λ_1, aber sehr langsam.

Fall 3: Es gibt verschiedene betragsmaximale Eigenwerte.

Wir behandeln wieder den einfachsten Spezialfall

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

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

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

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

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

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 4.2.1 *Die Potenzmethode konvergiert, wenn es einen betragsgrößten Eigenwert gibt. Stimmen für diesen die algebraische und geometrische Vielfachheit überein, so ist die Konvergenz (fast) geometrisch. Gibt es verschiedene betragsgleiche Eigenwerte, so ist die Potenzmethode nicht konvergent.*

4.3 Der LR- und der QR-Algorithmus

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 $|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n|$, und es gebe 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

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

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 $x_1^{(0)}$ 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. Daher subtrahieren wir ein geeignetes Vielfaches von $x_1^{(1)}$:

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

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:

$$r_{jj} x_j^{(1)} = A x_j^{(0)} - r_{1j} x_1^{(1)} - r_{2j} x_2^{(1)} - \ldots - r_{j-1,j} x_{j-1}^{(1)} , \quad j = 1, \ldots, n . \tag{3.1}$$

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

$$X_1 R_0 = A X_0 \ . \tag{3.2}$$

X_1 ist eine linke Dreiecksmatrix mit Hauptdiagonale 1. Wir haben hier also die LR-Zerlegung von AX_0 vorliegen. Die Potenzmethode läuft nun folgendermaßen: Sei $X_0 = I$.
Ist X_0 berechnet, so bilde man die LR-Zerlegung

$$X_{k+1} R_k = A X_k \tag{3.3}$$

von AX_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: Mit einer rechten Dreiecksmatrix R gilt

$$X_k \to XR \quad , \quad X = (x_1, \ldots, x_n) \, . \tag{3.4}$$

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

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

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

$$\begin{aligned} A_k &= X_k^{-1}AX_k &&= (L_kX_{k+1}^{-1})AX_k &&= L_kR_k \quad , \\ A_{k+1} &= (X_{k+1}^{-1}A)X_{k+1} &&= (R_kX_k^{-1})X_{k+1} &&= R_kL_k \quad . \end{aligned}$$

Schließlich erwarten wir noch, daß gemäß der vermuteten Konvergenz $X_k \to XR$ mit $k \to \infty$

$$A_k = X_k^{-1} A X_k \to R^{-1}X^{-1}AXR = R^{-1}JR \, ,$$

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

1) $A_0 = A$.

2) Ist A_k berechnet, so bilde man die LR-Zerlegung

$$A_k = L_k R_k$$

und setze

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

3) Für große k gilt

$$A_k \sim \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & x \\ & & \ddots & \\ & \mathrm{O} & & \\ & & & \lambda_n \end{pmatrix}$$

mit irgendwelchen Elementen oberhalb der Diagonalen.

Tatsächlich kann man unter gewissen Voraussetzungen nur zeigen, daß die Diagonale von A_k gegen eine Permutation der Eigenwerte konvergiert. Wir werden das LR-Verfahren jedoch nicht weiter verfolgen. Es hat den Nachteil, daß die LR-Zerlegung ja nicht immer durchführbar ist und numerische Stabilität nur durch Pivotisierung, also Zeilenvertauschungen, erreicht wird.

Durch eine leichte Modifikation des LR-Algorithmus kommen wir zum QR-Algorithmus: Wir bestimmen $r_{1,j}, \ldots, r_{j-1,j}$ in (3.1) so, daß $x_j^{(1)}$ orthogonal zu $x_1^{(1)}, \ldots, x_{j-1}^{(1)}$ ist und danach $r_{j,j}$ so, daß $x_j^{(1)}$ die Länge 1 hat. Dann hat man wieder (3.2), aber X_1 ist jetzt eine unitäre Matrix. Die Potenzmethode lautet wieder wie in (3.3), nur daß jetzt statt der LR-Zerlegung eine QR-Zerlegung mit unitärem Faktor X_{k+1} durchgeführt wird. Wieder erwarten wir (3.4). Mit den Matrizen

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

finden wir genau wie beim LR-Verfahren

$$A_k = Q_k R_k \quad , \quad A_{k+1} = R_k Q_k \; .$$

Damit haben wir den QR-Algorithmus gefunden:

1) $A_0 = A$

2) Ist A_k berechnet, so bilde man die QR-Zerlegung

$$A_k = Q_k R_k$$

und setze

$$A_{k+1} = R_k Q_k \; .$$

3) Wir erwarten

$$A_k \to \begin{pmatrix} \lambda_1 & & x \\ & \ddots & \\ \mathrm{O} & & \lambda_n \end{pmatrix} .$$

Lemma 4.3.1 *Sei $A_k \to A$ und $Q_k R_k = A_k$ die QR-Zerlegung von A_k. Dann gilt $Q_k \to Q$, $R_k \to R$, wo $QR = A$ die QR-Zerlegung von A ist.*

Beweis: Die erste Spalte von $Q_k R_k = A_k$ lautet

$$r_{11}^{(k)} q_1^{(k)} = a_1^{(k)} \; .$$

Wegen $r_{11}^{(k)} > 0$, $\|q_1^{(k)}\|_2 = 1$ und $a_1^{(k)} \to a_1$ konvergiert $r_{11}^{(k)}$, etwa gegen r_{11}. Damit konvergiert auch $q_1^{(k)}$, etwa gegen q_1. Die zweite Spalte von $Q_k R_k = A_k$ lautet

$$r_{12}^{(k)} q_1^{(k)} + r_{22}^{(k)} q_2^{(k)} = a_2^{(k)} \; .$$

Weil Q_k unitär ist, gilt $r_{12}^{(k)} = (a_2^{(k)}, q_1^{(k)}) \to r_{12}$. Also konvergiert auch $r_{22}^{(k)} q_2^{(k)}$ und wegen $\|q_2^{(k)}\|_2 = 1$ auch $r_{22}^{(k)} \to r_{22}$, mithin auch $q_2^{(k)} \to q_2$.

Es ist klar, daß man so fortfahren kann und

$$R_k \to R \quad , \quad Q_k \to Q$$

bekommt mit einer rechten Dreiecksmatrix R und einer unitären Matrix Q. Dies muß die QR-Zerlegung von A sein.

□

Satz 4.3.1 *A besitze n betragsmäßig verschiedene Eigenwerte $\lambda_1, \ldots, \lambda_n$. Sei $|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n| > 0$ und $A = XJX^{-1}$ die Jordan'sche Normalform von A. X^{-1} besitze eine LR-Zerlegung. Dann gilt für die Matrix A_k des QR-Algorithmus' für $k \to \infty$*

$$(A_k)_{i,j} \to \begin{cases} \lambda_i & , \quad j = i \\ 0 & \quad j < i \end{cases} .$$

Beweis: Sei $X^{-1} = L_- R_-$ die LR-Zerlegung von X^{-1}. Der Beweis beruht auf dem Vergleich zweier QR-Zerlegungen von A^k. Die erste dieser QR-Zerlegungen bekommen wir aus

$$A^k = XJ^kX^{-1} = XJ^kL_-R_- = XJ^kL_-J^{-k}J^kR_- .$$

Wegen unserer Numerierung der Eigenwerte ist $J^kL_-J^{-k}$ von der Form $I+F_k$ mit $F_k \to 0$. Also haben wir

$$A^k = X(I + F_k)J^kR_- .$$

Sei nun $X = QR$ die QR-Zerlegung von X. Dann gilt

$$\begin{aligned} A^k &= QR(I + F_k)J^kR_- \\ &= Q(I + G_k)RJ^kR_- \end{aligned}$$

mit $G_k = RF_kR^{-1} \to 0$. Sei $P_kS_k = I + G_k$ die QR-Zerlegung von $I + G_k$. Nach dem Lemma gilt $P_k \to I$, $S_k \to I$. Wir haben also

$$A^k = QP_kS_kRJ^kR_- \tag{3.5}$$

mit unitären Matrizen $P_k \to I$ und rechten Dreiecksmatrizen $S_k \to I$. Die zweite QR-Zerlegung von A^k ist

$$A^k = Q_0 \cdots Q_{k-1}R_{k-1} \cdots R_0 . \tag{3.6}$$

Dies ist der Fall $\ell = k$ der Identität

$$Q_0 \cdots Q_{k-\ell-1} A_{k-\ell}^{\ell} R_{k-\ell-1} \cdots R_0 = Q_0 \cdots Q_{k-1} R_{k-1} \cdots R_0 \ ,$$

die man für $\ell = 0, \ldots, k$ folgendermaßen beweist. Für $\ell = 0$ ist die Identität trivial. Es genügt also zu zeigen, daß die linke Seite von ℓ unabhängig ist, also

$$Q_{k-\ell-1} A_{k-\ell}^{\ell} R_{k-\ell-1} = A_{k-\ell-1}^{\ell+1} \ .$$

Wegen $A_{k-\ell} = R_{k-\ell-1} Q_{k-\ell-1}$, $A_{k-\ell-1} = Q_{k-\ell-1} R_{k-\ell-1}$ bestätigt man dies unmittelbar.

Aus dem Vergleich der QR-Zerlegungen (3.5), (3.6) bekommen wir

$$Q_0 \cdots Q_{k-1} = Q P_k \quad , \quad R_{k-1} \cdots R_0 = S_k R J^k R_- \ .$$

Aus der ersten dieser Beziehungen folgt

$$Q_k = (Q_0 \cdots Q_{k-1})^{-1} Q_0 \cdots Q_k = (Q P_k)^{-1} Q P_{k+1} \to I \ , \tag{3.7}$$

aus der zweiten

$$\begin{aligned} R_k = R_k R_{k-1} \cdots R_0 (R_{k-1} \cdots R_0)^{-1} &= S_{k+1} R J^{k+1} R_- R_-^{-1} J^{-k} R^{-1} S_k^{-1} \\ &= S_{k+1} R J R^{-1} S_k^{-1} \to R J R^{-1} \ . \end{aligned} \tag{3.8}$$

Also gilt

$$A_k = Q_k R_k \to R J R^{-1} \ ,$$

und dies ist eine Matrix der behaupteten Art.

□

Bemerkungen: Sind die Voraussetzungen des Satzes nicht erfüllt, so treten folgende Änderungen ein.

I. Ist λ_i mehrfacher Eigenwert mit $\sigma(\lambda_i) = \rho(\lambda_i)$, so gilt der Satz unverändert.

Dann ist nämlich die Matrix $J^k L_- J^{-k}$ von der Form $L + F_k$ mit einer linken Dreiecksmatrix L, und (3.5) gilt nach wie vor, wobei jetzt $P_k \to P$, $S_k \to S$ mit einer unitären Matrix P und einer rechten Dreiecksmatrix S ist, welche nicht mehr notwendig I sind. (3.7) gilt dann nach wie vor, während in (3.8) R durch SR ersetzt werden muß.

II. Wir lassen jetzt die Vorraussetzungen, daß die algebraische mit der geometrischen Vielfachheit übereinstimmt, fallen. A besitze also r Eigenwerte $\lambda_1, \ldots, \lambda_r$ mit $|\lambda_1| > |\lambda_2| > \cdots |\lambda_r| > 0$, und zu jedem Eigenwert λ_ℓ gehören

ein oder mehrere Jordankästchen, die wir zu J_ℓ zusammenfassen. J_ℓ hat dann die Gestalt

$$J_\ell = \begin{pmatrix} \lambda_\ell & \theta_1 & & \\ & \ddots & \ddots & \\ & & & \theta_\ell \\ & & & \lambda_\ell \end{pmatrix} \quad , \quad J = \begin{pmatrix} J_1 & & \\ & \ddots & \\ & & J_r \end{pmatrix} ,$$

wo $\theta_\ell = 0$ oder $\theta_\ell = 1$. J_ℓ ist ein $\sigma(\lambda_\ell) \times \sigma(\lambda_\ell)$-Matrix. Spaltet man L_- gemäß J auf, $L_- = (L_{i,j})$, so wird

$$\begin{aligned} \left(J^k L_- J^{-k}\right)_{i,j} &= J_i^k L_{i,j} J_j^{-k} \\ &= \left(\frac{\lambda_i}{\lambda_j}\right)^k M_{i,k} L_{ij} M_{j,k}^{-1} , \quad i > j \quad , \end{aligned}$$

wobei $M_{i,k}$, $M_{i,k}^{-1}$ Polynome höchstens vom Grade $< \sigma(\lambda_i)$ in k sind. Also gilt für $k \to \infty$ nach wie vor

$$\left(J^k L_- J^{-k}\right)_{i,j} \to 0 \quad , \quad i > j \, .$$

Die weitere (recht mühsame) Untersuchung zeigt nun, daß A_k entsprechend J aufgeteilt werden kann in Blöcke A_{ijk} der Dimension $\sigma(\lambda_i) \times \sigma(\lambda_j)$, wobei für $k \to \infty$

$$A_{ijk} \to 0 \qquad \text{für} \quad i > j \, ,$$

Alle $\sigma(\lambda_i)$ Eigenwerte von A_{iik} konvergieren gegen λ_i.

III. Den Fall betragsgleicher aber verschiedener Eigenwerte brauchen wir nicht zu betrachten, da wir ihn durch *shifts* vermeiden.

4.4 Praktische Durchführung des QR-Algorithmus

Die QR-Zerlegung nach Householder benötigt $\frac{2}{3}n^3$ *flops*, das Produkt RQ deren $\frac{1}{2}n^3$. Damit verlangt ein einziger QR-Schritt $\frac{7}{6}n^3$ *flops*.

Eine erhebliche Reduktion dieser Anzahl erreicht man, wenn man den QR-Algorithmus auf Hessenberg-Matrizen anwendet. Für eine Hessenberg-Matrix $H = (\ell_{i,j})$ ist $h_{i,j} = 0$ für $i > j + 1$. Man sieht leicht, daß ein QR-Schritt eine Hessenberg-Matrix immer wieder in eine solche überführt. Nach Aufgabe 15 benötigt die QR-Zerlegung für eine Hessenberg-Matrix nur $2n^2 + 0(n)$ flops.

Ist A eine beliebige (n, n)-Matrix, so kann man mit Hilfe des Householder-Verfahrens eine unitäre Matrix U berechnen, so daß $H = UAU^*$ eine Hessenberg - Matrix ist. Dazu setzt man $U = U_{n-1} \cdots U_1$, wobei U_i die Gestalt

$$U_i = \begin{pmatrix} I_i & \mathbf{O} \\ \mathbf{O} & S_i \end{pmatrix}$$

mit der i-dimensionalen Einheitsmatrix I_i und einer $(n-i)$-dimensionalen Spiegelung S_i ist. S_i wird so bestimmt, daß der $(n-i)$-dimensionale Vektor in der i-ten Spalte von $U_{i-1}\cdots U_1 A U_1^*\cdots U_{i-1}^*$ in und unterhalb der Hauptdiagonalen in ein Vielfaches des $(n-i)$-dimensionalen Einheitsvektors e_1 abgebildet wird.

Der QR-Schritt für Hessenberg-Matrizen kann durch die Givens-Rotation

$$U_{i,k} = \begin{pmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & c & & s & & \\ & & & \ddots & & & \\ & & -s & & c & & \\ & & & & & \ddots & \\ & & & & & & 1 \end{pmatrix}$$

ausgeführt werden. Die nichttrivialen Elemente $c = \cos(\varphi)$, $s = \sin(\varphi)$ stehen dabei in Zeilen und Spalten i, k. $U_{i,k}$ vermittelt eine Drehung in der $i-k$-Ebene um den Winkel φ. Wir verwenden nur $U_{k,k+1}$ mit den Elementen $c_k = \cos(\varphi_k)$, $s_k = \sin(\varphi_k)$. In einem ersten Schritt überschreibt man die Hessenberg-Matrix H durch den rechten Faktor R in der QR-Zerlegung $H = QR$:

$$\begin{array}{lll} \text{for} & k & = 1,\ldots,n-1 \\ & \{ & \text{Bestimme } \varphi_k \text{ so, daß } \begin{pmatrix} c_k & s_k \\ -s_k & c_k \end{pmatrix} \begin{pmatrix} h_{k,k} \\ h_{k+1,k} \end{pmatrix} = \begin{pmatrix} \times \\ 0 \end{pmatrix} \\ & & \text{for } j = k,\ldots,n \\ & & \begin{pmatrix} h_{k,j} \\ h_{k+1,j} \end{pmatrix} = \begin{pmatrix} c_k & s_k \\ -s_k & c_k \end{pmatrix} \begin{pmatrix} h_{k,j} \\ h_{k+1,j} \end{pmatrix} \\ & \} & \end{array}$$

Der unitäre Faktor Q ist dann natürlich $U_{1,2}^*\cdots U_{n-1,n}^*$. In einem zweiten Schritt wird RQ gebildet:

$$\begin{array}{l} \text{for } \; k = 1,\ldots,n-1 \\ \text{for } \; j = 1,\ldots,k+1 \\ \qquad (h_{j,k}, h_{j,k+1}) = (h_{j,k}, h_{j,k+1}) \begin{pmatrix} c_k & -s_k \\ s_k & c_k \end{pmatrix} . \end{array}$$

Jeder dieser Schritte benötigt $2n^2$, insgesamt also $4n^2$ *flops*. Das ist erheblich günstiger als die $\frac{7}{6}n^3$ *flops* eines QR-Schrittes mit einer allgemeinen Matrix.

Zur Konvergenzbeschleunigung führt man *shifts* durch, und zwar in der Form

$$\begin{array}{lll} A_k - \sigma_k I & = & Q_k R_k \\ A_{k+1} & = & R_k Q_k + \sigma_k I \; . \end{array}$$

Für σ_k verwendet man eine Näherung für den betragskleinsten Eigenwert, etwa das (n,n)-Element von A_k. Das führt zu einer schnellen Konvergenz von λ_n. Danach spaltet man Zeile und Spalte n ab und führt den QR-Algorithmus für die verbleibende $(n-1,n-1)$-Matrix weiter.

4.5 Fehlerabschätzung bei Eigenwertproblemen

Wir wollen zunächst einen Satz kennenlernen, der ohne viel Rechnung die grobe Lokalisierung der Eigenwerte einer Matrix gestattet.

Satz 4.5.1 *(Gerschgorin) Sei A (n,n)-Matrix und seien*

$$\begin{aligned} r_i &= \sum_{j\neq i} |a_{ij}| \\ K_i &= \{z \in C : |z - a_{ii}| \leq r_i\} \ . \end{aligned}$$

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 Eigenwerte zusammen genau die algebraische Vielfachheit m.

Beispiel:

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

Es liegen Eigenwerte jeweils der Gesamtvielfachheit 2 in $K_1 \cup K_4$ und $K_2 \cup K_3$.

Beweis:

(a) Sei λ Eigenwert und x Eigenvektor von A mit $\|x\|_\infty = 1 = |x_{i_0}|$. $Ax = \lambda x$ bedeutet

$$(a_{i,i} - \lambda)x_i = -\sum_{\substack{j=1 \\ j\neq i}}^{n} a_{i,j}x_j \ ,$$

also

$$|a_{i,i} - \lambda||x_i| \leq \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{i,j}||x_j| \leq r_i \ .$$

Für $i = i_0$ folgt $\lambda \in K_{i_0}$.

(b) Der Beweis beruht auf einem Stetigkeitsargument. Wir geben nur die Idee. Für eine exakte Fassung siehe etwa J. Werner, Kapitel 5.1.

Wir setzen $A(t) = D + t(A - D)$ mit der Diagonalen D von A. Dann ist $A(0) = D$, $A(1) = A$. Die Gerschgorin-Kreise von $A(t)$ sind $a_{ii} + tK_i$. Wir beweisen (b) für alle $A(t)$ mit $0 \leq t \leq 1$. Wir benutzen folgendes

Lemma 4.5.1 *Es gibt n stetige Funktionen $\lambda_1, \ldots, \lambda_n$ auf $[0,1]$, so daß $\lambda_1(t), \ldots, \lambda_n(t)$ die Eigenwerte von $A(t)$ sind.*

Nun argumentiert man folgendermaßen. Für $t = 0$ ist (b) offenbar richtig. Wir wählen nun $t_0 > 0$ so, daß für $t \leq t_0$ die Kreise $a_{ii} + tK_i$, $a_{ii} + tK_j$ für alle i, j mit $a_{i,i} \neq a_{j,j}$ punktfremd sind. Dann müssen für $t \leq t_0$ wegen des Lemmas in $a_{ii} + tK_i$ genau so viele $\lambda_j(t)$ liegen, wie es j gibt mit $a_{j,j} = a_{i,i}$. Also ist (b) richtig für $A(t)$ mit $t \leq t_0$.

Lassen wir jetzt t weiter anwachsen bis zum ersten Mal einige der bisher punktfremden $a_{ii} + tK_i$ zusammenstoßen. Die so entstehende Vereinigung von $a_{ii} + tK_i's$ enthält dann genau diejenigen Eigenwerte, welche bisher in den einzelnen $a_{ii} + tK_i's$ lagen. So fortfahrend erhält man schließlich das Resultat für $A(t)$ in $0 \leq t \leq 1$.

□

Für hermite'sche Matrizen kann man einfache und befriedigende Aussagen über die Lage von Eigenwerten machen. Zunächst haben wir folgenden Einschließungssatz.

Satz 4.5.2 *Sei A hermite'sch und seien λ, x Näherungen für einen Eigenwert von A mit zugehörigem Eigenvektor. Sei $d = Ax - \lambda x$.*

Dann gibt es einen Eigenwert λ_k von A mit

$$|\lambda_k - \lambda| \leq \frac{\|d\|_2}{\|x\|_2} .$$

Beweis: Sei $\{x_1, \ldots, x_n\}$ ein Orthonormalsystem von Eigenvektoren von A. Dann gilt

$$x = \sum_{i=1}^{n} c_i x_i, c_i = x_i^* x , \quad \|x\|_2^2 = \sum_{i=1}^{n} |c_i|^2$$

und

$$d = Ax - \lambda x = \sum_{i=1}^{n} c_i(\lambda_i - \lambda)x_i ,$$

$$\|d\|_2^2 = \sum_{i=1}^{n} |c_i|^2 |\lambda_i - \lambda|^2 \geq \sum_{i=1}^{n} |c_i|^2 \min_{1 \leq i \leq n} |\lambda_i - \lambda|^2 = \|x\|_2^2 |\lambda_k - \lambda|^2 .$$

□

Im allgemeinen Fall sind Störungsresultate sehr viel verwickelter und ungünstiger.

Satz 4.5.3 *Sei A eine (n,n)-Matrix mit Eigenwerten $\lambda_1,\ldots,\lambda_r$, und sei ν die maximale Länge der Jordan-Kästchen von A. Sei X eine Matrix, welche A auf Jordan'sche Normalform bringt. Sei $A_\varepsilon = A+\varepsilon F$, $0 \le \varepsilon \le 1$. Dann liegen sämtliche Eigenwerte von A_ε in der Vereinigung der Kreise*

$$K_\ell = \{z \in \mathbb{C} : |z-\lambda_\ell| \le \varepsilon^{1/\nu}(1+k_\infty(X))\|F\|_\infty\} .$$

Beweis: Es ist $A = XJX^{-1}$. Also haben $A+\varepsilon F$, $J+\varepsilon G$ mit $G = X^{-1}FX$ 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

$$J = \begin{pmatrix} \lambda & 1 & & & \\ & \lambda & 1 & & \\ & & \ddots & \ddots & \\ & & & & 1 \\ & & & & \lambda \end{pmatrix} \quad , \quad D^{-1}JD = \begin{pmatrix} \lambda & \varepsilon^{1/\nu} & & & \\ & \lambda & \varepsilon^{1/\nu} & & \\ & & \ddots & \ddots & \\ & & & & \varepsilon^{1/\nu} \\ & & & & \lambda \end{pmatrix} .$$

Die Matrix $D^{-1}(J+\varepsilon G)D$ hat die gleichen Eigenwerte wir $A+\varepsilon F$. Wir wenden den Satz von Gerschgorin auf $D^{-1}(J+\varepsilon G)D$ an. Die Gerschgorin-Kreise haben Mittelpunkt $\lambda+\varepsilon g_{i,i}$ und Radius

$$r_i \le \varepsilon^{1/\nu} + \varepsilon\varepsilon^{(1-\nu)/\nu}\sum_{\substack{j=1\\ j\ne i}}^{n}|g_{ij}| = \varepsilon^{1/\nu}\left(1+\sum_{\substack{j=1\\ j\ne i}}^{n}|g_{ij}|\right) .$$

Jeder Eigenwert μ von A_ε liegt in einem dieser Kreise, also

$$|\lambda+\varepsilon g_{ii}-\mu| \le r_i$$

für ein i. Es folgt

$$\begin{aligned} |\lambda-\mu| &\le \varepsilon|g_{ii}| + r_i \\ &\le \varepsilon^{1/\nu}\left(1+\sum_{j=1}^{n}|g_{ij}|\right) \\ &\le \varepsilon^{1/\nu}(1+\|G\|_\infty) \\ &\le \varepsilon^{1/\nu}(1+k_\infty(X)\|F\|_\infty) . \end{aligned}$$

2) J besteht aus mehreren Jordan-Kästchen $J_1,\ldots,J_r$. Dann setzt man D aus Diagonalmatrix $D_1,\ldots,D_r$ zusammen und hat dann

$$D^{-1}(J+\varepsilon G)D = \begin{pmatrix} D_1^{-1}J_1D_1 & & \\ & \ddots & \\ & & D_r^{-1}J_rD_r \end{pmatrix} + \varepsilon D^{-1}GD .$$

Anwendung von 1) ergibt die Behauptung.

□

Kapitel 5

Interpolation

5.1 Interpolation durch Polynome

Sei $\mathcal{P}_n$die Menge der Polynome mit komplexen Koeffizienten vom Grade $\leq n$. $\mathcal{P}_n$ besteht also aus den Ausdrücken der Form

$$\sum_{k=0}^{n} a_k x^k \quad , \quad a_k \in \mathbb{C} \, .$$

Als Polynominterpolation bezeichnet man folgende Aufgabe: Gegeben seien $n+1$ "Stützstellen" $x_0, \ldots, x_n \in \mathbb{C}$ und ebensoviel "Stützwerte" $y_0, \ldots, y_n \in \mathbb{C}$. Gesucht ist $p \in \mathcal{P}_n$ mit $p(x_j) = y_j$, $j = 0, \ldots, n$.

Satz 5.1.1 *p ist eindeutig bestimmt, falls die x_j paarweise verschieden sind.*

Beweis: Das Interpolationsproblems ist äquivalent dem linearen Gleichungssystem

$$\sum_{k=0}^{n} a_k x_j^k = y_j \quad , \quad j = 0, \ldots, n \tag{1.1}$$

für $a_0, \ldots, a_n$. Wir zeigen, daß dieses eindeutig lösbar ist. Dazu ist hinreichend, daß das zugehörige homogene System, also das System mit $y_0 = y_1 = \cdots = y_n = 0$, nur die Lösung $a_0 = a_1 = \cdots a_n = 0$ hat. Dies ist aber der Fall, weil ein Polynom vom Grade $\leq n$ nicht $n+1$ paarweise verschiedene Nullstellen haben kann ohne identisch zu verschwinden.

□

Zur Berechnung des interpolierenden Polynoms könnte man das lineare Gleichungssystem (1.1) lösen, etwa durch die Cramer'sche Regel. Die Determinante von (1.1) ist die Vandermond'sche Determinante

$$V = \begin{vmatrix} 1 & \cdots & 1 \\ x_0 & & x_n \\ \vdots & & \vdots \\ x_0^n & & x_n^n \end{vmatrix} = \prod_{i>k} (x_i - x_k) \; .$$

Diese ist also $\neq 0$, falls die x_j paarweise verschieden sind, was einen neuen Beweis von Satz 1 liefert. Wir ersetzen in V die k-te Spalte durch die Zahlen $y_0, \ldots, y_n$ und bezeichnen das Resultat mit V_k. Die Lösung des Interpolationsproblems ist dann

$$p(x) = \sum_{k=0}^{n} a_k x^k \quad , \quad a_k = \frac{V_k}{V} \; , \quad k = 0, \ldots, n \; .$$

Für das praktische Rechnen ist diese Lösung völlig ungeeignet. Wir werden drei sehr viel praktischere Darstellungen von p finden.

1. Die Form von Lagrange

Man setzt

$$\omega_j(x) = \prod_{i=0,\, i \neq j}^{n} \frac{x - x_i}{x_j - x_i} \quad , \quad j = 0, \ldots, n \; .$$

Es ist $\omega_j \in \mathcal{P}_n$ Lösung des speziellen Interpolationsproblems

$$\omega_j(x_k) = \begin{cases} 1 & , \quad k = j \; , \\ 0 & , \quad \text{sonst} \; . \end{cases}$$

Für das gesuchte Polynom gilt dann

$$p(x) = \sum_{j=0}^{n} y_j \omega_j(x) \; .$$

Beispiel: $n = 2$. Stützstellen und Stützwerte seien

j	x_j	y_j
0	0	1
1	1	3
2	3	2

Wir berechnen

$$\begin{aligned} \omega_0(x) &= \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} &&= \tfrac{1}{3}(x-1)(x-3) \\ \omega_1(x) &= \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} &&= -\tfrac{1}{2}x(x-3) \\ \omega_2(x) &= \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} &&= \tfrac{1}{6}x(x-1) \end{aligned}$$

und erhalten

$$p(x) = \frac{1}{3}(x-1)(x-3) - \frac{3}{2}x(x-3) + \frac{1}{3}x(x-1) = -\frac{5}{6}x^2 + \frac{17}{6}x + 1\,.$$

2. Die Rekursionsformel von Neville

Wir bezeichnen mit $p_{i,\ldots,k}, i \le k$, das nach Satz 1 eindeutig bestimmte Polynom $\in \mathcal{P}_{k-i}$, welches $y_i, \ldots, y_k$ an den Stellen $x_i, \ldots, x_k$ interpoliert. Das gesuchte Polynom ist dann $p = p_{0,\ldots,n}$.

Satz 5.1.2 *Es ist $p_i = y_i$ und für $i < k$*

$$p_{i,\ldots,k}(x) = \frac{1}{x_k - x_i}\ \left((x - x_i)p_{i+1,\ldots,k}(x) + (x_k - x)p_{i,\ldots,k-1}(x)\right)\ .$$

Beweis: Auf der rechten Seite steht ein Polynom vom Grade $k-i$, das an den Stellen $x_i, \ldots, x_k$ die Werte $y_i, \ldots, y_k$ annimmt. Nach Satz 1 ist dieses eindeutig bestimmt und muß daher mit $p_{i,\ldots,k}$ übereinstimmen.

□

Die Neville'sche Formel erlaubt die Berechnung des Interpolationspolynoms p nach folgendem Schema ($n = 3$):

$$\begin{array}{lllllll}
x_0 & y_0 = p_0 & & & & \\
 & & p_{0,1} & & & \\
x_1 & y_1 = p_1 & & p_{0,1,2} & & \\
 & & p_{1,2} & & p_{0,1,2,3} = p & \\
x_2 & y_2 = p_2 & & p_{1,2,3} & & \\
 & & p_{2,3} & & & \\
x_3 & y_3 = p_3 & & & &
\end{array}$$

Bei Hinzunahme einer weiteren Stützstelle (und Erhöhung des Polynomgrads) wird das Schema einfach um eine Zeile erweitert, ohne daß die bereits berechneten Teile des Schemas neu berechnet werden müßten.

3. Die Newton'sche Form

Für $p = p_{0,\ldots,n}$ machen wir den Ansatz

$$\begin{aligned} p_{0,\ldots,n}(x) \;=\; & A_0 + A_1(x - x_0) + A_2(x - x_0)(x - x_1) + \cdots \\ & + A_n(x - x_0)(x - x_1)\cdots(x - x_{n-1}) \end{aligned}$$

mit noch zu bestimmenden Koeffizienten $A_0, \ldots, A_n$. Die Bedingungen $p(x_j) = y_j,\ j = 0, \ldots, n$ ergeben für die A_k das lineare Gleichungssystem

$$\begin{aligned}
A_0 &= y_0\,, \\
A_0 + A_1(x_1 - x_0) &= y_1\,, \\
A_0 + A_1(x_2 - x_0) + A_2(x_2 - x_0)(x_2 - x_1) &= y_2\,, \\
&\cdots \\
A_0 + A_1(x_n - x_0) + \cdots + A_n(x_n - x_0)\cdots(x_n - x_{n-1}) &= y_n\,.
\end{aligned}$$

Die A_k können also durch Vorwärtseinsetzen berechnet werden. Für das Resultat gibt es eine elegante Darstellung durch "dividierte Differenzen" $[y_i, \ldots, y_k]$. Diese sind rekursiv definiert durch

$$\begin{aligned}
[y_i] &= y_i\,, \\
[y_i, \ldots, y_k] &= \frac{1}{x_k - x_i}([y_{i+1}, \ldots, y_k] - [y_i, \ldots, y_{k-1}])\,.
\end{aligned}$$

Z.B. ist

$$\begin{aligned}
[y_0, y_1] &= \frac{y_1 - y_0}{x_1 - x_0}\,, \\
[y_0, y_1, y_2] &= \frac{1}{x_2 - x_0}\left(\frac{y_1 - y_2}{x_1 - x_2} - \frac{y_0 - y_1}{x_0 - x_1}\right)\,.
\end{aligned}$$

Man ordnet die dividierten Differenzen im "Differenzenschema" an:

$$\begin{array}{llll}
x_0 & [y_0] & & \\
 & & [y_0, y_1] & \\
x_1 & [y_1] & & [y_0, y_1, y_2] \\
 & & [y_1, y_2] & \\
x_2 & [y_2] & & [y_1, y_2, y_3] \\
 & & [y_2, y_3] & \\
x_3 & [y_3] & &
\end{array}$$

Satz 5.1.3 *Für die Koeffizienten A_i gilt*

$$A_i = [y_0, \ldots, y_i]\,, \qquad i = 0, \ldots, n\,.$$

Beweis: Wir zeigen

$$\begin{aligned}
p_{i,\ldots,k}(x) &= [y_i] + [y_i, y_{i+1}](x - x_i) + [y_i, y_{i+1}, y_{i+2}](x - x_i)(x - x_{i+1}) \\
&\quad + \cdots + [y_i, \ldots, y_k](x - x_i)\cdots(x - x_{k-1})
\end{aligned} \tag{1.2}$$

durch Induktion nach k. Für $k = i$ ist (1.2) richtig. Sei (1.2) richtig für ein $k \geq i$. Dann ist

$$\begin{aligned}
p_{i,\ldots,k}(x) &= [y_i, \ldots, y_k](x - x_i)\cdots(x - x_{k-1}) + q_1(x) \\
&= [y_i, \ldots, y_k]x^{k-i} + q_2(x)
\end{aligned}$$

mit $q_\ell \in \mathcal{P}_{k-i-1}$, und ebenso gilt

$$p_{i+1,\ldots,k+1}(x) = [y_{i+1}, \ldots, y_{k+1}]x^{k-i} + q_3(x) \; .$$

Nach Satz 2 ist mit $r_\ell \in \mathcal{P}_{k-i}$

$$\begin{aligned} p_{i,\ldots,k+1}(x) &= \frac{1}{x_{k+1} - x_i}((x - x_i)p_{i+1,\ldots,k+1}(x) + (x_{k+1} - x)p_{i,\ldots,k}(x)) \\ &= \frac{1}{x_{k+1} - x_i}([y_{i+1}, \ldots, y_{k+1}] - [y_i, \ldots, y_k])x^{k-i+1} + r_1(x) \\ &= [y_i, \ldots, y_{k+1}]x^{k-i+1} + r_1(x) \\ &= [y_i, \ldots, y_{k+1}](x - x_i) \cdots (x - x_k) + r_2(x) \; . \end{aligned}$$

Offenbar muß $r_2(x_j) = y_j, \;\; j = i, \ldots, k$ sein. Nach Satz 1 ist also $r_2 = p_{i,\ldots,k}$. Da für $p_{i,\ldots,k}$ (1.2) bereits gilt, gilt (1.2) auch für $p_{i,\ldots,k+1}$.

□

Beispiel: Wir haben oben das Interpolationsproblem

j	x_j	y_j
0	0	1
1	1	3
2	3	2

durch die Lagrange'sche Form des Interpolationspolynoms gelöst. Zur Lösung des gleichen Problems mit der Newton'schen Form stellen wir zunächst einmal das Differenzenschema auf:

$$\begin{array}{cccc} 0 & 1 & & \\ & & 2 & \\ 1 & 3 & & -5/6 \\ & & -1/2 & \\ 3 & 2 & & \end{array}$$

Die Koeffizienten des Newton'schen Interpolationspolynoms stehen in der obersten Zeile:

$$p(x) = 1 + 2x - \frac{5}{6}x(x-1) = 1 + \frac{17}{x}x - \frac{5}{6}x^2 \; .$$

Fügen wir noch die Stützstelle $x_3 = 3$ mit dem Stützwert $y_3 = 4$ hinzu, so lautet das erweiterte Differenzenschema

$$\begin{array}{cccccc} 0 & 1 & & & & \\ & & 2 & & & \\ 1 & 3 & & -5/6 & & \\ & & -1/2 & & -1/3 & \\ 3 & 2 & & -3/2 & & \\ & & -2 & & & \\ 2 & 4 & & & & \end{array}$$

und das zugehörige Interpolationsproblem ist

$$p(x) = 1 + 2x - \frac{5}{6}x(x-1) - \frac{1}{3}x(x-1)(x-3) \; .$$

5.2 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 5.2.1 *Zu jedem $x \in [a,b]$ existiert ein $\tilde{x} \in [a,b]$, so daß gilt:*

$$f(x) - p(x) = w(x)\frac{f^{(n+1)}(\tilde{x})}{(n+1)!}$$

mit

$$w(x) = \prod_{j=0}^{n}(x - x_j)$$

Beweis: Sei $\overline{x} \in [a,b]$ beliebig gewählt mit $\overline{x} \neq x_j$, $j = 0,\ldots,n$. Wir setzen

$$F(x) = f(x) - p(x) - Kw(x)$$

mit einer Konstanten K. Dann ist

$$F(x_j) = 0 \quad , \quad j = 0,\ldots,n\,.$$

Wir wählen K nun so, daß auch $F(\overline{x})$ verschwindet, also

$$K = \left(\frac{f-p}{w}\right)(\overline{x})\,.$$

Damit hat F in $[a,b]$ mindestens die $n+2$ Nullstellen $\overline{x}$, $x_0,\ldots,x_n$. Durch wiederholte Anwendung des Satzes von Rolle folgt, daß $F^{(n+1)}$ mindestens eine Nullstelle $\tilde{x}$ in $[a,b]$ besitzt.
Aus der Beziehung

$$F^{(n+1)}(x) = f^{(n+1)}(x) - K(n+1)!$$

folgt

$$K = \left(\frac{f-p}{w}\right)(\overline{x}) = \frac{f^{(n+1)}(\tilde{x})}{(n+1)!}\,.$$

Also

$$f(\overline{x}) - p(\overline{x}) = w(\overline{x})\,\frac{f^{(n+1)}(\tilde{x})}{(n+1)!}\,.$$

Diese Beziehung ist offenbar auch für $\overline{x} = x_j$, $j = 0,\ldots,n$ erfüllt.

□

Für den Interpolationsfehler erhalten wir die Abschätzung

$$|f(x) - p(x)| \leq |w(x)| \max_{x \in [a,b]} \frac{|f^{(n+1)}(x)|}{(n+1)!} .$$

1) Wir betrachten zuerst den Fall gleichmäßig verteilter Stützstellen $x_j = a + jh$ mit der "Schrittweite" $h = \frac{b-a}{n}$. Für $x = a + \theta h$, $0 \leq \theta \leq n$ gilt

$$w(x) = h^{n+1} \prod_{j=0}^{n} (\theta - j) ,$$

also

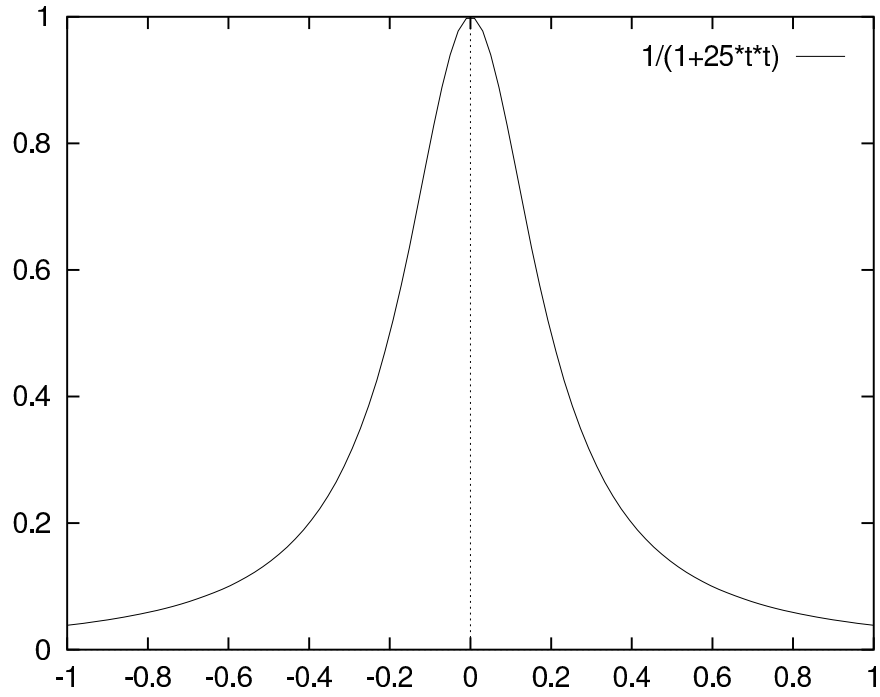
$$|f(x) - p(x)| \leq \frac{h^{n+1}}{(n+1)!} \left(\prod_{j=0}^{n} |\theta - j| \right) \max_{x \in [a,b]} |f^{(n+1)}(x)| .$$

(a) Für $b - a \to 0$ bei festem n erhalten wir

$$f(x) - p(x) = \mathrm{O}(h^{n+1}) .$$

(b) Der Fall $n \to \infty$ bei fester Intervallänge $b - a$ führt i.a. zu keiner Konvergenz. Wir betrachten hierzu das Beispiel von Runge:

$$f(x) = (1 + 25x^2)^{-1} \quad \text{in } [-1, 1]$$



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:

$$T_n(x) = \cos nt \quad , \quad x = \cos t \quad , \quad 0 \le t \le \pi \ .$$

In IV.3 haben wir gesehen, daß

$$T_{n+1} = 2xT_n - T_{n-1} \ .$$

Die Rekursion zeigt, daß T_n für $n \ge 1$ die Form

$$T_n(x) = 2^{n-1}x^n + \text{Polynom} \quad \in \mathcal{P}_{n-1}$$

haben muß.

Daher hat $w(x) = 2^{-n}T_{n+1}(x)$ den Höchstkoeffizienten 1 und es gilt

$$|w(x)| \le 2^{-n} \quad \text{in } [-1,1] \ .$$

Die Nullstellen $x_j = \cos \frac{(j+1/2)\pi}{(n+1)}$, $j = 0, \ldots, n$ von w sind unsere Stützstellen.

Bei dieser Wahl ergibt sich die Fehlerabschätzung

$$|f(x) - p(x)| \le \frac{\text{Max}|f^{(n+1)}(x)|}{2^n (n+1)!} \ .$$

Für das obige Beispiel ergeben sich folgende Werte:

n	$\max_{x\in[-1,1]} \lvert f(x)-p(x)\rvert$
1	0.93
5	0.56
13	0.12
19	0.04

Die Verbesserung ist erheblich, die Approximation aber immer noch unbefriedigend.

5.3 Trigonometrische Interpolation

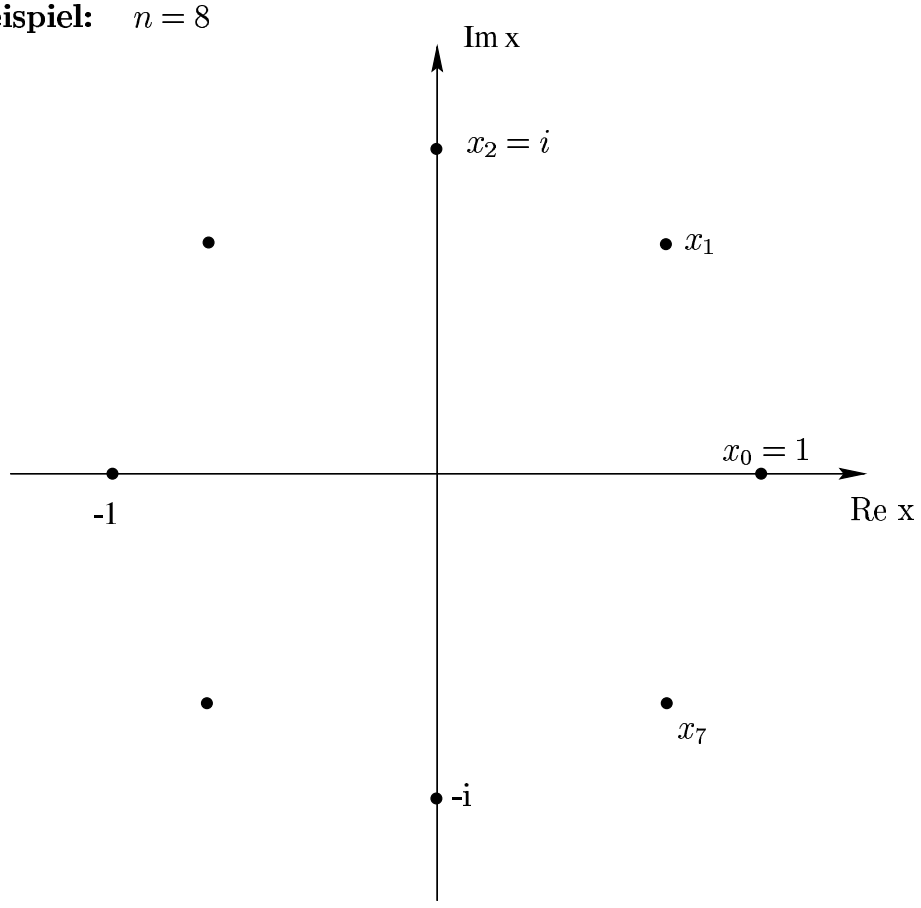
Wir betrachten in diesem Abschnitt einen wichtigen Spezialfall der Polynominterpolation, bei dem die Stützstellen in regelmäßigen Abständen auf dem komplexen Einheitskreis liegen.

Diese spezielle Problemstellung wird zu einem der wichtigsten Hilfsmittel der angewandten Mathematik führen: Zur diskreten Fouriertransformation.

Die Stützstellen seien gegeben durch

$$x_j = e^{tj} = \cos t_j + i \sin t_j \; ; \quad t_j = 2\pi j/n \; ; \quad j = 0, \ldots, n-1 \; .$$

Beispiel: $n = 8$



Nach Satz 5.1.1 gibt es ein eindeutig bestimmtes Polynom $p \in \mathcal{P}_{n-1}$ mit $p(x_j) = y_j, \; j = 0, \ldots, n-1$.

Die Koeffizienten von p bezeichnen wir mit $\hat{y}_k$:

$$p(x) = \sum_{k=0}^{n-1} \hat{y}_k x^k$$

Seien $y = (y_0, \ldots, y_{n-1})^T \in \mathbb{C}^n$ und $\hat{y} = (\hat{y}_0, \ldots, \hat{y}_{n-1})^T \in \mathbb{C}^n$. Dann können

wir die Interpolationsaufgabe

$$y_j = \sum_{k=0}^{n-1} \hat{y}_k x_j^k \; ; \quad j = 0, \ldots, n-1$$

in die Form

$$y = \begin{pmatrix} 1 & x_0 & \ldots & x_0^{n-1} \\ 1 & x_1 & \ldots & x_1^{n-1} \\ \vdots & & \ddots & \\ 1 & x_{n-1} & \ldots & x_{n-1}^{n-1} \end{pmatrix} \hat{y} = W\hat{y} \; .$$

umschreiben. Die Inversion der Matrix W erweist sich als sehr einfach:

Satz 5.3.1 *Mit obigen Definitionen für x_j, $j = 0, \ldots, n-1$ gilt für die Matrix W:*

$$WW^* = nI \; .$$

Beweis: Für $k, \ell = 0, \ldots, n-1$ gilt:

$$\begin{aligned} (WW^*)_{k\ell} &= \sum_{j=0}^{n-1} x_k^j \overline{x}_\ell^j \\ &= \sum_{j=0}^{n-1} e^{i(t_k - t_\ell)j} \\ &= \sum_{j=0}^{n-1} e^{2\pi i(k-\ell)j/n} \\ &= \sum_{j=0}^{n-1} q^j \qquad \text{mit } \; q = e^{2\pi i(k-\ell)/n} \\ &= \begin{cases} \frac{q^n - 1}{q-1} & , \text{ falls } \; q \neq 1 \\ n & , \text{ falls } \; q = 1 \end{cases} \\ &= \begin{cases} 0 & , \text{ falls } \; k \neq \ell \\ n & , \text{ falls } \; k = \ell \end{cases} \end{aligned}$$

□

Damit haben wir gleichzeitig eine Orthogonalitätseigenschaft der trigonometrischen Funktionen gezeigt:

$$\frac{1}{n} \sum_{j=0}^{n-1} e^{2\pi ijk/n} = \begin{cases} 1 & , \text{ falls } \; k \in n\mathbb{Z} \\ 0 & , \text{ sonst} \end{cases}$$

Aus Satz 5.3.1 können wir folgern:

$$W^{-1} = \frac{1}{n} W^*$$

und damit

$$\hat{y} = \frac{1}{n} W^* y \, .$$

In Komponenten:

$$\begin{aligned} \hat{y}_k &= \tfrac{1}{n} \sum_{j=0}^{n-1} e^{-2\pi i j k/n} y_j \, , \qquad (1) \\ y_j &= \sum_{k=0}^{n-1} e^{2\pi i j k/n} \hat{y}_k \quad , \qquad (2) \end{aligned}$$

Gleichung (1) heißt diskrete Fouriertransformation der Länge n, (2) heißt entsprechend inverse diskrete Fouriertransformation der Länge n. Beide werden in der Angewandten Mathematik sehr häufig benutzt. Man programmiert jedoch nicht nach den Formeln (1) und (2), was jeweils n^2 komplexe Rechenoperationen beanspruchen würde, sondern benutzt erheblich schnellere Algorithmen. Einen davon werden wir in 5.4 kennenlernen.

5.4 Schnelle Fouriertransformation

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 ist $q^n = 1$ und $q^m = q^{n/2} = -1$.

Die Idee ist nun, die Summe nach geraden und ungeraden Indizes zu zerlegen:

$$\begin{aligned} \hat{y}_k &= \sum_{\ell=0}^{m-1} q^{(2\ell)k} y_{2\ell} + \sum_{\ell=0}^{m-1} q^{(2\ell+1)k} y_{2\ell+1} \\ &= \sum_{\ell=0}^{m-1} (q^2)^{\ell k} y_{2\ell} + q^k \sum_{\ell=0}^{m-1} (q^2)^{\ell k} y_{2\ell+1} \\ &=: g_k + q^k u_k \end{aligned}$$

Wir sehen, 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 haben wir

$$\begin{aligned} g_{k+m} &= g_k \\ u_{k+m} &= u_k \end{aligned}$$

und damit für $k = 0, \ldots, m-1$

$$\begin{aligned} \hat{y}_k &= g_k + q^k u_k , \\ \hat{y}_{k+m} &= g_k + q^{k+m} u_k = g_k - q^k u_k . \end{aligned}$$

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 $A_0 = 0$, $M_0 = 0$ und

$$\begin{aligned} M_p &= 2M_{p-1} + 2^{p-1} , \\ A_p &= 2A_{p-1} + 2^p . \end{aligned}$$

Wir benutzen nun folgendes Lemma: Ist für $p = 1, 2 \ldots$

$$M_p = qM_{p-1} + r_p ,$$

so gilt

$$M_p = q^p M_0 + \sum_{j=1}^{p} r_j q^{p-j} .$$

Anwendung des Lemmas ergibt

$$\begin{aligned} M_p &= \tfrac{1}{2}p2^p &&= \tfrac{1}{2}(\log_2 n)\cdot n \\ A_p &= p2^p &&= (\log_2 n)\cdot n\,. \end{aligned}$$

Damit gilt der

Satz 5.4.1 *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.*

Die Implementierung der FFT durch ein rekursives Programm ist sehr einfach:

```
fft (y, n)
/* Führt die FFT der Länge n = 2^p durch */
{ m = n/2;
    for ℓ = 0, ..., m − 1 {
        g[ℓ] = y[2ℓ];
        u[ℓ] = y[2ℓ + 1];
    }
    if (m > 1) {
        fft (g, m);
        fft (u, m);
    }
    for k = 0, ..., m − 1 {
        u[k] = q^k * u[k];
        y[k] = g[k] + u[k];
        y[k + m] = g[k] − u[k];
    }
}
```

Die FFT nach Cooley-Tukey ist ein typisches Beispiel für das "divide and conquer"-Prinzip der Informatik:

1. Zerlege das Problem in Teilprobleme.
2. Löse die Teilprobleme.
3. Setze die Lösung der Teilprobleme zur Lösung des ganzen Problems zusammen.

Beispiele:

$$\begin{array}{lllll} n=1 & : & \hat{y}_0 & = & y_0 \\ n=2 & : & \hat{y}_0 & = & y_0+y_1 \\ & & \hat{y}_1 & = & y_0-y_1 \\ n=4 & : & q & = & p^{-2\pi i/4}=-i \\ & & g_0 & = & y_0+y_2 \qquad u_0=y_1+y_3 \\ & & g_1 & = & y_0-y_2 \qquad u_1=y_1-y_3 \\ & & \hat{y}_0 & = & g_0+u_0 \\ & & \hat{y}_1 & = & g_1-iu_1 \\ & & \hat{y}_2 & = & g_0-u_0 \\ & & \hat{y}_3 & = & g_1+iu_1 \end{array}$$

Wir können die FFT auch als Faktorisierung der Matrix W_n für die Fourier-Transformation der Länge n ansehen, also

$$W_n = \left(e^{-2\pi ik\ell/n}\right)_{k,\ell=0,\ldots,n-1} .$$

Sei P_n die Permutationsmatrix

$$P_n = \begin{pmatrix} e_0 \\ e_2 \\ \vdots \\ e_{n-2} \\ e_1 \\ e_3 \\ \vdots \\ e_{n-1} \end{pmatrix}$$

mit dem Einheitsvektor e_j^* des C^n, in dem wir die Komponenten von 0 bis $n-1$ durchnummerieren. Sei $m = n/2$ und I_m die (m,m) Einheitsmatrix, D_m die Diagonalmatrix

$$D_m = \begin{pmatrix} 1 & & & \\ & q & & \\ & & \ddots & \\ & & & q^{m-1} \end{pmatrix}, \qquad q = e^{-2\pi i/n} .$$

Dann ist

$$W_n = \begin{pmatrix} I_m & I_m \\ I_m & -I_m \end{pmatrix} \begin{pmatrix} I_m & 0 \\ 0 & D_m \end{pmatrix} \begin{pmatrix} W_m & 0 \\ 0 & W_m \end{pmatrix} P_n .$$

Die FFT ergibt sich durch sukzessives Anwenden dieser Formel. Die Durchführung der FFT verlangt also nur drei Operationen:

1) Permutationen

2) Multiplikation mit q^ℓ (Twiddle factor)

3) Die Anwendung von $\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ (Butterfly)

5.5 Splines

Ein Spline (spline (engl.) = Straklatte) ist ein dünner elastischer Stab, der zwischen Gewichte ("Knoten") gespannt Kurven darstellt. Sie wurden im Schiffsbau verwendet. Wir werden unten sehen, daß die Straklatte zwischen den Knoten durch Polynome beschrieben wird, die an den Knoten unter gewissen Differenzierbarkeitsbedingungen zusammengefügt werden. Unter Splines oder Splinefunktionen versteht man daher in der Numerik Funktionen, welche stückweise Polynome sind und die an den Nahtstellen gewissen Differenzierbarkeitsbedingungen genügen.

Definition 5.5.1 *Seien $x_0 < x_1 < \cdots < x_n$ reelle Zahlen. Eine Funktion s heißt Spline der Ordnung k zu $x_0, \ldots, x_n$, falls*

1) $s \in C^{k-2}(\mathbb{R}^1)$.

2) In jedem Intervall $[x_i, x_{i+1}]$ $i = -1, \ldots, n$ ist s ein Polynom vom Grade $\leq k-1$ ($x_{-1} = -\infty$, $x_{n+1} = +\infty$).

(Für $k = 1$ ist 1) leer.)

Beispiel: $n = 0$, $x_0 = t$,

$$f(x) = (x-t)_+^{k-1} = \begin{cases} (x-t)^{k-1} & , \quad x > t \, , \\ 0 & , \quad \text{sonst} \, . \end{cases}$$

f ist ein Spline der Ordnung k zu x_0.

Mit Hilfe von f können wir bereits sehr nützliche Splines erzeugen. Sei $x_0 < x_1 < \cdots < x_n$. Wir setzen $f_i = f(x_i)$, $i = 0, \ldots, n$. Die f_i sind Funktionen von t, was wir aber in der Bezeichnung nicht zum Ausdruck bringen. Dann sind auch die dividierten Differenzen $[f_i, \ldots, f_{i+k}]$ Funktionen von t. Wir setzen

$$B_{i,k}(t) = (x_{i+k} - x_i)[f_i, \ldots, f_{i+k}] \, . \tag{5.1}$$

Beispiel: $k = 2$, $f(x) = (x-t)_+$. Es ist

$$\begin{aligned} B_{i,2}(t) &= (x_{i+2} - x_i)[f_i, f_{i+1}, f_{i+2}] \\ &= [f_{i+1}, f_{i+2}] - [f_i, f_{i+1}] \\ &= \frac{f_{i+1} - f_{i+2}}{x_{i+1} - x_{i+2}} - \frac{f_i - f_{i+1}}{x_i - x_{i+1}} \, . \end{aligned}$$

Wir betrachten 4 Fälle.

1) $t < x_i$. Dann ist $f_i = x_i - t$, $f_{i+1} = x_{i+1} - t$, $f_{i+2} = x_{i+2} - t$, und es wird $B_{i,2}(t) = 0$.

2) $x_i \le t < x_{i+1}$. Jetzt ist $f_i = 0$, $f_{i+1} = x_{i+1} - t$, $f_{i+2} = x_{i+2} - t$, und es wird $B_{i,2}(t) = 1 - (x_{i+1} - t)/(x_{i+1} - x_i)$.

3) $x_{i+1} \le t < x_{i+2}$. Jetzt ist $f_i = f_{i+1} = 0$, $f_{i+2} = x_{i+2} - t$, also $B_{i,2}(t) = (x_{i+2} - t)/(x_{i+2} - x_{i+1})$.

4) $x_{i+2} \le t$. Es ist $f_i = f_{i+1} = f_{i+2} = 0$ und damit $B_{i,2}(t) = 0$.

Wir erhalten also die stückweise lineare stetige Funktion, die außerhalb $[x_i, x_{i+2}]$ verschwindet und die bei x_{i+1} den Wert 1 annimmt, also einen Spline der Ordnung 2.

Satz 5.5.1 *$B_{i,k}$ ist ein Spline der Ordnung k, der außerhalb $[x_i, x_{i+k}]$ verschwindet. Er heißt B-Spline (der Ordnung k zu $x_i, \ldots, x_{i+k}$).*

Beweis: f_j ist für $t \neq x_j$ ein Polynom vom Grade $\le k-1$ in t und für alle t $k-2$ mal stetig differenzierbar (für $k = 1$ ist die letzte Aussage leer). Als Linearkombination solcher Ausdrücke ist $[f_i, \ldots, f_{i+k}]$ in jedem Intervall $[x_j, x_{j+1}]$ $(j = i-1, \ldots, i+k)$ ein Polynom vom Grade $\le k-1$ und $k-2$ mal stetig differenzierbar, also ein Spline der Ordnung k zu $x_i, \ldots, x_{i+k}$. Für $t < x_i$ ist $f_j = (x_j - t)^{k-1}$ ein Polynom vom Grade $k-1$ in x_j für $j = i, \ldots, i+k$. Die dividierte Differenz $[f_i, \ldots, f_{i+k}]$ ist daher 0. Für $t > x_{i+k}$ ist $f_j = 0$ für $j = i, \ldots, i+k$. In beiden Fällen ist $B_{i,k}(t) = 0$.

□

Für das Weitere benötigen wir ein Hilfsmittel über dividierte Differenzen, das mit Splines eigentlich gar nichts zu tun hat.

Lemma 5.5.1 *(Leibniz'sche Formel) Sei $f_j = g_j h_j$, $j = i, \ldots, i+k$. Dann gilt*

$$[f_i, \ldots, f_{i+k}] = \sum_{r=i}^{i+k} [g_i, \ldots, g_r][h_r, \ldots, h_{i+k}] \, .$$

Beweis: Seien f, g, h die Interpolationspolynome vom Grade k zu den Stützstellen $x_i, \ldots, x_{i+k}$ und den Stützwerten f_j, g_j, h_j, $j = i, \ldots, i+k$. Das Newton'sche Interpolationspolynom für g, h lautet

$$\begin{aligned} g(x) &= [g_i] + [g_i, g_{i+1}](x - x_i) + \cdots [g_i, \ldots, g_{i+k}](x - x_i) \cdots (x - x_{i+k-1}) \\ &= \sum_{r=i}^{i+k} [g_i, \ldots, g_r](x - x_i) \cdots (x - x_{r-1}) \, , \end{aligned}$$

$$\begin{aligned} h(x) &= [h_{i+k}] + [h_{i+k}, h_{i+k-1}](x - x_{i+k}) + \cdots \\ &\qquad + [h_{i+k}, \ldots, h_i](x - x_{i+k}) \cdots (x - x_{i+1}) \\ &= \sum_{s=i}^{i+k} [h_s, \ldots, h_{i+k}](x - x_{i+k}) \cdots (x - x_{s+1}) \; . \end{aligned}$$

Multiplikation ergibt

$$(gh)(x) = \sum_{r,s=i}^{i+k} [g_i, \ldots, g_r][h_s, \ldots, h_{i+k}](x-x_i) \cdots (x-x_{r-1})(x-x_{s+1}) \cdots (x-x_{i+k}) \; . \tag{5.2}$$

Wir teilen die Summe auf in die Summe für $r > s$ und den Rest. Die erste Summe verschwindet für $x = x_i, \ldots, x_{i+k}$. Die restliche Summe ist ein Polynom vom Grade $\leq k$, welches an den Stellen $x_i, \ldots, x_{i+k}$ die Werte $f_i, \ldots, f_{i+k}$ annimmt und daher mit f übereinstimmen muß.

Der Höchstkoeffizient des restlichen Polynoms ergibt sich als Summe über $r = s$, also

$$\sum_{s=i}^{i+k} [g_i, \ldots, g_s][h_s, \ldots, h_{i+k}] \; ,$$

und dies muß mit dem Höchstkoeffizienten des Newton'schen Interpolationspolynoms, also $[f_i, \ldots, f_{i+k}]$ übereinstimmen. Dies beweist die Leibniz'sche Formel.

□

Satz 5.5.2 *Für $k \geq 2$ und $i = 0, \ldots, n-k$ 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 (5.5.1) ist

$$B_{i,k}(x) = (x_{i+k} - x_i)[f_i, \ldots, f_{i+k}] \; , \quad f_j = (x_j - x)_+^{k-1} \; .$$

Wir setzen $g_j = (x_j - x)_+^{k-2}$ und $h_j = x_j - x$. Dann ist $f_j = g_j h_j$, $j = i, \ldots, i+k$ und damit nach Leibniz

$$\begin{aligned} B_{i,k}(x) &= (x_{i+k} - x_i)[f_i, \ldots, f_{i+k}] \\ &= (x_{i+k} - x_i) \sum_{r=i}^{i+k} [g_i, \ldots, g_r][h_r, \ldots, h_{i+k}] \; . \end{aligned}$$

Da h_j ein Polynom vom Grade 1 in x_j ist, verschwindet $[h_r, \ldots, h_{i+k}]$ für $r < i+k-1$, und wir erhalten

$$\begin{aligned} B_{i,k}(x) &= (x_{i+k} - x_i)\{[g_i, \ldots, g_{i+k}][h_{i+k}] + [g_i, \ldots, g_{i+k-1}][h_{i+k-1}, h_{i+k}]\} \\ &= (x_{i+k} - x_i)\{[g_i, \ldots, g_{i+k}](x_{i+k} - x) + [g_i, \ldots, g_{i+k-1}]\} \\ &= (x_{i+k} - x_i)\left\{\frac{[g_{i+1}, \ldots, g_{i+k}] - [g_i, \ldots, g_{i+k-1}]}{x_{i+k} - x_i}(x_{i+k} - x) + [g_i, \ldots, g_{i+k-1}]\right\} \\ &= [g_{i+1}, \ldots, g_{i+k}](x_{i+k} - x) + [g_i, \ldots, g_{i+k-1}](x - x_i) \\ &= \frac{B_{i+1,k-1}}{x_{i+k} - x_{i+1}}(x_{i+k} - x) + \frac{B_{i,k-1}(x)}{x_{i+k-1} - x_i}(x - x_i) \; . \end{aligned}$$

□

Bemerkungen:

1) Zusammen mit

$$B_{i,1}(x) = \begin{cases} 1 & , \quad x_i \le x < x_{i+1} \; , \\ 0 & , \quad \text{sonst} \end{cases}$$

ermöglicht Satz 5.5.2 die rekursive Berechnung der $B_{i,k}$.

2) Diese Rekursion ist numerisch stabil, da nur Linearkombinationen positiver Zahlen gebildet werden.

Satz 5.5.3 *Für $k \ge 3$ gilt*

$$B'_{i,k}(x) = (k-1)\left\{\frac{B_{i,k-1}(x)}{x_{i+k-1} - x_i} - \frac{B_{i+1,k-1}(x)}{x_{i+k} - x_{i+1}}\right\} \; .$$

Beweis: Nach Definition ist

$$B_{i,k}(x) = (x_{i+k} - x_i)[f_i, \ldots, f_{i+k}] \; , \quad f_j = (x_j - x)_+^{k-1} \; .$$

Differentiation nach x ergibt

$$B'_{i,k}(x) = -(k-1)(x_{i+k} - x_i)[g_i, \ldots, g_{i+k}] \; ,$$

$$g_j = (x_j - x)_+^{k-2} \; .$$

Nach der rekursiven Definition der dividierten Differenzen ist

$$\begin{aligned} B'_{i,k}(x) &= -(k-1)\{[g_{i+1}, \ldots, g_{i+k}] - [g_i, \ldots, g_{i+k-1}]\} \\ &= (k-1)\left\{\frac{B_{i,k-1}(x)}{x_{i+k-1} - x_i} - \frac{B_{i+1,k-1}(x)}{x_{i+k} - x_{i+1}}\right\} \; . \end{aligned}$$

□

5.6 Interpolation mit Splines

Sei $x_0 < x_1 < \cdots < x_n$ und

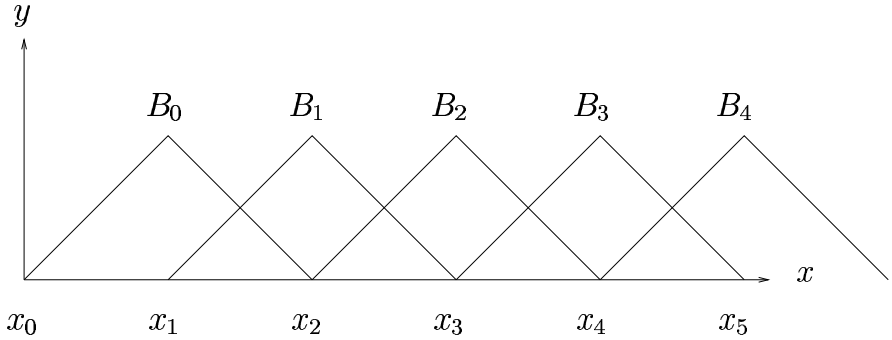
$$s(t) = \sum_{i=0}^{n-k} a_i B_{i,k}(t) \; .$$

Seien $n-k+1$ Stützstellen $t_0 < t_1 < \cdots < t_{n-k}$ und ebenso viele Stützwerte $y_0, \ldots, y_{n-k}$ gegeben. Wir wollen die a_i so bestimmen, daß $s(t_j) = y_j$, $j = 0, \ldots, n-k$ ist.

Beispiel: $k = 1$. Dann ist $s(t) = a_i$ für $x_i \leq t < x_{i+1}$. Das Interpolationsproblem ist genau dann eindeutig lösbar, wenn $x_i \leq t_i < x_{i+1}$, und zwar ist $a_i = y_i, \;\; i = 0, \ldots, n-k$.

Satz 5.6.1 *Das Interpolationsproblem ist eindeutig lösbar, wenn* $x_i < t_i < x_{i+k}, \;\; i = 0, \ldots, n-k$.

Beweis: Wir beginnen mit dem Fall $k = 2$ und schreiben B_i für $B_{i,k}$.



Wir haben zu zeigen, daß die $(n-1, n-1)$-Matrix

$$B = \begin{pmatrix} B_0(t_0) & B_1(t_0) & 0 & 0 \\ B_0(t_1) & B_1(t_1) & B_2(t_1) & 0 \\ 0 & B_1(t_2) & B_2(t_2) & B_3(t_2) \\ 0 & 0 & B_2(t_3) & B_3(t_3) \end{pmatrix} \qquad (n = 5)$$

invertierbar ist. Für $n = 2, 3$ ist dies elementar einzusehen. Den Fall $n > 3$ kann man auf $n = 2, 3$ zurückführen. Dazu zeigen wir, daß für $n > 3$ nicht alle Außerdiagonalelemente $\neq 0$ sein können. Ist nämlich etwa $B_0(t_1) \neq 0$, so ist $t_1 < x_2$ und damit $B_2(t_1) = 0$. Entsprechend argumentiert man in den anderen Fällen. Ist aber ein Außerdiagonalelement $= 0$, so zerfällt die Matrix in Teilmatrizen. Im Falle $B_2(t_1) = 0$ wären dies

$$\begin{pmatrix} B_0(t_0) & B_1(t_0) \\ B_0(t_1) & B_1(t_1) \end{pmatrix} \quad , \quad \begin{pmatrix} B_2(t_2) & B_3(t_2) \\ B_2(t_3) & B_3(t_3) \end{pmatrix} ,$$

im Falle $B_1(t_0) = 0$

$$B_0(t_0) \quad , \quad \begin{pmatrix} B_1(t_1) & B_2(t_1) & 0 \\ B_1(t_2) & B_2(t_2) & B_3(t_2) \\ 0 & B_2(t_3) & B_3(t_3) \end{pmatrix} .$$

Die letzte Matrix zerfällt wieder in eine $(1,1)$- und eine $(2,2)$-Matrix. Insgesamt genügt es also, die Invertierbarkeit der $(1,1)$- und der $(2,2)$-Matrizen nachzuweisen, also die Fälle $n = 2, 3$ zu betrachten. Damit ist der Satz für $k = 2$ bewiesen.

Sei nun die Behauptung richtig bis zur Ordnung $< k$ für ein $k \geq 3$. Der Fall $n = k$ ist trivial. Eine typische Stelle der Matrix für die Ordnung k sieht dann so aus:

$$\begin{matrix} & B_j(t_{j-1}) & B_{j+1}(t_{j-1}) \\ B_{j-1}(t_j) & B_j(t_j) & B_{j+1}(t_j) \end{matrix}$$

Wäre hier $B_{j+1}(t_j) = 0$, d.h. $t_j \leq x_{j+1}$, so wären erst recht alle Elemente rechts und oberhalb dieses Elementes 0. Entsprechend würde aus $B_{j-1}(t_j) = 0$, d.h. $t_j \leq x_{j+k-1}$, folgen, daß alle Elemente links und unterhalb dieses Elementes 0 wären. In beiden Fällen zerfiele die Matrix in kleinere, und wir könnten Induktion nach n treiben. Wir können also annehmen, daß $x_{j+1} < t_j < x_{j+k-1},\ j = 0, \ldots, n-k$.

Sei nun a ein Vektor mit den Komponenten $a_0, \ldots, a_{n-k}$ und $Ba = 0$. Der Spline

$$s = \sum_{j=0}^{n-k} a_j B_{j,k}$$

wäre eine C^{k-2}-Funktion mit den $n-k+3$ Nullstellen $t_0, \ldots, t_{n-k},\ x_0, x_n$. Nach dem Satz von Rolle hätte s' $n-k+2$ Nullstellen

$$\begin{aligned} \tau_0 \in (x_0, t_0) \ , \quad \tau_{n-k+1} \in (t_{n-k}, x_n) \ , \\ \tau_j \in (t_{j-1}, t_j) \ , \quad j = 1, \ldots, n-k \ . \end{aligned}$$

Wegen $x_{j+1} < t_j < x_{j+k-1}$ muß dann

$$\begin{aligned} \tau_0 \in (x_0, x_{k-1}) \ , \quad \tau_{n-k+1} \in (x_{n-k+1}, x_n) \ , \\ \tau_j \in (x_j, x_{j+k-1}) \ , \quad j = 1, \ldots, n-k \end{aligned}$$

gelten, wie man sich an Hand einer Figur klarmacht.

Mit $\tau_0, \ldots, \tau_{n-k+1}$ haben wir $n-k+2$ Nullstellen von s' gefunden, die in den Trägern von $B_{0,k-1}, \ldots, B_{n-k+1,k-1}$ liegen. s' löst also das Interpolationsproblem für Splines der Ordnung $k-1$ mit den Interpolationsstellen $\tau_0, \ldots, \tau_{n-k+1}$ und den Werten 0. Nach Induktionsannahmen ist dieses Problem eindeutig lösbar. Eine Lösung ist offenbar $s' = 0$. Also ist $s' = 0$ und damit $s = 0$. Damit ist auch $a = 0$.

□

Der Beweis des Satzes gibt auch eine Methode, den interpolierenden Spline zu berechnen. Man löse das Gleichungssystem

$$Ba = y\,, \quad a = \begin{pmatrix} a_0 \\ \vdots \\ a_{n-k} \end{pmatrix}\,, \quad y = \begin{pmatrix} y_0 \\ \vdots \\ y_{n-k} \end{pmatrix}$$

mit der Matrix B des Beweises. Diese Matrix ist eine Bandmatrix mit der Bandbreite $k+1$, der Rechenaufwand also entsprechend gering.

Bei speziellen Interpolationsproblemen kann man eine explizite Form des Gleichungssystems angeben. Unter dem natürlichen kubischen Spline verstehen wir einen Spline der Ordnung 4, der außerhalb $x_0, \ldots, x_n$ linear ist. Der interpolierende natürliche kubische Spline s erfüllt

$$\begin{aligned} s(x_j) &= y_j\,, \quad j = 0, \ldots, n \\ s''(x_0) &= s''(x_n) = 0\,. \end{aligned}$$

Wir werden zeigen, daß s eindeutig bestimmt und leicht zu berechnen ist. Wir machen in $[x_j, x_{j+1}]$ den Ansatz

$$\begin{aligned} s(x) &= y_j + \left(\frac{y_{j+1} - y_j}{h_j} - \frac{2M_j + M_{j+1}}{6}\, h_j \right)(x - x_j) \\ &\quad + \frac{M_j}{2}(x - x_j)^2 + \frac{M_{j+1} - M_j}{6h_j}(x - x_j)^3 \end{aligned}$$

mit $h_j = x_{j+1} - x_j$ und gewissen Zahlen M_j. Man bestätigt leicht

$$\begin{aligned} s(x_j \pm 0) &= y_j\,, \quad s''(x_j \pm 0) = M_j\,, \\ s'(x_j + 0) &= \frac{y_{j+1} - y_j}{h_j} - \frac{2M_j + M_{j+1}}{6} h_j\,, \\ s'(x_j - 0) &= \frac{y_j - y_{j-1}}{h_{j-1}} + \frac{2M_j + M_{j-1}}{6} h_{j-1}\,. \end{aligned}$$

s ist also genau dann Lösung unseres Interpolationsproblems, wenn

$$\frac{y_{j+1} - y_j}{h_j} - \frac{2M_j + M_{j+1}}{6} h_j = \frac{y_j - y_{j-1}}{h_{j-1}} + \frac{2M_j + M_{j-1}}{6} h_{j-1}\,, \quad j = 1, \ldots, n-1$$

$$M_0 = M_n = 0\,.$$

Dies ist ein lineares Gleichungssystem von $n-1$ Gleichungen für die $n-1$ Unbekannten $M_1, \ldots, M_{n-1}$. Es lautet in Matrix-Form

$$AM = b\,, \quad M = \begin{pmatrix} M_1 \\ \vdots \\ M_{n-1} \end{pmatrix}\,, \quad b = \begin{pmatrix} b_1 \\ \vdots \\ b_{n-1} \end{pmatrix}\,, \quad b_j = \frac{y_{j+1} - y_j}{h_j} - \frac{y_j - y_{j-1}}{h_{j-1}}$$

$$A = \frac{1}{3}\begin{pmatrix} h_0+h_1 & \frac{h_1}{2} & & \\ \frac{h_1}{2} & h_1+h_2 & \frac{h_2}{2} & \\ & & \cdots & \\ & & \frac{h_{n-1}}{2} & h_{n-2}+h_{n-1} \end{pmatrix}.$$

A ist offenbar invertierbar, und es gilt sogar (vergleiche den Beweis zu Satz 2.5.2)

$$k(A) = \|A\|_\infty \|A^{-1}\|_\infty \leq \frac{\max\limits_j (h_j + h_{j+1})}{\min\limits_j (h_j + h_{j+1})}.$$

A hat also gute Kondition, wenn die x_j nicht allzu unregelmäßig verteilt sind.

Kapitel 6

Numerische Integration und Differentiation

6.1 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:

$$I \simeq \sum_{k=0}^{n} A_k f(x_k)$$

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 die geschlossenen Newton-Cotes Formeln. Bei diesen sind die Stützstellen äquidistant und schließen Anfangs- und Endpunkte von $[a,b]$ ein:

$$\begin{aligned} x_k &= a + k \cdot h \quad , \quad h = \tfrac{b-a}{n} \quad k = 0, \ldots, n \\ I_n &= \sum_{k=0}^{n} A_k f(x_k) \end{aligned}$$

Die Gewichte A_k werden dadurch bestimmt, daß f an den Stützstellen durch ein Polynom p vom Grade n interpoliert wird und p an Stelle von f integriert wird.

In der Form von Lagrange lautet p

$$p(x) = \sum_{k=0}^{n} f(x_k)\omega_k(x) \quad , \quad \omega_k(x) = \prod_{\substack{\ell=0 \\ \ell \neq k}}^{n} \frac{x - x_\ell}{x_k - x_\ell} \, .$$

Wir setzen

$$I_n = \int_a^b p(x)dx = \sum_{k=0}^{n} \int_a^b \omega_k(x)dx f(x_k) \, ,$$

$$A_k = \int_a^b \omega_k(x)dx = \int_a^b \prod_{\substack{\ell=0 \\ \ell\neq k}}^{n} \frac{x - x_\ell}{x_k - x_\ell} dx \ .$$

Wir substituieren $x = ht + a$ und erhalten

$$A_k = h \int_0^n \prod_{\substack{\ell=0 \\ \ell\neq k}}^{n} \frac{t - \ell}{k - \ell} dt = h a_k \ .$$

Für $n = 1$ erhalten wir damit die "Trapezregel"

$$a_0 = \int_0^1 \frac{t-1}{-1} dt = \frac{1}{2} \ , \quad a_1 = \int_0^1 t dt = \frac{1}{2}$$

$$I_1 = \frac{h}{2}(f(a) + f(b)) = \frac{b-a}{2}(f(a) + f(b)) \ .$$

Das Integral wird also durch die Fläche eines Trapezes angenähert. Für $n = 2$ ergibt sich die trotz ihrer Einfachheit relativ genaue Simpson'sche Regel:

$$a_0 = \frac{1}{3} \quad , \quad a_1 = \frac{4}{3} \quad , \quad a_2 = \frac{1}{3}$$

$$I_2 = \frac{h}{3}\left(f(a) + 4f\left(\frac{b+a}{2}\right) + f(b)\right) \ .$$

Die folgende Tabelle enthält die Koeffizienten a_k für $n \leq 4$:

n	a_0	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 Genauigkeit konstruieren, indem man die oben angegebenen Regeln auf Teilintervalle anwendet.

Beispiel:

$$\begin{aligned} I &= \int_0^1 e^x dx = e - 1 &&= 1.7183 \\ I_1 &= \tfrac{1}{2}(1 + e) &&= 1.8591 \\ I_2 &= \tfrac{1}{6}(1 + 4e^{1/2} + e) &&= 1.7189 \\ I_3 &= \tfrac{1}{8}(1 + 3e^{1/3} + 3e^{2/3} + e) &&= 1.7185 \end{aligned}$$

Man sieht, daß der Übergang von I_1 nach I_2 einen großen Gewinn an Ge-

nauigkeit ergibt.
Der folgende Satz gibt eine Abschätzung für den Fehler $|I - I_n|$:

Satz 6.1.1 *i) Sei $f \in C^{n+1}[a,b]$. Dann gilt*

$$|I - I_n| \leq h^{n+2} c_n \max_{[a,b]} |f^{(n+1)}(x)| ,$$

$$c_n = \frac{1}{(n+1)!} \int_0^n \prod_{k=0}^n |t-k| dt .$$

ii) Sei n gerade und $f \in C^{n+2}[a,b]$. Dann gilt

$$|I - I_n| \leq h^{n+3} c_n^* \max_{[a,b]} |f^{(n+2)}(x)| , \quad c_n^* = \frac{n}{2} c_n .$$

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 5.2.1

$$(f-p)(x) = \frac{w(x)}{(n+1)!} f^{(n+1)}(\xi)$$

mit $w(x) = \prod_{k=0}^n (x - x_k)$. Also folgt

$$|I - I_n| \leq \frac{1}{(n+1)!} \int_a^b |w(x)| dx \max_{[a,b]} |f^{(n+1)}(x)| .$$

c_n ergibt sich aus der Berechnung von $\int_a^b |w(x)| dx$:

$$\begin{aligned} \int_a^b |w(x)| dx &= \int_a^b \prod_{k=0}^n |x - x_k| dx \\ &= h^{n+2} \int_0^n \prod_{k=0}^n |t-k| dt . \end{aligned}$$

ii) Für gerades n ist w ungerade bezüglich der Intervallmitte $c = \frac{a+b}{2}$, es gilt also

$$\int_a^b w(x)dx = 0 \, .$$

Damit hat man

$$\begin{aligned}
\int_a^b (f-p)(x)dx &= \frac{1}{(n+1)!} \int_a^b w(x) f^{(n+1)}(\xi)dx \\
&= \frac{1}{(n+1)!} \int_a^b w(x) \left\{ f^{(n+1)}(c) + (\xi - c) f^{(n+2)}(\eta) \right\} dx \\
&= \frac{1}{(n+1)!} \int_a^b w(x)(\xi - c) f^{(n+2)}(\eta)dx
\end{aligned}$$

Wegen $|\xi - c| \leq \frac{b-a}{2} = \frac{nh}{2}$ gilt

$$\begin{aligned}
\left| \int_a^b (f-p)(x)dx \right| &\leq \frac{1}{(n+1)!} \int_a^b |w(x)|dx \cdot \frac{nh}{2} \max_{[a,b]} |f^{(n+2)}(x)| \\
&= h^{n+2} c_n \frac{nh}{2} \max_{[a,b]} |f^{(n+2)}(x)| \\
&= h^{n+3} c_n^* \max_{[a,b]} |f^{(n+2)}(x)| \, .
\end{aligned}$$

□

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 Genauigkeit. 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:

$$\int_{x_k}^{x_{k+1}} f(x)dx \simeq \frac{h}{2}(f(x_k) + f(x_{k+1})) \, ,$$

$$\begin{aligned}
I = \int_{x_0}^{x_n} f(x)dx &= \sum_{k=0}^{n-1} \int_{x_k}^{x_{k+1}} f(x)dx \\
&\simeq \frac{h}{2} \sum_{k=0}^{n-1} (f(x_k) + f(x_{k+1}))
\end{aligned}$$

$$
\begin{aligned}
&= \frac{h}{2}(f_0 + 2f_1 + 2f_2 + \ldots + 2f_{n-1} + f_n) \\
&= T_h
\end{aligned}
$$

mit der Abkürzung $f_k := f(x_k)$.

Diese Formel heißt zusammengesetzte Trapezregel. Für den Fehler gilt:

$$
|I - T_h| = \sum_{k=0}^{n-1} \left| \int_{x_k}^{x_{k+1}} f(x)dx - \frac{h}{2}(f(x_k) + f(x_{k+1})) \right| .
$$

Nach Satz 6.1.1 gilt mit $c_1 = \frac{1}{2}\int_0^1 t(1-t)dt = \frac{1}{12}$:

$$
\begin{aligned}
|I - T_h| &\leq \frac{1}{12} \sum_{k=0}^{n-1} h^3 \max_{[x_k, x_{k+1}]} |f''(x)| \\
&\leq \frac{n}{12}\, h^3 \max_{[a,b]} |f''(x)| \\
&= \frac{b-a}{12} h^2 \max_{[a,b]} |f''(x)| .
\end{aligned}
$$

Für gerades n bilden wir nun analog die zusammengesetzte Simpson-Regel S_h, indem wir die Simpson-Regel auf jeweils zwei aufeinanderfolgende Teilintervalle anwenden und anschließend summieren:

$$
\begin{aligned}
I = \int_a^b f(x)dx &= \sum_{k=0}^{\frac{n}{2}-1} \int_{x_{2k}}^{x_{2k+2}} f(x)dx \\
&\simeq \frac{h}{3}(f_0 + 4f_1 + f_2 + f_2 + 4f_3 + f_4 + \ldots) \\
&= \frac{h}{3}(f_0 + 4f_1 + 2f_2 + 4f_3 + 2f_4 + \ldots + 2f_{n-2} + 4f_{n-1} + f_n) \\
&= S_h
\end{aligned}
$$

6.2 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 zusammengesetzte Trapezregel

$$T_1(h) = \frac{h}{2}(f_0 + 2f_1 + \ldots + 2f_{n-1} + f_n)$$

mit

$$f_i := f(x_i)\ , \quad x_i = a + ih\ , \quad i = 0, \ldots, n\ , \quad h = \frac{b-a}{n}\ .$$

Wir entwickeln nun $T_1(h)$ nach Potenzen von h.

Satz 6.2.1 *Sei $f \in C^{2m+2}[a,b]$. Dann gilt*

$$T_1(h) = I + c_1h^2 + c_2h^4 + \ldots + c_mh^{2m} + O(h^{2m+2})$$

mit

$$c_k = \frac{B_{2k}}{(2k)!}(f^{(2k-1)}(b) - f^{(2k-1)}(a))$$

und den Bernoulli'schen Zahlen B_k:

$$B_2 = \frac{1}{6} \quad , \quad B_4 = -\frac{1}{30} \quad , \quad B_6 = \frac{1}{42}, \ldots$$

Folgerung: $f \in C^{2m+2}(\mathbb{R}^1)$ 2π-periodisch. Dann gilt

$$\int_a^{a+2\pi} f(x)dx = T_1(h) + \ \mathrm{O}(h^{2m+2})\ .$$

Wenn sogar $f \in C^\infty(\mathbb{R}^1)$, so gilt

$$\int_a^{a+2\pi} f(x)dx - T_1(h) \to 0 \quad \text{schneller als jede Potenz von } \ h\ .$$

Periodische Funktionen lassen sich also über die volle Periode außerordentlich gut mit der Trapezregel integrieren. Diese vereinfacht sich in diesem Fall zu

$$T_1(h) = h\sum_{j=0}^{n-1} f_j\ .$$

Wir konstruieren nun Formeln hoher Genauigkeit für beliebige Funktionen.
Wir bilden dazu

$$\begin{aligned} T_1(h) &= I + c_1 h^2 + c_2 h^4 + \ldots + c_m h^{2m} + \mathrm{O}(h^{2m+2}) \,, \\ T_1(\frac{h}{2}) &= I + c_1 2^{-2} h^2 + c_2 2^{-4} h^4 + \ldots + c_m 2^{-2m} h^{2m} + \mathrm{O}(h^{2m+2}) \,. \end{aligned}$$

Damit ist

$$2^2 T_1(\frac{h}{2}) - T_1(h) = (2^2 - 1) I + c_2 (2^{-2} - 1) h^4 + \ldots + c_m (2^{2-2m} - 1) h^{2m} + \mathrm{O}(h^{2m+2}) \,,$$

$$I = \frac{1}{2^2 - 1} (2^2 T_1(\frac{h}{2}) - T_1(h)) - c_2 \frac{2^{-2} - 1}{2^2 - 1} h^4 - \ldots - c_m \frac{2^{2-2m} - 1}{2^2 - 1} h^{2m} + \mathrm{O}(h^{2m+2}) \,.$$

Mit

$$T_2(h) := \frac{1}{2^2 - 1} (2^2 T_1(\frac{h}{2}) - T_1(h))$$

erhalten wir also

$$I = T_2(h) + c_2' h^4 + \ldots + c_m' h^{2m} + \mathrm{O}(h^{2m+2}) \,.$$

Durch Wiederholen dieses Verfahrens erhalten wir Formeln höherer Ordnung:
Sei $T_k(h)$ eine Formel der Ordnung h^{2k}, d.h.

$$\begin{aligned} T_k(h) &= I + c_k h^{2k} + c_{k+1} h^{2k+2} + \ldots + c_m h^{2m} + \mathrm{O}(h^{2m+2}) \,, \\ T_k(\frac{h}{2}) &= I + c_k 2^{-2k} h^{2k} + \ldots + c_m 2^{-2m} h^2 2m \quad + \mathrm{O}(h^{2m+2}) \,. \end{aligned}$$

Damit wird

$$\begin{aligned} &2^{2k} T_k(\frac{h}{2}) - T_k(h) = \\ &= (2^{2k} - 1) I + c_{k+1} (2^{-2} - 1) h^{2k+2} + \ldots + c_m (2^{2k-2m} - 1) h^{2m} + \mathrm{O}(h^{2m+2}) \,. \end{aligned}$$

Mit

$$T_{k+1}(h) := \frac{1}{2^{2k} - 1} (2^{2k} T_k(\frac{h}{2}) - T_k(h))$$

gilt also

$$I = T_{k+1}(h) + \mathrm{O}(h^{2k+2}) \,.$$

Diese Konstruktion von Formeln höherer Ordnung läßt sich im sogenannten Romberg-Schema darstellen:

$$\begin{array}{llllllll} h & T_1(h) & & & & & & \\ & & \searrow & & & & & \\ \frac{h}{2} & T_1(\frac{h}{2}) & \longrightarrow & T_2(h) & & & & \\ & & \searrow & & \searrow & & & \\ \frac{h}{4} & T_1(\frac{h}{4}) & \longrightarrow & T_2(\frac{h}{2}) & \longrightarrow & T_3(h) & & \\ & & \searrow & & \searrow & & \searrow & \\ \frac{h}{8} & T_1(\frac{h}{8}) & \longrightarrow & T_2(\frac{h}{4}) & \longrightarrow & T_3(\frac{h}{2}) & \longrightarrow & T_4(h) \end{array}$$

Die Rekursion schreibt sich am besten in der Form

$$T_{k+1}(h) = T_k(\frac{h}{2}) + \frac{1}{2^{2k}-1}(T_k(\frac{h}{2}) - T_k(h)) \ .$$

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 im allgemeinen 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

$$\begin{aligned} T_1(\frac{h}{2}) &= \frac{h}{4}(f_0 + 2f_{1/2} + 2f_1 + \ldots + 2f_{n-1/2} + f_n) \\ &= \frac{h}{4}(f_0 + 2f_1 + 2f_2 + \ldots + 2f_{n-1} + f_n) \\ &\quad + \frac{h}{2}(f_{1/2} + f_{3/2} + \ldots + f_{n-1/2}) \\ &= \frac{1}{2}T_1(h) + \frac{h}{2}(f_{1/2} + f_{3/2} + \ldots + f_{n-1/2}) \ . \end{aligned}$$

Der erste Teil ist bereits bekannt. Nur der zweite Teil muß noch berechnet werden.

Beispiel: $I = \int_0^1 e^x dx = 1.718281828$

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 ε, 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

$$\begin{aligned} T_k(h) &= I + c_k h^{2k} + \mathrm{O}(h^{2k+2}) , \\ T_k(\frac{h}{2}) &= I + c_k 2^{-2k} h^{2k} + \mathrm{O}(h^{2k+2}) . \end{aligned}$$

Also folgt

$$\begin{aligned} T_k(\frac{h}{2}) - T_k(h) &= c_k(2^{-2k} - 1)h^{2k} + \mathrm{O}(h^{2k+2}) , \\ c_k h^{2k} &= \frac{1}{2^{-2k} - 1}(T_k(\frac{h}{2}) - T_k(h)) + \mathrm{O}(h^{2k+2}) \\ &= \varepsilon_k(\frac{h}{2}) + \mathrm{O}(h^{2k+2}) . \end{aligned}$$

Man prüft, ob $\varepsilon_k(\frac{h}{4}) \sim 2^{-2k}\varepsilon_k(\frac{h}{2})$. Falls die Abweichung unter 10% liegt, wird ε_k als Fehlerschätzung 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.

Wir müssen noch Satz 6.2.1 beweisen. Dazu einige Vorbereitungen. Seien B_k die Bernoulli-Polynome, d.h.

$$\begin{aligned} B_0 &= 1 , \\ B_k' &= B_{k-1} , \quad \int_0^1 B_k dx = 0 , \quad k = 1, 2, \ldots . \end{aligned}$$

Z.B. ist $B_1 = x - 1/2$, $B_2 = \frac{1}{2}x^2 - \frac{1}{2}x + 1/12$. Die $B_k = k! B_k(0)$ heißen Bernoulli-Zahlen.

Wir leiten einige Eigenschaften B_k her.

1) Für $k \geq 2$ ist $B_k(0) = B_k(1)$. Dies ist klar wegen

$$B_k(1) - B_k(0) = \int_0^1 B_k' dx = \int_0^1 B_{k-1} dx = 0 \quad \text{für} k \geq 2 .$$

2) Für $k \geq 1$ ist $B_{2k+1}(1) = B_{2k+1}(0) = 0$. Dazu setzen wir $P_k(x) = (-1)^k B_k(1 - x)$. Diese erfüllen die gleiche Rekursion wie die B_k. Also ist $P_k = B_k$, woraus die Behauptung mit Hilfe von 1) folgt.

Satz 6.2.2 *(Einfache Version der Euler'schen Summenformel) Sei $g \in C^{2m+2}[0,n]$, n ganz. Dann gilt*

$$\frac{1}{2}\,g(0)+g(1)+\cdots+g(n-1)+\frac{1}{2}g(n)-\int_0^n g\,dx$$

$$=-\int_0^n \overline{B}_{2m+2}g^{(2m+2)}dx+\sum_{k=0}^{m}\frac{B_{2k+2}}{(2k+2)!}\left(g^{(2k+1)}(n)-g^{(2k+1)}(0)\right)\,.$$

Dabei sind $\overline{B}_k$ die Funktionen in $\mathbb{R}^1$, welche aus B_k durch periodische Fortsetzung aus $[0,1]$ mit der Periode 1 entstehen.

Beweis: Die Eigenschaften der B_k führen zu folgenden Eigenschaften der $\overline{B}_k$: $\overline{B}_k$ ist stetig für $k \geq 2$, $\overline{B}_{2k+1}(0)=\overline{B}_{2k+1}(n)=0$, $\overline{B}_k=\overline{B}'_{k+1}$ für $k \geq 1$. Der Beweis beruht auf zwei verschiedenen Auswertungen von

$$\int_0^n \overline{B}_1 g'dx\,.$$

Einerseits haben wir

$$\begin{aligned}\int_0^n \overline{B}_1 g'dx &= \sum_{i=0}^{n-1}\int_i^{i+1}\overline{B}_1 g'dx=\sum_{i=0}^{n-1}\left\{\overline{B}_1 g\Big|_i^{i+1}-\int_i^{i+1}\overline{B}'_1 g dx\right\}\\ &= \sum_{i=0}^{n-1}\left\{\frac{g(i+1)+g(i)}{2}-\int_i^{i+1}g\,dx\right\}\\ &= \frac{1}{2}g(0)+g(1)+\cdots+g(n-1)+\frac{1}{2}g(n)-\int_0^n g\,dx\,.\end{aligned}$$

Andererseits gilt

$$\begin{aligned}\int_0^n \overline{B}_1 g'dx &= \int_0^n \overline{B}'_2 g'dx=-\int_0^n \overline{B}_2 g''dx+\overline{B}_2 g'\Big|_0^n\\ &= -\int_0^n \overline{B}'_3 g''dx+\overline{B}_2 g'\Big|_0^n\\ &= \int_0^n \overline{B}_3 g'''dx-\overline{B}_3 g''\Big|_0^n+\overline{B}_2 g'\Big|_0^n\,.\end{aligned}$$

So fortfahrend erhält man schließlich

$$\int_0^n \overline{B}_1 g'dx = -\int_0^n \overline{B}_{2m+2}g^{(2m+2)}dx$$

$$
\begin{aligned}
&+ \quad \left[+\overline{B}_{2m+2} g^{(2m+1)} - \overline{B}_{2m+1} g^{(2m)} + \cdots \; \overline{B}_2 g' \right]_0^n \\
&= \quad - \int_0^n \overline{B}_{2m+2} g^{(2m+2)} dx + \sum_{k=0}^{m} \frac{B_{2k+2}}{(2k+2)!} \left(g^{(2k+1)}(n) - g^{(2k+1)}(0) \right) .
\end{aligned}
$$

Aus dem Vergleich der beiden Darstellungen folgt die Behauptung.

□

Satz 6.1.1 folgt nun durch lineare Transformation des Intervalls $[a, b]$ auf $[0, n]$. Man setzt dazu einfach $g(x) = f(a + hx)$.

6.3 Integration nach Gauß

Wir suchen eine Integrationsformel für das Integral

$$If = \int_a^b w(x) f(x) dx$$

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 mit n Stützstellen (n ungerade) integrieren ein Polynom n-ten Grades exakt. Wir wollen nun fordern, daß

$$G_n f = If \quad \text{für} \quad f \in \mathcal{P}_{2n-1} .$$

Dies ergibt gerade $2n$ Bedingungen für die $2n$ Parameter. Der folgende Satz zeigt, daß diese Forderung maximal ist:

Satz 6.3.1 *Es gibt keine Formel G_n, die in $\mathcal{P}_{2n}$ exakt ist.*

Beweis: Wir nehmen an, G_n sei eine solche Formel, also $G_n f = If$ für alle $f \in \mathcal{P}_{2n}$. Dies wäre dann auch richtig für das Polynom

$$f(x) = \prod_{j=1}^{n} (x - x_j)^2 .$$

Offenbar ist aber $G_n f = 0$, $If \neq 0$.

□

Sei nun $H = C[a,b]$ und $(f,g) = \int_a^b wfgdx$ mit einer Funktion w, die "Gewichtsfunktion", welche in (a,b) stetig und positiv ist. Sei $u_0, u_1, \ldots$ l. u. Elemente von H und $v_0, v_1, \ldots$ die zugehörigen orthonormalisierten Elemente. 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 6.3.2 *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 $q = \prod_{i=1}^{m}(x - x_i) \in \mathcal{P}_m$. Wäre $m < k$, so wäre

$$(q, v_k) = \int_a^b wqv_k dx = 0\,.$$

Die Funktion qv_k hat konstantes Vorzeichen. Also folgt

$$q \cdot v_k \equiv 0 \quad \text{in} \quad (a,b)$$

und dies ist ein Widerspruch. Also ist $m \geq k$. Es kann nicht $m > k$ sein, weil v_k den Grad k hat. Also ist $m = k$.

□

Bemerkung: Die Nullstellen von v_k trennen die Nullstellen von v_{k+1}.

Satz 6.3.3 *Es gibt Konstanten α_k, β_k, γ_k, so daß $v_{k+1} = (\alpha_k x - \beta_k)v_k - \gamma_k v_{k-1}$.*

Beweis: Wir machen den Ansatz: $\tilde{v}_{k+1} = (x - \beta_k)v_k - \gamma_k v_{k-1}$. Die Konstanten β_k, γ_k werden so bestimmt, daß $\tilde{v}_{k+1}$ orthogonal zu $v_0, \ldots, v_k$ wird. Für $\ell \leq k-2$ gilt:

$$(\tilde{v}_{k+1}, v_\ell) = (xv_k, v_\ell) - \beta_k(v_k, v_\ell) - \gamma_k(v_{k-1}, v_\ell)\,.$$

Dies verschwindet, da $xv_\ell \in \mathcal{P}_{k-1}$.
Für $\ell = k-1$ und $\ell = k$ erhält man die Gleichungen

$$(xv_k, v_{k-1}) - \gamma_k = 0 \quad , \quad (xv_k, v_k) - \beta_k = 0\,.$$

Es gibt also β_k, γ_k, so daß $\tilde{v}_{k+1} \perp < v_0, \ldots, v_k >$. v_{k+1} entsteht aus $\tilde{v}_{k+1}$ durch Normieren.

□

Für die Werte der Koeffizienten α_k, β_k, γ_k gibt es Tabellen.

Beispiele: 1) $[a,b] = [-1,+1]$. Für $w = 1$ erhält man $v_k = c_k P_k$ mit den Legendre-Polynomen P_k:

$$\begin{aligned} P_0 &= 1 & P_3 &= \tfrac{1}{2}(5x^3 - 3x) \\ P_1 &= x & P_4 &= \tfrac{1}{8}(35x^4 - 30x^2 + 3) \\ P_2 &= \tfrac{1}{2}(3x^2 - 1) \end{aligned}$$

2) $[a,b] = [-1,+1]$, $w(x) = (1-x^2)^{-1/2}$.

$$T_k(x) = \cos(kt)\ , \quad \cos t = x\ .$$

Aus dem Additionstheorem folgt

$$T_{k+1}(x) + T_{k-1}(x) = 2xT_k(x)\ , \quad k = 1, 2, \ldots\ .$$

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.

Zur Konstruktion einer in $\mathcal{P}_{2n-1}$ exakten Formel G_n benutzen wir die orthonormalen Polynome v_n. Wir wissen: v_n hat in (a,b) genau n einfache Nullstellen.

Satz 6.3.4 *Es gibt Formeln G_n, welche auf $\mathcal{P}_{2n-1}$, exakt sind. Die x_j sind die Nullstellen von v_n und es gilt*

$$A_j = \int_a^b w(x) \prod_{\substack{i=1 \\ i\neq j}} \left(\frac{x - x_i}{x_j - x_i}\right)^2 dx\ .$$

Beweis: Sei G_n die Newton-Cotes Formel zu den Nullstellen $x_1, \ldots, x_n$ von v_n, also

$$G_n f = \sum_{j=1}^n \int_a^b w \prod_{\substack{i=1 \\ i\neq j}} \frac{x - x_i}{x_j - x_i} dx f(x_j)\ .$$

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

$$f = qv_n + r \quad \text{mit} \quad q, r \in \mathcal{P}_{n-1} \qquad \text{(Division mit Rest)} .$$

Dann ist

$$\int_a^b wf dx = \int_a^b wqv_n dx + \int_a^b wr dx .$$

Das erste Integral verschwindet wegen der Orthogonalitätseigenschaften der v_n. Das zweite ist $G_n r$, weil G_n ja auf $\mathcal{P}_{n-1}$ exakt ist. Also folgt

$$\int_a^b wf dx = G_n r = G_n(r + qv_n) = G_n f ,$$

weil v_n an den Stützstellen von G_n verschwindet.

Also ist G_n exakt in $\mathcal{P}_{2n-1}$ und wir müssen nur noch die Formel für die Gewichte bestätigen. Mit

$$w_j = \prod_{\substack{i=1 \\ i \neq j}} \frac{x - x_i}{x_j - x_i}$$

ist $w_j^2 \in \mathcal{P}_{2n-2}$, also

$$\int_a^b w w_j^2 dx = G_n(w_j^2) = \sum_{k=1}^{n} A_k w_j^2(x_k) = A_j .$$

□

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

$$G_3 = 2.350337$$

6.4 Numerische Differentiation

Sei $f \in C^{p+1}(\mathbb{R}^1)$. Wir interessieren uns für eine Näherung für $f^{(k)}(0)$, $k \leq p$, die mit Hilfe der Werte $f(x_j)$ an den Stützstellen $x_j = jh$ berechnet wird. Der Satz von Taylor liefert

$$f(ih) = \sum_{\ell=0}^{p} f^{(\ell)}(0) \frac{(ih)^\ell}{\ell!} + \mathrm{O}(h^{p+1}) \; .$$

Wir bilden die Linearkombination

$$\sum_{i=-q}^{q} \alpha_i f(ih) = \sum_{\ell=0}^{p} f^{(\ell)}(0) \frac{1}{\ell!} h^\ell \sum_{i=-q}^{q} i^\ell \alpha_i + \mathrm{O}(h^{p+1})$$

und wählen die α_i so, daß für $\ell = 0, \ldots, p$

$$\sum_{i=-q}^{q} i^\ell \alpha_i = \begin{cases} 1 & , \quad \ell = k \; , \\ 0 & , \quad \text{sonst} \end{cases} \quad .$$

Wir erhalten

$$\frac{1}{k!} h^k f^{(k)}(0) = \sum_{i=-q}^{q} \alpha_i f(ih) + \mathrm{O}(h^{p+1}) \; .$$

Für $2q+1 = p+1$ führt das Gleichungssystem für die α_i auf eine Vandermonde-Matrix, ist also eindeutig lösbar. Für $2q+1 > p+1$ setzt man einige $\alpha_i = 0$, bis man wieder eine Vandermonde-Matrix erhält.

Wir haben also mit

$$D_h^{(k)} f = \frac{k!}{h^k} \sum_{i=-q}^{q} \alpha_i f(ih) = f^{(k)}(0) + \mathrm{O}(h^{p+1-k})$$

eine Differentiationsformel der Ordnung h^{p+1-k}.

Beispiele: $k = 1$:

$$p = 1 \; , \; q = 1 \; , \; \alpha_{-1} = 0 \; , \; \alpha_0 = -1 \; , \; \alpha_1 = +1 \; ,$$

$$D_h^{(1)} f = \frac{f(h) - f(0)}{h} \; , \quad f'(0) = D_h^{(1)} f + \mathrm{O}(h) \quad \text{für} \quad f \in C^2 \; .$$

$$p = 2 \; , \; q = 1 \; , \; \alpha_{-1} = -\frac{1}{2} \; , \; \alpha_0 = 0 \; , \; \alpha_1 = \frac{1}{2} \; ,$$

$$D_h^{(1)} f = \frac{f(h) - f(-h)}{2h} , \quad f'(0) = D_h^{(1)} f + \mathrm{O}(h^2) \quad \text{für} \quad f \in C^3 .$$

$k = 2$:

$$p = 3 , \; q = 1 , \; \alpha_{-1} = \frac{1}{2} , \; \alpha_0 = -1 , \; \alpha_1 = \frac{1}{2} ,$$

$$D_h^{(2)} f = \frac{f(h) - 2f(0) + f(-h)}{h^2} , \; f''(0) = D_h^{(2)} f + \mathrm{O}(h^2) \quad \text{falls} \; f \in C^4 .$$

6.5 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 ε (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

$$|If - \tilde{I}f| \leq \int_a^b |(f - \tilde{f})(x)| dx \leq \varepsilon I|f| . \tag{5.1}$$

Falls $I|f|$, $|If|$ die gleiche Größenordnung haben (z.B. falls $f \geq 0$), so haben $|If - \tilde{I}f|$, $\varepsilon |If|$ die gleiche Größenordnung, also $\tilde{I}f$ einen relativen Fehler der Größenordnung ε.

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 ε.

Wir untersuchen die Fehlerfortpflanzung bei der Berechnung von f durch eine Integrationsformel der Ordnung p:

$$I_h f = \sum_{j=1}^n A_j f(x_j) , \quad |I_h f - If| \leq ch^p .$$

Es gilt

$$\begin{aligned} |I_h \tilde{f} - If| &\leq |I_h(\tilde{f} - f| + |(I_h - I)f| \\ &\leq \varepsilon \sum_{j=1}^n |A_j| |f(x_j)| + ch^p . \end{aligned}$$

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

$$|I_h\tilde{f} - If| \leq \varepsilon I|f| + ch^p \; . \tag{5.2}$$

Vergleich mit (5.1) zeigt, daß hier $\varepsilon I|f|$ der durch die Problemstellung verursachte, ch^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

$$ch^p \leq \varepsilon I|f| \, , \tag{5.3}$$

so ist der Algorithmusfehler akzeptabel. Dies ist nicht der Fall bei negativen Gewichten, denn dann kann

$$\varepsilon \sum_{j=1}^{n} |A_j||f(x_j)| \gg \varepsilon I|f|$$

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 $p+1-k$

$$D_h^{(k)} f = \frac{k!}{h^k} \sum_{i=-q}^{q} \alpha_i f(ih) \, , \quad |D_h^{(k)} f - f^{(k)}(0)| \leq ch^{p+1-k}$$

an, so wird

$$\begin{aligned} |D_h^{(k)}\tilde{f} - f^{(k)}(0)| &\leq |D_h^{(k)}(\tilde{f} - f)| + |D_h^{(k)} f - f^{(k)}(0)| \\ &\leq \varepsilon h^{-k} a + ch^{p+1-k} \, , \\ a &= k! \sum_{i=-q}^{q} |\alpha_i||f(ih)| \, . \end{aligned} \tag{5.4}$$

Hier kann man nun h nicht gegen Null gehen lassen, weil dabei der Fehler über alle Grenzen wächst. Es gibt hier ein optimales $h = h_0$, bei welchem der Fehler minimal wird. Größenordnungsmäßig kann man h_0 aus der Bedingung ("balancing terms")

$$\varepsilon h^{-k} a = ch^{p+1-k}$$

bestimmen zu $h_0 = \mathrm{O}(\varepsilon^{1/(1+p)})$. Für $h = h_0$ ist dann

$$D_h^{(k)}\tilde{f} - f^{(k)}(0) = \mathrm{O}(\varepsilon^{1-k/(p+1)}) \, .$$

Ein Fehler ε in f führt also - bei einer Formel der Ordnung p und optimaler Wahl von h - zu einem Fehler ε^α, $\alpha = 1 - \frac{k}{p+1} < 1$ in $f^{(k)}$. Dies bedeutet:

1) Numerische Differentiation führt zu einem Genauigkeitsverlust.

2) Dieser Verlust ist klein (d.h. $\alpha \sim 1$) falls, $p+1 \gg k$. Insbesondere hängt er von der Ordnung der verwendeten Differentiationsformel ab.

Beispiel: Berechnung von $f'(0)$ $(k=1)$

Formel	p	h_0	ε^α
$\frac{1}{h}(f(h)-f(0))$	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	$\varepsilon^{1/7}$	$\varepsilon^{7/8}$

6.6 Harmonische Analyse

Sei f 2π-periodisch. Solche Funktionen kann man in eine Fourier-Reihe entwickeln:

Satz 6.6.1 *Sei f stetig. Dann gilt*

$$\begin{aligned} f(x) &= \frac{a_0}{2} + \sum_{k=0}^{\infty}(a_k \cos kx + b_k \sin kx)\,, \\ a_k &= \frac{1}{\pi}\int_0^{2\pi} f(x)\cos kx dx \\ b_k &= \frac{1}{\pi}\int_0^{2\pi} f(x)\sin kx dx \end{aligned}$$

Beweis: Siehe etwa **Heuser** :*Lehrbuch der Analysis II.*

□

Bemerkungen:

1) Gilt sogar $f \in C^m$, so ist a_k, $b_k = \mathrm{O}(k^{-m})$.

2) Ist f stetig mit Ausnahme von x_0 und existieren dort die linken und rechten Grenzwerte $f(x_0 \pm 0)$, so konvergiert die Reihe in x_0 gegen $\frac{1}{2}\,(f(x_0+0)+f(x_0-0))$.

3) Die Berechnung der a_k, b_k aus f heißt Fourier-Analyse, die Berechnung von f aus a_k, b_k heißt Fourier-Synthese. a_k, b_k heißen Fourier-Koeffizienten von f, die Reihe Fourier-Reihe von f.

Beispiel: $f(x) = \begin{cases} 1 & , \ 0 \le x < \pi \\ -1 & , \ \pi \le x < 2\pi \end{cases}$. Offenbar ist $a_k = 0$ für alle k und $b_k = 0$ für k gerade, $b_k = 4/\pi k$ für k ungerade. Also lautet die Fourier-Reihe

$$f(x) = \frac{4}{\pi}\left(\sin x + \frac{1}{3}\sin 3x + \frac{1}{5}\sin 5x + \cdots\right) .$$

Zur numerischen Fourier-Analyse nehmen wir an, daß f an den Stützstellen $x_j = 2\pi j/n$, $j = 0, \ldots, n-1$ gegeben ist. Dann hat man die Approximationen

$$a_k \sim a_{k,n} = \frac{2}{n}\sum_{j=0}^{n-1} f(x_j)\cos kx_j \ , \quad b_k \sim b_{k,n} = \frac{2}{n}\sum_{j=0}^{n-1} f(x_j)\sin kx_j \ .$$

Im Abschnitt über numerische Integration haben wir gesehen, daß diese Näherungen (für 2π-periodisches f) kaum noch verbesserbar sind. Trotzdem verhalten sich die genäherten Fourier-Koeffizienten $a_{k,n}$, $b_{k,n}$ ganz anders als die exakten: $a_{k,n}$, $b_{k,n}$ haben in k die Periode n, während a_k, b_k für $k \to \infty$ gegen 0 streben. Man kann daher nur für $k \ll n$ erwarten, daß $a_{k,n}$, $b_{k,n}$ gute Näherungen für a_k, b_k darstellen.

Wir gehen zur komplexen Schreibweise und das Intervall $[-\pi, \pi]$ über und schreiben dann für Satz 1

$$\begin{aligned} f(x) &= \sum_{k=-\infty}^{+\infty} c_k \, e^{ikx} \ , \\ c_k &= \frac{1}{2\pi}\int_{-\pi}^{+\pi} f(x)\, e^{-ikx} dx \ . \end{aligned}$$

Die Approximation $c_{n,k}$ für c_k ist mit $x_j = \pi j/n$, $j = -n, \ldots, n-1$

$$c_{n,k} = \frac{1}{n}\sum_{j=-n}^{n-1} f_j \, e^{-ikx_j} = \frac{1}{n}\sum_{j=-n}^{n-1} f_j \, e^{-\pi ikj/n}$$

mit $f_j = f(x_j)$. Dies ist genau eine Fourier-Transformation der Länge $2n$. Die Fourier-Analyse kann also näherungsweise durch die schnelle Fourier-Transformation durchgeführt werden. Zur Fourier-Synthese verwenden wir aus den oben angedeuteten Gründen nur $c_{n,k}$ für $k = -n, \ldots, n-1$, d.h.

$$\begin{aligned} f(x_j) &\sim \sum_{k=-n}^{n-1} c_{n,k} \, e^{ikx_j} \\ &= \sum_{k=-n}^{n-1} c_{n,k} \, e^{\pi ikj/n} \ , \end{aligned}$$

weil $c_{n,k}$ in k die Periode n hat. Dies ist genau eine inverse diskrete Fourier-Transformation der Länge $2n$. Auch die Fourier-Synthese kann daher mit der schnellen Fourier-Transformation durchgeführt werden.

Um das asymptotische Verhalten der $c_{n,k}$ an das der c_k anzupassen, multipliziert man die $c_{n,k}$ noch mit "Abminderungsfaktoren" oder "Filterfaktoren" $d_{n,k}$ mit der Eigenschaft

$$\begin{aligned} d_{n,k} &\sim 1 \quad , \quad |k| \ll n \, , \\ d_{n,k} &\sim 0 \quad , \quad |k| \sim n \, . \end{aligned}$$

Eine häufige Wahl ist

$$d_{n,k} = \left(\frac{\sin k\pi/n}{k\pi/n} \right)^p$$

mit einem geeigneten p, etwa $p = 1, 2$ oder 3.

Kapitel 7

Gewöhnliche Differentialgleichungen

7.1 Anfangswertaufgaben gewöhnlicher Differentialgleichungen

Es soll eine ganz kurze Einführung in die Theorie der Anfangswertaufgabe 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 Anfangswertaufgabe besteht darin, eine solche Lösung zu finden, welche auch noch durch den Punkt (x_0, y_0) geht, d.h.

$$y(x_0) = y_0 \, .$$

Beispiele:

1) Bevölkerungswachstum.

Sei $p(t)$ die Größe der Bevölkerung zur Zeit t, $g(t,p)$ ihre Geburtsrate, $s(t,p)$ ihre Sterberate. p_0 sei die Größe der Bevölkerung zur Zeit t_0.

Die Funktion p löst offenbar die Anfangswertaufgabe

$$\frac{\dot{p}}{p} = g(t,p) - s(t,p) \quad , \quad p(t_0) = p_0 \; .$$

2) Lineare Differentialgleichung.

$$y' = p(x)y + q(x) \quad , \quad p, q \in C(a,b) \; .$$

Die Lösung läßt sich explizit angeben. Man betrachtet zunächst die homogene Differentialgleichung ($q = 0$)

$$y' = p(x)y \; .$$

Unter der Annahme $y(x) \neq 0$ erhält man der Reihe nach

$$\frac{y'}{y} = p(x) \quad , \quad \frac{d}{dx}\ell n \; y = p(x) \quad , \quad \ell n y = \int_{x_0}^{x} p(t)dt + \ell n y_0$$

$$y = y_0 e^{\int_{x_0}^{x} p(t)dt} \; .$$

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

$$y_H = e^{\int_{x_o}^{x} p(x)dt}$$

ergibt sich

$$c'(x) = q(x)e^{-\int_{x_0}^{x} p(x)dt} \quad , \quad c(x) = y_0 + \int_{x_0}^{x} q(t)e^{-\int_{x_0}^{x} p(s)ds} dt$$

mit einer Konstanten y_0, und weiter

$$y = \left(y_0 + \int_{x_0}^{x} q(t) e^{\int_{x_0}^{t} p(s)ds} dt \right) e^{\int_{x_0}^{x} p(t)dt}$$

$$= y_0 e^{\int_{x_0}^{x} p(t)dt} + \int_{x_0}^{x} q(t) e^{\int_{t}^{x} p(s)ds} dt \, .$$

Der erste Term ist eine Lösung der homogenen Gleichung mit Anfangswert y_0, 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 $C^\infty(\mathbf{R}^2)$ 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 Anfangswertaufgabe braucht also nicht eindeutig zu sein.

Zur Formulierung eines Existenz- und Eindeutigkeitssatzes benötigen wir folgende

Definition 7.1.1 *$f \in C(D)$ erfüllt in D eine Lipschitz-Bedingung, wenn es eine Konstante L gibt mit*

$$|f(x, y) - f(x, z)| \leq L|y - z| \, ,$$

wenn nur (x, y), $(x, z) \in D$. f erfüllt in D eine lokale Lipschitz-Bedingung, wenn es zu jedem Punkt in D eine Umgebung gibt, in der f eine Lipschitz-Bedingung erfüllt.

Satz 7.1.1 *Ist $f \in C^1(D)$, so erfüllt f in D eine lokale Lipschitz-Bedingung.*

Beweis: Mittelwertsatz der Differentialgleichung.

Satz 7.1.2 *f erfülle in D eine lokale Lipschitz-Bedingung. Dann gibt es zu jedem $(x_0, y_0) \in D$ eine Umgebung $(x_0 - \varepsilon, x_0 + \varepsilon)$ von x_0 und eine dort definierte Lösung y der Anfangswertaufgabe $y(x_0) = y_0$.*

Satz 7.1.3 *f erfülle in D eine lokale Lipschitz-Bedingung und $(x_0, y_0) \in D$. Dann gibt es eine Lösung y der Anfangswertaufgabe*

$$y' = f(x, y) \quad , \quad y(x_0) = y_0$$

mit folgender Eigenschaft: Jede weitere Lösung der Anfangswertaufgabe ist eine Restriktion von y.

Dabei heißt eine Funktion $\overline{y} : \overline{I} \to \mathbb{R}$ eine Restriktion von $y : I \to R$, wenn $\overline{I} \subseteq I$ und y, $\overline{y}$ auf $\overline{I}$ 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.

7.2 Einschrittverfahren für Anfangswertaufgaben

Die Anfangswertaufgabe

$$y' = f(x,y) \quad , \quad y(x_0) = y_0 \tag{2.1}$$

besitze in einer abgeschlossenen beschränkten Umgebung U von x_0 eine eindeutig bestimmte Lösung y. Wir wollen y auf dem Gitter $I_h : x_k = x_0 + kh$, $k = 0,1,\ldots$ berechnen. Dazu ersetzen wir (2.1) durch die Differenzengleichung

$$\frac{1}{h}(y_{k+1} - y_k) = f_h(x_k, y_k) \quad , \quad k = 0,1,\ldots \tag{2.2}$$

mit dem Startwert y_0 aus (2.1). 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

$$hf_h(x_k, y_k) \sim \int_{x_k}^{x_{k+1}} f(x, y(x))dx \tag{2.3}$$

sein. Die einfachste Weise, (2.3) zu erfüllen, ist

$$f_h(x_k, y_k) = f(x_k, y_k) \; .$$

Das so entstehende Einschrittverfahren

$$y_{k+1} = y_k + hf(x_k, y_k)$$

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 (2.1) in (2.2) 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 7.2.1 *Das Einschrittverfahren (2.2) heißt konsistent, falls*

$$\lim_{h\to 0} \max_{x_k\in U} |T_h(x_k)| = 0 \, .$$

Es heißt konsistent von der Ordnung p, falls für $h \to 0$

$$\max_{x_k\in U} |T_h(x_k)| = O(h^p) \, .$$

Beispiele:

1) Euler-Verfahren.
   Für $y' = f(x,y)$ ist

$$\begin{aligned} T_h(x_{k+1}) &= \frac{1}{h}(y(x_{k+1}) - y(x_k)) - f(x_k, y(x_k)) \\ &= y'(x_k) + \frac{h}{2}y''(\tilde{x}_k) - f(x_k, y(x_k)) \, , \quad \tilde{x}_k \in (x_k, x_{k+1}) \\ &= \frac{h}{2}y''(\tilde{x}_k) \, . \end{aligned}$$

   Also: Ist $y \in C^2(U)$, so ist das Euler-Verfahren konsistent von der Ordnung 1.

2) Verbessertes Euler-Verfahren.
   Nach der Trapezregel ist

$$\int_{x_k}^{x_{k+1}} f(x, y(x))dx = \frac{h}{2}\{f(x_k, y(x_k)) + f(x_{k+1}, y(x_{k+1}))\} + \mathrm{O}(h^3) \quad (2.4)$$

   für $f \in C^2$ (also $y \in C^3$). Weiter ist für $y' = f(x,y)$

$$\begin{aligned} y(x_{k+1}) &= y(x_k) + hy'(x_k) + \mathrm{O}(h^2) \\ &= y(x_k) + hf(x_k, y(x_k)) + \mathrm{O}(h^2) \, . \end{aligned}$$

   Setzt man dies in (2.4) ein, so entsteht

$$\frac{1}{h}\int_{x_k}^{x_{k+1}} f(x, y(x))dx = \frac{1}{2}\left\{f(x_k, y(x_k)) + f(x_{k+1}, y(x_k) + hf(x_k, y(x_k)))\right\} + \mathrm{O}(h^2) \, .$$

   Mit der Schrittfunktion

$$f_h(x,y) = \frac{1}{2}\{f(x,y) + f(x+h, y+hf(x,y))\}$$

hat man also

$$\begin{aligned} T_h(x_{k+1}) &= \frac{1}{h}(y(x_{k+1}) - y(x_k)) - f_h(x_k, y(x_k)) \\ &= \frac{1}{h}\int_{x_k}^{x_{k+1}} y'(x)dx - \frac{1}{h}\int_{x_k}^{x_{k+1}} f(x, y(x))dx + \mathrm{O}(h^2) \\ &= \mathrm{O}(h^2)\ . \end{aligned}$$

Also hat man Konsistenz von der Ordnung 2.

Beispiel: $y' = 1 + y^2$, $y'(0) = 0$, $h = 0.1$. 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

$$\begin{aligned} y_{k+1} &= y_k + h(\gamma_1 f_1 + \ldots + \gamma_m f_m)(x_k, y_k)\ , \\ f_1(x,y) &= f(x,y) \\ f_2(x,y) &= f(x + \alpha_2 h, y + h\beta_{2,1} f_1(x,y)) \\ &\vdots \\ f_m(x,y) &= f(x + \alpha_m h, y + h[\beta_{m,1} f_1 + \ldots + \beta_{m,m-1} f_{m-1}](x,y))\ . \end{aligned}$$

Man stellt alle Koeffizienten in dem Schema

$$\begin{array}{c|ccc} 0 & & & \\ \alpha_2 & \beta_{2,1} & & \\ \vdots & & & \\ \alpha_m & \beta_{m,1} & \ldots & \beta_{m,m-1} \\ \hline & \gamma_1 & \cdots & \gamma_m \end{array}$$

zusammen. Man nimmt übrigens immer $\alpha_k = \sum_{\ell=1}^{k-1} \beta_{k,\ell}$ und $1 = \gamma_1 + \cdots + \gamma_m$ an.

Beispiele:

$m = 1$

0	
	1

$y_{k+1} = y_k + hf(x_k, y_k)$
Euler, $p = 1$

$m = 2$

0		
1	1	
	$\frac{1}{2}$	$\frac{1}{2}$

$$y_{k+1} = y_k + \tfrac{h}{2}(f(x_k, y_k) + f(x_k + h, y_k + hf(x_k, y_k)))$$
Verbessertes Euler, $p = 2$

$m = 3$

0			
$\frac{1}{2}$	$\frac{1}{2}$		
1	-1	2	
	$\frac{1}{6}$	$\frac{4}{6}$	$\frac{1}{6}$

$$\begin{aligned} y_{k+1} &= y_k + \tfrac{h}{6}(f_1 + 4f_2 + f_3) \\ f_1 &= f(x_k, y_k) \\ f_2 &= f(x_k + \tfrac{h}{2}, y_k + \tfrac{h}{2} f_1) \\ f_3 &= f(x_k + h\,, y_k - hf_1 + 2hf_2) \end{aligned}$$
$p = 3$

$m = 4$

0				
$\frac{1}{2}$	$\frac{1}{2}$			
$\frac{1}{2}$	0	$\frac{1}{2}$		
1	0	0	1	
	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$

$$\begin{aligned} y_{k+1} &= y_k + \tfrac{h}{6}(f_1 + 2f_2 + 2f_3 + f_4) \\ f_1 &= f(x_k, y_k) \\ f_2 &= f(x_k + \tfrac{h}{2}, y_k + \tfrac{h}{2} f_1) \\ f_3 &= f(x_k + \tfrac{h}{2}, y_k + \tfrac{h}{2} f_2) \\ f_4 &= f(x_k + h, y_k + hf_3) \end{aligned}$$
(Standard) Runge-Kutta, $p = 4$.

7.3 Konvergenz von Einschrittverfahren

Vom lokalen Diskretisierungsfehler zu unterscheiden ist der globale Diskretisierungsfehler

$$D_h(x_k) = y_k - y(x_k) \; .$$

Definition 7.3.1 *Das Einschrittverfahren heißt konvergent, falls*

$$\lim_{h \to 0} \max_{x_k \in U} |D_h(x_k)| = 0 \; .$$

Es heißt konvergent von der Ordnung p, falls

$$\max_{x_k \in U} |D_h(x_k)| = O(h^p) \; .$$

Lemma 7.3.1 *1) Seien q, d_k, a_k nichtnegative Zahlen mit*

$$d_{k+1} \leq q d_k + a_k \qquad , \qquad k = 0, 1, \ldots \; .$$

Dann gilt

$$d_k \leq q^k d_0 + \sum_{j=0}^{k-1} q^{k-j-1} a_j \; .$$

2) Für $q \neq 1$ ist

$$\sum_{j=0}^{k-1} q^j = \frac{q^k - 1}{q - 1} \; .$$

3) Für $x \in \mathbf{R}$ ist $1 + x \leq e^x$.

Satz 7.3.1 *f_h erfülle in einer Umgebung D der Kurve $(x, y(x))_{x \in U}$ eine Lipschitz-Bedingung, d.h. es gebe eine von h, x unabhängige Zahl $L > 0$ mit*

$$|f_h(x, z_1) - f_h(x, z_2)| \leq L |z_1 - z_2| \; ,$$

falls (x, z_1), $(x, z_2) \in D$.

Dann gilt: Solange $(x_k, y_k) \in D$ ist, besteht die Abschätzung

$$|y(x_k) - y_k| \leq \frac{1}{L} \left(e^{L|x_k - x_0|} - 1 \right) \max_{j=1}^{k} |T_h(x_j)| \; .$$

Beweis: Aufgrund der Definition des lokalen Diskretisierungsfehlers haben wir

$$y(x_{k+1}) - y(x_k) = hf_h(x_k, y(x_k)) + hT_h(x_{k+1}) \; .$$

Das Verfahren lautet

$$y_{k+1} - y_k = hf_h(x_k, y_k) \; .$$

Subtraktion der beiden Beziehungen ergibt mit $d_k = y(x_k) - y_k$

$$d_{k+1} - d_k = h(f_h(x_k, y(x_k)) - f_h(x_k, y_k)) + hT_h(x_{k+1}) \; .$$

Solange $(x_k, y_k) \in D$ ist, folgt aus der Lipschitz-Bedingung

$$|d_{k+1} - d_k| \leq hL|d_k| + h|T_h(x_{k+1})|$$

oder

$$|d_{k+1}| \leq (1 + hL)|d_k| + h|T_h(x_{k+1})| \; .$$

Das Lemma ergibt nun unmittelbar

$$\begin{aligned} |d_k| &\leq h \sum_{j=0}^{k-1} (1 + hL)^{k-j-1} |T_h(x_{j+1})| \\ &\leq h \sum_{j=0}^{k-1} (1 + hL)^{k-j-1} \max_{j=1}^{k} |T_h(x_j)| \; . \end{aligned}$$

Die weiteren Aussagen des Lemmas führen zu der behaupteten Abschätzung.

□

Satz 7.3.2 *Das Einschrittverfahren sei konsistent (von der Ordnung p) und f_h erfülle die Voraussetzung von Satz 9.3.1. Dann ist das Verfahren konvergent (von der Ordnung p).*

Beweis: D enthält einen Streifen $\{(x, y) : |y - y(x)| \leq d, \; x \in U\}$ der Breite $2d > 0$. Man wähle h so klein, daß für alle $x_k \in U$

$$\frac{1}{L} \left(e^{L(x_k - x_0)} - 1 \right) \max_{j=1}^{k} |T_h(x_j)| \leq d \; .$$

Dann gilt die Abschätzung von Satz 7.3.1 für alle $x_h \in U$, und die Konvergenz (von der Ordnung p) folgt.

7.4 Mehrschrittverfahren

Ein (lineares) m-Schrittverfahren hat die Form ($\alpha_m \neq 0$)

$$\begin{aligned} \sum_{\nu=0}^{m} \alpha_\nu y_{k+\nu} &= h \sum_{\nu=0}^{m} \beta_\nu f(x_{k+\nu}, y_{k+\nu}) \;, \quad k = 0, 1, \ldots \\ y_k &= \overline{y}_k \quad , \qquad\qquad k = 0, \ldots, m-1 \;. \end{aligned}$$

Es benötigt also m Startwerte $\overline{y}_0, \ldots, \overline{y}_{m-1}$. Diese können etwa durch ein Einschrittverfahren gewonnen werden. Gegenüber den Einschrittverfahren besitzen Mehrschrittverfahren den Vorteil, daß sie pro Schritt nur eine Funktionswertung (nämlich $f_\nu(x_{k+\nu}, y_{k+\nu})$, ν der größte Index mit $\beta_\nu \neq 0$) benötigen. Ein Beispiel eines 2-Schrittverfahrens ist die Mittelpunktsregel

$$y_{k+2} - y_k = 2hf(x_{k+1}, y_{k+1}) \;.$$

Ein Mehrschrittverfahren 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

$$\begin{aligned} \alpha_m y_{k+m}^{(t+1)} + \sum_{\nu=0}^{m-1} \alpha_\nu y_{k+\nu} &= h\beta_m f(x_{k+m}, y_{k+m}^{(t)}) \\ &+ h \sum_{\nu=0}^{m-1} \beta_\nu f(x_{k+\nu}, y_{k+\nu}) \end{aligned}$$

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 $O(h)$ 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. Wie üblich ist $\nabla^0 y_k = y_k$.

Lemma 7.4.1 *Es gilt für $q \geq 0$*

$$\nabla^q y_k = \sum_{\nu=0}^{q} (-1)^\nu \begin{pmatrix} q \\ \nu \end{pmatrix} y_{k-\nu} \quad , \quad y_{k-q} = \sum_{\nu=0}^{q} (-1)^\nu \begin{pmatrix} q \\ \nu \end{pmatrix} \nabla^\nu y_k \; .$$

Beweis: Dies kann man natürlich durch Induktion nach q beweisen. Es geht aber auch einfacher. Wir definieren auf dem linearen Raum der Folgen $y = (y_k)_{k=-\infty,+\infty}$ den linearen Operator

$$(Ty)_k = y_{k-1} \; .$$

Die binomische Formel ergibt

$$(I - T)^q = \sum_{\nu=0}^{q} \begin{pmatrix} q \\ \nu \end{pmatrix} (-1)^\nu T^\nu \; .$$

Wegen $I - T = \nabla$, $(T^\nu y)_k = y_{k-\nu}$ ist dies die erste Formel. Die zweite bekommen wir ganz entsprechend aus

$$T^q = (I - \nabla)^q = \sum_{\nu=0}^{q} \begin{pmatrix} q \\ \nu \end{pmatrix} (-\nabla)^\nu \; .$$

□

Lemma 7.4.2 *Das Polynom p vom Grade $\leq q$ mit $p(x_{k-\ell}) = y_{k-\ell}$ $\ell = 0, \ldots, q$ ist*

$$p(x) = \sum_{\nu=0}^{q} (-1)^\nu \begin{pmatrix} -s \\ \nu \end{pmatrix} \nabla^\nu y_k \quad , \quad s = \frac{x - x_k}{h} \; .$$

Hier sind die Binomialkoeffizienten für reelle s durch

$$\begin{pmatrix} s \\ \nu \end{pmatrix} = \frac{1}{\nu!} s(s-1) \ldots (s - (\nu - 1)) \; .$$

erklärt.

Beweis: Dies ist nichts anderes als die Newtonsche Form des Interpolationspolynoms für die Stützstellen $x_{k-q}, \ldots, x_k$. Wir führen den Beweis aber direkt. p ist ein Polynom vom Grade $\leq q$, denn es ist

$$\begin{pmatrix} -s \\ \nu \end{pmatrix} = \frac{1}{\nu!} (-s - \nu + 1) \ldots (-s) \in \mathcal{P}_\nu \; .$$

Weiter gilt für $0 \leq \mu \leq q$

$$\begin{aligned} p(x_{k-\mu}) &= \sum_{\nu=0}^{q} (-1)^\nu \begin{pmatrix} \mu \\ \nu \end{pmatrix} \nabla^\nu y_k \\ &= \sum_{\nu=0}^{\mu} (-1)^\nu \begin{pmatrix} \mu \\ \nu \end{pmatrix} \nabla^\nu y_k \\ &= y_{k-\mu} \end{aligned}$$

nach Lemma 7.4.1.

□

Zur Aufstellung konkreter Mehrschrittverfahren gibt es grundsätzlich zwei Möglichkeiten:

(a) Integration.

Aus der Differentialgleichung folgt durch Integration

$$y(x_{k+m}) - y(x_{k+\ell}) = \int_{x_{k+\ell}}^{x_{k+m}} f(x, y(x))dx \; .$$

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

$$y_{k+m} - y_{k+\ell} = \int_{x_{k+\ell}}^{x_{k+m}} p(x)dx \; .$$

Als Stützwerte werden bei der Interpolation die Zahlen $f_j = f(x_j, y_j)$ genommen. Es ist dann $p(x)$ eine lineare Funktion der Zahlen f_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:

$$\begin{aligned} y_{k+m} - y_{k+m-1} &= \int_{x_{k+m-1}}^{x_{k+m}} p(x)dx \; , \quad p(x) = \sum_{\nu=0}^{m-1} (-1)^\nu \binom{-s}{\nu} \nabla^\nu f_{k+m-1} \\ &= h(\gamma_0 f_{k+m-1} + \gamma_1 \nabla^1 f_{k+m-1} + \ldots + \gamma_{m-1} \nabla^{m-1} f_{k+m-1}) \; . \\ \gamma_\nu &= \frac{1}{h} \int_{x_{k+m-1}}^{x_{k+m}} (-1)^\nu \binom{-s}{\nu} dx \quad , \quad s = \frac{x - x_{k+m-1}}{h} \; , \end{aligned}$$

$$= (-1)^\nu \int_0^1 \binom{-s}{\nu} ds \quad .$$

Adams-Moulton:

$$\begin{aligned} y_{k+m} - y_{k+m-1} &= \int_{x_{k+m-1}}^{x_{k+m}} p(x)dx \,, \quad p(x) = \sum_{\nu=0}^{m} (-1)^\nu \binom{-s}{\nu} \nabla^\nu f_{k+m} \\ &= h(\gamma_0 f_{k+m} + \gamma_1 \nabla^1 f_{k+m} + \ldots + \gamma_m \nabla^m f_{k+m}) \;, \\ \gamma_\nu &= \frac{1}{h}(-1)^\nu \int_{x_{k+m-1}}^{x_{k+m}} \binom{-s}{\nu} dx \,, \;\; s = \frac{x - x_{k+m}}{h} \,, \\ &= (-1)^\nu \int_{-1}^{0} \binom{-s}{\nu} ds \,. \end{aligned}$$

Nyström:

$$\begin{aligned} y_{k+m} - y_{k+m-2} &= h(\gamma_0 f_{k+m-1} + \gamma_1 \nabla^1 f_{k+m-1} + \ldots + \gamma_{m-1} \nabla^{m-1} f_{k+m-1}) \,, \\ \gamma_\nu &= (-1)^\nu \int_{-1}^{+1} \binom{-s}{\nu} ds \,. \end{aligned}$$

Milne-Simpson:

$$\begin{aligned} y_{k+m} - y_{k+m-2} &= h(\gamma_0 f_{k+m} + \gamma_1 \nabla^1 f_{k+m} + \ldots + \gamma_m \nabla^m f_{k+m}) \,, \\ \gamma_\nu &= (-1)^\nu \int_{-2}^{0} \binom{-s}{\nu} ds \;\; . \end{aligned}$$

Die γ_ν sind in Tabellen erfaßt (siehe z.B. Henrici, S. 191):

ν	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	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{29}{9}$
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 $\gamma_0 = 2$, $\gamma_1 = 0$ 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 $y_k, \ldots, y_{k+m}$:

$$\begin{aligned} p'(x_{k+\ell}) &= f(x_{k+\ell}, y_{k+\ell}) \ , \\ p(x) &= \sum_{\nu=0}^{m} (-1)^\nu \binom{-s}{\nu} \nabla^\nu y_{k+m} \ , \quad s = \frac{x - x_{k+m}}{h} \ . \end{aligned}$$

Dies ergibt ein Verfahren der Form

$$\begin{aligned} \sum_{\nu=0}^{m} \gamma_{\nu,m-\ell} \nabla^\nu y_{k+m} &= hf_{k+\ell} \ , \\ \gamma_{\nu,m-\ell} &= h \frac{d}{dx} (-1)^\nu \binom{-s}{\nu} |_{x=x_{k+\ell}} \\ &= (-1)^\nu \frac{d}{ds} \binom{-s}{\nu} |_{s=\ell-m} \quad . \end{aligned}$$

Offenbar ist $\gamma_{0,m-\ell} = 0$. 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

$$\begin{aligned} \nabla^1 y_{k+2} - \frac{1}{2} \nabla^2 y_{k+2} &= hf_{k+1} \qquad \text{oder} \\ y_{k+2} - y_k &= 2hf_{k+1} \ , \end{aligned}$$

also wieder die Mittelpunktsregel.

7.5 Konvergenz von Mehrschrittverfahren

Den lokalen Diskretisierungsfehler eines Mehrschrittverfahrens erklärt man wie beim Einschrittverfahren durch

$$\begin{aligned} T_h(x_{k+m}) &= \frac{1}{h}\sum_{\nu=0}^{m}\alpha_\nu y(x_{k+\nu}) - \sum_{\nu=0}^{m}\beta_\nu f(x_{k+\nu}, y(x_{k+\nu})) \\ &= \frac{1}{h}\sum_{\nu=0}^{m}\alpha_\nu y(x_{k+\nu}) - \sum_{\nu=0}^{m}\beta_\nu y'(x_{k+\nu}) \end{aligned}$$

für die exakte Lösung y. Die Definition der Konsistenz erfolgt dann wörtlich wie beim Einschrittverfahren.

Beispiele:

1) Um die Konsistenzordnung des Adams-Bashforth-Verfahrens zu bestimmen, erinnern wir uns an die Herleitung des Verfahrens. Danach ist

$$T_h(x_{k+m}) = \frac{1}{h}(y(x_{k+m}) - y(x_{k+m-1})) - \frac{1}{h}\int_{x_{k+m-1}}^{x_{k+m}} p(x)dx \ ,$$

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 I.5.2 ist für $f \in C^m$

$$p - f = \mathrm{O}(h^m) \ ,$$

so daß wir

$$T_h(x_{k+m}) = \frac{1}{h}\int_{x_{k+m-1}}^{x_{k+m}} (f(x, y(x)) - p(x))dx = \mathrm{O}(h^m)$$

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 $y \in C^4$

$$\begin{array}{rcccccccccc} y(x_{k+\nu}) &=& y(x_k) &+& \nu h y'(x_k) &+& \frac{1}{2}\nu^2 h^2 y''(x_k) &+& \frac{1}{6}\nu^3 h^3 y'''(x_k) &+& \mathrm{O}(h^4) \ . \\ y'(x_{k+\nu}) &=& y'(x_k) &+& \nu h y''(x_k) &+& \frac{1}{2}\nu^2 h^2 y'''(x_k) & & & + & \mathrm{O}(h^3) \end{array}$$

und damit

$$T_h(x_{k+m}) = \left(2 - (1+a) - \frac{3-a}{2} + \frac{1+a}{2}\right) y'(x_k)$$

$$
\begin{aligned}
&+ \quad h\left(\frac{4}{2} - \frac{1}{2}(1+a) - \frac{1}{2}(3-a)\right) y''(x_k) \\
&+ \quad h^2\left(\frac{8}{6} - \frac{1}{6}(1+a) - \frac{1}{4}(3-a)\right) y'''(x_k) + \mathrm{O}(h^3) \\
&= \quad h^2\left(\frac{5}{12} + \frac{a}{12}\right) y'''(x_k) + \mathrm{O}(h^3) \quad .
\end{aligned}
$$

Wir haben also Konsistenzordnung $p = 3$ für $a = -5$ und $p = 2$ sonst.

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 $\lambda \in \mathbb{C}$

$$
\rho(\lambda) = \sum_{\nu=0}^{m} \alpha_\nu \lambda^\nu \quad , \qquad \sigma(\lambda) = \sum_{\nu=0}^{m} \beta_\nu \lambda^\nu \; .
$$

Die Eigenschaften dieser beiden Polynome werden sich für das Verhalten des Mehrschrittverfahrens als wichtig erweisen.

Definition 7.5.1 *Ein Mehrschrittverfahren heißt stabil, wenn für die Nullstellen λ von ρ folgende Bedingung erfüllt ist:*

a) $|\lambda| \leq 1$

b) *Ist $|\lambda| = 1$, so ist λ einfache Nullstelle.*

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.

Wir benutzen dieses Verfahren für $y' = 1 + y^2$, $y(0) = 0$. Für die Schrittweite $h = 0.01$ und $y_0 = 0$, $y_1 = \tan(h)$ ergibt sich

k	x_k	$y_k(a=0)$	$y_k(a=1)$	$y_k(a=1.1)$	$y(x_k)$
0	0.0	0.00000	0.00000	0.00000	0.00000
20	0.2	0.20269	0.20251	0.20232	0.20271
40	0.4	0.42775	0.42185	0.41821	0.42279
60	0.6	0.68404	0.68139	0.64780	0.68414
80	0.8	1.02939	1.02239	0.76585	1.02964
100	1.0	1.55667	1.53706	−0.23823	1.55741
120	1.2	2.56893	2.49939	−6.70901	2.57215
140	1.4	5.75965	5.27453	−24.76387	5.79788

Für y_0, y_1 haben wir die exakten Werte genommen.

Für die instabilen Verfahren mit $a = 1$ und $a = 1.1$ treten offenbar Probleme auf.

Definition 7.5.2 *Ein Mehrschrittverfahren heißt konvergent, wenn für alle Startwerte* $\overline{y}_k$ *mit* $\lim_{h\to 0} |y(x_k) - \overline{y}_k| = 0$, $k = 0, \ldots, m-1$

$$\lim_{h\to 0} \max_{x_k \in U} |y(x_k) - y_k| = 0$$

gilt. Es heißt konvergent von der Ordnung p, wenn aus $y(x_k) - \overline{y}_k = O(h^p)$, $k = 0, \ldots, m-1$

$$\max_{x_k \in U} |y(x_k) - y_k| = O(h^p)$$

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

$$\sum_{\nu=0}^{m} \alpha_\nu \lambda^{k+\nu} = \lambda^k \rho(\lambda) = 0 \quad , \quad k = 0, 1, \ldots \ ,$$

d.h. wenn $\rho(\lambda) = 0$ ist. Ist λ eine zweifache Nullstelle von ρ, so ist $\rho'(\lambda) = 0$ und damit auch

$$\begin{aligned}\sum_{\nu=0}^{m} \alpha_\nu (k+\nu)\lambda^{k+\nu} &= \lambda^k \left\{ k \sum_{\nu=0}^{m} \alpha_\nu \lambda^\nu + \sum_{\nu=0}^{m} \alpha_\nu \nu \lambda^\nu \right\} \\ &= \lambda^k \{ k\rho(\lambda) + \lambda\rho'(\lambda) \} = 0 \ ,\end{aligned}$$

d.h. es ist auch $z_k = k\lambda^k$ Lösung. Genau so sieht man, daß im Falle einer r-fachen Wurzel λ 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 Differentialgleichungen gefunden.

Satz 7.5.1 *Seien $\lambda_1, \ldots, \lambda_n$ die Nullstellen von ρ mit den Vielfachheiten $r_1, \ldots, r_n$. Dann sind*

$$\lambda_j^k \ , \ k\lambda_j^k, \ldots, k^{r_j-1}\lambda_j^k \quad , \quad j = 1, \ldots, n$$

$m = r_1 + \ldots + r_n$ Lösungen der homogenen Differenzengleichung. Jede weitere Lösung z_k ist eine Linearkombination dieser Lösungen, d.h. es gilt mitKonstanten a_{jr}

$$z_k = \sum_{j=1}^{n} \sum_{r=0}^{r_j-1} a_{jr} k^r \lambda_j^k \ .$$

Die a_{jr} sind eindeutig bestimmt.

Beweis: Sei $\alpha_m = 1$. Die Differenzengleichung kann dann in der Form

$$Z_{k+1} = AZ_k \ , \ Z_k = \begin{bmatrix} z_k \\ \vdots \\ z_{k+m-1} \end{bmatrix} \ , \quad A = \begin{bmatrix} 0 & 1 & 0 & & \cdots & & 0 \\ 0 & 0 & 1 & 0 & & \cdots & 0 \\ & & \ddots & \ddots & & & \\ & & & \ddots & \ddots & & \\ & & & & \ddots & \ddots & \\ 0 & & & & & 0 & 1 \\ -\alpha_0 & -\alpha_1 & & \cdots & & & -\alpha_{m-1} \end{bmatrix}$$

geschrieben werden.

Dann ist also $Z_k = A^k Z_0$. Ist $J = X^{-1}AX$ die Jordan'sche Normalform, so ist

$$Z_k = XJ^kX^{-1}Z_0 \ .$$

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_ℓ haben wir schon in 4.2 ausgerechnet; wir fanden

$$J_\ell^k = \lambda_\ell^k (A_0 + kA_1 + \ldots + k^{r_\ell - 1} A_{r_\ell - 1})$$

mit gewissen Matrizen A_j, die noch von λ_ℓ abhängen. Z_k ist also wirklich als Linearkombination der genannten Ausdrücke darstellbar.

□

Als wichtige Folgerung aus Satz 7.5.1 haben wir:

Satz 7.5.2 *Ein Mehrschrittverfahren ist genau dann stabil, wenn alle Lösungen der Differenzengleichung*

$$\sum_{\nu=0}^{m} \alpha_\nu z_{k+\nu} = 0 \quad , \qquad k = 0, 1, \ldots$$

für $k \to \infty$ beschränkt bleiben.

Satz 7.5.3 *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 $\lambda_1, \ldots, \lambda_r$ die Eigenwerte von A mit $|\lambda_i| = \rho(A)$. Dann gibt es eine Matrix X, so daß

$$X^{-1}AX = \left(\begin{array}{ccc|c} \lambda_1 & & & \\ & \ddots & & \mathbf{O} \\ & & \lambda_r & \\ \hline \mathbf{O} & & & B \end{array}\right) \quad ,$$

wo die $(m-r, m-r)$-Matrix B nur noch die Eigenwerte mit Betrag $< \rho(A)$ hat. Nach Satz 2.5.1 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

$$\left\| \begin{pmatrix} x^r \\ x^{m-r} \end{pmatrix} \right\| = \max\{\|x^r\|_\infty, \|x^{m-r}\|_{m-r}\}$$

ein, so leistet die Norm $\|X^{-1}x\|$ das Gewünschte.

□

Satz 7.5.4 *f erfülle in einer Umgebung D der Lösung $(x, y(x))_{x \in U}$ eine Lipschitz-Bedingung.*

Das Mehrschrittverfahren sei stabil. Dann gibt es Konstanten C_1, C_2, $h_0 > 0$, so daß für $h < h_0$

$$|y(x_k) - y_k| \leq C_1 e^{C_2(x_k - x_0)} \left\{ \max_{k=0}^{m-1} |y(x_k) - y_k| + \max_{j=m}^{k} |T_h(x_j)| \right\} \ ,$$

solange $(x_k, y_k) \in D$.

Beweis: Nach Definition des lokalen Diskretisierungsfehlers ist

$$\sum_{\nu=0}^{m} \alpha_\nu y(x_{k+\nu}) = h \sum_{\nu=0}^{m} \beta_\nu f(x_{k+\nu}, y(x_{k+\nu})) + hT_h(x_{k+m}) \ ,$$

und das Verfahren lautet

$$\sum_{\nu=0}^{m} \alpha_\nu y_{k+\nu} = h \sum_{\nu=0}^{m} \beta_\nu f(x_{k+\nu}, y_{k+\nu}) \ .$$

Substraktion ergibt mit $d_k = y(x_k) - y_k$

$$\begin{aligned} \sum_{\nu=0}^{m} \alpha_\nu d_{k+\nu} &= h \sum_{\nu=0}^{m} \beta_\nu (f(x_{k+\nu}, y(x_{k+\nu})) - f(x_{k+\nu}, y_{k+\nu}) + hT_h(x_{k+m}) \\ &= hc_k \quad . \end{aligned}$$

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

$$D_{k+1} = AD_k + hc_k B$$

schreiben. A hat $\rho(\lambda)$ als charakteristisches Polynom. Nach Satz 7.3.1 gibt es also eine Vektornorm $\| \ \|$, so daß $\|A\| \leq 1$ und damit

$$\|D_{k+1}\| \leq \|D_k\| + h|c_k|\|B\| \ .$$

Solange $(x_k, y_k) \in D$ 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

$$\begin{aligned} &\|D_{k+1}\| \leq q\|D_k\| + ha_k \quad , \\ &q = (1 + h\|B\|K)/(1 - h\|B\|K) \ , \ \ a_k = |T_h(x_{k+m})|\|B\|/(1 - h\|B\|K) \ . \end{aligned}$$

Nach Lemma 7.3.1 folgt

$$\|D_k\| \leq q^k \|D_0\| + h \sum_{j=0}^{k-1} q^{k-j-1} a_j \ .$$

Nun benutzen wir die Ungleichung

$$\frac{1+x}{1-x} \leq 1+4x \quad , \quad 0 \leq x \leq \frac{1}{2} \, .$$

Dann wird für $h\|B\|K \leq 1/2 \quad q \leq 1+4h\|B\|K$ und damit

$$q^k \leq \left(1+4\|B\|K\frac{x_k-x_0}{k}\right)^k \leq e^{4\|B\|K(x_k-x_0)} \, .$$

Für D_k ergibt sich nun durch Aufsummieren der geometrischen Reihe

$$\begin{aligned} \|D_k\| &\leq q^k\|D_0\| + h\frac{q^k-1}{q-1} \max_{j=0}^{k-1} a_j \\ &\leq e^{4\|B\|K(x_k-x_0)} \left(\|D_0\| + \frac{1}{4\|B\|K} \max_{j=0}^{k-1} |T_h(x_{j+m})|\right) \, . \end{aligned}$$

Da in $\mathbb{R}^m$ alle Normen äquivalent sind, folgt die behauptete Ungleichung.

□

Hieraus folgt wie in §3 sofort

Satz 7.5.5 *f erfülle die Voraussetzungen von Satz 4. 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 7.5.6 *Ist ein Mehrschrittverfahren konvergent für $y' = 0$, $y(0) = 0$, so ist es stabil.*

Beweis: Sei λ eine Wurzel von ρ der Vielfachheit r. Wir geben die Anfangswerte

$$\overline{y}_k = k^{r-1}\lambda^k h \quad , \qquad k = 0, \ldots, m-1 \, .$$

vor. Das Verfahren lautet

$$\sum_{\nu=0}^{m} \alpha_\nu y_{k+\nu} = 0 \quad , \quad k = 0, 1, \ldots \quad , \quad y_k = \overline{y}_k \quad , \quad k = 0, \ldots, m-1 \, .$$

Nach Satz 7.5.1 ist

$$y_k = k^{r-1}\lambda^k h \quad , \qquad k = 0, 1, \ldots \quad ,$$

Wir lassen nun $h \to 0$ und $k \to \infty$ so streben, daß $x_k = hk = \overline{x} > 0$. Dann muß wegen der vorausgesetzten Konvergenz $y_k \to 0$ streben. Also folgt

$$\lim_{k\to\infty} \left(\frac{\overline{x}}{k}\right) k^{r-1}\lambda^k = 0 \, .$$

Dies ist nur möglich, wenn $|\lambda| \leq 1$ und $r = 1$ für $|\lambda| = 1$.

□

7.6 Konsistenz und Stabilität von 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 7.6.1 *Sei* $\varphi(\lambda) = \frac{\rho(\lambda)}{\ell n \lambda} - \sigma(\lambda)$. *Das Mehrschrittverfahren ist genau dann konsistent, wenn* $\varphi(1) = 0$. *Es ist genau dann konsistent von der Ordnung* p, *wenn* φ *bei* $\lambda = 1$ *eine Nullstelle der Ordnung* p *hat.*

Beweis: Für $y \in C^{p+1}$ ist

$$\begin{aligned} y(x_{k+\nu}) &= y(x_k) + \nu h y'(x_k) + \ldots + \tfrac{(\nu h)^p}{p!} y^{(p)}(x_k) + \mathrm{O}(h^{p+1}) \\ y'(x_{k+\nu}) &= y'(x_k) + \nu h y''(x_k) + \ldots + \tfrac{(\nu h)^{p-1}}{(p-1)!} y^{(p)}(x_k) + \mathrm{O}(h^p) \,. \end{aligned}$$

Dies ergibt für den lokalen Diskretisierungsfehler

$$\begin{aligned} T_h(x_{k+m}) &= \frac{1}{h} \sum_{\nu=0}^{m} \alpha_\nu y(x_{k+\nu}) - \sum_{\nu=0}^{m} \beta_\nu y'(x_{k+\nu}) \\ &= \frac{1}{h} C_0 y(x_k) + C_1 y'(x_k) + \ldots + h^{p-1} C_p y^{(p)}(x_k) + \mathrm{O}(h^p) \,, \end{aligned}$$

$$\begin{aligned} C_0 &= \sum_{\nu=0}^{m} \alpha_\nu \quad , \\ C_1 &= \sum_{\nu=0}^{m} \nu \alpha_\nu - \sum_{\nu=0}^{m} \beta_\nu \quad , \\ &\vdots \\ C_p &= \tfrac{1}{p!} \sum_{\nu=0}^{m} \nu^p \alpha_\nu - \tfrac{1}{(p-1)!} \sum_{\nu=0}^{m} \nu^{p-1} \beta_\nu \,. \end{aligned}$$

Sei nun

$$\chi(z) = \varphi(e^z) = \frac{1}{z} \sum_{\nu=0}^{m} \alpha_\nu e^{\nu z} - \sum_{\nu=0}^{m} \beta_\nu e^{\nu z} \,.$$

Die Potenzreihe um $z = 0$ für $e^{\nu z}$ ergibt für $z \to 0$

$$\begin{aligned} \chi(z) &= \frac{1}{z} \sum_{\nu=0}^{m} \alpha_\nu \sum_{\mu=0}^{p} \frac{(\nu z)^\mu}{\mu!} - \sum_{\nu=0}^{m} \beta_\nu \sum_{\mu=0}^{p-1} \frac{(\nu z)^\mu}{\mu!} + \mathrm{O}(z^p) \\ &= \frac{1}{z} C_0 + C_1 + \ldots + z^{p-1} C_p + \mathrm{O}(z^p) \quad . \end{aligned}$$

Nun gilt:

$$\begin{aligned} & \varphi \text{ hat } p\text{-fache Nullstelle bei } \lambda = 1 \\ \Leftrightarrow\ & \chi \text{ hat } p\text{-fache Nullstelle bei } z = 0 \\ \Leftrightarrow\ & C_0 = C_1 = \ldots = C_p = 0 \\ \Leftrightarrow\ & T_h(x_{k+m}) = \mathrm{O}(h^p) \qquad \text{für alle } y \in C^{p+1} \,. \end{aligned}$$

Dies erledigt den Fall der Konsistenzordnung p. Konsistenz schlechthin ist gleichbedeutend mit $C_0 = C_1 = 0$, d.h. mit $\chi(0) = 0$ und damit $\varphi(1) = 0$.

□

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 7.6.2 *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 gebrochen 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 0, i, ∞ über. Das Innere des Einheitskreises muß dabei in die linke Halbebene übergehen, weil 0 in -1 übergeht.

(ii) Die Koeffizienten eines reellen Polynoms, dessen Wurzeln nur Realteile ≤ 0 haben, haben alle das gleiche Vorzeichen.

Denn ist r ein solches Polynom und sind x_μ die reellen und $x_\nu \pm iy_\nu$ die konjugierten 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

$$\frac{z}{\ell n \frac{1+z}{1-z}} = c_0 + c_2 z^2 + c_4 z^4 + \ldots$$

sind negativ für $\nu > 0$ (siehe Henrici, S. 223).

Nun zum Beweis des Satzes! Seien ρ, σ die Polynome eines stabilen Verfahrens der Konsistenzordnung p. Wir setzen

$$r(w) = \left(\frac{1-w}{2}\right)^m \rho\left(\frac{1+w}{1-w}\right) \quad , \quad s(w) = \left(\frac{1-w}{2}\right)^m \sigma\left(\frac{1+w}{1-w}\right) .$$

Dann hat nach (i) und wegen der Stabilität r bei $w = 0$ eine einfache Nullstelle und sonst nur Nullstellen mit Realteil ≤ 0. 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_ℓ hat das Vorzeichen von a_1, $\ell = 2, \ldots, m$. Sei nun weiter

$$f(w) = \left(\frac{1-w}{2}\right)^m \varphi\left(\frac{1+w}{1-w}\right) \quad , \quad \varphi(z) = \frac{\rho(z)}{\ell n z} - \sigma(z) \, .$$

Nach Satz 7.6.1 ist die Ordnung p der Nullstelle von f bei 0 gleich der Konsistenzordnung des Verfahrens. Offenbar ist

$$\begin{aligned} f(w) &= \frac{r(w)}{\ell n \frac{1+w}{1-w}} - s(w) \\ &= b_0 + b_1 w + \ldots + b_{p-1} w^{p-1} + \ldots - s(w) \quad . \end{aligned}$$

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. Es ist dann $p \leq m + 1$ und der Satz richtig.

Wir berechnen nun die b_ν. Es ist nach (iii)

$$\begin{aligned} b_0 + b_1 w + \ldots &= \frac{w}{\ell n \frac{1+w}{1-w}} \, \frac{r(w)}{w} \\ &= (c_0 + c_2 w^2 + c_4 w^4 + \ldots)(a_1 + a_2 w + \ldots + a_m w^{m-1}) \end{aligned}$$

mit $c_{2\nu} < 0$, $\nu > 0$. Ausmultiplikation und Koeffizientenvergleich für die geraden Potenzen ergibt

$$b_{2\nu} = c_0 a_{2\nu+1} + c_2 a_{2\nu-1} + \ldots + c_{2\nu} a_1 \quad ,$$

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

$$b_{m+1} = c_0 a_{m+2} + c_2 a_m + \ldots + c_{m+1} a_1 \quad .$$

Es ist $a_{m+2} = 0$, $c_{2\nu} < 0$, die a_ℓ 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 $p \leq m + 1$ sein.

(b) m gerade. Wir setzen $2\nu = m + 2$ und bekommen

$$b_{m+2} = c_0 a_{m+3} + c_2 a_{m+1} + c_4 a_{m-1} + \ldots + c_{m+2} a_1 \, .$$

Wie oben folgt $b_{m+2} \neq 0$, d.h. es muß $p - 1 < m + 2$ oder $p \leq m + 2$ sein.

□

Definition 7.6.1 *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.

$$y_{k+2} - y_k = h(2f_{k+2} - 2\nabla f_{k+2} + \frac{1}{3}\nabla^2 f_{k+2})$$

ist identisch zu dem Verfahren für $m = 3$. Es hat also die Konsistenzordnung $3 + 1 = 4$ und ist daher optimal. Dagegen hat die Mittelpunktsregel

$$y_{k+2} - y_k = 2hf_{k+1}$$

nur die Konsistenzordnung 2 und ist also nicht optimal.

7.7 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

$$y(x,h) = y(x) + h^p e_p(x) + \ldots + h^q e_q(x) + \mathrm{O}(h^{q+1}) \tag{7.1}$$

mit von h unabhängigen Funktionen e_ℓ, so kann man ähnlich wie beim Romberg-Verfahren vorgehen: Man wiederholt die Rechnung mit kleinerer Schrittweite, etwa $\frac{h}{2}$, und hat dann

$$y(x,\frac{h}{2}) = y(x) + 2^{-p}h^p e_p(x) + \ldots + 2^{-q}h^q e_q(x) + \mathrm{O}(h^{q+1})\ . \tag{7.2}$$

Nun wird der Term der Ordnung h^p eliminiert:

$$\begin{aligned} 2^p y(x,\frac{h}{2}) - y(x,h) &= (2^p-1)y(x) + (2^{-1}-1)h^{p+1}e_{p+1}(x) + \ldots \\ &\quad +(2^{p-q}-1)h^q e_q(x) + \mathrm{O}(h^{q+1}) \quad . \end{aligned}$$

In

$$y^1(x,h) = \frac{1}{2^p-1}(2^p y(x,\frac{h}{2}) - y(x,h)) = y(x,\frac{h}{2}) + \frac{1}{2^p-1}(y(x,\frac{h}{2}) - y(x,h))$$

hat man also eine neue Formel mit der Ordnung $p+1$, und für diese gilt

$$y^1(x,h) = y(x) + h^{p+1}e^1_{p+1}(x) + \ldots + h^q e^1_q(x) + \mathrm{O}(h^{q+1})$$

mit neuen, ebenfalls von h unabhängigen Funktionen e^1_ℓ. Entsprechend konstruiert man Formeln $y^2, y^3, \ldots$ der Ordnungen $p+2, p+3, \ldots$.

Eine andere Anwendung von (7.1) betrifft die Schätzung des Fehlers. Subtraktion von (7.1), (7.2) ergibt

$$y(x,h) - y(x,\frac{h}{2}) = h^p(1-2^{-p})e_p(x) + \mathrm{O}(h^{p+1})\ ,$$

also

$$\begin{aligned} y(x,\frac{h}{2}) - y(x) &= 2^{-p}h^p e_p(x) + \mathrm{O}(h^{p+1}) \\ &= \frac{1}{2^p-1}(y(x,h) - y(x,\frac{h}{2})) + \mathrm{O}(h^{p+1}) \quad . \end{aligned}$$

Für hinreichend kleine h ist also

$$y(x,\frac{h}{2}) - y(x) \sim \frac{1}{2^p-1}(y(x,h) - y(x,\frac{h}{2}))\ .$$

Dies ist eine Schätzung für den Fehler von $y(x, \frac{h}{2})$.

Das Bestehen von (7.1) ist für Einschrittverfahren unter allgemeinen Voraussetzungen bewiesen (siehe Bulirsch-Stoer II, 7.2.3), für Mehrschrittverfahren im allgemeinen aber nicht.

Beispiel: Die Mittelpunktsregel für die Anfangswertaufgabe $y' = -y$, $y(0) = 1$ lautet

$$y_{k+2} = y_k - 2hy_{k+1} \quad , \qquad k = 0, 1, \ldots \ ,$$

und wir nehmen die Startwerte $y_0 = 1$, $y_1 = 1 - h \ \ (= y(h) + \mathrm{O}(h^2))$. Nach Satz 7.5.1 ist

$$y_k = C_1\lambda_1^k + C_2\lambda_2^k$$

mit den Wurzeln

$$\lambda_1 = \sqrt{1+h^2}\left(1 - \frac{h}{\sqrt{1+h^2}}\right) \ , \ \ \lambda_2 = -\sqrt{1+h^2}\left(1 + \frac{h}{\sqrt{1+h^2}}\right)$$

von $\rho(\lambda) = \lambda^2 + 2h\lambda - 1$ und mit Konstanten C_1, C_2 mit

$$\begin{array}{rcrcl} C_1 & + & C_2 & = & 1 \\ C_1\lambda_1 & + & C_2\lambda_2 & = & 1 - h \ . \end{array}$$

Sei nun $x > 0$ fest und $hk = x$. Dann ist

$$y(x,h) = C_1(h)\lambda_1(h)^{x/h} + C_2(h)\lambda_2(h)^{x/h} \ ,$$

wobei wir jetzt die Abhängigkeit von C_i, λ_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 $h \to 0$

$$\begin{aligned} \ell n\ \lambda_1(h)^{x/h} &= \frac{x}{h}\ell n\ \lambda_1(h) = \frac{x}{h}\ell n(1 + a_1h + a_2h^2 + \ldots) \\ &= \frac{x}{h}(b_1h + b_2h^2 + \ldots) = x(b_1 + b_2h + \ldots) \ . \end{aligned}$$

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 (7.1) ist daher keineswegs einfach. Das bekannteste Verfahren dieser Art stammt von Gragg: Sei n gerade.

$$\begin{array}{lcll} y_0 & = & y(x_0) & \\ y_1 & = & y_0 + hf(x_0, y_0) & \text{(Euler)} \\ y_{k+2} - y_k & = & 2hf(x_{k+1}, y_{k+1}) \ , \quad k = 0, \ldots, n-1 & \text{(Mittelpunktsregel)} \\ y_n & = & \frac{1}{4}y_{n+1} + \frac{1}{2}y_n + \frac{1}{4}y_{n-1} & \text{(Glättung)} \end{array}$$

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_ℓ. Dies führt natürlich zu (7.1), 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

$$\text{Gragg } (x, y, h, tol, f) \ ,$$

welches folgendes leistet: Man setzt der Reihe nach

$$\begin{array}{rcl} n & = & 2, 4, \ldots \\ h & = & \frac{h}{2}, \frac{h}{4}, \ldots \end{array}$$

und berechnet für $y(x+h)$ mit dem Gragg-Verfahren Näherungen

$$\begin{array}{rcl} T_1(h), T_1(\frac{h}{2}), \ldots, & & \text{und zwar mit} \\ T_1(h) & = & y_2 \text{ mit Schrittweite } h \ , \\ T_2(h) & = & y_4 \text{ mit Schrittweite } h/2 \text{ usw. .} \end{array}$$

Man erstellt etwa 4-6 Spalten des Romberg-Schemas und prüft, ob die Fehlerschätzung in der rechten Spalte zuverlässig ist und hinreichend klein ausfällt. Ist dies der Fall, so wird $x \to x + h$, $y \to y(x + h)$ 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 hinreichend kleiner Fehler auf, so wird h vergrößert.

7.8 Systeme von Differentialgleichungen und Differentialgleichungen höherer Ordnung

Wir betrachten nun die Anfangswertaufgabe für Systeme von n Differentialgleichungen 1. Ordnung

$$\begin{array}{rclcrcll} y_1' & = & f_1(x, y_1, \ldots, y_n) & , & y_1(x_0) & = & y_{10} & , \\ y_2' & = & f_2(x, y_1, \ldots, y_n) & , & y_2(x_0) & = & y_{20} & , \\ \vdots & & & & & & & \\ y_n' & = & f_n(x, y_1, \ldots, y_n) & , & y_n(x_0) & = & y_{n0} & . \end{array}$$

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

$$y' = f(x,y) \quad , \quad y(x_0) = y_0$$

schreiben. Damit übertragen sich alle Aussagen und Verfahren für skalare Differentialgleichungen unmittelbar auf Systeme. Zum Beispiel lautet die Lipschitzbedingung für Systeme

$$\|f(x,y) - f(x,\overline{y})\| \leq L\|y - \overline{y}\|$$

mit einer Vektornorm $\|\cdot\|$. Das Euler-Verfahren lautet

$$y_{k+1} = y_k + hf(x_k, y_k) \quad , \quad k = 0, 1, \ldots \quad ,$$

wobei jetzt y_k eine Näherung für den Vektor $y(x_k) = (y_1(x_k), \ldots, y_n(x_k))^T$ bedeutet.

Die Anfangswertaufgabe für Differentialgleichungen n-ter Ordnung lautet

$$\begin{array}{rcll} y^{(n)} & = & f(x, y, y', \ldots, y^{(n-1)}) & , \\ y^{(i)}(x_0) & = & y_0^{(i)} \ , \quad i = 0, 1, \ldots, n-1 & . \end{array}$$

Setzt man

$$y_1 = y \ , \ y_2 = y', \ldots, y_n = y^{(n-1)} \quad ,$$

so entsteht eine Anfangswertaufgabe für ein System 1. Ordnung:

$$\begin{array}{rclcrcll} y_1' & = & y_2 & , & y_1(x_0) & = & y_0 & , \\ y_2' & = & y_3 & , & y_2(x_0) & = & y_0' & , \\ \vdots & & & & \vdots & & & \\ y_{n-1}' & = & y_n & , & y_{n-1}(x_0) & = & y_0^{(n-2)} & , \\ y_n' & = & f(x, y_1, y_2, \ldots, y_n) & , & y_n(x_0) & = & y_0^{(n-1)} & . \end{array}$$

Entsprechend kann man Systeme höherer Ordnung auf Systeme 1. Ordnung zurückzuführen, 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 Erde und Mond in einer gemeinsamen Ebene beschreibt. Rechnet man im Schwerpunktsystem von Erde und Mond, so hat man folgende Konstellation:

Körper	Masse	Koordinaten
Mond	μ	$(1-\mu, 0)$
Erde	$1-\mu$	$(-\mu, 0)$
Satellit	0	$(x(t), y(t))$

Hier ist μ $(0 < \mu < 1)$ 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:

$$\begin{aligned} \ddot{x} &= 2\dot{y} + x + F_x(x,y) \\ \ddot{y} &= y - 2\dot{x} + F_y(x,y) \\ F(x,y) &= \frac{1-\mu}{((x+\mu)^2+y^2)^{1/2}} + \frac{\mu}{((x+\mu-1)^2+y^2)^{1/2}} \end{aligned} \qquad \dot{x} = \tfrac{d}{dt}x \text{ usw.}$$

Dazu kommen noch Anfangswerte für $x, \dot{x}, y, \dot{y}$. Setzen wir

$$y_1 = x \quad , \quad y_2 = y \quad , \quad y_3 = \dot{x} \quad , \quad y_4 = \dot{y} \, ,$$

so bekommt man das System 1. Ordnung

$$\begin{aligned} \dot{y}_1 &= y_3 \\ \dot{y}_2 &= y_4 \\ \dot{y}_3 &= 2y_4 + y_1 + F_x(y_1, y_2) \\ \dot{y}_4 &= -2y_3 + y_2 + F_y(y_1, y_2) \end{aligned}$$

$y_1(0) = x(0)$, $y_2(0) = y(0)$, $y_3(0) = \dot{x}(0)$, $y_4(0) = \dot{y}(0)$.

7.9 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

$$\begin{aligned} y' &= f(x,y) \quad , \quad a \le x \le b \\ g(y(a), y(b)) &= 0 \qquad\qquad . \end{aligned} \tag{9.1}$$

Man spricht dann von einer Randwertaufgabe.

Ein besonders einfacher Fall ist die lineare Randwertaufgabe 2. Ordnung

$$\begin{aligned} y'' + p(x)y' + q(x)y(x) &= f(x) \quad , \quad a \le x \le b \\ y(a) = 0 \quad , \quad y(b) &= 0 \quad , \end{aligned} \tag{9.2}$$

welche man natürlich in (9.1) überführen kann. Es seien $p, q, f \in C[a,b]$.

Satz 7.9.1 *Das homogene Problem*

$$y'' + py' + qy = 0\,, \quad y(a) = 0\,, \quad y(b) = 0$$

besitze nur die triviale Lösung $y = 0$. Dann ist (9.2) eindeutig lösbar.

Beweis: Die Eindeutigkeit ist klar. Wir brauchen daher nur die Lösbarkeit zu zeigen. Wir beschränken uns auf den Fall $p = 0$. Seien y_1, y_2 Lösungen der homogenen Gleichung mit

$$\begin{aligned} y_1(a) = 0 \quad , \quad y_1'(a) &= 1 \\ y_2(b) = 0 \quad , \quad y_2'(b) &= 1\,. \end{aligned}$$

Wir setzen

$$G(x,y) = c \begin{cases} y_1(x)y_2(x') & , \quad x \le x' \\ y_2(x)y_1(x') & , \quad x \ge x' \end{cases}$$

mit einer noch zu bestimmenden Konstanten c. Wir werden sehen, daß

$$y(x) = \int_a^b G(x,x')f(x')dx'$$

eine Lösung ist. Es ist

$$y(x) \;=\; c\int_a^x y_1(x)y_2(x')f(x')dx' + c\int_x^b y_2(x)y_1(x')f(x')dx'$$

$$
\begin{aligned}
y'(x) &= c\int_a^x y_1'(x)y_2(x')f(x')dx' + c\int_x^b y_2'(x)y_1(x')f(x')dx' \\
y''(x) &= c(y_1'y_2 - y_2'y_1)f(x) \\
&\quad + \int_a^x y_1(x'')y_2(x')f(x')dx' + \int_x^b y_2''(x)y_1(x')f(x')dx' \,,
\end{aligned}
$$

also

$$
y'' + qy = cWf \,, \quad W = y_1'y_2 - y_2'y_1 \,.
$$

Nun ist

$$
W' = y''_1 y_2 - y_2''y_1 = 0 \,, \quad W(a) = y_2(a) \neq 0 \,,
$$

also W eine Konstante $\neq 0$. Mit $c = \frac{1}{W}$ sind wir fertig.

□

Eine einfache und effiziente Methode zur Lösung von (9.2) 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

$$
\begin{aligned}
y''(x_i) &= \frac{y(x_{i+1}) - 2y(x_i) + y(x_{i-1})}{h^2} + \mathrm{O}(h^2) \,, \\
y'(x_i) &= \frac{y(x_{i+1}) - y(x_{i-1})}{2h} \qquad + \mathrm{O}(h^2) \,.
\end{aligned}
$$

vgl. §7.4. Es ist daher naheliegend, Näherungen y_i für $y(x_i)$ als Lösung des Systems

$$
\begin{gathered}
\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + p(x_i)\frac{y_{i+1} - y_{i-1}}{2h} + q(x_i)y_i = f(x_i) \,, \quad i = 1, \ldots, n-1 \,, \\
y_0 = 0 \quad , \quad y_n = 0
\end{gathered}
\tag{9.3}
$$

zu berechnen. Dies ist ein lineares System mit Tridiagonalmatrix.

Satz 7.9.2 *(9.2) sei eindeutig lösbar. Dann gibt es Konstanten h_0, $C > 0$, so daß für $h < h_0$ auch (9.3) 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

$$
L_h y_i = \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + p(x_i)\frac{y_{i+1} - y_{i-1}}{2h} + q(x_i)y_i \quad , \quad i = 1, \ldots, n-1 \,.
$$

Dann ist für $y \in C^4$ der "lokale Diskretisierungsfehler"

$$L_h y(x_i) - Ly(x_i) = T_h(x_i) = \mathrm{O}(h^2) \ .$$

Wegen $Ly = f$ folgt

$$L_h y(x_i) = f(x_i) + T_h(x_i) \ ,$$

und das Verfahren lautet

$$L_h y_i = f(x_i) \ .$$

Durch Subtraktion findet man für die Fehler $d_i = y(x_i) - y_i$

$$L_h d_i = T_h(x_i) \quad , \qquad i = 1, \ldots, n-1 \ .$$

Mit den Vektoren und Matrizen

$$d_h = \begin{pmatrix} d_1 \\ \vdots \\ d_{n-1} \end{pmatrix} \ , \quad T_h = \begin{pmatrix} T_h(x_1) \\ \vdots \\ T_h(x_{n-1}) \end{pmatrix} \ , \quad A_h = \begin{pmatrix} a_1 & b_1 & & & \\ c_2 & a_2 & b_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & b_{n-2} \\ & & & c_{n-1} & a_{n-1} \end{pmatrix} \ ,$$

$$a_i = -\frac{2}{h^2} + q(x_i) \ , \ \ b_i = \frac{1}{h^2} + \frac{p(x_i)}{2h} \ , \ \ c_i = \frac{1}{h^2} - \frac{p(x_i)}{2h}$$

schreibt sich dies

$$A_h d_h = T_h \ .$$

Ist A_h invertierbar, so folgt

$$\|d_h\|_\infty \leq \|A_h^{-1}\|_\infty \|T_h\|_\infty \ .$$

Kann man also für $h < h_0$ mit geeigneten $h_0 > 0$

$$\|A_h^{-1}\|_\infty \leq C \qquad \text{(unabhängig von } h\text{)} \tag{9.4}$$

zeigen, so folgt

$$\|d_h\|_\infty \leq C \|T_h\|_\infty = \mathrm{O}(h^2)$$

und der Satz ist bewiesen. (9.4) ist die Stabilitätsbedingung. Sie ist nicht leicht zu beweisen. Wir verweisen dazu auf die Literatur.

□

Allgemeiner kann man nichtlineare Randwertaufgaben

$$\begin{aligned} y'' &= f(x, y, y') \quad , \qquad a \leq x \leq b \\ y(a) &= \alpha \qquad , \quad y(b) = \beta \end{aligned} \tag{9.5}$$

betrachten. Zur Lösung kann man ebenso vorgehen wie bei der linearen Aufgabe (9.2) und erhält dann anstelle des linearen Gleichungssystems (9.3) ein nichtlineares Gleichungssystem, das man etwa mit Hilfe des Newton-Verfahrens (vgl. §3.3) lösen kann. Eine Alternative ist das "Schießverfahren": Man löst die Anfangswertaufgabe

$$\begin{aligned} y'' &= f(x,y,y') \quad , \\ y(a) &= \alpha \qquad , \quad y'(a) = s \end{aligned}$$

und versucht, s so zu bestimmen, daß $y(b) = \beta$. Bezeichnet man die Lösung der Anfangswertaufgabe mit $y(x,s)$, so löst man also die nichtlineare Gleichung $y(b,s) = \beta$. Dazu kann man irgendeine der Methoden aus Teil 3 verwenden. Die Berechnung von $y(b,s)$ erfolgt dabei durch irgendein Verfahren zur Lösung der Anfangwertaufgabe.

Beispiel: $y'' = \frac{3}{2}y^2$, $y(0) = 4$, $y(1) = 1$. Wir schreiben die Anfangswertaufgabe 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. §3.1) 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 (9.5) mit der analytischen Methode zu vergleichen. Für die einfache Gleichung $y'' = f(y)$ findet man der Reihe nach formal

$$\begin{aligned} y'y'' &= y'f(y) \qquad , \\ \tfrac{1}{2}\tfrac{d}{dx}y'^2 &= \tfrac{d}{dx}F(y) \qquad , \quad F(y) = \int^y f(z)dz \quad , \\ \tfrac{1}{2}y'^2 &= F(y) + c \qquad , \\ y' &= \pm\sqrt{2F(y)+c} \quad , \\ \frac{dy}{\pm\sqrt{2F(y)+c}} &= dx \qquad , \\ \pm \int_\alpha^{y(x)} \frac{dz}{\sqrt{2F(z)+c}} &= x - a \qquad . \end{aligned}$$

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.

Kapitel 8

Partielle Differentialgleichungen

8.1 Partielle Differentialgleichungen 1. Ordnung

Die Lösung partieller Differentialgleichungen 1. Ordnung kann man auf das Lösen von Systemen gewöhnlicher Differentialgleichungen zurückführen.

Wir zeigen dies zunächst für die lineare Gleichung

$$\sum_{i=1}^{n} a_i(x)\frac{\partial u}{\partial x_i} = b(x)u + c(x) \; . \tag{1.1}$$

Suche Kurve $x = x(t)$ auf Lösungsfläche $u = u(x)$.
Ansatz:

$$\dot{x} = a(x) \tag{1.2}$$

Dieses System gewöhnlicher Differentialgleichungen nennt man das charakteristische System von (1.1), seine Lösungen Charakteristiken.

Sei u Lösung. Dann gilt für $x = x(t)$

$$\frac{d}{dt}u(x) = \dot{x} \cdot \nabla u(x) = a \cdot \nabla u(x) = b(x)u + c(x) \; .$$

Kurven der gesuchten Art findet man also durch Lösen von $\dot{u} = b(x)u + c(x)$.

Sei $\Gamma : x = \xi(s)$, $s \in \mathbf{R}^{n-1}$ eine Anfangsfläche, entlang der u vorgegeben ist: $u(\xi(s)) = \mu(s)$. Wir lösen (1.2) mit den Anfangswerten $x(0, s) = \xi(s)$. Wir nehmen an, daß $x(t, s) = x$ nach x auflösbar ist:

$$t = t(x) \quad , \quad s = s(x) \; .$$

Dies ist in einer Umgebung von Γ der Fall, wenn

$$\det(x_t, x_s) = \det(a(\xi(s)), \xi_s(s)) \neq 0 \; , \qquad s \in \mathbf{R}^{n-1} \; .$$

Eine Fläche Γ mit dieser Eigenschaft nennt man nicht-charakteristisch.
Wir lösen dann $\dot{u} = b(x)u + c(x)$, $u(0) = \mu(s)$ und setzen

$$U(x) = u(t(x), s(x)) \; .$$

Es ist dann

$$U(x(t,s)) = u(t,s) \; .$$

Differentiation nach t ergibt

$$\nabla U(x(t,s)) \cdot \dot{x}(t,s) = \dot{u}(t,s)$$

oder

$$a(x(t,s)) \cdot \nabla U(x(t,s)) = b(x(t,s)) \; .$$

Also ist U Lösung der Anfangswertaufgabe in einer Umgebung von Γ.

Ganz ähnlich geht man vor bei der quasilinearen Gleichung

$$\sum_{i=1}^{n} a_i(x,u) \frac{\partial u}{\partial x_i} = b(x,u) \; . \tag{1.3}$$

Für Kurven auf der Lösungsfläche machen wir wieder den Ansatz

$$\dot{x} = a(x, u(x))$$

mit der - noch unbekannten - Lösung $u = u(x)$. Liegt $x = x(t)$ wirklich auf $u = u(x)$, so gilt für $x = x(t)$

$$\frac{d}{dt} u(x) = \dot{x} \cdot \nabla u(x) = a(x, u(x)) \cdot \nabla u(x) = b(x,u) \; .$$

Kurven der genannten Art findet man also durch Lösen von

$$\dot{x} = a(x,u) \; , \qquad \dot{u} = b(x,u) \; . \tag{1.4}$$

Dieses System heißt nun das charakteristische System zu (1.3), seine Lösungen Charakteristiken. Man beachte, daß die Charakteristiken nun Kurven im $\mathbf{R}^{n+1}$ sind.

Sei wieder $\Gamma : x = \xi(s)$ eine Anfangsfläche im $\mathbf{R}^n$, entlang der u vorgegeben ist: $u(\xi(s)) = \mu(s)$. Wir lösen (1.4) mit den Anfangswerten $x(0,s) = \xi(s)$, $u(0,s) = \mu(s)$. Wir nehmen an, daß $x(t,s)$ nach x auflösbar ist:

$$t = t(x) \; , \qquad s = s(x) \; .$$

Dies ist in einer Umgebung von Γ der Fall, wenn

$$\det(x_t, x_s) = \det(a(\xi(s), \mu(s)) \; , \; \xi_s(s)) \neq 0 \; . \tag{1.5}$$

Man beachte, daß dies nun eine Bedingung an die Fläche $s \to (\xi(s), \mu(s))$ in $\mathbf{R}^{n+1}$ ist. Man nennt eine Fläche mit (1.5) nicht-charakteristisch.

Wir setzen wieder

$$U(x) = u(t(x), s(x)) \ ,$$

also $U(x(t,s)) = u(t,s)$. Differenzieren nach t ergibt

$$\nabla U(x(t,s)) \cdot \dot{x}(t,s) = \dot{u}(t,s)$$

oder

$$a(x(t,s), u(t,s)) \cdot \nabla U(x(t,s)) = b(x(t,s) \ , \ u(t,s)) \ .$$

Also ist U Lösung der Anfangswertaufgabe.

Nun behandeln wir die allgemeine Gleichung

$$F(x,u,p) = 0 \ , \qquad p = \nabla u \ . \tag{1.6}$$

Wieder machen wir für eine Kurve auf der Lösungsfläche den Ansatz

$$\dot{x} = F_p(x,u,p) \ .$$

Liegt $x = x(t)$ auf der Lösungsfläche $u = u(x)$, so gilt für $x = x(t)$

$$\frac{d}{dt} u(x) = \nabla u(x) \cdot \dot{x} = p(x) \cdot F_p(x, u(x), p(x)) \ .$$

Differenzieren wir (1.6) nach t, so erhalten wir

$$\begin{aligned} 0 &= \frac{d}{dt} F(x, u(x), p(x)) \\ &= \dot{x} \cdot F_x + \left(\frac{d}{dt} u(x) \right) \cdot F_u + \left(\frac{d}{dt} p(x) \right) \cdot F_p \\ &= F_p \cdot F_x + (p(x) \cdot F_p) F_u + \left(\frac{d}{dt} p(x) \right) \cdot F_p \\ &= F_p \cdot \left(F_x + p(x) F_u + \frac{d}{dt} p(x) \right) \ ; \end{aligned}$$

die Argumente sind hier überall x, $u(x)$, $p(x)$. Dies ist erfüllt, wenn

$$\frac{d}{dt} p(x) = -F_x - p(x) F_u \ .$$

Damit haben wir das System

$$\dot{x} = F_p \quad , \quad \dot{u} = p \cdot F_p \quad , \quad \dot{p} = -F_x - p F_u \tag{1.7}$$

gefunden. Es heißt charakteristisches System von (1.6), seine Lösungen Charakteristiken. Dies sind nun Kurven im $\mathbf{R}^{2n+1}$.

Sei nun $\Gamma : x = \xi(s)$ wieder eine Anfangsmannigfaltigkeit und $u(\xi(s)) = \mu(s)$ vorgeschrieben. Wir wollen die Lösung der Anfangswertaufgabe wieder aus

Lösungen von (1.7) zusammensetzen. Dazu benötigen wir aber Anfangswerte für p. Ist $u = u(x)$ mit $p = \nabla u$ überhaupt eine Fläche - nicht notwendig eine Lösung von (1.6) - welche entlang Γ diese Werte annimmt, so muß

$$\begin{aligned} \frac{\partial \mu}{\partial s_k}(s) &= \sum_{i=1}^{n} \frac{\partial u}{\partial x_i}(\xi(s)) \frac{\partial \xi_i}{\partial s_k}(s) \\ &= p(\xi(s)) \cdot \frac{\partial \xi}{\partial s_k}(s) \,, \qquad k = 1, \ldots, n-1 \end{aligned}$$

gelten. Zu (1.6) tritt also noch

$$p \cdot \frac{\partial \xi}{\partial s_k} = \frac{\partial \mu}{\partial s_k} \,, \qquad k = 1, \ldots, n-1 \tag{1.8}$$

hinzu. (1.6), (1.8) bilden ein nichtlineares System von n Gleichungen für die n Unbekannten p. Es ist - jedenfalls lokal - nach p auflösbar, wenn

$$\det \left(F_p(\xi, \mu, p) \,, \frac{\partial \xi}{\partial s_1}, \ldots, \frac{\partial \xi}{\partial s_{n-1}} \right) \neq 0 \,. \tag{1.9}$$

Wir nennen die Anfangsmannigfaltigkeit $s \to (\xi(s), \mu(s), \psi(s))$ nicht charakteristisch, wenn (1.9) erfüllt ist für $p = \psi(s)$.

Nun gehen wir vor wie oben. Wir lösen (1.7) mit den Anfangswerten $x(0, s) = \xi(s)$, $u(0, s) = \mu(s)$, $p(0, s) = \psi(s)$. Unter der Bedingung (1.9) ist $x(t, s) = x$ nach t, s auflösbar, denn es ist

$$\det(x_t, x_s) = \det \left(F_p, \frac{\partial \xi}{\partial s_1}, \ldots, \frac{\partial \xi}{\partial s_{n-1}} \right) \neq 0 \,.$$

Mit $t = t(x)$, $s = s(x)$ setzen wir nun

$$U(x) = u(t(x), s(x)) \,.$$

Wieder kann man zeigen, daß U Lösung ist, aber nicht ganz so einfach wie oben.

Das wichtigste Beispiel einer allgemeinen Differentialgleichung 1. Ordnung ist die Eikonal-Gleichung

$$|p| = n(x) \,. \tag{1.10}$$

Das charakteristische System ist

$$\dot{x} = p/|p| \quad , \quad \dot{u} = |p| \quad , \quad \dot{p} = \nabla n \,. \tag{1.11}$$

Unter Benutzung von (1.10) können wir dafür auch

$$\dot{x} = p/n \quad , \quad \dot{u} = n \quad , \quad \dot{p} = \nabla n \tag{1.12}$$

schreiben. Man zeigt übrigens sofort, daß für jede Lösung von (1.11) $|p| = n$ ist bis auf eine additive Konstante.

Wir wollen nun das Anfangswertproblem

$$u(x) = \mu(s) \quad , \quad x = \xi(s) \quad , \quad s \in \mathbf{R}^{n-1}$$

lösen. Die Anfangswerte $\psi(s)$ für p ergeben sich aus

$$|p| = n \quad , \quad p^* \xi' = \mu' \; , \tag{1.13}$$

wobei

$$\xi' = \left(\frac{\partial \xi}{\partial s_1}, \ldots, \frac{\partial \xi}{\partial s_{n-1}} \right) \; , \quad \mu' = \left(\frac{\partial \mu}{\partial s_1}, \ldots, \frac{\partial \mu}{\partial s_{n-1}} \right) \; .$$

Die Lösung p_0 kleinster Norm des linearen unterbestimmten Systems ist

$$p_0 = \xi' (\xi'^* \xi')^{-1} \mu'^* \; ,$$

und es ist

$$|p_0|^2 = \mu' (\xi'^* \xi')^{-1} \mu'^* \; .$$

Das nichtlineare System (1.13) ist genau dann nach p auflösbar, wenn $|p_0| \geq n$, also

$$\mu' (\xi'^* \xi')^{-1} \mu'^* \geq n^2 \; . \tag{1.14}$$

In der Ebene bedeutet das

$$\frac{|\partial \mu / \partial s|}{|\partial \xi / \partial s|} \geq n \; .$$

Dies ist physikalisch leicht zu verstehen. Die Ausbreitungsgeschwindigkeit $|\partial \xi / \partial s| / |\partial \mu / \partial s|$ auf der Anfangsmannigfaltigkeit kann nicht größer sein als die lokale Ausbreitungsgeschwindigkeit $c = 1/n$.

Unter der strengen Bedingung (1.14) gibt es genau zwei Anfangswerte $\psi_\pm(s)$ für p. Die Bedingung (1.9) lautet

$$\det \left(\psi_\pm, \frac{\partial \xi}{\partial s_1}, \ldots, \frac{\partial \xi}{\partial s_{n-1}} \right) \neq 0 \; .$$

Sie ist unter der strengen Bedingung (1.14) immer erfüllt. Wäre sie verletzt, so wäre nämlich

$$\psi_\pm = \xi' \lambda \quad , \quad \lambda \in \mathbf{R}^{n-1} \; ,$$

und

$$\psi_\pm^* \xi' = \mu' \; .$$

Man könnte $\psi_\pm$ eliminieren und erhielte wegen (1.14)

$$\begin{aligned} \lambda &= (\xi'^* \xi')^{-1} \mu'^* \; , \\ \mu' \lambda &= \mu' (\xi'^* \xi')^{-1} \mu'^* < n^2 \; . \end{aligned}$$

Dies stünde im Widerspruch zu

$$\mu' \lambda = \psi_\pm'^* \xi' \lambda = \psi_\pm^* \psi_\pm = n^2 \; .$$

Insgesamt haben wir also die Lösbarkeit der Anfangswertaufgabe unter (1.14) gezeigt. Für die Lösung erhält man durch Integration der zweiten charakteristischen Gleichung

$$u(x(t,s)) = \mu(s) + \int_0^t n(x(t',s))dt' \ .$$

8.2 Lineare Differentialgleichung 2. Ordnung

$$\sum_{i,j=1}^{n} a_{ij}(x)\frac{\partial^2 u}{\partial x_i \partial x_j} + \sum_{i=1}^{n} a_i(x)\frac{\partial u}{\partial x_i} + a(x)u = f(x)$$

Beispiele:

1) Poisson'sche Differentialgleichung:

$$-\Delta u = -\sum_{i=1}^{n} \frac{\partial^2 u}{\partial x_i^2} = f$$

Typisch sind Randwertaufgaben, etwa

$$\begin{aligned} -\Delta u &= f \quad \text{in} \quad \Omega \subseteq \mathbb{R}^n \\ u &= g \quad \text{auf} \quad \partial\Omega \quad \text{(Dirichlet)} \\ \frac{\partial u}{\partial v} &= g \quad \text{auf} \quad \partial\Omega \quad \text{(Neumann)} \\ \frac{\partial u}{\partial v} + \sigma u &= g \quad \text{auf} \quad \partial\Omega \quad \text{(gemischt)} \end{aligned}$$

2) Wärmeleitungsgleichung:

$$\frac{\partial u}{\partial t} = \Delta u$$

Typisch sind Anfangswertaufgaben

$$\begin{aligned} \frac{\partial u}{\partial t} &= \Delta u \quad \text{in} \quad \Omega \times [0,T] \\ u &= g \quad \text{auf} \quad \partial\Omega \\ u(x,0) &= u_0(x) \quad \text{in} \quad \Omega \end{aligned}$$

3) Wellengleichung:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \Delta u$$

Typisch sind wieder Anfangswertaufgaben

$$\begin{aligned} \frac{\partial^2 u}{\partial t^2} &= c^2 \Delta u \quad \text{in} \quad \Omega \times [0,T]\,, \\ u &= g \quad \text{auf} \quad \partial\Omega \\ u(x,0) &= u_0(x)\,, \quad \frac{\partial u}{\partial t}(x,0) = u_1(x) \quad \text{in} \quad \Omega\,. \end{aligned}$$

8.3 Einfachste Differenzenverfahren

Wir beginnen mit der Anfangswertaufgabe der Wärmeleitungsgleichung.

$$\begin{aligned} &u_t = u_{xx} \quad , \qquad 0 \le x \le 1 \\ &u(x,0) = u_0(x) \ , \\ &u(t,0) = u(t,1) = 0 \ , \quad t \ge 0 \ . \end{aligned}$$

Wir führen ein Gitter

$$t_\ell = \ell \Delta t \quad , \quad \ell = 0,1,\dots \ , \ x_k = kh \ , \ k = 0,\dots,n \ , \ h = \frac{1}{n}$$

ein und suchen für $u(x_k, t_\ell)$ eine Näherung $u_{k,\ell}$, welche die der Differentialgleichung analoge Differenzengleichung

$$\frac{1}{\Delta t}(u_{k,\ell+1} - u_{k,\ell}) = \frac{1}{h^2}(u_{k+1,\ell} - 2u_{k,\ell} + u_{k-1,\ell}) \ ,$$

$$k = 1,\dots,n-1 \quad , \quad \ell = 0,1,\dots$$

erfüllt. Dazu kommen noch die Anfangs- und Randbedingungen

$$\begin{aligned} u_{k,0} &= u_0(x_k) \quad , \ k = 0,\dots,n \\ u_{0,\ell} &= u_{n,\ell} = 0 \quad , \ \ell = 1,2,\dots \ . \end{aligned}$$

Die Differenzengleichungen können nach $u_{k,\ell+1}$ aufgelöst werden. Mit $\lambda = \Delta t / h^2$ gilt

$$u_{k,\ell+1} = \lambda(u_{k+1,\ell} + u_{k-1,\ell}) + (1 - 2\lambda)u_{k,\ell} \ .$$

Sind also die Werte für die Zeit t_ℓ bekannt, so kann man sie für die Zeit $t_{\ell+1}$ berechnen. Für t_0 sind sie durch die Anfangsbedingungen gegeben.

Als Beispiel führen wir die Rechnung durch für die Anfangswerte

$$u_{k,0} = \begin{cases} 1 \ , & k = K \ , \quad K+1 \\ 0 \ , & \text{sonst} \quad . \end{cases}$$

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 λ abhängt. Die Rechnung muß also die zeitliche Entwicklung eines solchen "hot spot" zeigen.

Als weiteres Beispiel betrachten wir das Anfangswertproblem der Wellengleichung

$$\begin{aligned} u_{tt} &= u_{xx} \quad , \qquad 0 \le x \le 1 \\ u(x,0) &= u_0(x) \, , \\ u_t(x,1) &= u_1(x) \, , \\ u(0,t) &= u(1,t) = 0 \ . \end{aligned}$$

Die Differentialgleichung wird im Punkt (x_k, t_ℓ) 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 $O((\Delta t)^2 + (\Delta x)^2)$. Um diese Fehlerordnung auch bei der Diskretisierung von $u_t = u_1$ zu haben, führt man ein Zeitniveau t_{-1} ein und kann dann

$$\begin{aligned} \frac{1}{2\Delta t}(u_{k,1} - u_{k,-1}) &= u_1(x_k) \quad , \\ u_{k,0} &= u_0(x_k) \end{aligned}$$

setzen. Die Differenzengleichung wird dann für $\ell = 0, 1, \ldots$ benutzt. Man kann sie nach $u_{k,\ell+1}$ auflösen und erhält

$$u_{k,\ell+1} = (u_{k+1,\ell} + u_{k-1,\ell}) + 2(1-\lambda)u_{k,\ell} - u_{k,\ell-1} \, .$$

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 die Anfangswertaufgabe

$$\begin{aligned} u_t &= u_x \quad , \quad x \in \mathbb{R}^1 \\ u(x,0) &= u_0(x) \ . \end{aligned}$$

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) $\frac{1}{\Delta t}(u_{k,\ell+1} - u_{k,\ell}) = \frac{1}{2h}(u_{k+1,\ell} - u_{k-1,\ell})$

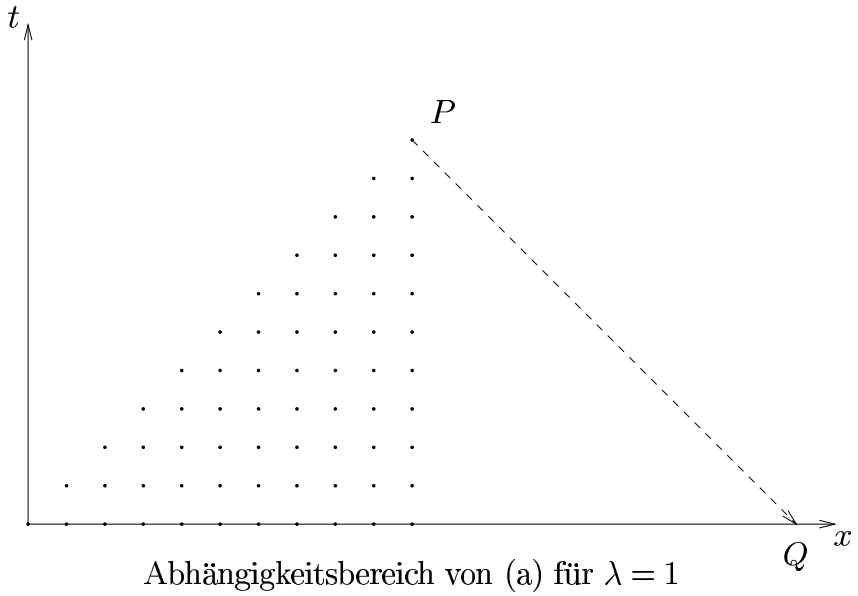
Auflösen nach $u_{k,\ell+1}$ ergibt mit $\lambda = \Delta t/\Delta x$

(a) $u_{k,\ell+1} = (1+\lambda)u_{k,\ell} - \lambda u_{k-1,\ell}$

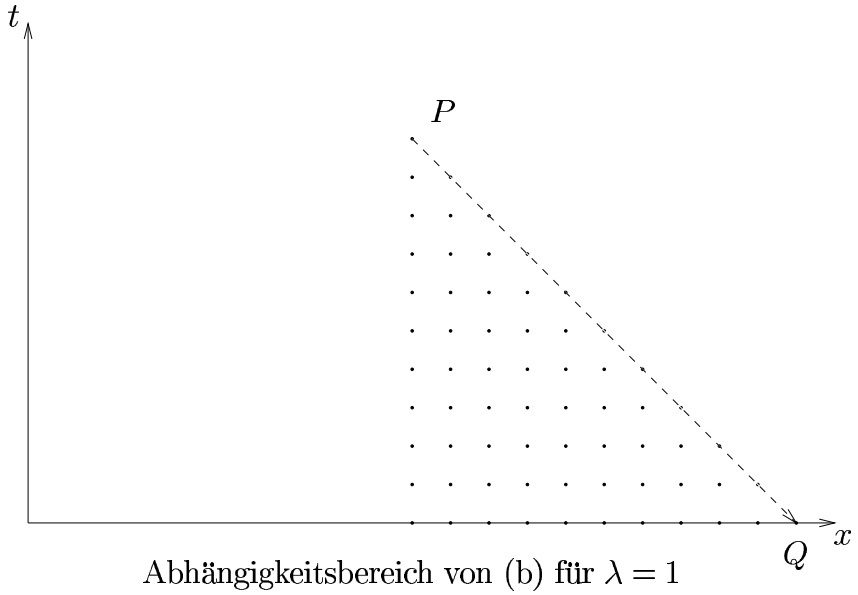
(b) $u_{k,\ell+1} = (1-\lambda)u_{k,\ell} + \lambda u_{k+1,\ell}$

(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.



Abhängigkeitsbereich von (a) für $\lambda = 1$



Abhängigkeitsbereich von (b) für $\lambda = 1$

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 $\lambda \leq 1$ gehört aber Q zum Abhängigkeitsbereich von (b).

Wir kommen so zu einer weiteren notwendigen Bedingung, der Courant - Friedrichs - Lewy - Bedingung (CFL):

> Das Abhängigkeitsgebiet für die Differentialgleichung
> muß im Abhängigkeitsgebiet des Differenzenverfahrens
> enthalten sein.

8.4 Stabilität

Wir gehen aus von der Anfangwertaufgabe

$$u_t = Au \quad , \quad u(x,0) = u_0(x) \quad , \quad a \le x \le b \, .$$

A sei ein Differentialoperator mit konstanten Koeffizienten, d.h. A_ρ hängt nicht von x ab. Dazu kommen unter Umständen noch Randbedingungen bei $x = a, b$.

Auf dem Gitter (t_ℓ, x_k) betrachten wir das Differenzenverfahren

$$\frac{1}{\Delta t}(u_{k,\ell+1} - u_{k,\ell}) = \sum_\nu B_\nu(h) u_{k+\nu,\ell} \ .$$

Wir führen die Vektoren und Matrizen

$$U_\ell = \begin{pmatrix} u_{0,\ell} \\ \vdots \\ u_{n,\ell} \end{pmatrix} \ , \ C(\Delta t) = I + \Delta t \begin{pmatrix} B_0 & B_1 & \cdots & \\ B_{-1} & B_0 & B_1 & \cdots \\ & & \cdots & \\ & & B_{-1} & B_0 \end{pmatrix} (h) \ , \quad h = g(\Delta t)$$

ein. Es entsteht

$$U_{\ell+1} = C(\Delta t) U_\ell \quad .$$

Definition 8.4.1 *Das Verfahren heißt stabil, wenn es für alle $T > 0$ eine Konstante $M(T)$ gibt, so daß für $\ell \Delta t \le T$*

$$\|(C(\Delta t))^\ell\|_\infty \le M(T) \ .$$

Beispiele:

1) Für das einfachste Differenzenverfahren bei der Wärmeleitungsgleichung ist

$$C(\Delta t) = \begin{pmatrix} 1-2\lambda & & & & \\ \lambda & 1-2\lambda & \lambda & & \\ & \ddots & & & \\ & & \lambda & 1-2\lambda \end{pmatrix} \ , \quad \|C(\Delta t)\|_\infty = |1-2\lambda| + 2\lambda \, .$$

Das Verfahren ist also jedenfalls für $\lambda \le 1/2$ stabil.

2) Für die Verfahren (a), (b), (c) für $u_t = u_x$ gilt der Reihe nach

$$\|C(\Delta t)\|_\infty = |1+\lambda| + |\lambda| \, , \quad |1-\lambda| + |\lambda| \, , \quad 1 + |\lambda| \, .$$

Danach sind (a), (c) immer instabil, (b) stabil für $\lambda \leq 1$.

Die Instabilitätsaussagen dieser Tabelle werden wir nach Satz 1 bestätigen.

Ein leistungsfähiges, wenngleich im allgemeinen nur notwendiges Stabilitätskriterium erhalten wir durch Fourier-Analyse. Setzen wir für reelles m

$$u_{k,\ell} = e^{ihkm} c_\ell \quad ,$$

so wird

$$\begin{aligned} u_{k,\ell+1} &= u_{k,\ell} + \Delta t \sum_\nu B_\nu(h) u_{k+\nu,\ell} \\ &= e^{ihkm} \left(I + \Delta t \sum_\nu B_\nu(h) e^{ih\nu m} \right) c_\ell \\ &= e^{ihkm} c_{\ell+1} \ , \quad c_{\ell+1} = G(\Delta t, m) c_\ell \ . \end{aligned}$$

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:

$$\begin{aligned} u_{k,\ell+1} &= (\lambda(e^{i(k+1)hm} + e^{-i(k+1)hm}) + (1-2\lambda)e^{ikhm}) c_\ell \\ &= (\lambda(e^{ihm} + e^{-ihm}) + 1 - 2\lambda) e^{ikhm} c_\ell \ , \\ c_{\ell+1} &= (2\lambda(\cos(hm) - 1) + 1) c_\ell \ . \end{aligned}$$

Also ist $G(\Delta t, m) = 2\lambda(\cos hm - 1) + 1$.

2) In Beispiel 3) aus §2 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

$$\begin{aligned} \frac{1}{\Delta t}(v_{k,\ell+1} - v_{k,\ell}) &= \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \frac{1}{h}(v_{k+1,\ell} - v_{k,\ell}) + \\ & \quad \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \frac{1}{h}(v_{k,\ell+1} - v_{k-1,\ell+1}) \ . \end{aligned} \tag{4.1}$$

Sei $u_{k,\ell}$ nach dem einfachsten Differenzenverfahren aus §3 berechnet, und sei

$$v^1_{k,\ell} = \frac{1}{\Delta t}(u_{k,\ell} - u_{k,\ell-1}) \ , \quad v^2_{k,\ell} = \frac{1}{h}(u_{k,\ell} - u_{k-1,\ell}) \ .$$

Dann ist

$$\begin{aligned}
\frac{1}{\Delta t}(v^1_{k,\ell+1} - v_{k,\ell}) &= \frac{1}{(\Delta t)^2}(u_{k,\ell+1} - 2u_{k,\ell} + u_{k,\ell-1}) \\
&= \frac{1}{h^2}(u_{k+1,\ell} - 2u_{k,\ell} + u_{k-1,\ell}) \\
&= \frac{1}{h}(v^2_{k+1,\ell} - v^2_{k,\ell}) \quad , \\
\frac{1}{\Delta t}(v^2_{k,\ell+1} - v^2_{k,\ell}) &= \frac{1}{h\Delta t}(u_{k,\ell+1} - u_{k-1,\ell+1} - u_{k,\ell} + u_{k-1,\ell}) \\
&= \frac{1}{h}(v^1_{k,\ell+1} - v^1_{k-1,\ell+1}) \qquad .
\end{aligned}$$

Dies bedeutet, daß $v_{k,\ell} = (v^1_{k,\ell}, v^2_{k,\ell})^T$ gerade (4.1) löst.
Mit $\lambda = \Delta t/\Delta x$ schreibt sich (4.1)

$$v_{k,\ell+1} = v_{k,\ell} + \lambda \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} (v_{k+1,\ell} - v_{k,\ell}) + \lambda \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} (v_{k,\ell+1} - v_{k-1,\ell+1}) \, .$$

Zur Berechnung der Amplifikationsmatrix setzen wir $v_{k,\ell} = e^{ihmk} c_\ell$. Es folgt

$$\begin{aligned}
e^{ihkm} c_{\ell+1} = e^{ihkm} c_\ell &+ \lambda \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} (e^{ihm} - 1) e^{ihkm} c_\ell \\
&+ \lambda \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} (1 - e^{-ihm}) e^{ihkm} c_{\ell+1} \quad .
\end{aligned}$$

Auflösen nach $c_{\ell+1}$ ergibt mit $a = \lambda(e^{ihm} - 1)$

$$c_{\ell+1} = \begin{pmatrix} 1 & 0 \\ \overline{a} & 1 \end{pmatrix}^{-1} \begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & a \\ -\overline{a} & 1 - |a|^2 \end{pmatrix} c_\ell$$

Damit haben wir die Amplifikationsmatrix gefunden:

$$G(\Delta t, m) = \begin{pmatrix} 1 & a \\ -\overline{a} & 1 - |a|^2 \end{pmatrix} \quad .$$

Satz 8.4.1 *(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.

$$\|C^\ell(\Delta t)U\|_\infty \le M(T)\|U\|_\infty \quad , \qquad \Delta t \cdot \ell \le T \ .$$

Setzen wir nun

$$U = \begin{pmatrix} u_0 \\ \vdots \\ u_n \end{pmatrix} \quad , \quad u_k = e^{ihmk}c \quad ,$$

so wird

$$C^\ell(\Delta t)U = \begin{pmatrix} G^\ell(\Delta t, m)u_0 \\ \vdots \\ G^\ell(\Delta t, m)u_n \end{pmatrix} \quad .$$

Also muß für alle m

$$||G^\ell(\Delta t, m)||_\infty \le M(T) \quad , \quad \Delta t \cdot \ell \le T$$

sein. Sei $\rho(\Delta t, m) = \max|\mu_i(\Delta t, m)|$ der Spektralradius von $G(\Delta t, m)$. Es ist $\rho^\ell(\Delta t, m) \le \|G^\ell(\Delta t, m)\|_\infty$. Also gilt

$$\rho^\ell(\Delta t, m) \le ||G^\ell(\Delta t, m)||_\infty \le M(T) \quad , \quad \Delta t \cdot \ell \le T \ .$$

Es folgt für $\Delta t\ell = T$

$$\rho(\Delta t, m) \le M(T)^{1/\ell} = M(T)^{\Delta t/T} \le 1 + K\Delta t \ .$$

□

Beispiele:

1) Für das einfachste Differenzenverfahren der Wärmeleitungsgleichung haben wir

$$G(\Delta t, m) = 2\lambda(\cos hm - 1) + 1$$

gefunden. Für den einzigen Eigenwert $\mu_1 = G(\Delta t, m)$ ist daher $1 - 2\lambda \le \mu_1 \le 1$, 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 $\lambda \le \frac{1}{2}$ ist die v. Neumann - Bedingung erfüllt. Das sagt aber zunächst nichts, weil die v. Neumann - Bedingung ja nur notwendig für Stabilität ist.

2) Für das der Wellengleichung äquivalente System haben wir

$$G(\Delta t, m) = \begin{pmatrix} 1 & a \\ -\overline{a} & 1 - |a|^2 \end{pmatrix} \quad , \quad a = \lambda(e^{ihm} - 1)$$

erhalten. Die Eigenwerte μ von $G(\Delta t, m)$ sind Lösungen von

$$\mu^2 + (\alpha - 2)\mu + 1 = 0 \ ,$$

$$\alpha = |a|^2 = \lambda^2((\cos hm - 1)^2 + (\sin hm)^2) = 2\lambda^2(1 - \cos hm) \ .$$

Also ist $0 \leq \alpha \leq 4\lambda^2$, und für beliebige h, m wird jeder Punkt dieses Intervalls erreicht. Die Lösungen $\mu_{1,2}$ der qudratischen Gleichung sind

$$\mu_{1,2} = 1 - \frac{\alpha}{2} \pm \sqrt{\alpha(\frac{\alpha}{4} - 1)} \ .$$

Für $\alpha > 4$ ist $\mu_2 < 1 - \frac{\alpha}{2} < -1$, und $|\mu_2| \leq 1 + K\Delta t$ ist nicht möglich. Die v. Neumann'sche Stabilitätsbedingung ist also für $\lambda > 1$ nicht erfüllt, das Verfahren also instabil. Für $\alpha \leq 4$ sind μ_1, μ_2 konjugiert komplex, und wegen $\mu_1\mu_2 = 1$ muß $|\mu_1| = |\mu_2| = 1$ sein. Also ist für $\lambda \leq 1$ 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

$$G(\Delta t, m) = 1 + \lambda - \lambda e^{-ihm} \quad \text{bzw.} \quad 1 - \lambda + \lambda e^{ihm} \ ,$$

also

$$\begin{aligned} |\mu_1|^2 &= (1 + \lambda(1 - \cos hm))^2 + \lambda^2(\sin hm)^2 \quad \text{bzw.} \\ |\mu_1|^2 &= (1 + \lambda(\cos hm - 1))^2 + \lambda^2(\sin hm)^2 \quad . \end{aligned}$$

Im ersten Fall sieht man sofort, daß die v. Neumann - Bedingung für kein $\lambda > 0$ erfüllt ist. Verfahren (a) ist also instabil für $\lambda > 0$. Für Verfahren (b) ist

$$|\mu_1|^2 = 1 + 2\lambda(\lambda - 1)(1 - \cos hm) \leq 1$$

für $0 < \lambda \leq 1$. Also ist für diese λ die v. Neumann - Bedingung erfüllt. Für (c) ist $G(\Delta t, m) = 1 + i\lambda \sin h\, m$, also

$$|\mu_1|^2 = 1 + \lambda^2(\sin h\, m)^2 \ .$$

Die v. Neumann-Bedingung ist also nie erfüllt und das Verfahren instabil.

Literatur über Numerik

1 Neuere Lehrbücher

- **Deuflhard - Hohmann**: Numerische Mathematik. Eine algorithmisch orientierte Einführung. Walter de Gruyter 1991.
- **Golub - Ortega**: Scientific Computing. Teubner 1995.
- **Hämmerlin - Hoffmann**: Numerische Mathematik. Springer 1989.
- **Opfer**: Numerische Mathematik für Anfänger. Vieweg, 3. Auflage 2001.
- **Plato**: Numerische Mathematik kompakt. Vieweg 2000.
- **Schaback - Werner, H.**: Numerische Mathematik. Springer 1992.
- **Schwarz**: Numerische Mathematik. Teubner 1988.
- **Shampine - Allen - Pruess**: Fundamentals of Numerical Computing. Wiley 1997.
- **Stoer**: Numerische Mathematik I. Springer 1989.
- **Stoer - Bulirsch**: Numerische Mathematik II. Springer 1990.
- **Stummel - Hainer**: Praktische Mathematik. Teubner 1982.
- **Werner, J.**: Numerische Mathematik 1 + 2, Vieweg 1992.

2 Ältere Lehrbücher

- **Acton**: Numerical Methods that Work. Harper 1970.
- **Björck - Dahlquist**: Numerische Methoden. Oldenbourg 1972.
- **Conte - de Boor**: Elementary Numerical Analysis. McGraw-Hill 1965, 1972.
- **Mennicken - Wagenführer**: Numerische Mathematik 1, 2, 3. Vieweg 1977.
- **Schmeisser - Schirmeier**: Praktische Mathematik. W. de Gruyter 1976.
- **Fröberg**: Introduction to Numerical Analysis. Addison-Wesley 1965.
- **Hamming**: Numerical Methods for Scientists and Engineers. McGraw-Hill, New York 1962.
- **Henrici**: Elements of Numerical Analysis.
- **Henrici**: Elements of Numerical Analysis. John Wiley & Sons, Inc., New York 1964.
- **Hildebrand**: Introduction to Numerical Analysis. McGraw-Hill, New York 1956.

- **Stiefel**: Einführung in die Numerische Mathematik. Teubner 1963.
- **Willers**: Methoden der praktischen Analysis. W. de Gruyter 1957.

3 Monographien

- **Beresin - Shidkow**: Numerische Methoden 1, 2. VEB, Deutscher Verlag der Wissenschaften, 1970.
- **Blum**: Numerical Analysis and Computation. Addison-Wesley & Sons, Inc., New York 1966.
- **Goldstine**: A History of Numerical Analysis. Springer 1977.
- **Hartree**: Numerical Analysis, 2d ed.. Oxford University Press, Fair Lawn, N.J. 1958.
- **Householder**: Principles of Numerical Analysis. McGraw-Hill, New York 1953.
- **Isaacson - Keller**: Analysis of Numerical Methods. John-Wiley & Sons, Inc., New York 1966.
- **Lanczos**: Applied Analysis. Prentice-Hall, Englewood Cliffs, N.J. 1956.
- **Marchuck**: Methods of Numerical Mathematics. Springer 1975.
- **Milne**: Numerical Calculus. Princeton University Press, Princeton, N.J. 1949.
- **Neumaier**: Introduction to Numerical Analysis. Cambridge 2001.
- **Ralston**: A First Course in Numerical Analysis. McGraw-Hill 1965.
- **Young - Gregory**: A Survey of Numerical Mathematics. Addison-Wesley 1972.

4 Programmbibliotheken

- **Press-Flannery-Teukolsky-Vetterling**, Numerical Recipes in **C** (auch FORTRAN, PASCAL erhältlich), Cambridge University Press).
- **IMSL-Bibliothek** (International Mathematical & Statistical Libraries, Inc., Houston).
- **NAG-Bibliothek** (The Numerical Analysis Group Ltd., Oxford).
- **MATLAB** (The MathWorks)