Numerische Mathematik
für Studierende
der Wirtschaftsmathematik,
der Lehrämter
und der Naturwissenschaften

April 2006

M. Hinze

Universität Hamburg

Vorbemerkung

Dieses Numerik–Skript will, kann und soll kein Lehrbuch ersetzen. Vielmehr möchte es den Studenten ermöglichen, der Vorlesung zu folgen, ohne den Zwang mitschreiben zu müssen. Es soll sie auch anleiten, den behandelten Stoff in unterschiedlichen Lehrbüchern nachzulesen und zu ergänzen und sie damit in Stand setzen, beim Kauf eines Lehrbuchs eine begründete Wahl zu treffen.

Daß dieses Skript geschrieben wurde, liegt auch am speziellen Hamburger Studienplan, der eine Einführung in die Numerische Mathematik parallel zu den Grundvorlesungen über Analysis, Lineare Algebra und Analytische Geometrie vorsieht. Dies hat, neben einer frühen Einführung in die Numerik, den Vorteil, daß theoretische Ergebnisse aus Analysis, Lineare Algebra und Analytische Geometrie sowohl ergänzt als auch motiviert werden können.

Natürlich ergeben sich aus dieser Situation auch inhaltliche Konsequenzen. Die Auswahl des Stoffes muß, so weit als möglich, danach ausgerichtet werden, was an mathematischen Kenntnissen durch die Schule oder die parallel laufenden Grundvorlesungen schon bereitgestellt worden ist. Deshalb können eine Reihe von Themen, die üblicherweise zu einer Einführung in die Numerische Mathematik gehören, nicht, noch nicht oder nur marginal behandelt werden. Auch die Reihenfolge des dargebotenen Stoffes ist diesen Rahmenbedingungen unterworfen. Themenkomplexe, die inhaltlich zusammengehören, müssen zum Teil zeitlich entzerrt werden, bis die notwendigen Vorkenntnisse bereitgestellt worden sind. Aus diesen Gründen sind Ergänzungen durch die Lehrbuchliteratur unverzichtbar.

Die meisten Numerik–Bücher setzen die Kenntnisse aus den Anfänger–Vorlesungen voraus und sind deshalb als einziges Vorlesungsbegleitmaterial nur bedingt geeignet. Auch dies ist ein Grund für die Erstellung dieses Skriptes.

Diese Schrift geht zurück auf ein Skript, das von Christoph Maas angefertigt wurde und von vielen Kollegen, die seither die Numerik gelesen haben (Werner, Hass, Opfer, Geiger, Hofmann, Ulbrich, um nur einige zu nennen), ergänzt, umgearbeitet und aktualisiert worden ist.

Inhaltsverzeichnis

0 Einführung **1**

0.1 Produktionsplanung, Arbeitswerttheorie: Das Leontief Modell 1

0.2 Elektrische Netzwerke 3

0.3 Populationsmodelle, hier das Räuber-Beute Modell 5

0.4 Portfolio Modellierung und Optimierung 6

1 Zahlendarstellung und Rundungsfehler **9**

1.1 Rundungsfehler 9

1.2 Gleitkommadarstellung 10

1.3 Konditionszahlen 11
1.4 Hilfe gegen rundungsbedingte Rechenfehler 13

2 Lineare Gleichungssysteme **15**
2.1 Das Gaußsche Eliminationsverfahren 16
2.1.1 Numerische Schwierigkeiten, Pivot–Suche 22
2.1.2 Bemerkungen zur Programmierung der Pivotsuche 26
2.1.3 Rechenaufwand 27
2.1.4 Variable rechte Seiten 27
2.2 Die Cholesky-Zerlegung 28
2.2.1 Normen und Fehlerabschätzungen 31
2.3 Iterative Lösung linearer Gleichungssysteme 34
2.3.1 Iterative Verfahren mit endlich vielen Iterationen 44

3 Nichtlineare Gleichungen **53**
3.1 Motivation 53
3.2 Verfahren und Konvergenzsätze 55

4 Interpolation **63**
4.1 Polynominterpolation 63
4.1.1 Interpolationsfehler 71
4.1.2 Optimale Stützstellen, Tschebyscheff-Knoten 75
4.2 Spline Interpolation 82
4.2.1 Kubische Splines 83
4.2.2 Splines k-ter Ordnung 87

5 Numerische Integration **89**
5.1 Interpolatorische Quadratur 89
5.2 Quadraturfehler bei interpolatorischen Verfahren 91
5.3 Zusammengesetzte Formeln 94
5.3.1 Summierte Trapez Regel 94
5.3.2 Summierte Simpson Regel 95

5.4 Adaptive Quadraturformeln 95
5.5 Extrapolation und Romberg Integration 99
5.6 Gauß Quadratur 100

6 Lineare Optimierung **105**

6.1 Lineare Optimierungsaufgaben: Formulierungen 110
6.2 Beschreibung von Ecken 111
6.3 Basislösungen 113
6.4 Das Simplex–Verfahren 115
6.5 Bestimmung einer Ausgangsbasislösung 122
6.6 Praktische Durchführung 124

0 Einführung

Unter Numerischer Mathematik versteht man die (zahlenmäßige) Lösung mathematischer Probleme mit Hilfe eines Computers. Insbesondere umfasst die Numerik folgende Aufgabenstellungen:

a) Entwicklung von problemangepassten Algorithmen

b) Analyse von Algorithmen im Hinblick auf

- Effizienz
- Zuverlässigkeit
- Genauigkeit
- Sensitivität (z.B. bzgl. Rundungsfehlern)

Vor der Lösung mathematischer Probleme steht deren Generierung. Mathematische Probleme resultieren häufig aus der mathematischen Modellierung wirtschaftlicher und technischer Aufgabenstellungen.

0.1 Produktionsplanung, Arbeitswerttheorie: Das Leontief Modell

In der Volkswirtschaftslehre muss man typischerweise eine ganze Reihe von Objekten untersuchen, die in wechselseitigen Abhängigkeiten stehen. Wir betrachten eine Volkswirtschaft bestehend aus n Sektoren. Jeder Sektor $i \in \{1, \ldots, n\}$ produziert Güter q_i (Angabe in Mengeneinheiten). Der Faktor $a_{ik} \in [0, 1]$ gebe an, welcher Anteil einer Mengeneinheit des Gutes q_k zur Produktion des i-ten Gutes benötigt wird.

Frage: Wieviele Mengeneinheiten q_i des i-ten Gutes muß jeder Sektor i produzieren, um einen Überschuß y_i zu garantieren $(i = 1, \ldots, n)$?

Wir bilanzieren:

$$y_i = q_i - \sum_{k=1}^{n} a_{ik} q_k \text{ für } i = 1, \ldots, n.$$

Sammeln wir die Nettoausgaben $y_1, \ldots, y_n$ und die zu produzierenden Güter $q_1, \ldots, q_n$ in Vektoren $y \in \mathbb{R}^n$ und $q \in \mathbb{R}^n$, so können wir diese Bilanzen kompakt in der Form

$$y = (E - A)q$$

schreiben. Dabei bezeichnet $E \in \mathbb{R}^{n,n}$ die Einheitsmatrix und $A := (a_{ik})_{i,k=1}^{n} \in \mathbb{R}^{nn}$ $(\equiv M(n \times n; \mathbb{R}))$ die Matrix der Produktionsfaktoren. Dieses Gleichungssystem kann nach q aufgelöst werden gdw $(E - A)^{-1}$ existiert. Nehmen wir also an, dass die Matrix $E - A$ invertierbar ist. Wir fragen uns als Nächstes, ob immer eine volkswirtschaftlich vernünftige Lösung unserer Aufgabenstellung existiert. Dazu fordern wir, dass aus $y \geq 0$ auch $q \geq 0$ folgt (Ungleichungen bei Matrizen und Vektoren immer Komponentenweise verstehen, falls nichts anderes festgelegt wird!). Hinreichend dafür ist sicherlich $(E - A)^{-1} \geq 0$, denn dann können wir folgern

$$(E - A)^{-1} \geq 0 \text{ und } y \geq 0 \Rightarrow q = (E - A)^{-1} y \geq 0.$$

Wir wollen die Matrix $E - A$ *produktiv* nennen, falls sie diese Eigenschaft besitzt. Es ist offensichtlich, dass der Nachweis dieser Eigenschaft der Matrix $E - A$ zusätzliche Anstrengungen erfordert.

Um das Vorgehen an einem Beispiel zu erläutern, betrachten wir eine Volkswirtschaft mit nur 3 Sektoren

- Landwirtschaft,
- produzierendes Gewerbe (Industrie) und
- Transportwesen.

Wir nehmen folgende Abhängigkeiten an:

a) Bei der Produktion landwirtschaftlicher Güter (Lebensmittel) im Werte von 1000,00 EURO werden landwirtschaftliche Güter im Wert von 300,00 EURO (Getreide, . . .), Transportleistungen im Wert von 100,00 EURO und industrielle Güter im Werte von 200,00 EURO (Landmaschinen, . . .) gebraucht.

b) Die Herstellung von Industrieprodukten im Wert von 1000,00 EURO benötigt Lebensmittel im Wert von 200,00 EURO, andere Industrieprodukte im Wert von 400,00 EURO und Transportleistungen im Wert von 100,00 EURO.

c) Zur Erbringung von Transportleistungen im Wert von 1000,00 EURO sind Lebensmittel im Wert von 100,00 EURO, industriell gefertigte Güter (Fahrzeuge, Treibstoff,. . .) im Wert von 200,00 EURO und Transportleistungen (z.B. Treibstoffversorgung) im Wert von 100,00 EURO erforderlich.

Welche Mengen müssen die einzelnen Sektoren produzieren, damit die Gesamtwirtschaft folgende Überschüsse erwirtschaftet;

Landwirtschaft	20.000	EURO
Industrie	40.000	EURO
Transportwesen	0	EURO

x_L gebe an, wieviele Mengeneinheiten im Wert von je 1000,00 EURO die Landwirtschaft produziert. x_I und x_T seien die entsprechenden Werte für Industrie und Transportwesen. Die gesuchten Mengen müssen also folgende Bedingungen erfüllen:

$$\begin{aligned} 0.7\,x_L - 0.2\,x_I - 0.1\,x_T &= 20 \\ -0.2\,x_L + 0.6\,x_I - 0.1\,x_T &= 40 \\ -0.1\,x_L - 0.2\,x_I + 0.9\,x_T &= 0 \end{aligned}$$

Diese Aufgabenstellung verlangt also — etwas allgemeiner formuliert — das Lösen eines Systems linearer Gleichungen

$$\begin{array}{lcccccl} a_{11}\,x_1 & + & a_{12}\,x_2 & + & \ldots & + & a_{1n}\,x_n = b_1 \\ a_{21}\,x_1 & + & a_{22}\,x_2 & + & \ldots & + & a_{2n}\,x_n = b_2 \\ \vdots & & & & & & \vdots \\ a_{m1}\,x_1 & + & a_{m2}\,x_2 & + & \ldots & + & a_{mn}\,x_n = b_m \end{array} \tag{1}$$

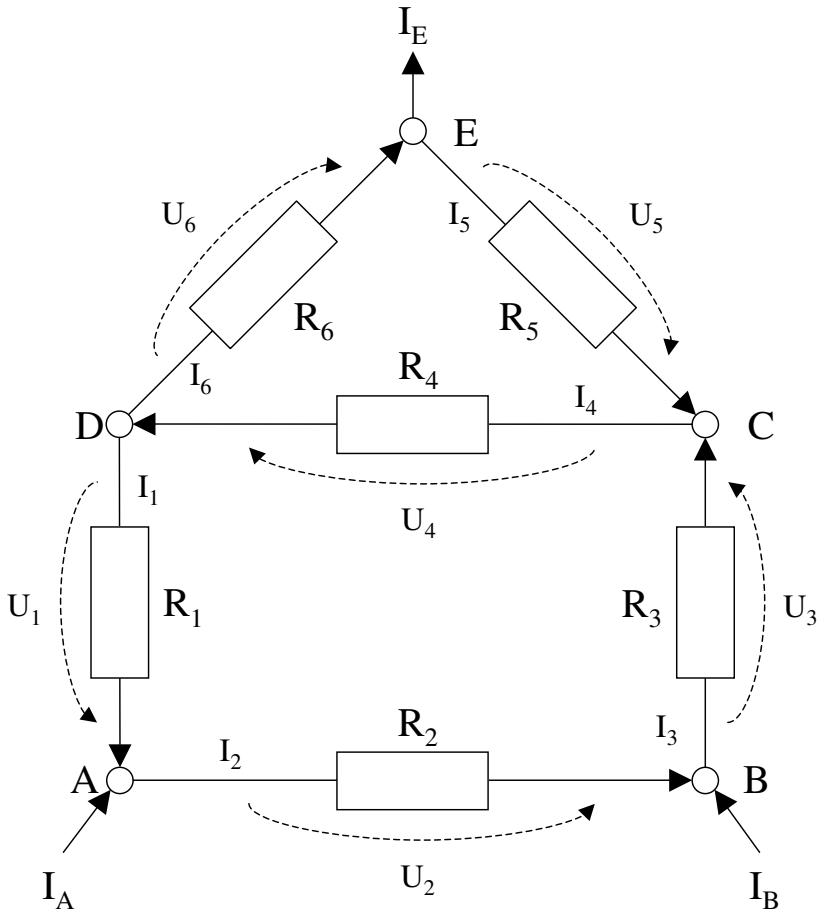
mit gegebenen Zahlen a_{ij} und b_i und gesuchten Größen x_j, $i = 1, \ldots, m$, $j = 1, \ldots, n$ (m Gleichungen, n Unbekannte).

In der Linearen Algebra wird untersucht, wann ein solches System lösbar ist und welche Struktur (Vektorraum, lineare Mannigfaltigkeit,. . .) die Menge der Lösungen hat.

In der numerischen Mathematik wollen wir untersuchen, wie wir in dem Spezialfall, dass das System genau eine Lösung hat, diese möglichst schnell und genau berechnen können.

0.2 Elektrische Netzwerke

Gegeben sei das abgebildete elektrische Netzwerk mit WiderständenR_k, Spannungen U_k und Strömen I_k, I_A, I_B, I_E, $k = 1, \ldots, 6$.



Gegeben seien die Widerstände $R_1, \ldots, R_6$ (Einheit Ohm) und die Ströme I_A, I_B (Einheit Ampere). Gesucht sind die Ströme $I_1, \ldots, I_6$ und I_E.

An jedem Widerstand gilt das Ohmsche Gesetz:

$$\text{Spannung} = \text{Widerstand} \times \text{Strom} \quad (\text{kurz: } U = R \cdot I)$$

In jedem Knoten gilt die erste Kirchhoffsche Regel:

$$\text{Summe der Teilströme im Knoten} = 0.$$

In jeder Masche gilt die zweite Kirchhoffsche Regel:

$$\text{Summe der Teilspannungen} = 0.$$

Dies ergibt (wir eliminieren die Spannungen U_k mit Hilfe des Ohmschen Gesetzes $U_k = R_k I_k$):

$$\begin{aligned}
I_1 - I_2 + I_A &= 0 && \text{(1. Kirchhoffsche Regel im Knoten } A\text{)}\\
I_2 - I_3 + I_B &= 0 && \text{(1. Kirchhoffsche Regel im Knoten } B\text{)}\\
I_3 - I_4 + I_5 &= 0 && \text{(1. Kirchhoffsche Regel im Knoten } C\text{)}\\
-I_1 + I_4 - I_6 &= 0 && \text{(1. Kirchhoffsche Regel im Knoten } D\text{)}\\
-I_5 + I_6 - I_E &= 0 && \text{(1. Kirchhoffsche Regel im Knoten } E\text{)}\\
R_1 I_1 + R_2 I_2 + R_3 I_3 + R_4 I_4 &= 0 && \text{(2. Kirchhoffsche Regel in der Masche } ABCD\text{)}\\
R_4 I_4 + R_5 I_5 + R_6 I_6 &= 0 && \text{(2. Kirchhoffsche Regel in der Masche } CDE\text{)}
\end{aligned}$$

In Matrixschreibweise:

$$\begin{pmatrix}
1 & -1 & 0 & 0 & 0 & 0 & 0\\
0 & 1 & -1 & 0 & 0 & 0 & 0\\
0 & 0 & 1 & -1 & 1 & 0 & 0\\
-1 & 0 & 0 & 1 & 0 & -1 & 0\\
0 & 0 & 0 & 0 & -1 & 1 & -1\\
R_1 & R_2 & R_3 & R_4 & 0 & 0 & 0\\
0 & 0 & 0 & R_4 & R_5 & R_6 & 0
\end{pmatrix}
\begin{pmatrix} I_1\\ I_2\\ I_3\\ I_4\\ I_5\\ I_6\\ I_E \end{pmatrix}
=
\begin{pmatrix} -I_A\\ -I_B\\ 0\\ 0\\ 0\\ 0\\ 0 \end{pmatrix}.$$

Konkrete Zahlenwerte:

$$R_1 = 1,\ \ R_2 = 2,\ \ R_3 = 5,\ \ R_4 = 4,\ \ R_5 = 1,\ \ R_6 = 17,\ \ I_A = 7,\ \ I_B = -1.$$

Dann ergibt sich das folgende **Lineare Gleichungssystem**:

$$\begin{pmatrix}
1 & -1 & 0 & 0 & 0 & 0 & 0\\
0 & 1 & -1 & 0 & 0 & 0 & 0\\
0 & 0 & 1 & -1 & 1 & 0 & 0\\
-1 & 0 & 0 & 1 & 0 & -1 & 0\\
0 & 0 & 0 & 0 & -1 & 1 & -1\\
1 & 2 & 5 & 4 & 0 & 0 & 0\\
0 & 0 & 0 & 4 & 1 & 17 & 0
\end{pmatrix}
\begin{pmatrix} I_1\\ I_2\\ I_3\\ I_4\\ I_5\\ I_6\\ I_E \end{pmatrix}
=
\begin{pmatrix} -7\\ 1\\ 0\\ 0\\ 0\\ 0\\ 0 \end{pmatrix}.$$

Lineare Gleichungssysteme effizient auf dem Rechner zu lösen, ist eine zentrale Aufgabenstellung der Numerik. In praktischen Anwendungen könnte das betrachtete Netzwerk aus 100000 Widerständen und mehr bestehen. wir hätten dann mindestens 100000 Unbekannte und damit ein riesiges Gleichungssystem zu lösen.

Zur effizienten Lösung linearer Gleichungssysteme werden wir u.A. das Gaußsche Eliminationsverfahren kennenlernen. Hierbei wird das System schrittweise auf obere Dreiecksgestalt überführt:

$$\left(\begin{array}{ccccccc|c}
1 & -1 & 0 & 0 & 0 & 0 & 0 & -7\\
0 & 1 & -1 & 0 & 0 & 0 & 0 & 1\\
0 & 0 & 1 & -1 & 1 & 0 & 0 & 0\\
-1 & 0 & 0 & 1 & 0 & -1 & 0 & 0\\
0 & 0 & 0 & 0 & -1 & 1 & -1 & 0\\
1 & 2 & 5 & 4 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 4 & 1 & 17 & 0 & 0
\end{array}\right)$$

$$\rightarrow \left(\begin{array}{ccccccc|c} 1 & -1 & 0 & 0 & 0 & 0 & 0 & -7 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & -1 & 0 & -7 \\ 0 & 0 & 0 & 0 & -1 & 1 & -1 & 0 \\ 0 & 3 & 5 & 4 & 0 & 0 & 0 & 7 \\ 0 & 0 & 0 & 4 & 1 & 17 & 0 & \end{array}\right)$$

$$\rightarrow \left(\begin{array}{ccccccc|c} 1 & -1 & 0 & 0 & 0 & 0 & 0 & -7 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & -1 & 0 & -6 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & 0 \\ 0 & 0 & 8 & 4 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 4 & 1 & 17 & 0 & 0 \end{array}\right)$$

$$\vdots$$

$$\rightarrow \left(\begin{array}{ccccccc|c} 1 & -1 & 0 & 0 & 0 & 0 & 0 & -7 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 12 & -8 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & -6 \\ 0 & 0 & 0 & 0 & 0 & 20.667 & 0 & 20.667 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & -6 \end{array}\right)$$

Dieses gestaffelte Gleichungssystem kann nun leicht aufgelöst werden:

$$I_E = \frac{-6}{-1} = 6, \quad I_6 = \frac{20.667}{20.667} = 1, \quad I_5 = \frac{-6 - (-1) \cdot 1}{1} = -5 \quad \text{usw.}$$

Insgesamt ergibt sich

$$(I_1, \ldots, I_6, I_E) = (-4, 3, 2, -3, -5, 1, 6).$$

0.3 Populationsmodelle, hier das Räuber-Beute Modell

Wir stellen uns einen Lebensraum vor mit unbeschränkten Ressourcen, der nur von 2 Spezies bewohnt wird, nämlich den Grfr und FrGr. Die GrFr könnten sich ungestört vermehren, wären da nicht die FrGr, deren einzige Nahrungsquelle die GrFr sind und welche letztendlich aussterben würden ohne diese Nahrungsquelle. Wir fragen uns nun, wie sich in dieser Situation die Sorten GrFr und FrGr ausgehend von Anfangsspopulationen G_0 und F_0 in dem Zeitraum $(0, T]$ mit $T > 0$ entwickeln werden.

Für die zeitlichen Populationsverläufe $G(t)$ der GrFr und $F(t)$ der FrGr erhalten wir folgendes mathematische Modell;

$$\left.\begin{array}{l} \dot{G}(t) = aG(t) - bG(t)F(t), \\ \dot{F}(t) = -cF(t) + dF(t)G(t), \\ G(0) = G_0,\ F(0) = F_0 \end{array}\right\} \text{ für } t \in (0, T]. \tag{2}$$

In diesem Modell bezeichnen a, b, c und d positive Proportionalitätskonstanten. Die Angabe von geschlossenen Populationsverläufen gelingt i.d.R. nicht (denn es handelt sich um ein System nichtlinearer Differentialgleichungen). Verbleibt aber immer noch die Möglichkeit, numerische Näherungslösungen für die Populationsverläufe $G(t_k), F(t_k)(k = 0, \ldots, n)$ zu den Zeitpunkten $0 \equiv t_0 < t_1 < \cdots < t_k < \cdots < t_n \equiv T$ zu berechnen. Dazu ersetzen wir in (2) Ableitungen durch Differenzenquotienten und erhalten für $k = 0, \ldots, n-1$

$$\begin{aligned} &\frac{G(t_{k+1}) - G(t_k)}{t_{k+1} - t_k} \approx aG(t_{k+1}) - bG(t_{k+1})F(t_{k+1}), \\ &\frac{F(t_{k+1}) - F(t_k))}{t_{k+1} - t_k} \approx -cF(t_{k+1}) + dF(t_{k+1})G(t_{k+1}), \\ &G(0) = G_0,\ F(0) = F_0. \end{aligned} \tag{3}$$

Gleichheit können wir in (3) natürlich nicht mehr erwarten, denn Differenzenquotienten approximieren Ableitungen (bestenfalls) ja nur. Wir sind allerdings guter Hoffnung, dass die aus dem nichtlinearen Gleichungssystem

$$\left.\begin{aligned} &\frac{G^{k+1} - G^k}{t_{k+1} - t_k} = aG^{k+1} - bG^{k+1}F^{k+1}, \\ &\frac{F^{k+1} - F^k}{t_{k+1} - t_k} = -cF^{k+1} + dF^{k+1}G^{k+1}, \\ &G^0 = G_0,\ F^0 = F_0 \end{aligned}\right\} \text{ für } k = 0, \ldots, n-1, \tag{4}$$

berechneten Größen G^k, F^k gute Näherungen der gesuchten Populationen $G(t_k), F(t_k)$ darstellen (falls die Feinheit $\max\{|t_{k+1} - t_k|, 0 \leq k \leq n-1\}$ des 'Gitters' $t_0 < \cdots < t_n$ klein genug ist). Die Gleichungen (4) definieren für jedes $k \in \{0, \ldots, n-1\}$ ein nichtlineares Gleichungssystem zur Bestimmung von G^{k+1}, F^{k+1} (bei gegebenen G^k, F^k). Tipp: Programmieren Sie die Aufgabenstellung aus (4) und tragen Sie F gegen G im sogenannten Phasendiagramm auf, vergleiche Abb. 1.

0.4 Portfolio Modellierung und Optimierung

Jeder Investor sollte sich darüber im Klaren sein, daß Gewinnvergrösserung einhergeht mit steigendem Investitionsrisiko. Stellen wir uns folgende Situation vor;

Es liegen n Investionsmöglichkeiten vor mit Renditen r_i $(i = 1, \ldots, n)$.

Die Renditen sind a-priori nicht bekannt und werden als normalverteilte Zufallsvariablen ($N(\mu, \sigma^2)$-verteilt) angenommen. Die Erwartungswerte seien $\mu_i = E[r_i]$ und die Varianzen $V(r_i) = \sigma_i^2 = E[(r_i - \mu_i)^2]$.

Die Menge an flüssigen Finanzmitteln sei gleich 1.

Wir konstruieren unser Portfolio, indem wir x_i Anteile unserer flüssigen Mittel in Investionsmöglichkeit i investieren. Dabei gilt $\sum_{i=1}^{n} x_i = 1$.

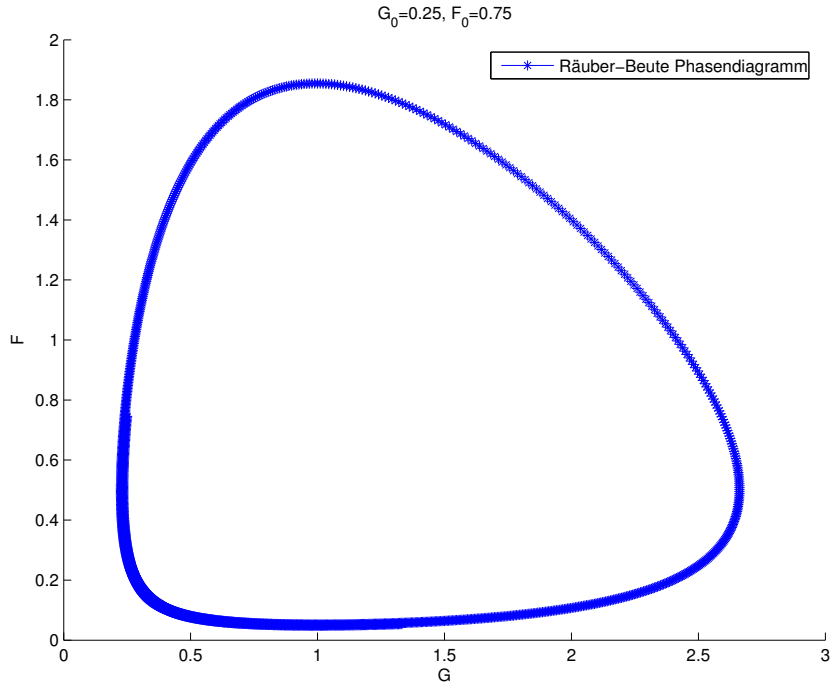


Abbildung 1: Phasendiagramm des Räuber-Beute Modells

Wie wird jetzt die Rendite modelliert? Es gilt natürlich

$$R = \sum_{i=1}^{n} x_i r_i.$$

Jetzt sollten wir noch ein Maß dafür aufstellen, wie wir ein Portfolio bewerten. Dazu betrachten wir den Erwartungswert der Rendite

$$E[R] = E\left[\sum_{i=1}^{n} x_i r_i\right] = \sum_{i=1}^{n} x_i E[r_i] = x^t \mu,$$

wobei wir die Anteile x_i und die Erwartungswerte μ_i in Vektoren $x = (x_1, \ldots, x_n)^t$ und $\mu = (\mu_1, \ldots, \mu_n)^t$ zusammengefaßt haben. Und schließlich noch die Varianz der Rendite,

$$E[(R - E[R])^2] = \sum_{i=1}^{n} \sum_{j=1}^{n} x_i x_j \sigma_i \sigma_j \rho_{ij} = x^t G x,$$

wobei $G = (g_{ij})_{i,j=1,\ldots,n}$, $g_{ij} := \sigma_i \sigma_j \rho_{ij}$ und

$$\rho_{ij} := \frac{E[(r_i - \mu_i)(r_j - \mu_j)]}{\sigma_i \sigma_j}, \quad i, j = 1, \ldots, n,$$

die Kovarianz zwischen den Anlagen i und j bezeichnet. Sie ist ein Maß dafür, inwieweit sich die Renditen der Anlagen i und j in die selbe Richtung entwickeln.

Wie soll unser Portfolio jetzt aussehen bzw. welches sind die Kriterien, nach denen wir unser Portfolio gestalten wollen? Der gesunde Menschenverstand schlägt uns vor, möglichst viel

Gewinn bei kleinen Risiken zu erwirtschaften (denn wir sind ja keine Zocker!), in den oben eingeführten Termen heißt das

Optimierungsziel: Maximale Rendite bei möglichst kleiner Varianz des Portfolios.

Wir landen somit bei der Optimierungsaufgabe

$$\max_{x\in\mathbb{R}^n} F(x) := x^t\mu - \kappa x^t G x \quad \text{bei } \sum_{i=1}^{n} x_i = 1,\ x \geq 0. \tag{5}$$

Dabei variiert der Parameter $\kappa \in [0, \infty)$ und ist offensichtlich ein Maß dafür, wie wichtig uns geringes Risiko bei der Portfoliogestaltung letztendlich ist (Großes κ meint kleines Risiko). Dieses Modell der Portfoliogestaltung geht zurück auf Markowitz [8].

Bis hierhin haben wir verschwiegen, wie wir uns die Daten μ_i und σ_i und ρ_{ij} für das Optimierungsproblem (5) verschaffen. Das ist wiederum ein Optimierungsproblem für sich. Bei klassischen Börsenwerten könnten z.B. die Daten aus der Vergangenheit genommen werden (die vergangenen 5 Jahre etwa). Das geht allerdings nicht bei Startups. Am besten ist wohl eine Mischung aus Erfahrungen bzw. Kenntnis der Materie und Daten aus der Vergangenheit, gepaart mit einem guten Schuß Gottvertrauen oder so.

Nun nehmen wir an, daß der Vektor μ, die Matrix G und der Parameter κ in (5) bekannt sind. Wir haben uns folgende Fragen zu stellen.

> Besitzt das Optimierungsproblem (5) eine Lösung?
>
> Sind mehrere Lösungen möglich?
>
> Ganz wichtig: Wie können Lösungen numerisch berechnet werden?

1 Zahlendarstellung und Rundungsfehler

Wir wollen hier nur eine vereinfachte, beispielorientierte und keineswegs vollständige Einführung in die Problematik des Rechnens an Computern geben. Eine ausführlichere, gut lesbare Darstellung dieses Themenkreises, die allerdings gewisse Grundkenntnisse der Analysis voraussetzt (Differenzierbarkeit, Taylorreihe), findet man z.B. in Stoer: Numerische Mathematik I, §1.

In den (von uns benutzten) Digitalrechnern hat man nicht die reellen Zahlen zur Verfügung, sondern nur eine **endliche** Menge A von Zahlen, die sog. Maschinenzahlen. Also gibt es in jedem Intervall zwischen zwei benachbarten Maschinenzahlen unendlich viele Zahlen, die der Computer nicht „kennt". Man muss also überlegen, wie man diese Maschinenzahlen am zweckmäßigsten auswählt und welche Konsequenzen diese Auswahl für die Ergebnisse unserer Rechnungen besitzt.

1.1 Rundungsfehler

Unabhängig von der Auswahl der Maschinenzahlen muss eine Zahl $x \notin A$ durch eine gerundete Zahl $rd(x) \in A$ angenähert werden. Vernünftig ist hierbei folgende Optimalitätsforderung

$$|rd(x) - x| \;\leq\; |y - x| \qquad \forall\, y \in A\,.$$

Liegt x genau in der Mitte zwischen zwei Maschinenzahlen, benötigt man eine Zusatzregel: man wählt z.B. die betragsgrößere der beiden möglichen Zahlen.

Wie muss nun eine Fehlergröße festgelegt werden, die Auskunft über die Genauigkeit einer Approximation gibt? Grundsätzlich gibt es zwei Fehlertypen:

1) absoluter Fehler $\quad e_{\text{abs}} := rd(x) - x$

2) relativer Fehler $\quad e_{\text{rel}} := \dfrac{rd(x) - x}{x}$

Ihre Bedeutung machen wir uns an Beispielen klar.

Die Entfernung der Mittelpunkte von Erde und Mond bis auf einen absoluten Fehler von $5\ m$ zu bestimmen, ist außerordentlich genau. Für die Angabe der Größe einer Parklücke ist eine Fehlermarke von $5\ m$ äußerst ungenügend. Ein absoluter Fehler von ca. $50\ cm$ ist hier angebrachter. Diese Fehlergröße ist wiederum bei der Bestimmung der Wellenlänge des sichtbaren Lichts (etwa $0.4 \cdot 10^{-6}\ m$ bis $0.8 \cdot 10^{-6}\ m$) mehr als wertlos.

Die Tolerierbarkeit eines Fehlers wird also weniger durch seine absolute Größe als durch sein Verhältnis zur Größenordnung des exakten Werts festgelegt (relativer Fehler).

Es hat sich deshalb als sinnvoll erwiesen, die Maschinenzahlen so zu verteilen, dass für jede Zahl x aus den Intervallen $[-M, -m]$ bzw. $[m, M]$ der Zahlen, die man im Rechner darstellen will, (m bzw. M sind die kleinste bzw. größte positive ganze Zahl, die man im Rechner darstellen will) der relative Fehler $\dfrac{rd(x) - x}{x}$ betragsmäßig eine möglichst kleine Schranke nicht übersteigt. Dies führt zur Benutzung der Gleitkommadarstellung für Zahlen im Rechner, die dieser Bedingung genügt, wie wir noch zeigen werden.

1.2 Gleitkommadarstellung

Eine Zahl ungleich Null wird dargestellt in der Form

$$a \quad = \quad \pm\, a_0 \,.\, a_1\, a_2 \;\; \ldots \;\; a_{t-1} \;\cdot\; g^p$$

Hierbei ist g die *Basis* ($g = 10,$ Dezimalsystem, wird bei der Eingabe von Zahlen in den Rechner und bei der Ausgabe von Ergebnissen benutzt, intern benutzen die Rechner das Dualsystem, $g = 2$).

Der *Exponent* p ist eine ganze Zahl, die betragsmäßig rechnerabhängig beschränkt wird (z.B. $|p| \leq 99$),

Die *Mantisse* $a_0.a_1 \ldots a_{t-1},\;\; a_j \in \{0, 1, 2, \ldots, g-1\},\;\; j = 0, \ldots, t-1$ ist eine Ziffernfolge der Mantissenlänge t ($\in \mathbb{N}$). t wird rechnerabhängig fixiert. Es wird gefordert

$$a_0 \;\neq\; 0.$$

Die Gleitkommazahl hat den Wert

$$a = (a_0 g^0 + a_1 g^{-1} + \cdots + a_1 g^{-(t-1)}) g^p.$$

Im Rechner wird zur Ein- und Ausgabe von Ergebnissen üblicherweise das Dezimalsystem $(g = 10)$ benutzt. Zahlen, die nicht in diese Darstellungsform passen (Zahlen mit größerer Mantissenlänge als t, z.B. Wurzeln und unendliche Dezimalbrüche) werden im allg. betragsmäßig nach folgender Vorschrift gerundet:

Für $|x| = a_0 \,.\, a_1\, a_2 \;\; \ldots \;\; a_t\, a_{t+1} \;\; \ldots \;\cdot\; 10^p$ ist

$$|rd(x)| \quad = \quad \begin{cases} a_0 \,.\, a_1 \;\; \ldots \;\; a_{t-1} \;\cdot\; 10^p, & \text{falls} \quad a_t \;<\; 5 \\ (a_0 \,.\, a_1 \;\; \ldots \;\; a_{t-1} + 10^{-(t-1)}) \;\cdot\; 10^p, & \text{falls} \quad a_t \;\geq\; 5 \end{cases}$$

Das Vorzeichen bleibt ungeändert.

Wir berechnen den maximalen relativen Fehler von $rd(x)$ der Rundung in der Gleitkommadarstellung mit $g = 10$.

Laut Rundungsvorschrift gilt: (falls $|x| \;\geq\; m$)

$$|rd(x) - x| \;\leq\; 5 \cdot 10^{-t} \cdot 10^p\,,$$

$$|x| \;\geq\; a_0 \,.\, a_1\, a_2 \ldots a_{t-1} \cdot 10^p \geq 10^p \quad (\text{beachte } a_0 \neq 0)$$

$$\text{also} \qquad \tfrac{1}{|x|} \;\leq\; 10^{-p}$$

$$\text{somit} \qquad \underline{\underline{\tfrac{|rd(x)-x|}{|x|}}} \;\leq\; 5 \cdot 10^{-t} \cdot 10^p \cdot 10^{-p} \quad = \quad \underline{\underline{\tfrac{1}{2} \cdot 10^{-t+1}}}\;.$$

Auf die gleiche Weise zeigt man für eine beliebige Basis g

$$\frac{|rd(x) - x|}{|x|} \quad \leq \quad \frac{1}{2} \cdot g^{-t+1} .$$

Diese Zahl heißt *Maschinengenauigkeit*.

Sie gilt einheitlich für den Gesamtbereich der darstellbaren Zahlen, und hängt bei vorgegebener Basis g nur von der Mantissenlänge t (der Anzahl der verwendeten Ziffern) ab.

Das folgende Beispiel zeigt, dass dies für den absoluten Fehler nicht gilt.

Beispiel.
$g = 10,\ t = 10,\ |p| \leq 99.$

$$\left.\begin{array}{l} 1.000000000 \cdot 10^{-99} \\ 1.000000001 \cdot 10^{-99} \end{array}\right\} \quad \text{Differenz} \qquad 1 \cdot 10^{-9} \cdot 10^{-99} = 10^{-108}$$

$$\left.\begin{array}{l} 9.999999998 \cdot 10^{99} \\ 9.999999999 \cdot 10^{99} \end{array}\right\} \quad \text{Differenz} \qquad 1 \cdot 10^{-9} \cdot 10^{99} = 10^{90}$$

Die absolute Differenz benachbarter Zahlen, und somit auch der Rundungsfehler, wächst mit dem Betrag der dargestellten Zahl.

Standards

Alle aktuellen Computer arbeiten mit Binärzahlen, d.h. $g = 2$. Als Standard hat sich der IEEE-Standard durchgesetzt:

a) Einfache Genauigkeit:

$$g = 2, \quad t = 24, \quad -126 \leq p \leq 127,$$

b) Doppelte Genauigkeit:

$$g = 2, \quad t = 53, \quad -1022 \leq p \leq 1023,$$

1.3 Konditionszahlen

Im Zusammenhang mit Rechenfehlern (z.B. durch Rundung) ist es von großer Bedeutung, wie stark Rechenfehler in den Eingangsdaten durch einen Algorithmus verstärkt werden.

Wir betrachten hierzu einen Algorithmus, der aus einer Zahl $x \in \mathbb{R}$ eine Zahl $y = f(x) \in \mathbb{R}$ berechnet. Sei nun $\tilde{x}$ ein Näherungswert für x (z.B. $\tilde{x} = rd(x)$). Wir bezeichnen mit Δx den absoluten Fehler, d.h.

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

Verwenden wir als Eingabe für den Algorithmus nicht x, sondern die Näherung $\tilde{x}$, und nehmen wir idealisierend an, dass der Algorithmus keine weiteren Rechenfehler hinzufügt, so erhalten

wir das Ergebnis $\tilde{y} = f(\tilde{x})$ anstelle des exakten Ergebnisses $y = f(x)$. Der absolute Fehler in y ist dann

$$\Delta y = \tilde{y} - y = f(\tilde{x}) - f(x).$$

Nehmen wir an, dass f stetig differenzierbar ist und dass $|\Delta x|$ klein ist, so gilt in guter Näherung

$$\Delta y = f(x + \Delta x) - f(x) = \frac{df}{dx}(x)\Delta x + \rho(\Delta x) \approx \frac{df}{dx}(x)\Delta x.$$

Hier haben wir die Taylor-Entwicklung verwendet und bezeichnen mit ρ das Restglied. Es gilt

$$\lim_{t \to 0} \frac{\rho(t)}{t} = 0.$$

Wegen

$$\Delta y \approx \frac{df}{dx}(x)\Delta x$$

gibt die Zahl $\frac{df}{dx}(x)$ in guter Näherung den Verstärkungsfaktor des absoluten Fehlers an.

Bezeichne

$$\varepsilon_x = \frac{\Delta x}{x}$$

den relativen Fehler von x. Für den relativen Fehler von y gilt dann

$$\varepsilon_y = \frac{\Delta y}{y} \approx \frac{\frac{df}{dx}(x)\Delta x}{y} = \frac{x\frac{df}{dx}(x)}{f(x)}\varepsilon_x.$$

Die Zahl $\frac{x\frac{df}{dx}(x)}{f(x)}$ gibt also in guter Näherung den Verstärkungsfaktor des relativen Fehlers an. Wir fassen zusammen:

Definition 1.1. Mit $\kappa_{\text{abs}} := \frac{df}{dx}(x)$ bezeichnen wir die *absolute Konditionszahl* des Algorithmus' $x \mapsto y = f(x)$. Mit $\kappa_{\text{rel}} := \frac{x\frac{df}{dx}(x)}{f(x)}$ bezeichnen wir die *relative Konditionszahl* des Algorithmus' $x \mapsto y = f(x)$.

Beispiel.
Zur Illustration betrachten wir die Subtraktion $y = x - 1$ bei $x = 1 + t$, $t \in \mathbb{R}$. Das Problem besteht also in der Auswertung von $x \mapsto y = f(x)$ mit $f(x) = x - 1$ bei $x = 1 + t$. Es gilt

$$\kappa_{\text{abs}} = \frac{df}{dx}(x) = 1,$$
$$\kappa_{\text{rel}} = \frac{x\frac{df}{dx}(x)}{f(x)} = \frac{1+t}{t} = \frac{1}{t} + 1.$$

Wir sehen, dass die relative Konditionszahl sehr groß ist, falls x nahe bei 1 liegt.

Dieses Phänomen nennt man *Auslöschung*. Betrachte z.B.

$$y = x - 1 = 1.00015 - 1 = 0.00015 \quad \text{für} \quad x = 1.00015.$$

Bei fünfstelliger Rechnung im Dezimalsystem gilt

$$\tilde{x} = rd(x) = 1.0002, \quad \Delta x = \tilde{x} - x = 0.00005, \quad \varepsilon_x = -4.99925 \cdot 10^{-5}.$$

Anstelle des exakten y wird berechnet:

$$\tilde{y} = y + \Delta y = rd(x) - 1 = 1.0002 - 1 = 0.0002.$$

Der absolute Fehler von y ist also

$$\Delta y = \tilde{y} - y = 0.0002 - 0.00015 = 0.00005.$$

Der relative Fehler von y ist

$$\varepsilon_y = \frac{\Delta y}{y} = \frac{0.00005}{0.00015} = 0.33333.$$

Mit Hilfe der Konditionszahl erhalten wir das gleiche Ergebnis:

$$\varepsilon_y = \kappa_{\text{rel}} \varepsilon_x = \frac{1.00015}{0.00015} \cdot 4.99925 \cdot 10^{-5} = 0.33333.$$

1.4 Hilfe gegen rundungsbedingte Rechenfehler

Hilfe gegen rundungsbedingte Rechenfehler ist nur begrenzt möglich, aber nicht zu vernachlässigen. Man kann zunächst eine

Fehleranalyse

durchführen. Diese kann man direkt durchführen oder aber mit Hilfe der Konditionszahlen. Zur Illustration führen wir eine direkte Fehleranalyse der Grundrechenarten durch (um zu wissen, welche Operationen wie gefährlich sind).

Es bezeichne $\tilde{x}$ eine Näherung von x, Δx den absoluten Fehler und ε_x den relativen Fehler, d.h.

$$\varepsilon_x \;=\; \frac{\Delta x}{x} = \frac{\tilde{x} - x}{x} \qquad \text{oder} \qquad \tilde{x} \;=\; x\,(1 + \varepsilon_x)\ .$$

Addition:

Wir berechnen den relativen Fehler ε_{x+y} der Summe $x + y$.

$$\varepsilon_{x+y} \quad = \quad \frac{\tilde{x} + \tilde{y} - (x + y)}{(x + y)}.$$

Durch Ausrechnen folgt

$$\begin{aligned} \varepsilon_{x+y} \;&=\; \frac{\tilde{x} - x}{x + y} \quad + \frac{\tilde{y} - y}{x + y} \\ &=\; \frac{x}{x + y}\,\varepsilon_x + \frac{y}{x + y}\,\varepsilon_y\ . \end{aligned}$$

FAZIT: Bei der Addition von Zahlen gleichen Vorzeichens addiert sich der relative Fehler der Eingangsdaten höchstens, da $\frac{x}{x+y},\ \frac{y}{x+y} \;\leq\; 1$.

Subtraktion:
Man ersetze in obiger Rechnung y durch $-y$ (also $x > 0,\ -y > 0$), dann erhält man

$$\varepsilon_{x-y} \quad \approx \quad \frac{x}{x-y}\,\varepsilon_x \;-\; \frac{y}{x-y}\,\varepsilon_y\,.$$

Hier kann ein Unglück passieren, wenn x und y fast gleich groß sind (vgl. das Beispiel a)).
Ist zum Beispiel y eine Maschinenzahl, also $\varepsilon_y = 0,\ x - y \approx 10^{-12},\ x \approx 5$, so folgt

$$\varepsilon_{x-y} \quad \approx \quad 5 \cdot 10^{12}\,\varepsilon_x\,.$$

Dieses Phänomen bezeichnet man als „*Auslöschung*“ (richtiger Ziffern).

Multiplikation:

$$\varepsilon_{x\cdot y} \;\approx\; \frac{\tilde{x}\,\tilde{y} - x\cdot y}{x\cdot y} \qquad = \quad \frac{x(1+\varepsilon_x)\cdot y(1+\varepsilon_y) - xy}{xy}$$

$$= \quad \frac{xy\,\varepsilon_x + xy\,\varepsilon_y + xy\,\varepsilon_x\,\varepsilon_y}{xy} \quad = \quad \varepsilon_x + \varepsilon_y + \varepsilon_x\,\varepsilon_y \;\approx\; \varepsilon_x + \varepsilon_y$$

also $\varepsilon_{xy} \;\approx\; \varepsilon_x \,+\, \varepsilon_y\,,$ also „relativ ungefährlich“ .

Division: Analog zu oben erhält man

$$\varepsilon_{x/y} \quad = \quad \varepsilon_x \;-\; \varepsilon_y\,.$$

Gefährlich ist also vor allem die Subtraktion annähernd gleich großer Zahlen. Wir werden in den Übungen Beispiele für ihre Auswirkung und ihre Vermeidung kennenlernen.

Natürlich sollte man Fehleranalysen wie oben für ganze mathematische Verfahren durchführen. Dies ist sehr aufwendig und im Rahmen dieser Veranstaltung nicht möglich (vgl. dazu etwa: Wilkinson: Rundungsfehler).

Wir müssen uns hier mit Hinweisen bzgl. der einzelnen zu behandelnden Verfahren zufrieden geben.

Allgemein kann man nur empfehlen:

- Vermeide Rechenoperationen, die Fehler verstärken (z.B. Auslöschung).
- Vermeide Verfahren, die eine zu genaue Angabe von Eingabedaten oder Zwischenergebnissen (vgl. etwa Beispiel b)) verlangen.
- Vermeide Eingabewerte, für die das Problem sich kaum von einem unlösbaren Problem unterscheidet, oder die in der Nähe von Werten liegen, für das sich das Verfahren unvorhersehbar verhält.
- Vermeide rundungsfehleranfällige Verfahren (Beispiele werden wir kennenlernen).

2 Lineare Gleichungssysteme

Wir erinnern an das Leontief Modell aus Abschnitt 0.1. Dort wird folgendes Gleichungssystem hergeleitet,

$$\begin{aligned} 0.7\,x_L - 0.2\,x_I - 0.1\,x_T &= 20, \\ -0.2\,x_L + 0.6\,x_I - 0.1\,x_T &= 40. \\ -0.1\,x_L - 0.2\,x_I + 0.9\,x_T &= 0, \end{aligned}$$

dessen Lösung wir etwa mit Hilfe von MATLAB berechnen können ($q = (E - A)$ *backslash* y, vergl. Abschnitt 0.1). Wir erhalten

$$\begin{pmatrix} x_L \\ x_I \\ x_T \end{pmatrix} = \begin{pmatrix} 58.2278 \\ 90.5063 \\ 26.5823 \end{pmatrix}.$$

Was aber steckt hinter der Befehlsequenz $q = (E - A)$ *backslash* y?

Diese Aufgabenstellung verlangt also — etwas allgemeiner formuliert — das Lösen eines Systems linearer Gleichungen der Form

$$\boldsymbol{A}\,\boldsymbol{x} \quad = \quad \boldsymbol{b}. \tag{6}$$

Dabei bezeichnen

$$\boldsymbol{A} \;:=\; \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ \vdots & \ddots & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}, \qquad \boldsymbol{b} \;:=\; \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}, \qquad \boldsymbol{x} \;:=\; \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

die Systemmatrix, den Vektor der rechten Seite, bzw. die gesuchte Lösung. $\boldsymbol{A}$ ist eine $(n \times n)$–Matrix (n Zeilen, n Spalten), die $(n \times 1)$–Matrizen $\boldsymbol{b}$ und $\boldsymbol{x}$ heißen (Spalten)–Vektoren. Wir bezeichnen weiter

$$\boldsymbol{A} \;\in\; \mathbb{R}^{n\times n}\,(= M(n \times m; \mathbb{R}), \qquad \boldsymbol{b}, \boldsymbol{x} \;\in\; \mathbb{R}^{n\times 1} \;=\; \mathbb{R}^n$$

Diese Bezeichnung deutet an, dass die Einträge der Matrizen reelle Zahlen sind. Natürlich ist auch z.B. $A \in \mathbb{C}^{n\times n}$ möglich. Der 1. obere Index bezeichnet immer die Zeilenzahl, der 2.te die Spaltenzahl.

Der Ausdruck $\boldsymbol{A}\,\boldsymbol{x}$ bezeichnet das Matrix-Vektor Produkt, falls $A \in \mathbb{R}^{n\times n}$ und $\boldsymbol{x} \in \mathbb{R}^n$. Das Bildungsgesetz dieses Produktes kennen Sie aus der Linearen Algebra (vgl. dazu auch Fischer, §2.4).

2.1 Das Gaußsche Eliminationsverfahren

Es macht keine Schwierigkeiten, ein lineares Gleichungssystem zu lösen, wenn es folgende Gestalt hat.

$$\begin{array}{ccccccccc} a_{11}\,x_1 & + & a_{12}\,x_2 & + & \dots & + & a_{1n}\,x_n & = & b_1 \\ & & a_{22}\,x_2 & + & \dots & + & a_{2n}\,x_n & = & b_2 \\ & & \ddots & & & & \vdots & & \\ & & & & \ddots & & \vdots & & \\ & & & & & & a_{nn}\,x_n & = & b_n \end{array} \tag{7}$$

Man sagt dann, die Matrix $\boldsymbol{A} = (a_{ij})_{i,j=1,\dots,n}$ hat *obere Dreiecksgestalt*: d.h. $a_{ij} = 0$ für alle $i > j$.

$$\boldsymbol{A} \quad := \quad \begin{pmatrix} a_{11} & a_{12} & a_{13} & & \dots & a_{1n} \\ 0 & a_{22} & a_{23} & & \dots & a_{2n} \\ 0 & 0 & \ddots & & & \vdots \\ \vdots & & \ddots & a_{ii} & & \vdots \\ & & & \ddots & \ddots & \\ 0 & \dots & \dots & \dots & 0 & a_{nn} \end{pmatrix}, \qquad \text{obere Dreiecksmatrix}\,. \tag{8}$$

Wenn alle *Diagonalelemente* $a_{ii} \neq 0,\ \ i = 1, \dots, n$ sind, erhält man die eindeutige Lösung von (7) durch **Rückwärtseinsetzen**:

$$\begin{array}{lcl} x_n & = & b_n/a_{nn} \\ x_{n-1} & = & (b_{n-1} \quad - \quad a_{n-1,n}\,x_n)/a_{n-1,n-1} \\ \vdots & & \vdots \qquad\qquad\quad \vdots \\ x_2 & = & (b_2 \quad - \quad a_{2,n}\,x_n \ - \ \dots \ - \ a_{2,3}\,x_3)/a_{22} \\ x_1 & = & (b_1 \quad - \quad a_{1,n}\,x_n \ - \ \dots \ - \ a_{1,3}\,x_3 - a_{1,2}\,x_2)/a_{11} \end{array} \tag{9}$$

oder in der Summenschreibweise

$$x_i \ = \ \left(b_i \ - \ \sum_{j=i+1}^{n} a_{ij}\,x_j \right) \Big/ \, a_{ii}\,, \qquad i = n,\ \ n-1, \dots, 1\,. \tag{10}$$

$$\left(\text{Beachte, dass } \sum_{j=h}^{\ell} \dots \ = \ 0, \ \text{ falls } \ \ell < h\,. \right)$$

Erfreulicherweise lassen sich alle eindeutig lösbaren linearen Gleichungssysteme auf die Form (7) bringen. Dies gelingt mit Hilfe der folgenden elementaren Umformungen, welche die Lösungsmenge des Systems, das sind alle Vektoren $\boldsymbol{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$, die dem System (1) für $m = n$ genügen, nicht ändern:

1) Man kann zu einer Gleichung (d.h. einer Zeile von $\boldsymbol{A}$ und der zugehörigen Komponente von $\boldsymbol{b}$) ein Vielfaches einer anderen Gleichung addieren.

2) Die Lösungsmenge ändert sich nicht, wenn man die Reihenfolge der Gleichungen vertauscht.

Wir benutzen diese beiden Eigenschaften, um das Gleichungssystem $\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{b}$ so umzuformen, dass alle Elemente a_{ij} unter der Hauptdiagonalen (also $i > j$) „zu Null gemacht“ (eliminiert) werden.

Wir bezeichnen das System in der Ausgangsform mit der vollbesetzten Matrix $\boldsymbol{A}$ mit

$$\boldsymbol{A}^{(0)}\,\boldsymbol{x} \quad = \quad \boldsymbol{b}^{(0)}\,.$$

Im 1. Schritt subtrahieren wir für $i = 2, \ldots, n$ von der i–ten Gleichung das $a_{i1}^{(0)} \,/\, a_{11}^{(0)}$–fache der 1. Gleichung und nennen das dadurch entstehende Gleichungssystem

$$\boldsymbol{A}^{(1)}\,\boldsymbol{x} \quad = \quad \boldsymbol{b}^{(1)}\,.$$

Das Bildungsgesetz lautet also für $i = 2, \ldots, n$:

$$\begin{aligned} \ell_{i1} \quad &:= \quad a_{i1}^{(0)} \,/\, a_{11}^{(0)}\,; \\ a_{ij}^{(1)} \quad &:= \quad a_{ij}^{(0)} \;-\; \ell_{i1} \cdot a_{1j}^{(0)}\,, \quad j = 1, \ldots, n \\ b_i^{(1)} \quad &:= \quad b_i^{(0)} \;-\; \ell_{i1} \cdot b_1^{(0)}\,, \end{aligned} \tag{11}$$

die 1. Zeile bleibt ungeändert.

Bemerkung:
Die Elemente $a_{i1}^{(1)}$ für $i > 1$ werden, falls sie später noch gebraucht werden sollten, nicht programmiert (gerechnet), sondern gesetzt, $a_{i1}^{(1)} = 0$, zur Vermeidung von unnötigen Rundungsfehlern.

Nach dem 1. Schritt hat das System also die Gestalt $\boldsymbol{A}^{(1)}\,\boldsymbol{x} = \boldsymbol{b}^{(1)}$ mit

$$\boldsymbol{A}^{(1)} \;=\; \begin{pmatrix} a_{11}^{(0)} & a_{12}^{(0)} & \ldots & a_{1n}^{(0)} \\ 0 & a_{22}^{(1)} & \ldots & a_{2n}^{(1)} \\ \vdots & a_{32}^{(1)} & \ldots & a_{3n}^{(1)} \\ \vdots & \vdots & & \\ 0 & a_{n2}^{(1)} & & a_{nn}^{(1)} \end{pmatrix}, \qquad \boldsymbol{b}^{(1)} \;=\; \begin{pmatrix} b_1^{(0)} \\ b_2^{(1)} \\ \vdots \\ \vdots \\ b_n^{(1)} \end{pmatrix}, \tag{12}$$

Im 2. Schritt subtrahieren wir für $i = 3, \ldots, n$ von der (i)ten Gleichung das $a_{i2}^{(1)} \,/\, a_{22}^{(1)}$ fache der 2. Gleichung, d.h. wir annullieren nun alle Elemente unter $a_{22}^{(1)}$. Das entstehende System nennen wir

$$\boldsymbol{A}^{(2)}\,\boldsymbol{x} \quad = \quad \boldsymbol{b}^{(2)}\,.$$

Bildungsgesetz: Für $i = 3, \ldots, n$ sei

$$\begin{aligned} \ell_{i2} &:= a_{i2}^{(1)} \,/\, a_{22}^{(1)} \,, \\ a_{ij}^{(2)} &:= a_{ij}^{(1)} \,-\, \ell_{i2}\, a_{2j}^{(1)} \,, \quad j = 3, \ldots, n \\ a_{i2}^{(2)} &:= 0, \\ b_i^{(2)} &:= b_i^{(1)} \,-\, \ell_{i2}\, b_2^{(1)} \,, \end{aligned} \tag{13}$$

die ersten beiden Zeilen bleiben ungeändert.

Dann hat $\boldsymbol{A}^{(2)}$ die Gestalt

$$\boldsymbol{A}^{(2)} \quad = \quad \begin{pmatrix} a_{11}^{(0)} & a_{12}^{(0)} & \ldots & \ldots & a_{1n}^{(0)} \\ 0 & a_{22}^{(1)} & \ldots & \ldots & a_{2n}^{(1)} \\ \vdots & 0 & a_{33}^{(2)} & \ldots & a_{3n}^{(2)} \\ \vdots & \vdots & a_{43}^{(2)} & & \vdots \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & a_{n3}^{(2)} & \ldots & a_{nn}^{(2)} \end{pmatrix} \tag{14}$$

Dieses Verfahren führt man fort. Im k–ten Schritt subtrahiert man für $i = k+1, \ldots, n$ von der i–ten Gleichung das $a_{ik}^{(k-1)} \,/\, a_{kk}^{(k-1)}$–fache der k–ten Gleichung, um die Elemente unter $a_{kk}^{(k-1)}$ zu annullieren, wir haben also das

Bildungsgesetz (k-ter Schritt):

Für $i = k+1, \ldots, n$ sei

$$\begin{aligned} \ell_{ik} &:= a_{ik}^{(k-1)} \,/\, a_{kk}^{(k-1)} \,, \\ a_{ij}^{(k)} &:= a_{ij}^{(k-1)} \,-\, \ell_{ik}\, a_{kj}^{(k-1)} \,, \quad j = k+1, \ldots, n \\ a_{ik}^{(k)} &:= 0, \\ b_i^{(k)} &:= b_i^{(k-1)} \,-\, \ell_{ik}\, b_k^{(k-1)} \,, \end{aligned} \tag{15}$$

die Zeilen $1, \ldots, k$ bleiben unverändert.

Nach $(n-1)$ Schritten erhalten wir schließlich $\boldsymbol{A}^{(n-1)}\, \boldsymbol{x} = \boldsymbol{b}^{(n-1)}$ mit der oberen Dreiecks-

matrix

$$\boldsymbol{A}^{(n-1)} \quad = \quad \begin{pmatrix} a_{11}^{(0)} & a_{12}^{(0)} & a_{13}^{(0)} & \dots & a_{1n}^{(0)} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & \dots & a_{2n}^{(1)} \\ \vdots & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & 0 & a_{nn}^{(n-1)} \end{pmatrix} \tag{16}$$

Das System $\boldsymbol{A}^{(n-1)}\,\boldsymbol{x} \;=\; \boldsymbol{b}^{(n-1)}$ hat die gleiche Lösungsmenge wie das Ausgangssystem. $\boldsymbol{A}^{(n-1)}\,\boldsymbol{x} = \boldsymbol{b}^{(n-1)}$ kann durch Rückwärtseinsetzen gelöst werden.

Mit Hilfe des Matrixkalküls kann man den k–ten Schritt des GEV wie folgt beschreiben:

$$\boldsymbol{A}^{(k)} \quad = \quad \boldsymbol{L}_{k-1}\,\boldsymbol{A}^{(k-1)} \tag{17}$$

mit der *Frobenius Matrix*

$$\boldsymbol{L}_{k-1} \quad = \quad \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & -\ell_{k+1,k} & \ddots & \\ & & \vdots & & \ddots \\ & & -\ell_{n,k} & & 1 \end{pmatrix} \tag{18}$$

mit ℓ_{ik} gemäß (15), also

$$\ell_{ik} \;=\; a_{ik}^{(k-1)} \,/\, a_{kk}^{(k-1)} \quad \text{für} \quad i = k+1, \dots, n$$

Man erhält somit insgesamt

$$\boldsymbol{A}^{(n-1)} \quad = \quad \boldsymbol{L}_{n-2}\,\boldsymbol{A}^{(n-2)} \quad = \quad \boldsymbol{L}_{n-2}\,\boldsymbol{L}_{n-3}\,\dots\,\boldsymbol{L}_0\,\boldsymbol{A}^{(0)}\,. \tag{19}$$

Es ist einfach nachzurechnen, dass das Produkt $\Lambda_{k-1}\,L_{k-1} = I$ (Einheitsmatrix) ist mit

$$\boldsymbol{\Lambda}_{k-1} \quad = \quad \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & \ell_{k+1,k} & \ddots & \\ & & \vdots & & \ddots \\ & & \ell_{n,k} & & 1 \end{pmatrix} \tag{20}$$

Etwas mühsamer ist es festzustellen, dass sich das Produkt

$$\boldsymbol{L} \;=\; \boldsymbol{\Lambda}_0\,\boldsymbol{\Lambda}_1\,\dots\,\boldsymbol{\Lambda}_{n-2} \;=\; \begin{pmatrix} 1 & & & & \\ \ell_{21} & 1 & & & \\ \vdots & \ell_{32} & \ddots & & \\ \ell_{n-1,1} & \vdots & \ddots & 1 & \\ \ell_{n,1} & \ell_{n,2} & \dots & \ell_{n,n-1} & 1 \end{pmatrix} \tag{21}$$

in der angegebenen einfachen Form als linke Dreiecksmatrix ausrechnen lässt.

Multiplizieren wir die Gleichung (19) der Reihe nach von links mit $\boldsymbol{\Lambda}_{n-2}, \boldsymbol{\Lambda}_{n-3}, \ldots, \boldsymbol{\Lambda}_0$, so erhalten wir wegen $\boldsymbol{\Lambda}_k \boldsymbol{L}_k = I$ und (21)

$$\boldsymbol{L}\, \boldsymbol{A}^{(n-1)} \quad = \quad \boldsymbol{A}^{(0)} \quad = \quad \boldsymbol{A}\,. \tag{22}$$

Die schließlich ausgerechnete Matrix $\boldsymbol{R} = \boldsymbol{A}^{(n-1)}$ ist eine (rechte) obere Dreiecksmatrix (vgl. (16)). D.h. wir haben mit dem GEV die Ausgangsmatrix $\boldsymbol{A}$ in ein Produkt

$$\boldsymbol{A} \quad = \quad \boldsymbol{L} \cdot \boldsymbol{R} \tag{23}$$

einer linken mit einer rechten Dreiecksmatrix zerlegt. Die Darstellung (23) heißt daher auch **LR–Zerlegung von $\boldsymbol{A}$**.

Damit lässt sich das Gaußverfahren prinzipiell in 3 Lösungsschritte aufteilen

$$\left.\begin{array}{lll} 1. & \boldsymbol{A} = \boldsymbol{L}\,\boldsymbol{R} & (\text{Zerlegung von } \boldsymbol{A}) \\ 2. & \boldsymbol{L}\,\boldsymbol{c} = \boldsymbol{b} & (\text{Vorwärtseinsetzen } \Rightarrow\ \boldsymbol{c} = \boldsymbol{L}^{-1}\boldsymbol{b} = \boldsymbol{b}^{(n-1)} = \boldsymbol{R}\,\boldsymbol{x}) \\ 3. & \boldsymbol{R}\,\boldsymbol{x} = \boldsymbol{c} & (\text{Rückwärtseinsetzen } \Rightarrow\ \boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b}) \end{array}\right\} \tag{24}$$

Bemerkungen

1. Im k–ten Schritt ändern sich die ersten k Zeilen von $\boldsymbol{A}^{(k-1)}$ und $\boldsymbol{b}^{(k-1)}$ nicht mehr.
2. Die bisher beschrieben Gauß-Elimination ist nur dann durchführbar, wenn $a_{kk}^{(k-1)} \neq 0$ gilt für $k = 1, \ldots, n$. Sollte im k-ten Schritt $a_{kk}^{(k-1)} = 0$ sein (das muß abgeprüft werden), so vertauschen wir die k–te Gleichung mit einer späteren s-ten Gleichung $(s > k)$, für welche $a_{sk}^{(k-1)} \neq 0$ gilt. Gibt es kein solches s, so sind die ersten k Spalten von $\boldsymbol{A}$ (als Vektoren aufgefaßt) linear abhängig und in der Linearen Algebra wird gezeigt, dass das Gleichungssystem dann nicht eindeutig lösbar ist.

   In Gedanken können wir die im Laufe der Elimination notwendigen Zeilenvertauschungen vorab durchführen (in der Praxis natürlich nicht). Sei $\boldsymbol{P} \in \mathbb{R}^{n\times n}$ die resultierende Permutationsmatrix. Dann haben wir am Ende die LR-Zerlegung der Matrix $\boldsymbol{P}\boldsymbol{A}$ berechnet:

   $$\boldsymbol{L}\boldsymbol{R} = \boldsymbol{P}\boldsymbol{A}.$$

   Die Matrix $\boldsymbol{P}$ erhält man, indem man die Zeilenvertauschungen in einem Vektor protokolliert, siehe später. Man muß hierbei beachten, dass bei Vertauschung zweier Zeilen von A im k-ten Schritt auch die entsprechenden Zeilen der bisher berechneten $l_{ij}, 1 \leq j < k \leq i \leq n$, vertauscht werden müssen.
3. Natürlich lassen sich mit dem GEV auch $(m \times n)$–Gleichungssysteme mit $n \geq m$ behandeln (vgl. Lineare Algebra).
4. Soll für eine quadratische Matrix $\boldsymbol{A}$ die Inverse $\boldsymbol{X}$ berechnet werden, $\boldsymbol{A}\,\boldsymbol{X} = \boldsymbol{I}$, so erhält man die Spalten $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^n$ von $\boldsymbol{X}$ durch Lösung der Systeme

   $$\boldsymbol{A}\,\boldsymbol{x}^k \ = \ \boldsymbol{e}^k, \qquad k = 1, \ldots, n, \qquad \boldsymbol{e}^k \ = \ k\text{–ter Einheitsvektor}\,.$$

   Zu ihrer Lösung wird man **nur einmal** $\boldsymbol{L}$ und $\boldsymbol{R}$ gemäß (23) berechnen und danach $\boldsymbol{L}\,\boldsymbol{c}^k = \boldsymbol{e}^k$ und $\boldsymbol{R}\,\boldsymbol{x}^k = \boldsymbol{c}^k$ lösen.

Beispiel zu 2.:

Das System $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$ ist ohne Zeilenvertauschung nicht mit dem Gauß Algorithmus lösbar.

Für invertierbare $n \times n$ Matrizen A gilt

Theorem 2.1. *Sei $A \in GL(n;\mathbb{R})$. Dann gibt es eine Permutationsmatrix $P \in GL(n;\mathbb{R})$, sowie eine obere Dreiecksmatrix $R \in GL(n;\mathbb{R})$ und eine untere Dreiecksmatrix $L \in GL(n;\mathbb{R})$ mit $L = (l_{ij})_{i,j=1}^{n-1}$, $l_{ii} = 1$ für $i = 1, \ldots, n$, so dass*

$$PA = LR$$

gültig ist.

Beweis: Wir führen Induktion über die Matrixdimension n. Für $n = 1$ ist die Aussage mit $P = L = 1$ und $R = A$ klar. Die Aussage des Satzes sei also gültig für $n-1$. Wir partitionieren $A \in GL(n;\mathbb{R})$ in der Form

$$A = \begin{pmatrix} B & a \\ b^t & c \end{pmatrix}$$

mit einer Matrix $B \in \mathbb{R}^{n-1,n-1}$, einem Skalar c und Vektoren $a, b \in \mathbb{R}^{n-1}$. Wir unterscheiden 2 Fälle:

1. B singulär, also nicht invertierbar (beachte, dass die Invertierbarkeit der Matrix A nicht jene der Matrix B induziert). Dann gilt $rgB = n-2$, denn $rgA = n$ impliziert $rg \begin{pmatrix} B \\ b^t \end{pmatrix} = n-1$. Weil $rgB = n-2$, gibt es mindestens eine Zeile $B_{i\cdot}$ von B, welche zu den restlichen Zeilen linear abhängig ist. Wir vertauschen in A die Zeile $A_{i\cdot}$ mit der Zeile $A_{n\cdot} = (b^t, c)$ und nennen das Resultat wieder A, in Matrixschreibweise mit der Permutationsmatrix P_i^n

   $$\tilde{A} = P_n^i A = \begin{pmatrix} \tilde{B} & \tilde{a} \\ \tilde{b}^t & \tilde{c} \end{pmatrix},$$

   wobei $\tilde{a}_i = c$ und $\tilde{c} = a_i$. In dieser Darstellung ist die Matrix $\tilde{B}$ aufgrund $rg \begin{pmatrix} B \\ b^t \end{pmatrix} = n-1$ regulär. Das ist der 2te Fall.

2. B regulär, also nach Induktionsvoraussetzung

   $$PB = \tilde{L}\tilde{R} \Longleftrightarrow B = P^{-1}\tilde{L}\tilde{R}$$

   mit $\tilde{L}, \tilde{R} \in \mathbb{R}^{n-1,n-1}$ untere bzw. obere Dreiecksmatrix und $P \in GL(n;\mathbb{R})$ Permutationsmatrix. Wir haben

   $$A = \begin{pmatrix} B & a \\ b^t & c \end{pmatrix} = A = \begin{pmatrix} P^{-1}\tilde{L}\tilde{R} & a \\ b^t & c \end{pmatrix} = \begin{pmatrix} P^{-1} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \tilde{L} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \tilde{R} & \tilde{a} \\ b^t & c \end{pmatrix} \quad (25)$$

mit $\tilde{a} = \tilde{L}^{-1}Pa$. Da $\tilde{R}$ eine reguläre obere Dreiecksmatrix darstellt (d.h. $\tilde{r}_{ii} \neq 0$!), besitzt die Matrix

$$\begin{pmatrix} \tilde{R} & \tilde{a} \\ b^t & c \end{pmatrix}$$

Nicht-Null Einträge nur in der letzten Zeile. Diese eliminieren wir mit Hilfe der Einträge $\tilde{t}_{ii}$ von $\tilde{R}$. Wir benutzen dazu in Analogie zu (15) Frobenius Matrizen $L_1, \ldots, L_{n-1} \in GL(n;\mathbb{R})$ und erhalten

$$\begin{pmatrix} \tilde{R} & \tilde{a} \\ 0 & \tilde{c} \end{pmatrix} = L_{n-1} \ldots L_1 \begin{pmatrix} \tilde{R} & \tilde{a} \\ b^t & c \end{pmatrix}.$$

Die Matrix L_k hat dabei in der $k-$ten Spalte unterhalb der Diagonalen höchstens einen von Null verschiedenen Eintrag $-b_k^{(k-1)}/\tilde{r}_{kk}$, wobei $b^0 := b$. In Matrixschreibweise

$$L_k = E + \begin{pmatrix} 0 \\ \vdots \\ 1 \end{pmatrix} \begin{pmatrix} 0 & \cdots - b_k^{(k-1)}/\tilde{r}_{kk} & \ldots & 0 \end{pmatrix}^t$$

Wir erhalten in (25)

$$A = \underbrace{\begin{pmatrix} P^{-1} & 0 \\ 0 & 1 \end{pmatrix}}_{=:P^{-1}} \underbrace{\begin{pmatrix} \tilde{L} & 0 \\ 0 & 1 \end{pmatrix} L_1^{-1} \ldots L_{n-1}^{-1}}_{=:L} \underbrace{\begin{pmatrix} \tilde{R} & \tilde{a} \\ 0 & \tilde{c} \end{pmatrix}}_{=:R},$$

womit der Beweis abgeschlossen ist. ■

2.1.1 Numerische Schwierigkeiten, Pivot–Suche

Die Bemerkung 2. (s.o.) ist oft nur von theoretischem Wert, da der Rechner auf Grund von Zahldarstellungsschwierigkeiten und Rundungsfehlern das Ergebnis Null einer Rechnung nur selten als Null erkennt, sondern z.B. $a_{kk}^{(k-1)} = 3.5 \cdot 10^{-25}$ erhält. Wird mit diesem Wert weiter gerechnet, so kann nur Unsinn herauskommen.

Wir zeigen dies an einem **1. Beispiel**, das wir der Einfachheit halber mit 6–stelliger Arithmetik rechnen (genauer: jeder Rechenschritt wird zunächst mit höherer Genauigkeit ausgeführt, dann wird auf sechs Dezimalstellen gerundet; dabei wird die Berechnung von $a + b \cdot c$ als ein

Rechenschritt angesehen):

$$\boldsymbol{A}^{(0)} = \begin{pmatrix} 11 & 44 & 1 \\ 0.1 & 0.4 & 3 \\ 0 & 1 & -1 \end{pmatrix}, \qquad \boldsymbol{b}^{(0)} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

1. Eliminationsschritt

$$\boldsymbol{A}^{(1)} = \begin{pmatrix} 11 & 44 & 1 \\ 0 & -4{\cdot}10^{-8} & 2.99091 \\ 0 & 1 & -1 \end{pmatrix}, \qquad \boldsymbol{b}^{(1)} = \begin{pmatrix} 1 \\ 0.990909 \\ 1 \end{pmatrix}$$

2. Eliminationsschritt

$$\boldsymbol{A}^{(2)} = \begin{pmatrix} 11 & 44 & 1 \\ 0 & -4{\cdot}10^{-8} & 2.99091 \\ 0 & 0 & 7.47727 \cdot 10^{7} \end{pmatrix}, \quad \boldsymbol{b}^{(2)} = \begin{pmatrix} 1 \\ 0.990909 \\ 2.47727 \cdot 10^{7} \end{pmatrix}$$

Lösung: $\mathbf{x}^T = (-41.8765,\ 10.4843,\ 0.331307)$

Zum Vergleich die exakte Lösung

$$\boldsymbol{x}^T = (-5.26444,\ 1.33131,\ 0.331307)$$

Grund:
Im 1. Eliminationsschritt ist $a_{22}^{(1)} = 0.4 - \frac{0.1}{11} \cdot 44$. Nun ist $\frac{0.1}{11}$ keine Maschinenzahl, weshalb der Rechner $a_{22}^{(1)} = 0$ nicht erhält. Das Ergebnis ist katastrophal.

Selbst wenn $a_{kk}^{(k-1)}$ tatsächlich $\neq 0$ ist, aber sehr klein im Vergleich zu anderen Matrixeinträgen, können sich erhebliche Schwierigkeiten ergeben, wie folgendes **2. Beispiel** zeigt (wieder mit 6–stelliger Arithmetik):

$$\boldsymbol{A}^{(0)} = \begin{pmatrix} 0.001 & 1 & 1 \\ -1 & 0.004 & 0.004 \\ -1000 & 0.004 & 0.000004 \end{pmatrix}, \quad \boldsymbol{b}^{(0)} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

1. Eliminationsschritt

$$\boldsymbol{A}^{(1)} = \begin{pmatrix} 0.001 & 1 & 1 \\ 0 & 1000 & 1000 \\ 0 & 1{\cdot}10^{6} & 1{\cdot}10^{6} \end{pmatrix}, \qquad \boldsymbol{b}^{(1)} = \begin{pmatrix} 1 \\ 1001 \\ 1{\cdot}10^{6} \end{pmatrix}.$$

2. Eliminationsschritt

$$\boldsymbol{A}^{(2)} = \begin{pmatrix} 0.001 & 1 & 1 \\ 0 & 1000 & 1000 \\ 0 & 0 & 0 \end{pmatrix}, \qquad \boldsymbol{b}^{(2)} = \begin{pmatrix} 1 \\ 1001 \\ -1000 \end{pmatrix}.$$

Gleichungssystem unlösbar.

Grund:

Das Unglück passiert schon im 1. Eliminationsschritt, nach welchem die Zeilen 2 und 3 von $\boldsymbol{A}^{(1)}$ linear abhängig sind. Der Rechner erhält

$$\begin{aligned} a_{22}^{(1)} &= a_{23}^{(1)} = 0.004 - \tfrac{(-1)}{0.001} = 0.004 + 1000 &&\approx 1000 \\ a_{32}^{(1)} &= 0.004 - \tfrac{(-1000)}{0.001} = 0.004 + 10^6 &&\approx 10^6 \\ a_{33}^{(1)} &= 0.000004 + 10^6 &&\approx 10^6 \end{aligned}$$

Er kann im Rahmen seiner Genauigkeit (6 Dezimalen) die Zahlen $10^3 + 0.004$ und 10^3 bzw. $10^6 + 0.004$ und 10^6 usw. bei der Differenz– bzw. Summenbildung nicht mehr unterscheiden. Dies liegt an der absoluten Größe der Faktoren

$$\begin{aligned} \ell_{21} &= a_{21}^{(0)} \,/\, a_{11}^{(0)} = -10^3 \\ \ell_{31} &= a_{31}^{(0)} \,/\, a_{11}^{(0)} = -10^6 \end{aligned}$$

Folge: Die Einträge der 2. und 3. Zeile „gehen unter“ in $A^{(1)}$.

Die folgenden beiden Regeln sollen helfen, das Auftreten dieser Phänomene einzuschränken.

1. einfache Pivotsuche (Spaltenpivotsuche)

Vor dem k–ten Eliminationsschritt suche man die betragsgrößte der Zahlen $a_{ik}^{(k-1)}, i \geq k$, also etwa $a_{i_0\,k}^{(k-1)}$, und vertausche dann in $\boldsymbol{A}^{(k-1)}$ die i_0–te Zeile mit der k–ten Zeile und entsprechend $b_{i_0}^{(k-1)}$ mit $b_k^{(k-1)}$. $a_{i_0\,k}^{(k-1)}$ heißt dann Pivotelement. (Man macht also durch geeignete Vertauschung der Gleichungen die Faktoren $\ell_{ik} = a_{ik}^{(k-1)} \,/\, a_{kk}^{(k-1)}$ betragsmäßig möglichst klein.)

2. totale Pivotsuche

Vor dem k–ten Eliminationsschritt suche man die betragsgrößte der Zahlen $a_{ij}^{(k-1)}, i \geq k,\ j \geq k$, diese sei $a_{rs}^{(k-1)}$. Dann vertausche man die k–te Zeile mit der r–ten Zeile — natürlich auch $b_k^{(k-1)}$ und $b_r^{(k-1)}$ — und die k–te Spalte mit der s–ten Spalte. $\left(a_{rs}^{(k-1)}\right.$ heißt Pivotelement$\left.\right)$.

Dadurch bekommen die Faktoren ℓ_{ik} betragsmäßig die kleinst möglichen Werte. Dadurch wird gesichert, dass bei der Differenzbildung (vgl. 5.9))

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

in der neuen i–ten Zeile möglichst viel Information der alten i–ten Zeile erhalten bleibt.

Die totale Pivotsuche gilt als die stabilste Variante des Gaußschen Eliminationsverfahrens (allerdings auch als die aufwendigste, da sehr viele Vergleiche nötig sind bis man das betragsgrößte Element gefunden hat). Zumindest auf die einfache Pivotsuche sollte man **keinesfalls verzichten**.

Schon mit der einfachen Pivotsuche liefern unsere Beispiele jetzt vernünftige Lösungen (wieder mit 6–stelliger Arithmetik).

Beispiel 1:
$A^{(0)}$ und $A^{(1)}$ wie auf S. 50, dann Vertauschung von Zeile 2 und 3 und 2. Eliminationsschritt

$$A^{(2)} = \begin{pmatrix} 11 & 44 & 1 \\ 0 & 1 & -1 \\ 0 & 0 & 2.99091 \end{pmatrix}, \qquad b^{(2)} = \begin{pmatrix} 1 \\ 1 \\ 0.990909 \end{pmatrix}$$

und die vernünftige Lösung

$$x^T = (-5.26444,\ 1.33131,\ 0.331037)$$

Beispiel 2:
$A^{(0)},\ b^{(0)}$ wie auf S. 51, Tausch von 1. und 3. Gleichung und 1. Eliminationsschritt liefern

$$A^{(1)} = \begin{pmatrix} -1000 & 0.004 & 0.000004 \\ 0 & 0.003996 & 0.004 \\ 0 & 1 & 1 \end{pmatrix}, \qquad b^{(1)} = \begin{pmatrix} 1 \\ 0.999 \\ 1 \end{pmatrix}$$

Tausch von 2. und 3. Gleichung und 2. Eliminationsschritt

$$A^{(2)} = \begin{pmatrix} -1000 & 0.004 & 0.000004 \\ 0 & 1 & 1 \\ 0 & 0 & 4{\cdot}10^{-6} \end{pmatrix}, \qquad b^{(2)} = \begin{pmatrix} 1 \\ 1 \\ 0.995004 \end{pmatrix}$$

Lösung: $x^T = (-0.995005,\ -2.48750 \cdot 10^5,\ 2.48751 \cdot 10^5)$.

Schließlich noch die LR Zerlegung mit partieller Pivotiserung in Form eines Algorithmus'

Algorithmus 2.2. (Gauß Algorithmus mit partieller Pivotisierung)
Do $j = 1, n$
  $i_p = \text{pivot}(a_{jj}, ..., a_{nj}) = \text{argmax}_{j \le l \le n} |a_{lj}|$
  if $(i_p == 0)$ Stop , A singulär
  vertausche Zeilen i_p und j in A und b_{i_p} und b_j
  Do $i = j + 1, n$
    $l_{ij} = a_{ij}/a_{jj}$
    $b_i = b_i - l_{ij} * b_j$
    do $k = j, n$
      $a_{ik} = a_{ik} - l_{ij} * a_{jk}$
    enddo
  enddo
enddo

2.1.2 Bemerkungen zur Programmierung der Pivotsuche

Die Vertauschung von Gleichungen (einfache Pivotsuche) hat keinen Einfluß auf die Lösungsmenge des Gleichungssystems, die Vertauschung von Spalten ebenfalls nicht, sie entspricht nur einer Änderung der Reihenfolge der Komponenten des Lösungsvektors, worüber man allerdings Buch führen muß. Wird das Verfahren programmiert, so muß man die Vertauschung von Zeilen und Spalten, die sehr aufwendig ist, nicht tatsächlich durchführen. Es ist einfacher, sich die Vertauschungen mit Hilfe von Merkvektoren zu merken.

Vor Beginn des Verfahrens definiert man für die Zeilen und Spalten von $\boldsymbol{A}$ Merkvektoren $z[i],\ \ s[j],\ \ i,j = 1,\ldots,n,$ die zunächst die natürliche Reihenfolge der Zeilen und Spalten speichern, d.h. $z[i] = s[i] = i,\ \ i = 1,\ldots,n.$

Die Matrixelemente $a_{ij},$ die rechten Seiten b_i und die Komponenten x_j des Lösungsvektors werden aufgerufen durch $a[z[i], s[j]],\ \ b[z[i]]$ und $x[s[j]].$

Soll beispielsweise die 3. und 7. Gleichung vertauscht werden, so setzt man mit einer Hilfsgröße h:

$$\begin{aligned} h &:= z[3]\,; \\ z[3] &:= z[7]\,; \\ z[7] &:= h\,; \end{aligned}$$

dann wird die neue 3. Zeile durch $a[z[3], s[k]],\ \ k = 1,\ldots,n,$ die neue rechte Seite durch $b[z[3]]$ aufgerufen. Entsprechend verfährt man beim Spaltentausch.

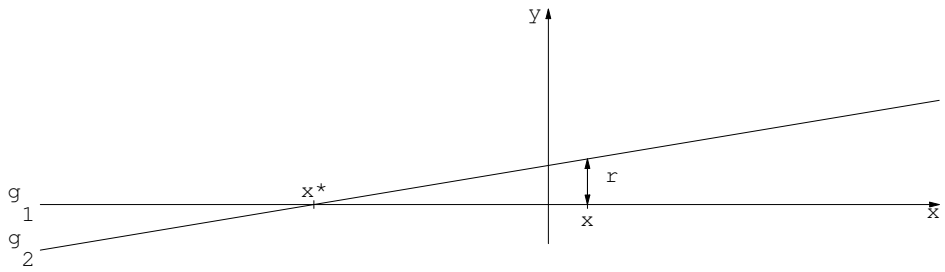
Es gibt einen Typ von Matrizen, die sog. *positiv definiten Matrizen*, bei denen man auf Zeilen– und Spaltenvertauschung verzichten kann und wobei man eine „sparsamere“ (Zahl der Rechenoperationen) Variante des GEV anwenden kann, das Cholesky–Verfahren, siehe Abschnitt 2.2.

Mit der Pivotsuche sind noch nicht alle auftretenden numerischen Probleme gelöst. Auf 2 Punkte wollen wir noch hinweisen:

1. Ein singuläres Gleichungssystem (d.h. die Zeilen von $\boldsymbol{A}$ sind linear abhängig) wird (jedenfalls theoretisch) vom GEV daran erkannt, dass alle Pivotsuchen erfolglos verlaufen, d.h. kein nicht verschwindendes Pivotelement finden. Durch Rundungsfehler könnte es aber passieren (Beispiel 1)), dass das System trotzdem als lösbar erscheint. Eine hieraus berechnete Lösung hat wenig Sinn. Man behilft sich in der Praxis überlicherweise so, dass man ein Gleichungssystem für nicht behandelbar erklärt, wenn ein Pivotelement betragsmäßig kleiner ist als ein vorgegebenes $\varepsilon > 0$ (z.B. $\varepsilon = 10^{-8}$). Dies Vorgehen erfolgt aber ausschließlich aus praktischen Gründen. Es kann durchaus vorkommen, dass man auf diese Weise ein problemlos lösbares Gleichungssystem für nicht behandelbar hält bzw., wenn man das Pivotelement und damit (bei totaler Pivotsuche) alle restlichen Elemente $= 0$ setzt, man damit numerisch linear abhängige Zeilen findet, die in Wirklichkeit gar nicht linear abhängig sind. Dies kann sich, zum Beispiel bei der späteren numerischen Behandlung von linearen Optimierungsaufgaben, als problematisch erweisen.
2. Auf Grund von Rechenungenauigkeiten wird in der Regel die berechnete Lösung $\hat{\boldsymbol{x}}$ nicht mit der exakten Lösung $\boldsymbol{x}^*$ übereinstimmen, es wird nur $\boldsymbol{A}\,\hat{\boldsymbol{x}} \approx \boldsymbol{b}$ gelten. Man muß sich deshalb fragen, ob die Größe des *Defekts* $\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{A}\,\hat{\boldsymbol{x}}$ (auch *Residuum* genannt) eine Aussage über die Größe des unbekannten Fehlers $\boldsymbol{\epsilon} = \boldsymbol{x}^* - \hat{\boldsymbol{x}}$ zuläßt.

Die in 2. angeschnittene Frage lässt sich nur bedingt mit ja beantworten. Wie eine Fehleranalyse des GEV (die wir hier und jetzt noch nicht durchführen können) zeigt, kann bei fast linear abhängigen Zeilen das Residuum sehr klein ausfallen, der Fehler $\boldsymbol{\epsilon} = \boldsymbol{x}^* - \hat{\boldsymbol{x}}$ jedoch sehr groß sein.

Man mache sich das geometrisch deutlich im Fall $n = 2$. Dann stellen die beiden Gleichungen 2 Geraden dar, ihre Lösung den Schnittpunkt der Geraden. Sind die Geraden fast linear abhängig, so ist $\boldsymbol{r}$ klein und $\boldsymbol{\epsilon}$ groß (vgl. das Beispiel: $g_1 = x$–Achse)



Wir nennen ein Gleichungssystem *schlecht konditioniert*, wenn kleine Änderungen der Eingabewerte (a_{ij}, b_i) große Änderungen der Lösung zur Folge haben. (Eine präzise Definition liegt im Augenblick noch ausserhalb unserer mathematischen Fähigkeiten.)

Man kann zeigen, dass dieses Phänomen bei linearen Fast–Abhängigkeiten auftritt
(**Aufgabe:** Man zeige dies an einem Beispiel für $n = 2$).

2.1.3 Rechenaufwand

Der Rechenaufwand der Gauß-Elimination ist kubisch in n, d.h. $O(n^3)$. Er wird dominiert durch die Anweisung

$$a_{ij}^{(k)} := a_{ij}^{(k-1)} \; - \; \ell_{ik}\, a_{kj}^{(k-1)}, \quad j = k+1, \ldots, n,$$

in (15), die für $k = 1, \ldots, n-1$ und $i = k+1, \ldots, n$ durchzuführen ist. Wegen

$$\begin{aligned}\sum_{k=1}^{n-1} \sum_{i=k+1}^{n} (n-k) = \sum_{k=1}^{n-1} (n-k)^2 = \sum_{k=1}^{n} (n-k)^2 = \sum_{k=1}^{n} (n^2 - 2kn + k^2) \\ = n^3 - 2n\frac{n(n+1)}{2} + \frac{2n^3+3n^2+n}{6} = \frac{n^3}{3} - \frac{n^2}{2} + \frac{n}{6}\end{aligned}$$

folgt

$$\text{Rechenaufwand(GEV)} = \frac{n^3}{3}\text{Add.} + \frac{n^3}{3}\text{Mult.} + O(n^2)\ \text{FLOPs.}$$

2.1.4 Variable rechte Seiten

Gelegentlich muß man dasselbe Gleichungssystem mit verschiedenen rechten Seiten lösen. Z.B. wollen wir für unser Modell der Volkswirtschaft berechnen, welche Mengen die Sektoren jeweils produzieren müssen, um verschiedene auswärtige Nachfragen befriedigen zu können.

Man wird dann nicht jedesmal das Gleichungssystem von neuem lösen. Man kann diesen Arbeitsaufwand reduzieren:

- Für die Matrix $\boldsymbol{A}$ wird einmal das Eliminationsverfahren vollständig durchgeführt.
- Im Rahmen einer Laufanweisung, die sich über alle rechten Seiten erstreckt, werden diese gemäß (15), letzte Zeile, umgeformt und danach wird, innerhalb derselben Laufanweisung, das Rückwärtseinsetzen für jede Seite durchgeführt. (Natürlich müssen dabei etwaige Zeilen– und Spaltenvertauschungen berücksichtigt werden.)

Formuliert man das im Matrixkalkül, so bedeutet das, dass einmal der 1. Schritt aus (24) ausgeführt wird und danach für jede rechte Seite die Schritte 2 und 3. Etwaige Zeilen– und Spaltenvertauschungen müssen durch Permutationsmatrizen berücksichtigt werden (vgl. Literatur).

Bemerkungen:

a) Variable rechte Seiten treten auch auf, wenn man die Inverse einer Matrix berechnen will.

b) Die Berechnung von Determinanten (zumindest für $n > 5$) wird mit Hilfe des GEV ausgeführt. Beachte dazu, daß die Determinanten der Umformungsmatrizen alle =1 sind. Die Determinante von **A** ist dann gleich dem Produkt der Diagonalelemente von $\boldsymbol{R}$.

c) Das GEV ist nicht das einzige Verfahren zur Lösung von Gleichungssystemen. Man kann dies (insbesondere bei großen und dünn besetzten Matrizen) auch mit Hilfe iterativer Verfahren tun. Wir kommen darauf zu einem späteren Zeitpunkt zurück.

2.2 Die Cholesky-Zerlegung

Ist die Koeffizientenmatrix $A \in \mathbb{R}^{n\times n}$ symmetrisch ($A = A^T$) und positiv definit ($v^T Av > 0$ für alle $v \in \mathbb{R}^n \setminus \{0\}$), so ist es effizienter, anstelle des Gaußschen Eliminationsverfahrens die *Cholesky-Zerlegung* zu verwenden. Eine in der Anwendung wichtige Klasse von Matrizen bilden die positiv definiten Matrizen.

Definition 2.3. Eine (komplexe) Matrix A heißt positiv definit:$\Longleftrightarrow$

a.) $A = A^H$, d.m. A ist hermitesche Matrix,

b.) $x^H Ax > 0$ für alle $x \in \mathbb{C}^n, x \neq 0$.

Folgende Eigenschaften positiv definiter Matrizen sind aus der Linearen Algebra bekannt.

Hilfsatz 2.4.

- A^{-1} existiert und ist positiv definit.
- Alle Hauptuntermatrizen von A sind positiv definit.
- Alle Hauptminoren von A sind positiv.

Für diese Klasse von Matrizen ist es effizienter, anstelle des Gaußschen Eliminationsverfahrens die *Cholesky-Zerlegung* zu verwenden.

Es gilt der

Satz 2.5. *Cholesky Zerlegung*
Sei $A \in \mathbb{C}^{n\times n}$ positiv definit. Dann gibt es genau eine untere Dreiecksmatrix L mit $l_{ik} = 0$ $(k > i)$, $l_{ii} > 0$ $(i = 1, \ldots, n)$ und

$$A = LL^H.$$

Ist A reell, so auch L.

Beweis: Mittels Induktion nach n.

Anfang $n = 1$: $A = a_{11} = l_{11}\bar{l}_{11}$ mit $l_{11} := \sqrt{a_{11}}$. Das ist möglich, da $a_{11} > 0$.

Annahme: $A_n = L_n L_n^H$, $l_{ik} = 0$ $(k > i)$, $l_{ii} > 0$ $(i = 1, \ldots, n)$.

Schritt $n \to n + 1$: Wir partitionieren $A \in \mathbb{C}^{n+1,n+1}$

$$A = \begin{pmatrix} A_n & b \\ b^H & a_{n+1n+1} \end{pmatrix},$$

mit $b \in \mathbb{C}^n$ und A_n positiv definit gemäß Hilfsatz 2.4. Also erfüllt A_n die Induktionsannahme und es gilt $A_n = L_n L_n^H$. Wir machen für die gesuchte Matrix L den Ansatz

$$L = \begin{pmatrix} L_n & 0 \\ c^H & \alpha \end{pmatrix}$$

und bestimmen c und α aus der Beziehung

$$\begin{pmatrix} L_n & 0 \\ c^H & \alpha \end{pmatrix} \begin{pmatrix} L_n^H & c \\ 0 & \alpha \end{pmatrix} = \begin{pmatrix} A_n & b \\ b^H & a_{n+1n+1} \end{pmatrix} \tag{26}$$

Es ergeben sich die Bedingungen

$$\begin{aligned} L_n c &= b \\ c^H c + \alpha^2 &= a_{n+1n+1}. \end{aligned}$$

Die erste Gleichung liefert den Vektor c, da die Matrix L_n regulär ist. Ferner gilt $\alpha^2 = a_{n+1n+1} - c^H c$ und $\det(L_n) > 0$. Aus (26) ergibt sich

$$\det(A) = |\det(L_n)|^2 \alpha^2,$$

also $\alpha^2 > 0$, da $\det(A) > 0$. Daher gibt es genau ein positives α in (26), nämlich $\alpha = \sqrt{a_{n+1n+1} - c^H c}$ und der Beweis fertig. ■

Die Cholesky Zerlegung einer positiv definiten Matrix kann natürlich auch algorithmisch realisiert werden. Direkt aus dem Beweis von Satz 2.5 ergibt sich

Algorithmus 2.6. Cholesky Zerlegung einer Matrix $A \in \mathbb{C}^{n\times n}$

1. $i = 1$ und $j = 1$.
2. $i > j$:

$$l_{ij} = \frac{a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk}}{l_{jj}}.$$

$i = j$:

$$l_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2}.$$

Ist $l_{ii} \leq 0$ oder l_{ii} nicht reell: Matrix nicht positiv definit, STOP.

$i < j$:

$$l_{ij} = 0.$$

3. $i \leq n-1$: $i := i+1$ und gehe zu 2.

   $j \leq n-1$: $j := j+1$, $i := 1$ und gehe zu 2.

   STOP.

Hinter diesem Algorithmus steckt einfaches Ausmultiplizieren von LL^T und eine Koeffizientenvergleich. Wir können spalten- und zeilenweise vorgehen;

Berechnung der Cholesky-Zerlegung (spaltenweise):

Für $k = 1, \ldots, n$:

$$l_{kk} := \sqrt{a_{kk} - \sum_{i=1}^{k-1} l_{ki}^2}$$

Für $j = k+1, \ldots, n$: $\quad l_{jk} := \dfrac{a_{jk} - \sum_{i=1}^{k-1} l_{ji} l_{ki}}{l_{kk}}$

Man kann die Matrix L auch zeilenweise berechnen:

Berechnung der Cholesky-Zerlegung (zeilenweise):

$l_{11} := \sqrt{a_{11}}$

Für $j = 2, \ldots, n$:

Für $k = 1, \ldots, j-1$: $\quad l_{jk} := \dfrac{a_{jk} - \sum_{i=1}^{k-1} l_{ji} l_{ki}}{l_{kk}}$

$$l_{jj} := \sqrt{a_{jj} - \sum_{i=1}^{j-1} l_{ji}^2}$$

In beiden Fällen ist der Rechenaufwand kubisch in n, also gleich $O(n^3)$. Genauer gilt

$$\sum_{k=1}^{n} \sum_{j=k+1}^{n} k = \sum_{k=1}^{n} k(n-k) = n \sum_{k=1}^{n} k - \sum_{k=1}^{n} k^2 = \frac{n^2(n+1)}{2} - \frac{2n^3 + 3n^2 + n}{6} = \frac{n^3 - n}{6}$$

und daher

$$\text{Rechenaufwand(Cholesky)} = \frac{n^3}{6}\text{Add.} + \frac{n^3}{6}\text{Mult.} + O(n^2)\ \text{FLOPs}.$$

Die Cholesky-Zerlegung ist also nur halb so aufwändig wie die LR-Zerlegung und daher bei symmetrisch positiv definiten Matrizen vorzuziehen.

2.2.1 Normen und Fehlerabschätzungen

Um messen zu können benötigen wir Normen.

Definition 2.7. Eine Abbildung $\|\cdot\| : \mathbb{C}^n \to \mathbb{R}$ heißt Norm, falls

a.) $\|x\| > 0$ für alle $x \in \mathbb{C}^n$, $x \neq 0$ (Definitheit)

b.) $\|\alpha y\| = |\alpha|\|x\|$ für alle $\alpha \in \mathbb{C}$, $x \in \mathbb{C}^n$ (Homogenität)

c.) $\|x + y\| \leq \|x\| + \|y\|$ für alle $x, y \in \mathbb{C}^n$ (Dreiecksungleichung)

Eigenschaften von Normen sind zusammengefaßt in

Aufgabe 2.8.

- Normen erfüllen die umgekehrte Dreiecksungleichung

$$|\|x\| - \|y\|| \leq \|x - y\| \text{ für alle } x, y \in \mathbb{R}^n(\mathbb{C}^n). \tag{27}$$

- Normen sind gleichmäßig stetig bzgl. der Metrik $\rho(x, y) := \max_i |x_i - y_i|$ des $\mathbb{R}^n(\mathbb{C}^n)$ (was ist denn eine Metrik?)

- Alle Normen auf dem $\mathbb{R}^n(\mathbb{C}^n)$ sind äquivalent, d.m. für jedes Paar $\|\cdot\|_a, \|\cdot\|_b$ von Normen gibt es Konstanten c, C derart, daß

$$c\|x\|_a \leq \|x\|_b \leq C\|x\|_a \text{ für alle } x, y \in \mathbb{R}^n(\mathbb{C}^n).$$

All' das sollen Sie nachweisen.

Für Matrizen $A \in M(m, n)$ (im Folgenden Matrizen mit m Zeilen, n Spalten und Einträgen aus $\mathbb{R}(\mathbb{C})$) werden Normen analog zu Definition 2.7 eingeführt (lediglich 'für alle $x \in \mathbb{C}^n$' durch 'für alle $A \in M(m, n)$' ersetzen. Gewöhnungsbedürftig ist

Definition 2.9.

- Eine Matrixnorm $\|\cdot\|$ heißt mit den Vektornormen $\|\cdot\|_a$ auf dem $\mathbb{C}^n$ und $\|\cdot\|_b$ auf dem $\mathbb{C}^m$, **verträglich (passend)**, falls

$$\|Ax\|_b \leq \|A\|\|x\|_a \text{ für alle } x \in \mathbb{C}^n, A \in M(m, n)$$

  gültig ist.

- Eine Matrixnorm $\|\cdot\|$ für quadratische Matrizen heißt **submultiplikativ**, falls

$$\|AB\| \leq \|A\|\|B\| \text{ für alle } A, B \in M(n, n).$$

- Sei $\|\cdot\|$ eine Vektornorm.

$$\text{lub}(A) := \max_{x \neq 0} \frac{\|Ax\|}{\|x\|}$$

  heißt **Grenznorm**. Dabei kommt 'lub' von least upper bound, was soviel heißt wie Kleinste obere Schranke.

- Für invertierbare Matrizen $A \in M(n,n)$ heißt $\kappa(A) := \|A\|\|A^{-1}\|$ Konditionszahl von A (bzgl. der Matrixnorm $\|\cdot\|$).

Aus der Definition 2.9 wird klar, daß die Grenznorm von der verwendeten Vektornorm abhängt, ebenso die Konditionszahl. Wir bezeichnen im Folgenden lub_2 bei Verwendung der Euklidischen, lub_∞ bei Verwendung der Maximum Norm in der Definition der Grenznorm, κ_2 und κ_∞ entsprechend.

Beispiel. Matrixnormen sind etwa

- $\|A\|_\infty := \text{lub}_\infty(A) = \max_{x\neq 0} \frac{\|Ax\|_\infty}{\|x\|_\infty} = \max_i \sum_{k=1}^{n} |a_{ik}|$ (Zeilensummen Norm)
- $\|A\|_1 := \text{lub}_1(A) = \max_{x\neq 0} \frac{\|Ax\|_1}{\|x\|_1} = \max_j \sum_{i=1}^{m} |a_{ij}|$ (Spaltensummen Norm)
- $\|A\|_G = \max_{i,j} |a_{ij}|$ (Gesamt Norm)
- $\|A\|_E = \left(\text{spur}(A^H A)\right)^{\frac{1}{2}}$ (Euklidische oder Schur Norm)
- $\|A\|_2 := \text{lub}_2(A) = \max_{x\neq 0} \frac{\|Ax\|_2}{\|x\|_2} = \max\{\sqrt{\lambda}; \lambda \in \mathbb{R}, A^H Ax = \lambda x \text{ für ein } x \in \mathbb{C}^n\}$ (Spektral oder Hilbert Norm).

Sind $\|\cdot\|_a$ auf dem $\mathbb{C}^n$, $\|\cdot\|_b$ auf dem $\mathbb{C}^m$ Vektornormen und ist $A \in M(m,n)$, so ist die Matrixnorm

$$\|A\| := \max_{x\neq 0} \frac{\|Ax\|_b}{\|x\|_a} \text{ Operator Norm}$$

mit $\|\cdot\|_a$ und $\|\cdot\|_b$ verträglich (passend) (Nachweis!, auch der Norm Eigenschaften).

Aufgabe 2.10. Weisen Sie nach, daß

- für jede mit der Vektornorm $\|\cdot\|_v$ verträgliche Matrix Norm $\|\cdot\|$ die Abschätzung $\text{lub}_v(A) \leq \|A\|$ gilt,
- $\text{lub}_\infty(A)$ der Zeilensummen Norm von A entspricht,
- $\text{lub}_2(A)$ der Spektral Norm von A entspricht,
- zu Vektornormen $\|\cdot\|_a$ auf dem $\mathbb{C}^n$, $\|\cdot\|_b$ auf dem $\mathbb{C}^m$ durch

$$\|A\| := \max_{x\neq 0} \frac{\|Ax\|_b}{\|x\|_a}$$

  eine Matrixnorm auf $M(m,n)$ definiert wird, welche mit $\|\cdot\|_a$ und $\|\cdot\|_b$ verträglich ist.

Jetzt zu Fehlerabschätzungen. Wir wollen untersuchen, wie numerische Fehler bei der Lösung von linearen Gleichungssystemen schlimmstenfalls verstärkt werden. Dazu betrachten wir zu $Ax = b$ mit invertierbarer Koeffizientenmatrix $A \in M(n,n)$ das gestörte Gls

$$(A + \delta A)(x + \delta x) = b + \delta b. \tag{28}$$

Ziel soll nun sein, in einer gegebenen Vektornorm $\|\cdot\|$ die Fehler

$$\|\delta x\| \text{ (absoluter Fehler) und } \frac{\|\delta x\|}{\|x\|} \text{ (relativer Fehler)}$$

in den Störungen δA und δb abzuschätzen. (28) bildet die Praxis ab, denn auf einem Computer wird nicht die exakte Lösung x von $Ax = b$, sondern eine Näherungslösung $x^* = x + \delta x \neq x$ berechnet. Von dieser nehmen wir an, dass sie die exakte Lösung des gestörten Systems (28) darstellt. Im Folgenden seien Vektor und Matrix Normen stets so gewählt, daß sie passend und submultiplikativ sind. Es gilt der

Satz 2.11. Es gelte für die Störung δA in (28) die Abschätzung

$$\kappa(A)\frac{\|\delta A\|}{\|A\|} < 1. \tag{29}$$

Dann erfüllt der Fehler δx die Abschätzungen

$$\|\delta x\| \leq \frac{\|A^{-1}\|}{1 - \kappa(A)\frac{\|\delta A\|}{\|A\|}} \left(\|\delta b\| + \|A^{-1}\|\|\delta A\|\|b\|\right) \tag{30}$$

und

$$\frac{\|\delta x\|}{\|x\|} \leq \kappa(A) \left(1 - \kappa(A)\frac{\|\delta A\|}{\|A\|}\right)^{-1} \left(\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|b\|}\right). \tag{31}$$

Die Konditionszahl (kurz Kondition) der Matrix spielt in diesen Abschätzungen die zentrale Rolle. In diesem Zusammenhang sprechen wir von gut konditionierten Problemen, falls der Verstärkungsfaktor für den relativen Fehler klein (d.m. moderat) ausfällt, andernfalls von schlecht konditionierten Problemen. Wegen (31) wird damit auch die Verwendung des Begriffs Konditionszahl für $\kappa(A)$ klar.

Beweis (von Satz 2.11): Aus (29) folgt mit Hilfe von Aufgabe (2.13), daß $A + \delta A$ invertierbar ist mit

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

Wegen

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

ergibt sich sofort

$$\begin{aligned}\|\delta x\| = \|(A + \delta A)^{-1} (\delta b - \delta A x)\| &\leq \|(A + \delta A)^{-1}\| \left(\|\delta b\| + \|\delta A\|\|x\|\right) \\ &\leq \frac{\|A^{-1}\|}{1 - \kappa(A)\frac{\|\delta A\|}{\|A\|}} \left(\|\delta b\| + \|\delta A\|\|x\|\right)\end{aligned}$$

und wegen $x = A^{-1}b$ unmittelbar (30). Division in dieser Ungleichung durch $\|x\|$, Anwendung der Dreiecksungleichung, Erweiterung des 2ten Terms in der rechten Klammer mit $\|A\|/\|A\|$ und die Tatsache $\|b\| \leq \|A\|\|x\|$ liefern schließlich (31). ■

Bemerkung 2.12. Was bedeuten die Fehlerabschätzungen (30) und (31) für die Lösung des linearen Gleichungssystems $\boldsymbol{A\,x = b}$?

1. Wir betrachten in (28) den Spezialfall $\delta A \equiv 0$. Dann stimmt $\delta b = b - Ax^*$ mit dem durch die gestörte Lösung x^* hervorgerufenen Residuum "uberein und es gilt

$$\|x - x^*\| = \|\delta x\| \leq \|A^{-1}\|\|\delta b\|,$$

bzw.

$$\frac{\|\delta x\|}{\|x\|} \leq \kappa(A)\frac{\|\delta b\|}{\|b\|}.$$

Die relative Fehler kann also groß werden, selbst wenn das durch die gestörte Lösung x^* hervorgerufene relative Residuum klein ist. Der mögliche Verstärkungsfaktor ist gerade durch die Kondition $\kappa(A)$ der Systemmatrix gegeben.

2. Egal, wie schlecht die Kondition der Systemmatrix auch sein mag, es gilt:

Der Gauß Algorithmus mit partieller Pivotisierung (Alg. 2.2) produziert immer kleine Residuen,

siehe [9] für eine ausführliche Diskussion dieses Sachverhaltes. Das ist allerdings nur die *halbe Miete*, denn

3. Falls $\boldsymbol{A}$ problembedingt eine schlechte Kondition hat, so ist das i.allg. nicht zu ändern. Rundungsfehler bedingen dann große relative Fehler und wir können dann nur darauf achten, dass das Verfahren, das zur Lösung von $\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{b}$ verwendet wird, die Kondition nicht verschlechtert. Dies ist beim GEV leider der Fall, denn „gelöst" wird nicht das Problem $\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{b}$, sondern das umgeformte Problem $\boldsymbol{A}^{(n-1)}\,\boldsymbol{x} = \boldsymbol{b}^{(n-1)}$ mit der oberen Dreiecksmatrix $\boldsymbol{A}^{(n-1)} = \boldsymbol{L}_{n-2}\,\ldots\,\boldsymbol{L}_1\,\boldsymbol{L}_0\,\boldsymbol{A}$ (vgl. (19)), und man kann ausrechnen, dass die Anwendung jedes einzelnen $\boldsymbol{L}_j$ die Kondition verschlechtert, d.h. die Kondition von $\boldsymbol{A}^{(n-1)}$ ist schlechter als die von $\boldsymbol{A}$, was besonders dann kritisch ist, wenn die Kondition von $\boldsymbol{A}$ ohnehin schon schlecht ist.

Aufgabe 2.13. Bezeichne $\|\cdot\|$ die Grenznorm. Weisen Sie nach, daß für $F \in M(n,n)$ mit $\|F\| < 1$ die Matrix $(E+F)^{-1}$ existiert und die Abschätzung

$$\|(E+F)^{-1}\| \leq \frac{1}{1-\|F\|}$$

erfüllt ist. Beweisen Sie mit diesem Hilfsmittel, daß mit den Notationen von Satz 2.11 aus (29) die Existenz von $(A+\delta A)^{-1}$ und die Abschätzung

$$\|(A+\delta A)^{-1}\| \leq \frac{\|A^{-1}\|}{1-\kappa(A)\frac{\|\delta A\|}{\|A\|}}$$

folgen.

Weitere, nach Prager und Oettli benannte Fehlerabschätzungen, welche auch ohne die Inverse von A auskommen, sind in [2, (4.4.19)Satz] zu finden.

2.3 Iterative Lösung linearer Gleichungssysteme

Wieder wollen wir das Gls

$$Ax = b$$

numerisch lösen, diesmal jedoch iterativ. Die Verfahren aus den vorangegangenen Kapiteln haben das Gleichungssystem immer exakt gelöst (bis auf Maschinengenauigkeit). In praktischen Aufgabenstellungen ist das allerdings nicht immer notwendig, insbesondere dann nicht, wenn das lineare Gleichungssystem elbst aus einem Modellierungsprozeß resultiert, welcher die *Realität* nur näherungsweise beschreibt, etwa mit einer Fehlertoleranz von 5%. In einem solchen Fall ist es sicherlich statthaft, bei der numerischen Lösung des linearen Gleichungssystems auch einen Fehler im Residuum von der gleichen Größenordnung (etwa 5%) zuzulassen. Mit iterativen Verfahren ist das möglich. Ferner werden wir die Näherungslösung häufig mit wesentlich geringerem Aufwand erhalten als die numerisch exakte Lösung.

Sei A wieder regulär, so daß zu jedem $b \in \mathbb{R}^n$ genau eine Lösung $x \in \mathbb{R}^n$ existiert. Wir spalten A auf in

$$A = W - R,$$

mit regulärem W. Dann gilt

$$Ax = b \Longleftrightarrow Wx = Rx + b.$$

Wir definieren

Definition 2.14. Sei $x^0 \in \mathbb{R}^n$ vorgelegt. Die Vorschrift

$$x^{m+1} = W^{-1}Rx^m + W^{-1}b =: Mx^m + c, \quad m \in \mathbb{N}, \quad N = W^{-1},\ M = W^{-1}R,\ c = Nb,$$

heißt **Basis Iterationsverfahren** zur Lösung von $Ax = b$.

Wir bemerken, daß das Basis Iterationsverfahren umgeschrieben werden kann als

$$W\left(x^{m+1} - x^m\right) = b - Ax^m := r^m, \quad m \in \mathbb{N}, \tag{32}$$

und so wird es auch praktisch implementiert!.

Spezielle Iterationsverfahren werden jetzt durch Variation von W und R bzw. M und N erhalten. Wir gehen im Folgenden davon aus, daß

$$A = D + L + U,$$

wobei D den Diagonalanteil, L das strikte untere und U das strikte obere Dreieck von A bezeichnen.

Definition 2.15. Das Iterationsverfahren $x^{m+1} = Mx^m + c$ mit

- $M = M^J := -D^{-1}(L + U)$, $c = c^J := D^{-1}b$ heißt **Jacobi Verfahren**. Hier ist $W = D$ und $R = -(L + U)$.
- $M = M^J_\omega := (1 - \omega)E - \omega D^{-1}(L + U)$, $c = c^J_\omega := \omega D^{-1}b$ heißt **relaxiertes Jacobi Verfahren** oder **gedämpftes Jacobi Verfahren**.
- $M = M^{GS} := -(L + D)^{-1}U$, $c = c^{GS} := (L + D)^{-1}b$ heißt **Gauß-Seidel Verfahren**. Hier ist $W = L + D$ und $R = -U$.
- $M = M^{SOR}_\omega := -(\omega L + D)^{-1}\left[(\omega - 1)D + \omega U\right]$, $c = c^{SOR}_\omega := \omega(\omega L + D)^{-1}b$ heißt **SOR- Verfahren** bzw. **relaxiertes oder gedämpftes Gauß-Seidel Verfahren**.

Der Begriff **Dämpfung** in diesem Zusammenhang wird nach Satz 2.16 klarer. Motivieren lassen sich die gedämpften (relaxierten) Verfahren wie folgt. Die Basis Iteration aus Definition 2.14 wird gemäß (32) geschrieben in der Form

$$Wx^{m+1} = Wx^m + r^m.$$

Wir können jetzt versuchen, die Korrektur $\delta x^m := x^{m+1} - x^m$ dadurch zu modifizieren (zu verbessern), daß in diesem Gleichungssystem r^m ersetzt wird durch ein gedämpftes (relaxiertes) Residuum ωr^m mit $\omega > 0$, d.m. wir lösen

$$Wx^{m+1} = Wx^m + \omega(b - Ax^m) = (1-\omega)Wx^m + \omega Rx^m + \omega b.$$

Schreiben wir diese Gleichung wieder in Form einer Fixpunkt Iteration, erhalten wir

$$x^{m+1} = \left((1-\omega)E + \omega W^{-1}R\right)x^m + \omega b =: M_\omega x^m + c_\omega$$

und bemerken, daß Dämpfung (oder Relaxierung) einer Modifikation der Iterationsmatrix M entspricht. Jetzt ist auch klar, wie gedämpfte Varianten des Jacobi und des Gauß–Seidel Verfahrens erhalten werden. In der voranstehenden Identität werden lediglich die entsprechenden Matrizen W und R des jeweiligen Verfahrens eingesetzt. Insbesondere wird so das gedämpfte/relaxierte Jacobi Verfahren aus Definition 2.15 erhalten. Die Herleitung des gedämpften/relaxierten Gauß–Seidel Verfahrens in Definiton 2.15 ist ein wenig subtiler; wegen $W = (L + D)$ gilt in (32) komponentenweise

$$a_{jj}x_j^{m+1} = -\sum_{k<j} a_{jk}x_k^{m+1} - \sum_{k>j} a_{jk}x_k^m + b_j \ (1 \le j \le n).$$

Wir ersetzen x_j^{m+1} durch

$$\tilde{x}_j^{m+1} := x_j^m + \omega(x_j^{m+1} - x_j^m) \text{ für } j = 1, \ldots, n$$

und erhalten in Matrixschreibweise

$$(\omega L + D)x^{m+1} = (\omega L + D)x^m + \omega(b - Ax^m),$$

bzw. in der Notation von Definition 2.14

$$x^{m+1} = M_\omega^{SOR}x^m + c_\omega^{SOR}.$$

Hier wird also nicht nur das Residuum r^m mit ω gedämpft/relaxiert, sondern auch die Matrix W wird durch eine Matrix $W(\omega)$ ersetzt.

Es ist klar, daß die o.g. Verfahren nur durchführbar sind, falls alle auftretenden Matrizen wohldefiniert sind.

Jetzt zur Konvergenz von iterativen Verfahren. Wir betrachten zunächst die Basis Iteration $x^{m+1} = Mx^m + c$ und beweisen

Satz 2.16. Das Basis Iterationsverfahren aus Definition 2.14 konvergiert genau dann gegen die Lösung x^* von $Ax^* = b$, wenn der Spektralradius

$$\rho(M) := \max\{|\lambda_i|; \lambda_i \text{ Eigenwert von } M\} < 1 \tag{33}$$

erfüllt.

Beweis: $\Longrightarrow$: Nach Konstruktion ist x^* Lösung der Gleichung $x^* = Mx^* + c$. Damit ergibt sich unter Verwendung des Startvektors x^0

$$x^* - x^{m+1} = M(x^* - x^m) = M^m(x^* - x^0) = \lambda_i^m(x^* - x^0),$$

falls $(x^* - x^0)$ zufällig ein Vielfaches eines Eigenvektors von M zum Eigenwert λ_i ist. Da

$$0 \leftarrow \|x^* - x^{m+1}\| = |\lambda_i|^m \|(x^* - x^0)\|,$$

muß natürlich $|\lambda_i|^m$ für $m \to \infty$ gegen Null konvergieren, also notwendig $|\lambda_i| < 1$ erfüllt sein.

$\Longleftarrow$: Wir zeigen, daß $M^m \to 0$ für $m \to \infty$. Damit folgt dann die Behauptung aus

$$x^* - x^{m+1} = M^m(x^* - x^0).$$

Wir bringen M auf Jordan'sche Normalform mit einer nichtsingulären Matrix T (siehe etwa [4, 4.6.7]),

$$M = TJT^{-1}.$$

Damit gilt auch

$$M^m = TJ^mT^{-1} \text{ für alle } m \in \mathbb{N}.$$

Es reicht offenbar aus, $J^m \to 0$ für $m \to \infty$ nachzuweisen. Dazu schreiben wir J hin;

$$J = \begin{bmatrix} J_1 & & 0 \\ & \ddots & \\ 0 & & J_k \end{bmatrix}, \quad J_i = J_i(\lambda_i) = \text{ diag}(B_i^l(\lambda_i)),\ B_i^l = \begin{bmatrix} \lambda_i & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_i \end{bmatrix} l-\text{ter Jordanblock zu } \lambda_i,$$

wobei nach Voraussetzung $|\lambda_i| < 1$ für $i = 1, \ldots, k$ und k die Anzahl der verschiedenen Eigenwerte von M bezeichnet. Es gilt jetzt

$$J^m = \begin{bmatrix} J_1^m & & 0 \\ & \ddots & \\ 0 & & J_k^m \end{bmatrix},$$

weshalb zu beweisen bleibt, daß $J_i^m \to 0$ für $m \to \infty$, $i = 1, \ldots, k$. Das ist sicherlich erfüllt, falls $B_i^{lm} \to 0$ für $m \to \infty$. Sei jetzt λ Eigenwert von M und $B = B(\lambda)$ Jordan Block zu λ der Länge s ($1 \le s \le$ Vielfachheit von λ). Dann gilt für die Einträge von $B^m(\lambda)$

$$e_i^t B^m(\lambda) e_k = \begin{cases} \lambda^{m-(k-i)} \binom{m}{k-i} & 1 \le i \le k \le s \\ 0 & \text{sonst.} \end{cases}$$

Das heißt aber, daß bei fixen k, i die Einträge von $B^m(\lambda)$ für $m \to \infty$ gegen Null konvergieren, siehe Aufgabe 2.17. Damit ist alles bewiesen. ■

Aus dem vorangegangenen Beweis wird ersichtlich, daß die Reduktion des Fehlers $x^m - x^*$ von Iterationsschritt m nach $m + 1$ im Wesentlichen bestimmt wird durch den betragsmässig größten Eigenwert der Iterationsmatrix M, denn es gilt ja

$$\|x^{m+1} - x^*\| = \|M(x^m - x^*)\| \approx \rho(M)\|x^m - x^*\|.$$

Je kleiner der Spektralradius, desto schneller wird der Fehler reduziert, desto schneller konvergiert demnach das Iterations Verfahren.

Merke 2.1. Die Größe des Spektralradius $\rho(M)$ der Iterationsmatrix bestimmt die Konvergenzgeschwindigkeit der Basis Iteration $x^{m+1} = Mx^m + c$.

Aufgabe 2.17. Weisen Sie nach, daß mit $B(\lambda)$ aus obigem Beweis

$$e_i^t B^m(\lambda) e_k = \begin{cases} \lambda^{m-(k-i)} \binom{m}{k-i} & 1 \leq i \leq k \leq s \\ 0 & \text{sonst.} \end{cases}$$

und daß für $|\lambda| < 1$ bei fixen k, i

$$\lambda^{m-(k-i)} \binom{m}{k-i} \to 0 \text{ für } m \to \infty.$$

Folgerung 2.18. Hinreichend für die Konvergenz der Basisiteration aus Definition 2.14 ist

$$\|M\| < 1.$$

Beweis: Ist x Eigenvektor von M zum Eigenwert λ mit $\|x\| = 1$, so folgt

$$|\lambda| = |\lambda|\|x\| = \|\lambda x\| = \|Mx\| \leq \|M\|\|x\| = \|M\|,$$

d.m. mit $\|M\| < 1$ ist auch $\rho(M) < 1$. Satz 2.16 liefert nun die Behauptung. ■

Für die numerische Realisierung iterativer Verfahren werden **Abbruch Bedingungen** benötigt. Abbruch Bedingungen sollten natürlich nur berechenbare Größen enthalten. Ist eine Toleranz $\epsilon > 0$ vorgelegt, so können wir die Iteration etwa stoppen, falls

1. $\|r^m\| = \|b - Ax^m\| \leq \epsilon \|r^0\|$, oder
2. $\|x^{m+1} - x^m\| \leq (1 - \rho_m)\epsilon \|x^m\|$

erfüllt ist. In der zweiten Bedingung ist

$$\rho_m = \frac{\|x^{m+1} - x^m\|}{\|x^m - x^{m-1}\|} \approx \rho(M) \text{ für große } m \in \mathbb{N}.$$

Das erste Abbruchkriterium muß nicht gewährleisten, dass bei Terminierung auch der relative Fehler $\frac{\|x^m - x^*\|}{\|x^*\|}$ in der numerischen Lösung klein ist. Insbesondere trifft das auf Matrizen A mit schlechter Kondition zu. Die zweite Abbruchkriterium schafft hier Abhilfe, denn sie gewährleistet, dass der realtive Fehler $\frac{\|x^m - x^*\|}{\|x^*\|}$ bei Terminierung klein ist. Dass dem so ist, lässt sich wie folgt motivieren;

1. $\|x^{m+1} - x^*\| \approx \rho(M)\|x^m - x^*\|$ und $\|x^{m+1} - x^m\| \approx \rho(M)\|x^m - x^{m-1}\|$ (letzteres für große m).
2. $\|x^m - x^*\| \leq \|x^{m+1} - x^m\| + \|x^{m+1} - x^*\| \leq \|x^{m+1} - x^m\| + \rho(M)\|x^m - x^*\|$. Demnach
3. $\|x^m - x^*\| \leq \frac{\|x^{m+1} - x^m\|}{1 - \rho(M)}$. Ersetze noch $\rho(M)$ gemäß
4. $\rho_m := \frac{\|x^{m+1} - x^m\|}{\|x^m - x^{m-1}\|} (\approx \rho(M))$ und $x^* \approx x^m$. Damit

5. $\frac{\|x^m - x^*\|}{\|x^*\|} \leq \epsilon$, (hoffentlich) falls $\|x^{m+1} - x^m\| \leq (1 - \rho_m)\epsilon\|x^m\|$.

Eine ausführliche Diskussion findet sich in [13, Kapitel 2.5]. Hier sei noch bemerkt, daß im Fall $\|M\| < 1$ die Mindestanzahl k_ϵ von Iterationen abgeschätzt werden kann, die zur Reduktion des Ausgangsfehlers $\|d^0\| = \|x^0 - x^*\|$ auf $\epsilon\|d^0\|$ benötigt werden. Dazu setzen wir an

$$\|M\|^{k_\epsilon}\|d^0\| \leq \epsilon\|d^0\| \Longrightarrow k_\epsilon \geq \frac{\ln \epsilon}{\ln \|M\|}.$$

Dabei wird natürlich davon ausgegangen, daß $\|M\|$ berechnet werden kann.

Jetzt noch einige Konvergenzresultate für die Verfahren aus Definition 2.15.

Satz 2.19. (Starkes Zeilen- und Spaltensummenkriterium)
Die Matrix A erfülle das starke Zeilensummenkriterium, d.m.

$$|a_{ii}| > \sum_{k \neq i} |a_{ik}| \text{ für alle } 1 \leq i \leq n, \tag{34}$$

oder das starke Spaltensummenkriterium, d.m.

$$|a_{kk}| > \sum_{i \neq k} |a_{ik}| \text{ für alle } 1 \leq k \leq n, \tag{35}$$

Dann sind Gesamt- und Einzelschrittverfahren für jeden Startwert $x^0 \in \mathbb{C}^n$ konvergent. Ferner gilt

$$\text{lub}_\infty(M^{GS}) \leq \text{lub}_\infty(M^J) < 1.$$

Beweis: (Nur für Gesamtschrittverfahren, vollständig siehe [3]). Wegen $M^J = -D^{-1}(L + U)$ gilt

$$\text{lub}_\infty(M^j) = \max_i \frac{1}{|a_{ii}|} \sum_{k \neq i} |a_{ik}| < 1.$$

Folgerung 2.18 liefert daher die Behauptung für das Zeilensummenkriterium. Das Spaltensummenkriterium entspricht dem Zeilensummenkriterium für A^t. Daher konvergiert das Gesamtschrittverfahren für A^t und es gilt nach Satz 2.16 $1 > \rho(M^J(A^t)) = \rho(M^{J^t}) = \rho(M^J)$, weil die Transponierte einer Matrix dieselben Eigenwerte wie die Matrix hat. Also konvergiert nach Satz 2.16 das Gesamtschrittverfahren auch für A. Der Nachweis für das Einzelschrittverfahren inc. der Abschätzung der Grenznormen findet sich etwa in [3]. ■

Satz 2.20. (Schwaches Zeilensummenkriterium]
Die Matrix A sei unzerlegbar, d.m. der ihr zugeordnete gerichtete Graph D mit Knoten $\{P_1, \ldots, P_n\}$ und Kanten $e_{ij} = P_i \to P_j$, falls $a_{ij} \neq 0$ sei wegzusammenhängend, d.m. von jedem Knoten P_i führt zu jedem anderen Knoten ein gerichteter Weg. Ferner sei das schwache Zeilen- oder Spaltensummenkriterium für A erfüllt, d.m.

$$|a_{ii}| \geq \sum_{k \neq i} |a_{ik}| \text{ für alle } 1 \leq i \leq n \text{ und } |a_{i_0 i_0}| > \sum_{k \neq i_0} |a_{i_0 k}| \text{ für ein } i_0. \tag{36}$$

Dann konvergieren Gesamt- und Einzelschrittverfahren.

Bemerkung 2.21. Matrizen A, welche das starke Zeilensummenkriterium erfüllen, heißen **stark diagonaldominant**, solche ‚die das schwache Zeilensummenkriterium erfüllen, heißen **irreduzibel diagonaldominant**. Dabei ist **irreduzibel** ein anderer Ausdruck für **unzerlegbar**.

Es ist leicht einzusehen, dass stark diagonaldominante und irreduzibel diagonaldominante Matrizen regulär sind, siehe etwa [12], wo auch einige der in diesem Kapitel aufgeführten Resultate diskutiert werden.

Beweis(von Satz 2.20): Nur für das Gauß–Seidel Verfahren. Da $a_{ii} \neq 0$ für alle $i \in \{1, \ldots, n\}$ (vergl. Bemerkung 2.21) ist das Gauß–Seidel Verfahren durchführbar. Wir zeigen jetzt $\rho(M^{GS}) < 1$, indem wir nachweisen, dass $M^{GS} - \lambda E$ regulär ist für $|\lambda| \geq 1$, denn dann kann M^{GS} bekanntlich keine Eigenwerte mit Betrag ≥ 1 haben und der Spektralradius muß kleiner sein als 1.

Weil $(L + D)^{-1}$ existiert, schließen wir $M^{GS} - \lambda E = -(L + D)^{-1}U - \lambda E$ regulär gdw $B := U + \lambda D + \lambda L$ regulär. B ist regulär, denn B ist irreduzibel diagonaldominant. Irreduzibel ist klar, weil A irreduzibel. Ferner für $k \in \{1, \ldots, n\}$ und wegen $|\lambda| \geq 1$

$$\sum_{j \neq k} |b_{kj}| = \sum_{j=1}^{k-1} |\lambda||a_{kj}| + \sum_{j=k+1}^{n} |a_{kj}| \leq |\lambda| \sum_{j \neq k} |a_{kj}| \leq |\lambda||a_{kk}| = |b_{kk}|,$$

wobei mindestens für ein k die Ungleichung strikt erfüllt wird. Also ist B irreduzibel diagonaldominant und somit regulär. ■

Der Beweis für das Jacobi Verfahren findet sich sich etwa in [12, 3].

Konvergenzaussagen für das relaxierte Gesamtschrittverahren finden sich in [14]. Jetzt noch einige Konvergenzaussagen für das relaxierte Einzelschrittverfahren.

Satz 2.22. Es gilt

$$\rho(M_\omega^{SOR}) \geq |\omega - 1|, \tag{37}$$

d.h. daß das SOR Verfahren nur für $0 < \omega < 2$ konvergent sein kann.

Beweis: Untersuche die Eigenwerte von $M_\omega^{SOR} = -(\omega L + D)^{-1}[(1 - \omega)D + \omega U]$. Zunächst bemerken wir, daß $\det(\omega L + D) = \det D$, also

$$\phi(\lambda) := \det(\lambda E - M_\omega^{SOR}) = \frac{1}{\det D}\det[(\omega L + D)(\lambda E - M_\omega^{SOR})] = \frac{1}{\det D}\det[(\lambda + 1 - \omega)D + \omega U + \lambda\omega L].$$

Ferner ist aus der linearen Algebra bekannt, daß $\phi(0) = \det M_\omega^{SOR} = \prod \lambda_i(M_\omega^{SOR})$. Demnach

$$\prod_i \lambda_i(M_\omega^{SOR}) = \phi(0) = \frac{1}{\det D}\det[(1 - \omega)D + \omega U] = (1 - \omega)^n.$$

■

Beispiel. Eine wichtige Klasse von Matrizen bilden Tridiagonalmatrizen. Sie entstehen etwa bei der Diskretisierung von 2-Punkt Randwertaufgaben der Form

$$-u''(x) + b(x)u'(x) + c(x)u(x) = f(x) \text{ für alle } x \in I = (0, 1), \text{ und } u(0) = u(1) = 0.$$

Dabei bezeichnen b, c, f gegebene, hinreichend glatte Funktionen, die Funktion u ist gesucht (und durch die Differentialgleichung und die Randbedingungen $u(0) = u(1) = 0$ eindeutig festgelegt). Zu $n \in \mathbb{N}$ bezeichnen wir mit $x_i := ih(i = 0, \ldots, n+1)$ ein Gitter "uber I, wobei $h := \frac{1}{n+1}$, und berechnen Näherungswerte u_i an die Funktionswerte $u(x_i)$, indem wir die in der Differentialgleichung auftretenden Ableitungen durch geeignete Differenzenquotienten ersetzen. Wir erhalten

$$-\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + b(x_i)\frac{u_{i+1} - u_{i-1}}{2h} + c(x_i)u_i = f(x_i) \text{ für } i = 1, \ldots, n, \text{ und } u_0 = u_{n+1} = 0.$$

Die Bestimmung des Lösungsvektors $U = [u_1, \ldots, u_n]^t$ führt auf die numerische Lösung des linearen Gleichungssystems

$$\begin{bmatrix} \beta_1 & \gamma_1 & & \\ \alpha_2 & \ddots & \ddots & \\ & \ddots & \ddots & \gamma_{n-1} \\ & & \alpha_n & \beta_n \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ \vdots \\ u_n \end{bmatrix} = \begin{bmatrix} f(x_1) \\ \vdots \\ \vdots \\ f(x_n) \end{bmatrix}.$$

Dabei ist $\alpha_i = -(\frac{1}{h^2} + \frac{b(x_i)}{2h})$, $\gamma_i = -(\frac{1}{h^2} - \frac{b(x_i)}{2h})$ und $\beta_i = \frac{2}{h^2} + c(x_i)$.

Für konstante α_i, β_i und γ_i können die Eigenwerte der Systemmatrix aus dem vorangegangenen Beispiel angegeben werden.

Satz 2.23. (Eigenwerte einer Tridiagonalmatrix)
Sei

$$B := \begin{bmatrix} \beta & \gamma & & \\ \alpha & \ddots & \ddots & \\ & \ddots & \ddots & \gamma \\ & & \alpha & \beta \end{bmatrix} \in \mathbb{R}^{nxn}$$

mit $\alpha \cdot \gamma > 0$. Dann besitzt B die Eigenwerte

$$\lambda_i = \beta + 2\sqrt{\alpha\gamma}\text{sign}(\alpha)\cos\frac{i\pi}{n+1},\ 1 \le i \le n \tag{38}$$

und die Eigenvektoren v^i mit den Komponenten

$$v^i_j = (\frac{\alpha}{\gamma})^{\frac{j-1}{2}}\sin\frac{ij\pi}{n+1},\ 1 \le i \le n, 1 \le j \le n. \tag{39}$$

Der Beweis dieses Satzes ist elementar.

Die Bedingung aus Satz 2.22 findet sich wieder im

Satz 2.24. (Ostrowski/Reich)
Ist A positiv definit, so gilt

$$\rho(M^{SOR}_\omega) < 1 \text{ für alle } 0 < \omega < 2. \tag{40}$$

Ein Beweis dieses Satzes wird etwa in [3] gegeben. Der Satz besagt, dass die Klasse der SOR–Verfahren f"ur positiv definite Matrizen konvergiert. Insbesondere konvergiert also das Gau"s–Seidel Verfahren. Die Aussage des Satzes ist nicht richtig f"ur das Jacobi-Verfahren, wie das Beispiel

$$A = \begin{bmatrix} 2 & -1 & 1 \\ -1 & 2 & -1 \\ 1 & -1 & 2 \end{bmatrix}$$

zeigt. A besitzt die Eigenwerte 4 und 1 (doppelt), ist also positiv definit, die Matrix M^J des Jacobi–Verfahrens die Eigenwerte -1 und 1/2 (doppelt), so dass $\rho(M^J) = 1$, das Jacobi–Verfahren nach Satz 2.16 demnach nicht konvergent sein kann.

Das nachfolgende Beispiel besitzt gro"se Praxisrelevanz und stellt eine positiv definite Matrix bereit, welcher wir noch h"aufig begegnen werden.

Beispiel. Wir betrachten die Finite Differenzen Diskretisierung der **Poisson Gleichung**

$$-\Delta u(x) = f(x) \text{ in } \Omega := (0,1) \times (0,1), \quad u(x) = 0 \text{ auf } \partial\Omega, \tag{41}$$

wobei $\Delta := \sum_i \frac{\partial^2}{\partial x_i^2}$ den Laplace Operator bezeichnet. Zur Diskretisierung mit finiten Differenzen benötigen wir ein Gitter. Dazu sei zu $n \in \mathbb{N}$ die Gitterweite $h := \frac{1}{n+1}$, $x_i := ih$, $y_j := jh$, $i, j = 0, \ldots n+1$ und

$$\Omega_h := \{(x_i, y_j); i, j = 1, \ldots, n\} \text{ Gitterpunkt Menge.}$$

Ziel ist die Berechnung von Näherungen $u_{ij} \approx u(x_i, y_j)$, wobei u die Lösung des Poisson Problems bezeichnet. Dazu ersetzen wir in der Differentialgleichung den Laplace Operator durch geeignete dividierte Differenzen und werten f an den Stellen (x_i, y_j) aus (Diskretisierung mittels 5-Punkt Stern):

$$\Delta u(x_i, y_j) \approx \frac{1}{h^2}\left(u_{i+1j} - 2u_{ij} + u_{i-1j} + u_{ij+1} - 2u_{ij} + u_{ij-1}\right) = f(x_i, y_j) \text{ für } i, j = 1, \ldots, n. \tag{42}$$

Wir erhalten n^2 Gleichungen für $(n+2)^2$ Unbekannte. Die restlichen $4n+4$ Unbekannten ergeben sich aus der Forderung $u = 0$ auf $\partial\Omega$, hier $u_{i0} = u_{in+1} = u_{0,j} = u_{n+1j} = 0$ für $i, j = 0, \ldots, n+1$. Arbeiten wir diese Bedingungen in (42) ein, erhalten wir ein lineares Gleichungssystem mit block-tridiagonaler Koeffizienten Matrix;

$$AU = F \Longleftrightarrow \begin{bmatrix} T & -I & & \\ -I & T & \ddots & \\ & \ddots & \ddots & -I \\ & & -I & T \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ \vdots \\ U_n \end{bmatrix} = h^2 \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_n \end{bmatrix}, \tag{43}$$

wobei

$$T := \begin{bmatrix} 4 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 4 \end{bmatrix} \in M(n,n),\ I \in M(n,n) \text{ Einheitsmatrix },\ U_i := \begin{bmatrix} u_{1i} \\ u_{2i} \\ \vdots \\ u_{ni} \end{bmatrix} \in \mathbb{R}^n,\ F_i \text{ analog.}$$

Aufgabe 2.25. Weisen Sie nach, daß die Matrix A aus (43) positiv definit ist. Tip (nur gültig mit Nachweis): Die Eigenvektoren z^{kl} und Eigenwerte λ^{kl} von A sind

$$z^{(kl)} \in \mathbb{R}^{n^2}, z_{ij}^{(kl)} = \sin k\pi ih \sin l\pi jh,\ \lambda^{(kl)} = 4 - 2\left(\cos k\pi h + \cos l\pi h\right).$$

Mit den Aussagen dieser Aufgabe folgern wir sofort aus dem Satz 2.24, daß das SOR Verfahren für unser Beispiel immer konvergiert, falls $0 < \omega < 2$ gilt.

Die Matrix A aus (43) ist ein Beispiel aus der Klasse der sogenannten konsistent geordneten Matrizen. Für Matrizen aus dieser Klasse kann der Relaxationsparameter ω^* angegeben werden, für welchen das SOR Verfahren am besten konvergiert.

Definition 2.26. Eine Matrix $A = (L + D + U) \in M(n, n)$ heißt **konsistent geordnet** $:\Longleftrightarrow$

$$J(\alpha) := \alpha D^{-1}L + \frac{1}{\alpha}D^{-1}U,\ \alpha \neq 0,$$

besitzt Eigenwerte, die unabhängig sind von α.

Beispiel. Die Matrix A aus (43) ist konsistent geordnet, den es gilt in diesem Fall

$$J(\alpha) = S_\alpha J(1) S_\alpha^{-1} \text{ mit } S_\alpha := \text{diag}(K, \alpha K, \ldots, \alpha^{n-1}K) \in M(n^2, n^2),\ \text{wobei } K := \text{diag}(1, \alpha, \ldots, \alpha^{n-1}).$$

Für konsistent geordnete Matrizen fassen wir zusammen (siehe [3, Kapitel 8]

Satz 2.27. (Varga/Young)
Sei A konsistent geordnet. Dann gilt mit der Notation aus Definition 2.15

i. $\rho(M^{GS}) = \rho^2(M^J)$.

ii. Seien alle Eigenwerte von M^J reell und $\rho(M^J) < 1$. Dann ist

$$\omega^* := \text{argmin}_{0<\omega<2}\rho(M_\omega^{SOR})$$

explizit bestimmbar und es gilt

$$\omega^* = \frac{2}{1 + \sqrt{1 - \rho(M^J)^2}},\ \text{sowie } \rho(M_{\omega^*}^{SOR}) = \omega^* - 1. \tag{44}$$

Für den Spektralradius von M_ω^{SOR} gilt

$$\rho(M_\omega^{SOR}) = \begin{cases} \omega - 1 & \text{für } \omega^* \leq \omega \leq 2 \\ 1 - \omega + \frac{1}{2}\omega^2\rho(M^J)^2 + \omega\rho(M^J)\sqrt{1 - \omega + \frac{1}{4}\omega^2\rho(M^J)^2} & \text{für } 0 \leq \omega \leq \omega^*. \end{cases} \tag{45}$$

Sie sind jetzt sicherlich in Lage, wieder mal eine Aufgabe zu lösen.

Aufgabe 2.28. Es bezeichne wieder A die Matrix aus der Diskretisierung des Poisson Problems (41). Dann gilt

i. $\rho(M^J) = \cos \pi h$.

ii. $\rho(M^{GS}) = \cos^2 \pi h$.

iii.
$$\omega^* = \frac{2}{1 + \sin \pi h} \text{ und } \rho(M^{SOR}_{\omega^*}) = \frac{\cos^2 \pi h}{(1 + \sin \pi h)^2}.$$

iv. Die Zahl $I(n)$ mit $\rho(M^J)^{I(n)} = \rho(M^{SOR}_{\omega^*})$ gibt an, wie viele Schritte des Jacobi Verfahrens dieselbe Fehlerreduktion wie ein Schritt des optimalen Relaxationsverfahrens liefern. Weisen Sie nach, daß asymptotisch gilt
$$I(n) = \frac{4(n+1)}{\pi}.$$

Hinweis für i.: Die Matrix M^J besitzt dieselben Eigenvektoren wie A.

Eine Folgerung aus der vorangegangenen Aufgabe ist, daß die Konvergenz der in Definition 2.15 definierten iterativen Verfahren bei der numerischen Lösung des Poisson Problems mit zunehmender Diskretisierungsfeinheit h abnimmt.

2.3.1 Iterative Verfahren mit endlich vielen Iterationen

Neben direkten und iterativen Verfahren zur numerischen Lösung von linearen Gleichungssystemen gibt es noch iterative Verfahren, die bei exakter Rechnung die Lösung des Gleichungssystems nach endlich vielen Schritten berechnen. Diese Klasse von Verfahren verquickt demnach die Möglichkeit, Lösungen exakt zu berechnen, mit der Flexibilität von iterativen Verfahren. Die Entwicklung solcher Verfahren erfordert weitreichende Kenntnisse der linearen Algebra und wird im Folgenden nur skizziert. Nachfolgend dargestellt Verfahren firmieren unter dem Oberbegriff der **Krylov–Raum Methoden**. Starten wollen wir mit Gleichungssystemen für positiv definite Matrizen. Das führt auf das Verfahren der konjugierten Gradienten, kurz

CG- Verfahren (Conjugate Gradient Method): Wir gehen aus von dem Gleichungssystem
$$Ax = b \tag{46}$$
mit positiv definiter Matrix A. Gleichungssysteme mit positiv definiter Systemmatrix haben die sehr schöne Eigenschaft, daß (46) die notwendige und hinreichende Bedingung dafür beschreibt, daß x das eindeutig bestimmte Minimum des quadratischen Funktionals
$$F(v) := \frac{1}{2}\langle v, Av\rangle - \langle b, v\rangle$$
darstellt, i.e.
$$x = \arg\min_{v \in \mathbb{R}^n} F(v),$$
wobei $\langle \cdot, \cdot \rangle$ das Euklidische Skalarprodukt bezeichnet. D.h., wir können zur numerischen Lösung von (46) auch Minimierungsalgorithmen verwenden. Die einfachsten Minimierungsalgorithmen sind die sogenannten **Koordinaten–Abstiegs–Verfahren**. Zu deren Illustration beschränken wir uns zunächst auf den Fall $A = E$. Dann ist die Lösung von (46) gegeben durch $x = b$. Der

nachfolgende Algorithmus berechnet x nach höchstens n Schritten, indem die Funktion $F(v)$ sukzessive entlang der Koordinatenrichtungen $e^1, \ldots, e^n$ minimerit wird (hier bezeichnet e^j den j-ten Einheitsvektor im $\mathbb{R}^n$);

Koordinaten–Abstiegs–Verfahren für $F(v) = \frac{1}{2}\langle v, Ev\rangle - \langle b, v\rangle$ bzgl. $e^1, \ldots, e^n$.

1. $x^0 \in \mathbb{R}^n$ gegeben, $i = 0$.
2. Setze
$$s_i := b_{i+1} - x^i_{i+1} \quad (\equiv \arg\min_{s\in\mathbb{R}} F(x^i + se^{i+1})),$$
3. Datiere auf
$$x^{i+1} = x^i + s_i e^{i+1},$$
4. Falls $|b - x^{i+1}| = 0$ Stop. Ansonsten setze $i = i + 1$ und gehe nach 2.

Es ist unmittelbar klar, daß dieser Algorithmus nach höchstens n Iterationen mit $x = b$ terminiert.

Wir "ubertragen die Idee des Koordinatenabstiegs jetzt auf positiv definite Matrizen A. Dazu bezeichne $p^0, \ldots p^{n-1}$ einen Satz A–**konjugierter Richtungen**, d.h. es gelte

$$\langle Ap^j, p^i\rangle = \begin{cases} 0, \text{ falls } j \neq i \\ \neq 0, \text{ falls } j = i. \end{cases}$$

Zunächst stellen wir fest, daß die Lösung x von $Ax = b$ bzgl. der Basis $p^0, \ldots, p^{n-1}$ dargestellt werden kann in der Form

$$x = \sum_{j=0}^{n-1} \tilde{\alpha}_j p^j \text{ mit } \tilde{\alpha}_j = \frac{\langle b, p^j\rangle}{\langle Ap^j, p^j\rangle}.$$

Wir wenden jetzt das Koordinaten–Abstiegs–Verfahren auf die Funktion $F(v) := \frac{1}{2}\langle v, Av\rangle - \langle b, v\rangle$ bzgl. des Koordinatensystems $p^0, \ldots, p^{n-1}$ an (die Numerierung ist jener in Algorithmus 2.29 geschuldet).

Koordinaten–Abstiegs–Verfahren für $F(v) = \frac{1}{2}\langle v, Av\rangle - \langle b, v\rangle$ bzgl. $p^0, \ldots, p^{n-1}$.

1. $x^0 \in \mathbb{R}^n$ gegeben, $i = 0$.
2. Setze
$$\alpha_i := \frac{\langle \overbrace{b - Ax^i}^{r^i}, p^i\rangle}{\langle Ap^i, p^i\rangle} \quad (\equiv \arg\min_{s\in\mathbb{R}} F(x^i + sp^i) \text{ (NACHWEIS!)}),$$
3. Datiere auf
$$x^{i+1} = x^i + \alpha_i p^i,$$
4. Falls $|r^{i+1}| = |b - Ax^{i+1}| = 0$ Stop. Ansonsten setze $i = i + 1$ und gehe nach 2.

Jetzt ist allerdings nicht mehr unmittelbar klar, daß dieser Algorithmus nach höchstens n Iterationen mit $x = A^{-1}b$ terminiert. Dass dem so ist, weisen wir nach, indem wir

$$\langle r^i, p^j \rangle = 0 \text{ für } 0 \leq j \leq i-1 \text{ und } 1 \leq i \leq n$$

nachweisen. Denn dann gilt natürlich

$$\langle b - Ax^n, p^j \rangle = \langle r^n, p^j \rangle = 0 \text{ für } j = 0, \ldots, n-1,$$

also $Ax^n = b$, denn die Vektoren $p^0, \ldots, p^{n-1}$ sind linear unabhängig. Der Beweis funktioniert mit Induktion nach n;

Anfang $n = 1$:

$$\langle r^1, p^0 \rangle = \langle b - Ax^0 - \alpha_0 Ap^0, p^0 \rangle = \langle r^0, p^0 \rangle - \frac{\langle r^0, p^0 \rangle}{\langle Ap^0, p^0 \rangle} \langle Ap^0, p^0 \rangle = 0.$$

Schritt $k \to k+1$: Für $j = 0, \ldots, k$

$$\langle r^{k+1}, p^j \rangle = \langle b - Ax^k - \alpha_k Ap^k, p^j \rangle = \overbrace{\langle r^k, p^j \rangle}^{=0 \text{ für } 0 \leq j \leq k-1} - \frac{\langle r^k, p^k \rangle}{\langle Ap^k, p^k \rangle} \overbrace{\langle Ap^k, p^j \rangle}^{=0 \text{ für } 0 \leq j \leq k-1} = 0.$$

■

In der Praxis stehen wir allerdings vor dem Problem, daß wir die konjugierten Richtungen $p^0, \ldots, p^{n-1}$ i.d.R. nicht kennen und deren numerische Berechnung (etwa mit dem Orthogonalisierungsverfahren von Gram/Schmidt) genau so aufwändig ist wie die numerische Lösung des linearen Gleichungssystems $Ax = b$. Die auf Hestenes und Stiefel zurückgehende Idee besteht nun darin, ausgehend von einer Ausgangsrichtung p^0, die konjugierten Richtungen p^{i+1} sukzessive aus dem aktuellen Residuum r^{i+1} und den Richtungen $p^0, \ldots, p^i$ $(1 \leq i \leq n-1)$ zu berechnen. Der nachfolgende Algorithmus fasst dieses Vorgehen zusammen (tatsächlich werden nur r^{i+1}, p^i zur Bestimmung von p^{i+1} benötigt!).

Algorithmus 2.29. (Basis CG-Verfahren)
Gegeben seien $b, x^0 \in \mathbb{R}^n$ und eine positiv definite Matrix $A \in M(n,n)$. Berechnet wird die Lösung x von (46).

Initialisierung

$$r^0 = b - Ax^0$$
$$p^0 = r^0$$
$$i = 0$$

Do While $i \leq n$ und $r^i \neq 0$

$$\alpha_i = (r^i, p^i)/(p^i, Ap^i)$$
$$x^{i+1} = x^i + \alpha_i p^i$$
$$r^{i+1} = r^i - \alpha_i Ap^i$$

$\beta_i = (r^{i+1}, Ap^i)/(p^i, Ap^i)$
$p^{i+1} = r^{i+1} - \beta_i p^i$
$i = i + 1$

Endwhile

Der numerische Aufwand in jedem Iterationsschritt besteht in einer Matrix-Vektor-Multiplikation und der Berechnung von 3 Skalarprodukten. Ein Skalarprodukt wird eingespart, falls

$$\left\{ \begin{array}{lcl} \alpha_i & = & |r^i|^2/(p^i, Ap^i) \\ \beta_i & = & |r^{i+1}|^2/|r^i|^2 \\ p^{i+1} & = & r^{i+1} + \beta_i p^i \end{array} \right. \tag{47}$$

im obigen Algorithmus gesetzt wird. Aufgrund von Orthogonalitätseigenschaften der Iterierten ergibt sich mit (47) ein zum Algorithmus 2.29 äquivalenter Algorithmus. Die Erfahrung zeigt allerdings, daß der numerische Mehraufwand stabilisierend wirken kann, d.h., daß die sparsame Variante häufiger versagt als die spendablere. Zunächst einige Aufgaben.

Aufgabe 2.30. cg-Verfahren
Berechnen Sie mit Hilfe des cg-Verfahrens Algorithmus 2.29 die **exakte** Lösung des Gleichungssystems

$$\begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix} x = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

Aufgabe 2.31. Eigenschaften des cg-Verfahrens
Betrachten Sie das cg-Verfahren aus Algorithmus 2.29.

1. Zeigen Sie, dass sowohl im Basis cg-Verfahren (Algorithmus 2.29) als auch in der Formulierung (47) stets r^{i+1} orthogonal zu p^i ist.
2. Zeigen Sie, dass beide Formulierungen des cg-Verfahrens äquivalent sind.

Der Algorithmus 2.29 hat folgende grundlegende Eigenschaften.

Satz 2.32. (CG-Verfahren ist ein direktes Verfahren)
Sei $A \in M(n, n)$ symmetrisch und positiv definit. Dann

1. erzeugt Algorithmus 2.29 konjugierte Richtungen p^i mit $\langle Ap^i, p^j \rangle = 0$ f"ur $i \neq j$,
2. gilt f"ur alle Residuen $\langle r^i, p^j \rangle = 0$ f"ur $j = 0, \ldots, i-1$,
3. konvergiert das CG-Verfahren in Algorithmus 2.29 nach höchstens n Schritten gegen die exakte Lösung x^* des Gleichungssystems (46), wobei die Wahl des Startwertes x^0 beliebig ist.

Beweis: [5, Satz 5.2], [12]. ■

Das CG-Verfahren ist demnach ein direktes Verfahren oder kann als solches aufgefaßt werden. Seine rekursive Formulierung läßt jedoch zu, die Lösung von (46) bis zu einer vorgelegten Genauigkeit zu berechnen. Die Anzahl der dazu benötigten Iterationen kann abgeschätzt werden.

Satz 2.33. (Konvergenzgeschwindigkeit des CG-Verfahrens)
Sei $A \in M(n,n)$ positiv definit und $x^0 \in \mathbb{R}^n$ beliebiger Startwert. Die Iterierten x^k des CG-Verfahrens erfüllen

$$\sqrt{(A(x^k - x^*), x^k - x^*)} \;\leq\; 2\left\{\frac{\sqrt{\lambda_{max}(A)} - \sqrt{\lambda_{min}(A)}}{\sqrt{\lambda_{max}(A)} + \sqrt{\lambda_{min}(A)}}\right\}^k \sqrt{(A(x^0 - x^*), x^0 - x^*)} \quad (48)$$

Beweis: [5, Satz 5.3] ■

Es bezeichne

$$|x|_A := \sqrt{x^t A x}$$

die zur Matrix A assoziierte Vektornorm und

$$\kappa(A) = \frac{\lambda_{max}(A)}{\lambda_{min}(A)} (= \kappa_2(A)) \quad (49)$$

die Kondition der symmetrischen, positiv definiten Matrix A. Aus (47) wird eine Abschätzung erhalten dafür, wieviele Iterationen des CG-Verfahrens höchstens durchzuführen sind, um den Ausgangsfehler auf die Größenordnung ϵ zu reduzieren. Hinreichend dafür ist, daß k

$$\left\{\frac{\sqrt{\lambda_{max}(A)} + \sqrt{\lambda_{min}(A)}}{\sqrt{\lambda_{max}(A)} - \sqrt{\lambda_{min}(A)}}\right\}^k \geq \frac{2}{\epsilon}$$

erfüllt. Mit Hilfe von (49) kann das auch umgeschrieben werden zu

$$\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)^k \;\geq\; \frac{2}{\epsilon},$$

also

$$k \;\geq\; \ln\frac{2}{\epsilon} \,/\, \ln\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}.$$

Wird noch

$$\ln\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1} \geq \frac{2}{\sqrt{\kappa}}, \quad \kappa > 1,\ \kappa = \kappa(A)$$

berücksichtigt, so ist für

$$k \;\geq\; \frac{1}{2}\sqrt{\kappa(A)}\ln\left(\frac{2}{\epsilon}\right) + 1 \quad (50)$$

sichergestellt, daß

$$|x^k - x^*|_A \;\leq\; \epsilon |x^1 - x^*|_A$$

gilt.

Die maßgebliche Größe für die Konvergenzgeschwindigkeit ist hier die Kondition der Matrix A. Ihre Güte hängt maßgeblich von der Breite des Spektrums von A ab, d.h., je weiter $\lambda_{max}(A)$ und $\lambda_{min}(A)$ auseinanderliegen, desto pessimistischer sind die Erwartungen hinsichtlich der Fehlerreduktion. Die Konvergenzgeschwindigkeit kann mittels Vorkonditionierung verbessert werden. Dazu bemerke, daß die eindeutige Lösung x^* von (46) auch das Minimum der Funktion

$$f(x) := \frac{1}{2}x^t Ax - b^t x$$

darstellt. Sei jetzt C eine nichtsinguläre Matrix. Dann ist mit

$$\tilde{A} := C^{-1}AC^{-t}, \quad \tilde{b} := C^{-1}b$$

das eindeutige Minimum y^* der Funktion

$$\tilde{f}(y) := \frac{1}{2}y^t \tilde{A}y - \tilde{b}^t y$$

durch

$$y^* = \tilde{A}^{-1}\tilde{b} = C^t x^*$$

gegeben, wobei x^* die Lösung von (46) darstellt. Ferner gilt wegen der positiven Definitheit von $\tilde{A}$ mit (48)

$$|y^k - y^*|_{\tilde{A}} \leq 2\left\{\frac{\sqrt{\kappa(\tilde{A})}-1}{\sqrt{\kappa(\tilde{A})}+1}\right\}^k |y^0 - y^*|_{\tilde{A}} \tag{51}$$

und zu gegebenen $\epsilon > 0$ gilt für die in (49) hergeleitete Zahl $k = k(\epsilon)$

$$k \geq \frac{1}{2}\sqrt{\kappa(\tilde{A})}ln\left(\frac{2}{\epsilon}\right) + 1.$$

Ist $\kappa(\tilde{A}) < \kappa(A)$, so wird wohl y^* schneller gut approximiert als x^*. Weil mit

$$x^* = C^{-t}y^*, \quad x^k = C^{-t}y^k$$

$$|y^k - y^*|_{\tilde{A}} = |x^k - x^*|_A$$

folgt, motiviert sich aus diesen Überlegungen unter Beachtung von

$$C^{-t}\tilde{A}C^t = C^{-t}C^{-1}A =: W^{-1}A$$

(d.h., $W^{-1}A$ und $\tilde{A}$ haben dieselben Eigenwerte) der

Algorithmus 2.34. (Vorkonditioniertes CG-Verfahren)
Gegeben seien $b, x^0 \in \mathbb{R}^n$, eine positiv definite Matrix $A \in M(n,n)$ und ein geeigneter **Vorkonditionierer** $W \in M(n,n)$, i.e. W positiv definit. Berechnet wird die Lösung x^* von (46).

Initialisierung

$$r^0 = b - Ax^0$$
$$p^0 = W^{-1}r^0$$
$$\beta_0 = (p^0, r^0)$$

$i = 0$

Do While $i \leq n$ und $r^i \neq 0$

$\alpha_i = \beta_i/(p^i, Ap^i)$
$x^{i+1} = x^i + \alpha_i p^i$
$r^{i+1} = r^i - \alpha_i A p^i$
$q^{i+1} = W^{-1} r^{i+1}$
$\beta_{i+1} = (q^{i+1}, r^{i+1})$
$p^{i+1} = q^{i+1} + \frac{\beta_{i+1}}{\beta_i} p^i$
$i = i + 1$

end while

Die Kunst besteht jetzt darin, die Matrix W so zu wählen, daß

i) $\kappa(\tilde{A})$ nahe bei 1 bzw. $K(\tilde{A}) << K(A)$ und/oder
ii) $q = W^{-1} r$ leicht auflösbar

gewährleistet werden kann. Diese beiden Eigenschaften widersprechen sich natürlich im Allgemeinen und gesucht ist hier der Königsweg.

Abschließend werden noch einige Vorkonditionierer angegeben und ihre Eigenschaften vorgestellt.

Beispiel. 1. Diagonale Skalierung. Dabei wird

$$W := \text{ diag}(A) \tag{52}$$

gewählt.

2. SOR-Vorkonditionierung. Dabei wird ausgehend von der Zerlegung $A = L + D + U$

$$W := \frac{1}{2-\omega} \left(\frac{1}{\omega} D + L\right) \left(\frac{1}{\omega} D\right)^{-1} \left(\frac{1}{\omega} D + U\right) \tag{53}$$

gewählt. Im symmetrischen Fall heißt das SSOR Vorkonditionierung.

3. Vorkonditionierung mittels unvollständiger LR-Zerlegung. Dabei wird $A = M + R$ mit $M = \tilde{L}\tilde{R}$ und $R = A - \tilde{L}\tilde{R}$ angesetzt und

$$W := \tilde{L}\tilde{R} \quad (= \tilde{L} D \tilde{L}^t \text{ für symmetrische Matrizen}) \tag{54}$$

gewählt, vergleiche [10]. Zur Berechnung der Faktoren $\tilde{L}$ und $\tilde{R}$ wird die L-R-Zerlegung etwa nur auf den Nicht-Nullelementen von A durchgeführt.

Es gilt

Satz 2.35. (Konditionsverbesserung)
Für die SSOR-Vorkonditionierung gilt

$$\min_{0<w<2} \kappa(\tilde{A})(\omega) \leq \frac{1}{2} + \sqrt{\frac{1}{2}\kappa(A)}, \tag{55}$$

für die Vorkonditionierung mittels unvollständiger LR-Zerlegung

$$\kappa(\tilde{A}) \leq C\sqrt{\kappa(A)}, \tag{56}$$

falls die LR-Zerlegung nur auf den Nicht-Nullelementen der Ausgangsmatrix A durchgeführt wird.

Beweis: [10, (1.73 c)](1.73 c) für (55), (56) in [1]. ■

Beispiel. Wählen Sie

$$A = \frac{1}{h^2}\begin{bmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix}, \quad h = \frac{1}{n+1}.$$

Dann ist A positiv definit und Satz 2.23 liefert für die Eigenwerte

$$\begin{aligned} h^2\lambda_{max}(A) &= 2 - 2\cos\tfrac{\pi n}{n+1} && \leq 4 \\ h^2\lambda_{min}(A) &= 2 - 2\cos\tfrac{\pi}{n+1} &= \tfrac{\pi^2}{(n+1)^2} + \mathcal{O}\left(\tfrac{1}{n^4}\right), \end{aligned}$$

also mit (49)

$$\kappa(A) \approx \frac{4}{\pi^2}h^{-2}.$$

Benötigt demnach das CG-Verfahren k Iterationen zur Reduktion des Ausgangsfehlers auf ϵ Teile seiner Ausgangsgröße, so kommen die mit SSOR und unvollständiger LR-Zerlegung vorkonditionierten Varianten mit $\mathcal{O}(\sqrt{\kappa})$ Iterationen aus.

Bemerkung 2.36. (Mehrgittermethoden)
Mit der Hilfe von Mehrgittermethoden lassen sich Vorkonditionierer entwickeln, mit deren Hilfe die Anzahl der zur Lösung notwendigen Iterationen im CG-Verfahren unabhängig von der Dimension des Gleichungssystems gehalten werden kann. Eine Ausführliche Diskussion zu Mehrgitterverfahren gibt Hackbusch in [7].

Bemerkung 2.37. F"ur Gleichungssysteme mit beliebiger regul"arer Koeffizientenmatrix A konvergiert das sogenannte **GMRES Verfahren** nach h"ochstens n Iterationen. Eine ausf"uhrliche Besprechung dieses Verfahrens wird etwa in [12] gegeben. Für reguläre indefinite Matrizen, wie sie häufig bei Sattelpunktproblemen auftauchen, gibt es Varianten des CG-Verfahrens, welche nicht soviel Speicherplatz wie GMRES benötigen. Genannt seien hier

- **BiCG (Biconjugate Gradients)**
- **BiCG$_3$(Biconjugate Gradients mit 3 Matrixmultiplikationen)**
- **BiCGSTAB(Biconjugate Gradients Stabilisiert)**
- **CGS(Conjugate Gradient Squared)**
- **CSBCG(Composite Step Biconjugate Gradients)**
- **LAL(Look ahead Lanzcos)**

- **QMR(Quasi Minimal Residual)**
- **TFQMR(Transpose Free QMR)**
- **SymmLQ(Symmetrische LQ-Zerlegung)**

Eine vergleichende Besprechung dieser Verfahren findet etwa in [16, 6] statt.

3 Nichtlineare Gleichungen

Jetzt sollen numerische Methoden zur Lösung von nichtlinearen Gleichungen der Form

$$F(x) = 0, \quad x \in \mathbb{R}^n,\ F : \mathbb{R}^n \to \mathbb{R}^n \tag{57}$$

entwickelt und untersucht werden.

3.1 Motivation

Aufgabenstellungen dieser Art treten sehr häufig bei der Modellierung physikalischer Vorgänge auf. So ergibt sich etwa nach geeigneter Diskretisierung der nichtlinearen Randwertaufgabe

$$u''(x) = e^{u(x)} \text{ für } x \in (0,1), \text{ und } u(0) = u(1) = 1$$

zu $n \in \mathbb{N}$ über dem Gitter $x_i := ih (i = 0, \ldots, n+1)$ mit Gitterweite $h := \frac{1}{n+1}$ ein nichtlineares Gleichungssystem der Form

$$Au + Z(u) = 0,$$

wobei $u = [u_1, \ldots, u_n]^t$. Dabei gilt unter Verwendung zentraler Differenzen für die Approximation von $u''(x_i)$

$$A = \frac{1}{h^2} \begin{bmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix}, \text{ und } Z(u_1, \ldots, u_n) = \begin{bmatrix} e^{u_1} \\ \vdots \\ \vdots \\ e^{u_n} \end{bmatrix}.$$

Die Gleichung (57) soll numerisch mit Hilfe einer **Fixpunkt Iteration**

$$x^0 \in \mathbb{R}^n \text{ gegeben }, x^{i+1} = G(x^i),\ i = i + 1, \tag{58}$$

gelöst werden. Die Funktion $G : \mathbb{R}^n \to \mathbb{R}^n$ heißt dabei **Iterations Funktion**. Iterations Funktionen ergeben sich häufig direkt aus der Aufgabenstellung, etwa falls F die spezielle Gestalt

$$F(x) = x - H(x)$$

besitzt. In unserem Beispiel wäre etwa $H(x) = -A^{-1}Z(x)$. Dann ist die natürliche Wahl der Iterations Funktion $G := H$.

Die Grundlegende Idee zur numerischen Lösung von (57) und zur Konstruktion von Iterationsfunktionen besteht in der Formulierung einer Sequenz von *einfachen* Ersatzaufgaben. Dazu wird die Funktion F lokal durch geeignete einfach Modelle ersetzt, deren Nullstellen einfach zu berechnen sind. Das führt auf die Betrachtung von Linearisierungen der Funkion F in der Nähe einer Nullstelle ξ. Es gilt dort für gegebenes $x \in \mathbb{R}^n$ (F als stetig differenzierbar vorausgesetzt) nach der Taylor Formel

$$0 = F(\xi) = F(x) + DF(x)(\xi - x) + \mathrm{o}(\|\xi - x\|) \text{ für } (x \to \xi),$$

wobei $DF(x) \in M(n,n)$ die Jacobi Matrix von F bei x bezeichnet. Dabei Es liegt jetzt auf der Hand, die Nullstelle ζ der affin–linearen Funktion

$$T_1[F,x](z) := F(x) + DF(x)(z-x)$$

als Approximation der Nullstelle ξ von F aufzufassen, denn $T_1[F,z]$ ist sicherlich eine gute Approximation an (brauchbares Modell für) die Funktion F, falls x nahe bei ξ liegt. Wir erhalten

$$\eta = x - DF(x)^{-1}F(x).$$

Die Berechnung von η ist hier natürlich nur möglich, falls $DF(x)^{-1}$ existiert. Ausgehend von x^0 in der Nähe von ξ iterieren wir dieses Vorgehen. Das sich daraus ergebende Iterationsverfahren heißt **Newton Verfahren** und hat die Form

Algorithmus 3.1. (Newton Verfahren)

1. $x^0 \in \mathbb{R}^n$ gegeben, $i = 0$,
2. löse nach Δx^i,
$$DF(x^i)\Delta x^i = -F(x^i),$$
3. datiere auf
$$x^{i+1} = x^i + \Delta x^i,$$
4. $i = i + 1$, gehe zu 2.

Die Iterations Funktion des Newton Verfahrens ist gegeben durch

$$G(x) = x - DF(x)^{-1}F(x). \tag{59}$$

Die Wahl des Startwertes x^0 für das Newton Verfahren ist kritisch. In praktischen Aufgabenstellungen resultiert das nichtlineare Gleichungssystem (57) selber aus einer mathematischen Modellierung, so dass wir vielleicht bereits eine Idee haben, welche Vektoren (Werte) für ξ in Frage kommen.

Bei der Motivation des Newton Verfahrens werden die **Landau'schen** Symbole o (und später auch) O verwendet, vergleiche [12]. Diese sind wie folgt definiert.

Definition 3.2. Seien f, g zwei Funktionen. Wir sagen

$f(x) = \mathrm{O}(g(x))$ $(x \to x_0)$ (sprich f ist groß O von g für x gegen x_0) $\Longleftrightarrow \exists C > 0, U(x_0)$ $\forall x \in U(x_0)$: $|f(x)| \leq C|g(x)|$.

$f(x) = \mathrm{o}(g(x))$ $(x \to x_0)$ (sprich f ist klein O von g für x gegen x_0) $\Longleftrightarrow \forall \epsilon > 0$ $\exists U(x_0)$ $\forall x \in U(x_0)$: $|f(x)| \leq \epsilon|g(x)|$.

Aufgabe 3.3. Sei $f : \mathbb{R} \to \mathbb{R}$ 2 mal stetig differenzierbar. Konstruieren Sie das **Newton-Raphson Verfahren 2ten Grades** zur numerischen Berechnung von Nullstellen einer Funktion f, welches durch die Iterations Funktion

$$G(x) := x - \frac{2f(x)}{f'(x) + \mathrm{sign}(f'(x))\sqrt{f'(x)^2 - 2f(x)f''(x)}} \tag{60}$$

gegeben ist. Tip: Ersetzen Sie die Funktion f durch ein Modell 2ter Ordnung, vergl. Motivation von Algorithmus 3.1.

3.2 Verfahren und Konvergenzsätze

Jetzt zu Konvergenzaussagen für Fixpunkt Iterationen. Dazu zunächst einige Begriffe.

Definition 3.4.

- ξ heißt **Fixpunkt** von $G :\Longleftrightarrow G(\xi) = \xi$.
- Sei ξ Fixpunkt von G und es gelte für alle Startvektoren x^0 für die mittels der Fixpunkt Iteration (58) erzeugten Iterierten $\{x^i\}_{i\in\mathbb{N}}$ für ein festes $p \geq 1$

  $$\|x^{i+1} - \xi\| \leq C\|x^i - \xi\|^p \text{ für alle } i \geq 0 \text{ und } C < 1 \text{ falls } p = 1.$$

  Dann wird das durch G erzeugte Iterations Verfahren als Verfahren von mindestens p-ter Ordnung bezeichnet.

Aus dieser Definition folgt sofort

Satz 3.5. Jedes Verfahren p-ter Ordnung zur Bestimmung eines Fixpunktes ξ ist lokal konvergent, d.m. es gibt eine Umgebung $U(\xi)$ derart, daß für alle Startwerte $x^0 \in U(\xi)$ die durch die Fixpunktiteration (58) erzeugten Iterierten $\{x^i\}_{i\in\mathbb{N}}$ gegen ξ konvergieren.

Beweis: Im Fall $p > 1$ wähle $U(\xi) := \{x; \|x - \xi\| < \delta\}$ so, daß $\|x - \xi\|C^{\frac{1}{p-1}} < \kappa < 1$ für alle $x \in U(\xi)$ gültig ist, also $\delta = C^{-\frac{1}{p-1}}$, wobei C die Konstante in der Definition der Ordnung bezeichnet. Das kann wie folgt eingesehen werden; Für die Iterierten gilt

$$\|x^{i+1} - \xi\| \leq C\|x^i - \xi\|^p \leq \cdots \leq C^{1+p+p^2+\cdots+p^i}\|x^0 - \xi\|^{p^{i+1}} = \\ = C^{-\frac{1}{p-1}}\left(C^{\frac{1}{p-1}}\|x^0 - \xi\|\right)^{p^{i+1}} \to 0 \ (i \to \infty),$$

falls $C^{\frac{1}{p-1}}\|x^0 - \xi\| < 1$ und $p > 1$. Im Fall $p = 1$ gilt $C < 1$, also

$$\|x^{i+1} - \xi\| \leq C\|x^i - \xi\| \leq \cdots \leq C^{i+1}\|x^0 - \xi\| \to 0 \ (i \to \infty),$$

so dass Konvergenz bei jeder Wahl des Startwertes vorliegt. ■

Wir sprechen im Fall $p = 1$ von **linearer Konvergenz**, und von **superlinearer Konvergenz**, falls

$$\|x^{i+1} - \xi\| \leq C_i\|x^i - \xi\| \text{ für alle } i \geq 0 \text{ mit } \lim_{i\to\infty} C_i = 0.$$

Für skalare Iterationen gilt

Satz 3.6. In (58) gelte $n = 1$. Ferner besitze G den Fixpunkt ξ, sei in einer Umgebung von ξ p-mal stetig differenzierbar und es gelte $G^{(k)}(\xi) = 0$ für $k = 1, \ldots, p-1$, aber $G^{(p)}(\xi) \neq 0$. Dann liegt für $p > 1$ ein Verfahren p-ter Ordnung vor. Ein Verfahren erster Ordnung liegt vor, falls zusätzlich zu $p = 1$ noch $|G'(\xi)| < 1$ gilt.

Der Beweis ist Gegenstand von

Aufgabe 3.7.

1. Beweisen Sie Satz 3.6.

2. Weisen Sie nach, daß das Newton Verfahren für skalare Funktionen $f : \mathbb{R} \to \mathbb{R}$ lokal mindestens von 2ter Ordnung konvergiert, sofern für die entsprechende Nullstelle ξ (der Fixpunkt der Verfahrens Funktion) $f'(\xi) \neq 0$ gilt.

Hinreichend für die lokale Konvergenz von Fixpunkt Iterationen ist die Kontraktionseigenschaft der Iterations Funktion G.

Definition 3.8. $G : \mathbb{R}^n \to \mathbb{R}^n$ heißt **stark kontrahierend** in einer Umgebung $U(\zeta) :\Longleftrightarrow$

$$\|G(x) - G(y)\| \leq K\|x - y\| \text{ für alle } x, y \in U(\zeta) \tag{61}$$

mit einem $K < 1$, bzw. **schwach kontrahierend** in einer Umgebung $U(\zeta) :\Longleftrightarrow$

$$\|G(x) - G(y)\| < \|x - y\| \text{ für alle } x, y \in U(\zeta). \tag{62}$$

Kontrahierend meint im Folgenden immer stark kontrahierend.

Die Fixpunkt Iteration einer kontrahierenden Abbildung ist in der Umgebung eines Fixpunktes konvergent mit mindestens Ordnung 1, wie der folgende Satz zeigt.

Satz 3.9. Die Funktion $G : \mathbb{R}^n \to \mathbb{R}^n$ besitze einen Fixpunkt ξ und sei in einer Umgebung $B_r(\xi) := \{x \in \mathbb{R}^n; \|x - \xi\| < r\}$ kontrahierend. Dann besitzt die Iterierten Folge $\{x^i\}_{i\in\mathbb{N}}$ der Fixpunkt Iteration (58) für jeden Startwert $x^0 \in B_r(\xi)$ die Eigenschaften

- $x^i \in B_r(\xi)$ für alle $i \in \mathbb{N}$,
- $\|x^i - \xi\| \leq K^i\|x^0 - \xi\|$ (was bedeutet, daß $\{x^i\}_{i\in\mathbb{N}}$ mindestens linear gegen ξ konvergiert).

Der Beweis dieses Satzes ist Gegenstand von

Aufgabe 3.10. Beweisen Sie Satz 3.9.

Bei lokal kontrahierenden Abbildungen kann aber auch auf die Existenz eines Fixpunktes geschlossen werden. Neben anderen Dingen besagt das der

Satz 3.11. Sei $x^0 \in \mathbb{R}^n$. Die Funktion $G : \mathbb{R}^n \to \mathbb{R}^n$ sei

i. in einer Umgebung $B_r(x^0) := \{x \in \mathbb{R}^n; \|x - x^0\| < r\}$ kontrahierend mit der Kontraktionskonstanten K,

ii. $\|x^1 - x^0\| = \|G(x^0) - x^0\| \leq (1 - K)r < r$.

Dann gilt für die Iterierten Folge $\{x^i\}_{i\in\mathbb{N}}$ der Fixpunkt Iteration (58) mit Startwert x^0

1. $x^i \in B_r(x^0)$ für alle $i \in \mathbb{N}$,
2. G besitzt in $\bar{B}_r(x^0)$ genau einen Fixpunkt ξ mit $\lim_{x\to\infty} x^i = \xi$ und

3. es gelten die Fehlerabschätzungen

$$\|x^{i+1} - \xi\| \leq K\|x^i - \xi\| \quad \text{und} \quad \|x^i - \xi\| \leq \frac{K^i}{1-K}\|x^1 - x^0\|. \tag{63}$$

Es ist zu beachten, dass im vorhergehenden Satz die Existenz eines Fixpunktes nicht vorausgesetzt, sondern gefolgert wird.

Beweis: (von Satz 3.11) 1. Wir zeigen $x^i \in B_r(x^0)$ für alle $i \in \mathbb{N}$ mit Hilfe von Induktion. Aus der Voraussetzung ii. folgt $x^1 \in B_r(x^0)$. Gelte also $x^j \in B_r(x^0)$ für $1 \leq j \leq i$. Aus i. erhalten wir sofort

$$\|x^{i+1} - x^i\| = \|G(x^i) - G(x^{i-1})\| \leq K\|x^i - x^{i-1}\| \leq K^i\|x^1 - x^0\|.$$

Mit Hilfe der Dreiecksungleichung und ii. folgt dann

$$\|x^{i+1} - x^0\| \leq \|x^1 - x^0\| \sum_{j=0}^{i} K^j \leq r(1-K)\sum_{j=0}^{i} K^j = (1 - K^{i+1})r < r,$$

also $x^{i+1} \in B_r(x^0)$.

2. Wir zeigen, daß $\{x^i\}_{i\in\mathbb{N}}$ Cauchy Folge ist. Dazu verwende die Dreiecksungleichung, ii., die geometrische Reihe und schließe für $m > l$

$$\|x^m - x^l\| \leq \|x^1 - x^0\|K^l \sum_{j=0}^{m-l-1} K^j < \frac{K^l}{1-K}\|x^1 - x^0\| < K^l r \to 0 \text{ für } m, l \to \infty.$$

Demnach ist $\{x^i\}_{i\in\mathbb{N}}$ Cauchy Folge und konvergiert wegen $x^i \in B_r(x^0)$ gegen ein $\xi \in \bar{B}_r(x^0)$. Dieses ξ ist Fixpunkt von G, denn

$$\|G(\xi) - \xi\| \leq \|G(\xi) - G(x^i)\| + \|G(x^i) - \xi\| \leq K\|x^i - \xi\| + \|x^{i+1} - \xi\| \to 0 \text{ für } i \to \infty.$$

Der Fixpunkt ist eindeutig, denn ist $\bar{\xi}$ ein weiterer Fixpunkt in $\bar{B}_r(x^0)$. Dann folgt $\bar{\xi} = \xi$ aus

$$\|\bar{\xi} - \xi\| = \|G(\bar{\xi}) - G(\xi)\| \leq K\|\bar{\xi} - \xi\|.$$

Schließlich noch die Fehlerabschätzungen;

$$\|\xi - x^l\| = \lim_{m\to\infty} \|x^m - x^l\| \leq \frac{K^l}{1-K}\|x^1 - x^0\|$$

und

$$\|\xi - x^{i+1}\| = \|G(\xi) - G(x^i)\| \leq K\|\xi - x^i\|.$$

Damit ist alles bewiesen. ■

Wir wollen jetzt das Newton Verfahren in seiner Basis Variante aus Algorithmus 3.1 untersuchen und beweisen zunächst

Satz 3.12. Sei $F : S \to \mathbb{R}^n$ stetig und auf der offenen Menge C mit $\bar{C} \subseteq S$ differenzierbar. Ferner gebe es zu $x^0 \in C$ positive Konstanten r, a, b, c, h mit den Eigenschaften

i. $B_r(x^0) = \{x \in \mathbb{R}^n; \|x - x^0\| < r\} \subseteq C$,

ii. $h := \frac{abc}{2} < 1$ und

iii. $r := \frac{a}{1-h}$.

Die Funktion F besitze die Eigenschaften

1) $\|DF(x) - DF(y)\| \leq c\|x - y\|$ für alle $x, y \in C$ (Lipschitz Stetigkeit der Ableitung),

2) $DF(x)^{-1}$ existiert für alle $x \in C$ und es gilt $\|DF(x)^{-1}\| \leq b$ und

3) $\|DF(x^0)^{-1}F(x^0)\| \leq a$.

Dann gilt

1. Ausgehend von x^0 ist die Folge der Newton Iterierten $\{x^i\}_{i\in\mathbb{N}}$ aus Algorithmus 3.1 wohldefiniert und es gilt $x^i \in B_r(x^0)$ für alle $i \in \mathbb{N}$.
2. $\lim_{i\to\infty} x^i = \xi \in \bar{B}_r(x^0)$ und es gilt $F(\xi) = 0$.
3. Es gilt die Fehlerabschätzung
$$\|x^i - \xi\| \leq a\frac{h^{2^i-1}}{1-h^{2^i}} \text{ für alle } i \in \mathbb{N}. \tag{64}$$
4. Das Verfahren ist lokal quadratisch konvergent, d.m. für die Iterierten gilt
$$\|x^{i+1} - \xi\| \leq K\|x^i - \xi\|^2$$
mit einer positiven Konstanten K.

Unter anderem ist das Newton Verfahren wegen $0 < h < 1$ dann mindestens quadratisch konvergent.

Beweis: 1. Wir schreiben Algorithmus 3.1, 2. um in die Form
$$x^{i+1} = x^i - DF(x^i)^{-1}F(x^i).$$
Sind die Iterierten x^j $(j = 0, \ldots, i)$ in $B_r(x^0)$, so ist x^{i+1} wegen Voraussetzung 2) wohldefiniert. Mit Voraussetzung 3) ist diese Aussage richtig für x^0, x^1. Seien also x^j $(j = 0, \ldots, i)$ in $B_r(x^0)$ für ein $i \geq 1$. Dann folgt aus 2) und der Iterationsvorschrift des Newton Verfahrens
$$\|x^{i+1}-x^i\| = \|DF(x^i)^{-1}F(x^i)\| \leq b\|F(x^i)\| = b\|F(x^i) \underbrace{-F(x^{i-1}) - DF(x^{i-1})(x^i - x^{i-1})}_{=0}\|.$$
Der Mittelwertsatz in Integralform liefert $(x := x^i, y := x^{i-1})$
$$F(x) - F(y) - DF(y)(x - y) = \int_0^1 DF(y + s(x - y))ds(x - y) - DF(y)(x - y)$$
$$= \int_0^1 (DF(y + s(x - y)) - DF(y))ds(x - y),$$

also

$$\|F(x)-F(y)-DF(y)(x-y)\| \leq \int_0^1 \|DF(y+s(x-y))-DF(y)\|ds\|x-y\| \leq c\int_0^1 sds\|x-y\|^2$$

wegen 1) und der Konvexität von C. Damit ergibt sich dann induktiv

$$\|x^{i+1}-x^i\| \leq \frac{cb}{2}\|x^i-x^{i-1}\|^2 \leq ah^{2^i-1}$$

wegen der Definition von h. Dies' wiederum liefert mit Hilfe der Dreiecksungleichung und er geometrischen Reihe

$$\|x^{i+1}-x^0\| \leq \sum_{j=0}^{i}\|x^{j+1}-x^j\| \leq a\sum_{j=0}^{i}h^{2^j-1} < \frac{a}{(1-h)} = r,$$

also $x^{i+1} \in B_r(x^0)$.

2. und 3. Wie in 2. des Beweises von Satz 3.11 wird jetzt bewiesen, daß $\{x^i\}_{i\in\mathbb{N}}$ eine Cauchy Folge ist und daher konvergiert. Es gilt nämlich für $m \geq n$

$$\|x^{m+1}-x^n\| \leq \sum_{j=n}^{m}\|x^{j+1}-x^j\| \leq ah^{2^n-1}\sum_{j=0}^{\infty}\left(h^{2^n}\right)^j < \frac{ah^{2^n-1}}{(1-h^{2^n})} < \epsilon \text{ für } n \geq n_0(\epsilon),$$

weil $0 < h < 1$. Damit gilt

$$\lim_{i\to\infty} x^i = \xi \in \bar{B}_r(x^0)$$

und auch als Folgerung aus der vorangegangenen Abschätzung

$$\lim_{m\to\infty}\|x^{m+1}-x^n\| = \|\xi-x^n\| \leq \frac{ah^{2^n-1}}{(1-h^{2^n})}.$$

Das ist die gewünschte Fehlerabschätzung. Verbleibt der Nachweis, daß ξ Nullstelle von F ist. Dazu schließe aus 1) und $x^1 \in B_r(x^0)$ für alle $i \in \mathbb{N}$

$$\|DF(x^i)\| \leq \|DF(x^0)\|+\|DF(x^i)-DF(x^0)\| \leq \|DF(x^0)\|+c\|x^i-x^0\| < cr+\|DF(x^0)\| =: K.$$

Damit

$$\|F(x^i)\| = \|-DF(x^i)(x^{i+1}-x^i)\| \leq K\|x^{i+1}-x^i\| \leq Kah^{2^i-1} \to 0 \text{ für } i\to\infty.$$

Da F auf S stetig ist gilt auch

$$\lim_{i\to\infty} F(x^i) = F(\xi),$$

also $F(\xi) = 0$.

Wir zeigen noch die quadratische Konvergenz. Dazu schreibe

$$F(x^i) = F(x^i)-F(\xi) = \int_0^1 (DF(\xi+s(x^i-\xi))-DF(x^i))ds(x^i-\xi)+DF(x^i)(x^i-\xi).$$

Damit ergibt sich

$$x^i - x^{i+1} = DF(x^i)^{-1}F(x^i) = x^i - \xi + DF(x^i)^{-1}\int_0^1 (DF(\xi + s(x^i - \xi)) - DF(x^i))ds(x^i - \xi),$$

also mit der Lipschitz Stetigkeit der Ableitungen

$$\|\xi - x^{i+1}\| \leq \|DF(x^i)^{-1}\|\frac{c}{2}\|\xi - x^i\|^2 \leq \frac{bc}{2}\|\xi - x^i\|^2,$$

das ist die Behauptung mit $K := \frac{bc}{2}$. ■

Eine wichtige Eigenschaft der Gleichung (57) ist deren Invarianz gegenüber affinen Transformationen. Denn es gilt ja

$$F(x) = 0 \quad \Longleftrightarrow \quad AF(x) = 0 \text{ für alle Matrizen } A \in GL(n), \tag{65}$$

d.m. Transformationen mit invertierbaren Matrizen ändern die Lösungsmenge nicht. Eine numerische Lösungsmethode sollte diese Invarianz dann auch berücksichtigen, also **konservativ** hinsichtlich der **affinen Invarianz** der Gleichung (57) sein. Das Newton Verfahren erfüllt diese Eigenschaft, denn es gilt

$$H(x) := AF(x) \;\Rightarrow DH(x)^{-1}H(x) = DF(x)^{-1}A^{-1}AF(x) = DF(x)^{-1}F(x).$$

Ein Konvergenz Beweis des Newton Verfahrens sollte daher nur mit Bedingungen formuliert werden, die auch affin invariant sind. In dieser Hinsicht waren wir bei der Formulierung des Satzes 3.12 nicht konsequent, wie das folgende Beispiel zeigt.

Beispiel. Betrachte

$$F(x) := \begin{bmatrix} x_1 - x_2 \\ (x_1 - 8)x_2 \end{bmatrix}$$

mit den beiden Nullstellen

$$x^* = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \text{ und } y^* = \begin{bmatrix} 8 \\ 8 \end{bmatrix}.$$

Wir formulieren unsere Aufgabenstellung (finde x mit $F(x) = 0$) in

$$S := \{x = (x_1, x_2); -1 < x_i < 3, i = 1, 2\} \ni x^*,$$

und schließen damit y^* von der Suche aus. Als Startwert für das Newton Verfahren wählen wir

$$x^0 = \begin{bmatrix} \frac{1}{8} \\ \frac{1}{8} \end{bmatrix}$$

und messen in der Maximum Norm. Dann gilt

- $a_F = \|DF(x^0)^{-1}F(x^0)\| = 0.127$,
- $b_F = \sup_S \|DF(x)^{-1}\| = 3$ und
- $c_F = \sup_S \frac{\|DF(x) - DF(y)\|}{\|x - y\|} = 2$.

Das Konvergenzkriterium $h_F = a_F b_F c_F/2 < 1$ ist hier wegen $h = 0.381$ erfüllt. Betrachte jetzt

$$H(x) := \begin{bmatrix} 1 & 1 \\ 0 & \frac{1}{2} \end{bmatrix} F(x).$$

Dann gilt

$$a_H = a_F,\ b_H = 8.5 \text{ und } c_H = 2 \ \Rightarrow\ h_H = 1.0795 > 1.$$

D.m. daß obwohl das Newton Verfahren invariant unter affinen Transformationen ist, gilt das nicht für die theoretische Charakterisierung in Satz 3.12.

Wir formulieren jetzt noch einen lokalen Konvergenzsatz für das Newton Verfahren, der nur affin invariante Voraussetzungen benötigt und zusätzlich auch die lokale Einzigkeit der Nullstelle ξ gewährleistet, siehe [15].

Satz 3.13. (Newton-Kantorovich)
Sei $F : \mathbb{R}^n \supseteq S \to \mathbb{R}^n$ sei stetig differenzierbar, S offen und $x^0 \in S$ mit $DF(x^0)^{-1}$ existent. Ferner nehme an, daß Konstanten $a > 0$, $\omega_0 > 0$ existieren mit den Eigenschaften

i. $\|DF(x^0)^{-1}F(x^0)\| \leq a$,

ii. $\|DF(x^0)^{-1}(DF(x) - DF(y))\| \leq \omega_0\|x - y\|$ für alle $x, y \in S$

iii. $h_0 := a\omega_0 \leq \frac{1}{2}$ und

iv. $\bar{B}_r(x^0) \subset S$, $r := \frac{1-\sqrt{1-2h_0}}{\omega_0}$.

Dann gilt

1. Ausgehend von x^0 ist die Folge der Newton Iterierten $\{x^i\}_{i\in\mathbb{N}}$ aus Algorithmus 3.1 wohldefiniert und es gilt $x^i \in B_r(x^0)$ für alle $i \in \mathbb{N}$.
2. $\lim_{i\to\infty} x^i = \xi \in \bar{B}_r(x^0)$ und es gilt $F(\xi) = 0$.
3. Die Lösung ξ ist eindeutig in
$$\bar{B}_r(x^0).$$

Die Konvergenz der Iteriertenfolge $\{x^i\}_{i\in\mathbb{N}}$ ist mindestens quadratisch.

Beweis: Siehe [15]. ■

Aufgabe 3.14. Prüfen Sie für Beispiel 3.2, ob die Voraussetzungen von Satz 3.13 affin invariant sind.

Eine vereinfachte Variante des Newton Verfahrens aus Algorithmus 3.1 verzichtet in Schritt 2. auf die Verwendung jeweils der exakten Jacobi Matrix $DF(x^i)$. Statt dessen wird die Jacobi Matrix aus einem der vorherigen Schritte eingefroren. Es lohnt sich dann, diese Matrix zu zerlegen, da in Schritt 2. (unten Schritt 3.) immer Gleichungssysteme mit derselben Koeffizientenmatrix zu lösen sind.

Algorithmus 3.15. (Vereinfachtes Newton Verfahren)

1. $x^0 \in \mathbb{R}^n$ gegeben, $i = 0$, $iaufdat \geq 1$.
2. Falls $\frac{i}{iaufdat} \in \mathbb{N} \cup \{0\}$ $A := DF(x^i)$
3. löse nach Δx^i,
$$A\Delta x^i = -F(x^i),$$
4. datiere auf
$$x^{i+1} = x^i + \Delta x^i,$$
5. $i = i + 1$, gehe zu 2.

Für diesen Algorithmus gelten lokale Konvergenzaussagen ähnlich denen aus den Sätzen 3.12 und 3.13, die Konvergenzrate ist jedoch i.d.R. nur linear, siehe dazu [15].

Weitere Varianten des Newton Verfahrens sind die sogenannten **Inexakten Newton Verfahren** und **Quasi Newton Verfahren**. Die Idee des inexakten Newton Verfahrens besteht darin, Schritt 2. des Newton Verfahrens 3.1 selbst iterativ zu lösen und die Genauigkeit dieser **inneren Iteration** an das Residuum der äußeren Iteration anzupassen. Diese Verfahren(sklassen) werden etwa in Grundvorlesungen zur Optimierung besprochen.

4 Interpolation

Interpolation von Funktionen und allgemeiner Daten ist ein häufig auftretendes Problem sowohl in der Mathematik als auch in vielen Anwendungen.

Dateninterpolation: Gegen seien Daten (x_i, f_i) $(i = 0, \ldots, n)$. Gesucht ist eine Funktion $p(x)$ mit

$$p(x_i) = f_i \quad i = 0, \ldots, n. \tag{66}$$

Daten könnten z.B. aus Messungen eines physikalischen Vorgangs erhalten werden, von dem wir annehmen wollen, dass er von einer kontinuierlichen Funktion f beschrieben wird (die wir leider nicht explizit kennen). Mit Hilfe der Daten wollen wir ein möglichst einfaches **Modell** p der Funktion f bestimmen, welches in den Daten mit f übereinstimmt, d.h. es soll

$$p(x_i) = f(x_i) \ (\equiv f_i) \text{ für alle } i = 0, \ldots, n$$

erfüllt sein. Das Modell p können wir dann dazu benutzen,

- an Stellen $x \neq x_i$ Näherungswerte $p(x)$ für $f(x)$ zu berechnen, oder auch
- mit Hilfe des Modells p **Ersatzaufgaben** zu formulieren, etwa für die Bestimmung von Nullstellen der Funktion f, d.m. die Aufgabenstellung *Finde* x^* *mit* $f(x^*) = 0$ wird ersetzt durch *Finde* x^* *mit* $p(x^*) = 0$.

Diesen Prozesse wollen wir natürlich zuverlässig gestalten. Daher ist es notwendig,

- das Modell p für beliebige (sinnvolle) Werte x effizient und exakt auswerten zu können, und
- den durch das Modell induzierten Fehler $\|p(x) - f(x)\|$ qualitativ zu beherrschen und nach Möglichkeit auch kontrollieren zu können.

Interpolieren können wir auf mannigfaltige Weise. Wir fangen an mit

4.1 Polynominterpolation

Wir wollen das Problem der Dateninterpolation zunächst mit **Polynomen** lösen, d.m. Funktionen der Form

$$p(x) := \sum_{i=0}^{m} a_i x^i \quad \text{(Polynom } m - \text{ten Grades)}. \tag{67}$$

Dabei bezeichnen $a_0, \ldots, a_m$ die (hier i.d.R. reellen) Koeffizienten. Wir setzen voraus, dass der Koeffizient $a_m \neq 0$ erfüllt, das Polynom also wirklich den Grad m besitzt. Polynome werden in der Algebra studiert, insbesondere “uber dem Körper der *Komplexen Zahlen*.

Bevor die eigentliche Aufgabenstellung mit der Konstruktion von **Interpolationspolynomen** angegangen wird, wollen wir effiziente Algorithmen zur Auswertung von Polynomen (und deren Ableitungen) angeben und noch einige nützliche Eigenschaften von Polynomen zusammentragen.

Eine elegante Methode zur Auswertung von Polynomen ist das Hornerschema. Dieses wird zunächst anhand des nachfolgenden Beispiels demonstiert.

Beispiel.

$$
\begin{array}{llcllllllllll}
n=4: & p(x) & = & 1 & + & 2x & + & 3x^2 & + & 4x^3 & + & 5x^4 & \\
 & & = & 1 & + & x(2 & + & 3x & + & 4x^2 & + & 5x^3\,) & \\
 & & = & 1 & + & x(2 & + & x(3 & + & 4x & + & 5x^2\,)) & \\
 & & = & 1 & + & x(2 & + & x(3 & + & x(4 & + & 5x\,))) & \\
 & & = & a_0 & + & x(a_1 & + & x(a_2 & + & x(a_3 & + & x\,a_4\,))) & \\
 & & = & a_{n-4} & + & x(a_{n-3} & + & x(a_{n-2} & + & x(\underbrace{a_{n-1} & + & \underbrace{x\,a_n}_{x\,b_{n-1}}}_{}\,))) &
\end{array}
$$

$$x(\underbrace{a_{n-2} + x \cdot b_{n-2}}_{})$$

$$= a_{n-4} + x(\underbrace{a_{n-3} + x \cdot b_{n-3}}_{})$$

$$= \underbrace{a_{n-4} + x \cdot b_{n-4}}_{b_{n-5}}$$

Diese Auswertung benötigt nur 4 Multiplikationen und 4 Additionen.

Wir fassen diese Klammer– und Multiplikationsmethode im sog. Horner–Schema zusammen:

Algorithmus 4.1. (Horner Schema zur Auswertung von $p(\hat{x})$)

$b_{n-1} := a_n,$

Für $i = n-1, \ldots, 0$ führe aus: $b_{i-1} := b_i \hat{x} + a_i,$

$p(\hat{x}) := b_{-1}.$

Schematisch:

$$
\begin{array}{ccccccc}
a_n & a_{n-1} & a_{n-2} & a_{n-3} & \cdots & a_1 & a_0 \\
\downarrow + & \downarrow + & \downarrow + & \downarrow + & & \downarrow + & \downarrow + \\
0 \nearrow & \hat{x}\,b_{n-1} \nearrow & \hat{x}\,b_{n-2} \nearrow & \hat{x}\,b_{n-3} \nearrow & \cdots & \hat{x}\,b_1 \nearrow & \hat{x}\,b_0 \\
\hline
b_{n-1} & b_{n-2} & b_{n-3} & b_{n-4} & & b_0 & b_{-1} \\
\| & & & & & & \| \\
a_n & & & & & & p(\hat{x})
\end{array}
$$

Wir halten fest

Satz 4.2. Sei $p_n(x) = \sum\limits_{i=0}^{n} a_i\, x^i$ ein beliebiges Polynom n–ten Grades $(n \in \mathbb{N})$ und $\hat{x} \in \mathbb{C}$ beliebig aber fest. Dann ist das Polynom $p_{n-1}(x) := \sum\limits_{i=0}^{n-1} b_i\, x^i$, dessen Koeffizienten b_i mit Hilfe des Horner–Schemas aus p_n und $\hat{x}$ berechnet werden, vom Grad $n-1$, und es gilt

$$p_n(x) = p_n(\hat{x}) + (x - \hat{x})\, p_{n-1}(x). \tag{68}$$

Durch Differenzieren von (68) erhalten wir

$$\begin{aligned} & p_n'(x) &=& \; p_{n-1}(x) \; + \; (x-\hat{x})\, p_{n-1}'(x)\,, \\ \text{also} \qquad & p_n'(\hat{x}) &=& \; p_{n-1}(\hat{x})\,. \end{aligned}$$

Dies liefert die

Korollar 4.3. Die Ableitung $p_n'(\hat{x})$ eines beliebigen Polynoms $p_n \in \Pi_n$ an einer vorgegebenen Stelle $\hat{x}$ ist als Polynomwert von $p_{n-1} \in \Pi_{n-1}$ berechenbar, wobei p_{n-1} aus p_n mittels Horner–Schema berechnet wurde.

Dies kann mit Hilfe des **doppelten Horner–Schemas** ausgeführt werden, das nur an einem Beispiel erläutert werden soll.

Beispiel. $p(x) = 1 - 2x + 3x^2 - 4x^3 + 5x^4, \qquad \hat{x} = 2$

$$\begin{array}{lccccc} & 5 & -4 & 3 & -2 & 1 \\ & (a_4) & (a_3) & (a_2) & (a_1) & (a_0) \\ & \downarrow + & \downarrow + & \downarrow + & \downarrow + & \downarrow + \\ \hat{x} = 2 & & 2 \cdot 5 \nearrow & 2 \cdot 6 \nearrow & 2 \cdot 15 \nearrow & 2 \cdot 28 \nearrow \\ \hline & 5 & 6 & 15 & 28 & 57 \quad = \quad p(2) \\ & (b_3) & (b_2) & (b_1) & (b_0) & (b_{-1}) \end{array}$$

$$\begin{array}{lcccc} & & 2 \cdot 5 & 2 \cdot 16 & 2 \cdot 47 \\ \hline & 5 & 16 & 47 & 122 \quad = \quad p'(2) \end{array}$$

Dieses Schema kann auf die Berechnung höherer Ableitungen erweitert werden. Vorsicht: Hierbei treten noch Faktoren auf (vgl. z.B. [11, Korollar 2.5]).

Eine einfache Folgerung aus Satz 4.2 lautet

Korollar 4.4. Ist $\hat{x}$ eine Nullstelle von $p_n(x)$, so zeigt Satz 4.2

$$p_n(x) = (x - \hat{x}) p_{n-1}(x). \tag{69}$$

Man kann mit dem Horner–Schema also einen Linearfaktor $(x - \hat{x})$ abspalten.

Definition 4.5. Hat ein Polynom $p \in \Pi_n$ eine Darstellung

$$p(x) = (x - \hat{x})^k \, q(x), \qquad k \in \mathbb{N},$$

mit einem Polynom $q \in \Pi_{n-k}$, für welches $q(\hat{x}) \neq 0$ erfüllt sei, so heißt $\hat{x}$ *k–fache Nullstelle* von p, und k heißt *algebraische Vielfachheit* der Nullstelle $\hat{x}$.

Unter Benutzung dieser Begriffsbildungen erhalten wir aus Satz 4.2 den fundamentalen

Hilfsatz 4.6. Ein Polynom $p_n \in \Pi_n$ mit $n+1$ Nullstellen (wobei mehrfache Nullstellen zugelassen sind: eine k–fache Nullstelle zählt dann als k Nullstellen) verschwindet identisch, d.h. $p_n(x) = 0$ für alle $x \in \mathbb{C}$. (Oder äquivalent: $p(x) = \sum_{i=0}^{n} a_i \, x^i$ und $a_i = 0$ für alle i)

Andere Formulierung: Ein Polynom p_n $n-$ten Grades hat höchstens n Nullstellen.

Beweis: durch vollständige Induktion (zunächst für paarweise verschiedene Nullstellen):
Induktionsanfang: $n = 0$, also $p_0(x) = a_0$, ist $p_0(\hat{x}) = 0$ für ein $\hat{x} \in \mathbb{C}$ so folgt $a_0 = 0$.

Induktionsvoraussetzung: Die Behauptung sei richtig für $n - 1$.

Induktionsschritt: $p_n \in \Pi_n$ habe die $n+1$ (paarweise verschiedenen) Nullstellen $x_1, \ldots, x_{n+1}$; dann gilt nach Satz 4.2

$$p_n(x) \;=\; p_n(x_{n+1}) + (x - x_{n+1})\, p_{n-1}(x) \;=\; (x - x_{n+1})\, p_{n-1}(x),\;\; p_{n-1} \in \Pi_{n-1}$$

p_{n-1} hat die Nullstellen $x_1, \ldots, x_n$, also ist $p_{n-1} \equiv 0$ nach Induktionsvoraussetzung.

Hat p_n mehrfache Nullstellen, so hat es nach Def. 4.5 eine Darstellung

$$p(x) \;=\; \left(\prod_{j=1}^{\ell} (x - x_j)^{k_j} \right) q_{n-m}(x), \quad m \;=\; \sum_{j=1}^{\ell} k_j, \quad k_j \;>\; 1, \quad q_{n-m} \;\in\; \Pi_{n-m}$$

mit einem Polynom q_{n-m}, das keine mehrfachen Nullstellen besitzt und auf das der vorige Beweis angewendet werden kann. ■

Die Ersatzaufgabe *Finde eine Nullstelle x^* von p* können wir mit dem Newton Verfahren aus Kapitel 3 numerisch lösen. Die Vorschrift lautet hier

$$x^0 \text{ gegeben, berechne } x^{i+1} = x^i - \frac{p_n(x^i)}{p_n'(x^i)},$$

wobei Nenner und Zähler der Iterationsvorschrift mit Hilfe des Horner–Schemas berechnet werden können. Die Auswahl des Startwertes x^0 wird bei Polynomen erleichtert durch

Satz 4.7. *(Einschließungssatz für Polynomnullstellen)* Hat das Polynom $p(x) = \sum_{i=0}^{n} a_i\, x^i$ mit $a_n = 1$ eine Nullstelle x^*, d.h. $p(x^*) = 0$, so gilt

a) $\quad |x^*| \;\leq\; \max\left(1, \sum_{i=0}^{n-1} |a_i|\right)$

b) $\quad |x^*| \;\leq\; \max_{i=1,\ldots,n-1} \left(|a_0|,\; 1 + |a_i|\right)$

Beweis: Wir zeigen: Außerhalb dieser Schranken gilt immer $|p(x)| > 0$.

a) Sei $|x| > \max\left(1, \sum_{i=0}^{n-1} |a_i|\right)$. Es gilt wegen $a_n = 1$ unter Anwendung der umgekehrten Dreiecksungleichung

$$\begin{aligned} |p(x)| \;=\; \left|x^n + \sum_{i=0}^{n-1} a_i\, x^i\right| \qquad & |x^n| \;-\; \sum_{i=0}^{n-1} |a_i\, x^i| \;=\; |x^n| \;-\; \sum_{i=0}^{n-1} |a_i|\, |x^i| \\ \geq \qquad & |x^n| \;-\; \sum_{i=0}^{n-1} |a_i|\, |x^{n-1}|\,, \quad \text{denn} \quad |x| \;>\; 1\,; \\ = \qquad & |x^{n-1}| \left(|x| \;-\; \sum_{i=0}^{n-1} |a_i|\right) \;>\; 0\,, \end{aligned}$$

denn $|x^{n-1}| > 1$ wegen $|x| > 1$ und $|x| - \sum_{i=0}^{n-1} |a_i| > 0$ nach obiger Voraussetzung.

b) Übung. ■

Kommen wir jetzt zur Bestimmung von Interpolationspolynomen zu einem Datensatz (x_i, f_i), $i = 0, \ldots, n)$.Es gilt

Satz 4.8. Seien Daten $(x_i, f_i) \in \mathbb{C} \times \mathbb{C}$ $(i = 0, \ldots, n)$ gegeben mit $x_i \neq x_j$ für $i \neq j$. Dann gibt es genau ein **Interpolationspolynom** $p(x) = \sum_{i=0}^{n} a_i x^i$ vom Grade n mit

$$p(x_i) = f_i, \quad i = 0, \ldots, n.$$

Beweis: Wir definieren zu x_i das **Lagrange Polynom** n-ten Grades gemäß

$$L_i(x) := \prod_{\substack{j=0 \\ x_j \neq x_i}}^{n} \frac{x - x_j}{x_i - x_j} \tag{70}$$

und setzen

$$p(x) := \sum_{i=0}^{n} f_i L_i(x).$$

Dann ist p ein Polynom n-ten Grades und es gilt $p(x_i) = f_i$ für $i = 0, \ldots, n$. Das sichert die Existenz des Polynoms. Gibt es Polynome $p_1 \neq p_2$ welche beide die Interpolationsaufgabe lösen, so ist $q := p_1 - p_2$ ein Polynom höchstens vom Grade n mit $n+1$ Nullstellen $x_0, \ldots, x_n$, muß also gemäß Hilfsatz 4.6 identisch verschwinden. Das sichert auch die Eindeutigkeit von p.

Die Auswertung des Interpolationspolynoms p in Lagrange Darstellung an einer Stelle x benötigt $O(n^2)$ mathematische Operationen. Ferner sollten die Nenner der Lagrange Polynome vorab berechnet werden, damit diese nicht bei jeder Auswertung neu berechnet werden müssen. Die Lagrange Darstellung von p ist nützlich immer dann, wenn nachträglich Meßwerte f_i geändert

werden, weil sich die Darstellung von p dann nicht "andert. Sollen allerdings Datenpunkte hinzu gefügt werden, ist diese Darstellung unpraktisch, weil dann alle Lagrange Polynome neu berechnet werden müssen. Günstiger ist dann die sogenannte **Newton Darstellung** des Interpolationspolynoms p.Dazu benötigen wir

Definition 4.9. (Devidierte Differenzen)
Zu x_i, f_i $(i = 0, \ldots, n)$ mit $x_i \neq x_j$ definieren wir die **Dividierten Differenzen** gemäß

$$f_{[x_i]} := f_i, \text{ und rekursiv } f_{[x_l,\ldots,x_{l+k}]} := \frac{f_{[x_{l+1},\ldots,x_{l+k}]} - f_{[x_l,\ldots,x_{l+k-1}]}}{x_{l+k} - x_l} \text{ für } 0 \leq l < l+k \leq n.$$

Damit gilt

Satz 4.10. (Newton Darstellung des Interpolationspolynoms)
Seien Daten (x_i, f_i) $(i = 0, \ldots, n)$ gegeben mit $x_i \neq x_j$ für $i \neq j$. Dann besitzt das Interpolationspolynom p vom Grade n mit

$$p(x_i) = f_i, \quad i = 0, \ldots, n.$$

die *Newton–Darstellung*

$$\begin{aligned} p_n(x) = f_{[x_0]} + f_{[x_0,x_1]}(x - x_0) + \cdots + f_{[x_0,x_1,\ldots,x_n]}(x - x_0)(x - x_1)\ldots(x - x_{n-1}) \\ =: c_0 + c_1(x - x_0) + \cdots + c_n(x - x_0)(x - x_1)\ldots(x - x_{n-1}). \quad (71) \end{aligned}$$

Beweis: $p_n \in \Pi_n$ folgt direkt aus der Darstellung (71). Die Berechenbarkeit der Konstanten c_j ergibt sich unmittelbar, indem man die Interpolationsbedingungen als Gleichungssystem aufschreibt:

$$\left.\begin{array}{ll} p_n(x_0) = c_0 & = f_0 \\ p_n(x_1) = c_0 + c_1(x_1 - x_0) & = f_1 \\ p_n(x_2) = c_0 + c_1(x_2 - x_0) + c_2(x_2 - x_0)\,(x_2 - x_1) & = f_2 \\ \vdots & \vdots \\ p_n(x_n) = c_0 + c_1(x_n - x_0) + c_2(x_n - x_0)\,(x_n - x_1) + \ldots + c_n \prod_{\nu=0}^{n-1}(x_n - x_\nu) & = f_n \end{array}\right\} \quad (72)$$

Die Konstanten c_j sind „von oben nach unten" berechenbar. Damit sind die Interpolationsbedingungen einschließlich der Gradforderung erfüllt. Die Eindeutigkeitsfrage ist bereits durch Satz 4.8 geklärt. ■

Aus der Darstellung (71) und dem zugehörigen Gleichungssystem (72) kann man folgende Eigenschaften ablesen:

1. Für jedes $j \in \{0, 1, \ldots, n\}$ wird der Koeffizient c_j nur aus den ersten $(j+1)$ Gleichungen $p_n(x_i) = f_i,\ i = 0, 1, \ldots, j,$ berechnet, ist also von den Interpolationsbedingungen für $i > j$ unabhängig. Wir schreiben:

$$\begin{aligned} c_0 &= f[x_0] = f_0 \\ c_i &= f[x_0, \ldots, x_i], \qquad i = 1, \ldots, n\,. \end{aligned}$$

Aus dieser Eigenschaft folgt insbesondere:

Ist $p_n \in \Pi_n$ Lösung der Interpolationsaufgabe für die Punkte $(x_i, f_i),\ i = 0, \ldots, n,$ und nimmt man einen weiteren Punkt (x_{n+1}, f_{n+1}) hinzu, so wird die Lösung $p_{n+1} \in \Pi_{n+1}$ des erweiterten Interpolationsproblems gemäß (71) gegeben durch

$$p_{n+1}(x) \quad = \quad p_n(x) \ + \ c_{n+1} \prod_{j=0}^{n} (x - x_j) \tag{73}$$

und c_{n+1} errechnet sich aus der Gleichung

$$c_{n+1} \quad = \quad \frac{f_{n+1} - p_n(x_{n+1})}{\prod\limits_{j=0}^{n} (x_{n+1} - x_j)}.$$

Man kann also die Lösung des Interpolationsproblems für $(n + 1)$ Punkte „ausbauen“ zur Lösung des Problems für $(n + 2)$ Punkte, ein Vorteil, den weder die Lösung obigen Gleichungssystem noch die Lagrange–Darstellung (70) bietet.

2. Vergleicht man das Newton-Polynom mit der üblichen Polynomdarstellung

$$p_n(x) \quad = \quad \sum_{j=0}^{n} c_j \prod_{\nu=0}^{j-1} (x - x_\nu) \quad = \quad \sum_{j=0}^{n} a_j \, x^j \, ,$$

so gilt für den Koeffizienten a_n der höchsten x-Potenz

$$a_n \ = \ c_n \qquad \text{(Beweis durch Ausmultiplizieren)}$$

Da Interpolationpolynome eindeutig sind (Satz 4.8), folgt hieraus, dass $a_n = c_n = f[x_0, \ldots, x_n]$ unabhängig ist von der Reihenfolge der Punkte $(x_i, f_i),\ i \leq n.$

3. Weiter zeigt (72): Ist a_j der Koeffizient der höchsten x-Potenz des Interpolationspolynoms für die Punkte $(x_i, f_i),\ i = 0, 1, \ldots, j,$ so folgt wie in 2. $a_j = c_j = f[x_0, \ldots, x_j]$ und man erhält analog:

   Alle $c_j = f[x_0, \ldots, x_j], \quad j = 0, 1, \ldots, n, \quad (c_0 = f[x_0]$ falls $j = 0),$ sind unabhängig von der Reihenfolge der Punkte $(x_i, f_i), i = 0, \ldots, j.$

4. Sind die c_i einmal bekannt, so erfordert die Auswertung des Polynoms $p_n(x)$ an einer Stelle x wesentlich weniger Rechenoperationen als bei Benutzung der Lagrange-Form, insbesondere auch, weil man (71) durch ein „Horner-ähnliches“ Schema programmieren kann (Nachweis!).

Für dividierte Differenzen fassen wir zusammen

Satz 4.11. (Eigenschaften dividierter Differenzen)

- $f_{[x_0,\ldots,x_l]} = f_{[x_{i_0},\ldots,x_{i_l}]}$ für jede Permutation $(i_0, \ldots, i_l)$ von $(0, \ldots, l)$.

- Sei f n-mal stetig differenzierbar und $f_i := f(x_i)$. Dann gilt

$$f_{[x_0,\ldots,x_n]} = \frac{f^{(n)}(\xi)}{n!} \text{ für ein } \xi \in I(x_0,\ldots,x_n),$$

  wobei $I(x_0,\ldots,x_n)$ das kleinste Intervall bezeichnet, welches $x_0,\ldots,x_n$ enthält.

- $f_{[x_l,\ldots,x_{l+k}]} = \frac{f^{(l)}(x_l)}{l!}$ falls $x_l = \cdots = x_{l+k}$. (Siehe auch Hermite Interpolation!)

Beweis: wird in Kapitel 4.1.1 nachgereicht. ■

Will man ein Interpolationspolynom nur an einer (oder nur an sehr wenigen) Stelle(n) auswerten, so empfiehlt sich dafür die Neville-Formel, die diese Auswertung ohne Berechnung der Koeffizienten c_i leistet.

Satz 4.12. *(Rekursionsformel für Polynome nach Neville)* Bei gegebenen Punkten $(x_j, f_j),\ j = 0,\ldots,n$, bezeichne $P_{i,i+1,\ldots,i+k}$ das Interpolationspolynom $\in \Pi_k$ zu den Punkten $(x_j, f_j), j = i, i+1,\ldots,i+k$. Dann gilt die Rekursionsformel

$$\begin{aligned} P_i(x) &= f_i\,, \\ P_{i,\ldots,i+k}(x) &= \frac{(x-x_i)\,P_{i+1,\ldots,i+k}(x) - (x-x_{i+k})\,P_{i,\ldots,i+k-1}(x)}{x_{i+k}-x_i}\,, \quad k > 0\,. \end{aligned} \tag{74}$$

Beweis:
Aus der Berechnungsformel (74) folgt $P_{i,\ldots,i+k} \in \Pi_k$ und durch Einsetzen der Werte $(x_j, f_j), j = i, i+1,\ldots,i+k$, erhält man direkt

$$P_{i,\ldots,i+k}(x_j) = f_j, \qquad j = i,\ldots,i+k\,,$$

also die Interpolationseigenschaft. Die Eindeutigkeit ist durch Satz 4.8 gesichert. ■

Die Berechnung eines einzelnen Polynomwertes unter Benutzung von Satz 4.12 gestaltet sich übersichtlich nach dem Neville-Schema, das wir für $n = 3$ angeben

Neville-Schema (für $n = 3$)

x	$P_i(x)$	$P_{i,i+1}(x)$	$P_{i,i+1,i+2}(x)$	$P_{i,i+1,i+2,i+3}(x) = p_3(x)$
x_0	$P_0(x) = f_0$			
		$P_{0,1}(x)$		
x_1	$P_1(x) = f_1$		$P_{0,1,2}(x)$	
		$P_{1,2}(x)$		$P_{0,1,2,3}(x)$
x_2	$P_2(x) = f_2$		$P_{1,2,3}(x)$	
		$P_{2,3}(x)$		
x_3	$P_3(x) = f_3$			

Für die Programmierung schreibt man die Neville-Formel am besten in der Gestalt

$$P_{i,\ldots,i+k}(x) = P_{i+1,\ldots,i+k}(x) + \frac{(P_{i+1,\ldots,i+k}(x) - P_{i,\ldots,i+k-1}(x))\,(x-x_{i+k})}{x_{i+k}-x_i}\,,$$

die weniger Multiplikationen als (74) benötigt.

Will man das Interpolationspolynom an mehreren Stellen auswerten, so ist es günstig (weil wenig Rechenoperationen benötigt werden), die Newton-Koeffizienten $c_i = f[x_0, \ldots, x_i]$ zu berechnen und dann (71) nach einem Horner-ähnlichen Schema auszuwerten.

Auch die Berechnung der dividierten Differenzen gemäß Definition 4.9 kann übersichtlich in einem Schema angeordnet werden.

Schema der dividierten Differenzen (hier für $n = 3$)

x	$f[\ ]$	$f[\ ,\ ]$	$f[\ ,\ ,\ ]$	$f[\ ,\ ,\ ,\ ]$
x_0	f_0			
		$f[x_0, x_1]$		
x_1	f_1		$f[x_0, x_1, x_2]$	
		$f[x_1, x_2]$		$f[x_0, x_1, x_2, x_3]$
x_2	f_2		$f[x_1, x_2, x_3]$	
		$f[x_2, x_3]$		
x_3	f_3			

In der oberen Diagonalen stehen die Koeffizienten c_i des Newton-Polynoms

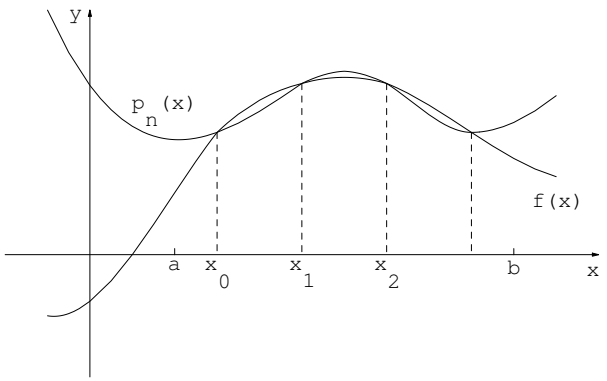
$$c_i \quad = \quad f[x_0, \ldots, x_i]\,.$$

Die Berechnung des Schemas benötigt insgesamt $n+(n-1)+\ldots+1 = n(n+1)/2$ Divisionen und doppelt soviele Subtraktionen. Dies ist ein konkurrenzlos geringer Aufwand im Vergleich zur Auflösung des Gleichungssystems (72) oder zu dem Aufwand, den die Lagrange-Formel bei der numerischen Auswertung benötigt. U.a. darum ist dieses Verfahren auch weniger rundungsfehleranfällig.

4.1.1 Interpolationsfehler

Interpolationspolynome stellen **Modelle** dar für Funktionen, deren Werte $f(x_j) \equiv f_j$ nur auf Stützstellen x_j bekannt sind. Wir wollen jetzt untersuchen, wie gut diese Modelle die Funktion f auf deren Definitonsintervall $[a, b]$ darstellen, d.h. wir möchten wissen, wie groß der Approximationsfehler ist:

$$\varepsilon(x) \;:=\; f(x) \;-\; p_n(x)\,, \qquad x \;\in\; [a, b]\,.$$



Um eine Aussage über den Approximationsfehler machen zu können, benötigt man natürlich weitere Informationen über f.

Wir untersuchen also folgende

Aufgabenstellung: $p_n \in \Pi_n$ interpoliere die Funktion f in den Stützstellen x_i, $i = 0, 1, \ldots, n$, die in einem Intervall $[a, b] \subset \mathbb{R}$ liegen mögen. Gesucht ist der Interpolationsfehler $\varepsilon(z)$ zwischen den Stützstellen

$$\varepsilon(z) \;=\; f(z) \,-\, p_n(z), \qquad z \,\in\, [a, b], \qquad z \;\neq\; x_i, \quad i = 0, 1, \ldots, n\,.$$

TRICK: Betrachte $z \quad (=: x_{n+1})$ als zusätzliche, beliebige aber feste (natürlich unbekannte) Stützstelle. $p_{n+1} \in \Pi_{n+1}$ interpoliere f in den Stellen $x_0, x_1, \ldots, x_n, z$. Insbesondere ist also $f(z) = p_{n+1}(z)$.

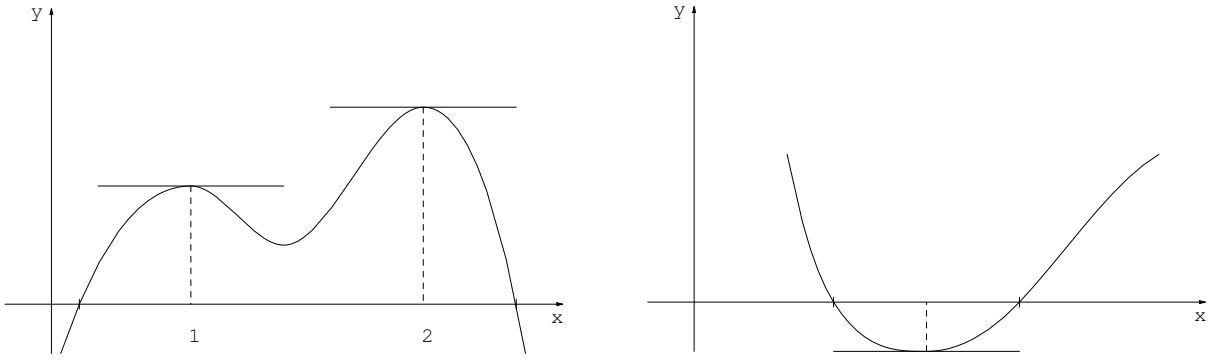
Berücksichtigt man die Eigenschaft (73) des Newtonschen Interpolationspolynoms, so gilt für den Fehler

$$\begin{aligned} \varepsilon(z) \;&=\; f(z) \,-\, p_n(z) \\ &=\; p_{n+1}(z) \,-\, p_n(z) \\ &=\; c_{n+1} \prod_{i=0}^{n} (z - x_i) \\ &=\; f[x_0, x_1, \ldots, x_n, z] \prod_{i=0}^{n} (z - x_i) \end{aligned} \tag{75}$$

Wir zeigen nun, wie die dividierte Differenz $f[x_0, x_1, \ldots, x_n, z]$ mit der Funktion f zusammenhängt. Dazu benötigen wir den Satz von Rolle, welcher folgendes besagt.

Hilfsatz 4.13. (Rolle) Sei $f : [\alpha, \beta] \to \mathbb{R}$ eine auf dem Intervall $[\alpha, \beta]$, $-\infty < \alpha < \beta < \infty$, stetige Funktion, die auf (a, b) differenzierbar ist. Gilt dann $f(\alpha) = f(\beta)$, so gibt es (mindestens) ein $\xi \in (\alpha, \beta)$ mit $f'(\xi) = 0$.

Veranschaulichung



Vorsicht:

Dieser Hilfsatz ist für komplexwertige Funktionen falsch!

Beispiel: $f(x) \;=\; e^{ix} - 1 \;=\; \cos\, x + i\, \sin\, x - 1, \;\; [\alpha, \beta] \;=\; [0, 2\pi]$.

Die folgenden Überlegungen sind also nur für reellwertige Funktionen richtig.

Wir setzen also voraus

$f : [a,b] \;\to\; \mathbb{R}$, (reellwertig)

$f \in C^{n+1}[a,b]$ (d.h. f gehört zur Menge der Funktionen, die auf dem Intervall (a,b) $(n+1)$-mal stetig differenzierbar sind und deren Ableitungen sich stetig auf $[a,b]$ fortsetzen lassen)

und wenden den Satz von Rolle an auf die Differenz $r(x)$ (vgl. (75))

$$\begin{aligned} r(x) &:= \varepsilon(x) - f[x_0,\ldots,x_n,z]\prod_{i=0}^{n}(x-x_i), \quad z \text{ beliebig aber fest} \\ &= f(x) - p_n(x) - f[x_0,x_1,\ldots,x_n,z]\prod_{i=0}^{n}(x-x_i) \\ &= f(x) - p_n(x) - f[x_0,x_1,\ldots,x_n,z]\omega(x) \end{aligned} \tag{76}$$

und auf deren Ableitungen. Hier bezeichnet $\omega(x) = \prod_{i=0}^{n}(x-x_i)$ das *Knotenpolynom*. Beachte, dass $r \in C^{n+1}[a,b]$.

$r(x)$ hat $n+2$ Nullstellen : $x_0,x_1,\ldots,x_n,z$. Zwischen je zwei Nullstellen von r liegt nach Hilfsatz 4.13 eine Nullstelle von $r'(x)$, also folgt

$r'(x)$ hat $n+1$ Nullstellen , und analog

$r''(x)$ hat n Nullstellen

$\vdots$

$r^{(n+1)}(x)$ hat eine Nullstelle ξ : $r^{(n+1)}(\xi) = 0, \quad \xi \in [a,b]$.

Nun ist $f \in C^{n+1}[a,b]$, $p_n \in \Pi_n$ also $p_n^{(n+1)} \equiv 0$, $f[x_0,x_1,\ldots,x_n,z]$ eine Zahl und $\omega(x)$ ein Polynom, dessen $(n+1)$ste Ableitung $= (n+1)!$ ist. Deshalb folgt aus (76) und $r^{(n+1)}(\xi) = 0$

$$r^{(n+1)}(\xi) = 0 = f^{(n+1)}(\xi) - f[x_0,x_1,\ldots,x_n,z]\cdot(n+1)!$$

bzw. (setze $z = x_{n+1}$)

$$f[x_0,x_1,\ldots,x_n,x_{n+1}] = \frac{1}{(n+1)!}f^{(n+1)}(\xi), \qquad \xi \in [a,b].$$

Wird dies in $\varepsilon(z)$ eingesetzt (vgl. (75)), so folgt

$$\varepsilon(z) = f(z) - p_n(z) = \frac{1}{(n+1)!}f^{(n+1)}(\xi)\omega(z) \tag{77}$$

mit einer von z abhängigen Stelle $\xi \in [a,b]$.

Die Herleitung gilt zunächst für $z \neq x_i$, doch ist (77) trivialerweise auch für $z = x_i$ richtig. Beachte ferner, dass bei festen Knoten x_i, $i = 0,\ldots,n$ die Zwischenstelle ξ von z abhängt. Zusammengefasst gilt also:

Satz 4.14. $p_n \in \Pi_n$ interpoliere die Funktion $f : [a,b] \to \mathbb{R}$ an den Stützstellen $x_i \in [a,b]$, $i = 0, 1, \ldots, n$, $-\infty < a, b < \infty$.

a) Dann gilt für den Interpolationsfehler (vgl. (75))

$$f(z) \; - \; p_n(z) \;\; = \;\; f[x_0, x_1, \ldots, x_n, z] \prod_{i=0}^{n} (z - x_i), \quad z \in [a,b], \;\; z \neq x_i, \;\; \forall\, i$$

b) Ist zusätzlich $f \in C^{n+1}[a,b]$, so gibt es ein $\xi \in [a,b]$ mit

$$f[x_0, x_1, \ldots, x_{n+1}] \;\; = \;\; \tfrac{1}{(1+n)!}\, f^{(n+1)}(\xi), \;\; x_i \in [a,b], \quad i = 0, \ldots, n+1,$$
$$x_i \neq x_j \;\text{ für }\; i \neq j$$

und der Interpolationsfehler kann dargestellt werden durch (77)

$$f(x) \; - \; p_n(x) \;\; = \;\; \tfrac{1}{(1+n)!}\, f^{(n+1)}(\xi_x) \;\cdot\; \prod_{i=0}^{n} (x - x_i),$$
$$\forall\, x \in [a,b], \quad \xi_x \in [a,b], \quad \text{abhängig von } x$$

bzw.

$$|f(x) \; - \; p_n(x)| \;\; \leq \;\; \frac{1}{(1+n)!} \max_{\xi \in [a,b]} \left| f^{(n+1)}(\xi) \right| \;\cdot\; \left| \prod_{i=0}^{n} (x - x_i) \right| .$$

Bemerkung 4.15.

1. $\max\limits_{\xi \in [a,b]} \left| f^{(n+1)}(\xi) \right|$ existiert unter den Voraussetzungen unseres Satzes existiert, wie in der Analysis bewiesen wird. Für $f(x) = \sqrt{x}$, $x \in [0,1]$ ist die Aussage des Satzes z.B. falsch (weil $f \notin C^{n+1}[0,1]$, sondern nur $\in C^{n+1}(0,1)$).

2. Da der Existenzbeweis für die Zwischenstelle ξ nicht konstruktiv ist, ist die betragsmäßige Abschätzung des Interpolationsfehlers für praktische Zwecke günstiger.

Als Beispiel interpolieren wir die Funktion $f(x) = \frac{1}{1+25x^2}$ im Intervall $[-1,1]$ an 11 äquidistanten Stützstellen. Wie Abb. 2 zeigt, stellt sich ein auffällig großer Fehler an den Intervallrändern ein. Die Fehlerabschätzung aus Satz 4.14 gibt Anlass zu überlegen, ob, und falls, wie man den Approximationsfehler verkleinern kann. Den Ausdruck $f^{(n+1)}(\xi)/(n+1)!$ kann man nicht direkt beeinflussen, da ξ nicht konstruiert wurde, wohl aber das **Knotenpolynom**

$$\omega(x) \;\; := \;\; \prod_{i=0}^{n} (x - x_i)\,,$$

denn über die Wahl der x_i wurde bisher noch nicht verfügt. Wir wollen die Stützstellen jetzt so bestimmen, daß der Beitrag des Notenpolynoms in (77) zum Fehler möglichst klein ausfällt. Das führt auf die Untersuchung der sogenannten

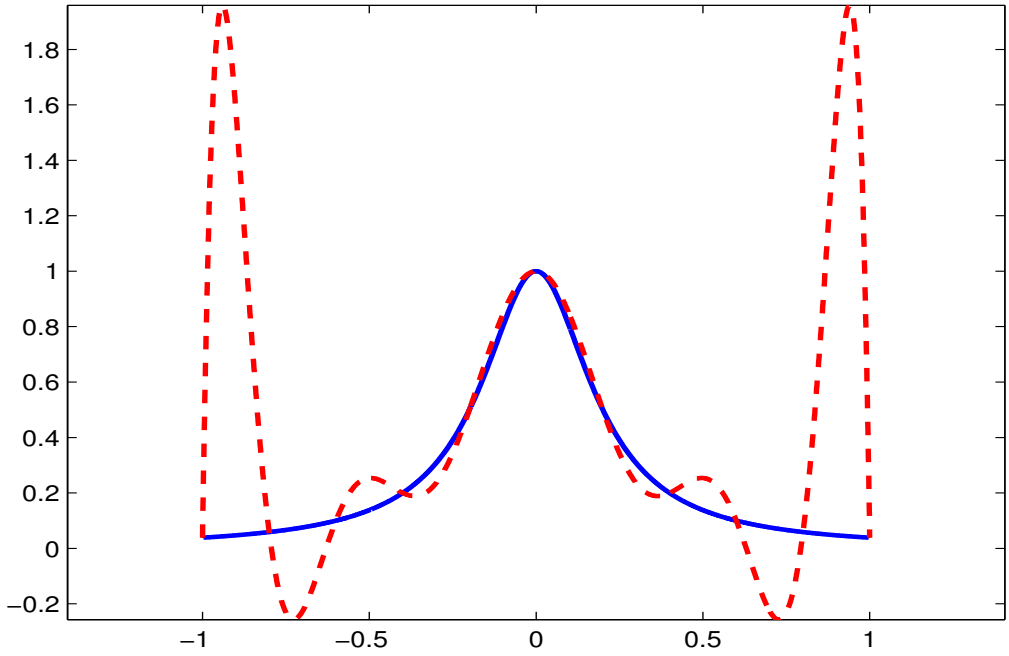


Abbildung 2: Interpolationspolynom 11ten Grades zu $f(x) = \frac{1}{1+25x^2}$

4.1.2 Optimale Stützstellen, Tschebyscheff-Knoten

Numerische Beispiele zeigen, dass bei äquidistanter Knotenwahl die Ausschläge von $\omega(x)$ an den Intervallenden besonders groß werden. Die maximalen Ausschläge werden kleiner, wenn man die Stützstellen mehr auf den Rand der Intervalle konzentriert. Dafür werden die Ausschläge in der Intervallmitte etwas größer. Typisch hierfür ist die Darstellung in Abb. 3, das den Funktionsverlauf von $\omega(x)$ in $[-1,1]$ für $n = 10$ (d.h. 11 äquidistant verteilte Stützstellen in $[-1,1]$) zeigt (durchgezogene Linie) und gestrichelt den Verlauf, welchen man erhält, falls als Stützstellen die sogenannten **Tschebyscheff-Knoten** gewählt werden (natürlich ebenfalls 11 Stück, vgl. dazu den folgenden Satz). Es stellt sich also folgende

Frage: Wie sollten die Stützstellen (Knoten) $x_i \in [a,b]$, $i = 0,1,\ldots,n$, gewählt werden, damit der maximale Funktionswert $\max\limits_{x\in[a,b]} \left|\prod\limits_{i=0}^{n}(x-x_i)\right|$ des Knotenpolynoms möglichst klein wird? D.h. gesucht wird

$$\min_{x_0,\ldots,x_n} \max_{x\in[a,b]} \left|\prod_{i=0}^{n}(x-x_i)\right|$$

(sofern das Minimum existiert, was a priori nicht klar ist).

Zunächst sollte jedoch im Vorwege geklärt werden, ob man dieses Min. (falls existent) für jedes Intervall gesondert suchen muss, oder ob man sich auf ein Referenzintervall beschränken kann. Letzteres ist tatsächlich der Fall, wie der nachfolgende Hilfsatz zeigt.

Hilfsatz 4.16. Werden das Intervall $[a,b], (-\infty < a < b < \infty)$, und die Stützstellen $x_i \in [a,b]$

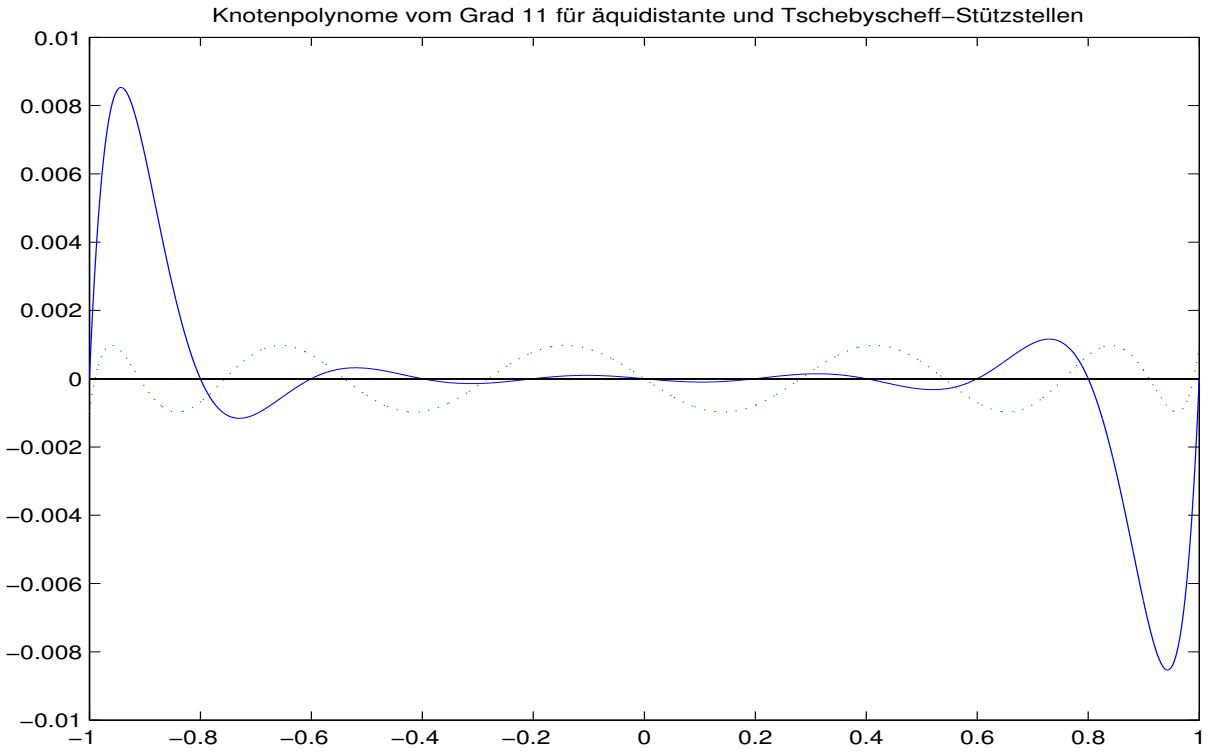


Abbildung 3: Das Knotenpolynom ω für 11 Stützstellen; durchgezogene Linie: äquidistante Stützstellen, gestrichelte Linie: Tschebyscheff Knoten

durch die umkehrbar eindeutige lineare Transformation

$$y \quad = \quad c + \frac{x-a}{b-a}\,(d-c) \tag{78}$$

auf das Intervall $[c,d]$, $(-\infty < c < d < \infty)$, und die Werte y_i abgebildet, so werden auch die Extrema von $\omega(x) = \prod_{i=0}^{n}(x-x_i)$ auf die Extrema von $\tilde{w}(y) = \prod_{i=0}^{n}(y-y_i)$ abgebildet.

Beweis: Die Behauptung ergibt sich aus der Beziehung

$$\begin{aligned}\tilde{w}(y) \;=\; \prod_{i=0}^{n}(y-y_i) \;&=\; \prod_{i=0}^{n}\left(\left[c+\frac{x-a}{b-a}\,(d-c)\right] \;-\; \left[c+\frac{x_i-a}{b-a}\,(d-c)\right]\right)\\ &=\; \left(\frac{d-c}{b-a}\right)^{n+1}\prod_{i=0}^{n}(x-x_i) \;=\; \left(\frac{d-c}{b-a}\right)^{n+1}\omega(x)\end{aligned}$$

und

$$\frac{d\tilde{w}(y(x))}{dx} \quad = \quad \tilde{w}'(y)\cdot\frac{dy(x)}{dx} \quad = \quad \left(\frac{d-c}{b-a}\right)^{n+1} w'(x)\,.$$

Es ist also $w'(x) = 0$ genau dann, wenn $\tilde{w}'(y) = 0$ mit $y = y(x)$ gem. (78), denn $\frac{dy}{dx} = \left(\frac{d-c}{b-a}\right) \neq 0$. ■

Wir können für die weiteren Untersuchungen also das Referenzintervall $[-1,1]$ $(c=-1,$ $d=+1)$ wählen, geeignete Stützstellen $y_i \in [-1,1]$ für $\tilde{w}(y)$ so bestimmen, dass $\max\limits_{y\in[-1,1]} |\prod\limits_{i=0}^{n}(y-y_i)|$ minimal wird und danach die Stützstellen $x_i \in [a,b]$ durch die Rücktransformation

$$x_i \quad = \quad a \,+\, \frac{y_i - c}{d-c}\,(b-a) \quad = \quad a \,+\, \frac{y_i+1}{2}\,(b-a)$$

bestimmen.

Unsere Frage wird jetzt beantwortet durch die nachfolgenden Sätze

Satz 4.17. (Tschebyscheff-Polynome)

a) Die Funktionen

$$T_n(y) \;=\; \cos\,(n \,\text{arc}\,\cos\, y), \qquad n=0,1,2,\ldots, \quad y\in[-1,1] \tag{79}$$

genügen der Rekursionsformel

$$T_{n+1}(y) \;=\; 2y\,T_n(y) \,-\, T_{n-1}(y), \qquad T_0(y) \,=\, 1, \quad T_1(y)=y, \quad n\in\mathbb{N}\,, \tag{80}$$

sind also Polynome vom Grad n, d.h. $T_n \in \Pi_n$ und heißen **Tschebyscheff-Polynome**.

b) T_n hat die n verschiedenen (Tschebyscheff-) Nullstellen

$$y_j \;=\; \cos\left(\frac{2j+1}{2n}\,\pi\right) \,\in\, (-1,1), \qquad j=0,1,\ldots,n-1 \tag{81}$$

und nimmt im Intervall $[-1,1]$ seine Extrema an in den $n+1$ Extremalstellen

$$y_j^{(e)} \;=\; \cos\left(\frac{j\,\pi}{n}\right) \,\in\, [-1,1], \qquad j=0,1,\ldots,n\,, \tag{82}$$

mit den Extremalwerten

$$T_n\left(y_j^{(e)}\right) \;=\; (-1)^j, \qquad j=0,1,\ldots,n\,. \tag{83}$$

c) T_n besitzt die Darstellung

$$T_n(y) \;=\; 2^{n-1} \,\prod_{j=0}^{n-1}\,(y-y_j)\,, \quad n\in\mathbb{N}\,. \tag{84}$$

Unsere Frage nach der bestmöglichen Fehlerabschätzung für die Interpolation wird dann beantwortet durch

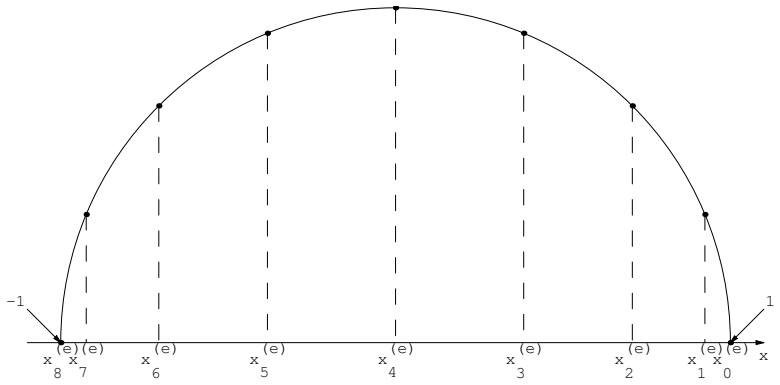
Satz 4.18. Der maximale Extremalwert von $|\omega(y)| = \left|\prod\limits_{j=0}^{n}\,(y-y_j)\right|$ für $y\in[-1,1]$ wird minimiert, falls für y_j die Nullstellen von T_{n+1} gewählt werden, d.h. für alle $q(y) := \prod\limits_{j=0}^{n}(y-\xi_j)$ mit paarweise verschiedenen $\xi_j \in \mathbb{R}$ gilt

$$\max_{y\in[-1,1]}\;|\omega(y)| \quad\leq\quad \max_{y\in[-1,1]}\;|q(y)| \tag{85}$$

Bemerkung 4.19.

1. Durch (79) sind die T_n nur für das Intervall $[-1,1]$ definiert, sie können aber natürlich mittels (80) auf $\mathbb{R}$ fortgesetzt werden.

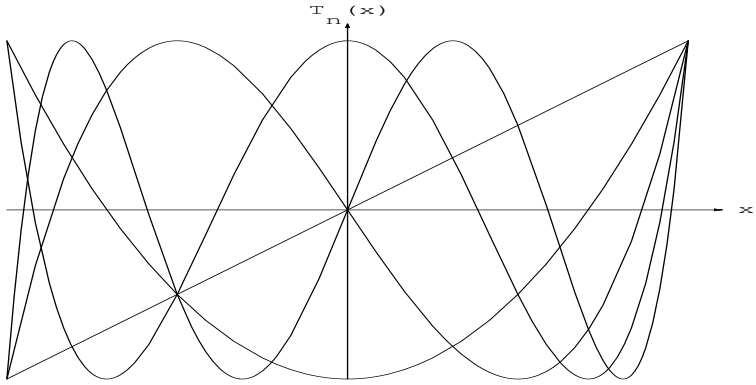
2. **Alle** Nullstellen der T-Polynome liegen in $(-1,1)$, sie sind, ebenso wie die Extremalstellen $y_j^{(e)}$, symmetrisch zum Nullpunkt verteilt. Man kann sie sich geometrisch vorstellen als Projektionen von regelmäßig auf dem Halbkreis verteilten Punkten, z.B.



Extremalstellen von $T_8(x)$

3. Die Nullstellen und Extremalstellen in (81), (82) sind in absteigender Reihenfolge geordnet: $y_{j+1} < y_j$.

4. Zwei der Extremalstellen $\in [-1,1]$ sind keine Waagepunkte von T_n sondern Randmaxima bzw. Randminima. Vergleiche dazu auch



Tschebyscheff-Polynome $T_n(x),\ n = 1(1)5$.

5. Aus (83) und (84) folgt

$$\left| \prod_{j=0}^{n-1} (y - y_j) \right| \;=\; 2^{-(n-1)} \, |T_n(y)| \;\leq\; 2^{-(n-1)} \quad \text{in} \quad [-1,1]\,.$$

Das Gleichheitszeichen wird in den Extremalstellen (82) angenommen.

Beweis Satz 4.17

a) Grundlage des Beweises ist die trigonometrische Identität

$$\cos[(n+1)z] \;+\; \cos[(n-1)z] \;\;=\;\; 2\cos\, z\, \cos(nz), \qquad n \in \mathbb{N}\,. \tag{86}$$

Man bestätigt sie sofort mit Hilfe der Additionstheoreme

$$\cos(\alpha \pm \beta) \quad = \quad \cos\,\alpha\, \cos\,\beta \mp \sin\,\alpha\, \sin\,\beta$$

indem man auf der linken Seite von (86) $\alpha = nz,\ \beta = z$ setzt.

Setzt man in (86) $z = \text{arc}\,\cos\, y$, so folgt

$$T_{n+1}(y) \;+\; T_{n-1}(y) \quad = \quad 2y\, T_n(y)\,.$$

Die Anfangswerte $T_0(y) = 1,\ T_1(y) = y$ erhält man sofort aus (79). Damit ist das Bildungsgesetz (80) für die Funktionen (79) bewiesen. Es zeigt – mittels vollständiger Induktion – sofort $T_n \in \Pi_n$.

b) Die $\cos$-Nullstellen sind bekannt. Aus (vgl. (79))

$$\cos(n\,\text{arc}\,\cos\, y) \;\;=\;\; 0 \quad \text{folgt}$$

$$n\,\text{arc}\,\cos\, y \;\;=\;\; (2k+1)\,\tfrac{\pi}{2}\,, \quad k \in \mathbb{Z}, \quad \text{bzw.}$$

$$y_j \;\;=\;\; \cos\left(\tfrac{2j+1}{2n}\,\pi\right)\,, \qquad j = 0, 1, \ldots, n-1, \quad n \geq 1, \quad \text{also (81)}\,.$$

Man kann sich auf $j = 0, 1, \ldots, n-1$ beschränken, danach wiederholen sich die Nullstellen auf Grund der Periodizität von $\cos$.

Die Extremalstellen der $\cos$-Funktion sind bekannt. Aus $|\cos(n\,\text{arc}\,\cos\, y)| = 1$ folgt $n\,\text{arc}\,\cos\, y = k\,\pi,\ k \in \mathbb{Z}$ also $y_j^{(e)} = \cos\,\frac{j\,\pi}{n},\ j = 0, \ldots, n$ (Periodizität) und $T(y_j^{(e)}) = (-1)^j$ durch Einsetzen in (79), also sind (80) – (83) bewiesen.

c) Da $T_n \in \Pi_n$ die n Nullstellen y_j besitzt, hat es die Form $T_n(y) = c_n \prod_{j=0}^{n-1} (y - y_j)$ mit einer Konstanten c_n. Dies folgt aus (69) (Horner-Schema).

Aus dem Bildungsgesetz (80) für T_n folgt, dass der Koeffizient c_n von y^n gleich 2^{n-1} ist, also gilt (84). ■

Beweis Satz 4.18 Wir nehmen an (vgl. (85)): Es gebe ξ_j, sodaß

$$\max_{y\in[-1,1]} |q(y)| \;\;=\;\; \max_{y\in[-1,1]} \left|\prod_{j=0}^{n} (y - \xi_j)\right| \;\;<\;\; \max_{y\in[-1,1]} |\omega(y)| \;\;=\;\; \max_{y\in[-1,1]} \left|\prod_{j=0}^{n} (y - y_j)\right|\,.$$

Wir leiten nun mit Hilfe von Satz 4.17 (für „$n+1$“ statt „n“, da Satz 4.18 Aussagen über T_{n+1} macht) einen Widerspruch her. Beachte dazu: Die Extremalstellen sind absteigend numeriert und ihre Funktionswerte haben gleichen Betrag aber alternierendes Vorzeichen.

Aus der Annahme folgt für die Funktionswerte $q(y_j^{(e)})$ in den Extremalstellen von T_{n+1} (vgl. (82)–(84) für $n+1$ statt n)

$$q\left(y_0^{(e)}\right) \leq \max_{y\in[-1,1]} |q(y)| < \max_{y\in[-1,1]} \left|\prod_{j=0}^{n}(y-y_j)\right| = 2^{-n} = w\left(y_0^{(e)}\right),$$

$$q\left(y_1^{(e)}\right) \geq -\max_{y\in[-1,1]} |q(y)| > -\max_{y\in[-1,1]} \left|\prod_{j=0}^{n}(y-y_j)\right| = -2^{-n} = w\left(y_1^{(e)}\right),$$

$$q\left(y_2^{(e)}\right) < 2^{-n} = w\left(y_2^{(e)}\right)$$

$$\vdots$$

allgemein also

$$q\left(y_j^{(e)}\right) \quad \begin{cases} < w\left(y_j^{(e)}\right), & \text{falls } j \text{ gerade} \\ > w\left(y_j^{(e)}\right), & \text{falls } j \text{ ungerade}, \quad j=0,1,\ldots,n+1\,. \end{cases}$$

Hieraus folgt für das Differenzpolynom $p(y) := \omega(y) - q(y)$

$$p\left(y_j^{(e)}\right) = w\left(y_j^{(e)}\right) - q\left(y_j^{(e)}\right) \quad \begin{cases} > 0, & \text{falls } j \text{ gerade} \\ < 0, & \text{falls } j \text{ ungerade}\,, \quad j=0,1,\ldots,n+1\,. \end{cases}$$

$p(y)$ hat also in $[-1,1]$ $n+1$ Vorzeichenwechsel, also $n+1$ Nullstellen. Nun ist aber $p \in \Pi_n$, denn sowohl bei $\omega(y)$ als auch bei $q(y)$ hat y^{n+1} den Koeffizienten 1, also verschwindet y^{n+1} bei der Differenzbildung. Damit folgt aus Lemma 4.6, dass $p \equiv 0$, also $\omega(y) = q(y)$. Dies ist aber ein Widerspruch zur Annahme. ∎

Wesentlich für den Beweis von Satz 4.18 ist die Tatsache, dass alle Extremalwerte von T_{n+1} den **gleichen** Betrag und alternierendes Vorzeichen haben (vgl. dazu **Tschebyscheff Approximation**).

Um die Auswirkung der Wahl der Tschebyscheff-Nullstellen auf die Approximationsgenauigkeit des Interpolationsverfahrens zu demonstrieren, greifen wir nochmals das Beispiel $f(x) = \frac{1}{1+25x^2}$ auf und stellen zum Vergleich die Ergebnisse der Interpolation mit äquidistanten- und Tschbebyscheff-Nullstellen in Abb. 4 nebeneinander. Wir stellen fest, daß die Ausschläge an den Intervallrändern, welche charakteristisch waren für Interpolation auf äquidistanten Gittern, jetzt nicht mehr auftreten.

Mit Hilfsatz 4.16 erhalten wir noch (dort $d=b, c=a, a=-1, b=1$)

$$\min_{\Delta: a\leq x_0<\cdots<x_n\leq b} \max_{x\in[a,b]} |\omega(x)| \quad \left(= \frac{(b-a)^{n+1}}{2\cdot 4^n} \text{ (Nachweis!)}\right). \tag{87}$$

Ferner folgern wir aus Satz 4.14 b.) für eine beliebige Knotenverteilung $\Delta^{(n)} : a \leq x_0 < \cdots < x_n \leq b$ direkt

$$\|f-p\|_\infty := \max_{x\in[a,b]} |f(x)-p(x)| \leq \frac{\|f^{(n+1)}\|_\infty (b-a)^{n+1}}{(n+1)!}, \tag{88}$$

und diese Abschätzung wollen wir uns merken. Denn wir erhalten mir deren Hilfe sofort

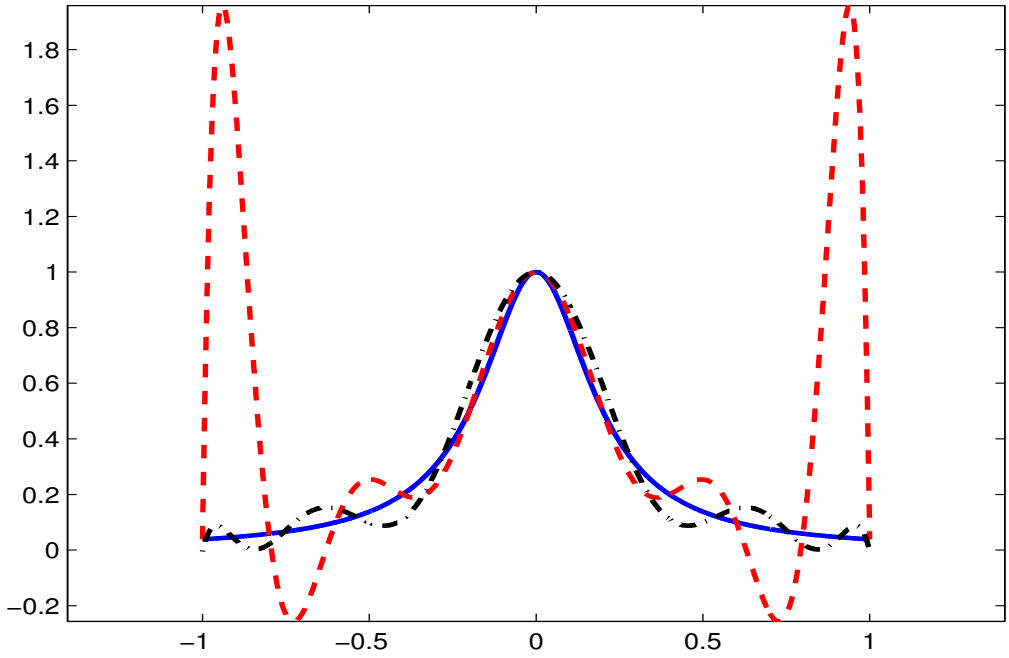


Abbildung 4: Interpolation auf 11 Knoten: äquidistante Knoten versus Tschebyscheff-Knoten

Satz 4.20. Sei f ∞–oft differenzierbar mit $\|f^{(k)}\|_\infty \leq M$ für alle $k \in \mathbb{N}$ mit einer positiven, von n unabhängigen Konstanten M. Ferner bezeichne p_n das eindeutig bestimmte Interpolationpolynom zu f auf der Knotenverteilung $\Delta^{(n)} : a \leq x_0 < \cdots < x_n \leq b$. Dann konvergieren die Interpolationspolynome p_n mit wachsenden n gleichmäßig gegen f, genauer gilt

$$\|f - p_n\|_\infty :=\leq \frac{M(b-a)^{n+1}}{(n+1)!} \to 0 \text{ für } n \to \infty.$$

Bemerken wollen wir hier, daß in Satz 4.20 nicht verlangt wird, daß die **Zerlegungsfeinheit** $|\Delta^{(n)}| :=\max\{x_{i+1} - x_i, 0 \leq i \leq n-1\}$ gegen Null tendiert, falls n nach ∞ strebt.

Leider kann **nicht** geschlossen werden, daß für feiner werdende Unterteilungen des Intervalls $[a, b]$ die zugehörigen Interpolationspolynome einer stetigen Funktion f gleichmßig gegen die Funktion f konvergieren, wie ein Resultat von Faber [12, Theorem 1.19] zeigt.

Die Voraussetzungen von Satz 4.20 enthalten die sehr starke Forderung $\|f^{(k)}\|_\infty \leq M$ gleichmäßig in $k \in \mathbb{N}$. In der Praxis sind solche Voraussetzungen für die zu modellierende Funktion f häufig nicht erfüllt. Andererseits entnehmen wir der Abschätzung (88), daß Interpolationspolynome p Funktionen f mit geringen Differenzierbarkeitseigenschaften gut approximieren, wenn $b - a$ klein ist. Diese Beobachtung legt es uns nahe, das **globale Modell Interpolationspolynom** (von hoher Ordnung) auf dem Intervall $[a, b]$ zu ersetzen durch **lokale Interpolationsmodelle** (von jeweils geringer Ordnung). Das führt auf

4.2 Spline Interpolation

Die Interpolationsaufgabe bestehe darin, vorgelegte Daten $(x_i, f_i := f(x_i))\ (i = 0, \ldots, n+1)$ zu interpolieren und damit ein geeignetes Modell für die Funktion $f : [a, b] \to \mathbb{R}$ zu erhalten. Wir geben uns ein Gitter (eine Zerlegung)

$$\Delta : a = x_0 < x_1 < \cdots < x_{n+1} = b$$

vor, definieren $\|\Delta\| := \max_i h_i$, $h_i := x_i - x_{i-1}\ (i = 1, \ldots, n+1)$ und verlangen jetzt von unserer interpolierenden Funktion S_Δ, daß sie auf dem Gitter ein **interpolierender Spline k-ter Ordnung** ist.

Definition 4.21. Sei $\Delta : a = x_0 < x_1 < \cdots < x_{n+1} = b$ eine Zerlegung des Intervalls $[a, b]$. Eine Funktion $S_{\Delta,k} : [a, b] \to \mathbb{R}$ heißt **Spline k-ter Ordnung** (zur Zerlegung Δ), falls

$$S_{\Delta,k}|_{[x_i,x_{i+1}]} \in \Pi_k, \text{ } S_\Delta \in C^{k-1}([a, b]) \text{(für } k \geq 1).$$

Eine Funktion $S_{\Delta,k} : [a, b] \to \mathbb{R}$ heißt **interpolierender Spline k-ter Ordnung** (zur Zerlegung Δ und Daten $(x_i, f_i := f(x_i))\ (i = 0, \ldots, n+1)$), falls

$S_{\Delta,k}$ Spline k–ter Ordnung und

$S_{\Delta,k}(x_i) = f_i$ für $i = 0, \ldots, n+1$.

Interpolierende Splines sind demnach auf jedem Teilintervall des Gitters Δ Polynome, modellieren die zu interpolierende Funktion f demnach **lokal** polynomial.

Bemerkung 4.22. Splines können natürlich auch ohne das Attribut **interpolierend** definiert werden. Zu einem Gitter $\Delta : a = x_0 < x_1 < \cdots < x_{n+1} = b$ wollen wir eine Funktion $S_{\Delta,k} : [a, b] \to \mathbb{R}$ mit $S_{\Delta,k}|_{[x_i,x_{i+1}]} \in \Pi_k$, $S_\Delta \in C^{k-1}([a, b])$(für $k \geq 1$) **Spline k-ter Ordnung** nennen. Zu Daten $(\xi_j, f_j), j = 0, \ldots, n+1$ mit $a \leq \xi_0 < \xi_1 < \cdots < \xi_{n+1} \leq b$ können wir die Interpolationsaufgabe etwas allgemeiner wie folgt formulieren; Finde einen Spline $S_{\Delta,k}$ k-ter Ordnung mit $S_{\Delta,k}(\xi_j) = f_j$ für $j = 0, \ldots, n+1$. Siehe dazu (99).

Wir wollen uns zwei Beispiele anschauen.

Beispiel. Sei $(x_i, f_i := f(x_i))\ (i = 0, \ldots, n+1)$ ein Datensatz. Der interpolierende Spline 0-ter Ordnung ist gegeben durch

$$S_{\Delta,0}(x) := f_i, \quad x \in [x_i, x_{i+1}),\ (i = 0, \ldots, n) \text{ und } S_{\Delta,0}(x_{n+1}) := f_{n+1}.$$

Der interpolierende **lineare Spline** (= interpolierender Spline erster Ordnung) ist gegben durch

$$S_{\Delta,1}(x) := \sum_{i=0}^{n+1} f_i b_i(x), \tag{89}$$

wobei die stückweise linearen, auf $[a,b]$ global stetigen Funktionen $b_i : [a,b] \to \mathbb{R}$ definiert sind durch

$$b_i(x) := \begin{cases} \frac{x-x_{i-1}}{x_i-x_{i-1}} & x \in [x_{i-1}, x_i) \\ \frac{x_{i+1}-x}{x_{i+1}-x_i} & x \in [x_i, x_{i+1}) \\ 0 & \text{sonst} \end{cases} \quad \text{für } i = 1, \ldots, n,$$

$$b_0(x) := \begin{cases} \frac{x_1-x}{x_1-x_0} & x \in [x_0, x_1) \\ 0 & \text{sonst} \end{cases}, \quad b_{n+1}(x) := \begin{cases} \frac{x-x_n}{x_{n+1}-x_n} & x \in [x_n, x_{n+1}) \\ 0 & \text{sonst} \end{cases} \tag{90}$$

Interpolierende Splines dritter Ordnung heißen

4.2.1 Kubische Splines

Wir wollen uns überlegen, wie wir **Kubische Splines** S_Δ numerisch berechnen können (wir lassen die 3 als Index der Einfachheit halber fort). Dazu stellen wir fest, daß mit $h_i := x_i - x_{i-1}$ und $S''_\Delta(x_i) =: M_i$ (den sogenannten **Momenten**)

$$S''_{\Delta_{|[x_{i-1},x_i]}}(x) = M_i \frac{x - x_{i-1}}{h_i} + M_{i-1} \frac{x_i - x}{h_i}$$

gültig ist. Integration liefert sofort

$$S'_{\Delta_{|[x_{i-1},x_i]}}(x) = M_i \frac{(x - x_{i-1})^2}{2h_i} - M_{i-1} \frac{(x_i - x)^2}{2h_i} + A_i$$

und

$$S_{\Delta_{|[x_{i-1},x_i]}}(x) = M_i \frac{(x - x_{i-1})^3}{6h_i} + M_{i-1} \frac{(x_i - x)^3}{6h_i} + A_i(x - x_{i-1}) + B_i$$

mit geeigneten Konstanten A_i, B_i. Diese Konstanten können jetzt durch die Momenten M_i, M_{i-1} und die Daten f_i, f_{i-1} ausgedrückt werden. Nach leichter Rechnung ergibt sich

$$B_i = f_{i-1} - M_{i-1} \frac{h_i^2}{6} \text{ und } A_i = \frac{f_i - f_{i-1}}{h_i} - \frac{h_i}{6}(M_i - M_{i-1})$$

Damit ist $S_{\Delta_{|[x_{i-1},x_i]}}$ durch M_{i-1}, M_i bestimmt, S_Δ demnach durch $M_0, \ldots, M_{n+1}$. Wie können wir jetzt die Momenten bestimmen? Dazu erinnern wir uns, daß $S_\Delta \in C^2$ gilt und daher auf den inneren Gitterpunkten $x_1, \ldots, x_n$ sicher

$$\lim_{x \nearrow x_i} S'_\Delta(x) = M_i \frac{h_i}{2} + A_i = -M_i \frac{h_{i+1}}{2} + A_{i+1} = \lim_{x \searrow x_i} S'_\Delta(x)$$

richtig ist. Es ergibt sich sofort mit

$$A_i = \frac{f_i - f_{i-1}}{h_i} - \frac{h_i}{6}(M_i - M_{i-1}) \text{ und } A_{i+1} = \frac{f_{i+1} - f_i}{h_{i+1}} - \frac{h_{i+1}}{6}(M_{i+1} - M_i)$$

das System von Gleichungen

$$M_{i-1} \frac{h_i}{6} + M_i \frac{h_i + h_{i+1}}{3} + M_{i+1} \frac{h_{i+1}}{6} = \frac{f_{i+1} - f_i}{h_{i+1}} - \frac{f_i - f_{i-1}}{h_i} \text{ für } i = 1, \ldots, n. \tag{91}$$

Das sind n Gleichungen für $n+2$ Unbekannte $M_0, \ldots, M_{n+1}$. Die Restlichen zwei Bedingungen werden i.d.R. wie folgt gestellt:

i. Natürlicher Spline: $S''_\Delta(a) = 0 = S''_\Delta(b)$, d.h. $M_0 = 0 = M_{n+1}$.

ii. Periodischer Spline: $S''_\Delta(a) = S''_\Delta(b)$ ($\Rightarrow M_0 = M_{n+1}$) und $S'_\Delta(a) = S'_\Delta(b)$.

iii. $S'_\Delta(a) = f'_0$ und $S'_\Delta(b) = f'_{n+1}$.

Damit ist das Interpolationsproblem für kubische Splines gelöst. Wir wollen abschließend oben formulierte Bedingungen in Form von linearen Gleichungssystemen formulieren. Dazu schreiben wir (91) um in

$$\mu_i M_{i-1} + 2M_i + \lambda_i M_{i+1} = d_i \text{ für } i = 1, \ldots, n, \tag{92}$$

wobei

$$\lambda_i := \frac{h_{i+1}}{h_{i+1} + h_i},\ \mu_i := 1 - \lambda_i \text{ und } d_i := \frac{6}{h_{i+1} + h_i} \left(\frac{f_{i+1} - f_i}{h_{i+1}} - \frac{f_i - f_{i-1}}{h_i} \right).$$

Wir setzen noch in den Fällen

i. Natürlicher Spline: $\lambda_0 := 0$, $d_0 := 0$, $\mu_{n+1} := 0$ und $d_{n+1} := 0$,

ii. Periodischer Spline: $\lambda_{n+1} := \frac{h_1}{h_{n+1}+h_1}$, $\mu_{n+1} := 1 - \lambda_{n+1} = \frac{h_{n+1}}{h_{n+1}+h_1}$ und $d_{n+1} := \frac{6}{h_{n+1}+h_1} \left(\frac{f_1 - f_{n+1}}{h_1} - \frac{f_{n+1} - f_n}{h_{n+1}} \right)$ (beachte, daß $f_0 = f_{n+1}$ wegen Periodizität), und

iii. Hermite Spline: $\lambda_0 := 1$, $d_0 := \frac{6}{h_1} \left(\frac{f_1 - f_0}{h_1} - f'_0 \right)$, $\mu_{n+1} := 1$ und $d_{n+1} := \frac{6}{h_{n+1}} \left(f'_{n+1} - \frac{f_{n+1} - f_n}{h_{n+1}} \right)$

und erhalten für die Fälle i. und iii. das lineare Gleichungssystem

$$\begin{bmatrix} 2 & \lambda_0 & & & 0 \\ \mu_1 & 2 & \lambda_1 & & \\ & \mu_2 & \ddots & \ddots & \\ & & \ddots & 2 & \lambda_n \\ 0 & & & \mu_{n+1} & 2 \end{bmatrix} \begin{bmatrix} M_0 \\ M_1 \\ \vdots \\ M_n \\ M_{n+1} \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_n \\ d_{n+1} \end{bmatrix}, \tag{93}$$

bzw. für den Fall ii.

$$\begin{bmatrix} 2 & \lambda_1 & & & \mu_1 \\ \mu_2 & 2 & \lambda_2 & & \\ & \mu_3 & \ddots & \ddots & \\ & & \ddots & 2 & \lambda_n \\ \lambda_{n+1} & & & \mu_{n+1} & 2 \end{bmatrix} \begin{bmatrix} M_1 \\ M_2 \\ \vdots \\ M_n \\ M_{n+1} \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \\ d_{n+1} \end{bmatrix}. \tag{94}$$

Damit ist das Problem der kubischen Spline Interpolation gelöst, denn es gilt der Satz

Satz 4.23. Die Gleichungssysteme (93) und (94) sind für jede Zerlegung $\Delta : a = x_0 < x_1 < \cdots < x_{n+1} = b$ des Intervalls $[a, b]$ eindeutig lösbar.

Beweis: Wir betrachten i. und iii., der Fall ii. wird analog behandelt. Sei

$$A := \begin{bmatrix} 2 & \lambda_0 & & & 0 \\ \mu_1 & 2 & \lambda_1 & & \\ & \mu_2 & \ddots & \ddots & \\ & & \ddots & 2 & \lambda_n \\ 0 & & & \mu_{n+1} & 2 \end{bmatrix} \in \text{MAT}(n+2, n+2).$$

Wir zeigen

$$Az = w \Longrightarrow \|z\|_\infty \leq \|w\|_\infty,$$

denn mit dieser Eigenschaft würde aus $Az = 0$ sofort $z = 0$, also die Injektivität des der Matrix A zugeordneten Endomorphismus' folgen. Sei also $\|z\|_\infty = |z_r|$. Dann gilt sicher wegen $\lambda_r + \mu_r = 1$ ($\mu_0 = 0 = \lambda_{n+1}$)

$$\|w\|_\infty \geq |w_r| \geq 2|z_r| - \mu_r|z_{r-1}| - \lambda_r|z_{r+1}| \geq 2|z_r| - \mu_r|z_r| - \lambda_r|z_r| \geq (2 - \mu_r - \lambda_r)|z_r| = \|z\|_\infty.$$

■

Bemerkung 4.24.

Analoges Vorgehen beweist, daß **strikt diagonaldominante** Matrizen regulär sind. Dabei heißt $A \in \text{Mat}(n, n)$ **strikt diagonaldominant**, falls A das starke Zeilensummenkriterium (34) erfüllt. Die System Matrizen aus (93) und (94) sind offensichtlich strikt diagonaldominant.

Die Gleichungssysteme in (93) und (94) sind mit O(n) arithmetische Operationen auflösbar. Im periodischen Fall ii. wird die Lösung des Gleichungssystems (94) mit Hilfe der **Sherman-Morrison-Woodbury Formel**

$$\left(B + UV^T\right)^{-1} = B^{-1} - B^{-1}U\left(I + V^T B^{-1} U\right)^{-1} V^T B^{-1}. \tag{95}$$

auf die Lösung von vier tridiagonalen Gleichungssystemen zurück geführt. Stellen wir uns geschickt an, so reicht auch die Lösung von 2 tridiagonalen Gleichungssystemen. In (95) ist $B \in \mathbb{R}^{n\times n}$ regulär, und $U,\, V \in \mathbb{R}^{n\times m}$ mit $m \leq n$. Natürlich müssen alle in (95) auftretenden Inversen existieren → Übung.

Kubische Spline Interpolierende verdienen ihren Namen, denn es gilt mit

$$h_{\max} := \max_{i=1,\ldots,n+1} h_i (= \|\Delta\|) \text{ und } h_{\min} := \min_{i=1,\ldots,n+1} h_i$$

Satz 4.25. Sei $f \in C^4([a, b])$. Die kubischen Spline Interpolierenden S_Δ zu den Daten $(x_i, f_i := f(x_i))$ $(i = 0, \ldots, n+1)$ und den Forderungen iii. (Hermite Splines) erfüllen die Fehlerabschätzungen

$$\|f^{(l)} - S_\Delta^{(l)}\|_\infty \leq C \frac{h_{\max}}{h_{\min}} h_{\max}^{4-l} \|f^{(4)}\|_\infty \text{ für } l = 0, 1, 2, 3, \tag{96}$$

wobei C eine positive Konstante bezeichnet, die nicht von $h_{\max}$ und $h_{\min}$ abhängt.

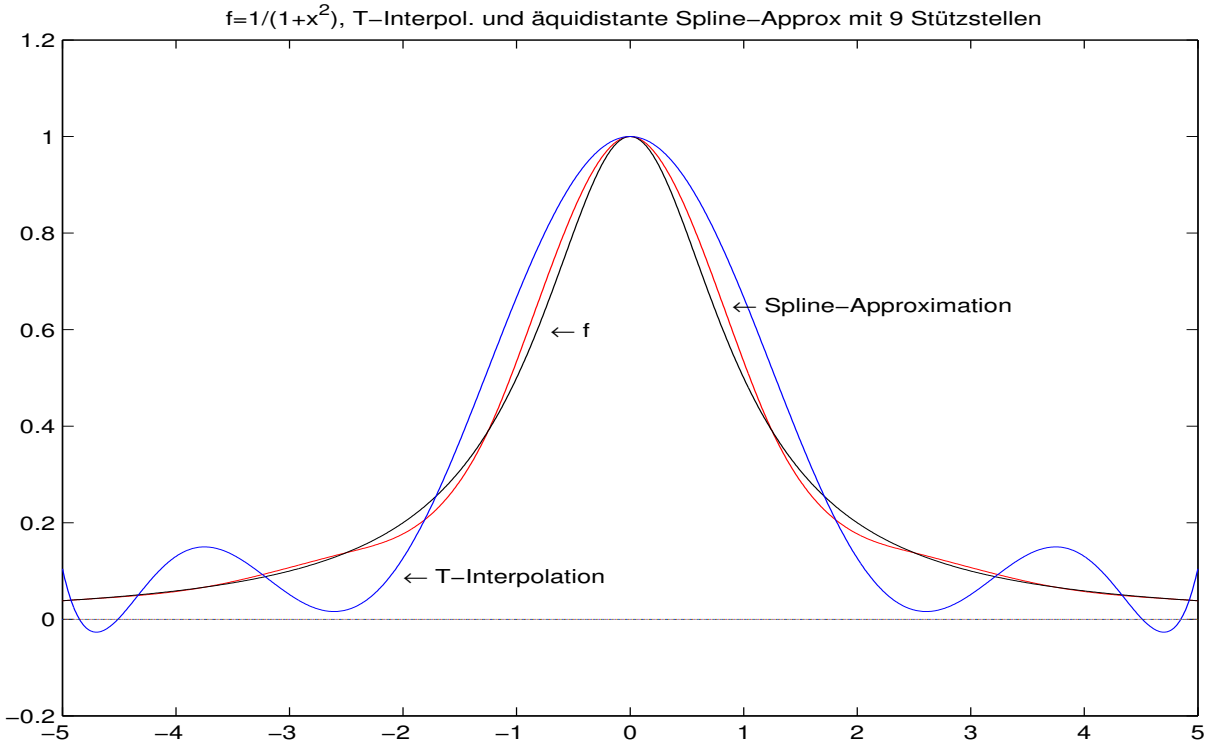


Abbildung 5: Spline- versus Tschebyscheff Interpolation, Splines auf äquidistanten Stützstellen

Einen Beweis dieser Aussage finden Sie etwa in [2, (2.4.3.3) Satz].

Zur Illustration vergleichen wir in Abb. 5 die Approximationsgüte der kubischen Spline-Approximation mit der der Tschebyschev-Interpolation am Beispiel der Funktion $f(x) = 1/(1+x^2)$, zunächst für äquidistante Spline-Stützstellen.

Werden die Stützstellen optimal gewählt, so kann man in der Zeichnung Funktion und Spline kaum mehr unterscheiden. Der Fehler der Spline-Approximation im gesamten Intervall ist dann $< 2.2 * 10^{-3}$, siehe Abb. 6.

Bemerkung 4.26. Spline Funktionen mit Eigenschaft i., ii. und f periodisch oder iii. mit $f'_0 = f'(a)$ und $f'_{n+1} = f'(b)$ haben noch die schöne Eigenschaft, daß sie unter allen 2 mal stetig differenzierbaren Funktionen g mit $g(x_i) = f_i$ $(i = 0, \ldots, n+1)$ diejenigen mit kleinster Gesamtkrümmung $\int_a^b |g''(x)|^2 dx$ sind, d.m.

$$\int_a^b |S''_\Delta(x)|^2 dx \leq \int_a^b |g''(x)|^2 dx \text{ für alle Funktionen } g \in C^2([a,b]) \text{ mit } g(x_i) = f_i (i = 0, \ldots, n+1).$$

vergleiche Aufgabenblatt 8. Dazu sei bemerkt, daß $g''(x)$ die exakte Krümmung $\frac{g''(x)}{\sqrt{1+g'(x)^2}^3}$ für kleine Ableitungen $g'(x)$ gut approximiert.

Jetzt noch kurz Interpolation mit Splines k-ter Ordnung.

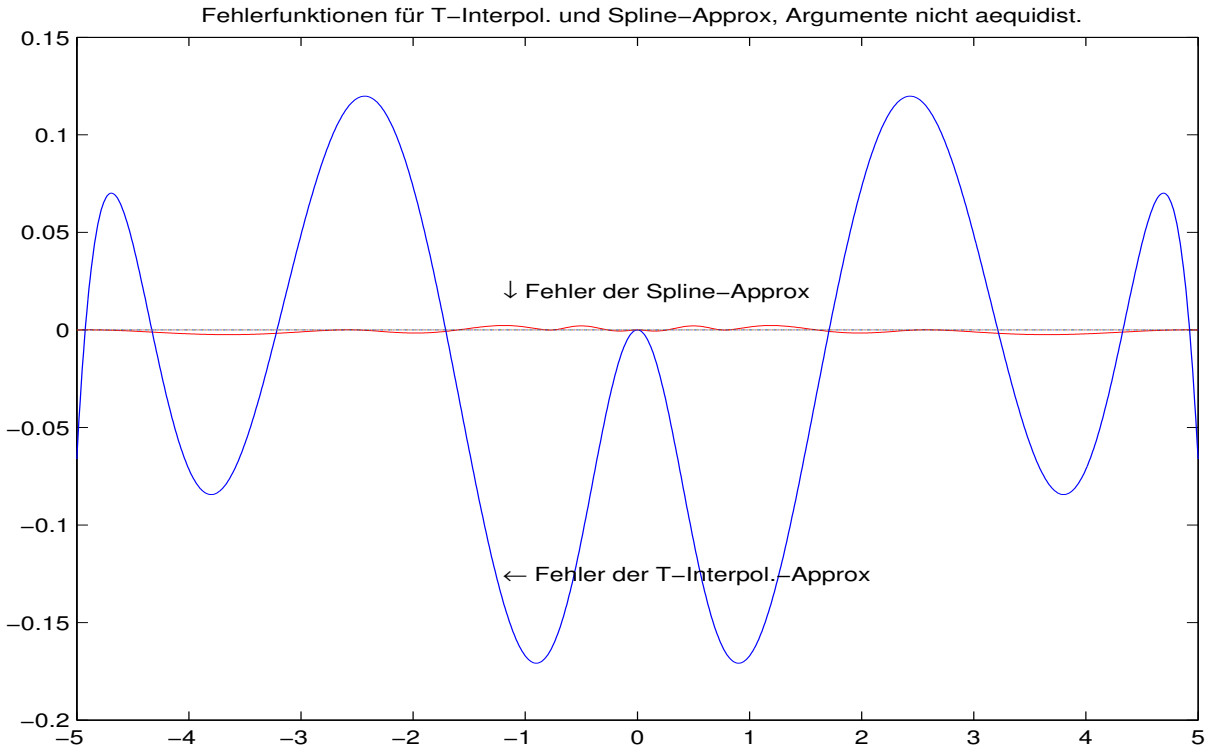


Abbildung 6: Spline- versus Tschebyscheff Interpolation, Fehler, jeweils auf optimal gewählten Stützstellen

4.2.2 Splines k-ter Ordnung

Wir ergänzen zunächst unsere Zerlegung Δ wie folgt um Hilfsgitterpunkte;

$$x_{-k} < \dots x_{-1} < a = x_0 < x_1 < \cdots < x_{n+1} = b < x_{n+2} < \cdots < x_{n+k+1}$$

und defineren für $i = -k, \dots, n$ zu x_i insgesamt $n + k + 1$ Funktionen $B_{i,k}$ wie folgt;

$$B_{i,0}(x) := \begin{cases} 1, & x \in [x_i, x_{i+1}) \\ 0, & \text{sonst} \end{cases} \quad \text{für } i = -k, \dots, n+k,$$

und rekursiv

$$B_{i,l}(x) := \frac{x - x_i}{x_{i+l} - x_i} B_{i,l-1}(x) + \frac{x_{i+l+1} - x}{x_{i+l+1} - x_{i+1}} B_{i+1,l-1}(x) \text{ für } l = 1, \dots k \text{ und } i = -k, \dots, n. \tag{97}$$

Die Funktionen $B_{i,k}$ heißen **B-Splines k-ter Ordnung** zur Zerlegung Δ. Es sei bereits hier bemerkt, daß die obige Konstruktion der B-Splines genau auf unsere Interpolationsaufgabe zugeschnitten ist. **B-Splines k-ter Ordnung** können auch ohne den Interpolationskontext definiert werden, siehe etwa [2, Kapitel 2.4.4].

Einige Eigenschaften der Funktionen $B_{i,k}$ sind zusammengefaßt in

Satz 4.27. Die in (97) definierten Funktionen $B_{i,k}$

sind nicht negativ,

sind linear unabhängig,

bilden eine Partition der 1, d.m. $\sum_i B_{i,k}(x) = 1$,

haben Träger (=Bereich, wo Funktion $\neq 0$ ist) $[x_i, x_{i+k+1}]$,

sind eingeschränkt auf $[x_i, x_{i+1}]$ Polynome vom Grad k und

sind $k-1$ mal stetig differenzierbar,

dessen Beweis Ihnen als Übungsaufgabe überlassen wird. Siehe dazu auch [2, (2.4.4.5) Satz]. Die letzten beiden Eigenschaften besagen, daß es sich bei den Funktionen $B_{i,k}$ um Splines k-ter Ordnung handelt, vergleiche Definition 4.21.

Wir sind nach diesen Vorbereitungen in der Lage, das Interpolationsproblem zu den Daten (x_i, f_i) $(i = 0, \ldots, n+1)$ mit Splines k-ter Ordnung zu lösen. Dazu machen wir für den interpolierenden Spline k-ter Ordnung den Ansatz

$$S_{\Delta,k}(x) := \sum_{i=0}^{n+1} \alpha_i B_{i-1,k}(x)$$

und fordern

$$S_{\Delta,k}(x_i) = f_i \text{ für } i = 0, \ldots, n+1.$$

In Matrix Schreibweise erhalten wir das $(n+2) \times (n+2)$ Gleichungssystem

$$\begin{bmatrix} B_{-1,k}(x_0) & \ldots & B_{n,k}(x_0) \\ \vdots & & \vdots \\ B_{-1,k}(x_{n+1}) & \ldots & B_{n,k}(x_{n+1}) \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \vdots \\ \alpha_{n+1} \end{bmatrix} = \begin{bmatrix} f_0 \\ \vdots \\ f_{n+1} \end{bmatrix} \Longleftrightarrow A\alpha = f. \tag{98}$$

Wir bemerken sofort, daß die Koeffizientenmatrix A Bandstruktur mit Bandbreite $k-1$ aufweist $(k \geq 1)$. Darüber hinaus ist die Matrix A regulär, das Interpolationsproblem in unserer Formulierung demnach eindeutig lösbar. Schließlich ergibt einfaches Vergleichen, daß mit den **Hütchen Funktionen** aus (90) die Beziehung

$$b_i = B_{i-1,1}$$

erfüllt wird, die Interpolationsaufgabe für lineare Splines also tatsächlich (89) liefert.

Abschließend sei bemerkt, daß die Interpolationsaufgabe auch für Stützstellen $a \leq \xi_0 < \cdots < \xi_{n+1} \leq b$ zu Stützwerten $f_0, \ldots, f_{n+1}$ formuliert werden kann. Das entsprechende Gleichungssystem verändert sich dann nur marginal;

$$\begin{bmatrix} B_{-1,k}(\xi_0) & \ldots & B_{n,k}(\xi_0) \\ \vdots & & \vdots \\ B_{-1,k}(\xi_{n+1}) & \ldots & B_{n,k}(\xi_{n+1}) \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \vdots \\ \alpha_{n+1} \end{bmatrix} = \begin{bmatrix} f_0 \\ \vdots \\ f_{n+1} \end{bmatrix}. \tag{99}$$

Nach [2, (2.4.5.7) Satz] besitzt (99) genau eine Lösung, wenn für die Diagonalelemente $B_{i-1,k}(\xi_i) \neq 0$ für $i = 0, \ldots, n+1$ erfüllt ist.

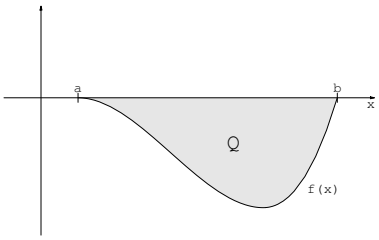
5 Numerische Integration

Im Rahmen dieser Veranstaltung beschränken wir uns auf die numerische Integration (auch numerische Quadratur genannt) von reellwertigen Funktionen f einer reellen Variablen. Berechnet werden soll

$$I(f) \quad = \quad \int_a^b f(x)\,dx\,.$$

Die Notwendigkeit der numerischen Integration verdeutlichen nachfolgende Beispiele.

1. Um die Wassermenge zu bestimmen, die ein Fluss pro Zeiteinheit transportiert, ist es nötig, den Flussquerschnitt $Q = \int_a^b -f(x)\,dx$ zu bestimmen. Dazu wird die Flusstiefe $f(x)$ an mehreren Stellen gemessen. Da sie damit nur an einzelnen Punkten bekannt ist, kann das Integral nicht mit Hilfe einer Stammfunktion integriert werden.



2. Bei vielen Integrationsaufgaben in der Mathematik (z.B. beim Lösen von Differentialgleichungen oder der Längenbestimmung von Kurven) ist die zu integrierende Funktion zwar bekannt, aber nicht geschlossen integrierbar, d.h. die Stammfunktion kann nicht explizit angegeben werden. Einfache Beispiele hierfür sind etwa $f(x) = e^{-x^2}$, $f(x) = \sin\, x^2$, $f(x) = \frac{\sin\, x}{\sqrt{x}}$. Auch hier muss man numerisch vorgehen.

5.1 Interpolatorische Quadratur

Die naheliegende Vorgehensweise besteht darin, $f(x)$ durch ein Interpolationspolynom $p_n(x)$ zu ersetzen und $\int_a^b p_n(x)\,dx$ anstelle von $\int_a^b f(x)\,dx$ zu berechnen. Wir bezeichnen mit

$$I_n(f) \quad = \quad \int_a^b p_n(x)\,dx$$

eine Integrationsformel mit einem Interpolationspolynom n-ter Ordnung für das exakte Integral

$$I(f) \quad = \quad \int_a^b f(x)\,dx.$$

$I_n(f)$ integriert demnach Polynome bis zum Höchstgrad n exakt.
Wird die Lagrange–Form des Interpolationspolynoms benutzt (vgl. Satz 4.8)

$$p_n(x) \;=\; \sum_{j=0}^{n} L_j(x)\, f(x_j)\,, \qquad L_j(x) \;=\; \prod_{\substack{\nu=0\\ \nu\neq j}}^{n} \frac{(x-x_\nu)}{(x_j-x_\nu)}\,,$$

so ist durch Integration unmittelbar einsichtig, dass interpolatorische Integrationsformeln die folgende Gestalt haben

$$I_n(f) \quad = \quad \int_a^b p_n(x)\,dx \quad = \quad \sum_{j=0}^{n} A_j\, f(x_j)$$

mit von den x_j abhängigen Gewichten (100)

$$A_j \quad = \quad \int_a^b L_j(x)\,dx\,.$$

Wählt man die Interpolationsknoten x_j äquidistant, so heißen die Integrationsformeln (100) **Newton–Cotes–Formeln**.

Am gebräuchlichsten sind die Formeln für $n = 0$ (Mittelpunktregel), für $n = 1$ (Trapezregel) und für $n = 2$ (Simpsonregel), die wir zunächst herleiten.

n = 0:

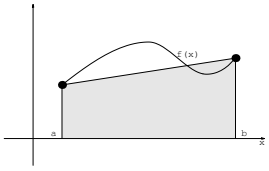
$$x_0 \;=\; \frac{a+b}{2}, \qquad p_0(x) \;=\; f\left(\frac{a+b}{2}\right), \qquad A_0 \;=\; \int_a^b 1\,dx, \qquad \text{also}$$

$$I_0(f) := (b-a) f\left(\frac{a+b}{2}\right) \quad \textbf{(Mittelpunktregel)}. \tag{101}$$

n = 1:

$$x_0 \;=\; a, \quad x_1 \;=\; b, \quad p_1(x) \;=\; \frac{x-b}{a-b}\,f(a) \;+\; \frac{x-a}{b-a}\,f(b)\,,$$

$$A_0 \;=\; \int_a^b \frac{x-b}{a-b}\,dx \;=\; \frac{b-a}{2}\,, \qquad A_1 \;=\; \int_a^b \frac{x-a}{b-a}\,dx \;=\; \frac{b-a}{2}\,,$$



also

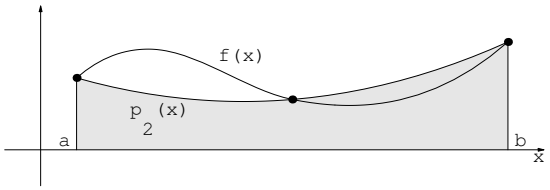
$$I_1(f) = \frac{b-a}{2}\;(f(a)+f(b)) \quad \textbf{(Trapezregel)} \tag{102}$$

n = 2:

$$x_0 \;=\; a, \quad x_1 \;=\; \frac{a+b}{2}\,, \quad x_2 \;= b\,, \quad A_0 \;=\; \int_a^b L_0(x)\,dx \;=\; \frac{b-a}{6}\,,$$

$$A_1 = \int_a^b L_1(x)\,dx = \frac{2}{3}(b-a),$$

$$A_2 = \int_a^b L_2(x)\,dx = \frac{b-a}{6},$$



und damit

$$I_2(f) = \int_a^b p_2(x)\,dx = \frac{b-a}{6}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right) \quad \textbf{(Simpsonregel)} \tag{103}$$

5.2 Quadraturfehler bei interpolatorischen Verfahren

Mit Hilfe der Darstellung des Interpolationsfehlers in Satz 4.14) erhalten wir durch Integration über den Interpolationsfehler eine Darstellung für den Integrationsfehler. Diese Idee wollen wir zunächst verfolgen. Aus Satz 4.14a) erhalten wir

$$I(f) - I_n(f) = \int_a^b f(x) - p_n(x)dx = \int_a^b f[x_0, \ldots, x_n, x] \prod_{j=0}^{n}(x - x_j)dx.$$

Im Fall $n = 1$ (Trapezregel) gilt somit ($x_0 = a, x_1 = b$) unter Verwendung des verallgemeinerten Mittelwertsatzes und Ausnutzung von $(x-a)(b-x) \geq 0$ auf $[a,b]$ mit einem $\tilde{\xi} \in [a,b]$

$$I(f) - I_1(f) = \int_a^b f[a,b,x](x-a)(x-b)dx =$$

$$= -f[a,b,\tilde{\xi}] \int_a^b (x-a)(b-x)dx = -f[a,b,\tilde{\xi}]\frac{(b-a)^3}{6}.$$

Ist zusätzlich $f \in C^2([a,b])$, so erhalten wir mit Satz 4.14b) für den Quadraturfehler der Trapezregel

$$I(f) - I_1(f) = -f''(\xi)\frac{(b-a)^3}{12} \text{ mit einem } \xi \in [a,b]. \tag{104}$$

Bemerkung 5.1. Auf Grund ihrer Herleitung integriert die Trapezregel Funktionen f, die Polynome vom Grad ≤ 1 sind, exakt (lineare Interpolation). Dies spiegelt sich auch in der Fehlerdarstellung wieder. Für Polynome 1. Grades ist $f'' = 0$, der Quadraturfehler also $= 0$.

Entsprechend wird man erwarten, dass die Simpsonregel Polynome vom Grad ≤ 2 exakt integriert. In der Tat integriert sie jedoch sogar Polynome vom 3. Grad exakt, **sofern man die Stützstellen geeignet wählt (etwa wie oben äquidistant)**. Die Ursache hierfür liegt in der Gültigkeit von

Hilfsatz 5.2. Sei $n \in \mathbb{N}_0$ gerade und seien paarweise verschiedene Knoten $x_j \in [a,b]$ gegeben, welche symmetrisch angeordnet sind, d.h. es gelte $x_j - a = b - x_{n-j}$ für $j = 0, \ldots, n$. Ferner

bezeichne p_n das Interpolationspolynom n-ten Grades zu f auf diesen Stützstellen und p_{n+1} das Interpolationspolynom vom Grade $n+1$ zu f mit den Stützstellen $x_i (i = 0, \ldots, n)$ und $z \in (a,b),\ z \neq x_j,\ j = 0, \ldots, n$. Dann gilt

$$\int_a^b p_n(x)\,dx = \int_a^b p_{n+1}(x)\,dx$$

Beweis: Für das Interpolationspolynom p_{n+1} in der Newton–Form gilt nach (73)

$$p_{n+1}(x) \;=\; p_n(x) \;+\; f[x_0, \ldots, x_n, z] \prod_{j=0}^{n} (x - x_j)\,, \quad \text{also}$$

$$\int_a^b p_{n+1}(x)\,dx \;=\; \int_a^b p_n(x)\,dx \;+\; f[x_0, \ldots, x_n, z] \int_a^b \prod_{j=0}^{n} (x - x_j)\,dx\,.$$

Nun ist aber mit $\omega(x) = \prod_{j=0}^{n} (x - x_j)$ hier $\omega(a+s) = -\omega(b-s)$, demnach

$$\int_a^b \omega(x)dx = 0.$$

■

Dieses Ergebnis hat zur Folge, dass man für die Darstellung des Quadraturfehlers der Simpsonregel den Interpolationsfehler für Interpolationspolynome 3. Grades benutzen kann, also mit $z \neq a, b, \frac{a+b}{2}$ und dem verallgemeinerten Mittelwertsatz

$$I_2(f) - I(f) = \int_a^b f(x) - p_2(x)dx = \int_a^b f(x) - p_3(x)dx =$$

$$= \int_a^b f[a, \frac{a+b}{2}, b, z, x](x-a)(x-\frac{a+b}{2})(x-b)(x-z)dx \to$$

$$\to \int_a^b f[a, \frac{a+b}{2}, \frac{a+b}{2}, b, x](x-a)(x-\frac{a+b}{2})^2(x-b)dx(z \to \frac{a+b}{2}) =$$

$$= -f[a, \frac{a+b}{2}, \frac{a+b}{2}, b, \tilde{\xi}] \int_a^b (x-a)(x-\frac{a+b}{2})^2(b-x)dx = -f[a, \frac{a+b}{2}, \frac{a+b}{2}, b, \tilde{\xi}]\frac{(b-a)^5}{120},$$

wobei $\tilde{\xi} \in [a,b]$. Die linke Seite dieser Ungleichung ist unabhängig von z, do dass der Grenzübergang $z \to \frac{a+b}{2}$ durchgeführt werden darf, ohne die Gültigkeit der Ungleichung zu verletzen. Ist $f \in C^4([a,b])$, so erhalten wir für den Quadraturfehler der Simpsonregel

$$I_2(f) - I(f) = -\frac{f^{(4)}(\xi)}{4!}\frac{(b-a)^5}{120} = -f^{(4)}(\xi)\frac{(b-a)^5}{2880} \text{ mit einem } \xi \in [a,b]. \qquad (105)$$

Vollkommen analoges Vorgehen liefert für den Quadraturfehler der Mittelpunktregel mit Hilfsatz 5.2 die Darstellung

$$I_0(f) - I(f) = -\frac{(b-a)^3}{24} f''(\xi) \text{ mit einem } \xi \in [a,b]. \tag{106}$$

Für gerades $n \in \mathbb{N}_0$ erhalten wir für Stützstellen, welche symmetrisch in $[a,b]$ liegen, die Darstellung

$$\begin{aligned} I_n(f) - I(f) &= \int_a^b f(x) - p_n(x)dx = \int_a^b f(x) - p_{n+1}(x)dx = \\ &= \int_a^b f[x_0,\ldots,x_n,z,x](x-z)\prod_{j=0}^{n}(x-x_j)dx = \\ &= \int_a^b f[x_0,\ldots,x_n,z,x](x-z)(x-x_{n/2})\prod_{j=0}^{n/2-1}(x-x_j)\prod_{j=n/2+1}^{n}(x-x_j)dx = \longrightarrow \\ &\to \int_a^b f[x_0,\ldots,x_n,z,x](x-x_{n/2})^2\prod_{j=0}^{n/2-1}(x-x_j)(x-x_{(n-j)})dx \text{ (für } z \to x_{n/2}). \end{aligned}$$

Der Mittelwertsatz ist nicht mehr ohne Weiteres anwendbar. Doch gestattet diese Darstellung des Fehlers dessen Abschätzung in gewohnter Manier;

$$\begin{aligned} |I_n(f)-I(f)| &\le (b-a)\|f[x_0,\ldots,x_n,z,x]\|_\infty \max_{x\in[a,b]} |(x-x_{n/2})^2\prod_{j=0}^{n/2-1}(x-x_j)(x-x_{(n-j)})| \le \\ &\le \frac{\|f^{n+2}\|_\infty}{(n+2)!}(b-a)(b-a)^{n/2}(b-a)^{n/2+2}2^{-(n/2+2)} = \frac{\|f^{n+2}\|_\infty}{(n+2)!}(b-a)^{n+3}2^{-(n/2+2)}, \end{aligned}$$

wobei wir in der letzten Ungleichung $f \in C^{n+2}([a,b])$ voraussetzen müssen.

Wir wollen der Vollständigkeit noch die Newton–Cotes Formeln höherer Ordnung angeben. Dazu sei

$$a =: x_0 < \cdots < x_n := b \text{ mit } x_i = a + ih, \text{ mit } h := \frac{b-a}{n}$$

eine äquidistante Unterteilung des Intervalls $[a,b]$. Mit den A_i aus (100) definieren wir

$$\frac{(b-a)\sigma_i}{ns} := A_i \text{ für } i = 0,\ldots,n,$$

wobei s so gewählt sei, dass σ_i ganzzahlig ist. Die Gewichte σ_i sind in Tabelle 1 bis $n = 6$ angegeben, vergleiche auch [2, Kapitel 3.1].

Für größere Werte von n treten auch negative Gewichte σ_i auf und die Formeln werden numerisch unbrauchbar, siehe etwa Diskussion [2, Kapitel3.1]. Die Newton-Cotes Formeln aus Tabelle 1 heißen abgeschlossen, weil für die gewählte Unterteilung $\Delta : a = x_0 < \cdots < x_n = b$ gilt, Intervallanfang und -ende demnach Stützstellen der Quadratur sind. Ein Vertreter der **Offenen Newton-Cotes Formeln** ist die **Mittelpunkt Regel** aus (101). Sie basiert auf der Wahl des Intervallmittelpunktes als Stützstelle für die Quadratur mit einem Polynom 0–ten Grades.

n	σ_1	σ_2	σ_3	σ_4	σ_5	σ_6	σ_7	ns	Fehler	Name
1	1	1						2	$h^3\frac{1}{12}f^{(2)}(\xi)$	Trapez Regel
2	1	4	1					6	$h^5\frac{1}{90}f^{(4)}(\xi)$	Simpson Regel
3	1	3	3	1				8	$h^5\frac{3}{80}f^{(4)}(\xi)$	$\frac{3}{8}$ Regel
4	7	32	12	32	7			90	$h^7\frac{8}{945}f^{(6)}(\xi)$	Milne Regel
5	19	75	50	50	75	91		288	$h^7\frac{275}{12096}f^{(6)}(\xi)$	ohne Namen
6	41	216	27	272	27	216	41	840	$h^9\frac{9}{1400}f^{(8)}(\xi)$	Weddle Regel

Tabelle 1: Abgeschlossene Newton-Cotes Formeln für $n = 1, \ldots, 6$

5.3 Zusammengesetzte Formeln

Ist das Intervall $[a, b]$ zu groß, ist das Interpolationspolynom $p_n \in \Pi_n$ nicht immer ein gutes Modell für die zu integrierende Funktion f. In solchen Fällen ist es wieder sinnvoller, auf den Teilintervallen $[x_{i-1}, x_i]$ einer Unterteilung $\Delta : a := x_0 < x_1 < \cdots < x_n := b$ stückweise zu interpolieren und als Quadraturformel für f die Summe der Newton-Cotes Quadraturen von f auf den einzelnen Teilintervallen zu verwenden. Sei dazu jetzt die Unterteilung Δ wieder äquidistant, d.m. $x_i = a + ih$, $h := \frac{b-a}{n}$.

5.3.1 Summierte Trapez Regel

Auf $[x_{i-1}, x_i]$ wird linear interpoliert, dort also die Trapez Regel zur Quadratur verwendet. D.m.

$$\int_{x_{i-1}}^{x_i} f(x)dx \approx \frac{h}{2}\left(f(x_{i-1}) + f(x_i)\right).$$

Die **Summierte Trapez Regel** ergibt sich dann zu

$$T(h) := \frac{h}{2}\sum_{i=1}^{n}\left\{(f(x_{i-1}) + f(x_i))\right\} = \\ = h\left[\frac{f(a)}{2} + f(a+h) + f(a+2h) + \cdots + f(a+(n-1)h) + \frac{f(b)}{2}\right]. \quad (107)$$

Der Fehler

$$T(h) - \int_a^b f(x)dx$$

ergibt sich für $f \in C^2$ als Summe der Fehler der Trapezregel über die Teilintervalle $[x_{i-1}, x_i]$ nach Tabelle 1 mit $h = \frac{b-a}{n}$ zu

$$T(h) - \int_a^b f(x)dx = \frac{h^2}{12}(b-a)\sum_{i=1}^{n}\frac{1}{n}f''(\xi_i) \text{ mit } \xi_i \in [x_{i-1}, x_i].$$

Anwendung des Zwischenwertsatzes auf f'' liefert die Existenz eines $\xi \in (\min \xi_i, \max \xi_i) \subseteq (a, b)$ mit

$$f''(\xi) = \frac{1}{n}\sum_{i=1}^{n} f''(\xi_i),$$

also für den Gesamtfehler

$$T(h) - \int_a^b f(x)dx = \frac{h^2}{12}(b-a)f''(\xi). \tag{108}$$

5.3.2 Summierte Simpson Regel

Ist n gerade, so kann die Simpson Regel zur Quadratur der Funktion f auf den Teilintervallen $[x_{2i-2}, x_{2i}]$ für $i = 1, \ldots, \frac{n}{2}$ angewendet werden. Deren Länge beträgt $2h$, so daß

$$\int_{x_{2i-2}}^{x_{2i}} f(x)dx \approx \frac{h}{3}\left(f(x_{2i-2}) + 4f(x_{2i-1}) + f(x_{2i})\right)$$

richtig ist. Die **Summierte Simpson Regel** ergibt sich damit zu

$$S(h) := \frac{h}{3}\sum_{i=1}^{\frac{n}{2}}\left(f(x_{2i-2}) + 4f(x_{2i-1}) + f(x_{2i})\right) =$$
$$\frac{h}{3}\left[f(a) + 4f(a+h) + 2f(a+2h) + \cdots + 2f(a+(n-2)h) + 4f(a+(n-1)h) + f(b)\right], \tag{109}$$

für deren Gesamtfehler gilt

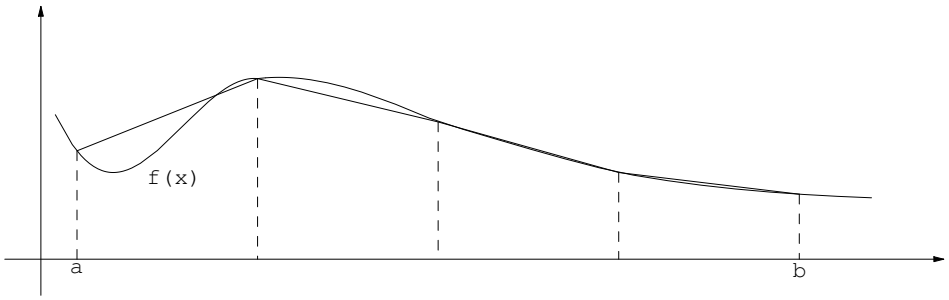
$$S(h) - \int_a^b f(x)dx = \frac{h^4}{180}(b-a)f^{(4)}(\xi) \text{ mit einem } \xi \in (a,b). \tag{110}$$

In analoger Weise können wir mit Hilfe der anderen Newton-Cotes Formeln auf Teilintervallen weitere zusammengesetzte Quadraturformeln erhalten.

Bemerkung 5.3. Die Fehlerdarstellungen bedeuten anschaulich: Wird die Schrittweite halbiert (n verdoppelt), so sinkt bei der zusammengesetzten Trapezregel die Fehlerschranke um den Faktor $\left(\frac{1}{2}\right)^2$, bei der zusammengesetzten Simpsonformel sogar um den Faktor $\left(\frac{1}{2}\right)^4$.

5.4 Adaptive Quadraturformeln

Fehlerschranken bieten die Möglichkeit, vor Beginn der Rechnung abzuschätzen, wie groß die Schrittweite h gewählt werden muss, damit der Fehler eine vorgegebene Schranke nicht übersteigt, vorausgesetzt natürlich, man kennt Schranken für die betreffenden Ableitungen. Dies ist oft nicht der Fall oder mit großen Mühen verbunden. Außerdem sind Formeln mit äquidistanten Stützstellen oft mit zu großem Rechenaufwand verbunden, weil sie auch in Regionen, in denen es nicht nötig ist, die Schrittweite verkleinern, wie man schon an folgendem einfachen Beispiel für die zusammengesetzte Trapezregel erkennt.



Es ist deshalb sinnvoll, nach Verfahren zu suchen, welche die Schrittweite selbst an die Eigenschaften der Funktion f anpassen (adaptieren), also kleine Schrittweiten wählen, wenn sich die Funktion (oder das Integral) stark ändern, große, wenn dies nicht der Fall ist.

Ziel: Entwickle ein Verfahren mit Schrittweitenanpassung, das für eine vorgegebene Funktion $\int_a^b f(x)\,dx$ bis auf einen vorgegebenen Fehler $\varepsilon > 0$ genau berechnet.

Ausgangspunkt der Überlegungen ist 1), dass man weiß, wie „in etwa“ sich der Fehler bei Intervallhalbierungen verhält (vgl. die obige Bemerkung) und 2), dass die numerische Differenz der Integrationswerte bei Intervallhalbierung ja numerisch anfällt. Beide Fakten kann man benutzen, um den wirklichen Integrationsfehler zu **schätzen** und daraus eine **Schätzung** für eine geeignete Schrittweite abzuleiten. Wir wollen dies (in einer Rohform – Verfeinerungen sind möglich –) für die zusammengesetzte Simpsonregel demonstrieren.

Sei also m die (vorläufig noch unbekannte) Zahl von Teilintervallen $I_j = [x_j, x_{j+1}]$ von $[a,b]$ der (vorläufig noch unbekannten) Länge $h_j = x_{j+1} - x_j$. Wir bezeichnen den exakten Integralwert im Intervall $[x_j, x_{j+1}]$ durch

$$I^j(f) \quad = \quad \int_{x_j}^{x_{j+1}} f(x)\,dx\,,$$

die Simpsonformel mit der Gitterweite $\frac{h_j}{2}$ in I_j durch (vergl. (103))

$$S^j(f) \quad = \quad \frac{h_j}{6}\left[f(x_j) \;+\; 4f\left(x_j + \frac{h_j}{2}\right) \;+\; f(x_{j+1})\right],$$

und die Simpsonformel mit halbierter Gitterweite $\frac{h_j}{4}$ durch

$$Q^j(f) \;=\; \frac{1}{6}\,\frac{h_j}{2}\left[f(x_j) + 4f\left(x_j + \frac{h_j}{4}\right) + 2f\left(x_j + \frac{h_j}{2}\right) + 4f\left(x_j + \frac{3h_j}{4}\right) + f(x_{j+1})\right].$$

Aus der Fehlerabschätzung (105) folgt die Existenz einer Konstanten c_j, so dass gilt

$$\left|I^j(f) - S^j(f)\right| = c_j\, h_j^5$$

Halbiert man das Intervall $[x_j, x_{j+1}]$ und wendet die Fehlerabschätzung auf beide Hälften an, so existieren Konstanten c_{j_1} und c_{j_2}, so dass

$$\left|I^j(f) - Q^j(f)\right| \quad = \quad (c_{j_1} + c_{j_2})\left(\frac{h_j}{2}\right)^5. \tag{111}$$

Wir nehmen nun an, dass sich c_{j_1} und c_{j_2} nicht wesentlich von c_j unterscheiden. Dies ist um so eher erfüllt, je weniger sich $f^{(4)}$ ändert, d.h. u.a. je kleiner die Teilintervalle werden. Setzt man also $c_{j_1} \approx c_{j_2} \approx c_j$, so geht (111) über in

$$\left|I^j(f) - Q^j(f)\right| \quad \approx \quad 2c_j \left(\frac{h_j}{2}\right)^5 = \frac{1}{16}\, c_j\, h_j^5 \approx \frac{1}{16}\left|I^j(f) - S^j(f)\right|.$$

1. Variante: Fehlerschätzer für S^j

Es gilt

$$\left|I^j(f) - S^j(f)\right| \le \left|I^j(f) - Q^j(f)\right| + \left|Q^j(f) - S^j(f)\right| \approx \frac{1}{16}\left|I^j(f) - S^j(f)\right| + \left|Q^j(f) - S^j(f)\right|.$$

Daraus folgt

$$\left|I^j(f) - S^j(f)\right| \lesssim \frac{16}{15}\left|Q^j(f) - S^j(f)\right|.$$

Die rechte Seite ist berechenbar und kann zur Kontrolle des Fehlers $|I^j(f) - S^j(f)|$ auf dem j-ten Teilintervall verwendet werden.

Bemerkung: Ersetzen wir den Faktor $\frac{16}{15}$ durch eine größere Zahl, so stimmt die Abschätzung auch noch, wenn sich c_j, c_{j_1} und c_{j_2} stärker unterscheiden.

2. Variante: Fehlerschätzer für Q^j

Da Q^j genauer ist als S^j, wäre es zu bevorzugen, S^j zu verwenden, um den Fehler von Q^j abzuschätzen.

Dies ist tatsächlich möglich:

$$\left|I^j(f) - Q^j(f)\right| \;\approx\; \frac{1}{16}\left|I^j(f) - S^j(f)\right| \;\le\; \frac{1}{16}\left(\left|I^j(f) - Q^j(f)\right| + \left|Q^j(f) - S^j(f)\right|\right),$$

bzw.

$$\left|I^j(f) - Q^j(f)\right| \quad \lesssim \quad \frac{1}{15}\left|Q^j(f) - S^j(f)\right|.$$

Bemerkung: Der Faktor $\frac{1}{15}$ hängt relativ sensibel von der Präzision der Annahme $c_{j_1} \approx c_{j_2} \approx c_j$ ab, so dass die Fehlerabschätzung für Q^j weniger robust ist als jene für S^j.

Wäre z.B. $c_{j_1} + c_{j_2} = 3c_j$ (anstelle von $2c_j$), dann hätten wir $|I^j(f) - Q^j(f)| = \frac{3}{32}|I^j(f) - S^j(f)|$ und

$$|I^j(f) - Q^j(f)| \le \frac{3}{29}|Q^j(f) - S^j(f)|.$$

Andererseits haben wir

$$|I^j(f) - S^j(f)| \le \frac{32}{29}|Q^j(f) - S^j(f)|.$$

Nun zurück zu unseren Fehlerabschätzungen:

Für den Gesamtfehler (wir diskutieren hier nur die zweite Variante) folgt

$$\begin{aligned}\left|I(f)-\sum_{j=0}^{m-1} Q^j(f)\right| &= \left|\sum_{j=0}^{m-1}\left(I^j(f)-Q^j(f)\right)\right| \\ &\le \sum_{j=0}^{m-1}\left|I^j(f)-Q^j(f)\right| \\ &\lesssim \tfrac{1}{15}\sum_{j=0}^{m-1}\left|Q^j(f)-S^j(f)\right| .\end{aligned} \tag{112}$$

Die Werte $|Q^j(f)-S^j(f)|$ sind nach der Intervallhalbierung und der numerischen Integration ja bekannt. Genügt nun h_j der **Forderung**

$$\left|Q^j(f)-S^j(f)\right| \quad\le\quad \frac{15h_j\cdot\varepsilon}{b-a}\,, \tag{113}$$

so folgt wegen $\sum_{j=0}^{m-1} h_j = b-a$ aus (113) und (112)

$$\left|I(f)-\sum_{j=0}^{m-1} Q^j(f)\right| \quad\lesssim\quad \varepsilon\,. \tag{114}$$

Umsetzung in ein Verfahren

Start: $x_0 := a,\ x_1 := b,$ berechne S^0, Q^0 und prüfe (113) für $h_0 = (b-a)$. Ist (113) erfüllt $\longrightarrow$ Ende, andernfalls

$$x_0 = a,\quad x_1 = \frac{a+b}{2},\quad x_2 := b,\quad I_0 = \left[a, \frac{a+b}{2}\right],\quad I_1 = \left[\frac{a+b}{2}, b\right],\quad h_0 = h_1 = \frac{b-a}{2}.$$

Prüfe (113) für beide Teilintervalle.

Für das (die) Teilintervall(e), in dem (denen) (113) nicht erfüllt ist, wird die Schrittweite halbiert, usw.

Ist (113) in allen Teilintervallen erfüllt, gilt (114).

Dass das beschriebene (sehr einfache) Verfahren ein Schätzverfahren ist (wie alle raffinierteren adaptiven Verfahren auch), belegt folgendes einfache

Beispiel. Berechne für $f(x) = \cos 4x$ das Integral $\int_0^{2\pi} \cos 4x\,dx$. Nun ist $\cos 4x = 1$ für $x = j\,\frac{\pi}{2},\ j = 0,1,2,3,4$. Also gilt $S^0(f) = 2\pi,\ Q^0(f) = 2\pi,$ das Verfahren bricht ab, weil (113) erfüllt ist, und liefert den Wert 2π statt 0.

Natürlich passiert dieser Zusammenbruch nicht, wenn man gleich mit einer höheren Zahl von Stützstellen beginnt, doch ist immer Vorsicht geboten.

5.5 Extrapolation und Romberg Integration

Die Idee der **Extrapolation** ist sehr einfach; Gegeben sei ein **Funktional** $F(h)$ zur näherungsweisen Auswertung der Funktion R. Wir wollen davon ausgehen, daß mit $h \to 0$ auch $F(h) \to R$ erfüllt ist. Ferner seien $0 < h_n < \cdots < h_0$ Stützstellen mit bekannten Stützwerten $F(h_i)$. Ein naheliegender Gedanke besteht nun darin, den nach **0 extrapolierten Wert** $p_n(0)$ des Interpolationspolynoms p_n zu den Daten $(h_i, F(h_i))$ als verbesserte Approximation von R zu betrachten.

Mit Hilfe der **Euler-Maclaurinschen Summenformel** läßt sich zeigen [2, Kapitel 3.4], daß für die Trapezsumme (107) eine Entwicklung in Potenzen von h^2 angegeben werden kann; präziser sei $f \in C^{2m+2}([a,b])$. Dann gilt

$$T(h) = \tau_0 + \tau_1 h^2 + \cdots + \tau_m h^{2m} + \alpha_{m+1}(h) h^{2m+2} \tag{115}$$

mit

$$\tau_0 = \int_a^b f(x)dx \text{ und } \alpha_{m+1}(h) = \frac{B_{2m+2}}{(2m+2)!}(b-a) f^{(2m+2)}(\xi(h)), \xi(h) \in (a,b),$$

wobei $B_i := (-1)^{i-1} \left[\frac{2i-1}{2(2i+1)} + (2i)! \sum_{k=1}^{i-1} \frac{B_k}{(2i-2k+1)!(2k)!} \right]$ die **Bernoulli Zahlen** bezeichnen.

Funktioniert Extrapolation nach 0 (und ob sie funktioniert), wird sich mit dem Interpolationspolynom

$$p_{2m}(h) = b_0 + b_1 h^2 + \cdots + b_m h^{2m}$$

zu den Daten $(h_i, T(h_i))$ mittels $p_{2m}(0) = b_0$ eine verbesserte Approximation des Integrals $\int_a^b f(x)dx$ ergeben. Ihr Fehler hängt natürlich von der Wahl der Folge $\{h_i\}_{i \in \mathbb{N}}$ ab. In der Praxis werden die Folgen

Definition 5.4.

- $h_0 = b - a$, $h_i = \frac{h_{i-1}}{2}$, $i \geq 1$ (**Romberg Folge**), bzw.
- $h_0 = b - a$, $h_1 = \frac{h_0}{2}$, $h_2 = \frac{h_0}{3}, \ldots, h_i = \frac{h_{i-2}}{2}$, $i \geq 3$ (**Bulirsch Folge**)

verwendet. Verbleibt noch bei gegebener Folge $\{h_i\}_{i \in \mathbb{N}}$ mit Stützwerten $T(h_i)$ die Berechnung von $p_{2m}(h)$ und dessen Auswertung an der Stelle 0. Dazu verwenden wir dividierte Differenzen aus Definition 4.9 zur Berechnung der dividierten Differenzen und schließlich das Neville Schema aus Satz 4.12 zur Auswertung des Interpolationspolynoms an der Stelle 0. Hierbei ist zu beachten, daß bei der Interpolation

$$h_0^2, h_1^2, \ldots h_m^2 \text{ als Stützstellen zu verwenden sind.}$$

Abschließend ein

Beispiel. 1. Sei $h_0 = b - a$, $h_1 = \frac{b-a}{2}$. Wir interpolieren zu den Daten

$$(h_0^2, T(h_0)) \text{ und } (h_1^2, T(h_1))$$

und erhalten

$$p_2(x) = T(h_0) + \frac{T(h_1) - T(h_0)}{h_1^2 - h_0^2}(x - h_0^2),$$

also nach kurzer Rechnung

$$p_2(0) = \frac{4}{3}T(h_1) - \frac{1}{3}T(h_0).$$

Wegen

$$T(h_0) = \frac{b-a}{2}\left(f(a) + f(b)\right) \text{ und } T(h_1) = \frac{b-a}{4}\left(f(a) + 2f(\frac{a+b}{2}) + f(b)\right)$$

ergibt sich

$$p_2(0) = \frac{b-a}{6}\left(f(a) + 4f(\frac{a+b}{2}) + f(b)\right),$$

also gerade die Simpson Regel, deren Fehler von der Ordnung h^4 ist. Der Fehler der Trapez Regel hingegen war nur von der Ordnung h^2.

2. Analog wird mit $h_2 = \frac{b-a}{4}$ über $p_4(0)$ die Milne Regel erhalten. Nachweis als Übungsaufgabe.

Bemerkung 5.5. Für gewisse Folgen können bei der Berechnung des Stützwertes $T(h_i)$ Funktionswerte verwendet werden, die schon bei der Berechnung von $T(h_{i-1})$ Verwendet wurden. So gilt etwa für die Romberg Folge

$$T(h_{i+1}) = T\left(\frac{h_i}{2}\right) = \frac{1}{2}T(h_i) + h_{i+1}\left[f(a + h_{i+1}) + f(a + 3h_{i+1}) + \ldots, +f(b - h_{i+1})\right].$$

Abschließend noch eine Fehlerabschätzung für die **Romberg Integration** (=Extrapolation der summierten Trapez Regel mit der Romberg Folge) nach [2, Kapitel 3.4].

Satz 5.6. Für den Fehler in der Romberg Integration gilt im Fall $f \in C^{2m+2}([a,b])$ die Fehler Darstellung

$$p_{2m}(0) - \int_a^b f(x)dx = (b-a)h_0^2h_1^2 \ldots h_m^2 \frac{(-1)^m B_{2m+2}}{(2m+2)!} f^{(2m+2)}(\xi) \text{ mit einem } \xi \in (a,b).$$

5.6 Gauß Quadratur

Bisher haben wir bei der Quadratur noch nicht weiter über die Wahl der Stützstellen $x_i \in [a,b]$ nachgedacht. Bei der Polynominterpolation in Kapitel 3 hat sich das ausgezahlt. Es wird sich herausstellen, daß mit Hilfe der Nullstellen x_i von gewissen **Orthogonalpolynomen** Gewichte w_i $(i = 1, \ldots, n)$ bestimmt werden können, so daß die Quadraturformeln der Form

$$\sum_{i=1}^{n} w_i f(x_i) \approx \int_a^b f(x)dx$$

bereitgestellt werden können, die Polynome bis zum Grad $2n-1$ exakt integrieren. Zur Einführung von Orthogonalpolynomen definieren wir zunächst für quadrat-integrierbare Funktionen f und g

$$(f,g) := \int_a^b f(x)g(x)dx.$$

Diese Form definiert ein Skalarprodukt auf dem Raum

$$L^2(a,b) := \{v : (a,b) \to \mathbb{R}; v \text{ meßbar und } \int_a^b |v(x)|^2 dx < \infty\}$$

der über (a,b) **quadrat-integrierbaren Funktionen**. Den folgenden Satz entnehmen wir [2, (3.6.3) Satz]. Sein Beweis gelingt vollständiger Induktion und mit Hilfe des **Gram-Schmidt Orthogonalisierungsverfahrens** (siehe Vorlesung).

Satz 5.7. Es gibt zu $j = 0, 1, 2, \ldots$ eindeutig bestimmte Polynome $p_j \in \Pi_j$ mit höchstem Koeffizienten 1 mit

$$(p_i, p_j) = 0 \text{ für } i \neq j. \tag{116}$$

Diese Polynome erfüllen die Rekursionsformel

$$p_0(x) = 1,\ p_1(x) = \left(x - \frac{(xp_0, p_0)}{(p_0, p_0)}\right) p_0(x) \text{ und}$$
$$p_{i+1}(x) = \left(x - \frac{(xp_i, p_i)}{(p_i, p_i)}\right) p_i(x) - \frac{(p_i, p_i)}{(p_{i-1}, p_{i-1})} p_{i-1}(x) \text{ für } i \geq 1. \tag{117}$$

Wir kommen jetzt zum Hauptresultat.

Satz 5.8.

i. Mit den Nullstellen $x_1, \ldots x_n$ von p_n bezeichnen $w_1 \ldots, w_n$ die Lösung des linearen Gleichungssystems

$$\sum_{i=1}^n p_k(x_i) w_i = \begin{cases} (p_0, p_0), & \text{falls } k = 0, \\ 0, & \text{falls } k \geq 1. \end{cases} \tag{118}$$

Dann gilt $w_i > 0$ für $i = 1, \ldots, n$ und

$$\sum_{i=1}^n w_i p_i(x_i) = \int_a^b p(x)dx \text{ für alle } p \in \Pi_{2n-1}, \tag{119}$$

d.m. Polynome vom Grad $\leq 2n-1$ werden von

$$Q(f) := \sum_{i=1}^n w_i f(x_i) \tag{120}$$

exakt integriert.

ii. Gilt für reelle Zahlen x_i, w_i $(i = 1, \ldots, n)$ (119) für alle $p \in \Pi_{2n-1}$, so sind die x_i Nullstellen von p_n und die reellen Zahlen w_i erfüllen das Gleichungssystem (118).

iii. Es gibt keine reellen Zahlen x_i, w_i $(i = 1, \ldots, n)$, so daß (119) für alle $p \in \Pi_{2n}$ richtig ist.

Der Beweis dieses Satzes ist etwa in [2, (3.6.12) Satz] zu finden und benutzt die nachfolgend genannten Eigenschaften der Orthogonalpolynome p_i.

Hilfsatz 5.9. Es gilt

i. $(p, p_n) = 0$ für alle $p \in \Pi_{n-1}$,

ii. die Nullstellen $x_1, \ldots, x_n$ von p_n sind reell, einfach und liegen im offenen Intervall (a, b),

iii. für beliebige $t_1 < t_2 < \cdots < t_n$ ist die $n \times n$ Matrix

$$\begin{bmatrix} p_0(t_1) & \ldots & p_0(t_n) \\ \vdots & & \vdots \\ p_{n-1}(t_1) & \ldots & p_{n-1}(t_n) \end{bmatrix} \tag{121}$$

nicht singulär. Das meint, daß $p_0, p_1, p_2, \ldots$ ein **Tschebyscheff System** bilden. Die Bedingung (121) heißt **Haar'sche Bedingung**.

Beweis:

i. $p \in \Pi_{n-1}$, dann p Linearkombination der $p_0, \ldots, p_{n-1}$, also $p \perp p_n$.

ii. Seien $a < x_1 \cdots < x_l < b$ die Nullstellen von p_n, an denen ein Vorzeichenwechsel stattfindet. Gilt $l < n$, so ist mit

$$q(x) := \prod_{i=1}^{l} (x - x_i) \in \Pi_l$$

wegen i. $(q, p_n) = 0$. Es ändert aber qp_n auf (a, b) sein Vorzeichen nicht, also kann wegen der Definition des Skalarproduktes $(\cdot, \cdot)$ diese Orthogonalität nicht erfüllt sein. Widerspruch.

iii. Sei $c \neq 0$ mit $A^t c = 0$. Dann hat $q(x) := \sum_{i=0}^{n-1} c_i p_i(x)$ die n parweise verschiedenen Nullstellen t_i, verschwindet demnach identisch. Ebenso die Ableitungen bis zur Ordnung $n - 1$. Der höchste Koeffizient der p_i ist jeweils 1, ihr Grad demnach i, also

$$0 = q^{(n-1)} = (n-1)! c_{n-1} \Rightarrow c_{n-1} = 0,\ q^{(n-2)} = 0 = (n-2)! c_{n-2} \Rightarrow c_{n-2} = 0,\ \ldots, \cdots \Rightarrow c_0 = 0.$$

Das ist ein Widerspruch zu $c \neq 0$.

Damit ist alles gezeigt. ■

Bemerkung 5.10.
1. Für $[a,b] = [-1,1]$ gehen die Resultate aus Satz 5.8 auf Gauß zurück, die zugehörigen Orthogonalpolynome sind (bis auf Normierung) die **Legendre Polynome**

$$p_k(x) := \frac{k!}{(2k)!}\frac{d^k}{dx^k}\left(x^2-1\right)^k,\ k = 0,1,\ldots$$

2. Obige Ausführungen können auch auf Skalarprodukte der Form

$$(f,g) := \int_a^b f(x)g(x)w(x)dx.$$

mit positiven **Gewichtsfunktionen** w verallgemeinert werden. Für spezielle Gewichtsfunktionen w und Intervalle sind die Orthogonalpolynome bekannt.

- $[a,b] = [-1,1]$, $w(x) = \frac{1}{\sqrt{1-x^2}}$. Orthogonalpolynome sind die Tschebyscheff Polynome T_n aus Satz 4.17.
- $[a,b] = [0,\infty]$, $w(x) = e^{-x}$. Orthogonalpolynome sind die **Laguerre Polynome** L_n.
- $[a,b] = [-\infty,\infty]$, $w(x) = e^{-x^2}$. Orthogonalpolynome sind die **Hermite Polynome** H_n.

3. Die Gewichte w_i und Nullstellen x_i sind in den gängigen Tafelwerken tabelliert.

Verbleibt noch die praktische Berechnung der Nullstellen x_i und der Gewichte w_i in Formel (119). Dazu definieren wir zu gegebenen Skalarprodukt $(\cdot,\cdot)$ und Orthogonalpolynomen p_k

$$d_{i+1} := \frac{(xp_i,p_i)}{(p_i,p_i)} \text{ für } i \geq 0, \quad g_{i+1} := \begin{cases} 0 & \text{für } i = 0 \\ \sqrt{\frac{(p_i,p_i)}{(p_{i-1},p_{i-1})}} & \text{für } i \geq 1. \end{cases}$$

Dann gilt [2, (3.6.20),(3.6.21) Satz]

Satz 5.11. Die Nullstellen x_i des n-ten Orthogonalpolynoms p_n sind gegeben durch die Eigenwerte der Tridiagonalmatrix

$$J_n := \begin{bmatrix} d_1 & g_2 & & & \\ g_2 & d_2 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & g_n \\ & & & g_n & d_n \end{bmatrix},$$

die Gewichte w_i in (120) sind gegeben durch

$$w_k = \left(v_1^k\right)^2,\ k = 1,\ldots,n,$$

wobei $v^k = [v_1^k,\ldots,v_n^k]^t$ den k-ten Eigenvektor zum Eigenwert x_k von J_n mit der Normierung $|v^k|_2^2 = (p_0,p_0)$ bezeichnet.

Damit ist die Quadratur nach Gauß auf die Berechnung von Eigenwerten und Eigenvektoren von Tridiagonalmatrizen zurückgeführt.

Abschließend noch der Quadraturfehler bei der Gauß Integration.

Satz 5.12. Mit den oben eingeführten Bezeichnungen gilt für $f \in C^{2n}([a,b])$

$$\sum_{i=1}^{n} w_i f(x_i) - \int_a^b f(x)dx = \frac{f^{(2n)}(\xi)}{(2n)!}(p_n, p_n) \tag{122}$$

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

Beweis: Der Beweis speziell dieser Aussage findet sich in [2, (3.6.24) Satz].

6 Lineare Optimierung

Um die Aufgabenstellung deutlich zu machen, beginnen wir mit einem (natürlich sehr vereinfachten) Beispiel:

Produktionsplan einer (zugegebenermaßen sehr kleinen) Schuhfabrik. Hergestellt werden sollen Damen– und Herrenschuhe, und zwar jeweils nur ein Modell. Die Produktionsbedingungen ergeben sich aus der folgenden Tabelle.

		Damenschuh	Herrenschuh	verfügbar
Herstellungszeit	$[h]$	20	10	8000
Maschinenbearbeitung	$[h]$	4	5	2000
Lederbedarf	$[dm^2]$	6	15	4500
Reingewinn	$[Euro]$	16	32	

Unter der Annahme, dass keine Absatzschwierigkeiten entstehen, soll berechnet werden, wieviele Damen– und Herrenschuhe hergestellt werden müssen, damit der Gewinn optimal wird, natürlich unter Einhaltung obiger Restriktionen.

Mathematische Formulierung:

Sei x_1 die Zahl der produzierten Damenschuhe ,

x_2 die Zahl der produzierten Herrenschuhe .

Dann lauten die Produktionsbedingungen:

$$\begin{cases} 20x_1 + 10x_2 \leq 8000 & \text{(i)} \\ 4x_1 + 5x_2 \leq 2000 & \text{(ii)} \\ 6x_1 + 15x_2 \leq 4500 & \text{(iii)} \\ x_1 \geq 0 & \text{(iv)} \\ x_2 \geq 0 & \text{(v)} \end{cases} \tag{123}$$

Gesucht sind Zahlen (x_1, x_2), die diesen Ungleichungen genügen und den Gewinn

$$f(x_1, x_2) := 16x_1 + 32x_2$$

maximieren.

Die Funktion f heißt *Zielfunktion*, die Ungleichungen (123) heißen *Nebenbedingungen*. Das Problem lautet also

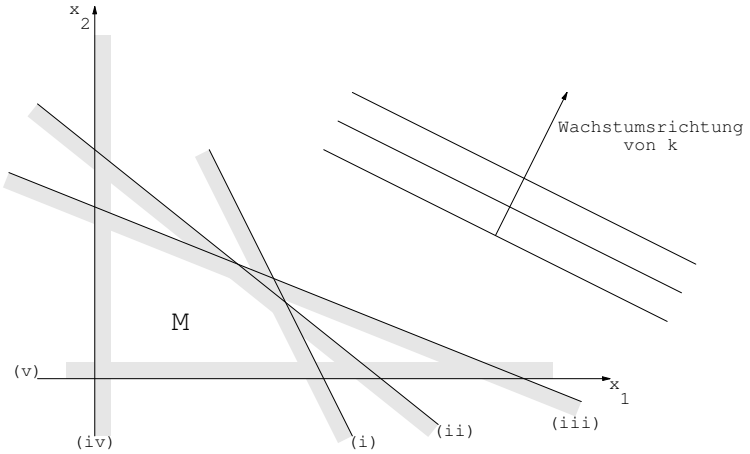
maximiere $16x_1 + 32x_2$

unter den Nebenbedingungen

$$\begin{aligned} 20x_1 + 10x_2 &\leq 8000 \\ 4x_1 + 5x_2 &\leq 2000 \\ 6x_1 + 15x_2 &\leq 4500 \\ x_1 &\geq 0 \\ x_2 &\geq 0. \end{aligned}$$

In diesem Beispiel kann man sich die Lösung geometrisch veranschaulichen. Gleichzeitig legt es die Lösungsidee für den allgemeinen Fall nahe.

Die Ungleichungen (123) beschreiben Halbebenen in $\mathbb{R}^2$, die durch Geraden begrenzt werden (vgl. Zeichnung). Der Durchschnitt M aller dieser Halbebenen beschreibt den *zulässigen Bereich* aller der Punktepaare (x_1, x_2), die den Ungleichungen (123) genügen. $f(x_1, x_2) = 16x_1 + 32x_2 = k$, mit variablem k, beschreibt eine Geradenschar. Es ist ein maximales k_m derart zu bestimmen, dass (mindestens) ein Punkt des zulässigen Bereichs M auf der Geraden $f(x_1, x_2) = k_m$ liegt.



Die geometrische Lösung ist nun einfach. Man verschiebt die Geradenschar so lange im Sinne fallender k, bis zum ersten Mal ein Punkt aus M auf einer Geraden liegt oder, falls die Geradenschar die gleiche Steigung hat wie eine Begrenzungsstrecke von M, sogar eine ganze Strecke. Dieser Punkt oder alle Punkte der genannten Strecke optimieren die Zielfunktion.

Im allgemeinen Fall wird ein lineares Optimierungsproblem die folgende Gestalt haben

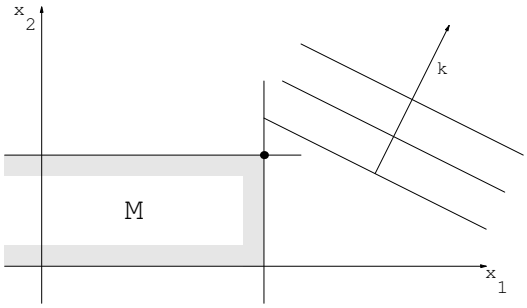
Standardform:

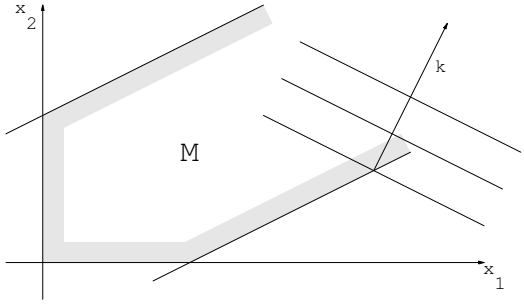
$$\left.\begin{array}{lrcrcrcrcl} \text{min} & c_1\,x_1 & + & c_2\,x_2 & + & \ldots & + & c_n\,x_n & & \\ \text{u.d.N.} & a_{11}\,x_1 & + & a_{12}\,x_2 & + & \ldots & + & a_{1n}\,x_n & \leq & b_1 \\ & a_{21}\,x_1 & + & \ldots & & & + & a_{2n}\,x_n & \leq & b_2 \\ & \vdots & & & & & & & & \\ & a_{m1}\,x_1 & + & & & & + & a_{mn}\,x_n & \leq & b_m \\ & & & & & & & x_i & \geq & 0, \quad i \in J. \end{array}\right\} \quad \text{(L')}$$

Hierbei sind $c_j, a_{ij}, b_i \in \mathbb{R}$ sowie $J \subset \{1, \ldots, n\}$ gegeben und die $x_j \in \mathbb{R}$ sind gesucht.

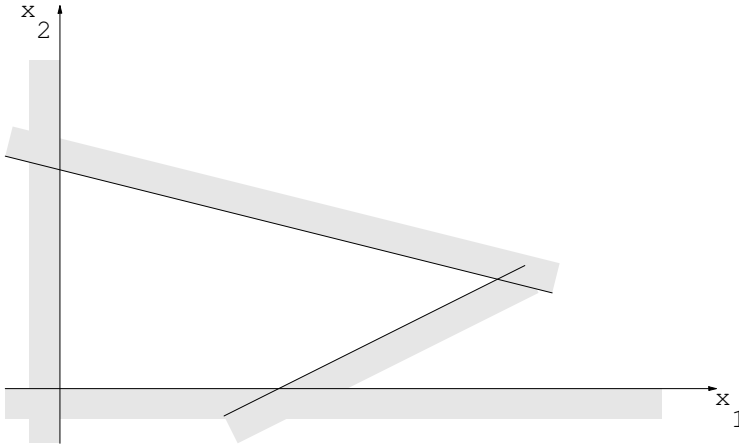
Wir wollen uns an einigen Beispielen klar machen, was je nach Beschaffenheit von M (zulässiger Bereich) und Zielfunktion passieren kann. In den Bildern ist $k = -f(\boldsymbol{x})$, $f(\boldsymbol{x}) = \sum_{j=1}^{n} c_j x_j$.

1) falls M nicht beschränkt ist, kann f ein Minimum annehmen, es muss dies aber nicht.

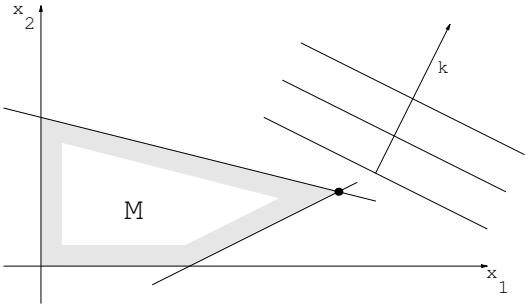


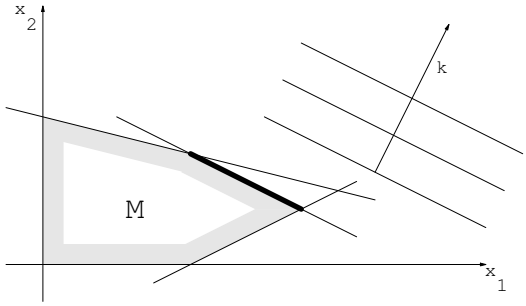


2) $M = \emptyset$ ist möglich.

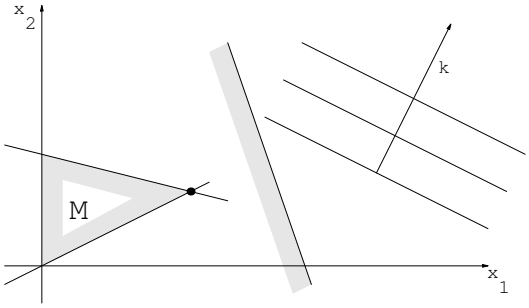


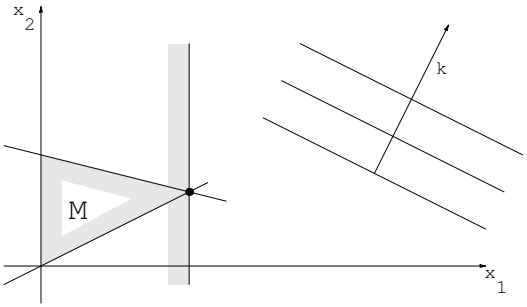
3) Die optimale Lösung kann, muss aber nicht eindeutig sein.





4) Es gibt überflüssige Nebenbedingungen.





Fazit:
In allen Beispielen gilt: Wenn eine optimale Lösung existiert, dann wird sie (vielleicht nicht nur, aber auch) in einer Ecke des zulässigen Bereichs angenommen.

Wir schreiben das Problem (L′) kurz in Matrixschreibweise. Mit $\boldsymbol{c} = (c_1, \ldots, c_n)^T \in \mathbb{R}^n$, $\boldsymbol{A} = (a_{ij}) \in \mathbb{R}^{m \times n}$, $\boldsymbol{x} = (x_1, \ldots, x_n)^T \in \mathbb{R}^n$, $\boldsymbol{b} = (b_1, \ldots, b_m)^T \in \mathbb{R}^m$, $J \subset \{1, \ldots, n\}$ folgt

$$\left.\begin{array}{rlll} \min & \boldsymbol{c}^T \boldsymbol{x} & & \\ \text{u.d.N.} & \boldsymbol{A}\,\boldsymbol{x} & \leq & \boldsymbol{b} \\ & x_i & \geq & 0, \quad i \in J. \end{array}\right\} \quad \text{(L')}$$

$\boldsymbol{c}^T \boldsymbol{x}$ heißt *Zielfunktion*, $M = \{\boldsymbol{x} \in \mathbb{R}^n, \ \boldsymbol{A}\,\boldsymbol{x} \leq \boldsymbol{b}, \ x_i \geq 0, \ i \in J\}$ *zulässiger Bereich*. $\boldsymbol{A}\,\boldsymbol{x} \leq \boldsymbol{b}$ ist im Sinne obiger Ungleichungen komponentenweise zu verstehen. Die Elemente $\in M$ heißen *zulässige Punkte* (zulässige Lösungen) und ein zulässiges $\boldsymbol{x} \in M$ heißt *optimal*, wenn für alle zulässigen Vektoren $\boldsymbol{y} \in M$ gilt $\boldsymbol{c}^T \boldsymbol{x} \geq \boldsymbol{c}^T \boldsymbol{y}$.

Natürlich sind auch andere Formulierungen von (L') möglich und gebräuchlich. Dies hängt von der jeweiligen Aufgabenstellung ab. Wir stellen sie hier kurz zusammen und zeigen, wie sie sich ineinander überführen lassen (um einen möglichst allgemeinen Typ von Optimierungsaufgaben behandeln zu können).

a) Eine Maximierungsaufgabe wird zu einer Minimierungsaufgabe durch Übergang zum Negativen der Zielfunktion.

$$\boldsymbol{c}^T \boldsymbol{x} \ = \ \max \qquad \Longleftrightarrow \qquad -\boldsymbol{c}^T \boldsymbol{x} \ = \ \min$$

b) Eine Ungleichung

$$a_{i1}\,x_1 \ + \ \ldots \ + \ a_{in}\,x_n \ \leq \ b_i$$

kann durch Einführen einer *Schlupfvariablen* $y_i \geq 0$ in eine Gleichung

$$a_{i1}\,x_1 \ + \ \ldots \ + \ a_{in}\,x_n \ + \ y_i \ = \ b_i$$

überführt werden.

c) Tritt eine Gleichung als Nebenbedingung auf,

$$a_{j1}\,x_1 \ + \ \ldots \ + \ a_{jn}\,x_n \ = \ b_j\,,$$

so kann sie durch 2 Ungleichungen ersetzt werden:

$$\begin{array}{rcccl} a_{j1}\,x_1 & + & \ldots & + & a_{jn}\,x_n \ \leq \ b_j \\ -a_{j1}\,x_1 & - & \ldots & - & a_{jn}\,x_n \ \leq \ -b_j \end{array}$$

d) Eine Komponente x_i von $\boldsymbol{x}$, für die keine Vorzeichenbedingung besteht, kann ersetzt werden durch den Ausdruck $x_i = x_{i+} - x_{i-}$ und die Forderung $x_{i+} \geq 0, \ x_{i-} \geq 0$.

Vorsicht: Diese Zerlegung muss auch in der Zielfunktion berücksichtigt werden.

e) Jede „$\leq$“–Ungleichung kann durch Multiplikation mit -1 in eine „$\geq$“–Ungleichung überführt werden (und umgekehrt).

Insbesondere kann also jede Optimierungsaufgabe mit linearer Zielfunktion und linearen Nebenbedingungen in die folgende Form gebracht werden

$$\left.\begin{array}{lrcl} \min & \boldsymbol{c}^T \boldsymbol{x} & & \\ \text{u.d.N.} & \boldsymbol{A}\,\boldsymbol{x} & = & \boldsymbol{b} \\ & \boldsymbol{x} & \geq & 0 \end{array}\right\} \quad \text{(L)}$$

wobei $\boldsymbol{c} \in \mathbb{R}^n$, $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $\boldsymbol{b} \in \mathbb{R}^m$ gegeben sind.

Wir verdeutlichen an einem Beispiel die Überführung einer vorgelegten Aufgabe in die Form (L). Man beachte dabei, dass sich bei Einführung von Schlupfvariablen die Zahl der Nebenbedingungen vergrößert und dass insbesondere bei Ersetzung einer nicht vorzeichenbeschränkten Variablen durch 2 Ungleichungen (im Beispiel $x_{1+} \geq 0,\ x_{1-} \geq 0$) sich auch i.a. die Variablenanzahl mit Koeffizienten $\neq 0$ in der Zielfunktion vergrößert.

$$\begin{array}{lrcl} \max & -2x_1 + 3x_2 & & \\ \text{u.d.N.} & x_1 + x_2 & \geq & 5 \\ & -x_1 + x_2 & \leq & 7 \\ & x_1 & \leq & 10 \\ & x_2 & \geq & 0 \end{array}$$

wird zu

$$\begin{array}{lrcl} \min & 2x_{1+} - 2x_{1-} - 3x_2 + 0y_1 + 0y_2 + 0y_3 & & \\ \text{u.d.N.} & x_{1+} - x_{1-} + x_2 - y_1 & = & 5 \\ & -x_{1+} + x_{1-} + x_2 + y_2 & = & 7 \\ & x_{1+} - x_{1-} + y_3 & = & 10 \\ & x_{1+} & \geq & 0 \\ & x_{1-} & \geq & 0 \\ & x_2 & \geq & 0 \\ & y_1 & \geq & 0 \\ & y_2 & \geq & 0 \\ & y_3 & \geq & 0 \end{array}$$

Bemerkungen zur Form (L):

Das Fazit aus den Beispielen 1) – 4) lässt vermuten, dass „Ecken“ bei der Lösung der Optimierungsprobleme eine wesentliche Rolle spielen werden. In $\mathbb{R}^2$ und $\mathbb{R}^3$ lassen sich Ecken als Schnittpunkte von Geraden oder Ebenen, also durch Gleichungen beschreiben. Setzen wir in der Formulierung (L) voraus, dass $\text{Rang}(\boldsymbol{A}) = m$ gilt, so besitzt der zulässige Bereich stets eine Ecke, sofern $M \neq \emptyset$ gilt. Außerdem existiert für den Fall, dass (L) lösbar ist, stets eine optimale Ecke. Wir werden diese Aussagen später nachweisen.

Um zu einer Lösungsidee für das Problem (L) zu kommen, kehren wir nochmals zu den anfangs betrachteten Beispielen 1) – 4) zurück.

Da die optimale Lösung auch immer in einer Ecke angenommen wird, liegt es nahe, alle Ecken zu bestimmen, in ihnen den Zielfunktionswert zu berechnen und die Ecke mit dem optimalen Zielfunktionswert auszuwählen.

Dieses Vorgehen ist problematisch, weil es einerseits, insbesondere bei vielen Ungleichungsrestriktionen, schwierig sein wird, alle Ecken zu bestimmen, und andererseits dabei viele Ecken umsonst berechnet werden.

Erfolgversprechend ist die nächste Idee: Man beginne mit einer Ecke (wie findet man die?), bestimme in dieser Ecke den Zielfunktionswert und gehe dann längs einer Kante, längs der der Zielfunktionswert abnimmt, zu einer Nachbarecke über und bestimme deren Zielfunktionswert usw., bis man am Minimum angelangt ist. (Wie erkennt man das?) Dabei können natürlich alle die Situationen eintreten, die wir in den Beispielen 1) – 4) kennengelernt haben. Insbesondere hoffen wir aber, dass man auf diese Weise nicht alle Ecken durchlaufen muss, bis man am Minimum angekommen ist.

Man mache sich diese Problematik auch an Beispielen im $\mathbb{R}^3$ klar.

Wir haben in der Tat mit der obigen Vorgehensweise die Grundidee des nun zu besprechenden Lösungsalgorithmus beschrieben. Bei der Umsetzung dieser geometrischen Lösungsidee in einen Lösungsalgorithmus für die Aufgabenstellung (L) treten eine Reihe von Problemen auf, wie etwa

- Wie beschreibt man Ecken im $\mathbb{R}^n$?
- Wie beschreibt man Kanten?
- Was heißt „Laufen auf einer Kante“?
- Wird der optimale Wert wirklich in einer Ecke angenommen?
- Was bedeuten die Beobachtungen aus den Beispielen 1) – 4) im allgemeinen Fall?

Diese und weitere auftauchende Fragen wollen wir im folgenden zu lösen versuchen. Dabei werden die Lösungsideen aus der anschaulichen Problemstellung (L') kommen (, die wir der Anschauung halber nochmals, samt ihrer geometrischen Interpretation skizzieren,) und werden dann in die algebraische und algorithmische Sprache für das Problem (L) übersetzt werden müssen.

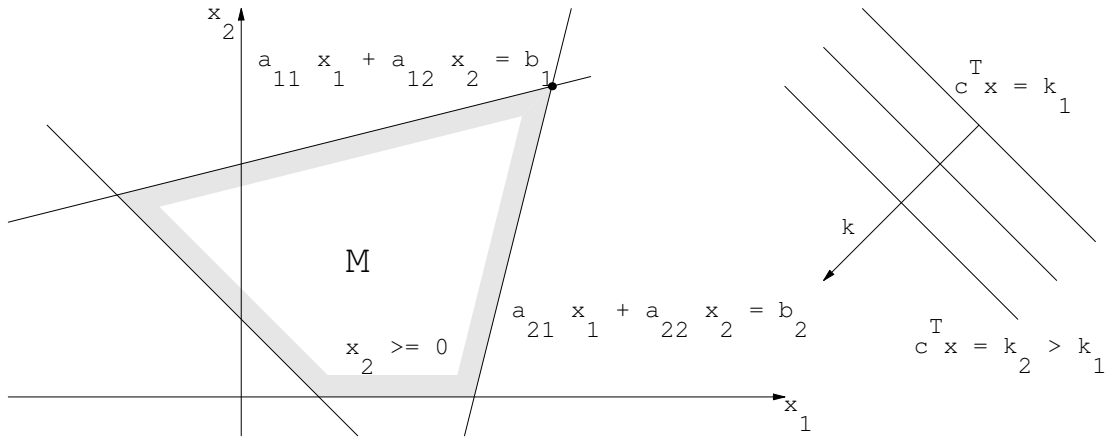
6.1 Lineare Optimierungsaufgaben: Formulierungen

Standardform des Linearen Optimierungsproblems:

$$\left.\begin{array}{ll} \min & \boldsymbol{c}^T\boldsymbol{x} \\ \text{u.d.N.} & \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}, \\ & x_i \geq 0, \;\; i \in J \end{array}\right\} \quad \text{(L')}$$

mit $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $\boldsymbol{c} \in \mathbb{R}^n$, $\boldsymbol{b} \in \mathbb{R}^m$, $J \subset \{1, \ldots, n\}$. Gesucht ist $\boldsymbol{x} \in \mathbb{R}^n$.

Im $\mathbb{R}^2$ (auch $\mathbb{R}^3$) hat man die Veranschaulichung



$M = \{\boldsymbol{x} \in \mathbb{R}^n;\ \boldsymbol{A}\,\boldsymbol{x} \leq \boldsymbol{b},\ x_i \geq 0,\ i \in J\}$ heißt *zulässiger Bereich*, $\boldsymbol{c}^T\boldsymbol{x}$ heißt *Zielfunktion*.

Im Folgenden verwenden wir die

Normalform des linearen Optimierungsproblems:

$$\left.\begin{array}{ll} \min & \boldsymbol{c}^T\boldsymbol{x} \\ \text{u.d.N.} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \\ & \boldsymbol{x} \geq 0 \end{array}\right\} \quad \text{(L)}$$

mit $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $\boldsymbol{c} \in \mathbb{R}^n$, $\boldsymbol{b} \in \mathbb{R}^m$, $J \subset \{1, \ldots, n\}$. Gesucht ist $\boldsymbol{x} \in \mathbb{R}^n$.

$M = \{\boldsymbol{x} \in \mathbb{R}^n;\ \boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{b},\ \boldsymbol{x} \geq \boldsymbol{0}\}$ heißt *zulässiger Bereich*, $\boldsymbol{c}^T\boldsymbol{x}$ heißt *Zielfunktion*, und $\boldsymbol{x}^*$ heißt *optimal*, falls $\boldsymbol{x}^* \in M$ und $\boldsymbol{c}^T\,\boldsymbol{x}^* \leq \boldsymbol{c}^T\,\boldsymbol{x}\ \forall\,\boldsymbol{x} \in M$.

Beachte: Die Vektoren $\boldsymbol{c}, \boldsymbol{b}$ und die Matrix $\boldsymbol{A}$ sind gemäß den Umformungen nicht die gleichen wie in (L').

6.2 Beschreibung von Ecken

Hier und im folgenden behandeln wir jeweils das Problem (L).

Der zulässige Bereich eines linearen Optimierungsproblems wird im $\mathbb{R}^2$ durch Geraden begrenzt, im $\mathbb{R}^3$ durch Ebenen, im $\mathbb{R}^n$ durch sog. Hyperebenen.

Geometrisch anschaulich ist eine Ecke eines solchen Bereichs M ein Punkt $\in M$, der nicht „im Inneren“der Verbindungsstrecke zweier verschiedener Punkte liegt, die auch in M liegen.

Mathematische Fassung dieses Sachverhalts:

Definition 6.1. 1) Eine Menge $K \subset \mathbb{R}^n$ heißt *konvex*

$$\overset{\text{(Def.)}}{\Longleftrightarrow} \quad \begin{cases} \forall\, \boldsymbol{x}^1, \boldsymbol{x}^2 \ \in \ K \quad \text{gilt} \\ \boldsymbol{x} \ = \ \lambda\, \boldsymbol{x}^1 + (1-\lambda)\, \boldsymbol{x}^2 \ \in \ K, \quad 0 \ \leq \ \lambda \ \leq \ 1 \end{cases}$$

$\boldsymbol{x}$ heißt *Konvexkombination* von $\boldsymbol{x}^1$ und $\boldsymbol{x}^2$.

2) Eine Konvexkombination heißt *echt*

$$\overset{\text{(Def.)}}{\Longleftrightarrow} \qquad \lambda \ \neq \ 0,\, 1$$

3) Sei $K \subset \mathbb{R}^n$ eine Menge, die durch Hyperebenen begrenzt wird.
$\boldsymbol{x} \in K$ heißt *Ecke von* K

$$\overset{\text{(Def.)}}{\Longleftrightarrow} \quad \boldsymbol{x} \quad \text{hat keine Darstellung als echte Konvex–kombination 2er verschiedener Punkte von } K$$

Hilfsatz 6.2. Die zulässige Menge M eines linearen Optimierungsproblems (in der Normalform)

$$M \ = \ \{\, \boldsymbol{x} \in \mathbb{R}^n, \ \boldsymbol{A}\, \boldsymbol{x} = \boldsymbol{b}, \ \boldsymbol{x} \geq 0\} \quad \text{wo} \quad \boldsymbol{A} \ \in \ \mathbb{R}^{m \times n}, \ \boldsymbol{b} \ \in \ \mathbb{R}^m, \ m \ < \ n\,,$$

ist konvex.

Beweis: Seien $\boldsymbol{x}^1, \boldsymbol{x}^2 \in M, \ \boldsymbol{x}^1 \neq \boldsymbol{x}^2$ und $\boldsymbol{x} = \lambda\, \boldsymbol{x}^1 + (1-\lambda)\, \boldsymbol{x}^2, \quad \lambda \in [0,1]$, so folgt

$$\begin{aligned} \boldsymbol{A}\, \boldsymbol{x} \ = \ \boldsymbol{A}(\lambda\, \boldsymbol{x}^1 + (1-\lambda)\, \boldsymbol{x}^2) \ &= \ \lambda\, \boldsymbol{A}\, \boldsymbol{x}^1 + (1-\lambda)\, \boldsymbol{A}\, \boldsymbol{x}^2 \\ &= \ \lambda\, \boldsymbol{b} \qquad + (1-\lambda)\, \boldsymbol{b} \ = \ \boldsymbol{b}\,. \end{aligned}$$

Aus $\boldsymbol{x} = \underbrace{\lambda}_{\geq 0}\ \underbrace{\boldsymbol{x}^1}_{\geq 0} \ + \ \underbrace{(1-\lambda)}_{\geq 0}\ \underbrace{\boldsymbol{x}^2}_{\geq 0}$ folgt $\boldsymbol{x} \geq \boldsymbol{0}$, also $\boldsymbol{x} \in M$. ■

Charakterisierung von Ecken

Sei $\boldsymbol{A} = (\boldsymbol{a}^1, \ldots, \boldsymbol{a}^n)$, d.h. $\boldsymbol{a}^i$ seien die Spalten von $\boldsymbol{A}$. Für $\boldsymbol{x} = (x_1, \ldots, x_n)^T$ kann $\boldsymbol{A}\, \boldsymbol{x} = \boldsymbol{b}$ geschrieben werden als $\sum\limits_{i=1}^{n} \boldsymbol{a}^i\, x_i = \boldsymbol{b}$.

Dann gilt folgende Charakterisierung von Ecken:

Satz 6.3. Sei $\boldsymbol{x}^* \in M \ = \{\boldsymbol{x} \in \mathbb{R}^n\,; \ \boldsymbol{A}\, \boldsymbol{x} = \boldsymbol{b}, \ \boldsymbol{x} \geq \boldsymbol{0}\}$. Dann ist $\boldsymbol{x}^*$ eine Ecke von M genau dann, wenn die Vektoren $\{\boldsymbol{a}^i; \ \boldsymbol{x}_i \neq 0\}$ linear unabhängig sind (im Fall $\boldsymbol{0} \in M$ ist $\boldsymbol{0}$ stets eine Ecke).

Beweis: Sei $J = \{i; \ \boldsymbol{x}_i \neq 0\}$.

„$\Longrightarrow$“: Seien $\{\boldsymbol{a}^i; \ i \in J\}$ linear abhängig, d.h. $\exists\, d_i \in \mathbb{R}$, $i \in J$, nicht alle gleich Null, mit $\sum\limits_{i \in J} \boldsymbol{a}^i\, d_i = \boldsymbol{0}$.

Setze $d_i = 0, i \in \{1, \dots, n\} \setminus J$. Dann gilt

$$\begin{aligned} &\boldsymbol{A}\boldsymbol{x}^* = \boldsymbol{b}, \quad x_i^* = 0,\ i \notin J, \quad x_i^* > 0,\ i \in J, \\ &\boldsymbol{A}\boldsymbol{d} = \boldsymbol{0}, \quad d_i = 0,\ i \notin J. \end{aligned}$$

Für $\tau > 0$ hinreichend klein gilt daher

$$\boldsymbol{A}(\boldsymbol{x}^* + t\boldsymbol{d}) = \boldsymbol{b}, \quad x_i^* + td_i = 0,\ i \notin J, \quad x_i^* + td_i \geq 0,\ i \in J \qquad \forall t \in [-\tau, \tau].$$

Daher gilt $\boldsymbol{x}^1 := \boldsymbol{x}^* + \tau\boldsymbol{d} \in M$, $\boldsymbol{x}^2 := \boldsymbol{x}^* - \tau\boldsymbol{d} \in M$, $\boldsymbol{x}^1 - \boldsymbol{x}^2 = 2\tau\boldsymbol{d} \neq \boldsymbol{0}$ und $\boldsymbol{x}^* = \frac{1}{2}(\boldsymbol{x}^1 + \boldsymbol{x}^2)$. Nach Definition ist daher $\boldsymbol{x}^*$ *keine* Ecke.

„$\Longleftarrow$“: Sei $\boldsymbol{x}^*$ *keine* Ecke, d.h.

$$\exists\, \boldsymbol{x}^1, \boldsymbol{x}^2 \in M,\ \boldsymbol{x}^1 \neq \boldsymbol{x}^2,\ \lambda \in (0,1): \quad \boldsymbol{x}^* = \lambda\boldsymbol{x}^1 + (1-\lambda)\boldsymbol{x}^2.$$

Für alle $i \notin J$ folgt

$$0 = x_i^* = \underbrace{\lambda}_{>0}\underbrace{x_i^1}_{\geq 0} + \underbrace{(1-\lambda)}_{>0}\underbrace{x_i^2}_{\geq 0} \quad \Longrightarrow \quad x_i^1 = x_i^2 = 0.$$

Wegen $\boldsymbol{A}\boldsymbol{x}^1 = \boldsymbol{b}$ und $\boldsymbol{A}\boldsymbol{x}^2 = \boldsymbol{b}$ folgt für $\boldsymbol{d} = \boldsymbol{x}^1 - \boldsymbol{x}^2$:

$$\boldsymbol{A}\boldsymbol{d} = \boldsymbol{A}\boldsymbol{x}^1 - \boldsymbol{A}\boldsymbol{x}^2 = \boldsymbol{b} - \boldsymbol{b} = \boldsymbol{0}$$

und weiter ist $d_i = 0$ für $i \notin J$. Wegen $\boldsymbol{d} = \boldsymbol{x}^1 - \boldsymbol{x}^2 \neq \boldsymbol{0}$ gibt es mindestens ein $i \in J$ mit $d_i \neq 0$. Wir erhalten

$$\sum_{i \in J} \boldsymbol{a}^i d_i = \boldsymbol{A}\boldsymbol{d} = \boldsymbol{0}.$$

Daraus folgt die lineare Abhängigkeit von $\{\boldsymbol{a}^i;\ i \in J\}$. ■

Korollar 6.4. Ist $\boldsymbol{x}^*$ eine Ecke von M, so hat $\boldsymbol{x}^*$ maximal Rang(A) positive Komponenten.

Eine Ecke $\boldsymbol{x}^*$ heißt *entartet*, wenn sie weniger als m positive Komponenten besitzt.

6.3 Basislösungen

Wir nehmen ab jetzt folgendes an:

Voraussetzung 6.5. Es gilt Rang$(\boldsymbol{A}) = m$.

Dies ist keine Einschränkung, denn im Fall, dass $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ Lösungen besitzt, können so lange (geeignete) Zeilen enfernt werden, bis A vollen Zeilenrang hat, ohne dass sich die Lösungsmenge ändert.

Definition 6.6. Gegeben sei das Problem (L) und es gelte Voraussetzung 6.5. Wir definieren:

$$\left.\begin{array}{l} \boldsymbol{x} \in M \text{ heißt } \textit{Basislösung} \\ \text{(Basisvektor) zur Indexmenge } J \end{array}\right\} \overset{\text{(Def.)}}{\Longleftrightarrow} \left\{\begin{array}{l} \exists\, m \text{ linear unabhängige Spalten} \\ \qquad \boldsymbol{a}^i \text{ von } \boldsymbol{A}, \\ i \in J,\ |J| = m, \quad \text{so dass} \\ x_i = 0 \quad \text{für} \quad i \notin J, \\ x_i \geq 0 \quad \text{für} \quad i \in J \quad \text{und} \\ \sum\limits_{i \in J} \boldsymbol{a}^i x_i = \boldsymbol{b} \end{array}\right.$$

Beachte: Es wird nicht gefordert $x_i > 0$ für alle $i \in J$. Jede Ecke $\boldsymbol{x} \in M$ definiert also eine Basislösung zu einer (im entarteten Fall nicht eindeutigen) Basis des $\mathbb{R}^m$ aus Spaltenvektoren von $\boldsymbol{A}$. Denn die linear unabhängigen Vektoren $\{\boldsymbol{a}^i \ : \ x_i \neq 0\}$ können durch weitere Spalten $\boldsymbol{a}^j$ zu einer Basis des $\mathbb{R}^m$ ergänzt werden. Umgekehrt beschreibt jede Basislösung eindeutig eine Ecke.

Beispiel. einer entarteten Ecke bzw. einer entarteten Basislösung: Wird im $\mathbb{R}^3$ M durch eine 4– seitige Pyramide, die auf der $(x_1, x_2)-$Ebene steht, dargestellt, so ist ihre Spitze eine entartete Ecke. (Formulierung als (L) im $\mathbb{R}^7 : x_1, x_2, x_3,$ 4 Schlupfvariable y_i, $\text{Rg}(\boldsymbol{A}) = 4$; $y_i = 0,\ i = 1...4$, beschreibt die Pyramidenspitze, das ist eine Restriktion zu viel.)

Als nächstes müssen wir uns fragen: Hat das Problem (L) überhaupt Ecken (im $\mathbb{R}^2$ wird z.B. durch $x_2 \geq 0,\ x_2 \leq 2$ ein Bereich ohne Ecken beschrieben), und wird, falls ja, ein optimaler Zielfunktionswert (falls einer existiert, vgl. Beispiel 1) auch in einer Ecke angenommen? Antwort gibt

Satz 6.7. Für (L) gilt:

a) $M \neq \emptyset \ \Rightarrow \ \exists$ eine Basislösung

(d.h. wenn zulässige Punkte existieren, so gibt es auch eine Basislösung).

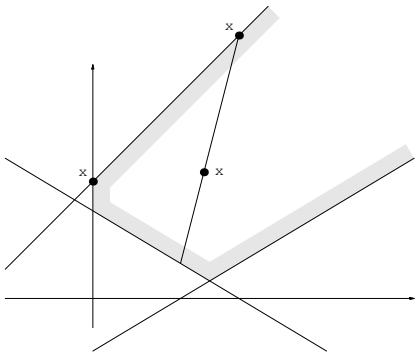
b) $\exists\, \boldsymbol{x}^* \in M$ optimal $\quad \Rightarrow \quad \exists$ eine optimale Basislösung.

Beweis:

zu a):

Geometrische Idee für a).

Ist $\boldsymbol{x}^* \in M$ keine Ecke, so existiert eine Gerade durch $\boldsymbol{x}^*$, auf der man in M in beiden Richtungen ein Stück laufen kann (vgl. Def. von Ecke), bis man zu einem Punkt $\overline{\boldsymbol{x}}$ einer Kante (bzw. einer Seite) kommt. $\overline{\boldsymbol{x}}$ erfüllt eine Gleichheitsrestriktion mehr als $\boldsymbol{x}^*$ (die Schlupfvariable der zugehörigen Ungleichheitsrestriktion ist $= 0$). $\overline{\boldsymbol{x}}$ hat also mehr Nullkomponenten als $\boldsymbol{x}^*$. Ist $\overline{\boldsymbol{x}}$ keine Ecke, so kann man obigen Prozess wiederholen.



Mathematische Umsetzung:

Sei $\boldsymbol{x}^* \in M$ *keine* Basislösung. Dann ist $\boldsymbol{x}^*$ auch keine Ecke und mit $J = \{i \ ; \ \boldsymbol{x}_i^* \neq 0\}$ gilt: $\{a_i \ ; \ i \in J\}$ ist linear abhängig. Es gibt dann $\boldsymbol{d} \in \mathbb{R}^n \setminus \{\boldsymbol{0}\}$ mit $d_i = 0, i \notin J$, und

$$\boldsymbol{A}\boldsymbol{d} = \boldsymbol{0}.$$

Sei $\boldsymbol{x}(t) = \boldsymbol{x}^* + t\boldsymbol{d}$, $t \in \mathbb{R}$. Dann gilt

$$\boldsymbol{A}\boldsymbol{x}(t) = \boldsymbol{b} \quad \forall\, t \in \mathbb{R}.$$

Wegen $x_i(0) = x_i^* > 0$ für alle $i \in J$ und und $x_i(t) = x_i^* = 0$ für alle $i \notin J$ und $t \in \mathbb{R}$ gilt $\boldsymbol{x}(t) \geq 0$ für kleine $|t| \geq 0$.

Hat $\boldsymbol{d}$ negative Komponenten, so gibt es ein maximales $\tau_+ > 0$ mit

$$\boldsymbol{x}(t) \geq 0 \quad \forall\, t \in [0, \tau_+], \quad \exists\, i: \;\; x_i(t) < 0 \quad \forall\, t > \tau_+.$$

Genauer:

$$\tau_+ = \min \left\{ -\frac{x_i^*}{d_i} \;;\; i \in J,\; d_i < 0 \right\}.$$

In diesem Fall bezeichne $j_+ \in J$ einen jener Indizes, für den das Minimum angenommen wird. Andernfalls setze $\tau_+ = \infty$.

Hat $\boldsymbol{d}$ positive Komponenten, so gibt es ein maximales $\tau_- > 0$ mit

$$\boldsymbol{x}(t) \geq 0 \quad \forall\, t \in [-\tau_-, 0], \quad \exists\, i: \;\; x_i(t) < 0 \quad \forall\, t < -\tau_-.$$

Genauer:

$$\tau_- = \min \left\{ \frac{x_i^*}{d_i} \;;\; i \in J,\; d_i > 0 \right\}.$$

Im Fall $\tau_+ < \infty$ ist $\overline{\boldsymbol{x}} := \boldsymbol{x}(\tau_+)$ zulässig und es gilt $\bar{J} := \{i \;;\; \bar{x}_i \neq 0\} \subset J \setminus \{j_+\}$.

Im Fall $\tau_- < \infty$ ist $\overline{\boldsymbol{x}} := \boldsymbol{x}(-\tau_-)$ zulässig und es gilt $\bar{J} := \{i \;;\; \bar{x}_i \neq 0\} \subset J \setminus \{j_-\}$.

In beiden Fällen haben wir $|\bar{J}| \leq |J| - 1$. Ist $\{\boldsymbol{a}^i \;:\; i \in J\}$ linear unabhängig, dann ist $\overline{\boldsymbol{x}}$ eine Ecke und somit eine Basislösung. Sonst wiederholen wir den Prozess.

zu b):

Sei $\boldsymbol{x}^*$ optimale Lösung. Ist $\boldsymbol{x}^*$ bereits eine Ecke (d.h. eine Basislösung), so sind wir fertig. Sonst verfahren wir wie in a).

Es muss dann $\boldsymbol{c}^T \boldsymbol{d} = 0$ sein, denn sonst gäbe es t mit

$$\boldsymbol{x}(t) \in M, \quad \boldsymbol{c}^T \boldsymbol{x}(t) < \boldsymbol{c}^T \boldsymbol{x}^*$$

Nun folgt $\boldsymbol{c}^T \overline{\boldsymbol{x}} = \boldsymbol{c}^T \boldsymbol{x}^*$ und $|\bar{J}| \leq |J| - 1$. Im Fall, dass $\{\boldsymbol{a}^i \;:\; i \in J\}$ linear unabhängig ist, sind wir fertig. Sonst wiederholen wir den Prozess.

6.4 Das Simplex–Verfahren

Wir beziehen uns auf die Aufgabenstellung:

$$\left.\begin{array}{lrcl} \min\limits_{\mathbf{x} \in \mathbb{R}^n} & \boldsymbol{c}^T \boldsymbol{x} & & \\ \text{u.d.N.} & \boldsymbol{A}\, \boldsymbol{x} & = & \boldsymbol{b} \\ & \boldsymbol{x} & \geq & 0 \end{array}\right\} \quad \text{(L)}$$

wobei $\boldsymbol{c} \in \mathbb{R}^n$, $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $\boldsymbol{b} \in \mathbb{R}^m$ gegeben sind. Es gelte

$$\text{Rang } \boldsymbol{A} = m < n.$$

Wir wissen: Wenn eine optimale Lösung existiert, so wird sie auch in einer Ecke des zulässigen Bereichs $M = \{\boldsymbol{x} \in \mathbb{R}^n,\;\; \boldsymbol{A}\, \boldsymbol{x} = \boldsymbol{b},\;\; \boldsymbol{x} \geq \boldsymbol{0}\}$ angenommen.

Idee des Verfahrens

1) Man geht aus von einer Ecke von M (wie man die findet, wird später untersucht) und bestimmt für diese Ecke den Zielfunktionswert.

2) Man untersucht, ob es „benachbarte“ Ecken gibt, die einen kleineren Zielfunktionswert haben und geht gegebenenfalls zu einer solchen Ecke über (Eckentausch). Diesen Eckentausch führt man so lange fort, bis keine benachbarte Ecke mit kleinerem Zielfunktionswert mehr zu finden ist.

1. Unterproblem: Wie vergleicht man formal (man sucht ja einen Rechenalgorithmus) am geeignetsten die Zielfunktionswerte zweier Ecken (bzw. einer Ecke und eines anderen zulässigen Punktes)?

2. Unterproblem: In welcher (formalen) Beziehung steht eine Ecke $\boldsymbol{x}^*$ zu einem anderen zulässigen Punkt $\overline{\boldsymbol{x}}$?

Sei also $\overline{\boldsymbol{x}} \in M$ und $\boldsymbol{x}^* \in M$ eine **Basislösung**, d.h. $\exists\, J \subset \{1, \ldots, n\},\ |J| = m,$ so dass

$$\begin{cases} \boldsymbol{a}^i, \qquad i \in J \quad \text{linear unabhängig}\,, \\ x_i^* = 0, \qquad i \notin J\,, \\ \sum\limits_{i \in J} x_i^*\, \boldsymbol{a}^i = \boldsymbol{b}, \qquad \boldsymbol{x}^* \geq \boldsymbol{0}\,. \end{cases}$$

Vergleich von $\boldsymbol{x}^*$ und $\overline{\boldsymbol{x}}$:
$\boldsymbol{x}^*, \overline{\boldsymbol{x}} \in M$ liefert

$$\sum_{i \in J} x_i^*\, \boldsymbol{a}^i = \boldsymbol{b} = \sum_{j=1}^{n} \overline{x}_j\, \boldsymbol{a}^j\,. \tag{124}$$

Idee:
Die $\boldsymbol{a}^i,\ i \in J$ bilden eine Basis für alle Spalten $\boldsymbol{a}^j$. Setzt man diese Basisdarstellung in obige Gleichung ein, so erhält man eine Beziehung zwischen x_i^* und $\overline{x}_i$.

Basisdarstellung: $\forall$ Spalten $\boldsymbol{a}^j$ von $\boldsymbol{A}$ $\quad \exists\, d_{ij} \in \mathbb{R},\ i \in J,\ j = 1, \ldots, n$ mit

$$\boldsymbol{a}^j = \sum_{i \in J} d_{ij}\, \boldsymbol{a}^i, \quad j = 1, \ldots, n, \quad \text{wobei für alle} \quad j \in J: \ d_{ij} = \begin{Bmatrix} 1 \text{ für } i = j \\ 0 \text{ für } i \neq j \end{Bmatrix}\,. \tag{125}$$

Einsetzen von (125) in (124) ergibt

$$\sum_{i \in J} x_i^*\, \boldsymbol{a}^i \quad = \quad \sum_{j=1}^{n} \overline{x}_j \left(\sum_{i \in J} d_{ij}\, \boldsymbol{a}^i \right) \quad = \quad \sum_{i \in J} \left(\sum_{j=1}^{n} d_{ij}\, \overline{x}_j \right) \boldsymbol{a}^i\,.$$

Koeffizientenvergleich der $\boldsymbol{a}^i,\ i \in J,$ ergibt für $i \in J$:

$$\begin{aligned} x_i^* &= \sum_{j=1}^{n} d_{ij}\,\overline{x}_j \\ &= \sum_{j\in J} d_{ij}\,\overline{x}_j + \sum_{j\notin J} d_{ij}\,\overline{x}_j \\ &= \overline{x}_i + \sum_{j\notin J} d_{ij}\,\overline{x}_j \qquad \text{gemäß (125)}\,, \end{aligned}$$

also folgt

$$\boxed{\overline{x}_i \;=\; x_i^* \;-\; \sum_{j\notin J} d_{ij}\,\overline{x}_j\,, \qquad i \;\in\; J} \tag{126}$$

Vergleich von $\boldsymbol{c}^T\boldsymbol{x}^*$ und $\boldsymbol{c}^T\overline{\boldsymbol{x}}$:
Zum Vergleich wird (126) in die Zielfunktion eingesetzt:

$$\begin{aligned} \boldsymbol{c}^T\overline{\boldsymbol{x}} &= \sum_{i\in J} c_i\,\overline{x}_i \;+\; \sum_{i\notin J} c_i\,\overline{x}_i \;=\; \sum_{i\in J} c_i\left(x_i^* \;-\; \sum_{j\notin J} d_{ij}\,\overline{x}_j\right) \;+\; \sum_{j\notin J} c_j\,\overline{x}_j \\ &= \sum_{i\in J} c_i\,x_i^* \;-\; \sum_{j\notin J}\sum_{i\in J} c_i\,d_{ij}\,\overline{x}_j \;+\; \sum_{j\notin J} c_j\,\overline{x}_j\,, \end{aligned}$$

also

$$\boxed{\boldsymbol{c}^T\overline{\boldsymbol{x}} \;=\; \boldsymbol{c}^T\boldsymbol{x}^* \;+\; \sum_{j\notin J}\left(c_j \;-\; \sum_{i\in J} c_i\,d_{ij}\right)\overline{x}_j} \tag{127}$$

Hieraus liest man ab (beachte: die d_{ij} sind unabhängig von $\overline{\boldsymbol{x}}$):

Satz 6.8. *(Optimalitätskriterium)* $c_j \;-\; \sum_{i\in J} c_i\,d_{ij} \;\geq\; 0 \;\forall\, j \notin J \quad\Rightarrow\quad \boldsymbol{x}^*$ optimal

bzw. ist $\boldsymbol{x}^*$ nicht optimal, dann gilt

$$\exists\, r \;\notin\; J \;:\quad c_r \;-\; \sum_{i\in J} c_i\,d_{ir} \;<\; 0\,. \tag{128}$$

Unter dieser Voraussetzung arbeiten wir weiter.

Herleitung des Austauschschrittes
Ist $\boldsymbol{x}^*$ nicht optimal, so sucht man eine neue (benachbarte) Ecke $\overline{\boldsymbol{x}}$ mit

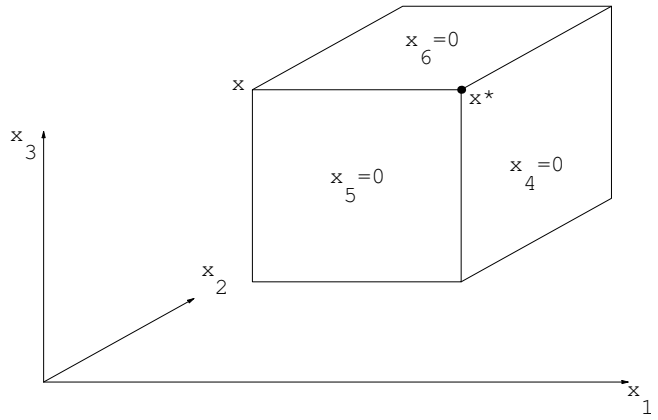
$$\boldsymbol{c}^T\overline{\boldsymbol{x}} \quad<\quad \boldsymbol{c}^T\boldsymbol{x}^*\,,$$

gegen die $\boldsymbol{x}^*$ ausgetauscht werden soll. Wir verdeutlichen das Vorgehen zunächst an einem Beispiel.

Beispiel: Eckentausch
Sei M etwa ein Würfel im $\mathbb{R}^3$, dessen 6 Seiten durch die Ungleichungen

$$\sum_{j=1}^{3} a_{ij}\,x_j \;\leq\; b_i, \qquad i = 1,\ldots,6 \quad (*)$$

beschrieben werden.



Nach Einführung von Schlupfvariablen $x_4, \ldots, x_9$ entsprechen $(*)$ die Gleichheitsrestriktionen

$$\sum_{j=1}^{3} a_{ij}\, x_j \;+\; x_{i+3} \;=\; b_i, \qquad i = 1, \ldots, 6$$

und die Vorzeichenrestriktionen

$$x_i \;\geq\; 0, \qquad i = 4, \ldots, 9\,.$$

Die Ecke $\boldsymbol{x}^*$ wird bestimmt durch die Restriktionen

$$x_j \;=\; 0, \qquad j = 4, 5, 6\,.$$

Will man zur Ecke $\overline{\boldsymbol{x}}$, so wird die Restriktion $x_4 = 0$ aufgegeben.

Eine Ecke $\boldsymbol{x}^*$ hat die Eigenschaft, dass sie mindestens $n - m$ Nullkomponenten hat (vgl. Korollar 6.4), d.h., dass sie mindestens $n - m$ Vorzeichenrestriktionen als Gleichungsrestriktionen erfüllt, d.h. dass sie auf mindestens $n - m$ Hyperebenen $x_\nu = 0$ liegt $(\nu \notin J)$ (vgl. Beispiel: Eckentausch). Man geht zu einer Nachbarecke $\overline{\boldsymbol{x}}$, indem man eine dieser Restriktionen aufgibt: d.h. für ein $r \notin J$ lässt man $\overline{x}_r = \delta > 0$ positiv werden, behält aber die anderen Restriktionen $\overline{x}_\nu = x^*_\nu = 0,\ \nu \notin J,\ \nu \neq r$ bei (im Beispiel sind das die Ebenen $x_5 = 0,\ x_6 = 0$). Wenn man sich auf der Kante von $\boldsymbol{x}^*$ nach $\overline{\boldsymbol{x}}$ bewegt, so ändern sich die Komponenten $x_i,\ i \in J$. Man macht für $\overline{\boldsymbol{x}}$ also zunächst den Ansatz

$$\begin{cases} \overline{x}_r = \delta > 0, \text{ für ein } r \notin J\,, \\ \overline{\boldsymbol{x}}_j = x^*_j - \alpha_j\,,\ j \in J,\ \alpha_j = \text{ die Änderungen}\,, \\ \overline{\boldsymbol{x}}_j = 0,\ \forall j \notin J,\ j \neq r \\ \qquad\qquad (\text{ die anderen Nullkomponenten will man beibehalten})\,. \end{cases} \tag{129}$$

r, δ und die Änderungen α_j müssen wir noch untersuchen.

Das $r \notin J$ wird man so wählen wollen, dass längs der ausgesuchten Kante der Zielfunktionswert möglichst schnell fällt. Setzt man $\overline{\boldsymbol{x}}$ in (127) ein, so folgt

$$\boldsymbol{c}^T \overline{\boldsymbol{x}} \quad = \quad \boldsymbol{c}^T \boldsymbol{x}^* \;+\; \left(c_r \;-\; \sum_{i \in J} c_i\, d_{ir} \right) \quad \underbrace{\overline{x}_r}_{>\,0}\,. \tag{130}$$

Man wird also ein r aus (128) so wählen, dass gilt

$$t_r \quad := \quad c_r \, - \, \sum_{i \in J} c_i \, d_{ir} \quad = \quad \min_{j \notin J} \left(c_j \, - \, \sum_{i \in J} c_i \, d_{ij} \right) \; < \; 0 \, , \tag{131}$$

d.h. man wählt die Kante aus längs der der Zielfunktionswert am schnellsten abnimmt, denn t_r ist die Ableitung der Zielfunktion nach $\overline{x}_r$.
Bewegt man sich auf der Kante, so bedeutet das, dass die Änderungen α_i so beschaffen sein müssen, dass $\overline{\boldsymbol{x}}$ zulässig bleibt, d.h. $\boldsymbol{A}\,\overline{\boldsymbol{x}} = \boldsymbol{b}$, $\overline{\boldsymbol{x}} \geq 0$. Es muss also gelten

$$\begin{aligned} \boldsymbol{A}\,\overline{\boldsymbol{x}} \; &= \; \sum_{j \in J} \left(x_j^* - \alpha_j \right) \boldsymbol{a}^j \, + \, \delta \, \boldsymbol{a}^r \quad \stackrel{!}{=} \quad \boldsymbol{b} \quad \text{(Zulässigkeitsbedingung)} \\ &= \quad \underbrace{\sum_{j \in J} x_j^* \, \boldsymbol{a}^j}_{\substack{= \; \boldsymbol{b} \\ \text{da } \boldsymbol{x}^* \in M}} \quad - \; \sum_{j \in J} \alpha_j \, \boldsymbol{a}^j \, + \, \delta \, \boldsymbol{a}^r \quad \stackrel{!}{=} \quad \boldsymbol{b} \, , \end{aligned}$$

und somit

$$\delta \, \boldsymbol{a}^r \, - \, \sum_{j \in J} \alpha_j \, \boldsymbol{a}^j \quad = \quad \boldsymbol{0} \, . \tag{132}$$

Benutzt man für $\boldsymbol{a}^r$ die Basisdarstellung (125), so folgt aus (132)

$$\boldsymbol{0} \quad = \quad \delta \sum_{j \in J} d_{jr} \, \boldsymbol{a}^j \, - \, \sum_{j \in J} \alpha_j \, \boldsymbol{a}^j \quad = \quad \sum_{j \in J} \left(\delta \, d_{jr} - \alpha_j \right) \boldsymbol{a}^j$$

und daraus wegen der Basiseigenschaft der $\boldsymbol{a}^j$: $\delta \, d_{jr} - \alpha_j = 0$ bzw.

$$\alpha_j \quad = \quad \delta \, d_{jr} \, .$$

Unser verbesserter Ansatz für $\overline{\boldsymbol{x}}$ muss also lauten:

$$\overline{\boldsymbol{x}} \; = \; \overline{\boldsymbol{x}}(\delta) \quad \text{mit} \quad \begin{cases} \overline{x}_r \; = \; \delta \; > \; 0 \, , & r \; \notin \; J, \quad r \text{ gemäß (131)} \\ \overline{x}_j \; = \; x_j^* - \delta \, d_{jr} \, , & j \; \in \; J \\ \overline{\boldsymbol{x}}_j \; = \; 0, \; \forall j \notin J, \; j \neq r \end{cases} \tag{133}$$

$\delta \; > \; 0$ nur so groß, dass $x_j^* - \delta \, d_{jr}$ $\begin{cases} \geq \; 0 \; \forall \, j \in J, \\ = \; 0 \text{ für ein } \nu \in J \text{ (falls möglich),} \end{cases}$
(man möchte ja eine neue Ecke haben,
braucht also eine neue Nullkomponente).

Was passiert, falls $d_{jr} \leq 0 \, \forall \, j \in J$? Dann ist $\overline{\boldsymbol{x}}(\delta) \geq 0 \, \forall \, \delta \in \mathbb{R}^+$ und wegen $\boldsymbol{A}\,\overline{\boldsymbol{x}}(\delta) = \boldsymbol{b}$ (so wurde der Ansatz konstruiert), gilt dann $\overline{\boldsymbol{x}}(\delta) \in M \; \forall \, \delta \in \mathbb{R}^+$, d.h. M ist nicht beschränkt, und wegen (128) und nach (130) ist auch die Zielfunktion nicht nach unten beschränkt, d.h. es gilt der

Satz 6.9. *(Abbruchkriterium)*Gilt für ein

$$\begin{aligned} r \notin J : \; c_r - \sum_{j \in J} c_j \, d_{jr} \; &< \; 0 \quad \text{und} \\ d_{jr} \; &\leq \; 0 \; \forall \, j \in J \, , \end{aligned}$$

so ist die Zielfunktion nicht nach unten beschränkt (d.h. es gibt keine optimale Lösung).

Wir setzen im folgenden also voraus

$$\exists\, j \in J \;:\; d_{jr} \;>\; 0, \qquad r \text{ gemäß (131)}\,. \tag{134}$$

Dann muss für den Ansatz (133) die Zulässigkeitsforderung $\overline{\boldsymbol{x}}(\delta) \geq 0$ gesichert werden, d.h.

$$\begin{aligned} & x_j^* - d_{jr}\,\delta \geq 0 \qquad \forall\, j \in J \text{ mit } d_{jr} > 0,\ r \text{ gemäß (131)} \\ \Longleftrightarrow \qquad & \delta \leq \tfrac{x_j^*}{d_{jr}} \quad \forall\, j \in J \text{ mit } d_{jr} > 0,\ r \text{ gemäß (131)}\,. \end{aligned}$$

Man wählt also in (133) (um eine neue Nullkomponente zu erhalten)

$$\delta \quad = \min_{j \in J, d_{jr} > 0} \frac{x_j^*}{d_{jr}} \quad =: \quad \frac{x_\nu^*}{d_{\nu r}} \quad (\nu \text{ nicht notwendig eindeutig})\,, \tag{135}$$

d.h. wegen $x_\nu^* - d_{\nu r}\,\delta = 0$ wird $\boldsymbol{a}^\nu$ aus der Basis ausgeschieden und durch $\boldsymbol{a}^r$ ersetzt.

Beachte: $\delta = 0$ ist leider möglich, wenn $x_\nu^* = 0$ (vgl. Bemerkung zu Satz Satz 6.10). Anschaulich müsste $\overline{\boldsymbol{x}}(\delta)$ gemäß (133), (135) wieder eine Ecke sein.

Satz 6.10. *(Austauschschritt)* Ist $\boldsymbol{x}^*$ Basislösung zur Indexmenge J, und existiert ein $r \notin J$ mit

$$c_r \;-\; \sum_{j \in J} d_{jr}\, c_j \;\;<\;\; 0$$

und ein $\mu \in J$ mit $d_{\mu r} > 0$, so ist für

$$\delta \quad = \quad \min_{\substack{\mu \in J \\ d_{\mu r} > 0}} \frac{x_\mu^*}{d_{\mu r}} \quad =: \quad \frac{x_\nu^*}{d_{\nu r}}$$

$$\overline{\boldsymbol{x}} \quad = \quad \begin{cases} \overline{x}_r \;=\; \delta \\ x_j^* - d_{jr}\,\delta, \;\; j \in J \\ \overline{\boldsymbol{x}}_j \;=\; 0, \;\; \forall j \notin J, \;\; j \neq r \end{cases}$$

eine Basislösung mit $\boldsymbol{c}^T\,\overline{\boldsymbol{x}} \leq \boldsymbol{c}^T\,\boldsymbol{x}^*$ zur Indexmenge $\overline{J} \;=\; (J \setminus \{\nu\}\,) \cup \{r\}$.

Beweis: $\overline{\boldsymbol{x}} \in M$ gilt laut Konstruktion. Für die Zielfunktion gilt nur (vgl. (130))

$$\boldsymbol{c}^T\,\overline{\boldsymbol{x}} \quad \leq \quad \boldsymbol{c}^T\,\boldsymbol{x}^* \;+\; \left(c_r \;-\; \sum_{i \in J} c_i\, d_{ir} \right) \quad \underbrace{\overline{x}_r}_{\geq\, 0}\,.$$

denn $x_r = \delta = 0$ ist möglich.

Zu zeigen bleibt: $a^j,\;\; j \in J,\;\; j \neq \nu$ und a^r sind linear unabhängig.

Annahme: Sie seien linear abhängig. Dann folgt: $\exists\, \lambda_i \in \mathbb{R},\;\; i \in J,\;\; i \neq \nu,\;\; \lambda_r,$ so dass

$$\sum_{\substack{i \in J \\ i \neq \nu}} \lambda_i\, \boldsymbol{a}^i \;+\; \lambda_r\, \boldsymbol{a}^r \;\;=\;\; \mathbf{0},\;\; \text{nicht alle } \lambda_i = 0,\;\; \text{insbesondere } \lambda_r \neq 0\,, \tag{136}$$

denn aus $\lambda_r = 0$ folgte $\lambda_i = 0 \ \forall\, i \in J,\ i \neq \nu$ (Basiseigenschaft der a^i). Wir setzen in (136) für $\boldsymbol{a}^r$ die Basisdarstellung $\boldsymbol{a}^r = \sum\limits_{i \in J} d_{ir}\, \boldsymbol{a}^i$ ein $\Rightarrow$

$$\mathbf{0} \quad = \quad \sum_{\substack{i \in J \\ i \neq \nu}} \lambda_i\, \boldsymbol{a}^i \ + \ \lambda_r \sum_{i \in J} d_{ir}\, \boldsymbol{a}^i \quad = \quad \sum_{\substack{i \in J \\ i \neq \nu}} (\lambda_i + \lambda_r\, d_{ir})\ \boldsymbol{a}^i \ + \ \lambda_r\, d_{\nu r}\, \boldsymbol{a}^\nu\,.$$

Da die $\boldsymbol{a}^i,\ i \in J$ eine Basis bilden, müssen alle Koeffizienten der $\boldsymbol{a}^i$ verschwinden, insbesondere $\lambda_r\, d_{\nu r} = 0$. Dies ist ein Widerspruch wegen $\lambda_r \neq 0,\ d_{\nu r} > 0$. ■

Bemerkung 6.11. (zu Satz 6.10)

1) Ist die Ecke $\boldsymbol{x}^*$ nicht entartet (d.h. $x_j^* > 0\ \forall\, j \in J$), so liefert der Austauschschritt (Satz 6.10) eine Ecke $\overline{\boldsymbol{x}}$ mit $\boldsymbol{c}^T \overline{\boldsymbol{x}} < \boldsymbol{c}^T \boldsymbol{x}^*$.

2) $\delta = 0$ kann vorkommen, wenn

$$x_j^* \ > \ 0 \quad \forall\, j \in J$$

nicht erfüllt ist. Dann ist die *Ecke* $\boldsymbol{x}^*$ *entartet*. Dann ist zwar $\overline{\boldsymbol{x}}$ wieder eine Basislösung mit der Indexmenge $\overline{J} \neq \boldsymbol{J}$. Aber $\boldsymbol{x}^*$ und $\overline{\boldsymbol{x}}$ beschreiben dieselbe Ecke. Es können Zyklen entstehen, bei denen in jedem Schritt eine neue Basislösung gefunden wird, die aber immer dieselbe Ecke beschreibt. Man kann solche Zyklen vermeiden, (vgl. Collatz/Wetterling), das Verfahren ist dann jedoch sehr aufwendig.

3) Ist $\boldsymbol{x}^*$ nicht entartete Ecke und lokales Minimum, so wird in $\boldsymbol{x}^*$ auch das globale Minimum angenommen (Übung!).

4) $\overline{\boldsymbol{x}}$ kann entartet sein (ohne dass $\boldsymbol{x}^*$ entartet ist), wenn δ im Satz 6.10 für mehr als einen Index ν angenommen wird.

Insgesamt erhalten wir das folgende Verfahren:

Algorithmus 6.12. (Simplex-Verfahren)

0. Bestimme eine Basislösung x mit zugehöriger Indexmenge J.
   Ist dies nicht möglich, STOP: Das Probelm besitzt keinen zulässigen Punkt.

Wiederhole:

1. Bestimme/aktualisiere d_{ij}, $i \in J$, $1 \leq j \leq n$, mit

$$\boldsymbol{a}^j = \sum_{i \in J} d_{ij} \boldsymbol{a}^i.$$

2. Besimme die Pivotspalte gemäß $r \notin J$,

$$t_r := c_r - \sum_{j \in J} d_{jr} c_j = \min_{i \notin J} \left(c_i - \sum_{j \in J} d_{ji} c_j \right).$$

   Falls $t_r \geq 0$, STOP: x ist optimale Ecke.

3. Falls $d_{jr} \leq 0$ für alle $j \in J$, STOP: Problem nicht nach unten beschränkt.
4. Bestimme die Pivotzeile gemäß $\nu \in J$, $d_{\nu r} > 0$,
$$\delta := \frac{x_\nu}{d_{\nu r}} = \min_{j \in J,\, d_{jr} > 0} \frac{x_j}{d_{jr}}.$$
5. Setze
$$x_j := \begin{cases} \delta & j = r \\ x_j - \delta d_{jr} & j \in J, \\ 0 & j \notin J,\ j \neq r. \end{cases}$$
und $J := (J \setminus \{\nu\}) \cup \{r\}$.

6.5 Bestimmung einer Ausgangsbasislösung

Manchen Optimierungsaufgaben kann man eine Ausgangsecke ansehen. Sind die Nebenbedingungen ursprünglich in der Form $\boldsymbol{A}\,\boldsymbol{x} \leq \boldsymbol{b},\ \boldsymbol{x} \geq 0$ mit einem Vektor $\boldsymbol{b} \geq 0$ (komponentenweise) gegeben und $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, so kann man durch Einführung von m Schlupfvariablen $y_i \geq 0,\ i = 1, \ldots, m$ die Nebenbedingungen auf Normalform bringen:

$$\begin{pmatrix} & & & \\ & & 1 & \\ \boldsymbol{A} & & & 1 \\ & & & & \ddots \\ & & & & & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \\ y_1 \\ \vdots \\ y_p \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ \vdots \\ b_m \end{pmatrix},$$

$$\underbrace{\qquad}_{n} \quad \underbrace{\qquad\qquad}_{m}$$

$$\boldsymbol{x} \ \geq\ 0, \qquad \boldsymbol{y} \ =\ (y_1, \ldots, y_m)^T \ \geq\ \boldsymbol{0}\,.$$

Dann ist $\text{Rang}\ (\boldsymbol{A}\,\boldsymbol{I}_m) = m$, $\boldsymbol{I}_m = m \times m$–Einheitsmatrix, und $\begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{pmatrix} = \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{b} \end{pmatrix}$ ist wegen $\boldsymbol{b} \geq \boldsymbol{0}$ eine Ausgangsbasislösung bzw. Ecke des Problems (vgl. Satz 6.3, Definition 6.6).

Kann man für die Aufgabe in der Normalform

$$\begin{cases} \boldsymbol{c}^T \boldsymbol{x} & \stackrel{!}{=} & \min, & \boldsymbol{c} \in \mathbb{R}^n \\ \boldsymbol{A}\,\boldsymbol{x} & = & \boldsymbol{b}, & \boldsymbol{A} \in \mathbb{R}^{m \times n}, \quad \boldsymbol{b} \in \mathbb{R}^m, \quad \text{Rg}\,\boldsymbol{A} = m \\ \boldsymbol{x} & \geq & \boldsymbol{0} \end{cases}$$

keine Basislösung (Ecke) finden, so kann das Lösen eines Hilfsproblems, zu dem eine Ausgangsbasislösung bekannt ist, mit Hilfe des beschriebenen Simplex–Verfahrens Abhilfe schaffen. Es gilt

Satz 6.13. Für das Problem (L) mit $\boldsymbol{b} \geq 0$ (das ist keine Einschränkung) definieren wir mit $\boldsymbol{e} = (1, \ldots, 1)^T \in \mathbb{R}^m$ das Hilfsproblem

$$(*) \quad \begin{cases} \min & \boldsymbol{e}^T \boldsymbol{y} \\ \text{u.d.N.} & \boldsymbol{A}\,\boldsymbol{x} + \boldsymbol{y} \ =\ \boldsymbol{b}, \quad \boldsymbol{y} \ =\ (y_1, \ldots, y_m)^T \\ & \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{pmatrix} \ \geq\ \boldsymbol{0} \end{cases}$$

a) Der Vektor $\begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{pmatrix}$, $\boldsymbol{x} = \boldsymbol{0}$, $\boldsymbol{y} = \boldsymbol{b}$ ist eine Basislösung von $(*)$, und $(*)$ hat eine optimale Basislösung $\begin{pmatrix} \boldsymbol{x}^* \\ \boldsymbol{y}^* \end{pmatrix}$.

b) Ist $\boldsymbol{y}^* \neq \boldsymbol{0}$, so hat (L) keine zulässigen Punkte, also auch keine Lösung.

c) Ist $\boldsymbol{y}^* = \boldsymbol{0}$, so wird durch $\boldsymbol{x}^*$ eine Ecke von (L) gegeben.

Beweis:

a) $\begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{pmatrix} = \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{b} \end{pmatrix}$ ist eine Ecke (wie oben), also besitzt $(*)$ zulässige Punkte und da die Zielfunktion nach unten beschränkt ist $(\boldsymbol{y} \geq \boldsymbol{0})$, existiert auch eine Lösung und damit auch eine Basislösung (Satz 6.7).

b) Hätte (L) einen zulässigen Punkt $\hat{\boldsymbol{x}}$, so wäre $\begin{pmatrix} \hat{\boldsymbol{x}} \\ \boldsymbol{0} \end{pmatrix} \in \mathbb{R}^{n+m}$ ein zulässiger Punkt von $(*)$ mit Zielfunktionswert Null im Widerspruch zur Voraussetzung b).

c) Ist $\begin{pmatrix} \boldsymbol{x}^* \\ \boldsymbol{0} \end{pmatrix}$ Lösung von $(*)$, so sind die zu positiven Komponenten von $\boldsymbol{x}^*$ gehörenden Spaltenvektoren von $\boldsymbol{A}$ linear unabhängig. Also ist $\boldsymbol{x}^*$ eine (möglicherweise entartete) Ecke von (L). ■

Eine weitere Möglichkeit zur Beschaffung einer Ausgangsecke und zur gleichzeitigen Lösung von (L) bietet

Satz 6.14. Für das Problem (L) mit $\boldsymbol{b} \geq \boldsymbol{0}$ definieren wir mit $\boldsymbol{e} = (1, \ldots, 1)^T \in \mathbb{R}^m$ und einem $S > 0$, $S \in \mathbb{R}$, das Hilfsproblem

$$(**) \quad \left\{ \begin{array}{ll} \min & \boldsymbol{c}^T \boldsymbol{x} + S\, \boldsymbol{e}^T \boldsymbol{y} \\ \text{u.d.N.} & \boldsymbol{A}\boldsymbol{x} + \boldsymbol{y} = \boldsymbol{b} \\ & \boldsymbol{z} = \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{pmatrix} \geq \boldsymbol{0} \end{array} \right.$$

a) Der Vektor $\boldsymbol{z} = \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{b} \end{pmatrix}$ ist eine Basislösung von $(**)$ zur Indexmenge $\{n+1,\ n+2,\ \ldots,\ n+m\}$.

b) Ist $S > 0$ hinreichend groß, so gilt: Hat $(**)$ eine Lösung $\begin{pmatrix} \boldsymbol{x}^* \\ \boldsymbol{y}^* \end{pmatrix}$ mit $\boldsymbol{e}^T \boldsymbol{y}^* > 0$, so hat das Ausgangsproblem (L) keine zulässigen Vektoren, ist also unlösbar.

c) Ist $\begin{pmatrix} \boldsymbol{x}^* \\ \boldsymbol{y}^* \end{pmatrix}$ eine Lösung von $(**)$ mit $\boldsymbol{e}^T \boldsymbol{y}^* = 0$, so ist $\boldsymbol{x}^*$ auch Lösung von (L).

Beweis:

a) $\boldsymbol{z}$ ist Basislösung nach Definition 6.6.

b) Dies Aussage kann hier nicht bewiesen werden, da hierzu zusätzliche Kenntnisse notwendig sind.

c) Laut Voraussetzung ist $\boldsymbol{c}^T \boldsymbol{x}^* \leq \boldsymbol{c}^T \tilde{\boldsymbol{x}} + S\, \boldsymbol{e}^T \tilde{\boldsymbol{y}}$ für alle für $(**)$ zulässigen Punkte $\tilde{\boldsymbol{z}} = \begin{pmatrix} \tilde{\boldsymbol{x}} \\ \tilde{\boldsymbol{y}} \end{pmatrix}$. Ist $\overline{\boldsymbol{x}}$ zulässig für (L), so ist $\overline{\boldsymbol{z}} = \begin{pmatrix} \overline{\boldsymbol{x}} \\ \boldsymbol{0} \end{pmatrix}$ zulässig für $(**)$, d.h. $\boldsymbol{c}^T \boldsymbol{x}^* \leq \boldsymbol{c}^T \overline{\boldsymbol{x}}$, d.h. $\boldsymbol{x}^*$ löst (L). ■

Bemerkung 6.15. Das in c) ausgerechnete $\boldsymbol{x}^*$ ist zur erhaltenen Indexmenge nicht notwendig ein Basisvektor für (L), wohl aber eine Ecke (vgl. dazu Collatz–Wetterling [71, p. 36–38]), denn $\begin{pmatrix} \boldsymbol{x}^* \\ \boldsymbol{y}^* \end{pmatrix}$ ist Basislösung von $(**)$ zu einer Indexmenge J^{**}, welche Indizes $> n$ enthalten kann.

6.6 Praktische Durchführung

Ein Iterationsschritt läßt sich jetzt folgendermaßen darstellen:

Sei eine m–elementige Indexmenge J, die zu einem Basisvektor $\boldsymbol{x}$ gehört, bekannt. Den Basisvektor $\boldsymbol{x}$ braucht man an dieser Stelle noch nicht zu kennen. Wir definieren zu dieser Indexmenge J die $(m \times m)$–Matrix

$$\boldsymbol{B} \;=\; (\boldsymbol{a}^i), \qquad i \in J, \qquad \boldsymbol{a}^i \;=\; i\text{–te Spalte von} \quad \boldsymbol{A}\,, \tag{137}$$

die definitionsgemäß regulär ist. Sei $\boldsymbol{D} = (d_{ij})$, $i \in J$, $j = 1, 2, \ldots, n$ die *Tableaumatrix*, die früher (vgl. (132)) durch

$$\boldsymbol{a}^j \;=\; \sum_{i \in J} d_{ij}\, \boldsymbol{a}^i, \quad j \;=\; 1, 2, \ldots, n\,, \quad \text{mit} \quad d_{ij} \;=\; \delta_{ij} \text{ für } j \in J \tag{138}$$

definiert wurde. Mit Hilfe von $\boldsymbol{B}$ aus (137) lautet (138)

$$\boldsymbol{A} \;=\; \boldsymbol{B}\,\boldsymbol{D} \quad \Longleftrightarrow \quad \boldsymbol{A}^T \;=\; \boldsymbol{D}^T\, \boldsymbol{B}^T\,. \tag{139}$$

Wir setzen in Übereinstimmung mit (134)

$$\boldsymbol{s} = (s_1, s_2, \ldots, s_n)^T\,, \quad s_j = \sum_{i \in J} d_{ij}\, c_i\,, \quad j = 1, 2, \ldots, n, \quad \boldsymbol{c}^J = (c_i),\, i \in J\,. \tag{140}$$

Dann ist wegen (140)

$$\boldsymbol{s} \quad = \quad \boldsymbol{D}^T\, \boldsymbol{c}^J\,. \tag{141}$$

Wir benutzen zum Rechnen jetzt nur die in (137) vorhandene Information, nämlich J. Der Vektor $\boldsymbol{s}$ kann aus (139) und (141) ohne Benutzung von $\boldsymbol{D}$ ausgerechnet werden: Multiplizieren wir den rechten Teil von (139) mit einem beliebigen Vektor $\boldsymbol{y} \in \mathbb{R}^m$, so erhalten wir

$\boldsymbol{A}^T\,\boldsymbol{y} = \boldsymbol{D}^T\,\boldsymbol{B}^T\,\boldsymbol{y}$. Bestimmen wir jetzt $\boldsymbol{y}$ so, daß $\boldsymbol{B}^T\,\boldsymbol{y} = \boldsymbol{c}^J$ ist, so ist $\boldsymbol{A}^T\,\boldsymbol{y} = \boldsymbol{D}^T\,\boldsymbol{c}^J = \boldsymbol{s}$, d.h. wir erhalten für $\boldsymbol{s}$ die Berechnungsvorschrift

$$\boldsymbol{B}^T\,\boldsymbol{y} \;=\; \boldsymbol{c}^J \qquad \Longleftrightarrow \qquad \boldsymbol{s} \;=\; \boldsymbol{A}^T\,\boldsymbol{y}\,. \tag{142}$$

Die *Pivotspalte* r erhält man daraus, sofern

$$t_r \;=\; c_r \,-\, s_r \;=\; \min_{j \notin J}\,(c_j - s_j) \;<\; 0\,; \tag{143}$$

man vgl. dazu (135) und Satz 6.7. Die r–te Spalte $\boldsymbol{d}^r$ von $\boldsymbol{D}$, die in (141) bzw. Satz 6.9 benötigt wird, ergibt sich aus dem linken Teil von (139), nämlich

$$\boldsymbol{B}\,\boldsymbol{d}^r \quad = \quad \boldsymbol{a}^r\,, \tag{144}$$

wobei $\boldsymbol{a}^r$, wie üblich, die r–te Spalte von $\boldsymbol{A}$ bezeichnet. Danach kann Satz 6.9 abgeprüft werden. Gleichzeitig kann man den entsprechenden Basisvektor $\boldsymbol{x}$ aus

$$\boldsymbol{B}\,\boldsymbol{x} \quad = \quad \boldsymbol{b} \tag{145}$$

ausrechnen. Sind $\boldsymbol{d}^r = (d_{ir})$ und $\boldsymbol{x}$ bekannt, so kann nach der Formel (141) die *Pivotzeile* ν bestimmt werden und damit auch die neue Indexmenge

$$\overline{J} \quad = \quad (J \,\cup\, \{r\}) \setminus \{\nu\}\,. \tag{146}$$

Die fehlenden Komponenten von $\boldsymbol{x}$ müssen am Schluß durch Null ergänzt werden.

Dieses Verfahren ist für große m aufwendig, für kleine m (etwa ≤ 10) durchaus passabel. Eine Verringerung des Rechenaufwandes kann erreicht werden, wenn nicht dreimal das entsprechende Gleichungssystem pro Schritt neu gelöst wird, sondern einmalig in jedem Schritt eine Zerlegung von $\boldsymbol{B}$ (etwa nach dem Gaußschen Eliminationsverfahren) hergestellt wird. Haben wir eine Zerlegung der Form

$$\boldsymbol{F}\,\boldsymbol{B} \;=\; \boldsymbol{R}, \qquad \boldsymbol{R} \text{ ist rechte Dreiecksmatrix}, \qquad \boldsymbol{F} \text{ regulär} \tag{147}$$

hergestellt, so kann man die auftretenden Gleichungssysteme vom Typ

$$\text{a)} \quad \boldsymbol{B}^T\,\boldsymbol{y} \;=\; \boldsymbol{c}^J\,, \qquad\qquad \text{b)} \quad \boldsymbol{B}\,\boldsymbol{x} \;=\; \boldsymbol{b} \tag{148}$$

folgendermaßen lösen:

$$\text{a)} \quad \boldsymbol{R}^T\,\boldsymbol{z} \;=\; \boldsymbol{c}^J\,, \quad \boldsymbol{y} \;=\; \boldsymbol{F}^T\,\boldsymbol{z} \qquad \text{b)} \quad \boldsymbol{R}\,\boldsymbol{x} \;=\; \boldsymbol{F}\,\boldsymbol{b}\,. \tag{149}$$

Fall a) kann also durch Vorwärtseinsetzen, b) durch Rückwärtseinsetzen gelöst werden. In jedem Fall kommt eine Multiplikation einer Matrix mit einem Vektor hinzu.
Man beachte, daß $\mathbf{F}$ keine Dreiecksmatrix sein muß, wenn das GEV Zeilenvertauschungen enthält, insbesondere also bei der Durchführung mit Spaltenpivotsuche. Ist $\mathbf{I}_{ik}$ die Matrix, die aus der Einheitsmatrix $\mathbf{I}$ entsteht durch Vertauschen der i-ten und k-ten Spalte, so sind in $\mathbf{I}_{ik}\mathbf{A}$ in $\mathbf{A}$ die Zeilen i und k vertauscht. Durch $\mathbf{I}_{ik}$ enthält $\mathbf{F}$ Elemente unterhalb und oberhalb der Diagonalen.
Hat man speziell die Zerlegung $\boldsymbol{B} = \boldsymbol{L}\,\boldsymbol{R}$, $\boldsymbol{L} =$ linke Dreiecksmatrix, so lauten a) und b)

$$\text{a)} \quad \boldsymbol{R}^T\,\boldsymbol{z} \;=\; \boldsymbol{c}^J\,, \quad \boldsymbol{L}^T\,\boldsymbol{y} \;=\; \boldsymbol{z} \qquad \text{b)} \quad \boldsymbol{L}\,\boldsymbol{z} \;=\; \boldsymbol{b}\,, \quad \boldsymbol{R}\,\boldsymbol{x} \;=\; \boldsymbol{z}\,. \tag{150}$$

Eine weitere Verringerung der Anzahl der Operationen kann erreicht werden, wenn man berücksichtigt, daß zwei aufeinanderfolgende Matrizen $\boldsymbol{B},\ \overline{\boldsymbol{B}},$ nämlich

$$\boldsymbol{B} \ = \ (\boldsymbol{a}^i)\,, \quad i \in J\,, \qquad \overline{\boldsymbol{B}} \ = \ (\boldsymbol{a}^j)\,, \quad j \in \overline{J} \ = \ (J \cup \{r\}) \setminus \{\nu\} \tag{151}$$

sich nur in einer Spalte unterscheiden, so daß auch die entsprechenden Zerlegungen $\boldsymbol{F}\boldsymbol{B} = \boldsymbol{R}$, $\overline{\boldsymbol{F}}\,\overline{\boldsymbol{B}} = \overline{\boldsymbol{R}}$ relativ leicht auseinander ausgerechnet werden können. Techniken zur Zerlegung von $\overline{\boldsymbol{B}}$ unter Benutzung der Zerlegung von $\boldsymbol{B}$ werden *Modifikationstechniken* (GLASHOFF–GUSTAFSON [78, p. 165]) oder *Updatingstechniken* genannt.

Updating von Matrixzerlegungen

Sei $\boldsymbol{B} = (\boldsymbol{b}^1, \ldots, \boldsymbol{b}^m)$ eine $m \times m$–Matrix mit den Spalten $\boldsymbol{b}^1, \ldots, \boldsymbol{b}^m$. Bekannt sei die Zerlegung

$$\boldsymbol{F}\,\boldsymbol{B} \ = \ \boldsymbol{R}, \qquad \boldsymbol{R} \text{ rechte Dreiecksmatrix}, \qquad \boldsymbol{F} \text{ regulär}\,. \tag{152}$$

(Eine solche Zerlegung kann man z.B. durch Gauß–Elimination oder mit dem Householder–Verfahren gewinnen). Sei nun $\boldsymbol{b}^*$ ein weiterer vorgegebener Vektor und $\overline{\boldsymbol{B}}$ die Matrix, die aus $\boldsymbol{B}$ entsteht durch Streichen von $\boldsymbol{b}^p$ und Hinzufügen von $\boldsymbol{b}^*$ als letzte Spalte $(p \in \{1, \ldots, m\})$:

$$\overline{\boldsymbol{B}} \ = \ \left(\boldsymbol{b}^1, \ldots, \boldsymbol{b}^{p-1},\ \boldsymbol{b}^{p+1}, \ldots, \boldsymbol{b}^m,\ \boldsymbol{b}^*\right)\,. \tag{153}$$

Gesucht ist eine (152) entsprechende Zerlegung für $\overline{\boldsymbol{B}}$, also

$$\overline{\boldsymbol{F}}\,\overline{\boldsymbol{B}} \ = \ \overline{\boldsymbol{R}}, \qquad \overline{\boldsymbol{R}} \text{ rechte Dreiecksmatrix}, \qquad \overline{\boldsymbol{F}} \text{ regulär}\,, \tag{154}$$

und zwar möglichst billig durch „Aufdatieren“ von (152).

Sehen wir uns zunächst die Matrix $\hat{\boldsymbol{R}} := \boldsymbol{F}\,\overline{\boldsymbol{B}}$ an! Es ist

$$\hat{\boldsymbol{R}} \quad = \quad \left(\boldsymbol{F}\boldsymbol{b}^1, \ldots, \boldsymbol{F}\boldsymbol{b}^{p-1},\ \boldsymbol{F}\boldsymbol{b}^{p+1}, \ldots, \boldsymbol{F}\boldsymbol{b}^m,\ \boldsymbol{F}\boldsymbol{b}^*\right)$$

$$= \begin{pmatrix}
\times & \times & \ldots & \times & \times & \ldots & & & \times \\
 & \times & \ldots & \times & \times & \ldots & & & \times \\
 & & \ddots & \vdots & \vdots & & & & \vdots \\
 & & & \times & \times & & & & \vdots \\
 & & & & \times & & & & \vdots \\
 & & & & \times & \times & & & \vdots \\
 & 0 & & & & \times & \ddots & & \vdots \\
 & & & & & & \ddots & & \vdots \\
 & & & & & & & \times & \times
\end{pmatrix}\,.$$

$\uparrow$

p–te Spalte

Die Matrix $\hat{\boldsymbol{R}}$ enthält als letzte Spalte den Vektor $\boldsymbol{F}\boldsymbol{b}^*$, davor die Spalten von $\boldsymbol{R}$ mit Ausnahme der p–ten Spalte. Die $m - p$ i.a. von 0 verschiedenen Matrixelemente unterhalb der Diagonalen von $\hat{\boldsymbol{R}}$ können nun leicht durch Gauß–Eliminationsschritte mit Spaltenpivotsuche

zu 0 gemacht werden; dabei braucht man nur eine Vertauschung *benachbarter* Zeilen in Betracht zu ziehen. Einem solchen Eliminationsschritt entspricht eine Multiplikation von links mit einer Matrix $\boldsymbol{G}_i$, die entweder von der Form

$$\begin{pmatrix} 1 & & & & & & & 0 \\ & \ddots & & & & & & \\ & & 1 & & & & & \\ & & & 1 & 0 & & & \\ & & & -\ell_{i+1} & 1 & & & \\ & & & & & 1 & & \\ & & & & & & \ddots & \\ 0 & & & & & & & 1 \end{pmatrix} \begin{matrix} \\ \\ \\ \leftarrow & i \\ \leftarrow & i+1 \\ \\ \\ \end{matrix}$$

(falls kein Zeilentausch stattfindet) oder von der Form

$$\begin{pmatrix} 1 & & & & & & & 0 \\ & \ddots & & & & & & \\ & & 1 & & & & & \\ & & & 0 & 1 & & & \\ & & & 1 & -\ell_{i+1} & & & \\ & & & & & 1 & & \\ & & & & & & \ddots & \\ 0 & & & & & & & 1 \end{pmatrix} \begin{matrix} \\ \\ \\ \leftarrow & i \\ \leftarrow & i+1 \\ \\ \\ \end{matrix}$$

(falls ein Zeilentausch stattfindet) ist. Aufgrund der Pivotsuche ist $|\ell_{i+1}| \leq 1$. Nach $m-p$ Eliminationsschritten hat man

$$\boldsymbol{G}_{m-1} \ \ldots \ \boldsymbol{G}_p \, \hat{\boldsymbol{R}} \ = \ \overline{\boldsymbol{R}} \tag{155}$$

mit einer rechten Dreiecksmatrix $\overline{\boldsymbol{R}}$. Setzt man hierin $\hat{\boldsymbol{R}} = \boldsymbol{F}\,\overline{\boldsymbol{B}}$ ein, so erhält man mit

$$\boldsymbol{G}_{m-1} \ \ldots \ \boldsymbol{G}_p \, \boldsymbol{F} \ =: \ \overline{\boldsymbol{F}} \tag{156}$$

eine Zerlegung der gewünschten Form (154).

Für die numerische Rechnung kann man aus (155) und (156) ablesen, daß man $\overline{\boldsymbol{R}}$ und $\overline{\boldsymbol{F}}$ dadurch erhält, daß man die Eliminationsschritte *simultan* auf $\hat{\boldsymbol{R}}$ und $\boldsymbol{F}$ anwendet. Während das direkte Ausrechnen einer Zerlegung (154) durch Gauß–Elimination ca. $c \cdot m^3$ Multiplikationen und Divisionen kostet, verbraucht das beschriebene Aufdatieren von (152) lediglich ca. $(m-p)^2$ Operationen.

Bemerkung:
Nach Abschnitt "Bestimmung einer Ausgangsbasislösung" ist die Ausgangsmatrix B in (152) oft die Einheitsmatrix, wodurch die Zerlegung (152) trivialerweise gegeben ist.

Beispiel: Wir lösen

$$\min \boldsymbol{c}^T \boldsymbol{x} \ \text{ u.d.N. } \ \boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{b}, \ \ \boldsymbol{x} \geq \boldsymbol{0}$$

mit

$$\boldsymbol{A} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 30 & 60 & 0 & 1 & 0 \\ 2 & 10 & 0 & 0 & 1 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 1200 \\ 42000 \\ 5200 \end{pmatrix}, \quad \boldsymbol{c} = \begin{pmatrix} -120 \\ -360 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

nach dem angegebenen Verfahren.

Erster Schritt: $J = \{3,4,5\}$, $\boldsymbol{B} = \boldsymbol{I}$ = Einheitsmatrix, $\boldsymbol{c}^J = (c_3, c_4, c_5)^T = (0,0,0)^T$. Also hat nach (142) das System $\boldsymbol{B}^T \boldsymbol{y} = \boldsymbol{c}^J$ die Lösung $\boldsymbol{y} = \boldsymbol{0}$ und somit $\boldsymbol{s} = \boldsymbol{0}$ und $t_2 = \min\limits_{j=1,2}(c_j - 0) = -360$, d.h. $r = 2$. Aus (144) folgt $\boldsymbol{B}\,\boldsymbol{d} = \boldsymbol{a}^2 = (1,60,10)^T$, d.h. $\boldsymbol{d}^r = \boldsymbol{d}^2 = \boldsymbol{a}^2$ und aus (145) folgt $\boldsymbol{x} = \boldsymbol{b} = (1200, 42000, 5200)^T$. Nach (135) erhält man $\delta = \min\limits_{j \in J} \left\{ \frac{x_j}{d_{j2}} \,:\, d^r_{j2} > 0 \right\} = \frac{x_5}{d_{52}} = 520$, und somit $\nu = 5$.

Zweiter Schritt: $J = \{3,4,5\} \cup \{r\} \backslash \{\nu\} = \{3,4,2\}$, $\boldsymbol{B} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 60 \\ 0 & 0 & 10 \end{pmatrix}$,
$\boldsymbol{c}^J = (0,0,-360)^T$, $\boldsymbol{B}^T \boldsymbol{y} = \boldsymbol{c}^J$ hat die Lösung $\boldsymbol{y} = (0,0,-36)^T$ und daraus folgt $\boldsymbol{s} = \boldsymbol{A}^T \boldsymbol{y} = (-72,-360,0,0,-36)^T$ und $t_1 = \min\limits_{j=1,5}(c_j - s_j) = -48$, $r = 1$. Aus $\boldsymbol{B}\,\boldsymbol{d}^1 = \boldsymbol{a}^1 = (1,30,2)^T$ folgt $\boldsymbol{d}^r = \boldsymbol{d}^1 = (4/5, 18, 1/5)^T$, und aus $\boldsymbol{B}\,\boldsymbol{x} = \boldsymbol{b}$ folgt $\boldsymbol{x} = (680, 10800, 520)^T$ und $\delta = \min\limits_{j \in J} \left\{ \frac{x_j}{d_{j1}} \,:\, d_{j1} > 0 \right\} = \frac{x_4}{d_{41}} = 600$, also $\nu = 4$.

Dritter Schritt: $J = \{3,1,2\}$, $\boldsymbol{B} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 30 & 60 \\ 0 & 2 & 10 \end{pmatrix}$, $\boldsymbol{c}^J = (0,-120,-360)^T$,
$\boldsymbol{B}^T \boldsymbol{y} = \boldsymbol{c}^J$ hat die Lösung $\boldsymbol{y} = (0,-8/3,-20)^T$ und daraus folgt

$$\boldsymbol{s} = \boldsymbol{A}^T \boldsymbol{y} = (-120,-360,0,-8/3,-20)^T$$

und alle $t_j \geq 0$, $j = 4,5$, d.h., wir sind bei der Lösung angekommen, die sich aus $\boldsymbol{B}\,\boldsymbol{x} = \boldsymbol{b}$ zu $\boldsymbol{x} = (x_3, x_1, x_2)^T = (200, 600, 400)^T$ berechnet. Die endgültige Lösung ist damit $\tilde{\boldsymbol{x}} = (600, 400, 200, 0, 0)^T$ und $\boldsymbol{c}^T \tilde{\boldsymbol{x}} = -216000$. Man beachte, daß die Indexmengen J zweckmäßigerweise nicht nach der Größe der Indizes geordnet werden sollten (bei der Updating–Methode ist entsprechend der Aufdatierung von (153) vorzugehen, d. h. ν entfernen, die Elemente aufrücken lassen und an letzter Stelle r hinzufügen).

Literatur

[1] Braess. *Methode der finiten Elemente*. Springer, 1994.

[2] Bulirsch/Stoer. *Numerische Mathematik I*. Springer, 1990.

[3] Bulirsch/Stoer. *Numerische Mathematik II*. Springer, 1990.

[4] G. Fischer. *Lineare Algebra*. Vieweg, 2000. 12. verbesserte Auflage.

[5] Großmann/Roos. *Numerik partieller Differentialgleichungen*. Teubner, 1994.

[6] Sabine Gutsch. Ein Vergleich CG-ähnlicher Verfahren zur Lösung indefiniter Probleme. Master's thesis, Institut für Informatik und Praktische Mathematik, Universität Kiel, 1994.

[7] Wolgang Hackbusch. *Iterative Lösung großer schwachbesetzter Gleichungssysteme*. Teubner, 1993.

[8] H.M. Markowitz. Portfolio selection. *Journal of Finance*, 8:77–91, 1952.

[9] Moler, C. *Numerical Computing with Matlab, siehe auch http://www.mathworks.com/moler/*. SIAM, 2004.

[10] V.A. Barker O. Axelsson. *Finite Element Solution of Boundary Value Problems*. Academic Press, 1984.

[11] Opfer. *Numerische Mathematik für Anfänger*. Vieweg Studium, 1994, 1994.

[12] Plato. *Numerische Mathematik Kompakt*. Vieweg, 2000.

[13] H. Schwetlick und H. Kretzschmar. *Numerische Verfahren für Naturwissenschaftler und Ingenieurer*. Fachbuch Verlag Leipzig, 1991.

[14] R. Schaback und H. Werner. *Numerische Mathematik*. Springer Lehrbuch, 1992.

[15] P. Deuflhard und R. Hohmann. *Numerische Mathematik 1*. De Gruyter, 2002.

[16] Barret; Chan; Demmel; Donato; Dongarra; Eijkhout; Pozo; Romine & van der Vorst. Templates for the solution of linear systems: Building blocks for iterative methods. preprint.