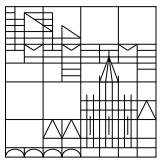


Universität Konstanz

Fachbereich Mathematik
und Statistik

Numerik I

Eberhard Luik

Vorwort

Der vorliegende Text ist das Skript zur Vorlesung "Numerik I". Diese Lehrveranstaltung gibt eine Einführung in grundlegende numerische Algorithmen. Es werden folgende Themen behandelt: Lineare Gleichungssysteme, linearer Ausgleich, lineare Optimierung, Interpolation, Nullstellenverfahren, Lösung von Anfangswertaufgaben, Numerische Integration, Iterationsverfahren für lineare bzw. nichtlineare Gleichungssysteme, Eigenwertaufgaben, Minimierung sowie Stabilitäts- und Störungsfragen. Anhand von anwendungsbezogenen Beispielen aus den Bereichen Wirtschaftswissenschaften bzw. Naturwissenschaften wird die Wirkungsweise der Algorithmen nochmals verdeutlicht.

Vorlesung und Übungen bilden eine Einheit. Deshalb ist die regelmäßige und aktive Teilnahme an den Übungen für das Verständnis des vorliegenden Textes und im Hinblick auf die anstehende Klausur unerlässlich.

Inhaltsverzeichnis

1 Lineare Gleichungssysteme

1.1 Einleitung

Gegeben sei das lineare Gleichungssystem

$$\begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \quad = \quad \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}$$

oder kurz

$$Ax \;=\; b \quad .$$

In der Linearen Algebra befasst man sich mit der Frage der Lösbarkeit und der Eindeutigkeit solcher linearer Gleichungssysteme. Wir setzen hier die eindeutige Lösbarkeit voraus (zum Beispiel durch die Forderung $\det A \neq 0$).

Zur numerischen Lösung linearer Gleichungssysteme gibt es zwei Typen von Verfahren:

direkte Verfahren: Sie lösen das Problem nach endlich vielen Schritten, so dass kein Formelfehler auftritt. Dagegen können Rundungsfehler das Ergebnis erheblich verfälschen.

indirekte Verfahren: Die Lösung des Problems wird durch Iteration näherungsweise bestimmt. Hier treten sowohl Formelfehler (Abbrechfehler) als auch Rundungsfehler auf. Trotzdem können iterative Verfahren durchaus vorteilhaft sein (insbesondere bei großem n).

In diesem Paragraphen werden die direkten Verfahren behandelt; mit den indirekten Verfahren beschäftigen wir uns in Abschnitt 9.3.

1.2 Gestaffelte Gleichungssysteme

Ist A eine untere (oder obere) Dreiecksmatrix und sind alle Diagonalelemente von Null verschieden, so kann man das lineare Gleichungssystem $Ax = b$ leicht rekursiv lösen. Für

$$A \;=\; \begin{pmatrix} a_{11} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ a_{n1} & \cdots & \cdots & a_{nn} \end{pmatrix}$$

gilt (Vorwärtsauflösung)

$$\begin{aligned} x_1 &= \frac{b_1}{a_{11}} \quad , \\ &\vdots \\ x_k &= \frac{b_k - a_{k1}x_1 - \cdots - a_{k,k-1}x_{k-1}}{a_{kk}} \qquad (k = 1, \ldots, n) \; . \end{aligned}$$

Im Falle einer oberen Dreiecksmatrix

$$A = \begin{pmatrix} a_{11} & \cdots & \cdots & a_{1n} \\ 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & a_{nn} \end{pmatrix}$$

ergibt sich (Rückwärtsauflösung)

$$\begin{aligned} x_n &= \frac{b_n}{a_{nn}} \quad , \\ &\vdots \\ x_k &= \frac{b_k - a_{k,k+1}x_{k+1} - \cdots - a_{kn}x_n}{a_{kk}} \qquad (k = n, \ldots, 1) \; . \end{aligned}$$

Deshalb versucht man bei den direkten Verfahren, das lineare Gleichungssystem $Ax = b$ auf eine der obigen Formen zu bringen. Dafür gibt es mehrere Möglichkeiten.

1.3 Die LR-Zerlegung

Sei A eine reguläre $n \times n$ - Matrix. Die *LR-Zerlegung* von A liefert eine Darstellung $A = LR$. Dabei ist L eine untere Dreiecksmatrix und R eine obere Dreiecksmatrix mit $r_{ii} \neq 0, \; i = 1, \ldots, n$:

$$\begin{pmatrix} a_{11} & \cdots & \cdots & a_{1n} \\ \vdots & & & \vdots \\ \vdots & & & \vdots \\ a_{n1} & \cdots & \cdots & a_{nn} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ l_{21} & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ l_{n1} & \cdots & l_{n,n-1} & 1 \end{pmatrix} \begin{pmatrix} r_{11} & \cdots & \cdots & r_{1n} \\ 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & r_{nn} \end{pmatrix} \tag{1}$$

Für das lineare Gleichungssystem $Ax = b$ ergibt sich

$$Ax = b \quad \Leftrightarrow \quad (LR)x = b \quad \Leftrightarrow \quad L(Rx) = b \quad \Leftrightarrow \quad Ly = b \text{ mit } y := Rx \; .$$

Fazit: Löse zuerst das gestaffelte Gleichungssystem

$$Ly = b \quad \text{(durch Vorwärtsauflösen)}$$

und dann das gestaffelte Gleichungssystem

$$Rx = y \quad \text{(durch Rückwärtsauflösen).}$$

Dabei ist auf diese Reihenfolge zu achten.

Wir müssen uns natürlich fragen, wann die LR-Zerlegung von A überhaupt existiert.

Satz: Ist A regulär (also $\det A \neq 0$), so kann man durch vorherige Zeilenvertauschungen bei A stets erreichen, dass die LR-Zerlegung existiert.

Beweis: vgl. Hämmerlin-Hoffmann [9].

Matrizentechnisch bedeutet dies

$$PA \;=\; LR \quad ,$$

wobei P ein Produkt von Permutationsmatrizen ist. Unter einer Permutationsmatrix versteht man eine Matrix der Form

$$P_{ij} \;:=\; \begin{pmatrix} 1 & & & & & & & \\ & 0 & & & & 1 & & \\ & & 1 & & & & & \\ & & & \ddots & & & & \\ & & & & 1 & & & \\ & 1 & & & & 0 & & \\ & & & & & & 1 & \end{pmatrix} \quad \begin{matrix} \\ \\ \\ \\ \\ \leftarrow \;\; i\text{-te Zeile} \\ \\ \leftarrow \;\; j\text{-te Zeile} \\ \\ \end{matrix}$$

Das Produkt $P_{ij}A$ vertauscht in A die i-te mit der j-ten Zeile.

Somit ergibt sich für das lineare Gleichungssystem $Ax = b$:

$$Pb \;=\; PAx \;=\; LRx \quad ,$$

und man löst dann nacheinander die Systeme

$$Ly \;=\; Pb \;\text{ und }\; Rx \;=\; y \quad .$$

Berechnung der LR-Zerlegung

A sei regulär und besitze eine LR-Zerlegung $A = LR$ mit L und R wie oben. Durch geschicktes Vorgehen beim Ausmultiplizieren von LR in (1) ergeben sich Formeln zur Berechnung der l_{ij} bzw. r_{ij} (*Verfahren von Crout*):

1. Für die 1. Zeile von A folgt

$$a_{1k} = r_{1k} \cdot 1 \quad (k = 1, \ldots, n) \; .$$

2. Für die 1. (Rest-)Spalte von A gilt

$$a_{i1} = l_{i1} \cdot r_{11} \quad (i = 2, \ldots, n) \; .$$

Daraus folgt (wegen A regulär ist $r_{11} \neq 0$)

$$l_{i1} \;=\; \frac{a_{i1}}{r_{11}} \; .$$

3. Für die 2. (Rest-)Zeile von A gilt

$$a_{2k} \;=\; l_{21} r_{1k} + l_{22} r_{2k} \quad (k = 2, \ldots, n) \; .$$

Da $l_{22} = 1$ und l_{21} bzw. r_{1k} schon berechnet sind, folgt daraus

$$r_{2k} \;=\; a_{2k} - l_{21} r_{1k} \quad (k = 2, \ldots, n) \; .$$

4. Für die 2. (Rest)-Spalte gilt

$$a_{i2} \;=\; l_{i1}r_{12} + l_{i2}r_{22} \quad (i = 3, \ldots, n)\ .$$

Da $l_{i1},\ r_{12},\ r_{22}$ schon berechnet sind und da $r_{22} \neq 0$ gilt, ergibt sich daraus

$$l_{i2} \;=\; \frac{a_{i2} - l_{i1}r_{12}}{r_{22}} \quad (i = 3, \ldots, n)\ .$$

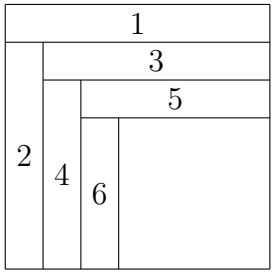
Allgemein erhält man für die j-te (Rest-)Zeile

$$r_{jk} \;=\; a_{jk} - l_{j1}r_{1k} - l_{j2}r_{2k} - \cdots - l_{j,j-1}r_{j-1,k} \quad (k = j, \ldots, n)$$

und für die j-te (Rest-)Spalte

$$l_{ij} \;=\; \frac{a_{ij} - l_{i1}r_{1j} - \cdots - l_{i,j-1}r_{j-1,j}}{r_{jj}} \quad (i = j+1, \ldots, n).$$

Schematisch sieht das Crout-Verfahren wie folgt aus:



Dies lässt sich beim Programmieren speicherplatzsparend ausnutzen.

Beispiel: Gegeben sei die Matrix

$$A \;=\; \begin{pmatrix} 3 & 1 & 6 \\ 1 & 1 & 1 \\ 2 & 1 & 3 \end{pmatrix} \quad .$$

Dann erhält man folgende LR-Zerlegung:

$$L \;=\; \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{3} & 1 & 0 \\ \frac{2}{3} & \frac{1}{2} & 1 \end{pmatrix} \quad , \qquad R \;=\; \begin{pmatrix} 3 & 1 & 6 \\ 0 & \frac{2}{3} & -1 \\ 0 & 0 & -\frac{1}{2} \end{pmatrix} \quad .$$

Nebenrechnungen:

$$\begin{aligned} r_{22} &= a_{22} - l_{21}r_{12} \;=\; 1 - \tfrac{1}{3} \;=\; \tfrac{2}{3}\ , \\ r_{23} &= a_{23} - l_{21}r_{13} \;=\; 1 - \tfrac{1}{3}6 \;=\; -1\ , \\ l_{32} &= (a_{32} - l_{31}r_{12})/r_{22} \;=\; \left(1 - \tfrac{2}{3}\right)/\tfrac{2}{3} \;=\; \tfrac{1}{2}\ , \\ r_{33} &= a_{33} - l_{31}r_{13} - l_{32}r_{23} \;=\; 3 - \tfrac{2}{3}6 - \tfrac{1}{2}(-1) \;=\; -\tfrac{1}{2}\ . \end{aligned}$$

Aufwand
Der Aufwand für die LR-Zerlegung beträgt ungefähr $\frac{n^3}{3}$ Punktoperationen (Multiplikationen bzw. Divisionen). Er wächst sehr rasch mit n und kann daher für große n selbst für Hochleistungsrechner zum Problem werden. Der Aufwand für das Auflösen eines gestaffelten Gleichungssystems beträgt ungefähr $\frac{n^2}{2}$ Punktoperationen.

Anmerkungen:
1. Ist die LR-Zerlegung bestimmt, so lassen sich mehrere Gleichungssysteme mit derselben Matrix, aber mit verschiedenen rechten Seiten, sehr rasch lösen.
2. In der Numerik vermeidet man die explizite Berechnung der inversen Matrix A^{-1}. Denn zum einen erfordert die Berechnung der Inversen ungefähr n^3 Punktoperationen und die Multiplikation von A^{-1} mit einem Vektor n^2 Punktoperationen. Zum anderen ist die Berechnung von A^{-1} mit Stabilitätsproblemen verbunden: ein kleiner Fehler in den Koeffizienten der Matrix A kann zu einer starken Abweichung bei der inversen Matrix A^{-1} führen.

1.4 Das Cholesky-Verfahren

Positiv definite Matrizen
Definition: Eine Matrix $A \in \mathbb{R}^{n\times n}$ heißt *positiv definit*, falls gilt

$$x^T A\, x \;>\; 0 \;\text{ für alle } x \in \mathbb{R}^n \setminus \{0\}\ .$$

Für die Numerik haben symmetrische, positiv definite Matrizen den Vorteil, dass eine einfachere Variante der LR-Zerlegung existiert, die sich mit ungefähr der Hälfte des Aufwandes berechnen lässt. Ferner ist eine größere Stabilität gewährleistet.

Die Cholesky-Zerlegung
Satz: Es sei $A \in \mathbb{R}^{n\times n}$ symmetrisch und positiv definit. Dann existiert eine Zerlegung der Form

$$A \;=\; L\cdot L^T$$

mit einer eindeutig bestimmten regulären unteren Dreiecksmatrix $L = (l_{ij})$ und $l_{ii} > 0$ für $i = 1, \ldots, n$.

Man hat somit eine Zerlegung der Gestalt

$$\begin{pmatrix} a_{11} & \cdots & \cdots & a_{1n} \\ \vdots & & & \vdots \\ \vdots & & & \vdots \\ a_{n1} & \cdots & \cdots & a_{nn} \end{pmatrix} = \begin{pmatrix} l_{11} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ l_{n1} & \cdots & \cdots & l_{nn} \end{pmatrix} \begin{pmatrix} l_{11} & \cdots & \cdots & l_{n1} \\ 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & l_{nn} \end{pmatrix} \quad . \tag{2}$$

Diese Darstellung heißt die *Cholesky-Zerlegung* von A. Beachten Sie, dass hier nicht $l_{ii} = 1$ gefordert wird (im Gegensatz zur LR-Zerlegung).

Beweis: (vollständige Induktion nach n).
Für $n = 1$ mit $A = (a_{11})$ und $a_{11} > 0$ ist $L = L^T = (\sqrt{a_{11}})$.
Sei nun $A \in \mathbb{R}^{n\times n}$ positiv definit. Wir spalten A auf in der Form

$$A \;=\; \begin{pmatrix} A_{n-1} & b \\ b^T & a_{nn} \end{pmatrix} \quad .$$

Dabei ist die $(n-1)\times(n-1)$ - Matrix A_{n-1} (Hauptuntermatrix) ebenfalls symmetrisch und positiv definit. Das Element a_{nn} ist positiv und $b \in \mathbb{R}^{n-1}$. Nach Induktionsvoraussetzung gibt es genau eine reguläre untere Dreicksmatrix L_{n-1} mit $A_{n-1} = L_{n-1}L_{n-1}^T$ und Diagonaleinträgen $l_{ii} > 0$ $(i = 1, \ldots, n-1)$. Damit hat die gesuchte Matrix L die Form

$$L \;=\; \begin{pmatrix} L_{n-1} & 0 \\ c^T & \alpha \end{pmatrix}$$

mit einem $c \in \mathbb{R}^{n-1}$ und $\alpha > 0$. Aus der Forderung $A = LL^T$, also

$$\begin{pmatrix} A_{n-1} & b \\ b^T & a_{nn} \end{pmatrix} = \begin{pmatrix} L_{n-1} & 0 \\ c^T & \alpha \end{pmatrix} \begin{pmatrix} L_{n-1}^T & c \\ 0 & \alpha \end{pmatrix} = \begin{pmatrix} L_{n-1}L_{n-1}^T & L_{n-1}c \\ c^T L_{n-1}^T & c^T c + \alpha^2 \end{pmatrix} \quad ,$$

ergeben sich die Beziehungen $L_{n-1}\, c \;=\; b$ und $c^T c + \alpha^2 = a_{nn}$. Wegen L_{n-1} regulär folgt $c = L_{n-1}^{-1} b$. Weiter gilt

$$0 \;<\; \det A \;=\; \alpha^2 \cdot (\det L_{n-1})^2$$

und somit $0 < \alpha^2 = a_{nn} - c^T c$. Also ist $l_{nn} = \alpha = \sqrt{a_{nn} - c^T c}$ reell und positiv. □

Berechnung der Cholesky-Zerlegung

Durch spaltenweises Vorgehen berechnet man aus (2) für $j = 1, \ldots, n$

$$l_{jj} \;=\; \sqrt{a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2} \quad ,$$

$$l_{ij} \;=\; \frac{1}{l_{jj}} \left(a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk} \right) \quad (i = j+1, \ldots, n).$$

Beispiel: Die symmetrische Matrix

$$A \;=\; \begin{pmatrix} 4 & -2 & 0 \\ -2 & 4 & -2 \\ 0 & -2 & 4 \end{pmatrix}$$

ist positiv definit und liefert folgende Cholesky-Zerlegung:

$$L \;=\; \begin{pmatrix} 2 & 0 & 0 \\ -1 & \sqrt{3} & 0 \\ 0 & -\frac{2\sqrt{3}}{3} & \frac{2\sqrt{6}}{3} \end{pmatrix} \quad .$$

Nebenrechnungen:

$$\begin{aligned}
l_{11} &= \sqrt{a_{11}} \;=\; 2\,, \\
l_{21} &= \frac{1}{l_{11}}\, a_{21} \;=\; -1\,, \\
l_{31} &= \frac{1}{l_{11}}\, a_{31} \;=\; 0\,, \\
l_{22} &= \sqrt{a_{22} - l_{21}^2} \;=\; \sqrt{4-1} \;=\; \sqrt{3}\,, \\
l_{32} &= \frac{1}{l_{22}} \left(a_{32} - l_{31} l_{21} \right) \;=\; \frac{1}{\sqrt{3}}(-2-0) \;=\; -\frac{2\sqrt{3}}{3}\,, \\
l_{33} &= \sqrt{a_{33} - l_{31}^2 - l_{32}^2} \;=\; \sqrt{4 - 0 - \frac{4}{3}} \;=\; \frac{2\sqrt{6}}{3}\,.
\end{aligned}$$

Aufwand
Der Aufwand beträgt ungefähr $\frac{n^3}{6}$ Punktoperationen, denn hier genügt es, L zu berechnen. Damit benötigt man ungefähr die Hälfte des Aufwandes der LR-Zerlegung. Allerdings kommen noch n Wurzelauswertungen hinzu.

1.5 Die QR-Zerlegung

Bei der *QR-Zerlegung* stellt man die $n \times n$ - Matrix A in der Form

$$A \;=\; QR$$

dar. Dabei ist Q eine orthogonale Matrix (d.h. $Q^TQ = I$, somit $Q^{-1} = Q^T$) und R eine obere Dreiecksmatrix.

Aus dem linearen Gleichungssystem $Ax = b$ wird

$$Ax \;= QRx \;=\; b$$

und somit

$$Rx \;=\; Q^Tb \;=:\; u \quad .$$

Wir lösen das gestaffelte System $Rx = u$ (Rückwärtsauflösen).

Satz: Die QR-Zerlegung existiert für jede reelle $n \times n$ - Matrix.

Der Beweis folgt aus dem unten angegebenen Konstruktionsverfahren. Q ergibt sich als Produkt von $(n-1)$ orthogonalen Matrizen, den sogenannten Householder-Matrizen. (Das Produkt zweier orthogonaler Matrizen ist wieder eine orthogonale Matrix.)

Householder-Matrizen
Es bezeichne I die $n \times n$ - Einheitsmatrix und $h^T = (0, \ldots, 0, \xi_k, \ldots, \xi_n) \in \mathbb{R}^n$ einen Vektor der Länge 1, also

$$1 \;=\; \|h\|_2 \;=\; \|h\|_2^2 \;=\; h^Th \quad .$$

Ferner ist hh^T eine $n \times n$ - Matrix:

$$hh^T \;=\; \begin{pmatrix} 0 & & & & \\ & \ddots & & 0 & \\ & & 0 & & \\ & & \xi_k^2 & \cdots & \xi_k\xi_n \\ & 0 & \vdots & & \vdots \\ & & \xi_k\xi_n & \cdots & \xi_n^2 \end{pmatrix} \quad .$$

Definition: Eine Matrix der Form $H := I - 2hh^T$ heißt *Householder-Matrix*.

H hat die Gestalt

$$H = \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & & 1-2\xi_k^2 & -2\xi_k\xi_{k+1} & \cdots & -2\xi_k\xi_n \\ & & & -2\xi_k\xi_{k+1} & 1-2\xi_{k+1}^2 & \ddots & \vdots \\ & & & \vdots & \ddots & \ddots & -2\xi_{n-1}\xi_n \\ & & & -2\xi_k\xi_n & \cdots & -2\xi_{n-1}\xi_n & 1-2\xi_n^2 \end{pmatrix} .$$

Man erkennt, dass H symmetrisch ist.

Lemma: H ist eine orthogonale Matrix.

Beweis: Unter Beachtung von $h^T h = 1$ ergibt sich

$$H^T H = HH = (I - 2hh^T)(I - 2hh^T) = I - 4hh^T + 4hh^T hh^T = I \quad . \quad \square$$

Konstruktion der QR-Zerlegung

Gegeben sei die Matrix

$$A^{(0)} = A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix} .$$

Es bezeichne $a_1 = (a_{11}, \ldots, a_{n1})^T$ die erste Spalte von A und $e_1 := (1, 0, \ldots, 0)^T \in \mathbb{R}^n$ den ersten Einheitsvektor. Konstruiere nun eine Householder-Matrix H, so dass

$$Ha_1 = \sigma e_1 \quad \text{mit } \sigma \in \mathbb{R}$$

gilt. Wegen

$$|\sigma|^2 = \|\sigma e_1\|_2^2 = \|Ha_1\|_2^2 = (Ha_1)^T(Ha_1) = a_1^T H^T H a_1 = a_1^T a_1 = \|a_1\|_2^2$$

ist σ bis auf das Vorzeichen festgelegt. Es gilt

$$(hh^T)a_1 = h(h^T a_1) = (h^T a_1)h$$

(da $h^T a_1 \in \mathbb{R}$) und somit

$$Ha_1 = a_1 - 2(hh^T)a_1 = a_1 - 2(h^T a_1)h = \sigma e_1 \quad ,$$

folglich

$$2(h^T a_1)h = a_1 - \sigma e_1 \quad .$$

Dies bedeutet, dass h ein Vielfaches von $a_1 - \sigma e_1$ ist. Wegen $\|h\|_2 = 1$ muss

$$h = \frac{a_1 - \sigma e_1}{\|a_1 - \sigma e_1\|_2} \tag{3}$$

gelten. Dabei ist noch das Vorzeichen von σ offen. Aus Stabilitätsgründen wählen wir dieses als $-\text{sign}(a_{11})$ (wobei noch sign $(0) = 1$ festgelegt wird), also

$$\sigma \;=\; -\text{sign}(a_{11})\,\|a_1\|_2 \quad .$$

In (3) ergibt sich dann

$$h \;=\; \frac{a_1 + \text{sign}(a_{11})\,\|a_1\|_2 e_1}{\left\| a_1 + \text{sign}(a_{11})\|a_1\|_2 e_1 \right\|} \quad .$$

Mit diesem Vektor $h =: h_1$ bildet man die Householder-Matrix

$$H_1 \;:=\; H \;:=\; I - 2h_1 h_1^T$$

und erhält dann

$$A^{(1)} \;:=\; H_1 A^{(0)} \;=\; H_1 A \;=\; \begin{pmatrix} * & * & \cdots & * \\ 0 & & & \\ \vdots & & \tilde{A}_1 & \\ 0 & & & \end{pmatrix} \quad .$$

Die Restmatrix $\tilde{A}_1$ hat die Dimension $(n-1) \times (n-1)$. Zunächst suchen wir eine $(n-1) \times (n-1)$ - Householder-Matrix $\tilde{H}_2$ mit

$$\tilde{H}_2 \tilde{a}_1 \;=\; \tilde{\sigma} \tilde{e}_1 \quad .$$

Dabei ist $\tilde{a}_1$ die 1. Spalte von $\tilde{A}_1$ und $\tilde{e}_1$ der 1. Einheitsvektor des $\mathbb{R}^{n-1}$. $\tilde{H}_2$ wird analog zu H_1 konstruiert, die Dimension ist jedoch um 1 niedriger.

Dann ist auch die $n \times n$ - Matrix

$$H_2 \;=\; \begin{pmatrix} 1 & 0 \\ 0 & \tilde{H}_2 \end{pmatrix}$$

eine Householer-Matrix, und es gilt

$$A^{(2)} \;:=\; H_2 A^{(1)} \;=\; H_2 H_1 A \;=\; \begin{pmatrix} * & * & \cdots & \cdots & * \\ 0 & * & \cdots & \cdots & * \\ \vdots & 0 & & & \\ \vdots & \vdots & & \tilde{A}_2 & \\ 0 & 0 & & & \end{pmatrix} \quad .$$

Im nächsten Schritt wird die $(n-2) \times (n-2)$ - Matrix $\tilde{A}_2$ betrachtet usw. Insgesamt erhält man nach $n-1$ Schritten

$$R \;:=\; A^{(n-1)} \;=\; H_{n-1} H_{n-2} \cdots H_2 H_1 A \quad ,$$

wobei R eine obere Dreiecksmatrix ist. Mit der orthogonalen Matrix

$$Q \;:=\; (H_{n-1} H_{n-2} \cdots H_2 H_1)^{-1} \;=\; H_1 H_2 \cdots H_{n-2} H_{n-1}$$

folgt schließlich $A \;=\; QR\,.$

Aufwand

Zur Berechnung der QR-Zerlegung mittels Householder-Matrizen benötigt man ungefähr $\frac{2}{3}n^3$ Punktoperationen. Dafür bieten orthogonale Matrizen den Vorteil einer größeren Stabilität. Zudem existiert die QR-Zerlegung immer (selbst für nicht quadratische Matrizen).

Die QR-Zerlegung wird uns später bei beim linearen Ausgleich und der Eigenwert-Bestimmung wieder begegnen.

1.6 Die Gauß-Elimination

Das Verfahren

Zu lösen sei das lineare Gleichungssystem $Ax = b$ mit

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix} \quad \text{und} \quad b = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} .$$

Bringe dieses System durch elementare Zeilenumformungen an der erweiterten Matrix

$$(A,b) = \left(\begin{array}{ccc|c} a_{11} & \cdots & a_{1n} & b_1 \\ \vdots & & \vdots & \vdots \\ a_{n1} & \cdots & a_{nn} & b_n \end{array}\right) \tag{4}$$

auf Zeilenstufenform

$$(R,c) = \left(\begin{array}{cccc|c} r_{11} & \cdots & \cdots & r_{1n} & c_1 \\ 0 & r_{22} & \cdots & r_{2n} & c_2 \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & r_{nn} & c_n \end{array}\right) . \tag{5}$$

Dieses Gleichungssystem $Rx = c$ lässt sich leicht lösen (Rückwärtsauflösung).

Dabei sind als elementare Zeilenumformungen in (4) zugelassen:

- Vertauschung von Zeilen,
- Addition eines Vielfachen einer Zeile zu einer anderen,
- Multiplikation einer Zeile mit einer von Null verschiedenen Zahl.

Diese Operationen verändern die Lösungsmenge des Gleichungssystems $Ax = b$ nicht.

Im Gauß-Algorithmus werden nun spaltenweise die Elemente unterhalb der Diagonalen zu Null gemacht:

Es sei $a_{11} \neq 0$. Mit den Operationen

$$\begin{aligned} l_{i1} &:= \frac{a_{i1}}{a_{11}} \qquad (i = 2,3,\ldots,n)\,, \\ a_{ik}^{(1)} &:= a_{ik} - l_{i1}a_{1k} \qquad (i,k = 2,3,\ldots,n)\,, \\ b_i^{(1)} &:= b_i - l_{i1}b_1 \qquad (i = 2,3,\ldots,n) \end{aligned} \tag{6}$$

wird das System (4) übergeführt in das äquivalente System

$$\left(\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ 0 & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} & b_2^{(1)} \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & a_{n2}^{(1)} & \cdots & a_{nn}^{(1)} & b_n^{(1)} \end{array}\right) \quad .$$

Damit lässt sich x_1 durch die übrigen Unbekannten ausdrücken:

$$x_1 \;=\; \frac{1}{a_{11}} \left(b_1 - \sum_{k=2}^{n} a_{1k} x_k \right) \quad ,$$

und es bleibt ein reduziertes Gleichungssystem mit $n-1$ Gleichungen für $n-1$ Unbekannte.

Dieser Vorgang wird als ein *Eliminationsschritt* bezeichnet. Nun wiederholt man diesen Prozess für das reduzierte Gleichungssystem und gelangt so nach $n-1$ Eliminationsschritten auf ein Gleichungssystem der Form (5).

Das gesamte Verfahren wird als *Gauß-Elimination* bezeichnet. Die Größen $a_{11},\ a_{22}^{(1)},\ a_{33}^{(2)}, \ldots$ heißen *Pivotelemente*.

Beispiel: Für

$$A \;:=\; \begin{pmatrix} 2 & 3 & -5 \\ 4 & 8 & -3 \\ -6 & 1 & 4 \end{pmatrix} , \qquad b \;:=\; \begin{pmatrix} 10 \\ 19 \\ 11 \end{pmatrix}$$

ergibt sich

$$\left(\begin{array}{ccc|c} 2 & 3 & -5 & 10 \\ 4 & 8 & -3 & 19 \\ -6 & 1 & 4 & 11 \end{array}\right) \quad \rightarrow \quad \left(\begin{array}{ccc|c} 2 & 3 & -5 & 10 \\ 0 & 2 & 7 & -1 \\ 0 & 10 & -11 & 41 \end{array}\right) \quad \rightarrow \quad \left(\begin{array}{ccc|c} 2 & 3 & -5 & 10 \\ 0 & 2 & 7 & -1 \\ 0 & 0 & -46 & 46 \end{array}\right) \quad .$$

Rückwärtsauflösen liefert

$$\begin{aligned} x_3 &= -1 \quad , \\ x_2 &= \frac{1}{2}(-1+7) \;=\; 3 \quad , \\ x_1 &= \frac{1}{2}(10-9-5) \;=\; -2 \quad . \end{aligned}$$

Wir müssen uns nun als nächstes fragen, wann die Gauß-Elimination durchführbar ist. Da in (6) durch die Pivotelemente dividiert wird, ist dies äquivalent zur Frage, wann die Pivotelemente von Null verschieden sind.

Satz: Die Matrix A sei regulär. Dann existiert vor jedem Eliminationsschritt eine Zeilenpermutation derart, dass das Diagonalelement (Pivotelement) von Null verschieden ist.

Beweis: (vollständige Induktion)
A regulär impliziert $\det A \neq 0$. Dann existiert in der 1. Spalte von A ein $a_{i1} \neq 0$ (denn sonst würde sofort $\det A = 0$ folgen). Vertausche dann die 1. und i. Zeile.
Sei nun die Behauptung schon richtig für die ersten k Spalten, d.h.

$$\left(\begin{array}{ccccccc|c} a_{11} & a_{12} & & & & & a_{1n} & b_1 \\ 0 & a_{22}^{(1)} & & & & & a_{2n}^{(1)} & b_2^{(1)} \\ & \ddots & \ddots & & & & \vdots & \vdots \\ \vdots & & \ddots & a_{kk}^{(k-1)} & \cdots & \cdots & a_{kn}^{(k-1)} & b_k^{(k-1)} \\ & & & 0 & & & & b_{k+1}^{(k)} \\ \vdots & & & \vdots & & \tilde{A}_k & & \vdots \\ & & & & & & & \\ 0 & \cdots & \cdots & 0 & & & & b_n^{(k)} \end{array}\right)$$

mit

$$\tilde{A}_k = \begin{pmatrix} a_{k+1,k+1}^{(k)} & \cdots & a_{k+1,n}^{(k)} \\ \vdots & & \vdots \\ a_{n,k+1}^{(k)} & \cdots & a_{nn}^{(k)} \end{pmatrix} .$$

Die verwendeten Zeilenoperationen verändern den Betrag der Determinante nicht. Deshalb gilt

$$|\det A| = \left| a_{11} a_{22}^{(1)} \cdots a_{kk}^{(k-1)} \det \tilde{A}_k \right| .$$

Somit folgt aus $\det A \neq 0$ die Beziehung $\det \tilde{A}_k \neq 0$. Dann gibt es ein $i \in \{k+1, \ldots, n\}$ mit $a_{i,k+1}^{(k)} \neq 0$ (denn sonst wäre $\det \tilde{A}_k = 0$). Die Vertauschung der $(k+1)$-ten mit der i-ten Zeile liefert ein von Null verschiedenes Pivot-Element. □

Definition: Eine Matrix A heißt *diagonal dominant*, falls gilt

$$|a_{ii}| > \sum_{\substack{k=1 \\ k \neq i}}^{n} |a_{ik}| \qquad (i = 1, 2, \ldots, n) .$$

Satz: Ist die Matrix A diagonal dominant, so ist Gauß-Elimination ohne Zeilenvertauschungen durchführbar.

Beweis: Übung

Aufwand

Löst man das lineare Gleichungssystem $Ax = b$ mittels Gauß-Elimination, so beträgt der Aufwand (ohne Berücksichtigung von Zeilenvertauschungen)

$$\begin{array}{ll} \frac{1}{3}n^3 + n^2 - \frac{1}{3}n & \text{Mult./Div.} \\ \frac{1}{3}n^3 + \frac{1}{2}n^2 - \frac{5}{6}n & \text{Add./Subtr.} \end{array}$$

Größenordnungsmäßig sind also $\mathcal{O}(n^3)$ Operationen erforderlich.

Pivotstrategien

Der obige Satz garantiert bei regulärem A für jeden Eliminationsschritt die Existenz eines von Null verschiedenen Pivotelementes. Er lässt jedoch die konkrete Wahl offen, falls es mehrere gibt.

Die Pivotwahl ist von großer Bedeutung für die Genauigkeit der berechneten Lösung (Rundungsfehler). Zudem benötigt ein Computerprogramm eine genau definierte Regel zur Bestimmung der Pivotelemente, die man als *Pivotstrategie* bezeichnet.

Beispiel: (vgl. Schwarz [2])
Gegeben sei das lineare Gleichungssystem

$$\left(\begin{array}{cc|c} 0.00035 & 1.2654 & 3.5267 \\ 1.2547 & 1.3182 & 6.8541 \end{array}\right) \quad .$$

Es werde nach jedem Schritt auf 5 Stellen gerundet (d.h. es wird ein Rechner mit einer 5-stelligen Mantissenlänge simuliert). Wird hier a_{11} als Pivotelement gewählt, so liefert die Gauß-Elimination

$$\left(\begin{array}{cc|c} 0.00035 & 1.2654 & 3.5267 \\ 0 & -4535.0 & -12636 \end{array}\right)$$

Daraus ergibt sich

$$x_2 \;=\; 2.7863$$

und

$$\begin{aligned} x_1 &= -(-3.5267 + 1.2654 * 2.7863)/0.00035 \\ &= -(-3.5267 + 3.5258)/0.00035 \;=\; 0.00009/0.00035 \;=\; 2.5714 \end{aligned} \qquad (7)$$

Das exakte Resultat ist jedoch $x_1 = 2.5354$.
(Subtraktion zweier ungefähr gleich großer Zahlen, anschließend wird durch eine sehr kleine Zahl dividiert, was den Fehler noch verstärkt).

Eine mögliche Pivotstrategie besteht deshalb in der Wahl des betragsmäßig größten Elements in der entsprechenden Restspalte:

$$\max_{i \geq k} \left| a_{ik}^{(k-1)} \right| \;=:\; \left| a_{pk}^{(k-1)} \right| \quad .$$

Vor dem k-ten Gauß-Eliminationsschritt werden die Zeilen k und p vertauscht.

Diese Strategie nennt man *betragsmaximale Spaltenpivotwahl*.

Wendet man die betragsmaximale Spaltenpivotwahl im obigen Beispiel an, so müssen zunächst die beiden Zeilen vertauscht werden:

$$\left(\begin{array}{cc|c} 1.2547 & 1.3182 & 6.8541 \\ 0.00035 & 1.2654 & 3.5267 \end{array}\right) \quad \longrightarrow \quad \left(\begin{array}{cc|c} 1.2547 & 1.3182 & 6.8541 \\ 0 & 1.2650 & 3.5248 \end{array}\right) \quad .$$

Daraus ergibt sich $x_1 = 2.5353$. Dieses Ergebnis ist wesentlich genauer als (7).

1.7 Gauß-Jordan-Elimination

Es sei $A \in \mathbb{R}^{n\times m}$ eine Matrix mit den Koeffizienten

$$A = \begin{pmatrix} a_{11} & \dots & a_{1k} & \dots & a_{1m} \\ \vdots & & \vdots & & \vdots \\ a_{i1} & \dots & a_{ik} & \dots & a_{im} \\ \vdots & & \vdots & & \vdots \\ a_{n1} & \dots & a_{nk} & \dots & a_{nm} \end{pmatrix}$$

und $a_{ik} \neq 0$. Durch elementare Zeilenoperationen kann man diese Matrix auf eine Form bringen, bei der in der k-ten Spalte der Einheitsvektor e_i steht. Dies wird als ein *Gauß-Jordan-Eliminations-Schritt* bezeichnet.

Als **Beispiel** betrachten wir

$$\begin{pmatrix} 4 & 1 & 1 & 2 & 0 & 3 \\ 1 & 2 & \boxed{4} & 1 & 1 & 1 \\ -1 & -1 & -3 & 1 & -1 & 2 \end{pmatrix}$$

mit dem Pivot $a_{23} = 4$. Zunächst wird die 2. Zeile durch das Pivot-Element dividiert:

$$\begin{pmatrix} 4 & 1 & 1 & 2 & 0 & 3 \\ \frac{1}{4} & \frac{1}{2} & 1 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ -1 & -1 & -3 & 1 & -1 & 2 \end{pmatrix}.$$

Dann macht man in der 3. Spalte die anderen Elemente zu Null (durch Zeilenkombinationen):

$$\begin{pmatrix} \frac{15}{4} & \frac{1}{2} & 0 & \frac{7}{4} & -\frac{1}{4} & \frac{11}{4} \\ \frac{1}{4} & \frac{1}{2} & 1 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ -\frac{1}{4} & \frac{1}{2} & 0 & \frac{7}{4} & -\frac{1}{4} & \frac{11}{4} \end{pmatrix} .$$

Anmerkung: Die Gauß-Jordan-Elimination wird uns bei der *linearen Optimierung* wieder begegnen.

1.8 Matrix-Zerlegungen mit Matlab

Matlab kennt die in diesem Paragraphen beschriebenen Matrix-Zerlegungen ebenfalls und stellt dafür Matlab-Funktionen bereit.

Das Kommando

```
[L,R,P] = lu(A)
```

liefert die LR-Zerlegung einer Matrix: $PA = LR$. Möglich ist auch

```
[S,R] = lu(A)
```

Hier gilt $A = SR$. Dabei ist jedoch `S` keine untere Dreiecksmatrix, sondern eine Zeilenpermutation einer unteren Dreiecksmatrix: Aus $PA = LR$ folgt $A = P^{-1}LR = P^TLR$

(beachte: Permutationsmatrizen sind orthogonale Matrizen). Damit erhält man den Zusammenhang

$$S = P^T L \text{ oder } L = PS \quad .$$

Das folgende **Beispiel** demonstriert diese Unterschiede:

```
A =
     0     1     2
     2     2     2
     4     2     1

[L,R,P] = lu(A)

L =
    1.0000         0         0
    0.5000    1.0000         0
         0    1.0000    1.0000

R =
    4.0000    2.0000    1.0000
         0    1.0000    1.5000
         0         0    0.5000

P =
     0     0     1
     0     1     0
     1     0     0

[S,R] = lu(A)

S =
         0    1.0000    1.0000
    0.5000    1.0000         0
    1.0000         0         0

R =
    4.0000    2.0000    1.0000
         0    1.0000    1.5000
         0         0    0.5000
```

Der Matlab-Befehl

```
R = chol(A)
```

liefert die Cholesky-Zerlegung von A. Dabei ist zu beachten, dass `R` eine obere Dreiecksmatrix ist (in unserer Notation also $R = L^T$).

Bei der Berechnung der Zerlegung wird von A nur die obere Dreiecksmatrix verwendet (Einträge unterhalb der Diagonalen werden ignoriert).

Die QR-Zerlegung von A erhalten wir durch das Kommando

```
[Q,R] = qr(A)
```

.

Für die Lösung des linearen Gleichungssystems $Ax = b$ kennt Matlab den Befehl

```
x = A \ b
```

.

Die Lösung wird dabei mittels Gauß-Elimination berechnet.

2 Linearer Ausgleich

2.1 Allgemeine Ausgleichsprobleme

In manchen Wissenschaftsgebieten (z.B. Chemie, Biologie, Physik) kommt das Problem vor, unbekannte Parameter $\alpha = (\alpha_1, \ldots, \alpha_n)$ einer Funktion $G(t;\alpha)$ anhand von m Messergebnissen zu bestimmen. Dabei ist die Struktur von $G(t;\alpha)$ aufgrund von Naturgesetzen oder Modellannahmen gegeben. In der Regel ist $m > n$.

Seien s_i $(i = 1, \ldots, m) \in \mathbb{R}$ die Messergebnisse zu den Zeitpunkten t_i. Wegen $m > n$ besitzt im Allgemeinen

$$G(t_i;\alpha) \;=\; s_i \qquad (i = 1, \ldots, m)$$

keine Lösung (überbestimmtes Gleichungssystem).

Deshalb geht man dazu über, den Parameter-Vektor α so zu bestimmen, dass

$$G(t_i;\alpha) \;\approx\; s_i \qquad (i = 1, \ldots, m)$$

möglichst gut erfüllt ist. Dies wird präzisiert zu

$$\sum_{i=1}^{m} (G(t_i;\alpha) - s_i)^2 \;=\; \min \tag{8}$$

(d.h. die Fehlerquadratsumme ist zu minimieren).

Diese Aufgabe wird als *allgemeines Ausgleichsproblem* bezeichnet.

Dazu betrachten wir einige **Beispiele** aus den Naturwissenschaften:

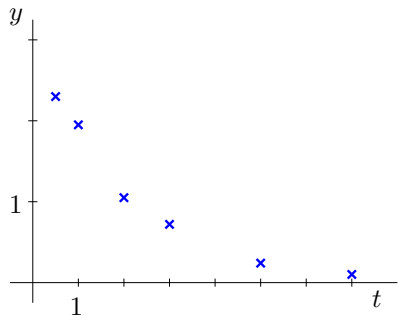
1. **Radioaktiver Zerfall (Chemie)**

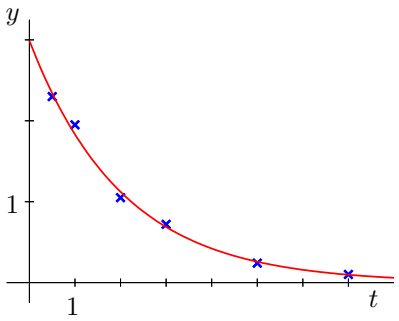
   Es bezeichne $y(t)$ die Konzentration einer radioaktiven Substanz zum Zeitpunkt t. Der radioaktive Zerfall wird beschrieben durch die Funktion

   $$y(t) \;=\; c\exp(\lambda t) \;=\; G(t;c,\lambda) \quad (c > 0,\ \lambda < 0).$$

   Dabei ist c die Konzentration der Substanz zum Zeitpunkt 0 und λ eine negative Material-Konstante. Eine Messreihe ist im linken Schaubild dargestellt.

   Unsere Aufgabe lautet nun: Bestimme anhand dieser Messwerte die Parameter c und λ so, dass die Theoriefunktion möglichst gut durch die Messpunkte geht (rechtes Schaubild).





2. **Logistisches Wachstum (Biologie)**
   Das Wachstum von vielen Populationen lässt sich durch die *logistische Kurve*

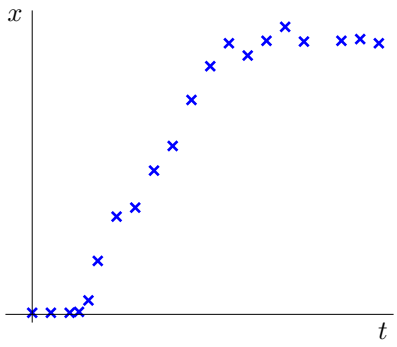
$$L(t) \;=\; \frac{a}{1+\exp(b-ct)} \;=\; G(t;a,b,c)$$

   mit Parametern $a>0,\ c>0$ und $b\in\mathbb{R}$ beschreiben. Die folgende Tabelle enthält die Ergebnisse eines Wachstumsexperimentes mit Käfern:

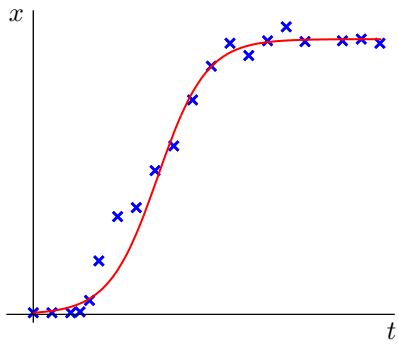
Tage t	0	14	28	35	42	49	63	77	91	105	119	133	$\cdots$	245	259
Käfer x	2	2	2	3	17	65	119	130	175	205	261	302	$\cdots$	335	330

   Auch hier sind die Parameter so zu bestimmen, dass die logistische Kurve möglichst gut durch die Messergebnisse geht.

Messergebnisse



optimale Kurve $L(t)$

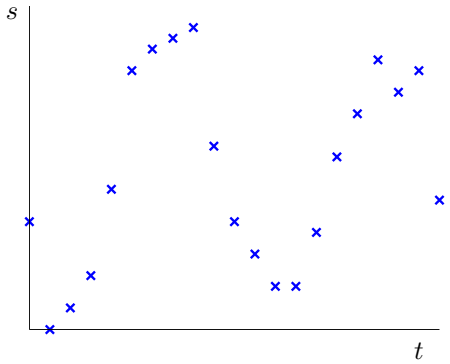


3. Aufgrund von physikalischen Überlegungen erwartet man zwischen Messgrößen s und t einen Zusammenhang der Form

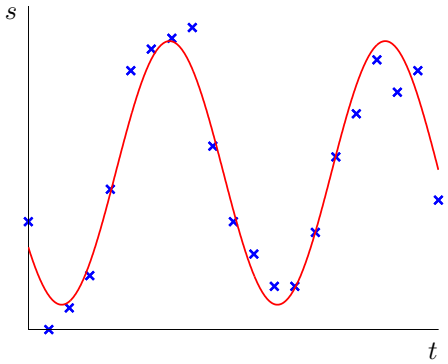
$$s(t) \;=\; \alpha_1\cdot\sin(\alpha_2 t+\alpha_3)+\alpha_4 \;=\; G(t;\alpha_1,\alpha_2,\alpha_3,\alpha_4) \quad .$$

   In den folgenden Schaubildern wird eine Messreihe und die optimale Kurve dargestellt.

Messergebnisse



optimale Kurve $s(t)$



Die numerische Lösung des allgemeinen Ausgleichsproblems ist kompliziert. Wesentlich einfacher ist die Situation, bei der die Parameter linear in die Theoriefunktion eingehen. Deshalb betrachten wir zunächst diesen Fall. Der allgemeine Fall wird in Paragraph 10 (Nichtlineare Minimierung) behandelt.

2.2 Lineare Ausgleichsprobleme

Definition: Ein Ausgleichsproblem heißt *linear*, falls die Theoriefunktion G linear von den Parametern $\alpha = (\alpha_1, \ldots, \alpha_n)$ abhängt, also die Form

$$G(t, \alpha) \;=\; \sum_{j=1}^{n} \alpha_j g_j(t)$$

hat mit gewissen Grundfunktionen $g_j(t),\ j = 1, \ldots, n$. Die zu minimierende Fehlerquadratsumme lautet dann

$$\sum_{i=1}^{m} \left(\sum_{j=1}^{n} \alpha_j g_j(t_i) \;-\; s_i \right)^2 \quad . \tag{9}$$

Beispiele:

1. $g_j(t) := t^{j-1}$ $(j = 1, \ldots, n)$ (Polynomausgleich):

$$G(t, a_0, \ldots, a_{n-1}) \;=\; \sum_{k=0}^{n-1} a_k t^k \quad .$$

   Im Sonderfall $n = 2$ erhält man als Theoriefunktion eine Gerade; die optimale Lösung wird als *Regressionsgerade* bezeichnet.

2. $g_j(t) := \cos(j\,t)$ bzw. $g_j(t) := \sin(j\,t)$. Als Theoriefunktion werden dann die *trigonometrischen Polyome* verwendet:

$$G(t, a_0, \ldots, a_k, b_1, \ldots, b_k) \;=\; a_0 \;+\; \sum_{j=1}^{k} \left(a_j \cos(jt) + b_j \sin(jt) \right) \quad .$$

Sei nun A die $m \times n$ - Matrix mit den Komponenten

$$a_{ij} \;:=\; g_j(t_i) \qquad (i = 1, \ldots, m;\ j = 1, \ldots, n)$$

und $s = (s_1, \ldots, s_m)^T$. Damit führt der lineare Ausgleich (9) auf das Problem

$$\|A\alpha - s\|_2^2 \;= \min \quad .$$

Beim Polynomausgleich hat A die Gestalt

$$A \;=\; \begin{pmatrix} 1 & t_1 & t_1^2 & \cdots & t_1^{n-1} \\ 1 & t_2 & t_2^2 & \cdots & t_2^{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & t_m & t_m^2 & \cdots & t_m^{n-1} \end{pmatrix} \quad .$$

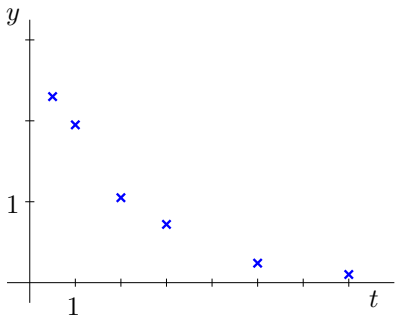
Sind hier die t_i paarweise verschieden, so hat A den Rang n, d.h. die Spalten sind linear unabhängig.

Anmerkung Gelegentlich gelingt es, ein nichtlineares Ausgleichsproblem in ein lineares zu transformierten. So erhalten wir in Beispiel 1 aus Abschnitt 2.1 durch den Übergang zu einem halblogarithmischen Maßstab ein lineares Ausgleichsproblem:

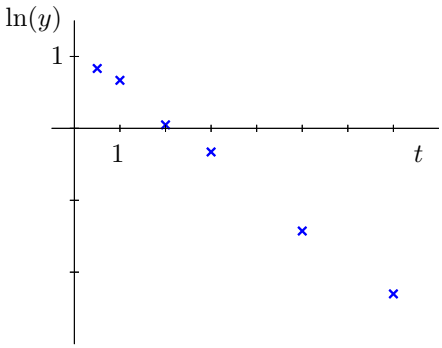
$$\ln(y(t)) \; = \ln(c\,\exp(\lambda t)) \; = \; \ln(c) \; + \; \lambda t \; = \; a \; + \; \lambda t \; = \; G(t, a, \lambda)$$

mit $a = \ln(c)$. Im rechten Schaubild erkannt man, dass die transformierten Messergebnisse ungefähr auf einer Geraden liegen.

normales Koordinatensystem



halblog. Koordinatensystem



2.3 Die Normalgleichungen

Gegeben sei das lineare Ausgleichsproblem

$$\|Ax - b\|_2^2 \; = \; \min$$

mit $A \in \mathbb{R}^{m\times n},\; x \in \mathbb{R}^n,\; b \in \mathbb{R}^m$ $(m > n)$. Man erhält dann

$$\begin{aligned} F(x) \; := \; \|Ax - b\|_2^2 \; &= \; (Ax - b)^T(Ax - b) \\ &= \; (x^T A^T - b^T)(Ax - b) \\ &= \; x^T A^T Ax - b^T Ax - x^T A^T b + b^T b \\ &= \; x^T A^T Ax - 2(A^T b)^T x + b^T b \end{aligned} \qquad (10)$$

(beachte $x^T A^T b = (b^T Ax)^T = b^T Ax$, da reelle Größe). Gesucht ist somit ein Minimum der Funktion

$$F \; : \; \mathbb{R}^n \; \to \; \mathbb{R} \quad .$$

F ist beliebig oft differenzierbar, denn F ist ein multivariates Polynom.

Notwendige Bedingung für das Vorliegen eines (lokalen) Minimums im Punkte x ist die Gradientenbedingung

$$\nabla F(x) \; = \; 0 \quad .$$

Aus (10) folgt

$$\nabla F(x) \; = \; 2\,A^T Ax - 2\,A^T b \; = \; 0$$

oder umgeformt

$$A^T A x \;=\; A^T b \quad . \tag{11}$$

Definition: Diese Beziehung nennt man die *Normalgleichungen.*

Anmerkung: $A^T A$ ist eine $n \times n$ – Matrix, $A^T b$ ist ein Vektor der Länge n. Die Normalgleichungen stellen ein lineares Gleichungssystem dar.

Satz: Jede Lösung der Normalgleichungen (11) ist ein globales Minimum der Funktion $F(x) = \|Ax - b\|_2^2$.

Beweis:
Sei $\overline{x}$ eine Lösung der Normalgleichungen, also $A^T A\overline{x} \;=\; A^T b$. Dann gilt für jedes $x \in \mathbb{R}^n$

$$\begin{aligned}
F(x) &= \|Ax - b\|_2^2 \;= < Ax - b, Ax - b > \\
&= < A\overline{x} - b + A(x - \overline{x}), A\overline{x} - b + A(x - \overline{x}) > \\
&= < A\overline{x} - b, A\overline{x} - b > \; + 2 < A\overline{x} - b, A(x - \overline{x}) > \; + \; < A(x - \overline{x}), A(x - \overline{x}) > \\
&= F(\overline{x}) \; + \; 2 < A\overline{x} - b, A(x - \overline{x}) > \; + \|A(x - \overline{x})\|_2^2 \\
&= F(\overline{x}) \; + \; 2 < A^T A\overline{x} - A^T b, x - \overline{x} > \; + \|A(x - \overline{x})\|_2^2 \\
&= F(\overline{x}) \; + \; 2 < 0, x - \overline{x} > \; + \|A(x - \overline{x})\|_2^2 \\
&= F(\overline{x}) \; + \; \|A(x - \overline{x})\|_2^2 \\
&\geq F(\overline{x}) \quad .
\end{aligned}$$

Somit ist $\overline{x}$ ein globales Minimum. □

Korollar: A habe den Rang n. Dann besitzt die Funktion F genau ein Minimum, nämlich die Lösung der Normalgleichungen.

Beweis: Die Matrix $A^T A$ ist positiv definit; denn

$$x^T A^T A x \;=\; (Ax)^T (Ax) \;=\; \|Ax\|_2^2 \;\geq 0$$

und

$$0 \;=\; x^T A^T A x \;=\; (Ax)^T (Ax) \quad \Leftrightarrow \quad Ax \;=\; 0 \quad \Leftrightarrow \quad x \;=\; 0$$

(da die Spalten von A linear unabhängig sind).
Eine positiv definite Matrix ist regulär. Somit besitzen die Normalgleichungen genau eine Lösung. Diese ist nach obigem Satz das globale Minimum. □

2.4 Berechnung einer Lösung

Zur Bestimmung der Lösung der Normalgleichungen

$$A^T A x \;=\; A^T b$$

(mit $\text{rang}(A) = n$) kann man die bekannten Verfahren (Gauß-Elimination, Cholesky-Zerlegung) verwenden. Dazu muss jedoch zunächst die Matrix $A^T A$ berechnet werden. Bei geschickter Anwendung der QR-Zerlegung lässt sich dies vermeiden.

Es existiert eine orthogonale Matrix $Q \in \mathbb{R}^{m\times m}$ und eine invertierbare obere Dreiecksmatrix $R \in \mathbb{R}^{n\times n}$ mit

$$QA \;=\; \begin{pmatrix} R \\ 0 \end{pmatrix} \quad \text{bzw.} \quad A = Q^T \begin{pmatrix} R \\ 0 \end{pmatrix} \quad .$$

Dabei ist Q das Produkt von n Householder-Matrizen.

Es folgt dann

$$A^T A \;=\; A^T Q^T Q A \;=\; (QA)^T QA \;=\; R^T R$$

und

$$A^T \;=\; (R^T, 0)Q \quad .$$

Somit erhält man für die Normalgleichungen

$$R^T Rx \;=\; (R^T, 0)Qb \;=\; R^T \begin{pmatrix} (Qb)_1 \\ \vdots \\ (Qb)_n \end{pmatrix} \quad .$$

Da R^T invertierbar ist, ergibt sich schließlich das gestaffelte Gleichungssystem

$$Rx \;=\; \begin{pmatrix} (Qb)_1 \\ \vdots \\ (Qb)_n \end{pmatrix} \quad .$$

Dieses System ist eindeutig lösbar (Rückwärtsauflösen).

3 Lineare Optimierung

3.1 Problemstellung

In der linearen Optimierung werden lineare Zielfunktionen unter linearen Nebenbedingungen maximiert bzw. minimiert. Solche Probleme treten häufig in der Betriebswirtschaft auf. Hier ist dies die Gewinnmaximierung (bzw. Kostenminimierung) unter gegebenen Einschränkungen an die Produktionsfaktoren (Kapital, Arbeit, Lagervorrat, gesetzliche Bestimmungen). Dazu betrachten wir ein

Beispiel: Ein Landwirt bewirtschaftet eine Fläche von 80 ha. Es sollen zwei Sorten von Getreide (Gerste und Weizen) angebaut werden. An Arbeitszeit stehen 560 Stunden zur Verfügung. Für einen ha Gerste sind 6 Std und für einen ha Weizen sind 10 Std Arbeitszeit erforderlich. An Kosten (Saatgut, Düngemittel usw.) entstehen pro ha bei Gerste 30 und bei Weizen 100 Euro. Insgesamt stehen dem Landwirt maximal 4700 Euro zur Verfügung.
An Gewinn sind bei Gerste 40 Euro und bei Weizen 50 Euro pro ha zu erwarten.
Wie hat der Landwirt die Ackerfläche aufzuteilen, um einen möglichst hohen Gewinn zu erzielen?

Bezeichnet man mit x_1 die Ackerfläche für Gerste und mit x_2 die Fläche für Weizen, so ergeben sich folgende Bedingungen:

$$\begin{array}{rcll} x_1 + x_2 & \leq & 80 & \text{(Beschränkung Anbaufläche)} \\ 6x_1 + 10x_2 & \leq & 560 & \text{(Beschränkung Arbeitszeit)} \\ 30x_1 + 100x_2 & \leq & 4700 & \text{(Beschränkung Kapital)} \\ x_1 & \geq & 0 & \\ x_2 & \geq & 0 & \end{array}$$

Die Aufgabe lautet nun:

$$\text{Maximiere} \quad g \;=\; g(x_1, x_2) \;=\; 40x_1 + 50x_2$$

unter den obigen Einschränkungen (*Nebenbedingungen*).

Wir gehen zur Matrixschreibweise über. Mit

$$x \;=\; \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}\;,\; P \;=\; \begin{pmatrix} 1 & 1 \\ 6 & 10 \\ 30 & 100 \end{pmatrix}\;,\; b \;=\; \begin{pmatrix} 80 \\ 560 \\ 4700 \end{pmatrix}\;,\; q \;=\; \begin{pmatrix} 40 \\ 50 \end{pmatrix}$$

lautet die Aufgabe

$$\begin{array}{rcl} g(x) & = & q^T x \to \max \\ Px & \leq & b \\ x & \geq & 0 \end{array}$$

(dabei sind sämtliche Ungleichungen komponentenweise zu verstehen).

Allgemeiner Fall:
Gegeben seien $q \in \mathbb{R}^n,\ b \in \mathbb{R}^m,\ P \in \mathbb{R}^{m\times n}$. Gesucht ist ein $x \in \mathbb{R}^n$ mit

$$\begin{array}{rcl} q^T x & \to & \max \\ Px & \leq & b \\ x & \geq & 0 \end{array} \quad . \tag{12}$$

Definitionen:

1. Diese Aufgabenstellung wird als *lineares Optimierungsproblem* (bzw. als *lineares Programm*) bezeichnet. Die Funktion $q^T x$ heißt *Zielfunktion*. Wir bezeichnen die Darstellung in (12) als Normalform 1.

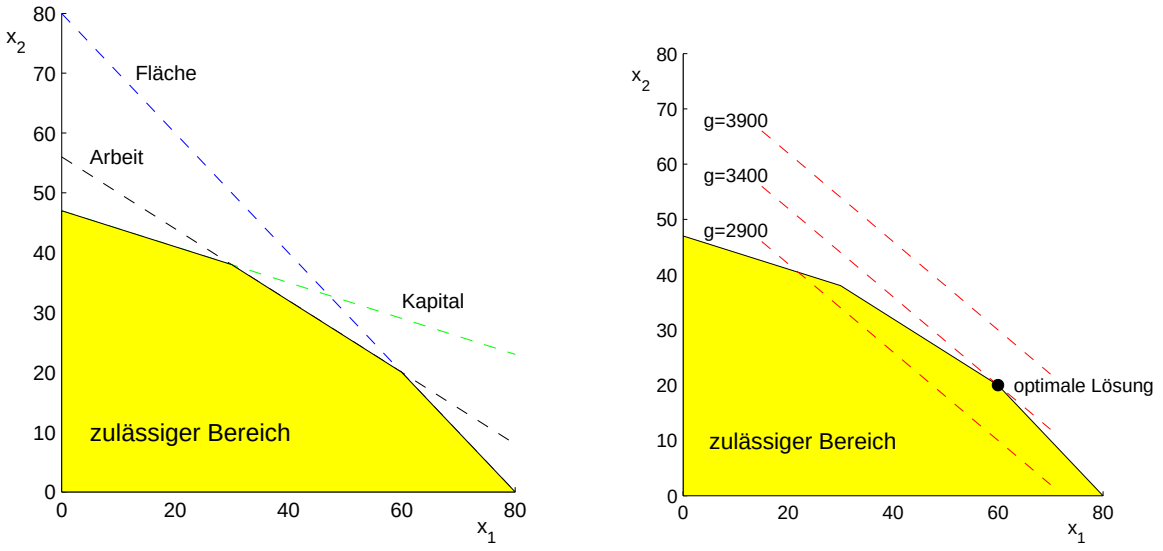
2. Die Menge
$$K := \{x \in \mathbb{R}^n : x \geq 0 \text{ und } Px \leq b\}$$
heißt *zulässige Menge (zulässiger Bereich)*, ihre Elemente nennt man *zulässige Punkte*.

3. Ein zulässiger Punkt heißt *optimale Lösung*, falls dort die Zielfunktion ihr Maximum annimmt.

3.2 Graphische Lösung

Im Fall $n = 2$ lässt sich das lineare Optimierungsproblem der Form (12) graphisch lösen. Wir demonstrieren dies anhand des obigen Beispiels.

Zunächst wird der zulässige Bereich dargestellt. Anschließend wird die Zielfunktion (Gerade)
$$g = 40x_1 + 50x_2$$
eingezeichnet und parallel so verschoben, dass sie den zulässigen Bereich gerade noch berührt (in einer Ecke).



Man erkennt, dass die optimale Lösung in einer Ecke angenommen wird. Die Ecken des zulässigen Bereichs spielen deshalb eine wichtige Rolle. Im obigen Beispiel sind die Ecken
$$(0, 0),\ (0, 47),\ (30, 38),\ (60, 20),\ (80, 0).$$

Auch im Fall $n = 3$ ist eine graphische Lösung möglich.

3.3 Normalformen

Während die Normalform 1 für die graphische Lösung günstig ist, erweist sich eine andere Darstellung (Normalform 3) für die Entwicklung von Lösungs-Algorithmen als geeigneter.

Die *Normalform 1*

$$\begin{array}{rcll} g(x) & = & q^T x & \rightarrow \quad \max \\ & & Px & \leq \quad b \\ & & x & \geq \quad 0 \end{array}$$

(mit $q \in \mathbb{R}^n,\ b \in \mathbb{R}^m,\ P \in \mathbb{R}^{m \times n}$) bringen wir durch die Einführung von *Schlupfvariablen* $s = (s_1, \ldots, s_m)^T$ in die

Normalform 2:

$$\begin{array}{rcll} g(x) & = & q^T x & \rightarrow \quad \max \\ & & Px + s & = \quad b \\ & & x & \geq \quad 0 \\ & & s & \geq \quad 0 \end{array} \quad . \tag{13}$$

Diese Darstellung ist als Sonderfall enthalten in

Normalform 3:
Gegeben seien $A \in \mathbb{R}^{m \times (m+n)}$ mit $\text{rang}(A) = m,\ b \in \mathbb{R}^m,\ c \in \mathbb{R}^{m+n}$.

$$\begin{array}{rcll} g(z) & = & c^T z & \rightarrow \quad \max \\ & & Az & = \quad b \\ & & z & \geq \quad 0 \end{array} \quad . \tag{14}$$

Mit den Setzungen

$$A = \left(P,\ \ I \right) \ , \quad z = \begin{pmatrix} x \\ s \end{pmatrix} \ , \quad c = \begin{pmatrix} q \\ 0 \end{pmatrix}$$

ergibt sich daraus die Normalform 2. Dabei ist I die $m \times m$ - Einheitsmatrix.

Anmerkungen: Alle anderen Ausgangsformen der linearen Optimierung mit linearen Nebenbedingungen lassen sich auf eine der obigen Normalformen bringen, z.B.:

1. min g ist äquivalent zu max $-g$
2. Eine Nebenbedingung der Form

$$a_{k1}x_1 + \cdots + a_{kn}x_n \ \geq \ b_k$$

wird zu

$$-a_{k1}x_1 - \cdots - a_{kn}x_n \ \leq \ -b_k$$

3. Falls eine Variable x_k keiner Vorzeichenbeschränkung unterliegt, so setzt man

$$x_k \ = \ x_{k1} - x_{k2} \ , \quad x_{k1} \geq 0,\ x_{k2} \geq 0 \ .$$

3.4 Charakterisierung des zulässigen Bereichs

Wir stellen die wesentlichen Eigenschaften des zulässigen Bereichs

$$K \ := \ \{x \in \mathbb{R}^n \ : \ x \geq 0,\ Px \leq b\} \quad \text{(Normalform 1)}$$

bzw.

$$K \;:=\; \left\{z \in \mathbb{R}^{m+n} \;:\; z \geq 0,\; Az = b\right\} \quad \text{(Normalformen 2 und 3)}$$

zusammen. Dazu sind einige Vorbereitungen notwendig.

Konvexe Mengen

Definition: Eine Teilmenge $M \subset \mathbb{R}^k$ heißt *konvex*, wenn für alle $x, y \in M$ und alle $0 \leq \lambda \leq 1$ gilt

$$z \;:=\; \lambda x \;+\; (1-\lambda)y \;\in\; M$$

(d.h. mit $x, y \in M$ liegt auch die Verbindungsstrecke $\overline{xy}$ in M).

Unmittelbar einsichtig sind folgende **Eigenschaften konvexer Mengen:**

1. Der Durchschnitt beliebig vieler konvexer Mengen ist konvex.

2. Es seien M konvex, $x^{(i)} \in M$ $(i = 1, \ldots, r)$, $\mu_i \geq 0$, und $\sum\limits_{i=1}^{r} \mu_i = 1$. Dann gilt

$$x \;:=\; \sum_{i=1}^{r} \mu_i x^{(i)} \;\in\; M \quad \textit{(konvexe Linearkombination).}$$

3. Für beliebige Vektoren $y^{(i)} \in \mathbb{R}^k$ $(i = 1, \ldots, r)$ ist die Menge

$$K \;=\; K\left(y^{(1)}, \ldots, y^{(r)}\right) \;:=\; \left\{ y \in \mathbb{R}^k \;:\; y = \sum_{i=1}^{r} \mu_i y^{(i)}\,,\; \mu_i \geq 0\,,\; \sum_{i=1}^{r} \mu_i = 1 \right\}$$

konvex. K wird als *konvexe Hülle* von $\{y^{(1)}, \ldots, y^{(r)}\}$ bezeichnet.

Definition: Sei M konvex. Ein $x \in M$ heißt *Ecke* (oder *Extrempunkt*), wenn aus $x = \mu x^{(1)} + (1-\mu)x^{(2)}$ mit $0 < \mu < 1$ und $x^{(1)}, x^{(2)} \in M$ stets $x = x^{(1)} = x^{(2)}$ folgt (d.h. x kann nicht als echte konvexe Linearkombination dargestellt werden).

Eine beliebige konvexe Menge kann unendlich viele Ecken haben (z.B. Kreis: jeder Randpunkt ist eine Ecke).

Hyperebenen und Halbräume

Definition: Sei $\alpha^T = (\alpha_1, \ldots, \alpha_k) \in \mathbb{R}^k \setminus \{0\}$ und $\gamma \in \mathbb{R}$. Die Menge

$$H \;:=\; \left\{x \in \mathbb{R}^k \;:\; \alpha^T x \;=\; \gamma\right\}$$

heißt *Hyperebene* (im $\mathbb{R}^k$).

Beispiele:

1. Hyperebenen im $\mathbb{R}^2$ sind Geraden.

2. Die Ebenen bilden die Hyperebenen im $\mathbb{R}^3$.

Eine Hyperebene teilt den $\mathbb{R}^k$ in die beiden *Halbräume*

$$H^+ \;:=\; \left\{x \in \mathbb{R}^k \;:\; \alpha^T x \;\geq\; \gamma\right\} \;\text{ und }\; H^- \;:=\; \left\{x \in \mathbb{R}^k \;:\; \alpha^T x \;\leq\; \gamma\right\} \quad .$$

Lemma: H, H^+ und H^- sind abgeschlossene konvexe Mengen.

Konvexität und Abgeschlossenheit des zulässigen Bereichs
Nebenbedingungen der Form

$$p_{i1}x_1 + \ldots + p_{in}x_n \;\leq\; b_i \quad \text{bzw.} \quad x_i \;\geq\; 0$$

liefern abgeschlossene Halbräume, Nebenbedingungen der Gestalt

$$a_{i1}z_1 + \ldots + a_{i,m+n}z_{m+n} \;=\; b_i$$

ergeben eine abgeschlossene Hyperebene. Somit ist der zulässige Bereich K der Durchschnitt von endlich vielen abgeschlossenen Hyperebenen und Halbräumen, und wir erhalten folgenden

Satz: Der zulässige Bereich K ist konvex und abgeschlossen.

In den Anwendungen ist im Normalfall der zulässige Bereich auch beschränkt, also kompakt. Der Fall $K = \emptyset$ ist natürlich möglich, dann gibt es keine optimale Lösung. Für beschränktes $K \neq \emptyset$ existiert stets eine optimale Lösung (stetige Funktion auf Kompaktum). Im Hinblick auf die Entwicklung von Algorithmen ist dann die Frage wichtig, wo die optimalen Lösungen zu suchen sind. Dabei spielen die Ecken des zulässigen Bereichs eine zentrale Rolle.

3.5 Ecken des zulässigen Bereichs

Wir beschränken uns auf die lineare Optimierungsaufgabe in der Normalform 3. Es sei

$$A \;=\; \left(a^{(1)}, a^{(2)}, \ldots, a^{(m+n)}\right)$$

mit $a^{(k)} \in \mathbb{R}^m$ (Spalten von A) und K der zulässige Bereich. Für $x \in K$ bezeichne

$$I(x) \;=\; \{j \in \{1, \ldots, m+n\} \;:\; x_j > 0\}$$

(Indizes mit positiven Komponenten).

Zwischen den Ecken des zulässigen Bereichs und den Spalten von A besteht folgender Zusammenhang:

Satz (Charakterisierungssatz):
Sei $x \in K$. Folgende Aussagen sind äquivalent:
(i) x ist eine Ecke von K.
(ii) Die Vektoren $a^{(j)},\ j \in I(x)$ sind linear unabhängig.

Beweis: (i) $\Rightarrow$ (ii)
Sei $x \in K$ eine Ecke und o.B.d.A $I(x) = \{1, 2, \ldots, r\},\ r \geq 1$. Dann gilt

$$b \;=\; Ax \;=\; \sum_{j=1}^{m+n} x_j a^{(j)} \;=\; \sum_{j=1}^{r} x_j a^{(j)} \quad ,$$

da $x_k = 0$ für $k > r$. Annahme $a^{(1)}, \ldots, a^{(r)}$ sind linear abhängig. Dann gibt es einen Vektor $(\alpha_1, \ldots, \alpha_r) \;\neq\; (0, \ldots, 0)$ mit

$$\sum_{j=1}^{r} \alpha_j a^{(j)} \;=\; 0 \;=\; A\alpha \quad .$$

Dabei wurde noch $\alpha = (\alpha_1, \ldots, \alpha_r, 0, \ldots, 0)$ gesetzt. Wegen $x_j > 0,\ j = 1, \ldots, r$ gibt es ein $\varepsilon > 0$ mit

$$x_j \pm \varepsilon\alpha_j \geq 0 \quad (j = 1, \ldots, r).$$

Damit bilden wir die Vektoren

$$\begin{aligned} y_+ &:= (x_1 + \varepsilon\alpha_1, x_2 + \varepsilon\alpha_2, \ldots, x_r + \varepsilon\alpha_r, 0, \ldots, 0) \geq 0 \quad , \\ y_- &:= (x_1 - \varepsilon\alpha_1, x_2 - \varepsilon\alpha_2, \ldots, x_r - \varepsilon\alpha_r, 0, \ldots, 0) \geq 0 \quad . \end{aligned}$$

Es gilt $y_+ \neq y_-$ (da $(\alpha_1, \ldots, \alpha_r) \neq 0$), $\frac{1}{2}y_+ + \frac{1}{2}y_- = x$ und

$$Ay_\pm = Ax \pm \varepsilon A\alpha = Ax \pm \varepsilon \sum_{j=1}^{r} \alpha_j a^{(j)} = Ax = b \quad ,$$

somit $y_+, y_- \in K$. Folglich ist x keine Ecke, was einen Widerspruch zur Voraussetzung ergibt.

(ii) $\Rightarrow$ (i)
Zu $x \in K$ seien $a^{(j)}, j \in I(x) = \{1, \ldots, r\}$ linear unabhängig. Es gelte eine Darstellung der Form

$$x = \lambda y + (1 - \lambda)z \quad \text{mit } y, z \in K \text{ und } 0 < \lambda < 1 \, .$$

Dann hat man

$$I(x) = I(y) \cup I(z) \quad .$$

Aus $Ay = Az = b$ ergibt sich

$$0 = Ay - Az = \sum_{j=1}^{m+n} (y_j - z_j)a^{(j)} = \sum_{j=1}^{r} (y_j - z_j)a^{(j)} \quad .$$

Wegen der linearen Unabhängigkeit folgt daraus

$$y_j = z_j \quad (j = 1, \ldots, r) \quad \text{und damit} \quad y = z \quad .$$

Folglich ist x eine Ecke. $\square$

Mit Hilfe dieses Satzes können wir nun die Existenz von Ecken des zulässigen Bereichs nachweisen.

Satz (Existenzsatz):
Der zulässige Bereich $K \subset \mathbb{R}^{m+n},\ K \neq \emptyset$, besitzt Ecken.

Beweis: Sei $I := \{|I(z)| : z \in K\} \subset \{0, \ldots, n + m\}$ und $\gamma = \min I \geq 0$. Wähle ein $x \in K$ mit $|I(x)| = \gamma$. Wir zeigen, dass dieses x eine Ecke ist. Der Fall $\gamma = 0$ bedeutet $x = 0 \in K$. Der Nullpunkt ist dann eine Ecke.
Für $\gamma > 0$ sei o.B.d.A. $x \in K$ so, dass $I(x) = \{1, \ldots, \gamma\}$ gilt. Annahme $a^{(1)}, \ldots, a^{(\gamma)} \in \mathbb{R}^m$ sind linear abhängig. Dann gibt es einen von Null verschiedenen Vektor $(\alpha_1, \ldots, \alpha_\gamma)$ mit

$$\sum_{j=1}^{\gamma} \alpha_j a^{(j)} = 0 \quad . \tag{15}$$

Wir setzen $\lambda := \min\left\{\frac{x_j}{|\alpha_j|} : \alpha_j \neq 0, 1 \leq j \leq \gamma\right\} =: \frac{x_k}{|\alpha_k|} = \frac{x_k}{\alpha_k}$

(O.B.d.A. sei $\alpha_k > 0$, sonst multipl. (15) mit -1) und bilden den Vektor

$$\bar{x} = (x_1 - \lambda\alpha_1, \ldots, x_\gamma - \lambda\alpha_\gamma, 0, \ldots, 0) \in \mathbb{R}^{m+n} \quad .$$

Es gilt dann

$$A\bar{x} = Ax - \lambda\sum_{j=1}^{\gamma}\alpha_j a^{(j)} = Ax = b \quad \text{und } \bar{x} \geq 0,$$

also $\bar{x} \in K$. Für den Index k gilt $\bar{x}_k = 0$, woraus $|I(\bar{x})| \leq \gamma - 1$ folgt. Dies ist ein Widerspruch zur Minimalität von γ. Also sind $a^{(1)}, \ldots, a^{(\gamma)}$ linear unabhängig. Nach dem Charakterisierungssatz ist dann x eine Ecke. □

Ist der zulässige Bereich nicht leer und beschränkt, so lässt sich jeder zulässige Punkt als Konvexkombination von Ecken darstellen, wie der folgende Satz zeigt.

Satz (Darstellungssatz):
Der zulässige Bereich sei beschränkt und nicht leer. Zu jedem $x \in K$ gibt es Ecken $z^{(1)}, \ldots, z^{(l)} \in K$ und reelle Zahlen $0 \leq \lambda_j \leq 1$, $\sum_{j=1}^{l}\lambda_j = 1$ mit

$$x = \sum_{j=1}^{l}\lambda_j z^{(j)} \quad .$$

Beweis: Siehe Hämmerlin-Hoffmann [9].

3.6 Basislösungen

Aus dem Charakterisierungssatz folgt, dass jede Ecke von K höchstens m positive Komponenten hat. Es können jedoch weniger sein. Deshalb führen wir noch den Begriff des Basispunktes ein.

Definition: Sei $A \in \mathbb{R}^{m\times(m+n)}$ mit Rang $A = m$ und $B = (a^{(i_1)}, \ldots, a^{(i_m)})$ eine Teilmatrix mit Rang $B = m$. Ein $x \in \mathbb{R}^{m+n}$ mit $x \geq 0$ heißt *Basispunkt zu* B, falls

$$x_j = 0 \text{ für } j \notin \{i_1, \ldots, i_m\} \quad \text{und} \quad \sum_{j=1}^{m} x_{i_j} a^{(i_j)} = b$$

gelten. Die Komponenten $x_{i_1}, \ldots, x_{i_m}$ werden als *Basisvariablen* bezeichnet.
Ein $x \in \mathbb{R}^{m+n}$ heißt *Basispunkt (Basislösung)*, wenn es eine Teilmatrix B von A gibt, so dass x ein Basispunkt von B ist.

Zwischen den Ecken von K und Basispunkten besteht folgender Zusammenhang:

Satz (Äquivalenzsatz):
Sei Rang $A = m$ und $x \in \mathbb{R}^{m+n}$. Dann sind äquivalent:
(i) x ist eine Ecke von K,
(ii) x ist eine Basislösung.

Beweis: (i) $\Rightarrow$ (ii)
Sei x eine Ecke von K mit den positiven Komponenten $x_{i_1}, \ldots, x_{i_p}$. Nach dem Charakterisierungssatz sind dann die Vektoren $a^{(i_1)}, \ldots, a^{(i_p)}$ linear unabhängig. Ist nun $p < m$, so ergänzen wir diese durch Hinzunahme von $m - p$ weiteren Spaltenvektoren von A zu einem System $a^{(i_1)}, \ldots, a^{(i_p)}, a^{(i_{p+1})}, \ldots, a^{(i_m)}$ linear unabhägiger Vektoren (wegen Rang $A = m$ ist dies möglich). Damit ist x ein Basispunkt.

(ii) $\Rightarrow$ (i)
Sei x ein Basispunkt. Dann gibt es eine Teilmatrix $B = (a^{(i_1)}, \ldots, a^{(i_m)})$ von A mit Rang $B = m$ und $x_j = 0$ für $j \notin \{i_1, \ldots, i_m\}$. Wegen

$$I(x) = \{k \in \{1, \ldots, m+n\} : x_k > 0\} \subset \{i_1, \ldots, i_m\}$$

sind auch $a^{(i)}$, $i \in I(x)$ linear unabhängig. Nach dem Charaktersierungssatz ist x eine Ecke. $\square$

Korollar: Die zulässige Menge K besitzt höchstens $\binom{m+n}{m}$ Ecken.

Beweis: Es gibt genau $\binom{m+n}{m}$ Möglichkeiten, m Spalten aus $m+n$ möglichen auszuwählen (siehe Kombinatorik), welche jedoch nicht immer linear unabhängig sind. $\square$

3.7 Lösung der linearen Optimierungsaufgabe

Wir befassen uns nun mit der Frage, wo die Lösung der linearen Optimierungsaufgabe zu suchen ist. Vorgelegt sei wieder die Normalform 3:

$$\begin{aligned} g(z) = c^T z &\to \max \\ Az &= b \\ z &\geq 0 \end{aligned}$$

mit dem zulässigen Bereich $K = \{z \in \mathbb{R}^{m+n} : Az = b,\ z \geq 0\}$.

Satz (Hauptsatz):
Der zulässige Bereich K sei beschränkt und nicht leer. Dann nimmt die Zielfunktion $g(z) = c^T z$ ihr Maximum in einer Ecke an.

Beweis: Da K kompakt ist (beschränkt und abgeschlossen), nimmt die Funktion g an einer Stelle $\bar{z} \in K$ ihr Maximum an. $\bar{z}$ lässt sich als Konvexkombination von Ecken darstellen:

$$\bar{z} = \sum_{j=1}^{l} \lambda_j x^{(j)} \quad \text{mit } x^{(j)} \in K \text{ Ecke}, 0 \leq \lambda_j \leq 1 \quad \text{und} \quad \sum_{j=1}^{l} \lambda_j = 1\,.$$

Dann gilt

$$\begin{aligned} \max\{c^T x^{(j)} : j = 1, \ldots, l\} &\leq \max\{c^T z : z \in K\} = c^T \bar{z} = \sum_{j=1}^{l} \lambda_j c^T x^{(j)} \\ &\leq \max\{c^T x^{(j)} : j = 1, \ldots, l\} =: c^T x^{(k)} \end{aligned}$$

und damit überall die Gleichheit. Also wird das Maximum auch in der Ecke $x^{(k)}$ angenommen. $\square$

Damit können wir uns bei der Suche nach dem Optimum auf die Ecken des zulässigen Bereichs beschränken. In einem ersten (naiven) Ansatz könnte man die Zielfunktion an allen (endlich vielen) Ecken auswerten und dann das Maximum bestimmen. Da aber $\binom{m+n}{m}$ sehr schnell wächst (für größere m und n), wäre diese Vorgehensweise sehr aufwändig (also nicht effizient).

Deshalb stellt sich sofort die Frage nach einem systematischen Vorgehen. Dies kommt beim *Simplex-Algorithmus* ("Eckenmarsch") zum Tragen. Dabei spielt der Übergang von einer Basislösung zu einer anderen die entscheidende Rolle, welchen man durch einen Gauß-Jordan-Schritt (vgl. Abschnitt 1.7) erreicht.

3.8 Der Simplex-Algorithmus

Wir betrachten die lineare Optimierungsaufgabe in der Normalform 2:

$$\begin{array}{rcl} q^T x + \alpha^{(0)} & \to & \max \\ Px + s & = & b \\ x & \geq & 0 \\ s & \geq & 0 \end{array} \quad ,$$

fordern zunächst noch $b \geq 0$ und ordnen alles in einer Matrix an:

$$H^{(0)} = \left(\begin{array}{c|c|c} P & I & b \\ \hline q^T & 0 & -\alpha^{(0)} \end{array}\right) = \left(\begin{array}{ccc|ccc|c} p_{11} & \cdots & p_{1n} & 1 & & & b_1 \\ \vdots & & \vdots & & \ddots & & \vdots \\ p_{m1} & \cdots & p_{mn} & & & 1 & b_m \\ \hline q_1 & \cdots & q_n & 0 & \cdots & 0 & -\alpha^{(0)} \end{array}\right) .$$

$H^{(0)}$ ist eine $(m+1) \times (m+n+1)$-Matrix. Hier gilt $\mathrm{Rang}(P, I) = m$ und damit die Äquivalenz von Ecken und Basislösung. Wegen der Zusatzannahme $b \geq 0$ können wir eine Basislösung (Ecke) sofort ablesen:

$$z^{(0)} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \\ s_1 \\ \vdots \\ s_m \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ b_1 \\ \vdots \\ b_m \end{pmatrix} .$$

Als Wert der Zielfunktion ergibt sich $(c^{(0)})^T z^{(0)} = 0$, welches der Eintrag in der rechten unteren Ecke ist.

Bezogen auf das Einführungsbeispiel (vgl. Abschnitt 3.1) hat man

$$H^{(0)} = \left(\begin{array}{cc|ccc|c} 1 & 1 & 1 & 0 & 0 & 80 \\ 6 & 10 & 0 & 1 & 0 & 560 \\ 30 & 100 & 0 & 0 & 1 & 4700 \\ \hline 40 & 50 & 0 & 0 & 0 & 0 \end{array}\right)$$

und als Basislösung

$$z^{(0)} \;=\; \begin{pmatrix} x_1 \\ x_2 \\ s_1 \\ s_2 \\ s_3 \end{pmatrix} \;=\; \begin{pmatrix} 0 \\ 0 \\ 80 \\ 560 \\ 4700 \end{pmatrix} \quad .$$

Nun suchen wir eine neue Basislösung (Ecke) mit einem höheren Wert der Zielfunktion. Dazu wird eine der Schlupfvariablen $s_1, \ldots, s_m$ gegen ein x_k ausgetauscht. Die Durchführung erfolgt durch einen Gauß-Jordan-Schritt, angewandt auf die komplette Matrix $H^{(0)}$. Ein solcher Schritt ist nur sinnvoll, wenn

– die neue Ecke wieder zulässig ist, d.h. nach der Transformation $b^{(1)} \geq 0$ gilt,

– der Wert der Zielfunktion nicht kleiner wird.

Es ist also ein geeignetes Pivot-Element für den Gauß-Jordan-Schritt zu wählen. Dazu betrachten wir zunächst in der letzten Zeile von $H^{(0)}$ die Einträge $q_1, \ldots, q_n$. Gilt $q_i \leq 0$ für alle $i = 1, \ldots, n$, so ist der Wert der Zielfunktion nicht mehr zu verbessern:

$$(x_1, \ldots, x_n) \;=\; (0, \ldots, 0)$$

ist die optimale Lösung. Andernfalls wähle ein $q_s > 0$ und in der zugehörigen s-ten Spalte von P ein Pivot-Element p_{rs}. Wegen der obigen Bedingungen muss dieses

$$\begin{array}{lrcl} (1) & p_{rs} & > & 0 \quad , \\ (2) & p_{rs} b_i & \geq & p_{is} b_r \quad (i = 1, \ldots, m) \text{ bzw.} \\ (2') & \frac{b_i}{p_{is}} & \geq & \frac{b_r}{p_{rs}} \quad (i = 1, \ldots, m) \end{array}$$

erfüllen. Die zweite Bedingung ergibt sich aus dem Gauß-Jordan-Schritt:

$$b_i^{(1)} \;=\; b_i \;-\; \frac{p_{is}}{p_{rs}} b_r \;\geq\; 0 \quad .$$

Im obigen Beispiel wählen wir $s = 2$ und bestimmen dann das Pivot:

$$H^{(0)} \;=\; \left(\begin{array}{cc|ccc|c||l} 1 & 1 & 1 & 0 & 0 & 80 & 80 : 1 = 80 \\ 6 & 10 & 0 & 1 & 0 & 560 & 560 : 10 = 56 \\ 30 & \boxed{100} & 0 & 0 & 1 & 4700 & 4700 : 100 = 47 \\ \hline 40 & 50 & 0 & 0 & 0 & 0 & \end{array}\right) \quad .$$

Das Pivotelement ist $p_{32} = 100$. Damit führen wir dann den Gauß-Jordan-Schritt durch:

$$\left(\begin{array}{cc|ccc|c} 1 & 1 & 1 & 0 & 0 & 80 \\ 6 & 10 & 0 & 1 & 0 & 560 \\ 0.3 & 1 & 0 & 0 & 0.01 & 47 \\ \hline 40 & 50 & 0 & 0 & 0 & 0 \end{array}\right) \quad , \qquad H^{(1)} \;=\; \left(\begin{array}{cc|ccc|c} 0.7 & 0 & 1 & 0 & -0.01 & 33 \\ 3 & 0 & 0 & 1 & -0.1 & 90 \\ 0.3 & 1 & 0 & 0 & 0.01 & 47 \\ \hline 25 & 0 & 0 & 0 & -0.5 & -2350 \end{array}\right) \quad .$$

Als neue Basislösung erhält man

$$z^{(1)} = \begin{pmatrix} x_1 \\ x_2 \\ s_1 \\ s_2 \\ s_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 47 \\ 33 \\ 90 \\ 0 \end{pmatrix}$$

und als neuen Wert für die Zielfunktion $c^T z^{(1)} = 2350$. Wir erkennen, dass $-c^T z^{(1)}$ in der rechten unteren Ecke des Tableaus steht.

Damit wurde eine neue Basislösung (Ecke) mit höherem Wert der Zielfunktion gefunden. Diese Gauß-Jordan-Austauschschritte werden nun wiederholt. Nach dem i-ten Schritt hat man eine Matrix

$$H^{(i)} = \left(\begin{array}{cccc|c} a_{11}^{(i)} & \cdots & a_{1n}^{(i)} & \cdots & a_{1,n+m}^{(i)} & b_1^{(i)} \\ \vdots & & \vdots & & \vdots & \vdots \\ a_{m1}^{(i)} & \cdots & a_{mn}^{(i)} & \cdots & a_{m,n+m}^{(i)} & b_m^{(i)} \\ \hline c_1^{(i)} & \cdots & c_n^{(i)} & \cdots & c_{n+m}^{(i)} & -\alpha^{(i)} \end{array}\right) \quad .$$

mit der Basislösung $z^{(i)}$. Vor dem nächsten Schritt werden die Optimalität und die Nichtlösbarkeit getestet.

Optimalität

Die Optimalität der Basislösung (Ecke) $z^{(i)}$ wird mit Hilfe $c^{(i)}$ (letzte Zeile von $H^{(i)}$) getestet.

Satz: Gilt $c_j^{(i)} \leq 0$ $(j = 1, \ldots, m+n)$, so ist die Basislösung $z^{(i)}$ optimale Lösung und $g(z^{(i)}) = (c^{(i)})^T z^{(i)} + \alpha^{(i)} = \alpha^{(i)}$ ist der maximale Wert.

Beweis: Nach dem Hauptsatz wird das Optimum in einer Basislösung (Ecke) angenommen. Sei also $\bar{z}$ eine beliebige Basislösung. Dann gilt

$$g(\bar{z}) = (c^{(i)})^T \bar{z} + \alpha^{(i)} = \sum_{j=1}^{m+n} \underbrace{\underbrace{c_j^{(i)}}_{\leq 0} \underbrace{\bar{z}_j}_{\geq 0}}_{\leq 0} + \alpha^{(i)} \leq \alpha^{(i)} = (c^{(i)})^T z^{(i)} + \alpha^{(i)} = g(z^{(i)})$$

(Beachte $c_l^{(i)} = 0$ für die Basisvariablen von $z^{(i)}$). □

Nichtlösbarkeit

Satz: Sei $c_s^{(i)} > 0$ und $a_{js}^{(i)} \leq 0$ für $j = 1, \ldots, m$. Dann besitzt die lineare Optimierungsaufgabe keine Lösung.

Beweis: Sei $z^{(i)}$ die Basislösung mit den Basisvariablen $z_{j_1}^{(i)}, \ldots, z_{j_m}^{(i)}$. Dann gilt

$$c_{j_1}^{(i)} = \cdots = c_{j_m}^{(i)} = 0 \ , \ \ z_k^{(i)} = 0 \text{ für } k \notin \{j_1, \ldots, j_m\} \tag{16}$$

und für die Zielfunktion

$$g(z^{(i)}) = (c^{(i)})^T z^{(i)} + \alpha^{(i)} = \alpha^{(i)} \ .$$

Außerdem hat die Matrix $H^{(i)}$ in der j_l-ten Spalte den Einheitsvektor e_l.
Zu $\lambda > 0$ konstruieren wir eine neue zulässige Lösung $y = (y_1, \ldots, y_{m+n})$ wie folgt:

$$\begin{aligned} y_{j_l} &= z_{j_l}^{(i)} - \lambda a_{ls}^{(i)} \;\geq\; 0 \;\; (l = 1, \ldots, m) \;, \\ y_s &= \lambda \;, \\ y_k &= 0 \;, \;\; k \notin \{j_1, \ldots, j_m, s\} \;. \end{aligned}$$

Dann gilt $y \geq 0$ und

$$A^{(i)} y \;=\; \sum_{l=1}^{m} \left(z_{j_l}^{(i)} - \lambda a_{ls}^{(i)} \right) e_l \;+\; \lambda \begin{pmatrix} a_{1s}^{(i)} \\ \vdots \\ a_{ms}^{(i)} \end{pmatrix} \;=\; \sum_{l=1}^{m} z_{j_l}^{(i)} e_l \;=\; A^{(i)} z^{(i)} \;=\; b^{(i)} \;.$$

Somit ist y eine zulässige Lösung. Für die Zielfunktion gilt unter Beachtung von (16)

$$g(y) \;=\; \alpha^{(i)} \;+\; \underbrace{c_s^{(i)}}_{>0} \lambda \quad \to \infty \;\text{ für } \lambda \to \infty \;.$$

Also gibt es kein Maximum. □

Neue Basislösung
Falls die obigen beiden Kriterien nicht erfüllt sind, so findet man eine neue Basislösung (Ecke) $z^{(i+1)}$ mit $g(z^{(i+1)}) \;\geq\; g(z^{(i)})$.

Wähle ein $c_s^{(i)} > 0$ und ein Pivot $a_{rs}^{(i)} > 0$ mit der Eigenschaft

$$b_j^{(i)} a_{rs}^{(i)} \;\geq\; b_r^{(i)} a_{js}^{(i)} \quad (j = 1, \ldots, m). \tag{17}$$

Satz: Mit diesem Pivot-Element liefert die Gauß-Jordan-Elimination eine neue Basislösung $z^{(i+1)}$ mit $g(z^{(i+1)}) \;\geq\; g(z^{(i)})$.

Beweis: Wir haben in der Ausgangssituation

$$g(z^{(i)}) \;=\; {c^{(i)}}^T z^{(i)} + \alpha^{(i)} \;=\; \alpha^{(i)}$$

(beachte $c_j^{(i)} = 0$ für die Basisvariablen und $z_k^{(i)} = 0$ für die Nichtbasisvariablen). Der Gauß-Jordan-Schritt liefert

$$b_j^{(i+1)} \;=\; b_j^{(i)} - \frac{a_{js}^{(i)}}{a_{rs}^{(i)}} b_r^{(i)} \;=\; \frac{\overbrace{b_j^{(i)} a_{rs}^{(i)} - b_r^{(i)} a_{js}^{(i)}}^{\geq 0}}{\underbrace{a_{rs}^{(i)}}_{>0}} \;\geq\; 0 \quad ,$$

woraus die Zulässigkeit folgt. Weiter erhalten wir

$$-g(z^{(i+1)}) \;=\; -\alpha^{(i+1)} \;=\; -\alpha^{(i)} - \frac{\overbrace{c_s^{(i)}}^{\geq 0}}{\underbrace{a_{rs}^{(i)}}_{>0}} \overbrace{b_r^{(i)}}^{\geq 0} \;\leq\; -\alpha^{(i)} \;=\; -g(z^{(i)})$$

und somit

$$g(z^{(i+1)}) \;\geq\; g(z^{(i)}) \quad . \;\square$$

Anmerkung: Die Eigenschaft (17) braucht man nur für die j mit $a_{js}^{(i)} > 0$ nachzuprüfen. Dann ist (17) aber äquivalent zu

$$\frac{b_j^{(i)}}{a_{js}^{(i)}} \;\geq\; \frac{b_r^{(i)}}{a_{rs}^{(i)}} \quad . \tag{18}$$

Diese Beziehung ist für die Handrechnung einfacher und liefert sofort Information, welches Pivot-Element zu wählen ist.

Wir demonstrieren die Funktionsweise des Simplex-Verfahrens am obigen Beispiel. Der erste Schritt wurde bereits vorgeführt:

$$H^{(1)} \;=\; \left(\begin{array}{ccc|cc|c} 0.7 & 0 & 1 & 0 & -0.01 & 33 \\ \boxed{3} & 0 & 0 & 1 & -0.1 & 90 \\ 0.3 & 1 & 0 & 0 & 0.01 & 47 \\ \hline 25 & 0 & 0 & 0 & -0.5 & -2350 \end{array}\right) \quad \text{mit } z^{(1)} \;=\; \begin{pmatrix} x_1 \\ x_2 \\ s_1 \\ s_2 \\ s_3 \end{pmatrix} \;=\; \begin{pmatrix} 0 \\ 47 \\ 33 \\ 90 \\ 0 \end{pmatrix} \quad .$$

Das Optimalitäts-Kriterium und das Nichtlösbarkeits-Kriterium sind nicht erfüllt. Wir wählen $c_1^{(1)} = 25$ und als Pivot $a_{21}^{(1)} = 3$ und führen den nächsten Gauß-Jordan-Schritt durch.

$$\left(\begin{array}{ccc|cc|c} 0.7 & 0 & 1 & 0 & -\frac{1}{100} & 33 \\ \boxed{1} & 0 & 0 & \frac{1}{3} & -\frac{1}{30} & 30 \\ 0.3 & 1 & 0 & 0 & \frac{1}{100} & 47 \\ \hline 25 & 0 & 0 & 0 & -\frac{1}{2} & -2350 \end{array}\right) \quad , \qquad H^{(2)} \;=\; \left(\begin{array}{ccc|cc|c} 0 & 0 & 1 & -\frac{7}{30} & \frac{4}{300} & 12 \\ 1 & 0 & 0 & \frac{1}{3} & -\frac{1}{30} & 30 \\ 0 & 1 & 0 & -\frac{1}{10} & \frac{1}{50} & 38 \\ \hline 0 & 0 & 0 & -\frac{25}{3} & \frac{1}{3} & -3100 \end{array}\right) \quad .$$

Es ist ein weiterer Gauß-Jordan-Schritt mit dem Pivot-Element $a_{15}^{(2)} = \frac{4}{300}$ möglich:

$$\left(\begin{array}{ccc|cc|c} 0 & 0 & \frac{300}{4} & -\frac{70}{4} & \boxed{1} & 900 \\ 1 & 0 & 0 & \frac{1}{3} & -\frac{1}{30} & 30 \\ 0 & 1 & 0 & -\frac{1}{10} & \frac{1}{50} & 38 \\ \hline 0 & 0 & 0 & -\frac{25}{3} & \frac{1}{3} & -3100 \end{array}\right) \quad , \qquad H^{(3)} \;=\; \left(\begin{array}{ccc|cc|c} 0 & 0 & \frac{300}{4} & -\frac{70}{4} & 1 & 900 \\ 1 & 0 & \frac{10}{4} & \frac{11}{12} & 0 & 60 \\ 0 & 1 & -\frac{3}{2} & \frac{1}{4} & 0 & 20 \\ \hline 0 & 0 & -\frac{100}{4} & -\frac{5}{12} & 0 & -3400 \end{array}\right) \quad .$$

Nun ist das Optimalitäts-Kriterium erfüllt. Als optimale Lösung ergibt sich

$$z \;=\; \begin{pmatrix} x_1 \\ x_2 \\ s_1 \\ s_2 \\ s_3 \end{pmatrix} \;=\; \begin{pmatrix} 60 \\ 20 \\ 0 \\ 0 \\ 900 \end{pmatrix}$$

und als Wert für die Zielfunktion $g(z) = 3400$.

Wir fassen diese Ergebnisse nochmals in einem Rechentableau zusammen (empfohlen für die Handrechnung). Dabei stehen oberhalb der ersten Zeile alle Variablen, links von der ersten Spalte die aktuellen Basisvariablen.

	x_1	x_2	s_1	s_2	s_3	b_i	
s_1	1	1	1	0	0	80	$80 : 1 = 80$
s_2	6	10	0	1	0	560	$560 : 10 = 56$
s_3	30	$\boxed{100}$	0	0	1	4700	$4700 : 100 = 47$
	40	50	0	0	0	0	
	1	1	1	0	0	80	
	6	10	0	1	0	560	
	0.3	1	0	0	0.01	47	
	40	50	0	0	0	0	
s_1	0.7	0	1	0	-0.01	33	$33 : 0.7 > 30$
s_2	$\boxed{3}$	0	0	1	-0.1	90	$90 : 3 = 30$
x_2	0.3	1	0	0	0.01	47	$47 : 0.3 > 30$
	25	0	0	0	-0.5	-2350	
	0.7	0	1	0	$-\frac{1}{100}$	33	
	1	0	0	$\frac{1}{3}$	$-\frac{1}{30}$	30	
	0.3	1	0	0	$\frac{1}{100}$	47	
	25	0	0	0	$-\frac{1}{2}$	-2350	
s_1	0	0	1	$-\frac{7}{30}$	$\boxed{\frac{4}{300}}$	12	$12 \cdot \frac{300}{4} = 900$
x_1	1	0	0	$\frac{1}{3}$	$-\frac{1}{30}$	30	
x_2	0	1	0	$-\frac{1}{10}$	$\frac{1}{50}$	38	$38 \cdot 50 = 1900$
	0	0	0	$-\frac{25}{3}$	$\frac{1}{3}$	-3100	
	0	0	$\frac{300}{4}$	$-\frac{70}{4}$	$\boxed{1}$	900	
	1	0	0	$\frac{1}{3}$	$-\frac{1}{30}$	30	
	0	1	0	$-\frac{1}{10}$	$\frac{1}{50}$	38	
	0	0	0	$-\frac{25}{3}$	$\frac{1}{3}$	-3100	
s_3	0	0	$\frac{300}{4}$	$-\frac{70}{4}$	1	900	
x_1	1	0	$\frac{10}{4}$	$-\frac{1}{4}$	0	60	
x_2	0	1	$-\frac{3}{2}$	$\frac{1}{4}$	0	20	
	0	0	$-\frac{100}{4}$	$-\frac{5}{2}$	0	-3400	

Nun ist das Optimalitäts-Kriterium erfüllt, und wir lesen aus dem Tableau die optimale Lösung ab:

$$x_1 \;=\; 60\,, \quad x_2 \;=\; 20\,, \quad s_3 \;=\; 900\,, \quad g \;=\; 3400\,.$$

Anmerkung: Die Schlupfvariable $s_3 = 900$ bedeutet, dass in der 3. Ungleichung des Ausgangsproblems (Beschränkung Kapital) noch "Luft" ist: Der maximale Kapitaleinsatz von 4700 Euro wurde um 900 Euro unterschritten.

3.9 Zweiphasenmethode

Das im vorigen Abschnitt geschilderte Simplex-Verfahren benötigt zum Start eine Basislösung. Für $b \geq 0$ ist dies

$$z^{(0)} \;=\; \begin{pmatrix} x \\ s \end{pmatrix} \;=\; \begin{pmatrix} 0 \\ b \end{pmatrix} \quad .$$

Sei nun ein $b_i < 0$; dann ist obiges $z^{(0)}$ nicht mehr zulässig, also auch keine Basislösung. Deshalb braucht man zunächst eine Anlaufrechnung, um eine zulässige Basislösung zu finden (*1. Phase*).

1. Phase:

Betrachte das lineare Optimierungsproblem in der Normalform 2:

$$\begin{aligned} g(x) \;&=\; q^T x + \alpha^{(0)} \;\to\; \max \\ Px + s \;&=\; b \\ x \;&\geq\; 0 \quad . \\ s \;&\geq\; 0 \end{aligned} \tag{19}$$

O.B.d.A. seien die Nebenbedingungen so geordnet, dass $b_i < 0$ für $i = 1, \ldots, p$ und $b_i \geq 0$ für $i = p+1, \ldots, m$ gilt. Für alle i mit $b_i < 0$ führen wir neue Hilfsvariablen ξ_i ein und betrachten dann

$$\begin{aligned} -\sum_{j=1}^{n} p_{ij} x_j \;-\; s_i \;+\; \xi_i \quad &=\quad -b_i \quad (i = 1, \ldots, p)\,, \\ \sum_{j=1}^{n} p_{ij} x_j \;+\; s_i \quad &=\quad b_i \quad (i = p+1, \ldots, m)\,, \\ x \geq 0, \quad s \geq 0, \quad \xi \geq 0\,. & \end{aligned} \tag{20}$$

Für $\xi = (\xi_i, \ldots, \xi_p) = (0, \ldots, 0)$ ist dies äquivalent zu (19). Man definiert nun die neue Zielfunktion

$$h(\xi) \;:=\; -\sum_{i=1}^{p} \xi_i$$

und löst dann das Hilfsproblem

$$\text{maximiere } h(\xi) \;\text{ unter den Nebenbedingungen (20)}$$

(dabei ist die Zielfunktion in den Nichtbasisvariablen anzugeben).

Dies stellt wieder ein lineares Optimierungsproblem dar, von dem man jedoch eine zulässige Basislösung sofort angeben kann:

$$\begin{aligned} y^{(0)} &= (x_1, \ldots, x_n, s_1, \ldots, s_p, s_{p+1}, \ldots s_m, \xi_1, \ldots, \xi_p) \\ &= (0, \ldots, 0, 0, \ldots, 0, b_{p+1}, \ldots, b_m, -b_1, \ldots, -b_p) \quad . \end{aligned} \tag{21}$$

Eigenschaften des Hilfsproblems:

1. (x, s, ξ) ist optimale Lösung des Hilfsproblems $\Leftrightarrow \quad \xi = 0$.
2. Ist (x, s, ξ) eine optimale Lösung des Hilfsproblems, so ist (x, s) eine zulässige Basislösung des Ausgangsproblems (19).

2. Phase

Löse mit der gefundenen Basislösung und einer entsprechend modifizierten Zielfunktion (Zielfunktion muss in den Nichtbasisvariablen dargestellt werden) das Ausgangsproblem mit dem Simplex-Algorithmus.

Beispiel: Maximiere die Zielfunktion $g(x) = -x_1 + 2x_2$ unter den Nebenbedingungen

$$\begin{aligned} 2x_1 + x_2 &\leq 200 \\ -x_1 + x_2 &\leq 40 \\ x_1 + x_2 &\geq 50 \\ x_1, x_2 &\geq 0 \end{aligned} \quad .$$

Zunächst bringen wir diese Aufgabe auf die Normalform 1:

$$\begin{aligned} 2x_1 + x_2 &\leq 200 \\ -x_1 + x_2 &\leq 40 \\ -x_1 - x_2 &\leq -50 \\ x_1, x_2 &\geq 0 \end{aligned}$$

und durch Einführen von Schlupfvariablen auf die Normalform 2:

$$\begin{aligned} 2x_1 + x_2 + s_1 &= 200 \\ -x_1 + x_2 + s_2 &= 40 \\ -x_1 - x_2 + s_3 &= -50 \quad \leftarrow b \geq 0 \text{ hier verletzt!} \end{aligned}$$

Deshalb betrachtet man in Phase 1 das Hilfsproblem

$$\begin{aligned} 2x_1 + x_2 + s_1 &= 200 \\ -x_1 + x_2 + s_2 &= 40 \\ x_1 + x_2 - s_3 + \xi_1 &= 50 \\ x_1, x_2, s_1, s_2, s_3, \xi_1 &\geq 0 \end{aligned}$$

mit der Zielfunktion

$$h(\xi) = -\xi_1 = x_1 + x_2 - s_3 - 50 \ \rightarrow \ \max$$

und löst dieses mit dem Simplex-Algorithmus:

	x_1	x_2	s_1	s_2	s_3	ξ_1	b_i	
s_1	2	1	1	0	0	0	200	$200 : 2 = 100$
s_2	-1	1	0	1	0	0	40	
ξ_1	$\boxed{1}$	1	0	0	-1	1	50	$50 : 1 = 50$
	1	1	0	0	-1	0	50	
s_1	0	-1	1	0	2	-2	100	
s_2	0	2	0	1	-1	1	90	
x_1	1	1	0	0	-1	1	50	
	0	0	0	0	0	-1	0	

Damit wurde eine optimale Lösung des Hilfsproblems erreicht:

$$(x_1, x_2, s_1, s_2, s_3, \xi_1) \;=\; (50, 0, 100, 90, 0, 0) \quad .$$

Dann ist $(x_1, x_2, s_1, s_2, s_3) \;=\; (50, 0, 100, 90, 0)$ eine zulässige Basislösung des Ausgangsproblems mit den Basisvariablen x_1, s_1, s_2 und den Nichtbasisvariablen x_2, s_3.

Die Zielfunktion $g(z) \;=\; -x_1 + 2x_2$ muss nun in den Nichtbasisvariablen geschrieben werden: Aus $50 = x_1 + x_2 - s_3$ folgt

$$g(z) \;=\; -50 + x_2 - s_3 + 2x_2 \;=\; 3x_2 - s_3 - 50 \; .$$

In der Phase 2 startet man das Simplex-Verfahren mit der in Phase 1 gefundenen Basislösung:

	x_1	x_2	s_1	s_2	s_3	b_i	
s_1	0	-1	1	0	2	100	
s_2	0	$\boxed{2}$	0	1	-1	90	$90 : 2 = 45$
x_1	1	1	0	0	-1	50	$50 : 1 = 50$
	0	3	0	0	-1	50	
s_1	0	0	1	$\frac{1}{2}$	$\boxed{\frac{3}{2}}$	145	
x_2	0	1	0	$\frac{1}{2}$	$-\frac{1}{2}$	45	
x_1	1	0	0	$-\frac{1}{2}$	$-\frac{1}{2}$	5	
	0	0	0	$-\frac{3}{2}$	$\frac{1}{2}$	-85	
s_3	0	0	$\frac{2}{3}$	$\frac{1}{3}$	1	$\frac{290}{3}$	
x_2	0	1	$\frac{1}{3}$	$\frac{2}{3}$	0	$\frac{280}{3}$	
x_1	1	0	$\frac{1}{3}$	$-\frac{1}{3}$	0	$\frac{160}{3}$	
	0	0	$-\frac{1}{3}$	$-\frac{5}{3}$	0	$-\frac{400}{3}$	

Hier ist das Optimalitätskriterium erfüllt:

$$x_1 = \frac{160}{3} \; , \quad x_2 = \frac{280}{3} \; , \quad g(x) = \frac{400}{3} \; .$$

4 Interpolation

4.1 Die Interpolationsaufgabe nach Lagrange

Gegeben seien Paare $(t_i, s_i) \in \mathbb{R}^2$ $(i = 0, 1, 2, \ldots, m)$; die t_i seien paarweise verschieden. Gesucht ist ein Polynom

$$p(t) \;:=\; \sum_{j=0}^{m} a_j t^j$$

vom Grad$(p) \leq m$ mit

$$p(t_i) \;=\; s_i \quad (i = 0, 1, \ldots, m). \tag{22}$$

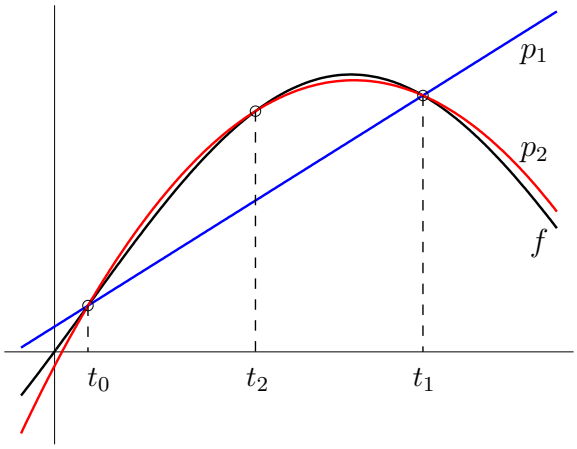
Definition: Diese Problemstellung nennt man die *Interpolationsaufgabe nach Lagrange*. Die t_i heißen die *Stützstellen* und die s_i die *Stützwerte*. Die Lösung $p(t)$ heißt das *Lagrange-Interpolationspolynom* zu den Paaren (t_i, s_i).

Anwendung: Seien $f : [a, b] \to \mathbb{R}$ eine reelle Funktion und $a \leq t_0 < t_1 < \cdots < t_m \leq b$ die Stützstellen. Wähle als Stützwerte nun

$$s_i \;:=\; f(t_i) \quad (i = 0, 1, \ldots, m).$$

Das Interpolationspolynom $p(t)$ liefert eine Näherung (Approximation) an die Funktion f und stimmt an den Stützstellen mit f überein (d.h. $p(t_i) = f(t_i),\ i = 0, 1, \ldots, m$). Man nennt p das *Interpolationspolynom zu f bzgl. der Stützstellen $t_0, \ldots, t_m$*.

Interpolationspolynome zu f



p_1 ist das Interpolationspolynom bzgl. der Stützstellen $t_0,\ t_1$

p_2 ist das Interpolationspolynom bzgl. der Stützstellen $t_0,\ t_1,\ t_2$

Vorteile:

1. $p(t)$ hat eine einfache Gestalt.
2. $p(t)$ kann leicht berechnet werden (vgl. Horner-Schema).
3. Polynome können einfach differenziert bzw. integriert werden.

Beachte: Das Interpolationspolynom hängt auch von der Stützstellenwahl ab.

Im Zusammenhang mit der Approximation sind Aussagen über den (Formel-)Fehler

$$f(t) \; - \; p(t) \quad (t \in [a,b])$$

von großer Bedeutung (vgl. unten).

Zunächst beschäftigen wir uns mit der Frage der Existenz und Eindeutigkeit der Interpolationsaufgabe nach Lagrange.

4.2 Existenz und Eindeutigkeit

Im Folgenden bezeichne Π_m den Raum der (reellen) Polynome vom Grad $\leq m$.

Satz: Die obige Interpolationsaufgabe besitzt genau eine Lösung.

Beweis: (i) Eindeutigkeit:
Seien $p(t)$ und $q(t)$ mit $p, q \in \Pi_m$ Lösungen der obigen Interpolationsaufgabe. Setze

$$r(t) \; := \; p(t) \; - \; q(t) \quad .$$

Dann gilt $r \in \Pi_m$ und

$$r(t_i) \; = \; p(t_i) - q(t_i) \; = \; s_i - s_i \; = \; 0 \qquad (i = 0, 1, \ldots, m). \tag{23}$$

Das Polynom r hat mindestens $m+1$ Nullstellen. Nach dem Fundamentalsatz der Algebra hat aber jedes Polynom vom Grad m, das ungleich dem Nullpolynom ist, höchstens m Nullstellen. Also folgt aus (23), dass r das Nullpolynom ist und somit $p = q$.
(ii) Existenz:
Wir geben ein $p \in \Pi_m$ explizit an und zeigen, dass dieses die Interpolationsaufgabe löst. Dazu sei

$$l_i(t) \; := \; \prod_{\substack{j \, = \, 0 \\ j \, \neq \, i}}^{m} \frac{t - t_j}{t_i - t_j} \quad (i = 0, 1, \ldots, m).$$

Die l_i heißen *Lagrange-Grundpolyome*. Es gilt $l_i \in \Pi_m$ und

$$l_i(t_k) \; = \; \begin{cases} 1 & \text{für } i = k \\ 0 & \text{für } i \neq k \end{cases} \qquad (i, k = 0, 1, \ldots, m). \tag{24}$$

Das Polynom

$$p(t) \; := \; \sum_{i=0}^{m} s_i \, l_i(t) \tag{25}$$

löst dann die Interpolationsaufgabe; denn es gilt $p \in \Pi_m$ und mit (24)

$$p(t_k) \; = \; \sum_{i=0}^{m} s_i l_i(t_k) \; = \; s_k \quad (k = 0, 1, \ldots, m) \, . \quad \square$$

Die Darstellung (25) nennt man *Interpolationsformel nach Lagrange*. Diese erweist sich als sehr nützlich für Beweise, ist jedoch für praktische Zwecke weniger geeignet (insbesondere für größere m).

Anmerkung: Einen alternativen Beweis erhält man durch den Ansatz

$$p(t) \;=\; a_0 \,+\, a_1 t \,+\, \ldots \,+\, a_m t^m \;.$$

Die Interpolationsbedingungen (22) liefern dann für die Koeffizienten $a_0, \ldots, a_m$ ein lineares Gleichungssystem, welches eindeutig lösbar ist (Vandermonde-Matrix).

4.3 Der Neville-Algorithmus

Es sei wieder $p \in \Pi_m$ das Interpolationspolynom zu den Paaren $(t_i, s_i),\ i = 0, 1, \ldots, m$, t_i paarweise verschieden.

Lemma: Es sei $p_{m-1}(t)$ das IP-Polynom zu den Paaren $(t_i, s_i),\ i = 0, 1, \ldots, m-1$, und $q_{m-1}(t)$ das IP-Polynom zu den Paaren $(t_i, s_i),\ i = 1, 2, \ldots, m$. Dann gilt

$$p(t) \;=\; \frac{(t - t_m)p_{m-1}(t) - (t - t_0)q_{m-1}(t)}{t_0 - t_m} \quad . \tag{26}$$

.

Beweis: Es bezeichne $r(t)$ die rechte Seite von (26). Es gilt $q_{m-1}, p_{m-1} \in \Pi_{m-1}$ und folglich $r \in \Pi_m$. Weiter erhält man

$$\begin{aligned}
r(t_i) \;&=\; \frac{(t_i - t_m)p_{m-1}(t_i) \;-\; (t_i - t_0)q_{m-1}(t_i)}{t_0 - t_m} \\
&=\; \frac{(t_i - t_m)s_i \;-\; (t_i - t_0)s_i}{t_0 - t_m} \;=\; s_i \quad (i = 1, \ldots, m-1)\,, \\
r(t_0) \;&=\; \frac{(t_0 - t_m)s_0}{t_0 - t_m} \;=\; s_0 \quad , \\
r(t_m) \;&=\; \frac{-(t_m - t_0)s_m}{t_0 - t_m} \;=\; s_m \quad .
\end{aligned}$$

Somit löst $r(t)$ die Interpolationsaufgabe zu den Paaren $(t_i, s_i),\ i = 0, 1, \ldots, m$. Aus der Eindeutigkeit der Interpolation folgt $r = p$. $\square$

Das obige Lemma liefert uns eine Rekursionsformel zur Berechnung von $p(\xi),\ \xi \in \mathbb{R}$. Mit $p_{i_0 \cdots i_k} \in \Pi_k$ bezeichnen wir das IP-Polynom zu den Paaren $(t_{i_0}, s_{i_0}), (t_{i_1}, s_{i_1}), \ldots, (t_{i_k}, s_{i_k})$. Zur Berechnung von $p(\xi) = p_{01\cdots m}(\xi)$ verwendet man folgendes Schema:

	$k=0$	$k=1$	$k=2$	$k=3$	$\ldots$	$k=m$
t_0	$s_0 = p_0(\xi)$					
		$p_{01}(\xi)$				
t_1	$s_1 = p_1(\xi)$		$p_{012}(\xi)$			
		$p_{12}(\xi)$		$p_{0123}(\xi)$		
t_2	$s_2 = p_2(\xi)$		$p_{123}(\xi)$		$\ddots$	
		$p_{23}(\xi)$		$\vdots$		$p_{012\cdots m}(\xi)$
t_3	$s_3 = p_3(\xi)$		$\vdots$			
		$\vdots$				
$\vdots$	$\vdots$					
		$p_{m-1,m}(\xi)$				
t_m	$s_m = p_m(\xi)$					

Dabei ergibt sich $p_{i_0 i_1 \cdots i_k}(\xi)$ aus $p_{i_0 i_1 \cdots i_{k-1}}(\xi)$ und $p_{i_1 i_2 \cdots i_k}(\xi)$ gemäß (26):

$$p_{i_0 i_1 \cdots i_k}(\xi) \;=\; \frac{(\xi - t_{i_k}) p_{i_0 \cdots i_{k-1}}(\xi) \;-\; (\xi - t_{i_0}) p_{i_1 \cdots i_k}(\xi)}{t_{i_0} \;-\; t_{i_k}} \quad . \tag{27}$$

So gilt zum Beispiel

$$p_{123}(\xi) \;=\; \frac{(\xi - t_3) p_{12}(\xi) \;-\; (\xi - t_1) p_{23}(\xi)}{t_1 - t_3} \quad .$$

$p_{i_0 i_1 \cdots i_k}(\xi)$ ist eine reelle Zahl, nämlich das Interpolationspolynom zu den Paaren $(t_{i_0}, s_{i_0}), \ldots, (t_{i_k}, s_{i_k})$, ausgewertet an der Stelle ξ.

Definition: Das obige Rechentableau wird als *Neville-Schema* bezeichnet.

Vorteil: Nimmt man ein weiteres Paar (t_{m+1}, s_{m+1}) zu den bisherigen Stützpaaren (t_i, s_i), $i = 0, 1, \ldots, m$ hinzu, so ergibt sich $p_{01 \cdots m+1}(\xi)$ durch Anfügen einer weiteren Zeile im Neville-Schema. Bisher schon berechnete Werte können verwendet werden. Bei der Darstellung nach Lagrange (vgl. (25)) ist dagegen eine komplette Neuberechnung erforderlich.

Beispiel: Es seien $m = 2,\ \xi = 2$ und

$$\begin{array}{lll} t_0 = 0, & t_1 = 1, & t_2 = 3, \\ s_0 = 1, & s_1 = 3, & s_2 = 2. \end{array}$$

Dann erhält man

	$k=0$	$k=1$	$k=2$
$t_0 = 0$	$s_0 = p_0(2) = 1$		
		$p_{01}(2) = 5$	
$t_1 = 1$	$s_1 = p_1(2) = 3$		$p_{012}(2) = \frac{10}{3}$
		$p_{12}(2) = \frac{5}{2}$	
$t_2 = 3$	$s_2 = p_2(2) = 2$		

Die Nebenrechnungen lauten:

$$\begin{aligned} p_{01}(2) &= \tfrac{(2-1)1-(2-0)3}{0-1} = 5\,, \\ p_{12}(2) &= \tfrac{(2-3)3-(2-1)2}{1-3} = \tfrac{5}{2}\,, \\ p_{012}(2) &= \tfrac{(2-3)5-(2-0)\frac{5}{2}}{0-3} = \tfrac{10}{3} = p(\xi)\,. \end{aligned}$$

4.4 Der Interpolationsfehler

Wir nehmen nun an, dass die Stützwerte s_i von einer Funktion $f : [a, b] \to \mathbb{R}$ herrühren, d.h. $s_i = f(t_i)$. Ferner gelte $a \le t_0 < \cdots < t_m \le b$. Uns interessiert dann die Frage, wie gut das Interpolationspolynom $p(t)$ zu den Paaren

$$(t_i, f(t_i)) \qquad (i = 0, 1, \ldots, m)$$

die Funktion f approximiert. Mit $C^k[a, b]$ bezeichnen wir den Raum der k-mal stetig differenzierbaren Funktionen $f : [a, b] \to \mathbb{R}$.

Satz: Sei $f \in C^{m+1}[a,b]$ und $z \in [a,b]$ fest. Wir setzen $\alpha := \min\{t_0, z\}$ und $\beta := \max\{t_m, z\}$. Dann gibt es ein $\xi \in (\alpha, \beta)$, so dass für den Interpolationsfehler gilt

$$f(z) - p(z) \;=\; \frac{f^{(m+1)}(\xi)}{(m+1)!} \cdot \prod_{i=0}^{m} (z - t_i) \quad .$$

Beweis: Ist z eine Stützstelle (d.h. $z = t_i$ für ein i), so ist die Behauptung trivialerweise erfüllt. Im Folgenden sei deshalb $z \notin \{t_0, t_1, \ldots, t_m\}$. Zur Abkürzung setzen wir

$$w(t) \;:=\; \prod_{i=0}^{m} (t - t_i)$$

und betrachten die Hilfsfunktion

$$\Phi(t) \;:=\; f(t) - p(t) - \frac{w(t)}{w(z)} \cdot (f(z) - p(z)) \quad .$$

Dann gilt $\Phi \in C^{m+1}[a,b]$ und

$$\begin{aligned} \Phi^{(m+1)}(t) &= f^{(m+1)}(t) - p^{(m+1)}(t) - \frac{w^{(m+1)}(t)}{w(z)} \cdot (f(z) - p(z)) \\ &= f^{(m+1)}(t) - \frac{(m+1)!}{w(z)} \cdot (f(z) - p(z)) \quad . \end{aligned} \tag{28}$$

Weiter erhält man

$$\begin{aligned} \Phi(t_i) &= 0 \qquad (i = 0, 1, \ldots, m), \\ \Phi(z) &= f(z) - p(z) - \frac{w(z)}{w(z)} \cdot (f(z) - p(z)) \;=\; 0 \quad . \end{aligned}$$

Somit hat Φ im Intervall $[\alpha, \beta]$ mindestens $m+2$ verschiedene Nullstellen. Der Satz von Rolle liefert nun jeweils:

Φ' hat in (α, β) mindestens $m+1$ verschiedene Nullstellen,

$\Phi^{(2)}$ hat in (α, β) mindestens m verschiedene Nullstellen,

$\Phi^{(3)}$ hat in (α, β) mindestens $m-1$ verschiedene Nullstellen,

$\vdots$

$\Phi^{(m)}$ hat in (α, β) mindestens 2 verschiedene Nullstellen,

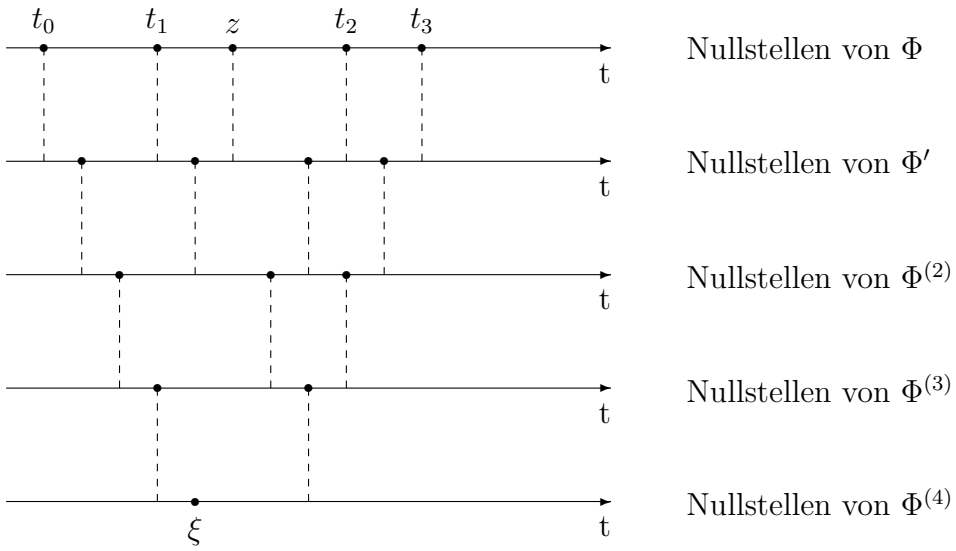
$\Phi^{(m+1)}$ hat in (α, β) mindestens eine Nullstelle ξ.

Diese Nullstelle setzen wir in (28) ein:

$$0 \;=\; \Phi^{(m+1)}(\xi) \;=\; f^{(m+1)}(\xi) - \frac{(m+1)!}{w(z)} \cdot (f(z) - p(z)) \quad .$$

Daraus folgt die Behauptung. $\square$

Wir veranschaulichen die obige Schlussweise nochmals für $m = 3$:



Im Normalfall ist jedoch der Zwischenwert ξ und damit auch $f^{(m+1)}(\xi)$ nicht explizit bekannt. Deshalb ist die Fehlerdarstellung im obigen Satz für die Praxis nicht brauchbar. Sie bildet aber die Grundlage für Fehlerabschätzungen.

Fehlerschranken
Wir versehen den Raum $C[a, b]$ mit der Maximum-Norm:

$$\|g\|_{[a,b]} \;:=\; \|g\|_\infty \;:=\; \|g\| \;:=\; \max\{|g(t)| \,:\, t \in [a, b]\} \quad .$$

Weiter sei wieder

$$w(t) \;:=\; \prod_{i=0}^{m}(t - t_i) \quad \text{(Knotenpolynom)}$$

und $f \in C^{m+1}[a, b]$. Dann erhält man aus dem obigen Satz die Abschätzungen

$$|f(z) - p(z)| \;\le\; \frac{\left\|f^{(m+1)}\right\|}{(m+1)!}\cdot|w(z)| \qquad \text{für jedes } z \in [a, b]$$

bzw.

$$\|f - p\| \;\le\; \frac{\left\|f^{(m+1)}\right\|}{(m+1)!}\cdot\|w\| \quad . \tag{29}$$

Das Knotenpolynom $w(t)$ und damit auch $\|w\|$ hängen von den gewählten Stützstellen $t_0, t_1, \ldots, t_m$ ab. Deshalb stellt sich sofort die Frage nach einer "günstigen" Stützstellenwahl: Wie sind die t_i – paarweise verschieden – zu wählen, damit $\|w\|$ minimal wird? Dabei beschränken wir uns zunächst auf das Intervall $[-1, 1]$.

$w(t)$ ist ein Polynom vom Grad $m + 1$ mit Hauptkoeffizient $= 1$, d.h.

$$w(t) \;=\; t^{m+1} + a_m t^m + \cdots + a_1 t + a_0 \quad . \tag{30}$$

Gesucht ist also ein Polynom vom Grad $m + 1$ mit Hauptkoeffizient $= 1$ und $m + 1$ einfachen reellen Nullstellen in $[-1, 1]$, welches eine minimale Norm hat (unter allen Polynomen der Gestalt (30)), falls ein solches überhaupt existiert.

Hierzu benötigen wir die sogenannten Tschebyscheff-Polynome.

4.5 Tschebyscheff-Polynome

Definition der Tschebyscheff-Polynome
Zu jedem $n \in \mathbb{N}$ definiert man die Funktion

$$T_n \,:\, [-1,1] \to \mathbb{R} \;:\; t \mapsto T_n(t) := \cos\left(n \arccos(t)\right) \quad . \tag{31}$$

und zeigt, dass T_n ein Polynom vom Grad n ist.

Lemma: Für $n \geq 1$ gilt

$$T_{n+1}(t) \;=\; 2t \cdot T_n(t) \;-\; T_{n-1}(t) \quad . \tag{32}$$

Beweis: Die Additionstheoreme für cos liefern:

$$\begin{aligned} T_{n+1}(t) &= \cos\left((n+1)\arccos(t)\right) \\ &= \cos\left(n\arccos(t)\right)\cdot\cos\left(\arccos(t)\right) \;-\; \sin\left(n\arccos(t)\right)\cdot\sin\left(\arccos(t)\right) \;, \\ T_{n-1}(t) &= \cos(n-1)\arccos t \\ &= \cos\left(n\arccos(t)\right)\cdot\cos\left(\arccos(t)\right) \;+\; \sin\left(n\arccos(t)\right)\cdot\sin\left(\arccos(t)\right) \;. \end{aligned}$$

Daraus ergibt sich

$$T_{n+1}(t) + T_{n-1}(t) \;=\; 2\cdot\cos\left(n\arccos(t)\right)\cdot\cos\left(\arccos(t)\right) \;=\; 2t\cdot T_n(t)\;. \quad \square$$

Damit können wir nun folgenden Satz beweisen:

Satz: T_n ist ein Polynom vom Grad n. Für $n \geq 1$ hat T_n den Hauptkoeffizient 2^{n-1} .

Beweis: (durch vollständige Induktion)
Aus (31) folgt sofort

$$T_0(t) \equiv 1\;, \quad T_1(t) = t\;.$$

Also ist die Behauptung für den Induktionsanfang erfüllt.

Sei nun die Behauptung richtig für alle $k \leq n$, d.h. T_k ist ein Polynom vom Grad k und

$$T_k(t) \;=\; 2^{k-1}t^k + c_{k-1}^{(k)}t^{k-1} + \cdots + c_1^{(k)}t + c_0^{(k)} \quad .$$

Mit Hilfe des obigen Lemmas folgt

$$\begin{aligned} T_{n+1}(t) &= 2t\cdot\left(2^{n-1}t^n + c_{n-1}^{(n)}t^{n-1} + \cdots + c_0^{(n)}\right) - \left(2^{n-2}t^{n-1} + c_{n-2}^{(n-1)}t^{n-2} + \cdots + c_0^{(n-1)}\right) \\ &= 2^n t^{n+1} + c_n^{(n+1)}t^n + \cdots + c_o^{(n+1)} \quad . \qquad \square \end{aligned}$$

Definition: Die Polynome T_n bezeichnet man als *Tschebyscheff-Polynome 1. Art.*
(P. L. Tschebyscheff, 1821 - 1894, russischer Mathematiker)

Beispiele:
$$\begin{aligned} T_2(t) &= 2t^2 - 1\;, \\ T_3(t) &= 4t^3 - 3t\;, \\ T_4(t) &= 8t^4 - 8t^2 + 1\;. \end{aligned}$$

Eigenschaften

Wir stellen einige wichtige Eigenschaften der Tschebyscheff-Polynome zusammen:

1. Symmetrie: Es gilt
$$T_n(-t) \;=\; (-1)^n T_n(t)$$
(d.h. Achsensymmetrie für gerades und Punktsymmetrie für ungerades n).

2. Aus (31) folgt sofort
$$\begin{aligned} |T_n(t)| &\leq 1 \quad \text{für alle } t \in [-1,1]\,, \\ T_n(1) &= 1\,, \\ T_n(-1) &= (-1)^n\,, \\ \|T_n\|_{[-1,1]} &= 1\,. \end{aligned}$$

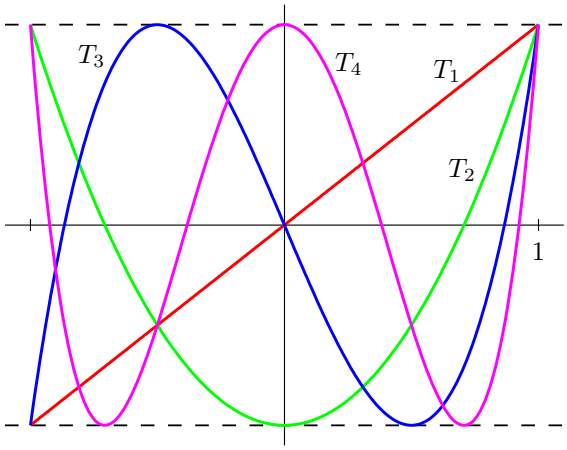
3. Die Tschebyscheff-Polynome $T_0, T_1, \ldots, T_n$ bilden eine Basis des Polynomraumes Π_n; d.h. jedes $q \in \Pi_n$ besitzt eine eindeutige Darstellung der Form
$$q(t) \;=\; b_n T_n(t) + b_{n-1} T_{n-1}(t) + \cdots + b_1 T_1(t) + b_0 T_0(t)$$
mit $b_i \in \mathbb{R},\ i = 0, 1, \ldots, n$.

   Beispiele:
$$\begin{aligned} t^2 &= \tfrac{1}{2}\left(T_0(t) + T_2(t)\right)\,, \\ t^3 &= \tfrac{1}{4}\left(3\,T_1(t) + T_3(t)\right)\,, \\ t^4 &= \tfrac{1}{8}\left(3\,T_0(t) + 4\,T_2(t) + T_4(t)\right)\,. \end{aligned}$$

4. Nullstellen: T_n hat n einfache reelle Nullstellen im offenen Intervall $(-1, 1)$. Diese sind gegeben durch
$$t_i \;=\; \cos\left(\frac{2i+1}{2n}\,\pi\right) \qquad (i = 0, 1, \ldots, n-1); \tag{33}$$
denn aus (31) folgt sofort
$$\begin{aligned} T_n(t_i) &= \cos\left(n \arccos\left(t_i\right)\right) \;=\; \cos\left(n \arccos\left(\cos\left(\frac{2i+1}{2n}\,\pi\right)\right)\right) \\ &= \cos\left(n \cdot \frac{2i+1}{2n} \cdot \pi\right) \;=\; \cos\left((2i+1)\,\frac{\pi}{2}\right) \;=\; 0 \quad . \end{aligned}$$

5. Extremalstellen: Im Intervall $[-1, 1]$ hat T_n die $n+1$ Extremalstellen
$$z_i \;=\; \cos\left(\frac{i}{n}\pi\right) \qquad (i = 0, 1, 2, \ldots, n). \tag{34}$$
Ferner gilt
$$T_n(z_i) = (-1)^i \qquad (i = 0, 1, 2, \ldots, n).$$
Alle Extremalwerte sind also betragsmäßig gleich.

Mit Hilfe der obigen Eigenschaften können wir nun die Tschebyscheff-Polynome skizzieren.



Minimalitätseigenschaft

Für $n \geq 1$ sei Q_n die Menge aller Polynome vom Grad n mit Hauptkoeffizient 1, also

$$Q_n \;:=\; \{q \in \Pi_n \;:\; q(t) = t^n + a_{n-1}t^{n-1} + \cdots + a_1 t^1 + a_0\,;\; a_i \in \mathbb{R}\}\;.$$

Setzt man $\hat{T}_n \;:=\; \dfrac{T_n}{2^{n-1}}$, so gilt $\hat{T}_n \in Q_n$.

Satz: Unter allen Polynomen aus Q_n $(n \geq 1)$ hat $\hat{T}_n$ die kleinste Maximum-Norm im Intervall $[-1, 1]$, d.h. es gilt

$$\min\{\|q\| \;:\; q \in Q_n\} \;=\; \|\hat{T}_n\| \;=\; \frac{1}{2^{n-1}} \quad . \tag{35}$$

Beweis: Nach (34) gilt für die Extremalstellen z_i von T_n

$$\hat{T}_n(z_i) \;=\; \frac{(-1)^i}{2^{n-1}} \qquad (i = 0, 1, 2, \ldots, n).$$

Annahme: es gibt ein $\hat{q} \in Q_n$ mit $\|\hat{q}\| \;<\; \|\hat{T}_n\| \;=\; \dfrac{1}{2^{n-1}}$. Das Differenzpolynom

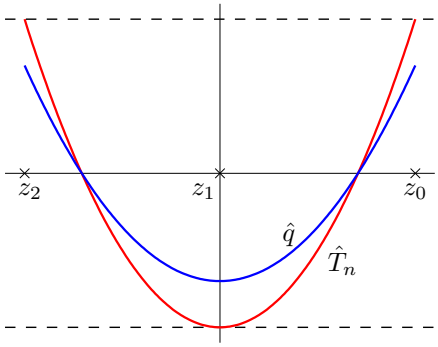
$$r(t) \;:=\; \hat{T}_n(t) - \hat{q}(t)$$

erfüllt $r \in \Pi_{n-1} \setminus \{0\}$, folglich hat r höchstens $n-1$ Nullstellen.
Es gelten folgende Ungleichungen:

$$\begin{aligned}
r(z_0) \;&=\; \hat{T}_n(z_0) - \hat{q}(z_0) \;=\; \frac{1}{2^{n-1}} - \hat{q}(z_0) \;>\; 0\;,\\
r(z_1) \;&=\; \hat{T}_n(z_1) - \hat{q}(z_1) \;=\; -\frac{1}{2^{n-1}} - \hat{q}(z_1) \;<\; 0\;,\\
r(z_2) \;&=\; \hat{T}_n(z_2) - \hat{q}(z_2) \;=\; \frac{1}{2^{n-1}} - \hat{q}(z_2) \;>\; 0\;,\\
&\;\;\vdots\\
r(z_n) \;&=\; \hat{T}_n(z_n) - \hat{q}_n(z_n) \;\begin{cases} > \;0 & \text{für } n \text{ gerade} \\ < \;0 & \text{für } n \text{ ungerade} \end{cases} \quad .
\end{aligned}$$

Nach dem Zwischenwertsatz hat somit r im IV $[-1, 1]$ mindestens n verschiedene Nullstellen. Dies ist ein Widerspruch. □

Wir veranschaulichen die obige Schlussweise für den Fall $n = 2$.



Nullstellen als Stützstellen bei der Interpolation

Mit Hilfe des obigen Satzes können wir nun die Frage nach einer günstigen Stützstellenwahl bei der Interpolation beantworten.

Wähle als Stützstellen bei der Interpolation die Nullstellen des Tschebyscheff-Polynoms T_{m+1}, also

$$t_i := \cos\left(\frac{2i+1}{2m+2}\,\pi\right) \qquad (i = 0, 1, \ldots, m).$$

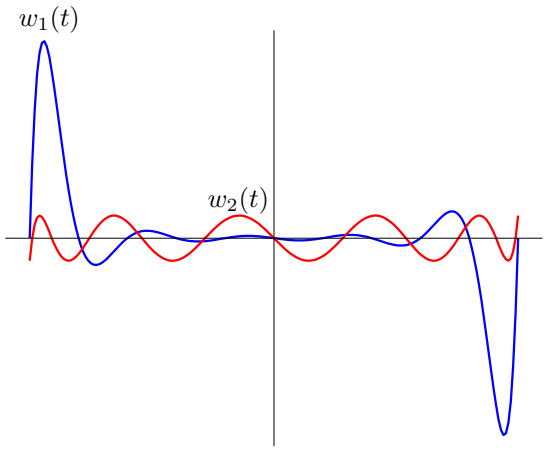
Dann gilt

$$w(t) = \hat{T}_{m+1}(t) \quad \text{und somit} \quad \|w\| = \frac{1}{2^m} \quad .$$

Aus (29) folgt für den Interpolationsfehler

$$\|f - p\| \leq \frac{\left\|f^{(m+1)}\right\|}{2^m (m+1)!} \quad . \tag{36}$$

Das folgende Schaubild zeigt die Knotenpolynome $w_1(t)$ (äquidistante Stützstellen) und $w_2(t)$ (Tschebyscheff-Nullstellen).



Die Aussage (36) gilt für das Intervall $[-1,1]$. Mittels affiner Transformation erhält man Fehlerschranken für das Intervall $[a,b]$.

Es sei $g : [a,b] \to \mathbb{R}$. Mit der affinen Transformation

$$\varphi \ : \ [-1,1] \to [a,b] \ : \ \varphi(t) := \frac{b-a}{2} \cdot t \ + \ \frac{a+b}{2}$$

ist $f(t) := g(\varphi(t))$ eine Funktion $f \ : \ [-1,1] \to \mathbb{R}$. Es gilt

$$f^{(m+1)}(t) \ = \ \left(\frac{b-a}{2} \right)^{m+1} \cdot g^{(m+1)}(\varphi(t)) \quad .$$

Für das Interpolationspolynom q bezüglich der Stützstellen $x_i := \varphi(t_i)$, $i = 0, \ldots, m$, erhält man

$$\|g - q\|_{[a,b]} \ \leq \ \frac{(b-a)^{m+1} \cdot \left\| g^{(m+1)} \right\|_{[a,b]}}{2^{2m+1}(m+1)!} \quad .$$

4.6 Konvergenzfragen

Es sei $f \in C[a,b]$. Wir betrachten zu jedem $m \in \mathbb{N}$ Stützstellen $t_{m0}, \ldots, t_{mm} \in [a,b]$ (paarweise verschieden) und bilden das Interpolationspolynom $p_m(t)$ zu $f(t)$. Dann interessieren folgende Fragen:

- Gilt $\lim\limits_{m \to \infty} \|f - p_m\| \ = \ 0$ (gleichmäßige Konvergenz)?
- Gilt $\lim\limits_{m \to \infty} p_m(t) \ = \ f(t)$ für alle $t \in [a,b]$ (punktweise Konvergenz)?

Die obigen Stützstellen werden üblicherweise in einer (unendlichen) Matrix angeordnet:

$$S \ = \ \begin{pmatrix} t_{00} & & & & & \\ t_{10} & t_{11} & & & & \\ t_{20} & t_{21} & t_{22} & & & \\ \vdots & \vdots & \vdots & \ddots & & \\ t_{m0} & t_{m1} & t_{m2} & \cdots & t_{mm} & \\ \vdots & \vdots & \vdots & \cdots & \vdots & \ddots \end{pmatrix} \quad .$$

Diese Matrix nennt man eine *Stützstellenmatrix*.

Satz (Faber): Zu jeder vorgegebenen Stützstellenmatrix S gibt es eine Funktion $f \in C[a,b]$, so dass die Folge $(p_m)_{m \in \mathbb{N}}$ der Interpolationspolynome nicht gleichmäßig gegen f konvergiert.

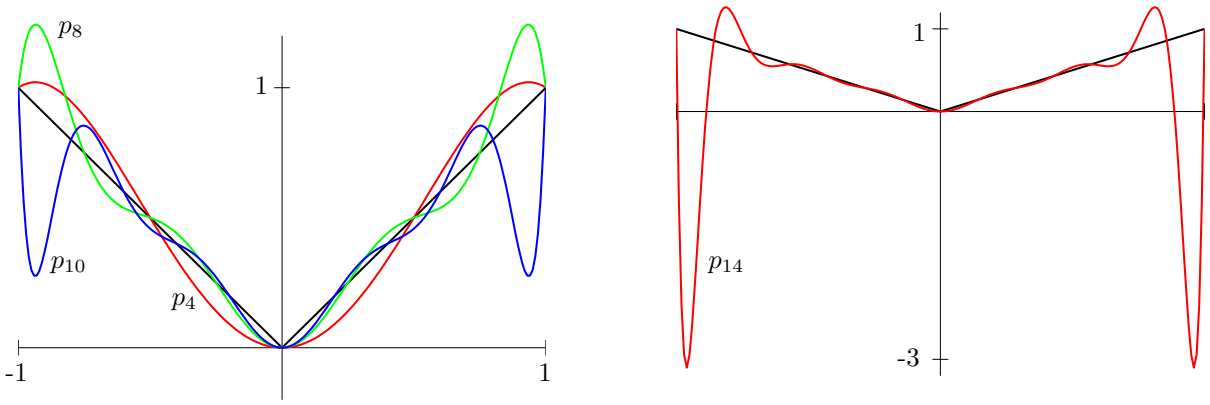
Satz (Marcinkiewicz): Zu jeder stetigen Funktion $f \in C[a,b]$ gibt es eine Stützstellenmatrix S, so dass die Folge $(p_m)_{m \in \mathbb{N}}$ der Interpolationspolynome gleichmäßig gegen f konvergiert.

Anmerkung: Der Satz von Marcinkiewicz liefert zu vorgegebenem f nur die Existenz einer solchen Stützstellenmatrix, die Stützstellen sind i.A. nicht explizit bekannt.

Beispiel 1: Sei $[a,b] = [-1,1]$ und $f(t) := |t|$. Bei äquidistanter Stützstellenwahl, d.h.

$$t_{mi} \;:=\; -1 + \frac{2i}{m} \qquad (i = 0,1,\ldots,m;\; m \geq 1)$$

divigiert die Folge $(p_m(t))_{m\in\mathbb{N}}$ für alle t mit $0 < |t| < 1$.
In den beiden folgenden Schaubildern sind einige dieser Interpolationspolynome dargestellt:



Beispiel 2: Seien $f(t) := e^t$ und $a, b \in \mathbb{R}$ mit $a < b$. Für jede beliebige Stützstellenmatrix S konvergiert die Folge der Interpolationspolynome $(p_m)_{m\in\mathbb{N}}$ im Intervall $[a,b]$ gleichmäßig gegen f. Denn aus (29) folgt

$$\|f - p_m\| \;\leq\; \frac{e^b}{(m+1)!}\cdot(b-a)^{m+1}$$

und somit

$$\lim_{m\to\infty} \|f - p_m\| \;=\; e^b \cdot \lim_{m\to\infty} \left(\frac{(b-a)^{m+1}}{(m+1)!}\right) \;=\; 0 \qquad .$$

4.7 Numerische Differentiation

Es seien $f \in C^1[a,b]$ und $\tau \in (a,b)$ fest. Als Anwendung der Interpolation leiten wir nun Formeln zur numerischen Bestimmung der Ableitung $f'(\tau)$ her und setzen

$$g(h) \;:=\; g_{\tau,f}(h) \;:=\; \frac{f(\tau+h) - f(\tau-h)}{2h}$$

$(h \neq 0,\; \tau \pm h \in [a,b])$. Dann gilt $g(-h) = g(h)$ und (nach Definition der Differenzierbarkeit)

$$\lim_{h\to 0} g(h) = f'(\tau) \qquad .$$

Beachten Sie, dass g nur in einer Umgebung der Null definiert ist.
Die Funktion

$$\varphi(h) \;:=\; \begin{cases} g(h) & \text{für } h \neq 0 \\ f'(\tau) & \text{für } h = 0 \end{cases} \qquad (h \text{ hinreichend klein}) \tag{37}$$

ist stetig, erfüllt $\varphi(-h) = \varphi(h)$, und $\varphi(0) = f'(\tau)$ ist der gesuchte Wert.

Sei $q \in \Pi_{2m+1}$ das Interpolationspolynom zu den Stützpaaren

$$(h_i, \varphi(h_i)),\ (-h_i, \varphi(h_i))\,, \quad i = 0, 1, \ldots m\ .$$

Dann ist $q(0)$ ein Näherungswert für $f'(\tau)$.

Die Berechnung von $q(0)$ erfolgt mit Hilfe des Neville-Algorithmus unter Verwendung des folgenden Lemmas.

Lemma: Es seien $\varphi : [-\alpha, \alpha] \to \mathbb{R}$ stetig mit $\varphi(t) = \varphi(-t)$ für alle $t \in [-\alpha, \alpha]$ und $0 < h_0 < h_1 < \cdots < h_m < \alpha$. Es bezeichne $q(t)$ das Interpolationspolynom zu den $2m+2$ Paaren

$$(\pm h_i, \varphi(h_i)) \qquad (i = 0, 1, \ldots, m)$$

und $p(t)$ das Interpolationspolynom zu den $m + 1$ Paaren

$$(h_i^2, \varphi(h_i)) \qquad (i = 0, 1, \ldots, m). \tag{38}$$

Dann gilt $q(t) = p(t^2)$.

Beweis:
Es gilt $p(t) \in \Pi_m$ und folglich $p(t^2) \in \Pi_{2m}$; weiter ist $q(t) \in \Pi_{2m+1}$. Man erhält

$$\begin{aligned} p((+h_i)^2) &= \varphi(h_i) = q(h_i) \quad (i = 0, \ldots, m)\ , \\ p((-h_i)^2) &= p(h_i^2) = \varphi(h_i) = \varphi(-h_i) = q(-h_i) \quad (i = 0, \ldots, m)\ . \end{aligned}$$

Somit ist $p(t^2)$ das Interpolationspolynom zu den Paaren

$$(h_0, \varphi(h_0)), \ldots, (h_m, \varphi(h_m)), (-h_0, \varphi(-h_0)), \ldots, (-h_m, \varphi(-h_m))\ .$$

Aus der Eindeutigkeit der Interpolation folgt dann $q(t) = p(t^2)$. $\square$

Anmerkung: Mit diesem Lemma erhalten wir insbesondere

$$f'(\tau) = \varphi(0) \approx q(0) = p(0)\ .$$

Deshalb verwendet man im Neville-Algorithmus vorteilhaft die Paare (38), wodurch sich der Aufwand reduziert.

Beispiel 1: Es sei $m := 0,\ h_0 := h$ ($h > 0$ fest). Dann ergibt sich sofort

$$f'(\tau) \approx p(0) = \frac{f(\tau + h) - f(\tau - h)}{2h} \quad .$$

Beispiel 2: Es sei $m := 1,\ h_0 := h,\ h_1 := 2h$ ($h > 0$ fest). Das Schema von Neville liefert dann (mit $\xi = 0$)

$h_0^2 = h^2$	$p_0(0) = \varphi(h) = \frac{f(\tau+h) - f(\tau-h)}{2h}$	
		$p_{01}(0) = \frac{(0-4h^2)p_0(0) - (0-h^2)p_1(0)}{h^2 - 4h^2}$
$h_1^2 = 4h^2$	$p_1(0) = \varphi(2h) = \frac{f(\tau+2h) - f(\tau-2h)}{4h}$	

Somit

$$f'(\tau) \approx p_{01}(0) = \frac{1}{12h}\Big(f(\tau-2h) - 8f(\tau-h) + 8f(\tau+h) - f(\tau+2h)\Big) .$$

4.8 Inverse Interpolation

Sei $f : [a,b] \to \mathbb{R}$ eine reelle Funktion, deren Umkehrabbildung f^{-1} existiert. Ferner seien $a \le t_0 < t_1 < \cdots < t_m \le b$. Bildet man nun das Interpolationspolynom $q(t) \in \Pi_m$ zu den Paaren

$$(f(t_i), t_i) \qquad (i = 0, 1, \ldots, m),$$

so ist $q(t)$ eine Näherung an $f^{-1}(t)$.

Anmerkung: Hier wurden Stützstellen und Stützwerte vertauscht. Da nach Voraussetzung f^{-1} existiert, sind auch die $f(t_i)$ paarweise verschieden. Damit ist $q(t)$ eindeutig bestimmt (vgl. Abschnitt 4.2 über Existenz und Eindeutigkeit). Zur besseren Unterscheidung bezeichnen wir dieses Interpolationspolynom mit $q(t)$.

Definition: Diese Vorgehensweise heißt *inverse Interpolation*.

Anwendung: Man kann die inverse Interpolation benutzen, um näherungsweise eine Nullstelle ζ der Funktion $f(t)$ zu bestimmen. $f(\zeta) = 0$ ist äquivalent zu $\zeta = f^{-1}(0)$; folglich ist

$$\zeta \approx q(0) \quad .$$

Die Durchführung erfolgt wieder mit dem Algorithmus von Neville und führt auf folgendes Schema:

	$k=0$	$k=1$	$k=2$	$\cdots$	$k=m$
$f(t_0)$	$t_0 = q_0(0)$				
		$q_{01}(0)$			
$f(t_1)$	$t_1 = q_1(0)$		$q_{012}(0)$		
		$q_{12}(0)$		$\ddots$	
$f(t_2)$	$t_2 = q_2(0)$		$\vdots$		$q_{012\cdots m}(0)$
		$\vdots$		$\cdots$	
$\vdots$	$\vdots$		$q_{m-2,m-1,m}(0)$		
		$q_{m-1,m}(0)$			
$f(t_m)$	$t_m = q_m(0)$				

Aus (27) erhält man hier die Rekursionsformel

$$q_{i_0 i_1 \cdots i_k}(0) = \frac{f(t_{i_0}) q_{i_1 \cdots i_k}(0) - f(t_{i_k}) q_{i_0 \cdots i_{k-1}}(0)}{f(t_{i_0}) - f(t_{i_k})} \quad .$$

Dann ist $q_{012\cdots m}(0)$ eine Näherung an die gesuchte Nullstelle ζ. Um diese Näherung noch zu verbessern, bietet sich folgende Vorgehensweise an: Setze

$$t_{m+1} := q_{012\cdots m}(0)$$

und bilde das IP-Polynom $q_{012\cdots m+1}$ zu den Paaren

$$(f(t_0), t_0),\ \ldots\ , (f(t_m), t_m),\ (f(t_{m+1}), t_{m+1}) \quad .$$

Häufig ist dann $q_{012\cdots m+1}(0)$ eine bessere Näherung für die gesuchte Nullstelle. Im Neville-Schema bedeutet dies, dass man eine weitere Zeile hinzufügt:

	$k=0$	$k=1$	$k=2$	$\cdots$	$k=m$	$k=m+1$
$f(t_0)$	$t_0 = q_0(0)$					
		$q_{01}(0)$				
$f(t_1)$	$t_1 = q_1(0)$		$q_{012}(0)$			
		$q_{12}(0)$		$\ddots$		
$f(t_2)$	$t_2 = q_2(0)$		$\vdots$		$q_{012\cdots m}(0) =: t_{m+1}$	
		$\vdots$		$\cdots$		$q_{012\cdots m+1}(0)$
$\vdots$	$\vdots$		$q_{m-2,m-1,m}(0)$		$q_{123\cdots m+1}(0)$	
		$q_{m-1,m}(0)$		$\cdots$		
$f(t_m)$	$t_m = q_m(0)$		$q_{m-1,m,m+1}(0)$			
		$q_{m,m+1}(0)$				
$f(t_{m+1})$	$t_{m+1} = q_{m+1}(0)$					

Dieser Vorgang lässt sich iterativ fortsetzen (*iterative inverse Interpolation*).

4.9 Weitere Interpolationsarten

Wir geben einen Überblick über einige andere wichtige Interpolationsarten.

Hermite-Interpolation

Hier werden auch Bedingungen für die Ableitungen des IP-Polynoms an den Stützstellen gefordert. Im einfachsten Fall (nur 1. Ableitung) lautet die Interpolationsaufgabe dann: Gegeben seien Tripel $(t_i, s_i^{(0)}, s_i^{(1)}) \in \mathbb{R}^3$ $(i = 0, 1, \ldots, m)$; die t_i seien wieder paarweise verschieden. Gesucht ist ein Polynom $p \in \Pi_{2m+1}$ mit

$$p(t_i) \ = \ s_i^{(0)} \quad \text{und} \quad p'(t_i) \ = \ s_i^{(1)} \quad (i = 0, 1, \ldots, m).$$

Diese Problemstellung wird als *Interpolationsaufgabe nach Hermite* bezeichnet.

Satz: Es existiert genau ein solches Polynom.

Der Beweis der Eindeutigkeit verläuft ähnlich wie bei der Lagrange-Interpolation. Bei der Existenz zeigt man, dass das Polynom

$$p(t) \ := \ \sum_{k=0}^{m} s_k^{(0)}\, l_k^2(t) \left(1 - 2l_k'(t_k)\,(t - t_k)\right) \ + \ \sum_{k=0}^{m} s_k^{(1)}\, l_k^2(t)\,(t - t_k)$$

diese Interpolationsaufgabe erfüllt (die $l_k(t)$ bezeichnen wieder die Lagrange-Grundpolynome zu den Stützstellen $t_0, \ldots, t_m$).

$p(t)$ heißt *Hermite-Interpolationspolynom.*

Anwendung: Sei $f \in C^1[a,b],\ a \le t_0 < \cdots < t_m \le b$. Für die Stützwerte wählt man dann

$$s_i^{(0)} := f(t_i)\,, \quad s_i^{(1)} := f'(t_i) \qquad (i = 0, \ldots, m).$$

(*Hermite-Interpolationspolynom* zu f bezüglich der Stützstellen $t_0, \ldots, t_m$).

Hier ergibt sich für den Interpolationsfehler folgende Darstellung:

Satz: Sei $f \in C^{2m+2}[a,b]$ und $z \in [a,b]$. Dann gilt

$$f(z) - p(z) \;=\; \frac{f^{(2m+2)}(\xi)}{(2m+2)!} \prod_{i=0}^{m} (z - t_i)^2$$

mit einem $\xi \in (a,b)$.

Beweis: Übung.

Anmerkungen:

1. Eine Verallgemeinerung ergibt sich durch Berücksichtigung höherer Ableitungen, eventuell sogar unterschiedlich vieler in den einzelnen Stützstellen (*verallgemeinerte Hermite-Interpolation*).

2. Das Taylor-Polynom kann als verallgemeinertes Hermite-Interpolationspolynom mit einer Stützstelle aufgefasst werden.

Rationale Interpolation:

Gegeben seien $n+m+1$ Paare $(t_i, s_i) \in \mathbb{R}^2$, die t_i wieder paarweise verschieden. Gesucht ist eine rationale Funktion

$$R_{m,n}(t) \;:=\; \frac{a_0 + a_1 t + \cdots + a_m t^m}{b_0 + b_1 t + \cdots + b_n t^n}$$

mit

$$R_{m,n}(t_i) \;=\; s_i \qquad (i = 0, 1, 2, \ldots m+n).$$

Im Gegensatz zur Polynom-Interpolation liegen hier kompliziertere Verhältnisse vor. So ist z.B. die Existenz nicht immer gesichert (vgl. Stoer [7]).

Trigonometrische Interpolation:

Gegeben seien $2m+1$ Paare $(t_i, s_i) \in \mathbb{R}^2$ mit $-\pi \le t_0 < \ldots < t_{2m} < \pi$. Gesucht ist ein trigonometrisches Polynom

$$S_m(t) \;:=\; a_0 + \sum_{j=1}^{m} (a_j \cos jt \;+\; b_j \sin jt)$$

mit der Eigenschaft

$$S_m(t_i) \;=\; s_i \qquad (i = 0, 1, 2, \ldots, 2m).$$

Diese Interpolationsaufgabe besitzt eine eindeutige Lösung. S_m ist eine 2π-periodische Funktion. Diese Interpolationsart ist deshalb bei periodischen Funktionen $f : \mathbb{R} \to \mathbb{R}$ angebracht.

Spline-Interpolation:
Gegeben seien $m+1$ Paare $(t_i, s_i) \in \mathbb{R}^2$ mit $a = t_0 < \ldots < t_m = b$.

Definition: Unter einer *(kubischen) Spline-Funktion* $h(t)$ versteht man eine reelle Funktion $h : [a,b] \to \mathbb{R}$ mit folgenden Eigenschaften:

a) $h \in C^2[a,b]$,

b) auf jedem Teil-Intervall $[t_i, t_{i+1}]$ stimmt h mit einem Polynom vom Grad ≤ 3 überein.

Bei der *Spline-Interpolation* ist ein kubischer Spline h gesucht mit der Eigenschaft

$$h(t_i) \;=\; s_i \qquad (i = 0, 1, \ldots, m).$$

Diese Interpolations-Aufgabe besitzt stets eine Lösung; sie ist jedoch nicht eindeutig. Eindeutigkeit erreicht man z.B. durch die Zusatzforderung

$$h^{(2)}(t_0) \;=\; h^{(2)}(t_m) \;=\; 0 \qquad .$$

Dann heißt h ein *natürlicher kubischer Spline*.

Zur Bestimmung des natürlichen kubischen Splines machen wir für jedes Teilintervall $[t_i, t_{i+1}]$ den Ansatz

$$h_i(t) \;=\; a_i(t-t_i)^3 \,+\, b_i(t-t_i)^2 \,+\, c_i(t-t_i) \,+\, d_i \quad (i = 0, \ldots, m-1)\,. \tag{39}$$

Wegen $h(t_i) = s_i$ lassen sich die d_i sofort angeben:

$$d_i \;=\; s_i \quad (i = 0, \ldots, m-1)\,.$$

Für die $3m$ Unbekannten a_i, b_i, c_i $(i = 0, m-1)$ liefern die $3m$ Bestimmungsgleichungen

$$\begin{aligned}
h_i(t_{i+1}) &= s_{i+1} && (i = 0, \ldots, m-1)\,,\\
h_i'(t_{i+1}) &= h_{i+1}'(t_{i+1}) && (i = 0, \ldots, m-2)\,,\\
h_i^{(2)}(t_{i+1}) &= h_{i+1}^{(2)}(t_{i+1}) && (i = 0, \ldots, m-2)\,,\\
h_0^{(2)}(t_0) &= h_{m-1}^{(2)}(t_m) \;=\; 0
\end{aligned}$$

ein lineares Gleichungssystem, welches sich unter den gemachten Annahmen eindeutig lösen lässt (vgl. Übungen).

5 Nullstellenbestimmung

In diesem Paragraphen befassen wir uns mit der Frage der näherungsweisen Berechnung einer Nullstelle von $f \in C[a,b]$ (d.h. gesucht ist ein Wert $\xi \in [a,b]$ mit $f(\xi) = 0$).

5.1 Das Bisektionsverfahren

Es sei $f \in C[a,b]$ mit

$$f(a) \cdot f(b) \; < \; 0 \quad .$$

Nach dem Zwischenwertsatz (vgl. Analysis I) existiert dann ein $\xi \in (a,b)$ mit

$$f(\xi) \; = \; 0 \quad .$$

Die Existenz einer Lösung ist also gesichert.

Intervallhalbierung

Hier liegt die Idee zugrunde, durch Intervallhalbierung eine Nullstelle genauer zu lokalisieren. Das Ausgangsintervall sei $I_0 = [a,b] = [a_0, b_0]$. Man bildet die Intervallmitte $s := \frac{a+b}{2}$ und betrachtet dann $f(s)$. Dabei tritt genau einer der folgenden drei Fälle auf:

(a) $f(s) = 0$: dann ist s eine gesuchte Nullstelle ξ ,

(b) $f(a) \cdot f(s) < 0$: dann liegt eine Nullstelle ξ im Intervall $I_1 = [a_1, b_1] = \left[a, \frac{a+b}{2}\right]$,

(c) $f(s) \cdot f(b) < 0$: dann liegt eine Nullstelle ξ im Intervall $I_1 = [a_1, b_1] = \left[\frac{a+b}{2}, b\right]$.

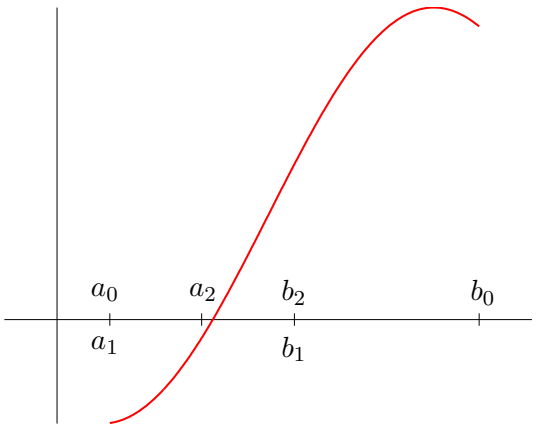
Wir haben somit das Intervall halbiert, in dem eine Nullstelle liegt. Die Methode der Intervallhalbierung lässt sich nun mehrfach anwenden. Wählt man nach dem n-ten Schritt ein u_n aus dem verbleibenden Intervall I_n als Näherung, so gilt

$$|\xi - u_n| \; \leq \frac{b-a}{2^n} \quad .$$

Durch fortgesetzte Intervallhalbierung kann eine Nullstelle von f beliebig genau berechnet werden.

Definition: Diese Methode heißt *Bisektionsverfahren*.

Bisektionsverfahren



5.2 Das Newton-Verfahren

Herleitung

Es sei $f \in C^2[a,b]$ und $\xi \in (a,b)$ eine Nullstelle von f mit $f'(\xi) \neq 0$ (aufgrund der Stetigkeit gilt dann $f'(t) \neq 0$ in einer Umgebung von ξ). Diese Nullstelle ξ sei nicht explizit bekannt. Zur Verfügung stehe dagegen ein Näherungswert x_0 mit $f'(x_0) \neq 0$.

Ziel: Bestimmung einer besseren Näherung x_1 an die gesuchte Nullstelle.
Dazu betrachten wir die Taylorentwicklung um x_0

$$0 \;=\; f(\xi) \;=\; f(x_0) + f'(x_0)(\xi - x_0) + \frac{f^{(2)}(\eta)}{2!}(\xi - x_0)^2 \tag{40}$$

mit einem Zwischenwert η.

Linearisierung liefert

$$0 \;=\; f(\xi) \;\approx\; f(x_0) + f'(x_0)(\xi - x_0)$$

(Begründung: Falls x_0 schon nahe bei ξ liegt, so kann der Rest in (40) vernachlässigt werden.)
Daraus folgt

$$\xi \;\approx x_0 - \frac{f(x_0)}{f'(x_0)} \;=:\; x_1 \quad .$$

Im nächsten Schritt sucht man ausgehend von x_1 eine neue (hoffentlich bessere) Näherung x_2, usw.

Diese Überlegungen motivieren folgendes Verfahren zur Nullstellenbestimmung (für ein $f \in C^2[a,b]$ mit $f'(t) \neq 0$ in $[a,b]$) :
Wähle einen geeigneten Startwert $x_0 \in [a,b]$ und definiere neue Werte durch die Vorschrift

$$x_{i+1} \;:=\; x_i - \frac{f(x_i)}{f'(x_i)} \qquad (i = 0, 1, 2, \ldots,). \tag{41}$$

Definition: Die obige Methode heißt *Newton-Verfahren.*

Für die Praxis treten sofort einige Fragen auf:

- Ist die Iteration überhaupt durchführbar, d.h. gilt $x_{i+1} \in [a,b]$?
- Konvergenz gegen eine Nullstelle?
- Wenn Konvergenz vorliegt, wie rasch ist sie?
- Wie wählt man den Startwert x_0 geschickt?

Auf diese Fragen gehen wir im folgenden Abschnitt ein.

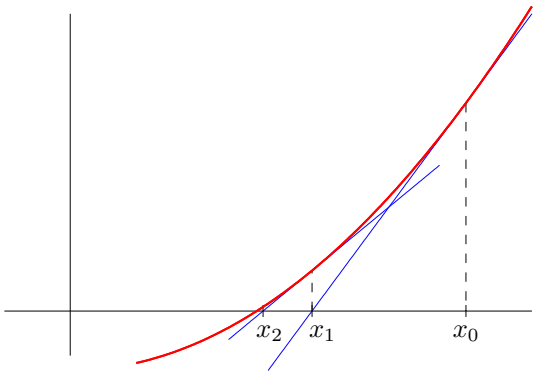
Anmerkung: Allgemein heißt ein Verfahren, bei dem ein Startwert x_0 vorgegeben und dann $x_{i+1} = \Phi(x_i)$ gesetzt wird (mit einer gewissen Funktion Φ), ein *Iterationsverfahren.* Φ wird als *Iterationsfunktion* bezeichnet. Hier interessiert man sich für Fixpunkte. Dabei heißt ein $\xi \in [a,b]$ *Fixpunkt*, falls $\Phi(\xi) = \xi$ gilt.

Beim Newton-Verfahren haben wir

$$\Phi(x) \;=\; x - \frac{f(x)}{f'(x)} \quad .$$

Graphische Darstellung
Ist x_n schon berechnet, so erhält man x_{n+1} auf folgende Weise: es wird die Tangente an f im Punkt x_n bestimmt; der Schnittpunkt dieser Tangente mit der x-Achse ergibt dann x_{n+1}.

Newton-Verfahren



5.3 Konvergenz des Newton-Verfahrens

Definition: (lokale und globale Konvergenz)
Es sei $\Phi : [a,b] \to \mathbb{R}$ eine Iterationsfunktion und $\xi \in [a,b]$ ein Fixpunkt. Konvergiert die Iterationsfolge nur für Anfangswerte x_0 aus einer Umgebung $U \subset [a,b]$ des Fixpunktes ξ, so heißt die Iteration *lokal konvergent*. Konvergiert die Iteration für jeden Startwert aus $[a,b]$, so heißt das Verfahren *global konvergent*.

Für die Numerik ist neben der Konvergenz die Frage nach der Konvergenzgeschwindigkeit von großer Bedeutung. Bei rascher Konvergenz genügen wenige Schritte für eine gute Näherung an die gesuchte Nullstelle.

Definition: (Konvergenzordnung)
Es seien ξ ein Fixpunkt von Φ und U eine Umgebung von ξ. Erfüllen für jeden Startwert $x_0 \in U$ die Iterierten $x_{i+1} = \Phi(x_i)$ eine Ungleichung der Form

$$|x_{i+1} - \xi| \;\leq\; C\,|x_i - \xi|^p \qquad \text{für alle } i \geq 0$$

mit $C \in \mathbb{R},\ p \geq 1,\ p \in \mathbb{R}$ (für $p = 1$ gelte zusätzlich $C < 1$), so heißt die von Φ erzeugte Iteration ein Verfahren der *Konvergenzordnung* p.

Der Fall $p = 1$ wird als *lineare* und der Fall $p = 2$ als *quadratische* Konvergenz bezeichnet.

Satz: Es seien $f \in C^2[a,b]$, $\xi \in [a,b]$ eine Nullstelle und $f'(t) \neq 0$ für alle $t \in [a,b]$. Dann liefert das Newton-Verfahren für jedes $x_0 \in [a,b]$ die Ungleichung

$$|x_1 - \xi| \;\leq\; C\,|x_0 - \xi|^2 \tag{42}$$

mit einer von x_0 unabhängigen Konstanten $C > 0$.

Beweis: Aus den Voraussetzungen folgt

$$\begin{aligned} C_1 &:= \max\{\left|f^{(2)}(t)\right| : t \in [a,b]\} < \infty \quad , \\ C_2 &:= \min\{|f'(t)| : t \in [a,b]\} > 0 \quad . \end{aligned}$$

Die Taylor-Entwicklung ergibt

$$0 = f(\xi) = f(x_0) + f'(x_0)(\xi - x_0) + \frac{f^{(2)}(\eta)}{2!}(\xi - x_0)^2$$

mit einem η zwischen x_0 und ξ. Daraus folgt

$$(\xi - x_0) + \frac{f(x_0)}{f'(x_0)} = -\frac{f^{(2)}(\eta)}{2!\, f'(x_0)}(\xi - x_0)^2 \quad .$$

Somit erhält man

$$\begin{aligned} |\xi - x_1| &= \left|\xi - x_0 + \tfrac{f(x_0)}{f'(x_0)}\right| \\ &= \frac{\left|f^{(2)}(\eta)\right|}{2!\,|f'(x_0)|}(\xi - x_0)^2 \\ &\leq \frac{C_1}{2C_2}(\xi - x_0)^2 = C(\xi - x_0)^2 \end{aligned}$$

mit $C := \frac{C_1}{2C_2}$. □

Aus diesem Ergebnis erhalten wir einerseits die lokale Konvergenz des Newton-Verfahrens und andererseits die Konvergenzordnung 2.

Satz: Es seien $f \in C^2[a,b]$, $\xi \in (a,b)$ mit $f(\xi) = 0$ und $f'(\xi) \neq 0$. Dann gilt:

(i) Das Newton-Verfahren besitzt die Konvergenzordnung 2.

(ii) Es gibt eine Umgebung U von ξ, so dass das Newton-Verfahren für alle Startwerte $x_0 \in U$ konvergiert (lokale Konvergenz).

Beweis: Wegen der Stetigkeit von f' und $f'(\xi) \neq 0$ gibt es eine (abgeschlossene) Umgebung $[\alpha,\beta] = [\xi - \varepsilon_1\,,\, \xi + \varepsilon_1] \subset [a,b]$ mit $f'(t) \neq 0$ für alle $t \in [\alpha,\beta]$. Nach obigem Satz gilt für alle Startwerte $x_0 \in [\alpha,\beta]$ die Ungleichung (42) mit einer Konstanten $C > 0$ Setze $\varepsilon := \min\{\frac{1}{2C}, \varepsilon_1\}$ und wähle als Umgebung $U = U_\varepsilon(\xi) = (\xi - \varepsilon, \xi + \varepsilon)$. Dann ergibt das Newton-Verfahren für jeden Startwert $x_0 \in U$

$$|x_1 - \xi| \leq C\,|x_0 - \xi|^2 < C\varepsilon^2 \leq C\left(\frac{1}{2C}\right)\varepsilon = \frac{1}{2}\cdot\varepsilon \quad .$$

Somit $x_1 \in U$ und folglich $x_i \in U$ für alle $i \in \mathbb{N}$. Aus (42) folgt dann die Konvergenzordnung 2 und ferner per Induktion

$$|x_{i+1} - \xi| \leq C\,|x_i - \xi|^2 < \left(\frac{1}{2}\right)^{i+1} \cdot \varepsilon$$

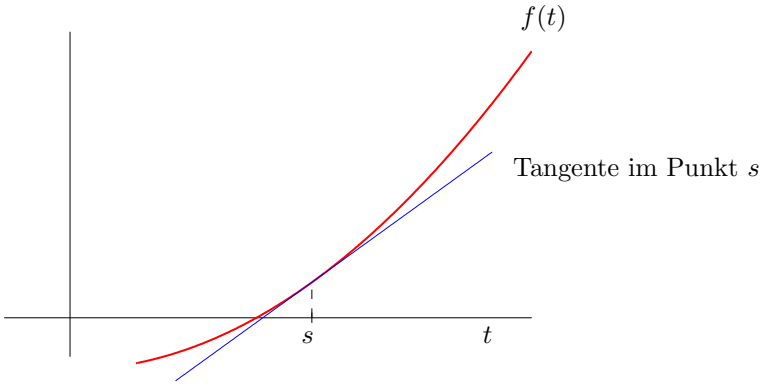
$\Big($denn $|x_{i+1} - \xi| \leq C\,|x_i - \xi|^2 < \left(\frac{1}{2}\right)^i \varepsilon\, C|x_i - \xi| \leq \left(\frac{1}{2}\right)^i \varepsilon\, C\, \frac{1}{2C}\Big)$, woraus sich die Konvergenz dieser Newton-Folge gegen ξ ergibt. □

Unter gewissen Zusatzvoraussetzungen kann man globale Konvergenz erreichen. Dazu betrachten wir eine Funktion $f : [a, b] \to \mathbb{R}$ mit folgenden Eigenschaften:

$$\begin{aligned}
&\text{(V1)} \quad \text{es gibt ein } \xi \in [a,b] \text{ mit } f(\xi) = 0 \quad , \\
&\text{(V2)} \quad f \in C^1[a,b] \text{ und } f'(t) \neq 0 \text{ für alle } t \in [a,b] \quad , \\
&\text{(V3)} \quad f'(s)\,(t-s) + f(s) \leq f(t) \text{ für alle } s,t \in [a,b] \quad , \\
&\text{(V4)} \quad \text{Für jeden Startwert } x_0 \in [a,b] \text{ gelte } x_1 := x_0 - \frac{f(x_0)}{f'(x_0)} \in [a,b] \quad .
\end{aligned} \tag{43}$$

Anmerkungen:

1. Aus (V2) folgt entweder $f'(t) > 0$ in $[a, b]$ oder $f'(t) < 0$ in $[a, b]$. Somit ist f streng monoton und hat höchstens eine Nullstelle in $[a, b]$.
2. Wegen (V1) und (V2) besitzt f genau eine Nullstelle in $[a, b]$.
3. (V3) bedeutet, dass in jedem Punkt $s \in [a, b]$ die Tangente an f unterhalb von f liegt, denn auf der linken Seite steht das Taylor-Polynome vom Grad 1 zu $f(t)$ im Punkt s.



Die Bedingung (V3) ist im Allgemeinen schwer nachzuprüfen; deshalb ist die folgende hinreichende Bedingung sehr nützlich.

Lemma: Aus $f \in C^2[a, b]$ und $f^{(2)}(t) \geq 0$ für alle $t \in [a, b]$ folgt die Bedingung (V3).

Beweis: Taylorentwicklung liefert:

$$f(t) \;=\; f(s) + f'(s)(t-s) + \frac{f^{(2)}(\eta)}{2!}(t-s)^2$$

für ein $\eta \in (\min\{s,t\}, \max\{s,t\})$. Daraus ergibt sich

$$f(t) - f(s) - f'(s)(t-s) \;=\; \frac{f^{(2)}(\eta)}{2!}(t-s)^2 \;\geq\; 0$$

und folglich die Behauptung. □

Satz (monotone Konvergenz):
Es seien die Voraussetzungen (V1) bis (V4) erfüllt und $x_0 \in [a,b]$ ein beliebiger Startwert. Dann gilt für das Newton-Verfahren:

(i) $\xi \le x_{i+1} \le x_i \quad (i = 1,2,3,\ldots), \quad$ falls $f'(t) > 0$ in $[a,b]$,

(ii) $x_i \le x_{i+1} \le \xi \quad (i = 1,2,3,\ldots), \quad$ falls $f'(t) < 0$ in $[a,b]$,

(iii) $\lim\limits_{i\to\infty} x_i \;=\; \xi$.

Beweis: (i) Sei $f'(t) > 0$ in $[a,b]$. Das Newton-Verfahren ergibt

$$x_1 \;=\; x_0 - \frac{f(x_0)}{f'(x_0)} \quad (\;\Leftrightarrow\; f'(x_0)(x_1 - x_0) + f(x_0) \;=\; 0\;)\,. \tag{44}$$

Falls $x_1 = \xi$ gilt, so ist man fertig (denn dann ist $x_i = \xi$ für alle $i \ge 1$). Sei also $x_1 \ne \xi$. Aus (44) und (V3) folgt die Beziehung

$$f(\xi) \;=\; 0 \;=\; f'(x_0)(x_1 - x_0) + f(x_0) \;\le\; f(x_1) \quad .$$

Der MWS liefert dann (beachte $f'(t) > 0$ in $[a,b]$)

$$\frac{f(x_1) - f(\xi)}{x_1 - \xi} \;=\; f'(\eta) \;>\; 0$$

und somit $x_1 - \xi > 0$. Also $\xi < x_1$ und nach Voraussetzung $x_1 \in [a,b]$.
Insgesamt hat man also $\xi \le x_1$ (Induktionsanfang).

Es gelte nun bereits

$$\xi \;\le\; x_n \;\le\; x_{n-1} \;\le\; \cdots \le x_1 \qquad \text{mit } f(x_n) \ge 0\,.$$

Wie oben folgt dann wieder

$$f(\xi) \;=\; 0 \;=\; f'(x_n)(x_{n+1} - x_n) + f(x_n) \;\le\; f(x_{n+1}) \tag{45}$$

und (falls $\xi \ne x_{n+1}$) aus dem MWS $\xi < x_{n+1}$. Wegen $f(x_n) \ge 0,\ f'(x_n) > 0$ gilt ferner

$$x_{n+1} \;=\; x_n - \frac{f(x_n)}{f'(x_n)} \;\le\; x_n \quad .$$

Insgesamt ergibt sich also $\xi \le x_{n+1} \le x_n$. Damit ist (i) mittels vollständiger Induktion bewiesen.

(ii) analog.

(iii) Die Folge $(x_n)_{n\in\mathbb{N}}$ ist monoton und beschränkt, also konvergent, d.h.

$$\gamma \;=\; \lim_{n\to\infty} x_n \quad .$$

Aufgrund der Stetigkeit der Iterationsfunktion $\Phi(t) = t - \frac{f(t)}{f'(t)}$ gilt

$$\Phi(\gamma) \;=\; \Phi(\lim_{n\to\infty} x_n) \;=\; \lim_{n\to\infty}(\Phi(x_n)) \;=\; \lim_{n\to\infty} x_{n+1} \;=\; \gamma$$

und folglich

$$f(\gamma) \;=\; 0\,, \quad \text{d.h. } \gamma \;=\; \xi$$

(da es nur eine Nullstelle in $[a,b]$ geben kann). $\square$

Beispiel: Berechnung von $\sqrt{a}$ für $a > 1$. Gesucht ist also eine positive Nullstelle der Funktion $f(t) := t^2 - a$. Es gilt

$$\begin{aligned} f'(t) &= 2t > 0 \quad \text{für } t > 0 \quad , \\ f^{(2)}(t) &= 2 > 0 \quad \text{für } t > 0 \quad , \\ f(1) &= 1 - a < 0 \quad (\text{da } a > 1) \quad , \\ f(a) &= a^2 - a > 0 \quad (\text{da } a > 1) \quad . \end{aligned}$$

Nach dem Zwischenwertsatz gibt es ein $\xi \in (1,a)$ mit $f(\xi) = 0$.
Sei $x_0 \in [1,a]$ ein beliebiger Startwert für das Newton-Verfahren. Für den ersten Newton-Schritt gilt:

$$x_1 \;=\; x_0 - \frac{f(x_0)}{f'(x_0)} \;=\; \frac{x_0}{2} + \frac{a}{2x_0} \;\le\; \frac{a}{2} + \frac{a}{2\cdot 1} \;=\; a$$

und

$$x_1 \;=\; \frac{x_0}{2} + \frac{a}{2x_0} \;\ge\; \frac{1}{2} + \frac{a}{2a} \;=\; 1 \quad .$$

Somit gilt $x_1 \in [1,a]$.
Es sind alle Voraussetzungen für den Satz über die monotone Konvergenz erfüllt. Für jeden Startwert $x_0 \in [1,a]$ konvergiert das Newton-Verfahren gegen $\sqrt{a}$.

5.4 Das Sekantenverfahren

Ein Nachteil des Newton-Verfahrens besteht darin, dass in jedem Schritt die Berechnung einer Ableitung von f (nämlich $f'(x_i)$) erforderlich ist. Dieses Problem lässt sich vermeiden, indem man im Newton-Verfahren die Tangente durch eine Sekante (oder - analytisch ausgedrückt - den Differentialquotienten durch einen Differenzenquotienten) ersetzt.

Das Verfahren

Gegeben seien $f \in C[a,b]$ sowie Startwerte x_0 und x_1 mit $x_0 \neq x_1$ (hier benötigt man zwei Startwerte). Für $i = 1,2,3,\ldots$ wird $f'(x_i)$ ersetzt durch

$$\frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}}$$

(dabei sei $f(x_i) \neq f(x_{i-1})$). Damit erhalten wir die Vorschrift

$$x_{i+1} \;:=\; x_i \;-\; \frac{(x_i - x_{i-1})\cdot f(x_i)}{f(x_i) - f(x_{i-1})} \;=\; \frac{f(x_i)x_{i-1} - f(x_{i-1})x_i}{f(x_i) - f(x_{i-1})} \quad . \tag{46}$$

Definition: Diese Methode heißt *Sekantenverfahren.*

Anmerkungen

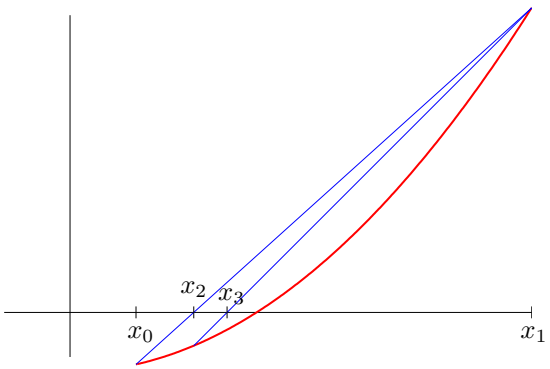
1. Dieses Verfahren ergibt sich auch bei der inversen Interpolation, jeweils an den beiden Paaren
$$(x_{i-1}, f(x_{i-1})) \text{ und } (x_i, f(x_i)) ,$$
und Auswertung mit dem Neville-Algorithmus (vgl. Abschnitt 4.8).

2. Das Sekantenverfahren hat die Konvergenzordnung $p = \frac{1+\sqrt{5}}{2} = 1.618\ldots$ (vgl. Stoer [7]). Im Gegensatz zum Newton-Verfahren ist in jedem Schritt nur eine Funktionsauswertung erforderlich.

Graphische Darstellung

Geometrisch ist x_{i+1} der Schnittpunkt der Sekante durch die Punkte $(x_{i-1}, f(x_{i-1}))$ und $(x_i, f(x_i))$ mit der x-Achse.

Sekantenverfahren



Variante

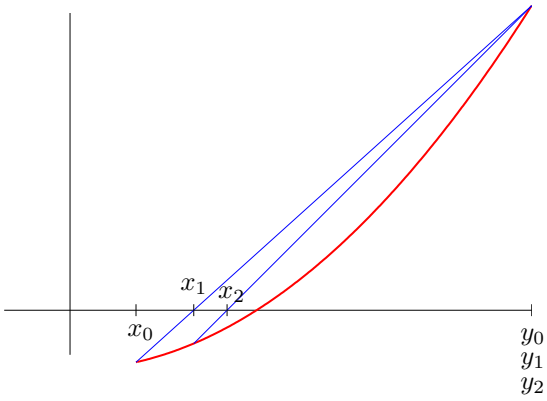
Zu stetigem f seien zwei Startwerte $x_0 < y_0$ gegeben mit $f(x_0)\, f(y_0) < 0$. Dann liegt eine Nullstelle im Intervall $[x_0, y_0]$. Beim Sekantenverfahren kann diese Information verloren gehen. Deshalb bietet sich folgende Modifikation an: Berechne wie beim Sekantenverfahren
$$s_i := \frac{f(y_i)x_i - f(x_i)y_i}{f(y_i) - f(x_i)}$$
Gilt $f(s_i) = 0$, so ist man fertig. Sonst setze (wie bei Bisektionsverfahren)
$$\begin{aligned} x_{i+1} &:= s_i \ , \ y_{i+1} := y_i \quad \text{falls} \quad f(s_i)\cdot f(y_i) < 0 \\ x_{i+1} &:= x_i \ , \ y_{i+1} := s_i \quad \text{falls} \quad f(x_i)\cdot f(s_i) < 0 \end{aligned}$$
$(i = 0, 1, 2\ldots)$.

Definition: Dieses Verfahren nennt man *regula falsi-Verfahren.*

Anmerkungen:

1. Auch das regula falsi-Verfahren kann über inverse Interpolation hergeleitet werden.
2. Ein Vorteil besteht darin, dass eine gesuchte Nullstelle im Intervall $[x_i, y_i]$ liegt.
3. Die Konvergenzordnung ist mindestens linear, d.h. $p \geq 1$ (vgl. Stoer [7]).

regula falsi-Verfahren



5.5 Anwendung aus dem Bereich der Finanzmathematik

Effektivzins bei prämienbegünstigten Sparverträgen
Bei dieser Geldanlageform zahlt der Sparer zu Beginn des Jahres jeweils einen festen, gleichbleibenden Betrag auf ein Sparbuch ein. Dieser wird mit einem gleichbleibenden Zinssatz p am Jahresende verzinst und dem Sparbuch gutgeschrieben. Nach Ablauf von n Jahren (z.B. n = 7) zahlt das Kreditinstitut dem Sparer dann für die eingezahlten Beträge eine Prämie von 14 %, und der Kunde kann über das gesamte Kapital (Sparbeträge, Zinsen, Prämie) frei verfügen.

Gefragt ist nun die *Effektivverzinsung*, d.h. welcher Zinssatz q ist anzusetzen, damit der Kunde für seine Sparbeträge das gleiche Kapital ausbezahlt bekommt, wenn keine Prämie gewährt wird?

Lösung:
Zur Abkürzung setzen wir $s := 1 + \frac{p}{100}$. Der jährliche Sparbetrag sei R. Unter Berücksichtigung von Zinsen und Zinseszinsen hat man nach n Jahren:

für die Einzahlung R zu Beginn des 1. Jahres:	$R(1+\frac{p}{100})^n$	$=$	Rs^n
für die Einzahlung R zu Beginn des 2. Jahres:			Rs^{n-1}
für die Einzahlung R zu Beginn des 3. Jahres:			Rs^{n-2}
$\vdots$			$\vdots$
für die Einzahlung R zu Beginn des n. Jahres:			Rs

Insgesamt also an Sparbeiträgen und Zinsen (geom. Reihe)

$$R\left(\sum_{i=1}^{n} s^i\right) \;=\; R\left(\frac{s^{n+1}-s}{s-1}\right) \quad .$$

Die gesamten Sparbeiträge sind nR; somit beträgt die Prämie $0.14 nR$. Der Kunde erhält folgendes Kapital ausbezahlt:

$$K \;=\; R\left(\frac{s^{n+1}-s}{s-1}\right) + 0.14nR \quad .$$

Sei q der Effektivzins und $r := 1 + \frac{q}{100}$. Dann muss gelten

$$K \;=\; R\left(\frac{r^{n+1}-r}{r-1}\right) \quad ,$$

dabei ist r die Unbekannte, während K, R, n gegeben sind.

O.B.d.A. kann man $R = 1$ annehmen. Es ist also eine Gleichung der Form

$$f(t) \;:=\; \frac{t^{n+1}-t}{t-1} - c \;=\; 0$$

zu lösen, wobei $c > 0,\ n \geq 1$ und $t > 1$ gelten muss.
Äquivalent dazu ist

$$f(t) \;=\; t^n + t^{n-1} + \cdots + t - c \;=\; 0 \quad . \tag{47}$$

Wir prüfen nun die Bedingungen (V1) bis (V4) nach.

Es gilt $f \in C[0,\infty)$ und

$$f'(t) \;=\; nt^{n-1} + \cdots + 2t + 1 \;>\; 0 \qquad \text{in } [0,\infty)\ .$$

Somit ist die Bedingung (V2) erfüllt. Ferner hat man

$$f^{(2)}(t) \;=\; n(n-1)t^{n-2} + \cdots + 6t + 2 \;>\; 0 \qquad \text{in } [0,\infty).$$

Daraus folgt die Bedingung (V3).
Aus

$$\begin{aligned} f(0) \;&=\; -c \;<\; 0\,, \\ f(c) \;&=\; c^n + \cdots + c^2 \;>\; 0 \end{aligned}$$

folgt nach dem Zwischenwertsatz die Existenz einer Nullstelle $r \in [0, c]$ und damit auch in $[0,\infty)$. Wegen (V2) ist r die einzige Nullstelle.

Für jeden beliebigen Startwert $x_0 \in [0,\infty)$ gilt wegen (V3)

$$f'(x_0)(r - x_0) \;+\; f(x_0) \;\leq\; f(r) \;=\; 0 \quad ,$$

woraus wir

$$r \;\leq\; x_0 \;-\; \frac{f(x_0)}{f'(x_0)} \;=\; x_1$$

erhalten und somit $x_1 \in [0,\infty)$.

Insgesamt sind die Bedingungen (V1) - (V4) erfüllt.

Deshalb wenden wir hier das Newton-Verfahren an. Für jeden Startwert $x_0 \in [0,\infty)$ erhält man monotone Konvergenz gegen die gesuchte Nullstelle r. Als Startwert für unser Ausgangsproblem empfiehlt sich $x_0 := 1 + \frac{p}{100}$.

5.6 Nullstellenberechnung mit Matlab

In Matlab steht uns zur Nullstellenberechnung die Funktion `fzero` zur Verfügung. Durch Eingabe von

```
doc fzero
```

erhält man eine ausführliche Beschreibung.

Als Beispiel berechnen wir die positive Nullstelle von der Funktion $f(x) := x^2 - 2$. Zunächst wird eine Matlab-Funktion erstellt:

```
function y = funk1(t)
y = t^2 - 2;
```

Diese wird in die Datei namens `funk1.m` gespeichert. Der Aufruf der Nullstellenfunktion `fzero` lautet dann

```
z = fzero(@funk1,5)
z = fzero(@funk1,[0,3])
```

Beachten Sie, dass die Übergabe von Matlab-Funktionen als Paramter an andere Funktionen mit vorangestelltem `@` erfolgt. Im ersten Fall wird als Startwert für die Nullstellensuche der Wert 5 verwendet. Im zweiten Fall wird ein Intervall angegeben, in dem sich eine Nullstelle befindet.

Ferner kann man noch zwei Optionen setzen:

- Gewünschte Genauigkeit:
  ```
  options = optimset('TolX',1e-12)
  ```
- Ausgabe der Zwischenergebnisse
  ```
  options = optimset('Display','iter')
  ```

Selbstverständlich können beide Optionen gleichzeitig gesetzt werden:

```
options = optimset('TolX',1e-12,'Display','iter')
```

Die Optionen werden als weiterer Parameter an die Nullstellenroutine übergeben:

```
z = fzero(@funk1,[0,3],options)
```

Wir erhalten dann folgende Ausgabe:

```
Func-count      x          f(x)         Procedure
    1           0            -2         initial
    2           3             7         initial
    3    0.666667      -1.55556         interpolation
    4     1.83333       1.36111         bisection
    5     1.28889     -0.338765         interpolation
    6     1.39739    -0.0473004         interpolation
    7     1.41441   0.000547369         interpolation
    8     1.41421  -3.27474e-06         interpolation
    9     1.41421  -2.24031e-10         interpolation
   10     1.41421   4.44089e-16         interpolation
   11     1.41421   -7.9996e-12         interpolation
Zero found in the interval: [0, 3].
z =
   1.41421356237310
```

Die Funktion `fzero` kombiniert verschiedene numerische Verfahren (Bisektion, Sekantenverfahren, inverse Interpolation). Die letzte Spalte gibt an, welches Verfahren im jeweiligen Zwischenschritt verwendet wurde.

Funktionen mit zusätzlichen Parametern

An die Nullstellenroutine dürfen wir auch Funktionen übergeben, die ihrerseits wieder Parameter haben. Diese müssen dann als zusätzliche Parameter an `fzero` übergeben werden. Als Beispiel betrachten wir die Funktion

$$f(t) \;=\; t^n \;-\; c \qquad (n \in \mathbb{N},\; c > 0).$$

In die Datei `funk2.m` schreibt man wieder die Matlab-Funktion

```
function y = funk2(t,n,c)
y = t^n - c;
```

In der Matlab-Sitzung geben wir dann folgende Befehle ein:

```
n = 3;
c = 5;
fzero(@funk2,[0,3],options,n,c)
```

Konkret wird dann $\sqrt[3]{5}$ berechnet. Möglich ist auch

`fzero(@funk2,[0,3],options,3,5)` .

Sind keine Optionen definiert, so muss hier trotzdem ein Platzhalter angegeben werden:

`fzero(@funk2,[0,3],[ ],n,c)`

6 Anfangswertaufgaben

6.1 Einleitung

Betrachtet man Entwicklungen von Vorgängen in der Natur, so führt uns das in der Regel auf Differentialgleichungen. Allgemein hat eine Differentialgleichung die Form

$$\dot{x} \;=\; f(t,x) \quad .$$

Dabei ist f eine Funktion, welche auf einer Menge $D \subset \mathbb{R}^2$ definiert ist. Sei I ein Intervall. Eine differenzierbare Funktion $x : I \to \mathbb{R}$ heißt eine Lösung der Differentialgleichung, wenn der Graph von x in D liegt und

$$\dot{x}(t) \;=\; f(t,x(t)) \quad \text{für alle } t \in I$$

gilt.

Beispiele:

1. Es bezeichne $x(t)$ den Umfang einer Population (z.B. Tiere, Organismen) zum Zeitpunkt t. In der Biologie interessiert man sich für das Wachstum dieser Population. Die einfachste Vorstellung nimmt an, dass das Wachstum (Veränderungsrate) proportional zum Umfang der Population ist. Dies bedeutet

$$\dot{x} \;=\; Rx \tag{48}$$

   mit einem Proportionalitätsfaktor $R \in \mathbb{R},\ R > 0$. Diese Dgl wird als *Ratengleichung* bezeichnet.

2. Ein verfeinerter Ansatz geht davon aus, dass das Wachstum durch Umwelteinflüsse begrenzt (negativ beeinflusst) wird. Dies führt uns auf die sogenannte *Verhulst-Gleichung:*

$$\dot{x} \;=\; Rx\left(1 - \frac{x}{K}\right) \tag{49}$$

   mit $R, K \in \mathbb{R},\ R > 0,\ K > 0$.

3. Wir betrachten einen Fallschirmspringer mit Masse m. Für die Fallgeschwindigkeit $v(t)$ erhält man die Differentialgleichung

$$\dot{v} \;=\; g \;-\; \frac{k(t)v^2}{m} \quad . \tag{50}$$

   Dabei ist $g = 9.81$ die Gravitationskonstante und die Funktion $k(t)$ die Reibung zum Zeitpunkt t. Diese kann etwa durch die Funktion

$$k(t) \;:=\; \begin{cases} k_1 & \text{für } 0 \le t < t_1 \\ k_1 + \frac{(t-t_1)(k_2-k_1)}{t_2-t_1} & \text{für } t_1 < t < t_2 \\ k_2 & \text{für } t \ge t_2 \end{cases}$$

   beschrieben werden mit $0 < t_1 < t_2$ und $k_1 < k_2$ (lineare Entfaltung des Fallschirms).

Häufig treten in den Anwendungen Differentialgleichungssysteme auf:

$$\begin{aligned} \dot{x}_1 &= f_1(t, x_1, \ldots, x_n) \\ &\vdots \\ \dot{x}_n &= f_n(t, x_1, \ldots, x_n) \end{aligned}$$

mit gewissen Funktionen $f_i(t, x_1, \ldots, x_n)$ oder in Vektorschreibweise

$$\dot{x} = F(t, x) \quad .$$

Beispiel: (Räuber-Beute-Modell)
Bezeichnet man mit $y(t)$ die Räuber (z.B. Bussarde) und mit $z(t)$ die Beute (z.B. Mäuse) zum Zeitpunkt t, so ergibt sich folgendes Modell:

$$\begin{aligned} \dot{y} &= \alpha z y - \beta y \\ \dot{z} &= -\gamma z y + \delta z \end{aligned} \tag{51}$$

mit positiven Konstanten $\alpha,\ \beta,\ \gamma,\ \delta$.

In (48),(49) und (51) tritt die Variable t auf der rechten Seite nicht explizit auf. Solche Differentialgleichungen heißen *autonom*. Sie sind also von der Form

$$\dot{x} = F(x) \quad .$$

Dagegen tritt in der rechten Seite von (50) die Variable t explizit auf. Man spricht deshalb von einer *nichtautonomen* Differentialgleichung.

Wir kehren zum 1. Beispiel zurück und verifizieren sofort, dass die Funktion

$$x(t) := a \cdot \exp(Rt)$$

für jedes beliebige $a \in \mathbb{R}$ eine Lösung der Ratengleichung ist. Dies ändert sich, wenn man zusätzlich zur Dgl noch eine Anfangsbedingung fordert, z.B. $x(0) = 20$. Dann ist

$$x(t) := 20 \exp(Rt)$$

die einzige Funktion, welche die Differentialgleichung und die Anfangsbedingung erfüllt.

Eine Differentialgleichung zusammen mit einer Anfangsbedingung nennt man eine *Anfangswertaufgabe* (AWA). Im allgemeinen Fall haben wir also

$$\begin{aligned} \dot{x}_1 &= f_1(t, x_1, \ldots, x_n)\,, \quad x_1(t_0) &= y_1^{(0)} \\ &\vdots & \vdots \\ \dot{x}_n &= f_n(t, x_1, \ldots, x_n)\,, \quad x_n(t_0) &= y_n^{(0)} \end{aligned} \tag{52}$$

oder in Vektorschreibweise

$$\dot{x} = F(t, x)\,, \quad x(t_0) = y^{(0)} \quad .$$

Unter gewissen Voraussetzungen (z.B. $F \in C^1$) besitzt diese Anfangswertaufgabe genau eine Lösung (Existenz- und Eindeutigkeitssatz, vgl. Analysis III).

Wir setzen in diesem Paragraphen die eindeutige Lösbarkeit von (52) voraus und beschäftigen uns mit numerischen Verfahren zur Berechnung dieser Lösung in einem kompakten Intervall $[t_0, b]$.

Anmerkung: Bei der Herleitung von numerischen Verfahren können wir uns auf autonome Systeme

$$\dot{x} \;=\; F(x)\,, \quad x(t_0) \;=\; y^{(0)} \tag{53}$$

mit $F \in C^1(\mathbb{R}^n, \mathbb{R}^n)$ beschränken. Denn:

1. Jedes nichtautonome System kann in ein autonomes transformiert werden. Mit

   $$z \;=\; (z_0, z_1, \ldots, z_n) \;=\; (t, x_1, \ldots, x_n) \qquad \text{und} \qquad z_0(t_0) = t_0$$

   ergibt sich aus dem nichtautonomen System (52) das autonome System

   $$\dot{z} \;=\; G(z)\,, \quad z(t_0) \;=\; \begin{pmatrix} t_0 \\ y^{(0)} \end{pmatrix} \qquad \text{mit} \qquad G(z) \;:=\; (1, f_1(z), \ldots, f_n(z)) \quad .$$

2. Auch ein Dgl-System höherer Ordnung lässt sich in ein autonomes Dgl-System 1. Ordnung transformieren. Wir demonstrieren dies für die Pendelgleichung ohne Reibung:

   $$\ddot{x} \;=\; -\lambda \sin(x)\,, \quad x(t_0) = \alpha\,,\ \dot{x}(t_0) = \beta\,.$$

   Mit $z_1 := x$ und $z_2 := \dot{x}$ erhält man

   $$\begin{aligned} \dot{z}_1 \;&=\; \dot{x} \;=\; z_2\,, & z_1(t_0) \;&=\; \alpha \quad , \\ \dot{z}_2 \;&=\; \ddot{x} \;=\; -\lambda \sin(z_1)\,, & z_2(t_0) \;&=\; \beta \quad . \end{aligned}$$

6.2 Einteilung der Näherungsverfahren

Kontinuierliche Verfahren

Sie liefern eine Funktion $\varphi(t)$ als Näherung an die wahre Lösung von (53), die mit $x(t)$ bezeichnet wird. Dazu gehört z. B. die *Picard-Iteration:*

$$\begin{aligned} \varphi_0(t) \;&:=\; y^{(0)} \quad , \\ \varphi_{n+1}(t) \;&:=\; y^{(0)} + \int_{t_0}^{t} F(\varphi_n(s))\, ds \quad . \end{aligned}$$

Als Beispiel betrachten wir die skalare Anfangswertaufgabe

$$\dot{x} \;=\; x\,, \quad x(0) \;=\; 1 \quad .$$

Dann erhält man

$$\begin{aligned}
\varphi_0(t) &= 1 \quad , \\
\varphi_1(t) &= 1 + \int_0^t 1\, ds \qquad\quad = 1 + t \quad , \\
\varphi_2(t) &= 1 + \int_0^t (1+s)\, ds \quad = 1 + t + \tfrac{1}{2}\, t^2 \quad , \\
&\vdots \\
\varphi_n(t) &= 1 + \int_0^t \varphi_{n-1}(s)\, ds \quad = 1 + t + \tfrac{1}{2}\, t^2 + \cdots + \tfrac{1}{n!}\, t^n \quad .
\end{aligned}$$

Diskrete Verfahren

Diese Verfahren liefern in Gitterpunkten $t_0 < t_1 < t_2 < \ldots$ Näherungswerte $y^{(k)}$ für die exakte Lösung zu den Zeitpunkten t_k, also

$$y^{(k)} \;\approx\; x(t_k) \quad (k = 1, 2, 3, \ldots).$$

Die diskreten Verfahren werden unterteilt in

— **Einschrittverfahren**

Der Näherungswert $y^{(k)}$ wird berechnet aus $y^{(k-1)}$ $(k = 1, 2, 3, \ldots)$.

— **Mehrschrittverfahren**

Der Näherungswert $y^{(k)}$ wird berechnet aus $y^{(k-1)}, \ldots, y^{(k-l)}$ (l-Schrittverfahren). Man braucht hier l Startwerte $y^{(0)}$, $y^{(1)}$, …, $y^{(l-1)}$.

Im Folgenden gehen wir nur auf die Einschrittverfahren ein.

6.3 Einschrittverfahren

Wir wählen eine feste Schrittweite, betrachten die Gitterpunkte

$$t_j \;=\; t_0 + jh \quad (j \in \mathbb{N}) \qquad \text{bzw.} \qquad t_j \;=\; t_{j-1} + h \tag{54}$$

und suchen dann Näherungswerte $y^{(j)}$ an die tatsächliche Lösung $x(t_j)$ der gegebenen AWA, also $y^{(j)} \approx x(t_j)$.

Ein Einschrittverfahren geht von einer sogenannten *Verfahrensfunktion (numerischen Zuwachsfunktion)*

$$V(h, x) \;\in\; C(\mathbb{R}^{n+1}, \mathbb{R}^n)$$

aus. Beginnend mit dem $y^{(0)}$ aus der gegebenen Anfangsbedingung bildet man iterativ die Vektoren

$$y^{(k+1)} \;=\; y^{(k)} + h \cdot V(h, y^{(k)}) \qquad (k = 0, 1, 2, \ldots). \tag{55}$$

Anmerkung: Für eine nichtautonome AWA bekommt man

$$y^{(k+1)} \;=\; y^{(k)} + h \cdot V(h, t_k, y^{(k)}) \qquad (k = 0, 1, 2, \ldots)$$

(hier ist $V \in C(\mathbb{R}^{n+2}, \mathbb{R}^n)$).

Das Euler-Cauchy-Verfahren

Die einfachste Wahl für eine Verfahrensfunktion ist

$$V(h,x) := F(x) \quad .$$

Zu der Anfangsbedingung $(t_0, y^{(0)})$ ergibt sich folgendes Verfahren:

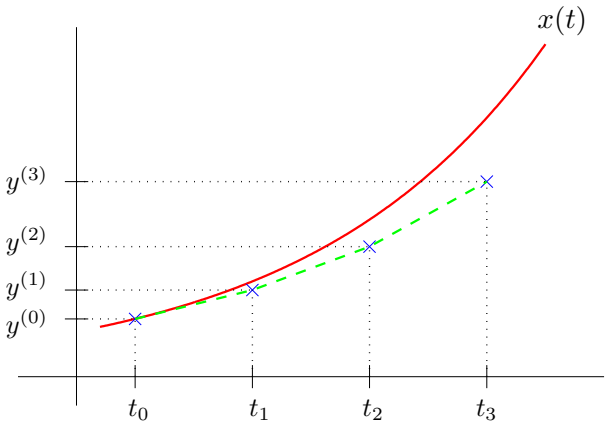
$$y^{(k+1)} = y^{(k)} + hF(y^{(k)}) \qquad (k = 0, 1, 2, \ldots)$$

oder in Komponentenschreibweise

$$y_i^{(k+1)} = y_i^{(k)} + hf_i(y^{(k)}) \qquad (i = 1, \ldots, n) \; .$$

Definition: Dieses Verfahren heißt *Euler-Cauchy-Verfahren.*

Im skalaren Fall hat das Euler-Cauchy-Verfahren die folgende graphische Veranschaulichung:



Sei $y^{(k)}$ schon berechnet. Die Dgl liefert im Punkt $(t_k, y^{(k)})$ die Steigung $K := F(y^{(k)})$. Dann ist $y^{(k+1)}$ der Schnittpunkt der Geraden durch $(t_k, y^{(k)})$ mit der Steigung K und der Geraden $t = t_{k+1}$.

Explizite Runge-Kutta-Verfahren

Wir betrachten wieder die autonome AWA

$$\dot{x} = F(x) \; , \qquad x(t_0) = y^{(0)} \quad (F : \mathbb{R}^n \to \mathbb{R}^n).$$

Ein explizites Runge-Kutta-Verfahren der Stufe s wird dargestellt durch die Matrix mit reellen Einträgen

$$\left(\begin{array}{cccccc}
\beta_{21} & & & & & \\
\beta_{31} & \beta_{32} & & & & \\
\beta_{41} & \beta_{42} & \beta_{43} & & & \\
\vdots & \vdots & \vdots & \ddots & & \\
\beta_{s1} & \beta_{s2} & \beta_{s3} & \cdots & \beta_{s,s-1} & \\
\hline
\gamma_1 & \gamma_2 & \gamma_3 & \cdots & \gamma_{s-1} & \gamma_s
\end{array}\right)$$

Dadurch wird folgendes Verfahren beschrieben:

$$\begin{aligned}
K^{(1)}(h,x) &= F(x) \\
K^{(2)}(h,x) &= F\left(x + h\beta_{21}K^{(1)}(h,x)\right) \\
K^{(3)}(h,x) &= F\left(x + h\beta_{31}K^{(1)}(h,x) + h\beta_{32}K^{(2)}(h,x)\right) \\
&\vdots \\
K^{(j)}(h,x) &= F\left(x + \sum_{i=1}^{j-1} h\beta_{ji}K^{(i)}(h,x)\right) \qquad (j = 2,\ldots,s).
\end{aligned}$$

Mit diesen Hilfsgrößen bildet man die Verfahrensfunktion

$$V(h,x) = \sum_{j=1}^{s} \gamma_j K^{(j)}(h,x) \quad .$$

Da zur Berechnung von $K^{(j)}$ nur die bereits berechneten Größen $K^{(1)},\ldots,K^{(j-1)}$ herangezogen werden, nennt man diese Verfahren *explizite Runge-Kutta-Verfahren.*

Beispiele:

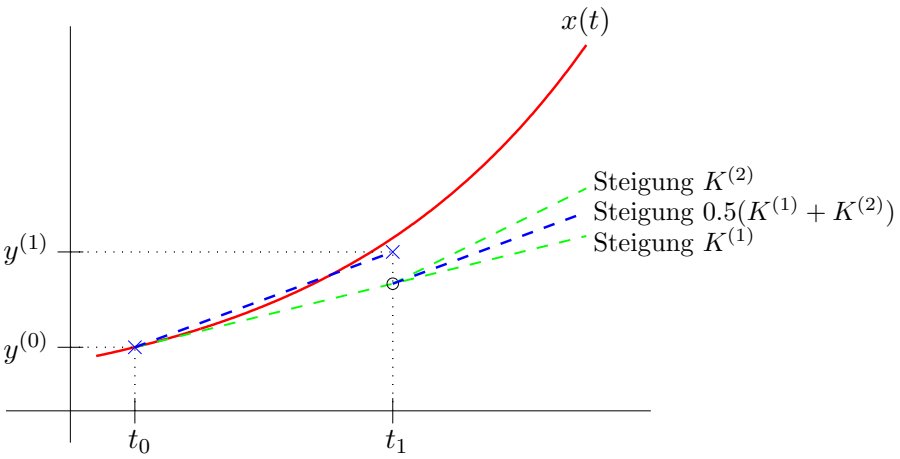
1. *Stufe $s = 2$:*

$$\begin{pmatrix} 1 & \\ \hline \frac{1}{2} & \frac{1}{2} \end{pmatrix} \qquad \text{(zweistufiges Verfahren von Heun).}$$

Wir wollen uns dieses Verfahren für skalaren Fall $\dot{x} = f(x)$, $x(t_0) = y^{(0)}$, anschauen. Hier ergibt sich folgender Algorithmus $(k = 0, 1, 2, \ldots)$

$$\begin{aligned}
K^{(1)} &= f(y^{(k)}) \,, \\
K^{(2)} &= f(y^{(k)} + hK^{(1)}) \,, \\
y^{(k+1)} &= y^{(k)} + \frac{h}{2}\left(K^{(1)} + K^{(2)}\right) \,.
\end{aligned}$$

Dieses Verfahren besitzt folgende graphische Darstellung:



$$\left(\begin{array}{cc} \frac{1}{2} & \\ \hline 0 & 1 \end{array}\right) \qquad \text{(Halbschrittverfahren).}$$

2. *Stufe* $s = 3$*:*

$$\left(\begin{array}{ccc} \frac{1}{3} & & \\ 0 & \frac{2}{3} & \\ \hline \frac{1}{4} & 0 & \frac{3}{4} \end{array}\right) \qquad \text{(3-stufiges Verfahren von Heun),}$$

$$\left(\begin{array}{ccc} \frac{1}{2} & & \\ -1 & 2 & \\ \hline \frac{1}{6} & \frac{4}{6} & \frac{1}{6} \end{array}\right) \qquad \text{(Verfahren von Kutta).}$$

3. *Stufe* $s = 4$*:*

$$\left(\begin{array}{cccc} \frac{1}{2} & & & \\ 0 & \frac{1}{2} & & \\ 0 & 0 & 1 & \\ \hline \frac{1}{6} & \frac{2}{6} & \frac{2}{6} & \frac{1}{6} \end{array}\right) \qquad \text{(klassisches Runge-Kutta-Verfahren).}$$

Implizite Runge-Kutta-Verfahren

Wir betrachten wieder die autonome AWA

$$\dot{x} \;=\; F(x)\,, \qquad x(t_0) \;=\; y^{(0)} \quad (F \,:\, \mathbb{R}^n \to \mathbb{R}^n). \tag{56}$$

Ein implizites Runge-Kutta-Verfahren der Stufe s wird dargestellt durch die Matrix

$$\left(\begin{array}{cccccc} \beta_{11} & \beta_{12} & \beta_{13} & \cdots & \beta_{1,s-1} & \beta_{1s} \\ \beta_{21} & \beta_{22} & \beta_{23} & \cdots & \beta_{2,s-1} & \beta_{2,s} \\ \beta_{31} & \beta_{32} & \beta_{33} & \cdots & \beta_{3,s-1} & \beta_{3,s} \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ \beta_{s1} & \beta_{s2} & \beta_{s3} & \cdots & \beta_{s,s-1} & \beta_{ss} \\ \hline \gamma_1 & \gamma_2 & \gamma_3 & \cdots & \gamma_{s-1} & \gamma_s \end{array}\right)$$

mit $\beta_{ij} \neq 0$ für mindestens ein $j \geq i$.

Dadurch wird folgendes Verfahren beschrieben:

$$\begin{aligned} K^{(1)}(h,x) &= F\left(x + \sum_{i=1}^{s} h\beta_{1i}K^{(i)}(h,x)\right) \\ K^{(2)}(h,x) &= F\left(x + \sum_{i=1}^{s} h\beta_{2i}K^{(i)}(h,x)\right) \\ &\vdots \\ K^{(j)}(h,x) &= F\left(x + \sum_{i=1}^{s} h\beta_{ji}K^{(i)}(h,x)\right) \qquad (j = 1,\ldots,s). \end{aligned}$$

Daraus bildet man die Verfahrensfunktion

$$V(h,x) \;=\; \sum_{j=1}^{s} \gamma_j K^{(j)}(h,x) \quad .$$

Die Näherungswerte $y^{(k+1)}$ werden somit gemäß

$$y^{(k+1)} \;:=\; y^{(k)} \;+\; h\sum_{j=1}^{s} \gamma_j K^{(j)}(h,y^{(k)}) \qquad (k = 0,1,2,\ldots)$$

berechnet.

Zur Bestimmung von $K^{(j)}(h,y^{(k)})$ werden die unbekannten Größen $K^{(1)}(h,y^{(k)}),\ldots,$ $K^{(s)}(h,y^{(k)})$ herangezogen. Die $K^{(j)}(h,y^{(k)})$ sind durch eine implizite Gleichung gegeben, welche in jedem Schritt (d.h. für jedes k) neu gelöst werden muss.
Deshalb spricht man hier von *impliziten Runge-Kutta-Verfahren.*

Beispiel (Stufe $s = 1$): $\left(\dfrac{1}{1}\right)$.

Diese Matrix beschreibt die Verfahrensfunktion

$$V(h,x) \;=\; K^{(1)}(h,x) \;=\; F\left(x + hK^{(1)}(h,x)\right)$$

und somit

$$\begin{aligned} y^{(0)} &= x(t_0) \qquad \text{(Anfangsbedingung)}, \\ y^{(k+1)} &= y^{(k)} + hK^{(1)}\left(h,y^{(k)}\right) \;=\; y^{(k)} + hF\left(y^{(k+1)}\right) \quad (k = 0,1,2,\ldots)\,. \end{aligned}$$

Dieses Verfahren heißt *Rückwärts-Euler-Verfahren* (oder *implizites Euler-Verfahren).*

6.4 Konsistenz und Konvergenz

Diskretisierungsfehler

Die Anfangswertaufgabe

$$\dot{x} \;=\; F(x)\,,\; x(t_0) = y^{(0)}$$

werde durch ein Einschrittverfahren mit der Verfahrensfunktion $V(h,x)$ näherungsweise gelöst. Dabei sei $x(t)$ die exakte Lösung dieser AWA.

Wir fragen nun nach dem Fehler, der bei einem Iterationsschritt entsteht. Dazu definiert man

$$\Delta(h,y^{(0)}) \;:=\; \begin{cases} \frac{x(t_0+h)-x(t_0)}{h} & \text{für } h \neq 0 \\ F(y^{(0)}) & \text{für } h = 0 \end{cases} \quad .$$

Δ ist die *wahre Zuwachsfunktion*. Würde man den Näherungswert $y^{(1)}$ mit dieser (unbekannten) Steigung berechnen, also

$$y^{(1)} \;=\; y^{(0)} \;+\; h{\cdot}\Delta(h,y^{(0)}) \quad ,$$

so ergäbe sich für $y^{(1)}$ die exakte Lösung $x(t_1)$.

Definition: Die Funktion

$$\tau(h,y^{(0)}) \;:=\; \Delta(h,y^{(0)}) - V(h,y^{(0)}) \tag{57}$$

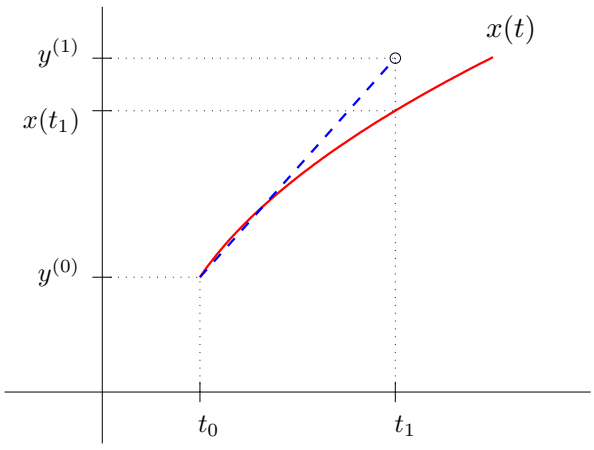
heißt *lokaler Diskretisierungsfehler* (Beachte: τ hängt auch von F ab).

Für den Schrittfehler folgt

$$\begin{aligned} x(t_1) - y^{(1)} \;&=\; x(t_0+h) - y^{(1)} \;=\; x(t_0+h) - y^{(0)} - hV(h,y^{(0)}) \\ &=\; h\left(\frac{x(t_0+h)-x(t_0)}{h} - V(h,y^{(0)})\right) \\ &=\; h{\cdot}\tau(h,y^{(0)}) \quad . \end{aligned}$$

Fazit: Schrittfehler = Schrittweite * lokaler Diskretisierungsfehler.

Schrittfehler



Von einem numerischen Verfahren erwartet man, dass der lokale Diskretisierungsfehler klein wird für kleine Schrittweite h, und zwar für jeden Startwert $(t_0, y^{(0)})$ aus dem Definitionsgebiet der Dgl und für jedes F, das gewissen Glattheitsbedingungen genügt. Dies führt uns auf die Begriffe

Konsistenz und Konsistenzordnung

Sei $N \in \mathbb{N}$. Mit B_N bezeichnen wir die Menge der Funktionen $F : \mathbb{R}^n \to \mathbb{R}^n$, für die gilt

- F ist stetig und beschränkt,
- es existieren sämtliche partielle Ableitungen von F bis zur Ordnung N, diese sind stetig und beschränkt.

Definitionen:

1. Das von der Verfahrensfunktion V erzeugte Verfahren heißt *konsistent*, wenn gilt

$$\lim_{h \to 0} \tau(h, y^{(0)}) \;=\; 0 \tag{58}$$

   für jeden AW $(t_0, y^{(0)})$ und für jedes $F \in B_1$.

2. Das von V erzeugte Einschrittverfahren heißt ein Verfahren der *Konsistenzordnung* p, wenn es ein $p \in \mathbb{N}_1$ gibt, so dass für jeden AW $(t_0, y^{(0)})$ und für jedes $F \in B_p$ gilt

$$\tau(h, y^{(0)}) \;=\; \mathcal{O}(h^p) \qquad (h \to 0).$$

Es ist sofort klar, dass ein Verfahren der Konsistenzordnung p insbesondere konsistent ist.

Aus der Konsistenzbedingung (58) folgt für die Verfahrensfunktion

$$V(0, y^{(0)}) \;=\; F(y^{(0)}) \quad ,$$

$$\text{denn} \quad V(0, y^{(0)}) \;=\; \lim_{h \to 0} V(h, y^{(0)}) \;=\; \lim_{h \to 0} \frac{x(t_0 + h) - x(t_0)}{h} \;=\; F(y^{(0)}) \quad .$$

Die Konsistenzordnung dient uns als Gütemaß, um verschiedene Einschrittverfahren miteinander zu vergleichen. Allerdings muss man dies in Relation sehen zum benötigten Rechenaufwand.

Beispiel: Wir bestimmen die Konsistenzordnung beim Euler-Cauchy-Verfahren (im skalaren Fall). Es sei $x(t)$ die exakte Lösung der Anfangswertaufgabe

$$\dot{x} \;=\; f(x)\,,\; x(t_0) = y^{(0)}$$

und $f \in B_1$. Die Taylor-Entwicklung liefert

$$x(t_0 + h) \;=\; x(t_0) \;+\; \dot{x}(t_0) \cdot h \;+\; \ddot{x}(\xi) \cdot \frac{h^2}{2}$$

mit einem ξ zwischen t_0 und $t_0 + h$. Weiter gilt

$$\begin{aligned} \dot{x}(t_0) \;&=\; f(x(t_0)) \;=\; f(y^{(0)}) \quad , \\ \ddot{x}(\xi) \;&=\; f'(x(\xi)) \cdot \dot{x}(\xi) \;=\; f'(x(\xi)) \cdot f(x(\xi)) \quad . \end{aligned}$$

Somit erhalten wir für den wahren Zuwachs

$$\Delta(h, y^{(0)}) \;=\; \frac{x(t_0 + h) - x(t_0)}{h} \;=\; f(y^{(0)}) \;+\; \frac{h}{2} f'(x(\xi)) \cdot f(x(\xi))$$

und für den lokalen Diskretisierungsfehler (beachte $V(h, y^{(0)}) = f(y^{(0)})$)

$$\tau(h, y^{(0)}) \;=\; \Delta(h, y^{(0)}) \;-\; V(h, y^{(0)}) \;=\; \frac{h}{2}\cdot f'(x(\xi))\cdot f(x(\xi)) \;=\; \mathcal{O}(h) \quad .$$

Das Euler-Cauchy-Verfahren besitzt damit die Konsistenzordnung 1.

Für Runge-Kutta-Verfahren liefert die Konsistenzforderung $V(0, x) = F(x)$ die Bedingung

$$\sum_{j=1}^{s} \gamma_j \;=\; 1 \quad .$$

Fordert man noch eine bestimmte Konsistenzordnung, so ergeben sich nichtlineare Bedingungsgleichungen für die β_{ji}. Die Zusammenhänge sind allerdings sehr kompliziert. Die Frage, welche Konsistenzordnung mit einem s-stufigen expliziten RKV maximal erreicht werden kann, ist bisher nur für kleinere s gelöst. Es gilt

Stufe s	1	2	3	4	5	6	7	8	9	10	$s \geq 10$
Max. KO	1	2	3	4	4	5	6	6	7	7	$\leq s-3$

Wir geben noch die Konsistenzordnung einiger Verfahren an (vergleiche auch Übungen):

Ordnung	Verfahren
1	Euler-Cauchy
2	Halbschritt, Heun
3	3-stufiges Heun, Verfahren von Kutta
4	klassisches Runge-Kutta-Verfahren

Konvergenz

Während bei der Konsistenz nur der Fehler bei einem Iterationsschritt interessiert, untersucht man bei der Konvergenz das Verhalten der Näherungslösung auf einem vorgegebenen Intervall $[t_0, b]$. Dabei ist zu beachten, dass die Einschrittverfahren nur Näherungen $y^{(k)}$ in Gitterpunkten t_k liefern (die von der gewählten Schrittweite h abhängen, z.B. $t_k = t_{k-1} + h$), während die wahre Lösung $x(t)$ eine stetige Funktion auf $[t_0, b]$ ist. Deshalb wird der Fehler nur in den Gitterpunkten untersucht.

Zu jedem $h > 0$ betrachtet man die Gitterpunkte $t_i^h := t_0 + i\cdot h,\ i = 1, \ldots, R$. Dabei sei $R = R(h) \in \mathbb{N}$ so gewählt, dass $t_{R-1} < b \leq t_R$ gilt. Mit einem numerischen Verfahren berechnen wir zur Schrittweite h die Näherungen $y_h^{(i)},\ i = 1, \ldots, R$ und definieren dann den (globalen) Fehler

$$\begin{aligned} e_h^{(i)} \;&:=\; x(t_i^h) \;-\; y_h^{(i)}\,, \quad i = 0, \ldots, R\,, \\ E(h) \;&:=\; \max\left\{ \left\| e_h^{(i)} \right\| \;:\; i = 0, \ldots, R \right\} \end{aligned}$$

mit einer Norm $\|\cdot\|$ auf dem $\mathbb{R}^n$.

Definitionen:

1. Das von der Verfahrensfunktion V erzeugte Einschrittverfahren heißt *konvergent*, falls gilt
$$\lim_{h\to 0} E(h) \;=\; 0$$
für jedes $F \in B_1$, für jeden AW $(t_0, y^{(0)})$ und jedes Intervall $[t_0, b]$.

2. Das Verfahren heißt von der *Konvergenzordnung* p, falls
$$E(h) \;\leq\; c{\cdot}h^p \quad (\text{also } E(h) \;=\; \mathcal{O}(h^p))$$
gilt mit einer von h unabhängigen Konstanten c.

Neben Konvergenzaussagen ist in der Praxis natürlich auch der Rundungsfehler zu berücksichtigen. Eine sehr kleine Schrittweite bedeutet, dass viele Schritte zu rechnen sind, um im Intervall $[t_0, b]$ Näherungen zu bestimmen. Dadurch verstärken sich die Rundungsfehlereinflüsse. Deshalb sollte die Schrittweite auch nicht zu klein gewählt werden. Falls $h_2 < h_1$ gilt und h_2 nicht zu klein ist, so kann man aufgrund der Konvergenz erwarten, dass die Schrittweite h_2 bessere Näherungen liefert als h_1.

Anmerkung: Bei expliziten Runge-Kutta-Verfahren folgt aus der Konsistenz bereits die Konvergenz.

6.5 Schrittweitensteuerung

Bisher haben wir stets mit einer festen Schrittweite h gerechnet. In der Praxis kommen jedoch Verfahren zum Einsatz, bei denen in jedem Schritt die Schrittweite h neu gewählt wird.

Der Formelfehler setzt sich zusammen aus dem Fortpflanzungsfehler und dem Schrittfehler. Der Fortpflanzungsfehler lässt sich bei der Ausführung des $(k+1)$-ten Schrittes nicht kontrollieren, da er durch die bisherige Rechnung entstanden ist. Dagegen kann man den Schrittfehler ungefähr schätzen.

Grundlage für die Fehlerschätzungen bilden asymptotische Entwicklungen des Schrittfehlers nach h-Potenzen. Im Gitterpunkt t_k sei ein Näherungswert $y^{(k)}$ bereits berechnet und akzeptiert. Dann betrachten wir die Anfangswertaufgabe

$$\dot{x} \;=\; F(x)\,, \qquad x(t_k) \;=\; y^{(k)} \qquad , \tag{59}$$

deren wahre Lösung mit $u(t)$ bezeichnet wird. Wir rechnen mit der bisherigen Schrittweite h einen Schritt:

$$y_h^{(k+1)} \;=\; y^{(k)} + hV(h, y^{(k)}) \quad . \tag{60}$$

Besitzt das Verfahren die Konsistenzordnung p und gilt $F \in B_{p+1}$, so erhält man für den Schrittfehler

$$u(t_k + h) - y_h^{(k+1)} \;=\; d{\cdot}h^{p+1} \;+\; \mathcal{O}(h^{p+2}) \tag{61}$$

mit einer Konstanten d, die zwar vor t_k und $y^{(k)}$ aber nicht von h abhängt.

Unser Ziel ist es nun, das Fehlerhauptglied $d \cdot h^{p+1}$ näherungsweise zu berechnen und

dann die Schrittweite h so zu wählen, dass

$$\varepsilon_1 \;\leq\; |d \cdot h^{p+1}| \;\leq\; \varepsilon_2$$

erfüllt ist für zwei vorgegebene Schranken $0 < \varepsilon_1 \;<\; \varepsilon_2$.

Schätzung des Schrittfehlers

Neben $y_h^{(k+1)}$ berechnen wir einen weiteren Näherungswert $y_{h/2}^{(k+1)}$ für $u(t_k + h)$, und zwar zwei Schritte mit der halben Schrittweite:

$$\begin{aligned} z^{(k+1)} &:= y^{(k)} + \frac{h}{2} V\left(\frac{h}{2}, y^{(k)}\right) \\ y_{h/2}^{(k+1)} &:= z^{(k+1)} + \frac{h}{2} V\left(\frac{h}{2}, z^{(k+1)}\right) \quad . \end{aligned} \tag{62}$$

Mit Hilfe der beiden Näherungen $y_h^{(k+1)}$ und $y_{h/2}^{(k+1)}$ lässt sich das Fehlerhauptglied in (61) ungefähr angeben.

Satz: Gegeben sei ein RKV der Konsistenzordnung p und $F \in B_{p+1}$. Dann gilt für die oben berechneten Näherungslösungen

$$u(t_k + h) \;-\; y_h^{(k+1)} \;=\; \frac{y_{h/2}^{(k+1)} - y_h^{(k+1)}}{1 - 2^{-p}} \;+\; \mathcal{O}(h^{p+2}) \quad .$$

Wir nennen dieses Ergebnis ohne Beweis.

Korollar: Bildet man den Näherungswert

$$y^{(k+1)} \;:=\; y_h^{(k+1)} \;+\; \frac{y_{h/2}^{(k+1)} \;-\; y_h^{(k+1)}}{1 - 2^{-p}} \;=\; \frac{2^p y_{h/2}^{(k+1)} \;-\; y_h^{(k+1)}}{2^p \;-\; 1} \quad ,$$

so gilt

$$u(t_k + h) \;-\; y^{(k+1)} \;=\; \mathcal{O}(h^{p+2}) \quad .$$

Anmerkungen:

1. $y^{(k+1)}$ ist ebenfalls eine Näherung an die exakte Lösung $u(t_k + h)$.
2. Hier liegt ein allgemeines Prinzip (Extrapolation) zugrunde: Aus zwei Nähungswerten für eine gesuchte Größe bekommt man ohne großen Rechenaufwand eine neue (bessere) Näherung. Wir werden in Abschnitt 7.4 auf dieses Prinzip ausführlich eingehen.

Praktische Durchführung der Schrittweitensteuerung

Man nutzt die obigen Ergebnisse, um eine Schrittweitensteuerung in die Runge-Kutta-Verfahren einzubauen. Dabei ist auch darauf zu achten, dass man die Schrittweite wieder größer macht, sofern es die Fehlerschätzung erlaubt (um in t-Richtung voranzukommen und Rundungsfehlereinflüsse gering zu halten).

Als exemplarischen Fall betrachten wir das klassische Runge-Kutta-Verfahren für eine skalare nichtautonome Dgl. Für die AWA

$$\dot{x} \;=\; f(t, x)\,, \qquad x(t_0) \;=\; y^{(0)}$$

sei eine Näherung für die exakte Lösung x im Punkte b gesucht. Gegeben seien die Anfangsschrittweite h sowie Toleranzgrenzen $\varepsilon_1 < \varepsilon_2$.

Das klassische RKV hat die Konsistenzordnung $p = 4$. Damit gelangen wir zu folgendem **Verfahren**:

1. Setze $k = 0$.
2. Sei $y^{(k)}$ als Näherung im Punkt t_k schon berechnet und akzeptiert, weiter sei h die aktuelle Schrittweite.
   Bestimme mit der Schrittweite h die beiden Näherungen $y_h^{(k+1)}$ und $y_{h/2}^{(k+1)}$ wie in (60) und (62) beschrieben.
3. Berechne dann die Fehlerschätzung
$$E \;:=\; \frac{16}{15}\left|y_{h/2}^{(k+1)} \;-\; y_h^{(k+1)}\right| \quad .$$
4. Gilt $\varepsilon_2 < E$, so setze $h := \frac{h}{2}$ und gehe wieder zu Schritt 2.
5. Falls $E \leq \varepsilon_2$ gilt, so bilde gemäß des obigen Korollars
$$y^{(k+1)} \;=\; \frac{16y_{h/2}^{(k+1)} \;-\; y_h^{(k+1)}}{15}$$
   und $t_{k+1} \;:=\; t_k + h$.
   $y^{(k+1)}$ wird als Näherungswert für $x(t_{k+1})$ akzeptiert.
6. Gilt $E \;<\; \varepsilon_1$, so setzt man noch $h := 2h$.
7. Falls $t_{k+1} < b$ gilt, so setze $k = k + 1$ und gehe zu Schritt 2. Im letzten Schritt ist h so zu wählen, dass $t_{R-1} + h = b$ gilt.

Vergleiche auch Übungen.

6.6 Stabilität

Hier betrachtet man die skalare Test-Differentialgleichung

$$\dot{x} \;=\; \lambda x\,, \qquad x(0) \;=\; a\,, \qquad \lambda < 0 \quad , \tag{63}$$

welche die exakte Lösung $\bar{x}(t) \;=\; a\exp(\lambda t)$ und das Langzeitverhalten

$$\lim_{t\to\infty} \bar{x}(t) \;=\; 0 \tag{64}$$

besitzt.

Diese Anfangswertaufgabe wird mit einem Runge-Kutta-Verfahren mit einer festen Schrittweite $h > 0$ näherungsweise in den Gitterpunkten $t_k \;=\; kh$ gelöst. Man erwartet dann von der Näherungslösung ebenfalls ein abklingendes Verhalten wie in (64).

Zunächst betrachten wir das Euler-Cauchy-Verfahren und bekommen für die obige Anfangswertaufgabe die Näherungen

$$y^{(0)} \;=\; a\,, \qquad y^{(k)} \;=\; y^{(k-1)} + h\lambda y^{(k-1)} \;=\; (1 + h\lambda)y^{(k-1)} \quad .$$

Daraus folgt (durch Induktion)

$$y^{(k)} \;=\; (1+\lambda h)^k y^{(0)} \;=\; (1+\lambda h)^k a \quad . \tag{65}$$

Für ein abklingendes Verhalten muss

$$0 \;=\; \lim_{k\to\infty} y^{(k)} \;=\; \lim_{k\to\infty} (1+\lambda h)^k a$$

gelten und folglich

$$|1+\lambda h| \;<\; 1 \quad .$$

Äquivalent dazu ist $\lambda h \in (-2, 0)$.

Fazit Um beim Euler-Cauchy-Verfahren qualitativ das gleiche Verhalten wie für die exakte Lösung zu erhalten, darf die Schrittweite h nicht zu groß gewählt werden.

Beispiel: Wir lösen die AWA

$$\dot{x} \;=\; -10x \qquad x(0) \;=\; 1$$

näherungsweise mit dem Euler-Cauchy-Verfahren zu verschiedenen Schrittweiten.

Schrittweite h = 0.25

t	Näherung	exakte L.	Fehler
0.250	-1.50000	0.08208500	1.58208
0.500	2.25000	0.00673795	-2.24326
0.750	-3.37500	0.00055308	3.37555
1.000	5.06250	0.00004540	-5.06245
1.250	-7.59375	0.00000373	7.59375
1.500	11.39062	0.00000031	-11.39062
1.750	-17.08594	0.00000003	17.08594
2.000	25.62891	0.00000000	-25.62891
2.250	-38.44336	0.00000000	38.44336
2.500	57.66504	0.00000000	-57.66504
2.750	-86.49756	0.00000000	86.49756
3.000	129.74634	0.00000000	-129.74634
3.250	-194.61951	0.00000000	194.61951
3.500	291.92926	0.00000000	-291.92926
3.750	-437.89389	0.00000000	437.89389
4.000	656.84084	0.00000000	-656.84084
4.250	-985.26125	0.00000000	985.26125
4.500	1477.89188	0.00000000	-1477.89188
4.750	-2216.83782	0.00000000	2216.83782
5.000	3325.25673	0.00000000	-3325.25673

Schrittweite h = 0.0625

t	Näherung	exakte L.	Fehler
0.2500	0.01977539	0.08208500	0.06230961
0.5000	0.00039107	0.00673795	0.00634688
0.7500	0.00000773	0.00055308	0.00054535
1.0000	0.00000015	0.00004540	0.00004525
1.2500	0.00000000	0.00000373	0.00000372
1.5000	0.00000000	0.00000031	0.00000031
1.7500	0.00000000	0.00000003	0.00000003
2.0000	0.00000000	0.00000000	0.00000000
2.2500	0.00000000	0.00000000	0.00000000
2.5000	0.00000000	0.00000000	0.00000000

Mit $W(z) := 1 + z$ erhält man in (65)

$$y^{(k)} \;=\; W(\lambda h) y^{(k-1)} \;=\; W(\lambda h)^k a \quad ,$$

und die Stabilitätsbedingung lautet dann

$$|W(\lambda h)| \;<\; 1 \quad .$$

Die Funktion $W(z)$ heißt *Stabilitätsfunktion* (des Euler-Cauchy-Verfahrens).

Allgemein ordnet man jedem Runge-Kutta-Verfahren

$$\beta \;=\; \left(\begin{array}{ccc} \beta_{11} & \cdots & \beta_{1s} \\ \vdots & & \vdots \\ \beta_{s1} & \cdots & \beta_{s,s} \\ \hline \gamma_1 & \cdots & \gamma_s \end{array}\right)$$

eine Stabilitätsfunktion $W_\beta(z)$ zu und erhält dann für die Näherungen der AWA (63) die Beziehung

$$y^{(k)} \;=\; W_\beta(\lambda h) \cdot y^{(k-1)} \;=\; \cdots \;=\; W_\beta(\lambda h)^k \cdot a \quad .$$

Damit die Näherungslösung dasselbe Langzeitverhalten wie die exakte Lösung besitzt, muss

$$|W_\beta(\lambda h)| \;<\; 1$$

gelten.

Beispiel: Für das klassische Runge-Kutta-Verfahren ergibt sich die Stabilitätsfunktion

$$W_\beta(z) \;=\; 1 + z + \frac{1}{2} z^2 + \frac{1}{6} z^3 + \frac{1}{24} z^4 \quad .$$

Definition: Ein konsistentes RKV heißt *A-stabil* für λh, falls $|W_\beta(\lambda h)| < 1$ gilt.
Die Menge

$$S_\beta \;:=\; \{ z \in \mathbb{R} \;:\; |W_\beta(z)| < 1 \}$$

heißt *Stabilitätsbereich des RKV* β.

Beispiele:

1. Euler-Cauchy-Verfahren: $S_\beta = (-2, 0)$.
2. klass. Runge-Kutta-Verfahren: $S_\beta = (-2.78, 0)$.
3. Rückwärts-Euler-Verfahren: $S_\beta \supset (-\infty, 0)$.

   Hier ist die Stabilitätsfunktion $W_\beta(z) = \frac{1}{1-z}$. Dieses Verfahren ist stabil für jedes $h > 0$.

Anmerkung: Allgemeiner betrachtet man die komplexe Testfunktion

$$\dot{x} \;=\; \lambda x \quad \text{mit } \lambda \in \mathbb{C} \text{ und } \mathrm{Re}(\lambda) < 0 \quad .$$

Dann ergeben sich komplexe Stabilitätsfunktionen und komplexe Stabilitätsbereiche.

6.7 Lösung von Anfangswertaufgaben mit Matlab

Explizite Verfahren

In Matlab gibt es zwei Funktionen, welche Anfangswertaufgaben mit Hilfe von expliziten Runge-Kutta-Verfahren lösen. Im Standardfall verwendet man `ode45`. Ist nur eine grobe Genauigkeit der Näherungslösung gewünscht, so wird die Funktion `ode23` empfohlen, da sie weniger Rechenaufwand erfordert. Beide Routinen unterscheiden sich in ihrer programmiertechnischen Handhabung nicht. Deshalb beschreiben wir hier nur `ode45`.

Sie löst den folgenden Typ von Anfangswertaufgaben:

$$\begin{array}{rcl} \dot{y}_1 &=& f_1(t, y_1, \ldots, y_n)\,, \quad y_1(t_0) = y_1^{(0)} \\ &\vdots& \qquad\qquad\qquad\qquad \vdots \\ \dot{y}_n &=& f_n(t, y_1, \ldots, y_n)\,, \quad y_n(t_0) = y_n^{(0)} \end{array}$$

Aufruf

Die Matlab-Funktion `ode45` hat die Syntax

```
[T,Y] = ode45(odefun,tspan,y0)
```

.

Dabei ist `odefun` eine vektorwertige Funktion für die rechte Seite der Differentialgleichung. `tspan` gibt das Zeitintervall $[t_0, b]$ an, in dem eine Näherungslösung gesucht wird. Der Vektor `y0` enthält die Anfangswerte $(y_1^{(0)}, \ldots, y_n^{(0)})$.

Nach dem Aufruf der Funktion enthält der Vektor `T` die Zeitpunkte $t_0, t_1, \ldots, t_R$, in denen eine Näherungslösung berechnet wurde. `Y` ist eine Matrix und enthält in der k-ten Zeile die Näherungslösung $y^{(k)} = (y_1^{(k)}, \ldots, y_n^{(k)})$ zum Zeitpunkt t_k.

Als Beispiel lösen wir die Anfangswertaufgabe

$$\begin{array}{rcrcl} \dot{y}_1 &=& -y_2, & y_1(0) &=& 1 \\ \dot{y}_2 &=& y_1, & y_2(0) &=& 0 \end{array} \quad .$$

Die exakte Lösung dieser Anfangswertaufgabe ist $y(t) = (y_1(t), y_2(t)) = (\cos(t), \sin(t))$.

Zuerst erstellen wir eine Matlab-Funktion für die Differentialgleichung, welche in die Datei `dgl1.m` gespeichert wird:

```
function dy = dgl1(t,y)
  dy = zeros(size(y));
  dy(1) = -y(2);
  dy(2) = y(1);
```

Beachten Sie die Zeile `dy = zeros(size(y));`, damit bekommt `dy` die gleiche Dimension wie `y`. Danach geben wir

```
[T,Y] = ode45(@dgl1,[0,10],[1 0])
```

ein. Das Phasenbild erhalten wir mittels

```
plot(Y(:,1),Y(:,2))
```

und das Zeitbild durch die Sequenz

```
plot(T,Y(:,1),'r');
hold on;
plot(T,Y(:,2),'g')
```

Sind die Näherungslösungen zu gewissen Zeitpunkten $t_0, t_1, \ldots t_R$ gesucht, so müssen diese als Vektor in der Variablen `tspan` angegeben werden:

```
tspan = 0:1:10;
[T,Y] = ode45(@dgl1,tspan,[1 0]);
text = sprintf('%6.2f   %8.4f   %8.4f \n',[T,Y]');
disp(text)
```

Dieses Programm liefert die Ausgabe

```
 0.00    1.0000    0.0000
 1.00    0.5403   -0.8416
 2.00   -0.4164   -0.9094
 3.00   -0.9900   -0.1409
 4.00   -0.6533    0.7570
 5.00    0.2842    0.9588
 6.00    0.9604    0.2787
 7.00    0.7534   -0.6576
 8.00   -0.1463   -0.9890
 9.00   -0.9114   -0.4111
10.00   -0.8383    0.5448
```

Das `sprintf`-Kommando gibt eine Matrix **spaltenweise** aus. Deshalb müssen wir hier die transponierte Matrix `[T,Y]'` angeben.

Weitere Optionen

Daneben können weitere Optionen gesetzt werden, um etwa die Genauigkeit oder die Ausgabe der Näherungslösung zu verändern. Welche Optionen es insgesamt gibt und welche Voreinstellungen verwendet werden, erfahren Sie durch Eingabe von `odeset`. Diese Funktion wird auch verwendet, um gewisse Parameter zu setzen und hat dann die Form

```
options = odeset('name1',wert1,'name2',wert2,...)
```

.

So werden durch

```
options = odeset('RelTol',1e-6,'AbsTol',1e-8)
[T,Y] = ode45(@dgl1,[0,10],[1 0],options)
```

die Genauigkeitsschranken gesetzt. Die Art der Ausgabe wird durch `OutputFcn` gesteuert: durch

```
options=odeset('OutputFcn','odeplot');
[T,Y] = ode45(@dgl1,[0,10],[1 0],options)
```

wird das Zeitbild gezeichnet. Das Phasenbild erhält man durch

```
options=odeset('OutputFcn','odephas2');
```

Besitzt die Differentialgleichung selbst Parameter, so haben wir zwei Möglichkeiten.

– Übergabe als zusätzliche Parameter an die `ode45`-Routine:

```
[T,Y] = ode45(@dgl1,[0,10],[1 0],options,parameter)
```

Diese müssen dann auch bei der Funktionsdefinition auftreten:

```
function dy = dgl1(t,y,parameter)
```

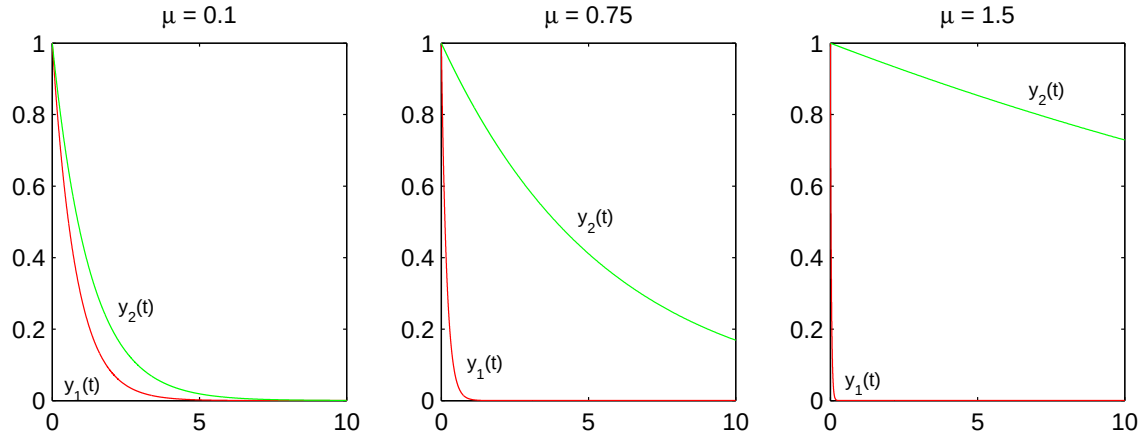
– Verwendung der Matlab-Anweisung `global` . Informationen dazu erhalten Sie in Matlab durch das Kommando `doc global`.

Implizite Verfahren

Die bisher behandelten Matlab-Funktionen `ode45` bzw. `ode23` beruhen auf relativ einfachen numerischen Verfahren (expliziten Runge-Kutta-Verfahren). Bei Differentialgleichungen mit gewissen Eigenschaften (den sogenannten *steifen* Differentialgleichungen) sind diese Verfahren nicht empfehlenswert. Wir demonstrieren dies anhand eines Beispiels:

$$\begin{array}{rclcrcl} \dot{y}_1 &=& -10^{\mu} y_1 + 0.0001 y_2 &,& y_1(0) &=& 1 \\ \dot{y}_2 &=& -10^{-\mu} y_2 &,& y_2(0) &=& 1 \end{array} \quad . \tag{66}$$

mit einem Parameter $\mu \in \mathbb{R}$. Mit Hilfe von `ode45` wird diese Anfangswertaufgabe für verschiedene μ's gelöst:



Wir erkennen, dass für größere μ die 1. Lösungskomponente $y_1(t)$ rasch gegen Null geht, die Komponente $y_2(t)$ hingegen sehr langsam. Differentialgleichungen mit stark unterschiedlich rasch abklingenden Lösungskomponenten werden als *steif* bezeichnet. Das obige Beispiel ist also für größere μ eine steife Differentialgleichung. Für solche sind explizite Runge-Kutta-Verfahren (und damit die Matlab-Funktionen `ode23` bzw. `ode45`) ungeeignet: die Schrittweitensteuerung wird hier eine sehr kleine Schrittweite wählen. Dies bedeutet dann sehr viele Rechenschritte, bis das vorgegebene Zeitintervall durchlaufen ist. Die folgende Tabelle demonstriert dies für die Lösung der obigen Anfangswertaufgabe im Zeitintervall $[0, 10]$ eindrucksvoll:

μ	0.5	1	2	3	4	5
Anzahl der Schritte	1789	1745	2721	13569	122041	1206741
Rechenzeit	1.3 sec	1.3 sec	2.0 sec	10.5 sec	105 sec	2028 sec

Für steife Differentialgleichungen empfiehlt sich daher die Verwendung von impliziten Näherungsmethoden. Hierzu bietet Matlab zwei Funktionen an.

Die Matlab-Funktionen `ode15s` und `ode23t`

Die Standard-Routine für steife Differentialgleichungen ist die Funktion `ode15s`. Ist die Steifheit nicht zu hoch, so kann man auch `ode23t` verwenden, welche auf einem einfacheren impliziten Verfahren beruht. Die Syntax dieser beiden Routinen ist dieselbe wie

bei der bereits behandelten Matlab-Funktion `ode45`:

```
[T,Y] = ode15s(odefun,tspan,y0,options)   .
```

Als Beispiel lösen wir die Anfangswertaufgabe (66). Dazu erstellt man zunächst eine Matlab-Funktion für die Differentialgleichung, welche in `dgl2.m` gespeichert wird:

```
function dy = dgl2(t,y,mu)
dy = zeros(2,1);     % dy muss ein Spaltenvektor sein
dy(1) = (-10.^mu).*y(1) + 0.0001.*y(2);
dy(2) = (-10.^(-mu)).*y(2);
```

Dann wird noch die gewünschte Genauigkeit festgelegt:

```
options = odeset('RelTol',1e-12,'AbsTol',1e-12);    .
```

Nun können wir für verschiedene Parameter μ diese Anfangswertaufgabe mit Hilfe der Matlab-Funktion `ode15s` lösen und die Ergebnisse mit der obigen Tabelle vergleichen:

```
mu = 0.5;
[T,Y] = ode15s(@dgl2,[0,10],[1,1],options,mu);
```

μ	0.5	1	2	3	4	5
Anzahl der Schritte	649	656	634	633	635	639
Rechenzeit	1.8 sec	1.8 sec	1.8 sec	1.8 sec	1.8 sec	1.8 sec

Mit der Matlab-Funktion `odeset` können eine ganze Reihe von Optionen gesetzt werden. Durch den Befehl

```
doc odeset
```

erhalten Sie dazu ausführliche Informationen.

7 Numerische Integration

7.1 Einleitung

In diesem Paragraphen befassen wir uns mit der näherungsweisen Berechnung eines Integrals. Zu $f \in C[a,b]$ existiert das Integral

$$\int_a^b f(t)\,dt \quad . \tag{67}$$

Ist eine Stammfunktion $F(t)$ zu $f(t)$ bekannt, so gilt (Hauptsatz der Differential- und Integralrechnung)

$$\int_a^b f(t)\,dt \;=\; F(b) - F(a) \quad ,$$

und man kann diese Formel zur Integralberechnung heranziehen (z.B. günstig, falls f ein Polynom ist).

Es gibt jedoch auch Situationen, wo diese Vorgehensweise nicht möglich bzw. nicht praktisch ist:

- Die Stammfunktion ist nicht in expliziter Form bekannt, z.B. für $f(t) := \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}}$ (Dichtefunktion der Standard-Normalverteilung).
- Eine Stammfunktion ist zwar in expliziter Form bekannt, sie ist jedoch von sehr komplizierter Bauart. Dies tritt häufig bei rationalen Funktionen auf.
- Der Integrand f ist nur in gewissen Punkten $a \le t_0 < t_1 < \cdots < t_m \le b$ gegeben, z.B. als Ergebnis von Messungen.

Aus diesem Grunde benötigen wir Näherungsverfahren zur Berechnung des Integrals (67). Solche Näherungsformeln haben die Gestalt

$$\sum_{i=0}^{m} w_i f(t_i)$$

mit $a \le t_0 < t_1 < \cdots < t_m \le b$ und $w_i \in \mathbb{R}$. Diesen Ausdruck nennt man eine *Quadratursumme*, die w_i heißen die *Gewichte* und die t_i die *Stützstellen* der Quadratursumme. Die Gewichte hängen nicht vom Integranden ab.
Den Fehler bezeichnen wir mit $R[f]$, schreiben

$$\int_a^b f(t)\,dt \;=\; \sum_{i=0}^{m} w_i f(t_i) \;+\; R[f] \tag{68}$$

und nennen diese Darstellung auch eine *Quadraturformel*.

Anmerkung: Der Quadraturfehler $R : C[a,b] \to \mathbb{R}$ ist ein lineares Funktional, d.h

$$R[\alpha\cdot f + \beta\cdot g] \;=\; \alpha\cdot R[f] \;+\; \beta\cdot R[g] \quad \text{für alle } f,g \in C[a,b],\ \alpha,\beta \in \mathbb{R}\,.$$

Definitionen:

1. Die Quadraturformel heißt *exakt* für ein $g \in C[a,b]$, falls $R[g] = 0$ gilt.
2. Sie hat den *Exaktheitsgrad* n ($n \in \mathbb{N}$), falls gilt

$$R[q] = 0 \quad \text{für alle } q \in \Pi_n \qquad \text{und} \qquad R[\bar{q}] \neq 0 \quad \text{für ein } \bar{q} \in \Pi_{n+1} \ .$$

Für die Numerik kommt es darauf an, die Stützstellen und die Gewichte "geschickt" zu wählen, damit die Quadratursumme einen guten Näherungswert für das Integral liefert. Hier kommen drei Methoden in Betracht:

- Der Integrand f wird durch ein Polynom p angenähert (approximiert), das Integral über p wird als Näherungswert für das gesuchte Integral (67) akzeptiert. Solche Formeln bezeichnet man als *interpolatorische Quadraturformeln*. Das Integral von Polynomen lässt sich leicht berechnen.
- Das Intervall $[a,b]$ wird in Teil-Intervalle zerlegt; für jedes Teil-Intervall wird eine (einfache) Quadratursumme verwendet. Diese Formeln heißen *zusammengesetzte Quadraturformeln*.
- Durch geeignete Kombination von (zusammengesetzten) Quadratursummen werden neue gewonnen, die i.A. bessere Näherungen an das Integral darstellen (*Romberg-Verfahren*).

Affine Transformation

Ist eine Quadraturformel für das Intervall $[a,b]$ gegeben, so erhält man daraus durch affine Transformation eine Quadraturformel für das Intervall $[c,d]$.

Sei $g \in C[c,d]$ gegeben. Betrachte die affine Transformation

$$\varphi : [a,b] \to [c,d]\,, \ \varphi(t) := \alpha t + \beta \quad \text{mit} \quad \varphi(a) = c, \ \varphi(b) = d \ .$$

Durch die beiden Bedingungen werden α und β eindeutig bestimmt:

$$\alpha = \frac{d-c}{b-a} \qquad \text{und} \qquad \beta = \frac{bc-ad}{b-a} \quad .$$

Dann gilt $f := g \circ \varphi \in C[a,b]$ und

$$\int_c^d g(x)\,dx \ = \ \int_a^b g(\varphi(t))\,\varphi'(t)\,dt \ = \ \alpha \int_a^b g(\alpha t + \beta)\,dt \quad .$$

Für die Funktion $f(t) = g(\varphi(t))$ wenden wir die Quadraturformel (68) bezüglich des Intervalls $[a,b]$ an und erhalten eine Formel für das Intervall $[c,d]$:

$$\begin{aligned} \int_c^d g(x)\,dx \ &= \ \alpha \sum_{i=0}^{m} w_i\, g(\alpha t_i + \beta) \ + \ R[g] \\ &= \ \sum_{i=0}^{m} \bar{w}_i\, g(x_i) \ + \ R[g] \end{aligned}$$

mit $\bar{w}_i := \alpha w_i$ und $x_i := \alpha t_i + \beta$.

7.2 Interpolatorische Formeln

Herleitung

Es seien $f \in C[a,b]$ und $a \le t_0 < \cdots < t_m \le b$ gegeben. Bilde das Interpolationspolynom zu f bezüglich dieser Stützstellen:

$$p(t) \;=\; \sum_{i=0}^{m} f(t_i)\, l_i(t) \quad ,$$

wobei die l_i wieder die Lagrange-Grundpolynome sind (vgl. Abschnitt 4.2). Da das Integral über Polynome leicht berechnet werden kann, bietet es sich an, den Integranden f durch das Interpolationspolynom p zu ersetzen. Man erhält dann

$$\begin{aligned}
\int_a^b f(t)\,dt &= \int_a^b p(t)\,dt \;+\; R[f] \\
&= \int_a^b \left(\sum_{i=0}^{m} f(t_i)\, l_i(t) \right) dt \;+\; R[f] \\
&= \sum_{i=0}^{m} \left(f(t_i) \int_a^b l_i(t)\,dt \right) \;+\; R[f] \\
&= \sum_{i=0}^{m} w_i\, f(t_i) \;+\; R[f] \quad \text{mit} \;\; w_i := \int_a^b l_i(t)\,dt \quad .
\end{aligned}$$

Definition: Eine solche Quadraturformel heißt eine *interpolatorische Quadraturformel*.

Satz: Eine interpolatorische Quadraturformel (mit $m+1$ Stützstellen) besitzt mindestens den Exaktheitsgrad m.

Beweis: Sei $q \in \Pi_m$ und p das zugehörige Interpolationspolynom. Aufgrund der Eindeutigkeit der Interpolation stimmt p mit q überein. □

Anmerkung: Für eine interpolatorische Quadraturformel gilt $\displaystyle\sum_{i=0}^{m} w_i \;=\; b-a \quad .$

Einige elementare Formeln

Im Folgenden geben wir einige einfache interpolatorische Quadraturformeln an. Zur Abkürzung wird noch $h_i(t) := t^i$ (Monome) gesetzt.

Mittelpunktsregel: $m := 0, t_0 := \frac{a+b}{2}$.

$$\int_a^b f(t)\,dt \;=\; (b-a) f\left(\frac{a+b}{2}\right) \;+\; R[f] \quad . \tag{69}$$

Exaktheitsgrad: $M = 1$, denn man verifiziert sofort $R[h_1] = 0$ und $R[h_2] \neq 0$.

Quadraturfehler: Sei $f \in C^2[a,b]$. Dann gibt es ein $\xi \in (a,b)$ mit

$$R[f] \;=\; \frac{1}{24}(b-a)^3 f^{(2)}(\xi) \quad .$$

Auf die Bestimmung des Quadraturfehlers kommen wir in Abschnitt 7.7 zurück.

Trapezregel: $m = 1,\ t_0 := a,\ t_1 := b$

$$\int_a^b f(t)\,dt \;=\; \frac{b-a}{2}\,(f(a)+f(b)) \;+\; R[f] \quad . \tag{70}$$

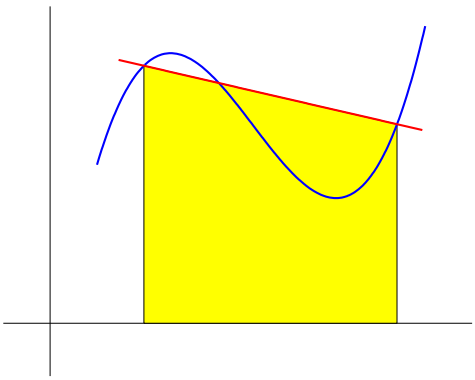
Denn für die Gewichte erhält man $w_1 = b - a - w_0$ und

$$w_0 \;=\; \int_a^b l_0(t)\,dt \;=\; \int_a^b \frac{t-b}{a-b}\,dt \;=\; \frac{1}{2(a-b)}\,(t-b)^2\Big|_a^b \;=\; \frac{b-a}{2} \quad .$$

Exaktheitsgrad: $M = 1$

Quadraturfehler: Sei $f \in C^2[a,b]$. Dann gilt

$$R[f] \;=\; -\frac{(b-a)^3}{12}\,f^{(2)}(\xi) \qquad \text{für ein } \xi \in (a,b) \quad . \tag{71}$$



Die Quadratursumme liefert den Flächeninhalt eines Trapezes.

Keplersche Fassregel: (Johannes Kepler, 1571-1630)
$m = 2,\ t_0 := a,\ t_1 := \frac{a+b}{2},\ t_2 := b$

$$\int_a^b f(t)\,dt \;=\; \frac{b-a}{6}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right) \;+\; R[f] \quad . \tag{72}$$

Exaktheitsgrad: $M = 3$

Quadraturfehler: Sei $f \in C^4[a,b]$. Dann gilt

$$R[f] \;=\; -\frac{(b-a)^5}{2880}\,f^{(4)}(\xi) \qquad \text{für ein } \xi \in (a,b) \quad . \tag{73}$$

Newton-Cotes-Formeln: (Roger Cotes, 1682-1716; Isaac Newton, 1642-1727)
Die Trapezregel und die Keplersche Fassregel sind Sonderfälle einer allgemeineren Klasse von Quadraturformeln: Für jedes $m \in \mathbb{N}$ wählt man die Stützstellen äquidistant, also

$$t_i \;:=\; a + i\,\frac{b-a}{m} \qquad (i = 0, 1, \ldots, m)$$

und bildet die zugehörige interpolatorische Quadraturformel. Diese heißen *Newton-Cotes-Formeln*.

So liefert $m = 1$ die Trapezregel und $m = 2$ die Keplersche Fassregel (weitere Beispiele finden Sie in Schwarz [2]).

7.3 Zusammengesetzte Quadraturformeln

Hier wird das Intervall $[a, b]$ in Teil-Intervalle zerlegt, für jedes Teil-Intervall wird eine einfache Quadraturformel verwendet. Dazu sei $a = x_0 < x_1 < \cdots < x_{N-1} < x_N = b$. Für das Integral gilt dann

$$\int_a^b f(t)\,dt \;=\; \sum_{i=1}^{N} \int_{x_{i-1}}^{x_i} f(t)\,dt \qquad . \tag{74}$$

Hier empfiehlt es sich zunächst, die Teil-Intervalle alle gleich groß zu machen und jeweils dieselbe Quadraturformel zu verwenden. Sei also

$$x_i \;:=\; a + i\,\frac{b-a}{N} \qquad (i = 0, \ldots, N).$$

Die Trapezregel für das Teil-Intervall $[x_{i-1}, x_i]$ lautet dann

$$\int_{x_{i-1}}^{x_i} f(t)\,dt \;=\; \frac{b-a}{2N}\,(f(x_{i-1}) + f(x_i)) \;+\; R_i[f]$$

(beachte $x_i - x_{i-1} = \frac{b-a}{N}$) mit

$$R_i[f] \;=\; -\frac{(b-a)^3}{12N^3}\,f^{(2)}(\xi_i) \qquad \text{für ein } \xi_i \in (x_{i-1}, x_i) \quad .$$

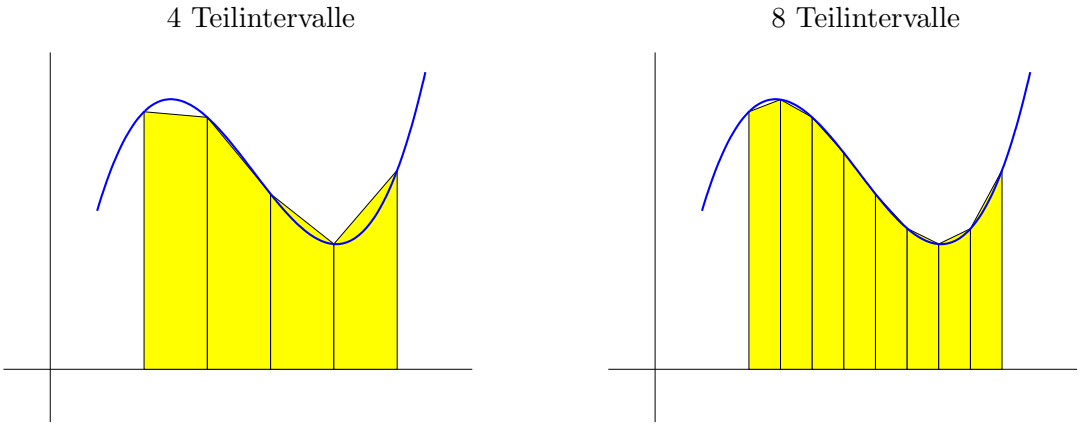
Für das Gesamt-Intervall ergibt sich mit (74)

$$\int_a^b f(t)\,dt \;=\; \frac{b-a}{N}\left(\frac{1}{2}f(a) + \frac{1}{2}f(b) + \sum_{i=1}^{N-1} f(x_i)\right) \;+\; R[f] \qquad . \tag{75}$$

Definition: Diese Quadraturformel heißt *zusammengesetzte Trapezregel*.

Satz: Sei $f \in C^2[a, b]$. Dann gilt für den Quadraturfehler bei der zusammengesetzten Trapezregel mit N Teilintervallen

$$R[f] \;=\; -\frac{(b-a)^3}{12N^2}\,f^{(2)}(\xi) \qquad \text{für ein } \xi \in (a, b). \tag{76}$$



Anwendung: Sei $f \in C^2[a,b]$ und $\varepsilon > 0$ vorgegeben. Man berechne für das Integral

$$J \;:=\; \int_a^b f(t)\,dt$$

einen Näherungswert I, so dass $|J - I| < \varepsilon$ gilt.

Lösung: Wähle N so, dass

$$\frac{(b-a)^3 \left\| f^{(2)} \right\|}{12N^2} \;<\; \varepsilon$$

erfüllt ist und bilde dann die zusammengesetzte Trapezregel mit N Teil-Intervallen. Dabei werden Rundungsfehler vernachlässigt.

Die Verwendung der Mittelpunktsregel (69) als Grundregel liefert

$$\int_a^b f(t)\,dt \;=\; \frac{b-a}{N} \sum_{j=1}^{N} f\left(a + \frac{2j-1}{2N}(b-a)\right) \;+\; R[f] \tag{77}$$

(*zusammengesetzte Mittelpunktsregel*).

Für den Fehler bei der zusammengesetzten Mittelpunktsregel gilt (für $f \in C^2[a,b]$)

$$R[f] \;=\; \frac{(b-a)^3}{24N^2}\, f^{(2)}(\xi) \qquad \text{mit einem } \xi \in (a,b) \quad .$$

Für die zusammengesetzte Keplersche Fassregel erhält man (für $f \in C^4[a,b]$)

$$R[f] \;=\; -\frac{(b-a)^5}{2880N^4}\, f^{(4)}(\xi) \qquad \text{mit einem } \xi \in (a,b) \quad .$$

7.4 Das Prinzip der Extrapolation

Sei $\varphi : [-a,a] \to \mathbb{R}$ eine reelle Funktion mit der Eigenschaft

$$\varphi(-h) \;=\; \varphi(h) \qquad \text{für alle } h \in [-a,a]\,.$$

Weiter seien

$$0 < h_m < h_{m-1} < \cdots < h_0 \leq a$$

gegeben. Gesucht ist ein Näherungswert für $\varphi(0)$.

Man bildet das Interpolationspolynom $p(t)$ zu den Stützpaaren

$$\left(h_i^2, \varphi(h_i)\right) \ , \quad i = 0, \ldots, m$$

und verwendet $p(0)$ als Näherungwert für $\varphi(0)$. Die Durchführung erfolgt dabei mit dem Neville-Algorithmus (vgl. Abschnitt 4.7 über Numerische Differentiation).

Definition: Diese Vorgehensweise nennt man *Extrapolation.*

Anmerkung: Es handelt sich also um eine Interpolation, wobei jedoch das IP-Polynom an einer Stelle ausgewertet wird, die nicht im Bereich der Stützstellen liegt.

Sei φ nun von der Gestalt

$$\varphi(h) \ = \ \tau_0 + \tau_1 h^2 + \tau_2 h^4 + \tau_3 h^6 + \cdots \tag{78}$$

(mit $\tau_i \in \mathbb{R}$; Entwicklung in h^2-Potenzen). Dann ist $\tau_0 = \varphi(0)$ der gesuchte Wert.

Weiter seien $h_0, h_1 \in \mathbb{R}$ gegeben mit $h_0 > h_1$, also $h_1 = \alpha h_0$ mit $\alpha := \frac{h_1}{h_0}$. Man erhält

$$\begin{aligned} p_0 \ = \ \varphi(h_0) \ &= \ \tau_0 + \tau_1 h_0^2 + \tau_2 h_0^4 + \tau_3 h_0^6 + \cdots \quad , \\ p_1 \ = \ \varphi(h_1) \ &= \ \tau_0 + \tau_1 h_1^2 + \tau_2 h_1^4 + \tau_3 h_1^6 + \cdots \quad . \end{aligned}$$

Der Neville-Algorithmus (vgl. (27)) liefert

$$\begin{aligned} p_{01} = p_{01}(0) \ &= \ \frac{(0 - h_1^2)p_0 - (0 - h_0^2)p_1}{h_0^2 - h_1^2} \\ &= \ \frac{h_0^2 p_1 - h_1^2 p_0}{h_0^2 - h_1^2} \\ &= \ \frac{h_0^2(\tau_0 + \tau_1 h_1^2 + \tau_2 h_1^4 + \tau_3 h_1^6 + \cdots) - h_1^2(\tau_0 + \tau_1 h_0^2 + \tau_2 h_0^4 + \tau_3 h_0^6 + \cdots)}{h_0^2 - h_1^2} \\ &= \ \tau_0 - \tau_2 h_0^2 h_1^2 - \cdots \\ &= \ \tau_0 + \bar{\tau}_2 h_0^4 + \bar{\tau}_3 h_0^6 + \cdots \end{aligned}$$

(mit gewissen $\bar{\tau}_i \in \mathbb{R}$, die sich aus der obigen Beziehung genau angeben lassen, z.B. $\bar{\tau}_2 = -\alpha^2 \tau_2$). Somit gilt

$$p_{01} - \varphi(0) \ = \ p_{01} - \tau_0 \ = \ \bar{\tau}_2 h_0^4 + \bar{\tau}_3 h_0^6 + \cdots \quad .$$

Folgerung: Für betragsmäßig kleines h_0 ist p_{01} eine bessere Näherung an $\varphi(0)$ als p_0.

Dieses Prinzip lässt sich fortsetzen: Seien $h_0 > h_1 > h_2 > 0$ und $\varphi(h)$ wie in (78). Dann liefert der Neville-Algorithmus

$$p_{012} - \varphi(0) \ = \ p_{012} - \tau_0 \ = \ \hat{\tau}_3 h_0^6 + \hat{\tau}_4 h_0^8 + \cdots \quad \text{mit gewissen } \hat{\tau}_i \in \mathbb{R}.$$

usw.

7.5 Das Romberg-Verfahren

Das Extrapolationsprinzip lässt sich auf die zusammengesetzte Trapezregel anwenden (mit $N_1 < N_2$ Teil-Intervallen). Dabei muss man zunächst eine Funktion $\varphi(h)$ finden mit einer Entwicklung der Form (78).

Wir betrachten die zusammengesetzte Trapezregel mit N Teil-Intervallen:

$$T(h) \;:=\; h\left(\frac{1}{2}f(a)+\frac{1}{2}f(b)+\sum_{i=1}^{N-1} f(x_i)\right) \tag{79}$$

mit $h := \frac{b-a}{N}$ und $x_i := a+ih$. Beachten Sie, dass $T(h)$ zunächst nur für gewisse positive h definiert ist. Unser Ziel ist eine Entwicklung von $T(h)$ der Form (78):

$$T(h) \;=\; \tau_0 + \tau_1 h^2 + \tau_2 h^4 + \tau_3 h^6 + \cdots \quad \text{mit} \quad \tau_0 \;=\; \int_a^b f(t)\,dt \quad . \tag{80}$$

Die Summenformel von Euler-MacLaurin

(Leonhard Euler 1707 - 1783; Colin MacLaurin, 1698 - 1746)

Wir betrachten zunächst das Intervall $[0,1]$, setzen $B_0(t) := 1$, $B_1(t) := t - \frac{1}{2}$ und definieren weitere Polynome $B_k(t)$ vom Grad k durch die Bedingungen

$$\begin{aligned} B'_{k+1}(t) &= (k+1)B_k(t) \qquad (k=1,2,\ldots)\,, \\ B_{2k+1}(0) &= B_{2k+1}(1) \;=\; 0 \quad \text{für } k \geq 1 \quad , \end{aligned}$$

welche die $B_k(t)$ eindeutig festgelegen. So erhält man zum Beispiel

$$B_2(t) \;=\; t^2 - t + \frac{1}{6}\;, \quad B_3(t) \;=\; t^3 - \frac{3}{2}t^2 + \frac{1}{2}t\;.$$

Definition: Die Polynome $B_k(t)$ heißen *Bernoulli-Polynome*. Die Zahlen $B_k := B_k(0)$ nennt man *Bernoulli-Zahlen* (Jacob Bernoulli, 1654 - 1705).

Anmerkung: $B_{k+1}(t)$ ist bis auf einen Faktor eine Stammfunktion von $B_k(t)$. Für die Bernoulli-Polynome gilt ferner $B_k(1-t) = (-1)^k B_k(t)$, d.h. Symmetrie bezüglich $t = \frac{1}{2}$. Insbesondere hat man für alle $k \geq 2$ die Beziehung $B_k(1) = B_k(0) = B_k$.

Sei nun $f \in C^{2m+2}[a,b]$ für ein gewisses $m \in \mathbb{N}$. Dann gilt für die zusammengesetzte Trapezregel mit N Teilintervallen (d.h. $h = \frac{b-a}{N}$):

$$\begin{aligned} \int_a^b f(t)\,dt \;&=\; h\left(\frac{f(a)}{2}+\frac{f(b)}{2}+\sum_{i=1}^{N-1} f(a+ih)\right) \;+\; \sum_{k=1}^{m} h^{2k}\frac{B_{2k}}{(2k)!}\left(f^{(2k-1)}(a)-f^{(2k-1)}(b)\right) \\ &+\; h^{2m+2}\hat{r}_{m+1} \quad . \end{aligned}$$

Definition: Diese Darstellung heißt *Summenformel von Euler-MacLaurin.*

Somit erhält man eine Darstellung der Form (80) mit

$$\tau_0 \;:=\; \int_a^b f(t)\,dt \quad \text{und} \quad \tau_i \;:=\; \frac{B_{2i}}{(2i)!}\left(f^{(2i-1)}(b)-f^{(2i-1)}(a)\right).$$

Damit können wir nun das Extrapolations-Prinzip auf die zusammengesetzte Trapezregel anwenden.

Das Romberg-Verfahren

Wähle $N_0 := 1,\ N_1 := 2,\ N_2 := 4, \ldots,\ N_i := 2^i$ (die Anzahl der Teil-Intervalle wird jeweils verdoppelt) und entprechend $h_i := \frac{b-a}{N_i} = \frac{b-a}{2^i}$. Dann werden die zusammengesetzten Trapezregeln

$$T(h_i) = \frac{b-a}{2^i}\left(\frac{f(a)}{2} + \frac{f(b)}{2} + \sum_{j=1}^{2^i-1} f(a+jh_i)\right) \tag{81}$$

gebildet und auf die Paare $(h_i^2, T(h_i)),\ i = 0, \ldots, I$ ($I \in \mathbb{N}$ gegeben) der Neville-Algorithmus angewandt (Auswertung an der Stelle Null). Hier bezeichnet man die Einträge im Neville-Schema mit T_j^k, der untere Index gibt dabei die Spalte im Tableau an.

$$\begin{array}{c|cccc}
h_0^2 & T(h_0) = T_0^0 & & & \\
 & & T_1^0 & & \\
h_1^2 & T(h_1) = T_0^1 & & T_2^0 & \\
 & & T_1^1 & & T_3^0 \\
h_2^2 & T(h_2) = T_0^2 & & T_2^1 & \ddots \\
 & & T_1^2 & \vdots & \\
h_3^2 & T(h_3) = T_0^3 & \vdots & & \\
 & \vdots & & & \\
\vdots & \vdots & & &
\end{array}$$

mit der Rekursionsformel

$$T_j^k = \frac{4^j T_{j-1}^{k+1} - T_{j-1}^k}{4^j - 1} \qquad (j = 1, 2, \ldots;\ k = 0, 1, \ldots)\ . \tag{82}$$

Beweis:
In der Notation des Neville-Schemas entspricht $T_j^k = p_{k\cdots k+j}(0)$ (beachte: hier ist $\xi = 0$). Für die h_i gilt

$$h_{k+j} = \frac{h_k}{2^j} \qquad \text{also} \qquad h_{k+j}^2 = \frac{h_k^2}{4^j}\ .$$

Damit erhält man

$$\begin{aligned}
T_j^k = p_{k\cdots k+j}(0) &= \frac{(0 - h_{k+j}^2)p_{k\cdots k+j-1}(0) - (0 - h_k^2)p_{k+1\cdots k+j}(0)}{h_k^2 - h_{k+j}^2} \\
&= \frac{h_k^2 T_{j-1}^{k+1} - h_{k+j}^2 T_{j-1}^k}{h_k^2 - h_{k+j}^2} \\
&= \frac{T_{j-1}^{k+1} - \frac{1}{4^j} T_{j-1}^k}{1 - \frac{1}{4^j}} = \frac{4^j T_{j-1}^{k+1} - T_{j-1}^k}{4^j - 1}\ . \ \square
\end{aligned}$$

Definition: Das obige Schema heißt *Romberg-Schema (Romberg-Tableau).*

Praktische Durchführung des Romberg-Verfahrens
Ziel: Programmierung so, dass möglichst wenig Rechenaufwand erforderlich ist.

Die Intervallhalbierung hat den Vorteil, dass $T(h_i)$ zur Berechnung von $T(h_{i+1})$ verwendet werden kann. Dazu bezeichne $M(h_i)$ die zusammengesetzte Mittelpunktsregel:

$$\begin{aligned} M(h_i) &= \frac{b-a}{2^i} \sum_{j=1}^{2^i} f\left(a + \frac{2j-1}{2 \cdot 2^i}(b-a)\right) \\ &= h_i \sum_{j=1}^{2^i} f(a + (2j-1)h_{i+1}) \quad . \end{aligned}$$

Lemma: Es gilt

$$T(h_{i+1}) = \frac{1}{2}\Big(T(h_i) + M(h_i)\Big) \quad .$$

Beweis: Wir erhalten

$$\begin{aligned} T(h_{i+1}) &= \frac{b-a}{2^{i+1}} \left(\frac{f(a)}{2} + \frac{f(b)}{2} + \sum_{j=1}^{2^{i+1}-1} f\left(a + j\frac{b-a}{2^{i+1}}\right)\right) \\ &= \frac{1}{2}\left[\frac{b-a}{2^i}\left(\frac{f(a)}{2} + \frac{f(b)}{2} + \sum_{j \text{ gerade}} f\left(a + j\frac{b-a}{2^{i+1}}\right) + \sum_{j \text{ ungerade}} f\left(a + j\frac{b-a}{2^{i+1}}\right)\right)\right] \\ &= \frac{1}{2}\left[\frac{b-a}{2^i}\left(\frac{f(a)}{2} + \frac{f(b)}{2} + \sum_{k=1}^{2^i-1} f\left(a + 2k\frac{b-a}{2^{i+1}}\right) + \sum_{k=1}^{2^i} f\left(a + (2k-1)\frac{b-a}{2^{i+1}}\right)\right)\right] \\ &= \frac{1}{2}\left[\frac{b-a}{2^i}\left(\frac{f(a)}{2} + \frac{f(b)}{2} + \sum_{k=1}^{2^i-1} f\left(a + k\frac{b-a}{2^i}\right) + \sum_{k=1}^{2^i} f\left(a + (2k-1)\frac{b-a}{2^{i+1}}\right)\right)\right] \\ &= \frac{1}{2}\left(T(h_i) + M(h_i)\right) \quad . \ \square \end{aligned}$$

Der Algorithmus zur Berechnung des Romberg-Tableaus wird folgende Punkte berücksichtigen:

- Das Tableau wird zeilenweise erstellt.
- Berechne $T_0^0 = T(h_0) = h_1(f(a) + f(b))$.
- Für $i \geq 1$ (i-te Zeile)
  berechne $M(h_{i-1})$ (zusammengesetzte Mittelpunktsregel) und dann
  $T_0^i = T(h_i) = \frac{1}{2}\left(T_0^{i-1} + M(h_{i-1})\right)$.
- Für $j \geq 1$ berechne T_j^k $(j + k = i)$ gemäß der Formel (82).
- Abbruchbedingung: Gilt für gegebenes $\varepsilon > 0$
  $\left|T_{i-1}^1 - T_{i-1}^0\right| \leq \varepsilon$, dann wird T_i^0 als Näherung akzeptiert.

7.6 Gauß-Quadraturformeln

In Abschnitt 7.2 wurde gezeigt, dass eine interpolatorische Quadraturformel (mit $m+1$ Stützstellen) mindestens den Exaktheitsgrad m hat. Es stellt sich die Frage, ob durch geeignete Wahl der Stützstellen ein noch höherer Exaktheitsgrad erreicht werden kann.

Lemma: Eine Quadraturformel mit $m+1$ Stützstellen hat höchstens den Exaktheitsgrad $2m+1$.

Beweis: Seien $a \le t_0 < \ldots < t_m \le b$ beliebige Stützstellen. Für das Polynom

$$q(t) \;:=\; \prod_{j=0}^{m} (t - t_j)^2$$

gilt $q \in \Pi_{2m+2}$ und

$$\int_a^b q(t)\,dt \;>\; 0 \qquad \text{sowie} \qquad \sum_{i=0}^{m} w_i q(t_i) \;=\; 0 \quad ,$$

also $R[q] \neq 0$. $\square$

Legendre-Polynome

Wir beschränken uns nun auf das Intervall $[-1,1]$. Durch

$$< f, g > := \int_{-1}^{1} f(t)g(t)\,dt \qquad (f, g \in C[-1,1])$$

ist ein Skalarprodukt gegeben. Zwei Polynome $p \neq 0$ und $q \neq 0$ heißen *orthogonal*, falls $< p, q > = 0$ gilt.

Satz (ohne Beweis): Es gibt eine Folge $(P_k)_{k\in\mathbb{N}}$ orthogonaler Polynome mit Grad $P_k = k$ und Hauptkoeffizient > 0 (d.h. $< P_i, P_j > = 0$ für $i \neq j$).

Anmerkungen:

1. Diese Polynome können mit Hilfe des Schmidtschen Orthogonalisierungsverfahrens rekursiv konstruiert werden und heißen *Legendre-Polynome*.

2. Die orthogonalen Polynome $P_0, P_1, \ldots, P_n$ bilden eine Basis des Polynomraumes Π_n. Daraus folgt
$$< P_n, q > = \; 0 \qquad \text{für alle } q \in \Pi_{n-1} \; . \tag{83}$$

3. Für die Legendre-Polynome gilt eine dreigliedrige Rekursionsformel:
$$P_{n+1}(t) \;=\; \frac{2n+1}{n+1}\, t\, P_n(t) - \frac{n}{n+1}\, P_{n-1}(t) \qquad (n \ge 1) \tag{84}$$
mit $P_0(t) \equiv 1$ und $P_1(t) = t$.

Satz: P_n hat n einfache reelle Nullstellen im offenen Intervall $(-1,1)$.

Beweis: Annahme P_n hat in $(-1,1)$ nur $l < n$ reelle Nullstellen $-1 < t_1 < \cdots < t_l < 1$ mit ungerader Ordnung. Dann gilt

$$q(t) \;:=\; \prod_{i=1}^{l}(t - t_i) \quad \in \Pi_l$$

und nach (83)

$$< P_n, q > \quad = \int_{-1}^{1} P_n(t)q(t)\,dt \;=\; 0 \qquad . \tag{85}$$

Andererseits ist $P_n(t)q(t)$ ein Polynom, das in $(-1,1)$ nur Nullstellen gerader Ordnung hat, folglich gilt $P_n(t)q(t) \geq 0$. Somit

$$\int_{-1}^{1} P_n(t)q(t)\,dt \;>\; 0 \qquad .$$

Dies ist ein Widerspruch zu (85). Also hat P_n in $(-1,1)$ n reelle Nullstellen ungerader Ordnung. Da aber $P_n \in \Pi_n$ gilt, sind diese Nullstellen alle einfach, und es gibt keine weiteren (Fundamentalsatz der Algebra). $\square$

Wähle nun die Nullstellen $-1 < t_0 < \cdots < t_m < 1$ des Legendre-Polynoms P_{m+1} als Stützstellen für eine interpolatorische Quadraturformel.

Definition: Diese Formeln heißen *Gauß-Quadraturformeln.*
(Carl Friedrich Gauß, 1777-1855)

Satz: Die Gauß-Quadraturformel mit $m+1$ Stützstellen hat den Exaktheitsgrad $M = 2m+1$.

Beweis: Sei $q \in \Pi_{2m+1}$ beliebig. Dann gibt es Polynome $r, s \in \Pi_m$ mit

$$q \;=\; r \cdot P_{m+1} \;+\; s$$

(P_{m+1} Legendre-Polynom, Division mit Rest). Es gilt $R[s] \;=\; 0$ (da eine interpolatorische QF zugrunde liegt) und

$$R[r \cdot P_{m+1}] \;=\; \int_{-1}^{1} r(t)P_{m+1}(t)\,dt \;-\; \sum_{i=0}^{m} w_i\, r(t_i)\, P_{m+1}(t_i) \;=\; 0$$

(Orthogonalität, vgl. (83); t_i Nullstelle von P_{m+1}). Insgesamt erhalten wir

$$R[q] \;=\; R[r \cdot P_{m+1} + s] \;=\; R[r \cdot P_{m+1}] \;+\; R[s] \;=\; 0 \quad . \;\square$$

Beispiel:
Aus der Rekursionsformel der Legendre-Polynome (vgl. (84)) errechnet man

$$P_2(t) \;=\; \frac{3}{2}t^2 \;-\; \frac{1}{2}$$

und somit als Stüztstellen

$$t_0 \;=\; -\frac{\sqrt{3}}{3}\,, \qquad t_1 \;=\; \frac{\sqrt{3}}{3} \quad .$$

Als Gauß-Quadraturformel für $m = 1$ ergibt sich

$$\int_{-1}^{1} f(t)\,dt \;=\; f\left(-\frac{\sqrt{3}}{3}\right) \;+\; f\left(\frac{\sqrt{3}}{3}\right) \;+\; R[f] \quad .$$

Exaktheitsgrad: $M = 3$.

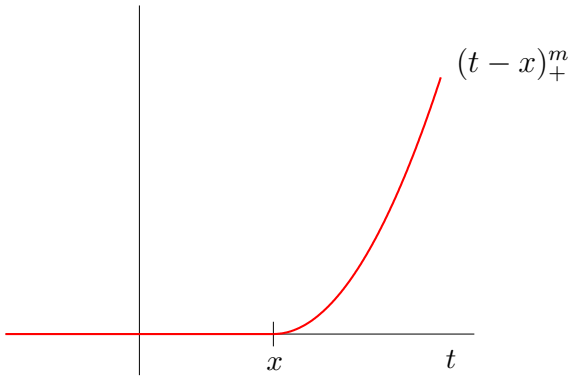
Fehler: Sei $f \in C^4[-1,1]$. Dann gilt $R[f] \;=\; \dfrac{f^{(4)}(\xi)}{135}$ für ein $\xi \in (-1,1)$.

7.7 Fehlerbetrachtungen

Wir führen für $m \geq 0$ folgende Bezeichnung ein:

$$(t-x)_+^m \;:=\; \begin{cases} (t-x)^m & \text{für } t \geq x \\ 0 & \text{für } t < x \end{cases}$$

(abgeschnittene Potenzfunktion, truncated power function).



Die Quadraturformel (68) habe den Exaktheitsgrad M.

Satz (Peano): Sei $f \in C^{M+1}[a,b]$. Dann gilt für den Quadraturfehler

$$R[f] \;=\; \int_a^b f^{(M+1)}(s)\,K(s)\,ds \tag{86}$$

$$\text{mit} \qquad K(s) \;:=\; \frac{1}{M!}\int_a^b (t-s)_+^M\,dt \;-\; \frac{1}{M!}\sum_{i=0}^{m} w_i (t_i - s)_+^M \quad . \tag{87}$$

Definition: Diese Fehlerdarstellung nennt man *Kerndarstellung von Peano*. $K(s)$ wird als *Peano-Kern* bezeichnet.

Beweis: Sei nun $f \in C^{M+1}[a,b]$. Die Taylor-Entwicklung um den Punkt a liefert

$$f(t) \;=\; f(a) + f'(a)(t-a) + \cdots + \frac{f^{(M)}(a)}{M!}\,(t-a)^M \;+ r_M(t)$$

mit dem Restglied (in Integralform)

$$r_M(t) \;=\; \frac{1}{M!}\int_a^t f^{(M+1)}(s)(t-s)^M\,ds \;=\; \frac{1}{M!}\int_a^b f^{(M+1)}(s)(t-s)_+^M\,ds \quad .$$

Somit gilt für den Quadraturfehler

$$\begin{aligned}
R[f(t)] &= f(a)\,R[1] + f'(a)\,R[(t-a)] + \cdots + \frac{f^{(M)}(a)}{M!}\,R[(t-a)^M] \;+\; R[r_M(t)] \\
&= R[r_M(t)] \\
&= \int_a^b r_M(t)\,dt \;-\; \sum_{i=0}^{m} w_i r_M(t_i) \\
&= \frac{1}{M!}\int_a^b \left(\int_a^b f^{(M+1)}(s)(t-s)_+^M\,ds\right) dt \;-\; \frac{1}{M!}\sum_{i=0}^{m} w_i \int_a^b f^{(M+1)}(s)(t_i-s)_+^M ds \\
&= \frac{1}{M!}\int_a^b \left(\int_a^b f^{(M+1)}(s)(t-s)_+^M\,dt\right) ds \;-\; \frac{1}{M!}\int_a^b \left(f^{(M+1)}(s)\sum_{i=0}^{m} w_i(t_i-s)_+^M\right) ds \\
&= \int_a^b f^{(M+1)}(s)\,\left(\frac{1}{M!}\int_a^b (t-s)_+^M\,dt \;-\; \frac{1}{M!}\sum_{i=0}^{m} w_i(t_i-s)_+^M\right) ds \\
&= \int_a^b f^{(M+1)}(s)\,K(s)\,ds \quad .
\end{aligned}$$

Im obigen Doppelintegral darf man die Integrationsreihenfolge vertauschen (Satz von Fubini). □

Anmerkungen:

1. Der Peano-Kern $K(t)$ hängt nur von der gegebenen Quadraturformel ab, nicht jedoch vom Integranden f.
2. Für $M \geq 1$ ist $K(s)$ stetig, was aus (87) folgt.

Besitzt $K(s)$ im Intervall $[a,b]$ keinen Vorzeichenwechsel, so lässt sich auf die Kerndarstellung von Peano der erweiterte Mittelwertsatz der Integralrechnung anwenden.

Korollar: Sei $f \in C^{M+1}[a,b],\ M \geq 1$, und $K(s) \geq 0$ für alle $s \in [a,b]$ (oder $K(s) \leq 0$ für alle $s \in [a,b]$). Dann gibt es ein $\xi \in (a,b)$ mit

$$R[f] \ = \ f^{(M+1)}(\xi) \int_a^b K(s)\, ds \quad . \tag{88}$$

Aus (86) erhalten wir die Fehlerabschätzung

$$|R[f]| \ \leq \ \left\| f^{(M+1)} \right\|_\infty \cdot \int_a^b |K(s)|\, ds \quad .$$

Beispiel: Für die Mittelpunktsregel

$$\int_a^b f(t)\, dt \ = \ (b-a) f\left(\frac{a+b}{2}\right) \ + \ R[f]$$

haben wir in Abschnitt 7.2 die Fehlerdarstellung

$$R[f] \ = \ \frac{1}{24}(b-a)^3 f^{(2)}(\xi) \quad \text{für ein } \xi \in (a,b)$$

(dabei sei $f \in C^2[a,b]$) behauptet. Wir beweisen nun diese Aussage.

Die Mittelpunktsregel hat den Exaktheitsgrad $M = 1$. Deshalb liefert der Satz von Peano für $f \in C^2[a,b]$ die Darstellung

$$R[f] \ = \ \int_a^b f^{(2)}(s) K(s)\, ds$$

mit

$$K(s) \ = \ \frac{1}{1!} \int_a^b (t-s)_+^1\, dt \ - \ (b-a) \left(\frac{a+b}{2} - s\right)_+^1 \quad .$$

Für $K(s)$ erhält man

$$\begin{aligned} K(s) \ &= \ \int_s^b (t-s)^1\, dt \ - \ (b-a) \left(\frac{a+b}{2} - s\right)_+^1 \\ &= \ \frac{1}{2}(b-s)^2 \ - \ (b-a) \left(\frac{a+b}{2} - s\right)_+^1 \\ &= \ \begin{cases} \frac{1}{2}(b-s)^2 \ - \ (b-a)\left(\frac{a+b}{2} - s\right) & \text{für } s \leq \frac{a+b}{2} \\ \frac{1}{2}(b-s)^2 & \text{für } s > \frac{a+b}{2} \end{cases} \end{aligned}$$

Eine leichte Umrechnung zeigt:

$$K(s) \;=\; \begin{cases} \frac{1}{2}(a-s)^2 & \text{für } s \le \frac{a+b}{2} \\ \frac{1}{2}(b-s)^2 & \text{für } s > \frac{a+b}{2} \end{cases}$$

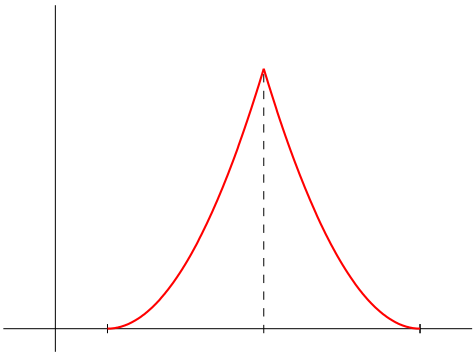
Da $K(s)$ keinen Vorzeichenwechsel hat, gilt mit (88)

$$\begin{aligned} R[f] &= f^{(2)}(\xi) \int_a^b K(s)\,ds \\ &= \frac{1}{24}(b-a)^3 f^{(2)}(\xi) \quad , \end{aligned}$$

denn

$$\begin{aligned} \int_a^b K(s)\,ds &= \frac{1}{2}\int_a^{\frac{a+b}{2}} (a-s)^2\,ds \;+\; \frac{1}{2}\int_{\frac{a+b}{2}}^b (b-s)^2\,ds \\ &= -\frac{1}{6}\,(a-s)^3\Big|_a^{\frac{a+b}{2}} \;-\; \frac{1}{6}\,(b-s)^3\Big|_{\frac{a+b}{2}}^b \\ &= -\frac{1}{6}\left(\frac{a-b}{2}\right)^3 + \frac{1}{6}\left(\frac{b-a}{2}\right)^3 \\ &= \frac{1}{48}(b-a)^3 \;+\; \frac{1}{48}(b-a)^3 \;=\; \frac{1}{24}(b-a)^3 \quad . \end{aligned}$$

Peano-Kern für Mittelpunktsregel



8 Eigenwertaufgaben

8.1 Eigenwerte und Eigenvektoren

Sei $A \in \mathbb{R}^{n\times n}$. Dann lautet das Eigenwertproblem: Gesucht ist ein $\lambda \in \mathbb{C}$ und ein Vektor $x \in \mathbb{C}^n \setminus \{0\}$ mit

$$Ax \;=\; \lambda x\;.$$

Die Zahl λ heißt *Eigenwert* und der Vektor x ein zugehöriger *Eigenvektor* von A.

In der linearen Algebra lernen Sie, dass die Eigenwerte die Nullstellen der charakteristischen Polynoms

$$p(t) \;:=\; \det(A - tI)$$

sind. Somit kann man die Eigenwertberechnung auf die Nullstellenberechnung eines Polynomes zurückführen. Aus der Sicht der Numerik ist diese Vorgehensweise jedoch sehr problematisch, da sie zu einer hohen Störempfindlichkeit in den Koeffizienten des char. Polynoms führt (in Paragraph 11 gehen wir noch ausführlich darauf ein). Deshalb kommen in der Numerik andere Methoden zum Zug, die die explizite Berechnung des charakteristischen Polynoms vermeiden.

Vielmehr macht man von der Tatsache Gebrauch, dass ähnliche Matrizen dieselben Eigenwerte haben: Sei $Q \in \mathbb{R}^{n\times n}$ invertierbar, dann hat die Matrix

$$B \;:=\; QAQ^{-1}$$

dieselben Eigenwerte wie A. Ist x ein Eigenvektor von A zum Eigenwert λ, so ist $y := Qx$ ein Eigenvektor von B zum Eigenwert λ, denn

$$By \;=\; QAQ^{-1}y \;=\; QAQ^{-1}Qx \;=\; QAx \;=\; Q(\lambda x) \;=\; \lambda Qx \;=\; \lambda y \quad .$$

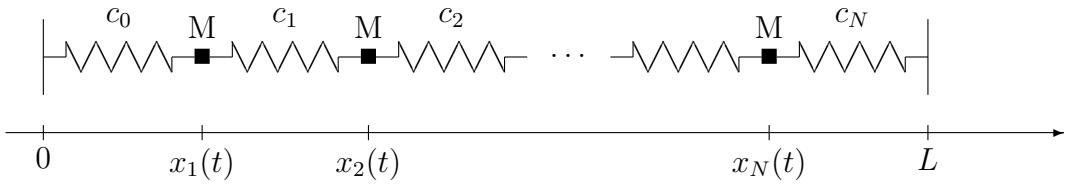
Aus Sicht der Numerik (Stabilität, Rechenaufwand) sind dabei orthogonale Matrizen Q (d.h. $Q^{-1} = Q^T$) besonders interessant.

Dabei geht man meistens zweistufig vor: Zuerst wird die allgemeine Matrix A durch eine Ähnlichkeitstransformation auf eine einfachere Form gebracht (Vortransformation). Anschließend werden die Eigenwerte der transformierten Matrix bestimmt, in der Regel durch ein iteratives Verfahren. Zum Beispiel versucht man, durch iterative Ähnlichkeitstransformationen näherungsweise eine Gestalt zu erhalten, bei der die Eigenwerte sofort abgelesen werden können (Diagonalgestalt, Dreiecksgestalt).

8.2 Anwendung aus der Physik

Schwingendes Federsystem (lineare Kette)

Vorgelegt seien N Massen M, die durch $N+1$ Federn linear gekoppelt sind. Sie sollen zwischen zwei festen Enden im Abstand L eingespannt sein:

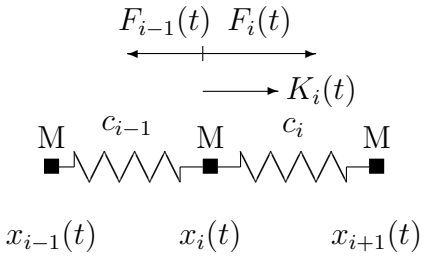


Die i-te Feder habe dabei die Ruhelänge L_0 und die Federkonstante $c_i > 0$.

Wir betrachten das schwingende System ohne Reibung. Es sei $x_i(t)$ die Position der i-ten Masse zum Zeitpunkt t. Dann ist $\dot{x}(t)$ die Geschwindigkeit und $\ddot{x}(t)$ die Beschleunigung der i-ten Masse. Die Einspannung des Federsystems berücksichtigen wir durch die Setzung

$$x_0(t) \;:=\; 0\,, \qquad x_{N+1}(t) \;:=\; L \quad .$$

Weiter sei $K_i(t)$ die im Zeitpunkt t in der i-ten Masse resultierende Kraft.



Die i-te Masse genügt der Bewegungsgleichung

$$M\ddot{x}_i(t) \;=\; K_i(t) \quad . \tag{89}$$

Die Kraft $K_i(t)$ ergibt sich dabei durch Überlagerung der beiden Nachbarkräfte $F_{i-1}(t)$ und $F_i(t)$, also

$$K_i(t) \;=\; F_i(t) - F_{i-1}(t)\,.$$

Das Hookesche Gesetz liefert für die i-te Feder

$$F_i(t) \;=\; c_i(x_{i+1}(t) - x_i(t) - L_0) \qquad (i = 0, \ldots, N)$$

und somit für die resultierende Kraft

$$\begin{aligned} K_i(t) &= c_i(x_{i+1}(t) - x_i(t) - L_0) - c_{i-1}(x_i(t) - x_{i-1}(t) - L_0) \\ &= c_{i-1}x_{i-1}(t) - (c_i + c_{i-1})x_i(t) + c_i x_{i+1}(t) + (c_{i-1} - c_i)L_0 \quad . \end{aligned}$$

Die Bewegungsgleichung (89) lautet dann für $i = 1, \ldots, N$

$$\ddot{x}_i(t) \;=\; \frac{1}{M}\left\{c_{i-1}x_{i-1}(t) - (c_i + c_{i-1})x_i(t) + c_i x_{i+1}(t) + (c_{i-1} - c_i)L_0\right\} \quad . \tag{90}$$

Wir fassen die $x_i(t)$ bzw. $\ddot{x}_i(t)$ zu Vektoren zusammen:

$$x(t) \;:=\; \begin{pmatrix} x_1(t) \\ \vdots \\ x_N(t) \end{pmatrix}\,, \qquad \ddot{x}(t) \;:=\; \begin{pmatrix} \ddot{x}_1(t) \\ \vdots \\ \ddot{x}_N(t) \end{pmatrix} \quad .$$

Mit

$$A \;=\; \frac{1}{M}\begin{pmatrix} c_0 + c_1 & -c_1 & 0 & \cdots & 0 \\ -c_1 & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -c_{N-1} \\ 0 & \cdots & 0 & -c_{N-1} & c_{N-1} + c_N \end{pmatrix}$$

und

$$b \;=\; \frac{1}{M}\begin{pmatrix} (c_0 - c_1)L_0 \\ (c_1 - c_2)L_0 \\ \vdots \\ (c_{N-2} - c_{N-1})L_0 \\ (c_{N-1} - c_N)L_0 + c_N L \end{pmatrix}$$

erhält man aus (90) die Darstellung

$$\ddot{x}(t) \;=\; b \,-\, Ax(t) \quad . \tag{91}$$

Dies ist ein *Differentialgleichungssystem 2. Ordnung.*

Von der Physik her erwartet man, dass diese Differentialgleichungen die Positionen $x_i(t)$ der Massen für alle $t \geq 0$ eindeutig bestimmen, sofern

– die Anfangspositionen $x_i(0)\,,\ i = 1, \ldots, N$, und

– die Anfangsgeschwindigkeiten $\dot{x}_i(0)\,,\ i = 1, \ldots, N$,

vorgegeben werden.

Ruhelage

Das Federsystem befindet sich in der Ruhelage, falls alle resultierenden Kräfte Null sind, d.h. $K_i(t) = 0$ $(i = 1, \ldots, N)$ gilt. Aus (91) folgt, dass die Lösung $\overline{x}$ des linearen Gleichungssystems

$$Ax \;=\; b$$

die Ruhelage liefert.

Dann löst $x(t) \;:=\; \bar{x}$ das Dgl-System (91) zu den Anfangswerten

$$x_i(0) \;:=\; \bar{x}_i \quad \text{ und } \quad \dot{x}_i(0) \;=\; 0 \quad (i = 1, \ldots, N).$$

Diese Lösung nennt man einen *stationären Zustand* des Dgl-Systems (91).

Die Matrix A besitzt Zusatzeigenschaften, die wir bei der Lösung des obigen Gleichungssystems ausnutzen:

Lemma: Gilt $c_i > 0$ $(i = 0, \ldots, N)$, so ist die Matrix A positiv definit.

Beweis: Übung.

Transformation auf ein homogenes System

Sei $\bar{x}$ die Ruhelage des Federsystems (d.h. $A\bar{x} = b$). Wir transformieren das Dgl-System (91) in ein homogenes System. Dazu sei

$$y_i(t) \;:=\; x_i(t) - \bar{x}_i \quad (i = 1, \ldots, N).$$

Physikalisch ist $y_i(t)$ die Auslenkung der i-ten Masse aus der Ruhelage. Dann gilt

$$\ddot{y}(t) \;=\; \ddot{x}(t) \;=\; b - A(y(t) + \bar{x}) \;=\; -Ay(t) \quad .$$

Es genügt also, das System

$$\ddot{y}(t) \;=\; -Ay(t) \tag{92}$$

zu lösen. Ist $y(t)$ eine Lösung von (92), so löst

$$x(t) \;:=\; y(t) + \bar{x}$$

das System (91).

Lösungen des homogenen Systems
Aus der linearen Algebra wissen wir, dass eine positiv definite Matrix A lauter positive reelle Eigenwerte besitzt. Seien $0 < \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N$ die Eigenwerte von A und $v^{(1)}, \ldots, v^{(N)}$ zugehörige (linear unabhängige) Eigenvektoren.

Alle Lösungen des homogenen Systems (92) sind von der Form

$$y(t) \;=\; \sum_{j=1}^{N} \Big(\alpha_j \, \cos(\sqrt{\lambda_j} t) v^{(j)} \;+\; \beta_j \, \sin(\sqrt{\lambda_j} t) \, v^{(j)} \Big) \qquad , \tag{93}$$

wobei $\alpha_j, \beta_j \in \mathbb{R}$ beliebige Konstanten sind. Dass dies Lösungen des homogenen Systems sind, ergibt sich sofort durch Nachrechnen.

Lösungen des schwingenden Federsystems
Aus (93) erhält man als Lösungen für das schwingende Federsystem

$$x(t) \;=\; \sum_{j=1}^{N} \Big(\alpha_j \, \cos(\sqrt{\lambda_j} t) v^{(j)} \;+\; \beta_j \, \sin(\sqrt{\lambda_j} t) \, v^{(j)} \Big) \;+\; \overline{x} \qquad ,$$

wobei $\overline{x}$ wieder die Ruhelage des Federsystems ist. In der Komponentenschreibweise bedeutet dies

$$x_i(t) \;=\; \sum_{j=1}^{N} \Big(\alpha_j \, \cos(\sqrt{\lambda_j} t) v_i^{(j)} \;+\; \beta_j \, \sin(\sqrt{\lambda_j} t) \, v_i^{(j)} \Big) \;+\; \overline{x}_i \quad (i = 1, \ldots, N). \tag{94}$$

Für die Ableitung folgt daraus

$$\dot{x}_i(t) \;=\; \sum_{j=1}^{N} \Big(-\sqrt{\lambda_j} \cdot \alpha_j \, \sin(\sqrt{\lambda_j} t) v_i^{(j)} \;+\; \sqrt{\lambda_j} \cdot \beta_j \, \cos(\sqrt{\lambda_j} t) \, v_i^{(j)} \Big) \qquad . \tag{95}$$

Seien nun Anfangsbedingungen

$$x_i(0) \;\text{ und }\; \dot{x}_i(0) \quad (i = 1, \ldots, N)$$

vorgegeben. Dann erhalten wir aus (94) bzw. (95) für die α_j und die β_j die Beziehungen

$$x_i(0) \;=\; \sum_{j=1}^{N} \alpha_j \cdot v_i^{(j)} \;+\; \overline{x}_i \qquad (i = 1, \ldots, N) \;,$$

$$\dot{x}_i(0) \;=\; \sum_{j=1}^{N} \beta_j \Big(\sqrt{\lambda_j} v_i^{(j)} \Big) \qquad (i = 1, \ldots, N) \;.$$

Dies sind zwei lineare Gleichungssyteme. Für die α_j ergibt sich das lineare Gleichungssystem $C\alpha = d$ mit

$$C = \begin{pmatrix} v_1^{(1)} & \cdots & v_1^{(N)} \\ \vdots & & \vdots \\ v_N^{(1)} & \cdots & v_N^{(N)} \end{pmatrix}, \quad d = \begin{pmatrix} x_1(0) - \overline{x}_1 \\ \vdots \\ x_N(0) - \overline{x}_N \end{pmatrix}, \quad \alpha = \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_N \end{pmatrix}.$$

Die β_j erhält man aus dem linaren Gleichungssystem $\overline{C}\beta = \overline{d}$ mit

$$\overline{C} = \begin{pmatrix} \sqrt{\lambda_1} v_1^{(1)} & \cdots & \sqrt{\lambda_N} v_1^{(N)} \\ \vdots & & \vdots \\ \sqrt{\lambda_1} v_N^{(1)} & \cdots & \sqrt{\lambda_N} v_N^{(N)} \end{pmatrix}, \quad \overline{d} = \begin{pmatrix} \dot{x}_1(0) \\ \vdots \\ \dot{x}_N(0) \end{pmatrix}, \quad \beta = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_N \end{pmatrix}.$$

Beachten Sie, dass die beiden obigen Matrizen regulär sind. Deshalb sind die α_j und β_j eindeutig bestimmt.

Fazit: Bei gegebenen Anfangsbedingungen gibt es genau eine Lösung des schwingenden Federsystems. Dies stimmt mit den physikalischen Erwartungen überein.

Zusammenfassung
Die Bestimmung der Lösung des schwingenden Federsystems führt uns auf folgende numerische Problemstellungen:

1. Lösung eines linearen Gleichungssystems mit positiv definiter Koeffizienten-Matrix (Berechnung der Ruhelage).
2. Bestimmung aller Eigenwerte und zugehöriger Eigenvektoren einer reellen symmetrischen Matrix.
3. Lösung von linearen Gleichungssystemen mit beliebiger regulärer Koeffizientenmatrix (Bestimmung der α_j bzw β_j).

8.3 Lokalisierung der Eigenwerte

Der folgende Satz liefert Aussagen über die ungefähre Lage der Eigenwerte.

Satz (Gerschgorin): Sei $A \in \mathbb{R}^{n\times n}$. Die Vereinigung aller Kreisscheiben

$$K_i := \left\{ z \in \mathbb{C} \ : \ |z - a_{ii}| \ \leq \sum_{\substack{k=1 \\ k \neq i}}^{n} |a_{ik}| \right\} \quad , \qquad i = 1, \ldots, n$$

enthält alle Eigenwerte von A.

Beweis: Sei λ ein Eigenwert und x ein zugehöriger Eigenvektor, also $Ax = \lambda x$ oder in Komponenten

$$\sum_{j=1}^{n} a_{lj} x_j \ = \ \lambda x_l \qquad (l = 1, \ldots, n).$$

Daraus erhalten wir

$$(\lambda - a_{ll})x_l \;=\; \sum_{\substack{j\,=\,1\\ j\,\neq\,l}}^{n} a_{lj}x_j \qquad (l = 1, \ldots, n).$$

Sei nun $i \in \{1, \ldots, n\}$ ein Index mit $|x_i| = \|x\|_\infty$. Dann ergibt sich

$$|\lambda - a_{ii}| \;=\; \left| \sum_{\substack{j\,=\,1\\ j\,\neq\,i}}^{n} a_{ij}\frac{x_j}{x_i} \right| \;\leq\; \sum_{\substack{j\,=\,1\\ j\,\neq\,i}}^{n} |a_{ij}|\frac{|x_j|}{|x_i|} \;\leq\; \sum_{\substack{j\,=\,1\\ j\,\neq\,i}}^{n} |a_{ij}|$$

und folglich die Behauptung. □

Anmerkungen:

1. Die K_i bezeichnet man als *Gerschgorin-Kreise*.
2. Man kann ferner zeigen, dass K_i genau einen Eigenwert enthält, falls K_i disjunkt ist zu allen anderen Kreisen (vgl. Stoer-Bulirsch [8]).
3. Da A^T dieselben Eigenwerte hat wie A, kann man zusätzlich die Gerschgorin-Kreise zu A^T bilden und damit u.U. weitere Informationen über die Lage der Eigenwerte erhalten.

Beispiel: (vgl. Stoer-Bulirsch [8])

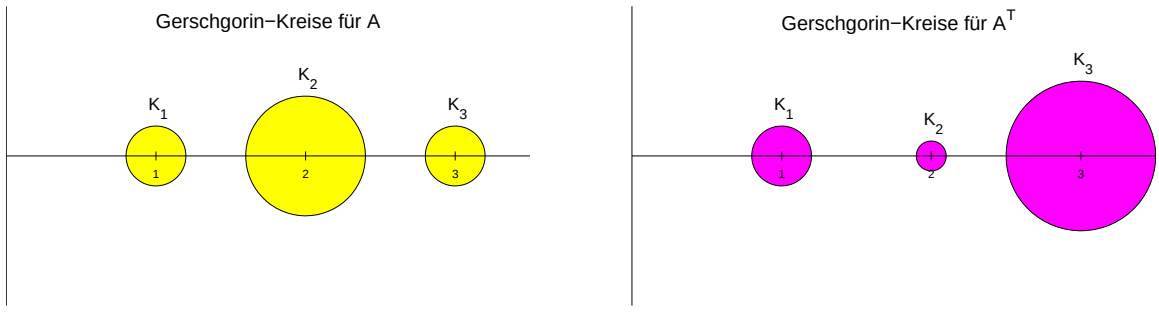
Es sei $A = \begin{pmatrix} 1 & 0.1 & -0.1 \\ 0 & 2 & 0.4 \\ -0.2 & 0 & 3 \end{pmatrix}$.

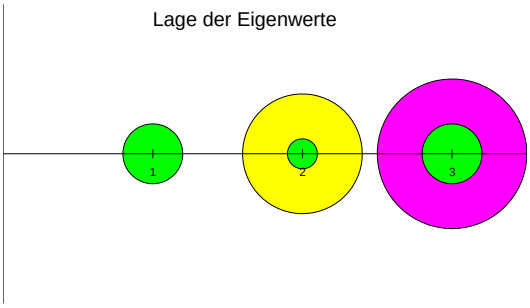
Gerschgorin-Kreise für A

$$\begin{aligned} K_1 &= \{z \in \mathbb{C} \,:\, |z-1| \leq 0.2\} \\ K_2 &= \{z \in \mathbb{C} \,:\, |z-2| \leq 0.4\} \\ K_3 &= \{z \in \mathbb{C} \,:\, |z-3| \leq 0.2\} \end{aligned}$$

Gerschgorin-Kreise für A^T

$$\begin{aligned} \overline{K}_1 &= \{z \in \mathbb{C} \,:\, |z-1| \leq 0.2\} \\ \overline{K}_2 &= \{z \in \mathbb{C} \,:\, |z-2| \leq 0.1\} \\ \overline{K}_3 &= \{z \in \mathbb{C} \,:\, |z-3| \leq 0.5\} \end{aligned}$$





8.4 Transformation auf obere Hessenberg-Gestalt

Definition: Eine $n \times n$ - Matrix der Form

$$\begin{pmatrix} * & \cdots & \cdots & \cdots & * \\ * & \ddots & & & \vdots \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & * & * \end{pmatrix}$$

heißt *obere Hessenberg-Matrix*.

Satz: Sei $A \in \mathbb{R}^{n\times n}$. Dann gibt es eine orthogonale Matrix Q, so dass QAQ^T eine obere Hessenberg-Matrix ist.

Anmerkung: Q kann als Produkt von $(n-2)$ Householder-Matrizen gewählt werden. Im Gegensatz zur QR-Zerlegung kann man hier i.A. keine obere Dreiecksmatrix erwarten, da auch von rechts mit Q^T multipliziert wird.

Beweisidee: Wir zerlegen die Matrix A wie folgt:

$$A = \left(\begin{array}{c|ccc} a_{11} & a_{12} & \cdots & a_{1n} \\ \hline a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{array}\right) = \left(\begin{array}{c|c} a_{11} & b \\ \hline c & A_1 \\ & \end{array}\right) \quad . \tag{96}$$

Es gibt eine $(n-1)\times(n-1)$ Householder-Matrix $\tilde{H}_1$ mit

$$\tilde{H}_1 c = \sigma e_1 \qquad (\sigma \in \mathbb{R}),$$

wobei $e_1 \in \mathbb{R}^{n-1}$ den ersten Einheitsvektor bezeichnet (vgl. Abschnitt 1.5). Dann ist

$$H_1 := \left(\begin{array}{c|c} 1 & 0 \\ \hline 0 & \tilde{H}_1 \\ & \end{array}\right) \tag{97}$$

eine $n \times n$ Householdermatrix, und es gilt

$$H_1 A = \left(\begin{array}{c|ccc} a_{11} & a_{12} & \cdots & a_{1n} \\ \hline \sigma & & & \\ 0 & & & \\ \vdots & & \tilde{A}_1 & \\ 0 & & & \end{array}\right) \quad .$$

Beachten Sie, dass die erste Zeile von A nicht verändert wird. Entsprechend verändert die Rechtsmultiplikation von $(H_1 A)$ mit $H_1^T = H_1$ die erste Spalte von $(H_1 A)$ nicht. Also liefert die Ähnlichkeitstransformation

$$A^{(1)} := H_1 A H_1^{-1} = H_1 A H_1 = \left(\begin{array}{c|ccc} a_{11} & \bar{a}_{12} & \cdots & \bar{a}_{1n} \\ \hline \sigma & & & \\ 0 & & & \\ \vdots & & \bar{A}_1 & \\ 0 & & & \end{array}\right) \quad .$$

Der korrekte Beweis verwendet vollständige Induktion, wobei die obigen Ausführungen dem Induktionsschluss entsprechen. □

Hieraus wird auch ersichtlich, dass eine obere Dreiecksmatrix i.A. nicht erreichbar ist. Denn würde man eine Householder-Matrix T verwenden mit

$$T \begin{pmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{n1} \end{pmatrix} = \sigma \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad ,$$

so folgt zwar zunächst

$$TA = \begin{pmatrix} \sigma & * & \cdots & * \\ 0 & * & & \vdots \\ \vdots & \vdots & & \vdots \\ 0 & * & \cdots & * \end{pmatrix} \quad ,$$

aber die anschließende Rechtsmultiplikation von (TA) mit $T^{-1} = T$ zerstört die Nullen in der 1. Spalte wieder.

Korollar: Sei $A \in \mathbb{R}^{n \times n}$ symmetrisch. Dann existiert eine orthogonale Matrix $Q \in \mathbb{R}^{n \times n}$, so dass QAQ^T eine symmetrische Tridiagonalmatrix ist.

Beweis: Für eine symmetrische Matrix hat man in (96) die Aufteilung

$$A = \left(\begin{array}{c|c} a_{11} & c^T \\ \hline c & A_1 \\ & \end{array}\right) \quad .$$

Mit der in (97) definierten Householder-Matrix folgt dann

$$H_1 A = \left(\begin{array}{c|c} a_{11} & c^T \\ \hline \sigma & \\ 0 & \\ \vdots & \tilde{A}_1 \\ 0 & \end{array}\right) \quad \text{und} \quad (H_1 A) H_1 = \left(\begin{array}{c|c} a_{11} & c^T \tilde{H}_1 \\ \hline \sigma & \\ 0 & \\ \vdots & \tilde{A}_1 \tilde{H}_1 \\ 0 & \end{array}\right) \quad ,$$

und $\tilde{H}_1 A_1 \tilde{H}_1$ ist wieder symmetrisch. Nun gilt

$$c^T \tilde{H}_1 = c^T \tilde{H}_1^T = (\tilde{H}_1 c)^T = (\sigma e_1)^T = (\sigma, 0, \ldots, 0) \quad . \quad \square$$

Insgesamt haben wir also das Problem, Eigenwerte und Eigenvektoren von einer beliebigen Matrix zu berechnen, zurückgeführt auf das Bestimmen von Eigenwerten und Eigenvektoren einer oberen Hessenberg-Matrix. Im symmetrischen Fall wird das Problem sogar auf eine symmetrische Tridiagonalmatrix reduziert.

8.5 Das QR-Verfahren

QR-Zerlegung und Ähnlichkeitstransformationen

Sei $A \in \mathbb{R}^{n\times n}$ beliebig. In Abschnitt 1.5 wurde gezeigt, dass sich A zerlegen lässt in der Form

$$A \;=\; QR$$

mit einer orthogonalen Matrix Q und einer oberen Dreiecksmatrix R (QR-Zerlegung). Dann erhält man

$$Q^TAQ \;=\; Q^T(QR)Q \;=\; RQ \quad .$$

Also sind A und RQ ähnlich und haben deshalb dieselben Eigenwerte. Gleichzeitig bedeutet dies, dass man hier bei der Ähnlichkeitstransformation mit einer Matrizenmultiplikation auskommt.

Definition: RQ heißt die *QR-Transformierte* von A.

In diesem Zusammenhang erweisen sich Hessenberg-Matrizen bzw. Tridiagonalmatrizen als besonders günstig. Denn einerseits spart man Rechenaufwand, andererseits gilt der folgende

Satz: Sei $B \in \mathbb{R}^{n\times n}$ eine obere Hessenberg-Matrix mit der QR-Zerlegung $B = QR$. Dann ist die QR-Transformierte wieder eine obere Hessenberg-Matrix.

Beweis: Übung.

Korollar: Ist B eine symmetrische Tridiagonalmatrix, so ist die QR-Transformierte wieder eine symmetrische Tridiagonalmatrix.

Beweis: Die QR-Transformierte

$$C \;:=\; Q^TBQ \;=\; RQ$$

ist nach obigem Satz eine obere Hessenberg-Matrix. Weiter gilt

$$C^T \;=\; (Q^TBQ)^T \;=\; Q^TB^TQ \;=\; Q^TBQ \;=\; C \quad ,$$

also ist C auch symmetrisch. Eine symmetrische Hessenberg-Matrix ist aber eine symmetrische Tridiagonalmatrix. □

Der QR-Algorithmus

1. Sei $A \in \mathbb{R}^{n\times n}$. Setze $A^{(1)} := A$.
2. Für $k = 1, 2, \ldots$ bilde von

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

die QR-Zerlegung $A^{(k)} = Q^{(k)}R^{(k)}$ und setze dann

$$A^{(k+1)} \;:=\; R^{(k)}Q^{(k)} \quad \text{(QR-Transformierte)}.$$

Diese Iterationsvorschrift nennt man den *QR-Algorithmus.*

Um den Rechenaufwand erträglich zu halten, wird bei der praktischen Durchführung zunächst die Vortransformation auf Hessenberg-Gestalt durchgeführt und der QR-Algorithmus nur auf die Hessenberg-Matrix angewandt. Deshalb sei im Folgenden A schon eine Hessenberg-Matrix. Dann sind alle $A^{(k)}$ ebenfalls obere Hessenberg-Matrizen (vgl. oben).
Beim QR-Algorithmus besteht natürlich der Wunsch, dass die Folge der $A^{(k)}$ gegen eine Matrix konvergiert, bei der man die Eigenwerte leicht ablesen kann. Hier haben wir folgende Aussagen:

1. **Betragsmäßig verschiedene Eigenwerte**
Es sei A eine obere Hessenberg-Matrix. Sind alle Eigenwerte von A betragsmäßig verschieden, d.h.

$$|\lambda_1| \ > \ |\lambda_2| \ > \ \cdots \ > |\lambda_n| \quad ,$$

so kann man unter einer weiteren Voraussetzung

$$\left|a_{j+1,j}^{(k)}\right| \ \leq \ c \left|\tfrac{\lambda_{j+1}}{\lambda_j}\right|^k \qquad (j = 1, \ldots, n-1) \tag{98}$$

zeigen mit einer Konstanten $c \in \mathbb{R}$. Daraus folgt $a_{j+1,j}^{(k)} \to 0$ für $k \to \infty$ und somit

$$a_{jj}^{(k)} \ \to \ \lambda_j \quad \text{ für } k \to \infty \quad (j = 1, \ldots, n).$$

Die Konvergenz kann aber sehr langsam sein. Um eine Konvergenzbeschleunigung zu erzielen, benutzt man die

2. **Shift-Strategie**
Sei μ eine Schätzung für λ_n. Berechne dann

$$A^{(k)} - \mu I \ = \ Q^{(k)} R^{(k)} \quad \text{ und setze } \quad A^{(k+1)} \ := \ R^{(k)} Q^{(k)} + \mu I \quad .$$

Auch hier ist $A^{(k+1)}$ ähnlich zu $A^{(k)}$, denn

$$\begin{aligned} A^{(k+1)} \ &= \ R^{(k)} Q^{(k)} + \mu I \ = \ (Q^{(k)})^T Q^{(k)} R^{(k)} Q^{(k)} + \mu (Q^{(k)})^T Q^{(k)} \\ &= \ (Q^{(k)})^T (Q^{(k)} R^{(k)} + \mu I) Q^{(k)} \ = \ (Q^{(k)})^T (A^{(k)} - \mu I + \mu I) Q^{(k)} \\ &= \ (Q^{(k)})^T A^{(k)} Q^{(k)} \quad . \end{aligned}$$

In (98) hat man dann die Abschätzung

$$\left|a_{n,n-1}^{(k)}\right| \ \leq \ c \left|\tfrac{\lambda_n - \mu}{\lambda_{n-1} - \mu}\right|^k$$

mit dem (hoffentlich) kleineren Faktor $\left|\frac{\lambda_n - \mu}{\lambda_{n-1} - \mu}\right|$. In jedem Iterationsschritt kann μ neu gewählt werden, zweckmäßigerweise als $\mu = a_{nn}^{(k)}$.

3. **Reduktion**
Hat $A^{(k)}$ nach wenigen Iterationen die Form

$$A^{(k)} \ \approx \ \left(\begin{array}{ccc|c} & & & * \\ & B & & \vdots \\ & & & * \\ \hline 0 & \cdots & 0 & \lambda_n \end{array}\right) \quad ,$$

so ist ein Eigenwert hinreichend gut lokalisiert, und man setzt den QR-Algorithmus nur noch für die Untermatrix B fort.

4. **Komplexe Eigenwerte**
Sei nun

$$|\lambda_1| > \cdots > |\lambda_r| = |\lambda_{r+1}| > \cdots > |\lambda_n|$$

und $\lambda_{r+1} = \bar{\lambda}_r$ (konjugiert komplexe Eigenwerte). Für große k gilt dann

$$A^{(k)} \approx \begin{pmatrix} \lambda_1 & * & \cdots & \cdots & \cdots & \cdots & \cdots & * \\ & \ddots & \ddots & & & & & \vdots \\ & & \lambda_{r-1} & \ddots & & & & \vdots \\ & & & a_{rr}^{(k)} & a_{r,r+1}^{(k)} & & & \vdots \\ & & & a_{r+1,r}^{(k)} & a_{r+1,r+1}^{(k)} & \ddots & & \vdots \\ & & & & & \lambda_{r+2} & \ddots & \vdots \\ & & & & & & \ddots & * \\ & & & & & & & \lambda_n \end{pmatrix}$$

mit der 2×2-Matrix

$$\Lambda^{(k)} = \begin{pmatrix} a_{rr}^{(k)} & a_{r,r+1}^{(k)} \\ a_{r+1,1}^{(k)} & a_{r+1,r+1}^{(k)} \end{pmatrix} .$$

Zwar konvergiert die Folge $\Lambda^{(k)}$ i.A. nicht gegen eine 2×2-Matrix; aber $\Lambda^{(k)}$ besitzt ein Paar konjugiert komplexer Eigenwerte $(\varrho^{(k)}, \bar{\varrho}^{(k)})$, welches für $k \to \infty$ gegen $(\lambda_r, \lambda_{r+1})$ konvergiert.
Auch hier gibt es geeignete Shift-Strategien.

8.6 Das Verfahren von Hyman

Das charakteristische Polynom einer Hessenberg-Matrix

Ist $A \in \mathbb{R}^{n \times n}$ eine beliebige Matrix, so ist zur Eigenwertbestimmung der Weg über die Nullstellenberechnung des charakteristischen Polynoms

$$p(\mu) = \det(A - \mu I)$$

nicht empfehlenswert (Instabilitäten, Aufwand).

Anders ist die Situation, wenn die Matrix eine obere Hessenberg-Gestalt (bzw. eine symmetrische Tridiagonalgestalt) hat. Denn in diesem Fall ist es relativ einfach, bei vorgegebenem Argument μ den Funktionswert $p(\mu)$ und die Ableitung $p'(\mu)$ des charakteristischen Polynoms zu berechnen. Dabei ist es nicht nötig, das charakteristische Polynom als Ganzes durch Koeffizienten darzustellen. Die Situation ist ähnlich wie bei der Auswertung des Interpolationspolynoms ohne eine analytische Darstellung durch den Neville-Algorithmus.

Sind wir in der Lage, Funktionswert (und Ableitung) bei jedem Argument bequem zu ermitteln, so können wir zur Berechnung einer Nullstelle die in Paragraph 5 angegebenen Verfahren verwenden. Das Newton-Verfahren wird die schnellste Konvergenz zeigen, ist

aber durch den lokalen Charakter (vgl. Satz über die lokale Konvergenz) eingeschränkt. Im Allgemeinen wird man deshalb zunächst mit einem anderen Nullstellenverfahren (z.B. iterierte inverse Interpolation) beginnen. Ist man nach einigen Schritten in der Nähe einer Nullstelle angekommen, so kann man auf das Newton-Verfahren umsteigen.

Sei nun

$$B = \begin{pmatrix} b_{11} & \cdots & \cdots & \cdots & b_{1n} \\ b_{21} & \ddots & & & \vdots \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & b_{n,n-1} & b_{nn} \end{pmatrix}$$

eine obere Hessenberg-Matrix mit $b_{i+1,i} \neq 0$ $(i = 1, ..., n-1)$.

Eine solche Matrix heißt *unzerlegbar*.

Diese Zusatzannahme bedeutet keine wesentliche Einschränkung. Denn falls $b_{k+1,k} = 0$ gilt für ein k, so folgt

$$\det(B - \mu I) = \det(B_1 - \mu I_1) \cdot \det(B_2 - \mu I_2)$$

mit zwei oberen Hessenberg-Matrizen kleinerer Dimension. Man betrachtet dann diese Matrizen mit kleinerer Dimension.

Die folgende Methode zur Berechnung von $p(\mu)$ und $p'(\mu)$ stammt von Hyman.
Sei $\mu \in \mathbb{R}$ fest. Betrachte das lineare Gleichungssystem

$$(B - \mu I)x = \alpha e_1 \tag{99}$$

oder ausgeschrieben

$$\begin{aligned} (b_{11} - \mu)x_1 + b_{12}x_2 + \cdots + b_{1n}x_n &= \alpha \\ b_{21}x_1 + (b_{22} - \mu)x_2 + \cdots + b_{2n}x_n &= 0 \\ &\vdots \\ b_{n,n-1}x_{n-1} + (b_{nn} - \mu)x_n &= 0 \end{aligned}$$

Anmerkung: Jeder Eigenvektor y zum Eigenwert λ einer unzerlegbaren oberen Hessenberg-Matrix erfüllt $y_n \neq 0$; denn mit $y_n = 0$ folgt aus $(B - \lambda I)y = 0$ durch Rückwärtsauflösen $y_{n-1} = y_{n-2} = \ldots = y_1 = 0$, im Widerspruch zur Definition eines Eigenvektors.

Setze nun $x_n := 1$. Damit lassen sich $\alpha,\ x_1, \ldots, x_{n-1}$ eindeutig bestimmen (hängen aber vom gegebenen μ ab):

- Die letzte Gleichung im obigen System liefert x_{n-1},
- die vorletzte Gleichung liefert x_{n-2},

  usw.
- die 2. Gleichung ergibt x_1,
- aus der 1. Gleichung erhält man schließlich $\alpha = \alpha(\mu)$.

Bis auf einen Faktor stimmt $\alpha(\mu)$ mit dem charakteristischen Polynom von B überein:

Lemma: Es gilt

$$\alpha \;=\; \alpha(\mu) \;=\; \frac{(-1)^{n+1}}{b_{21}b_{32}\cdots b_{n,n-1}}\det(B-\mu I) \quad .$$

Beweis: Betrachte das lineare Gleichungssystem (99). Für x_n liefert die Cramersche Regel

$$1 \;=\; x_n \;=\; \frac{\det C}{\det(B-\mu I)} \tag{100}$$

mit

$$C \;=\; \begin{pmatrix} b_{11}-\mu & b_{12} & \cdots & b_{1,n-1} & \alpha \\ b_{21} & \ddots & & \vdots & 0 \\ 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & b_{n-1,n-1}-\mu & \vdots \\ 0 & \cdots & 0 & b_{n,n-1} & 0 \end{pmatrix}$$

Der Laplacesche Entwicklungssatz liefert

$$\det C \;=\; (-1)^{n+1}\alpha b_{21}b_{32}\cdots b_{n,n-1} \quad .$$

Einsetzen in (100) und Auflösen nach α ergibt dann die Behauptung. □

Betrachtet man $\alpha, x_1, \ldots, x_{n-1}$ als Funktionen von μ und differenziert diese, so folgt aus dem obigen Gleichungssystem (mit $x_i' := x_i'(\mu)$)

$$\begin{aligned} (b_{11}-\mu)x_1' - x_1 + b_{12}x_2' + \;\cdots\; + b_{1,n-1}x_{n-1}' &= \alpha'(\mu) \\ b_{21}x_1' + (b_{22}-\mu)x_2' - x_2 + \;\cdots\; + b_{2,n-1}x_{n-1}' &= 0 \\ &\vdots \\ b_{n,n-1}x_{n-1}' - x_n &= 0 \end{aligned}$$

(beachte $x_n = 1$, also $x_n' = 0$). Nachdem für festes μ die Größen $\alpha, x_1, \ldots, x_{n-1}$ schon berechnet sind, bestimmt man hieraus durch Rückwärtsauflösen die Größen $x_{n-1}', \ldots, x_1'$ und schließlich $\alpha'(\mu)$. Wir lösen also (bei festem μ) das Gleichungssystem

$$(B-\mu I)x' \;=\; \alpha'(\mu)e_1 + x \quad .$$

Fazit: Für ein gegebenes μ können wir $\alpha(\mu)$ und $\alpha'(\mu)$ einfach bestimmen (jeweils durch Lösen eines gestaffelten Gleichungssystems) und dann die bekannten Verfahren zur Berechnung der Nullstellen von $\alpha(\mu)$ anwenden. Diese Nullstellen sind nach dem obigen Lemma die (reellen) Eigenwerte von B.

Bestimmung von Eigenvektoren

Das Verfahren von Hyman liefert auch Approximationen für einen Eigenvektor. Dabei ist aber zu beachten, dass wir von einer Hessenberg-Matrix ausgehen und nur Eigenvektoren zu dieser Matrix berechnet werden. Eigenvektoren der ursprünglichen Matrix sind dann über die Transformationen zurück zu verfolgen, welche die Ausgangsmatrix auf Hessenberg-Gestalt gebracht haben.

Beim Verfahren von Hyman wird eine Nullstelle der Funktion $\alpha(\mu)$ berechnet. Ist $\bar{\mu}$ eine Nullstelle, so gilt in (99)

$$(B - \bar{\mu}I)x \;=\; 0 \qquad \text{und somit} \qquad Bx \;=\; \bar{\mu}x \qquad .$$

Das bedeutet, dass die im Algorithmus (mit $\bar{\mu}$) berechneten Zahlen $x_1, \ldots, x_{n-1}$ und $x_n = 1$ die Komponenten eines Eigenvektors $x = (x_1, \ldots, x_n)$ sind.

Da die Komponenten bei der Auswertung der Funktion α ohnehin anfallen, liefert (beim Verfahren von Hyman) die Nullstellenberechnung ohne zusätzlichen Aufwand eine Approximation für die zugehörigen Eigenvektoren.

8.7 Eigenwerte und Eigenvektoren in Matlab

Zur Bestimmung der Eigenwerte und Eigenvektoren einer Matrix A gibt es in Matlab die Funktion `eig`. Der Aufruf

```
d = eig(A)
```

liefert die Eigenwerte von A in einem Vektor `d`. Dagegen liefert

```
[V,D] = eig(A)
```

als Ergebnis zwei Matrizen derselben Größe wie A. Dabei ist `D` eine Diagonalmatrix mit den Eigenwerten in der Diagonalen, `V` enthält in den **Spalten** die zugehörigen Eigenvektoren, jeweils zur euklidschen Länge 1 normiert. Es gilt dann die Beziehung $VD = AV$. Ist die Matrix A diagonalisierbar, so bedeutet dies $D = V^{-1}AV$ (Ähnlichkeitstransformation).

Für schwachbesetzte Matrizen (sparse matrices) gibt die Matlab-Funktion `eigs`.

Abschließend wollen wir das QR-Verfahren (vgl. Abschnitt 8.5) mit Hilfe eines Matlab-Programmes veranschaulichen. Die Vortransformation auf obere Hessenberg-Gestalt erreicht man in Matlab durch

```
[P,H] = hess(A)
```

Dabei ist `H` die Hessenberg-Matrix und `P` die Transformationsmatrix, so dass $H = P^T AP$ gilt. Die QR-Zerlegung einer Matrix A erhalten wir in Matlab durch

```
[Q,R] = qr(A)
```

Dabei ist `Q` eine orthogonale und `R` eine obere Dreiecksmatrix. Das folgende Matlab-Programm demonstriert die einzelnen Schritte des QR-Verfahrens:

```
%   Es wird das QR-Verfahren demomstriert
disp(' Ausgangsmatrix A = ');
A = [1 1 1 1;1 2 2 1;1 2 3 4; 1 1 4 1]
ant = input(' Weiter mit Eingabe-Taste ','s');
disp('  Transformation auf Hessenberg-Form ');
[P,H] = hess(A)
ant = input(' Weiter mit Eingabe-Taste ','s');
disp('  Probe:  A = P*H*P'' ');
A
P*H*P'
ant = input(' Weiter mit Eingabe-Taste ','s');
```

```
disp('   QR-Verfahren (mit der Hessenberg-Matrix)');
weiter='j';
count = 0;
while weiter=='j'
  count = count + 1;
  [Q,R] = qr(H);
  text=sprintf(' nach dem %g-ten Schritt: \n',count);
  disp(text);
  H = R*Q
  ant = input(' Weiter mit Eingabe-Taste ','s');
end
```

9 Iterative Verfahren

9.1 Iteration

Das Newton-Verfahren (vgl. Abschnitt 5.2) ist ein Spezialfall einer allgemeineren Klasse von Verfahren, nämlich der Iterationsverfahren.
Es sei $D \subset \mathbb{R}^n$ und $\Phi : D \to D$ eine stetige Funktion.

Definition: Ein $\xi \in D$ heißt *Fixpunkt* von Φ, falls gilt

$$\xi \;=\; \Phi(\xi) \qquad .$$

Man kann das Auffinden einer Nullstelle von $f(s)$ auf das Suchen eines Fixpunktes zurückführen. Aus dem Newton-Verfahren (vgl. (41)) ergibt sich die Fixpunktform

$$\Phi(t) \;:=\; t - \frac{f(t)}{f'(t)} \qquad . \tag{101}$$

Denn aus $\xi = \Phi(\xi)$ folgt sofort

$$\xi \;=\; \xi - \frac{f(\xi)}{f'(\xi)}$$

und somit $f(\xi) = 0$.

Fixpunkte werden iterativ berechnet: Sei ein Startwert $x^{(0)}$ vorgegeben. Dann setzt man

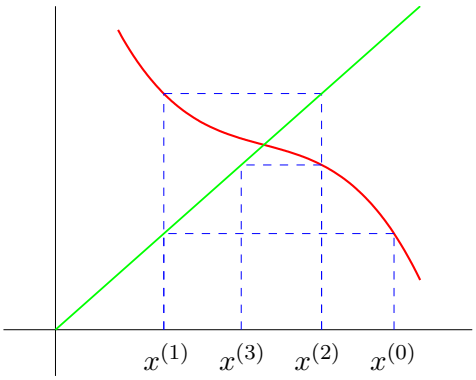
$$x^{(i+1)} := \Phi(x^{(i)}) \qquad (i = 0, 1, 2, \ldots) \; . \tag{102}$$

Frage: Wann existiert $\lim\limits_{i\to\infty} x^{(i)}$?

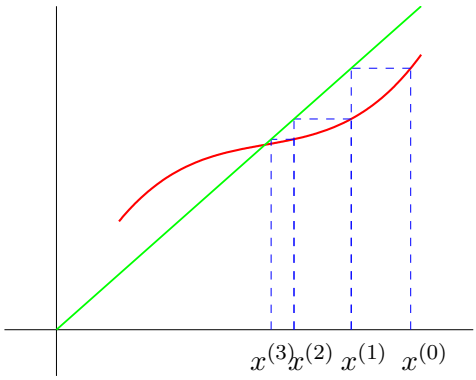
Falls der Grenzwert existiert, so ist $\xi := \lim\limits_{i\to\infty} x^{(i)}$ ein Fixpunkt von Φ (folgt aus der Stetigkeit von Φ).

Graphische Darstellung eines Iterationsverfahrens

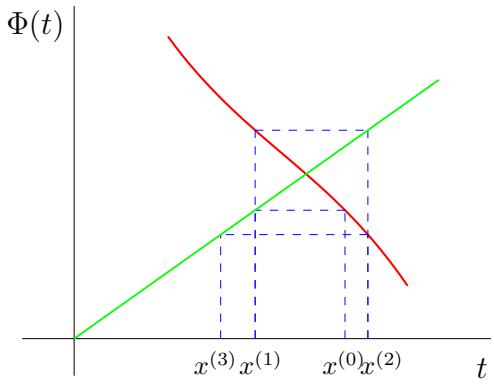
alternierende Konvergenz



monotone Konvergenz



keine Konvergenz



Zur Formulierung eines Konvergenzsatzes betrachten wir allgemeiner metrische Räume.

Definition: Sei (X, d) ein metrischer Raum und $\Phi : X \to X$ eine Abbildung. Φ heißt *kontrahierend*, falls es eine reelle Zahl $L < 1$ gibt mit der Eigenschaft

$$d(\Phi(t), \Phi(s)) \;\leq\; L \cdot d(t, s) \qquad \text{für alle } s, t \in X \quad .$$

Satz (Banachscher Fixpunktsatz):
Sei (X, d) ein vollständiger metrischer Raum und $\Phi : X \to X$ kontrahierend. Dann gilt

1. Φ besitzt genau einen Fixpunkt.
2. Für jeden Startvektors $x^{(0)} \in X$ konvergiert die Iterationsfolge $x^{(i+1)} := \Phi(x^{(i)})$ gegen den Fixpunkt.

Beweis: vgl. z. B. Hämmerlin-Hoffmann [9].

Für das Newton-Verfahren haben wir $X = [a, b]$ und als Metrik $d(s, t) = |s - t|$. Hier lautet die Kontraktionsbedingung

$$|\Phi(t) - \Phi(s)| \;\leq\; L\,|t - s| \qquad \text{für alle } s, t \in [a, b].$$

Diese Bedingung ist zum Beispiel erfüllt, falls Φ stetig differenzierbar ist und

$$\|\Phi'\|_\infty \;:=\; \max\{|\Phi'(t)| \,:\, t \in [a, b]\} \;\leq L < 1 \tag{103}$$

gilt (Mittelwertsatz).

An dieser Stelle erinnern wir nochmals an die Begriffe *lokale und globale Konvergenz*:

Definition:
Konvergiert die Folge der Iterierten (102) nur für Anfangswerte x_0 aus einer Umgebung $U \subset D$ des Fixpunktes ξ, so heißt die Iteration *lokal konvergent*. Konvergiert die Iteration für jeden Startwert aus dem gesamten Definitionsbereich D, so heißt das Verfahren *global konvergent*.

Anmerkung: Lokale Konvergenz liegt vor, wenn Φ nur auf $U \subset D$ mit $\xi \in U$ kontrahierend ist, nicht jedoch auf ganz D.

Konvergenz des Newton-Verfahrens

Den Satz über die lokale Konvergenz des Newton-Verfahrens (vgl. Abschnitt 5.3) kann man auch mit Hilfe des Banachschen Fixpunktsatzes beweisen:

Satz: (lokale Konvergenz). Sei $f \in C^2[a,b]$ und $\xi \in [a,b]$ mit $f(\xi) = 0$ und $f'(\xi) \neq 0$. Dann gibt es eine Umgebung U von ξ, so dass für jeden Startwert $x^{(0)} \in U$ das Newton-Verfahren konvergiert.

Beweis: Für das Newton-Verfahren gilt

$$\Phi(t) \;=\; t - \frac{f(t)}{f'(t)} \quad .$$

Φ ist stetig differenzierbar mit

$$\Phi'(t) \;=\; 1 - \frac{f'(t)^2 - f(t)f^{(2)}(t)}{f'(t)^2} \;=\; \frac{f(t)f^{(2)}(t)}{f'(t)^2} \quad .$$

Insbesondere gilt dann $\Phi'(\xi) = 0$. Sei $0 < L < 1$ fest gegeben. Aus den Voraussetzungen $f(\xi) = 0$ und $f'(\xi) \neq 0$ folgt, dass es eine Umgebung $U = [\xi - \varepsilon, \xi + \varepsilon]$ gibt mit

$$|\Phi'(t)| \;=\; \left|\tfrac{f(t)f^{(2)}(t)}{f'(t)^2}\right| \;\leq\; L \;<\; 1 \qquad \text{für alle } t \in U,$$

nach (103) ist Φ kontrahierend auf U. Der Banachsche Fixpunktsatz liefert dann die Behauptung. □

9.2 Vektornormen und Matrixnormen

Es sei $\mathbb{K}$ der Körper der reellen oder komplexen Zahlen ($\mathbb{K} = \mathbb{R}$ oder $\mathbb{K} = \mathbb{C}$). Mit $\mathbb{K}^{n\times n}$ bezeichnen wir den Raum der $n \times n$ - Matrizen.

Definition: $\|\ \|$ heißt eine Norm auf $\mathbb{K}^n$ (oder Vektornorm), wenn gilt:

(N1) $\|x\| > 0$ für alle $x \in \mathbb{K}^n \setminus \{0\}$,

(N2) $\|\lambda x\| = |\lambda| \cdot \|x\|$ für alle $x \in \mathbb{K}^n$ und $\lambda \in \mathbb{K}$,

(N3) $\|x + y\| \leq \|x\| + \|y\|$ für alle $x, y \in \mathbb{K}^n$.

Beispiele: Es sei $x = (x_1, \ldots, x_n)^T$. Vektornormen sind

$$\|x\|_\infty \;:=\; \max_{i=1,\ldots,n}\{|x_i|\} \qquad \text{(Maximum-Norm)},$$

$$\|x\|_2 \;:=\; \sqrt{\sum_{i=1}^{n} |x_i|^2} \qquad \text{(Euklidsche Norm)},$$

$$\|x\|_1 \;:=\; \sum_{i=1}^{n} |x_i| \qquad (l_1\text{-Norm}).$$

Ebenso lassen sich Normen auf dem Raum der Matrizen definieren. Man spricht dann von Matrixnormen. Zur besseren Unterscheidung bezeichnen wir diese mit $N(A)$.

Definitionen:

1. $N(\,\cdot\,)$ heißt eine *Matrixnorm* auf $\mathbb{K}^{n\times n}$, wenn gilt:

$$\begin{aligned}
&\text{(N1)} \quad N(A) > 0 \text{ für alle } A \in \mathbb{K}^{n\times n} \setminus \{0\} \quad , \\
&\text{(N2)} \quad N(\lambda A) = |\lambda| \cdot N(A) \text{ für alle } A \in \mathbb{K}^{n\times n} \text{ und } \lambda \in \mathbb{K} \quad , \\
&\text{(N3)} \quad N(A+B) \leq N(A) + N(B) \text{ für alle } A, B \in \mathbb{K}^{n\times n} \quad .
\end{aligned}$$

2. Die Matrixnorm N heißt *submultiplikativ*, falls gilt:

$$N(A\cdot B) \leq N(A)\cdot N(B) \quad \text{für alle } A, B \in \mathbb{K}^{n\times n} \, .$$

3. Die Matrixnorm N und die Vektornorm $\|\ \|$ heißen *verträglich*, falls gilt:

$$\|Ax\| \leq N(A)\cdot\|x\| \quad \text{für alle } A \in \mathbb{K}^{n\times n} \text{ und } x \in \mathbb{K}^n \, .$$

Beispiele: Es sei

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix} \quad .$$

Dann erhält man Matrixnormen durch

$$N_E(A) := \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} |a_{ij}|^2} \quad \text{(Euklidische Norm, Schur-Norm)} \, ,$$

$$N_Z(A) := \max_{i=1,\ldots,n} \left\{ \sum_{k=1}^{n} |a_{ik}| \right\} \quad \text{(Zeilensummennorm)} \, ,$$

$$N_S(A) := \max_{k=1,\ldots,n} \left\{ \sum_{i=1}^{n} |a_{ik}| \right\} \quad \text{(Spaltensummennorm)} \, .$$

Verträglich sind zum Beispiel:

N_E mit $\|\ \|_2$,

N_Z mit $\|\ \|_\infty$,

N_S mit $\|\ \|_1$.

Dagegen ist N_Z nicht verträglich mit $\|\ \|_1$.

Jede Vektornorm $\|\ \|$ definiert eine submultiplikative, verträgliche Matrixnorm durch

$$\text{lub}(A) := \max\{\|Ax\| \,:\, x \in \mathbb{K}^n \text{ mit } \|x\| = 1\} \quad .$$

Diese nennt man die von $\|\ \|$ *induzierte Matrixnorm* oder auch *lub-Norm* .

So liefert zum Beispiel $\| \ \|_1$ die Spaltensummennorm N_S und $\| \ \|_\infty$ die Zeilensummennorm N_Z.

Anmerkungen:
1. lub bedeutet least upper bound.
2. $c := \text{lub}(A)$ ist die kleinste Konstante, so dass für alle $x \in \mathbb{K}^n$ die Abschätzung $\|Ax\| \leq c \cdot \|x\|$ gilt.

Definition: Für $A \in \mathbb{K}^{n \times n}$ heißt die Größe

$$\varrho(A) \ = \ \max\{\, |\lambda| \in \mathbb{C} \ : \ \lambda \text{ ist ein Eigenwert von } A\}$$

(Betrag des betragsmäßig größten Eigenwertes) der *Spektralradius* von A.

Für jede lub-Norm gilt

$$\varrho(A) \ \leq \ \text{lub}(A) \quad . \tag{104}$$

Um dies einzusehen, betrachten wir einen Eigenwert λ_i von A und einen zugehörigen Eigenvektor v_i mit $\|v_i\| = 1$. Dann gilt

$$\|Av_i\| \ = \ \|\lambda_i v_i\| \ = \ |\lambda_i| \, \|v_i\| \ = \ |\lambda_i| \quad ,$$

woraus sich dann $\text{lub}(A) \geq \varrho(A)$ ergibt.

Lemma: Sei $A \in \mathbb{K}^{n \times n}$ mit $\varrho(A) < 1$. Dann gibt es eine Vektornorm $\| \ \|$, so dass für die zugehörige lub-Norm gilt

$$\varrho(A) \ \leq \ \text{lub}(A) \ < \ 1 \quad .$$

Beweis: vgl. Stoer-Bulirsch [8].

9.3 Indirekte Verfahren für lineare Gleichungssysteme

Fixpunktformen

Gegeben sei eine reguläre Matrix $A \in \mathbb{K}^{n \times n}$. Zu lösen sei das lineare Gleichungssystem

$$Ax \ = \ b \quad . \tag{105}$$

Bei iterativen Verfahren sucht man eine äquivalente Fixpunktform

$$x \ = \ Tx + g \ =: \Phi(x) \tag{106}$$

und bestimmt dann mittels der Iterationsvorschrift

$$x^{(i+1)} \ := \ \Phi(x^{(i)})$$

einen Näherungswert für die gesuchte Lösung von (105), wobei zunächst ein geeigneter Startwert zu wählen ist.

Lemma: Sei $B \in \mathbb{K}^{n \times n}$ regulär. Durch Zeilenvertauschungen kann B in eine Matrix A übergeführt werden, deren Diagonalelemente von Null verschieden sind.

Beweis: Übung.

Sei also

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & & \vdots \\ \vdots & & & \vdots \\ a_{n1} & \cdots & \cdots & a_{nn} \end{pmatrix}$$

mit $a_{ii} \neq 0$ für $i = 1, \ldots, n$. Wir bilden folgende Matrizen:

$$D := \begin{pmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & a_{nn} \end{pmatrix} , \qquad L := \begin{pmatrix} 0 & \cdots & \cdots & 0 \\ a_{21} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ a_{n1} & \cdots & a_{n,n-1} & 0 \end{pmatrix} ,$$

$$U := \begin{pmatrix} 0 & a_{12} & \cdots & a_{1n} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & a_{n-1,n} \\ 0 & \cdots & \cdots & 0 \end{pmatrix} .$$

Damit ergibt sich die Zerlegung

$$A = L + D + U \quad .$$

Das Gleichungssystem $Ax = b$ ist somit gleichwertig zu $(L+D+U)x = b$. Hieraus lassen sich nun Fixpunktformen herleiten. Da nach Voraussetzung $a_{ii} \neq 0$ $(i = 1, \ldots, n)$ gilt, sind die Matrizen $(L + D)$ und D invertierbar.

Es gilt

$$(L + D + U)x = b \quad \Leftrightarrow \quad Dx = -(L + U)x + b \quad \Leftrightarrow \quad x = -D^{-1}(L + U)x + D^{-1}b \, .$$

Hier erhält man die Fixpunktform (106) mit

$$T := -D^{-1}(L + U) \quad \text{und} \quad g := D^{-1}b \quad . \tag{107}$$

Definition: Das hier erzeugte Iterationsverfahren heißt *Gesamtschrittverfahren (Jacobi-Verfahren).*

Eine zweite Möglichkeit für eine Fixpunktform folgt aus den Umformungen

$$(L+D+U)x = b \quad \Leftrightarrow \quad (L+D)x = -Ux+b \quad \Leftrightarrow \quad x = -(L+D)^{-1}Ux+(L+D)^{-1}b \, .$$

Hier sind

$$T := -(L + D)^{-1}U \, , \qquad \text{und} \quad g := (L + D)^{-1}b \, . \tag{108}$$

Definition: Die so erzeugte Iteration heißt *Einzelschrittverfahren (Gauß-Seidel-Verfahren).*

Gesamtschrittverfahren

Man wählt einen beliebigen Startwert $x^{(0)} \in \mathbb{K}^n$ und verwendet die Iterationsvorschrift

$$x^{(i+1)} := -D^{-1}(L + U)x^{(i)} + D^{-1}b$$

oder in Komponentendarstellung

$$x_k^{(i+1)} \;=\; \frac{1}{a_{kk}} \left(-\sum_{\substack{j\,=\,1\\ j\,\neq\,k}}^{n} a_{kj} x_j^{(i)} + b_k \right) \qquad (k = 1, 2, \ldots, n)\,.$$

Einzelschrittverfahren

Man wählt $x^{(0)} \in \mathbb{K}^n$ beliebig und verwendet dann die Iterationsvorschrift

$$x^{(i+1)} \;:=\; -(L+D)^{-1} U x^{(i)} + (L+D)^{-1} b \qquad (i = 0, 1, 2, \ldots)\,.$$

Aus

$$(L+D)x^{(i+1)} \;=\; -Ux^{(i)} + b$$

folgt für die k-te Komponente

$$\sum_{j=1}^{k} a_{kj} x_j^{(i+1)} \;=\; -\sum_{j=k+1}^{n} a_{kj} x_j^{(i)} + b_k \qquad .$$

Auflösen nach $x_k^{(i+1)}$ liefert für das Einzelschrittverfahren die komponentenweise Darstellung

$$x_k^{(i+1)} \;=\; -\frac{1}{a_{kk}} \left(\sum_{j=1}^{k-1} a_{kj} x_j^{(i+1)} + \sum_{j=k+1}^{n} a_{kj} x_j^{(i)} - b_k \right) \qquad (k = 1, \ldots, n).$$

Unterschiede

Beim Gesamtschrittverfahren wird jede Komponente $x_k^{(i+1)}$ aus $x_1^{(i)}, \ldots, x_n^{(i)}$ berechnet:

$$\begin{pmatrix} x_1^{(i)} \\ x_2^{(i)} \\ \vdots \\ x_n^{(i)} \end{pmatrix} \quad \to \quad \begin{pmatrix} x_1^{(i+1)} \\ x_2^{(i+1)} \\ \vdots \\ x_n^{(i+1)} \end{pmatrix} \qquad .$$

Dagegen berechnet man bei Einzelschrittverfahren die Komponente $x_k^{(i+1)}$ aus $x_1^{(i+1)}, \ldots, x_{k-1}^{(i+1)}, x_{k+1}^{(i)}, \ldots, x_n^{(i)}$:

$$\begin{pmatrix} x_1^{(i)} \\ x_2^{(i)} \\ x_3^{(i)} \\ \vdots \\ x_{n-1}^{(i)} \\ x_n^{(i)} \end{pmatrix} \quad \to \quad \begin{pmatrix} x_1^{(i+1)} \\ x_2^{(i)} \\ x_3^{(i)} \\ \vdots \\ x_{n-1}^{(i)} \\ x_n^{(i)} \end{pmatrix} \quad \to \quad \begin{pmatrix} x_1^{(i+1)} \\ x_2^{(i+1)} \\ x_3^{(i)} \\ \vdots \\ x_{n-1}^{(i)} \\ x_n^{(i)} \end{pmatrix} \quad \to \quad \begin{pmatrix} x_1^{(i+1)} \\ x_2^{(i+1)} \\ x_3^{(i+1)} \\ \vdots \\ x_{n-1}^{(i)} \\ x_n^{(i)} \end{pmatrix}$$

$$\to \cdots \to \quad \begin{pmatrix} x_1^{(i+1)} \\ x_2^{(i+1)} \\ x_3^{(i+1)} \\ \vdots \\ x_{n-1}^{(i+1)} \\ x_n^{(i)} \end{pmatrix} \quad \to \quad \begin{pmatrix} x_1^{(i+1)} \\ x_2^{(i+1)} \\ x_3^{(i+1)} \\ \vdots \\ x_{n-1}^{(i+1)} \\ x_n^{(i+1)} \end{pmatrix} \qquad .$$

Aufwand: Für einen Iterationsschritt benötigt man bei beiden Verfahren ungefähr n^2 Punkt- und n^2 Strichoperationen. Für große Gleichungssysteme (z.B. $n = 1000$) bedeuten dann 10 oder 20 Iterationsschritte wesentlich weniger Aufwand als bei direkten Verfahren (z.B. LR-Zerlegung, dort werden ungefähr $\frac{n^3}{3}$ Punktoperationen und $\frac{n^3}{3}$ Strichoperationen benötigt).

Konvergenzfragen

Für Konvergenzaussagen bei Iterationsverfahren steht uns der Banachsche Fixpunktsatz zur Verfügung, den wir hier auf die Iterationsfunktion

$$\Phi(x) \ := \ Tx + g \tag{109}$$

anwenden (T eine $n \times n$ - Matrix). Sei $\| \ \|$ eine Vektornorm und $N(\cdot)$ eine verträgliche Matrixnorm. Dann gilt für alle $x, y \in \mathbb{K}^n$

$$\|\Phi(x) - \Phi(y)\| \ = \ \|Tx - Ty\| \ = \ \|T(x-y)\| \ \leq \ N(T) \cdot \|x - y\| \quad .$$

Damit ist die Kontraktionsbedingung erfüllt, falls

$$N(T) \ < \ 1 \tag{110}$$

gilt für irgend eine Matrixnorm (beachte: alle Matrixnormen sind äquivalent). Man hat dann Konvergenz für jeden Startwert $x^{(0)} \in \mathbb{K}^n$.

Genau-dann-Kriterium

Satz: Genau dann konvergiert die Iterationsfolge

$$x^{(i+1)} \ := \ Tx^{(i)} + g$$

für jeden Startwert $x^{(0)} \in \mathbb{K}^n$ gegen den Fixpunkt $\overline{x}$, wenn $\varrho(T) < 1$ gilt.

Beweis: Sei $\bar{x}$ ein Fixpunkt von $Tx + g$ und $x^{(0)} \in \mathbb{K}^n$ ein beliebiger Startwert. Dann gilt

$$\begin{aligned} x^{(i)} - \bar{x} \ &= \ Tx^{(i-1)} + g - T\bar{x} - g \ = \ T(x^{(i-1)} - \bar{x}) \\ &= \ T(Tx^{(i-2)} + g - T\bar{x} - g) \ = \ T^2(x^{(i-2)} - \bar{x}) \\ &\vdots \\ &= \ T^i(x^{(0)} - \bar{x}) \quad . \end{aligned}$$

"$\Rightarrow$" Sei λ ein Eigenwert von T und y ein zugehöriger Eigenvektor. Wähle als Startwert $x^{(0)} := y + \bar{x}$. Dann erhält man

$$0 \ = \ \lim_{i\to\infty} (x^{(i)} - \bar{x}) \ = \ \lim_{i\to\infty} T^i(x^{(0)} - \bar{x}) \ = \ \lim_{i\to\infty} T^i y \ = \ \lim_{i\to\infty} \lambda^i y \quad .$$

Daraus folgt $|\lambda| < 1$ und somit $\varrho(T) < 1$.

"$\Leftarrow$" Sei nun $\varrho(T) < 1$. Dann gibt es nach dem Lemma aus Abschnitt 9.2 eine Vektornorm $\| \ \|$, so dass für die zugehörige induzierte Matrixnorm gilt

$$\varrho(T) \ \leq \ \mathrm{lub}(T) \ < \ 1 \quad .$$

Nach (110) folgt die Konvergenz der Iterationsfolge für jeden beliebigen Startwert. □

Anmerkung: $\varrho(T)$ ist i.A. nur schwer zu bestimmen. Deshalb sucht man nach anderen (hinreichenden) Konvergenzkriterien, die leicht nachprüfbar sind.

Hinreichende Bedingung für das Gesamtschrittverfahren
Beim Gesamtschrittverfahren gilt

$$T_G \;:=\; -D^{-1}(L+U)$$

oder ausgeschrieben

$$T_G \;=\; -\begin{pmatrix} \frac{1}{a_{11}} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \frac{1}{a_{nn}} \end{pmatrix} \begin{pmatrix} 0 & a_{12} & \cdots & a_{1n} \\ a_{21} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{n-1,n} \\ a_{n1} & \cdots & a_{n,n-1} & 0 \end{pmatrix} \quad .$$

T_G lässt sich sofort mit Hilfe der Komponenten von A angeben. Wählt man als Matrixnorm die Zeilensummennorm N_Z, so folgt aus (110) als hinreichendes Kriterium

$$\sum_{\substack{j\,=\,1 \\ j\,\neq\,k}}^{n} |a_{kj}| \;<\; |a_{kk}| \qquad (k = 1, \cdots, n)\,. \tag{111}$$

Diese Bedingung nennt man das *Zeilensummenkriterium.*

Aus dem Zeilensummenkriterium ergibt sich durch Übergang zu transponierten Matrizen als hinreichendes Kriterium

$$\sum_{\substack{k\,=\,1 \\ k\,\neq\,j}}^{n} |a_{kj}| \;<\; |a_{jj}| \qquad (j = 1, \cdots n). \tag{112}$$

Diese Bedingung nennt man das *Spaltensummenkriterium.*

Beweis: (für das Spaltensummenkriterium)
Das Spaltensummenkriterium entspricht dem Zeilensummenkriterium für A^T. Also gilt für die Iterationsmatrix $T = -D^{-1}(U^T + L^T)$ die Bedingung $\rho(T) < 1$. Daraus folgt $\rho(T^T) < 1$. Weiter ist $T^T = -(L+U)D^{-1}$ und $D^{-1}T^TD = -D^{-1}(L+U) = T_G$, also auch $\rho(T_G) < 1$. □

Hinreichende Bedingungen für das Einzelschrittverfahren
Beim Einzelschrittverfahren lässt sich die Iterationsmatrix

$$T_E \;:=\; -(L+D)^{-1}U$$

nicht so einfach mit Hilfe der Koeffizienten von A ausdrücken, so dass man hier $N_Z(T_E)$ nicht so rasch bestimmen kann.

Satz: Das Zeilensummenkriterium und das Spaltensummenkriterium sind auch hinreichende Konvergenz-Bedingungen für das Einzelschrittverfahren.

Beweis: (für das Zeilensummenkriterium)
Sei $y \in \mathbb{K}^n$ beliebig und $z := T_E\, y$. Mittels vollständiger Induktion zeigen wir für die Komponenten von z

$$|z_k| \;\leq\; \frac{1}{|a_{kk}|} \sum_{\substack{j\,=\,1 \\ j\,\neq\,k}}^{n} |a_{kj}|\, \|y\|_\infty \;\leq\; N_Z(T_G)\, \|y\|_\infty \qquad (k = 1, \ldots, n). \tag{113}$$

Daraus erhält man sofort

$$\|z\|_\infty \;=\; \|T_E y\|_\infty \;\leq\; N_Z(T_G)\,\|y\|_\infty \qquad \text{für alle } y \in \mathbb{K}^n.$$

Da aber N_Z die von $\|\ \|_\infty$ induzierte Matrixnorm (lub-Norm) ist, folgt

$$N_Z(T_E) \;\leq\; N_Z(T_G) \;<\; 1$$

und damit die Behauptung.

Beweis der Behauptung (113) mittels Induktion:
Wir schreiben die Gleichung $z = T_E y$ um in $(L+D)z = -Uy$. Daraus folgt für die k-te Komponente

$$z_k \;=\; -\frac{1}{a_{kk}}\left(\sum_{j=1}^{k-1} a_{kj} z_j \;+\; \sum_{j=k+1}^{n} a_{kj} y_j\right) \qquad (k = 1, \cdots, n).$$

Für $k = 1$ gilt (Induktionsanfang)

$$\begin{aligned} |z_1| \;&\leq\; \frac{1}{|a_{11}|}\sum_{j=2}^{n} |a_{1j}||y_j| \;\leq\; \frac{1}{|a_{11}|}\sum_{j=2}^{n} |a_{1j}|\|y\|_\infty \\ &\leq\; N_Z(T_G)\,\|y\|_\infty \quad . \end{aligned}$$

Mit Hilfe der Induktionsvoraussetzung und $N_Z(T_G) < 1$ folgt für die k-te Komponente

$$\begin{aligned} |z_k| \;&\leq\; \frac{1}{|a_{kk}|}\left(\sum_{j=1}^{k-1} |a_{kj}||z_j| \;+\; \sum_{j=k+1}^{n} |a_{kj}||y_j|\right) \\ &\leq\; \frac{1}{|a_{kk}|}\left(\sum_{j=1}^{k-1} |a_{kj}|\, N_Z(T_G)\,\|y\|_\infty \;+\; \sum_{j=k+1}^{n} |a_{kj}|\|y\|_\infty\right) \\ &\leq\; \frac{1}{|a_{kk}|}\left(\sum_{j=1}^{k-1} |a_{kj}| \;+\; \sum_{j=k+1}^{n} |a_{kj}|\right)\|y\|_\infty \\ &\leq\; N_Z(T_G)\,\|y\|_\infty \quad . \;\square \end{aligned}$$

9.4 Nichtlineare Gleichungssysteme

Das mehrdimensionale Newton-Verfahren

Sei $U \subset \mathbb{R}^n$ und $F : U \to \mathbb{R}^n$ hinreichend glatt (mindestes einmal stetig differenzierbar). Gesucht ist ein $x \in U$ mit $F(x) = 0$ oder in Komponentendarstellung

$$\begin{aligned} f_1(x_1, \ldots, x_n) \;&=\; 0 \\ &\vdots \qquad . \\ f_n(x_1, \ldots, x_n) \;&=\; 0 \end{aligned}$$

Wie im univariaten Fall (vgl. Abschnitt 5.2) wird auch hier F lokal durch eine affin lineare Funktion ersetzt. Von dieser kann man eine Nullstelle bestimmen (Lösung eines linearen Gleichungssystems).

Als Ausgangspunkt dient uns dabei die Taylor-Entwicklung für den mehrdimensionalen Fall. Mit $DF(x)$ bezeichnen wir die Funktionalmatrix von F:

$$DF(x) \;=\; \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(x) & \cdots & \frac{\partial f_1}{\partial x_n}(x) \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1}(x) & \cdots & \frac{\partial f_n}{\partial x_n}(x) \end{pmatrix} \quad .$$

Ist F mindestens zweimal stetig differenzierbar, so gilt für $x, y \in U$ die Taylor-Darstellung

$$F(x) \;=\; F(y) \;+\; DF(y)(x-y) \;+\; \text{Rest} \quad .$$

Der Rest enthält dabei nur höhere partielle Ableitungen und ist von der Größenordnung $\mathcal{O}(\|x-y\|^2)$. Für festes y ist die Funktion

$$G(x) \;:=\; F(y) \;+\; DF(y)(x-y)$$

affin linear. Damit erhält man einen Näherungswert für eine Nullstelle von F durch Lösen des linearen Gleichungssystems

$$DF(y)x \;=\; DF(y)y \;-\; F(y) \quad . \tag{114}$$

Ist $DF(y)$ invertierbar, so folgt daraus

$$x \;=\; y - (DF(y))^{-1}F(y) \quad .$$

Sei nun $DF(x)$ invertierbar für alle $x \in U$ und $y^{(0)} \in U$ gegeben. Bilde die Iterationsvorschrift

$$y^{(k+1)} \;:=\; y^{(k)} \;-\; (DF(y^{(k)}))^{-1}F(y^{(k)}) \qquad (k = 0, 1, 2, \ldots).$$

Dies ist das *(mehrdimensionale) Newton-Verfahren.*

Im Fall $n = 1$ ergibt sich wieder das aus Abschnitt 5.2 bekannte (eindimensionale) Newton-Verfahren.

Praktische Durchführung

In der Numerik bestimmt man die Iterierten $y^{(k)}$ durch das lineare Gleichungssystem (114). Mit

$$v \;:=\; y^{(k)} \;-\; y^{(k+1)}$$

lösen wir das lineare Gleichungssystem

$$DF(y^{(k)})v \;=\; F(y^{(k)}) \tag{115}$$

und setzen dann

$$y^{(k+1)} \;:=\; y^{(k)} - v \quad .$$

Breche die Iteration ab, falls für vorgegebenes $\varepsilon > 0$ und für eine gewählte Vektornorm (z.B. Maximum-Norm oder Euklidsche Norm) gilt

$$\left\|F(y^{(k+1)})\right\| \;<\; \varepsilon \qquad \text{und} \qquad \left\|y^{(k+1)} - y^{(k)}\right\| \;<\; \varepsilon \quad .$$

Vereinfachte Form

Beim Newton-Verfahren ist in (115) in jedem Schritt ein Gleichungssystem mit der Koeffizientenmatrix $A_k := DF(y^{(k)})$ zu lösen. Dies bedeutet, dass in jedem Schritt zuerst diese Koeffizientenmatrix berechnet werden muss, was unter Umständen viel Rechenzeit kostet (n^2 Funktionsauswertungen).

Es gibt deshalb eine vereinfachte Form des Newton-Verfahrens, bei dem diese Koeffizientenmatrix konstant gehalten wird (d.h. $A_k = A$). Eine beliebte Wahl ist $A := DF(y^{(0)})$, dabei wird der Startvektor $y^{(0)}$ natürlich so gewählt, dass $DF(y^{(0)})$ invertierbar ist.

Dann erhält man

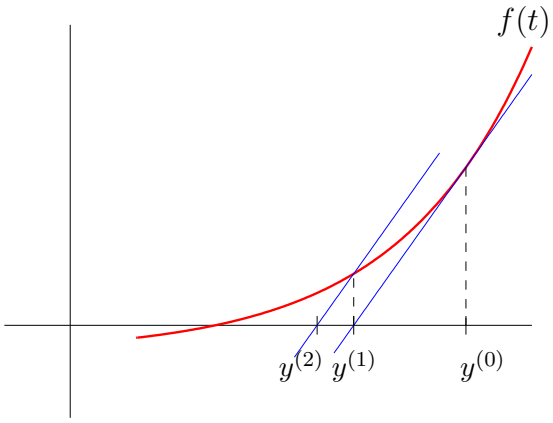
$$Av \;=\; F(y^{(k)})\,, \qquad y^{(k+1)} \;:=\; y^{(k)} - v \qquad .$$

Dieses Verfahren heißt *Parallelverfahren*.

Bei der praktischen Durchführung wird vorab von A eine Zerlegung (z.B. LR-Zerlegung oder QR-Zerlegung) bestimmt. Dann kann man die anfallenden Gleichungssysteme rasch lösen.

Zur Verdeutlichung des Parallelverfahrens schauen wir uns den skalaren Fall an. Wählt man dabei $A = f'(y^{(0)})$, so ergibt sich

$$y^{(k+1)} \;=\; y^{(k)} \;-\; \frac{f(y^{(k)})}{f'(y^{(0)})} \qquad .$$



Parallelverfahren im skalaren Fall

Im skalaren Fall spielt diese Methode eine geringere Rolle, da hier andere Alternativen zur Anwendung kommen (z.B. Bisektionsverfahren oder Sekantenverfahren).

Konvergenz des Newton-Verfahrens

Die Konvergenzverhältnisse sind beim n-dimensionalen Newton-Verfahren sehr verwickelt. Es ist möglich, dass die Newton-Folge (selbst wenn man sicherstellen kann, dass sie für alle Iterationsschritte gebildet werden kann) im Raum umherirrt, ohne sich irgendwo festzusetzen. Ferner können die Iterierten eine Zyklus bilden, d.h. es gilt $y^{(N)} = y^{(0)}$ für ein gewisses $N \in \mathbb{N}_1$.

Es gibt hier auch einen Satz, der Aussagen über die Konvergenz macht. Dieser ist jedoch von theoretischer Art, die praktische Anwendung ist sehr schwierig.

Satz (lokale Konvergenz):
Sei $U \subset \mathbb{R}^n$ offen und $F : U \to \mathbb{R}^n$ zweimal stetig differenzierbar. Es existiere ein $\bar{u} \in U$ mit $F(\bar{u}) = 0$. Ferner sei die Funktionalmatrix $DF(\bar{u})$ invertierbar. Dann gelten:

1. Es gibt eine Kugel

$$K_\varrho(\bar{u}) \;=\; \{x \in \mathbb{R}^n \;:\; \|x - \bar{u}\|_2 \;\leq\; \varrho\} \;\subset\; U$$

   mit $F(x) \neq 0$ für alle $x \in K_\varrho(\bar{u}) \setminus \{\bar{u}\}$ (d.h. $\bar{u}$ ist die einzige Nullstelle von F in dieser Kugel).

2. Für jeden Anfangswert $y^{(0)} \in K_\varrho(\bar{u})$ existiert die Newton-Folge $y^{(k)}$. Es gilt $y^{(k)} \in K_\varrho(\bar{u})$ und

$$\lim_{k\to\infty} y^{(k)} \;=\; \bar{u} \qquad \text{(lokale Kovergenz)}.$$

3. Weiter gibt es eine (von $\bar{u}$ und ϱ abhängige) Konstante $c > 0$, so dass für alle $k \in \mathbb{N}$ gilt

$$\left\|\bar{u} - y^{(k+1)}\right\|_2 \;\leq\; c{\cdot}\left\|\bar{u} - y^{(k)}\right\|_2^2 \qquad \text{(Konvergenzordnung 2)}.$$

Beweis: vgl. Kress [13] oder Werner [11].

10 Minimierung

10.1 Einleitung

Gesucht ist ein (lokales) Minimum einer Funktion $h : D \subset \mathbb{R}^n \to \mathbb{R}$, d.h. ein Vektor $\overline{z} = (\overline{z}_1, \ldots, \overline{z}_n)$ mit

$$h(\overline{z}) \leq h(z) \quad \text{für alle } z \in D \quad \text{(globales Minimum) bzw.}$$

$$h(\overline{z}) \leq h(z) \quad \text{für alle } z \in U,\ U \text{ Umgebung von } \overline{z} \quad \text{(lokales Minimum).}$$

Wir geben uns mit dem Auffinden der lokalen Minima zufrieden und setzen die Existenz der 1. und 2. partiellen Ableitungen von h voraus ($h \in C^2(\mathbb{R}^n, \mathbb{R})$). Als lokale Minima kommen dann nur Vektoren $\overline{z} = (\overline{z}_1, \ldots, \overline{z}_n)$ in Frage, welche

$$\frac{\partial}{\partial z_i} h(\overline{z}_1, \ldots, \overline{z}_n) = 0 \quad (i = 1, \ldots, n)$$

oder in anderer Schreibweise

$$\operatorname{grad} h(\overline{z}) = (h_{z_1}(\overline{z}), \ldots, h_{z_n}(\overline{z})) = (0, \ldots, 0) = 0 \tag{116}$$

erfüllen.

Numerische Verfahren bestimmen deshalb in der Regel zuerst einen Vektor $\overline{z}$, für den (116) gilt (und zwar näherungsweise im Rahmen einer vorgegebenen Genauigkeit) und testen dann, ob dieser Punkt $\overline{z}$ ein lokales Minimum ist (zum Beispiel mit Hilfe der Hesse-Matrix von h).

10.2 Nichtlinearer Ausgleich

In den Anwendungen ist h häufig eine Fehlerquadratsumme:

$$h(z) = h(z_1, \ldots, z_n) = \sum_{j=1}^{m} (G(t_j, z_1, \ldots, z_n) - s_j)^2$$

mit einer Theoriefunktion $G(t, z) = G(t, z_1, \ldots, z_n)$. Dabei sind die Paare (t_j, s_j), $j = 1, \ldots, m$, Messergebnisse eines Experimentes (vgl. Abschnitt 2.1).

Falls in G die Parameter $z_1, \ldots, z_n$ linear eingehen, so ergibt sich ein lineares Ausgleichsproblem, welches bereits in Paragraph 2 behandelt wurde. Hier interessieren wir uns für die nichtlineare Situation. Einige Beispiele wurden schon in Abschnitt 2.1 angegeben. Trotzdem betrachten wir hier zunächst noch eine weitere Anwendung.

Beispiel aus der Biologie: Enzym-Aktionen

Viele Stoffwechselvorgänge in Lebewesen erfolgen mit Hilfe von Enzymen, welche aus der Reaktion unverändert hervorgehen. Zum Beispiel:

$$\text{Saccharose} \xrightarrow[\text{Invertase}]{} \text{D-Glucose} + \text{D-Fructose}$$

In allgemeiner Form haben wir die Situation

$$\text{Substrat} \xrightarrow[\text{Enzym}]{} \text{Produkt}$$

oder kurz

$$X \xrightarrow[E]{} P$$

(wobei X das Substrat, P das Produkt und E das Enzym bezeichnet). Bei einer solchen Enzym-Aktion werden zunächst Substrat-Moleküle an das Enzym gebunden; es entsteht ein sogenannter Enzym-Substrat-Komplex (den wir mit ES bezeichnen). In einer zweiten Stufe entsteht aus diesem Enzym-Substrat-Komplex das Produkt und wieder das Enzym:

$$X + E \rightleftarrows ES \longrightarrow P + E \quad .$$

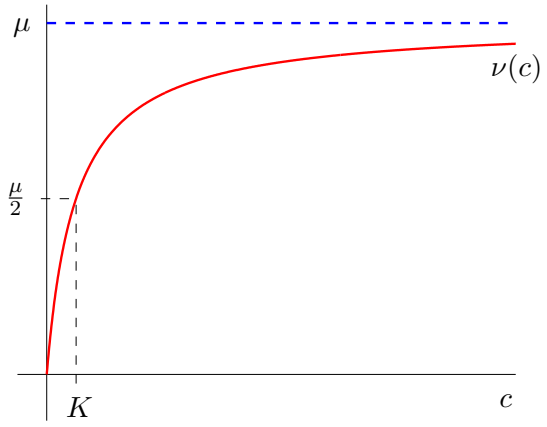
Die erste Reaktion ist umkehrbar, die zweite nicht.

Aus biologischer Sicht ist es nun von Bedeutung, die Geschwindigkeit des Substratabbaus in Abhängigkeit von der Menge des angebotenen Substrates zu kennen. Bezeichnet man mit c die Konzentration von X und mit ν die Reaktionsgeschwindigkeit, so ist eine Funktion $\nu(c)$ gesucht.

Nach Michaelis-Menten (1913) hat diese Funktion die Form

$$\nu(c) = \frac{\mu \cdot c}{K + c} \qquad \text{mit } K > 0 \text{ und } \mu > 0.$$

Dabei sind die Konstanten K und μ zunächst unbekannt. Graphisch hat diese Funktion den folgenden Verlauf:



An dieser Stelle sei noch erwähnt, dass nicht jede Enzymreaktion einen Verlauf wie oben zeigt. Solche Enzyme werden hier nicht betrachtet.

Im biologischen Experiment wird zu gewissen Konzentrationen c_i die Reaktionsgeschwindigkeit ν_i gemessen $(i = 1, \ldots, m)$. Aufgabe ist es nun, anhand dieser Messergebnisse die Parameter K und μ so zu bestimmen, dass

$$\nu_i \approx \nu(c_i) \qquad (i = 1, \ldots, m)$$

möglichst gut erfüllt ist. Dies wird mathematisch präzisiert zu: Bestimme das Minimum der Funktion

$$h(K, \mu) := \sum_{i=1}^{m} \left(\frac{\mu \cdot c_i}{K + c_i} - \nu_i \right)^2$$

(vergleiche Abschnitt 2.1). Die Theoriefunktion ist hier

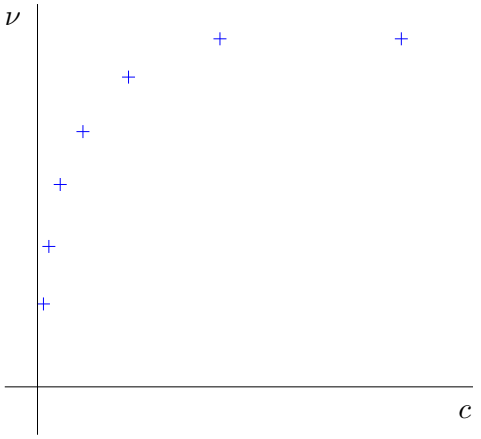
$$g(c;\, \mu,\, K) = \frac{\mu \cdot c}{K + c} \quad .$$

Dieser Ansatz stellt ein nichtlineares Ausgleichsproblem dar. Biologisch sinnvoll ist dabei nur $K > 0$ und $\mu > 0$, was eine Einschränkung des Definitionsbereichs von h bedeutet.

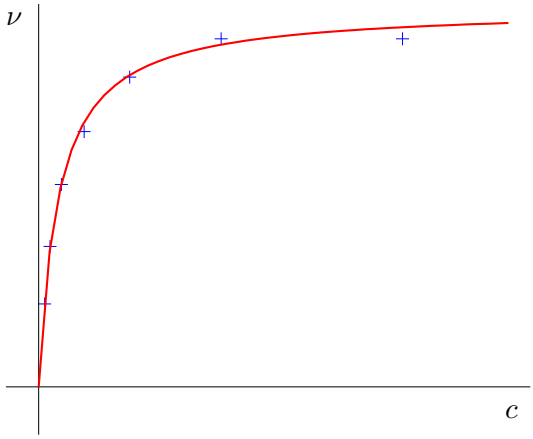
Für die oben erwähnte Enzym-Aktion haben Michaelis und Menten im Experiment folgende Werte erhalten:

c	0.3330	0.1670	0.0833	0.0416	0.0208	0.0104	0.0054
ν	3.6360	3.6360	3.2360	2.6660	2.1140	1.4660	0.8660

Messergebnisse



optimaler Fit für $\mu = 3.95,\ K = 0.018$



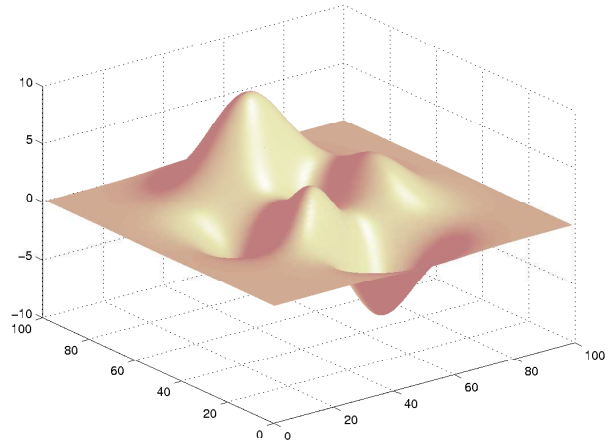
Bei den gängigen Algorithmen zur nichtlinearen Minimierung spielt die Wahl eines geeigneten Startwertes eine zentrale Rolle (denn in der Regel liegt nur lokale Konvergenz vor). Bei Fitproblemen mit zwei Parametern lassen sich eventuell aus einer *Höhenkarte* geeignete Startwerte ablesen.

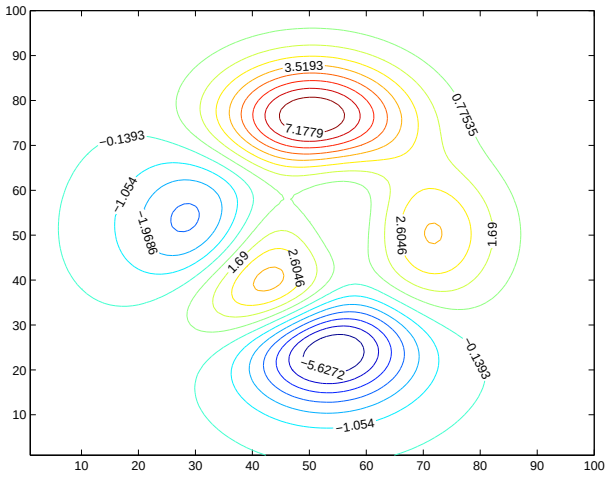
10.3 Höhenlinien

Im Fall $n = 2$ erhält man Hinweise über die ungefähre Lage der Extremalstellen der Funktion $h(z_1, z_2)$ durch Zeichnen der Höhenlinien von h. Sei $c \in \mathbb{R}$, so ist

$$H_c = \{(z_1, z_2) \in D \,:\, h(z_1, z_2) = c\}$$

die *Höhenlinie zum Niveau c*. Dazu betrachten wir das folgende Beispiel:





Praktische Bestimmung der Höhenlinien

Höhenlinien von h bestimmt man durch numerisches Lösen der Anfangswertaufgabe

$$\begin{aligned} \dot{z_1} &= h_{z_2}(z_1, z_2)\,, \quad z_1(0) = \alpha\,, \\ \dot{z_2} &= -h_{z_1}(z_1, z_2)\,, \quad z_2(0) = \beta\,, \end{aligned}$$

dabei nehmen wir an, dass die partiellen Ableitungen von h existieren und stetig sind.

Lemma: Die Lösung $(z_1(t), z_2(t))$ dieser Anfangswertaufgabe liefert (teilweise) die Höhenlinie H_c mit $c := h(\alpha, \beta)$; denn es gilt

$$(z_1(t), z_2(t)) \;\in\; H_c$$

für alle t aus dem Existenzintervall I der Lösung.

Beweis: Die Funktion

$$\varphi \;:\; I \to \mathbb{R}\,,\; \varphi(t) := h(z_1(t), z_2(t))$$

ist differenzierbar, und es gilt

$$\begin{aligned}\dot{\varphi} &= h_{z_1}(z_1(t), z_2(t))\cdot\dot{z}_1(t) + h_{z_2}(z_1(t), z_2(t))\cdot\dot{z}_2(t) \\ &= h_{z_1}(z_1(t), z_2(t))\cdot h_{z_2}(z_1(t), z_2(t)) - h_{z_2}(z_1(t), z_2(t))\cdot h_{z_1}(z_1(t), z_2(t)) \\ &= 0 \quad .\end{aligned}$$

Damit ist die Funktion φ konstant, also

$$h(z_1(t), z_2(t)) = \varphi(t) = \varphi(0) = h(z_1(0), z_2(0)) = h(\alpha, \beta) \quad \text{für alle } t \in I. \ \square$$

10.4 Abstiegsverfahren

Abstiegsrichtung

Gegeben sei ein Vektor $z \in \mathbb{R}^n$. Wir fragen uns, ob es einen neuen Vektor $z^* \in \mathbb{R}^n$ gibt mit

$$h(z^*) < h(z) \quad .$$

Anschaulich bedeutet dies, dass wir uns in einem Gebirge an einer bestimmten Stelle z befinden und ins Tal wollen. Wegen Nebels kann man jedoch nur die unmittelbare Umgebung erfassen. Eine naheliegende Methode besteht darin, zunächst einmal zu prüfen, ob es eine von dem aktuellen Standpunkt z ausgehende Richtung gibt, in der es wenigstens ein kleines Stück abwärts geht.

Wir fragen uns also: gibt es eine Richtung $p = (p_1, \ldots, p_n)^T \in \mathbb{R}^n$ und ein $t_0 > 0$, so dass

$$h(z + t\cdot p) < h(z) \quad \text{für alle } t \in (0, t_0]$$

gilt. Ein solches p nennt man eine *Abstiegsrichtung* von h im Punkt z.

Beachten Sie, dass z und p Vektoren der Länge n sind und dass t eine reelle Zahl ist:

$$z + t\cdot p = \begin{pmatrix} z_1 + t\cdot p_1 \\ \vdots \\ z_n + t\cdot p_n \end{pmatrix} \quad .$$

Zu festem $z, p \in \mathbb{R}^n$ betrachten wir die reelle Funktion

$$\varphi(t) := h(z + t\cdot p) \quad ,$$

welche auf einem gewissen Intervall $I \supset [0, t_0]$ definiert ist und

$$\varphi(0) = h(z)$$

erüllt. Ferner ist φ differenzierbar (da $h \in C^2(\mathbb{R}^n, \mathbb{R})$ vorausgesetzt wird), und es gilt

$$\dot{\varphi}(t) = \operatorname{grad} h(z + t\cdot p)\cdot p = h_{z_1}(z + t\cdot p)\cdot p_1 + \ldots + h_{z_n}(z + t\cdot p)\cdot p_n$$

(dabei fassen wir $\operatorname{grad} h(z + t\cdot p)$ als Zeilenvektor und p als Spaltenvektor auf).

Insbesondere erhält man

$$\dot{\varphi}(0) \;=\; \operatorname{grad} h(z) \cdot p \quad .$$

Gilt nun

$$\dot{\varphi}(0) \;=\; \operatorname{grad} h(z) \cdot p \;<\; 0 \quad , \tag{117}$$

so dürfen wir aufgrund der Stetigkeit von $\dot{\varphi}$ daraus $\dot{\varphi}(t) \;<\; 0$ für alle t aus einer geeigneten Umgebung von 0 schließen, d.h. es gibt ein $t_1 > 0$ mit

$$\dot{\varphi}(t) \;<\; 0 \quad \text{für } 0 \le t \le t_1 \quad .$$

Dann ist φ streng monoton fallend in diesem Bereich, also

$$h(z + t{\cdot}p) \;=\; \varphi(t) \;<\; \varphi(0) \;=\; h(z) \quad \text{für } 0 < t \le t_1 \quad .$$

Fazit:

1. Wählt man den Vektor $p \in \mathbb{R}^n$ so, dass (117) gilt, dann ist p eine Abstiegsrichtung von h im Punkt z.
2. Ist $\operatorname{grad} h(z) \neq 0$, so gibt es immer eine Abstiegsrichtung, nämlich $p = -\operatorname{grad} h(z)$.

Basisalgorithmus

Der vorige Abschnitt motiviert folgendes Verfahren zur Berechnung eines Punktes $\overline{z}$ mit $\operatorname{grad} h(\overline{z}) = 0$:

1. Sei ein $z^{(0)} \in \mathbb{R}^n$ gegeben. Setze $k = 0$.
2. Berechne $\operatorname{grad} h(z^{(k)})$. Gilt $\operatorname{grad} h(z^{(k)}) \;=\; 0$, so beende das Verfahren.
   Andernfalls bestimmt man eine Abstiegsrichtung $p^{(k)} \in \mathbb{R}^n$ und ein $t^{(k)} > 0$ mit

$$h(z^{(k)} + t^{(k)}{\cdot}p^{(k)}) \;<\; h(z^{(k)}) \quad . \tag{118}$$

   Setze dann $z^{(k+1)} \;:=\; z^{(k)} \;+\; t^{(k)}{\cdot}p^{(k)}$, $k := k+1$ und wiederhole Schritt 2.

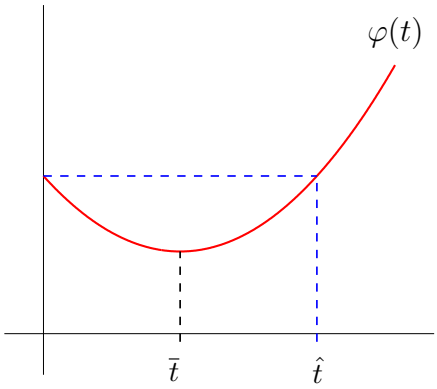
Eine solche Iterationsvorschrift wird als *Abstiegsverfahren* bezeichnet. Es besteht aus einer *Richtungs-* und einer *Schrittweitenstrategie*: für jedes $k = 0, 1, 2, \ldots$ ist eine Abstiegsrichtung $p^{(k)}$ und eine Schrittweite $t^{(k)}$ so zu wählen, dass (118) erfüllt ist.

Schrittweitenstrategie

Seien $z^{(k)} \in \mathbb{R}^n$ und eine Abstiegsrichtung $p^{(k)}$ von h im Punkte $z^{(k)}$ schon bestimmt. Dann betrachten wir wieder die Funktion

$$\varphi(t) \;:=\; h\left(z^{(k)} + t{\cdot}p^{(k)}\right) \quad .$$

Unter gewissen Voraussetzungen hat φ qualitativ folgendes Aussehen:



Hieraus erkennt man

$$h\left(z^{(k)}+t\cdot p^{(k)}\right) \;=\; \varphi(t) \;<\; \varphi(0) \;=\; h\left(z^{(k)}\right) \quad \text{für alle } t\in(0,\hat{t})$$

und

$$\varphi(\bar{t}) \;\leq\; \varphi(t) \quad \text{für alle } t\in(0,\bar{t}) \quad .$$

Dieses $\bar{t}$ ist das erste relative Minimum von φ, also die kleinste positive Nullstelle von $\dot{\varphi}$. In unserem Basisalgorithmus können wir dann $t^{(k)} := \bar{t}$ wählen, welches als *exakte Schrittweite* bezeichnet wird. $t^{(k)}$ ist die Lösung einer skalaren Minimierungsaufgabe. Beachten Sie, dass sich für jedes k ein anderes $t^{(k)}$ ergibt, da φ jeweils von $z^{(k)}$ und $p^{(k)}$ abhängt.

Richtungsstrategien

Ist der Punkt $z^{(k)}$ schon bestimmt und gilt $\operatorname{grad} h(z^{(k)}) \neq 0$, so kann man als Abstiegsrichtung $p^{(k)}$ den negativen Gradienten wählen:

$$p^{(k)} \;:=\; -\operatorname{grad} h\left(z^{(k)}\right)^T \;=\; \left(-h_{z_1}\left(z^{(k)}\right),\ldots,-h_{z_n}\left(z^{(k)}\right)\right)^T \quad .$$

Die so erzeugte Iterationsvorschrift heißt *Gradientenverfahren.*

Konvergenz

Unter gewissen Voraussetzungen (die in der Praxis allerdings nur schwer zu verifizieren sind) erhält man Konvergenz gegen ein lokales Minimum. Wir geben hier nur einen Satz für das Gradientenverfahren an.

Satz: Seien $h\in C^2(D,\mathbb{R})$, $D\subset\mathbb{R}^n$ offen und $z^{(0)}\in D$ der Startwert des Gradientenverfahrens. Die Niveaumenge

$$L_0 \;:=\; \{x\in D \;:\; h(x)\leq h(z^{(0)})\}$$

sei kompakt und besitze genau ein $\overline{z}\in L_0$ mit $\operatorname{grad} h(\overline{z})=0$.
Dann konvergiert das Gradientenverfahren (mit Startwert $z^{(0)}$) gegen $\overline{z}$.

Beweis: vgl. Werner [12].

Wird im Basisalgorithmus eine andere Abstiegsrichtung (als der negative Gradient) verwendet, so sind für die Konvergenz weitere Voraussetzungen erforderlich.

10.5 Differentialgleichungsmethoden

Stationäre Punkte einer Differentialgleichung

Gegeben sei die Differentialgleichung

$$\dot{z} \;=\; F(z)$$

mit $F \in C(\mathbb{R}^n, \mathbb{R}^n)$. Die Nullstellen von F liefern konstante Lösungen der Differentialgleichung: Sei $\bar{z} \in \mathbb{R}^n$ eine Nullstelle von F, so ist

$$z(t) \;=\; \bar{z} \quad \text{für alle } t \in \mathbb{R}$$

eine Lösung der obigen Differentialgleichung, denn

$$\dot{z}(t) \;=\; 0 \;=\; F(\bar{z}) \;=\; F(z(t)) \quad .$$

Man bezeichnet deshalb $\bar{z}$ als *stationären Punkt*. Unter gewissen Voraussetzungen ist ein stationärer Punkt *asymptotisch stabil*. Dies bedeutet, dass alle in einer (kleinen) Umgebung um $\bar{z}$ gestarteten Lösungen $z(t)$ der Differentialgleichung gegen den stationären Punkt konvergieren (für $t \to \infty$).
Diese Eigenschaft können wir für die Numerik ausnutzen.

Nullstellen des Gradienten als stationäre Punkte einer Dgl

In der Einleitung zu diesem Paragraphen haben wir erwähnt, dass als lokale Minima der Funktion h nur solche Vektoren $\overline{z} \in \mathbb{R}^n$ in Frage kommen, welche

$$\operatorname{grad} h(\overline{z}) \;=\; 0$$

erfüllen. Betrachtet man nun die Differentialgleichung

$$\dot{z} \;=\; -\operatorname{grad} h(z) \tag{119}$$

oder ausführlich hingeschrieben

$$\begin{aligned} \dot{z}_1 &= -h_{z_1}(z_1, \ldots, z_n) \\ &\vdots \qquad\qquad\qquad\qquad , \\ \dot{z}_n &= -h_{z_n}(z_1, \ldots, z_n) \end{aligned}$$

so wissen wir, dass die Nullstellen von $g(z) \;:=\; -\operatorname{grad} h(z)$ stationäre Punkte der Differentialgleichung (119) sind. Das bedeutet, dass die stationären Punkte dieser Dgl. als lokale Minima von h in Betracht kommen.

Sei nun ein Vektor $z^{(0)} \in \mathbb{R}^n$ gegeben. Mit $\Psi(t) \;=\; (\Psi_1(t), \ldots, \Psi_n(t))$ bezeichnen wir die Lösung der Anfangswertaufgabe

$$\dot{z} \;=\; -\operatorname{grad} h(z)\,, \quad z(0) \;=\; z^{(0)} \quad . \tag{120}$$

Für die Funktion

$$\varphi(t) \;:=\; h(\Psi(t))$$

(beachte $\Psi(t)$ ist eine Kurve im $\mathbb{R}^n$) errechnet man

$$\begin{aligned}\dot{\varphi}(t) \;=\; \frac{d}{dt}\,\varphi(t) \;&=\; h_{z_1}(\Psi(t))\cdot\frac{d}{dt}\Psi_1(t) \,+\ldots+\, h_{z_n}(\Psi(t))\cdot\frac{d}{dt}\Psi_n(t)\\ &=\; h_{z_1}(\Psi(t))\cdot\dot{\Psi}_1(t) \,+\ldots+\, h_{z_n}(\Psi(t))\cdot\dot{\Psi}_n(t)\\ &=\; -h_{z_1}(\Psi(t))\cdot h_{z_1}(\Psi(t)) \,-\ldots-\, h_{z_n}(\Psi(t))\cdot h_{z_n}(\Psi(t))\\ &\leq\; 0 \quad \text{für alle } t\,.\end{aligned}$$

Daraus folgt, dass $\varphi(t)$ monoton fallend ist, also

$$h(\Psi(t_1)) \;=\; \varphi(t_1) \;\leq\; \varphi(t_2) \;=\; h(\Psi(t_2)) \quad \text{für alle } t_1 > t_2\,.$$

Unter gewissen Voraussetzungen (z.B. $\overline{z}$ ein stabiler stationärer Punkt und $z^{(0)}$ hinreichend nahe bei $\overline{z}$) gilt

$$\lim_{t\to\infty}\; \Psi(t) \;=\; \overline{z}$$

(d.h. $\lim\limits_{t\to\infty}\; \Psi_i(t) \;=\; \overline{z}_i$ für jede Komponente).

Numerisches Verfahren

Für uns ergibt sich daraus das folgende numerische Verfahren:

1. Wähle einen Vektor $z^{(0)} \in \mathbb{R}^n$ (möglichst in der Nähe eines lokalen Minimums) und eine feste Schrittweite $\sigma > 0$. Setze $k = 1$.
2. Berechne mit einem numerischen Verfahren die Lösung der Anfangswertaufgabe (120) zum Zeitpunkt $k\cdot\sigma$:

   $$z^{(k)} \;\approx\; \Psi(k\cdot\sigma)$$

   oder in Komponentenschreibweise

   $$\begin{aligned} z_1^{(k)} \;&\approx\; \Psi_1(k\cdot\sigma)\\ &\;\;\vdots\\ z_n^{(k)} \;&\approx\; \Psi_n(k\cdot\sigma)\end{aligned}$$

   sowie

   $$\operatorname{grad} h\left(z^{(k)}\right) \;=\; \left(h_{z_1}\left(z^{(k)}\right),\ldots,h_{z_n}\left(z^{(k)}\right)\right) \quad .$$

   Gilt dann mit vorgegebener Genauigkeitschranke $\varepsilon > 0$

   $$\left|h_{z_i}\left(z^{(k)}\right)\right| \;<\; \varepsilon \quad \text{für alle } i = 1,\ldots,n\,,$$

   so beenden wir das Verfahren und akzeptieren $z^{(k)}$ als Näherungswert für $\overline{z}$.
3. Andernfalls setzt man $k = k + 1$. Ist $k \leq \overline{K}$ (wobei $\overline{K}$ vorgegeben ist und die maximale Anzahl der Iterationsschritte bedeutet), so wiederholen wir den Schritt 2.
4. Ist $k > \overline{K}$, so wird das Verfahren (ohne brauchbares Ergebnis) abgebrochen. Dann liegt die Vermutung nahe, dass der von uns gewählte Startvektor $z^{(0)}$ ungünstig war. Deshalb starten wir das Verfahren erneut mit einem anderen $z^{(0)}$.

Aus Stabilitätsgründen empfiehlt sich hier der Einsatz von impliziten Verfahren zur Lösung der AWA (120). Matlab bietet dafür geeignete Unterprogramme an (vergleiche Abschnitt 6.7).

11 Stabilitäts- und Störungsfragen

11.1 Einleitung

Stabilität

Gegeben sei eine Funktion $F : \mathbb{R}^n \to \mathbb{R}^n$, von der eine Nullstelle $\bar{x}$ gesucht ist (d.h. $F(\bar{x}) = 0$). Bekannt sei ein Wert y mit

$$F(y) \approx 0 \quad .$$

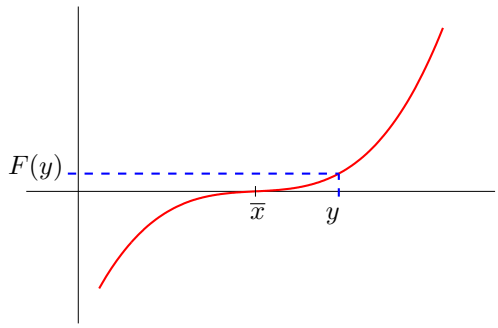
Wir untersuchen, unter welchen Bedingungen dann y eine "gute Näherung" für die gesuchte Nullstelle $\bar{x}$ ist. Diese Fragestellung nennt man das *Stabilitätsproblem.*

Dies ist etwa von Bedeutung für Abbruchkriterien bei Iterationsverfahren, denn dort hat man im Allgemeinen nur den Defekt $F(y^{(k)})$ und die Veränderung $y^{(k)} - y^{(k-1)}$ als Gütewert zur Verfügung.

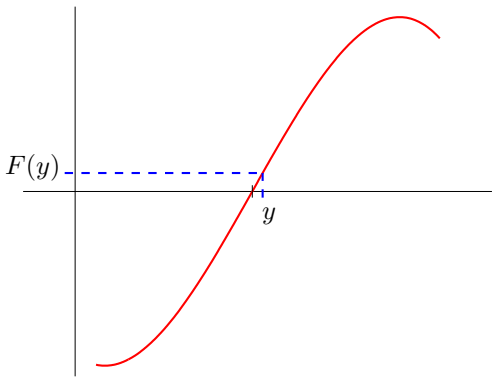
Man spricht von einer *guten Stabilität* der Nullstelle $\bar{x}$, falls wir bei relativ großem Defekt $F(y)$ auf eine gute Näherung y schließen können. Demgegenüber ist bei einer *schlechten Stabilität* ein sehr kleiner Defekt $F(y)$ zu erreichen, um eine gute Näherung zu garantieren.

Im skalaren Fall gibt es eine graphische Veranschaulichung:

schlechte Stabilität



gute Stabilität



Störempfindlichkeit

Eine andere Situation liegt vor, falls die Funktion F nicht genau bekannt ist. Dann ist man darauf angewiesen, die Nullstelle einer gestörten Funktion $G(x)$ zu ermitteln.

So können bei der Berechnung der Koeffizienten eines Polynomes $p \in \Pi_n$ (z.B. beim charakteristischen Polynom) Rundungsfehler entstehen. Statt der Nullstelle $\bar{x}$ von p wird eine Nullstelle y eines gestörten Polynoms $\bar{p}$ berechnet.

Im Gegensatz zum Stabilitätsproblem nehmen wir beim Störungsproblem an, dass die Nullstelle y von G exakt bekannt ist und interessieren uns dann für die Differenz $\bar{x} - y$. Die zentrale Frage des Störungsproblems lautet: Unter welchen Voraussetzungen ist bei einer kleinen Störung der Wert y eine gute Approximation an die gesuchte Größe $\bar{x}$?

In der Praxis treten Stabilitäts- und Störungsprobleme häufig gleichzeitig auf.

Die Störung modellieren wird durch einen q-dimensionalen Parametervektor und betrachten eine Familie von Funktionen

$$G \,:\, \mathbb{R}^{n+q} \to \mathbb{R}^n \,:\, (x,\mu) \to G(x,\mu) \quad .$$

mit $G(x,\bar{\mu}) = F(x)$ für ein gewisses $\bar{\mu}$. So setzt man zum Beispiel für ein in den Koeffizienten gestörtes Polynom

$$G(x,\mu_0,\ldots,\mu_n) \;=\; p(x) \;+\; \sum_{i=0}^{n} \mu_i x^i \quad .$$

Gesucht ist ein $\bar{x}$ mit

$$0 \;=\; F(\bar{x}) \;=\; G(\bar{x},\bar{\mu}) \qquad \text{(Ausgangsproblem)}.$$

Wir kennen dagegen ein (y,η) mit $G(y,\eta) = 0$ (Lösung des gestörten Problems).

11.2 Eine allgemeine Fehlerdarstellung

Gegeben sei eine stetig differenzierbare Funktion

$$G \,:\, \mathbb{R}^{n+q} \to \mathbb{R}^n \,:\, (u,v) \to G(u,v)$$

oder in Komponentendarstellung

$$G(u,v) \;=\; \begin{pmatrix} g_1(u_1,\ldots,u_n,v_1,\ldots,v_q) \\ \vdots \\ g_n(u_1,\ldots,u_n,v_1,\ldots,v_q) \end{pmatrix} \quad .$$

Für jedes $j = 1,\ldots,n$ gilt folgender "Mittelwertsatz":

$$\begin{aligned} g_j(x,\mu) - g_j(y,\eta) \;&=\; \sum_{i=1}^{n} \frac{\partial}{\partial u_i} g_j\left(y + t_j(x-y), \eta + t_j(\mu-\eta)\right)\cdot(x_i - y_i) \\ &\quad + \sum_{i=1}^{q} \frac{\partial}{\partial v_i} g_j\left(y + t_j(x-y), \eta + t_j(\mu-\eta)\right)\cdot(\mu_i - \eta_i) \\ &=\; D_u g_j\left(y + t_j(x-y), \eta + t_j(\mu-\eta)\right)\cdot(x-y) \\ &\quad + D_v g_j\left(y + t_j(x-y), \eta + t_j(\mu-\eta)\right)\cdot(\mu-\eta) \end{aligned}$$

mit einem Zwischenwert $t_j \in (0,1)$, genauer $t_j = t_j(x,y,\mu,\eta)$.

Dieses Ergebnis der Analysis ergibt sich, wenn man auf die univariate Funktion

$$\varphi(s) \;=\; g_j(y + s(x-y), \eta + s(\mu-\eta))$$

den Mittelwertsatz anwendet und φ nach der Kettenregel differenziert:

$$\begin{aligned} g_j(x,\mu) \;-\; g_j(y,\eta) \;&=\; \varphi(1) \;-\; \varphi(0) \;=\; \varphi'(t_j) \\ &=\; Dg_j(y + t_j(x-y), \eta + t_j(\mu-\eta))\cdot \begin{pmatrix} x-y \\ \mu-\eta \end{pmatrix} \end{aligned}$$

($Dg_j = (D_u g_j, D_v g_j)$ ist der Gradient von g_j).

In kompakter Form schreiben wir

$$\begin{aligned} G(x,\mu) - G(y,\eta) &= (D_u G, D_v G) \begin{pmatrix} x-y \\ \mu-\eta \end{pmatrix} \\ &= D_u G \cdot (x-y) + D_v G \cdot (\mu - \eta) \quad . \end{aligned} \tag{121}$$

Dabei ist $D_u G$ die $n \times n$ - Matrix mit den Elementen

$$\frac{\partial}{\partial u_i} g_j(y + t_j(x-y), \eta + t_j(\mu - \eta)) \qquad (i,j = 1,\ldots,n)$$

und $D_v G$ die $n \times q$ - Matrix mit den Elementen

$$\frac{\partial}{\partial v_i} g_j(y + t_j(x-y), \eta + t_j(\mu - \eta)) \qquad (j = 1,\ldots,n \text{ und } i = 1,\ldots,p).$$

Diese Darstellung gilt auch für $q = 0$ und lautet dann

$$F(x) - F(y) = DF \cdot (x-y) \quad . \tag{122}$$

In (121) sind in der Funktionalmatrix $(D_u G, D_v G)$ verschiedene Zwischenpunkte zugelassen. Im Folgenden werten wir die Funktionalmatrix an der festen Stelle $(\bar{x}, \bar{\mu})$ aus; in (121) gilt dann nur approximativ die Gleichheit.

Stabilitätsproblem

Hier hängt G von keinen Parametern ab (d.h. $q = 0$ und somit $G(x) = F(x)$). Sei $\bar{x}$ eine Nullstelle von F und y ein Näherungswert. Dann gilt

$$-F(y) = F(\bar{x}) - F(y) \approx DF(\bar{x}) \cdot (\bar{x} - y)$$

und falls $DF(\bar{x})$ invertierbar ist

$$\bar{x} - y \approx -(DF(\bar{x}))^{-1} \cdot F(y) \quad . \tag{123}$$

Fazit: Aus (123) erkennt man, dass ein kleiner Defekt für eine gute Stabilität noch nicht ausreicht. Vielmehr ist auch noch eine kleine Norm der Inversen der Funktionalmatrix erforderlich.

Im skalaren Fall ist $(DF(\bar{x}))^{-1} = \frac{1}{F'(\bar{x})}$, und diese Überlegungen stimmen mit der graphischen Veranschaulichung überein.

Störungsproblem

Sei $G(\bar{x}, \bar{\mu}) = 0$. Berechnet wurde die Lösung $G(y,\eta) = 0$ eines gestörten Problems. Aus (121) erhält man dann die approximative Beziehung

$$0 \approx D_u G(\bar{x}, \bar{\mu}) \cdot (\bar{x} - y) + D_v G(\bar{x}, \bar{\mu}) \cdot (\bar{\mu} - \eta) \quad .$$

Ist $D_u G(\bar{x}, \bar{\mu})$ invertierbar, so ergibt sich

$$\bar{x} - y \approx -(D_u G(\bar{x}, \bar{\mu}))^{-1} \cdot D_v G(\bar{x}, \bar{\mu}) \cdot (\bar{\mu} - \eta) \quad . \tag{124}$$

Die Güte der Näherung y wird in Abhängigkeit der Differenz der Parametervektoren gemessen. Das auftretende Produkt der Funktionalmatrizen bewirkt eine Verstärkung oder eine Dämpfung.

Fazit:

1. Für die Störempfindlichkeit ist die Norm der Matrix

$$(D_u G(\bar{x}, \bar{\mu}))^{-1} \cdot D_v G(\bar{x}, \bar{\mu})$$

   verantwortlich.

2. Eine kleine Norm bedeutet nach (124), dass bei einer kleinen Störung $\bar{\mu} - \eta$ auch ein kleiner Fehler $\bar{x} - y$ garantiert ist. Es liegt eine *kleine Störempfindlichkeit* (oder *Störunempfindlichkeit*) vor.

3. Dagegen bedeutet eine große Norm, dass selbst bei einer sehr kleinen Störung $\bar{\mu} - \eta$ ein sehr großer Fehler $\bar{x} - y$ auftreten kann. Hier spricht man von einer *großen Störempfindlichkeit*.

11.3 Stabilität und Störempfindlichkeit einer Nullstelle

Stabilität

Gegeben sei die skalare Nullstellen-Situation

$$f \in C^1(\mathbb{R}) \,, \qquad f(\bar{x}) \;=\; 0 \,, \qquad f'(\bar{x}) \;\neq\; 0$$

und ein Näherungwert y. Mit der allgemeinen Theorie (vgl. (123)) finden wir

$$y \;-\; \bar{x} \;\approx\; \frac{1}{f'(\bar{x})} f(y) \qquad .$$

Konkret erhalten wir dann folgende Aussage:

$$|f'(\bar{x})| \;\approx\; 10^k \,, \quad |f(y)| \;\approx\; 10^d \quad \Rightarrow \quad |y - \bar{x}| \;\approx 10^{d-k} \qquad .$$

Um also $|y - \bar{x}| \;\approx\; \varepsilon := 10^{-\eta}$ zu erhalten, muss der Defekt $|f(y)| \;\approx\; 10^{k-\eta}$ erfüllen.

Für $k \geq 1$ wird man von einer guten Stabilität reden (trotz eines relativ großen Defekts ist y eine gute Näherung für $\bar{x}$), für $k \leq -1$ dagegen von einer schlechten Stabilität.

Anmerkungen:

1. Betrachtet man die exakte Fehlerdarstellung (122), so ergibt sich

$$y \;-\; \bar{x} \;=\; \frac{1}{f'(\xi)} f(y) \qquad \text{mit einem } \xi \in (\min\{\bar{x}, y\} \,, \max\{\bar{x}, y\}).$$

   Daraus wird ersichtlich, dass für das Stabilitätsverhalten genau genommen nicht nur $f'(\bar{x})$ entscheidend ist, sondern das Verhalten von f' in einer Umgebung der Nullstelle $\bar{x}$.

2. Hat f mehrere Nullstellen, so kann das Stabilitätsverhalten in den einzelnen Nullstellen sehr unterschiedlich sein.

Störempfindlichkeit einer Nullstelle

Vorgelegt sei ein skalares Störungsproblem mit einem Parameter:

$$G(x,\mu) \;=\; f(x) \;+\; \mu{\cdot}g(x)\;, \qquad f,g \in C^1(\mathbb{R})$$

sowie

$$G(\bar{x},0) \;=\; f(\bar{x}) \;=\; 0\;, \qquad G(y,\eta) \;=\; f(y) \;+\; \eta{\cdot}g(y) \;=\; 0 \quad .$$

Aus der allgemeinen Darstellung (124) erhalten wir für diesen Sonderfall

$$\bar{x} \;-\; y \;\approx\; -\frac{g(\bar{x})}{f'(\bar{x})}(-\eta) \;=\; \frac{g(\bar{x})}{f'(\bar{x})}\,\eta \quad , \tag{125}$$

dabei sei $f'(\bar{x}) \neq 0$ vorausgesetzt.

Definition: Die Größe $\left|\frac{g(\bar{x})}{f'(\bar{x})}\right|$ heißt *Konditionszahl (der Nullstelle $\bar{x}$).*

Daraus erkennt man eine hohe Störempfindlichkeit, falls die Konditionszahl groß ist.

Beispiel: (vgl. Schwarz [3])
Gegeben sei das Polynom vom Grad 12 mit den Nullstellen $\bar{x}_k = k$ $(k = 1,\ldots,12)$:

$$\begin{aligned} f(x) \;=\; \prod_{k=1}^{12}(x-k) \;&=\; x^{12} - 78x^{11} + - \ldots - 6926634x^7 + - \ldots + 4790001600 \\ &=\; \sum_{i=0}^{12} p_i x^i \quad . \end{aligned}$$

Wir betrachten eine Störung im Koeffizienten p_j, d.h.

$$\mu g(x) \;=\; \mu p_j x^j \quad ,$$

und erhalten

$$|f'(\bar{x}_k)| \;=\; |f'(k)| \;=\; (k-1)!(12-k)!$$

$$\left(f'(x) \;=\; \sum_{j=1}^{12}\prod_{i\neq j}(x-i) \qquad \text{somit} \;\; f'(k) \;=\; \prod_{i\neq k}(k-i) \;=\; (-1)^{12-k}(k-1)!(12-k)! \right)$$

sowie $g(k) = p_j k^j$. Damit ergeben sich die Konditionszahlen

$$c_{kj} \;:=\; \left|\frac{g(k)}{f'(k)}\right| \;=\; \frac{|p_j|k^j}{(k-1)!(12-k)!} \quad .$$

In der folgenden Tabelle werden einige c_{kj} angegeben.

	k = 1	k = 3	k = 9	k = 12
j=0	$1.20{\cdot}10^{1}$	$6.60{\cdot}10^{2}$	$1.98{\cdot}10^{3}$	$1.20{\cdot}10^{1}$
j=3	$3.54{\cdot}10^{1}$	$5.26{\cdot}10^{4}$	$4.26{\cdot}10^{6}$	$6.12{\cdot}10^{4}$
j=7	$1.17{\cdot}10^{-1}$	$2.09{\cdot}10^{4}$	$1.37{\cdot}10^{8}$	$6.22{\cdot}10^{6}$
j=11	$1.95{\cdot}10^{-6}$	$1.90{\cdot}10^{1}$	$1.01{\cdot}10^{7}$	$1.45{\cdot}10^{6}$

Für die kleineren Nullstellen (insbesondere für $\bar{x}_1 = 1$) liegt in den Koeffizienten zu den höheren Potenzen eine Störunempfindlichkeit vor.
Dagegen erkennt man für die großen Nullstellen eine sehr hohe Störempfindlichkeit in den Koeffizienten.

Wir wollen nun speziell die Nullstelle $\bar{x}_9$ bei einer Störung von p_7 anschauen und stellen uns einen Rechner mit 10-stelliger Mantisse vor. Der Koeffizient p_7 wird in der letzten Stelle um eine Einheit gestört:

$$p_7^\star \;=\; p_7 + \mu p_7 \;=\; 6926634.001\,, \qquad \text{also } \mu \;=\; \frac{p_7^\star - p_7}{p_7} \;=\; 1.444{\cdot}10^{-10} \quad .$$

Formel (125) liefert für die Nullstelle y_9 des gestörten Problems $y_9 \;\approx\; \bar{x}_9 - \mu c_{97} = 8.980$.
Dieser Wert wird durch die Rechnung mit sehr hoher Genauigkeit bestätigt.

Fazit:

1. Das obige Beispiel zeigt eine hohe Empfindlichkeit der Nullstellen gegenüber Störungen in den Koeffizienten des Polynoms. Diese Empfindlichkeit verstärkt sich in der Regel mit zunehmendem Polynomgrad.
2. Dieser Aspekt ist ein wesentlicher Grund, warum man bei der Eigenwertbestimmung Verfahren verwendet, die die explizite Berechnung der Koeffizienten des charakteristischen Polynoms vermeiden.

11.4 Störempfindlichkeit bei Eigenwertaufgaben

Eigenwerte gestörter Matrizen

Es sei $\|\cdot\|$ eine Vektornorm und $\mathrm{lub}(\cdot)$ die induzierte Matrixnorm:

$$\mathrm{lub}(A) \;:=\; \max\{\|Ax\| \;:\; x \in \mathbb{R}^n \text{ mit } \|x\| = 1\} \quad .$$

In diesem Abschnitt untersuchen wir die Frage, wie weit Eigenwerte benachbarter Matrizen voneinander abweichen können. Der folgende Satz liefert die Grundlage für Eigenwertabschätzungen:

Satz: Seien $A, B \in \mathbb{R}^{n\times n}$. Ist $\lambda \in \mathbb{C}$ ein Eigenwert von A und kein Eigenwert von B, so gilt

$$1 \;\le\; \mathrm{lub}\left((\lambda I - B)^{-1}(A - B)\right) \;\le\; \mathrm{lub}\left((\lambda I - B)^{-1}\right){\cdot}\mathrm{lub}(A - B) \quad . \tag{126}$$

Beweis: Die zweite Ungleichung ergibt sich sofort aus der Submultiplikativität der lub-Norm. Ist v ein Eigenvektor mit $\|v\| = 1$ von A zum Eigenwert λ, so erhält man

$$(A - B)v \;=\; (\lambda I - B)v \quad .$$

$(\lambda I - B)$ ist invertierbar, denn λ ist kein Eigenwert von B. Also folgt

$$(\lambda I - B)^{-1}(A - B)v \;=\; v$$

und damit nach Definition der lub-Norm

$$\mathrm{lub}\left((\lambda I - B)^{-1}(A - B)\right) \;\ge\; 1 \quad . \quad \square$$

Die Ungleichung (126) wird benutzt, um die Eigenwerte einer gestörten Matrix abzuschätzen. Dazu verwenden wir die euklidsche Vektornorm $\| \ \|_2$; $\varrho(A)$ bezeichne wieder den Spektralradius der Matrix A.

Lemma: Ist $A \in \mathbb{R}^{n\times n}$ symmetrisch, so gilt für die zur euklidschen Vektornorm gehörige lub-Norm

$$\mathrm{lub}_2(A) \ = \ \varrho(A) \quad .$$

Für eine Diagonalmatrix A ergibt sich

$$\mathrm{lub}_2(A) \ = \ \max_{1\le i\le n} |a_{ii}| \quad . \tag{127}$$

Beweis: Übung.

Anmerkung: Für eine beliebige Matrix $A \in \mathbb{R}^{n\times n}$ gilt

$$\mathrm{lub}_2(A) \ = \ \sqrt{\varrho(A^T A)}$$

(vgl. z.B. Hämmerlin-Hoffmann [9]).

Definition: Es sei A eine invertierbare Matrix, $\|\cdot\|$ eine Vektornorm und $\mathrm{lub}(\cdot)$ die induzierte Matrixnorm. Dann heißt

$$\mathrm{cond}(A) \ = \ \mathrm{lub}(A)\cdot\mathrm{lub}(A^{-1})$$

die *Konditionszahl* von A. Die Konditionszahl hängt von der gewählten Vektornorm ab.

Satz: Sei $B \in \mathbb{R}^{n\times n}$ diagonalisierbar mit den Eigenwerten $\lambda_1(B), \ldots, \lambda_n(B)$. Es gelte also die Darstellung $B = PDP^{-1}$ mit $D = \mathrm{diag}(\lambda_1(B), \ldots, \lambda_n(B))$ und einer invertierbaren Matrix P. Ist A eine beliebige Matrix, so gibt es zu jedem Eigenwert λ von A einen Eigenwert $\lambda_j(B)$ mit

$$|\lambda - \lambda_j(B)| \ \le \ \mathrm{cond}_2(P)\cdot\mathrm{lub}_2(A-B) \quad .$$

Beweis: Ist λ auch ein Eigenwert von B, so ist die Behauptung trivial. Sei also im Folgenden λ kein Eigenwert von B. Dann erhält man unter Beachtung von (127)

$$\begin{aligned}
\mathrm{lub}_2\left((\lambda I - B)^{-1}\right) &= \mathrm{lub}_2\left((\lambda PP^{-1} - PDP^{-1})^{-1}\right) \\
&= \mathrm{lub}_2\left([P(\lambda I - D)P^{-1}]^{-1}\right) \\
&= \mathrm{lub}_2\left(P(\lambda I - D)^{-1}P^{-1}\right) \\
&\le \mathrm{lub}_2\left((\lambda I - D)^{-1}\right)\cdot\mathrm{lub}_2(P)\cdot\mathrm{lub}_2(P^{-1}) \\
&= \max_{1\le i\le n}\left\{\frac{1}{|\lambda - \lambda_i(B)|}\right\}\mathrm{cond}_2(P) \\
&= \frac{1}{|\lambda - \lambda_j(B)|}\,\mathrm{cond}_2(P) \qquad \text{für ein gewisses } j\,.
\end{aligned}$$

Dies setzen wir in (126) ein und erhalten

$$\begin{aligned}
1 &\le \mathrm{lub}_2\left((\lambda I - B)^{-1}\right)\cdot\mathrm{lub}_2(A-B) \\
&\le \frac{1}{|\lambda - \lambda_j(B)|}\cdot\mathrm{cond}_2(P)\cdot\mathrm{lub}_2(A-B) \quad ,
\end{aligned}$$

woraus die Behauptung folgt. □

Korollar: Ist die Matrix B symmetrisch, so gibt es zu jedem Eigenwert λ einer beliebigen Matrix A einen Eigenwert $\lambda(B)$ von B mit

$$|\lambda - \lambda(B)| \;\leq\; \mathrm{lub}_2(A - B) \quad . \tag{128}$$

Beweis: Bei symmetrischen Matrizen ist P eine orthogonale Matrix. Für eine orthogonale Matrix P gilt $\mathrm{lub}_2(P) = \mathrm{lub}_2(P^{-1}) = \mathrm{lub}_2(P^T) \;=\; 1$ und damit $\mathrm{cond}_2(P) = 1$; denn für jedes $x \in \mathbb{R}^n$ erhält man

$$\|x\|_2^2 \;=\; x^T x \;=\; x^T P^T P x \;=\; (Px)^T Px \;=\; \|Px\|_2^2 \quad .$$

Damit folgt die Behauptung aus dem obigen Satz. □

Fazit:

1. Das obige Korollar besagt, dass sich bei einer kleinen Störung einer symmetrischen Matrix die Eigenwerte auch nur wenig ändern. Das Eigenwertproblem (bei einer symmetrischen Matrix) ist deshalb stör**un**empfindlich. Man sagt auch, das Eigenwertproblem ist *gut konditioniert*.
2. Demgegenüber steht eine hohe Störempfindlichkeit der Koeffizienten des charakteristischen Polynoms (vgl. Abschnitt 11.3). Es ist also nicht ratsam, das charakteristische Polynom durch seine Koeffizienten darzustellen und dann die Nullstellen zu berechnen.

Störempfindlichkeit bei Ähnlichkeitstransformationen

Das QR-Verfahren (vgl. Abschnitt 8.5) basiert auf einer Folge von Ähnlichkeitstransformationen (mit orthogonalen Matrizen). Wir wollen nun untersuchen, wie sich eine Störung bei einer Ähnlichkeitstransformation $B = P^{-1}AP$ auswirkt.

Sei $A \in \mathbb{R}^{n\times n}$ und ΔA eine Störung. Für eine invertierbare Matrix P folgt dann

$$P^{-1}(A + \Delta A)P \;=\; P^{-1}AP \;+\; P^{-1}\cdot\Delta A\cdot P \;=\; B + P^{-1}\cdot\Delta A\cdot P \quad .$$

Die Störung ΔA wird also zu einer Störung $P^{-1}\cdot\Delta A\cdot P$ auf B weitergegeben. Es gilt

$$\begin{aligned} \mathrm{lub}_2(P^{-1}\cdot\Delta A\cdot P) &\leq \mathrm{lub}_2(P^{-1})\cdot\mathrm{lub}_2(P)\cdot\mathrm{lub}_2(\Delta A) \\ &= \mathrm{cond}_2(P)\cdot\mathrm{lub}_2(\Delta A) \quad . \end{aligned}$$

Dies bedeutet, dass sich eine Störung im nächsten Schritt verstärkt, falls die Transformationsmatrix P eine große Konditionszahl besitzt. Bei den Transformationsalgorithmen ist deshalb darauf zu achten, dass diese Konditionszahl klein bleibt.

Verwendet man orthogonale Ähnlichkeitstransformationen $B = Q^T AQ$, so gilt

$$\mathrm{cond}_2(Q) \;=\; 1$$

(vgl. oben). Eine einmal gemachte Störung wird im nächsten Schritt nicht verstärkt. Dies ist ein weiterer Vorteil von orthogonalen Ähnlichkeitstransformationen.

Fazit: Orthogonale Ähnlichkeitstransformationen sind stör**un**empfindlich.

11.5 Störempfindlichkeit linearer Gleichungssysteme

Störung in der rechten Seite

Gegeben sei das lineare Gleichungssystem

$$Ax \;=\; b$$

mit einem invertierbaren $A \in \mathbb{R}^{n\times n}$ und $b \in \mathbb{R}^n$. Es bezeichne $\bar{x} \in \mathbb{R}^n$ die exakte Lösung des linearen Gleichungsystems, also $A\bar{x} \;=\; b$. Wir betrachten zunächst nur eine Störung Δb in der rechten Seite b. Es sei $y \in \mathbb{R}^n$ die Lösung des gestörten Problems:

$$Ay \;=\; b + \Delta b \quad .$$

Mit $y = \bar{x} + \Delta x$ ergibt sich

$$A(\bar{x} + \Delta x) \;=\; b + \Delta b$$

und daraus schließlich

$$\Delta x \;=\; A^{-1}\,\Delta b \quad . \tag{129}$$

Sei $\|\ \|$ eine beliebige Vektornorm und $\mathrm{lub}(\cdot)$ die induzierte Matrixnorm. Dann erhalten wir für den absoluten Fehler

$$\|\bar{x} - y\| \;=\; \|\Delta x\| \;\le\; \mathrm{lub}(A^{-1})\|\Delta b\| \quad .$$

Fazit: Bei einer großen Norm der inversen Matrix liegt eine hohe Empfindlichkeit gegenüber Störungen in der rechten Seite vor (bezogen auf den absoluten Fehler).

Für den relativen Fehler ergibt sich unter Beachtung von

$$\|b\| \;\le\; \mathrm{lub}(A)\|\bar{x}\| \quad \text{und somit} \quad \frac{1}{\|\bar{x}\|} \;\le\; \frac{\mathrm{lub}(A)}{\|b\|}$$

die Ungleichung

$$\frac{\|\Delta x\|}{\|\bar{x}\|} \;\le\; \frac{\mathrm{lub}(A^{-1})\cdot\|\Delta b\|}{\|\bar{x}\|} \;\le\; \mathrm{lub}(A^{-1})\cdot\mathrm{lub}(A)\cdot\frac{\|\Delta b\|}{\|b\|} \;\le\; \mathrm{cond}(A)\cdot\frac{\|\Delta b\|}{\|b\|} \quad .$$

Fazit: Für die Störempfindlichkeit ist die Konditionszahl von A entscheidend.

Anmerkung: Das Ergebnis (129) erhalten wir auch mit dem allgemeinen Konzept aus Abschnitt 11.2. Dazu setzt man

$$G(x,\mu) \;:=\; Ax \;-\; b \;-\; \mu \quad .$$

Dabei ist $\mu = (\mu_1, \ldots, \mu_n)^T$ ein Vektor. Man findet dann

$$D_x G(\bar{x},\bar{\mu}) \;=\; A\,, \qquad D_\mu G(\bar{x},\bar{\mu}) \;=\; -I$$

und damit für $\mu = \Delta b$ und $\overline{\mu} = 0$

$$-\Delta x \;=\; \bar{x} - (\bar{x} + \Delta x) \;=\; -[D_x G(\bar{x},0)]^{-1}\cdot D_\mu G(\bar{x},0)\cdot(0 - \Delta b) \;=\; -A^{-1}\Delta b \quad .$$

Hier steht sogar die Gleichheit, da die Ableitungen $D_x G$ und $D_\mu G$ konstant sind.

Störung in den Koeffizienten der Matrix

Sei $A \in \mathbb{R}^{n\times n}$ invertierbar. Neben der tatsächlichen Lösung $A\bar{x} = b$ betrachten wir das in den Koeffizienten gestörte Problem

$$(A + \Delta A)y \;=\; (A + \Delta A)(\bar{x} + \Delta x) \;=\; b$$

mit $\Delta x = y - \bar{x}$. Ist $(A + \Delta A)$ invertierbar, so erhält man daraus

$$\begin{aligned} \Delta x &= (A + \Delta A)^{-1}(b - (A + \Delta A)\bar{x}) \\ &= -(A + \Delta A)^{-1}\Delta A\,\bar{x} \end{aligned}$$

und somit für eine beliebige Vektornorm und die zugehörige lub-Norm

$$\|\Delta x\| \;\le\; \mathrm{lub}\left((A + \Delta A)^{-1}\right)\cdot \mathrm{lub}(\Delta A)\cdot\|\bar{x}\| \quad . \tag{130}$$

Um nun $\mathrm{lub}\,((A + \Delta A)^{-1})$ abzuschätzen, benötigen wir folgenden

Hilfssatz: Sei $B \in \mathbb{R}^{n\times n}$. Für eine beliebige Vektornorm $\|\ \|$ und eine verträgliche Matrixnorm gelte $N(B) < 1$. Dann existiert $(I + B)^{-1}$, und es gilt

$$\mathrm{lub}\left((I + B)^{-1}\right) \;\le\; \frac{1}{1 - \mathrm{lub}(B)} \quad .$$

Beweis: Sei $x \neq 0$. Dann gilt

$$\begin{aligned} \|(I + B)x\| &= \|x + Bx\| \;\ge\; \|x\| - \|Bx\| \;\ge\; \|x\| - N(B)\|x\| \\ &= (1 - N(B))\|x\| \;>\; 0 \quad . \end{aligned}$$

Somit besitzt das homogene lineare Gleichungssystem $(I + B)x = 0$ nur die triviale Lösung $x = 0$. Damit ist die Matrix $I + B$ regulär. Für die zu $\|\ \|$ gehörige lub-Norm gilt (vgl. Abschnitt 9.2)

$$\mathrm{lub}(B) \;\le\; N(B) \quad \text{und} \quad \mathrm{lub}(I) \;=\; 1 \quad .$$

Weiter erhalten wir

$$\begin{aligned} 1 \;=\; \mathrm{lub}(I) &= \mathrm{lub}\left((I + B)(I + B)^{-1}\right) \\ &= \mathrm{lub}\left((I + B)^{-1} + B(I + B)^{-1}\right) \\ &\ge \mathrm{lub}\left((I + B)^{-1}\right) \;-\; \mathrm{lub}\left(B(I + B)^{-1}\right) \\ &\ge \mathrm{lub}\left((I + B)^{-1}\right) \;-\; \mathrm{lub}(B)\cdot\mathrm{lub}\left((I + B)^{-1}\right) \\ &= \mathrm{lub}\left((I + B)^{-1}\right)\cdot(1 - \mathrm{lub}(B)) \end{aligned}$$

und somit die Behauptung. □

Satz: Es seien A invertierbar und $\mathrm{lub}(A^{-1})\cdot\mathrm{lub}(\Delta A) < 1$. Dann gilt für den relativen Fehler des in den Koeffizienten gestörten linearen Gleichungssystems

$$\frac{\|\Delta x\|}{\|\bar{x}\|} \;\le\; \frac{\mathrm{cond}(A)}{1 - \mathrm{lub}(A^{-1})\mathrm{lub}(\Delta A)}\cdot\frac{\mathrm{lub}(\Delta A)}{\mathrm{lub}(A)} \quad .$$

Beweis: Es gilt

$$\mathrm{lub}(A^{-1}\,\Delta A) \;\le\; \mathrm{lub}(A^{-1})\cdot\mathrm{lub}(\Delta A) \;<\; 1 \quad .$$

Wir setzen im obigen Hilfssatz $B := A^{-1}\,\Delta A$ und erhalten dann

$$\begin{aligned}
\mathrm{lub}\left((A+\Delta A)^{-1}\right) &= \mathrm{lub}\left([A(I+A^{-1}\,\Delta A)]^{-1}\right) \\
&= \mathrm{lub}\left((I+A^{-1}\,\Delta A)^{-1}A^{-1}\right) \\
&\le \mathrm{lub}\left((I+A^{-1}\,\Delta A)^{-1}\right)\cdot\mathrm{lub}(A^{-1}) \\
&\le \frac{1}{1-\mathrm{lub}(A^{-1})\cdot\mathrm{lub}(\Delta A)}\cdot\mathrm{lub}(A^{-1}) \quad .
\end{aligned}$$

Damit folgt aus (130) für den relativen Fehler

$$\begin{aligned}
\frac{\|\Delta x\|}{\|\bar{x}\|} &\le \frac{\mathrm{lub}(A^{-1})\cdot\mathrm{lub}(A)}{1-\mathrm{lub}(A^{-1})\cdot\mathrm{lub}(\Delta A)}\cdot\frac{\mathrm{lub}(\Delta A)}{\mathrm{lub}(A)} \\
&= \frac{\mathrm{cond}(A)}{1-\mathrm{lub}(A^{-1})\cdot\mathrm{lub}(\Delta A)}\cdot\frac{\mathrm{lub}(\Delta A)}{\mathrm{lub}(A)} \quad . \;\square
\end{aligned}$$

Fazit: Für die Störempfindlichkeit ist wieder die Konditionszahl entscheidend. Denn bei einer sehr kleinen Störung ΔA wird $1-\mathrm{lub}(A^{-1})\cdot\mathrm{lub}(\Delta A) \;\approx\; 1$ gelten.

Störung in den Koeffizienten und in der rechten Seite

Wir lassen nun allgemeine Störungen zu und haben dann

$$A\bar{x} \;=\; b \quad \text{und} \quad (A+\Delta A)(\bar{x}+\Delta x) \;=\; b+\Delta b \quad .$$

Aus der 2. Gleichung folgt (für invertierbares $A+\Delta A$)

$$\Delta x \;=\; (A+\Delta A)^{-1}(\Delta b - \Delta A\,\bar{x})$$

und daraus für den relativen Fehler

$$\frac{\|\Delta x\|}{\|\bar{x}\|} \;\le\; \mathrm{lub}\left((A+\Delta A)^{-1}\right)\left(\frac{\|\Delta b\|}{\|\bar{x}\|}+\mathrm{lub}(\Delta A)\right) \quad .$$

Es sei wieder $\mathrm{lub}(A^{-1})\cdot\mathrm{lub}(\Delta A) < 1$. Mit dem Hilfssatz erhält man dann (analog zum Beweis des vorigen Satzes)

$$\frac{\|\Delta x\|}{\|\bar{x}\|} \;\le\; \frac{\mathrm{cond}(A)}{1-\mathrm{lub}(A^{-1})\cdot\mathrm{lub}(\Delta A)}\left(\frac{\|\Delta b\|}{\|b\|}+\frac{\mathrm{lub}(\Delta A)}{\mathrm{lub}(A)}\right) \quad . \tag{131}$$

Wir wollen uns die numerischen Konsequenzen aus der obigen Ungleichung veranschaulichen und unterstellen einen Rechner mit d-stelliger Mantisse. Störungen der Größenordnung

$$\frac{\mathrm{lub}(\Delta A)}{\mathrm{lub}(A)} \;\approx\; 5\cdot 10^{-d}\,, \qquad \frac{\|\Delta b\|}{\|b\|} \;\approx\; 5\cdot 10^{-d}$$

und eine Konditionszahl $\mathrm{cond}(A) \approx 10^{\alpha}$ ergeben in (131) die Abschätzung

$$\frac{\|\Delta x\|}{\|\bar{x}\|} \;\le\; 10^{\alpha-d+1} \quad .$$

Fazit: Wird ein lineares Gleichungssystem $Ax = b$ mit d-stelliger Gleitpunktrechnung gelöst und beträgt die Konditionszahl $\mathrm{cond}(A) \approx 10^{\alpha}$, so sind in der berechneten Lösung nur $d-\alpha-1$ Stellen sicher.

Literatur

[1] **Bollhöfer; Mehrmann:** Numerische Mathematik, Vieweg 2004.

[2] **Schwarz; Köckler:** Numerische Mathematik, 8. Auflage, Vieweg + Teubner 2011.

[3] **Schwarz:** Numerische Mathematik, 3. Auflage, Teubner 1993.

[4] **Hermann:** Numerische Mathematik, 3. Auflage, Oldenbourg-Verlag 2011.

[5] **Gramlich; Werner:** Numerische Mathematik mit Matlab. dpunkt.verlag 2000.

[6] **Dahmen; Reusken:** Numerik für Ingenieure und Naturwissenschaftler, 2. Auflage, Springer 2008.

[7] **Stoer; Bulirsch:** Numerische Mathematik 1, 10. Auflage, Springer-Verlag 2007.

[8] **Stoer; Bulirsch:** Numerische Mathematik 2, 5. Auflage, Springer-Verlag 2005.

[9] **Hämmerlin; Hoffmann:** Numerische Mathematik, 4. Auflage, Springer-Verlag 2007.

[10] **Opfer:** Numerische Mathematik für Anfänger, 5. Auflage, Vieweg + Teubner 2008.

[11] **Werner:** Numerische Mathematik 1, Vieweg-Verlag 1992.

[12] **Werner:** Numerische Mathematik 2, Vieweg-Verlag 1992.

[13] **Kress:** Numerical Analysis, Springer-Verlag 1998.

[14] **Deuflhard:** Numerische Mathematik, Band 1: Eine algorithmisch orientierte Einführung, 3. Auflage, Gruyter-Verlag 2002.

[15] **Deuflhard:** Numerische Mathematik, Band 2: Gewöhnliche Differentialgleichungen, 3. Auflage, Gruyter-Verlag 2008.

[16] **Hanke-Bourgeois:** Grundlagen der Numerischen Mathematik und des Wissenschaftlichen Rechnens, 3. Auflage, Vieweg + Teubner 2009.

[17] **Knorrenschild:** Numerische Mathematik. Eine beispielorientierte Einführung, Hanser-Verlag 2017.