Vorlesungsskript
Numerische Mathematik I

F. Natterer

Institut für Numerische
und instrumentelle Mathematik

WS 1995/96, Di/Fr 11-13, M 4

Inhaltsverzeichnis

Kapitel 1

Lineare Gleichungssysteme

1.1 Das Eliminationsverfahren

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\begin{aligned}
a_{1,1}x_1 + a_{1,2}x_2 \quad + \cdots + \quad a_{1,n}x_n &= b_1\,,\\
a_{2,1}x_1 + a_{2,2}x_2 \quad + \cdots + \quad a_{2,n}x_n &= b_2\,,\\
\cdots \qquad & \\
a_{n,1}x_1 + a_{n,2}x_2 \quad + \cdots + \quad a_{n,n}x_n &= b_n\,.
\end{aligned} \tag{1.5}$$

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

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

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

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

$$\begin{array}{ccc} a_{2,2}^{(2)}x_2 & + \cdots + & a_{2,n}^{(2)}x_n = b_2^{(2)} \\ & \cdots & \\ a_{n,2}^{(2)}x_2 & + \cdots + & a_{n,n}^{(2)}x_n = b_n^{(2)} \end{array} \tag{1.7}$$

$$\begin{array}{rclcll} a_{i,k}^{(2)} & = & a_{i,k} - \ell_{i,1}a_{i,k} & , & i,k = 2, \ldots, n & , \\ b_i^{(2)} & = & b_i - \ell_{i,1}b_1 & , & i = 2, \ldots, n & , \\ \ell_{i,1} & = & a_{i,1}/a_{1,1} & , & i = 2, \ldots, n & . \end{array}$$

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

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

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

Wir wollen für das Eliminationsverfahren ein rekursives Programm schreiben:

```
elim (A, b, x, j, n)
/* Löst  Σ_{k=j}^{n} a_{i,k} x_k = b_i ,   i = j, ..., n */
{   Zeilenvertauschung () ;
    for   i = j + 1, ..., n
      {   ℓ = a_{i,j} / a_{j,j} ;
          for  k = j + 1, ..., n   a_{i,k} = a_{i,k} − ℓ * a_{j,k} ;
          b_i = b_i − ℓ * b_j ;
      }
    if   (j < n) elim (A, b, x, j + 1, n);
    x_j = b_j;
    for  k = j + 1, ..., n   x_j = x_j − a_{j,k} * x_k;
    x_j = x_j / a_{i,j};
}
```

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

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

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

Zeilenvertauschung ();

```
{  Suche i_0;
   for k = j, ..., n    swap (a_{i_0,k}, a_{j,k});
   swap (b_{i_0}, b_j);
}
```

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

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

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

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

$$1.2547 - 1.2534 = 0.0013 \ .$$

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

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

Die einzige Rechenoperation, die beim Eliminationsverfahren ausgeführt wird, ist

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

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

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

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

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

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

Spaltenpivotsuche ();

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

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

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

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

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

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

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

also

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

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

1.2 Die LR-Zerlegung

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

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

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

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

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

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

Zur Herstellung der LR-Zerlegung verwenden wir die Elementarmatrizen.

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

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 : $L_j A$ entsteht aus A durch Subtraktion des $\ell_{i,j}$-fachen der j-ten Zeile von der i-ten Zeile, $i = j+1, \dots, n$. Dies ist genau die Operation, die wir beim j-ten Eliminationsschritt durchgeführt haben.

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

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

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

Neben den Elementarmatrizen benötigen wir noch Permutationsmatrizen. Sei $\{i_1, \dots, i_n\}$ eine Permutation von $\{1, \dots, n\}$ 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 $\{i_1, \dots, i_n\}$ 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 $\{i_1, \dots, i_n\}$.

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

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. $i_1 = 1, \dots, i_j = j$. Dann gilt $PL_jP^* = L'_j$, wobei L'_j die gleiche Form wie L_j hat, aber mit (gemäß der Permutation) vertauschten Elementen $\ell'_{k,j} = \ell_{i_k,j},\ \ k > j$. Dies sieht man sofort, wenn man

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

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

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

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

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

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

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

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

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

Bemerkungen:

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

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

1.3 Fehlerabschätzung bei linearen Gleichungssystemen

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

Beispiel:

Löse $Ax = b$ mit

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

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

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

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

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

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

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

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

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

Beispiele:

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

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

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

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

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

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

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

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

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

Beweis: In jedem der angegebenen Lehrbücher.

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

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

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

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

Wir wollen nun spezielle Normen im Vektorraum der Matrizen definieren.

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

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

die zugeordnete Matrizennorm.

Bemerkung: Es gilt

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

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

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

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

3) $\|AB\| \leq \|A\|\|B\|$.
Beweis: Sei $B \neq 0$.

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

Beispiele:

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

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

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

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

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

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

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

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

c) T–Norm:

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

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

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

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

gilt.

Beweis: Sei

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

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

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

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

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

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

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

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

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

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

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

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

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

Beweis:

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

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

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

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

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

Für jedes $x \neq 0$ gilt damit

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

oder

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

Damit gilt:

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

zu b) Wir betrachten zunächst das Gleichungssystem

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

Durch Subtraktion erhalten wir

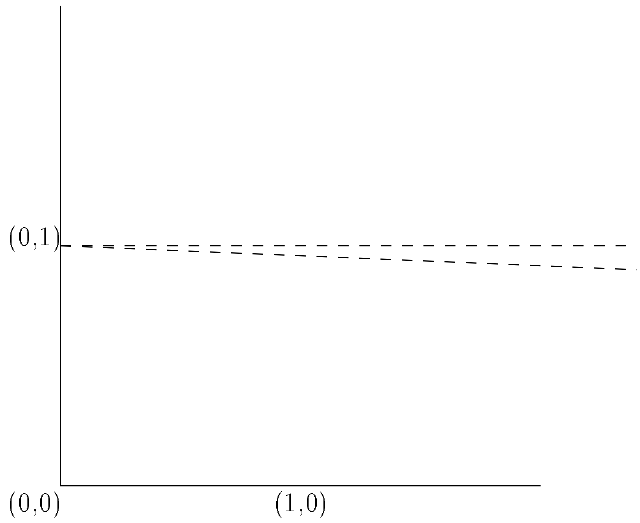
$$(A + \Delta A)\Delta x = b - \Delta A \cdot x$$

und damit
$$\Delta x = (A + \Delta A)^{-1}(b - \Delta A \cdot x).$$
Mit den Norm–Rechenregeln gilt:
$$\begin{aligned}
\frac{\|\Delta x\|}{\|x\|} &\leq \frac{1}{\|x\|}\|(A+\Delta A)^{-1}\|\|b - \Delta A \cdot x\| \\
&\leq \frac{\|A^{-1}\|}{1 - \|A^{-1}\|\|\Delta A\|}\left(\frac{\|\Delta b\|}{\|x\|} + \|\Delta A\|\right) \\
&= \frac{k(A)}{1 - k(A)\frac{\|\Delta A\|}{\|\Delta\|}}\left(\frac{\|\Delta b\|}{\|A\|\|x\|} + \frac{\|\Delta A\|}{\|A\|}\right) \\
&\leq \frac{k(A)}{1 - k(A)\frac{\|\Delta A\|}{\|A\|}}\left(\frac{\|\Delta b\|}{\|b\|} + \frac{\|\Delta A\|}{\|\Delta\|}\right)
\end{aligned}$$

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

Wir wollen nun unser einführendes Beispiel aufklären. Es galt:
$$A = \begin{pmatrix} 1 & 1 \\ 1 & 0.99 \end{pmatrix}, \quad A^{-1} = \begin{pmatrix} 99 & -100 \\ -100 & 100 \end{pmatrix}.$$
Wir erhalten:
$$k(A)_\infty = 2 \cdot 200 = 400 \,.$$
Wir müssen also damit rechnen, daß ein gegebener Anfangsfehler in A und b sich um einen Faktor 400 verstärkt in x auswirkt.

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



Grafische Darstellung des Modellproblems

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

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

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

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

Es ist dann

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

Die Anwendung des Satzes verlangt nun zunächst einmal

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

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

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

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

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

1.4 Die Cholesky-Zerlegung

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

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

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

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

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

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

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

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

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

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

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

Damit ist die erste Spalte von L bestimmt.

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

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

Für $j = 2$ ergibt sich

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

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

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

Damit ist auch die zweite Spalte von L bestimmt.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
{ for   j = 1, ..., n
           for   i = j + 1, ..., 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

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

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

1.5 Die QR-Zerlegung

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

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

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

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

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

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

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

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

Dann setzen wir

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

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

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

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

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

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

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

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

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

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

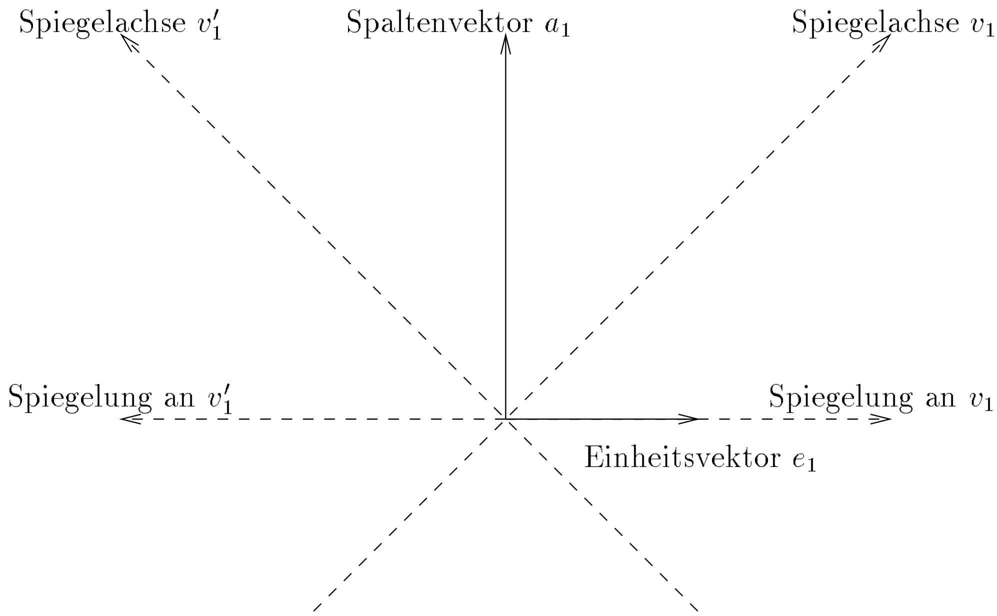
$$S = I - 2vv^* \quad , \quad \|v\| = 1$$

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

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

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

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



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

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

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

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

Die weiteren Spalten sind

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

$S_1 A$ hat die Gestalt

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

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

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

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

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

mit der rechten Dreiecksmatrix

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

Es folgt

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

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

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

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

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

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

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

$qr\_$ dump (A, α, n, m)

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

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

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

$q\_$ mal $x(A, x, n, m)$

```
/ ⋆   Überschreibt x mit Sx nach Aufruf von
      qr_ dump (A, α, n, m) ⋆ /
{     for j = m, …, 1
      { γ = 2 ⋆ (a_{j,j} ⋆ x_j + ⋯ + a_{n,j} ⋆ x_n);
        for i = j, …, n  x_i = x_i − γ ⋆ a_{i,j};
      }
}
q*_ mal_x(A, x, n, m)
/ ⋆   Überschreibt x mit S*x nach Aufruf von
      qr_ dump (A, α, n, m) ⋆ /
{       Wie oben, aber  j = 1, …, m}
```

1.6 Unter- und überbestimmte lineare Systeme

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

□

Bemerkungen:

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

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

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

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

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

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

1) x^+ ist Kleinste-Quadrate-Lösung.

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

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

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

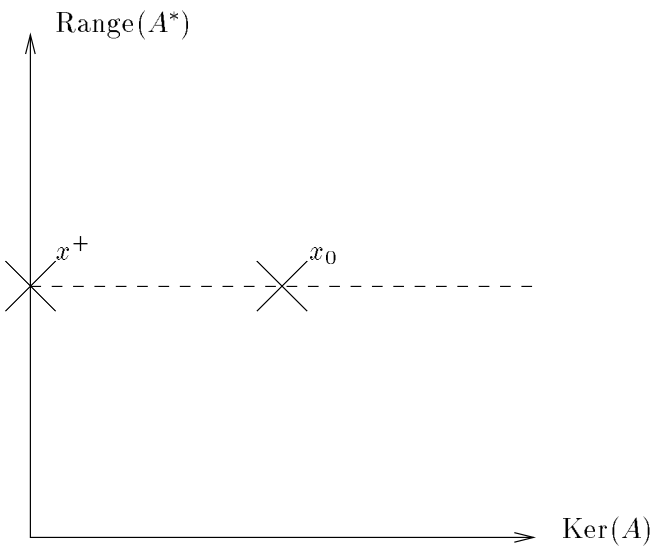
Beweis: Sei x_0 Kleinste-Quadrate-Lösung, also $A^*Ax_0 = A^*b$. Nach Satz 1 ist $x_0 = x_1 + x_2$ mit $x_1 \in$ range(A^*) , $x_2 \in$ ker(A). Auch x_1 erfüllt die Normalgleichungen und ist damit Kleinste-Quadrate-Lösung. Es gibt also immer eine Kleinste-Quadrate-Lösung x_1 in range (A^*). Jede weitere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

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

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

□

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



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

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

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

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

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

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

1) Wir betrachten das Beispiel

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

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

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

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

   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 2

Lineare Optimierung

2.1 Ein einführendes Beispiel

Wir wenden uns nun einem praxisorientiertes Gebiet der Numerischen Mathematik zu. In der Linearen Optimierung werden lineare Zielfunktionen unter linearen Bedingungen optimiert. Häufig ist dies die Gewinnmaximierung unter gegebenen Einschränkungen durch Produktionskapazität, Lagervorrat und gesetzliche Bestimmungen. Diese Anwendung macht die lineare Optimierung zu einem der wichtigsten mathematischen Methoden in der Betriebswirtschaft. Wir werden sie zunächst informell behandeln. Wir diskutieren das folgende Beispiel:

Ein Chemiekonzern stellt die Chemikalien C_1 und C_2 her. Bei der Produktion fallen die Schadstoffe S_1, S_2 und S_3 an (laut Tabelle). Der maximale Schadstoffausstoß pro Tag ist begrenzt. Erarbeiten Sie einen Produktionsplan, der einen maximalen Gewinn sicherstellt.

Tabelle 1:

	$C_1(^{mg}/\ell)$	$C_2(^{mg}/\ell)$	Höchstgrenze $(^{mg}/d)$
S_1	20	10	8000
S_2	4	5	2000
S_3	6	15	4500
Gewinn DM	16	32	

Wir formulieren die Optimierungsaufgabe wie folgt:

Seien c_1, c_2 die pro Tag von C_1, C_2 produzierten Mengen in Liter. Es muß gelten

$$\begin{array}{rcrcrcl} c_1 & \geq & 0 & & c_1 & \geq & 0 \\ c_2 & \geq & 0 & \text{oder} & c_2 & \geq & 0 \\ 20c_1 + 10c_2 & \leq & 8000 & & 8000 - 20c_1 - 10c_2 & \geq & 0 \\ 4c_1 + \ 5c_2 & \leq & 2000 & & 2000 - \ 4c_1 - \ 5c_2 & \geq & 0 \\ 6c_1 + 15c_2 & \leq & 4500 & & 4500 - \ 6c_1 - 15c_2 & \geq & 0 \end{array}$$

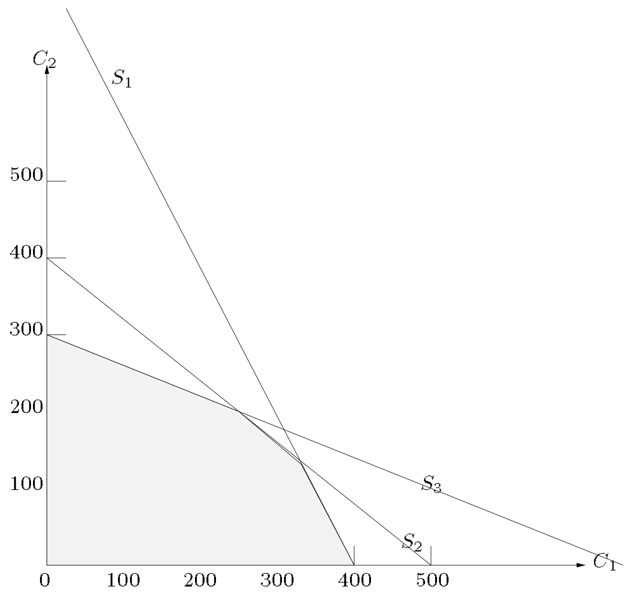
Mit der Definition

$$c = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} , \quad b = \begin{pmatrix} 8000 \\ 2000 \\ 4500 \end{pmatrix} , \quad A = \begin{pmatrix} -20 & -10 \\ -4 & -5 \\ -6 & -15 \end{pmatrix}$$

muß daher gelten

$$c \geq 0 \qquad \text{und} \qquad b + Ac \geq 0 \; .$$

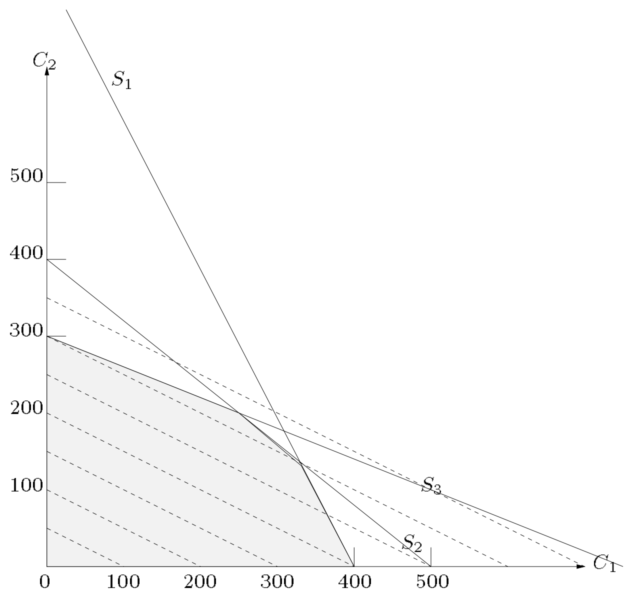
Wir zeichnen den durch diese Ungleichungen bestimmten Lösungsbereich L.



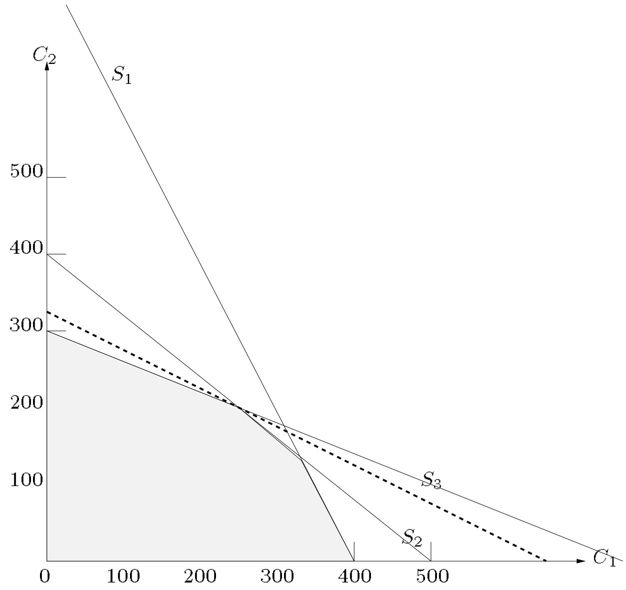
Es gilt: Gewinn (in DM) $= 16c_1 + 32c_2$. Es ist also genau dann möglich, einen Gewinn von g DM zu erzielen, wenn die durch $16c_1 + 32c_2 = g$ bestimmte Gerade $G(g)$ einen Schnittpunkt mit L besitzt.

Wir suchen also den größten Wert von g, so daß $G(g)$ einen Schnittpunkt mit L hat.

Im folgenden Bild ist $G(g)$ für einige Werte von g gezeichnet worden.



Für steigendes g verlagert sich die Linie nach oben. Die höchste Linie, die noch einen Schnittpunkt mit L hat, ist im folgenden gezeichnet:



Wir bestimmen den Schnittpunkt als Lösung des Gleichungssystems

$$\begin{aligned} 4c_1 + 5c_2 &= 2000 \\ 6c_1 + 15c_2 &= 4500 \end{aligned}$$

als $c = (250, 200)$.

Es ist anschaulich, daß es immer eine Ecke von L gibt, in der ein optimales Ergebnis erzielt wird. "Ecken" sind dabei Lösungen eines linearen Gleichungssystems in k Variablen, wenn eine Zielfunktion im $\mathbb{R}^k$ optimiert werden soll.

Wir können daher folgenden einfachen Algorithmus zur Lösung der linearen Optimierungsaufgabe angeben.

Für alle Schnittpunkte von k Ungleichungen:

> Stellen Sie fest, ob der Schnittpunkt zulässig ist (d.h., ob alle Ungleichungen erfüllt sind).
>
> Bestimme unter allen zulässigen Schnittpunkten das Zielfunktionsmaximum.

Zur Bestimmung der Schnittpunkte müssen bei n Ungleichungen $\binom{n}{k}$ Gleichungssysteme der Ordnung k gelöst werden. Da typischerweise n einige tausend, k einige hundert beträgt, ist der Algorithmus nicht durchführbar.

2.2 Die allgemeine lineare Optimierungsaufgabe

Wir wollen nun die allgemeine lineare Optimierungsaufgabe formulieren.

Sei A eine reelle (m,n)-Matrix und $c \in \mathbb{R}^n$. Sei

$$M = \Big\{x \in \mathbb{R}^n \;:\; x_j \geq 0\ ,\ j = 1,\ldots,n_0\ ,\ \sum_{j=1}^{n} a_{i,j}x_j \leq b_i\ ,\ \ i = 1,\ldots,m_0\ ,$$
$$\sum_{j=1}^{n} a_{i,j}x_j = b_i\ ,\ \ i = m_0+1,\ldots,m\Big\}\ . \tag{2.1}$$

Die allgemeine lineare Optimierungsaufgabe lautet nun: Minimiere

$$z(x) = \sum_{j=1}^{n}\ c_j x_j \tag{2.2}$$

in M! Man nennt M die Menge der zulässigen Punkte und jedes $x \in M$ zulässig. Die lineare Funktion (2.2) heißt Zielfunktion.

Die Aufgabe (2.1), (2.2) heißt in Normalform, wenn $n_0 = n$ und $m_0 = 0$. Dann kann man M in der Form

$$M = \{x \in \mathbb{R}^n : x \geq 0\ , \quad Ax = b\} \tag{2.3}$$

schreiben. Ungleichungen zwischen Vektoren werden hier und unten immer komponentenweise verstanden. Man kann (2.1) immer in der Form (2.3) schreiben. Ist $n_0 < n$, so setzt man

$$x_j = y_j - z_j \quad , \quad j = n_0+1,\ldots,n \quad , \quad y_j\ ,\ z_j \geq 0\ .$$

Ist $m_0 > 0$, treten also Ungleichungen auf, so führt man die "Schlupfvariablen"

$$u_i = b_i - \sum_{i=1}^{n}\ a_{ij}x_j \quad , \quad i = 1,\ldots,m_0$$

ein. Dann hat man ein Problem in Normalform mit den nicht negativen Variablen

$$\begin{array}{ll} x_j & ,\ \ j = 1,\ldots,n_0 \\ y_j, z_j & ,\ \ j = n_0+1,\ldots,n \\ u_i & ,\ \ i = 1,\ldots,m_0 \end{array}$$

und den Gleichungen

$$
\begin{aligned}
u_i + \sum_{j=1}^{n_0} a_{i,j} x_j \quad &+ \sum_{j=n_0+1}^{n} a_{i,j}(y_j - z_j) = b_i \ , \quad i = 1, \ldots, m_0 \\
\sum_{j=1}^{n_0} a_{i,j} x_j \quad &+ \sum_{j=n_0+1}^{n} a_{i,j}(y_j - z_j) = b_i \ , \quad i = m_0 + 1, \ldots, m \ .
\end{aligned}
$$

Nun führen wir einige geometrische Begriffe ein.

1) Eine Menge $U \subseteq \mathbb{R}^n$ heißt konvex, wenn mit $x, y \in U$ auch $\lambda x + (1-\lambda) y \in U$ für $0 \leq x \leq 1$, d.h. wenn mit je zwei Punkten deren Verbindungsgerade in U liegt. Z.B. sind die Mengen M aus (2.1), (2.3) konvex.

2) Sei $U \subseteq \mathbb{R}^n$ konvex. $x \in U$ heißt Ecke von U, wenn x nicht im Innern einer Verbindungsgeraden zweier Punkte von U liegt. Mit anderen Worten: Es gilt nicht $x = \lambda y + (1-\lambda) z$ mit $y, z \in U, \ y \neq z, \ 0 < \lambda < 1$.

3) Sind $u_i \in \mathbb{R}^n$ und $\lambda_i \geq 0, \ \sum\limits_i \lambda_i = 1$, so heißt $\sum\limits_i \lambda_i u_i$ Konvexkombination der u_i. Die Menge aller Konvexkombinationen heißt konvexe Hülle der u_i.

Von dem einleitenden Beispiel in §1 ist klar, daß die Ecken von M eine besondere Rolle spielen. Wir wollen diese Ecken algebraisch charakterisieren. Im folgenden nehmen wir immer an, daß A den Rang m hat, und für M nehmen wir immer die Normalform (2.3).

Definition 2.2.1 *Sei A eine reelle (m,n)-Matrix. Eine Lösung x von $Ax = b$ heißt Basis-Lösung, wenn es eine Teilmenge I von $\{1, \ldots, n\}$ mit genau m Elementen gibt, so daß gilt:*

i) $x_i = 0 \ , \ \ i \notin I$

ii) $a_i \ , \ i \in I$ *sind linear unabhängig.*

I heißt Basis von x.

Satz 2.2.1 *$x \in \mathbb{R}^n$ ist genau dann Ecke von M, wenn x zulässige Basis-Lösung von $Ax = b$ ist.*

Beweis:

a) Sei x Ecke von M. Sei I die Menge der Indizes i mit $x_i > 0$. Wären a_i, $i \in I$ linear abhängig, so gäbe es Zahlen α_i, $i \in I$ mit

$$\sum_{i \in I} \alpha_i a_i = 0 \; , \quad \sum_{i \in I} |\alpha_i| > 0 \; .$$

Wir könnten die Punkte $x^{\pm}$ mit den Komponenten

$$\begin{array}{rcll} x_i^{\pm} & = & x_i \pm \varepsilon \alpha_i & , \quad i \in I \; , \\ x_i^{\pm} & = & 0 & , \quad i \notin I \end{array}$$

bilden. Es wäre $Ax^{\pm} = b$ und $x^{\pm} \geq 0$ für hinreichend kleines ε, also $x^{\pm} \in M$. Außerdem wäre $x = \frac{1}{2}(x^+ + x^-)$. Dies ist nicht möglich, da x Ecke ist. Also sind die a_i, $i \in I$ linear unabhängig. Insbesondere hat I höchstens m Elemente. Hat I weniger als m Elemente, so können wir I zu einer m-elementigen Menge ergänzen. x ist damit zulässige Basis-Lösung mit Basis I.

b) Sei x umgekehrt zulässige Basis-Lösung mit Basis I. Wäre x nicht Ecke, so gäbe es x^1, $x^2 \in M$, $x^1 \neq x^2$, und λ mit $0 < \lambda < 1$, so daß $x = \lambda x^1 + (1 - \lambda) x^2$. Für $i \notin I$ müßte dann $x_i^1 = x_i^2 = 0$ sein, und wegen

$$\sum_{i \in I} x_i^1 a_i = \sum_{i \in I} x_i^2 a_i \; .$$

müßte $x_i^1 = x_i^2$ auch für $i \in I$ sein. Im Widerspruch zur Annahme wäre also $x^1 = x^2$.

Satz 2.2.2 *Sei M beschränkt. Dann ist M die konvexe Hülle seiner Ecken.*

Beweis: Es ist zu zeigen, daß jedes $x \in M$ Konvexkombination der Ecken von M ist.

Sei also $x \in M$ und p die Anzahl der positiven Komponenten von x. Wir führen den Beweis durch Induktion nach p.

Ist $p = 0$, so ist x zulässige Basis-Lösung (mit einer beliebigen Wahl von m Basis-Vektoren) und damit selbst Ecke. Sei die Behauptung richtig bis $p-1$. Ist x Ecke, so sind wir fertig. Andernfalls sind die p Vektoren a_i, $x_i > 0$ linear abhängig, d.h. es gibt $w \in \mathbb{R}^n$, $w \neq 0$ mit $Aw = 0$ und $w_i = 0$ für $x_i = 0$. Da M beschränkt ist, muß w sowohl positive als auch negative Komponenten haben. Also kann man t_1, $t_2 > 0$ so finden, daß

$$x^1 = x - t_1 w \quad , \quad x^2 = x + t_2 w$$

jeweils höchstens $p-1$ positive Komponenten haben und x^1, $x^2 \in M$. Nach Induktionsannahme sind x^1, x^2 Konvexkombinationen der Ecken von M, also auch

$$x = \frac{t_2}{t_1+t_2} x^1 + \frac{t_1}{t_1+t_2} x^2 \; .$$

□

Satz 2.2.3 *M sei beschränkt und nicht leer. Dann ist die lineare Optimierungsaufgabe lösbar, und es gibt sogar eine Ecke von M, die Lösung ist.*

Beweis: Die Zielfunktion z nimmt als stetige Funktion auf dem Kompaktum M ihr Minimum an, etwa in $x^0 \in M$. Seien $x^1, \ldots, x^k$ die Ecken von M. Nach Satz II.2.2 ist x Konvexkombination von $x^1, \ldots, x^k$, also

$$x^0 = \sum_{i=1}^k \lambda_i x^i \quad , \quad \lambda_i \geq 0 \quad , \quad \sum_{i=1}^k \lambda_i = 1 \; .$$

Wäre nun $z(x^0) < z(x^i)$ für alle i, so wäre

$$z(x^0) = \sum_{i=1}^k \lambda_i z(x^i) < \sum_{i=1}^k \lambda_i z(x^0) = z(x^0) \; ,$$

und dies ist ein Widerspruch. Also gibt es ein i mit $z(x^0) = z(x^i)$, und dieses x^i ist Lösung.

□

2.3 Das Simplex-Verfahren: Phase II

Das Simplex-Verfahren ist das Standard-Verfahren zur numerischen Lösung linearer Optimierungsaufgaben. Es wurde gegen Ende des zweiten Weltkriegs von Dantzig entwickelt und zur optimalen Ladungsverteilung von der amerikanischen Marine benutzt. Bei diesen Anwendungen war die Menge M ein Simplex. Wir beschreiben das sogenannte revidierte Simplex-Verfahren, welches für sehr große Probleme besonders geeignet ist.

Wir beschreiben das Verfahren für die Normalform

$$\text{Minimiere} \quad z(x) = \sum_{i=1}^{n} c_i x_i \ \text{ in } \ M = \{x \in \mathbb{R}^n : x \geq 0 \ \ , Ax = b\} \tag{3.1}$$

Wie immer soll die (m,n)-Matrix A den Rang m besitzen. Ecken x von M sind dann dadurch gekennzeichnet, daß die Vektoren a_i mit $x_i > 0$ linear unabhängig sind. Natürlich kann es nicht mehr als m solcher Komponenten x_i geben. Sind es genau m, so heißt die Ecke nicht entartet, andernfalls entartet. Für nicht entartete Ecken gibt es immer genau eine Basis $I = \{i : x_i > 0\}$. Diese enthält genau m Elemente, und die Basis-Vektoren a_i, $i \in I$ sind linear unabhängig. Wir wollen annehmen, daß keine Ecke von M entartet ist.

Das Simplex-Verfahren konstruiert eine endliche Folge von Ecken von M mit abnehmenden Werten von z und endet mit einer Lösung oder der Information, daß das Problem nicht lösbar ist. Jede Ecke kann als Anfang dieser Folge gewählt werden. Die Berechnung einer Ausgangsecke von M ist Gegenstand der Phase I des Simplex-Verfahrens. Bei der jetzt zu schildernden Phase II nehmen wir an, wir hätten eine Ecke x^0 bereits gefunden. Es sei I die Basis von x^0 und J die Menge der nicht in I liegenden Indizes oder die Nicht-Basis-Indizes. Es ist also $x_i^0 > 0$ für $i \in I$, $x_j^0 = 0$ für $j \in J$, und $a_{i,i\in I}$, sind linear unabhängig. Wir können $Ax = b$ nach den Basis-Variablen x_i, $i \in I$ auflösen und erhalten

$$z = z(x^0) + \sum_{j \in J} p_j x_j \tag{3.2}$$

$$x_i = x_i^0 + \sum_{j \in J} c_{i,j} x_j \quad , \quad i \in I\ . \tag{3.3}$$

Ist $A_I = (a_i)_{i \in I}$, $A_J = (a_j)_{j \in J}$, so gilt für die $(m, n-m)$-Matrix $C = (c_{ij})_{i \in I, j \in J}$

$$C = A_I^{-1} A_J\ . \tag{3.4}$$

Die Koeffizienten von (3.2)-(3.3) fassen wir ins Simplex-Tableau der Ecke x^0 zusammen:

		J
	$z(x^0)$	p_j
I	x_i^0	$c_{i,j}$

(3.5)

Nun kommt die Beschreibung des Simplex-Schrittes, welcher die Ecke x^0 in eine neue Ecke x^1 mit kleinerem Wert der Zielfunktion überführt. Er besteht aus folgenden Teilschritten:

1. Suche $j_0 \in J$ mit $p_{j_0} < 0$.

Gibt es kein solches, so ist $p_j \geq 0$ für alle $j \in J$ und damit nach (3.2) $z(x) \geq z(x_0)$ für alle $x \in M$. Also ist x^0 bereits Lösung und das Verfahren beendet.

2. Suche nach einem $i_0 \in I$ mit $c_{i_0,j_0} < 0$ und

$$\frac{x_{i_0}^0}{-c_{i_0,j_0}} \leq \frac{x_i^0}{-c_{i,j_0}} \quad \text{für alle} \quad i \in I \;\text{ mit }\; c_{i,j_0} < 0 \;.$$

Gibt es kein solches i_0, so ist $c_{i,j_0} \geq 0$ für alle $i \in I$, und der Punkt $x(t) = x^0 + te_{j_0}$ ist wegen (3.3) zulässig für alle $t \geq 0$, während wegen (2.2) $z(x(t)) \to -\infty$ für $t \to +\infty$. Also ist z nach unten nicht beschränkt auf M, die Optimierungsaufgabe unlösbar, und das Verfahren beendet.

3. Konstruktion der neuen Ecke x^1.

Gibt es $i \in I$ mit $c_{i,j_0} < 0$, so ist wegen (3.3) $x(t) = x^0 + te_{j_0} \in M$ genau für $x_i(t) = x_i^0 + tc_{i,j_0} \geq 0$ für alle diese i, d.h. für

$$t \leq \frac{x_i^0}{-c_{i,j_0}} \quad \text{für } i \in I \;\text{ mit }\; c_{i,j_0} < 0 \;.$$

Nach Wahl von i_0 ist $t_0 = x^0_{i_0}/(-c_{i_0,j_0})$ das größte t mit dieser Eigenschaft. Wir setzen

$$x^1 = x(t_0) = x^0 + t_0 e_{j_0} \; .$$

Es ist

$$\begin{array}{rcl} z(x^1) & = & z(x^0) + t_0 p_{j_0} < z(x^0) \; , \\ x^1_{i_0} & = & 0 \quad , \quad x^1_{j_0} = t_0 > 0 \; . \end{array}$$

Wir zeigen, daß x^1 eine Ecke mit Basis $I \setminus \{i_0\} \cup \{j_0\}$ ist, d.h. daß die Vektoren $a_i, \;\; i \in I \setminus \{i_0\} \cup \{j_0\}$ linear unabhängig sind. Wäre dies nicht der Fall, so gäbe es Zahlen $\alpha_i, \;\; i \in I \setminus \{i_0\} \cup \{j_0\}$, die nicht alle verschwinden, so daß

$$\sum_{i \in I - \{i_0\}} \alpha_i a_i + \alpha_{j_0} a_{j_0} = 0 \; .$$

Nach (3.4) ist $A_I C = A_J$. Die j_0-te Spalte hiervon lautet

$$\sum_{i \in I} c_{i,j_0} a_i = a_{j_0} \; .$$

Damit haben wir zwei Darstellungen von a_{j_0} durch die linear unabhängigen Vektoren a_i, $i \in I$. In der ersten verschwindet der Koeffizient von a_{i_0}, in der zweiten ist er $c_{i_0,j_0} \neq 0$. Dieser Widerspruch zeigt, daß die a_i mit $i \in I \setminus \{i_0\} \cup \{j_0\}$ linear unabhängig sind und damit x^1 eine Ecke mit der Basis $I \setminus \{i_0\} \cup \{j_0\}$ ist.

4. Berechnung des neuen Tableaus.

Dazu löst man $Ax = b$ oder (3.3) nach den neuen Basis-Variablen x_i, $i \in I \setminus \{i_0\} \cup \{j_0\}$ auf. Aus der Gleichung (3.3) für $i = i_0$ erhalten wir zunächst

$$x_{j_0} = (x_{i_0} - x^0_{i_0} - \sum_{j \in J \setminus \{j_0\}} c_{i_0,j} x_j)/c_{i_0,j_0}$$

und dann aus den restlichen Gleichungen von (3.3)

$$\begin{array}{rcl} x_i & = & x^0_{i_0} + \displaystyle\sum_{j \in J \setminus \{j_0\}} c_{i,j} x_j + c_{i,j_0} x_{j_0} \\ & = & x^0_{i_0} + \displaystyle\sum_{j \in J \setminus \{j_0\}} c_{i,j} x_j + \frac{c_{i,j_0}}{c_{i_0,j_0}} \left(x_{i_0} - x^0_{i_0} - \displaystyle\sum_{j \in J \setminus \{j_0\}} c_{i_0,j} x_j \right) \\ & = & x^0_{i_0} - \dfrac{c_{i,j_0}}{c_{i_0,j_0}} x^0_{i_0} + \dfrac{c_{i,j_0}}{c_{i_0,j_0}} x_{i_0} + \displaystyle\sum_{j \in J - \{j_0\}} \left(c_{i,j} - \frac{c_{i,j_0}}{c_{i_0,j_0}} c_{i_0,j} \right) x_j \end{array}$$

für $i \in I\backslash\{i_0\}$. Auch die Zielfunktion muß in den neuen Nicht-Basis-Variablen ausgedrückt werden. Wir erhalten

$$\begin{aligned} z &= z(x^0) + \sum_{j \in J\backslash\{j_0\}} p_j x_j + p_{j_0} x_{j_0} \\ &= z(x^0) - \frac{p_{j_0}}{c_{i_0,j_0}} x^0_{i_0} + \sum_{j \in J\backslash\{j_0\}} \left(p_j - \frac{p_{j_0}}{c_{i_0,j_0}} c_{i_0,j} \right) x_j \, . \end{aligned}$$

Das (i_0, j_0)-Element im Simplex-Tableau heißt Pivotelement, die zugehörige Zeile bzw. Spalte Pivotzeile bzw. Pivotspalte. Das Simplex-Tableau für x^1 hat die Form

		$j \in J \setminus \{j_0\}$	i_0
	$z(x^0) - \frac{p_{j_0}}{c_{i_0,j_0}} x^0_{i_0}$	$p_j - \frac{p_{j_0}}{c_{i_0,j_0}} c_{i_0,j}$	$\frac{p_{j_0}}{c_{i_0,j_0}}$
$i \in I \setminus \{i_0\}$	$x^0_{i_0} - \frac{c_{i,j_0}}{c_{i_0,j_0}} x^0_{i_0}$	$c_{i,j} - \frac{c_{i,j_0}}{c_{i_0,j_0}} c_{i_0,j}$	$\frac{c_{i,j_0}}{c_{i_0,j_0}}$
j_0	$-\frac{x^0_{i_0}}{c_{i_0,j_0}}$	$-\frac{c_{i_0,j}}{c_{i_0,j_0}}$	$\frac{1}{c_{i_0,j_0}}$

Dabei schreibt man die neue Zeile j_0 (Spalte i_0) zweckmäßigerweise in die alte Zeile i_0 (Spalte j_0).

Sind, wie vorausgesetzt, alle Ecken von M nicht entartet, so ist $t_0 > 0$, und die Zielfunktion nimmt bei jedem Schritt echt ab. Das Simplex-Verfahren bricht also nach endlich vielen Schritten ab, entweder mit der Lösung, oder mit der Aussage, daß z auf M nicht nach unten beschränkt ist.

Wir schreiben nun ein Programm, das den Übergang von einem Tableau zum nächsten bewerkstelligt. Wir werden sehen, daß das ganze Tableau, also auch erste Spalte und Zeile, einheitlich behandelt werden können. Deshalb nehmen wir das Tableau in der Form

$$\begin{array}{c|ccc}
c_{0,0} & c_{0,1} & \cdots & c_{0,q} \\
\hline
c_{1,0} & c_{1,1} & \cdots & c_{1,q} \\
\vdots & \vdots & & \vdots \\
c_{m,0} & c_{m,1} & \cdots & c_{m,q}
\end{array}$$

an mit $q = n - m$. Die Zeilenindizes $1, \ldots, m$ entsprechen also den Basis-Indizes $i_1, \ldots, i_n$, die Spaltenindizes $1, \ldots, q$ den Nicht-Basis-Indizes $j_1, \ldots, j_q$.

austausch (C, i_0, j_0, m, q)

/ $\star$ Berechnet nach Bestimmung von i_0, j_0 das neue Tableau.
Schreibt den neuen Basis-Index j_{j_0} in Zeile i_0 und den
neuen Nicht-Basis-Index i_{i_0} in die Spalte j_0 / $\star$

{ $c_{i_0,j_0} = 1/c_{i_0,j_0}$;
for $i = 0, \ldots, m$ if $(i \neq i_0) c_{i,j_0} = c_{i,j_0} * c_{i_0,j_0}$;
for $i = 0, \ldots, m$ for $j = 0, \ldots, q$
if $(i \neq i_0$ und $j \neq j_0)$ $c_{i,j} = c_{i,j} - c_{i,j_0} * c_{i_0,j}$;
for $j = 0, \ldots, q$ if $(j \neq j_0)$ $c_{i_0,j} = -c_{i_0,j} * c_{i_0,j_0}$;
}

Der Austauschschritt benötigt $qm+$ O$(m + q)$ flops.

Mit Hilfe der Austauschroutine kann man nun leicht ein Programm für die Phase II des Simplex-Verfahrens schreiben:

simplex (C, I, J, m, q)

/ $\star$ Bei Aufruf enthält C das Tableau der Ausgangs-Ecke, I die Basis-,
J die Nicht-Basis-Indizes. Nach Ablauf steht das Minimum der
Zielfunktion in $c_{0,0}$, die Lösung in $c_{i,0}$, $i = 1, \ldots, m$
und die Basis-Indizes der Lösung in I / $\star$

$\{$ while $(\exists\, j \in \{1,\ldots,q\} : c_{0,j} < 0)$
$\quad\{$ Wähle j_0;
Wähle $i_0 \in \{1,\ldots,m\}$, so daß $\frac{c_{i,0}}{-c_{i,j_0}}$ minimal für
$i = i_0$ unter allen i mit $c_{i,j_0} < 0$;
breche ab mit Meldung "Problem nicht lösbar"
falls es keine solchen i gibt;
$I = I \setminus \{i_{i_0}\} \cup \{j_{j_0}\}$; $J = J \setminus \{j_{j_0}\} \cup \{i_{i_0}\}$;
austausch (C, i_0, j_0, m, q);
$\quad\}$
$\}$

Der Austauschschritt kann auch zum Invertieren von Matrizen benutzt werden. Sei C eine $(n+1, n+1)$-Matrix und $y = Cx$, also

$$\begin{array}{lcccc} y_0 & = & c_{0,0}x_0 & +\cdots+ & c_{0,n}x_n \\ \vdots & & & & \\ y_n & = & c_{n,0}x_0 & +\cdots+ & c_{n,n}x_n\ . \end{array}$$

Wir lösen die nullte Gleichung nach x_0 auf und setzen x_0 in die restlichen Gleichungen ein. Es entsteht

$$\begin{array}{lccccccc} x_0 & = & c'_{0,0}y_0 & + & c'_{0,1}x_1 & +\cdots+ & c'_{0,n}x_n \\ y_1 & = & c'_{1,0}y_0 & + & c'_{1,1}x_1 & +\cdots+ & c'_{1,n}x_n \\ \vdots & & & & & & \\ y_n & = & c'_{n,0}y_0 & + & c'_{n,1}x_1 & +\cdots+ & c'_{n,n}x_n \end{array}$$

mit neuen Elementen $c'_{i,j}$, die wir durch den Aufruf

austausch $(C, 0, 0, n, n)$

berechnen können (falls $c_{0,0} \neq 0$). Danach lösen wir die erste Gleichung nach x_1 auf und setzen x_1 in die restlichen Gleichungen ein. Dann sind x_0, x_1, $y_2,\ldots,y_n$ durch y_0, y_1, $x_2,\ldots,x_n$ ausgedrückt. So fortführend können wir schließlich $x_0,\ldots,x_n$ durch $y_0,\ldots,y_n$ ausdrücken. Die Koeffizienten dieser Formeln sind dann gerade die Elemente von C^{-1}. Das Programm

for $i = 0,\ldots,n$
austausch (C, i, i, n, n)

invertiert also C. Das geht natürlich nur gut, wenn alle Pivotelemente $\neq 0$ sind.

2.4 Das Simplex-Verfahren: Phase I

Die in §3 beschriebene Phase II des Simplex-Verfahrens setzt voraus, daß wir eine Ecke x^0 von M kennen. Eine solche Ecke liefert uns die Phase I. Dabei nehmen wir wieder an, daß M in der Normalform

$$M = \{x \in \mathbb{R}^n : x \geq 0 \ , \quad Ax = b\}$$

vorliegt mit einer (m, n)-Matrix A. Wir können zusätzlich $b \geq 0$ annehmen.

Wir führen nun ein Hilfsproblem in den $n+m$ Variablen $x_1, \ldots, x_n, y_1, \ldots, y_m$ ein. Seine zulässige Menge ist

$$\tilde{M} = \{x, y : x \in \mathbb{R}^n, \ y \in \mathbb{R}^m, \ x, y \geq 0, \ Ax + y = b\} \ ,$$

und seine Zielfunktion

$$\tilde{z}(x, y) = \sum_{i=1}^{m} y_i \ .$$

Wir zeigen nun: M ist nicht leer genau dann, wenn das Hilfsproblem eine Lösung mit $y = 0$ besitzt. Ist nämlich $x \in M$, so ist $\begin{pmatrix} x \\ 0 \end{pmatrix}$ Lösung des Hilfsproblems. Ist umgekehrt $\begin{pmatrix} x \\ 0 \end{pmatrix}$ Lösung des Hilfsproblems, so ist $x \in M$.

Das Hilfsproblem ist - für $b > 0$ - genau von der Form, die wir für die Phase II vorausgesetzt haben. Eine Ecke von $\tilde{M}$ ist $\begin{pmatrix} 0 \\ b \end{pmatrix}$, die zugehörigen Basis-Variablen sind $y = b - Ax$, und die Zielfunktion ist

$$\tilde{z}(x, y) = e^*(b - Ax) \ , \qquad e^* = (1, \ldots, 1) \ .$$

Man kann nun die Phase II des Simplex-Verfahrens für das Hilfsproblem durchführen. Endet dies mit einer Lösung mit $y = 0$, so hat man eine Startecke von M gefunden. Andernfalls ist M leer.

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 Cordani'schen Formeln erfolgen:

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

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

Auch hier tritt unter Umständen, nämlich 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 $((k - \frac{1}{2}), (k + \frac{1}{2})\pi)$, $k \in \mathbb{Z}$, 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

```
fa = f(a)   ;   fb = f(b);
     while      (b - a >= ε)
            {   c = (b + a)/2;
                fc = f(c);
                if (fa fc < 0)  {b = c; fb = fc; }
                else {a = c; fa = fc; }
            }
```

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.

□

Beispiele:

1) 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.

2) Der Brouwer'sche Satz wird häufig auf Probleme der Volkswirtschaft angewendet. Ein Markt bestehe aus n Gütern mit Preisen $x_1, \ldots, x_n \geq 0$ und aus m Teilnehmern (Produzenten oder Konsumenten). Bei den Preisen $x = (x_1, \ldots, x_n)$ kauft oder produziert der ℓ-te Teilnehmer die Menge $f_i^\ell(x)$ von Gut i. Die totale Nachfrage bzw. Produktion von Gut i ist dann $f_i(x) = \sum\limits_{\ell=1}^{m} f_i^\ell(x)$. Man sagt, der Markt sei im Gleichgewicht, wenn

$$f_i(x) \leq 0\ , \quad i = 1, \ldots, n, \quad f_i(x) = 0 \quad \text{falls} \ \ x_i > 0\ .$$

Wir machen folgende Voraussetzungen:

1) Die Nachfrage hängt nur von den relativen Preisen ab, d.h.

$$f_i^\ell(tx) = f_i^\ell(x) \quad , \quad t > 0\ .$$

2) Der Markt ist abgeschlossen, d.h.

$$\sum_{i=1}^{n} x_i f_i^\ell(x) = 0 \quad , \quad \ell = 1, \ldots, m$$

3) Die f_i^ℓ sind stetig.

Dann gibt es einen Preisvektor x, bei welchem der Markt im Gleichgewicht ist.

Zum Beweis setzen wir $D = \{x \in \mathbb{R}^n : x \geq 0,\ \sum\limits_{i=1}^{n} x_i = 1\}$ und auf D

$$g_i(x) = \frac{x_i + \ \mathrm{Max}\ (0, f_i(x))}{1 + \sum\limits_{k=1}^{n}\ \mathrm{Max}\ (0, f_k(x))}\ , \quad i = 1, \ldots, n\ .$$

Die Abbildung $g(x) = (g_1(x), \ldots, g_n(x))$ ist dann stetig auf der konvexen kompakten Menge D und bildet D in sich ab. Nach dem Satz gibt es ein $x \in D$ mit $g(x) = x$, also

$$x_i = \frac{x_i + \text{ Max } (0, f_i(x))}{1 + \sum\limits_{k=1}^{n} \text{ Max } (0, f_k(x))} \quad , \quad i = 1, \ldots, n \; .$$

Hieraus folgt, daß x Gleichgewichtspunkt ist (Aufgabe 21).

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*

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

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

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

also

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

weil D konvex ist.

$\square$

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-Konstante 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

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

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 $\overline{x} \in \mathbb{R}^n$. 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

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

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

3) Fehlerabschätzung

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

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

□

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 $g'(x) = 1/\cos^2 x \ge 1$. 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 $|g'(x)| \le 1/(1 + \pi^2/4) < 1$, und g bildet D in sich ab. Nach Satz

III.2.1 ist g in D kontrahierend mit $q = 1/(1+\pi^2/4) = 0.29$. 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	$\frac{q^k}{1-q}\|x^1 - x^0\|$	$\overline{x} - x^k$
0	3.1416		–
1	4.4042	0.41	0.0892
2	4.4891	0.19	0.0043
3	4.4932	0.034	0.0002
4	4.4934	0.010	–

Wir sehen, daß die Fehlerabschätzung viel zu pessimistisch ist. Man kann auch $D = [\pi, \frac{3\pi}{2}]$ wählen mit $q = 1/(1+\pi^2) = q = 0.092$ 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 die "anziehenden" Fixpunkt sind, d.h. diejenigen mit $|g'(\overline{x})| < 1$. 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 $D = \{x \in K^n : \|x - \overline{x}\| \le r\}$ mit einem, $r > 0$ kontrahierend ist, und zwar mit der Lipschitz-Konstanten $q < 1$. Dann ist für $x \in D$

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

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

$\square$

Bemerkungen:

1) Sei $K = \mathbb{R}$. Die Bedingung des Satzes III.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 $\mathbb{R}^n$ 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 III.2.1 ist g in dieser Umgebung kontrahierend.

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 III.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 $f(x) = 0$, $f : D \to \mathbb{R}^n$, $D \subseteq \mathbb{R}^n$. 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

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

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 $f \in C^2(D)$

$$f(x) = f(x^k) + f'(x^k)(x - x^k) + \text{ O } (\|x - x^k\|^2) \,.$$

Man vernachlässigt nun $\text{O}(\|x - x^k\|^2)$ 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 $\mathbb{R}^1$ 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 $\mathbb{R}^1$ 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	1.417	3
3	1.414216	6
4	1.414213562	10

Satz 3.3.1 *Die Abbildung $f : \mathbb{R}^n \to \mathbb{R}^n$ 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 III.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

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

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

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

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

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

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

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 III.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

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

Hierzu schreiben wir

$$\begin{aligned} x^{k+1} - \overline{x} &= x^k - \overline{x} - \frac{x^k - x^{k-1}}{f(x^k) - f(x^{k-1})} f(x^k) \\ &= (x^k - \overline{x}) \frac{\dfrac{f(x^k) - f(x^{k-1})}{x^k - x^{k-1}} - \dfrac{f(x^k) - f(\overline{x})}{x^k - \overline{x}}}{\dfrac{f(x^k) - f(x^{k-1})}{x^k - x^{k-1}}} \\ &= (x^k - \overline{x}) \frac{\int\limits_0^1 \{f'(x^{k-1} + t(x^k - x^{k-1})) - f'(\overline{x} + t(x^k - \overline{x}))\} dt}{f'(\xi^k)} \end{aligned}$$

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 $f_k \leq F_k$, $k = 0, 1, \dots$. 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

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

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

$\square$

Kapitel 4

Iterationsverfahren für lineare Gleichungssysteme

4.1 Gesamt- und Einzelschrittverfahren

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

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

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

$GS(X,n)$ /$\star$ Führt einen Gesamtschritt durch $\star$/
{ for $i = 1, \ldots, n$
$x_i^1 = (b_i - \sum_{j \neq i} a_{i,j} x_j)/a_{i,i};$
$x = x^1;$
}
$ES(X,n)$ /$\star$ Führt einen Einzelschritt durch $\star$/
{ for $i = 1, \ldots, n$
$x_i = (b_i - \sum_{j \neq i} a_{i,j} x_j)/a_{i,i};$
}

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

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

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

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

4.2 Konvergenz

Alle unsere Konvergenzaussagen werden auf folgendem Satz beruhen:

Satz 4.2.1 *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 I.3.2 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 III.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 4.2.1 *Eine (n,n)-Matrix A über K erfüllt das starke Zeilensummenkriterium, wenn $|a_{i,i}| > \sum\limits_{j \neq i} |a_{i,j}|, \quad i = 1, \ldots, n$. 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 4.2.2 *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 zu 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 nur $\rho(D^{-1}(L + R)) \leq 1$. Es genügt daher zu zeigen, daß $D^{-1}(L + R)$ keinen Eigenwert λ mit $|\lambda| = 1$ hat. Wäre λ ein solcher und x ein dazugehöriger 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 4.2.3 *Die (n,n)-Matrix A sei positiv definit. Dann konvergiert das SOR-Verfahren zur Lösung von $Ax = b$ genau dann 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, $\beta \in \mathbb{R}$ 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

$$|1 - \omega - \omega\alpha'| < |1 + \omega\alpha'| \; . \tag{2.1}$$

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 (2.1) weglassen und erhalten

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

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

$\square$

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

Satz 4.2.4 *Sei B_ω die SOR-Matrix für eine beliebige Matrix mit nichtverschwindenden Diagonalelementen. Dann gilt*

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

Beweis: Es ist

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

Die Diagonale von B_ω ist also $(1-\omega)I$. Sind $\lambda_1, \ldots, \lambda_n$ die Eigenwerte von A (nicht notwendig verschieden), so gilt

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

und daraus folgt die Behauptung.

$\square$

4.3 Das Verfahren der konjugierten Gradienten (CG)

Ein Problem des SOR-Verfahrens liegt in der Bestimmung des optimalen Wertes von ω. Das CG-Verfahren kommt ohne einen solchen Parameter aus und hat trotzdem gute Konvergenzeigenschaften.

Sei A eine reelle positiv definite (n,n)-Matrix. Zu lösen sei $Ax = b$. Wir setzen

$$f(x) = \frac{1}{2}(x, Ax) - (x, b) \ .$$

Satz 4.3.1 *$\overline{x}$ ist genau dann Lösung von $A\overline{x} = b$, wenn $f(\overline{x}) \leq f(x)$ für alle $x \in \mathbb{R}^n$.*

Beweis: Ist $\overline{x}$ die gesuchte Lösung, so ist

$$\frac{1}{2}(A(x-\overline{x}), (x-\overline{x})) = \frac{1}{2}(Ax, x) - (x, b) + \frac{1}{2}(A\overline{x}, \overline{x}) \ .$$

$\overline{x}$ ist also der eindeutig bestimmte Vektor, welcher $f(x)$ minimiert.

□

Anstelle einer Lösung von $Ax = b$ können wir also genausogut das Minimum von $f(x)$ suchen. Sei x^0 eine Ausgangsnäherung für $\overline{x}$. Wir suchen eine bessere Näherung für $\overline{x}$ in der Form $x^0 + \alpha d^0$, wo $d^0 \neq 0$ eine "Suchrichtung" ist. Wir bestimmen α so, daß $f(x^0 + \alpha d^0)$ minimal ist. Wegen $f'(x) = Ax - b$ ist

$$\begin{aligned}
\frac{d}{d\alpha} f(x^0 + \alpha d^0) &= (f'(x^0 + \alpha d^0), d^0) \\
&= (A(x^0 + \alpha d^0) - b, d^0) \\
&= (r^0, d^0) + \alpha(Ad^0, d^0) \ , \\
\frac{d^2}{d\alpha^2} f(x^0 + \alpha d^0) &= (Ad^0, d^0) > 0 \ ,
\end{aligned}$$

wobei wir hier und unten $r^k = Ax^k - b$ setzen. Entlang $x^0 + \alpha d^0$ ist also f genau dann minimal, wenn die erste Ableitung von f verschwindet. Dies ist für

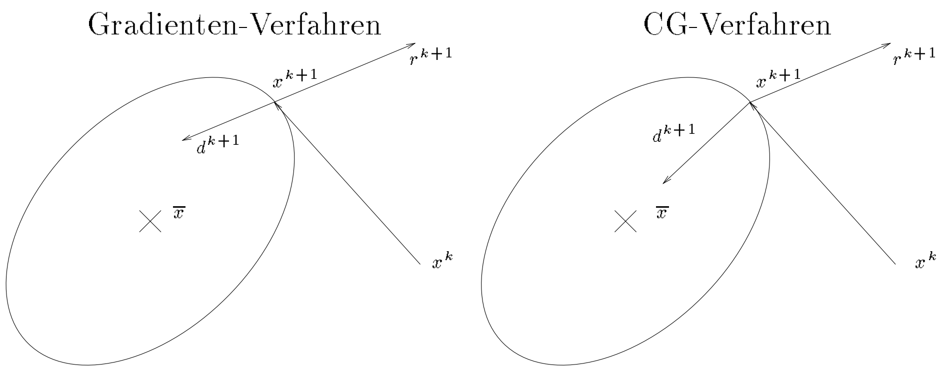
$$\alpha_0 = -\frac{(r^0, d^0)}{(Ad^0, d^0)}$$

der Fall. Geometrisch bedeutet diese Wahl, daß $r^1 \perp d^0$, und d^0 liegt im Tangentialraum des Ellipsoids $f(x) = f(x^1)$ im Punkte x^1. Die neue Approximation für $\overline{x}$ ist dann $x^1 = x^0 + \alpha d^0$.

Wie wählt man nun d^0? Eine naheliegende Wahl ist $d^0 = -r^0$, d.h. man läuft in Richtung des steilsten Abstiegs von f. Dies führt zu den Verfahren des steilsten Abstiegs (steepest descent) oder Gradientenverfahren:

Initialisierung	:	Wähle x^0, $r^0 = Ax^0 - b$, $d^0 = -r^0$.
Für $k = 0, 1, 2, \ldots$	:	$z = Ad^k$
		$\alpha_k = -\dfrac{(d^k, d^k)}{(z, d^k)}$
		$x^{k+1} = x^k + \alpha_k d^k \quad , \quad d^{k+1} = d^k + \alpha_k z$.

Die Suchrichtung beim k-ten Schritt ist $d^k = -r^k$.



Wir haben gesehen, daß für das Gradientenverfahren $r^{k+1} \perp d^k$ ist, also $d^{k+1} \perp d^k$. Dies bedeutet, daß x^k sich $\overline{x}$ auf einem Zick-Zack-Kurs annähert. Dies läßt für die Konvergenzgeschwindigkeit nichts Gutes hoffen.

Beim CG-Verfahren ersetzt man die eindimensionale Minimierung durch eine zweidimensionale. Sind x^0, x^1 nach den Gradientenverfahren berechnet, so setzt man

$$x^2 = x^1 - \alpha r^1 + \beta d^0$$

und bestimmt α, β so, daß $f(x^2)$ minimal ist. Die Bedingungen für ein Minimum sind

$$\begin{aligned} \frac{\partial f}{\partial \alpha}(x^2) &= (Ax^2 - b, r^1) = 0 \\ \frac{\partial f}{\partial \beta}(x^2) &= (Ax^2 - b, d^0) = 0 \;. \end{aligned}$$

Auflösen dieser Gleichungen nach α, β liefert x^2. Wir benutzen nur die zweite Gleichung, welche wir in der Form

$$(r^1 - \alpha A r^1 + \beta A d^0, d^0) = 0$$

schreiben können. Wegen $(r^1, d^0) = 0$ folgt

$$\beta = \alpha \beta_0 \quad , \quad \beta_0 = \frac{(Ar^1, d^0)}{(Ad^0, d^0)}$$

und damit

$$x^2 = x^1 + \alpha d^1 \quad , \quad d^1 = -r^1 + \beta_0 d^0 \;.$$

Nun können wir α durch eine eindimensionale Minimierung bestimmen. Das Resultat α_1 kennen wir schon vom Gradientenverfahren, nämlich

$$\alpha_1 = -\frac{(r^1, d^1)}{(d^1, Ad^1)} \;.$$

Damit ist x^2 bestimmt, allerdings in einer sehr unpraktischen Form: Die Berechnung von x^2 erfordert drei Anwendungen der Matrix A auf einen Vektor. Wir wollen mit einer Anwendung auskommen. Dazu schreiben wir β_0 in der Form

$$\beta_0 = \frac{(r^1, Ad^0)}{(d^0, Ad^0)} = \frac{(r^1, (r^1 - r^0)/\alpha_0)}{(d^0, Ad^0)} = \frac{(r^1, r^1)}{(r^0, r^0)} \;,$$

wobei, wie beim Gradientenverfahren, $\alpha_0 = (r^0, r^0)/(d^0, Ad^0)$. Für α_1 haben wir die α_0 entsprechende Form

$$\alpha_1 = -\frac{(r^1, -r^1 + \beta_0 d^0)}{(d^1, Ad^1)} = \frac{(r^1, r^1)}{(d^1, Ad^1)} ,$$

denn $(r^1, d^0) = 0$. Damit lautet das CG-Verfahren:

$$\begin{array}{lcl}
\text{Initialisierung} & : & \text{Wähle } x^0,\ r^0 = Ax^0 - b,\ \ d^0 = -r^0 \\
\text{Für } k = 0, 1, \ldots & : & z = Ad^k \\
 & & \alpha_k = \dfrac{(r^k, r^k)}{(d^k, z)} \\
 & & x^{k+1} = x^k + \alpha_k d^k \ ,\ \ r^{k+1} = r^k + \alpha_k z \\
 & & \beta_k = \dfrac{(r^{k+1}, r^{k+1})}{(r^k, r^k)} \\
 & & d^{k+1} = -r^{k+1} + \beta_k d^k \quad .
\end{array}$$

Für $k = 0$ macht also das CG-Verfahren einen Gradientenschritt. Die weiteren Schritte führen wie das Gradientverfahren eine eindimensionale Minimierung durch, aber nicht entlang des steilsten Abstiegs $-r^k$, sondern entlang der Richtung d^k. Das Verfahren bricht ab, wenn $r^k = 0$. In diesem Fall ist $x^k = \overline{x}$.

Satz 4.3.2 *Das CG-Verfahren bricht nach höchstens n Schritten mit $r^k = 0$ ab. Solange es nicht abbricht, gilt*

(a) $(d^i, r^k) = 0$ *für* $i < k$,

(b) $(r^k, d^k) = -\|r^k\|^2$,

(c) $(r^i, r^k) = 0$ *für* $i < k$,

(d) $(d^i, Ad^k) = 0$ *für* $i < k$,

(e) $sp\ \{d^0, \ldots, d^k\} = sp\ \{r^0, \ldots, r^k\}$.

Beweis: Wir beweisen (a) - (e) durch Induktion nach k. Die Aussage über den Abbruch folgt dann aus (c), weil es nicht mehr als n linear unabhängige Vektoren r^k gibt. Solange $r^k \neq 0$ ist, ist nach (b) auch $d^k \neq 0$. Das Verfahren bricht also nur mit $r^k = 0$ ab. Für $k = 0$ sind (a) - (e) trivial. Sei (a) - (e) richtig für ein $k \geq 0$. Wir zeigen, daß (a) - (e) dann auch für $k+1$ richtig ist, wenn nur $r^{k+1} \neq 0$.

(a) Es ist für $i < k$

$$(d^i, r^{k+1}) = (d^i, r^k + \alpha_k A d^k) = 0$$

wegen (a) und (d) für k. Für $i = k$ ist

$$(d^k, r^{k+1}) = 0 \ ,$$

weil x^{k+1} durch Minimierung entlang d^k entsteht.

(b) Es ist

$$(r^{k+1}, d^{k+1}) = (r^{k+1}, -r^{k+1} + \beta_k d^k) = -\|r^{k+1}\|^2 \ ,$$

weil (a) auch für $k+1$ gilt.

(c) Weil (a) auch für $k+1$ gilt, ist für $i < k+1$ mit $\beta_{-1} = 0$

$$(r^i, r^{k+1}) = (-d^i + \beta_{i-1} d^{i-1}, r^{k+1}) = 0 \ .$$

(d) Aus (b) für $k+1$ folgt $d^{k+1} \neq 0$. Für $i < k$ ist

$$\begin{aligned} (d^i, A d^{k+1}) &= (d^i, A(-r^{k+1} + \beta_k d^k)) \\ &= -(A d^i, r^{k+1}) + \beta_k (d^i, A d^k) \\ &= \frac{1}{\alpha_i}(r^i - r^{i+1}, r^{k+1}) + \beta_k (d^i, A d^k) \ . \end{aligned}$$

Dies verschwindet wegen (c) für $k+1$ und (d) für k. Für $i = k$ ist

$$\begin{aligned} (d^k, A d^{k+1}) &= (A d^k, d^{k+1}) = \frac{1}{\alpha_k}(r^{k+1} - r^k, -r^{k+1} + \beta_k d^k) \\ &= \frac{1}{\alpha_k}((r^{k+1}, r^{k+1}) - \beta_k (r^k, d^k)) \end{aligned}$$

wegen (a), (c) für $k+1$, und dies verschwindet nach Definition von β_k.

(e) Es ist $\mathrm{sp}\,\{d^0, \ldots, d^k\} = \mathrm{sp}\,\{r^0, \ldots, r^k\}$ und $d^{k+1} = -r^{k+1} + \beta_k d^k$. Also ist auch $\mathrm{sp}\,\{d^0, \ldots, d^{k+1}\} = \mathrm{sp}\,\{r^0, \ldots, r^{k+1}\}$.

$\square$

Bemerkung: Man nennt Vektoren x, y mit $(x, Ay) = 0$ konjugiert (bezüglich A). Nach Teil (d), (e) des Satzes entstehen die Richtungen d^k aus den Gradienten r^k durch Orthogonalisierung bezüglich des inneren Produktes (x, Ay). Dies erklärt den Namen "konjugierte Gradienten".

Der folgende Satz gibt eine Erklärung für die Effizienz des CG-Verfahrens.

Satz 4.3.3 *Solange das CG-Verfahren nicht abbricht, minimiert x^k die Funktion $f(x)$ in $x^0 +$ sp $\{d^0, \dots, d^{k-1}\}$.*

Beweis: Sei $f(x)$ minimal für $x = x^0 + \sum_{i=0}^{k-1} c_i d^i$, also

$$\frac{\partial}{\partial c_i} f\left(x^0 + \sum_{i=0}^{k-1} c_i d^i\right) = 0 \ , \qquad i = 0, \dots, k-1 \ .$$

Wegen

$$\begin{aligned}
f\left(x^0 + \sum_{i=0}^{k-1} c_i d^i\right) &= \frac{1}{2}\left(x^0 + \sum_{i=0}^{k-1} c_i d^i, A\left(x^0 + \sum_{i=0}^{k-1} c_i d^i\right)\right) - \left(x^0 + \sum_{i=0}^{k-1} c_i d^i, b\right) \\
= \quad & f(x^0) + \frac{1}{2}\sum_{i,j=0}^{k-1} c_i c_j (d^i, Ad^j) + \sum_{i=0}^{k-1} c_i (d^i, Ax^0 - b) \\
= \quad & f(x^0) + \frac{1}{2}\sum_{i=0}^{k-1} c_i^2 (d^i, Ad^i) + \sum_{i=0}^{k-1} c_i (d^i, r^0) \ ,
\end{aligned}$$

wo wir Satz 3.1 (d) benutzt haben, gilt

$$c_i = -\frac{(d^i, r^0)}{(d^i, Ad^i)} \ , \qquad i = 0, \dots, k-1 \ .$$

Nach Konstruktion des CG-Verfahrens ist

$$\begin{aligned}
d_i &= -r^i + \beta_{i-1} d^{i-1} \\
&= -r^i - \beta_{i-1}(r^{i-1} + \beta_{i-2} d^{i-2}) \\
&= \cdots \\
&= -r^i - \beta_{i-1} r^{i-1} - \beta_{i-1}\beta_{i-2} r^{i-2} - \cdots + \beta_{i-1}\cdots\beta_0 r^0 \ .
\end{aligned}$$

Nach Satz 3.1 (c) gilt also

$$(d^i, r^0) = -\beta_{i-1} \cdots \beta_0 (r^0, r^0) = (r^i, r^i)$$

aufgrund der Definition der β_k. Also haben wir

$$c_i = \frac{(r^i, r^i)}{(d^i, Ad^i)} = \alpha_i \ , \qquad i = 0, \dots, k-1 \ .$$

Also gilt für den minimalen Vektor x in der Tat

$$x = x^0 + \sum_{i=0}^{k-1} c_i d^i = x^0 + \sum_{i=0}^{k-1} \alpha_i d^i = x^k \ .$$

$\square$

Satz 4.3.4 *Für das CG-Verfahren gilt die Fehlerabschätzung*

$$\|x^k - \overline{x}\|_A \leq 2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k \|x^0 - \overline{x}\|_A \ ,$$

wo κ die Kondition von A in der euklidischen Norm und $\| \cdot \|_A$ die Energie-Norm $\|x\|_A = (x, Ax)^{1/2}$ bedeutet.

Beweis: Nach Satz 3.1 ist $f(x) = \frac{1}{2}\|x - \overline{x}\|_A^2 - \frac{1}{2}\|\overline{x}\|_A^2$. Nach Satz 3.3 minimiert x^k die Funktion $f(x)$ in x^0+ sp $\{d^0, \dots, d^{k-1}\}$, also minimiert x^k auch $\|x - \overline{x}\|_A$ in diesen Raum, der nach Satz 3.2 identisch ist mit x^0+ sp $\{r^0, \dots, r^{k-1}\}$. Wir zeigen

$$\mathrm{sp}\{r^0, \dots, r^{k-1}\} = \mathrm{sp}\{r^0, Ar^0, \dots, A^{k-1}r^0\} \ . \tag{3.1}$$

Für $k = 1$ ist dies richtig. Sei es richtig für ein $k \geq 1$. Es ist $r^k = r^{k-1} + \alpha_{k-1} Ad^{k-1}$, und nach Satz IV.3.2 (e) ist $d^{k-1} \in$ sp $\{r^0, \dots, r^{k-1}\} =$ sp $\{r^0, \dots, A^{k-1}r^0\}$, also $Ad^{k-1} \in$ sp $\{r^0, \dots, A^k r^0\}$ und damit $r^k \in$ sp $\{r^0, \dots, A^k r^0\}$. Umgekehrt ist dann auch $A^k r^0 \in$ sp $\{r^0, \dots, r^k\}$. Damit ist (3.1) gezeigt.

x^k hat also die Gestalt

$$x^k = x^0 + c_0 r^0 + \cdots c_{k-1} A^{k-1} r^0$$

oder

$$\begin{aligned} x^k - \overline{x} &= x^0 - \overline{x} + (c_0 I + \cdots + c_{k-1} A^k)(x^0 - \overline{x}) \\ &= Q(A)(x^0 - \overline{x}) \end{aligned}$$

mit einem Polynom Q der Ordnung k und $Q(0) = 1$. Da A positiv definiti ist, existiert $A^{1/2}$, und $A^{1/2}A = AA^{1/2}$. Also ist

$$\begin{aligned} \|x^k - \overline{x}\|_A &= (x^k - \overline{x}, A(x^k - \overline{x}))^{1/2} \\ &= \|A^{1/2}(x^k - \overline{x})\|_2 \\ &= \|A^{1/2} Q(A)(x^0 - \overline{x})\|_2 \\ &= \|Q(A) A^{1/2}(x^0 - \overline{x})\|_2 \\ &\leq \|Q(A)\|_2 \|A^{1/2}(x^0 - \overline{x})\|_2 \\ &= \|Q(A)\|_2 \|x^0 - \overline{x}\|_A \ . \end{aligned}$$

Seien $0 < \lambda_1 \leq \cdots \leq \lambda_n$ die Eigenwerte von A. Dann ist

$$\|Q(A)\|_2 = \max_{\lambda_1 \leq \lambda \leq \lambda_n} |Q(\lambda)| \ .$$

Wir suchen nun Q so zu bestimmen, daß $Q(1) = 1$ und $|Q|$ möglichst klein in $[\lambda_1, \lambda_n]$. Dazu führen wir die Tschebitscheff-Polynome T_k erster Art ein. Es ist

$$T_0(x) = 1 \ , \ T_1(x) = x \ , \ T_{k+1}(x) + T_{k-1}(x) = 2x T_k(x) \ .$$

Man bestätigt leicht, daß

$$T_k(x) = \cos kt \quad , \quad x = \cos t \quad , \quad 0 \leq t \leq \pi \ .$$

Wir setzen

$$Q(x) = \frac{T_k(\dfrac{\lambda - \lambda_1 - \lambda_n}{\lambda_1 - \lambda_n})}{T_k(-\dfrac{\lambda_1 + \lambda_n}{\lambda_1 - \lambda_n})}$$

Offenbar ist Q ein Polynom vom Grade k mit $Q(0) = 1$, und es ist

$$\max_{\lambda_1 \leq \lambda \leq \lambda_n} |Q(\lambda)| = \frac{1}{|T_k(-\frac{\lambda_1 - \lambda_n}{\lambda_1 - \lambda_n})|}$$

Nun gilt aber für $x \geq 1$

$$T_k(x) \geq \frac{1}{2}(x + \sqrt{x^2 - 1})^k \; . \tag{3.2}$$

Dazu setzen wir $x = \cosh t$, also $t = \operatorname{arccosh} x = \ell n(x + \sqrt{x^2 - 1})$. Aus der Rekursion für T_k folgt

$$\begin{aligned} T_k(x) &= \cosh kt \\ &\geq \frac{1}{2} e^{kt} \\ &= \frac{1}{2}(x + \sqrt{x^2 - 1})^k \end{aligned}$$

und dies ist (3.2). Mit $x = \frac{\lambda_1 + \lambda_n}{\lambda_1 - \lambda_n} = \frac{1+\kappa}{1-\kappa}$, $\kappa = \kappa(A) = \frac{\lambda_n}{\lambda_1}$ haben wir

$$x + \sqrt{x^2 - 1} = \frac{\sqrt{\kappa} + 1}{\sqrt{\kappa} - 1} \; .$$

Damit folgt die Behauptung.

$\square$

4.4 Vorkonditionierung

Die großen linearen Gleichungssysteme mit dünnbesetzter Matrix, für welche iterative Verfahren besonders geeignet sind, haben häufig eine sehr große Kondition. Wir wollen uns dies an dem Beispiel

$$-\Delta u = f \quad \text{in} \quad \Omega = (0,1)^2\ , \quad u = 0 \quad \text{auf}\ \partial\Omega$$

klarmachen. Die diskrete Version aus §1 ist

$$\begin{aligned} 4u_{k,\ell} - u_{k+1,\ell} - u_{k-1,\ell} - u_{k,\ell+1} - u_{k,\ell-1} &= h^2 f_{k,\ell} \\ k,\ell = 1,\ldots,n-1\ , \quad h &= 1/n\ , \\ u_{k,\ell} = 0 \quad \text{falls} \quad k \in \{0,n\} \ \text{ oder }\ \ell &\in \{0,n\}\ . \end{aligned} \tag{4.1}$$

Dabei bedeutet $u_{k,\ell}$ eine Näherung für u in dem Gitterpunkt $(hk, h\ell)$. Wir numerieren die Gitterpunkte und die Gleichungen (4.1) etwa zeilenweise von 1 bis $N = (n-1)^2$. Man sieht dann leicht, daß die Matrix A von (4.1) eine dünnbesetzte symmetrische (N,N)-Matrix ist.

Wir berechnen die Eigenwerte von A. Mit $v_k = \sin\alpha\pi kh$ ist wegen der Additionstheoreme

$$2v_k - v_{k+1} - v_{k-1} = 2(1-\cos\alpha\pi h)v_k\ .$$

Mit $v_{k,\ell} = \sin\alpha\pi kh \sin\beta\pi\ell h$ ist also

$$4v_{k,\ell} - v_{k+1,\ell} - v_{k-1,\ell} - v_{k,\ell+1} - v_{k,\ell-1} = 2(2-\cos\alpha\pi h - \cos\beta\pi h)v_{k,\ell}\ .$$

Wählen wir nun α, β ganz und $1 \le \alpha$, $\beta < n$, so ist $v_{k,\ell} = 0$ für $k \in \{0,n\}$ oder $\ell \in \{0,n\}$ und damit der Vektor $(v_{k,\ell})_{k,\ell=1,\ldots,n-1}$ Eigenvektor von A zum Eigenwert

$$\lambda_{\alpha,\beta} = 2(2-\cos\alpha\pi h - \cos\beta\pi h)\ .$$

Damit haben wir alle N Eigenwerte von A gefunden. Die Kondition $\kappa(A)$ von A bezüglich der euklidischen Norm berechnet sich nun als Quotient von größtem und kleinstem Eigenwert von A, also

$$\kappa(A) = \frac{\lambda_{n-1,n-1}}{\lambda_{1,1}} = \frac{1+\cos h\pi}{1-\cos h\pi}\ .$$

Für kleine h verhält sich dies wie $4/(h\pi)^2$, ist also sehr groß.

Berechnen wir die Konvergenzgeschwindigkeit des Gesamtschrittverfahrens! Die Iterationsmatrix ist $B = -D^{-1}(L+R)$ mit $A = D+L+R$, $D = 4I$. Sei μ ein Eigenwert von B. Dann ist $(1-\mu)/4$ Eigenwert von A, also $\mu = 1-\lambda/4$. Die Eigenwerte von B sind also

$$\mu_{\alpha,\beta} = 1 - \lambda_{\alpha,\beta}/4 = \frac{1}{2}(\cos\alpha\pi h + \cos\beta\pi h) \ .$$

Damit erhalten wir

$$\rho(B) = \max_{1\leq\alpha,\beta<n} |\mu_{\alpha,\beta}| = \cos\pi h \ .$$

Für kleine h verhält sich dies wie $1-\pi^2h^2/2$. Die Konvergenz des Gesamtschrittverfahrens ist also sehr langsam.

Gesamtschritt- und CG-Verfahren konvergieren also beide langsam, doch mit einem bedeutenden Unterschied: Während sich der Fehler beim Gesamtschrittverfahren wie $(1-\pi^2h^2/2)^k$ verhält, verhält er sich beim CG-Verfahren (allerdings in einer anderen Norm) wie $(1-h\pi)^k$. Das CG-Verfahren konvergiert also weniger langsam als das Gesamtschrittverfahren. Man kann zeigen, daß das SOR-Verfahren bei optimaler Wahl von ω ungefähr so schnell wie das CG-Verfahren konvergiert. Insgesamt muß man also bei allen diesen Verfahren mit langsamer Konvergenz rechnen, und zwar wird die Konvergenz umso langsamer, je kleiner h wird.

Abhilfe schafft hier die Vorkonditionierung. Dazu schreiben wir $Ax = b$ in der Form

$$C^{-1}A(C^*)^{-1}C^*x = C^{-1}b \tag{4.2}$$

mit einer nichtsingulären Matrix C. C wird so gewählt, daß

$$\kappa(C^{-1}A(C^*)^{-1}) \ll \kappa(A) \ .$$

(4.2) hat dann die Gestalt

$$\tilde{A}\tilde{x} = \tilde{b} \ ,$$

$$\tilde{A} = C^{-1}A(C^*)^{-1} \ , \quad \tilde{x} = C^*x \ , \quad \tilde{b} = C^{-1}b \ ,$$

wobei $\tilde{A}$ eine viel bessere Kondition hat als A. Überdies ist $\tilde{A}$ hermitesch (positiv definit), falls dies für A der Fall ist.

Wegen

$$\begin{aligned} (C^{-1})^* \tilde{A} C^* &= (C^{-1})^* C^{-1} A (C^*)^{-1} C^* = M^{-1} A \,, \\ M &= CC^* \end{aligned}$$

ist $\kappa(\tilde{A}) = \kappa(M^{-1}A)$. Um für $\tilde{A}$ eine gute Kondition zu bekommen, wird man also $M \sim A$ wählen wollen. Gleichzeitig muß M aber eine Cholesky-Faktorisierung $M = CC^*$ besitzen, so daß C^{-1}, $(C^*)^{-1}$ leicht auf Vektoren anwendbar ist. Für $A = I + L + L^*$ ist

$$M = (I + L)(I + L^*)$$

eine Möglichkeit.

Andere Möglichkeiten sind Dekompositionsmethoden und Mehrgitterverfahren. Diese führen zu von h unabhängigen Konvergenzraten. Sie sind Gegenstand weiterführender Vorlesungen, wie etwa “Numerik partieller Differentialgleichungen”.

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

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

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

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

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 mehr als $n+1$ paarweise verschiedene Nullstellen haben kann.

□

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

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

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 $p_{i,\ldots,k}$ 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*

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

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 ($n=3$):

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

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

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

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

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

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:

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

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 $m = k - i$. Für $m = 0$ ist (1.2) richtig. Sei (1.2) richtig für ein $m \geq 0$. 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 $p_{i,\ldots,k+1}$.

$\square$

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:

$$
\begin{array}{ccccc}
0 & 1 & & & \\
 & & 2 & & \\
1 & 3 & & & -5/6 \\
 & & -1/2 & & \\
3 & 2 & & &
\end{array}
$$

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

$$|f(x) - p(x)| \le |w(x)| \max_{x \in [a,b]} \frac{|f^{(n+1)}(x)|}{(n+1)!} \; .$$

1) Wir betrachten zuerst den Fall gleichmäßig verteilter Stützstellen $x_j = a + jh$ mit der "Schrittweite" $h = \frac{b-a}{n}$. Für $x = a + \theta h$, $0 \le \theta \le n$ gilt

$$w(x) = h^{n+1} \prod_{j=0}^{n} (\theta - j) \; ,$$

also

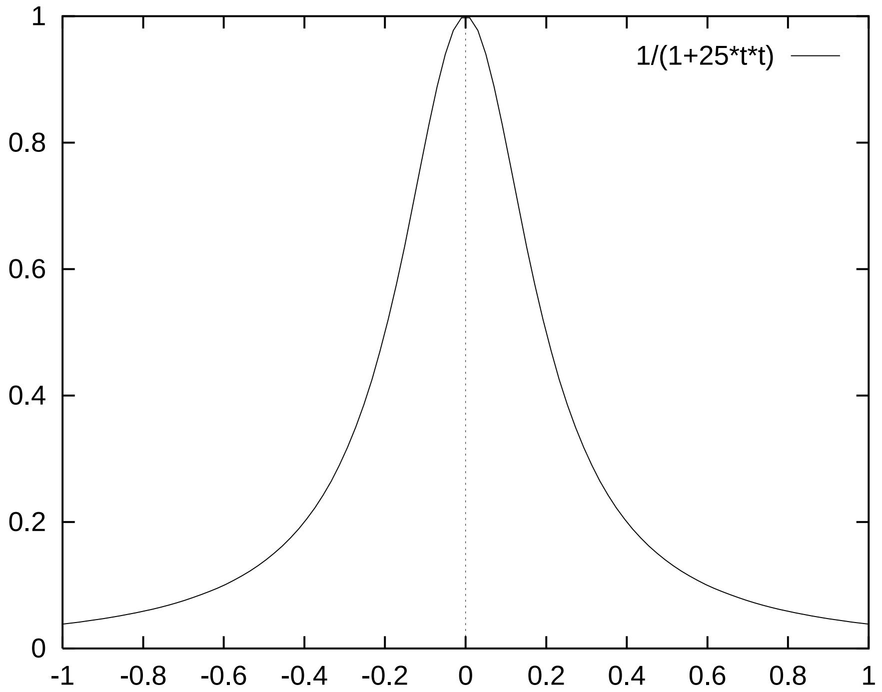
$$|f(x) - p(x)| \le \frac{h^{n+1}}{(n+1)!} \left(\prod_{j=0}^{n} |\theta - j|\right) \max_{x \in [a,b]} |f^{(n+1)}(x)| \; .$$

(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 $\max_{[a,b]} \lvert w(x) \rvert$ 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:

$$T_n(x) = \cos nt \quad , \quad x = \cos t \quad , \quad 0 \leq t \leq \pi \; .$$

In IV.3 haben wir gesehen, daß

$$T_{n+1} = 2xT_n - T_{n-1} .$$

Die Rekursion zeigt, daß T_n für $n \geq 1$ 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

$$|w(x)| \leq 2^{-n} \quad \text{in } [-1, 1] .$$

Die Nullstellen $x_j = \cos\frac{(j+1/2)\pi}{n}$, $j = 0, \ldots, n$ von w sind unsere Stützstellen. Bei dieser Wahl ergibt sich die Fehlerabschätzung

$$|f(x) - p(x)| \leq \frac{\text{Max}|f^{(n+1)}(x)|}{2^n(n+1)!} .$$

Für das obige Beispiel ergeben sich folgende Werte:

n	$\max_{x\in[-1,1]} \lvert f(x) - p(x)\rvert$
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

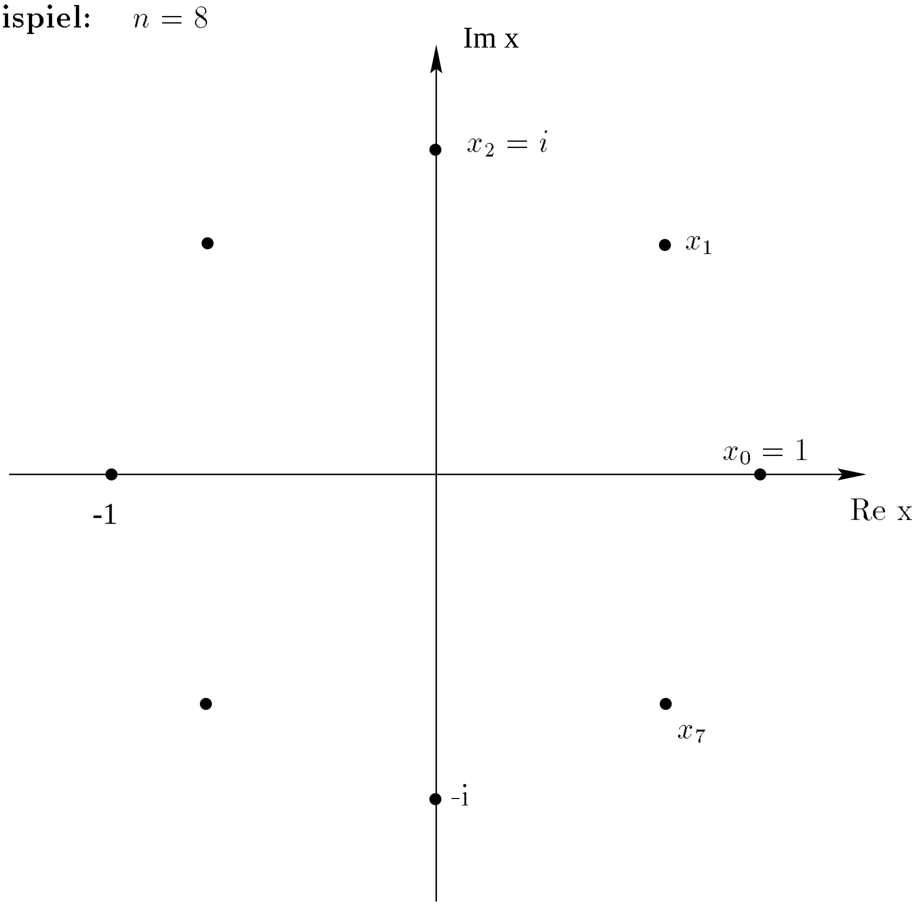
Wir betrachten in diesem Abschnitt einen wichtigen Spezialfall der Polynominterpolation, bei dem die Stützstellen in regelmäßigen Abständen auf dem komplexen Einheitskreis liegen.

Diese spezielle Problemstellung wird zu einem der wichtigsten Hilfsmittel der angewandten Mathematik führen: Zur diskreten Fouriertransformation.

Die Stützstellen seien gegeben durch

$$x_j = e^{it_j} = \cos t_j + i \sin t_j \ ; \quad t_j = 2\pi j/n \ ; \quad j = 0, \ldots, n-1 \ .$$

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

$$y = \begin{pmatrix} 1 & x_0 & \ldots & x^{n-1} \\ 1 & x_1 & \ldots & x_1^{n-1} \\ \vdots & & \ddots & \\ 1 & x_{n-1} & \ldots & x_{n-1}^{n-1} \end{pmatrix} \cdot \hat{y} = W\hat{y} \ .$$

umschreiben. Die Inversion der Matrix W erweist sich als sehr einfach:

Satz 5.3.1 *Mit obigen Definitionen für x_j, $j = 0, \ldots, n-1$ und die Matrix W gilt:*

$$WW^* = nI \ .$$

Beweis: Für k, $\ell = 0, \ldots, n-1$ gilt:

$$\begin{aligned}
(WW^*)_{k\ell} &= \sum_{j=0}^{n-1} x_k^j \overline{x}_\ell^j \\
&= \sum_{j=0}^{n-1} e^{i(t_k - t_\ell)j} \\
&= \sum_{j=0}^{n-1} e^{2\pi i (k-\ell)j/n} \\
&= \sum_{j=0}^{n-1} q^j \qquad \text{mit} \ \ q = e^{2\pi i (k-\ell)/n}
\end{aligned}$$

$$= \begin{cases} \frac{q^n-1}{q-1} & , \text{ falls } q \neq 1 \\ n & , \text{ falls } q = 1 \end{cases}$$

$$= \begin{cases} 0 & , \text{ falls } k \neq \ell \\ n & , \text{ falls } k = \ell \end{cases}$$

□

Damit haben wir gleichzeitig eine Orthogonalitätseigenschaft der trigonometrischen Funktionen gezeigt:

$$\frac{1}{n} \sum_{j=0}^{n-1} e^{2\pi ijk/n} = \begin{cases} 1 & , \text{ falls } k \in n\mathbb{Z} \\ 0 & , \text{ sonst} \end{cases}$$

Wegen $(WW^*)_{k,\ell} = (W^*W)_{k,\ell}$, $k, \ell = 0, \ldots, n-1$, können wir folgern:

$$W^{-1} = \frac{1}{n} W^*$$

und damit

$$\hat{y} = \frac{1}{n} W^* y \ .$$

In Komponenten:

$$\boxed{\begin{aligned} \hat{y}_k &= \tfrac{1}{n} \sum_{j=0}^{n-1} e^{-2\pi ijk/n} y_j \ , && (1) \\ y_j &= \sum_{k=0}^{n-1} e^{2\pi ijk/n} \hat{y}_k \quad , && (2) \end{aligned}}$$

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.

Zunächst stellen wir jedoch die Beziehung zum Titel dieses Paragraphen her und drücken $p(x)$ durch trigonometrische Funktionen aus:

Wir setzen

$$\begin{aligned} a_k &= \frac{2}{n} \sum_{j=0}^{n-1} y_j \cos(2\pi kj/n) \; , \\ b_k &= \frac{2}{n} \sum_{j=0}^{n-1} y_j \sin(2\pi kj/n) \; . \end{aligned}$$

Dann ist

$$\begin{aligned} \hat{y}_k &= \frac{1}{n} \sum_{j=0}^{n-1} y_j e^{-2\pi ijk/n} \\ &= \frac{1}{n} \sum_{j=0}^{n-1} y_j (\cos(2\pi jk/n) - i \sin(2\pi jk/n)) \\ &= \frac{1}{2}(a_k - ib_k) \; . \end{aligned}$$

Man beachte, daß dies keine Zerlegung in Real- und Imaginärteil von $\hat{y}_k$ darstellt, da die a_k und b_k nicht notwendig reell sind. Es gilt:

$$a_{n-k} = a_k \quad \text{und} \quad b_{n-k} = -b_k$$

und damit

$$\hat{y}_{n-k} = \frac{1}{2}(a_k + ib_k) \; .$$

Sei n nun ungerade, also $n = 2m + 1$. Dann erhalten wir

$$\begin{aligned} p(x_j) &= \sum_{k=0}^{n-1} \hat{y}_k e^{2\pi ijk/n} \\ &= \hat{y}_0 + \sum_{k=1}^{m} \hat{y}_k e^{2\pi ijk/n} + \sum_{k=m+1}^{n-1} \hat{y}_k e^{2\pi ijk/n} \\ &= \hat{y}_0 + \sum_{k=1}^{m} \hat{y}_k e^{2\pi ijk/n} + \sum_{k'=1}^{m} \hat{y}_{n-k'} e^{2\pi ij(n-k')/n} \quad ; \quad k' = n - k \end{aligned}$$

$$
\begin{aligned}
&= \hat{y}_0 + \frac{1}{2}\sum_{k=1}^{m}\{(a_k - ib_k)(\cos(2\pi jk/n) + i\sin(2\pi jk/n)) \\
&\qquad +(a_k + ib_k)(\cos(2\pi jk/n) - i\sin(2\pi jk/n))\} \\
\Rightarrow p(x_j) = y_j &= \frac{a_0}{2} + \sum_{k=1}^{m}(a_k\cos(2\pi jk/n) + b_k\sin(2\pi jk/n))\ . \qquad (\star)
\end{aligned}
$$

Für gerades n, $n = 2m$, erhält man analog

$$
\begin{aligned}
p(x_j) = y_j = \frac{a_0}{2} &+ \sum_{k=1}^{m-1}(a_k\cos(2\pi jk/n) + b_k\sin(2\pi jk/n)) \\
&+ \frac{a_0}{2}\cos(2\pi jm/n)\ , \qquad (\star\star)
\end{aligned}
$$

wenn man beachtet, daß b_m verschwindet.

Wir bezeichnen die Menge

$$
\mathcal{T}_m = \left\{\frac{a_0}{2} + \sum_{k=1}^{m}(a_k\cos kt + b_k\sin kt)\ ;\ a_k, b_k \in \mathbb{C}\right\}
$$

als die Menge der trigonometrischen Polynome vom Grad m.

/$\star$ vgl. Aufgabe 38 $\star$/

Damit haben wir die folgende Aussage über die trigonometrische Interpolation gewonnen:

Satz 5.3.2 *Gesucht sei $T \in \mathcal{T}_m$ mit*

$$
\begin{aligned}
T(t_j) &= y_j \quad , \quad j = 0, \ldots, n-1 \\
t_j &= 2\pi j/n\ .
\end{aligned}
$$

Für $n = 2m+1$ ist die Aufgabe eindeutig lösbar und die Lösung ist durch ($\star$) gegeben.

Für $n = 2m$ gibt es genau eine Lösung mit $b_n = 0$, die durch ($\star\star$) gegeben ist.

5.3.1 Algorithmus von Goertzel und Reinsch

Für die trigonometrische Interpolation und besonders für die diskrete Fouriertransformation ist es wichtig, effiziente Algorithmen zur Auswertung von Ausdrücken der Form

$$c = \sum_{k=0}^{n-1} y_k \cos k\varphi \quad \text{und} \quad s = \sum_{k=0}^{n-1} y_k \sin k\varphi$$

zu kennen.

Goertzel hat hierfür 1958 ein Verfahren vorgeschlagen, das im folgenden Satz dargestellt wird:

Satz 5.3.3 *Für $\varphi \notin \pi\mathbb{Z}$, $n \in \mathbb{N}$ sei*

$$\begin{aligned} u_j &= \frac{1}{\sin\varphi} \sum_{k=j}^{n-1} y_k \sin(k-j+1)\varphi \quad , \quad j = 0, \ldots, n-1 \\ u_n &= u_{n+1} = 0 \; . \end{aligned}$$

Dann gilt:

(a) $\quad u_j = y_j + 2\cos\varphi\, u_{j+1} - u_{j+2} \quad , \quad j = n-1, n-2, \ldots, 0$

(b) $\quad \sum\limits_{k=0}^{n-1} y_k \sin k\varphi = u_1 \sin\varphi$

(c) $\quad \sum\limits_{k=0}^{n-1} y_k \cos k\varphi = u_1 \cos\varphi\, u_1 - u_2$

Beweis: Zu (a): Das Additionstheorem für den Sinus

$$\sin(\alpha+\beta) = \sin\alpha\cos\beta + \cos\alpha\sin\beta$$

liefert uns die Formel

$$\sin(k-j\pm 1)\varphi = \sin(k-j)\varphi\cos\varphi \pm \cos(k-j)\varphi\sin\varphi \; .$$

Hiermit können wir (a) durch einfaches Nachrechnen zeigen:

$$\begin{aligned}
&y_j + 2\cos\varphi\, u_{j+1} - u_{j+2} \\
&= y_j + \frac{1}{\sin\varphi}\left\{2\cos\varphi \sum_{k=j+1}^{n-1} y_k \sin(k-j)\varphi - \sum_{k=j+2}^{n-1} y_k \sin(k-j-1)\varphi\right\} \\
&= y_j + \frac{1}{\sin\varphi}\sum_{k=j+1}^{n-1} y_k \left\{2\cos\varphi\sin(k-j)\varphi - \sin(k-j-1)\varphi\right\} \\
&= y_j + \frac{1}{\sin\varphi}\sum_{k=j+1}^{n-1} y_k \left\{2\cos\varphi\sin(k-j)\varphi - (\sin(k-j)\varphi\cos\varphi - \cos(k-j)\varphi\sin\varphi)\right\} \\
&= y_j + \frac{1}{\sin\varphi}\sum_{k=j+1}^{n-1} y_k \left\{\cos\varphi\sin(k-j)\varphi + \sin\varphi\cos(k-j)\varphi\right\} \\
&= \frac{1}{\sin\varphi}\left\{y_j\sin\varphi + \sum_{k=j+1}^{n-1} y_k \sin(k-j+1)\varphi\right\} \\
&= \frac{1}{\sin\varphi}\sum_{k=j}^{n-1} y_k \sin(k-j+1)\varphi \\
&= u_j
\end{aligned}$$

(b) Dies folgt direkt aus der Definition von u_1.

(c) Übungsaufgabe 36.

□

Pseudo - Programm:

goertzel (φ, y, n, s, c)
$\{u_n = u_{n+1} = 0;$
  $\gamma = \cos\varphi;$
  for $j = n-1,\, n-2, \ldots, 1$
    $u_j = y_j + 2\gamma u_{j+1} - u_{j+2};$
  $c = y_0 + u_1\gamma - u_2;$
  $s = u_1 \sin\varphi;$
$\}$

Zahlenbeispiel: $n = 1001$, $\varphi = \frac{\pi}{2(n-1)}$, $y_j = \begin{cases} 1 & , \quad j = n-1 \\ 0 & , \quad \text{sonst} . \end{cases}$

Korrekt: $c = \cos\frac{\pi}{2} = 0$, $s = \sin\frac{\pi}{2} = 1$

Sun 4 mit float-Zahlen: $\tilde{c}_G = -0.012144$, $\tilde{s}_G = 0.992777$

Das ist nicht sehr befriedigend.

Das wesentliche Problem steckt hier in der Verwendung von $\gamma = \cos\varphi$. Wir berechnen c als Funktion von γ. Es zeigt sich, daß c bei kleinem φ empfindlich auf kleine Fehler in γ reagiert:

Sei $\varphi > 0$, ε bezeichne die Maschinengenauigkeit.

$$
\begin{aligned}
c &= \sum_{k=0}^{n-1} y_k \cos k\varphi = \sum_{k=0}^{n-1} y_k \cos(k \arccos\gamma) \\
\frac{\partial c}{\partial \gamma} &= \sum_{k=0}^{n-1} y_k(-\sin(k \cdot \arccos\gamma)) \cdot k \cdot \left(-\frac{1}{\sqrt{1-\gamma^2}}\right) \\
&\quad \frac{1}{\sqrt{1-\gamma^2}} \sum_{k=0}^{n-1} k y_k \sin(k \cdot \arccos\gamma) \\
&\quad \frac{1}{\sin\varphi} \sum_{k=0}^{n-1} k y_k \sin(k\varphi) = \frac{1}{\sin\varphi}\left(-\frac{\partial c}{\partial \varphi}\right) \\
\Rightarrow \Delta c &= \left|\frac{\partial c}{\partial \gamma} \cdot \Delta\gamma\right| = \left|\frac{1}{\sin\varphi}\,\frac{\partial c}{\partial \varphi} \cdot \varepsilon \cdot \cos\varphi\right| = \left|\varepsilon \cdot cot\varphi \frac{\partial c}{\partial \varphi}\right|
\end{aligned}
$$

Bei kleinem φ führen Ungenauigkeiten in γ wegen dem Faktor $cot\varphi$ zu viel größeren Fehlern, als Ungenauigkeiten in φ. Damit haben wir offenbar einen unnötig instabilen Weg gewählt, um c zu berechnen.

Eine Verbesserung bringt die Variante von Reinsch:

Wir beschränken uns auf den Fall $\cos\varphi \geq 0$. Als Hilfsgröße wird

$$w_j := u_j - u_{j+1}$$

eingeführt. Die Rekursionsformel aus Satz 3 kann dann wie folgt umgeschrieben werden:

$$\begin{aligned}
u_j &= y_j + 2\cos\varphi\, u_{j+1} - u_{j+2} \\
\Leftrightarrow u_j - u_{j+1} &= y_j + \underbrace{2(\cos\varphi - 1)}_{=:\lambda} u_{j+1} + u_{j+1} - u_{j+2} \\
\Leftrightarrow w_j &= y_j + \lambda u_{j+1} + w_{j+1}\,.
\end{aligned}$$

Mit

$$\lambda = 2(\cos\varphi - 1) = -4\sin^2\frac{\varphi}{2}$$

und

$$c = y_0 + \cos\varphi u_1 - u_2 = \ldots = w_0 - \frac{\lambda}{2} w_1$$

erhalten wir den folgenden Algorithmus:

reinsch (φ, n, y, s, c)
{ $u_{n+1} = w_n = 0$;
  $\lambda = -4\sin^2\frac{\varphi}{2}$;
  for $j = n-1, \ldots, 0$
    { $u_{j+1} = w_{j+1} + u_{j+2}$;
      $w_j = y_j + \lambda u_{j+1} + w_{j+1}$;
    }
  $c = w_0 - \frac{\lambda}{2} w_1$;
  $s = u_1 \sin\varphi$;
}

Obiges Zahlenbeispiel:

$$\tilde{c}_R = 0.000785 \quad , \quad \tilde{s}_R = 1.000000$$

Der Algorithmus von Reinsch scheint bessere Resultate zu liefern.

Wir führen dieselbe Fehlerrechnung wie oben durch. Hierbei können wir wegen $\lambda = 2(\gamma - 1)$ die Kettenregel anwenden und das obige Ergebnis benutzten:

$$\frac{\partial c}{\partial \lambda} = \frac{\partial c}{\partial \gamma} \cdot \underbrace{\frac{\partial \gamma}{\partial \lambda}}_{=\frac{1}{2}} = \frac{1}{2} \frac{1}{\sin\varphi} \left(-\frac{\partial c}{\partial \varphi} \right)$$

Dieses Ergebnis zeigt noch keine Verbesserung gegenüber dem Algorithmus von Goertzel. Wir berechnen den Fehler und benutzen $\sin x = 2\sin\frac{x}{2}\cos\frac{x}{2}$:

$$\begin{aligned}
\Delta c &= \left|\Delta\lambda\cdot\frac{\partial c}{\partial\lambda}\right| = \left|\varepsilon 4\sin^2\frac{\varphi}{2}\cdot\frac{1}{2}\frac{1}{2\sin\frac{\varphi}{2}\cos\frac{\varphi}{2}}\left(-\frac{\partial c}{\partial\varphi}\right)\right| \\
&= \left|\varepsilon\cdot\tan\frac{\varphi}{2}\frac{\partial c}{\partial\varphi}\right|
\end{aligned}$$

Für kleines φ ist auch $\tan\frac{\varphi}{2}$ klein. Der Vorteil ergibt sich in diesem Fall nur aus dem Unterschied zwischen $\Delta\gamma$ und $\Delta\lambda$, er ist also nur darin begründet, daß wir bei kleinem φ den Wert von $\sin\varphi$ genau berechnen und v.a. speichern können, als den Wert $\cos\varphi$.

Der Vorteil des Verfahrens von Reinsch geht verloren, wenn wir φ größer wählen. Bei $\varphi = \frac{\pi}{16}$ erhalten wir ($n = 1001$, $c = 0$, $s = 1$)

$$\begin{array}{lcrclcrcr}
\text{Goertzel} & : & \tilde{s}_G & = & 1,000001 & , & c_G & = & -0,000147 \\
\text{Reinsch} & : & \tilde{s}_R & = & 1,000001 & , & c_R & = & 0,096608
\end{array}$$

Für die weiteren Anwendungen ist der Fall $\varphi \ll 1$ aber wichtiger.

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 sind $q^n = 1$ und $q^m = q^{n/2} = -1$.

Die Idee ist nun, die Summe nach geraden und ungeraden Indizes zu zerlegen:

$$\hat{y}_k = \sum_{\ell=0}^{m-1} q^{(2\ell)k} y_{2\ell} + \sum_{\ell=0}^{m-1} q^{(2\ell+1)k} y_{2\ell+1}$$

$$
\begin{aligned}
&= \sum_{\ell=0}^{m-1} (q^2)^{\ell k} y_{2\ell} + q^k \sum_{\ell=0}^{m-1} (q^2)^{\ell k} y_{2\ell+1} \\
&=: g_k + q^k u_k
\end{aligned}
$$

Wir sehen nun, daß sich g_k und u_k als Fouriertransformationen der Länge $m = n/2$ berechnen, da gilt:

$$q^2 = e^{-2\pi i/(n/2)}$$

Weiter erhalten wir

$$
\begin{aligned}
g_{k+m} &= g_k \\
u_{k+m} &= u_k
\end{aligned}
$$

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

$$
\begin{aligned}
M_{p+1} &= 2M_p + 2^p \ , \\
A_{p+1} &= 2A_p + 2^{p+1} \ ,
\end{aligned}
$$

also

$$
\begin{aligned}
M_p &= \tfrac{1}{2} p 2^p = \tfrac{1}{2}(\log_2 n) \cdot n \\
A_p &= p 2^p = (\log_2 n) \cdot n \ .
\end{aligned}
$$

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.*

Pseudoprogramm:

```
fft (y, n)
{ 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:

$$
\begin{array}{rcrcl}
n=1 & : & \hat{y}_0 & = & y_0 \\
n=2 & : & \hat{y}_0 & = & y_0 + y_1 \\
 & & \hat{y}_1 & = & y_0 - y_1 \\
n=4 & : & q & = & p^{-2\pi i/4} = -i \\
 & & g_0 & = & y_0 + y_2 \qquad u_0 = y_1 + y_3 \\
 & & g_1 & = & y_0 - y_2 \qquad u_1 = y_1 - y_3 \\
 & & \hat{y}_0 & = & g_0 + u_0 \\
 & & \hat{y}_1 & = & g_1 - iu_1 \\
 & & \hat{y}_2 & = & g_0 - u_0 \\
 & & \hat{y}_3 & = & g_1 + iu_1
\end{array}
$$

5.5 Differenzengleichungen

Eine lineare Differentialgleichung der Ordnung m mit konstanten Koeffizienten lautet

$$\sum_{\nu=0}^{m} u_{k+\nu}\alpha_\nu = f_k \ , \qquad k = 0, 1, \ldots \tag{5.1}$$

Beispiele:

1) Die Formel von Goertzel (Satz 3.3), also

$$u_k = y_k + 2u_{k+1} \cos\xi - u_{k+2} \ .$$

2) Für große k kann $C_k = \cos kx$ durch die Rekursion

$$C_{k+1} + C_{k-1} = 2\cos xC_k \ , \quad C_0 = 1 \ , \quad C_1 = \cos x$$

berechnet werden.

3) Entsprechendes gilt für die Tschebyscheff-Polynome $T_k(x)$,

$$T_{k+1}(x) + T_{k-1}(x) = 2xT_k(x) \ , \quad T_0 = 1 \ , \quad T_1 = x \ .$$

4) Ebenfalls linear, aber nicht mit konstanten Koeffizienten ist die Differenzengleichung

$$J_{k+1}(x) + J_{k-1}(x) = \frac{2k}{x} J_k(x)$$

für die Bessel-Funktion $J_k(x)$ 1. Art der Ordnung k.

Die Differenzengleichung (5.1) heißt homogen, falls $f_k = 0$ ist für $k = 0, 1, \ldots,$ andernfalls inhomogen. Der homogenen Gleichung ordnen wir das charakteristische Polynom

$$p(\lambda) = \sum_{\nu=0}^{m} \alpha_\nu \lambda^\nu$$

zu.

Satz 5.5.1 *p habe die Nullstellen $\lambda_1, \ldots, \lambda_r$ mit den Vielfachheiten $\sigma_1, \ldots, \sigma_r$, $\sigma_1 + \cdots + \sigma_r = m$. Dann ist*

$$u_k = k^\sigma \lambda_\ell^k \ , \qquad 0 \le \sigma < \sigma_\ell \ , \quad \ell = 1, \ldots, r \tag{5.2}$$

Lösung der homogenen Gleichung (5.1), und jede Lösung der homogenen Gleichung (5.1) ist eine Linearkombination solcher Lösungen.

Beweis: Für $\sigma = 0$ ist

$$\begin{aligned} \sum_{\nu=0}^{m} u_{k\nu}\alpha_\nu &= \sum_{\nu=0}^{m} \lambda_\ell^{k+\nu}\alpha_\nu = \lambda_\ell^k \sum_{\nu=0}^{n} \lambda_\ell^\nu \alpha_\nu \\ &= \lambda_\ell^k p(\lambda_\ell) = 0 \ . \end{aligned}$$

Ist λ_ℓ eine σ_ℓ-fache Wurzel von p, so ist für $0 \le \sigma < \sigma_\ell$

$$\left(\frac{d}{d\lambda}\right)^\sigma p(\lambda)\Big|_{\lambda=\lambda_\ell} = \sum_{\nu=0}^{m} \nu(\nu-1)\cdots(\nu-\sigma+1)\lambda_\ell^{\nu-\sigma}\alpha_\nu = 0 \ .$$

Es folgt

$$\sum_{\nu=0}^{m} \nu^\sigma \lambda_\ell^\nu \alpha_\nu = 0 \ , \quad \sigma = 0, \ldots, \sigma_\ell - 1$$

und damit

$$\sum_{\nu=0}^{m} u_{k+\nu}\alpha_\nu = \sum_{\nu=0}^{m} (k+\nu)^\sigma \lambda_\ell^{k+\nu}\alpha_\nu = 0 \ , \quad \sigma = 0, \ldots, \sigma_\ell - 1 \ .$$

Also sind (5.2) tatsächlich Lösungen der homogenen Gleichung (5.1).

Sei umgekehrt u_k eine beliebige Lösung der homogenen Gleichung (5.1). Wir versuchen, Zahlen $C_{\ell,\sigma}$, $0 \le \sigma < \sigma_\ell$ so zu bestimmen, daß

$$u_k = \sum_{\ell=1}^{r} \sum_{\sigma=0}^{\sigma_\ell - 1} C_{\ell,\sigma} k^\sigma \lambda_\ell^k \ , \quad k = 0, \ldots, m-1 \ . \tag{5.3}$$

Dies ist ein System von m linearen Gleichungen für die m Zahlen $C_{\ell,\sigma}$. Für $\sigma_\ell = 1$, $\ell = 1, \ldots, r$ ist die Determinante gerade die Van der Monde'sche Determinante der paarweise verschiedenen Zahlen $\lambda_1, \ldots, \lambda_r$ $(r = m)$ und damit von Null verschieden. Für den allgemeinen Fall kann man dies auch zeigen. Damit gilt (5.3). Da eine Lösung von (5.1) durch ihre Werte

$u_0, \ldots, u_{m-1}$ eindeutig bestimmt ist, ist u_k tatsächlich eine Linearkombination der Lösungen (5.2).

□

Man nennt eine Differenzengleichung stabil, wenn es eine Konstante C gibt, so daß für jede Lösung der homogenen Gleichung

$$|u_k| \leq C \max_{0 \leq \ell < m} |u_\ell|$$

für alle $k \geq 0$ gilt. Sind dann für eine Lösung u_k der inhomogenen Gleichung $u_0, \ldots, u_{m-1}$ mit dem Fehler ε bekannt, so kann u_k für $k \geq m$ mit dem Fehler $\leq C\varepsilon$ berechnet worden.

Satz 5.5.2 *Die Differenzengleichung (5.1) ist genau dann stabil, wenn für* $\ell = 1, \ldots, r$ *gilt:*

$$|\lambda_\ell| < 1 \quad \textit{oder} \quad |\lambda_\ell| = 1 \quad \textit{und} \quad \sigma_\ell = 1 \ .$$

Beweis: Folgt sofort aus Satz 5.1.

□

Beispiele:

1) Bei der Formel von Goertzel ist $p(\lambda) = 1 - 2\lambda\cos\xi + \lambda^2$, also $\lambda_{1,2} = \cos\xi \pm i\sin\xi$. Die Rekursion ist stabil für $\xi \neq 0$, instabil für $\xi = 0$. Man hat Probleme für $\xi \sim 0$.

2) Es gibt Probleme bei $x = 0$.

3) Es gibt Probleme bei $|x| \geq 1$.

4) Hier ist die Theorie nicht streng anwendbar, da die Differenzengleichung nicht konstante Koeffizienten hat. Aus dem Vergleich mit 3) vermutet man aber, daß man Probleme hat für $|k/x| \geq 1$. In diesem Fall verwendet man dann die Rekursion einfach für absteigende Werte von k und benutzt als Startwerte asymptotische Näherungen von J_k für große k.

5.6 Harmonische Analyse

Sei f 2π-periodisch. Solche Funktionen kann man in eine Fourier-Reihe entwickeln:

Satz 5.6.1 *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 $a_k = 0$ 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 werden wir sehen, 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.

Wie in §3 gehen wir 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

$$\begin{array}{lcl} d_{n,k} \sim 1 & , & |k| \ll n \, , \\ d_{n,k} \sim 0 & , & |k| \sim n \, . \end{array}$$

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.

5.7 Splines

Ein Spline (spline (engl.) = Straklatte) ist ein dünner elastischer Stab, der zwischen Gewichte ("Knoten") gespannt Kurven darstellt, wie sie im Schiffsbau verwendet wurden. 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 5.7.1 *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

$$B_{i,k}(t) = (x_{i+k} - x_i)[f_i, \ldots, f_{i+k}]\ . \tag{7.1}$$

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) $x_i \leq t < x_{i+1}$. 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) $x_{i+1} \leq t < x_{i+2}$. 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) $x_{i+2} \leq t$. 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 5.7.1 *$B_{i,k}$ ist ein Spline der Ordnung k, der außerhalb $[x_i, x_{i+k}]$ verschwindet.*

Beweis: f_j ist für $t \neq x_j$ ein Polynom vom Grade $\leq k-1$ 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 $\leq k-1$ 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 $B_{i,k}(t) = 0$.

□

Lemma 5.7.1 *(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. Das Newton'sche Interpolationspolynom für g, h lautet

$$\begin{aligned} g(x) &= [g_i] + [g_i, g_{i+1}](x - x_i) + \cdots [g_i, \ldots, g_{i+k}](x - x_i) \cdots (x - x_{i+k-1}) \\ &= \sum_{r=i}^{i+k} [g_i, \ldots, g_r](x - x_i) \cdots (x - x_{r-1}) , \\ h(x) &= [h_{i+k}] + [h_{i+k}, h_{i+k-1}](x - x_{i+k}) + \cdots \\ &\qquad + [h_{i+k}, \ldots, h_i](x - x_{i+k}) \cdots (x - x_{i+1}) \\ &= \sum_{s=i}^{i+k} [h_s, \ldots, h_{i+k}](x - x_{i+k}) \cdots (x - x_{s+1}) . \end{aligned}$$

Multiplikation ergibt

$$f(x) = (gh)(x) = \sum_{r,s=i}^{i+k} [g_i, \ldots, g_r][h_s, \ldots, h_{i+k}](x-x_i) \cdots (x-x_{r-1})(x-x_{s+1}) \cdots (x-x_{i+k}) . \tag{7.2}$$

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 5.7.2 *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 (7.1) 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 $f_j = g_j h_j, \quad j = i, \ldots, i+k$ und damit nach Leibniz

$$B_{i,k}(x) = (x_{i+k} - x_i) \sum_{r=i}^{i+k} [g_i, \ldots, g_r][h_r, \ldots, h_{i+k}] \ .$$

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

$$B_{i,1}(x) = \begin{cases} 1 & , \quad x_i \leq x < x_{i+1} \ , \\ 0 & , \quad \text{sonst} \end{cases}$$

ermöglicht Satz 3 die rekursive Berechnung der $B_{i,k}$.

2) Diese Rekursion ist numerisch stabil, da nur Linearkombination positiver Zahlen gebildet werden.

Satz 5.7.3 *Für $k \geq 3$ 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

$$\begin{aligned} B'_{i,k}(x) &= -(k-1)\{[g_{i+1}, \ldots, g_{i+k}] - [g_i, \ldots, g_{i+k-1}]\} \\ &= (k-1)\left\{\frac{B_{i,k-1}(x)}{x_{i+k-1} - x_i} - \frac{B_{i+1,k-1}(x)}{x_{i+k} - x_{i+1}}\right\} . \end{aligned}$$

□

5.8 Interpolation mit Splines

Sei $x_0 < x_1 < \cdots x_n$ und

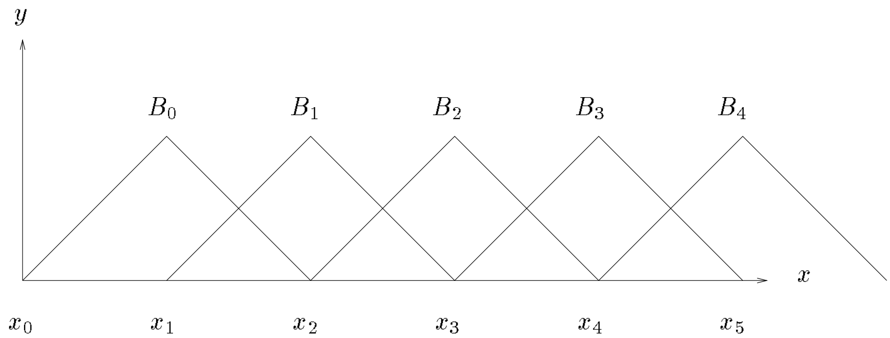
$$s(t) = \sum_{i=0}^{n-k} a_i B_{i,k}(t)\ .$$

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 5.8.1 *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

$$\begin{pmatrix} B_0(t_0) & B_1(t_0) & 0 & 0 \\ B_0(t_1) & B_1(t_1) & B_2(t_1) & 0 \\ 0 & B_1(t_2) & B_2(t_2) & B_3(t_2) \\ 0 & 0 & B_2(t_3) & B_3(t_3) \end{pmatrix} \qquad (n = 5)$$

invertierbar ist. Wir zeigen, daß für $n \geq 3$ 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. Die Matrix zerfällt dann in zwei kleinere Teilmatrizen, im Falle

$$\begin{pmatrix} B_0(t_0) & B_1(t_0) \\ B_0(t_1) & B_1(t_1) \end{pmatrix} \quad , \quad \begin{pmatrix} B_2(t_2) & B_3(t_2) \\ B_2(t_3) & B_3(t_3) \end{pmatrix} ,$$

im Falle $B_1(t_0) = 0$ in

$$B_1(t_0) \quad , \quad \begin{pmatrix} B_1(t_1) & B_2(t_1) & 0 \\ B_1(t_2) & B_2(t_2) & B_3(t_2) \\ 0 & B_2(t_3) & B_3(t_3) \end{pmatrix} .$$

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. Dies ist aber leicht zu sehen.

Sei nun die Behauptung richtig bis zur Ordnung $< k$ für ein $k \geq 2$. 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 O. 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 O wären. In beiden Fällen zerfiele die Matrix in kleinere, und wir könnten uns auf den trivialen Fall $n = k$ beschränken. 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^{n-k}-Funktion mit den $n-k+3$ Nullstellen $t_0, \ldots, t_{n-i,k}, \; x_0, \, x_n$. Nach dem Satz von Rolle hätte $s \;\; n-k+2$ Nullstellen

$$\tau_{-1} \in (x_0, t_0) \quad , \quad \tau_j \in (t_j, t_{j+1}) \, , \quad j = 0, \ldots, n-k-1 \, ,$$
$$\tau_{n-k} \in (t_{n-k}, x_n) \, .$$

Wegen $x_{j+1} < t_j < x_{j+k-1}$ muß dann auch

$$x_0 < \tau_{-1} < x_{k-1} \quad , \quad \begin{array}{l} x_{j+1} < \tau_j < x_{j+k} \,, \quad j = 0, \ldots, n-k-1 \,, \\ \tau_{n-k} \in [x_{n-k+1}, x_n] \end{array}$$

gelten. Damit haben wir $n - k + 2$ Nullstellen von s' gefunden, welche hinsichtlich der x_j die Voraussetzung des Satzes für $k-1$ erfüllen. Nach Satz 7.3 ist s' aber eine Linearkombination der $B_{j,k-1}$. Nach Induktionsannahme folgt also $s' = 0$, also $s = 0$ und damit $a = 0$. Also ist B auch für die Ordnung k invertierbar.

□

5.9 Rationale Interpolation

Seien wieder paarweise verschiedene Stützstellen x_j und Stützwerte y_j gegeben. Gesucht ist eine rationale Funktion

$$R^{m,n}(x) = \frac{P^{m,n}(x)}{Q^{m,n}(x)} = \frac{a_0 + a_1 x + \cdots + a_m x^m}{b_0 + b_1 x + \cdots + b_n x^n} \ ,$$

welche diese Werte interpoliert. Da $\mathbb{R}^{m,n} m+n+1$ wesentliche Parameter enthält, wird man $j = 0, \ldots, m+n$ annehmen. Die rationale Interpolationsaufgabe $A^{m,n}$ wird also

$$R^{m,n}(x_j) = y_j \ , \qquad j = 0, \ldots, m+n$$

lauten. Wir stellen ihr zur Zeit die Aufgabe $S^{m,n}$, welche

$$P^{m,n}(x_j) - y_j Q^{m,n}(x_j) = 0 \ , \qquad j = 0, \ldots, m+n$$

lautet. Der Zusammenhang zwischen diesen beiden Aufgaben ist leider nicht so einfach, wie es auf den ersten Blick aussieht.

Beispiel: Sei $m = n = 1$. Stützstellen und -werte seien

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

$S^{1,1}$ bedeutet das lineare Gleichungssystem

$$\begin{aligned} a_0 - b_0 &= 0 \\ a_0 + a_1 - 2(b_0 + b_1) &= 0 \\ a_0 + 2a_1 - 2(b_0 + 2b_1) &= 0 \ . \end{aligned}$$

Dieses System hat die bis auf einen gemeinsamen Faktor eindeutig bestimmte Lösung

$$b_1 = 1 \quad , \quad a_1 = 2 \quad , \quad a_0 = b_0 = 0 \ .$$

Dies führt zu

$$R^{1,1}(x) = \frac{P^{1,1}(x)}{Q^{1,1}(x)} = \frac{2x}{x} = 2 \ ,$$

und dies ist offenbar keine Lösung von $A^{1,1}$.

Wir müssen uns genauer ansehen, wie rationale Funktionen als Paare (P, Q) von Polynomen dargestellt werden können. Ist a konstant, so betrachten wir (P, Q) und (aP, aQ) als identisch. Gilt für die Paare (P_1, Q_1) und (P_2, Q_2) $Q_1 P_2 = Q_2 P_1$, so betrachten wir die rationalen Funktionen $R_1 = P_1/Q_1$, $R_2 = P_2/Q_2$ als äquivalent: $R_1 \sim R_2$. Mit $\tilde{R}$ bezeichnen wir die rationale Funktion R, bei der Zähler und Nenner soweit wie möglich gekürzt wurden.

Satz 5.9.1 *$S^{m,n}$ besitzt stets eine Lösung, deren Nenner nicht identisch verschwindet. Je zwei Lösungen von $S^{m,n}$ sind äquivalent.*

Beweis: $S^{m,n}$ hat als lineares homogenes System von $n+m+1$ Gleichungen in $n+m+2$ Unbekannten stets eine Lösung $a_0, \ldots, a_m, b_0, \ldots, b_n$, in der nicht alle Zahlen verschwinden. Wäre $Q^{m,n} \equiv 0$, so wäre $b_0 = \cdots = b_n = 0$ und wegen der Interpolationsbedingung $P^{m,n}(x_j) = 0$, $j = 0, \ldots, m+n$. Also hätte $P^{m,n}$ mindestens $n+1$ Nullstellen und würde identisch verschwinden. Dann wäre aber auch $a_0 = \cdots = a_n = 0$.

Sind $R_1 = P_1/Q_1$, $R_2 = P_2, Q_2$ Lösungen von $S^{m,n}$, so folgt für $P = P_1 Q_2 - P_2 Q_1$

$$\begin{aligned} P(x_j) &= P_1(x_j) Q_2(x_j) - P_2(x_j) Q_1(x_j) \\ &= y_j (Q_1 Q_2)(x_j) - y_j (Q_2 Q_1)(x_j) = 0 \; . \end{aligned}$$

Also hat das Polynom P vom Grade $n+m$ die $n+m+1$ Nullstellen $x_0, \ldots, x_{n+m}$ und verschwindet damit identisch. Also ist $R_1 \sim R_2$.

□

Sei nun $R^{m,n} = P^{m,n}/Q^{m,n}$ Lösung von $S^{m,n}$. Ist $Q^{m,n}(x_j) \neq 0$, $j = 0, \ldots, m$, so ist $R^{m,n}(x_j) = y_j$, $j = 0, \ldots, m+n$ und damit $R^{m,n}$ Lösung von $A^{m,n}$. Ist aber für ein j $Q^{m,n}(x_j) = 0$, so muß auch $P^{m,n}(x_j) = 0$ sein, und $P^{m,n}$, $Q^{m,n}$ haben einen gemeinsamen Faktor $(x - x_j)^r$. Wir kürzen nun alle solche Faktoren und kommen so zu $\tilde{R}^{m,n} \sim R^{m,n}$. Natürlich braucht nicht $\tilde{R}^{m,n}(x_j) = y_j$, $j = 0, \ldots, m+n$ zu gelten. Tritt ein j mit $\tilde{R}^{m,n}(x_j) \neq y_j$ auf, so ist $A^{m,n}$ nicht lösbar. Wir nennen (x_j, y_j) einen unerreichbaren Punkt.

Die Berechnung der rationalen Interpolation erfolgt mit Hilfe von Kettenbrüchen. Dies sind Ausdrücke der Form

$$\phi_n = b_0 + \frac{a_1}{b_1+} \frac{a_2}{b_2+} = b_0 + \frac{a_1}{b_1+} \frac{a_2}{b_2+} \cdots \frac{a_n}{b_n} \, .$$

Rekursiv kann man Kettenbrüche durch

$$r_n = b_n \, , \quad r_k = b_k + \frac{a_{k+1}}{r_{k+1}} \, , \quad k = n-1, \ldots, 0 \, , \quad \phi_n = r_0$$

beschreiben. Dies mache man sich etwa an dem Fall $n = 3$ klar.

Satz 5.9.2 *Seien x_j, φ_j komplexer Zahlen und*

$$\phi_n(x) = \varphi_0 + \frac{x - x_0}{\varphi_1 +} \frac{x - x_1}{\varphi_2 +} \cdots \frac{x - x_{n-1}}{\varphi_n} \, .$$

Dann gibt es Polynome P, Q der Grade $\lfloor \frac{n+1}{2} \rfloor$ bzw. $\lfloor \frac{n}{2} \rfloor$ mit $\phi_n = P/Q$.

Beweis: Wir treiben Induktion nach n. Für $n = 0$ ist $\phi_n = \varphi_0$ ein Polynom vom Grade 0 und damit alles klar. Sei der Satz richtig für ein $n \geq 0$. Dann ist

$$\begin{aligned} \phi_{n+1} &= \varphi_0 + \frac{x - x_0}{\phi_n} = \varphi_0 + \frac{x - x_0}{P/Q} \\ &= \frac{\varphi_0 P + (x - x_0) Q}{P} \end{aligned}$$

mit Polynomen P, Q der Grade $\lfloor \frac{n+1}{2} \rfloor$ bzw. $\lfloor \frac{n}{2} \rfloor$. Dies ist aber ein Quotient von Polynomen der Grade $\lfloor \frac{n+2}{2} \rfloor$ bzw. $\lfloor \frac{n+1}{2} \rfloor$. Also gilt die Behauptung auch für $n + 1$.

Ähnlich wie die dividierten Differenzen führen wir nun die inversen Differenzen ein. Seien x_j, y_i, $j = 0, \ldots, n$ Stützwerte und Stützstellen.

Die inverse Differenz $\langle y_i, \ldots, y_k \rangle$ ist dann rekursiv definiert durch

$$\begin{aligned} \langle y_i \rangle &= y_i \, , \\ \langle y_i \cdots y_j y_k y_\ell \rangle &= \frac{x_k - x_\ell}{\langle y_i, \ldots, y_j, y_k \rangle - \langle y_i, \ldots, y_j, y_\ell \rangle} \, . \end{aligned}$$

Zum Beispiel ist

$$\langle y_k, y_\ell \rangle = \frac{x_k - x_\ell}{y_k - y_\ell} \,.$$

Die inversen Differenzen stellen wir im inversen Differenzenschema zusammen:

$$\begin{array}{lllll}
\langle y_0 \rangle & & & & \\
\langle y_1 \rangle & \langle y_0, y_1 \rangle & & & \\
\langle y_2 \rangle & \langle y_0, y_2 \rangle & \langle y_0, y_1, y_2 \rangle & & \\
\langle y_3 \rangle & \langle y_0, y_3 \rangle & \langle y_0, y_1, y_3 \rangle & \langle y_0, y_1, y_2, y_3 \rangle & \\
\langle y_4 \rangle & \langle y_0, y_4 \rangle & \langle y_0, y_1, y_4 \rangle & \langle y_0, y_1, y_2, y_4 \rangle & \langle y_0, y_1, y_2, y_3, y_4 \rangle
\end{array}$$

Satz 5.9.3 *(Thiele'scher Kettenbruch): Sei $\varphi_i = \langle y_0, y_1, \ldots, y_i \rangle$, $i = 0, \ldots, n$. Sei $\phi_n(x)$ die in Satz 5.14.2 definierte rationale Funktion. Dann gilt:*

$$\phi_n(x_j) = y_i \,, \qquad j = 0, \ldots, n \,.$$

Beweis: Wir berechnen ϕ_n durch die Rekursion

$$r_n(x) = \varphi_n \,, \quad r_k(x) = \varphi_k + \frac{x - x_k}{r_{k+1}(x)} \,, \qquad k = n-1, \ldots, 0 \,, \quad \phi_n(x) = r_0(x)$$

und zeigen für $n - k \le j \le n$

$$r_{n-k}(x_j) = \langle y_0, y_1, \ldots, y_{n-k-1}, y_j \rangle \,. \tag{9.1}$$

Die Behauptung ergibt sich dann für $k = n$.

Wir beweisen (9.1) durch Induktion nach k. Für $k = 0$ ist $r_n(x_n) = \varphi_n = \langle y_0, \ldots, y_{n-1}, y_n \rangle$ und (9.1) damit richtig. Sei (9.1) richtig für ein $k < n$. Dann ist für $n - k \le j \le n$

$$\begin{aligned}
r_{n-k-1}(x_j) &= \varphi_{n-k-1} + \frac{x_j - x_{n-k-1}}{r_{n-k}(x_j)} \\
&= \varphi_{n-k-1} + \frac{x_j - x_{n-k-1}}{\langle y_0, \ldots, y_{n-k-1}, y_j \rangle} \\
&= \varphi_{n-k-1} + \frac{x_j - x_{n-k-1}}{\frac{x_{n-k-1} - x_j}{\langle y_0, \ldots, y_{n-k-1} \rangle - \langle y_0, \ldots, y_{n-k-2}, y_j \rangle}} \\
&= \varphi_{n-k-1} + \langle y_0, \ldots, y_{n-k-2}, y_j \rangle - \langle y_0, \ldots, y_{n-k-1} \rangle \\
&= \langle y_0, \ldots, y_{n-k-2}, y_j \rangle \,.
\end{aligned}$$

Für $j = n - k - 1$ ist aufgrund der Rekursion

$$r_{n-k-1}(x_{n-k-1}) = \varphi_{n-k-1} = \langle y_0, \ldots, y_{n-k-1} \rangle \ .$$

Damit ist (9.1) auch für $k + 1$ richtig.

$\square$

Kapitel 6

Eigenwertprobleme

6.1 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 6.1.1 *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 6.1.2 *Jedem Eigenwert λ von A ordnen wir zwei Vielfachheiten zu:*

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

Seine geometrischeVielfachheit $\rho(\lambda)$ = 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 6.1.3 *Die (n,n)-Matrizen A, B heißen ähnlich, wenn es eine nichtsinguläre (n,n)-Matrix x gibt mit*

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

Satz 6.1.1 *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 Eigenvektor $x_1, \ldots, x_\rho$ 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. Eigenvektor 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 und Beispiel 1) ist $\lambda_k = d_k$, $\rho(\lambda_k) = \sigma(\lambda_k)$, $k = 1, \ldots, n$. Dem Beweis von Satz 1.1 und Beispiel 1) entnimmt man die Eigenvektor $x_k = xe_k = k$-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

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

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 6.1.2 *(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:

$$J = \begin{pmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & 2 & 1 & & & & \\ & & & 2 & & & & \\ & & & & 2 & & & \\ & & & & & i1 & & \\ & & & & & & i1 & \\ & & & & & & & i \end{pmatrix}$$

λ	$\rho(\lambda)$	$\sigma(\lambda)$
1	2	2
2	2	3
i	1	3

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:

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

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

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

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 $x_1, \dots, x_\nu$ die Spalten von X_ℓ, also $X_\ell = (x_1, \dots, x_\nu)$. Dann lautet die Gleichung

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

oder

$$\begin{array}{rcll} Ax_1 & = & \lambda x_1 & , \\ Ax_1 & = & \lambda x_i + x_{i-1} , & i = 2, \dots, \nu - 1 . \end{array}$$

x_1 ist also Eigenvektor zum Eigenwert λ_ℓ, wie wir schon aus Beispiel 3 und Satz 1.1 wissen. Für die weiteren Vektoren x_i gilt

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

also

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

Definition 6.1.4 *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 Eigenwert 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 6.1.5 *Eine (n, n)-Matrix A heißt hermitesch, wenn $A = A^*$ mit $A^* = \overline{A}^T$.*

Beispiel: Folgende Matrizen sind hermitesch:

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

Insbesondere sind reell-symmetrische Matrizen hermitesch.

Satz 6.1.3 *Sei A hermitesch. Dann sind alle Eigenwerte von A reell.*

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.

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

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

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

6.2 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

$$\begin{aligned}
x^{(0)} &= \sum_{i=1}^n c_i x_i \quad ,\\
x^{(k)} &= A^k x^{(0)}=\sum_{i=1}^n c_i A^k x_i=\sum_{i=1}^n c_i\lambda_i^k x_i\\
&= \lambda_1^k(c_1x_1+r_k)\ ,\\
r_k &= \sum_{i=2}^n c_i\left(\frac{\lambda_i}{\lambda_1}\right)^k x_i\ .
\end{aligned}$$

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 hängt von $\left| \frac{\lambda_2}{\lambda_1} \right|$ ab.

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 $c = (1, 1, 1)^T$ 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

$$\lambda_1 = 490 \quad , \quad \lambda_2 = 20 \quad , \quad x_1 = \begin{pmatrix} 07. \\ 1 \\ 0.7 \end{pmatrix} \quad .$$

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

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

Die Eigenwerte λ_ℓ sind also nach geometrischer Vielfachheit gezählt und nach abnehmenden Beträgen geordnet:

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

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

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

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 $k \geq \nu_\ell$

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

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

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

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 λ_1 maximalen Betrages, und es ist $\rho(\lambda_1) = \sigma(\lambda_1) = \rho$.

Es ist also

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

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 $\nu = \text{Max}\ \nu_\ell$. 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 λ_1 maximalen Betrages, aber es ist $\rho(\lambda_1) < \sigma(\lambda_1)$.

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

$$\begin{aligned} x^{(k)} &= \lambda_1^k x_1 M_{1k} y_1^{(0)} + \sum_{\ell=2}^{r} \lambda_\ell^k X_\ell M_{\ell k} y_\ell^{(0)} \\ &= \lambda_1^k \left\{ X_1 M_{1k} y_1^{(0)} + \sum_{\ell=2}^{r} \left(\frac{\lambda_\ell}{\lambda_1}\right)^k X_\ell M_{\ell k} y_\ell^{(0)} \right\} \\ &= \lambda_1^k \left\{ X_1 M_{1k} y_1^{(0)} + r_k \right\} \ . \end{aligned}$$

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. Bildet man nun (x^k, d) und bildet 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 $(a, d) \neq 0$. 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 6.2.1 *Die Potenzmethode konvergiert, wenn es genau einen betragsgrößten Eigenwert gibt. Stimmen für diesen die algebraische und geometrische Vielfachheit überein, so ist die Konvergenz geometrisch. Gibt es verschiedene betragsgleiche Eigenwerte, so ist die Potenzmethode nicht konvergent.*

6.3 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 gibt n l.u. Eigenvektoren $x_1, \ldots, x_n$. Wenden wir die Potenzmethode auf n Startvektoren $x_1^{(0)}, \ldots, x_n^{(0)}$ gleichzeitig an, also

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

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 und subtrahieren daher 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 $k \to \infty$

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

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

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

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

$$\begin{array}{rcccccl} A_k & = & X_k^{-1} A X_k & = & (L_k X_{k+1}^{-1}) A X_k & = & L_k R_k \ , \\ A_{k+1} & = & (X_{k+1}^{-1} A) X_{k+1} & = & (R_k X_k^{-1}) X_{k+1} & = & R_k L_k \ . \end{array}$$

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

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

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) Gemäß der vermuteten Konvergenz $X_k \to XR$ erwarten wir, daß

$$\begin{aligned} A_k &= X_k^{-1} A X_k \to R^{-1} X^{-1} A X R = R^{-1} J R \\ &= \begin{pmatrix} \lambda_1 & & & & \\ & \lambda_2 & & x & \\ & & \ddots & & \\ & \mathrm{O} & & & \\ & & & & \lambda_n \end{pmatrix} \end{aligned}$$

mit irgendwelchen Elementen oberhalb der Diagonalen.

Tatsächlich kann man unter gewissen Voraussetzungen 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 $r_{1,j}, \dots, r_{j-1,j}$ in (3.1) so, daß $x_j^{(1)}$ orthogonal zu $x_1^{(1)}, \dots, x_{j-1}^{(1)}$ 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 6.3.1 *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 $r_{22} q_2^{(k)}$ 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

$\square$

Satz 6.3.1 *A besitze n betragsmäßige verschiedene Eigenwerte $\lambda_1, \ldots, \lambda_n$. Sei $|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n| > 0$ und $A = XJ^{-1}$ die Jordan'sche Normalform von A. Dann gilt für die Matrix A_k der 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 $\ell = 0, \ldots, k$ durch Induktion nach ℓ beweist. Aus dem Vergleich der QR-Zerlegungen (3.5), (3.6) bekommen wir

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

Aus der ersten dieser Beziehungen folgt

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

aus der zweiten

$$R_k = R_k R_{k-1} \cdots R_0 (R_0 \cdots R_{k-1})^{-1} = S_{k+1} RJR^{-1} S_k^{-1} \to RJR^{-1} \ . \tag{3.8}$$

Also gilt

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

und dies ist eine Matrix der behaupteten Art.

$\square$

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}, 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.

6.4 Praktische Durchführung des QR-Algorithmus

Die QR-Zerlegung nach Householder benötigt $\frac{1}{3}n^3$ *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 $H = (L_{i,j})$ 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.

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 rechten Faktor R in der QR-Zerlegung $H = QR$:

for $k = 1, \ldots, n-1$

$\{$ Bestimme φ_k so, daß $\begin{pmatrix} c_k & s_k \\ -s_k & c_k \end{pmatrix} \begin{pmatrix} c_k & s_k \\ -s_k & c_k \end{pmatrix} \begin{pmatrix} h_{k,k} \\ h_{k+1,k} \end{pmatrix} = \begin{pmatrix} x \\ 0 \end{pmatrix}$

for $j = k, \ldots, n$

$$\begin{pmatrix} h_{k,j} \\ h_{k+1,j} \end{pmatrix} = \begin{pmatrix} c_k & s_k \\ -s_k & c_k \end{pmatrix} \begin{pmatrix} h_{k,j} \\ h_{k+1,j} \end{pmatrix}$$

$\}$

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

for $k = 1, \ldots, n-1$
for $j = 1, \ldots, k-1$

$$(h_{j,k}, h_{j,k+1}) = (h_{j,k}, h_{j,k+1}) \begin{pmatrix} c_k & -s_k \\ s_k & c_k \end{pmatrix} .$$

Jeder dieser Schritte benötigt $2n^2$, insgesamt also $4n^2$ *flops*.

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

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

Für σ_k verwendet man eine Näherung für den betragskleinsten Eigenwerte, 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.

6.5 Fehlerabschätzung bei Eigenwertproblemen

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

Satz 6.5.1 *(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 $\bigcup\limits_{i=1}^{n} K_i$ 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

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

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 tK_i. Wir beweisen (b) für alle $A(t)$ mit $0 \leq t \leq 1$. Wir benutzen folgendes

Lemma 6.5.1 *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 $t \leq t_0$ tK_i, 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 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 tK_i zusammenstoßen. Die so entstehende Vereinigung von $tK_i's$ enthält dann genau diejenigen Eigenwerte, welche bisher in den einzelnen $tK_i's$ 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 6.5.2 *Sei A hermite'sch und seien λ, x Näherungen für einen Eigenwert von A mit zugehörigem Eigenvektor.*

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

$$\begin{aligned} d = Ax - \lambda x &= \sum_{i=1}^{n} c_i(\lambda_i - \lambda)x_i \ , \\ \|d\|_2^2 = \sum_{i=1}^{n} |c_i|^2|\lambda_i - \lambda|^2 &\geq \sum_{i=1}^{n} |c_i|^2 \min_{1\leq i\leq n} |\lambda_i - \lambda|^2 = \|x\|_2^2|\lambda_k - \lambda|^2 \ . \end{aligned}$$

□

Über die Eigenwerte gestörter Matrizen hat man im hermite'schen Fall folgendes einfache Resultat.

Satz 6.5.3 *Seien A, B hermite'sche (n,n)-Matrizen. Dann kann man die Eigenwerte λ_i von A und die Eigenwerte μ_i von B so anordnen, daß*

$$|\lambda_i - \mu_i| \leq \|a - B\|_2$$

ist.

Beweis: Sei λ_i Eigenwert zum Eigenvektor x_i von A. Dann ist

$$\begin{aligned} d = Bx_i - \lambda_i x_i &= Ax_i - \lambda_i x_i + (B - A)x_i \\ &= (B - A)x_i \end{aligned}$$

und damit

$$\|d\| \leq \|B - A\|_2 \|x_i\|_2 \ .$$

Nach Satz 5.2 gibt es einen Eigenwert μ_i von B mit

$$|\lambda_i - \mu_i| \leq \frac{\|d\|_2}{\|x_i\|_2} \leq \|B - A\|_2 \ .$$

□

Im allgemeinen Fall sind Störungsresultate sehr viel verwickelter und ungünstiger.

Satz 6.5.4 *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

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

Die Matrix $D^{-1}(H+\varepsilon G)D$ 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

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

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

$$D^{-1}(J+G)D = \begin{pmatrix} D_1^{-1} \quad J_1 & \quad D_1 & & \\ & \ddots & & \\ & & D_r^{-1} \quad J_r \quad D_r \end{pmatrix} + \varepsilon D^{-1}GD \; .$$

Anwendung von 1) ergibt die Behauptung.

$\square$

Kapitel 7

Approximation

7.1 Approximation in normierten Räumen

Eine vorgegebene Funktion $f \in C[a,b]$ soll durch eine Funktion u aus einem Unterraum $U \subset C[a,b]$ approximiert werden. Wir untersuchen in diesem Paragraphen, ob es eine optimale Wahl u^* für u gibt.

Wir führen zunächst auf $C[a,b]$ eine Norm $\|\cdot\|$ ein.

Für $1 \leq p < \infty$ ist beispielsweise

$$\|f\|_p = \left(\int_a^b |f(x)|^p dx\right)^{1/p}$$

eine Norm. Einen wichtigen Fall erhält man für $p \to \infty$:

$$\|f\|_\infty = \lim_{p\to\infty} \|f\|_p = \max_{[a,b]} |f(x)|$$

Wir legen uns im folgenden jedoch auf keine bestimmte Norm fest. Sei U der von den linear unabhängigen Funktionen $u_0, \ldots, u_n \in C[a,b]$ aufgespannte Unterraum:

$$U = \left\{\sum_{k=0}^{n} a_k u_k \ , \ a_k \in \mathbb{R}\right\}$$

und sei

$$\varepsilon(f) := \operatorname{Inf}_{u\in U} \|f-u\| \ .$$

Dann heißt unsere Aufgabe:

Finde $u^* \in U$ mit $\|f - u^*\| = \varepsilon(f)$.

Satz 7.1.1 *Das Approximationsproblem ist immer lösbar.*

Beweis: Für $a := (a_0, \ldots, a_n)^T \in \mathbb{R}^{n+1}$ sei

$$F(a) = \|f - \sum_{k=0}^{n} a_k u_k\|$$

$F(a)$ ist eine stetige Funktion von a. Bei der Suche nach einem Minimum von F in $\mathbb{R}^{n+1}$ können wir uns auf die Menge $M = \{a \in \mathbb{R}^{n+1} : F(a) \leq F(a)\}$ beschränken. M ist abgeschlossen. M ist auch beschränkt. Denn die Abbildung $a \to \|\sum_{k=0}^{n} a_k u_k\|$ ist eine Norm in $\mathbb{R}^{n+1}$. Da je zwei Normen in $\mathbb{R}^{n+1}$ äquivalent sind, gibt es eine Konstante $c > 0$ mit

$$\|\sum_{k=0}^{n} a_k u_k\| \geq c\|a\|_\infty \ .$$

Mit $\|\alpha\|_\infty \to \infty$ strebt also auch $F(a) \to \infty$, und damit ist M beschränkt.

M ist also kompakt, und die stetige Funktion F nimmt auf M ihr Minimum an, etwa in a^*.

Also existiert ein $a^* \in \mathbb{R}^{n+1}$ mit $F(a^*) = \varepsilon(f)$. $u^* = \sum_{k=0}^{n} a_k^* u_k$ ist dann Lösung des Approximationsproblems.

$\square$

Definition 7.1.1 *Eine Norm heißt strikt, wenn gilt:*

$$\|f + g\| = \|f\| + \|g\| \Rightarrow f, g \ \textit{linear abhängig}$$

Beispiele: $\| \ \|_2$ ist strikt, $\| \ \|_\infty$ hingegen nicht. Letzteres sieht man an dem Beispiel $f = 1$, $g = x$ in $C[0, 1]$.

Satz 7.1.2 *Das Approximationsproblem für strikte Normen ist eindeutig lösbar.*

Beweis: Seien u_1^* und u_2^* Lösungen und $u^* := \frac{1}{2}(u_1^* + u_2^*)$. Dann gilt:

$$\|f - u^*\| \leq \frac{1}{2}\|f - u_1^*\| + \frac{1}{2}\|f - u_2^*\| = \varepsilon(f)$$

u^* ist also ebenfalls eine Lösung und in der Ungleichung gilt Gleichheit. Da die Norm streng ist, gibt es $\alpha, \beta \in \mathbb{R}$ mit $|\alpha| + \beta| > 0$ und

$$\frac{\alpha}{2}(f - u_1^*) = \frac{\beta}{2}(f - u_2^*)$$

oder

$$(\alpha - \beta)f = \alpha u_1^* - \beta u_2^* \, .$$

Ist $\alpha = \beta$, so folgt $u_1^* = u_2^*$. Ist $\alpha \neq \beta$, so folgt $f \in U$, also $f = u_1^* = u_2^*$. In jedem Fall ist also $u_1^* = u_2^*$.

7.2 Tschebyscheff-Approximation

Wir betrachten im folgenden die Approximationsaufgabe mit der Norm

$$\|f\| = \max_{[a,b]} |f(x)|$$

Definition 7.2.1 *Seien* $u_0, \ldots, u_n \in C[a,b]$ *linear unabhängig.* $U = \langle u_0, \ldots, u_n \rangle$ *heißt unisolvent, wenn für jede Unterteilung* $a \leq x_0 < x_1 < \ldots < x_n \leq b$ *und für jede Wahl von* y_j, $j = 0, \ldots, n$ *genau ein* $u^* \in U$ *existiert mit* $u^*(x_j) = y_j$, $j = 0, \ldots, n$.

Bemerkung: U unisolvent $\Leftrightarrow$ det $(u_k(x_j)) \neq 0$ für jede Unterteilung.
$\Leftrightarrow$ Jede Funktion $u \in U$ mit mehr als n Nullstellen verschwindet identisch.

Beispiele:

1) $U = \mathcal{P}_n$.

2) $u_k(x) = e^{\lambda k x}$, $k = 0, \ldots, n$, $\lambda \neq 0 \in \mathbb{R}^1$.
   det $(u_k(x_j)) = \left(e^{\lambda x_j}\right)^k = (z_j)^k$ ist gerade die Vandermonde-Determinante von $(z_0, \ldots, z_n)$.

3) $U = \mathcal{T}_m$, $n = 2m + 1$ für $0 \leq a < b < 2\pi$.

4) $< 1, x^2 >$ ist in $[-1, +1]$ nicht unisolvent.

5) $< B_{0,k}, \ldots, B_{n-k,k} >$ in $[x_0, x_n]$ nicht unisolvent.

Satz 7.2.1 *Sei* U *unisolvent,* $u \in U$ *und* $a \leq x_0 < x_1 < \ldots < x_{n+1} \leq b$. $(f-u)(x_j)$ *habe alternierende Vorzeichen, d.h. es gebe ein* $\sigma \in \mathbb{R}$ *mit* $|\sigma| = 1$, *so daß*

$$sgn[(f-u)(x_j)] = (-1)^j \sigma\ , \quad j = 0, \ldots, n+1\ .$$

Dann gilt:

$$\min_{j=0}^{n+1} |(f-u)(x_j)| \leq \varepsilon(f) \leq \|f - u\|\ .$$

Beweis: Wir nehmen an, es wäre

$$\min_{j=0}^{n+1} |(f-u)(x_j)| > \varepsilon(f) \ .$$

Dann gäbe es ein $v \in U$ mit

$$\min_{j=0}^{n+1} |(f-u)(x_j)| > \|f-v\| \ .$$

Es folgt

$$|(f-u)(x_j)| > |(f-v)(x_j)| \ , \quad j = 0, \ldots, n+1 \ .$$

Nun ist aber

$$|(f-u)(x_j)| = (\operatorname{sgn}(f-u)(x_j))(f-u)(x_j) = \sigma(-1)^j(f-u)(x_j) \ ,$$

und es folgt weiter

$$\sigma(-1)^j(f-u)(x_j) > |(f-v)(x_j)| \geq \sigma(-1)^j(f-v)(x_j) \ , \quad j = 0, \ldots, n+1 \ .$$

Dies ist nur möglich, wenn

$$\sigma(-1)^j(v-u)(x_j) > 0 \ , \quad j = 0, \ldots, n+1 \ .$$

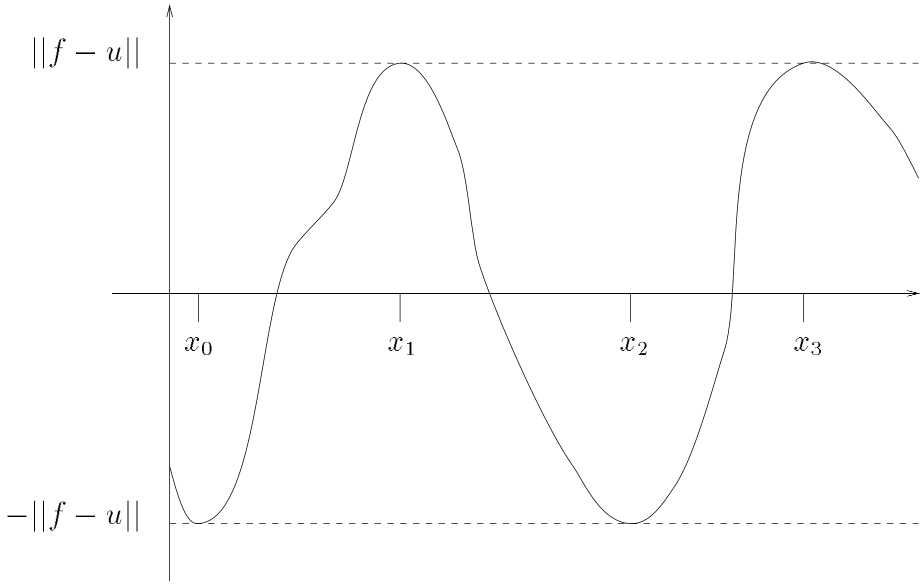
$v-u$ hat also mindestens $n+1$ Nullstellen. Da U unisolvent ist, folgt $v = u$. Dies ist ein Widerspruch.

□

Definition 7.2.2 $x_0 < x_1 < \ldots < x_{n+1}$ *heißt Alternante (der Länge $n+2$) zu $f-u$, falls $f-u$ an den Stellen x_j mit alternierendem Vorzeichen seinen Maximalwert annimmt, d.h. es gibt $\sigma \in \mathbb{R}$ mit $|\sigma| = 1$, so daß*

$$(f-u)(x_j) = \sigma(-1)^j \|f-u\|$$

Beispiel: $n = 2$



Satz 7.2.2 *Sei U unisolvent und $u \in U$. $f - u$ besitze eine Alternante der Länge $n+2$. Dann ist u Lösung des Approximationsproblems, d.h. $\|f-u\| = \varepsilon(f)$.*

Beweis: Nach Satz 7.2.1 gilt:

$$\|f-u\| = \min_{j=0}^{n+1} |(f-u)(x_j)| \leq \varepsilon(f) \leq \|f-u\|$$

Bemerkung: Es gilt auch die umgekehrte Behauptung, d.h. zu jeder Lösung gibt es eine Alternante. Der Beweis ist schwieriger (vgl. G. Meinardus, Approximation von Funktionen und ihre numerische Behandlung).

Satz 7.2.3 *Sei U unisolvent und $a \leq x_0 < \ldots < x_{n+1} \leq b$. Dann gibt es genau ein $u \in U$ und $d \in \mathbb{R}^1$ mit*

$$u(x_j) = f(x_j) - d(-1)^j \quad , \quad j = 0, \ldots, n+1$$

und es gilt:

$$|d| \leq \varepsilon(f) \leq \|f-u\|$$

Beweis: Sind u, d von der genannten Art, so ist

$$(f-u)(x_j) = d(-1)^j \quad , \quad j = 0, \dots, n+1$$

und nach Satz 7.2.1 folgt

$$|d| = \min_{j=0}^{n+1} |(f-u)(x_j)| \le \varepsilon(f) \ .$$

Für d und die Koeffizienten a_k von $u = \sum_{k=0}^{n} a_k u_k$ hat man die Gleichungen

$$\sum_{k=0}^{n} a_k u_k(x_j) + d(-1)^j = f(x_j) \ , \quad j = 0, \dots, n+1 \ .$$

Das zugehörige homogene System lautet

$$u(x_j) = \sum_{k=0}^{n} a_k u_k(x_j) = -d(-1)^j \ , \quad j = 0, \dots, n+1$$

u hat also mindestens $n+1$ Nullstellen.

Da U unisolvent ist, müssen die a_k und damit auch d verschwinden, das homogene System ist also nur trivial lösbar. Damit sind u, d eindeutig bestimmt.

□

Satz 7.2.3 bildet die Grundlage des Remes - Algorithmus zur Lösung der Approximationsaufgabe:

Sei $f \in C[a,b]$ und U unisolvent. Gesucht wird ein $u^* \in U$ mit

$$\|f - u^*\| \le \|f - u\| \ , \qquad \forall \quad u \in U \ .$$

Wir konstruieren eine Folge von Unterteilungen

$$x^{(k)} = (x_0^{(k)}, \dots, x_{n+1}^{(k)}) \ , \quad a \le x_0^{(k)} < \dots < x_{n+1}^{(k)} \le b$$

und finden die nach Satz 7.2.3 eindeutig bestimmten $u^{(k)}$, $d^{(k)}$ mit

$$\begin{aligned} u^{(k)}(x_j) &= f(x_j^{(k)}) - d^{(k)}(-1)^j \quad , \quad j = 0, \dots, n+1 \ , \\ |d^{(k)}| &\le \varepsilon(f) \le \|f - u^{(k)}\| =: \varepsilon^{(k)} \quad . \end{aligned}$$

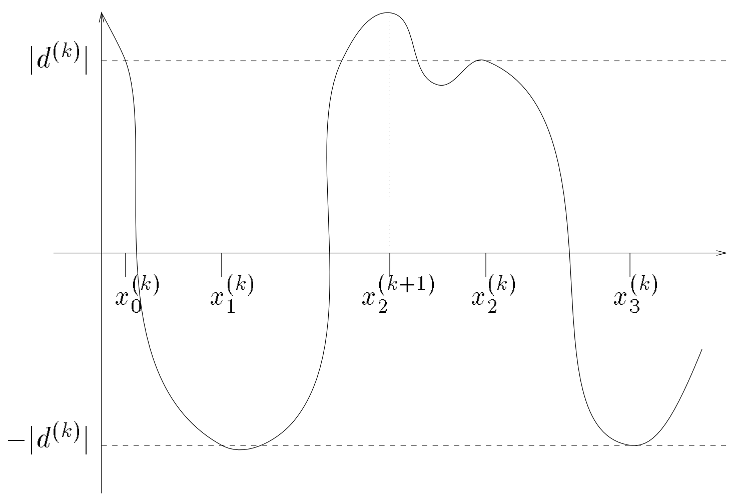
Die Unterteilung $X^{(0)}$ kann beliebig gewählt werden, es ist jedoch ein wichtiges praktisches Problem, ein geeignetes $X^{(0)}$ zu finden.

Sei $X^{(k)}$ berechnet. Falls $|d^{(k)}| = \varepsilon^{(k)}$, so ist $\|f - u^{(k)}\| = \varepsilon(f)$ und das Verfahren kann abgebrochen werden. Für $|d^{(k)}| < \varepsilon^{(k)}$ wird $X^{(k+1)}$ so gewählt, daß die folgenden Bedingungen erfüllt sind:

(a) $\quad (f - u^{(k)})(x_j^{(k+1)})$ $\qquad$ hat alternierendes Vorzeichen

(b) $\quad |(f - u^{(k)})(x_j^{(k+1)})| \geq |d^{(k)}| \qquad j = 0, \ldots, n+1$

(c) $\quad |(f - u^{(k)})(x_j^{(k+1)})| > |d^{(k)}|$ $\quad$ für mindestens ein j



In dem skizzierten Beispiel sind die Bedingungen (a) bis (c) erfüllt, wenn $x_2^{(k+1)}$ wie eingezeichnet gewählt wird und die übrigen Unterteilungspunkte nicht verändert werden.

Satz 7.2.4 *Sei U unisolvent und seien die Bedingungen (a), (b), (c) erfüllt. Dann gilt*

$$|d^{(k+1)}| > |d^{(k)}| \,.$$

Beweis: Wir entwickeln $u^{(k+1)}$ nach der Basis $\{u_0, \ldots, u_n\}$ von U:

$$u^{(k+1)} = \sum_{i=0}^{n} a_i^{(k+1)} u_i$$

Für die Koeffizienten $a_i^{(k+1)}$ und für $d^{(k+1)}$ gelten die Gleichungen

$$\sum_{i=0}^{n} a_i^{(k+1)} u_i(x_j^{(k+1)}) + d^{(k+1)}(-1)^j = f(x_j^{(k+1)})\,, \quad j = 0, \ldots, n+1\ .$$

Nach der Cramer'schen Regel gilt für $d^{(k+1)}$:

$$d^{(k+1)} \;=\; \frac{\begin{vmatrix} u_0(x_0^{(k+1)}) & \cdots & u_n(x_0^{(k+1)}) & f(x_0^{(k+1)}) - u^{(k)}(x_0^{(k+1)}) \\ \vdots & & \vdots & \vdots \\ u_0(x_{n+1}^{(k+1)}) & \cdots & u_n(x_{n+1}^{(k+1)}) & f(x_{n+1}^{(k+1)}) - u^{(k)}(x_{n+1}^{(k+1)}) \end{vmatrix}}{\begin{vmatrix} u_0(x_0^{(k+1)}) & \cdots & u_n(x_0^{(k+1)}) & 1 \\ \vdots & & \vdots & \vdots \\ u_0(x_j^{(k+1)}) & \cdots & u_n(x_j^{(k+1)}) & (-1)^j \\ \vdots & & \vdots & \vdots \\ u_0(x_{n+1}^{(k+1)}) & \cdots & u_n(x_{n+1}^{(k+1)}) & (-1)^{n+1} \end{vmatrix}}$$

Dabei haben wir im Zähler von der letzten Spalte $\left(u^{(k)}(x_0^{(k+1)}), \ldots, u^{(k)}(x_{n+1}^{(k+1)})\right)^T$ subtrahiert. Dies ändert den Wert der Determinante jedoch nicht, da der subtrahierte Vektor eine Linearkombination der ersten $n+1$ Spalten ist.

Wir entwickeln beide Determinanten nach der letzten Spalte:

$$d^{(k+1)} = \frac{\sum\limits_{j=0}^{n+1} (f - u^k)(x_j^{(k+1)})(-1)^{n+1+j} D_j}{\sum\limits_{j=0}^{n+1} (-1)^j (-1)^{n+1+j} D_j}$$

D_j entsteht aus beiden Determinanten durch Streichen der j-ten Zeile und der letzten Spalte.

Sei $\mu_j = D_j / \left(\sum\limits_{i=0}^{n+1} D_i\right)$. Dann gilt:

$$d^{(k+1)} = \sum_{j=0}^{n+1} (-1)^j (f - u^{(k)})(x_j^{(k+1)}) \mu_j$$

Wir zeigen nun, daß die μ_j alle streng positiv sind. Da U unisolvent, gilt $D_j \neq 0,\ \ j = 0, \ldots, n+1$. Betrachten wir die Unterteilung $X^{(k+1)}$. D_j hängt stetig von den $x_i^{(k+1)},\ \ i \neq j$ ab. Denken wir uns nun $x_{j-1}^{(k+1)}$ in Richtung von $x_j^{(k+1)}$ verschoben, so nähert sich D_j dem Wert von D_{j-1} stetig an, darf aber für keinen Wert von $x_{j-1}^{(k+1)}$ verschwinden. Daher müssen D_j und D_{j-1} gleiches Vorzeichen haben, woraus folgt, daß alle D_i dasselbe Vorzeichen haben, die μ_i also streng positiv sind.

Aus der Voraussetzung (a) folgt, daß auch das Vorzeichen von $(-1)^j(f-u^k)(x_j^{(k+1)})$ von j unabhängig ist. Damit gilt nach (b), (c)

$$|d^{(k+1)}| = \sum_{j=0}^{n+1} |(f-u^{(k)})|\mu_j > |d^{(k)}| \sum_{j=0}^{n+1} \mu_j = |d^{(k)}| \ ,$$

da $\sum\limits_{j=0}^{n+1} \mu_j = 1$.

□

Wir betrachten nun den Fall der Approximation mit Polynomen n-ten Grades und untersuchen, wir gut bereits das Interpolationspolynom approximiert.

Satz 7.2.5 *Sei $U = \mathcal{P}_n$. Für $u \in U$ gelte $u(x_j) = f(x_j),\ \ j = 0, \ldots, n$, wobei die x_j aus $[a,b]$ und paarweise verschieden seien.*

Dann gilt:

$$\|f-u\| \leq V_n \cdot \varepsilon(f)$$

mit

$$V_n = 1 + \|\sum_{j=0}^{n} |\omega_j(x)\| \ , \ \ \omega_j(x) = \prod_{\substack{j=0\\ i\neq j}}^{n} \frac{x-x_i}{x_j-x_i} \ .$$

Beweis: Sei L_n der Operator, der eine Funktion in ihr Interpolationspolynom zur Unterteilung $x_0, \ldots, x_n$ überführt. Wir benutzen die Lagrange'sche Form des Interpolationspolynoms und erhalten $u = L_n f = \sum\limits_{j=0}^{n} f(x_j)\omega_j$. Es folgt:

$$
\begin{aligned}
f-u &= f-u^*+u^*-u \\
&= f-u^*+L_nu^*-L_nf \qquad (\text{da } u^*=L_nu^*) \\
&= f-u^*+L_n(u^*-f) \qquad (\text{da } L_n \text{ linear ist}) .
\end{aligned}
$$

Also ist

$$
\begin{aligned}
|(f-u)(x)| &\le |(f-u^*)(x)| + \sum_{j=0}^{n} |(u^*-f)(x_j)| \ \ |\omega_j(x)| \\
&\le \|f-u^*\| \left(1+\sum_{j=0}^{n} |\omega_j(x)|\right)
\end{aligned}
$$

und damit

$$
\|f-u\| \le \left(1+\left\|\sum_{j=0}^{n} |\omega_j(x)|\right\|\right) \varepsilon(f) .
$$

Beispiel: $[a,b]=[-1,+1]$, x_j seien die Nullstellen des Tschebyscheff-Polynoms $T_{n+1}(x)=\cos(n+1)t$, $x=\cos t$:

$$
x_j=\cos\left(\frac{j+\frac{1}{2}}{n+1}\pi\right) \qquad , \quad j=0,\ldots,n .
$$

Durch numerisches Rechnen erhält man die folgenden Werte für V_n:

n	V_n
4	3.0
10	3.5
100	4.9

Im für die Praxis interessanten Bereich $n\le 10$ vergrößert man den Fehler nur um einen Faktor kleiner gleich 3.5, wenn man mit dem Interpolationspolynom an Tschebyscheffstellen approximiert.

7.3 Approximation nach Gauß

Sei H ein unitärer Raum. Wir benutzen die durch das innere Produkt gegebene Norm:

$$\|f\| = (f,f)^{1/2}$$

Seien $u_0, \ldots, u_n \in H$ linear unabhängig und sei $U = < u_0, \ldots, u_n >$. Die Approximationsaufgabe lautet dann:

Für gegebenes $f \in H$ wird $u^* \in U$ gesucht mit

$$\|f - u^*\| = \operatorname{Inf}_{u \in U} \|f - u\| \ .$$

Satz 7.3.1 *u^* existiert und ist eindeutig bestimmt.*

Beweis: Die Existenz von u^* folgt aus Satz 1.1. Die durch das innere Produkt induzierte Norm ist strikt. Aus Satz 1.2 folgt dann die Eindeutigkeit.

Satz 7.3.2 *Es gilt $u^* = \sum\limits_{i=0}^{n} a_i^* u_i$, wobei die Koeffizienten a_i^* durch die Normalgleichungen*

$$(*) \qquad \sum_{i=0}^{n} a_i^*(u_i, u_k) = (f, u_k) \qquad k = 0, \ldots, n$$

bestimmt sind.

Beweis: Wir formen die Normalgleichungen etwas um:

$$\left(u_k \ , \ \sum_{i=0}^{n} a_i^* u_i - f \right) = 0 \qquad k = 0, \ldots, n$$

$\sum\limits_{i=0}^{n} a_i^* u_i - f$ ist also orthogonal zu U. Für ein beliebiges $u = \sum\limits_{i=0}^{n} a_i u_i$ gilt:

$$\|f - u\|^2 = \left\| f - \sum_{i=0}^{n} a_i u_i \right\|^2 = \left\| f - \sum_{i=0}^{n} a_i^* u_i + \sum_{i=0}^{n} (a_i^* - a_i) u_i \right\|^2$$

$$
\begin{aligned}
&= \left\| f - \sum_{i=0}^{n} a_i^* u_i \right\|^2 + \left\| \sum_{i=0}^{n} (a_i^* a_i) u_i \right\|^2 \\
&\geq \left\| f - \sum_{i=0}^{n} a_i^* u_i \right\|^2
\end{aligned}
$$

$\square$

Beispiele:

1) $H = \mathbb{C}^p$

   Die Basisvektoren $u_0, \ldots, u_n$ von U bilden die Matrix $A = (u_0, \ldots, u_n)$. Für $f \in \mathbb{C}^p$ führt die Aufgabe $\|f - Ax\|$ zu minimieren auf ein überbestimmtes lineares Gleichungssystem, siehe I.1.6.

2) $H = C[-1,1]$, $U =< 1, t, t^2 >$, $\ f(t) = e^t$

$$\|f\| = \left(\int_{-1}^{1} |f(t)|^2 dt \right)^{1/2}$$

$(u_i, u_k) = \int_{-1}^{1} u_i u_k dt = \int_{-1}^{1} t^{i+k} dt = \frac{1+(-1)^{i+k}}{i+k+1}$

$$
\begin{aligned}
(u_0, f) &= \int_{-1}^{1} 1 \cdot e^t dt = e - 1/e \\
(u_1, f) &= \int_{-1}^{1} t e^t dt = 2/e \\
(u_2, f) &= \int_{-1}^{1} t^2 e^t dt = e - 5/e
\end{aligned}
$$

Die Normalgleichungen lauten damit

$$
\begin{pmatrix} 2 & 0 & \frac{2}{3} \\ 0 & \frac{2}{3} & 0 \\ \frac{2}{3} & 0 & \frac{2}{5} \end{pmatrix}
\begin{pmatrix} a_0^* \\ a_1^* \\ a_2^* \end{pmatrix}
=
\begin{pmatrix} e - 1/e \\ 2/e \\ e - 5/e \end{pmatrix}
$$

und führen auf

$$\begin{aligned} a_0^* &= 0.99629 \\ a_1^* &= 1.10364 \\ a_2^* &= 0.53672 \end{aligned}$$

Es gilt:

$$\max_{t\in[-1,1]} |(e^t - a_0^* - a_1^* - a_1^* t - a_2^* t^2| = 0.082$$

3) Die Normalgleichungen vereinfachen sich bedeutend, wenn $u_0, \ldots, u_n$ ein Orthonormalsystem bilden:

$$(u_i, u_k) = \begin{cases} 1 & i = k \\ 0 & \text{sonst} \end{cases}$$

Dann gilt einfach

$$(u_i, u_k) = \begin{cases} 1 & i = k \\ 0 & \text{sonst} \end{cases}$$

Dann gilt einfach

$$a_k^* = (f, u_k)$$

und

$$u^* = \sum_{i=0}^{n} (f, u_i) u_i \ .$$

Die a_k^* heißen in diesem Fall verallgemeinerte Fourier-Koeffizienten von f bezüglich $u_0, \ldots, u_n$.

4) $H = C[0, 2\pi]$

$(f, g) = \int\limits_0^{2\pi} f\overline{g} dx$

$$u_k = \frac{1}{\sqrt{2\pi}} e^{ikx} \qquad k = 0\ , \ \neq 1, \ldots, \neq m\ , n = 2m$$

$$\int\limits_0^{2\pi} u_k u_\ell dx = \frac{1}{2\pi} \int\limits_0^{2\pi} e^{i(k-\ell)x} dx = \begin{cases} 1 & k = \ell \\ 0 & \text{sonst} \end{cases}$$

Hier sind die $a_k^* = (f, u_k)$ die üblichen Fourier-Koeffizienten.

$$u^* = \sum_{k=-n}^{n} (f, u_k) u_k = \frac{1}{2\pi} \sum_{k=-n}^{n} \left(\int\limits_0^{2\pi} f e^{-ikt} dt \right) e^{ikx}$$

heißt endliche Fourier-Reihe von f. Für $n \to \infty$ konvergiert u^* in vielen Fällen gegen f.

5) $\{u_0, \ldots, u_{2m}\} = \left\{\frac{1}{\sqrt{2\pi}}, \frac{1}{\sqrt{\pi}} \cos x, \sin mx\right\}$

$$u^*(x) = \frac{a_0}{2} + \sum_{k=1}^{m} (a_k \cos kx + b_k \sin kx)$$

heißt ebenfalls Fourier-Reihe. Es gilt

$$\left.\begin{array}{l} a_k \\ b_k \end{array}\right\} = \frac{1}{\pi} \int_0^{2\pi} \left\{\begin{array}{c} \cos kx \\ \sin kx \end{array}\right\} f(x) dx$$

Satz 7.3.3 *Seien $u_0, \ldots, u_n$ linear unabhängig. Dann gibt es ein Orthonormalsystem $v_0, \ldots, v_n$ mit*

$$< u_0, \ldots, u_m > = < v_0, \ldots, v_m > , \quad m = 0, \ldots, n .$$

Beweis: Durch das E. Schmidt'sche Orthonormalisierungsverfahren.

□

Sei nun $H = C[a, b]$ und $(f, g) = \int_a^b wfgdx$ mit einer Funktion w, die "Gewichtsfunktion", welche stetig und in (a, b) positiv ist. Ist $u_0 = 1$, $u_1 = x, \ldots, u_n = x^n$, so nennt man die v_m "orthogonale Polynome" (zu (a, b) und w).

Satz 7.3.4 *v_k hat in (a, b) genau k einfache Nullstellen.*

Beweis: $x_1, \ldots, x_m$ seien die Zeichenwechsel von v_k in (a, b). Wir zeigen, daß $m = k$ gilt.

Sei $q = \sum\limits_{i=1}^{m} (x - x_i) \in \mathcal{P}_m$. Dann ist für $m < k$

$$(q, v_k) = \int\limits_a^b w q v_k dx = 0 \; .$$

Die Funktion $q \cdot v_k$ hat konstantes Vorzeichen. Also folgt

$$q \cdot v_k \equiv 0 \quad \text{in} \quad (a, b)$$

und dies ist ein Widerspruch.

□

Bemerkung: Die Nullstellen von v_k trennen die Nullstellen von v_{k+1}.

Satz 7.3.5 *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 werden so bestimmt, daß v_{k+1} orthogonal zu $v_0, \dots, v_k$ wird. Für $\ell \leq k - 2$ gilt:

$$(\tilde{v}_{k+1}, v_\ell) = (x v_k, v_\ell) - \beta_k (v_k, v_\ell) - \gamma_k (v_{k-1}, v_\ell) = 0 \; .$$

Dies ist automatisch erfüllt, da $x v_\ell \in \mathcal{P}_{k-1}$.

Für $\ell = k - 1$ und $\ell = k$ erhält man die Gleichungen

$$(x v_k, v_{k-1}) - \gamma_k = 0 \quad , \quad (x v_k, v_k) - \beta_k = 0 \; .$$

Es gibt also β_k, γ_k, so daß $\tilde{v}_{k+1} \perp < v_0, \dots, v_k >$. 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: $[a,b] = [-1,+1]$. Für $w = 1$ erhält man $v_k = c_k P_k$ mit den Legendre-Polynomen P_k:

$$\begin{aligned} P_0 &= 1 & \qquad P_3 &= \tfrac{1}{2}(5x^3 - 3x) \\ P_1 &= x & P_4 &= \tfrac{1}{8}(35x^4 - 30x^2 + 3) \\ P_2 &= \tfrac{1}{2}(3x^2 - 1) \end{aligned}$$

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.

Kapitel 8

Numerische Integration und Differentiation

8.1 Die Formeln von Newton-Cotes

Sei $f \in C[a, b]$. Wir wollen das Integral $I = \int\limits_a^b f(x)dx$ 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, d.h. die Randpunkte des Intervalls sind Stützstellen:

$$\begin{aligned} x_k &= a + k \cdot h \quad , \qquad h = \tfrac{b-a}{n} \qquad k = 0, \ldots, n \\ I_n &= \sum_{k=0}^{n} A_k f(x_k) \end{aligned}$$

Eine Möglichkeit, die Gewichte A_k zu bestimmen, ist, die Funktion f durch ihr Interpolationspolynom p zu ersetzen, das an ihrer Stelle integriert wird:

In der Form von Lagrange lautet p

$$p(x) = \sum_{k=0}^{n} f(x_k)\omega_k(x) \quad , \quad \omega_k(x) = \prod_{\substack{\ell=0 \\ \ell \neq k}}^{n} \frac{x - x_\ell}{x_k - x_\ell} \, .$$

Es folgt

$$\begin{aligned} I_n &= \int_a^b p(x)dx = \sum_{k=0}^{n} \int_a^b \omega_k(x)dx f(x_k) \, , \\ A_k &= \int_a^b \omega_k(x)dx = \int_a^b \prod_{\substack{\ell=0 \\ \ell \neq k}}^{n} \frac{x - x_\ell}{x_k - x_\ell} dx \, . \end{aligned}$$

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"

$$\begin{aligned} a_0 &= \int_0^1 \frac{t-1}{-1} dt = \frac{1}{2} \, , \quad a_1 = \int_0^1 t dt = \frac{1}{2} \\ I_1 &= \frac{h}{2}(f(a) + f(b)) = \frac{b-a}{2}(f(a) + f(b)) \, . \end{aligned}$$

Das Integral wird also durch die Fläche eines Trapezes angenähert. Für $n = 2$ ergibt sich die trotz ihrer Einfachheit relativ genaue Simpson'sche Regel:

$$\begin{aligned} a_0 &= \frac{1}{3} \quad , \quad a_1 = \frac{4}{3} \quad , \quad a_2 = \frac{1}{3} \\ I_2 &= \frac{h}{3}\left(f(a) + 4f\left(\frac{b+a}{2}\right) + f(b)\right) \, . \end{aligned}$$

Die folgende Tabelle enthält die Koeffizienten a_j 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:

$$\begin{array}{rcll}
I & = & \int\limits_0^1 e^x dx = e - 1 & = \ 1.7183 \\
I_1 & = & \frac{1}{2}(1+e) & = \ 1.8591 \\
I_2 & = & \frac{1}{6}(1+4e^{1/2}+e) & = \ 1.7189 \\
I_3 & = & \frac{1}{8}(1+3e^{1/3}+3e^{2/3}+e) & = \ 1.7185
\end{array}$$

Man sieht, daß der Übergang von I_1 nach I_2 einen großen Gewinn an Genauigkeit ergibt.

Der folgende Satz gibt eine Abschätzung für den Fehler $|I - I_n|$:

Satz 8.1.1 *i) Sei $f \in C^{n+1}[a,b]$. Dann gilt*

$$|I - I_n| \leq h^{n+2} c_n \max_{[a,b]} |f^{(n+1)}(x)| \; ,$$

$$c_n = \frac{1}{(n+1)!} \int\limits_0^n \prod_{k=0}^n |t-k| dt \; .$$

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 V.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

$$|I - I_n| \le \frac{1}{(n+1)!} \int_a^b |w(x)|dx \; \max_{[a,b]} |f^{(n+1)}(x)| \, .$$

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

$$\begin{aligned}
\left|\int_a^b (f-p)(x)dx\right| &\leq \frac{1}{(n+1)!}\int_a^b |w(x)|dx \cdot \frac{nh}{2}\max_{[a,b]}|f^{(n+2)}(x)| \\
&= h^{n+2}c_n\frac{nh}{2}\max_{[a,b]}|f^{(n+2)}(x)| \\
&= h^{n+2}c_n^*\max_{[a,b]}|f^{(n+2)}(x)| \,.
\end{aligned}$$

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:

$$\int_{x_k}^{x_{k+1}} f(x)dx \simeq \frac{h}{2}(fx_k) + f(x_{k+1})) \,,$$

$$I = \int_{x_0}^{x_n} f(x)dx = \sum_{k=0}^{n-1}\int_{x_k}^{x_{k+1}} f(x)dx$$

$$
\begin{aligned}
&\simeq \frac{h}{2}\sum_{k=0}^{n-1}(f(x_k)+f(x_{k+1})) \\
&= \frac{h}{2}(f_0+2f_1+2f_2+\ldots+f_{n-1}+f_n) = \\
&= T_h
\end{aligned}
$$

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 8.1 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, indem wir die Simpson-Regel auf jeweils zwei aufeinanderfolgende Teilintervalle anwenden und anschließend summieren:

$$
\begin{aligned}
I = \int_a^b f(x)dx &= \sum_{k=0}^{\frac{n}{2}-1}\int_{x_{2k}}^{x_{2k+2}} f(x)dx \\
&\simeq \frac{h}{3}(f_0+4f_1+f_2+f_2+4f_3+f_4+\ldots) \\
&= \frac{h}{3}(f_0+4f_1+2f_2+4f_3+2f_4+\ldots+2f_{n-2}+4f_n) \\
&= S_h
\end{aligned}
$$

8.2 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 Trapezregel:

$$T_1(h) = \frac{h}{2}(f_0 + 2f_1 + \ldots + 2f_{n-1} + f_n)$$

mit

$$f_i := f(x_i)\ , \quad x_i = a + ih\ , \quad i = 0, \ldots, n\ , \quad h = \frac{b-ab}{n}\ .$$

Wir entwickeln nun $T_1(h)$ nach Potenzen von h.

Satz 8.2.1 *Sei $f \in C^{2m+2}[a,b]$. Dann gilt*

$$T_1(h) = I + c_1 h^2 + c_2 h^4 + \ldots + c_m h^{2m} + O(h^{2m+2})$$

mit

$$c_k(-1)^{k+1}\frac{B_{2k}}{(2k)!}(f^{(2k-1)}(b) - f^{(2k-1)}(a))$$

und den Bernoulli'schen Zahlen B_k:

$$B_1 = \frac{1}{6} \quad , \quad B_2 = \frac{1}{30} \quad , \quad B_3 = \frac{1}{42}, \ldots$$

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 f sogar $\in C^\infty(\mathbb{R}^1)$, 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

$$\begin{aligned} & 2^{2k} T_k(\frac{h}{2}) - T_k(h) = \\ & \quad = (2^{2k}-1)I + c_{k+1}(2^{-2}-1)h^{2k+2} + \ldots + c_m(2^{2k-2m})h^{2m} + \ \mathrm{O}(h^{2m+2}) \ . \end{aligned}$$

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:

$$\begin{array}{ccccccccc}
h & T_1(h) & & & & & & & \\
 & & \diagdown & & & & & & \\
\frac{h}{2} & T_1(\frac{h}{2}) & \text{———} & T_2(h) & & & & & \\
 & & \diagdown & & \diagdown & & & & \\
\frac{h}{4} & T_1(\frac{h}{4}) & \text{———} & T_2(\frac{h}{2}) & \text{———} & T_3(h) & & & \\
 & & \diagdown & & \diagdown & & \diagdown & & \\
\frac{h}{8} & T_1(\frac{h}{8}) & \text{———} & T_2(\frac{h}{4}) & \text{———} & T_3(\frac{h}{2}) & \text{———} & T_4(h) &
\end{array}$$

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: $I = \int\limits_0^1 e^x dx = 1.718281828$

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 Auswertung 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

$$T_k(\frac{h}{2}) - T_k(h) = c_k(2^{-2k} - 1)h^{2k} + \mathrm{O}(h^{2k+2}) ,$$

$$c_k h^{2k} = \frac{1}{2^{-2k} - 1}(T_k(\frac{h}{2}) - T_k(h)) + \mathrm{O}(h^{2k+2})$$

$$= \qquad \varepsilon_k(\frac{h}{2}) + \ \mathrm{O}(h^{2k+2}) \ .$$

Man prüft, ob $\varepsilon_k(\frac{h}{4} \sim 2^{-2k}\varepsilon_k(\frac{h}{2})$. Falls die Abweichung unter 10% liegt, wird 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 1 beweisen. Dazu einige Vorbereitungen. Seien B_k die Bernoulli-Polynome, d.h.

$$\begin{aligned} B_0 &= 1 , \\ B_k' &= B_{k-1} , \quad \int_0^1 B_k dx = 0 , \quad k = 1, 2, \dots . \end{aligned}$$

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 $B_{2k+1}(1) = B_{2k+1} = 0$. Dazu setzen wir $P_k(x) = (-1)^k B_k(1-x)$. Diese erfüllen die gleiche Rekursion wir die B_k. Also ist $P_k = B_k$, woraus die Behauptung folgt.

Satz 8.2.2 *(Euler'sche Summenformel) Sei $g \in C^{2m+2}[0,n]$, n ganz. Dann gilt*

$$\begin{aligned} & g(0) + g(1) + \cdots + g(n-1) + \frac{1}{2}g(n) - \int_0^n g dx \\ & = -\int_0^n \overline{B}_{2m+2} g^{(2m+2)} dx + \sum_{i=0}^m B_{2k+2}(0) \left(g^{(2n+1)}(n) - g^{(2k+1)}(0) \right) \ . \end{aligned}$$

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

$$\begin{aligned}
\int_0^n \overline{B}_1 g' dx &= \sum_{i=0}^{n-1} \int_i^{i+1} \overline{B}_1 g' dx = \sum_{i=0}^{n-1} \left\{ \overline{B}_1 g|_i^{i+1} - \int_i^{i+1} \overline{B}_1' g dx \right\} \\
&= \sum_{i=0}^{n-1} \left\{ \frac{g(i+1)+g(i)}{2} - \int_i^{i+1} g dx \right\} \\
&= \frac{1}{2} g(0) + g(1) + \cdots + g(n-1) + \frac{1}{2} g(n) - \int_0^n g dx \; .
\end{aligned}$$

Andererseits gilt

$$\begin{aligned}
\int_0^n \overline{B}_1 g' dx &= \int_0^n \overline{B}_2' g dx = -\int_0^n \overline{B}_2 g'' dx + \overline{B}_2 g' \Big|_0^n \\
&= -\int_0^n \overline{B}_3' g'' dx + \overline{B}_2 g' \Big|_0^n \\
&= -\int_0^n \overline{B}_3 g''' dx - \overline{B}_3 g'' \Big|_0^n + \overline{B}_2 g' |_0^n \; .
\end{aligned}$$

So fortfahrend erhält man schließlich

$$\begin{aligned}
\int_0^n \overline{B}_1 g' dx &= -\int_0^n \overline{B}_{2m+2} g^{(2m+2)} dx \\
&+ \left[-\overline{B}_{2m+2} g^{(2m+1)} - \overline{B}_{2m+1} g^{(2m)} + \cdots \; \overline{B}_2 g' \right]_0^n \; .
\end{aligned}$$

Aus dem Vergleich der beiden Darstellungen folgt die Behauptung.

□

Satz 1 folgt nun durch lineare Transformation des Intervalls $[0, b]$ auf $[0, n]$.

8.3 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 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 8.3.1 *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 $G_n f = 0$, $If \neq 0$.

□

Zur Konstruktion einer in $\mathcal{P}_{2n-1}$ exakten Formel G_n benutzen wir die nach dem Schmidt'schen Verfahren konstruierten orthonormalen Polynome p_n:

$$\int_a^b w p_n p_m dx = \begin{cases} 1 & m = n \\ 0 & m \neq n \end{cases}$$

Wir wissen: p_n hat in (a, b) genau n einfache Nullstellen.

Satz 8.3.2 *Es gibt Formeln G_n, welche auf $\mathcal{P}_{2n-1}$, exakt sind. Die x_j sind die Nullstellen von p_n und es gilt*

$$A_j = \int_a^b w(x) \prod_{\substack{i=1 \\ i \neq j}} \left(\frac{x - x_i}{x_j - x_i} \right)^2 dx \,.$$

Beweis: Sei G_n die Newton-Cotes Formel zu den Nullstellen $x_1, \ldots, x_n$ von p_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

$$f = q p_n + r \quad \text{mit} \quad q, r \in \mathcal{P}_{n-1} \qquad \text{(Division mit Rest)} \,.$$

Dann ist

$$\int_a^b w f dx = \int_a^b w q p_n dx + \int_a^b w r dx \,.$$

Das erste Integral verschwindet wegen der Orthogonalitätseigenschaften der p_n. Das zweite ist $G_n r$, weil G_n ja auf $\mathcal{P}_{n-1}$ exakt ist. Also folgt

$$\int_a^b w f dx = G_n r = G_n(r + q p_n) = G_n f \,,$$

weil p_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

$$\int_a^b w w_j^2 dx = G_n(w_j^2) = \sum_{k=1}^{n} A_k w_j^2(x_k) = A_j \ .$$

$\square$

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	$-\sqrt{\frac{1}{3}}$	$+\sqrt{13}$		1	1	
3	$-\sqrt{35}$	0	$\sqrt{35}$	$\frac{5}{9}$	$\frac{8}{9}$	$\frac{5}{9}$

$$I = \int_{-1}^{1} e^x dx = 2.350402$$

Die Simpson-Regel liefert:

$$I_2 = 2.362054$$

Dagegen ist mit gleich vielen Funktionswertungen

$$G_3 = 2.350337$$

8.4 Numerische Differentiation

Sei $f \in C^{p+1}(\mathbb{R}^1)$. Wir interessieren uns für eine Näherung für $f^{(k)}(0)$, 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

$$\sum_{i=-q}^{q} i^\ell \alpha_i = \begin{cases} 1 & , \quad \ell = k < p \\ 0 & , \quad \text{sonst} \end{cases} \quad , \qquad \ell = 0, \ldots, p \, .$$

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: $k = 1$:

$p = 1 \, , \; q = 1 \, , \; \alpha_{-1} = 0 \, , \; \alpha_0 = -1 \, , \; \alpha_1 = +1 \, ,$

$$D_h^{(1)} f = \frac{f(h) - f(0)}{h} \, , \quad f'(0) = D_h^{(1)} f + \mathrm{O}(h) \quad \text{für} \quad f \in C^2 \, .$$

$$p = 2\ ,\ q = 1\ ,\ \alpha_{-1} = -\frac{1}{2}\ ,\ \alpha_0 = 0\ ,\ \alpha_1 = \frac{1}{2}\ ,$$

$$D_h^{(1)} f = \frac{f(h) - f(-h)}{2h}\ , \quad f'(0) = D_h^{(1)} f + \mathrm{O}(h^2) \quad \text{für} \quad f \in C^3\ .$$

$k = 2$:

$$p = 3\ ,\ q = 1\ ,\ \alpha_{-1} = \frac{1}{2}\ ,\ \alpha_0 = -1\ ,\ \alpha_1 = \frac{1}{2}\ ,$$

$$D_h^{(2)} f = \frac{f(h) - 2f(0) + f(-h)}{h^2}\ ,\ f''(0) = D_h^{(2)} f + \mathrm{O}(h^2) \quad \text{falls} \ \ f \in C^4\ .$$

8.5 Der Fehler bei Integration und Differentiation

Sei $If = \int\limits_a^b f(x)dx$ zu berechnen. Steht anstelle von f nur eine Näherung $\tilde f$ mit relativem Fehler ε (also $|(f - \tilde f)| \le \varepsilon |f(x)|$) zur Verfügung, so kann nur die Näherung $I\tilde f$ für If berechnet werden. Es gilt

$$|If - \tilde I f| \le \int\limits_a^b |(f - \tilde f)(x)| dx \le \varepsilon I|f|\ . \tag{5.1}$$

Falls $I|f|$, $|If|$ die gleiche Größenordnung haben (z.B. falls $f \ge 0$), so haben $|If - \tilde I f|$, $\varepsilon |If|$ die gleiche Größenordnung, also $\tilde I f$ einen relativen Fehler der Größenordnung ε. Die Integration ist in diesem Fall also eine gut konditionierte Aufgabe.

Ist aber $I|f|$ viel größer als $|If|$ (z.B. wenn f eine stark oszillierende Funktion ist), dann ist der relative Fehler von $I\tilde f$ viel größer als ε. Die Integration ist dann schlecht konditioniert.

Wir untersuchen nun, ob die Berechnung von f durch eine Integrationsformel der Ordnung p

$$I_h f = \sum_{j=1}^{n} A_j f(x_j)\ , \quad |I_h f - If| \le ch^p$$

ein gutartiger Algorithmus ist. Es gilt

$$\begin{aligned} |I_h\tilde{f} - If| &\leq |I_h(\tilde{f} - f| + |(I_h - I)f| \\ &\leq \varepsilon \sum_{j=1}^{n} |A_j||f(x_j)| + ch^p \ . \end{aligned}$$

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 unvermeidliche, 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 Algorithmus gutartig. Dies ist nicht der Fall bei negativen Gewichten, denn dann kann

$$\varepsilon \sum |A_j||f(x_j)| \gg \varepsilon I|f|$$

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

$$\begin{aligned} |D_h^{(k)}\tilde{f} - f^{(k)}(0) &\leq |D_h^{(k)}(\tilde{f} - f)| + |D_h^{(k)}f - f^{(k)}(0)| \\ &\leq \varepsilon h^{-k}a + ch^{p+1-k} \ , \qquad (5.4) \\ a &= k! \sum_{i=-q}^{q} |\alpha_i||f(ih)| \ . \end{aligned}$$

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")

$$\varepsilon h^{-k} a = c h^{p+1-k}$$

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
$\frac{1}{h}(f(h) - f(0))$	1	$\varepsilon^{1/2}$	$\varepsilon^{1/2}$
$\frac{1}{2h}(f(h) - (-h))$	2	$\varepsilon^{1/3}$	$\varepsilon^{2/3}$
	4	$\varepsilon^{1/5}$	$\varepsilon^{4/5}$
	6	$\varepsilon^{1/7}$	$\varepsilon^{7/8}$

Kapitel 9

Gewöhnliche Differentialgleichungen

9.1 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

$$c'(x) = q(x)e^{-\int\limits_{x_0}^{x} p(x)dt} \,, \quad c(x) = y_0 + \int\limits_{x_0}^{x} q(t)e^{-\int\limits_{x_0}^{x} p(s)ds} dt$$

mit einer Konstanten c_0, und weiter

$$\begin{aligned} y &= \left(y_0 + \int\limits_{x_0}^{x} q(t)e^{\int\limits_{x_0}^{t} p(s)ds} dt \right) e^{\int\limits_{x_0}^{x} p(t)dt} \\ &= y_0 e^{\int\limits_{x_0}^{x} p(t)dt} + \int\limits_{x_0}^{x} q(t)e^{\int\limits_{t}^{x} p(s)ds} dt \,. \end{aligned}$$

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 $C^\infty(\mathbb{R}^2)$ ist).

4) $y' = y^{1/3}$, $y(0) = 0$ hat für jedes $c \geq 0$ die Lösung

$$y(x) = \begin{cases} \left(\frac{2}{3}(x - c)\right)^{3/2} & , \quad x \geq c \\ 0 & \quad \text{sonst} \,. \end{cases}$$

Die Lösung einer Anfangswertaufgabe braucht also nicht eindeutig zu sein.

Zur Formulierung eines Existenz- und Eindeutigkeitssatzes benötigen wir folgende

Definition 9.1.1 *$f \in C(D)$ 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 9.1.1 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.

9.2 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 9.2.1 *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

$$\begin{aligned} T_h(x_{k+1}) &= \frac{1}{4}(y(x_{k+1}) - y(x_k)) - f(x_k, y(x_0)) \\ &= y'(x_k) + \frac{h}{2}y''(\tilde{x}_k) - f(x_k, y(x_k)) , \quad \tilde{x}_k \in (x_k, x_{k+1}) \\ &= \frac{h}{2}y''(\tilde{x}_k) . \end{aligned}$$

   Also: Ist $y \in C^2(U)$, so ist das Euler-Verfahren konsistent mit der Ordnung 1.

2) Verbessertes Euler-Verfahren.

   Nach Trapezregel ist

$$\int_{x_k}^{x_{k+1}} f(x, y(x))dx = \frac{h}{2}\{f(x_k, y(x_k)) + f(x_{k+1}, y_{k+1}))\} + \mathrm{O}(h^2) \qquad (2.4)$$

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

$$\frac{1}{h} \int\limits_{x_k}^{x_{k+1}} f(x,y(x))dx = \frac{1}{2} \{f(x_k, y(x_k)) + f(x_{k+1}, y(x_k) + hf(x_k, y(x_k)))\} + \mathrm{O}(h^2) \; .$$

Mit der Schrittfunktion

$$f_h(x,y) = \frac{1}{2}\{f(x,y) + f(x+h, y+hf(x,y))\}$$

hat man also

$$\begin{aligned} T_h(x_{k+1}) &= \frac{1}{h}(y(x_{k+1}) - y(x_k)) - f_h(x_k, y(x_k)) \\ &= \frac{1}{h} \int\limits_{x_k}^{x_{k+1}} y'(x)dx - \frac{1}{h} \int\limits_{x_k}^{x_{k+1}} f(x,y(x))dx + \mathrm{O}(h^2) \\ &= \mathrm{O}(h^2) \; . \end{aligned}$$

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

$$
\begin{array}{lcl}
y_{k+1} & = & y_k + h(\gamma_1 f_1 + \ldots + \gamma_m f_m)(x_k, y_k)\ , \\
f_1(x,y) & = & f(x,y) \\
f_2(x,y) & = & f(x + \alpha_2 h, y + h\beta_{21} f_1(x,y)) \\
\vdots & & \\
f_m(x,y) & = & f(x + \alpha_m h, y + h[\beta_{m1} f_1 + \ldots + \beta_{m,m-1} f_{m-1}](x,y))\ .
\end{array}
$$

Man stellt alle Koeffizienten in dem Schema

$$
\begin{array}{l|l}
0 & \\
\alpha_2 & \beta_{21} \\
\vdots & \\
\alpha_m & \beta_{m1}\ \ldots\ \beta_{m-1} \\
\hline
 & \gamma_1\ \ldots\ \gamma_m
\end{array}
$$

zusammen. Man nimmt übrigens immer $\alpha_k = \sum\limits_{\ell=1}^{k-1} \beta_{k\ell}$ an.

Beispiele:

$m = 1$

$$\begin{array}{c|c} 0 & \\ \hline & 1 \end{array}$$

$y_{k+1} = y_k + hf(x_k, y_k)$
Euler, $\quad p = 1$

$m = 2$

$$\begin{array}{c|cc} 0 & & \\ 1 & 1 & \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

$$\begin{aligned} y_{k+1} &= y_k + \tfrac{h}{2}(f(x_k, y_k)+ \\ &\quad +f(x_k + h, y_k + hf(x_k, y_k))) \end{aligned}$$

Verbessertes Euler, $\quad p = 2$

$m = 4$

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ 1 & 0 & 0 & 1 & \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}$$

$$\begin{aligned} y_{k+1} &= y_k + \tfrac{h}{6}(f_1 + 2f_2 + 2f_3 + f_4) \\ f_1 &= f(x_k, y_k) \\ f_2 &= f(x_k + \tfrac{h}{2}, y_k + \tfrac{h}{2} f_1) \\ f_3 &= f(x_k + \tfrac{h}{2}, y_k + \tfrac{h}{2} f_2) \\ f_4 &= f(x_k + h, y_k + hf_3) \end{aligned}$$

(Standard) Runge-Kutta, $\quad p = 4$.

$m = 3$

$$\begin{array}{c|ccc} 0 & & & \\ \frac{1}{2} & \frac{1}{2} & & \\ 1 & -1 & 2 & \\ \hline & \frac{1}{6} & \frac{4}{6} & \frac{1}{6} \end{array}$$

$$\begin{aligned} y_{k+1} &= y_k + \tfrac{h}{6}(f_1 + 4f_2 + 2f_3) \\ f_1 &= f(x_k, y_k) \\ f_2 &= f(x_k + \tfrac{h}{2}, y_k + \tfrac{h}{2} f_1) \\ f_3 &= f(x_k + h\ , y_k - hf_1 + 2hf_2) \\ & \quad p = 3 \end{aligned}$$

9.3 Konvergenz von Einschrittverfahren

Vom lokalen Diskretisierungsfehler zu unterscheiden ist der globale Diskretisierungsfehler

$$D_h(x_h) = y_k - y(x_k)\ .$$

Definition 9.3.1 *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 9.3.1 *Seien q, d_k, a_k nichtnegative Zahlen mit*

$$d_{k+1} \le q d_k + a_k \qquad , \qquad k = 0, 1, \dots\ .$$

Dann gilt

$$d_k \le q^k d_0 + \sum_{j=0}^{k-1} q^{k-j-1} a_j\ .$$

Beweis: Durch vollständige Induktion.

Satz 9.3.1 *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*

$$|f_h(x, z_1) - f_h(x, z_2)| \le L|z_1 - z_2|\ ,$$

falls (x, z_1), $(x, z_2) \in D$.

Dann gilt: Solange $(x_k, y_k) \in D$ ist, besteht die Abschätzung

$$|y(x_k) - y_k| \le \frac{1}{L}\left(e^{L|x_k - x_0|} - 1\right) \max_{j=1}^{k} |T_h(x_j)|\ .$$

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 behauptete Abschätzung ergibt sich durch Aufsummieren der geometrischen Reihe.

□

Satz 9.3.2 *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 9.3.1 für alle $x_h \in U$, und die Konvergenz (von der Ordnung p) folgt.

9.4 Mehrschrittverfahren

Ein (lineares) m-Schrittverfahren hat die Form ($\alpha_m \neq 0$)

$$\begin{aligned} \sum_{\nu==}^{m} \alpha_\nu y_{k+\nu} &= h\sum_{\nu=0}^{m} \beta_\nu f(x_{k+\nu}, y_{k+\nu})\,, \quad k = 0, 1, \ldots \\ y_k &= \overline{y}_k \quad , \qquad\qquad k = 0, \ldots, m-1\,. \end{aligned}$$

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 <u>eine</u> 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 Einschrittverfahren heißt explizit, wenn $\beta_m = 0$. Dann läßt sich y_{k+m} unmittelbar durch $y_{k+m-1}, \ldots, y_k$ ausdrücken. Ist $\beta_m \neq 0$, so tritt y_{k+m} auch auf der rechten Seite auf und man muß y_{k+m} durch Lösen einer nichtlinearen Gleichung berechnen. Dies kann iterativ in der Form

$$\begin{aligned} \alpha_m y_{k+m}^{(t+1)} + \sum_{\nu=0}^{m-1} \alpha_\nu y_{k+\nu} &= h\beta_m f(x_{k+m}, y_{k+m}^{(t)}) \\ &+ h\sum_{\nu=0}^{m-1} \beta_\nu f(x_{k+\nu}, y_{k+\nu}) \end{aligned}$$

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 $\mathrm{O}(h)$ 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.

Lemma 9.4.1 *Es gilt für $q \geq 0$*

$$
\nabla^q y_k = \sum_{\nu=0}^{q} (-1)^\nu \binom{q}{\nu} y_{k-\nu} \quad , \quad y_{k-q} = \sum_{\nu=0}^{q} (-1)^\nu \binom{q}{\nu} \nabla^\nu y_k \; .
$$

Beweis: 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

$$
(I - T)^q = \sum_{\nu=0}^{q} \binom{q}{\nu} (-1)^\nu T^\nu \; .
$$

Wegen $I - T = \nabla$, $(T^\nu y)_k = y_{k-\nu}$ ist dies die erste Formel. Die zweite bekommen wir ganz entsprechend aus

$$
T^q = (I - \nabla)^q = \sum_{\nu=0}^{q} \binom{q}{\nu} (-\nabla)^\nu \; .
$$

□

Lemma 9.4.2 *Das Polynom p vom Grade $\leq q$ mit $p(x_{k-\ell}) = y_{k-\ell}$ $\ell = 0, \ldots, q$ ist*

$$
p(x) = \sum_{\nu=0}^{q} (-1)^\nu \binom{-s}{\nu} \nabla^\nu y_k \quad , \quad s = \frac{x - x_k}{h} \; .
$$

Hier sind die Binomialkoeffizienten für reelle s durch

$$\binom{s}{\nu} = \frac{1}{\nu!} s(s-1)\dots(s-(\nu-1)) \; .$$

erklärt.

Beweis: p ist ein Polynom vom Grade $\leq q$, denn es ist

$$\binom{-s}{\nu} = \frac{1}{\nu!}(s-\nu+1)\dots(-s) \in \mathcal{P}_\nu \; .$$

Weiter gilt für $0 \leq \mu \leq q$

$$\begin{aligned} p(x_{k-\mu}) &= \sum_{\nu=0}^{q} (-1)^\nu \binom{\mu}{\nu} \nabla^\nu y_k \\ &= \sum_{\nu=0}^{\mu} (-1)^\nu \binom{\mu}{\nu} \nabla^\nu y_k \\ &= y_{k-\mu} \end{aligned}$$

nach Lemma 9.4.1.

□

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 $x_k, \dots, x_{k+m}$ (implizites Verfahren) oder vom Grade $m-1$ an den Stellen $x_k, \dots, x_{k+m-1}$ (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:

$$\begin{aligned} y_{k+m} - y_{k+m-1} &= \int\limits_{x_{k+m-1}}^{x_{k+m}} p(x)dx\ , \quad p(x) = \sum_{\nu=0}^{m-1} (-1)^\nu \begin{pmatrix} -s \\ \nu \end{pmatrix} \nabla^\nu f_{k+m-1} \\ &= h(\gamma_0 f_{k+m-1} + \gamma_1 \nabla^1 f_{k+m-1} + \ldots + \gamma_{m-1} \nabla^{m-1} f_{k+m-1})\ . \\ \gamma_\nu &= \frac{1}{h} \int\limits_{x_{k+m-1}}^{x_{k+m}} (-1)^\nu \begin{pmatrix} -s \\ \nu \end{pmatrix} dx \quad , \quad s = \frac{x - x_{k+m-1}}{h}\ , \\ &= (-1)^\nu \int\limits_0^1 \begin{pmatrix} -s \\ \nu \end{pmatrix} ds \quad . \end{aligned}$$

Adams-Moulton:

$$\begin{aligned} y_{k+m} - y_{k+m-1} &= \int\limits_{x_{k+m-1}}^{x_{k+m}} p(x)dx\ , \quad p(x) = \sum_{\nu=0}^{m} (-1) \begin{pmatrix} -s \\ \nu \end{pmatrix} \nabla^\nu f_{k+m} \\ &= h(\gamma_0 f_{k+m} + \gamma_1 \nabla^1 f_{k+m} + \ldots + \gamma_m \nabla^m f_{k+m})\ , \end{aligned}$$

$$
\begin{aligned}
\gamma_\nu &= \frac{1}{h}(-1)^\nu \int\limits_{x_{k+m-1}}^{x_{k+m}} \binom{-s}{\nu} dx \ , \quad s = \frac{x - x_{k+m}}{h} \ , \\
&= (-1)^\nu \int\limits_{-1}^{0} \binom{-s}{\nu} ds \ .
\end{aligned}
$$

Nyström:

$$
\begin{aligned}
y_{k+m} - y_{k+m-2} &= h(\gamma_0 f_{k+m-1} + \gamma_1 \nabla^1 f_{k+m-1} + \ldots + \gamma_{m-1} \nabla^{m-1} f_{k+m-1}) \ , \\
\gamma_\nu &= (-1)^\nu \int\limits_{-1}^{+1} \binom{-s}{\nu} ds \ .
\end{aligned}
$$

Milne-Simpson:

$$
\begin{aligned}
y_{k+m} - y_{k+m-2} &= h(\gamma_0 f_{k+m} + \gamma_1 \nabla^1 f_{k+m} + \ldots + \gamma_m \nabla^m f_{k+m}) \ , \\
\gamma_\nu &= (-1)^\nu \int\limits_{-2}^{0} \binom{-s}{\nu} ds \quad .
\end{aligned}
$$

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 $m = 2$

$$
y_{k+2} - y_k = 2h f_{k+1} \ ,
$$

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

$$\begin{aligned} \sum_{\nu=0}^{m} \gamma_{\nu,m-\ell} \nabla^\nu y_{k+m} &= h f_{k+\ell}\ , \\ \gamma_{\nu,m-\ell} &= h \frac{d}{dx} (-1)^\nu \binom{-s}{\nu} \Big|_{x=x_{k+\ell}} \\ &= (-1)^\nu \frac{d}{ds} \binom{-s}{\nu} \Big|_{s=\ell-m}\ . \end{aligned}$$

Offenbar ist $\gamma_{0,m-\ell} = 0$. Einige $\gamma_{\nu,r}$ finden sich in folgender Tabelle:

$r \backslash \nu$	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

$$\begin{aligned} \nabla^1 y_{k+2} - \frac{1}{2} \nabla^2 y_{k+2} &= h f_{k+1} \qquad \text{oder} \\ y_{k+2} - y_k &= 2h f_{k+1}\ , \end{aligned}$$

also wieder die Mittelpunktsregel.

9.5 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

$$T_h(x_{k+m}) = \frac{1}{h}(y(x_{k+m}) - y(x_{k+m-1})) - \frac{1}{h}\int\limits_{x_{k+m-1}}^{x_{k+m}} p(x)dx \ ,$$

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

$$T_h(x_{k+m}) = \frac{1}{h}\int\limits_{x_{k+m-1}}^{x_{k+m}} (f(x, y(x)) - p(x))dx = \mathrm{O}(h^m)$$

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

$$\begin{aligned} T_h(x_{k+m}) &= h\left(\frac{4}{2} - \frac{1}{2}(1+a) - \frac{1}{2}(3-a)\right) y''(x_k) \\ &+ h^2\left(\frac{8}{6} - \frac{1}{6}(1+a) - \frac{1}{4}(3-a)\right) y'''(x_k) + \mathrm{O}(h^3) \\ &= h^2\left(\frac{5}{12} + \frac{a}{12}\right) y'''(x_k) + \mathrm{O}(h^3) \quad . \end{aligned}$$

Wir haben also Konsistenzordnung $p = 2$ für $a \neq -5$, sonst $p = 3$.

Wir kommen nun zu einem wichtigen Begriff. Numerische Experimente mit dem im letzten Beispiel angegebenen Verfahren ergeben befriedigende Resultate für $a = 0$ (d.h. Adams-Bashforth mit $m = 2$), aber unbrauchbare für $a = -5$. Die Konsistenzordnung kann also für die Konvergenz nicht, wie bei den Einschrittverfahren, das einzig Maßgebende sein. Wir werden sehen, daß bei Mehrschrittverfahren neben der Konsistenz die Stabilität notwendig für Konsistenz ist.

Sei für $\lambda \in \mathbb{C}$

$$\rho(\lambda) = \sum_{\nu=0}^{m} \alpha_\nu \lambda^\nu \quad , \qquad \sigma(\lambda) = \sum_{\nu=0}^{m} \beta_\nu \lambda^\nu \ .$$

Die Eigenschaften dieser beiden Polynome werden sich als für die Eigenschaften des Mehrschrittverfahrens wichtig erweisen.

Definition 9.5.1 *Ein Mehrtschrittverfahren heißt stabil, wenn für die Nullstellen λ von ρ folgende Bedingung erfüllt ist*

a) $|\lambda| \leq 1$

b) *Ist $|\lambda| = 1$, so ist λ einfach.*

Beispiele:

1) Adams-Bashforth.

Es ist $\rho(\lambda) = \lambda^{m-1}(\lambda - 1)$. Nullstellen sind $\lambda = 0 \quad ((m-1)$-fach) und $\lambda = 1$ (einfach). Also ist das Verfahren stabil.

2) Nyström.

Es ist $\rho(\lambda) = \lambda^{m-2}(\lambda^2 - 1)$. Nullstellen sind $\lambda = 0 \quad ((m-2)$-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 diese 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

	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 die instabilen Verfahren mit $a = 1$ und $a = 1.1$ treten offenbar Probleme auf.

Definition 9.5.2 *Ein Mehrschrittverfahren heißt konvergent, wenn für alle Startwerte* $\overline{y}_k$ *mit* $\lim_{h\to 0} |y(x_k) - \overline{y}_k| = 0, \quad k = 0, \ldots, m-1$

$$\lim_{h\to 0} \max_{x_k \in U} |y(x_k) - y_k| = 0$$

gilt. Es heißt konvergent von der Ordnung p, wenn aus $y(x_k) - \overline{y}_k = O(h^p)$, $k = 0, \dots, m-1$

$$\max_{x_k \in U} |y(x_k) - y_k| = O(h^p)$$

folgt.

Wir wollen zeigen, daß Konvergenz gleichbedeutend ist mit Konsistenz und Stabilität. Dazu benötigen wir einige einfache Tatsachen über Differenzengleichungen.

Unter einer linearen Differentialgleichung mit konstanten Koeffizienten versteht man eine Gleichung der Form

$$\sum_{\nu=0}^{m} \alpha_\nu z_{k+\nu} = c_k \quad , \quad k = 0, 1, \dots \quad .$$

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

$$\sum_{\nu=0}^{m} \alpha_\nu \lambda^{k+\nu} = \lambda^k \rho(\lambda) = 0 \quad , \quad k = 0, 1, \dots \quad ,$$

d.h. wenn $\rho(\lambda) = 0$ ist. Ist λ eine zweifache Nullstelle von ρ, so ist $\rho'(\lambda) = 0$ und damit auch

$$\begin{aligned}\sum_{\nu=0}^{m} \alpha_\nu (k+\nu) \lambda^{k+\nu} &= \lambda^k \left\{ k \sum_{\nu=0}^{m} \alpha_\nu \lambda^\nu + \sum_{\nu=0}^{m} \alpha_\nu \nu \lambda^\nu \right\} \\ &= \lambda^k \{ k\rho(\lambda) + \rho'(\lambda) \} = 0 \ ,\end{aligned}$$

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 $z_k = \lambda^k$, $z_k = k\lambda^k, \dots, z_k = k^{r-1}\lambda^k$ Lösungen sind. Damit hat man aber auch schon alle Lösungen der homogenen Differentialgleichungen gefunden.

Satz 9.5.1 *Seien* $\lambda_1, \dots, \lambda_n$ *die Nullstellen von* ρ *mit den Vielfachheiten* $r_1, \dots, r_n$. *Dann sind*

$$\lambda_j^k \ , \ k\lambda_j^k, \dots, k^{r_j - 1}\lambda_j^k \qquad , \quad j = 1, \dots, n$$

$m = r_1 + \ldots + r_n$ Lösungen der homogenen Differenzengleichung. Jede weitere Lösung z_k ist eine Linearkombination dieser Lösungen, d.h. es gilt mit Konstanten a_{jr}

$$z_k = \sum_{j=1}^{n} \sum_{r=0}^{r_j-1} a_{jr} k^r \lambda_j^k \quad .$$

Die a_{rj} sind eindeutig bestimmt.

Beweis: Sei $\alpha_m = 1$. Die Differenzengleichung kann dann in der Form

$$Z_{k+1} = AZ_k \ , \quad Z_k = \begin{bmatrix} z_k \\ \vdots \\ z_{k+m-1} \end{bmatrix} \quad , \quad A = \begin{bmatrix} 0 & 1 & 0 & & \cdots & & 0 \\ 0 & 0 & 1 & 0 & & \cdots & 0 \\ & & \ddots & \ddots & & & \\ & & & \ddots & \ddots & & \\ & & & & \ddots & \ddots & \\ 0 & & & & & 0 & 1 \\ -\alpha_0 & -\alpha_1 & & \cdots & & & -\alpha_{m-1} \end{bmatrix}$$

geschrieben werden.

Dann ist also $D_k = A^k D_0$. Ist $J = X^{-1}AX$ die Jordan'sche Normalform, so ist

$$D_k = X^{-1} J^k X D_0 \quad .$$

In unserem Fall haben die Jordan-Kästchen $J_1, \ldots, J_n$ zu den Eigenwerten $\lambda_1, \ldots, \lambda_n$ die Dimensionen $r_1, \ldots, r_n$, vergleiche Übungsaufgabe 3. Die Potenzen von J_ℓ haben wir schon in 6.2 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. D_k ist also wirklich als Linearkombination der genannten Ausdrücke darstellbar.

$\square$

Als wichtige Folgerung aus Satz IX.9.1 haben wir:

Satz 9.5.2 *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 9.5.3 *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 1.3.2 gibt es eine Norm $\| \; \|_{m-r}$ in $\mathbb{C}^{m-r}$ mit $\|B\|_{m-r} \le \rho(A)$. 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 9.5.4 *f erfülle in einer Umgebung D der Kurve $(x, y(x))_{x \in U}$ eine Lipschitz-Bedingung.*

Das Mehrschrittverfahren sei stabil. Dann gibt es Konstanten C_1, C_2, $h_0 > 0$, so daß für $h < h_0$

$$|y(x_k) - y_k| \le C_1 e^{C_2(x_k - x_0)} \left\{ \max_{k=0}^{m-1} |y(x_k) - y_k| + \max_{j=m}^{k} |T_h(x_j)| \right\} \; ,$$

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 9.5.3 gibt es also eine Vektornorm $\| \ \|$, so daß $\|A\| \leq 1$.

$$\|D_{k+1}\| \leq \|D_k\| + h|c_k|\|B\| .$$

Solange $(x_k, y_k) \in D$ gilt, haben wir

$$|c_k| \leq L \sum_{\nu=0}^{m} |\beta_\nu||d_{k+\nu}| + |T_h(x_{k+m})| \leq K(\|D_k\|)|T_h(x_{k+m})|$$

mit einer geeigneten Konstanten K. Für $h\|B\|K < 1$ ist also

$$\|D_{k+1}\| \le q\|D_k\| + ha_k \quad ,$$
$$q = (1+h\|B\|K)/(1-h\|B\|K)\ , \quad a_k = |T_h(x_{k+m})|\|B\|/(1-h\|B\|K)\ .$$

Nach Lemma 3.1 folgt

$$\|D_k\| \le q^k\|D_0\| + h\sum_{j=0}^{k-1} q^{k-j-1}a_j\ .$$

Nun benutzen wir die Ungleichung

$$\frac{1+x}{1-x} \le 1+4x \quad , \quad 0 \le x \le \frac{1}{2}\ .$$

Dann wird für $h\|B\|K \le 1/2 \quad q \le 1+4h\|B\|K$ und damit

$$q^k \le \left(1+4\|B\|K\frac{x_k-x_0}{k}\right)^k \le e^{4\|B\|K(x_k-x_0)}\ .$$

Für D_k ergibt sich nun durch Aufsummieren der geometrischen Reihe

$$\begin{aligned}\|D_k\| &\le q^k\|D_0\| + h\frac{q^k-1}{q-1}\max_{j=0}^{k-1} a_j \\ &\le e^{4\|B\|K(x_k-x_0)}\left(\|D_0\| + \frac{1}{2K}\max_{j=0}^{k-1}|T_h(x_{j+m})|\right)\ .\end{aligned}$$

Da in $\mathbb{R}^m$ alle Normen äquivalent sind, folgt die behauptete Ungleichung.

□

Hieraus folgt wie in §3 sofort

Satz 9.5.5 *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 9.5.6 *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

$$\sum_{\nu=0}^{m} \alpha_\nu y_{k+\nu} = 0 \ , \quad k = 0, 1, \ldots \ , \quad y_k = \overline{y}_k \ , \quad k = 0, \ldots, m-1 \ .$$

Nach Satz 9.5.1 ist

$$y_k = k^{r-1}\lambda^k \varepsilon(h) \quad , \qquad k = 0, 1, \ldots \ ,$$

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 $|\lambda| = 1$.

$\square$

9.6 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 9.6.1 *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

$$\begin{array}{rcl} y(x_{k+\nu}) & = & y(x_k) + \nu h y'(x_k) + \ldots + \frac{(\nu h)^p}{p!} y^{(p)}(x_k) + \mathrm{O}(h^{p+1}) \\ y'(x_{k+\nu}) & = & y'(x_k) + \nu h y''(x_k) + \ldots + \frac{(\nu h)^{p-1}}{(p-1)!} y^{(p)}(x_k) + \mathrm{O}(h^p) \, . \end{array}$$

Dies ergibt für den lokalen Diskretisierungsfehler

$$\begin{aligned} T_h(x_{k+m}) &= \frac{1}{h} \sum_{\nu=0}^{m} \alpha_\nu y(x_{k+\nu}) - \sum_{\nu=0}^{m} \beta_\nu y'(x_{k+\nu}) \\ &= \frac{1}{h} C_0 y(x_k) + C_1 y'(x_k) + \ldots + h^{p-1} C_p y^{(p)}(x_k) + \mathrm{O}(h^p) \, , \end{aligned}$$

$$\begin{aligned} C_0 &= \sum_{\nu=0}^{m} \alpha_\nu \quad , \\ C_1 &= \sum_{\nu=0}^{m} \nu \alpha_\nu - \sum_{\nu=0}^{m} \beta_\nu \quad , \\ &\vdots \\ C_p &= \frac{1}{p!} \sum_{\nu=0}^{m} \nu^p \alpha_\nu - \frac{1}{(p-1)!} \sum_{\nu=0}^{m} \nu^{p-1} \beta_\nu \, . \end{aligned}$$

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 $z \to 0$

$$\begin{aligned}
\chi(z) &= \frac{1}{z}\sum_{\nu=0}^{m}\alpha_\nu \sum_{\mu=0}^{p}\frac{(\nu z)^\mu}{\mu!} - \sum_{\nu=0}^{m}\beta_\nu \sum_{\mu=0}^{p-1}\frac{(\nu z)^\mu}{\mu!} + \mathrm{O}(z^p) \\
&= \frac{1}{z}C_0 + C_1 + \ldots + z^{p-1}C_p + \mathrm{O}(z^p) \ .
\end{aligned}$$

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 9.6.2 *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 negativem Realteil. Nach (ii) ist r also von der Form

$$r(w) = a_1 w + a_2 w^2 + \ldots + a_m w^m$$

mit $a_1 \neq 0$, und a_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 9.6.1 ist die Ordnung p der Nullstelle von f bei 0 gleich der Konsistenzordnung des Verfahrens. Offenbar ist

$$\begin{aligned} f(w) &= \frac{r(w)}{\ell n \frac{1+w}{1-w}} - s(w) \\ &= b_0 + b_1 w + \ldots + b_{p-1} w^{p-1} + \ldots - s(w) \; . \end{aligned}$$

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 unterschreiben 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.

$\square$

Definition 9.6.1 *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.

Kapitel 10

Numerik partieller Differentialgleichungen

10.1 Anfangswertaufgaben partieller Differentialgleichungen

Treten in einer Differentialgleichung Ableitungen nach mehr als einer Variablen auf, so spricht man von partieller Differentialgleichung. Bei den Anfangswertaufgaben spielt eine dieser Variablen die Rolle der Zeit. Wir bezeichnen sie daher mit t.

Wir betrachten die partielle Differentialgleichung

$$u_t = Au \qquad \text{in} \quad [a,b] \times [0,\infty)$$

für die vektorwertige Funktion $u = (u_1, \ldots, u_m)^T, \;\; u_i = u_i(x,t)$. A ist der Differentialausdruck r-ter Ordnung

$$Au = \sum_{\rho=0}^{r} A_\rho(x) \frac{\partial^\rho}{\partial x^\rho} u$$

der Ordnung r mit (m,m)-Matrizen $A(x)$. Zur eindeutigen Festlegung von u gehört noch eine Anfangsbedingung

$$u(x,0) = u_0(x) \qquad , \qquad a \le x \le b$$

und unter Umständen, in Abhängigkeit von A, Randbedingungen am Rande von $[a, b]$, also Bedingungen für die Funktion

$$u(a,t) \quad , \quad u(b,t) \quad , \qquad t \geq 0 \ .$$

Dies ist die Anfangswertaufgabe oder auch Anfangsrandwertaufgabe.

Beispiele:

1) $\quad u_t = u_x \ , \ \ x \in \mathbb{R}^1$.

Die Gleichung verlangt, daß u entlang der Geraden $t + x = C$ konstant ist. Also ist

$$u(x,t) = u(x+t, 0) = u_0(x+t)$$

die Lösung der Anfangswertaufgabe.

2) $\quad u_t = Du_{xx}, \ \ 0 \leq x \leq 1$. Dies ist die Wärmeleitungs- oder Diffusionsgleichung. Sie beschreibt die Temperatur $u(x,t)$ eines Stabes der Länge 1 an der Stelle x zur Zeit t; $u_0(x)$ ist demgemäß die Temperatur zur Zeit 0. Als Randbedingungen kommen z.B. in Frage

$$\begin{aligned} u(0,t) \quad , \quad u(1,t) \quad &= \quad 0 \quad \text{(Enden gekühlt)} \\ u_x(0,t) \quad , \quad u_x(1,t) \quad &= \quad 0 \quad \text{(Enden wärmeisoliert).} \end{aligned}$$

Die exakte Lösung für die ersten Randbedingungen ist

$$\begin{aligned} u(x,t) \quad &= \quad \sum_{\ell=1}^{\infty} \hat{u}_\ell \sin(\ell\pi x) e^{-\pi^2\ell^2 tD} \quad , \\ \hat{u}_\ell \quad &= \quad 2\int_0^1 u_0(x) \sin \ell\pi x dx \quad , \end{aligned}$$

vgl. Satz 5.6.1.

3) $\quad u_{tt} = c^2 u_{xx}, \ \ 0 \leq x \leq 1$. Dies ist die Wellengleichung. Sie beschreibt die Auslenkung $u(x,t)$ zur Zeit t an der Stelle x einer schwingenden Saite der Länge 1. Zur eindeutigen Festlegung von u braucht man demgemäß die Auslenkung zur Zeit 0 sowie die Geschwindigkeit zur Zeit 0, also

$$u(x,0) = u_0(x) \ , \qquad u_t(x,0) = u_1(x) \ .$$

Als Randbedingung tritt etwa auf

$$u(0,t) = u(1,t) = 0 \qquad \text{(Enden fest eingespannt)} \ .$$

Die exakte Lösung unter diesen Randbedingungen ist

$$\begin{aligned} u(x,t) &= \sum_{\ell=0}^{\infty} (\hat{u}_{0\ell} \cos(c\ell\pi t) + \hat{u}_{1\ell} \sin(c\ell\pi t)) \sin \ell\pi x \ , \\ \hat{u}_0 &= 2\int_0^1 u_0(x) \sin(\ell\pi x) dx \ , \\ \hat{u}_1 &= \frac{2}{c\ell\pi} \int_0^1 u_1(x) \sin(\ell\pi x) dx \ , \quad \ell > 0 \ . \end{aligned}$$

Man kann dieses Problem in unser Schema einordnen, wenn man $v_1 = u_x$, $v_2 = u_t$ setzt. Man erhält dann das System

$$\begin{pmatrix} v_1 \\ v_2 \end{pmatrix}_t = \begin{pmatrix} 0 & 1 \\ c^2 & 0 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}_x \quad , \quad \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} (x,0) = \begin{pmatrix} u_0(x) \\ u_1(x) \end{pmatrix} \ .$$

4) Allgemeiner betrachten wir das hyperbolische System 1. Ordnung

$$u_t + A u_x = 0 \quad , \qquad u = \begin{pmatrix} u_1 \\ \vdots \\ u_m \end{pmatrix}$$

mit einer konstanten (m,m)-Matrix A, welche m verschieden reelle Eigenwerte $\lambda_1, \ldots, \lambda_m$ hat. Dann ist $A = Y^{-1} J Y$ mit

$$J = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_m \end{pmatrix} \ , \quad Y = \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix} \ , \quad y_i(A - \lambda_i I) = 0 \ .$$

Mit $v = Yu$ entsteht

$$v_t + J v_x = 0 \ .$$

Dies ist ein zerfallendes System von m Differentialgleichungen

$$\frac{\partial}{\partial t} v_i + \lambda_i \frac{\partial}{\partial x} v_i = 0 \ .$$

Entlang der Geraden $x = \lambda_i t + C$ ist v_i konstant, denn es ist

$$\frac{d}{dt} v_i(x,t) = \frac{\partial}{\partial t} v_i(x,t) + \lambda_i \frac{\partial}{\partial x} v_i(x,t) = 0 \quad .$$

Die Lösung der Anfangswertaufgabe ist nun wie in Beispiel 1 möglich: Für jeden Punkt (x_0, t_0) der x-ten Ebene bestimmt man die Gerade $x = \lambda_i t_0 + C$, welche durch diesen Punkt geht (es ist diejenige mit $x_0 = \lambda_i t_0 + C$) und sucht deren Schnittpunkt mit $t = 0$ (dies ist in $x = C = x_0 - \lambda_i t_0$). Dann ist

$$v_i(x_0, t_0) = v_i(x_0 - \lambda_i t_0, 0)$$

und dies ist aus der Anfangsbedingung bekannt.

Die Geraden $x = \lambda_i t + C$ nennt man Charakteristiken. Man sieht, daß $u(x_0, t_0)$ nur von den Anfangswerten $u_0(x_0 - \lambda_i t_0),\ i = 1, \ldots, m$ abhängt. Man nennt daher das Intervall $[\min(x_0 - \lambda_i t_0), \max(x_0 - \lambda_i t_0)]$ das Abhängigkeitsgebiet von (x_0, t_0).

Für die Wellengleichung haben wir $A = \begin{pmatrix} 0 & 1 \\ c^2 & 0 \end{pmatrix}$ und damit $\lambda_{1,2} = \pm c$. Die Charakteristiken sind also Geraden mit der Steigung $\pm\frac{1}{c}$. Das Abhängigkeitsgebiet von (x_0, t_0) ist $[x_0 - ct_0, x_0 + ct_0]$. Eine Störung, die zur Zeit 0 bei x_1 ist, hat also zur Zeit $t_0 = (x_1 - x_0)/c$ den Punkt x_0 erreicht. c hat also die Bedeutung einer Ausbreitungsgeschwindigkeit.

10.2 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

$$t_\ell = \ell \Delta t \quad , \quad \ell = 0,1,\dots \quad , \quad x_k = kh \ , \quad k = 0,\dots,n \ , \quad h = \frac{1}{n}$$

ein und suchen für $u_k, t_\ell)$ 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

$$\begin{aligned}
u_{k0} &= u_0(x_k) \quad , \quad k = 0,\dots,n \\
u_{0\ell} &= u_{n\ell} = 0 \quad , \quad \ell = 1,2,\dots \quad .
\end{aligned}$$

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 Anfangswertbedingungen gegeben.

Als Beispiel führen wir die Rechnung durch für die Anfangswerte

$$u_{k,0} = \left\{ \begin{array}{lll} 1 & , & k = K \ , \quad K+1 \\ 0 & , & \text{sonst} \quad . \end{array} \right.$$

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 entscheiden von λ abhängt.

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

$$\begin{aligned} \frac{1}{\Delta t}(u_{k,1} - u_{k,-1}) &= u_1(x_k) \quad , \\ u_{k,0} &= u_0(x_k) \end{aligned}$$

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.

Empfohlenes Buch: J. Werner, Numerische Mathematik 1 + 2, Vieweg 1992, je DM 38,–

Weitere Literatur:

1 Neuere Lehrbücher

- **Deuflhard - Hohmann**: Numerische Mathematik. Eine algorithmisch orientierte Einführung. Walter de Gruyter 1991.
- **Hämmerlin - Hoffmann**: Numerische Mathematik. Springer 1989.
- **Stoer**: Numerische Mathematik I. Springer 1989.
- **Stoer - Bulirsch**: Numerische Mathematik II. Springer 1990.
- **Stummel - Hainer**: Praktische Mathematik. Teubner 1982.
- **Schwarz**: Numerische Mathematik. Teubner 1988.
- **Schaback - Werner, H.**: Numerische Mathematik. Springer 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. 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.
- **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).