Algorithmische Mathematik
Vorlesung WS 98/99

Winfried Hochstättler
Zentrum für paralleles Rechnen
Universität zu Köln

17. März 1999

Inhaltsverzeichnis

1 Notation, allgemeine Grundlagen und Rechnerproblematik **3**

1.1 Algorithmus 4

1.2 Etwas Notation 4

1.3 Kodierung von Zahlen 6

1.4 Fehlerquellen und Beispiele 9

1.5 Komplexität 11

2 Lineare Gleichungssysteme **14**

2.1 Gaußelimination und LU-Zerlegung, Pivotstrategien 14

2.2 LU-Zerlegung 16

2.3 Gauß-Jordan Algorithmus 20

2.4 Wiederholung über Eigenwerte 21

2.5 Cholesky-Faktorisierung 22

2.6 Matrixnormen 25

2.7 Kondition 28

3 Lineare Optimierung **31**

3.1 Modellbildung 31

3.2 Farkas' Lemma 36

3.3 Dualitätssatz 38

3.4 Das Simplexverfahren 41

3.5 Tableauform des Simplexalgorithmus 46

3.6 Pivotwahl, Entartung, Endlichkeit 46
3.7 Bemerkungen zur Numerik 49
3.8 Die Zweiphasenmethode 50
3.9 Sensitivitätsanalyse 53

4 Nichtlineare Optimierung 56
4.1 Wiederholung aus der mehrdimensionalen Differentialrechnung . 57
4.1.1 Kurven 57
4.1.2 Partielle Ableitungen 58
4.2 Notwendige und hinreichende Bedingungen für Extremwerte . . . 61
4.3 Bedingungen für Extrema auf (Un)gleichungs-definierten Mengen 65
4.3.1 Exkurs Mannigfaltigkeiten und Tangentialräume 65
4.3.2 Lagrange-Multiplikatoren 67
4.3.3 Kuhn-Tucker Bedingungen 69

5 Numerische Verfahren zur Nichtlinearen Programmierung 73
5.1 Das allgemeine Suchverfahren 73
5.2 Spezielle Suchverfahren 76
5.3 Koordinatensuche, Methode des steilsten Abstiegs 81
5.4 Newtonverfahren 85
5.5 Verfahren der konjugierten Richtungen 88

Übungsaufgaben 95

Weihnachtsvorlesung 120

Pathologisches Beispiel zu Satz 4.2.8 127

Index 128

Kapitel 1

Notation, allgemeine Grundlagen und Rechnerproblematik

Was ist ein *Algorithmus*? Zunächst einmal handelt es sich um eine Verunstaltung des Eigennamens *Abu 'Abdallah Muhammad ibn Musa al-Khwarizmi*. Dieser Mathematiker und Astronom, um 790 in Khorezm im heutigen Usbekistan geboren, leitete am Hofe des siebten Kalifen der Abbessiden 'Abdallah Al Ma'mun das Haus der Weisheiten. Von seinen Schriften sind zwei besonders interessant:

a) al-Kitab al-mukhtasar fi hisab al-jabr wa'l-muqabala (Rechnen durch Ergänzen und Ausgleichen in Kürze). Dieses Buch ist der Lösung linearer und quadratischer Gleichungen gewidmet, und ihm verdanken wir das Wort „Algebra“.

b) Kitab hisab al'adad al-hindi, der lateinischen Übersetzung „Algoritmi de numero Indorum“verdanken wir neben dem Wort Algorithmus auch die irreführende Bezeichnung „arabische Zahlen“. Al-Khwarizmi rechnet in 10er Notation mit Platzhalter für die Null.

Nach der Bereitstellung des rechnerischen Handwerkszeugs, beschäftigt sich das erste Algebrabuch mit Maßen und Erbschaftsangelegenheiten. Die lateinische Übersetzung des Buches über die Zahlen der Inder führte dazu, daß bei Rechenproblemen im Italien des ausgehenden Mittelalters die Kaufleute nachschauten, was *Algoritmi dixit*. In der Grundschule lernen wir die Algorithmen zur Addition, Subtraktion, Multiplikation und Division.

Ältere Prominente Algorithmen sind das Sieb des Erathostenes und der euklidische Algorithmus zur Bestimmung des größten gemeinsamen Teilers zweier Zahlen.

1.1 Algorithmus

Unter einem Algorithmus verstehen wir eine "Rechenvorschrift", die bei *Eingabe* einer Probleminstanz eine *Ausgabe* berechnet und

a) in einem Text *endlicher Länge* beschrieben werden kann,

b) deren einzelne Schritte exakt und *eindeutig* festgelegt sind, dabei muß jede mögliche Situation erfaßt sein und die zugehörige Vorgehensweise genau spezifiziert und in endlicher Zeit durchführbar sein.

c) die in endlichen vielen Schritten *terminiert*

Diese informelle Definition eines Algorithmus soll uns hier genügen. In den Informatikvorlesungen haben Sie gelernt, daß man Algorithmen formal über geeignete Maschinenmodelle, wie RAMs oder Turingmaschinen definiert.

1.2 Etwas Notation

Der Allquantor ist $\forall$, der Existenzquantor $\exists$. Wir bezeichnen mir $\mathbb{N}$ $(\mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C})$ die Mengen der natürlichen (ganzen, rationalen, reellen bzw. komplexen) Zahlen. Die Menge $\mathbb{N}$ enthält nicht die Null. Mit $\mathbb{Z}_+$, $(\mathbb{Q}_+, \mathbb{R}_+)$ bezeichnen wir die nichtnegativen ganzen (rationalen, reellen) Zahlen.

Für $n \in \mathbb{N}$ bezeichnet $\mathbb{N}^n$ $(\mathbb{Z}^n, \mathbb{Q}^n, \mathbb{R}^n, \mathbb{C}^n)$ die Menge der Vektoren mit n Komponenten mit Einträgen in $\mathbb{N}$ $(\mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C})$. Sind E und R Mengen, so bezeichnet R^E die Menge aller Abbildungen von E nach R. Wenn E endlich ist, so betrachten wir die Elemente von R^E auch als $|E|$-Vektoren und schreiben

$$x = (x_e)_{e \in E}.$$

Soweit nicht explizit anders gesagt, sind Vektoren stets *Spaltenvektoren*. Ein hochgestelltes $\top$ bedeutet Transposition, also ist für $x, y \in \mathbb{R}^n$ $x^\top$ ein Zeilenvektor und als Matrixprodukt ist

$$x^\top y = \sum_{i=1}^{n} x_i y_i.$$

Ist α eine reelle Zahl, so bezeichnen wir mit

$$\lceil \alpha \rceil := \min\{x \in \mathbb{Z} \mid \alpha \leq x\}, \quad \lfloor \alpha \rfloor := \max\{x \in \mathbb{Z} \mid \alpha \geq x\}.$$

Sind $a, b \in \mathbb{R}^n$, so bedeutet

$$a \leq b \Leftrightarrow \forall i = 1, \ldots, n : a_i \leq b_i.$$

Sind M, N Mengen, so schreiben wir

$M \subseteq N$ falls M eine Teilmenge von N ist,

$M \subset N$ falls $M \neq N$ eine Teilmenge von N ist,

$M \setminus N := \{x \in M \mid x \notin N\}$ für die Differenz

$M \triangle N := (M \setminus N) \cup (N \setminus M)$ für die symmetrische Differenz

$2^M := \{X \mid X \subseteq M\}$ für die Potenzmenge von M.

Sind darüber hinaus $M, N \subseteq \mathbb{R}^n$ und $\alpha \in \mathbb{R}$, so gelte

$M + N := \{x + y \mid x \in M, y \in N\}$.

$M - N := \{x - y \mid x \in M, y \in N\}$.

$\alpha M := \{\alpha x \mid x \in M\}$.

$M^\perp := \{y \in \mathbb{R}^n \mid \forall x \in M : x^\top y = 0\}$.

Ist R eine Menge, so bezeichnen wir mit $R^{m \times n}$ die Menge der (m, n)-Matrizen mit Einträgen in R, das sind Matrizen mit m Zeilen und n Spalten. Ist $A \in R^{m \times n}$ so heißt der Eintrag in der i-ten Zeile und j-ten Spalte $a_{i,j}$ oder $A_{i,j}$ manchmal auch ohne Komma im Index. Wir schreiben auch $A = (a_{i,j})_{\substack{i=1,\ldots,m \\ j=1,\ldots,n}}$ oder kurz $A = (a_{i,j})$. Sind M, N die Zeilen- bzw. Spaltenindexmenge von A, $I \subseteq M$ und $J \subseteq N$, so bezeichnen wir mit $A_{I,J}$ die Matrix $(a_{i,j})_{i \in I, j \in J}$, statt $A_{I,N}$ $(A_{M,J})$ schreiben wir kurz $A_{I.}(A_{.J})$. Mit I_n oder kurz I bezeichnen wir die *Einheitsmatrix*, d.i. die $(n \times n)$-Matrix mit Einsen auf der Diagonale und sonst lauter Nullen.

Mit $e_i \in \mathbb{R}^n$ bezeichnen wir den i-ten Einheitsvektor, d.h. $(e_i)_j = 1$ falls $i = j$ und 0 sonst.

Ist $x \in \mathbb{R}^n$, so bezeichnet, falls nicht ausdrücklich anders definiert, $\|x\|$ stets die *euklidische Norm* oder L_2-Norm

$$\|x\|_2 := \sqrt{x^\top x} = \sqrt{\sum_{i=1}^{n} x_i^2}.$$

Manchmal verwenden wir auch andere Normen, wie etwa

die L_1- oder 1-Norm: $\|x\|_1 := \sum_{i=1}^n |x_i|$

die L_∞- oder Maximumsnorm: $\|x\|_\infty := \max_{1\leq i\leq n} |x_i|$.

Ist $P \subseteq \mathbb{R}^n$ und $\varepsilon > 0$ so bezeichnen wir mit

$$U_\varepsilon(P) := \{x \in \mathbb{R}^n \mid \exists p \in P : \|x-p\| < \varepsilon\}$$

die offene ϵ-Umgebung der Menge P. Ist $P = \{p\}$ einelementig, so schreiben wir statt $U_\varepsilon(\{p\})$ kurz $U_\varepsilon(p)$. Zusätzlich definieren wir $U_\varepsilon(\{\infty\}) = \{x \in \mathbb{R}^n \mid \|x\| > \frac{1}{\varepsilon}\}$.

1.3 Kodierung von Zahlen

Wie wir bereits oben erwähnten, war eine der wesentlichen Leistungen von al-Khwarizmi, daß er die Stellennotation im Abendland bekannt machte. Mit römischen Zahlen läßt sich nämlich schlecht rechnen. Was steckt nun genau hinter dieser Zahldarstellung? Jede Stelle steht für eine Zehnerpotenz, wir haben Basis Zehn. Mit dem folgenden Satz werden wir beweisen, daß man bzgl. jeder beliebigen Basis eine eindeutige Darstellung jeder reellen Zahl gewinnen kann.

Satz 1.3.1 *Sei $B \in \mathbb{N}, B \geq 2$, und sei $x \in \mathbb{R} \setminus \{0\}$. Dann gibt es genau eine Darstellung der Gestalt*

$$x = \sigma B^n \sum_{i=1}^{\infty} x_{-i} B^{-i} \tag{1.1}$$

mit $\sigma \in \{+1, -1\}, n \in \mathbb{Z}$ und $x_{-i} \in \{0, 1, \ldots, B-1\}, x_{-1} \neq 0$ und zusätzlich $\forall j \in \mathbb{N} \exists k \geq j : x_{-k} \neq B-1$.

Beweis. Existenz. Wir setzen

$$\sigma := \operatorname{sign}(x), n := \lfloor \log_B(|x|) \rfloor + 1 \text{ und } x_{-i} := \lfloor |x| B^{i-n} \rfloor \bmod B$$

und müssen nachweisen, daß dann die Reihe gegen x konvergiert. Dazu zeigen wir mittels vollständiger Induktion zunächst

$$x_{-i} = \lfloor (|x| - B^n \sum_{j=1}^{i-1} x_{-j} B^{-j}) B^{i-n} \rfloor. \tag{1.2}$$

Nach Definition von n ist $B^{n-1} \leq |x| < B^n$ und somit zunächst einmal

$$\begin{aligned} x_{-1} &= \lfloor |x| B^{1-n} \rfloor \bmod B \\ &= \lfloor |x| B^{1-n} \rfloor \\ &= \lfloor (|x| - B^n \sum_{j=1}^{1-1} x_{-j} B^{-j}) B^{1-n} \rfloor \\ &\geq 1 \end{aligned}$$

und somit (1.2) gezeigt für $i = 1$. Sei nun $i_0 \geq 2$ und die Behauptung für alle $i \leq i_0 - 1$ bewiesen. Nach Induktionsvoraussetzung ist dann

$$0 \leq (|x| - B^n \sum_{j=1}^{i_0-2} x_{-j} B^{-j}) B^{i_0-1-n} - x_{i_0-1} < 1.$$

Wir schließen hieraus

$$0 \leq (|x| - B^n \sum_{j=1}^{i_0-2} x_{-j} B^{-j}) B^{i_0-n} - B^{n-i_0+1+(i_0-n)} x_{i_0-1} < B$$

und somit

$$\begin{aligned} x_{-i_0} &= \lfloor |x| B^{i_0-n} \rfloor \bmod B \\ &= \lfloor (|x| - B^n \sum_{j=1}^{i_0-1} x_{-j} B^{-j}) B^{i_0-n} \rfloor \end{aligned}$$

womit (1.2) bewiesen ist.

Für den Konvergenzbeweis müssen wir nach Mathematik für Wirtschaftsinformatiker I zeigen, daß es zu jedem $\varepsilon > 0$ ein $i_0 \in \mathbb{N}$ gibt mit

$$\forall N \geq i_0 : |x - \sigma B^n \sum_{i=1}^{N} x_{-i} B^{-i}| < \varepsilon.$$

Sei also $\varepsilon > 0$ vorgegeben und $i_0 = \lceil n + 2 - \log_B \varepsilon \rceil$. Nach dem soeben Bewiesenen ist

$$0 \leq (|x| - B^n \sum_{j=1}^{i_0-1} x_{-j} B^{-j}) B^{i_0-n} < B$$

und somit

$$\begin{aligned}
|x - \sigma B^n \sum_{j=1}^{N} x_{-j} B^{-j}| &= \left| |x| - B^n \sum_{j=1}^{N} x_{-j} B^{-j} \right| \\
&\leq \left| |x| - B^n \sum_{j=1}^{i_0-1} x_{-j} B^{-j} \right| + B^n \sum_{j=i_0}^{\infty} (B-1) B^{-j} \\
&< B^{n+1-i_0} + \frac{B^n (B-1)}{B^{i_0}(1-\frac{1}{B})} \\
&\leq B^{n+2-\lceil n+2-\log_B \varepsilon \rceil} \\
&= B^{\lfloor \log_B \varepsilon \rfloor} \leq \varepsilon.
\end{aligned}$$

Nehmen wir nun an, es gäbe ein $j_0 \in \mathbb{N}$ mit $\forall k \geq j_0 : x_{-k} = B - 1$. Dann ist

$$\begin{aligned}
x_{-(j_0-1)} &= \lfloor (|x| - B^n \sum_{k=1}^{j_0-2} x_{-k} B^{-k}) B^{j_0-1-n} \rfloor \\
&= \lfloor (B^n \sum_{i=1}^{\infty} x_{-i} B^{-i} - B^n \sum_{k=1}^{j_0-2} x_{-k} B^{-k}) B^{j_0-1-n} \rfloor \\
&= \lfloor B^{j_0-1} \sum_{i=j_0-1}^{\infty} x_{-i} B^{-i} \rfloor \\
&= \lfloor x_{-(j_0-1)} + (B-1) \sum_{i=j_0}^{\infty} B^{j_0-1-i} \rfloor \\
&= x_{-(j_0-1)} + \lfloor (B-1) \sum_{i=1}^{\infty} B^{-i} \rfloor \\
&= x_{-(j_0-1)} + \lfloor (B-1) \frac{1}{B} \frac{1}{1-\frac{1}{B}} \rfloor \\
&= x_{-(j_0-1)} + 1.
\end{aligned}$$

Aus diesem Widerspruch folgt die Behauptung. Es bleibt die Eindeutigkeit zu zeigen. Angenommen also $\sigma_1 B^{n_1} \sum_{i=1}^{\infty} x_{-i} B^{-i} = \sigma_2 B^{n_2} \sum_{i=1}^{\infty} y_{-i} B^{-i}$. Offensichtlich muß wegen $x \neq 0$ dann $\sigma_1 = \sigma_2$ sein. Aus $x_{-1} \neq 0 \neq y_{-1}$ und

$$\begin{aligned}
\sum_{i=2}^{\infty} z_{-i} B^{-i} &< \sum_{i=2}^{\infty} (B-1) B^{-i} \\
&= \frac{B-1}{B^2} \frac{1}{1-\frac{1}{B}} \\
&= B^{-1}
\end{aligned}$$

falls alle $z_{-i} \in \{0,\dots,B-1\}$ aber nicht alle identisch $B-1$ sind, schließen wir $n_1 = n_2$. Es bleibt zu zeigen $x_{-i} = y_{-i}$ für alle i. Angenommen dies wäre nicht so. Dann betrachten wir $0 = \sum_{i=1}^{\infty}(x_{-i}-y_{-i})B^{-i}$. Nach Annahme gibt es einen kleinsten Index i_0 mit $x_{-i_0} \neq y_{-i_0}$. Durch eventuelles Vertauschen der Namen können wir o.B.d.A. (ohne Beschränkung der Allgemeinheit) annehmen, daß $x_{-i_0} < y_{-i_0}$. Dann erhalten wir

$$\begin{aligned}
1 \le y_{-i_0} - x_{-i_0} &= \sum_{i=i_0+1}^{\infty}(x_{-i} - y_{-i})B^{-i+i_0} \\
&= \sum_{i=1}^{\infty}(x_{-i-i_0} - y_{-i-i_0})B^{-i} \\
&\le \sum_{i=1}^{\infty}(B-1)B^{-i} \\
&\le (B-1)\frac{1}{B-1} = 1.
\end{aligned}$$

Demnach muß in allen Ungleichungen dieser Kette Gleichheit herrschen. Hieraus folgt aber, daß $x_i = B-1$ und $y_i = 0$ für alle $i > i_0$, was wir ausgeschlossen hatten. □

Im Rechner verwenden wir bekanntermaßen $B \in \{2, 8, 16\}$.

1.4 Fehlerquellen und Beispiele

Man kann keine Reihen mit unendlich vielen Gliedern bei vorgegebenem endlichen Speicher darstellen. Beim praktischen Rechnen ist man deshalb gezwungen zu *runden*. Es seien $x, \tilde{x} \in \mathbb{R}$ und $\tilde{x}$ eine Näherung für x sei. Dann heißen

a) $x - \tilde{x}$ der *absolute Fehler* und

b) für $x \neq 0$, $\frac{x-\tilde{x}}{x}$ der *relative Fehler.*

Bei der Darstellung von Zahlen bzgl. einer geraden Basis $B \in \mathbb{N}, B \ge 2$ wollen wir folgende Rundungsvorschrift für t-stellige Darstellung mit $x \in \mathbb{R} \setminus \{0\}$ für $x = \sigma B^n \sum_{i=1}^{\infty} x_{-i}B^{-i}$ benutzen:

$$Rd_t(x) := \begin{cases} \sigma B^n \sum_{i=1}^{t} x_{-i}B^{-i} & \text{falls } x_{-t-1} < \frac{B}{2} \\ \sigma B^n (B^{-t} + \sum_{i=1}^{t} x_{-i}B^{-i}) & \text{falls } x_{-t-1} \ge \frac{B}{2} \end{cases}$$

Sei nun eine *Maschinengenauigkeit* $t \in \mathbb{N}$ vorgegeben. Dann bezeichnen wir Zahlen mit einer Darstellung der Gestalt $\sigma B^n \sum_{i=1}^{t} x_{-i} B^{-i}$ als *Maschinenzahlen* oder *Gleitkommazahlen*. Dabei heißt n der *Exponent*, σ das *Vorzeichen* und $\sum_{i=1}^{t} x_{-i} B^{-i}$ die *Mantisse* der Zahl. Wir sagen auch die Zahl habe eine *t-stellige Mantisse*.

Satz 1.4.1 *Sei $k \in \mathbb{N}, B = 2k, t \in \mathbb{N}$ und $x = \sigma B^n \sum_{i=1}^{\infty} x_{-i} B^{-i} \neq 0$. Dann gilt:*

a) *$Rd_t(x)$ hat eine Darstellung der Gestalt $\sigma B^{n'} \sum_{i=1}^{t} x'_{-i} B^{-i} \neq 0$,*

b) *der absolute Fehler ist beschränkt durch $|x - Rd_t(x)| \leq \frac{B^{n-t}}{2}$,*

c) *der relative Fehler ist beschränkt durch $\left|\frac{x - Rd_t(x)}{x}\right| \leq \frac{B^{1-t}}{2}$,*

d) *der relative Fehler bzgl. $Rd_t(x)$ ist beschränkt durch $\left|\frac{x - Rd_t(x)}{Rd_t(x)}\right| \leq \frac{B^{1-t}}{2}$.*

Beweis. Übung.

Bei der Verknüpfung zweier Gleitkommazahlen, wie Addition, Subtraktion, Multiplikation und Division werden wir nun zunächst das Ergebnis möglichst genau berechnen und dann auf t Stellen runden. Betrachten wir dies zum Beispiel bei der Addition.

Beispiel 1.4.2 *Wir wollen mit 4-stelliger Arithmetik (d.h. 4-stelliger Mantisse) zur Basis 10 die Zahlen 12,34 und -0,09876 addieren.*

$$
\begin{array}{cccccccccccc}
+ & . & 1 & 2 & 3 & 4 & & & & & & 2 \\
- & . & 9 & 8 & 7 & 6 & & & & & & -1 \\
 & & & & & & & & & & & \\
+ & . & 1 & 2 & 3 & 4 & 0 & 0 & 0 & 0 & & 2 \\
- & . & 0 & 0 & 0 & 9 & 8 & 7 & 6 & 0 & & 2 \\
\hline
+ & . & 1 & 2 & 2 & 4 & 1 & 2 & 4 & 0 & & 2 \\
\hline
+ & . & 1 & 2 & 2 & 4 & & & & & & 2
\end{array}
$$

Das Runden von Zahlen liefert also ein zusätzliches Fehlerpotential bei der numerischen Lösung eines Problems. Andere Fehlerquellen sind *Datenfehler* oder *Verfahrensfehler*. Letzteres sind etwa Ungenauigkeiten, die dadurch enstehen, daß iterative, konvergente Prozesse nach endlich vielen Schritten abgebrochen werden. Alle diese Fehler können durch *Fehlerfortpflanzung* verstärkt werden, wie in der „Mathematik für Chemiker und Wirtschaftsinformatiker“ diskutiert.

In den Übungen wollen wir diskutieren, wie man durch geschickte Auswertung von Formeln die Auswirkungen von Rundungsfehlern verkleinern kann.

1.5 Komplexität

Unter der *Komplexität* eines Algorithmus verstehen wir üblicherweise das Laufzeitverhalten des Verfahrens. Um dies zu definieren müssen wir uns zunächst auf einen Satz von Elementarschritten einigen, deren Ausführung in einem Schritt bewältigt werden kann. Wir haben eben gesehen, daß die Addition zweier Maschinenzahlen konstanten Aufwand verursacht. Der Einfachheit halber woller wir $+, -, \cdot, :$ sowie Vergleichsoperationen $=, <, >, \leq$ etc. und Zuweisungen zu den Elementarschritten zählen. Mit diesen Definitionen können wir bei gegebenem Input die Laufzeit des Algorithmus A bei Input I definieren

$$L_A(I) := |\text{Elementarschritte, die } A \text{ ausführt, bis er terminiert.}|$$

Da wir bei Algorithmen üblicherweise an dem Verhalten bzgl. mehrer Inputbeispiele interessiert sind, betrachten wir das Laufzeitverhalten in Abhängigkeit von der Länge $\langle I \rangle$ der (sinnvoll kodierten) Inputdaten $\langle I \rangle$ und bezeichen als *worst-case Komplexität*

$$L_A(n) := \max\{L_A(I) \mid \langle I \rangle \leq n\}.$$

Beispiel 1.5.1 *Der euklidische Algorithmus zur Bestimmung des größten gemeinsamen Teilers zweier ganzer Zahlen m und n läßt sich wie folgt darstellen.*

```
#!/zpr/local/bin/python         # Pfad fuer den Interpreter
import sys                      # Modul fuer sys
import string                   # Modul fuer string

m=string.atoi(sys.argv[1])      # lese erste Zahl ein
n=string.atoi(sys.argv[2])      # lese zweite Zahl ein
r= m % n                        # r = m (mod) n
while r!=0:
  m=n
  n=r
  r= m % n                      # r = m (mod) n
print n
```

Sei m_i der Inhalt der Variablen m in der i-ten Iteration. Betrachten wir die Änderung der Variablen m beim Durchlauf zweier Schleifen. Offensichtlich ist stets $m_{i+1} < m_i$. Gilt nun $m_{i+1} \leq \frac{m_i}{2}$, so ist offensichtlich $m_{i+2} + m_{i+1} \leq 2m_{i+1} \leq m_i$. Andererseits folgt aber aus $m_{i+1} > \frac{m_i}{2}$ sofort $m_{i+2} = m_i - m_{i-1}$ also wiederum $m_{i+2} + m_{i+1} \leq m_i$. In zwei Schleifendurchläufen wird also die Größe der Zahl m mindestens halbiert. Hieraus schließen wir, daß die Gesamtlaufzeit des Algorithmus beschränkt ist durch $4(\log_2(m) + 1)$. *Also terminiert der Algorithmus.*

Wir schreiben $d|n$ falls d die Zahl n teilt, *bzw.* $d \not| n$ *im anderen Fall. Es bleibt zu zeigen, daß das Verfahren tatsächlich die größte Zahl $ggT(m,n)$ berechnet, die m und n teilt. Dafür zeigen wir, daß, wenn $m > n$, $n \not| m$ aber $d|n$ und $d|m$ auch gilt: $d|(m \bmod n)$. Denn $m = dk_1, n = dk_2$ und $m \bmod n = m - k_3 n$ impliziert $m \bmod n = (k_1 - k_2 k_3)d$. Der Algorithmus terminiert im r-ten Schritt, wenn $n_r|m_r$ und nach dem Gezeigten gilt dann $ggT(n,m)|n_r = m_{r-1} \bmod n_{r-1}$ und somit $ggT(n,m)|n_r$.*

Exaktes Buchhalten bei Algorithmen ist schwierig. Es erweist sich als nützlich für das Verhalten der Kompexitätsfunktion die sogenannten *Landau-Symbole* einzuführen. Seien dazu $f, g : D \to \mathbb{R}$ zwei Funktionen mit $D \subseteq \mathbb{R}$ und $x_0 \in \mathbb{R} \cup \{\infty\}$.

a) f heißt *von der Ordnung* $O(g)$, wir schreiben $f = O(g)$, *für x gegen x_0*, falls es $C > 0$, $\delta > 0$ gibt mit

$$|f(x)| \leq C|g(x)| \text{ für } x \in U_\delta(x_0).$$

b) f heißt *von der Ordnung* $o(g)$, wir schreiben $f = o(g)$, *für x gegen x_0*, falls für alle $C > 0$ es ein $\delta > 0$ gibt mit

$$|f(x)| \leq C|g(x)| \text{ für } x \in U_\delta(x_0).$$

Den wichtigen Spezialfall $n \to \infty$, den wir für die Laufzeitfunktion benötigen, wollen wir noch einmal extra notieren.

Proposition 1.5.2 *Sei A ein Algorithmus mit Laufzeitfunktion L_A und $g : \mathbb{N} \to \mathbb{N}$. Dann ist L_A in $O(g)$*[1] *genau dann, wenn*

$$\exists C > 0\, \exists N_0 \in \mathbb{N}\, \forall N \geq N_0 : L_A(N) \leq Cg(N).$$

Beweis. $L_A = O(g)$ für x gegen ∞, wenn es ein $C > 0$, und ein $\delta > 0$ gibt mit $|L_A(N)| \leq C|g(N)|$ für $N \in U_\delta(\infty)$.

„⇒“ Wir setzen $N_0 = \frac{1}{\delta} + 1$. Dann ist $N \in U_\delta(\infty) \Leftrightarrow N > \frac{1}{\delta} = N_0 - 1 \Leftrightarrow N \geq N_0$.

„⇐“ Wir setzen $\delta = \frac{1}{N_0 - 1}$ und schließen wie eben.

□

[1] Warum ist hier nur $x_0 = \infty$ sinnvoll?

Beispiel 1.5.3 *Wir betrachten die Laufzeitfunktion des euklidischen Algorithmus. Die Inputgröße der Daten ist bei sinnvoller Kodierung* $\log_2(m) + \log_2(n) + 3$*. Wir dürfen annehmen* $n \geq 2$*. Für die Laufzeitfunktion* L_{euklid} *gilt also*

$$L_{euklid}(\log_2(m) + \log_2(n) + 3) \leq 4(\log_2(m) + 1).$$

Nach Definition ist die Laufzeitfuntion monoton wachsend und wir schließen

$$L_{euklid}(\log_2(m)) \leq 4 \log_2(m) \textit{ für } m \geq n \geq 2.$$

Somit ist also $L_{euklid}(k) = O(k)$*, wir sagen, der euklidische Algorithmus ist linear. Üblich ist auch die Bezeichnung* $L_{euklid} = O(n)$*, weil mit n oft die Inputlänge bezeichnet wird, was in diesem Fall eine Fußfalle darstellt.*

Wir wollen auch die *Komplexität eines Problems* definieren und zwar als als die Laufzeitfunktion eines *besten* Algorithmus. Wir können also z.B. etwas salopp sagen, daß das Problem der Bestimmung des größten gemeinsamen Teilers zweier Zahlen $O(n)$ ist.

Üblicherweise definiert ein Problem im mathematische Sinne eine Funktion, die einer gegebenen Instanz des Problems die (eine) zugehörige Lösung zuordnet. Umgekehrt können wir jede gegebene Funktion f als Problem betrachten und nach einem Algorithmus fragen, der diese Funktion *berechnet*. Falls ein solcher existiert, nennen wir die Funktion *berechenbar* sonst *nicht berechenbar*. Der folgende Satz zeigt, daß es bei der Lösung mancher Probleme prinzipielle Schwierigkeiten geben kann.

Satz 1.5.4 *Es gibt eine nicht berechenbare Funktion* $\pi : \mathbb{N} \to \{0, 1\}$.

Beweis. Nach Definition läßt sich jeder Algorithmus in einem endlichen Text beschreiben, insbesondere also auch jeder Algorithmus, der eine Funktion mit Werten in $\{0, 1\}$ berechnet. Indem wir diese Algorithmen nach ihrer Länge und dann lexikographisch ordnen, können wir sie abzählen, also jedem $n \in \mathbb{N}$ einen Algorithmus A_n zuordnen. Wir definieren nun $\pi(n) = 1 - A_n(n)$. Dann stimmt π mit keiner Funktion überein, die von einem Algorithmus berechnet wird. $\square$

Kapitel 2

Lineare Gleichungssysteme

Eine der wichtigsten numerischen Aufgaben ist das Lösen von Gleichungssystemen, denn dieses Problem taucht in vielen Verfahren als Teilproblem auf. Hierzu gibt es im wesentlichen zwei Typen von Algorithmen. Die direkten Verfahren, die in endlich vielen Schritten eine Lösung berechnen, die mit Rundungsfehlern und Ihren Konsequenzen behaftet ist. Andererseits die indirekten, iterativen Verfahren, die abbrechen, wenn eine „gewisse "Genauigkeit erreicht ist. Wir werden uns aus Zeitmangel nur mit direkten Verfahren beschäftigen.

Im gesamten Kapitel werden wir folgende Fragestellung untersuchen. Gegeben seien $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$. Gesucht ist ein Vektor $x \in \mathbb{R}^n$ mit $Ax = b$.

2.1 Gaußelimination und LU-Zerlegung, Pivotstrategien

Diese Aufgabenstellung ist bereits in der „Mathematik für Chemiker und Wirtschaftsinformatiker" behandelt worden. Wir wollen das dort vorgestellte Eliminationsverfahren etwas implementationsnäher darstellen und untersuchen.

Znächst einmal wiederholen wir die Vorgehensweise anhand eines Beispiels.

Beispiel 2.1.1 *Sei*

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \end{pmatrix}, \quad b = \begin{pmatrix} 10 \\ -4 \\ -3 \\ -2 \end{pmatrix}.$$

$$\begin{array}{rrrr|r} 1 & 1 & 1 & 1 & 10 \\ -1 & 0 & 0 & 0 & -4 \\ 0 & -1 & 0 & 0 & -3 \\ 0 & 0 & -1 & 0 & -2 \end{array} \qquad \begin{array}{cccc|c} \boxed{1} & 1 & 1 & 1 & 10 \\ 0 & \boxed{1} & 1 & 1 & 6 \\ 0 & 0 & \boxed{1} & 1 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{array},$$

Und wir erhalten als Lösung $x_4 = 1, x_3 = 3 - 1 = 2, x_2 = 6 - 2 - 1 = 3, x_1 = 10 - 3 - 2 - 1 = 4$.

Vereinfacht formuliert geht man so vor, daß man von links nach rechts mit den Diagonalelementen eine *Gaußelimination* durchführt, d.h. man formt das System so äquivalent um, daß unter dem *Pivotelement* nur noch Nullen stehen. Notieren wir zunächst den *Gaußeliminationsschritt bzgl. des Elementes* a_{ij}.

```
def gausselim(A,i,j):
   for l in range(i+1,m):
      A_lj=A[l,j]
      A[l,:]=A[l,:]-A_lj/A[i,j]*A[i,:]
```

Bei der Bestimmung des Pivotelementes entlang der Diagonalen kann der Fall eintreten, daß $a_{jj} = 0$ ist. Gibt es dann ein $a_{kl} \neq 0$ mit $k \geq j$ und $l \geq j$, so vertauschen wir die k-te und j-te Zeile, sowie l-te und j-te Spalte, merken uns, daß in der Lösung die Indizes l und j vertauscht werden müssen und führen eine Gaußelimination mit a_{jj} durch.

```
def gaussalg(A,b):
  d=min([m,n])
  index=range(0,n)                # Permutation der Variablen
  C=concatenate((A,b),1)          # Verschmelze A,b nebeneinander
  for k in range(d):
    if C[k,k]==0:
      pivotfound,i,j = findpivot(C,index,k)
      if pivotfound == 0:
        break
      else:
        if i != k:                    # swap rows i and k
          swaprows(C,i,k)
        if j != k:                    # swap columns
          swapcolumns(C,index,j,k)    # and variables j and k
    gausselim(C,k,k)
  solvable, x = compute_x(C,index)
  return solvable,x
```

Abschließend können wir nun anhand der transformierten Matrix feststellen, ob das Gleichungssystem lösbar ist und, wenn ja, eine spezielle Lösung bestimmen.

```
def compute_x(C,ind):
# C=[A|b], wobei A eine obere Dreiecksmatrix mit m<= n ist.
# Wir loesen das System Ax=b unter Beruecksichtigung der
# Permutation index fuer x.
       .
       .
  if solvable:
    x[ind[d]]=C[d,n]/C[d,d]
    for i in range(1,d+1):              # for(i=1,i<d+1,i++)
      x[ind[d-i]]=C[d-i,n]
      for j in range(0,i):
        x[ind[d-i]]=x[ind[d-i]]-C[d-i,d-j]*x[ind[d-j]]
      x[ind[d-i]]=x[ind[d-i]]/C[d-i,d-i]
  return solvable,x
```

Bemerkung 2.1.2 *Grundsätzlich sollte man bei numerischen Berechnungen wegen der Rundungsfehler Abfragen nach Gleichheit vermeiden. So sollte für wirkliche Berechnungen in den oberen Routinen z.B. stets anstatt der Abfrage* $x! = 0$ *besser* $|x| \geq \epsilon$ *für eine kleine Konstante abgefragt werden.*

2.2 *LU*-Zerlegung

Wir wollen nun noch etwas detaillierter den Fall untersuchen, daß A ein reguläre Matrix ist. Speziell ist die Matrix dann quadratisch und wir definieren:

Definition 2.2.1 *Sei* A *eine* $(n \times n)$ *Matrix.* A *heißt* obere Dreiecksmatrix, *wenn* $\forall 1 \leq i \leq n \forall i > j \leq n : A_{ij} = 0$ *ist. Analog definieren, die* untere Dreiecksmatrix, *wenn* $\forall 1 \leq i \leq n \forall i < j \leq n : A_{ij} = 0$ *ist.*

Die folgenden Überlegungen sind nützlich, wenn das System $Ax = b$ mit festem A aber variierendem b zu lösen ist. Zunächst halten wir fest, daß wir uns bei regulärem A bei der Suche eines Pivotelementes auf die aktuelle Spalte beschränken können.

Proposition 2.2.2 *Sei* A *eine reguläre* $(n \times n)$*-Matrix und* $1 \leq d \leq n$ *so, daß für alle* $0 < j < d$ *und alle* $i > j$ $A_{ij} = 0$ *ist, d.h.* A *hat in den ersten* $d - 1$ *Spalten die Gestalt einer oberen Dreiecksmatrix. Dann gilt:* $\exists d \leq i \leq n : A_{id} \neq 0$.

Beweis. Wenn $\forall d \leq i \leq n : A_{id} = 0$, so haben die ersten d Spalten Nichtnulleinträge nur in den ersten $d-1$ Zeilen, sind also linear abhängig im Widerspruch zur angenommenen Regularität von A. $\square$

Unter der Annahme, daß nur Zeilenvertauschungen durchgeführt werden, können wir nun alle Transformationen, die im Gaußalgorithmus durchgeführt werden als Multiplikation des Systems $[A|b]$ von links mit einer geeigneten Matrix schreiben.

Definition 2.2.3 *Eine $(n \times n)$ Matrix der Gestalt*

$$P^n_{ij} = \begin{array}{cc} & \begin{array}{cccccccccc} & & i & & & & j & & & \end{array} \\ \begin{array}{c} \\ \\ i \\ \\ \\ \\ j \\ \\ \\ \\ \end{array} & \begin{pmatrix} 1 & & & & & & & & & \\ & 1 & & & & & & & 0 & \\ & & 0 & \dots & \dots & \dots & 1 & & & \\ & & \vdots & \ddots & & & \vdots & & & \\ & & \vdots & & 1 & & \vdots & & & \\ & & \vdots & & & \ddots & \vdots & & & \\ & & 1 & \dots & \dots & \dots & 0 & & & \\ & & & & & & & 1 & & \\ & 0 & & & & & & & \ddots & \\ & & & & & & & & & 1 \end{pmatrix} \end{array}$$

heißt Transpositionsmatrix. *Ein Produkt von Transpositionsmatrizen nennen wir eine* Permutationsmatrix.

Bemerkung 2.2.4 *Die Linksmultiplikation einer $(m \times n)$-Matrix A mit P^m_{ij} bewirkt die Vertauschung der i-ten mit der j-ten Zeile. Die Rechtsmultiplikation von A mit P^n_{ij} bewirkt die Vertauschung der i-ten mit der j-ten Spalte.*

Proposition 2.2.5 *a) Transpositionsmatrizen sind selbstinvers, d.h. $P^n_{ij} P^n_{ij} = I_n$.*

b) Eine $(n \times n)$-Matrix A ist eine Permutationsmatrix genau dann, wenn $A \in \{0,1\}^{n \times n}$ und für jedes $i \in \{1,\dots,n\}$ ein $j_1 \in \{1,\dots,n\}$ und ein $j_2 \in \{1,\dots,n\}$ existieren mit $Ae_i = e_{j_1}$ und $e_i^\top A = e_{j_2}^\top$.

Beweis. Übung. $\square$

Auch ein Gaußeliminationsschritt läßt sich als Linksmultiplikation mit einer geeigneten Matrix schreiben.

Definition 2.2.6 *Eine $(n \times n)$ Matrix der Gestalt*

$$G_d = \begin{array}{cc} & d \\ d & \begin{pmatrix} 1 & & & & \\ & \ddots & & 0 & \\ & & 1 & & \\ & & -g_{d+1,d} & & \\ & 0 & \vdots & \ddots & \\ & & -g_{n,d} & & 1 \end{pmatrix} \end{array}$$

heißt Frobeniusmatrix. *Sie unterscheidet sich nur in der d-ten Spalte von einer Einheitsmatrix. Wir können also schreiben*

$$G_d = I_n - g_d e_d^\top \ \mathit{mit}\ g_d = (0, \dots, 0, g_{d+1,d}, \dots, g_{n,d}).$$

Ein Gaußeliminationsschritt bzgl. des Elementes a_{dd} wird bewirkt durch Linksmultiplikation mit einer Frobeniusmatrix, bei der $g_{id} = \frac{A_{id}}{A_{dd}}$.

Bemerkung 2.2.7 *Die Frobenius Matrix G_d ist regulär und man rechnet nach:*

$$G_d^{-1} = I_n + g_d e_d^\top = \begin{array}{cc} & d \\ d & \begin{pmatrix} 1 & & & & \\ & \ddots & & 0 & \\ & & 1 & & \\ & & g_{d+1,d} & & \\ & 0 & \vdots & \ddots & \\ & & g_{n,d} & & 1 \end{pmatrix} \end{array}.$$

Proposition 2.2.8 *a) Ist P^n_{ij} eine Transpositionsmatrix mit $i, j > d$, so gilt*

$$P^n_{ij} G_d^{-1} P^n_{ij} = P^n_{ij}(I_n + g_d e_d^\top) P^n_{ij} = I_n + (P^n_{ij} g_d) e_d^\top .$$

b) Sind $P^n_{i_1 j_1}, \dots, P^n_{i_k j_k}$ Transpositionsmatrizen mit $i_l, j_l > d$, so gilt

$$I_n + ((\prod_{l=1}^{k} P^n_{i_l j_l}) g_d) e_d^\top = \prod_{l=1}^{k} P^n_{i_l j_l} G_d^{-1} \prod_{l=1}^{k} P^n_{i_{k+1-l} j_{k+1-l}} .$$

Beweis. Übung. □

Führen wir den Gaußalgorithmus bei einer regulären Matrix A durch, so können wir dies auch durch eine Multiplikation mit einer Folge von Frobenius- und Transpositionsmatrizen beschreiben. Nach einer Gaußelimination können wir den Speicherplatz unterhalb des Diagonalelementes g_{dd} zur Abspeicherung des Vektors g_d nutzen. Der folgende Satz verdeutlicht, warum es sinnvoll ist, alle weiteren Zeilenvertauschungen auch mit dem g_d-Vektor durchzuführen. Genauer gilt:

Satz 2.2.9 (Satz von der Dreieckszerlegung) *Sei $A \in \mathbb{R}^{n\times n}$ eine reguläre Matrix und seien $P^n_{i_1 1}, \ldots P^n_{i_{n-1} n-1}$ und $G_1, \ldots, G_{n-1}$ die benötigten Transpositions- bzw. Frobeniusmatrizen. Setzen wir*

$$P = \prod_{k=1}^{n-1} P^n_{i_k k} \quad \textit{und} \quad L = \prod_{k=1}^{n-1} \left(I_n + \left(\prod_{j=1}^{n-k-1} P^n_{i_{n-j} n-j} \right) g_k e_k^\top \right)$$

und bezeichnen mit U die Transformierte von A, dann ist L eine untere Dreiecksmatrix und es gilt

$$PA = LU.$$

Beweis. Zunächst einmal ist $I_n + \left(\prod_{j=1}^{n-k-1} P^n_{i_{n-j} n-j} \right) g_k e_k^\top$ offensichtlich wieder eine Frobeniusmatrix. Somit ist L das Produkt von Frobeniusmatrizen mit von rechts nach links wachsendem Index, also (!) eine untere Dreiecksmatrix. Die Behauptung folgt nun, da $U = \prod_{k=1}^{n-1} G_{n-k} P^n_{i_{n-k} k} A$ und

$$\begin{aligned}
LU &= \prod_{k=1}^{n-1} \left(I_n + \left(\prod_{j=1}^{n-k-1} P^n_{i_{n-j} n-j} \right) g_k e_k^\top \right) \left(\prod_{k=1}^{n-1} G_{n-k} P^n_{i_{n-k} k} \right) A \\
&= \prod_{k=1}^{n-2} \left(I_n + \left(\prod_{j=1}^{n-k-1} P^n_{i_{n-j} n-j} \right) g_k e_k^\top \right) G_{n-1}^{-1} G_{n-1} \\
&\qquad\qquad P^n_{i_{n-1} n-1} \left(\prod_{k=2}^{n-1} G_{n-k} P^n_{i_{n-k} k} \right) A \\
&= \prod_{k=1}^{n-3} (\sim)(I_n + P^n_{i_{n-1} n-1} g_{n-2} e_{n-2}^\top) \\
&\qquad\qquad P^n_{i_{n-1} n-1} \left(\prod_{k=2}^{n-1} G_{n-k} P^n_{i_{n-k} k} \right) A \\
&= \prod_{k=1}^{n-3} (\sim) P^n_{i_{n-1} n-1} G_{n-2}^{-1} P^n_{i_{n-1} n-1} P^n_{i_{n-1} n-1} G_{n-2}
\end{aligned}$$

$$
\begin{aligned}
& \qquad P^n_{i_{n-2}n-2}\left(\prod_{k=3}^{n-1} G_{n-k}P^n_{i_{n-k}k}\right)A \\
= & \qquad \vdots \\
= & \prod_{k=1}^{n-d-1}(\sim)\left(\prod_{j=1}^{d-1} P^n_{i_{n-j}n-j}\right)G^{-1}_{n-d}\left(\prod_{j=1}^{d-1} P^n_{i_{n-d+j}n-d+j}\right) \\
& \left(\prod_{k=1}^{d-1} P^n_{i_{n-k}n-k}\right)G_{n-d}P^n_{i_{n-d}n-d}\left(\prod_{k=d+1}^{n-1} G_{n-k}P^n_{i_{n-k}k}\right)A \\
= & \; PA.
\end{aligned}
$$

□

Bemerkung 2.2.10 *Wir hatten bereits angedeutet, daß die LU-Zerlegung nützlich ist, wenn man $Ax = b$ für verschieden Vektoren b lösen muß. Dafür berechnet man zunächst $Lc = Pb$. Da L eine untere Dreiecksmatrix ist, läßt sich das durch eine einfache Rekursion lösen. Nun berechnet man $Ux = c$. Insgesamt gilt dann $Ax = P^{-1}PAx = P^{-1}LUx = P^{-1}Lc = P^{-1}Pb = b$.*

Analysieren wir die Komplexität des Algorithmus, so sind für den d-ten Gaußeliminationsschritt $(n-d+(n-d)^2)$ Multiplikationen und $(n-d)^2$ Additionen durchzuführen. Abgesehen von der Pivotsuche erält man einen Aufwand von $\frac{n^3}{3}-\frac{n}{3}$ Multiplikationen und $\frac{n^3}{3}-\frac{n^2}{2}+\frac{n}{6}$ Additionen. Bei der Lösung von $Ux = Pb$ beträgt der Aufwand $\binom{n}{2} = \frac{n^2}{2}-\frac{n}{2}$ Additionen und $\binom{n+1}{2} = \frac{n^2}{2}+\frac{n}{2}$ Multiplikationen.

Bemerkung 2.2.11 *Bei der Pivotsuche sollte man aus Gründen der numerischen Stabilität nach möglichst großen Einträgen suchen. Führt man allerdings eine Spaltenpivotsuche durch, so empfiehlt es sich, vorher das Maximum der Zeilen zu skalieren.*

2.3 Gauß-Jordan Algorithmus

Anstatt nur unterhalb des Pivotelementes die Variable zu eliminieren, kann man dies natürlich auch oberhalb der Diagonale tun. Wir erhalten so die Gauß-Jordan-Elimination.

```
def gaussjordanelim(A,i,j):
   A_ij=A[i,j]
   A[i,:]=A[i,:]/A_ij
```

```
for k in range(i)+range(i+1,m):
   A_kj=A[k,j]
   A[k,:]=A[k,:]-A[i,:]*A_kj
```

Analog zu dem Vorherigen kann man diese Eliminationsschritte zu einem Algorithmus kombinieren, bei dem in A eine Einheitsmatrix übrigbleibt und rechter Hand eine Lösung des Gleichungssystems steht.

Vorteile hat man beim Einsatz von Vektorrechnern, ansonsten ist die Komplexität um einen Faktor drei schlechter als eben. Den Gauß-Jordan-Eliminationsschritt werden wir allerdings in der Linearen Programmierung noch intensiv benutzen.

Auch den Gauß-Jordan-Eliminationsschritt kann man als Linksmultiplikation mit einer Matrix beschreiben. Dies hat bei eine Pivot auf dem Element a_{ij} die Gestalt

$$G_{i,j} = \begin{array}{cc} & \begin{array}{ccccc} & & i & & \end{array} \\ i & \begin{pmatrix} 1 & & \frac{-a_{1j}}{a_{ij}} & & \\ & \ddots & & 0 & \\ & & \frac{1}{a_{ij}} & & \\ & & \frac{-a_{i+1,j}}{a_{ij}} & & \\ & 0 & \vdots & \ddots & \\ & & \frac{-a_{nj}}{a_{ij}} & & 1 \end{pmatrix} \end{array}$$

und wird als η-Matrix bezeichnet. Die i-te Spalte wird auch η-Vektor genannt.

2.4 Wiederholung über Eigenwerte

Definition 2.4.1 *Sei A eine $(n \times n)$-Matrix über einem Körper $\mathbb{K}$. Eine Zahl $\lambda \in \mathbb{K}$ heißt* Eigenwert *von A, wenn es ein $x \in \mathbb{K}^n \setminus \{0\}$ gibt mit $Ax = \lambda x$. Jeder solche Vektor $x \neq 0$ heißt* Eigenvektor von A zum Eigenwert λ. *Ist A eine Matrix mit komplexwertigen Einträgen, so bezeichnen wir mit A^* die zu A transponierte, komplexkonjugierte Matrix. Ist $A = A^*$, so nennen wir A* hermitesch. *Eine Matrix Q heißt* orthogonal, *wenn $Q^\top Q = QQ^\top = I$.*

Eine reellwertige Matrix ist hermitesch genau dann, wenn sie symmetrisch ist.

Lemma 2.4.2 *Hermitesche Matrizen haben nur reelle Eigenwerte.*

Beweis. $Ax = \lambda x \Rightarrow x^*A^* = x^*A = \bar{\lambda}x^* \Rightarrow \lambda x^*x = x^*A^*x = \bar{\lambda}x^*x \Rightarrow \lambda = \bar{\lambda}.$ □

Satz 2.4.3 *Ist A eine reellwertige, symmetrische Matrix, dann gibt es eine orthogonale Matrix Q, so daß $QAQ^\top = D$ eine Diagonalmatrix ist. Die Spalten von Q bilden eine* Orthonormalbasis aus Eigenvektoren.

Beweis. vgl. Mathematik für Wirtschaftsinformatiker, Kapitel V. □

Wir nennen $A = Q^\top DQ$ die *Hauptachsentransformation* von A, da die Eigenwerte von A die Diagonaleinträge von D sind und die Eigenvektoren die Zeilen von Q. Ist D eine Diagonalmatrix und y der Vektor der Diagonaleinträge, so schreiben wir auch $D = Diag(y)$.

2.5 Cholesky-Faktorisierung

Die Resultate des letzten Abschnitts besagen, daß man eine symmetrische Matrizen A durch Drehung und Spiegelung des Koordinatensystems in eine Diagonalmatrix überführen kann, welche die Eigenwerte von A auf der Diagonale hat. Sind diese alle nichtnegativ, so kann man aus A die „Wurzel ziehen“. Solche Matrizen werden beim sogenannten Newton-Verfahren in der nichtlinearen Programmierung eine Rolle spielen.

Definition 2.5.1 *Sei $A \in \mathbb{R}^{n\times n}$ eine symmetrische Matrix. Dann heißt A* positiv definit *falls $x^\top Ax > 0$ für alle $x \in \mathbb{R}^n \setminus \{0\}$.*

Im folgenden werden wir zeigen, wie man bei positiv definiten Matrizen das Gleichungssystem $Ax = b$ noch effizienter lösen kann.

Proposition 2.5.2 *Eine reellwertige, symmetrische Matrix ist positiv definit genau dann, wenn der kleinste Eigenwert $\lambda_n > 0$ ist.*

Beweis. Die Bedingung offensichtlich notwendig, da aus $Ax = \lambda x$ sofort $x^\top Ax = \lambda\|x\|^2$ folgt. Umgekehrt, sei $b_1, \ldots, b_n$ eine Orthonormalbasis aus Eigenvektoren zu den Eigenwerten $\lambda_1 \geq \ldots \geq \lambda_n > 0$. Dann ist

$$
\begin{aligned}
x^\top Ax &= \left(\sum_{i=1}^n (x^\top b_i)b_i^\top\right) A \left(\sum_{j=1}^n (x^\top b_j)b_j\right) \\
&= \left(\sum_{i=1}^n (x^\top b_i)b_i^\top\right) \left(\sum_{j=1}^n (x^\top b_j)Ab_j\right)
\end{aligned}
$$

$$\begin{aligned}
&\overset{b_j \text{ sind EV}}{=} \quad \sum_{i,j=1}^{n} (x^\top b_i)(x^\top b_j)\lambda_j b_i^\top b_j \\
&\overset{b_i \text{ sind ONB}}{=} \quad \sum_{j=1}^{n} \lambda_i (x^\top b_j)^2 \\
&\overset{x \neq 0, \lambda_i > 0}{>} \quad 0.
\end{aligned}$$

□

Proposition 2.5.3 *Sei $A \in \mathbb{R}^{n\times n}$ symmetrisch und positiv definit und $I \subseteq \{1, \ldots n\}$. Dann ist auch $A_{I,I}$ symmetrisch und positiv definit.*

Beweis. Offensichtlich ist $A_{I,I}$ symmetrisch. Angenommen sie wäre nicht positiv definit, dann gäbe es ein $\tilde{x} \in \mathbb{R}^I \setminus \{0\}$ mit $\tilde{x}^\top A_{I,I}\tilde{x} \leq 0$. Wir definieren $x \in \mathbb{R}^n$ durch

$$x_i := \begin{cases} \tilde{x}_i & \text{falls } i \in I \\ 0 & \text{sonst.} \end{cases}$$

Dann ist $x \neq 0$ und $x^\top Ax = \tilde{x}^\top A_{I,I}\tilde{x} \leq 0$ im Widerspruch zur positiven Definitheit von A. □

Nach diesen Vorbereitungen können wir nun folgenden Satz zeigen.

Satz 2.5.4 (Satz von der Cholesky-Faktorisierung) *Sei $A \in \mathbb{R}^{n\times n}$ symmetrisch und positiv definit. Dann existiert eine eindeutig bestimmte, reguläre untere Dreiecksmatrix L mit $A = LL^\top$ und $L_{ii} > 0$ für $i = 1, \ldots, n$.*

Beweis. Wir führen vollständige Induktion über n. Im Fall $n = 1$ ist $A = (a_{11})$ mit $a_{11} > 0$ und wir setzen $L = (\sqrt{a_{11}})$. Sei also $n > 1$. Wir zerlegen A als

$$A = \begin{pmatrix} A_{n-1,n-1} & b \\ b^\top & a_{n,n} \end{pmatrix}.$$

Wie oben gezeigt ist $A_{n-1,n-1}$ positiv definit und symmetrisch. Nach Induktionsvoraussetzung gibt es also genau eine reguläre untere Dreiecksmatrix L_{n-1} mit positiven Diagonalelementen und $L_{n-1}L_{n-1}^\top = A_{n-1,n-1}$. Jedes L wie in dem Satz behauptet hat also notwendig die Form

$$L = \begin{pmatrix} L_{n-1,n-1} & 0 \\ c^\top & l_{n,n} \end{pmatrix}.$$

Setzen wir ein, so erhalten wir

$$A = \begin{pmatrix} A_{n-1,n-1} & b \\ b^\top & a_{n,n} \end{pmatrix} = \begin{pmatrix} L_{n-1} & 0 \\ c^\top & l_{n,n} \end{pmatrix} \begin{pmatrix} L_{n-1}^\top & c \\ 0 & l_{n,n} \end{pmatrix}$$

und somit als notwendige und hinreichende Bedingungen $L_{n-1}c = b$ und $\|c\|^2 + l_{n,n}^2 = a_{n,n}$. L_{n-1} ist regulär also können wir $c = L_{n-1}^{-1}b$ setzen. Die Behauptung folgt nun, wenn wir zeigen können, daß $a_{n,n} - \|c\|^2 > 0$ ist. Dafür betrachten wir $x^\top = (c^\top L_{n-1}^{-1}, -1)$. Dann ist $x \neq 0$ und somit

$$\begin{aligned}
0 &< x^\top A x \\
&= x^\top \begin{pmatrix} A_{n-1,n-1} & b \\ b^\top & a_{n,n} \end{pmatrix} \begin{pmatrix} (L_{n-1}^{-1})^\top c \\ -1 \end{pmatrix} \\
&= x^\top \begin{pmatrix} L_{n-1}L_{n-1}^\top (L_{n-1}^{-1})^\top c - b \\ c^\top L_{n-1}^\top (L_{n-1}^{-1})^\top c - a_{n,n} \end{pmatrix} \\
&= (c^\top L_{n-1}^{-1}, -1) \begin{pmatrix} 0 \\ c^\top c - a_{n,n} \end{pmatrix} \\
&= a_{n,n} - \|c\|^2.
\end{aligned}$$

□

Für die Berechnung der Cholesky-Faktorisierung betrachten wir das Zustandekommen der einzelnen Einträge von A.

$$a_{ij} = L_{i.}L_{.j}^\top = \sum_{k=1}^{n} l_{ik}l_{jk} = \sum_{k=1}^{\max\{i,j\}} l_{ik}l_{jk}.$$

Wir können daraus leicht rekursiv die Einträge von L ausrechnen und erhalten folgenden Algorithmus:

```
def cholesky(A):
  n=A.shape[0]
  L=zeros((n,n))
  try:
    for k in range(n):
      L[k,k]=sqrt(A[k,k]-dot(L[k,:],L[k,:]))
      for l in range(k+1,n):
        L[l,k]=(A[l,k]-dot(L[l,:],L[k,:]))/L[k,k]
  except:
    print "Matrix nicht positiv definit"
  return L
```

Analysieren wir den Rechenaufwand des Verfahrens, so haben wir bei festem Zeilenindex l einen Aufwand von $\sum_{i=1}^{l-1} i = \binom{l}{2}$ Additionen, $\binom{l}{2}$ Multiplikationen, $l-1$ Divisionen und einer Quadratwurzel. Aufsummiert erhalten wir $\frac{1}{2}\sum_{l=1}^{n} l^2 - l = \frac{n(n+1)(2n+1)-3n(n+1)}{12} = \frac{n^3-n}{6}$ Additionen und Multiplikationen, $\binom{n}{2}$ Divisionen und n Quadratwurzeln. Die Cholesky-Faktorisierung läßt sich analog zur LU-Zerlegung zur Lösung von Gleichungssystemen $Ax = b$ benutzen. Der Aufwand beträgt aber nur etwa die Hälfte des Gaußverfahrens. Außerdem ist dieser Algorithmus numerisch stabiler.

Beispiel 2.5.5 *Wir betrachten die Matrix*

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

und berechnen

$l_{1,1} = \sqrt{1},\ l_{2,1} = -1,\ l_{3,1} = -1,\ l_{4,1} = -1,$
$l_{2,2} = \sqrt{2-(-1)^2} = 1,\ l_{3,2} = (0-(-1)*(-1))/1 = -1,$
$l_{4,2} = (0-(-1)*(-1))/1 = -1,$
$l_{3,3} = \sqrt{3-1-1} = 1,\ l_{4,3} = (-1-(-1)*(-1)-(-1)*(-1))/1,$
$l_{4,4} = \sqrt{4-1-1-1} = 1$

also

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

.

2.6 Matrixnormen

Im diesem Abschnitt wollen wir Überlegungen zur numerischen Stabilität von Operationen der Linearen Algebra beginnen. In der Mathematik für Chemiker und Wirtschaftsinformatiker ist bereits Fehlerfortpflanzung von rellwertigen Funktionen besprochen worden. Wir erinnern daran, daß dort die Linearisierung (Ableitung) der Funktion eine Maßzahl für die numerische Problematik des Ausdrucks gab. Wie sieht das im Mehrdimensionalen aus? Dafür werden wir Normen von Matrizen studieren.

Definition 2.6.1 *Sei X ein Vektorraum über $\mathbb{R}$. Eine Abbildung $\|\cdot\| : X \to \mathbb{R}$ heißt* Norm, *wenn*

(N1) $\|x\| = 0 \Leftrightarrow x = 0$,

(N2) $\forall \alpha \in \mathbb{R} : \|\alpha x\| = |\alpha|\|x\|$; (Homogenität)

(N3) $\|x+y\| \leq \|x\| + \|y\|$; Dreiecksungleichung).

Ein Vektorraum mit Norm heißt normierter Vektorraum.

Wegen $0 = \|x-x\| \leq \|x\| + \|-x\| = 2\|x\|$ gilt stets $\|x\| \geq 0$. Zu einem normierten n-dimensionalen $\mathbb{R}$-Vektorraum X und einem normierten m-dimensionalen $\mathbb{R}$-Vektorraum Y gibt es eine natürliche Norm auf dem Vektorraum der $(m \times n)$-Matrizen über $\mathbb{R}$.

Proposition 2.6.2 *Seien X, Y normierte $\mathbb{R}$-Vektorräume der Dimension n bzw. m mit Normen $\|\cdot\|_X$, $\|\cdot\|_Y$. Dann ist die Abbildung $\|\cdot\|_{X\to Y} : \mathbb{R}^{m\times n} \to \mathbb{R}$ definiert durch*

$$\|A\|_{X\to Y} := \sup_{x\in\mathbb{R}^n\setminus\{0\}} \frac{\|Ax\|_Y}{\|x\|_X} \stackrel{!}{=} \max_{\|x\|_X=1} \|Ax\|_Y$$

eine Norm auf dem Vektorraum der linearen Abbildungen von X nach Y, die natürliche Norm.

Beweis. ad N1: $\max_{\|x\|_X=1} \|Ax\|_Y = 0$ impliziert mit der Homogenität, daß $Ax = 0$ für alle $x \in \mathbb{R}^n$. Also stellt A die Nullabbildung dar und somit haben wir $A = 0$.

ad N2: $\max_{\|x\|_X=1} \|\alpha Ax\|_Y = \max_{\|x\|_X=1} |\alpha|\|Ax\|_Y = |\alpha| \max_{\|x\|_X=1} \|Ax\|_Y$.

ad N3:

$$\begin{aligned}
\max_{\|x\|_X=1} \|(A+B)x\|_Y &\leq \max_{\|x\|_X=1} (\|Ax\|_Y + \|Bx\|_Y) \\
&\leq \max_{\|x\|_X=1} \|Ax\| + \max_{\|y\|_X=1} \|By\|_Y.
\end{aligned}$$

□

Wegen $\|A\|_{X\to Y} \geq \|Ax\|_Y / \|x\|_X$ gilt nun stets $\|Ax\|_Y \leq \|A\|_{X\to Y}\|x\|_X$. Für natürliche Matrixnormen von Endomorphismen (linearen Abbildungen innerhalb eines festen Vektorraumes X) gilt $\|I\| = 1$ und $\|AB\| \leq \|A\|\|B\|$. Letzteres folgt aus

$$\|ABx\|_X \leq \|A\|_{X\to X}\|Bx\|_X \leq \|A\|_{X\to X}\|B\|_{X\to X}\|x\|_X.$$

Beispiel 2.6.3 *Wir betrachten die im letzten Kapitel bereits eingeführten Normen $\|\cdot\|_1, \|\cdot\|_2, \|\cdot\|_\infty$. Sind dann $X = \mathbb{R}^n = Y$, so schreiben wir für die zugehörigen Matrixnormen kurz ebenso $\|A\|_1, \|A\|_2, \|A\|_\infty$. Es gilt:*

a) $\|A\|_1 = \max_{j\in\{1,\ldots n\}} \sum_{i=1}^n |a_{ij}|$ *(Spaltensummennorm),*

b) $\|A\|_\infty = \max_{i\in\{1,\ldots n\}} \sum_{j=1}^n |a_{ij}|$ *(Zeilensummennorm),*

c) $\|A\|_2 = \max\{\sqrt{\lambda} \mid \lambda$ *ist Eigenwert von* $A^\top A\}$ *(Spektralnorm).*

Beweis. *Zum Beweis von a). sei x mit* $\|x\|_1 = \sum_{j=1}^n |x_j| = 1$. *Dann gilt*

$$\begin{aligned}
\|Ax\|_1 &= \sum_{i=1}^n \left|\sum_{j=1}^n a_{ij}x_j\right| \\
&\le \sum_{i,j=1}^n |a_{ij}||x_j| \\
&= \sum_{j=1}^n \left(|x_j| \sum_{i=1}^n |a_{ij}|\right) \\
&\le \sum_{j=1}^n \left(|x_j| \max_{k\in\{1,\ldots,n\}} \sum_{i=1}^n |a_{ik}|\right) \\
&\overset{\|x\|_1=1}{=} \max_{j\in\{1,\ldots,n\}} \sum_{i=1}^n |a_{ij}|.
\end{aligned}$$

Die erste Behauptung folgt nun aus $\|Ae_j\|_1 = \sum_{i=1}^n |a_{ij}|$.

ad b).: Sei nun x mit $\|x\|_\infty = \max_{j\in\{1,\ldots,n\}} |x_j| = 1$. *Dann gilt*

$$\begin{aligned}
\|Ax\|_\infty &= \max_{i\in\{1,\ldots,n\}} \left|\sum_{j=1}^n a_{ij}x_j\right| \\
&\le \max_{i\in\{1,\ldots,n\}} \left\{\sum_{j=1}^n |a_{ij}||x_j|\right\} \\
&\overset{|x_j|\le 1}{\le} \max_{i\in\{1,\ldots,n\}} \sum_{j=1}^n |a_{ij}|.
\end{aligned}$$

Die Behauptung folgt somit, wenn man den $(1,-1)$*-Vektor betrachtet, der die gleichen Vorzeichen hat wie die Zeile, in der das Maximum angenommen wird.*

Für den Beweis von c) erinnern wir uns zunächst daran, daß es, da $A^\top A$ symmetrisch ist, es eine Orthonormalbasis $b_1, \dots, b_n$ aus Eigenvektoren zu Eigenwerten $\lambda_1 \geq \dots \geq \lambda_n \overset{!}{\geq} 0$ von $A^\top A$ gibt. Sei nun $x = \sum_{i=1}^n \beta_i b_i$ mit $\|x\|_2 = \sqrt{x^\top x} = \sqrt{\sum_{i=1}^n \beta_i^2} = 1$. Dann ist

$$\begin{aligned}
\|Ax\|_2^2 &= x^\top A^\top Ax \\
&= (\sum_{i=1}^n \beta_i b_i^\top) A^\top Ax \\
&= (\sum_{i=1}^n \beta_i \lambda_i b_i^\top)(\sum_{j=1}^n \beta_j b_j) \\
&= \sum_{i=1}^n \beta_i^2 \lambda_i \\
&\leq \lambda_1.
\end{aligned}$$

Die Behauptung folgt nun aus $b_1^\top A^\top A b_1 = \lambda_1 b_1^\top b_1$. □

2.7 Kondition

Die motivierende Fragestellung dieses Abschnitts ist: Wie wirken sich Datenfehler bei der Aufgabe $Ax = b$ mit einer regulären Matrix A (bei exakter Lösung) auf den Vektor x aus. In diesem Kapitel gehen wir davon aus, daß eine feste Vektornorm $\|\cdot\|_{\mathbb{R}^n}$ mit zugehöriger Matrixnorm $\|\cdot\|_{\mathbb{R}^n \to \mathbb{R}^n}$ gegeben ist.

Beschränken wir uns zunächst auf eine fehlerbehaftete rechte Seite

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

Hieraus ergibt sich $\Delta x = A^{-1}\Delta b$ und somit

$$\|\Delta x\| \leq \|A^{-1}\| \, \|\Delta b\|.$$

Mit $\|b\| \leq \|A\| \, \|x\|$ erhalten wir $\|x\| \geq \frac{\|b\|}{\|A\|}$ und hieraus folgende Abschätzung für den relativen Fehler

$$\frac{\|\Delta x\|}{\|x\|} \leq \|A^{-1}\| \, \|A\| \frac{\|\Delta b\|}{\|b\|}.$$

Definition 2.7.1 *Sei $A \in \mathbb{R}^{n \times n}$ eine reguläre Matrix. Die Zahl $cond(A) := \|A^{-1}\| \, \|A\|$ heißt* Kondition *der Matrix A.*

Die Kondition ist abhängig von der gewählten Norm. Für die natürliche Matrixnorm eines normierten Raumes gilt

$$\operatorname{cond}(A) = \|A^{-1}\|\,\|A\| \geq \|A^{-1}A\| = \|I_n\| = 1.$$

Um auch Auswirkungen von Störungen von A abschätzen zu können, beweisen wir zunächst:

Lemma 2.7.2 *Sei $A \in \mathbb{R}^{n\times n}$ und $\|A\| < 1$. Dann ist $I_n + A$ regulär und*

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

Beweis. Wegen $\|x + Ax - Ax\| \leq \|x + Ax\| + \|Ax\|$ gilt:

$$\|(I_n+A)x\| = \|x+Ax\| \geq \|x\| - \|A\|\|x\| \geq (1-\|A\|)\|x\|.$$

Folglich kann $(I_n + A)x = 0$ nur gelten, wenn $x = 0$ ist, somit muß $I_n + A$ regulär sein. Ferner haben wir

$$1 \leq \|(I_n+A)^{-1}\|\,\|I_n+A\| \overset{(N3)}{\leq} \|(I_n+A)^{-1}\|(1+\|A\|)$$

und

$$\begin{aligned}
\|(I_n+A)^{-1}\| &= \|(I_n+A)^{-1} + (I_n+A)^{-1}A - (I_n+A)^{-1}A\| \\
&\leq \|(I_n+A)^{-1}(I_n+A)\| + \|(I_n+A)^{-1}A\| \\
&\leq 1 + \|(I_n+A)^{-1}\|\,\|A\|
\end{aligned}$$

also

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

□

Wir sind an den Auswirkungen einer Störung von A bei der Lösung von $Ax = b$ und damit an einer Abschätzung von $\operatorname{cond}(A + \Delta A)$ interessiert. Wir folgern deshalb:

Lemma 2.7.3 (Störungslemma) *Seien $A, B \in \mathbb{R}^{n\times n}$, A regulär und $\|A^{-1}\|\,\|B - A\| < 1$. Dann ist auch B regulär und*

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

Beweis. Nach Voraussetzung ist $\|A^{-1}(B-A)\| \leq \|A^{-1}\| \, \|(B-A)\| < 1$. Nach Lemma 2.7.2 ist also $(I_n + A^{-1}B - I_n)$ regulär, also auch B regulär, und es gilt

$$\begin{aligned} \|B^{-1}\| \leq \|B^{-1}A\| \, \|A^{-1}\| &\leq \|A^{-1}\| \frac{1}{1 - \|A^{-1}B - I_n\|} \\ &\leq \|A^{-1}\| \frac{1}{1 - \|A^{-1}\| \, \|B-A\|}. \end{aligned}$$

□

Nun können wir abschließend folgenden Satz beweisen.

Satz 2.7.4 *Seien $A, \Delta A \in \mathbb{R}^{n\times n}$ und gelte $\|A^{-1}\| \, \|\Delta A\| < 1$. Seien x bzw. $x + \Delta x$ Lösungen des Systems $Ax = b$ bzw. $(A + \Delta A)(x + \Delta x) = b$. Dann läßt sich der relative Fehler abschätzen durch*

$$\frac{\|\Delta x\|}{\|x\|} \leq \frac{cond\,(A)}{1 - cond(A)\frac{\|\Delta A\|}{\|A\|}} \frac{\|\Delta A\|}{\|A\|}.$$

Beweis. Nach Voraussetzung ist $\|A^{-1}\| \, \|A + \Delta A - A\| < 1$. Also ist nach dem Störungslemma $A + \Delta A$ regulär und

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

Aus $(A + \Delta A)(x + \Delta x) - Ax = 0$ schließen wir $\Delta x = -(A + \Delta A)^{-1}\Delta A x$. Durch Einsetzen erhalten wir

$$\begin{aligned} \|\Delta x\| &\leq \frac{\|A^{-1}\|}{1 - \|A^{-1}\| \, \|\Delta A\|} \|\Delta A\| \, \|x\| \\ &= \frac{\|A^{-1}\| \, \|A\|}{1 - \|A^{-1}\| \|A\| \, \frac{\|\Delta A\|}{\|A\|}} \frac{\|\Delta A\|}{\|A\|} \, \|x\|, \end{aligned}$$

woraus die Behauptung folgt □

Kapitel 3

Lineare Optimierung

In den folgenden Kapiteln wollen wir uns mit Optimierungsproblemen und ihren Algorithmen beschäftigen. Ein Optimierungsproblem besteht für uns aus einem zulässigen Bereich $S \subseteq \mathbb{R}^n, \mathbb{Z}^n$ und einer Zielfunktion $c : S \to \mathbb{R}$. Ziel ist die Bestimmung von inf, sup, max oder $\min_{x \in S} c(x)$. Im allgemeinen sind solche Probleme beliebig schwer. Deswegen behandeln wir die Aufgabenstellung nicht in aller Allgemeinheit.

Beschränken wir uns zunächst auf den einfachsten Fall, daß $c(x) = c^\top x$ eine lineare Funktion ist und S eine Teilmenge des $\mathbb{R}^n$ ist, die durch eine Menge linearer Ungleichungen der Form $a^\top x \leq b$ gegeben ist.

3.1 Modellbildung

Lineare Programmierungsaufgaben aus der industriellen Praxis sind üblicherweise groß (tausende Variablen und Ungleichungen). Aus Gründen der Übersichtlichkeit behandeln wir hier eher "toy problems".

Beispiel 3.1.1 *Eine Ölraffinierie hat vier verschiede Sorten Rohbenzin zur Verfügung und mischt daraus Benzin in vier verschiedenen Oktanstärken. Dafür gelten folgende Daten*

Rohstoffsorte	*Oktanzahl*	*Fässer verfügbar*	*Preis pro Faß*
1	*68*	*4000*	*\$ 31.02*
2	*86*	*5050*	*\$ 33.15*
3	*91*	*7100*	*\$ 36.35*
4	*99*	*4300*	*\$ 38.75*

Benzinsorte	*Mindestoktanzahl*	*Nachfrage*	*Preis pro Faß*
1	*85*	$\geq$ *15000*	*\$ 40.99*
2	*90*	*beliebig*	*\$ 42.95*
3	*95*	$\leq$ *10000*	*\$ 45.15*

Gesucht ist ein Produktionsprozeß, der den Gewinn maximiert. Wir setzen

x_{ij} := *Fässer des Rohstoffs i, die zur Produktion von Sorte j benutzt werden.*

Die Daten aus der ersten Tabelle liefern dann folgende Restriktionen.

$$\begin{aligned} x_{1,1}+x_{1,2}+x_{1,3} &\leq 4000 \\ x_{2,1}+x_{2,2}+x_{2,3} &\leq 5050 \\ x_{3,1}+x_{3,2}+x_{3,3} &\leq 7100 \\ x_{4,1}+x_{4,2}+x_{4,3} &\leq 4300 \end{aligned}$$

Auch die Anforderungen an die Qualität der Mischungen ergeben lineare Bedingungen, denn wir haben z.B.

$$\frac{68x_{1,1}+86x_{2,1}+91x_{3,1}+99x_{4,1}}{x_{1,1}+x_{2,1}+x_{3,1}+x_{4,1}} \geq 85.$$

Wir erhalten somit folgende drei Bedingungen

$$\begin{aligned} 68x_{1,1}+86x_{2,1}+91x_{3,1}+99x_{4,1}-85(x_{1,1}+x_{2,1}+x_{3,1}+x_{4,1}) &\geq 0 \\ 68x_{1,2}+86x_{2,2}+91x_{3,2}+99x_{4,2}-90(x_{1,2}+x_{2,2}+x_{3,2}+x_{4,2}) &\geq 0 \\ 68x_{1,3}+86x_{2,3}+91x_{3,3}+99x_{4,3}-95(x_{1,3}+x_{2,3}+x_{3,3}+x_{4,3}) &\geq 0 \end{aligned}$$

Der Mindest und Höchstabsatz ergeben:

$$\begin{aligned} x_{1,1}+x_{2,1}+x_{3,1}+x_{4,1} &\geq 15000 \\ x_{1,3}+x_{2,3}+x_{3,3}+x_{4,3} &\leq 10000 \end{aligned}$$

Als Zielfunktion, die wir maximieren wollen, erhalten wir $c(x) = 40.99(x_{1,1}+x_{2,1}+x_{3,1}+x_{4,1})+42.95(x_{1,2}+x_{2,2}+x_{3,2}+x_{4,2})+45.15(x_{1,3}+x_{2,3}+x_{3,3}+x_{4,3})-31.02(x_{1,1}+x_{1,2}+x_{1,3})-33.15(x_{2,1}+x_{2,2}+x_{2,3})-36.35(x_{3,1}+x_{3,2}+x_{3,3})-38.75(x_{4,1}+x_{4,2}+x_{4,3})$ *und bemerken abschließend, daß alle* $x_{ij} \geq 0$ *sein sollten.*

In Tabellenform erhalten wir also zunächst folgende Aufgabenstellung:

$x_{1,1}$	$x_{2,1}$	$x_{3,1}$	$x_{4,1}$	$x_{1,2}$	$x_{2,2}$	$x_{3,2}$	$x_{4,2}$	$x_{1,3}$	$x_{2,3}$	$x_{3,3}$	$x_{4,3}$		
9.97	*7.84*	*4.64*	*2.24*	*11.93*	*9.80*	*6.60*	*4.20*	*14.13*	*12.00*	*8.80*	*6.40*		
1	*0*	*0*	*0*	*1*	*0*	*0*	*0*	*1*	*0*	*0*	*0*	$\leq$	*4000*
0	*1*	*0*	*0*	*0*	*1*	*0*	*0*	*0*	*1*	*0*	*0*	$\leq$	*5050*
0	*0*	*1*	*0*	*0*	*0*	*1*	*0*	*0*	*0*	*1*	*0*	$\leq$	*7100*
0	*0*	*0*	*1*	*0*	*0*	*0*	*1*	*0*	*0*	*0*	*1*	$\leq$	*4300*
-17	*1*	*6*	*14*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	$\geq$	*0*
0	*0*	*0*	*0*	*-22*	*-4*	*1*	*9*	*0*	*0*	*0*	*0*	$\geq$	*0*
0	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*-27*	*-9*	*-4*	*4*	$\geq$	*0*
1	*1*	*1*	*1*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	$\geq$	*15000*
0	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*1*	*1*	*1*	*1*	$\leq$	*10000*

Zur allgemeinen Behandlung des Problems ist es ungünstig, eine Mischung aus $\leq, \geq$ und = Restriktionen zu haben. Außerdem wollen wir im Falle der Linearen Programmierung – wie im vorliegenden Beispiel und abweichend von der Einleitung – lieber Maximierungsprobleme als Minimierungsprobleme betrachten. Das macht keinen Unterschied, da $\max_{x \in S} c(x) = -(\min_{x \in S} -c(x))$. Deswegen definieren wir.

Definition 3.1.2 *Sei $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $b \geq 0$ und $c \in \mathbb{R}^n$. Ferner sei $m \geq n$ und A habe vollen Rang, also rang$(A) = n$. Die Aufgabenstellung*

$$\begin{array}{ll} \max & c^\top x \\ \textit{unter} & Ax = b \\ & x \geq 0 \end{array}$$

nennen wir Lineares Optimierungsproblem in Standardform. *Ist $x \geq 0$ mit $Ax = b$, so sagen wir x ist* zulässig *für das Problem.*

Um Beispiel 3.1.1 in Standardform zu bringen, führen wir für die neun Ungleichungen sogenannte *Schlupfvariabelen* $y_1, \ldots, y_9$ ein und setzen

$$A = \begin{pmatrix}
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
-17 & 1 & 6 & 14 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -22 & -4 & 1 & 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -27 & -9 & -4 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}$$

$$b^\top = (4000, 5050, 7100, 4300, 0, 0, 0, 15000, 10000))$$

und $c^\top =$
(9.97,7.84,4.64,2.24,11.93,9.80 ,6.60,4.20,14.13,12.00,8.80,6.40,0,0,0,0,0,0,0,0,0).

Bemerkung 3.1.3 *Nicht vorzeichenbeschränkte Variablen u kann man mit $u = u^+ - u^-$ und $u^+, u^- \geq 0$ durch zwei vorzeichenbeschränkte ersetzen und so in Standardform bringen.*

Der älteste und immer noch praktisch wichtigste Algorithmus zur Lösung linearer Programme ist der Simplexalgorithmus. In den letzten zehn Jahren haben die innere-Punkt-Verfahren, die Ideen aus der Nichtlinearen Programmierung benutzen, an Bedeutung gewonnen.

Zur Lösung linearer Programme gibt es Standardpakete wie CPLEX, XPRESS oder OSL. Für nichtkommerzielle Belange gibt es auch hinreichend gute Software in der Public Domain. Ein Startpunkt ist die Electronic Library of Mathematical Software http://elib.zib.de.

CPLEX liefert als Lösung des obigen Problems:

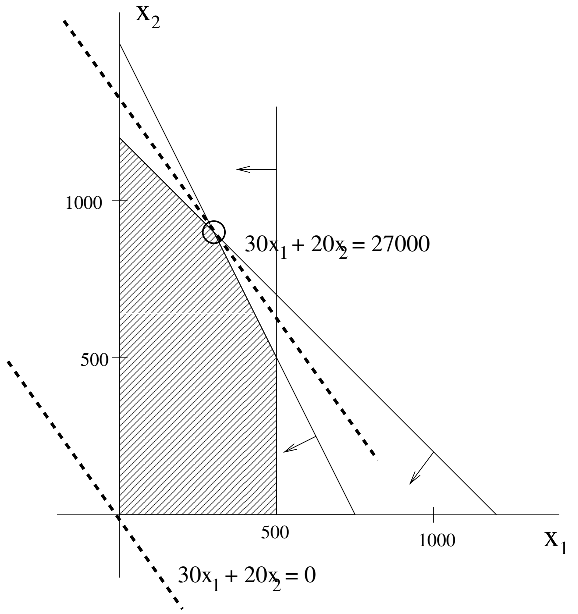
```
CPLEX> display problem all
Minimize
 ZIEL: - 9.97 X11 - 11.93 X12 - 14.13 X13 - 7.84 X21 - 9.8 X22 - 12 X23
 - 4.64 X31 - 6.6 X32 - 8.8 X33 - 2.24 X41 - 4.2 X42 - 6.4 X43
Subject To
 RB1: X11 + X12 + X13 <= 4000
 RB2: X21 + X22 + X23 <= 5050
 RB3: X31 + X32 + X33 <= 7100
 RB4: X41 + X42 + X43 <= 4300
 O85: - 17 X11 + X21 + 6 X31 + 14 X41 >= 0
 O90: - 22 X12 - 4 X22 + X32 + 9 X42 >= 0
 O95: - 27 X13 - 9 X23 - 4 X33 + 4 X43 >= 0
 NF1: X11 + X21 + X31 + X41 >= 15000
 NF2: X13 + X23 + X33 + X43 <= 10000
Bounds
 All variables are >= 0.
CPLEX> optimize
Primal - Optimal:  Objective =   -1.3862560000e+05
Solution time =    0.00 sec.  Iterations = 10 (4)

CPLEX> display solution variables -
Variable Name           Solution Value
X11                        4000.000000
```

```
X21                        3823.846154
X23                        1226.153846
X31                        7100.000000
X41                        1541.153846
X43                        2758.846154
All other variables in the range 1-12 are zero.
```

Die algorithmische Idee des Simplexverfahrens und die Geometrie der Aufgabenstellung erkennt man am leichtesten an Beispielen wie dem folgenden:

Beispiel 3.1.4 (Graphische Lösung von Problemen mit 2 Variablen) *Eine Düngemittelfabrik stellt zwei Sorten Dünger A und B aus drei Ausgangsstoffen C,D,E her. Vom Ausgangstoff C stehen pro Periode 1500 Tonnen zur Verfügung, von D 1200 t und von E 500 t. Zur Produktion einer Tonne A werden 2 Tonnen C, und jeweils 1 Tonne D und E benötigt, für B jeweils eine Tonne C und D. Der Gewinn pro Tonne A beträgt 30,- DM, pro Tonne B 20,-. Wir erhalten also das Programm.*

$$
\begin{array}{rrcrcr}
\max & 30x_1 & + & 20x_2 & & \\
\textit{unter} & 2x_1 & + & x_2 & \leq & 1500 \\
& x_1 & + & x_2 & \leq & 1200 \\
& x_1 & & & \leq & 500 \\
& x_1 & , & x_2 & \geq & 0
\end{array}
$$



Die Ungleichungen definieren jeweils einen Halbraum. Der zulässige Bereich ist als Schnitt von Halbräumen ein sogenanntes Polyeder. *Die Isogewinnflächen sind*

Hyperebenen (in der Ebene Geraden). Die Optimallösung erhält man, in dem man die Isogewinnfläche so weit wie möglich in Richtung wachsender Erlöse verschiebt, bis sie das Polyeder nur noch berührt. Betrachten wir die Ecken des Polyeders etwas genauer. Sie sind dadurch definiert, daß zwei Ungleichungen nicht strikt sind, sondern mit Gleichheit angenommen werden. Dies ist zunächst $x_1 = 0, x_2 = 0$. *Wenn wir mit der Isogewinnfläche in Richtung wachsender Erlöse passieren wir* $(x_1 = 500, x_2 = 0), (x_1 = 500, 2x_2 + x_1 = 1500), (x_1 + x_2 = 1200, 2x_1 + x_2 = 1500)$. *Aus letzterem erhalten wir* $x_1 = 300, x_2 = 900$.

Bevor wir in Abschnitt 3.4 auf diese Idee zurückkommen, müssen wir den Korrektheitsbeweis mit etwas Theorie vorbereiten. Abschließend definieren wir Polyeder noch formal.

Definition 3.1.5 *Eine Menge* $P \subseteq \mathbb{R}^n$ *heißt Polyeder, wenn es ein* $m \in \mathbb{N}$, *eine* $\mathbb{R}^{m\times n}$ *Matrix* A *und ein* $b \in \mathbb{R}^m$ *gibt mit*

$$P = \{x \in \mathbb{R}^n \mid Ax \leq b\}.$$

Übung 3.1.6 *Zeigen Sie: Der zulässige Bereich eines linearen Programms in Standardform ist ein Polyeder.*

3.2 Farkas' Lemma

Lemma 3.2.1 (Farkas' Lemma) *Sei* $L \subseteq \mathbb{R}^n$ *ein Vektorraum. Dann gilt genau eine der beiden folgenden Altenativen:*

a) $\exists x \in L : x_1 > 0, x \geq 0$

b) $\exists y \in L^\perp : y_1 > 0, y \geq 0$.

Beweis. Offensichtlich können nicht beide Alternativen gleichzeitig gelten, da es ansonsten x, y gäbe mit $0 = x^\top y = \sum_{i=1}^n x_i y_i \geq x_1 y_1 > 0$.

Für den Rest der Behauptung führen wir vollstandige Induktion über n.

$n = 1$: Hier ist entweder $L = \mathbb{R}$ oder $L^\perp = \mathbb{R}$. Also gilt die Behauptung.

$n \geq 2$. Wir betrachten folgende Vektorräume(!) in $\mathbb{R}^{n-1}$

$$\begin{aligned}
\tilde{L} &:= \{\tilde{x} \in \mathbb{R}^{n-1} \mid \exists x \in L, x_n = 0, x_i = \tilde{x}_i, i = 1, \ldots, n-1\} \\
\hat{L} &:= \{\hat{x} \in \mathbb{R}^{n-1} \mid \exists x \in L, x_i = \hat{x}_i, i = 1, \ldots, n-1\} \\
\tilde{L^\perp} &:= \{\tilde{y} \in \mathbb{R}^{n-1} \mid \exists y \in L^\perp, y_n = 0, y_i = \tilde{y}_i, i = 1, \ldots, n-1\} \\
\hat{L^\perp} &:= \{\hat{y} \in \mathbb{R}^{n-1} \mid \exists y \in L^\perp, y_i = \hat{y}_i, i = 1, \ldots, n-1\}
\end{aligned}$$

Offensichtlich gilt dann $(\tilde{L})^\perp = \hat{L^\perp}$ und $(\hat{L})^\perp = \tilde{L^\perp}$. Falls es nun ein $\tilde{x} \in \tilde{L}$ mit $\tilde{x}_1 > 0, \tilde{x} \geq 0$ gibt, so leistet das zugehörige x das Gewünschte. Ebenso sind wir fertig, falls es ein $\tilde{y} \in \tilde{L^\perp}$ gibt mit $y_1 > 0, y \geq 0$. Wir können also davon ausgehen, daß jeweils die andere Alternative gilt, d.h. es gibt $x \in L, y \in L^\perp$ mit $x_1, y_1 > 0$ und $x_i, y_i \geq 0$ für $i = 1, \ldots, n-1$. Aus $x^\top y = 0$ erhalten wir somit

$$\begin{aligned}
-x_n y_n &= x_1 y_1 + \sum_{i=2}^{n-1} x_i y_i \\
&\geq x_1 y_1 \\
&> 0.
\end{aligned}$$

Wir schließen, daß genau einer von x_n und y_n echt positiv ist. Falls $x_n > 0$ ist, so leistet x das Gewünschte für die erste Alternative, im anderen Fall y für die zweite. □

Farkas' Lemma ist in einigen anderen Erscheinungsformen bekannt. Am häufigsten wird es wohl zitiert als

> Ist $P \subseteq \mathbb{R}^n$ ein Polytop und $x \in \mathbb{R}^n$, so gilt entweder $x \in P$ oder es gibt eine affine Hyperebene, die x und P trennt.

Ein Polytop ist ein beschränktes Polyeder, das in diesem Satz als *konvexe Hülle* von Punkten gegeben ist und nicht als Schnitt von Halbräumen. Die beiden Polyederbegriffe sind äquivalent, was wir im Rahmen dieser Vorlesung nicht beweisen werden. Bevor wir diese Version von Farkas' Lemma herleiten, erinnern wir zunächst an eine Tatsache aus der Linearen Algebra und definieren die konvexe Hülle.

Proposition 3.2.2 *Sei $A \in \mathbb{R}^{m \times n}$. Dann sind*

$$L = ker(A) := \{x \in \mathbb{R}^n \mid Ax = 0\}$$

und

$$im(A^\top) := \{x \in \mathbb{R}^n \mid \exists y \in \mathbb{R}^m : x = A^\top y\}$$

ein orthokomplementäres Paar von Unterräumen des $\mathbb{R}^n$, d.h. $im(A^\top) = L^\perp$.

Definition 3.2.3 *Seien $x_1, \dots, x_k \in \mathbb{R}^n$ und $\lambda \in \mathbb{R}^k$. Dann heißt*

$$x = \sum_{i=1}^{k} \lambda_i x_i$$

Linearkombination *von $x_1, \dots, x_k$. Gilt zusätzlich*

$$\left\{\begin{array}{ll} \lambda \geq 0 & \\ & \sum_{i=1}^k \lambda_i = 1 \\ \lambda \geq 0, & \sum_{i=1}^k \lambda_i = 1 \end{array}\right\}, \textit{ so heißt } x \left\{\begin{array}{l} \text{konische} \\ \text{affine} \\ \text{Konvex-} \end{array}\right\} \text{Kombination.}$$

Die Kombination heißt echt, wenn es $\lambda_i \neq 0 \neq \lambda_j$ mit $x_i \neq x_j$ gibt, oder der einzige Nichtnulleintrag von λ, etwa λ_i, von Eins verschieden ist und $x_i \neq 0$ ist. $S \subseteq \mathbb{R}^n$ sei

$$\left\{\begin{array}{l} lin\,(S) \\ cone\,(S) \\ aff\,(S) \\ conv\,(S) \end{array}\right\}, \textit{ die Menge aller } \left\{\begin{array}{l} \text{linearen} \\ \text{konischen} \\ \text{affinen} \\ \text{konvexen} \end{array}\right\} \textit{Kombinationen von}$$

Elementen in S. Wir sprechen von der Hülle.

Kommen wir also zum angekündigten Trennungssatz.

Korollar 3.2.4 *Seien $v_1, \dots, v_k, x \in \mathbb{R}^n$. Dann gilt:*
Entweder $x \in conv\,(\{v_1, \dots, v_k\})$ oder es gibt $a \in \mathbb{R}^n, b \in \mathbb{R}$, so daß $a^\top v_i \leq b$ für $i = 1, \dots, k$ und $a^\top x > b$.

Beweis. Wir setzen

$$A = \begin{pmatrix} -x & v_1 & \dots & v_k \\ -1 & 1 & \dots & 1 \end{pmatrix}$$

und wenden Farkas' Lemma auf das Paar $L = \ker(A), L^\perp = \operatorname{im}(A)$ an. Dann gilt: entweder es gibt $(\mu_0, \mu) \in \mathbb{R}^{n+1}$, $\mu \geq 0, \mu_0 > 0$ mit $A\mu = 0$ oder es gibt ein $\binom{a}{a_{k+1}} \in \mathbb{R}^{k+1}$ mit $A^\top a \geq 0$ und $(A^\top \binom{a}{a_{k+1}})_1 > 0$. Letzteres bedeutet aber $(-a)^\top x > -a_{k+1}$ und $a^\top v_i \geq a_{k+1}$, also $(-a)^\top v_i \leq -a_{k+1}$ für $i = 1, \dots, k$. Im ersten Fall setzen wir $\lambda_i := \frac{\mu_i}{\mu_0}$ und erhalten $-x + \sum_{i=1}^k \lambda_i v_i = 0$ und $-1 + \sum_{i=1}^k \lambda_i = 0$. □

3.3 Dualitätssatz

Wir wollen nun die Resultate aus dem letzten Kapitel auf die Lineare Programmierung anwenden. Dafür leiten wir zunächst eine „gute Charakterisierung“ wie folgt her. Betrachten wir die Behauptung

Das Lineare Programm

$$(P)\qquad \begin{array}{ll} \max & c^\top x \\ \text{unter} & Ax = b \\ x \geq 0 & \end{array}$$

hat mindestens einen Wert von z_0.

Die Richtigkeit dieser Behauptung läßt sich durch Angabe eines $x \geq 0$ mit $Ax = b$ sowie $c^\top x \geq z_0$ beweisen. Wenn es kein solches x gibt, hilft uns Farkas' Lemma, dafür einen Beweis anzugeben. Betrachten wir nämlich den Kern der Matrix.

$$B := \begin{pmatrix} -b & A & 0 \\ -z_0 & c^\top & -1 \end{pmatrix}$$

so enthält dieser kein nichtnegatives Element, das in der ersten Komponente echt positiv ist. Denn wäre (x_0, x, x_{n+1}) ein solches Element, so haben wir $A\frac{x}{|x_0|} = b$, $c^\top \frac{x}{|x_0|} \geq z_0$. Nach Farkas' Lemma gibt es also ein (y, t) mit $y^\top b + tz_0 < 0, y^\top A + tc^\top \geq 0, t \leq 0$. Dies motiviert folgende Definition:

Definition 3.3.1 *Das Lineare Programm*

$$(D)\qquad \begin{array}{ll} \min & y^\top b \\ \textit{unter} & y^\top A \geq c^\top \end{array}$$

heißt das duale Programm *zum Linearen Programm in Standardform.*

Wir werden im folgenden zeigen, daß, im wesentlichen, das duale und primale Programm den gleichen Zielfunktionswert haben. Dies hilft beim Algorithmendesign.

Lemma 3.3.2 (Schwache Dualität) *Ist $x \geq 0$ zulässig für das primale Programm und y für das duale Programm, so gilt $c^\top x \leq y^\top b$.*

Beweis.

$$c^\top x \overset{x \geq 0}{\leq} y^\top Ax = y^\top b.$$

□

Satz 3.3.3 (Dualitätssatz der Linearen Programmierung) *Seien $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$ und $c \in \mathbb{R}^n$. Dann gilt genau eine der folgenden vier Alternativen:*

a) *Das primale Programm ist zulässig und beschränkt. Ist dann z_0 eine kleinste obere Schranke, so ist auch das duale Programm zulässig und beschränkt und es gibt eine Optimallösung y^* mit $z_0 = y^{*\top} b$.*

b) *Das primale Programm ist unbeschränkt, d.h. es gibt ein $x_0 \geq 0$ mit $Ax_0 = b$ und ein $x_1 \geq 0$ mit $Ax_1 = 0$ und $c^\top x_1 > 0$.*

c) *Das duale Programm ist unbeschränkt, d.h. es gibt ein y_0 mit $y_0^\top A \geq c$ und ein y_1 mit $y_1^\top A \geq 0$ und $y_1^\top b < 0$.*

d) *Beide Programme sind unzulässig, d.h. es gibt weder ein $x \geq 0$ mit $Ax = b$, noch ein $y \in \mathbb{R}^m$ mit $y^\top A \geq c^\top$.*

Beweis. Wegen Lemma 3.3.2 kann stets höchstens eine Alternative gelten. Sei zunächst das primale Programm zulässig. Ist es unbeschränkt, so gilt Alternative b). Also nehmen wir an, das primale Programm sei beschränkt, sei dann z_0 eine kleinste obere Schranke. Da z_0 obere Schranke ist, gibt es kein $x \geq 0$ mit $Ax - b = 0$ und $c^\top x > z_0$. Wir betrachten den Kern von

$$B := \begin{pmatrix} 0 & A & -b \\ 1 & -c^\top & z_0 \end{pmatrix}.$$

Falls $(w_0, w^\top, w_{n+1})$ im Kern von B existiert mit $w_0 > 0, w \geq 0, w_{n+1} \geq 0$, so impliziert die Schrankeneigenschaft von z_0 sofort $w_{n+1} = 0$. Denn ansonsten wäre $x := \frac{1}{w_{n+1}} w$ zulässig für das primale Programm und $c^\top x > z_0$. Daraus folgt aber $Aw = 0$ und $c^\top w = w_0 > 0$, d.h. das primale Programm ist unbeschränkt im Widerspruch zur Annahme. Nach Farkas' Lemma gibt es also ein $(\tilde{y}, y_{m+1})$ mit $y_{m+1} > 0, \tilde{y}^\top A - c^\top y_{m+1} \geq 0, y_{m+1} z_0 \geq \tilde{y}^\top b$. Nach Division durch y_{m+1} haben wir somit ein y mit $y^\top A \geq c$ und $y^\top b \leq z_0$. Da z_0 kleinste obere Schranke ist und wegen Lemma 3.3.2 $y^\top b$ obere Schranke ist, gilt ebenso $y^\top b \geq z_0$.

Ist nun das primale Programm unzulässig, betrachten wir $(-b, A)$ und Farkas' Lemma impliziert die Existenz eines y mit $y^\top b < 0$ und $y^\top A \geq 0$. Also ist das duale Programm, falls es zulässig ist, unbeschränkt. □

Daß auch die letzte Alternative eintreten kann, zeigt folgendes Beispiel:

Beispiel 3.3.4

$$\begin{array}{llrcrcl} & \max & x_1 & + & x_2 & & \\ (P) & \textit{unter} & x_1 & - & x_2 & = & 1 \\ & & -x_1 & + & x_2 & = & 1 \\ & & x_1 & , & x_2 & \geq & 0 \end{array}$$

$$
\begin{array}{llrcrcl}
 & \min & y_1 & + & y_2 & & \\
(D) & \textit{unter} & y_1 & - & y_2 & \geq & 1 \\
 & & -y_1 & + & y_2 & \geq & 1
\end{array}
$$

Übung 3.3.5 *Zeigen Sie: Das duale Programm des dualen Programms ist das primale Programm. (Bringen Sie das duale Programm in Standardform und dualisieren Sie.) Folgern Sie hieraus: Ist das primale Programm zulässig und beschränkt, so gibt es für beide Programme Optimallösungen x^* und y^* und es gilt: $c^\top x^* = {y^*}^\top b$. Diskutieren Sie den Unterschied zu Satz 3.3.3 a).*

Haben wir nun ein duales Paar linearer Programme, bei der die erste Alternative von Satz 3.3.3 gilt, so betrachten wir ein duales Paar von Optimallösungen x^*, y^*. Es ist $(y^*)^\top b - c^\top x^* = ({y^*}^\top A - c^\top)x^* = 0$. Da $x^* \geq 0$ und ${y^*}^\top A - c^\top \geq 0$, schließen wir hieraus den

Satz 3.3.6 (Satz vom komplementären Schlupf) *Seien $x^* \in \mathbb{R}^n_+$ mit $Ax = b$ und $y^* \in \mathbb{R}^m$ mit ${y^*}^\top A \geq c^\top$. Dann sind x^*, y^* genau dann ein Paar von Optimallösungen des primalen bzw. dualen Programmes, wenn gilt:*

a) $x_i^* \neq 0 \Rightarrow (c^\top = {y^*}^\top A)_i$ *und*

b) $(c^\top > {y^*}^\top A)_i \Rightarrow x_i^* = 0.$

Beweis. Die Notwendigkeit der Bedingung hatten wir vor Formulierung des Satzes hergeleitet. Aus den Bedingungen folgt nun

$$c^\top x^* = \sum_{i=1}^n c_i x_i^* = \sum_{i=1}^n ({y^*}^\top A)_i x^* i = {y^*}^\top A x^* = {y^*}^\top b$$

also folgt aus Lemma 3.3.3 die Behauptung.

3.4 Das Simplexverfahren

Die geometrische Idee des Simplexverfahrens hatten wir im zweidimensionalen Fall schon erläutert. Wir gehen so lange von einer Ecke zu einer besseren Ecke, bis es nicht mehr besser geht. Ecken des Standardprogramms erhalten wir dadurch, daß wir Ungleichungen zu Gleichungen machen, hier also durch Nullsetzen möglichst vieler Variablen.

Definition 3.4.1 *Sei P ein Polyeder. Dann heißt $x \in P$* Ecke *von P, wenn x nicht echte Konvexkombination von Elementen in P ist.*

Lemma 3.4.2 *Habe $A \in \mathbb{R}^{m\times n}$ vollen Zeilenrang und seien $b \in \mathbb{R}^m, x \in \mathbb{R}^n, x \geq 0$. Dann ist x Ecke von $P := \{x \in \mathbb{R}^n \mid Ax = b, x \geq 0\}$ genau dann, wenn es $B \subseteq \{1,\dots,n\}$ gibt, mit $A_{.B}$ regulär, $x_B = A_{.B}^{-1}b$ und mit $N := \{1,\dots,n\} \setminus B$ gilt $x_N = 0$.*

Beweis. Seien $x^1,\dots,x^k \in P, \lambda_1,\dots,\lambda_k > 0, \sum_{i=1}^n \lambda_i = 1$ mit $x = \sum_{i=1}^n \lambda_i x_i$. Da die $x^i \geq 0$ und die $\lambda_i > 0$ sind, folgt aus $x_N = 0$, daß $x_N^i = 0$ für alle i. Aus $A_{.B}x_B^i = Ax_i = b$ schließen wir somit $x_B^i = A_{.B}^{-1}b = x_B$ für alle i. Also kann die Konvexkombination nicht echt sein und x ist eine Ecke. Nehmen wir nun für die andere Richtung der Aussage an, daß es ein solches B nicht gibt. Sei C die Menge der Indizes mit $x_C > 0$. Da C nicht zu einem B ergänzt werden kann, hat die Matrix $A_{.C}$ nicht vollen Spaltenrang. Also gibt es $0 \neq y_C \in \ker(A_{.C})$ und ein $\varepsilon > 0$ mit $x_C^1 := x_C + \varepsilon y_C > 0$ und $x_C^2 := x_C - \varepsilon y_C > 0$. Wir setzen $x^1_{\{1,\dots,n\}\setminus C} = 0 = x^2_{\{1,\dots,n\}\setminus C}$. Dann ist $x = \frac{1}{2}x^1 + \frac{1}{2}x^2$ keine Ecke. □

Definition 3.4.3 *Habe $A \in \mathbb{R}^{m\times n}$ vollen Zeilenrang und sei $b \in \mathbb{R}^m$. Wir sagen $B \subseteq \{1,\dots,n\}$ ist eine* Basis *von A, falls $A_{.B}$ regulär ist, B heißt* zulässige Basis, *wenn $A_{.B}^{-1}b \geq 0$ ist. Ist B eine (zulässige) Basis, so heißt $N := \{1,\dots,n\} \setminus B$ (zulässige) Nichtbasis und der Vektor $x \in \mathbb{R}^n$ mit $x_B = A_{.B}^{-1}b, x_N = 0$ (zulässige) Basislösung.*

Als nächstes werden wir untersuchen, wie man von einer zulässigen Ecke eine benachbarte bessere findet oder feststellt, daß die Ecke eine Optimallösung darstellt.

Proposition 3.4.4 (Optimalitätskriterium) *Habe $A \in \mathbb{R}^{m\times n}$ vollen Rang und sei $b \in \mathbb{R}^m$. Sei B eine zulässige Basis. Dann ist die zulässige Basislösung x eine Optimallösung des linearen Programms, wenn*

$$c^\top - c_B^\top A_{.B}^{-1}A \leq 0.$$

Beweis. Wir setzen $y^\top = c_B^\top A_{.B}^{-1}$. Dann ist $y^\top A \geq c^\top$ und $y^\top b = c_B^\top A_{.B}^{-1}b = c_B^\top x_B = c^\top x$ und die Behauptung folgt aus der schwachen Dualität. □

Der Term $c^\top - c_B^\top A_{.B}^{-1} A$ liefert nicht nur eine Optimalitätskriterium sondern, wie wir gleich sehen werden, auch Information darüber, welche Nachbarecken lohnend sind.

Definition 3.4.5 *Ist B eine zulässige Basis, so heißen $c^\top - c_B^\top A_{.B}^{-1} A$* die reduzierten Kosten.

Der Begriff reduzierte Kosten kommt daher, daß man früher zuerst Minimierungsprobleme eingeführt hat. In unserem Falle spräche man besser von reduzierten Gewinnen.

Definition 3.4.6 *Zwei Basen B_1, B_2 heißen benachbart, wenn $|B_1 \triangle B_2| = 2$.*

Lemma 3.4.7 *Seien B und $B' = B \cup \{j\} \setminus \{i\}$ zwei benachbarte zulässige Basen mit zugehörigen Basislösungen x und x'. Dann ist*

$$c^\top x' - c^\top x = x'_j (c_j - c_B^\top A_{.B}^{-1} A_{.j}).$$

Beweis. Wegen $A(x' - x) = 0$ sind die Nichtnulleinträge von $x' - x$ im Kern von $A_{.,B' \cup B}$, wobei $(x' - x)_j = x'_j$ und $(x' - x)_i = -x_i$. Also ist $-x'_j A_{.j} = A_{.B}(x' - x)_B$ und somit $(x' - x)_B = -x'_j A_{.B}^{-1} A_{.j}$. Somit erhalten wir

$$\begin{aligned} c^\top x' - c^\top x &= c_{B' \cup B}^\top (x' - x)_{B' \cup B} \\ &= c_j x'_j + c_B^\top (x' - x)_B \\ &= x'_j c_j - x'_j c_B^\top A_{.B}^{-1} A_{.j} \\ &= x'_j (c_j - c_B^\top A_{.B}^{-1} A_{.j}). \end{aligned}$$

□

Wenn x'_j und die j-ten reduzierten Kosten beide echt positiv sind, verbessern wir uns beim Wechsel von x nach x'. Haben also die reduzierten Kosten noch einen positiven Eintrag, so wollen wir den zugehörigen Index, etwa j einer solchen Variablen in die Basis B aufnehmen. Im Kern der Matrix $A_{.,B \cup j}$ gibt es genau eine vom Nullvektor verschiedene Richtung, "nämlich $e_j - A_{.B}^{-1} A_{.j}$". Diese verläuft von der Basislösung in Richtung einer *Kante* des Polyeders. In dieser Richtung wird die Variable x_j von Null auf einen positiven Wert gehoben, bis eine Bedingung $x_i \geq 0$ greift. Da x_i auf 0 gesetzt wird, nennen wir es das *basisverlassende Element*. Dies ist also der Index, bei dem bei Erhöhungen der neuen Basisvariablen der zugehörige x-Wert als erstes auf Null, fällt:

Proposition 3.4.8 *Ist B eine zulässige Basis, $i \in B \not\ni j$ und*

$$i \in argmin \left\{ \frac{(A_{.B}^{-1} b)_i}{(A_{.B}^{-1} A_{.j})_i} \mid (A_{.B}^{-1} A_{.j})_i > 0 \right\},$$

wobei argmin die Menge aller Indizes ist, an denen das Minimum angenommen wird, so ist $B' := B \cup \{j\} \setminus \{i\}$ eine zulässige Basis.

Beweis. Ist $x = (x_B, x_N) = (x_B, 0)$ zulässige Basislösung, $j \notin B$ und d definiert durch

$$d_k := \begin{cases} -A_{.B}^{-1} A_{.j} & \text{falls} \quad k \in B \\ 1 & \text{falls} \quad k = j \\ 0 & \text{sonst,} \end{cases}$$

so gilt für alle $\lambda : A(x + \lambda d) = b + \lambda(Ae_j - A_{.j}) = b$. Ist nun

$$\lambda_0 = \min\{\frac{(A_{.B}^{-1} b)_i}{(A_{.B}^{-1} A_{.j})_i} \mid (A_{.B}^{-1} A_{.j})_i > 0\},$$

so gilt außerdem für alle $k \in B$:

$$\begin{aligned} (x + \lambda_0 d)_k &= (A_{.B}^{-1} b)_k + \lambda_0(-A_{.B}^{-1} A_{.j})_k \\ &\geq (A_{.B}^{-1} b)_k + \frac{(A_{.B}^{-1} b)_k}{(A_{.B}^{-1} A_{.j})_k}(-A_{.B}^{-1} A_{.j})_k = 0. \end{aligned}$$

Es bleibt zu zeigen, daß $A_{.B'}$ vollen Rang hat. Angenommen dies wäre nicht der Fall und $\tilde{\mu} \neq 0$ mit $A_{.B'} \tilde{\mu} = 0$. Sei dann μ definiert durch $\mu_{B'} = \tilde{\mu}$ und $\mu_{N'} = 0$. Dann gilt $\kappa := \mu - \mu_j d \neq 0$, denn $\kappa_i = -(A_{.B}^{-1} A_{.j})_i < 0$ und $\kappa_j = 0$. Somit definieren die Nichtnulleinträge von κ ein nichttriviales Element im Kern von $A_{.B}$ im Widerspruch dazu, daß B eine Basis ist. $\square$

Bevor wir den Simplexalgorithmus formulieren, überlegen wir zunächst, was passiert, wenn die Menge, über die wir durch diese Minimumbildung das basisverlassende Element finden wollen leer ist. Anschaulich heißt das, daß nie mehr eine Bedingung $x \geq 0$ greift, wir also beliebig viel weiteren Profit einstreichen können.

Lemma 3.4.9 *Ist B eine zulässige Basis mit Basislösung x, $c_j - c_B^\top A_{.B}^{-1} A_{.j} > 0$ und $(A_{.B}^{-1} A_{.j})_i \leq 0$ für alle i, so ist das lineare Programm unbeschränkt.*

Beweis. Sei $d \in \mathbb{R}^n$ mit $d_j = 1, d_B = -A_{.B}^{-1} A_{.j}$ Dann ist für $\lambda \geq 0 : x + \lambda d \geq 0, A(x + \lambda d) = b$ und $c^\top(x + \lambda d) = c^\top x + \lambda(c_j - c_B^\top A_{.B}^{-1} A_{.j}) \stackrel{\lambda \to \infty}{\to} \infty$. $\square$

Wir haben jetzt fast alle Fakten beisammen, um den Simplexalgorithmus skizzieren und beweisen zu können. Wir beschränken die Eingabedaten allerdings zunächst noch auf den günstigen Fall, daß A vollen Zeilenrang hat und eine zulässige *Startbasis* B gegeben ist. Wie man den allgemeinen Fall darauf reduziert, werden wir später vorstellen.

Schematische Skizze des Simplexalgorithmus:

Eingabedaten: $A \in \mathbb{R}^{m\times n}$ mit vollem Zeilenrang, $b \in \mathbb{R}^m$, $b \geq 0$, zulässige Basis B, $c \in \mathbb{R}^n$.

Solange $c^\top - c_B^\top A_{.B}^{-1} A \not\leq 0$:

Spaltenwahl: Wähle j mit $c_j - c_B^\top A_{.B}^{-1} A_{.j} > 0$.

Zeilenwahl: Berechne $i \in \operatorname{argmin}\{\frac{(A_{.B}^{-1}b)_i}{(A_{.B}^{-1}A_{.j})_i} \mid (A_{.B}^{-1}A_{.j})_i > 0\}$. Falls diese Menge leer ist. STOP. Das Programm ist unbeschränkt.

Basiswechsel: Setze $B = B \setminus \{i\} \cup \{j\}$.

oder als Pythoncode

```
def simplex(c,A,b,B):
  m=A.shape[0]
  n=A.shape[1]
  C=concatenate((A,b),1)      # 1 heisst nebeneinander
  c=concatenate((c,[[0]]),1)
  D=concatenate((c,C),0)      # 0 heisst untereinander
  initialize_basis(D,B)
  redcostneg=0
  while redcostneg==0:
    column=choosecolumn(D)
    if column<=n:
      row=chooserow(D,column)
      if row==0:
        print "Programm ist unbeschraenkt"
        redcostneg=1
      else:
        gaussjordanelim(D,row,column)
        B[row-1]=column
    else:
      redcostneg=1
  return D,B
```

3.5 Tableauform des Simplexalgorithmus

Alle die oben angeführten Operationen lassen sich mit Hilfe des Gauß-Jordan Eliminationsschrittes sehr leicht in einer Tableauform formalisieren. Wir nehmen wieder an, daß A vollen Zeilenrang hat und eine zulässige *Startbasis* B gegeben ist. Dann lautet das Tableau zur Basis B:

$$\begin{array}{c|c} c^\top - c_B^\top A_{.B}^{-1}A & -c_B^\top A_{.B}^{-1}b \\ \hline A_{.B}^{-1}A & A_{.B}^{-1}b \end{array}$$

Der Basiswechsel von B nach $B \setminus \{i\} \cup \{j\}$ wird nun mittels eines Gauß-Jordan Eliminationsschrittes mit Pivotelement $(A_{.B}^{-1}A)_{ij}$ durchgeführt. Als Beispiel wollen wir die Düngemittelfabrik durchrechnen. Startbasis sind hier die drei Schlupfvariablen y_1, y_2, y_3.

x_1	x_2	$\underline{y_1}$	$\underline{y_2}$	$\underline{y_3}$	$-ZF$
30	20	0	0	0	0
2	1	1	0	0	1500
1	1	0	1	0	1200
$\boxed{1}$	0	0	0	1	500

$\underline{x_1}$	x_2	$\underline{y_1}$	$\underline{y_2}$	y_3	ZF
0	20	0	0	−30	−15000
0	$\boxed{1}$	1	0	−2	500
0	1	0	1	−1	700
1	0	0	0	1	500

$\underline{x_1}$	$\underline{x_2}$	y_1	$\underline{y_2}$	y_3	$-ZF$
0	0	−2	0	10	−25000
0	1	1	0	−2	500
0	0	−1	1	$\boxed{1}$	200
1	0	0	0	1	500

$\underline{x_1}$	$\underline{x_2}$	y_1	y_2	$\underline{y_3}$	ZF
0	0	−10	−10	0	−27000
0	1	−1	2	0	900
0	0	−1	1	1	200
1	0	1	−1	0	300

Bemerkung 3.5.1 *In einigen Lehrbüchern wird das Tableau für Minimierungsprobleme eingeführt, dann wählt man Spalten mit negativen reduzierten Kosten, bis alle positiv sind. Auch wird oft die Zielfunktion wird unter die Matrix geschrieben.*

3.6 Pivotwahl, Entartung, Endlichkeit

Die oben angegebene Skizze ist strenggenommen noch kein Algorithmus, da, insbesondere bei der Spaltenwahl noch viel Freiheit herrscht. Auch bei der Zeilenwahl gibt es bei nicht eindeutigem Minimum Zweideutigkeiten.

Eine feste Vorschrift der Auswahl der Pivotspalte bezeichnet man als *Pivotregel*. Wir wollen hier drei erwähnen.

Steilster Anstieg: Wähle die Spalte mit den größten reduzierten Kosten.

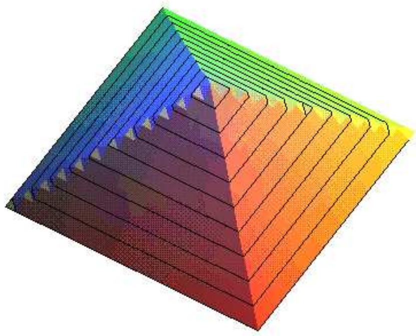
Größter Fortschritt: Wähle die Spalte deren Aufnahme in die Basis die größte Verbesserung in der Zielfunktion verspricht.

Bland's rule: Wähle stets die Variable mit dem kleinsten Index (sowohl bei Spalten als auch bei Zeilenwahl).

Die erste Regel ist billig zu implementieren mit zufriedenstellendem Ergebnis, die zweite Regel ist numerisch aufwendig und die dritte ist praktisch fast bedeutungslos hat aber ein wesentliches theoretisches Feature. Das liegt daran, daß ein Basiswechsel nicht notwendig zu einem Wechsel der Ecke führt.

Definition 3.6.1 *Eine Ecke x eines LP in Standardform heißt* entartet*, wenn es mindestens zwei Basen $B \neq B'$ gibt mit $x_B = A_{.B}^{-1}b$ und $x_{B'} = A_{.B'}^{-1}b$, d.h. die zugehörigen Basislösungen zu B und B' sind gleich. Ebenso nennen wir einen Pivotschritt* entartet*, der die Basislösung nicht verändert.*

Geometrisch liegt eine entartete Ecke auf mehr Hyperebenen „als nötig“. Im zweidimensionalen kann man Entartung nur durch überflüssige Ungleichungen erzeugen. Im dreidimensionalen ist die Spitze einer Viereckspyramide entartet.



Proposition 3.6.2 *Ist x eine entartete Ecke, so hat x weniger als m Nichtnulleinträge.*

Beweis. x kann nur auf $B \cap B'$ von Null verschieden sein. □

Proposition 3.6.3 *a) Ein Pivot von der Basis B nach B' mit Pivotelement a_{ij} ist genau dann entartet, wenn $(A_{.B}^{-1}b)_i = 0 = (A_{.B'}^{-1}b)_i$ ist.*

b) Ist ein Pivot von Basis B zu B' nicht entartet, so ist

$$c^\top x' = c_{B'}^\top A_{.B'}^{-1} b > c_B^\top A_{.B}^{-1} b = c^\top x.$$

Beweis. Wir betrachten $A_{.B'}^{-1}b - A_{.B}^{-1}b = (\eta - I_n)A_{.B}^{-1}b$ mit einer η Matrix wie in Abschnitt 2.3 angegeben, die den Gauß-Jordan-Schritt beschreibt. Die Matrix $(\eta - I_n)$ hat genau eine von Null verschiedene Spalte, nämlich die i-te. Folglich ist $A_{.B'}^{-1}b = A_{.B}^{-1}b \leftrightarrow (A_{.B}^{-1}b)_i = 0$. Die zweite Behauptung folgt nun mit Lemma 3.4.7. □

Wir werden in den Übungen ein Beispiel kennenlernen, bei dem man unter Anwendung der steilste-Anstieg-Regel in einer degenerierten Ecke hängenbleibt und *zykelt*, d.h. es gibt eine Folge von Basen $B_1, \ldots, B_k = B_1$, so daß der Simplexalgorithmus unter Anwendung dieser Pivotregel von B_i nach B_{i+1} wechselt.

Kleine Rundungsfehler heben Degeneration auf. Es war lange Zeit Lehrmeinung, daß in der Praxis Zykeln nicht auftritt. In letzter Zeit wird häufen sich aber Berichte über Zykeln bei sehr großen, strukturierten Probleminstanzen.

Satz 3.6.4 *Bei Anwendung von Bland's rule zykelt das Simplexverfahren nicht.*

Beweis. Sei $B_1, \ldots, B_k = B_1$ ein Zykel und $B_{i+1} = B_i \setminus \{e_i\} \cup \{f_i\}$ und

$$J = \bigcup_{i=1}^{k} \{e_i\} \stackrel{!}{=} \bigcup_{i=1}^{k} \{f_i\}.$$

Sei $t = e_i = f_j$ der größte Index in J. Wenn t in die Basis B_j aufgenommen wird, müssen wegen Bland's rule die reduzierten Kosten aller Elemente in J nicht-positiv sein, speziell gilt dies für das Element $s := f_i$, welches in die Basis aufgenommen wird, wenn t die Basis verläßt. Wir haben also

$$\begin{aligned} c_s - c_{B_j}^\top A_{.B_j}^{-1} A_{.s} &\leq 0 \\ c_s - c_{B_i}^\top A_{.B_i}^{-1} A_{.s} &> 0 \end{aligned}$$

und somit

$$c_{B_i}^\top A_{.B_i}^{-1} A_{.s} - c_{B_j}^\top A_{.B_j}^{-1} A_{.s} < 0. \tag{3.1}$$

Bei der Wahl des basisverlassenden Elementes kommen jeweils die Elemente k mit $(A_{.B}^{-1}b)_k = 0$ in Betracht. Wenn t die Basis verläßt, muß also wegen Anwendung von Bland's rule $(A_{.B_i}^{-1}A.s)_k \leq 0$ für alle $k \in B_i \cap J \setminus \{t\}$ sein. Da ferner $B_i \setminus J \subseteq B_j$ ist erhalten wir insgesamt

$$
\begin{aligned}
c_{B_i}^\top A_{.B_i}^{-1} A_{.s} - c_{B_j}^\top A_{.B_j}^{-1} A_{.s} &= (c_{B_i}^\top - c_{B_j}^\top A_{.B_j}^{-1} A_{.B_i}) A_{.B_i}^{-1} A_{.s} \\
&= \underbrace{(c_t - c_{B_j}^\top A_{.B_j}^{-1} A_t)}_{>0} \underbrace{(A_{.B_i}^{-1} A_{.s})_t}_{>0} \\
&+ \underbrace{(c_{J\cap B_i\setminus\{t\}}^\top - c_{B_j}^\top A_{.B_j}^{-1} A_{.J\cap B_i\setminus\{t\}})}_{\leq 0} \underbrace{(A_{.B_i}^{-1} A_{.s})_{J\cap B_i\setminus\{t\}}}_{\leq 0} \\
&+ \underbrace{(c_{B_i\cap B_j}^\top - c_{B_j}^\top A_{.B_j}^{-1} A_{.B_i\cap B_j})}_{=0} (A_{.B_i} A_{.s})_{B_i\cap B_j} \\
&> 0
\end{aligned}
$$

im Widerspruch zu (3.1). □

Damit können wir nun Endlichkeit und Korrektheit des Simplexverfahrens beweisen.

Satz 3.6.5 (Korrektheit des Simplexverfahrens) *a) Bei Anwendung von Bland's rule stoppt das Simplexverfahren nach endlich vielen Schritten.*

b) Wenn das Simplexverfahren stoppt, und $c^\top - c_B^\top A_{.B}^{-1} A \not\leq 0$ ist, so ist das Problem unbeschränkt, andernfalls ist B eine optimale Basis.

Beweis. Es gibt nur endlich viele Basen, nämlich $\binom{m}{n}$ viele. Wegen Satz 3.6.4 und Lemma 3.4.7 wird jede Basis höchstens einmal besucht. Der Rest folgt aus Proposition 3.4.4 und Lemma 3.4.9.

3.7 Bemerkungen zur Numerik

Es ist nicht-trivial, einen numerisch stabilen LP-Code zu schreiben. Es gibt im Netz eine lange Liste von Standardbeispielen, die gängige numerische Schwächen von Codes testen. Wir wollen hier mit ein paar Bemerkungen Probleme und Möglichkeiten der LP-Numerik anreißen.

Bei einer Implementierung wird man natürlich nicht das ganze Tableau abspeichern. Für die Basisspalten genügt es, ihre Nummern zu kennen, da sie nur die Einheitsmatrix enthalten (genauer sollte man sich die zugehörige Permutation merken). Das ergibt allerdings zusätzlichen Buchhaltungsaufwand, den wir uns hier sparen.

In der Praxis hat man es häufig mit dünnbesetzten Matrizen zu tun, das sind solche, bei denen die meisten Einträge Null sind. Hier setzt man (wie auch schon in der numerischen linearen Algebra) "sparse matrix" Techniken ein.

Üblicherweise wird man auch weder stets neu $A_{.B}^{-1}$ noch das Tableau berechnen. Beim Basiswechsel von B nach B' geht nach Abschnitt 2.3 $A_{.B'}^{-1}$ durch Multiplikation mit einer η-Matrix aus $A_{.B'}^{-1}$ hervor. Außerdem benötigt man zur Auswahl der nächsten Basis nur die reduzierten Kosten und die Daten der Pivotspalte. Also genügt es z.B., solange Einträge der reduzierten Kosten zu berechnen, bis man einen positiven Eintrag gefunden hat. Dann berechnet man die Einträge in der Pivotspalte zur Wahl der Pivotzeile, ändert die Basis und die Inverse. Es empfiehlt sich, wegen der Fehlerfortpflanzung bei der Matrixmultiplikation hin und wieder eine volle Matrixinversion durchzuführen.

3.8 Die Zweiphasenmethode

Wir wollen nun zu einem beliebigen linearen Programm in Standardform ein Hilfsproblem konstruieren, das die Voraussetzungen zur Anwendung des Simplexverfahrens erfüllt und dessen Optimallösung eine zulässige Basis des Ausgangsproblems bildet. Sei also

$$\begin{array}{lll} \max & & c^\top x \\ (P) & \text{unter} & Ax = b \\ & & x \geq 0 \end{array}$$

ein lineares Programm in Standardform mit einer beliebigen Matrix $A \in \mathbb{R}^{m\times n}$.

Das zugehörige Hilfsproblem ist dann

$$\begin{array}{lll} & \max & -\sum_{i=1}^m y_i \\ (H) & \text{unter} & Ax + y = b \\ & & x, y \geq 0 \end{array}$$

mit Startbasis $B = \{n+1, \ldots, m+n\}$. Die Startbasis besteht also nur aus den *künstlichen Schlupfvariablen*, deren Summe wir minimieren wollen.

Proposition 3.8.1 *a)* (A, I) *hat vollen Rang und B ist zulässige Startbasis.*

b) Ist der optimale Zielfunktionswert von (H) *ungleich* 0*, so ist* (P) *unzulässig.*

c) Ist $(x, 0)$ *eine optimale Basislösung zur Basis* $B' \subseteq \{1, \ldots, n\}$ *für* (H)*, so ist x zulässig Basislösung zur Basis B' für* (P) *und A hat vollen Rang.*

Beweis. Die erste Behauptung ist wegen $b \geq 0$ trivial. Wenn x zulässig für (P) ist, so ist $(x, 0)$ zulässig für (H). Da der Zielfunktionswert von $(x, 0)$ bzgl. (H) 0 ist und $y \geq 0$ gilt, folgt der Rest. □

Wenn man nun nach Lösen des Hilfsproblems (wir sprechen von Phase I) einen Zielfunktionswert 0 erreicht hat aber noch eine künstliche Schlupfvariable y_i in der Basis ist, so können wir diese gegen ein beliebiges Element $x_k, k \in \{1, \ldots, n\}$ mit $(A^{-1}_{.B'}A_{.k})_i \neq 0$ austauschen, denn mit $B'' := B' \setminus \{n+i\} \cup \{k\}$ haben wir, wenn wir $\tilde{A} := A^{-1}_{.B'}A$

$$A^{-1}_{.B''} = \eta A^{-1}_{.B'}$$

mit

$$\eta = \begin{array}{cc} & \begin{array}{ccccccc} & & & i & & \end{array} \\ \begin{array}{c} \\ \\ \\ i \\ \\ \\ \end{array} & \begin{pmatrix} 1 & 0 & \ldots & -\frac{\tilde{a}_{1k}}{\tilde{a}_{ik}} & \ldots & 0 \\ 0 & 1 & \ldots & -\frac{\tilde{a}_{2k}}{\tilde{a}_{ik}} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ 0 & 0 & \ldots & \frac{1}{\tilde{a}_{ik}} & \ldots & 0 \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & -\frac{\tilde{a}_{mk}}{\tilde{a}_{ik}} & \ldots & 1 \end{pmatrix} \end{array}.$$

Nun ist $(A^{-1}_{.B'}b)_i = 0$, also $A^{-1}_{.B''}b = \eta A^{-1}_{.B'}b = A^{-1}_{.B'}b$.

Gibt es kein solches x_k, so ist wegen $(A^{-1}_{.B'})_{i.}(A, b) = (0, \ldots, 0)$ die Zeile redundant, kann also gestrichen werden. (Genauer, wenn y_j das i-te Basiselement ist, so ist $(A^{-1}_{.B'})_{i.}e_j = e_i$ also $(A^{-1}_{.B'})_{ij} = 1$, also ist die j-te Zeile von (A, b) eine Linearkombination der übrigen Zeile. Die Matrix hatte also nicht vollen Rang. Da nur Äquivalenzumformungen durchgeführt wurden, hat die transformierte Matrix, aus der die Nullzeile gestrichen wurde, den gleichen Rang, definiert also ein äquivalentes Gleichungssystem).

Iteriert man diese Vorgehensweise, so endet man mit einer regulären Matrix $\hat{A}$ und einer Basis, die keine künstliche Variable mehr enthält. Nun kann man die künstlichen Variablen mit ihren Spalten streichen und den Simplexalgorithmus bzgl. der Originalzielfunktion starten.

Wir erhalten also folgenden schematischen Ablauf:

Eingabedaten: $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m, b \geq 0$, partielle zulässige Basis B (eventuell leer), $c \in \mathbb{R}^n$.

Phase 1a: Ergänze B durch künstliche Schlupfvariablen zu voller Basis, und minimiere die Summe der künstlichen Schlupfvariablen mit dem Simplexalgorithmus.

Phase 1b: Enthält die optimale Basis noch künstliche Variablen, pivotiere sie hinaus. Ist dies nicht möglich, streiche die zugehörige Zeile, Spalte und das Basiselement.

Phase 1c: Streiche alle Spalten künstlicher Schlupfvariablen.

Phase 2: Optimiere das Originalproblem mit dem Simplexverfahren.

Beispiel 3.8.2 *Wir betrachten folgendes Lineares Programm*

$$\begin{array}{rrcrcrcrcrcl} \max & -3x_1 & + & 2x_2 & + & 2x_3 & - & 4x_4 & - & 2x_5 & & \\ \textit{unter} & -2x_1 & + & x_2 & + & x_3 & & & + & x_5 & = & 3 \\ & -2x_1 & + & x_2 & + & & & x_4 & & & = & 2 \\ & x_1 & + & x_2 & & & + & x_4 & & & = & 7 \end{array}$$

Als erste Basisvariable können wir x_3 oder x_5 wählen. Für die zweite und dritte Ungleichung brauchen wir künstliche Schlupfvariablen. *Das Hilfsproblem lautet dann:*

$$\begin{array}{rrrrl} \max & & -y_1 - y_2 & & \\ \textit{unter} & -2x_1 + x_2 + x_3 + x_5 & & & = 3 \\ & -2x_1 + x_2 + x_4 & +y_1 & & = 2 \\ & x_1 + x_2 + x_4 & & + \;\; y_2 & = 7 \end{array}$$

Wir erhalten also folgende Tableaus:

0	0	0	0	0	−1	−1	0
−3	2	2	−4	−2	0	0	0
−2	1	1	0	1	0	0	3
−2	1	0	1	0	1	0	2
1	1	0	1	0	0	1	7

−1	2	0	2	0	0	0	9
−7	4	4	−4	0	0	0	6
−2	1	1	0	1	0	0	3
−2	$\boxed{1}$	0	1	0	1	0	2
1	1	0	1	0	0	1	7

3	0	0	0	0	−2	0	5
1	0	4	−8	0	−4	0	−2
0	0	1	−1	1	−1	0	1
−2	1	0	1	0	1	0	2
$\boxed{3}$	0	0	0	0	−1	1	5

0	0	0	0	0	−1	−1	0
0	0	4	−8	0	$\frac{-11}{3}$	$\frac{-1}{3}$	$\frac{-11}{3}$
0	0	$\boxed{1}$	−1	1	−1	0	1
0	1	0	1	0	$\frac{1}{3}$	$\frac{2}{3}$	$\frac{16}{3}$
1	0	0	0	0	$\frac{-1}{3}$	$\frac{1}{3}$	$\frac{5}{3}$

Beim letzten Tableau ist die künstliche Zielfunktion Null und man hat mit $(\frac{5}{3}, \frac{16}{3}, 1, 0, 0)$ eine zulässige Basislösung zur Basis $B = \{1, 2, 3\}$ gefunden. Nun kann man die *künstliche Zielfunktion sowie die künstlichen Variablen streichen und erhält das optimale Tableau*

$$\begin{array}{ccccc|c} 0 & 0 & 0 & -4 & -4 & \frac{-23}{3} \\ \hline 0 & 0 & 1 & -1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & \frac{16}{3} \\ 1 & 0 & 0 & 0 & 0 & \frac{5}{3}. \end{array}$$

Wir haben nun auch noch insgesamt folgenden Satz gezeigt:

Satz 3.8.3 *Wenn ein lineares Programm zulässig und beschränkt ist, so wird das Optimum an einer Ecke angenommen.*

Beweis. Ist das Programm zulässig, so findet man in Phase 1 eine zulässige Ecke. In Phase 2 terminiert der Simplexalgorithmus, da das Programm beschränkt ist in einer optimalen Ecke. □

3.9 Sensitivitätsanalyse

Anhand des optimalen Tableaus kann man verschieden Informationen über Änderungen der Lösung bei Änderung der Eingangsdaten herleiten. Wir wollen dies anhand zweier Beispiele diskutieren.

Beispiel 3.9.1 *Angenommen in dem Beispiel der Düngemittelfabrik (3.1.4) erfände ein Chemiker eine neue Formel für einen Dünger, der als Rohstoffvektor* $A_{.3} = (3,2,2)^\top$ *pro Tonne benötigt. Wie teuer muß das Unternehmen das Produkt absetzen können, damit die Produktion nach dieser Formel sich lohnt? Anstatt das lineare Programm von vorne zu lösen, können wir auch einfach ausnutzen, daß die reduzierten Kosten bzgl. der gegenwärtigen Basis* $c_3 - c_B^\top A_{.B}^{-1} A_{.3}$ *sind. Damit eine Aufnahme dieses Vektors in die Basis eine echte Verbesserung bringt, muß also*

$$\begin{aligned} c_3 &> (10,10,0)\begin{pmatrix} 3 \\ 2 \\ 2 \end{pmatrix} \\ &= 50 \end{aligned}$$

sein. Anhand der reduzierten Kosten kann man also Auswirkungen bei Einführung einer neuen Variablen studieren. Außerdem ist offensichtlich die Optimallösung bei Änderungen der Kostenfunktion in der Nichtbasis je sensibler, je geringer die reduzierten Kosten sind.

Im zweiten Beispiel wollen wir die Sensitivität gegenüber Änderungen der rechten Seite studieren. Offensichtlich bleibt, da sich die reduzierten nicht ändern, eine Basis optimal gegen Änderungen der rechten Seite um einen Vektor ε, solange $A_{.B}^{-1}(b+\varepsilon) \geq 0$ ist. Aus dieser Beziehung kann man für die rechte Seite obere und untere Schranken ausrechnen. Standardpakete für die lineare Programmierung bieten eine Sensitivitätsanalyse an.

Beispiel 3.9.2 *Betrachten wir das Beispiel der Ölraffinerie so liefert CPLEX aber sogar die Schranken, in denen sich die rechte Seite ändern darf, ohne daß der Zielfunktionswert sich wesentlich ändert.*

```
CPLEX> display sensitivity rhs
Display RHS sensitivity for which constraint(s): -
                                                RHS Sensitivity Ranges
Constraint Name      Dual Price            Down         Current              Up
RB1                     -2.8980       3457.4074       4000.0000       6344.1176
RB2                     -8.2560       3422.2222       5050.0000      27311.1111
RB3                     -7.1360       3437.5000       7100.0000      10810.1852
RB4                     -8.0640       1453.5714       4300.0000       7962.5000
O85                      0.4160     -14650.0000            zero      39850.0000
O90                      0.4293            zero            zero      25617.8571
O95                      0.4160     -14650.0000            zero      11385.7143
NF1                        zero       -infinity      15000.0000      16465.0000
NF2                        zero       3985.0000      10000.0000       +infinity
```

Setzen wir die rechte Seite von O85 allerdings nur auf .000018, so scheint sich die Lösung bereits zu ändern. Allerdings haben wir es hier offensichtlich mit Entartung zu tuen, so daß wir nicht auf eine Änderung der Basis schließen dürfen.

```
CPLEX> display problem all
Minimize
 ZIEL: - 9.97 X11 - 11.93 X12 - 14.13 X13 - 7.84 X21 - 9.8 X22 - 12 X23
 - 4.64 X31 - 6.6 X32 - 8.8 X33 - 2.24 X41 - 4.2 X42 - 6.4 X43
Subject To
 RB1: X11 + X12 + X13 <= 4000
 RB2: X21 + X22 + X23 <= 5050
 RB3: X31 + X32 + X33 <= 7100
 RB4: X41 + X42 + X43 <= 4300
 O85: - 17 X11 + X21 + 6 X31 + 14 X41 >= 0.000018
 O90: - 22 X12 - 4 X22 + X32 + 9 X42 >= 0
 O95: - 27 X13 - 9 X23 - 4 X33 + 4 X43 >= 0
 NF1: X11 + X21 + X31 + X41 >= 15000
 NF2: X13 + X23 + X33 + X43 <= 10000
Bounds
 All variables are >= 0.
CPLEX> optimize
```

```
Using devex.

Primal - Optimal:  Objective =   -1.3862559999e+05
Solution time =    0.00 sec.  Iterations = 0 (0)

CPLEX> display solution variables -
Variable Name           Solution Value
X11                        3485.806452
X13                         514.193548
X21                        5050.000000
X31                        7100.000000
X41                         829.193550
X43                        3470.806450
All other variables in the range 1-12 are zero.
```

Kapitel 4

Nichtlineare Optimierung

Wir wollen nun etwas allgemeinere Optimierungsprobleme betrachten. Das allgemeine Problem lautete

$$\min_{x \in S} f(x).$$

Bei einer linearen Zielfunktion ist diese Fragestellung nur sinnvoll, wenn der Bereich S beschränkt ist, bei nicht-linearen Zielfunktionen ist oft schon die unbeschränkte Optimierungsaufgabe mit $S = \mathbb{R}^n$ schwer.

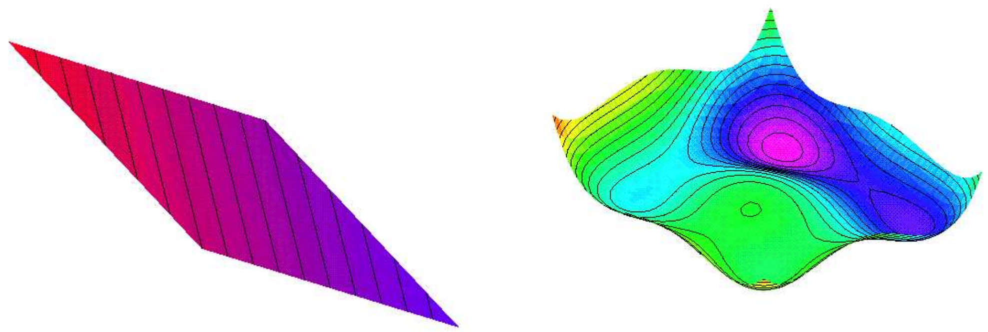


Abbildung 4.1: Lineare Funktion gegen $x^4 + y^4 - 5x^2 - 4y^2 + 5x + 2y - 1.5$

Bei der Minimierung nicht-linearen Funktionen spielt die folgende einfache Strategie eine zentrale Rolle. Ausgehend von einem Punkt x_i suche eine Abstiegsrichtung und gehe in dieser Richtung bestmöglich zu x_{i+1}. Iteriere bis es keine Abstiegsrichtung mehr gibt. Im allgemeinen findet man so kein *globales* Minimum, sondern nur *lokale* (oder auch relative) Minima. Aber auch die Theorie macht oftmals nur Aussagen über *lokale* Minima.

Definition 4.0.3 *Sei $S \subseteq \mathbb{R}^n, f : S \to \mathbb{R}$ und $x^* \in S$. Dann heißt x^* ein* lokales *Minimum von f über S (oder auch* relatives *Minimum), wenn es ein $\varepsilon > 0$ gibt mit $\forall x \in S \cap U_\varepsilon(x^*) : f(x) \geq f(x^*)$. Gilt sogar $\forall x \in S \cap U_\varepsilon(x^*) : f(x) > f(x^*)$, so ist x^* ein* striktes lokales *Minimum.*

Falls $\forall x \in S : f(x) \geq f(x^*)$, so heißt x^* ein globales *Minimum und analog zum Vorigen sprechen wir von einem* strikten globalen *Minimum falls die letzte Ungleichung stets strikt ist.*

4.1 Wiederholung aus der mehrdimensionalen Differentialrechnung

4.1.1 Kurven

Wie bereits erwähnt haben wir im wesentlichen nur Mittel zur Bestimmung lokaler Minima zur Hand. Die Werkzeuge dafür liefert die Analysis. Eines der zentralen Anliegen der Analysis ist es, Funktionen lokal durch lineare Funktionen (und evtl. Terme höherer Ordnung) zu approximieren. Dafür muß die Funktion aber lokal „hinreichend dicht“definiert sein. Oftmals betreibt man deshalb Analysis nur auf *offenen Mengen U*, das sind Mengen, bei denen zu jedem $x \in U$ ein $\varepsilon > 0$ mit $U_\varepsilon(x) \subseteq U$ existiert. Unsere Mengen S werden im allgemeinen nicht offen sein, aber wir werden stets annehmen, daß die Kostenfunktion auf einer offenen Menge U definiert ist, die S enthält.

Um die Optimierungsstrategie aus der Einleitung dieses Kapitels präzisieren zu können, wiederholen wir nun Wege und Richtungen von Wegen im $\mathbb{R}^n$.

Definition 4.1.1 *Sei $I \subseteq \mathbb{R}$ ein Intervall. Eine* Kurve *oder ein* Weg *im $\mathbb{R}^n$ ist eine stetige Abbildung*

$$\begin{array}{rll} c: & I & \to \mathbb{R}^n \\ & t & \mapsto \quad c(t) = (c_1(t), \ldots, c_n(t))^\top, \end{array}$$

d.h. alle Komponentenfunktionen sind stetig. Analog heißt eine Kurve $k-$fach (stetig) differenzierbar, *wenn alle Komponentenfunktionen $k-$fach (stetig) differenzierbar sind, für $k \in \mathbb{N}$.*

Ist $t_0 \in I$, so nennen wir $c'(t_0) := (c_1'(t_0), \ldots, c_n'(t_0))$ den Tangentialvektor an *c in t_0.*

Beispiel 4.1.2 *Wir betrachten die übliche Parametrisierung des Einheitskreises*

$$\begin{array}{rccl} c: & [0,2\pi] & \to & \mathbb{R}^2 \\ & t & \mapsto & c(t) = (\sin(t), \cos(t)), \end{array}$$

in $t_0 = \pi$. *Dann ist* $c(t_0) = (0, -1)$ *und* $c'(t_0) = (\cos(\pi), -\sin(\pi)) = (-1, 0)$.

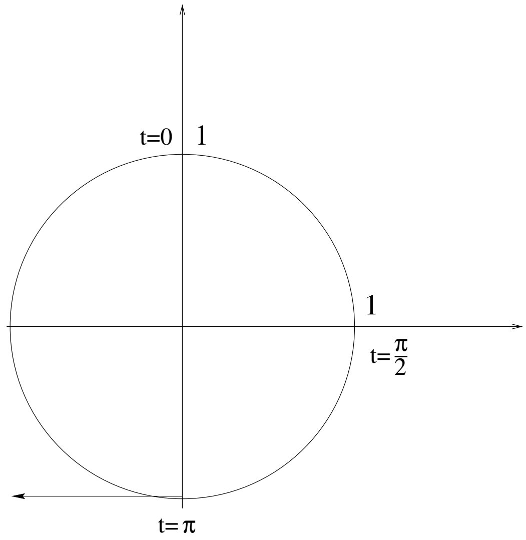


Abbildung 4.2: Eine Tangente an den Kreis

Proposition 4.1.3 *Sei* $S \subseteq \mathbb{R}^n, f : S \to \mathbb{R}$ *und* $x^* \in S$. *Ist* x^* *ein lokales Minimum (striktes lokales Minimum) von* f*, so gilt für jeden Weg* $c : [0, \varepsilon] \to S$ *mit* $c(0) = x^*$ *und* $c'(0) \neq 0$*:*

$$\exists \delta \leq \varepsilon \, \forall 0 < \alpha \leq \delta : (f \circ c)(\alpha) := f(c(\alpha)) \geq f(x^*) \text{ (bzw. } (f(c(\alpha)) > f(x^*)).$$

Beweis. Ist x ein (striktes) lokales Minimum, so gibt es ein η mit $f(x) \overset{>}{\geq} f(x^*)$ für alle $x \in U_\eta(x^*)$. Sei nun $c : [0, \varepsilon] \to S$ ein Weg mit $c(0) = x^*$ und $c'(0) \neq 0$. Da c stetig ist, gibt es ein $\delta > 0$ mit $0 < \alpha \leq \delta$ impliziert $c(t) \in U_\varepsilon(x^*)$. □

4.1.2 Partielle Ableitungen

Im letzten Abschnitt haben wir Abbildungen $\mathbb{R} \to \mathbb{R}^n$ untersucht. In der Optimierung haben wir es bei der Zielfunktion üblicherweise mit Abbildungen $\mathbb{R}^n \to \mathbb{R}$ zu tun (vgl. Abbildung 4.1). Dafür definieren wir Richtungsableitungen entlang eines Weges. Eine besondere Rolle spielen hierbei die Richtungen der Koordinatenachsen.

Definition 4.1.4 *Sei $I \subseteq \mathbb{R}$ ein Intervall, $t_0 \in I$, $S \subseteq \mathbb{R}^n$, $c : I \to \mathbb{R}^n$ eine stetig differenzierbare Kurve und $f : S \to \mathbb{R}$ mit $p = c(t_0) \in S$ und $c'(t_0) = d$. Existiert dann $(f \circ c)'(t_0)$, so heißt diese Zahl die* Richtungsableitung $\frac{\partial f}{\partial d}$ von f in Richtung d in p. *Ist $c'(t_0) = e_i$, so heißt*

$$\frac{\partial f}{\partial x_i}(p) := (f \circ c)'(t_0)$$

die i-te partielle Ableitung von f. *Der* Gradient $\nabla f(p)$ von f in p *ist der Zeilenvektor der partiellen Ableitungen* $(\frac{\partial f}{\partial x_1}(p), \ldots, \frac{\partial f}{\partial x_n}(p))$.

Ist $h : U \to \mathbb{R}^k$ eine vektorwertige Ableitung, so bezeichnen wir mit Jh die Jacobische *d.i. die Matrix der partiellen Ableitungen*

$$Jh(x) := \begin{pmatrix} \frac{\partial h_1}{\partial x_1} & \cdots & \frac{\partial h_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_n}{\partial x_1} & \cdots & \frac{\partial h_n}{\partial x_n} \end{pmatrix}.$$

Existieren alle partiellen Ableitungen einer reellwertigen Funktion f und sind stetig, so sagen wir f ist stetig differenzierbar. *Wir können nun die Definition der Jacobischen auch auf den Gradienten anwenden. Zunächst definieren wir dafür die* zweiten partiellen Ableitungen

$$\frac{\partial^2 f}{\partial x_i \partial x_j} := \frac{\partial \left(\frac{\partial f}{\partial x_i}\right)}{\partial x_j} =: \frac{\partial}{\partial x_j}\left(\frac{\partial f}{\partial x_i}\right).$$

Für $\frac{\partial^2 f}{\partial x_i \partial x_i}$ schreiben wir auch kürzer $\frac{\partial^2 f}{\partial x_i^2}$. Existieren alle zweiten partiellen Ableitungen und sind stetig, so heißt f zweimal stetig differenzierbar. Die Jacobische des Gradienten nennen wir Hessematrix $\nabla^2 f(x)$. *Die Hessematrix ist also die Matrix der zweiten partiellen Ableitungen*

$$\nabla^2 f(x) := \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}.$$

Ist $U \subseteq \mathbb{R}^n$ und f k-fach stetig differenzierbar in U, so schreiben wir $f \in C^k(U)$.

Bemerkung 4.1.5 *Genau genommen ist der von uns hier gewählte Zugang nicht ganz sauber, da nicht klar ist, daß die oben eingeführte Richtungsableitung unabhängig von der Wahl des Weges c ist. Wir berufen uns dafür auf Bekanntes aus der Mathematik für Chemiker und Wirtschaftsinformatiker, da wir zur Herleitung der Zuasammenhänge hier keine Zeit haben.*

Proposition 4.1.6 *Ist f zweimal stetig differenzierbar, so ist*

$$\frac{\partial f}{\partial d}(p) = \nabla f(p) d.$$

Beweis. Anwendung der Kettenregel aus Mathematik für Chemiker und Wirtschaftsinformatiker Kapitel IV. □

Beispiel 4.1.7 *Wir betrachten die Funktion $f(x, y) = x^4 + y^4 - 5x^2 - 4y^2 + 5x + 2y - 1.5$. Dann ist $\nabla f(x, y) = (4x^3 - 10x + 5, 4y^3 - 8y + 2)$ und*

$$\nabla^2 f(x) := \begin{pmatrix} 12x^2 - 10 & 0 \\ 0 & 12y^2 - 8 \end{pmatrix}.$$

Wir wollen diese Wiederholung abschließen mit einer mehrdimensionalen Konsequenz aus dem Satz von Taylor.

Satz 4.1.8 *Sei $p \in S \subseteq \mathbb{R}^n$ und $f : S \to \mathbb{R}$ zweimal stetig differenzierbar. Sei c ein zweimal stetig differenzierbarer Weg mit $c(t_0) = p$ und $c'(t_0) = d$. Dann ist*

$$(f \circ c)(t) = f(p) + \nabla f(p) d(t - t_0) + \frac{1}{2} \nabla f(p) c''(t_0)(t - t_0)^2 + \frac{1}{2} d^\top \nabla^2 f(p) d(t - t_0)^2 + o((t - t_0)^2).$$

Beweis. Nach dem Satz von Taylor für Variablen einer Veränderlichen ist $(f \circ c)(t) = f(p) + (f \circ c)'(t_0)(t - t_0) + \frac{1}{2}(f \circ c)''(t_0)(t - t_0)^2 + o((t - t_0)^2)$. Nach Definition der Richtungsableitung und Proposition 4.1.6 ist $(f \circ c)'(t_0) = \nabla f(p) d$. Zu zeigen ist also nur $(f \circ c)''(t_0) = \left(((\nabla f) \circ c)\, c'(t)\right)'(t_0) = \nabla f(p) c''(t_0) + d^\top \nabla^2 f(p) d$ (vgl. Produktregel). Dafür betrachten wir die Funktion

$$\begin{aligned} (f \circ c)'(t) &= \nabla f(c(t)) c'(t) \\ &= \sum_{i=1}^{n} c_i'(t) \frac{\partial f}{\partial x_i}(c(t)) \\ &= \sum_{i=1}^{n} c_i'(t) \left(\frac{\partial f}{\partial x_i} \circ c \right)(t). \end{aligned}$$

Nach Additionsregel dürfen wir Summandenweise ableiten und berechnen zunächst mit der Produktregel und Proposition 4.1.6:

$$(c_i' \frac{\partial f}{\partial x_i} \circ c)'(t_0) = c_i''(t_0) \frac{\partial f}{\partial x_i}(p) + d_i \nabla \frac{\partial f}{\partial x_i} d.$$

Fassen wir die Summanden zusammen, so erhalten wir:

$$\begin{aligned}(f \circ c)''(t_0) &= \sum_{i=1}^{n} \left(c_i''(t_0) \frac{\partial f}{\partial x_i}(p) + d_i \nabla \frac{\partial f}{\partial x_i} d \right) \\ &= \nabla f(p) c''(t_0) + d^\top \nabla^2 f(p) d.\end{aligned}$$

□

4.2 Notwendige und hinreichende Bedingungen für Extremwerte

Bei den folgenden Überlegungen wollen wir den zulässigen Bereich S in unsere Überlegungen miteinbeziehen. Kommen wir zurück auf die algorithmische Idee aus der Einleitung dieses Kapitels, so müssen wir irgendwie ausdrücken, was es heißt, daß es in keine Richtung mehr „bergab“ geht.

Definition 4.2.1 *Sei $S \subseteq \mathbb{R}^n, x \in S$ und $d \in \mathbb{R}^n$. Dann heißt d* zulässige Richtung *für x bzgl. S , wenn es ein $\varepsilon > 0$ und einen differenzierbaren Weg $c : [0, \varepsilon] \to S$, gibt mit $c(0) = x$ und $c'(0) = d$.*

Notwendig für ein lokales Minimum ist, daß es keine zulässige Abstiegsrichtung gibt.

Proposition 4.2.2 (Notwendige Bedingung erster Ordnung) *Sei $S \subseteq U \subseteq \mathbb{R}^n$, U offen, $f \in C^1(U)$. Ist dann x^* ein relatives Minimum von f in S, so gilt für jede zulässige Richtung d für x bzgl. S:*

$$\nabla f(x^*) d \geq 0.$$

Beweis. Da d zulässige Richtung ist, gibt es ein $\varepsilon > 0$ und einen differenzierbaren Weg $c : [0, \varepsilon] \to S$ mit $c(0) = x^*$ und $c'(0) = d$. Wir betrachten wieder die Funktion $f \circ c$. Nach Definition der Ableitung ist

$$(f \circ c)'(0) = \lim_{h \to 0} \frac{(f \circ c)(h) - f(x^*)}{h}.$$

Wäre nun $(f \circ c)'(0) < 0$, so folgte hieraus die Existenz eines $0 < \delta < \varepsilon$ mit

$$\frac{(f \circ c)(\alpha) - f(x^*)}{\alpha} < 0 \text{ für alle } 0 < \alpha < \delta$$

und somit $(f \circ c)(\alpha) < f(x^*)$ für alle $\alpha < \delta$. Dies widerspricht laut Proposition 4.1.3 der Minimalität von x^*. Also folgt notwendig $(f \circ c)'(0) \overset{4.1.6}{=} \nabla f(x^*) d \geq 0$. □

Für den Spezialfall eines Minimum im Innern von S erhalten wir folgende Aussage, die ganz analog zur notwendigen Bedingung für ein Extremum aus der Kurvendiskussion ist.

Korollar 4.2.3 *Ist x^* ein relatives Minimum von f im Innern von S, d.h. es gibt $\varepsilon > 0$ mit $U_\varepsilon(x) \subseteq S$, so ist $\nabla f(x^*) = 0$.*

Beweis. Nach Proposition 4.2.2 gilt $\nabla f(x^*)d \geq 0$ für alle zulässigen Richtungen. Nach Voraussetzung sind alle $d \in \mathbb{R}^n \setminus \{0\}$ zulässig, betrachte dazu jeweils den Weg $c_d(t) = x^* + td$. Da mit jedem d auch $-d$ zulässig ist, schließen wir $\nabla f(x^*)d = 0$ für alle $d \in \mathbb{R}^n$ und somit $\nabla f(x^*) = 0$. □

Beispiel 4.2.4 *Wir betrachten das Optimierungsproblem*

$$\begin{aligned} &\min f(x_1, x_2) = x_1^2 - x_1 + x_2 + x_1 x_2 \\ &\textit{unter } x_1 \geq 0,\ x_2 \geq 0. \end{aligned}$$

Der Gradient dieser Funktion ist $\nabla f(x_1, x_2) = (2x_1 - 1 + x_2, x_1 + 1)$. Im Innern der zulässigen Bereiches S ist $x_1 > 0$, also kann der Gradient nicht verschwinden, folglich hat die Funktion im Innern kein lokales Minimum. Am Rand gilt $x_1 = 0$ oder $x_2 = 0$. Im ersten Fall sind die zulässigen Richtungen genau die Vektoren d mit $d_1 \geq 0$ und wir haben als Gradienten $(x_2 - 1, 1)$. Die Bedingung aus Proposition 4.2.2 kann hier nicht erfüllt werden, da z.B. $(0, -1)$ stets eine zulässige Richtung ist. Im zweiten Fall ist der Gradient $(2x_1 - 1, x_1 + 1)$ und eine Richtung ist zulässig genau dann, wenn $d_2 \geq 0$. Somit ist die Richtung $(1 - 2x_1, 0)$ zulässig im Punkt (x_1, x_2) und wir erhalten als Bedingung $-(2x_1 - 1)^2 \geq 0$. Wir schließen hieraus, daß der einzige Kandidat für ein lokales Minimum $x^* = (\frac{1}{2}, 0)$ ist. Wir haben $f(x^*) = -\frac{1}{4}$. Dieser Wert ist auch das globale Minimum der Funktion $x_1^2 - x_1$ und wegen $x_1, x_2 \geq 0$ haben wir $f(x_1, x_2) \geq x_1^2 - x_1$ also hat f in x^* sogar ein globales Minimum.

In Abbildung 4.3 haben wir die Isoquanten der Funktion geplottet. Der Gradient steht stets senkrecht auf diesen Isoquanten. Die Hyperebene definiert durch die Gleichung $\nabla f(x^*)x = 0$ berührt also die Isoquantenfläche.

In der Kurvendiskussion haben Sie gelernt, daß die zweite Ableitung Auskunft über die Krümmung einer Funktion gibt. Auch das letzte Beispiel deutet an, daß die Isokostenfläche sich von der Tangentialhyperebene „wegkrümmen“ muß. Die in der Kurvendiskussion kennengelernten notwendigen Bedingungen zweiter Ordnung für lokale Minima gelten nun für alle Richtungen.

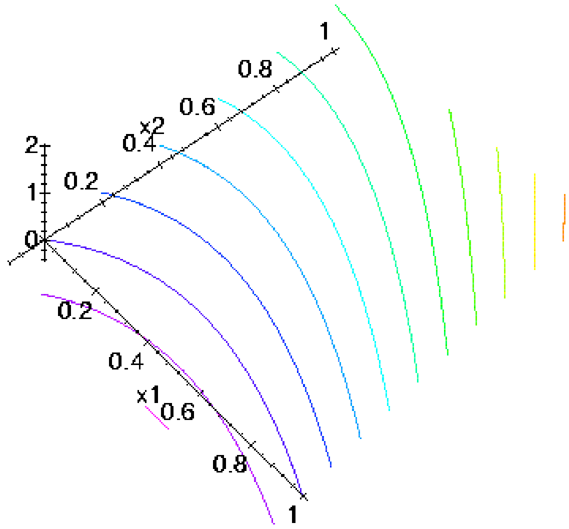


Abbildung 4.3: Isoquanten von 4.2.4

Proposition 4.2.5 (Notwendige Bedingungen zweiter Ordnung) *Sei $S \subseteq U \subseteq \mathbb{R}^n$, U offen, $f \in C^2(U)$. Ist dann x^* ein relatives Minimum von f in S, so gilt für jedes d, für das es ein ϵ gibt mit $x^* + \alpha d \in S$ für $0 \leq \alpha \leq \varepsilon$:*

a) $\nabla f(x^*)d \geq 0$,

b) *falls* $\nabla f(x^*)d = 0$, *so ist* $d^\top \nabla^2 f(x)d \geq 0$.

Beweis. Wir haben nur die zweite Behauptung zu zeigen. Sei also $c : [0, \varepsilon] \to S$ definiert durch $c(t) = x^* + td$. Dann ist d zulässige Richtung, $c(0) = x^*$ und $c'(0) = d$. Weil $c'(t) = d$ konstant ist, verschwindet $c''(0)$ und nach Satz 4.1.8 ist, $(f \circ c)(\alpha) := f(x^*) + \nabla f(x^*)d\alpha + \frac{1}{2}d^\top \nabla^2 f(x^*)d\alpha^2 + o(\alpha^2) = f(x^*) + \frac{1}{2}d^\top \nabla^2 d\alpha^2 + o(\alpha^2)$ für $\alpha \to 0$. Angenommen nun $d^\top \nabla^2 f(x)d < 0$, dann ist für hinreichend kleines α nach Definition der Landausymbole $\alpha^2 \frac{1}{2} d^\top \nabla^2 f(x)d + o(\alpha^2) < 0$ und somit $(f \circ c)(\alpha) < f(x^*)$ im Widerspruch zur Minimalität von x^*. □

Auch hier wollen wir wieder den Fall eines inneren Punktes gesondert notieren.

Korollar 4.2.6 *Ist x^* ein relatives Minimum von c im Innern von S, so gilt für alle $d \in \mathbb{R}^n$.*

a) $\nabla f(x^*) = 0$,

b) $d^\top \nabla^2 f(x) d \geq 0.$

$\square$

Für die letzte Ungleichung sagen wir auch: die Hessematrix ist *positiv semidefinit* (vgl. Definition 2.5.1).

Ähnlich wie im Eindimensionalen lassen sich die notwendigen Bedingung zweiter Ordnung zu hinreichenden verschärfen.

Proposition 4.2.7 (Hinreichende Bedingungen zweiter Ordnung) *Sei $U \subseteq \mathbb{R}^n$, U offen, $f \in C^2(U)$ und $x^* \in U$. Gilt dann*

a) $\nabla f(x^*) = 0$,

b) und $\nabla^2 f(x^*)$ ist positiv definit,

so ist x^* ein striktes lokales Minimum von f.

Beweis. Wie im Beweis von Proposition 4.2.5 entwickeln wir $f(x^* + \alpha d) = f(x^*) + \alpha^2 \frac{1}{2} d^\top \nabla^2 f(x^*) d + o(\alpha^2)$. Für α hinreichend klein ist wegen $d^\top \nabla^2 f(x^*) d > 0$ folglich $f(x^* + \alpha d) > f(x^*)$. Hieraus folgt, daß x^* auf jedem Strahl ein striktes lokales Minimum ist. Die Behauptung folgt dann aus dem folgenden Satz. $\square$

Satz 4.2.8 *Sei $x^* \in U \subseteq \mathbb{R}^n$ offen und $f \in C(U)$ eine Funktion. Gibt es dann für alle $d \in \mathbb{R}^n \setminus \{0\}$ ein $\alpha_d > 0$, so daß 0 striktes lokales Minimum der Funktion $f_d : [0, \alpha_d] \to \mathbb{R}$, definiert durch $f_d(t) := f(x^* + td)$, ist, so ist x^* striktes lokales Minimum von f.*

Beweis. Sei $S^{n-1} := \{d \in \mathbb{R}^n \mid \|d\| = 1\}$ die Menge aller Einheitsrichtungen. Nach Voraussetzung gibt es für jedes $d \in S^{n-1}$ ein α_d mit $f_d(t) > f_d(0)$ für $0 < t < \alpha_d$. Sei für jedes d ein solches $\alpha_d \in]0, \infty[$ maximal gewählt. (Beachten Sie, daß das Maximum, falls es von ∞ verschieden ist, existiert, da wir $f_d(t) > f_d(0)$ für $t < \alpha_d$ fordern.) Sei nun $U_k := \{d \in S^{n-1} \mid \alpha_d > \frac{1}{k}\}$. Dann gilt $U_k \subseteq U_{k+1}$ und $S^{n-1} = \bigcup_{i=1}^\infty U_i$. Gibt es nun ein n_0 mit $S^{n-1} = U_{n_0}$, so ist x^* ein globales Minimum, wenn man f auf $U_{\frac{1}{n_0}}$ einschränkt, also speziell ein lokales Minimum. Nehmen wir also an, daß dies nicht der Fall ist. Da S^{n-1} beschränkt und abgeschlossen ist, können wir eine konvergente Folge d_k mit $d_k \in S^{n-1} \setminus U_k$ konstruieren. Sei $d = \lim_{k \to \infty} d_k$. Sei $k_0 \geq \frac{1}{\alpha_d}$. Dann ist $f(x^* + td) > f(x^*)$ für $0 < t < \frac{1}{n_0}$. Da f stetig ist, gilt diese Ungleichung in einer ganzen Umgebung $U_{\varepsilon_1}(x^* + td) \subseteq \mathbb{R}^n$. Somit ist aber eine ganze Umgebung $U_{\varepsilon_2}(d) \subseteq S^{n-1}$ in $U_{\frac{1}{n_0}}$ enthalten, im Widerspruch dazu, daß $d = \lim_{n \to \infty} d_n$.

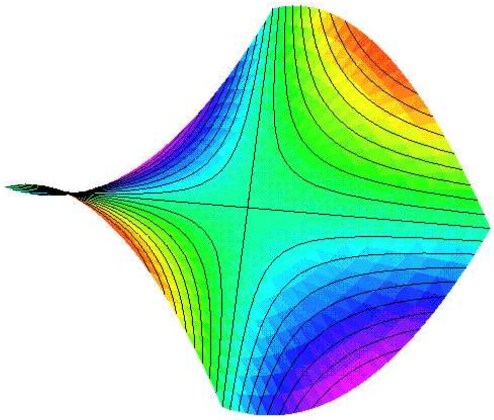
$\square$

4.3 Bedingungen für Extrema auf (Un)gleichungs-definierten Mengen

Wir wollen uns nun mit etwas spezielleren zulässigen Bereichen beschäftigen. Und zwar wollen wir auch hier eine verallgemeinerte Version der Nebenbedingungen der linearen Programmierung betrachten, nämlich Teilmengen des $\mathbb{R}^n$, die sich durch Ungleichungen $g_1(x) \leq 0, \ldots, g_k \leq 0$ mit differenzierbaren Funktionen $g_1, \ldots, g_k : \mathbb{R}^n \to \mathbb{R}$ beschreiben lassen. Wie wir im letzten Kapitel gesehen haben, ist die Situation „im Innern" eines solchen Gebildes relativ einfach. Um die Ränder genauer zu untersuchen wollen wir zunächst gleichungsdefinierte Mengen, sogenannte *Mannigfaltigkeiten*, betrachten.

4.3.1 Exkurs Mannigfaltigkeiten und Tangentialräume

Seien $h_1, \ldots, h_k : \mathbb{R}^n \to \mathbb{R}$ differenzierbare Funktionen und $k \leq n$. Wir wollen die Lösungsmenge der Gleichung $h(x) = 0$ untersuchen, wobei $h = (h_1, \ldots, h_n)^\top$. Betrachten wir hierzu als Beispiel die Funktion $h(x, y) = (x^2 - y^2)^2 - a$ für $a \geq 0$. Die betrachtete Menge ist also gerade die Höhenlinie der Funktion $(x^2 - y^2)^2$ zum Wert a.



Im allgemeinen können wir erwarten, daß eine Gleichung eine Höhenhyperfläche definiert, da sie einen „Freiheitsgrad einschränkt". Allerdings möchten wir Zerteilungspunkte, wie in der Abbildung im Ursprung zu sehen, ausschließen. Man beachte, daß der Gradient der dargestellten Funktion im Ursprung verschwindet. Bei mehreren Funktionen h_i ist man auf der sicheren Seite, wenn die Gradienten linear unabhängig sind. Dies ist die Aussage des Satzes über implizit definierte Funktionen, der einer der zentralen Sätze der Differentialrechnung mehrerer Veränderlicher ist. Wir werden ihn im Rahmen dieser Vorlesung nicht beweisen.

Satz 4.3.1 (Satz über implizit definierte Funktionen) *Sei $T \subseteq \mathbb{R}^{l+k}, x^* \in \mathbb{R}^l$, $y^* \in \mathbb{R}^k$ und (x^*, y^*) ein innerer Punkt von T. Sei $h = (h_1, \ldots, h_k) : T \to \mathbb{R}^k$ stetig differenzierbar, die Matrix $\frac{\partial h}{\partial y}(x^*, y^*)$ regulär und $h(x^*, y^*) = 0$. Dann gibt es $\varepsilon_1, \varepsilon_2 > 0$ mit $U_{\varepsilon_1}(x^*) \times U_{\varepsilon_2}(y^*) \subseteq T$ und genau eine stetig differenzierbare Funktion $f : U_{\varepsilon_1} \to U_{\varepsilon_2}$ mit $h(x, f(x)) = 0$ und $Jf(x^*) = -(\frac{\partial h}{\partial y}(x^*, y^*))^{-1} \frac{\partial h}{\partial x}(x^*, y^*)$.*

Der Satz besagt also, daß die Lösungsmenge solcher „gutartiger Gleichungssysteme“ lokal wie ein verformter $\mathbb{R}^l$ aussieht. Überprüfen wir den Spezialfall linearer Gleichungssysteme:

Beispiel 4.3.2 *Sei $A = (A_1, A_2) \in \mathbb{R}^{(l+k) \times n}, b \in \mathbb{R}^k$ und A_2 regulär. Sei $h(x) = Ax - b = A_1 x_1 + A_2 x_2 - b$. Wir berechnen daraus $x_2 = A_2^{-1}(b_2 - A_1 x_1)$.*

Als Anwendung des Satzes über implizit definierte Funktionen werden wir nun zeigen, daß die Menge aller Tangentialvektoren an die Mannigfaltigkeit in einem regulären Punkt p^*, das sind die Tangentialvektoren von Wegen, die ganz in der Mannigfaltigkeit verlaufen und p^* enthalten, ein Vektorraum ist.

Satz 4.3.3 *Sei $T \subseteq \mathbb{R}^{l+k}, h = (h_1, \ldots, h_k) : T \to \mathbb{R}^k$ stetig differenzierbar im inneren Punkt $p^* \in T, h(p^*) = 0$ und die Gradienten $\nabla h_1(p^*), \ldots, \nabla h_k(p^*)$ linear unabhängig. Dann ist die Menge der Tangentialvektoren an die Mannigfaltigkeit $S = \{x \in \mathbb{R}^{l+k} \mid h(x) = 0\}$ in p^* der l-dimensionale Untervektorraum L des $\mathbb{R}^{l+k}$:*

$$L = ker \begin{pmatrix} \nabla h_1(p^*) \\ \vdots \\ \nabla h_k(p^*) \end{pmatrix} = kerJh = \{d \in \mathbb{R}^{l+k} \mid \nabla h_i(p^*)d = 0, i = 1, \ldots, k\}.$$

Beweis. Ist d ein Tangentialvektor und $c : [0, \varepsilon] \to S$ mit $c(0) = p^*$ und $c'(0) = d$, so ist $h_i \circ c(t) = 0$ für alle i und t und somit $0 = (h_i \circ c)'(0) = \nabla h_i(p^*)d$. Sei nun umgekehrt d mit $\nabla h_i(p^*)d = 0$ für alle i gegeben. Da die Gradienten von $h_1, \ldots, h_k$ linear unabhängig sind, können wir k Koordinaten $x_{i_1}, \ldots, x_{i_k}$ wählen, so daß die Matrix

$$\frac{\partial h}{\partial(x_{i_1}, \ldots, x_{i_k})} := \begin{pmatrix} \frac{\partial h_1}{\partial x_{i_1}} & \cdots & \frac{\partial h_1}{\partial x_{i_k}} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_k}{\partial x_{i_1}} & \cdots & \frac{\partial h_k}{\partial x_{i_n}} \end{pmatrix}$$

regulär ist. Nach eventueller Umnummerierung können wir annehmen, daß dies die letzten k Koordinaten sind. Wir spalten d und p^* dazu passend auf in $d = (d_1, d_2)$ und $p^* = (p_1, p_2)$. Nach dem Satz 4.3.1 über implizit definierte Funktionen, gibt es $\varepsilon_1, \varepsilon_2 > 0$ mit $U_{\varepsilon_1}(p_1) \times U_{\varepsilon_2}(p_2) \subseteq T$ und genau eine differenzierbare Funktion $f : U_{\varepsilon_1}(p_1) \to U_{\varepsilon_2}(p_2)$ mit $h(x_1, f(x_1)) = 0$ für alle $x_1 \in U_{\varepsilon_1}(p_1)$. Wir betrachten den Weg $c : [0, \frac{\varepsilon_1}{2}] \to S$ definiert durch $c(t) = (c_1(t), c_2(t)) := (p_1 + td_1, f(p_1 + td_1))$. Dann ist c ein differenzierbarer Weg in S mit $c(0) = p^*$. Nach Kettenregel (oder zeilenweise mit Proposition 4.1.6) ist $c_2'(0) = Jf(p_1)d_1$. Nach dem Satz über implizit definierte Funktionen ist

$$Jf(p_1) = -\left(\frac{\partial h}{\partial(x_{l+1}, \ldots, x_{l+k})}\right)^{-1}(p^*)\frac{\partial h}{\partial(x_1, \ldots, x_l)}. \tag{4.1}$$

Da nach Voraussetzung $0 = Jhd = \frac{\partial h}{\partial(x_1,\ldots,x_l)}d_1 + \frac{\partial h}{\partial(x_{l+1},\ldots,x_{l+k})}d_2$ erhalten wir

$$d_2 = -\left(\frac{\partial h}{\partial(x_{l+1}, \ldots, x_{l+k})}\right)^{-1}(p^*)\frac{\partial h}{\partial(x_1, \ldots, x_l)}d_1. \tag{4.2}$$

Insgesamt erhalten wir also

$$c'(0) = (d_1, Jf(p_1)d_1) = (d_1, d_2) = d. \tag{4.3}$$

□

4.3.2 Lagrange-Multiplikatoren

Die Erkenntnisse des letzten Paragraphen wollen wir nun dazu benutzen, eine notwendige Bedingung für einen lokalen Extremwert auf einer gleichungsdefinierten Mannigfaltigkeit zu formulieren. Die geometrische Bedeutung des folgenden Satzes ist, daß in einem lokalen Minimum, der Gradient der Zielfunktion senkrecht auf der Mannigfaltigkeit stehen muß.

Satz 4.3.4 *Seien $h : \mathbb{R}^{l+k} \to \mathbb{R}^k, f : \mathbb{R}^{k+l} \to \mathbb{R}$ differenzierbare Funktionen und $x^* \in S$ sei ein lokales Minimum der Optimierungsaufgabe*

$$\begin{array}{cc} \min & f(x) \\ \textit{unter} & h(x) = 0 \end{array}.$$

Ferner seien $\nabla h_1(x^*), \ldots, \nabla h_k(x^*)$ linear unabhängig. Dann gibt es $\lambda_1, \ldots, \lambda_k \in \mathbb{R}$ mit

$$\nabla f(x^*) = \lambda_1 \nabla h_1(x^*) + \ldots + \lambda_k \nabla h_k(x^*).$$

Die λ_i nennt man Langrange'sche Multiplikatoren.

Beweis. Gibt keine solchen Lagrange'schen Multiplikatoren, so liegt $\nabla f(x^*)$ nicht im Vektorraum $L^\perp$ aller Linearkombinationen von $\nabla h_1(x^*), \ldots, \nabla h_k(x^*)$. Wählen wir eine Orthonormalbasis $b_1, \ldots, b_k$ von $L^\perp$, so ist also

$$d^\top := -\nabla f(x^*) + \sum_{i=1}^{k} (\nabla f(x^*) b_i) b_i^\top \neq 0.$$

Das heißt, der Gradient von f hat eine nicht verschwindende Komponente im Tangentialraum, denn offensichtlich ist $d^\top b_i = 0$ für alle b_i. Da die b_i den gleichen Raum aufspannen wie die $\nabla h_i(x^*)$, gilt also auch $\nabla h_i(x^*)d = 0$. Nach Satz 4.3.3 gibt es also einen differenzierbaren Weg $c : [0, \varepsilon] \to S$ mit $c(0) = x^*, c'(0) = d$, wobei $S = \{x \in \mathbb{R}^n \mid h(x) = 0\}$. Angenommen nun, x^* wäre ein lokales Minimum, dann gäbe es ein $\alpha > 0$ mit $f(x^*) \leq (f \circ c)(t)$ für alle $t \leq \alpha$. Wir schließen hieraus

$$(f \circ c)'(0) := \lim_{t \searrow 0} \frac{(f \circ c)(t) - f(x^*)}{t} \geq 0. \tag{4.4}$$

Andererseits ist aber nach Proposition 4.1.6

$$\begin{aligned} (f \circ c)'(0) &= \nabla f(x^*) d \\ &= (-d^\top + \sum_{i=1}^{k} (\nabla f(x^*) b_i) b_i^\top) d \\ &= -d^\top d < 0. \end{aligned}$$

Aus diesem Widerspruch folgt die Behauptung. □

Auch hier können wir wieder notwendige Bedingungen 2. Ordnung angeben.

Satz 4.3.5 *Unter den Voraussetzungen des letzten Satzes gilt: Ist $x^* \in S$ ein lokales Minimum der Optimierungsaufgabe, so gibt es $\lambda_1, \ldots, \lambda_k \in \mathbb{R}$ mit*

a) $\nabla f(x^*) = \sum_{i=1}^{k} \lambda_i \nabla h_i(x^*)$ *und*

b) die Matrix $L := \nabla^2 f(x^*) - \sum_{i=1}^{k} \lambda_i \nabla^2 h_i(x^*)$ ist positiv semidefinit auf dem Tangentialraum von $S = \{x \in \mathbb{R}^{l+k} \mid h(x) = 0\}$ in x^*, d.h. für alle Vektoren $d \in ker(Jh(x^*))$ gilt $d^\top Ld \geq 0$.

Beweis. Wir haben nur die zweite Bedingung zu zeigen. Sei also $Jh(x^*)d = 0$ und $c : [0, \varepsilon] \to S$ ein Weg mit $c(0) = x^*$ und $c'(0) = d$. Da f auch lokales Minimum der Funktion $f \circ c$ ist, gilt bekanntlich $(f \circ c)''(0) \geq 0$. Wir hatten bereits für den Taylorschen Satz ausgerechnet

$$(f \circ c)''(0) = \nabla f(x^*)c''(0) + d^\top \nabla^2 f(x^*)d.$$

Genauso erhalten wir durch zweimaliges Differenzieren:

$$(h_i \circ c)''(0) = \nabla h_i(x^*)c''(0) + d^\top \nabla^2 h_i(x^*)d$$

und somit wegen $(h_i \circ c) \equiv 0 \,:\, \nabla h_i(x^*)c''(0) = -d^\top \nabla^2 h_i(x^*)d$. Wir setzen dies unter Ausnutzung von $\nabla f(x^*) = \sum_{i=1}^k \lambda_i \nabla h_i(x)^*$ zusammen zu

$$\begin{aligned}
0 &\leq \nabla f(x^*)c''(0) + d^\top \nabla^2 f(x^*)d \\
&= \sum_{i=1}^k \lambda_i \nabla h_i(x)^* c''(0) + d^\top \nabla^2 f(x^*)d \\
&= \sum_{i=1}^k -\lambda_i d^\top \nabla^2 h_i(x^*)d + d^\top \nabla^2 f(x^*)d \\
&= d^\top \left(\nabla^2 f(x^*) - \sum_{i=1}^k \lambda_i \nabla^2 h_i(x^*) \right) d.
\end{aligned}$$

□

Den Beweis für die hinreichenden Bedingungen zweiter Ordnung wollen wir uns sparen.

Satz 4.3.6 *Seien $h : \mathbb{R}^{k+l} \to \mathbb{R}^k$, $f : \mathbb{R}^{k+l} \to \mathbb{R}$ differenzierbare Funktionen. Gelte $h(x^*) = 0, \nabla f(x^*) + \sum_{i=1}^k \lambda_i \nabla h_i(x^*)$ für einen Vektor λ. Sei ferner die Matrix $\nabla^2 f(x^*) + \sum_{i=1}^k \lambda_i \nabla^2 h_i(x^*)$ positiv definit. Dann ist x^* striktes lokales Minimum der Optimierungsaufgabe*

$$\begin{array}{ll} \min & f(x) \\ \textit{unter} & h(x) = 0. \end{array}$$

Beweis. Luenberger Seite 227.

4.3.3 Kuhn-Tucker Bedingungen

In diesem letzten Paragraphen wollen wir die Ergebnisse des letzten Abschnitts auf Mengen übertragen, die durch Gleichungen und Ungleichungen definiert sind.

Dafür benötigen wir ein Analogon zu Satz 4.3.3. Da die zugehörige Menge anschaulich nicht mehr wie ein „Tangentialraum aussieht“ wollen wir hier wieder von zulässigen Richtungen sprechen. Im folgenden setzen wir nicht mehr voraus, daß $n = k + l$!

Satz 4.3.7 *Seien $h : \mathbb{R}^n \to \mathbb{R}^k$ und $g : \mathbb{R}^n \to \mathbb{R}^l$ stetig differenzierbare Funktionen und*

$$S := \{x \in \mathbb{R}^n \mid h(x) = 0, g(x) \leq 0\}.$$

Für $x \in S$ sei $eq(x) = \{i \in \{1, \dots, l\} \mid g_i(x) = 0\}$. Sei x^* ein regulärer Punkt *von S, d.h. $\nabla h_1(x^*), \dots, \nabla h_k(x^*), \nabla g_{i_j}(x^*)$ für $i_j \in eq(x^*)$ seien linear unabhängig. Dann ist $d \in \mathbb{R}^n$ eine zulässige Richtung für x^* genau dann, wenn $Jhd = 0$ und $\nabla g_{i_j}(x^*)d \leq 0$ falls $i_j \in eq(x^*)$.*

Beweis. Ist d eine zulässige Richtung und $c : [0, \varepsilon]$ ein zugehöriger Weg, so folgt wie in Satz 4.3.3 $\nabla h_i(x^*)d = 0$ und $\nabla g_{i_j} d \leq 0$ falls $i_j \in eq(x^*)$. Sei nun umgekehrt d gegeben mit $\nabla h_i(x^*)d = 0$ und $\nabla g_{i_j} d \leq 0$ falls $i_j \in eq(x^*)$. Sei $\tilde{S} := \{x \in \mathbb{R}^n \mid h(x) = 0$ und $g_{i_j}(x) = 0$ falls $\nabla g_{i_j} d = 0\}$. Dann ist $x^* \in \tilde{S}$ und die Gradienten der definierenden Gleichungen von $\tilde{S}$ sind linear unabhängig in x^*. Nach Satz 4.3.3 ist d im Tangentialraum von $\tilde{S}$, also gibt es $c : [0, \varepsilon] \to T$ mit $c(0) = x^*$ und $c'(0) = d$. Um zu zeigen, daß c, zumindest nahe bei der Null, in S verläuft, müssen wir nun noch die g_{i_j} mit $\nabla g_{i_j} d < 0$ untersuchen. Nach Proposition 4.1.6 wissen wir $0 > \nabla g_{i_j}(x^*)d = (g_{i_j} \circ c)'(0) = \lim_{t \to 0} \frac{g(c(t))}{t}$ woraus die Behauptung folgt. □

Somit kommen wir nun zum Hauptresultat dieses Kapitels, das besagt, daß in einem relativen Minimum der Gradient der Zielfunktion sekrecht auf allen Gleichungs- und mit Gleichheit angenommenen Ungleichungrestriktionen stehen muß, und im letzten Falle zusätzlich in die zulässige Menge hineinzeigen muß.

Satz 4.3.8 (Kuhn-Tucker Bedingungen) *Seien $f : \mathbb{R}^n \to \mathbb{R}, h : \mathbb{R}^n \to \mathbb{R}^k$ und $g : \mathbb{R}^n \to \mathbb{R}^l$ stetig differenzierbar und x^* ein relatives Minimum des Problems*

$$\begin{array}{ll} \min & f(x) \\ \textit{unter} & h(x) = 0 \\ & g(x) \leq 0 \end{array}$$

und ein regulärer Punkt. Dann gibt es Koeffizienten $\lambda_1, \dots, \lambda_k \in \mathbb{R}$ und $\mu_1, \dots, \mu_l \in \mathbb{R}, \mu_i \leq 0$ mit

$$\nabla f(x^*) = \sum_{i=1}^{k} \lambda_i \nabla h_i(x^*) + \sum_{j=1}^{l} \mu_j \nabla g_j(x^*)$$

und $\sum_{j=1}^{l} \mu_j g_j(x^*) = 0$.

Beweis. Die letzte Bedingung besagt offensichtlich, daß $\mu_i \neq 0 \Rightarrow g(x^*) = 0$. Sei wieder $eq(x^*)$ die Menge der „aktiven Bedingungen". Außerhalb von $eq(x^*)$ setzen wir die μ_i schon mal Null. Da x^* auch lokales Minimum des Problems

$$\begin{array}{rl} \min & f(x) \\ \text{unter} & h(x) = 0 \\ & g_j(x) = 0 \text{ für } j \in eq(x^*) \end{array}$$

ist, gibt es nach Satz 4.3.4 Langrangeparameter $\lambda_1, \ldots, \lambda_k \in \mathbb{R}$ und $\mu_1, \ldots, \mu_l \in \mathbb{R}$ mit $\nabla f(x^*) = \sum_{i=1}^k \lambda_i \nabla h_i(x^*) + \sum_{j=1}^l \mu_j \nabla g_j(x^*)$. Es bleibt zu zeigen, daß stets $\mu_j \leq 0$ gilt. Angenommen dies wäre nicht der Fall und $\mu_{j_0} > 0$ für $j_0 \in eq(x^*)$. Wir wählen dann eine Orthonormalbasis $b_1, \ldots, b_t$ des von $\nabla h_1(x^*), \ldots, \nabla h_k(x^*)$ und den $\nabla g_j(x^*)$ für $j \in eq(x^*) \setminus \{j_0\}$ aufgespannten Vektorraumes und, da x^* ein regulärer Punkt und $\mu_{j_0} \neq 0$ ist, gilt auch $d^\top := -\nabla f(x^*) + \sum_{i=1}^t (\nabla f(x^*) b_i) b_i^\top \neq 0$. Wir schließen aus dem letzten Satz, daß d eine zulässige Richtung ist, denn $d^\top b_j = 0$ für alle j impliziert $\nabla h_i(x^*) d = 0$ und $\nabla g_j(x^*) d = 0$ für $j \in eq(x^*) \setminus \{j_0\}$ und

$$\begin{aligned}
\mu_{j_0} \nabla g_{j_0}(x^*) d &= \left(\nabla f(x^*) - \sum_{i=1}^k \lambda_i \nabla h_i(x^*) - \sum_{\substack{j=1 \\ j \neq i}}^l \mu_j \nabla g_j(x^*) \right) d \\
&= \nabla f(x^*) d \\
&= \nabla f(x^*) \left(-\nabla f(x^*) + \sum_{i=1}^t (\nabla f(x^*) b_i) b_i^\top \right)^\top \\
&= -\|\nabla f(x)\|^2 + \sum_{i=1}^t (\nabla f(x^*) b_i)^2 \\
&= -\|\nabla f(x^*) - \sum_{i=1}^t (f(x^*) b_i) b_i\|^2
\end{aligned}$$

$$= -\| - d\|^2 < 0.$$

Also ist d zulässig und Abstiegsrichtung im Widerspruch zu Proposition 4.2.2. □

Beispiel 4.3.9 *Wir betrachten das Problem*

$$\begin{array}{rl} \max & f(x,y) = 14x - x^2 + 6y - y^2 + 7 \\ \textit{unter} & g_1(x,y) = x + y - 2 \leq 0 \\ & g_2(x,y) = x + 2y - 3 \leq 0. \end{array}$$

Wir berechnen

$$\begin{aligned}
\nabla(-f)(x,y) &= (2x - 14, 2y - 6) \\
\nabla g_1(x) &= (1,1) \\
\nabla g_2(x) &= (1,2)
\end{aligned}$$

und überprüfen zunächst, wo der Gradient von f verschwindet. Dies ist der Fall in $(7,3)$*, das nicht im zulässigen Bereich liegt.*

Falls nur die erste Ungleichung aktiv ist, erhalten wir als Bedingungen:

$$\begin{aligned}(2x-14, 2y-6) &= (\mu, \mu),\\ x+y &= 2,\end{aligned}$$

woraus wir $x = 3$ *und* $y = -1$ *berechnen. Dieser Punkt ist zulässig für* $\mu = -* \leq 0$. *Für die zweite Ungleichung haben wir*

$$\begin{aligned}(2x-14, 2y-6) &= (\mu, 2\mu),\\ x+2y &= 3,\end{aligned}$$

und somit

$$\begin{aligned}4x-2y &= 22\\ x+2y &= 3,\end{aligned}$$

und wir berechnen $x = 5, y = -1$. *Dieser Punkt ist nicht zulässig.*

Schließlich betrachten wir noch den Fall, daß beide Ungleichungen aktiv sind. Dies ist der Fall in $(1,1)$. *Als Kuhn-Tuckerbedingung haben wir dann* $(-12,-4) = (\mu_1 + \mu_2, \mu_1 + 2\mu 2)$*, woraus wir* $\mu_2 = 8, \mu_1 = -20$ *berechnen. Da* $\mu_2 > 0$ *ist, kann hier auch kein lokales Minimum vorliegen.*

Da die Funktion bei betraglich wachsendem x oder y gegen $-\infty$ *geht, muß das globale Maximum in einem lokalen Maximum angenommen werden. Da* $(3,-1)$ *der einzige Kandidat hierfür ist, ist das Maximum der Funktion also* $f(3,-1) = 33$.

Wir wollen dieses Kapitel abschließen mit den Bedingungen zweiter Ordnung.

Satz 4.3.10 *Unter den Bedingungen des letzten Satzes gilt außerdem: die Matrix* $\nabla^2 f(x^*) - \sum_{i=1}^k \lambda_i \nabla^2 h_i^* - \sum_{j=1}^l \mu_j \nabla^2 g_j(x^*)$ *ist positiv semidefinit auf dem Tangentialraum der aktiven Nebenbedingungen von* x^*.

Beweis. Da x^* auch ein relatives Minimum für das entsprechende gleichungsdefinierte Problem ist, folgt die Behauptung sofort aus Satz 4.3.5. □

Die analogen hinreichenden Bedingungen wollen wir nicht mehr extra anführen.

Kapitel 5

Numerische Verfahren zur Nichtlinearen Programmierung

Nachdem wir im letzten Kapitel etwas Theorie betrieben haben, wollen wir uns nun den Algorithmen zuwenden. Die zentralen Stichworte in der Nichtlinearen Optimierung sind

- Suchrichtung und
- Schrittweite.

Im letzten Kapitel haben wir schon eine Charakterisierung von guten Suchrichtungen, nämlich die Abstiegsrichtungen, kennengelernt. Deswegen wollen wir uns hier zunächst stärker mit der Schrittweitensteuerung beschäftigen. Ist die Suchrichtung festgelegt, haben wir es mit einem Suchproblem der Funktion $f(p_0+\alpha d) : \mathbb{R}_+ \to \mathbb{R}$ zu tun.

5.1 Das allgemeine Suchverfahren

Sicherlich ist über eine Suche bei allgemeinen Funktionen $f : \mathbb{R} \to \mathbb{R}$, die uns nur durch eine Unterroutine gegeben sind, keine allgemeine Aussage möglich. Bei der Klassifizierung der Suchverfahren beschränken wir uns deshalb auf Funktionen, die ein eindeutiges Minimum haben. Die folgenden Voraussetzungen garantieren eine beweisbar richtige Suche. Die Suchverfahren oder eventuelle Modifikationen kann aber auch anwenden, wenn die Voraussetzungen nicht im strengen Sinne erfüllt sind.

Definition 5.1.1 *Sei* $[a, b] \subseteq \mathbb{R}$ *ein Intervall und* $f : [a, b] \to \mathbb{R}$ *eine Funktion. Dann heißt* f strikt unimodal auf [a,b], *wenn* f *genau ein lokales Minimum in* $[a, b]$ *hat.*

Proposition 5.1.2 *Sei* $[a, b] \subseteq \mathbb{R}$ *ein Intervall und* $f : [a, b] \to \mathbb{R}$ *eine Funktion. Dann ist* f *strikt unimodal genau dann, wenn für* $a \leq x < y \leq b$ *und* $\lambda \in]0, 1[$ *stets gilt:*

$$f(\lambda x + (1 - \lambda)y) < \max\{f(x), f(y)\}.$$

Beweis. Beweis durch Kontraposition. Sei $f : [x, y] \to \mathbb{R}$ eine Funktion und $\lambda \in]0, 1[, p = \lambda x + (1 - \lambda)y$ mit

$$f(p) \geq \max\{f(x), f(y)\}.$$

Wir können also ein lokales Minimum $z_1 \neq p$ von f in $[x, p]$ und ein lokales Minimum $z_2 \neq p$ von f in $[p, y]$ wählen. Da z_1 und z_2 von p verschieden sind, sind sie lokale Minima von f. Somit ist f nicht strikt unimodal. Sei nun umgekehrt $g : [a, b] \to \mathbb{R}$ nicht strikt unimodal, und seien $z_1 < z_2$ lokale Minima. Wir dürfen oBdA. annehmen (Symmetrie!), daß $g(z_1) \leq g(z_2)$. Da z_2 ein lokales Minimum von g ist, gibt es ein $1 > \varepsilon > 0$ mit $g(z_2 + \varepsilon(z_1 - z_2)) \geq g(z_2) = \max\{g(z_1), g(z_2)\}$. Somit ist $g(\varepsilon z_1 + (1 - \varepsilon)z_2) \not< \max\{g(z_1), g(z_2)\}$. □

Beispiel 5.1.3 (konvex quadratische Probleme) *Sei* $Q \in \mathbb{R}^{n \times n}$ *eine quadratische, symmetrische, positiv definite Matrix,* $b, x_0 \in \mathbb{R}^n$ *und* $d \in \mathbb{R}^n \setminus \{0\}$. *Dann ist die Funktion* $f : \mathbb{R} \to \mathbb{R}$ *definiert durch*

$$f(t) = \frac{1}{2}(x_0 + td)^\top Q(x_0 + td) + b(x_0 + td)$$

strikt unimodal auf $\mathbb{R}$, *genauer gilt für* $s < t$ *und* $\lambda \in]0, 1[$:

$$f(\lambda s + (1 - \lambda)t) < \lambda f(s) + (1 - \lambda)f(t) \leq \max\{f(s), f(t)\}.$$

Für die Gültigkeit der ersten Ungleichung in der letzten Zeile sagen wir auch: Die Funktion f *ist* strikt konvex.

Beweis.

$$\begin{aligned}
& \lambda f(s) + (1 - \lambda)f(t) - f(\lambda s + (1 - \lambda)t) \\
= \; & \lambda \left(\frac{1}{2}(x_0 + sd)^\top Q(x_0 + sd) + b(x_0 + sd)\right) + (1 - \lambda)\left(\frac{1}{2}(x_0 + td)^\top Q(x_0 + td) + b(x_0 + td)\right) \\
& -\frac{1}{2}\left((x_0 + (\lambda s + (1 - \lambda)t)d)^\top Q(x_0 + (\lambda s + (1 - \lambda)t)d)\right) + b\,(x_0 + (\lambda s + (1 - \lambda)t)d)
\end{aligned}$$

$$
\begin{aligned}
&= \lambda\left(\frac{1}{2}(x_0+sd)^\top Q(x_0+sd)\right)+(1-\lambda)\left(\frac{1}{2}(x_0+td)^\top Q(x_0+td)\right)\\
&\quad -\frac{1}{2}\left(\lambda(x_0+sd)+(1-\lambda)(x_0+td)\right)^\top Q\left((\lambda(x_0+sd)+(1-\lambda)t)d\right)\\
&= \frac{1}{2}\lambda(1-\lambda)\big((x_0+sd)^\top Q(x_0+sd)-(x_0+sd)^\top Q(x_0+td)\\
&\quad -(x_0+td)^\top Q(x_0+sd)+(x_0+td)^\top Q(x_0+td)\big)\\
&= \frac{1}{2}\lambda(1-\lambda)(x_0+sd-x_0-td)^\top Q(x_0+sd-x_0+td)\\
&\overset{Q\ pd}{>} 0
\end{aligned}
$$

□

Bemerkung 5.1.4 *Im allgemeinen heißt eine Funktion $f : S \to \mathbb{R}$* konvex*, wenn S konvex ist und*

$$f(\lambda x+(1-\lambda)y) \leq \lambda f(x)+(1-\lambda)f(y).$$

Dies ist genau dann der Fall, wenn für der Epigraph der Funktion epi(f) $:= \{(x,y) \in \mathbb{R}^{n+1} \mid x \in \mathbb{R}^n, y \geq f(x)\}$ eine konvexe Menge ist. Mit Hilfe des Mittelwertsatzes der Differentialrechnung kann man zeigen, daß eine Funktion, bei der die Hesse-Matrix stets positiv definit bzw. positiv semidefinit ist, strikt konvex bzw. konvex ist.

Bei strikt unimodalen Funktionen können wir bei der Auswertung der Funktion an zwei Punkten $x, y \in]a, b[$ feststellen, in welchem der beiden Intervalle $[a, y]$ oder $[x, b]$ das globale Minimum der Funktion im Intervall $[a, b]$ liegt.

Proposition 5.1.5 *Sei $f : [a, b] \to \mathbb{R}$ strikt unimodal und $a < x < y < b$. Dann gilt*

a) $f(x) \geq f(y) \Rightarrow \min_{[a,b]} f \in [x, b]$.

b) $f(x) \leq f(y) \Rightarrow \min_{[a,b]} f \in [a, y]$.

Beweis. Aus Symmetriegründen brauchen wir nur eine der beiden Aussagen zu zeigen. Sei also $f(x) \geq f(y)$. Für $z \in [a, x[$ sei $\lambda_z := \frac{y-x}{y-z} \in]0, 1[$. Dann ist $x = \lambda_z z + (1-\lambda_z)y$ und, da f strikt unimodal ist, gilt $f(x) < \max\{f(z), f(y)\}$. Wegen $f(x) \geq f(y)$ gilt somit für alle $z \in [a, x[: f(z) > f(x)$. □

Aus diesem Ergebnis können sie sich als „Faustregel" merken: Minimieren konvexer Funktionen über konvexen Mengen ist eine gutartige Aufgabenstellung.

Nach diesen Vorbereitungen sollte unser allgemeines Suchverfahren klar sein.

```
def findmin(f,a,x,b):
  fx=f(x)
  while (b-a)/(abs(b)+abs(a)) >= eps:
    x,y,fx,fy=choosepoint(f,a,x,b,fx)
    if fx >= fy:
      a=x
      x=y
      fx=fy
    else:
      b=y
  else:
    return (a+b)/2
```

Die hier angegeben Formulierung hat den Vorteil, daß die Funktion an jeder Stelle höchstens einmal ausgewertet wird. Der Aufwand in der nichtlinearen Programmierung wird oft mit der Anzahl der Funktionsauswertungen angegeben. So kann die Funktion etwa über eine Suchrichtung und eine mehrdimensionale kompliziertete Funktion gegeben sein.

5.2 Spezielle Suchverfahren

In diesem Abschnitt wollen wir zwei spezielle Varianten (Implementierungen) des allgemeinen Suchverfahrens diskutieren. Gütemaß für ein allgemeines Suchverfahren kann nur die Geschwindigkeit der Reduktion der Intervallänge sein. Ein natürlicher Gedanke ist es, binäre Suche zu implementieren. Man überlegt sich leicht, daß hierbei in zwei Schritten die Intervallänge halbiert wird. Bereits bei der Analyse des Euklidischen Algorithmus hatten wir in den Übungen gesehen, daß mit den Fibonacci-Zahlen eine bessere Abschätzung möglich ist.

Betrachten wir auch hier zwei aufeinanderfolgende Iterationen. Zunächst haben wir $a < x < y < b$ und entfernen entweder $[a, x]$ oder $[y, b]$. Im darauffolgenden Schritt wird ein z in $[x, b]$ bzw. in $[a, y]$ plaziert. Egal wie dies geschieht, es wird, falls im ersten Fall das linke Teilstück des Intervalls entfernt wird stets mindestens $[y, b]$ übrigbleiben bzw. im zweiten Fall mindestens $[a, x]$. Da man bei allgemeinen Funktionen im voraus nicht wissen kann, welche Stücke entfernt werden, haben wir somit gezeigt, daß man in einem Schritt im ungünstigsten Fall nicht mehr entfernen kann, als im nächsten Schritt übrigbleibt. Oder formaler

Proposition 5.2.1 *Sei S ein Suchverfahren und für $k \in \mathbb{Z}_+$*

$$S_k := \max\{\frac{b_k - a_k}{b - a} \mid a_k, b_k \text{ nach } k\text{-ter Iteration bei strikt unimodalen } f\}$$

dann gilt

$$S_k \leq S_{k+1} + S_{k+2}.$$

Beweis. Wir haben eben gezeigt, daß $(S_k - S_{k+1})(b-a) \leq \max\{x-a, b-y\}$ und $S_{k+2}(b-a) \geq \max\{x-a, b-y\}$. Also gilt $S_k - S_{k+1} \leq S_{k+2}$. □

Die Konsequenz der letzten Proposition ist, daß sich mittels der Fibonaccizahlen ein beweisbar bestes Suchverfahren konstruieren läßt. Sei dazu F_k die k-te Fibonaccizahl, also $F_0 = F_1 = 1$ und $F_{n+2} = F_{n+1} + F_n$. Sei die Anzahl N der Iterationen des Algorithmus vorgegeben. Für $k \leq N$ definieren wir im k-ten Schritt der Fibonaccisuche x_k, y_k wie folgt:

$$\begin{aligned} x_k &:= a_k + \frac{F_{N-k}}{F_{N+2-k}}(b_k - a_k) \\ y_k &:= a_k + \frac{F_{N+1-k}}{F_{N+2-k}}(b_k - a_k). \end{aligned}$$

Im N-ten Schritt setzen wir

$$\begin{aligned} x_N &:= y_N - \varepsilon = x_{N-1} - \varepsilon \quad &&\text{falls } b_N = y_{N-1}, \\ y_N &:= x_N + \varepsilon = y_{N-1} + \varepsilon \quad &&\text{falls } a_N = x_{N-1}, \end{aligned}$$

wobei ε die Maschinengenauigkeit ist.

Proposition 5.2.2 *a) Die Fibonaccisuche ist eine Implementierung des allgemeinen Suchverfahrens, d.h.*

$$\text{falls } a_{k+1} = x_k, \text{ so ist } x_{k+1} = y_k, \text{ falls } a_{k+1} = a_k, \text{ so ist } y_{k+1} = x_k.$$

b) Die Fibonaccisuche plaziert x und y symmetrisch in $[a, b]$, d.h. für $k < N$ gilt

$$x_k - a_k = b_k - y_k = \frac{F_{N-k}}{F_{N+2-k}}(b_k - a_k).$$

c) Für $k \leq N$ ist $b_k - a_k = \frac{F_{N+2-k}}{F_{N+1}}(b-a)$ und $b_{N+1} - a_{N+1} = \frac{1}{F_{N+1}}(b-a) + \varepsilon$

Beweis. Ist $a_{k+1} = x_k$, so ist $b_{k+1} = b_k$ und wir haben $a_{k+1} = a_k + \frac{F_{N-k}}{F_{N+2-k}}(b_k - a_k)$ und

$$\begin{aligned}
x_{k+1} &= a_{k+1} + \frac{F_{N-1-k}}{F_{N+1-k}}(b_{k+1} - a_{k+1}) \\
&= a_k + \frac{F_{N-k}}{F_{N+2-k}}(b_k - a_k) + \frac{F_{N-1-k}}{F_{N+1-k}}\left(b_k - a_k - \frac{F_{N-k}}{F_{N+2-k}}(b_k - a_k)\right) \\
&= a_k + \left(\frac{F_{N-k}}{F_{N+2-k}} + \frac{F_{N-1-k}}{F_{N+1-k}}(1 - \frac{F_{N-k}}{F_{N+2-k}})\right)(a_k - b_k) \\
&= a_k + \left(\frac{F_{N-k}}{F_{N+2-k}} + \frac{F_{N-1-k}}{F_{N+1-k}}(\frac{F_{N+1-k}}{F_{N+2-k}})\right)(a_k - b_k) \\
&= a_k + \left(\frac{F_{N-k}}{F_{N+2-k}} + \frac{F_{N-1-k}}{F_{N+2-k}}\right)(a_k - b_k) \\
&= a_k + \frac{F_{N+1-k}}{F_{N+2-k}}(a_k - b_k) = y_k
\end{aligned}$$

Im zweiten Fall berechnet man analog $y_{k+1} = x_k$ oder folgert es aus der Symmetrie (zweite Behauptung). Für diese zweite Behauptung haben wir

$$\begin{aligned}
b_k - y_k &= (b_k - a_k) - \frac{F_{N+1-k}}{F_{N+2-k}}(b_k - a_k) \\
&= (b_k - a_k)(1 - \frac{F_{N+1-k}}{F_{N+2-k}}) \\
&= (b_k - a_k)(\frac{F_{N-k}}{F_{N+2-k}}) = x_k - a_k.
\end{aligned}$$

Die letzte Behauptung zeigen wir mittels vollständiger Induktion. Die Verankerung besagt $b-a = b-a$ ist also sicher wahr. Wegen der eben gezeigten Symmetrie gilt nun

$$\begin{aligned}
b_{k+1} - a_{k+1} &= b_k - x_k \\
&= b_k - a_k - \frac{F_{N-k}}{F_{N+2-k}}(b_k - a_k) \\
&= (b_k - a_k)(1 - \frac{F_{N-k}}{F_{N+2-k}}) \\
&= \frac{F_{N+1-k}}{F_{N+2-k}}(b_k - a_k) \\
&\overset{I.V.}{=} \frac{F_{N+1-k}}{F_{N+2-k}} \frac{F_{N+2-k}}{F_{N+1}}(b - a) \\
&= \frac{F_{N+1-k}}{F_{N+1}}(b - a).
\end{aligned}$$

Schließlich ist $b_{N+1} - a_{N+1} = \frac{1}{2}(b_N - a_N) + \varepsilon = \frac{2}{2F_{N+1}}(b - a) + \varepsilon$. □

Satz 5.2.3 *Sei S wie in Proposition 5.2.1. Dann gilt:*

$$S_N \geq \frac{1}{F_{N+1}}.$$

Also kann kein Suchverfahren bei fest vorgewählter Schrittzahl eine stärkere Reduktion des Suchintervalls garantieren.

Beweis. Offensichtlich ist $S_{N-1} \leq 2S_N$. Also haben wir für $k = 1, 2 : S_{N+1-k} \leq F_k S_N$. Wir beweisen nun per Induktion, daß diese Aussage für $k \leq N+1$ gilt. Nach Proposition 5.2.1 haben wir für $k \in 3, \ldots, N+1$

$$S_{N+1-k} \leq S_{N+1-(k-1)} + S_{N+1-(k-2)} \overset{i.V.}{\leq} F_{k-1}S_N + F_{k-2}S_N = F_k S_N.$$

Die Behauptung folgt nun aus $S_0 = 1$. □

Die Fibonacci-Suche hat zwei Nachteile:

a) man muß im Voraus wissen, wieviel Iterationen man machen will, bzw. wie klein das Suchintervall werden soll. Im allgemeinen wird man jedoch auch die Funktionswerte miteinbeziehen.

b) man muß eine Tabelle der Fibonaccizahlen bereitstellen. Diese sind entweder in Gleitkommadarstellung nur angenähert oder man muß Langzahlarithmetik verwenden.

Statt dessen betrachtet man direkt das Verhalten des Quotienten benachbarter Fibonacci-Zahlen (vgl. Übung 3.Serie):

$$\begin{aligned}
\frac{F_{n+1}}{F_n} &= \frac{\left(\frac{1+\sqrt{5}}{2}\right)^{n+1} - \left(\frac{1-\sqrt{5}}{2}\right)^{n+1}}{\left(\frac{1+\sqrt{5}}{2}\right)^{n} - \left(\frac{1-\sqrt{5}}{2}\right)^{n}} \\
&= \frac{\left(\frac{1+\sqrt{5}}{2}\right) - \left(\frac{1-\sqrt{5}}{2}\right)\left(\frac{1-\sqrt{5}}{1+\sqrt{5}}\right)^{n}}{1 - \left(\frac{1-\sqrt{5}}{1+\sqrt{5}}\right)^{n}} \\
&\overset{n\to\infty}{\to} \frac{1+\sqrt{5}}{2}.
\end{aligned}$$

Die Zahl $\frac{1+\sqrt{5}}{2}$ ist auch als *Goldener Schnitt* bekannt.

> Goldener Schnitt Gesetz zur Konstruktion harmonischer Proportionen. Beim G.S. verhält sich das längere Teilstück einer Strecke zum kürzeren Teilstück wie das längere Teilstück zur gesamten Strecke. Am häufigsten kommen die Verhältniszahlen 3:5, 5:8, 8:13 und 13:21 zur Anwendung. Beim typografischen Gestalten läßt sich der G.S. auf das Verhältnis von Abständen, Schriftgrößen, Seitenproportionen etc. anwenden.
> Quelle: Kleines Glossar Typographie und Layout im Desktop-Publishing, Zusammengestellt von Jürgen F. Schopp Universität Tampere, Finnland.

Im Grenzwert wird also aus der Fibonacci-Suche die *Goldener-Schnitt-Suche*. Wir plazieren x so, daß x, y symmetrisch in $[a, b]$ liegen und a, x, y ein Goldener Schnitt ist. Setzen wir also $l_1 = x - a, l_2 = y - a$, so haben wir aus Symmetriegründen $l_1 + l_2 = b - a$. Damit wir einen Goldenen Schnitt erhalten, müssen x, y so gewählt werden, daß

$$\frac{l_1}{l_2} = \frac{l_2 - l_1}{l_1}. \tag{5.1}$$

Da nun $l_1 = b - a - l_2$ ist, erhalten wir hieraus:

$$l_2^2 + (b-a)l_2 - (b-a)^2 = 0. \tag{5.2}$$

Diese quadratische Gleichung hat genau eine positive Nullstelle, nämlich

$$\frac{\sqrt{5}-1}{2}(b-a) = \left(\frac{1+\sqrt{5}}{2}\right)^{-1}(b-a).$$

Also wird y an dieser Stelle plaziert und auch a, y, b ist ein Goldener Schnitt.
Wegen $1 - \frac{\sqrt{5}-1}{2} = \frac{3-\sqrt{5}}{2}$ erhalten wir als Vorschrift für die Goldener Schnitt-Suche.

$$\begin{aligned} x_k &:= a_k + \frac{3-\sqrt{5}}{2}(b_k - a_k) \\ y_k &:= a_k + \frac{\sqrt{5}-1}{2}(b_k - a_k). \end{aligned}$$

In Pseudocode erhalten wir:

```
leftfak=(3-sqrt(5))/2
rightfak=(sqrt(5)-1)/2

def choosepoint(f,a,x,b,fx):
  if b-x <= x-a:
    return a+leftfak*(b-a),x,f(a+leftfak*(b-a)),fx
  else:
    return x,a+rightfak*(b-a),fx,f(a+rightfak*(b-a))

def findmin(f,a,x,b):
  fx=f(x)
  while (b-a)/(abs(b)+abs(a)) >= eps:
    x,y,fx,fy=choosepoint(f,a,x,b,fx)
    if fx >= fy:
      a=x
      x=y
      fx=fy
    else:
      b=y
  else:
    return (a+b)/2
```

Bemerkung 5.2.4 *Wir haben hier nur Verfahren angesprochen, welche die Stetigkeit der Funktion ausnutzen. Es gibt einige Verfahren, welche Differenzierbarkeitsinformation ausnutzen, also etwa lineare oder quadratische Annäherung, auf die wir hier aber nicht näher eingehen wollen. Allerdings kann man die eindimensionale Variante des Newton-Verfahrens, das wir im übernächsten Abschnitt diskutieren als Beispiel heranziehen.*

5.3 Koordinatensuche, Methode des steilsten Abstiegs

In diesem Abschnitt wollen wir uns der Frage geeigneter Suchrichtungen zuwenden. Dabei wollen wir in unseren Untersuchungen Nebenbedingungen vernachlässigen. Die hier vorgestellten Algorithmen sind allerdings eher „prinzipiell“ zu verstehen und haben sich in der Praxis als nicht besonders effizient erwiesen. Wir werden auf diese Problematik später noch etwas näher eingehen.

Alle Verfahren benutzen line search als Unterroutine. Bei unseren theoretischen Überlegungen in diesem Abschnitt wollen wir von der idealisierten Vorstellung ausgehen, daß der line search stets das Minimum findet.

Das simpelste Verfahren ist die sogenannte *Koordinatenabstiegsmethode*:

Sei $\bar{x} \in \mathbb{R}^n$ gegeben. Wir fixieren alle Koordinaten bis auf die i-te und lösen

$$\min_{x_i \in \mathbb{R}} f(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_{i-1}, x_i, \bar{x}_{i+1}, \ldots, \bar{x}_n).$$

Ist x^* die optimale Lösung dieses Subproblems, so setzen wir $\bar{x} = x^*$, wählen eine andere Koordinate und fahren fort. Wir erhalten also folgenden Algorithmus (hier enden leider unsere Möglichkeiten, mit python „ausführbaren Pseudocode“ zu erzeugen):

while $\|x - x\_old\| > \varepsilon$:
  for i in [n]:
    x_old[i]=x[i]
    $\lambda = \text{argmin}_{\lambda} f(x + \lambda e_i)$ # löse mit line search
    $x = x + \lambda e_i$

Da die Auswahl der Koordinaten zyklisch erfolgt, nennt man obige Methode *zyklisches Abstiegsverfahren*. Werden die Koordinaten in der Reihenfolge $1, 2, \ldots, n-1, n, n-1, \ldots, 2, 1, 2, \ldots$ abgearbeitet, so trägt das Verfahren den Namen „Aitken double sweep method“. Nutzt man zusätzlich Differenzierbarkeitsinfomation aus und wählt stets die Koordinate mit dem größten Absolutwert im Gradienten, so erhalten wir das *Gauß-Southwell-Verfahren*.

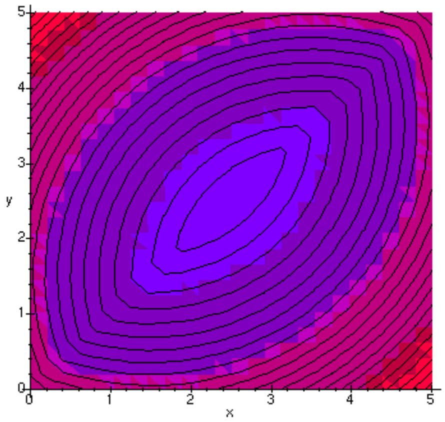
Die dargestellten Verfahren scheinen sinnvoll. Konvergenz gegen „etwas Sinnvolles“ kann man jedoch nur garantieren, wenn f differenzierbar ist. Betrachten wir zunächst ein Beispiel.

Beispiel 5.3.1 *Sei die stetige (!) Funktion $f : \mathbb{R}^2 \to \mathbb{R}$ wie folgt definiert:*

$$f(x,y) := \begin{cases} (x+y-5)^2 + (x-y-2)^2 & \textit{falls} \quad x \leq y \\ (x+y-5)^2 + (x-y+2)^2 & \textit{falls} \quad x > y. \end{cases}$$

Wir führen eine Koordinatensuche beginnend in $(0,0)$ durch. Suchen wir in x-Richtung stellen wir zunächst fest, daß das Minimum in positive Richtung x-Richtung zu suchen ist. Also müssen wir zunächst die Funktion $(x-5)^2+(x+2)^2$ minimieren. Aus Symmetriegründen oder mittels nachrechnen findet man das Minimum in $x = 1.5$. In y Richtung minimieren wir also nun die Funktion $(y-3.5)^2 + (-y+3.5)^2$ für $y \leq 1.5$ und $(y-3.5)^2+(-y-0.5)^2$ für $y > 1.5$. Diese Funktion hat ihr Minimum in $y = 1.5$. Wieder in x-Richtung betrachten wir nun die Funktion $(x-3.5)^2+(x-3.5)^2$ für $x \leq 1.5$ und $(x-3.5)^2 + (x+0.5)^2$ für $x > 1.5$. Diese ist minimal in $x = 1.5$. In y-Richtung $(y-3.5)^2 + (-y-0.5)^2$ für $1.5 \leq y$ und $(y-3.5)^2 + (-y+3.5)^2$ für

$1.5 > y$ erhalten wir die gleiche Funktion. Die Koordinatensuche terminiert also mit dem Wert $4+4=8$, das Minimum liegt aber in $(2.5, 2.5)$ mit dem Wert 4.



Ist hingegen f differenzierbar, so können wir zeigen:

Satz 5.3.2 *Ist $f : \mathbb{R}^n \to \mathbb{R}$ stetig differenzierbar und ist $(x_i)_{i\in\mathbb{N}}$ eine Folge, die von einem Koordinatensuchverfahren erzeugt wird, so konvergiert jede konvergente Teilfolge von $(x_{i_j})_{j\in\mathbb{N}}$ gegen ein x^* mit $\nabla f(x^*) = 0$.*

Beweis. Angenommen $\nabla f(x)^* \neq 0$. Sei dann $(y_i)_{i\in\mathbb{N}}$ ein Teilfolge von $(x_{i_j})_{j\in\mathbb{N}}$, bei der stets in Richtung e_{i_0} mit $\nabla f(x^*)_{i_0} \neq 0$ gesucht wird. Eine solche Teilfolge gibt es offensichtlich in allen drei Varianten des Algorithmus. Da $-f(x-te_{i_0})'(0) = f(x+te_{i_0})'(0) = \nabla f(x^*)_{i_0} \neq 0$ ist, gibt es ein $\alpha \neq 0$ mit $f(x^*+te_{i_0}) < f(x^*)$ für alle $t \in]0, \alpha]$ bzw. $t \in [\alpha, 0[$. Da f stetig ist, gibt es ein ε mit $f(x) < f(x^*)$ für alle $x \in U_\varepsilon(x^*+\alpha e_{i_0})$. Sei nun y_j ein Folgenelement mit $y_j \in U_\varepsilon(x^*)$. Dann ist $y_j+\alpha e_{i_0} \in U_\varepsilon(x^*+\alpha e_{i_0})$ und somit $f(y_j + \alpha e_{i_0}) < f(x^*)$ also gilt auch für den Nachfolger x^j von y_j in der Folge $f(x^j) < f(x^*)$. Dies widerspricht aber der Tatsache, daß $(f(x_i))_{i\in\mathbb{N}}$ monoton fallend ist und gegen $f(x^*)$ konvergiert. □

Koordinatenabstiegsvefahren haben sich in der Praxis nur in ganz wenigen Spezialfällen (z.B. Probleme mit Rechtecknebenbedingungen) bewährt, i.a. ist ihr Konvergenzverhalten schlecht, so daß selbst Probleme mit geringer Variablenzahl kaum gelöst werden können. Es scheint, daß die einzige einigermaßen erfolgreiche Variante die *Methode von Rosenbrock* ist, bei der in jeder Iteration ein neues „Koordinatensystem“ gewählt wird.

Als nächstes wollen wir in Richtung des Gradienten suchen.

Methode des steilsten Abstiegs:

while $\|\nabla f(x)\| > \varepsilon$:
  $\lambda = \text{argmin}_\lambda f(x - \lambda \nabla f(x))$ # löse mit line search
  $x = x - \lambda \nabla f(x)$.

Auch hier weisen wir die Konvergenz nach:

Satz 5.3.3 *Sei $f : \mathbb{R}^n \to \mathbb{R}$ stetig differenzierbar und sei $(x_i)_{i\in\mathbb{N}}$ eine konvergente Teilfolge einer von der Methode des steilsten Abstiegs erzeugten Punktfolge. Dann konvergiert (x_i) gegen einen stationären Punkt x^*, d.h. $\nabla f(x^*) = 0$.*

Beweis. Angenommen $\nabla f(x^*) \neq 0$. Da $-\nabla f(x^*)$ dann eine Abstiegsrichtung ist, gibt es ein α mit $f(x - \alpha \nabla f(x^*)) < f(x^*)$. Da f stetig ist, gibt es ein $\varepsilon > 0$ mit $f(x) < f(x^*)$ für alle $x \in U_\varepsilon(x - \alpha \nabla f(x^*)$. Da f stetig differenzierbar ist, gilt $\lim_{n\to\infty} \nabla f(x_n) = \nabla f(x^*)$. Da außerdem $x_i \to x^*$ geht, können wir N_0 so wählen, daß

$$\|x_{N_0} - x^*\| < \frac{\varepsilon}{2} \text{ und } \|\nabla f(x_{N_0}) - \nabla f(x^*)\| < \frac{\varepsilon}{2|\alpha|}.$$

Dann ist aber

$$\begin{aligned}
\|x_N - \alpha \nabla f(x_N) - (x^* - \alpha \nabla f(x^*))\| &= \|x_N - x^* - (\alpha \nabla f(x_N) - \alpha \nabla f(x^*))\| \\
&\leq \|x_N - x^*\| + \|\alpha \nabla f(x_N) - \alpha \nabla f(^*))\| \\
&= \|x_N - x^*\| + |\alpha| \|\nabla f(x_N) - \nabla f(^*))\| \\
&< \frac{\varepsilon}{2} + |\alpha| \frac{\varepsilon}{2|\alpha|} = \varepsilon
\end{aligned}$$

und der Widerspruch folgt wie eben. □

Obwohl dieses Verfahren lokal die „beste“ Richtung benutzt ist sein Konvergenzverhalten eher mäßig. Eine Analyse der Konvergenzrate ist etwas aufwendiger. Wir verweisen auf Luenberger S. 149–154 und zitieren:

Satz 5.3.4 *Sei $f : \mathbb{R}^n \to R$ zweimal stetig differenzierbar und x^* ein relatives Minimum von f. Sei ferner die Hessematrix $\nabla^2 f(x^*)$ positiv definit mit größtem Eigenwert $\lambda_1 > 0$ und kleinstem Eigenwert $\lambda_n > 0$. Ist dann $(x_i)_{i\in\mathbb{N}}$ eine von dem Gradientenabstiegsverfahren erzeugte, gegen x^* konvergente Folge, dann konvergiert die Folge der Zielfunktionswerte $(f(x_i))_{i\in\mathbb{N}}$ linear gegen $f(x^*)$ mit einer Konvergenzrate von höchstens $\left(\frac{\lambda_1 - \lambda_n}{\lambda_1 + \lambda_n}\right)$.*

Bemerkung 5.3.5 *Die Gradienten aufeinanderfolgender Iterationspunkte stehen senkrecht aufeinander, d.h. es gilt stets:*

$$\nabla f(x^k) \nabla f(x^{k+1})^\top = 0.$$

Beweis. $0 = f(x_k + t\nabla f(x_k))'(\lambda) = \nabla f(x_k) \nabla f(x_{k+1})^\top$. □

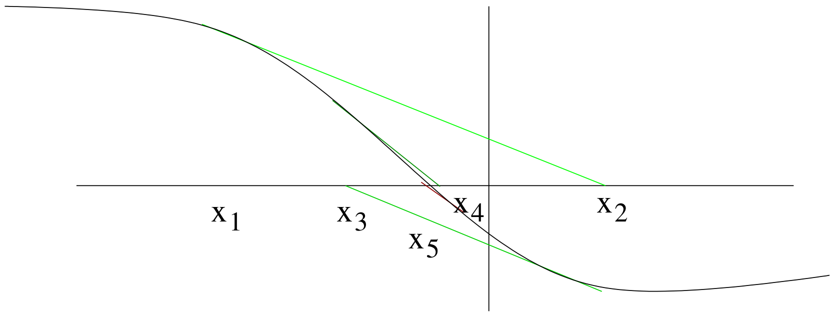
Bemerkung 5.3.6 *Bei der Benutzung von Ableitungen in numerischen Algorithmen nähert man diese üblicherweise nur an, d.h. der Gradient $\nabla f(x^*)$ wird etwa angenähert durch den Ausdruck*

$$\frac{1}{h}(f(x^* + he_1) - f(x^*), \dots, f(x^* + he_n) - f(x^*)).$$

5.4 Newtonverfahren

Der Hauptvorteil des Newtonverfahrens ist, daß das lokale Konvergenzverhalten deutlich besser als beim line search ist.

Vielleicht kennen Sie das Newton-Verfahren zur Bestimmung der Nullstelle einer Funktion noch aus der Schule: $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$. Hierbei wird iterativ die Funktion lokal durch eine lineare Funktion angenähert.



Relaxieren wir nun die Suche nach einem lokalen Minimum zu der Suche nach einem stationären Punkt, so erhalten wir die Vorschrift:

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}.$$

Hier wird also die Funktion f lokal durch die quadratische Funktion $q(x) = f(x_k) + f'(x_k)(x - x_k) + \frac{1}{2}f''(x_k)(x - x_k)^2$ approximiert und für den nächsten Iterationspunkt deren eindeutiger stationärer Punkt berechnet.

In dieser Form und Interpretation können wir das Newton-Verfahren direkt auf die Situation einer Funktion $f : \mathbb{R}^n \to \mathbb{R}$ übertragen. Wir erhalten dann folgendes Verfahren:

while $\|x - x\_old\| > \varepsilon$:
  $x\_old = x$
  $x = x - \left(\nabla^2 f(x)\right)^{-1} \nabla f(x)^\top$

Im allgemeinen können bei dieser Vorgehensweise schon bei der Bestimmung einer Nullstelle im Eindimensionalen Schwierigkeiten auftreten, nämlich, daß die Ableitung Null wird, weil man sich einem stationären Punkt nähert. Im allgemeinen können folgende Probleme auftreten

a) Im Laufe des Verfahrens kann die Hesse-Matrix singulär oder schlecht konditioniert werden.

b) Es kann passieren, daß $f(x_{k+1}) > f(x_k)$ ist.

c) Die Folge der generierten Punkte kann gegen einen Sattelpunkt konvergieren.

Wir werden nun aber nachweisen, daß lokal das Konvergenzverhalten sehr gut ist.

Satz 5.4.1 *Sei $f : \mathbb{R}^n \to \mathbb{R}$ zweimal stetig differenzierbar, sei $\nabla f(x^*) = 0$, $\nabla^2 f(x^*)$ regulär und x_1 ein Startpunkt, so daß es δ_1, δ_2 gibt mit $\delta_1\delta_2 < 1$ und für alle x mit $\|x - x^*\| < \|x_1 - x^*\|$ gilt:*

a) $\| \left(\nabla^2 f(x)\right)^{-1} \|_2 \leq \delta_1$,

b) $\dfrac{\|\nabla f(x^*) - \nabla f(x) - \nabla^2 f(x)(x - x^*)\|}{\|x - x^*\|} \leq \delta_2.$

Dann konvergiert das Newtonverfahren gegen x^*.

Beweis.

$$\begin{aligned}
\|x_{k+1} - x^*\| &= \|x_k - \left(\nabla^2 f(x_k)\right)^{-1} \nabla f(x_k) - x^*\| \\
&= \|(x_k - x^*) - \left(\nabla^2 f(x_k)\right)^{-1} (\nabla f(x_k) - \nabla f(x^*))\| \\
&= \| \left(\nabla^2 f(x_k)\right)^{-1} \left(\nabla f(x^*) - \nabla f(x_k) - \left(\nabla^2 f(x_k)\right)(x^* - x_k)\right) \| \\
&\leq \| \left(\nabla^2 f(x_k)\right)^{-1} \| \| \left(\nabla f(x^*) - \nabla f(x) - \left(\nabla^2 f(x_k)\right)(x^* - x_k)\right) \| \\
&\leq \delta_1\delta_2\|x^* - x_k\| \leq \|x^* - x_k\|.
\end{aligned}$$

Wir sagen, die Methode ist kontraktiv. Also bildet $(\|x^* - x_k\|)_{k\in\mathbb{N}}$ eine streng monoton fallende nichtnegative Folge, die somit konvergieren muß. Falls diese Folge gegen Null konvergiert, sind wir fertig. Angenommen also die Folge konvergierte gegen einen Wert > 0. Da die Werte der x_k betraglich beschränkt sind, können wir eine konvergente Teilfolge $(x_{k_i})_{i\in\mathbb{N}}$ wählen, konvergiere diese gegen y_1. Da die Abbildung $x \mapsto x - \left(\nabla^2 f(x)\right)^{-1} \nabla f(x)$ mit ∇f und $\nabla^2 f$ stetig ist, konvergiert die Folge

$(x_{k_i+1} = x_{k_i} - \left(\nabla^2 f(x_{k_i})\right)^{-1} \nabla f(x_{k_i}))_{i\in\mathbb{N}}$ gegen $y_2 = y_1 - \left(\nabla^2 f(y_1)\right)^{-1} \nabla f(y_1))$. Also ist y_2 ein anderer Häufungspunkt des Algorithmus. Somit gilt auch $\|x^* - y_1\| = \|x^* - y_2\|$. Andererseits folgt mit der gleichen Rechnung wie eben: $\|x^* - y_1\| < \|x^* - y_2\|$. Widerspruch. □

Da das Newton-Verfahren aus einer quadratischen Annäherung an die Funktion abgeleitet ist, erwarten wir lokal quadratische Konvergenz:

Satz 5.4.2 *Sei $f : \mathbb{R}^n \to \mathbb{R}$ viermal stetig differenzierbar, x^* ein Punkt mit $\nabla f(x^*) = 0$ und $\nabla^2 f(x^*)$ regulär. Sei $(x_k)_{k\in N}$ eine vom Newton-Verfahren erzeugte, gegen x^* konvergente Folge, also $x_{k+1} = N(x_k) := x_k - (\nabla^2 f(x^*))^{-1}\nabla f(x^*)^\top$. Dann konvergiert die Folge quadratisch.*

Beweis. Die Funktion $N(x)$ ist als Verknüpfung zweimal stetig differenzierbarer Funktionen zweimal stetig differenzierbar. Wir berechnen den Eintrag $(JN(x))_{ij}$ der Jacobischen. Dafür bezeichnen wir die ik-te Komponentenfunktionen der Matrix $G(x) = \left(\nabla^2 f(x)\right)^{-1}$ mit $G_{ik}(x)$. Dann ist

$$\begin{aligned}
JN(x))_{ij} &= \frac{\partial}{\partial x_j}\left(x_i - \sum_{k=1}^n G_{ik}(x)\frac{\partial f}{\partial x_k}(x)\right) \\
&= \delta_{ij} - \sum_{k=1}^n \frac{\partial G_{ik}}{\partial x_j}(x)\frac{\partial f}{\partial x_k}(x) - \sum_{k=1}^n G_{ik}(x)\frac{\partial^2 f}{\partial x_k \partial x_j}(x) \\
&= \delta_{ij} - \frac{\partial}{\partial x_j}(G_{i.})(x)\nabla f(x)^\top - (G_{i.})(\nabla^2 f(x))_{.j}) \\
&= -\frac{\partial}{\partial x_j}(G_{i.})(x)\nabla f(x)^\top,
\end{aligned}$$

wobei

$$\delta_{ij} = \begin{cases} 1 \text{ falls } i = j \\ 0 \text{ sonst} \end{cases}.$$

Da nach Annahme $\nabla f(x^*) = 0$ ist, gilt insbesondere $JN(x^*) = 0$ und wir erhalten

$$\begin{aligned}
|x_{k+1} - x^*| &= |N(x_k) - N(x^*)| \\
&\le |JN(x^*)(x_k - x^*)| + \frac{1}{2}|N''(\bar{x})||x_k - x^*|^2.
\end{aligned}$$

Hierbei nutzen wir den Satz von Taylor und den Mittelwertsatz, also ist $\bar{x}$ ein Punkt auf der Verbindungslinie von x_k nach x^*. Wir bitten die etwas laxe Schreibweise zu entschuldigen. Z.B. ist $N''(x)$ ein Gebilde, bei dem bei der „Multiplikation“ mit zwei Vektoren einen Vektor erhält. Man mache sich die Situation im Eindimensionalen klar.

Wir erhalten also mit einer Abschätzung c des Terms $\frac{1}{2}|N''(\bar{x})|$ in der Nähe von x^* nach oben:

$$|x_{k+1} - x^*| \leq c|x_k - x^*|^2.$$

□

Leider konvergiert dieses Verfahren nicht unbedingt. Man kann es auf verschiedene Arten modifizieren, um Konvergenz zu erzwingen. Wir wollen hier eine Möglichkeit darstellen.

Dazu betrachten wir allgemein Verfahren, bei denen die Iterationsvorschrift durch

$$x_{k+1} = x_k - \alpha_k M_k \nabla f(x_k)^\top$$

mit einem Suchparameter α_k und einer positiv definiten Matrix M_k gegeben ist. Dann ist in erster Näherung (bei der Entwicklung in erster Näherung bleiben quadratische und höhere Terme „übrig“)

$$\begin{aligned} f(x_{k+1}) &= f(x_k) + \nabla f(x_k)(x_{k+1} - x_k) + O(|x_{k+1} - x_k|^2) \\ &= f(x_k) - \alpha_k \nabla f(x_k) M_k \nabla f(x_k)^\top + O(\alpha_k^2). \end{aligned}$$

Nahe bei Null dominiert der in α_k lineare Term, also garantiert die positive Definitheit von M_k, daß $M_k \nabla f(x_k)^\top$ eine Abstiegsrichtung ist. Für $M_k = I$ erhalten wir das steilste Abstiegsverfahren. Genauso wie dort kann man auch globale Konvergenz nachweisen. Nahe bei einem lokalen Minimum mit positiv definiter Hessematrix erhalten wir ein parametrisiertes Newtonverfahren.

Nun ist die Hessematrix bei zweimal stetig differenzierbaren Funktionen stets symmetrisch, also gibt es nach Satz 2.4.3 eine orthogonale Matrix Q und eine Diagonalmatrix D mit $\nabla^2 f(x_k) = Q^\top DQ$, wobei auf der Diagonale von D die Eigenwerte von $\nabla^2 f(x_k)$ stehen. Die Diagonaleinträge d_{ii} ersetzt man nun durch $\max\{\delta, d_{ii}\}$, wobei $\delta > 0$ ein Steuerungsparameter ist. Nahe bei dem lokalen Minimum sind die Eigenwerte alle $\geq \delta$ und die Methode wird zum Newtonverfahren.

5.5 Verfahren der konjugierten Richtungen

Der Ansatz der konjugierten Richtungen ist ein weiterer Versuch, die Vorteile des steilsten Abstiegsverfahrens (globale Konvergenz) mit denen des Newton-Verfahrens (Ausnutzung von Information zweiter Ordnung) zu verknüpfen. Die Ideen und Eigenschaften lassen sich besser am Spezialfall eines quadratischen Programms studieren. Wir betrachten also zunächst folgendes Problem:

Sei $Q \in \mathbb{R}^{n\times n}$ eine quadratische, symmetrische, positiv definite Matrix und $b \in \mathbb{R}^n$. Wir betrachten das Minimierungsproblem

$$\min g(x) = \frac{1}{2}x^\top Qx + b^\top x.$$

Eine Möglichkeit dies Problem zu lösen, wäre die notwendigen Bedingungen zu betrachten und

$$\nabla g(x)^\top = Qx - b = 0 \tag{5.3}$$

zu lösen. Wir wollten aber näher an einer Richtungssuche bleiben und das aufwendige Invertieren der Matrix umgehen. Nach einem line search hat man die Optimierungsaufgabe optimal auf einem 1-dimensionalen affinen Teilraum des Lösungsraumes bewältigt. Wir werden nun iterativ Suchrichtungen konstruieren, so daß die Dimension des Teilraumes, auf dem das Problem optimal gelöst ist, stets um eins wächst. Dafür definieren wir zunächst.

Definition 5.5.1 *Sei $Q \in \mathbb{R}^{n\times n}$ eine quadratische, symmetrische Matrix. Dann heißen zwei Vektoren $d_1, d_2 \in \mathbb{R}^n$* Q-orthogonal, Q-konjugiert *oder auch kurz* konjugiert, *wenn $d_1^\top Qd_2 = 0$ gilt. Eine endliche Menge von Vektoren $d_1, \ldots, d_k \in \mathbb{R}^n$ heißt* Q-orthogonal, *wenn sie paarweise konjugiert sind.*

Bilden $d_1, \ldots, d_n$ eine Orthonormalbasis von Eigenvektoren einer symmetrischen Matrix, so sind sie Q-orthogonal und orthogonal im euklidischen Sinne. Im allgemeinen fallen die Begriffe nicht zusammen. Für den positiv definiten Fall gilt allerdings:

Proposition 5.5.2 *Sei $Q \in \mathbb{R}^{n\times n}$ eine quadratische, symmetrische, positiv definite Matrix und $d_1, \ldots, d_k \in \mathbb{R}^n \setminus \{0\}$ Q-orthogonal. Dann sind diese Vektoren linear unabhängig.*

Beweis. Sei $\sum_{i=1}^k \alpha_i d_i = 0$. Dann ist auch $0 = d_j^\top Q0 = \sum_{i=1}^k \alpha_i d_j^\top Qd_i = \alpha_j d_j^\top Qd_j$. Da Q positiv definit ist schließen wir, daß $\alpha_j = 0$ für $j = 1, \ldots, k$ gelten muß. □

Haben wir also in unserer Aufgabenstellung der quadratischen Optimierung konjugierte Richtungen $d_1, \ldots, d_n$ gegeben, so ist die Optimallösung x^* eine Linearkombination dieser Vektoren:

$$x^* = \sum_{i=1}^n \alpha_i d_i.$$

Aus dieser Gleichung können wir unter Berücksichtigung von Gleichung (5.3) herleiten:

$$\alpha_i = \frac{d_i^\top Q x^*}{d_i^\top Q d_i} = \frac{d_i^\top b}{d_i^\top Q d_i}. \tag{5.4}$$

Somit können wir x^* durch Skalar- und Matrixprodukte ausrechnen:

$$x^* = \sum_{i=1}^{n} \frac{d_i^\top Q x^*}{d_i^\top Q d_i} d_i. \tag{5.5}$$

Wir können diese Linearkombination allerdings erst berechnen, wenn wir eine konjugierte Basis haben. Dann ist es allerdings leicht, auf von einer Teilmenge der d_i aufgespannten Unterraum das quadratische Problem zu lösen.

Satz 5.5.3 *Seien Q, b wie eben, $d_1, \ldots, d_n$ Q-orthogonal und $x_0 \in \mathbb{R}^n$. Bezeichne B_i den von $d_1, \ldots, d_i$ an x_0 aufgespannten affinen Unterraum*

$$B_i := \{x_0 + \sum_{j=1}^{i} \lambda_j d_j \mid \lambda_j \in \mathbb{R}\} = \{y \in \mathbb{R}^n \mid d_k^\top (y - x_0) = 0, k = i+1, \ldots, n\}.$$

Seien nun $x_1, \ldots, x_n$ definiert durch

$$x_k := x_{k-1} - \frac{x_{k-1}^\top Q d_k - b^\top d_k}{d_k^\top Q d^k} d_k. \tag{5.6}$$

Dann ist x_k die Optimallösung des Problems

$$\min_{x \in B_k} \frac{1}{2} x^\top Q x - b^\top x.$$

Insbesondere löst x_n das volle quadratische Problem.

Beweis. Wir zeigen dies mittels vollständiger Induktion. Für $k = 0$ ist die Behauptung offensichtlich richtig. Sei also $k > 0$. Da die Nebenbedingungen hier linear sind, sind die Kuhn-Tucker Bedingungen äquivalent dazu, daß $\nabla g(x_k) = Qx_k - b$ senkrecht auf $d_1, \ldots, d_k$ steht. Nun ist aber

$$x_k^\top Q d_k - b^\top d_k = x_{k-1}^\top Q d_k - \frac{x_{k-1}^\top Q d_k - b d^k}{d_k^\top Q d^k} d_k^\top Q d_k - b d_k = 0 \tag{5.7}$$

und für $j < k$.

$$\begin{aligned} x_k^\top Q d_j - b^\top d_j &= x_{k-1}^\top Q d_j - \frac{x_{k-1}^\top Q d_k - b d_k}{d_k^\top Q d_k} d_k^\top Q d_j - b d_j \\ &= x_{k-1}^\top Q d_j - b d_j \\ &= \nabla g(x_{k-1}) d_j \overset{I.V.}{=} 0. \end{aligned}$$

□

Ein großer Nachteil der bisherigen Überlegungen ist, daß sie stets davon ausgehen, daß eine konjugierte Basis bekannt ist. Im folgenden werden wir diese dynamisch erzeugen.

Methode der konjugierten Gradienten Sei $x_0 \in \mathbb{R}^n$ und $d_1 = -g_0 = b - Qx_0$. Iterativ berechnen wir

$$x_k = x_{k-1} - \frac{g_{k-1}^\top d_k}{d_k^\top Q d_k} d_k \tag{5.8}$$

$$g_k = Qx_k - b \tag{5.9}$$

$$d_{k+1} = -g_k + \frac{g_k^\top Q d_k}{d_k^\top Q d_k} d_k. \tag{5.10}$$

Wenn wir nun nachweisen, daß $d_1, \ldots, d_n$ Q-konjugiert sind, so erfüllt das Verfahren offensichtlich die Voraussetzungen von Satz 5.5.3. Dies wird mit dem folgenden Satz getan:

Satz 5.5.4 *Die in (5.8, 5.9, 5.10) defnierte Methode der konjugierten Gradienten erfüllt die Voraussetzungen von Satz 5.5.3. Insbesondere gilt, falls das Verfahren nicht in x_k terminiert, daß*

a) $lin(\{g_0, g_1, \ldots, g_{k-1}\}) = lin(\{g_0, Qg_0, Q^2 g_0, \ldots, Q^k g_0\})$,

b) $lin(\{d_0, d_1, \ldots, d_k\}) = lin(\{g_0, Qg_0, Q^2 g_0, \ldots, Q^k g_0\})$,

c) $d_k^\top Q d_i = 0$ *für* $i < k$,

d) $-\dfrac{g_{k-1}^\top d_k}{d_k^\top Q d_k} = \dfrac{g_{k-1}^\top g_{k-1}}{d_k^\top Q d_k}$,

e) $\dfrac{g_k^\top Q d_k}{d_k^\top Q d_k} = \dfrac{g_k^\top g_k}{g_{k-1}^\top g_{k-1}}$.

Beweis. Wir zeigen zunächst die ersten drei Behauptungen mittels vollständiger Induktion, wobei die Verankerung klar ist. Sei also $k > 0$. Dann ist $g_k = Qx_k - b = g_{k-1} - \frac{g_{k-1}^\top d_k}{d_k^\top Q d_k} Q d_k$. Nach Induktionsvoraussetzung sind

$$g_{k-1}, Qd_k \in \operatorname{lin}(\{g_0, Qg_0, Q^2 g_0, \ldots, Q^k g_0\}),$$

somit gilt dies auch für g_k. Andererseits ist $g_k \notin \operatorname{lin}(\{g_0, Qg_0, Q^2 g_0, \ldots, Q^{k-1} g_0) = \operatorname{lin}(d_1, \ldots, d_k)$, da ansonsten das Minimum der Zielfunktion innerhalb B_k liegen würde und somit nach Induktionsvoraussetzung für die Konjugiertheit der Richtungen in x_k angenommen würde. Somit wäre dann $g_{k+1} = 0$ und der Algorithmus terminiert in x_k im Widerspruch zur Voraussetzung. Die zweite Behauptung folgt nun sofort aus der Formel (5.10) und der ersten.

Nun ist $d_{k+1}^\top Q d_i = -g_k^\top Q d_i + \frac{g_k^\top Q d_k}{d_k^\top Q d_k} d_k^\top Q d_i$. Für $i = k$ evaluiert man den Ausdruck zu Null, für $i < k$ sind beide Summanden Null, der zweite nach Induktionsvoraussetzung und der erste, weil $Qd_i \in \operatorname{lin}(d_1, \ldots, d_{i+1})$ und g_k senkrecht auf diesem Raum steht (wegen Satz 5.5.3).

Schließlich berechnen wir

$$-g_{k-1}^\top d_k = g_{k-1}^\top g_{k-1} + \frac{g_k^\top Q d_k}{d_k^\top Q d_k} g_{k-1}^\top d_{k-1},$$

wobei der zweite Summand Null ist, da g_{k-1} senkrecht auf B_{k-1} steht. Aus (5.9) schließen wir

$$Qd_k = -\frac{d_k^\top Q d_k}{g_{k-1}^\top d_k}(g_k - g_{k-1}). \tag{5.11}$$

Wegen $a), b)$ und Satz 5.5.3 ist $g_{k-1}^\top g_k = 0$ und somit $g_k^\top Q d_k = -\frac{d_k^\top Q d_k}{g_{k-1}^\top d_k} g_k^\top g_k$. □

Zum Abschluß der Vorlesung wollen wir zwei mögliche Erweiterungen auf nichtquadratische Probleme diskutieren. Eine naheliegende ist die Methode der quadratischen Approximation, bei der wir stets die Matrix Q durch die momentan aktuelle Hessematrix ersetzen.

Wir betrachten also nun wieder ein nichtlineares Optimierungsproblem

$$\min f(x).$$

Quadratische Approximation Sei $x_0 \in \mathbb{R}^n$ und $d_1 = -g_0 = \nabla f(x_0)$.

While Abbruchbedingung noch nicht erfüllt
  For $i = 1, \ldots, n$:
    $x_k = x_{k-1} - \dfrac{g_{k-1}^\top d_k}{d_k^\top \nabla^2 f(x_{k-1}) d_k} d_k$
    $g_k = \nabla f(x_k)$
    if $k \neq n$:
      $d_{k+1} = -g_k + \dfrac{g_k^\top \nabla^2 f(x_{k-1}) d_k}{d_k^\top Q d_k} d_k$
    else:
      $x_0 = x_n$.

Der Vorteil dieser Methode ist, daß man keinen expliziten line search durchführen muß. Dennoch hat dieser Ansatz zwei gravierende Nachteile. Zum einen ist die ständige Neuberechnung der Hessematrix sehr aufwendig, zum anderen ist die Methode in dieser Form im allgemeinen nicht global konvergent.

Der folgende Ansatz von Fletcher und Reeves berücksichtigt diese Nachteile, indem er einerseits in jedem Schritt einen line search durchführt und andererseits die Hessematrix durch die Identität abschätzt.

Methode nach Fletcher-Reeves Sei $x_0 \in \mathbb{R}^n$ und $d_1 = -g_0 = \nabla f(x_0)$.

> While Abbruchbedingung noch nicht erfüllt
> For $i = 1, \ldots, n$:
> $\alpha_k = \operatorname{argmin} f(x_{k-1} + \alpha_k d_k)$ # löse mit line search
> $x_k = x_{k-1} + \alpha_k d_k$
> $g_k = \nabla f(x_k)$
> if $k \neq n$:
> $d_{k+1} = -g_k + \dfrac{g_k^\top g_k}{g_{k-1}^\top g_{k-1}} d_k$
> else:
> $x_0 = x_n$.

Die globale Konvergenz dieses Verfahrens wird dadurch sichergestellt, daß einerseits der Wert der Zielfunktion nie steigt, da ein line search durchgeführt wird, und andererseits alle n Schritte ein Gradientenabstiegsschritt durchgeführt wird. Dies ist ein Beispiel für einen *Spacer Step*. Ganz allgemein kann man in Abstiegsverfahren durch „Untermischen“ von unendlich vielen Schritten eines global konvergenten Algorithmus globale Konvergenz erzwingen. Genauer gilt:

Satz 5.5.5 (Spacer Step Theorem) *Sei A eine stetige Funktion und sei $x_{k+1} = A(x_k)$ die Iterationsvorschrift eines auf der Menge $X \subseteq \mathbb{R}^n$ global gegen ein Elemente der Menge Γ (z.B. die Menge der stationären Punkte) konvergenten Verfahrens und f eine stetige Funktion mit*

$$f(A(x_k)) < f(x_k) \text{ für alle } x \in X \setminus \Gamma \tag{5.12}$$

und

$$f(A(x_k)) \leq f(x_k) \text{ für alle } x \in \Gamma. \tag{5.13}$$

Sei nun $(y_n)_{n\in\mathbb{N}}$ eine Folge, $\mathcal{K} \subseteq \mathbb{N}$ eine unendliche Indexmenge mit $y_{n+1} = A(y_n)$ falls $n \in \mathcal{K}$ und $f(y_{n+1}) \leq f(y_n)$ für alle $n \in N$. Dann konvergiert jede konvergente Teilfolge von $(y_k)_{k\in\mathcal{K}}$, gegen ein Element in Γ.

Beweis. Sei $(z_i)_{i\in\mathbb{N}}$ ein solche Folge und $\lim_{i\to infty} z_i = x^*$. Da A stetig ist, konvergiert die Folge $(A(z_i))_{i\in\mathbb{N}}$ gegen $A(x^*)$. Wegen (5.12) und (5.13) konvergiert $f(y_n)$ gegen $f(x^*) = f(A(x^*))$. Letzteres impliziert aber wegen (5.12) $x^* \in \Gamma$. $\square$

Ende der Vorlesung

Appendix A: Übungsaufgaben (mit Dr. Thomas Korb)

Aufgabe 1 (mdl.)
Der folgende Kartentrick beruht auf einer Tatsache, welche unter Mathematikern und Zauberern als das *Gilbreath-Prinzip* bekannt ist. (Benannt nach dem Mathematiker und Amateur-Zauberer *Norman Gilbreath*, der es im Jahre 1958 entdeckte. Es handelt sich um ein Beispiel für verborgene Strukturen, welche in einem scheinbar ungeordnetem System zur *Cluster-Bildung* führen.)

Ein Zauberer sortiert ein Kartenspiel nach den vier Farben, also z.B. Kreuz–Pik–Herz–Karo, Kreuz–Pik–Herz–Karo und so weiter. Dann läßt er einen Zuschauer den Kartenstapel abheben. Evtl. läßt er auf Wunsch des Publikums noch andere Zuschauer abheben. Anschließend fragt er, wieviele Karten er von dem Kartenstapel auf einen neuen Stapel herunterblättern soll. (Hierdurch wird die Reihenfolge der Karten in dem neu entstehenden Stapel vertauscht!) Die beiden so entstehenden Stapel läßt er von einem Zuschauer amerikanisch *mischen (oder einfach ineinander schieben). Der Zauberer steckt nun die Karten in die Hosentasche und behauptet, er könne die Karten in der Tasche wieder so sortieren, daß in jeweils vier aufeinanderfolgenden Karten wieder jede Farbe nur einmal vorkommt. Er zieht die Karten wieder aus der Hosentasche und blättert die ersten vier Karten auf. Jede Farbe kommt nur einmal vor. Das gleiche gilt für die nächsten vier und so weiter.*

Eine einfachere Version dieses Kartentricks erhält man, indem man die Karten nicht nach den vier Farben sondern einfach nach Rot und Schwarz sortiert und dann wie oben beschrieben vorgeht. Jeder 2-er Pack Karten sollte dann am Ende des Tricks aus einer roten und einer schwarzen Karte bestehen. Versuchen Sie, sich den Trick in dieser einfacheren Version (anschaulich) klar zu machen. Was macht der Zauberer in der Hosentasche? Sollte man sein BAFöG in Pokerrunden zu vermehren suchen, in denen der Geber amerikanisch mischt?

Aufgabe 2 (mdl.)

Das folgende Rätsel stammt von *Raymond Smullyan*, welcher in Deutschland durch Bücher wie *'Alice im Rätselland'*, *'Logik-Ritter und andere Schurken'*, *'Dame oder Tiger'*, *'Simplicius und der Baum'* und *'Spottdrosseln und Metavögel'* bekannt wurde. Er wurde 1918 in New York geboren, wo er mit 12 Jahren die High School verließ, um im Selbststudium moderne Algebra und Logik zu lernen. (Nicht zur Nachahmung empfohlen! Aber dafür sind Sie ja sowieso schon zu alt!) Später wurde Raymond Smullyan dann Professor für mathematische Logik und lehrte an diversen Universitäten in New York.

Auf einer Bank sitzen 3 Personen nebeneinander, von denen Sie wissen, daß eine immer die Wahrheit sagt, eine immer lügt und eine manchmal die Wahrheit sagt und manchmal lügt, ganz nach Belieben. Sie sollen nun mit 3 Fragen herausfinden, wer wer ist. Nachdem Sie die erste Frage gestellt haben, dürfen Sie entscheiden, wie die zweite Frage lautet und an welche der Personen Sie sie richten wollen etc. Die Fragen sollten von den Personen eindeutig mit Ja oder Nein beantwortbar sein. Wie gehen Sie vor? (Verwechseln Sie dieses Rätsel nicht mit der wohlbekannten simplen Version, in der nur der Wahrheitsliebende und der Lügner vorkommen!)

Raymond Smullyan stellte dieses Rätsel einmal auf einer Konferenz im Mathematischen Forschungsinstitut Oberwolfach, welche ich besuchte. Er sagte, er habe dieses Rätsel in keines seiner Bücher aufgenommen, da es ihm zu schwer erscheine und er sich Sorgen mache, daß manche seiner Leser wahnsinnig würden, wenn Sie nicht auf die Lösung kämen. (Er hatte aus Leserbriefen erfahren, wie lange und intensiv sich manche Leser mit den von ihm gestellten Aufgaben beschäftigten!) Ich habe die Lösung des Rätsels übrigens geträumt! Ich bin im Schlaf hochgeschreckt, habe die geträumte Lösung aufgeschrieben und am nächsten Morgen kontrolliert. Und siehe da, sie stimmte! Klingt wie ein Klischee, ist aber wahr! Thomas.

Aufgabe 3 (mdl.)

Sie werden an der Uni studentische Hilfskraft (wie ihr Übungsgruppenleiter) und verdienen daher enorm viel Geld! (Achtung: Satire!) Dieses Geld tragen Sie ins Casino, wo ein neues Spiel angeboten wird. Die Bank hat einen Kartenstapel vor sich, in dem sich ebensoviele rote wie schwarze Karten befinden (z.B. 5 rote und 5 schwarze). Leider wird der Stapel gemischt, so daß Sie die Reihenfolge der Karten nicht kennen. Sie müssen sich nun für ein Startkapital entscheiden (z.B. 100,- DM) und jedesmal die Hälfte ihres Kapitals einsetzen. Die Bank zieht eine Karte. Ist sie schwarz, so haben Sie gewonnen und ist sie rot, so gewinnt die Bank. Gespielt wird, bis der Kartenstapel aufgebraucht ist und in jeder Runde müssen sie genau die Hälfte ihres Kapitals setzen. Spielen Sie mit?

Wer einen Computer oder programmierbaren Taschenrechner hat, sollte ruhig erst *einmal eine Simulation dieses Spiels programmieren, um einen ersten Eindruck zu gewinnen. Sie werden erstaunt sein!*

Aufgabe 4 (12 Punkte)
Beweisen Sie *Satz 1.4.1* der Vorlesung (Rundung, absoluter Fehler, relativer Fehler).

Aufgabe 5 (9 Punkte)
Wir wollen nun das Rechnen mit *kurzer Arithmetik* betrachten. (Leider gibt es hier keine allgemeingültigen Regeln bzw. Konzepte, sondern es muß in jedem Einzelschritt der Berechnung überprüft werden, daß dieser numerisch stabil ausgeführt wurde.)

(a) Berechnen Sie in 2-stelliger und in 3-stelliger Arithmetik zur Basis 10 den Ausdruck $x^3 \cdot 1000$ für $x = 10^{-1}$.

(b) Bekanntlich gilt: $\sin(x)^2 + \cos(x)^2 = 1$. Berechnen Sie in 4-stelliger Arithmetik zur Basis 10 die Ausdrücke $\sin(x)^2$ und $1 - \cos(x)^2$ für $x = 10^{-2}$ und diskutieren Sie die Ergebnisse.

Hinweis: Benutzen Sie die Reihendarstellung von Sinus und Kosinus.

(c) Es seien $a, b, c \in \mathbb{R}$ mit $a > 0$. Wie Sie wissen, gelten für die *quadratische Gleichung*

$$ax^2 + bx + c = 0$$

mit $|4ac| < b^2$ die Lösungsformeln

$$x_1 = \frac{1}{2a}(-b - \operatorname{sgn}(b)\sqrt{b^2 - 4ac}), \quad x_2 = \frac{1}{2a}(-b + \operatorname{sgn}(b)\sqrt{b^2 - 4ac}).$$

Wir betrachten nun den Fall: $|4ac| \ll b^2$. Überlegen Sie sich, warum hier bei der Berechnung von x_2 numerische Instabilität auftritt, wenn man mit kurzer Arithmetik arbeitet, wohingegen die Berechnung von x_1 relativ unproblematisch bleibt. Geben Sie eine andere Formel zur Berechnung von x_2 an, bei deren Auswertung der Fehler in der gleichen Größenordnung bleibt wie bei der Berechnung von x_1.

Hinweis: x_2 kann durch x_1 ausgedrückt werden.

Aufgabe 6 (9 Punkte)
Stellen Sie die Zinseszinsformeln für die folgenden Fälle auf:

(a) Sie zahlen bei einer Bank ein Startkapital x_0 auf ein Konto ein. Die Bank verzinst Ihnen dieses mit $p\%$ pro Jahr. Wieviel Geld x_n besitzen Sie nach n Jahren? Lösen Sie die Formel für x_n auch nach x_0, p und n auf.

(b) Sie gehen vor wie in Fall (a), vereinbaren mit der Bank jedoch eine jährliche Zuzahlung einer festen Summe Z beginnend mit dem Ablauf des ersten Jahres. Wie hoch ist nun ihr Kapital nach n Jahren?

Wer mag, der kann auch die Zinseszinsformel für den häufig auftretenden Fall einer festen Verzinsung mit *monatlicher* Zuzahlung aufstellen.

Hinweis: Benutzen Sie vollständige Induktion und — für (b) — ihr Wissen über die geometrische Reihe.

Aufgabe 7 (mdl.)
Viele bekannte kombinatorische Probleme beschäftigen sich mit *Dominosteinen*. Über die Geschichte des Dominospiels ist überraschend wenig bekannt. In der abendländischen Literatur gibt es erst ab der Mitte des 18. Jahrhunderts Hinweise darauf. In China war es schon viele Jahrhunderte früher bekannt (wenn auch in einer etwas anderen Version; s. z.B. Stewart Culins Buch *Games of the Orient* von 1895, welches 1958 von Charles Tuttle neu aufgelegt wurde). [Fällt Ihnen bei den beiden vorhergehenden Jahreszahlen etwas auf? Ist das Buch 1985 vielleicht noch einmal erschienen?] Das westliche Standardspiel besteht aus 28 Steinen, die alle möglichen Paarungen der Augenzahlen von Null bis Sechs zeigen. Wir wollen einen Satz *vollständig* nennen, wenn er alle Kombinationen von Doppel-Null bis Doppel-n für eine natürliche Zahl n enthält und darüber hinaus keine weiteren Steine. Eines der ältesten kombinatorischen Probleme mit Dominosteinen ist es, herauszufinden, auf wieviele verschiedene Arten ein vollständiger Satz in einer Reihe — den Dominoregeln entsprechend — aneinandergelegt werden kann. (Die Dominoregel besagt, daß sich berührende Steine an der Kontaktstelle mit den Augenzahlen übereinstimmen müssen.) Lassen wir den trivialen Satz, bestehend aus nur einem Stein, beiseite und betrachten den einfachsten vollständigen Satz ($n = 1$). Hier ergibt sich nur eine Reihe, nämlich $0–0$, $0–1$, $1–1$ und deren Umkehrung. Betrachten Sie die Fälle $n = 2$ und $n = 3$.

Hinweis: Das Problem läßt sich formalisieren indem man Pfade in geeigneten Graphen betrachtet. Überlegen Sie sich dies. Es hat auch etwas mit dem berühmten *Königsberger Brückenproblem* zu tun. (Vielleicht finden Sie das selbst heraus. Ansonsten wird Ihr Übungsgruppenleiter es Ihnen schildern.) Im Fall $n = 3$ kann man übrigens eine sehr kurze Antwort mit Begründung geben!

Aufgabe 8 (12 Punkte)

Die durch die Rekursionsvorschrift

$$a_1 := a_2 := 1, \qquad a_n := a_{n-1} + a_{n-2} \quad (n \geq 2),$$

definierten Zahlen werden *Fibonaccizahlen* genannt. Benannt sind sie nach Leonardo von Pisa (1170–1250 ???), welcher besser unter dem Namen Fibonacci bekannt ist. Er entdeckte diese Zahlenreihe als Lösung zu folgendem Problem:

Angenommen ein Kaninchenpaar wird in einen Käfig gesperrt, um sich zu vermehren. Nehmen wir weiter an, daß Kaninchen zwei Monate nach ihrer Geburt erstmalig Junge werfen — jeweils ein Paar — und von da ab jeden Monat. Wenn keine Kaninchen sterben, wieviele Kaninchenpaare sind dann nach einem Jahr in dem Käfig?

Die Berühmtheit der Fibonaccizahlen beruht natürlich *nicht* auf dieser etwas dubiosen Aussage zur Kaninchenvermehrung, sondern auf ihrer Nützlichkeit für viele mathematische Problemstellungen und einer unglaublichen Fülle von interessanten Resultaten, welche im Laufe der Zeit über sie entdeckt worden sind. Aus Teil (a) dieser Aufgabe läßt sich z.B. leicht ableiten, daß $\lim_{n\to\infty} a_n/a_{n+1} = g$ ist, wobei g den *Goldenen Schnitt* bezeichnet, ein Teilverhältnis, welches in der Architektur der Antike und der italienischen Renaissance eine hervorragende Rolle spielte.

(a) Beweisen Sie folgendes überraschendes Ergebnis:

$$a_n = \frac{\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n}{\sqrt{5}} \qquad (\forall n \in \mathbb{N}).$$

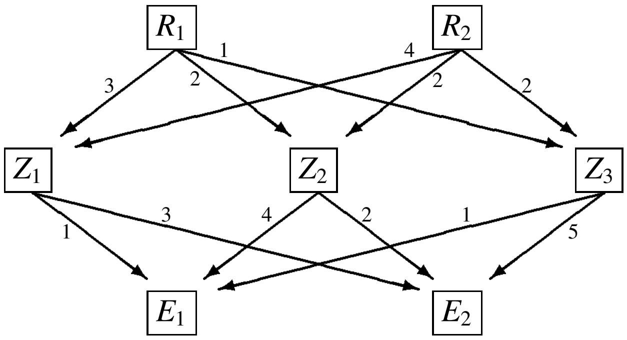
Hinweis: Benutzen Sie vollständige Induktion. Das können Sie natürlich nur, weil Ihnen schon bekannt ist, welche Formel Sie beweisen sollen! Weiß man dies nicht, so würde man ein solches Problem z.B. durch Lösen der Rekursionsgleichungen mit der Methode der *erzeugenden Funktionen* angehen. Ist $(a_n)_{n\in\mathbb{N}}$ eine Folge reeller Zahlen und besitzt die Potenzreihe $f(x) = \sum_{n=0}^{\infty} a_{n+1}x^n$ einen positiven Konvergenzradius, so nennt man f die *erzeugende Funktion* der Folge $(a_n)_{n\in\mathbb{N}}$. Im Falle der Fibonacci-Folge ergibt sich $f(x) = -\frac{1}{x^2+x-1}$ für $|x| < \frac{1}{2}$. Die Nullstellen des Polynoms x^2+x-1 sind $x_1 := \frac{-1+\sqrt{5}}{2}$ und $x_2 := \frac{-1-\sqrt{5}}{2}$. Durch weitere Analyse und Verwendung des *Identitätssatzes* für Potenzreihen erhält man dann die Formel in Teil (a). (Siehe: H. Heuser / Lehrbuch der Analysis / Teil 1 / B.G. Teubner Stuttgart / Kap. VIII.64.15.)

(b) In der Vorlesung wurde die Laufzeitfunktion des *euklidischen Algorithmus* zur Bestimmung des größten gemeinsamen Teilers zweier ganzer Zahlen $m \geq n \geq 2$ betrachtet. Es wurde gezeigt, daß die Gesamtlaufzeit des Algorithmus durch $4(\log_2(m)+1)$ beschränkt ist. Überlegen Sie sich, daß diese Schranke mit Hilfe der Fibonaccizahlen verbessert werden kann.

Hinweis: Rufen Sie sich noch einmal den Beweis von *Beispiel 1.5.1* der Vorlesung ins Gedächtnis.

Aufgabe 9 (8 Punkte)

Ein Betrieb stellt 2 Endprodukte E_1, E_2 her, welche aus 2 Rohstoffen R_1, R_2 mittelbar über 3 Zwischenprodukte Z_1, Z_2, Z_3 gewonnen werden. Der Materialverbrauch (in betriebsinternen Einheiten) für die verschiedenen Produktionsstufen wird durch folgendes Flußdiagramm wiedergegeben:



Ein Kunde bestellt nun 300 Einheiten E_1 und 100 Einheiten E_2. Berechnen Sie den Rohstoffbedarf, welcher für die Produktion dieser Bestellung notwendig ist.

Hinweis: Dies ist eine typische Aufgabe zur *Bedarfsrechnung*. Tip: Matrizen!

Aufgabe 10 (mdl.)

In der Vorlesung wurde definiert, wann eine Funktion *berechenbar* heißt. Es wurde auch gezeigt, daß es eine *nicht berechenbare* Funktion $\pi : \mathbb{N} \longrightarrow \{0, 1\}$ gibt. Überlegen Sie sich, ob die Funktion

$$f = \begin{cases} 1, & \text{falls Gott existiert,} \\ 0, & \text{falls Gott } \textit{nicht} \text{ existiert.} \end{cases}$$

berechenbar ist.

Aufgabe 11 (10 Punkte)

Beweisen Sie *Proposition 2.2.5* der Vorlesung (Transpositionsmatrizen sind selbstinvers; Charakterisierung von Permutationsmatrizen). **Aufgabe 12** (10 Punkte)

Betrachten Sie das lineare Gleichungssystem

$$\begin{pmatrix} 3 & 1 & 6 \\ 2 & 1 & 3 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 7 \\ 4 \end{pmatrix}.$$

Führen Sie eine LU–Zerlegung durch und lösen Sie dann zur Kontrolle das Gleichungssystem mit Hilfe dieser Zerlegung.

Aufgabe 13 (10 Punkte)
INPUT-OUTPUT-ANALYSE [nach W. Leontief].
Eine Volkswirtschaft bestehe aus n Wirtschaftszweigen (oder: *Sektoren*). Jeder dieser Sektoren bezieht in einer solchen Volkswirtschaft Güter oder Leistungen aus anderen Sektoren (*Input*) und gibt wiederum selbst Güter oder Leistungen an andere Sektoren ab (*Output*).

Die Produktion x_i des i-ten Sektors setzt sich wie folgt zusammen:

- x_{ii} := der Teil von x_i, welcher im Sektor i selbst benötigt wird.
- x_{ij} := der Teil von x_i, welcher in den Sektor j geht ($i \neq j$).
- y_i := der Teil von x_i, welcher die Volkswirtschaft verläßt, also konsumiert oder exportiert wird. (Kurz: *Konsum.*)

Es gilt also: $x_i = \sum_{j=1}^{n} x_{ij} + y_i \qquad (i = 1, \ldots, n)$.

Wir nehmen nun der Einfachheit halber an, daß die benötigten Inputs der Sektoren linear vom Output abhängen, d.h.:

$$x_{ij} = a_{ij} x_j \qquad (i, j = 1, \ldots, n).$$

Die Koeffizienten a_{ij} heißen die *Verbrauchskoeffizienten* und die $(n \times n)$–Matrix $A := (a_{ij})$ die *Verflechtungsmatrix* der Volkswirtschaft.

Es sei nun $A = \left(\begin{smallmatrix} 1/2 & 1/3 \\ 1/4 & 1/3 \end{smallmatrix} \right)$ die Verflechtungsmatrix einer *sehr kleinen* Volkswirtschaft, welche nur aus 2 Sektoren besteht. Für die nächste Planungsperiode wird ein Konsum von $y_1 = 10$ und $y_2 = 30$ geschätzt. Berechnen Sie, welche Produktion x_1, x_2 notwendig ist um diesen Konsumbedarf zu befriedigen.

Die oben vorgestellte Input-Output-Analyse läßt sich natürlich statt auf eine Volkswirtschaft auch auf einen einzelnen Betrieb anwenden, welcher aus mehreren Sektoren (z.B. Abteilungen) besteht.

Aufgabe 14 (10 Punkte)
Durch die *symmetrische* Matrix

$$A = \begin{pmatrix} a & b \\ b & c \end{pmatrix} \qquad (a, b, c \in \mathbb{R}, a \neq 0)$$

wird eine *quadratische Form* $\Phi : \mathbb{R}^2 \longrightarrow \mathbb{R}$ definiert. Diagonalisieren Sie A durch geeignete Koordinatentransformation, d.h. bringen Sie Φ auf eine Gestalt, in welcher nur noch rein quadratische Terme auftreten.

Hinweis: Erinnern Sie sich, daß die quadratische Form Φ durch den Ausdruck $x^T Ax$ für $x \in \mathbb{R}^2$ gegeben ist. Schreiben Sie Φ konkret hin und führen Sie eine *quadratische Ergänzung* durch.

Aufgabe 15 (mdl.)

In der letzten Übungsserie haben Sie die *Fibonaccizahlen* kennengelernt. Wenn Sie nun vor die Aufgabe gestellt würden, diese in einem Computerprogramm zu implementieren, dann könnten Sie folgende Prozedur schreiben, welche an Einfachheit und Transparenz kaum zu überbieten ist:

```
fib := proc(n)
begin
if n<2 then n else fib(n-1)+fib(n-2) end_if;
end_proc:
```

(Die Notation hier entspricht dem Computeralgebrasystem MuPAD, sollte aber selbsterklärend sein.) Überlegen Sie sich, warum diese Implementierung der Fibonaccizahlen problematisch ist und finden sie andere Möglichkeiten, die Fibonaccizahlen einem Programm als Prozedur zur Verfügung zu stellen.

Hinweis: Wer einen Computer oder programmierbaren Taschenrechner hat, der sollte obige Prozedur einmal programmieren und z.B. durch inkrementieren einer globalen Variablen überprüfen, wie oft die Prozedur sich selbst aufruft für n = 1,2,3,....

Nachtrag: Fibonaccizahlen haben übrigens mehr mit Drohnen (männliche Bienen) zu tun als mit Kaninchen. Eine Drohne hat nämlich keinen Vater. Somit gestaltet sich die Ahnenreihe einer Drohne wie folgt: 1 Elternteil (seine Mutter), 2 Großeltern (die Eltern der Mutter), 3 Urgroßeltern (weil der Vater der Mutter keinen Vater hatte), 5 Ururgroßeltern und so weiter in der Fibonaccifolge.

Aufgabe 16 (6 Punkte)

Für $x \in \mathbb{R}^n$ wurde in der Vorlesung folgende Notation eingeführt:

(i) $||x||_1 := \sum_{i=1}^n |x_i|$,

(ii) $||x||_2 := \sqrt{x^T x} = \sqrt{\sum_{i=1}^n x_i^2}$,

(iii) $||x||_\infty := \max_{1\le i\le n}\{|x_i|\}$.

Beweisen Sie, daß dies tatsächlich *Normen* sind.

Aufgabe 17 (10 Punkte)
Beweisen Sie folgenden Satz, welcher besagt, daß alle Normen auf dem $\mathbb{R}^n$ bzw. $\mathbb{C}^n$ auf eine gewisse Art und Weise zueinander äquivalent sind.

Für jedes Paar von Normen $n_1(x)$, $n_2(x)$ auf dem $\mathbb{R}^n$ (bzw. $\mathbb{C}^n$) existieren Konstanten $k, K > 0$, so daß gilt:

$$k \cdot n_2(x) \leq n_1(x) \leq K \cdot n_2(x) \qquad \forall x \in \mathbb{R}^n \quad (bzw.\ \mathbb{C}^n).$$

Im Beweis dürfen Sie benutzen, daß jede Norm auf dem $\mathbb{R}^n$ bzw. $\mathbb{C}^n$ eine (gleichmäßig) *stetige* Funktion bzgl. der Metrik $\rho(x, y) := \max_{1 \leq i \leq n}\{|x_i - y_i|\}$ ist. (Wer mag, der sollte sich auch einmal den Beweis für diesen Satz überlegen.)

Hinweis: Betrachten Sie den Fall $n_2(x) := ||x||_1$. Die Menge $M := \{x \in \mathbb{C}^n \mid \max_i\{|x_i|\} = 1\}$ ist eine *kompakte* Teilmenge des $\mathbb{C}^n$. Da $n_1(x)$ *stetig* ist, existieren somit...

Aufgabe 18 (10 Punkte)
Überprüfen Sie, welche der beiden folgenden symmetrischen Matrizen *positiv definit* ist.

$$(a) \quad \begin{pmatrix} 10 & 5 & 10 \\ 5 & -14 & 2 \\ 10 & 2 & -11 \end{pmatrix} \qquad\qquad (b) \quad \begin{pmatrix} 6 & -2 & 2 \\ -2 & 5 & 0 \\ 2 & 0 & 7 \end{pmatrix}$$

Sollte eine der Matrizen positiv definit sein, so bestimmen Sie deren Cholesky-Faktorisierung und lösen Sie auf diesem Wege das folgende Gleichungssystem

$$Ax = \begin{pmatrix} 10 \\ 10 \\ -5 \end{pmatrix}$$

wobei A die positiv definite Matrix aus (a) bzw. (b) bezeichnet.

Aufgabe 19 (4 Punkte)
Betrachten Sie das lineare Gleichungssystem

$$Ax = b \qquad \text{mit} \qquad A = (a_{ij}) = \begin{pmatrix} 1 & 200 \\ 1 & 1 \end{pmatrix} \qquad \text{und} \qquad b = \begin{pmatrix} 100 \\ 1 \end{pmatrix}$$

und lösen Sie dieses mit 2-stelliger Arithmetik, indem Sie die Gaußsche Eliminationsmethode durchführen. Wählen Sie einmal a_{11} und einmal a_{21} als Pivotelement im 1. Schritt. Vergleichen Sie die Lösungen, welche Sie so erhalten! (Dieses Beispiel illustriert *Bemerkung 2.2.11* im Skript!)

Aufgabe 20 (mdl.)
Wie würden Sie vorgehen, wenn Sie die Wurzel aus einer Zahl bestimmen sollten (z.B. $\sqrt{2}$) und nicht die gleichsam magischen Fähigkeiten eines Taschenrechners mit Wurzeltaste zur Verfügung hätten? Einem sehr alten Verfahren *(Heron-Verfahren)* zur Lösung dieses Problems liegt folgende geometrische Idee zu Grunde:

Versuche ein Quadrat zu konstruieren, dessen Flächeninhalt n ist. Die Seitenlänge dieses Quadrates ist dann $\sqrt{n}$. Starte mit einem Rechteck, welches die Seitenlängen $a_0 = n$ und $b_0 = 1$ hat. Dessen Flächeninhalt ist natürlich n. Konstruiere nun ein neues Rechteck, dessen eine Seite die Länge a_1 = arithmetisches Mittel aus a_0 und *b_0 hat und dessen andere Seite eine Länge b_1 hat, welche so gewählt wird, daß das Rechteck wieder einen Flächeninhalt von n aufweist. Fahre nun so fort und konstruiere auf diese Weise immer neue Rechtecke. Diese nähern sich immer mehr dem gesuchten Quadrat.*

Man erhält also zwei Folgen (a_i) und (b_i). Zeigen Sie, daß diese von oben bzw. von unten gegen $\sqrt{n}$ konvergieren und eine Intervallschachtelung für den gesuchten Wert liefern.

Bemerkungen: Man kann bei diesem Verfahren sehr schön sehen, auf wieviele Stellen genau man die gesuchte Wurzel schon berechnet hat! (Woran?) Außerdem konvergieren die beiden Folgen sehr schnell gegen den gesuchten Wert.

Aufgabe 21 (6 Punkte)
Es sei $A \in \mathbb{R}^{n\times n}$ eine *orthogonale* Matrix. Zeigen Sie, daß gilt: $||A||_2 = 1$.

Bemerkung: Benutzen Sie bitte nur die Definitionen von $||\_||_2$ und orthogonalen Matrizen. Weitere Ergebnisse sind nicht notwendig!

Aufgabe 22 (12 Punkte)
Zeigen Sie folgende Verallgemeinerung von *Satz 2.6.4* der Vorlesung (unter den dort angenommenen Voraussetzungen).

Seien x und $x+\Delta x$ Lösungen der Systeme $Ax = b$ bzw. $(A+\Delta A)(x+\Delta x) = (b+\Delta b)$. Dann läßt sich der relative Fehler abschätzen durch:

$$\frac{||\Delta x||}{||x||} \leq \frac{\mathrm{cond}(A)}{1-\mathrm{cond}(A)\frac{||\Delta A||}{||A||}} \left(\frac{||\Delta A||}{||A||} + \frac{||\Delta b||}{||b||} \right).$$

Bemerkung: Im Gegensatz zu *Satz 2.6.4* lassen wir hier also auch eine fehlerbehaftete rechte Seite des Gleichungssystems zu.

Aufgabe 23 (12 Punkte)

Modellbildung: Das Ende der Prohibition in Amerika stellte die *'Familie'* vor das Problem, das nun legale Alkoholgeschäft nach marktwirtschaftlichen Grundsätzen zu betreiben. Glücklicherweise hatte der Pate seinen Sohn in weiser Voraussicht Wirtschaftsinformatik studieren lassen. Da es leider noch nicht viele Computer gab, war dieser natürlich arbeitslos, aber sein theoretisches Wissen konnte er nun sinnvoll in die Familiengeschäfte einbringen. Im Lagerhaus der Familie fand er neben den üblichen Mafiautensilien (Beton, Maschinenpistolen, etc.) noch 2000 Flaschen Whisky der Marke Winfriddich, 2500 Flaschen Korbelschnaps und 1200 Flaschen 50-prozentiger Brennspiritus. Nach einer feuchtfröhlichen Marktanalyse (in verschiedenen Bars und Kneipen) sah der Sohn des Paten keine großen Absatzchancen für diese Originalprodukte. Daher beschloß er, durch Mischen die neuen Whiskysorten Winnie Talker, Blonde Beauty und Simple produzieren zu lassen, welche zu 22.5 cts, 28.5 cts bzw. 34 cts pro Flasche verkauft werden sollten. Als Wiederbeschaffungspreis für die Ausgangsprodukte Winfriddich, Korbelschaps und Brennspiritus konnte er 35 cts, 25 cts bzw. 20 cts pro Flasche ermitteln. Auf einigen Familienfeiern wurden durch exzessives Probieren folgende Vorgaben für die Mischungsverhältnisse festgelegt:

Winnie Talker	wenigstens	60% Winfriddich
	höchstens	20% Brennspiritus
Blonde Beauty	wenigstens	15% Winfriddich
	höchstens	60% Brennspiritus
Simple	höchstens	50% Brennspiritus

Absatzschwierigkeiten waren auf Grund der langjährigen guten Kontakte in der Alkoholbranche und einem familieneigenen Talent, Kunden mit *handfesten* Argumenten zu überzeugen, nicht zu befürchten. Wie sah das Lineare Programm aus, das der Sohn des Paten für dieses Problem im Hinblick auf eine Gewinnmaximierung (Erlös – Wiederbeschaffungspreis) aufstellte?

Bemerkung: Leider konnte die Familie dann doch nicht im legalen Alkoholgeschäft Fuß fassen, da der Sohn des Paten sein Studium abgebrochen hatte, bevor er lernen konnte, wie man ein solches Lineares Programm löst!

Aufgabe 24 (mdl.)
In folgendes Schema soll eine 10-stellige Zahl eingetragen werden, welche sich in folgendem Sinne selbst beschreibt: Die Ziffer im Feld n soll angeben, wie oft die Ziffer n in der Zahl selbst vorkommt. Steht also z.B. eine 3 im Feld 5, so muß die 5 dreimal in der Zahl als Ziffer vorkommen.

0	1	2	3	4	5	6	7	8	9

Wieviele Lösungen existieren zu diesem Problem und wie lauten sie?

Hinweis: Natürlich könnten Sie ein Computerprogramm schreiben, welches alle möglichen Zahlen daraufhin überprüft, ob sie sich in obiger Art und Weise selbst beschreiben. Aber das sind immerhin 10 Milliarden Zahlen und das Erstellen des Programms ist auch ein gewisser Zeitaufwand. Eine Lösung mit Papier und Bleistift (bei geeignetem systematischen Ansatz) sollte deutlich schneller zu erreichen sein.

Aufgabe 25 (10 Punkte)
Das *Farkas' Lemma* der Vorlesung wird oft in folgender äquivalenter Formulierung angegeben:

Es seien $A \in \mathbb{R}^{m\times n}$ *und* $b \in \mathbb{R}^m$ *gegeben. Dann gilt*

entweder (a) $\exists x \in \mathbb{R}^n : Ax \leq b$,

oder (b) $\exists u \in \mathbb{R}^m_+ : u^T A = 0$ *und* $u^T b < 0$,

aber nicht beides.

Leiten Sie diese Form des *Farkas' Lemmas* aus der Ihnen bekannten her.

Aufgabe 26 (10 Punkte)
Beweisen Sie die Aussage der *Übung 3.1.6* des Skripts:

Der zulässige Bereich eines linearen Programms in Standardform ist ein Polyeder.

Aufgabe 27 (10 Punkte)
Für zwei Punkte $p, q \in \mathbb{R}^n$ bezeichnen wir mit $[p, q]$ die *Verbindungsstrecke* zwischen diesen beiden Punkten, d.h.

$$[p, q] := \{(1-\lambda)p + \lambda q \mid \lambda \in [0, 1] \subset \mathbb{R}\} = \{\lambda p + \mu q \mid \lambda, \mu \in \mathbb{R}_+, \lambda + \mu = 1\}.$$

Wir sagen, eine Teilmenge $K \subset \mathbb{R}^n$ sei *konvex*, wenn mit je zwei Punkten $p, q \in K$ auch $[p, q] \subset K$ gilt. Beweisen Sie nun folgenden Satz:

Ist $K \subset \mathbb{R}^n$ konvex und sind $p_0, \ldots, p_k \in K$, so enthält K jede Konvexkombination $\lambda_0 p_0 + \ldots + \lambda_k p_k$.

Hinweis: Erinnern Sie sich: Konvekkombination heißt, daß $\lambda \geq 0$ und $\lambda_0 + \ldots + \lambda_k = 1$ gilt. Führen Sie Induktion nach k durch.

Aufgabe 28 (mdl.)

Zeichnen Sie auf ein Blatt Papier 3 Häuser, ein Kraftwerk, ein Gaswerk und einen Wasserturm. Jedes der Häuser soll nun mit allen 3 Versorgern verbunden werden (angedeutet durch einen Strich oder eine Kurve mit dem Stift), jedoch so, daß sich die Verbindungslinien *nicht* überschneiden! *(Grund: Der Boden in dieser Gegend ist ab einer geringen Tiefe schon sehr hart und man kann Rohre nicht übereinander verlegen. So eine Situation trifft man z.B. in Helsinki/Finnland an. Die Stadt ist quasi auf Granit gebaut.)* Ist dieses Problem *planar* (d.h. auf Ihrem Blatt Papier) lösbar?

Zusatz: Was wäre, wenn Helsinki sich nicht in Finnland sondern auf einem *Möbiusband* befinden würde? Natürlich gehen wir auch hier davon aus, daß es in Helsinki nur 3 Häuser und 3 Versorger gibt!

Aufgabe 29 (6 Punkte)

Dualisieren Sie die beiden folgenden linearen Programme:

$$(a) \quad \begin{array}{ll} \max & c^T x \\ \text{unter} & Ax \leq b \\ & x \geq 0 \end{array} \qquad\qquad (b) \quad \begin{array}{ll} \min & u^T b \\ \text{unter} & u^T A \geq c^T \\ & u \geq 0 \end{array}$$

(Notation wie in der Vorlesung.)

Aufgabe 30 (10 Punkte)

Zeigen Sie *Übung 3.3.5* des *neuen* Skripts:

Das duale Programm des dualen Programms ist das primale Programm. Folgern Sie hieraus: Ist das primale Programm zulässig und beschränkt, so gibt es für beide Programme Optimallösungen x^* und y^* und es gilt: $c^T x^* = {y^*}^T b$.

Diskutieren Sie ebenfalls den Unterschied zu Satz 3.3.3 (a) der Vorlesung.

Hinweis: Bringen Sie das duale Programm in Standardform und dualisieren Sie.

Aufgabe 31 (8 Punkte)

Wahrscheinlich wissen Sie alle, was ein *Graph* ist. Er besteht aus *Knoten* und *Kanten*. (In einer früheren Übungsserie haben wir schon einmal Graphen betrachtet;

und zwar hinsichtlich der Lösung des Problems mit den Dominosteinen!) Formal ist ein Graph G ein geordnetes Paar von disjunkten Mengen (V, E), so daß E eine Untermenge der Menge der ungeordneten Paare von Elementen aus V ist. Ist G ein Graph, so nennt man $V = V(G)$ die Menge der Knoten und $E = E(G)$ die Menge der Kanten von G. (Die Bezeichnungen stammen natürlich aus dem Englischen: set of *v*ertices, set of *e*dges.) Ist $e = (v_1, v_2) \in E$, so heißt dies, daß e eine Kante ist, welche die Knoten v_1 und v_2 miteinander verbindet. Das folgende ist ein Beispiel für einen einfachen Graphen mit 5 Knoten und 4 Kanten:

$$\begin{array}{lllll}
v_1 & e_1 & v_2 & e_2 & v_3 \\
\mathrm{o} & — & \mathrm{o} & — & \mathrm{o} \\
\mid e_3 & & \mid e_4 & & \\
\mathrm{o} & & \mathrm{o} & & \\
v_4 & & v_5 & &
\end{array}$$

Ein wichtiges Hilfsmittel zur Untersuchung von Graphen ist die sogenannte *(Knoten–Kanten)–Inzidenzmatrix*. Sei $G = (V, E)$ ein Graph mit $V = \{v_1, \ldots, v_n\}$ und $E = \{e_1, \ldots, e_m\}$, so ist die Inzidenzmatrix von G die $n \times m$ Matrix A, welche gegeben ist durch:

$$a_{ij} := \begin{cases} 1, & \text{falls zu dem Knoten } v_i \text{ die Kante } e_j \text{ hinführt,} \\ 0, & \text{falls zu dem Knoten } v_i \text{ die Kante } e_j \textit{ nicht} \text{ hinführt.} \end{cases}$$

Wir wollen nun spezielle Graphen betrachten. Ein Graph G heißt *bipartit* mit Knotenmengen V_1, V_2, wenn $V(G) = V_1 \cup V_2$, $V_1 \cap V_2 = \emptyset$ gilt und jede Kante einen Knoten aus V_1 mit einem Knoten aus V_2 verbindet. Es werden also keine Knoten von V_1 untereinander verbunden. Das gleiche gilt für Knoten aus V_2. (Wenn Sie auf ein Blatt Papier verschiedene Mädchen und Jungen malen und jedes Mädchen mit jedem Jungen verbinden, den sie schon einmal geküsst hat, dann erhalten Sie einen bipartiten Graphen. Etwas albern vielleicht, aber in der Literatur werden viele Probleme und Ergebnisse zu bipartiten Graphen über solche Mädchen–Junge, Mann–Frau Beziehungen verdeutlicht!) Lösen Sie nun folgende Aufgaben:

(a) Stellen Sie die Inzidenzmatrix zu obigem Beispiel (Graph mit 5 Knoten und 4 Kanten) auf.

(b) Berechnen Sie die Determinante der Inzidenzmatrix eines Graphen mit 3 Knoten, welche alle durch Kanten miteinander verbunden sind. (Der Graph sieht also wie ein Dreieck aus!)

(c) Überlegen Sie sich, daß alle Unterdeterminanten der Inzidenzmatrix eines *bipartiten* Graphen den Wert 0, 1 oder -1 haben. (Solche Matrizen nennt man *total unimodular*.)

Bemerkung: Warum die Determinanten von Inzidenzmatrizen für uns interessant sind, sehen Sie in der nächsten Aufgabe, welche in der kommenden Übungsserie wieder aufgegriffen wird!

Aufgabe 32 (6 Punkte)
Es sei A eine reguläre Matrix. Wir betrachten ein lineares Gleichungssystem $Ax = b$ und nehmen an, daß die Komponenten von b alle ganzzahlig sind. Zeigen Sie nun: Falls A die Inzidenzmatrix eines bipartiten Graphen ist, dann ist obiges Gleichungssystem ganzzahlig lösbar.

Hinweis: Erinnern Sie sich daran, was Determinanten mit dem Lösen von linearen Gleichungssystemen bzw. inversen Matrizen zu tun haben und benutzen Sie dann Aufgabe 3 (c).

Aufgabe 33 (mdl.)
Viele Zaubertricks und Rechenkunststücke beruhen auf sogenannten *zyklischen Zahlen*. Hierunter versteht man eine natürliche Zahl mit n Stellen (führende Nullen werden zugelassen und als Stellen mitgezählt!), welche folgende ungewöhnliche Eigenschaft hat: Wenn die Zahl mit einer natürlichen Zahl von 1 bis einschließlich n multipliziert wird, so enthält das Produkt die *gleichen* n Ziffern der ursprünglichen Zahl in *derselben* zyklischen Reihenfolge. (Die Ziffern der Zahl werden bei der Multiplikation also nur zyklisch permutiert.) Die einfachste zyklische Zahl ist natürlich die 1. Können Sie die nächste zyklische Zahl finden? Diese hat übrigens keine führende Null (ist also eine *echte* natürliche Zahl). Außer der trivialen 1 ist sie aber auch die einzige zyklische Zahl ohne führende Null. (Sie werden sehen, warum!) Alle Methoden sind zugelassen: Computer, theoretischer Ansatz oder einen Zauberer fragen. Von letzteren wird Ihnen wahrscheinlich fast jeder die Zahl nennen können, da sie in deren Kreisen extrem bekannt ist.

Aufgabe 34 (10 Punkte)
Wir wollen noch einmal bipartite Graphen betrachten (s. letzte Übungsserie). Ein wichtiger Begriff in diesem Zusammenhang ist das sog. *matching*. Sei G ein bipartiter Graph mit Knotenmengen V_1 und V_2. Man spricht von einem *perfekten matching*, wenn jeder Knoten aus V_1 über eine Kante mit einem Knoten aus V_2 verbunden ist, ohne daß zwei Knoten aus V_1 demselben Knoten aus V_2 zugeordnet werden müssen. Sei V_1 z.B. eine Menge von Frauen und V_2 eine Menge von Männern. Eine Kante zwischen $v_1 \in V_1$ und $v_2 \in V_2$ soll bedeuten, daß die Frau v_1 den Mann v_2 heiraten könnte (weil sie sich kennen und mögen und die Familie nichts dagegen hat etc. etc.). Kann man nun alle Frauen verheiraten, so erhält man ein perfektes matching (wobei man je nach Definition eigentlich noch die verbleibenden Männer aus dem Graphen streichen müßte). Ist es nicht möglich, alle Frauen zu verheiraten und betrachtet man nur die maximale Anzahl von Frauen,

welche verheiratet werden können, so erhält man den Begriff des *maximalen matching* (bzw. genauer: *kardinalitätsmaximales matching*). Maximales matching bedeutet also, daß man den größten Untergraphen von G betrachtet, in welchem man perfektes matching erhält. Folgender *Satz von König* beschreibt maximales matching:

$$\text{max. \# matching} = \text{min. \# Überdeckung.}$$

Gemeint ist hiermit, daß die Anzahl der Knoten aus V_1, welche zu einem maximalen matching gehören, mit der minimalen Anzahl der Knoten (aus V_1 und V_2) übereinstimmt, welche benötigt werden, damit jede Kante des gesamten Graphen mit einem dieser Knoten inzidiert (d.h. verbunden ist; diese Knoten *überdecken* also die Kantenmenge des Graphen). Betrachten Sie z.B. den bipartiten Graphen $G = (V_1 \cup V_2, E)$ mit

$$V_1 = \{v_1, v_2, v_3\}, \quad V_2 = \{v_4, v_5, v_6\} \text{ und } E = \{(v_1, v_5), (v_2, v_4), (v_2, v_6), (v_3, v_5)\}.$$

Ein maximales matching wird hier z.B. durch die *zwei* Paare (v_1, v_5) und (v_2, v_4) gegeben. Und die *zwei* Knoten v_2 und v_5 inzidieren mit allen Kanten des Graphen (und man kann nicht weniger Knoten mit dieser Eigenschaft finden). Dies ist die Aussage des Satzes von König, welchen Sie nun beweisen sollen! Definieren Sie sich hierzu in geeigneter Weise zwei Vektorräume: einen *Knotenraum* und einen *Kantenraum* und überlegen Sie sich, wie die *Knoten-Kanten-Inzidenzmatrix* des Graphen (s. letzte Übungsserie) zwischen diesen Vektorräumen als Abbildung funktioniert. Beschreiben Sie dann maximales matching in einem bipartiten Graphen durch ein lineares Programm. Dualisieren liefert dann die Aussage des Satzes von König.

Aufgabe 35 (8 Punkte)
Lösen Sie folgendes Problem mit Hilfe des Simplexverfahrens:

$$\begin{array}{lrcl} \min & 10x_1 + 3x_2 & & \\ \text{unter} & x_1 - 3x_2 & \leq & 3 \\ & x_1 + x_2 & \geq & 3 \\ & x_1 & \geq & 1 \\ & x & \geq & 0 \end{array}$$

Aufgabe 36 (Vorsicht falsch! 12 Punkte)
Wie im Skript versprochen (neuer Skript, S. 48) wollen wir nun ein Beispiel für

mögliches *Kreisen* (oder: *Zykeln*) des Simplexverfahrens betrachten:

$$\begin{array}{lrcl} \min & 2x_2+4x_4+4x_6 & & \\ \text{unter} & x_1-3x_2-x_3-x_4-x_5+6x_6 & = & 0 \\ & 2x_2+x_3-3x_4-x_5+2x_6 & = & 0 \\ & x & \geq & 0 \end{array}$$

(a) Bestimmen Sie das zu $B=\{1,2\}$ gehörige Tableau.

(b) Iterieren Sie (Simplexverfahren) unter Verwendung der *Steilster-Anstieg-Regel* (s. neuer Skript §3.6), bis Kreisen eintritt.

(c) Überlegen Sie sich, wie man das Kreisen verhindern kann.

Zusatzaufgabe: *(mdl.)* Stellen Sie unter Verwendung des dualen Programms die einzelnen Iterationsschritte graphisch dar.

Aufgabe 37 (mdl.)

Die drei Mathematiker Alfred, Bill und Charley gehen an jedem Werktag gemeinsam zum Mittagessen. Irgendwann stellten Sie fest, daß ihre Gewohnheit einen Martini zu trinken durch folgende Regeln beschrieben werden kann:

1) Wenn Alfred einen Martini bestellt, dann macht Bill das auch.

2) Entweder Bill oder Charley bestellt einen Martini, aber nie beide beim gleichen Mittagessen.

3) Alfred oder Charley oder beide bestellen immer einen Martini.

4) Wenn Charley einen Martini bestellt, dann macht Alfred das auch.

Können Sie einfacher sagen, wer wann einen Martini trinkt?

Tip: Es gibt sicherlich viele Möglichkeiten, dieses Problem zu lösen. Eine elegante Lösung erhält man jedoch durch Verwendung sog. *Venn-Diagramme*. Das sind diese Kreise, welche die meisten von Ihnen wahrscheinlich schon im Mengenlehre-Unterricht der Grundschule gezeichnet haben. Identifizieren Sie *'Martini bestellen'* mit *wahr* und *'keinen Martini bestellen'* mit *falsch* und zeichnen Sie dann geeignete Venn-Diagramme.

Aufgabe 38 (10 Punkte)

Lösen Sie folgendes Problem mit Hilfe der *Zweiphasenmethode*:

$$\begin{array}{lrcrcrcl} \min & 4x_1 & + & x_2 & + & x_3 & & \\ \text{unter} & 2x_1 & + & x_2 & + & 2x_3 & = & 4 \\ & 3x_1 & + & 3x_2 & + & x_3 & = & 3 \\ & & & & & x & \geq & 0 \end{array}$$

Aufgabe 39 (10 Punkte)
Lösen Sie folgendes Problem mit Hilfe der *Zweiphasenmethode*:

$$
\begin{array}{rrcrcrcl}
\min & 4x_1 & + & x_2 & + & x_3 & & \\
\text{unter} & 2x_1 & + & x_2 & + & 2x_3 & = & 4 \\
& 3x_1 & + & 3x_2 & + & x_3 & = & 3 \\
& & & & & x & \geq & 0
\end{array}
$$

Aufgabe 40 (10 Punkte)
In der Praxis ist es oftmals so, daß die Variablen in linearen Programmen sowohl unteren als auch *oberen Schranken* unterliegen. (Z.B. ist der Grad einer Produktion durch die Kapazitäten der Fabrik beschränkt bzw. die Menge der verfügbaren Rohstoffe durch Transportkapazitäten etc. etc.) Also gelten üblicherweise folgende Einschränkungen für eine Variable x_i in einem linearen Programm:

$$u_i \leq x_i \leq o_i \qquad (u_i = \underline{\text{u}}\text{ntere Schranke}, \ o_i = \underline{\text{o}}\text{bere Schranke}).$$

öberlegen Sie sich, daß jedes lineare Programm mit unteren und oberen Schranken für die Variablen (üblicherweise *Lineares Programm mit oberen Schranken* genannt) z.B. in die folgende Form gebracht werden kann:

$$
\begin{array}{ll}
\max & c^T x \\
\text{unter} & Ax = b \\
& 0 \leq x \leq s
\end{array}
$$

öberlegen Sie sich nun, wie die obige Form in die übliche *Standardform* überführt werden kann (Tip: *Schlupfvariablen*), auf welche dann das normale Simplexverfahren anwendbar ist. Wie groß wird die Matrix des so erhaltenen linearen Programms in Standardform?

In den öbungen werden Sie eine effektivere Methode (leicht modifizierte Form des Simplexverfahrens) kennenlernen, um lineare Programme mit oberen Schranken zu lösen.

Aufgabe 41 (10 Punkte)
Redundanz in linearen Programmen. Wir betrachten ein System von linearen Ungleichungen in Standardform:

$$
\begin{array}{rcl}
Ax & = & b \\
x & \geq & 0
\end{array}
$$

Folgende 3 Fälle von *Redundanz* können in diesem System auftreten:

(1) *Redundante Gleichungen:* Eine der Gleichungen kann als Linearkombination der anderen ausgedrückt werden.

(2) *Null-Variablen:* Eine Variable x_i ist in allen Lösungen des Systems gleich Null.

(3) *Nicht-extremale Variablen:* Für eine Variable x_i ist die Ungleichung $x_i \geq 0$ redundant (d.h. es ergibt sich schon aus den Gleichungen, daß $x_i \geq a$ für ein $a > 0$ gelten muß).

Beantworten Sie nun folgende Fragestellungen:

(a) Gibt es Null-Variablen in dem folgenden System?

$$\begin{array}{rcrcrcrcl} 2x_1 & + & 3x_2 & + & 4x_3 & + & 4x_4 & = & 6 \\ x_1 & + & x_2 & + & 2x_3 & + & x_4 & = & 3 \\ & & & & & & x & \geq & 0 \end{array}$$

(b) Gibt es nicht-extremale Variablen in dem folgenden System?

$$\begin{array}{rcrcrcl} x_1 & + & 3x_2 & + & 4x_3 & = & 4 \\ 2x_1 & + & x_2 & + & 3x_3 & = & 6 \\ & & & & x & \geq & 0 \end{array}$$

(c) Wie kann man ein System von linearen Ungleichungen in Standardform, in welchem einer der obigen Fälle von Redundanz auftritt, vereinfachen? Wie sieht eine solche Vereinfachung für (a) und (b) aus?

Aufgabe 42 (mdl.)

Sie erholen sich in einem Ruderboot auf einem kreisrunden See. Plötzlich entdecken Sie am Ufer einen Mathematiker mit einer neuen öbungsserie, welche er Ihnen aufdrängen will! Nach einiger Zeit der Beobachtung kommen Sie zu dem Schluß, daß der Mathematiker wohl Nichtschwimmer ist, aber genau viermal so schnell laufen kann, wie Sie rudern. Er kann aber nicht so schnell laufen wie Sie. Könnten Sie also mit dem Boot eine Stelle des Ufers erreichen, ohne dort den Mathematiker mit der öbungsserie zu treffen, so wären Sie gerettet, da Sie schneller laufen können als er!

Können Sie einen Weg finden, wie Sie einen Punkt des Ufers vor dem Mathematiker erreichen?

Aufgabe 43 (6 Punkte)
Bestimmen Sie alle Funktionen $f : \mathbb{R}^2 \longrightarrow \mathbb{R}$ mit $\nabla f(x) = x$ für alle $x \in \mathbb{R}^2$.

Aufgabe 44 (10 Punkte)
Die Funktionen $f, g : S \subset \mathbb{R}^n \longrightarrow \mathbb{R}$ (S offen) mögen in $x \in S$ Gradienten besitzen. Dann trifft dies auch für $f+g, fg, \alpha f$ ($\alpha \in \mathbb{R}$) und f/g (falls $g(x) \neq 0$) zu. Zeigen Sie nun, daß die folgenden Formeln gelten (wobei als Argument jeweils x zu ergnzen ist):

(a) $\nabla(f+g) = \nabla f + \nabla g$ (b) $\nabla(fg) = f\nabla g + g\nabla f$

(c) $\nabla(\alpha f) = \alpha \nabla f$ (d) $\nabla\left(\dfrac{f}{g}\right) = \dfrac{g\nabla f - f\nabla g}{g^2}$

Aufgabe 45 (8 Punkte)
Stellen Sie die Hessematrix der Funktion $f(x, y) = x^3 + y^3 - 3xy$ auf dem $\mathbb{R}^2$ auf.

Aufgabe 46 (6 Punkte)
Geben Sie eine Parametrisierung für eine *Zykloide* an. *(Die Zykloide beschreibt die Bahn eines Punktes auf der Peripherie eines Kreises (vom Radius r), wenn letzterer auf der positiven x-Achse der x−y-Ebene abrollt und besagter Punkt sich zu Beginn der Bewegung im Nullpunkt befand. Oder anders: Der Weg eines Nagels in einem Autoreifen (vorausgesetzt der Reifen wird nicht kleiner, weil er Luft verliert)!)*

Aufgabe 47 (mdl.)
Zeichnen Sie auf ein Blatt Papier eine 4×4-Matrix und schreiben Sie die Zahlen von 1 bis 16 von links nach rechts, angefangen mit dem Feld $(1, 1)$ in die Matrix. (In der ersten Zeile steht also $1, 2, 3, 4$, in der zweiten steht $5, 6, 7, 8$ etc.) Whlen Sie nun ein beliebiges Feld der Matrix und kreisen Sie die Zahl dort ein. Ziehen Sie nun einen vertikalen Strich durch die Spalte, in der die eingekreiste Zahl steht und einen horizontalen Strich durch die entsprechende Reihe. Kreisen Sie dann eine der verbleibenden Zahlen ein (also eine, welche noch nicht durchgestrichen ist) und ziehen Sie wieder die entsprechenden vertikalen und horizontalen Striche. Nun wiederholen Sie dieses Verfahren noch einmal und zuletzt kreisen Sie die einzig übriggebliebene Zahl ein. Wenn man nun die 4 eingekreisten Zahlen addiert, so ergibt sich als Summe 34. Warum?

Wenn Sie erst einmal verstanden haben, wie dieser Trick funktioniert, dann können Sie sich beliebig viele Variationen davon ausdenken. Versuchen Sie es einmal!

Aufgabe 48 (mdl.)
Magische Quadrate kann man natürlich noch auf die Spitze treiben. Wer mag, der

versuche einmal einen *magischen Würfel* zu konstruieren (z.B. einen $3 \times 3 \times 3$-Würfel; hier müssen Sie die Zahlen von 1 bis 27 so auf die Felder verteilen, daß sich in alle Richtungen (auch auf den Diagonalen!) immer die gleiche Summe ergibt). **Aufgabe 49** (8 Punkte)

Untersuchen Sie noch einmal *Bsp. 4.2.4* der Vorlesung, d.h. das Optimierungsproblem

$$\begin{array}{ll} \min & f(x_1,x_2) = x_1^2 - x_1 + x_2 + x_1x_2 \\ \text{unter} & x_1 \geq 0, \quad x_2 \geq 0, \end{array}$$

im Hinblick auf die *'Notwendigen Bedingungen zweiter Ordnung'* (*Proposition 4.2.5*).

Aufgabe 50 (12 Punkte)
Betrachten Sie das Optimierungsproblem

$$\begin{array}{ll} \min & f(x_1,x_2) = x_1^3 - x_1^2x_2 + 2x_2^2, \\ \text{unter} & x_1 \geq 0, \quad x_2 \geq 0, \end{array}$$

und überprüfen Sie, ob im *Inneren* des zulssigen Bereiches Lösungen existieren.

Aufgabe 51 (10 Punkte)
Betrachten Sie die Gleichung $x_1^2 + x_2 = 0$. Eine offensichtliche Lösung ist $x_1 = 0$ und $x_2 = 0$. Gibt es eine Umgebung dieser Lösung, auf der eine Funktion Φ existiert mit $x_1 = \Phi(x_2)$? (Wenn nicht, dann überlegen Sie sich, welche Voraussetzung des *Satzes über implizit definierte Funktionen* nicht erfüllt ist. Wie sieht es mit der Existenz eines solchen Φ für die anderen Lösungen dieser Gleichung aus?)

Aufgabe 52 (mdl.)
3 Quickies *(Wenig Mathematik erforderlich, aber dennoch nett!)*

- In einem Zimmer befindet sich auf einem Tisch eine Glühbirne, vor dem Zimmer 4 Lichtschalter, welche jeweils auf 'AN' bzw. 'AUS' gestellt werden können. Aber nur einer der Lichtschalter schaltet die Glühbirne wirklich an bzw. aus. Die drei anderen sind ohne Funktion. Sie dürfen nun mit den Lichtschaltern herumspielen. Danach gehen Sie in das Zimmer und können dann beantworten, welcher der Schalter die Glühbirne steuert. (Natürlich können Sie, whrend Sie die Lichtschalter ausprobieren, nicht in das Zimmer sehen! Die Tür zu dem Zimmer ist verschlossen und lßt kein Licht durch.) Wie machen Sie das?

- Sie sind Kandidat bei einer Gameshow. Sie konnten bisher alle Fragen richtig beantworten und haben alle Aufgaben erfolgreich gelöst. Nun, in der entscheidenden Endrunde, dürfen Sie sich eine von 10 Türen aussuchen. Hinter 9 Türen befindet sich nichts, hinter der verbleibenden Tür der große Preis. Sie whlen eine Tür. Von den 9 anderen Türen öffnet der Showmaster (welcher weiß, wo sich der Preis befindet) nun 8 und zeigt, daß sich nichts dahinter befindet. Dann fragt er Sie, ob Sie ihre Tür behalten wollen, oder ob Sie lieber die von ihm noch nicht geöffnete 9te Tür nehmen wollen. Wie verhalten Sie sich?

- Ein böser Zauberer droht damit, 3 Prinzen in Frösche zu verwandeln. Er gibt ihnen jedoch noch eine Chance. Er stellt Sie hintereinander in einer Reihe auf, so daß der letzte seine beiden Vordermnner sehen kann, der mittlere nur seinen Vordermann und der erste keinen der beiden anderen. Nun verbindet der Zauberer den Prinzen die Augen und setzt jedem von ihnen einen Hut auf. Er sagt ihnen, daß die Hüte rot oder schwarz sind und daß nicht alle 3 Prinzen einen Hut der gleichen Farbe tragen. Dann nimmt er ihnen die Augenbinden wieder ab und sagt, daß er alle 3 Prinzen nicht verzaubern wird, wenn mindestens einer von ihnen innerhalb von 10 Minuten sagen kann, welche Farbe sein Hut hat. Der Zauberer aktiviert eine große Sanduhr und die Zeit luft. Nach 9 Minuten und 59 Sekunden antwortet einer der Prinzen richtig und rettet damit alle drei! Welcher Prinz war das und wie hat er die Aufgabe gelöst?

Aufgabe 53 (8 Punkte)

Betrachten Sie das Problem

$$\begin{array}{rl} \min & x_1x_2 + x_2x_3 + x_1x_3 \\ \text{unter} & x_1 + x_2 + x_3 = 3 \end{array}$$

und lösen Sie es unter Verwendung von *Lagrange-Multiplikatoren*.

Tip: Aus dem Satz über Lagrange-Multiplikatoren erhalten Sie 3 Gleichungen. Zusammen mit der Bedingung der Optimierungsaufgabe ergibt sich dann ein Gleichungssystem mit 4 Gleichungen in 4 Unbekannten, welches es zu lsen gilt.

Aufgabe 54 (12 Punkte)

Finden Sie Kandidaten für relative Minima des Problems

$$\begin{array}{rl} \min & 2x_1^2 + 2x_1x_2 + x_2^2 - 10x_1 - 10x_2 \\ \text{unter} & x_1^2 + x_2^2 \le 5 \\ & 3x_1 + x_2 \le 6 \end{array}$$

indem Sie die *Kuhn-Tucker Bedingungen* überprüfen.

Hinweis: Achten Sie darauf, daß Sie verschiedene Kombinationen von *aktiven Bedingungen* annehmen knnen. Im vorliegenden Fall sind dies entweder keine, eine oder zwei Bedingungen.

Aufgabe 55 (10 Punkte)
Betrachten Sie die beiden folgenden Flächen im $\mathbb{R}^3$:

(a) $x^2 + y^2 = z^2$,

(b) $xy + x^3 + y^3 = 0$. *(Kein Fehler, z kommt hier nicht vor!)*

Skizzieren Sie die Flächen, bestimmen Sie ihre singulären Punkte (d.h. ihre *nicht* regulären Punkte) und geben Sie die Tangentialräume in den regulären Punkten der Flächen an.

Aufgabe 56 (mdl.)
Das folgende Rätsel kennen einige von ihnen schon (hat Winfried Hochstättler nach der Vorlesung erzählt). Es ist aber so schn, daß wir es noch einmal für alle aufschreiben:

Sie haben zwei Zündschnüre und ein Feuerzeug. Beide Zündschnüre brennen jeweils genau 60 Sekunden, aber leider nicht regelmäßig. (D.h., sie brennen nicht soundsoviel Zentimeter pro Sekunde, sondern mal mehr und mal weniger. Sie wissen nur, daß nach genau 60 Sekunden die gesamte Zündschnur verbrannt ist.) Sie sollen nun mit Hilfe der beiden Zündschnüre und des Feuerzeugs einen Zeitraum von 45 Sekunden mglichst genau messen. Wie gehen Sie vor? **Aufgabe 57** (10

Punkte)
Betrachten Sie das *quadratische* Programm (Notation wie immer)

$$\begin{array}{rl} \min & \frac{1}{2}x^T Qx - b^T x \\ \text{unter} & Ax = c \end{array}$$

und überlegen Sie sich, daß ein Punkt x^* genau dann ein *lokales* Minimum ist, wenn er auch ein *globales* Minimum ist.

Aufgabe 58 (10 Punkte)
Beweisen Sie nochmals die LP-Dualitt (*Lineare Programme*) indem Sie die *Kuhn-Tucker Bedingungen* verwenden.

Aufgabe 59 (10 Punkte)
Zur Wiederholung: Untersuchen Sie folgende Funktion auf Minima

$$f(x,y,z) = 2x^2 + xy + y^2 + yz + z^2 - 6x - 7y - 8z + 9$$

indem sie die notwendigen bzw. hinreichenden Bedingungen erster und zweiter Ordnung überprüfen. (Finden Sie sogar ein *globales* Minimum?)

Aufgabe 60 (mdl.)
Das Ehepaar Feierschön organisiert eine Party, zu der sie 4 weitere Ehepaare eingeladen haben. Als die Gste eintreffen geben sich diejenigen die Hand, welche sich bisher noch nicht kannten. Alle anderen begrüßen sich einfach so. Nach der Begrüßung fragt Herr Feierschön alle anderen Anwesenden, mit wieviel Leuten Sie Hnde geschüttelt haben. Bemerkenswert ist, daß jeder eine andere Anzahl antwortet! Können Sie aus diesen Informationen ableiten, welche Zahl Frau Feierschön angegeben hat?

- Jeder kennt natürlich sich selbst und seinen Ehepartner.
- Herr Feierschön hat sich selbst *nicht* gefragt, wieviel Leuten er die Hand gegeben hat.

Aufgabe 61 (8 Punkte)
Beweisen Sie die beiden folgenden Aussagen, welche zeigen, daß die Kombination von *konvexen* Funktionen wieder *konvexe* Funktionen ergibt.

(a) Es seien f_1 und f_2 konvexe Funktionen definiert auf einer konvexen Menge S. Dann ist auch die Funktion $f_1 + f_2$ konvex auf S.

(b) Es sei f eine konvexe Funktion definiert auf einer konvexen Menge S. Dann ist auch αf konvex für jedes $\alpha \in \mathbb{R}_+$.

Aufgabe 62 (10 Punkte)
Zeigen Sie folgenden Satz:

Es sei f eine konvexe Funktion auf einer konvexen Menge S. Dann gilt für jede reelle Zahl c, daß die Menge $\Gamma_c := \{x \in S \,|\, f(x) \leq c\}$ konvex ist.

Angenommen Sie haben nun konvexe Funktionen $f_1, \ldots, f_m$ auf einer konvexen Menge S. Gilt dann auch, daß die Menge der Punkte x, welche

$$f_1(x) \leq c_1, f_2(x) \leq c_2, \ldots, f_m(x) \leq c_m \qquad (c_1, \ldots, c_m \in \mathbb{R})$$

simultan erfüllen, konvex ist?

Aufgabe 63 (12 Punkte)
Beweisen Sie folgendes klassisches Resultat über die Minimierung von konvexen Funktionen:

Es sei f eine konvexe Funktion über einer konvexen Menge S. Dann ist jedes relative Minimum von f ein globales Minimum.

Tip: berlegen Sie sich zuerst (mit Aufgabe 2), daß die Menge, in der f ihr Minimum annimmt, konvex ist. Dann lßt sich obige Aussage leicht durch einen Widerspruchsbeweis zeigen.

Aufgabe 64 (mdl.)
Whrend der Klausur zur Algorithmischen Mathematik schlendert der Assistent der Vorlesung irgendwann zwischen 9 und 13 Uhr durch den Hörsaal C des Hörsaalgebudes und kontrolliert die Personal- und Studentenausweise der Klausurteilnehmer. Nachdem er damit fertig ist, sagt er (der für sein phnomenales Gedchtnis bekannt ist! [Achtung: Satire!]): *„Schade! Niemand hat heute, also am 6.2.'99, Geburtstag. Aber mir ist aufgefallen, daß zwei von Ihnen am selben Tag Geburtstag feiern! So ein Zufall!"*

Hat er recht damit? Ist das wirklich so ein Zufall? (Konkretere Aufgabenstellung: *Wieviele Personen braucht man, damit die Wahrscheinlichkeit, daß zwei von Ihnen am selben Tag Geburtstag haben, mindestens* $1/2$ *betrgt?*)

Appendix B: Weihnachtsvorlesung

Magische Quadrate

Ein magisches Quadrat ist eine $n \times n$-Matrix A, deren Einträge die Zahlen von 1 bis n^2 sind und bei der alle Zeilen und Spaltensummen gleich sind. Folgendes magische Quadrat ist als Dürers magisches Quadrat bekannt geworden, da es im Hintergrund des Bildes Melencolia 1 zu sehen ist.

16	3	2	13
5	10	11	8
9	6	7	12
4	15	14	1

Wir wollen heute eine Methode vorstellen, mit der man magische Quadrate konstruieren kann.

Leonhard Eulers Problem mit 36 Offizieren

Leonhard Euler wurde 1707 in Basel geboren und arbeitete ab 1766 in Petersburg zur Zeit von Katherina der Großen. Es wird erzählt, daß Katherina ihm folgende Aufgabe stellte:

> Er solle 36 Offiziere aus 6 Regimentern mit in 6 verschiedenen Of-

fiziersrängen, von jedem Regiment einer jeden Ranges, so auf einem 6×6-Schachbrett anordnen, daß in jeder Zeile und in jeder Spalte je ein Offizier von jedem Regiment und von jedem Rang vorkomme.

Für ein $n \times n$ Schachbrett können wir die Frage also so verallgemeinern, daß man in einer $n \times n$-Matrix Tupel die Tupel (i, j) mit $i, j = 0, \ldots n-1$ so anordnen soll, daß in jeder Spalte und Zeile in beiden Koordinaten alle Zahlen vorkommen. Da sich an der Eigenschaft durch Permutation der Symbole und Spaltenvertauschungen nichts ändert, können wir annehmen, daß $a_{1j} = (j-1, j-1)$ und $a_{i1} = (i-1, *)$ ist. Wir nennen so eine Matrix ein *orthogonales Paar lateinischer Quadrate.*

n=1,2

Für $n = 1$ ist nichts zu tun. Sei also $a_{11} = (0, 0)$ und $a_{12} = (1, 1)$.Für a_{21} bleibt nun, da die 0en vergeben sind, auch nur $(1, 1)$. Also kann es keine solche Anordnung geben.

n=3,4

Man findet z.B. durch Ausprobieren

00	11	22
12	20	01
21	02	10

00	11	22	33
12	03	30	21
23	32	01	10
31	20	13	02

n=5

Probieren wird langsam etwas mühsam. Hier kommt ein Trick. Und zwar tragen wir in die erste Koordinate "$(i-1)+(j-1)$" ein und in die zweite "$2*(i-1)+j-1$". Wir müssen mit diesen Zahlen nur "richtig" rechnen. Bei Primzahlen ist das leicht, es genügt Modulo-Rechnung also

0+0, 2*0+0	0+1, 2*0+1	0+2, 2*0+2	0+3, 2*0+3	0+4, 2*0+4
1+0, 2*1+0	1+1, 2*1+1	1+2, 2*1+2	1+3, 2*1+3	1+4, 2*1+4
2+0, 2*2+0	2+1, 2*2+1	2+2, 2*2+2	2+3, 2*2+3	2+4, 2*2+4
3+0, 2*3+0	3+1, 2*3+1	3+2, 2*3+2	3+3, 2*3+3	3+4, 2*3+4
4+0, 2*4+0	4+1, 2*4+1	4+2, 2*4+2	4+3, 2*4+3	4+4, 2*4+4

0,0	1,1	2,2	3,3	4,4
1,2	2,3	3,4	4,0	0,1
2,4	3,0	4,1	0,2	1,3
3,1	4,2	0,3	1,4	2,0
4,3	0,4	1,0	2,1	3,2

.

Vielleicht erkennen Sie die Ähnlichkeit mit dem Fall $n = 3$. Dort haben wir genauso gerechnet. Wie ist im Falle $n = 4$? Der *Körper* mit 4 Elementen hat etwas eigenwillige Rechenregeln.

+	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

*	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	3	1
3	0	3	1	2

Fassen wir zusammen. Wenn wir auf $\mathbb{F} := \{0, 1, \dots, n-1\}$ Addition und Multiplikation invertierbar (es gibt Subtraktion und Division) und verträglich miteinander definieren können, so nennen wir $\mathbb{F}$ mit dieser Definition einen Körper. Solche Körper gibt es genau dann, wenn $n = p^m$ eine Primzahlpotenz ist.

Satz 5.5.6 *Sei $\mathbb{F} = \{0, \dots, n-1\}$ ein Körper mit $n \geq 3$ Elementen und $a \neq b \in \mathbb{F} \setminus \{0\}$. Dann definiert der Eintrag $a_{i,j} := a * (i-1) + (j-1), b * (i-1) + (j-1)$ ein orthogonales Paar lateinischer Quadrate.*

Beweis. Wir zeigen zunächst, daß in jeder Zeile und Spalte in beiden Koordinaten jedes Element c vorkommt. Betrachten wir dazu zunächst die erste Zeile in Koordinate i.

$$\begin{aligned} a * (i-1) + (j-1) &= c \\ \Leftrightarrow (j-1) &= c - a * (i-1). \end{aligned}$$

Bei vorgegebener Spalte j berechnen wir

$$\begin{aligned} a * (i-1) + (j-1) &= c \\ \Leftrightarrow a * (i-1) &= (c - (j-1)) \\ \Leftrightarrow (i-1) &= a^{-1}(c - (j-1)). \end{aligned}$$

Die zweite Koordinate berechnet man analog. Es bleibt zu zeigen, daß jedes Paar genau einmal vorkommt. Betrachten wir also das Paar (c, d), so erhalten wir aus

$$\begin{aligned}
(a * (i-1) + (j-1), b * (i-1) + (j-1)) &= (c, d) \\
(a-b)(i-1) &= c - d \qquad \text{und} \\
a * (i-1) + ab^{-1}(j-1) &= ab^{-1}d, \qquad \text{also} \\
(1 - ab^{-1})(j-1) &= (c - ab^{-1}d).
\end{aligned}$$

Da $a \neq b$ ist, sind $(a-b)$ und $(1 - ab^{-1})$ von Null verschieden und wir können i, j berechnen als

$$\begin{aligned}
(i-1) &= (a-b)^{-1}(c-d) \qquad \text{und} \\
(j-1) &= (1 - ab^{-1})^{-1}(c - ab^{-1}d).
\end{aligned}$$

$\square$

Haben wir zwei Paare orthogonaler lateinischer Quadrate der Größen n und m, so können wir leicht ein orthogonales Quadrat der Größe nm konstruieren.

Satz 5.5.7 *Sind A und B orthogonale lateinische Quadrate der Ordnungen m bzw. n, so ist $A \times B$ definiert durch*

$$(A \times B)_{in+k, jn+l} := (nA^1_{ij} + B^1_{kl}, nA^2_{ij} + B^2_{kl})$$

ein orthogonales Paar lateinischer Quadrate.

Beispiel 5.5.8 $m = 3, n = 4$

0,0	1,1	2,2	3,3	4,4	5,5	6,6	7,7	8,8	9,9	10,10	11,11
1,2	0,3	3,0	2,1	5,6	4,7	7,4	6,5	9,10	8,11	11,8	10,9
2,3	3,2	0,1	1,0	6,7	7,6	4,5	5,4	10,11	11,10	8,9	9,8
3,1	2,0	1,3	0,2	7,5	6,4	5,7	4,6	11,9	10,8	9,11	8,10
4,8	5,9	6,10	7,11	8,0	9,1	10,2	11,3	0,4	1,5	2,6	3,7
5,10	4,11	7,8	6,9	9,2	4,3	11,0	10,1	1,6	0,7	3, 4	2, 5
6,11	7,10	4,9	5,8	10,3	11,2	8,1	9,0	2,7	3,6	0, 5	1, 4
7,9	6,8	5,11	4,10	3,9	2,8	1,11	0,10	7,1	6,0	5, 3	4, 2
8,4	9,5	10,6	11,7	0,8	1,9	2,10	3,11	4,0	5,1	6,2	7,3
9,6	8,7	11,4	10,5	1,10	0,11	3,8	2,9	5,2	4,3	7, 0	6,1
10,7	11,6	8,5	9,4	2,11	3,10	0,9	1,8	6,3	7,2	0,5	5,0
11,5	10,4	11,3	8,6	3,9	2,8	1,11	0,10	7,1	6,0	5,3	4,2

Die letzten beide Sätze ergeben kombiniert:

Korollar 5.5.9 *Ist $n \not\equiv 2\,mod\,4$ so gibt es ein Paar orthogonaler lateinischer Quadrate der Größe n.*

Beweis. Wir führen Induktion über die Anzahl der Primzahlen, die n teilen. Sei also p ein Primzahlen, die n teilt und p^r die höchste Potenz, in der p in n vorkommt. Nach Satz 5.5.6 gibt es ein Paar für p^r. Nun ist $m := \frac{n}{q^r}$ eine natürliche Zahl mit $m \not\equiv 2\,mod\,4$ und einem Primteiler weniger als n. Nach Induktionsvoraussetzung gibt es ein Paar für m also nach Satz 5.5.7 auch für n. □

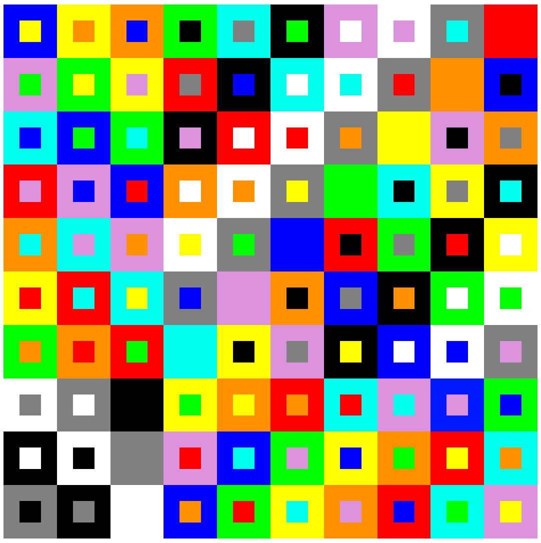
Frage: Wo ist in dem Beweis die Induktionsverankerung geblieben?

Wie steht es mit Eulers Problem, also $n = 6$? Herr Euler vermutete, daß 2 und 6 schon schiefgehen und der Rest relativ einfach ist, daß $n \equiv 2\,mod\,4$ nicht geht. Für $n = 6$ zitieren wir:

> Surprisingly there is no such solution. Whether he was asked by Catherine or not, Euler did consider the problem in 1779 and convinced himself that it was impossible. Euler was capable of monumental calculations, but it is not clear that he had really examined all the cases. This was systematically done in 1900 by G. Tarry. Today a computer can do this easily.
>
> *J.H. van Lint & R.M. Wilson "A Course in Combinatorics"*

Zum letzten Satz fällt mir nicht viel ein. Es mag sein, daß die Fallunterscheidungen von einem Computer leicht ausgeixt werden können, der Programmierer aber dennoch etwas Zeit und Intelligenz investieren muß.

Allerdings lag Herr Euler völlig schief. 2 und 6 sind, wie man inzwischen weiß, die einzigen Ausnahmen. Die folgende Grafik zeigt $n = 10$.



Magische Quadrate

Eigentlich sind orthogonale Paare lateinischer Quadrate schon magisch, wenn man die Einträge nur richtig liest, nämlich als zweistellige Ziffern im n-aren System. Offensichtlich hat dann jede Zeile und Spalte genau die Summe

$$(n+1)(\sum_{i=0}^{n-1} i).$$

Die Transformation zu der am Anfang gewünschten Gestalt ist völlig einfach. Der Eintrag (i, j) wird ersetzt durch $ni + j + 1$. Wir erhalten also für $n = 5$ das magische Quadrat

1	7	13	19	25
8	14	20	21	2
15	16	22	3	9
17	23	4	10	11
24	5	6	12	18

Man hat so ein Konstruktionsverfahren, das für alle $n \in \mathbb{N} \setminus \{2, 6\}$ funktioniert. Das heißt aber nicht, daß es keinemagischen Quadrate der Größe 6 gäbe. Schon

Dürers Quadrat hat einen anderen Typ als die hier vorgestellten und auch für $n = 6$ gibt es magische Quadrate.

32	29	4	1	2	21
30	31	2	3	22	23
12	9	17	20	28	25
10	11	18	19	26	27
13	16	36	33	5	8
14	15	34	35	6	7

Dieses hat wie auch schon Dürers Beispiel den zusätzlichen Vorteil, daß die magische Summe auch auf der Hauptdiagonalen (bei Dürer auch auf der Nebendiagonalen) angenommen wird, was von manchem Spielverderber schon in der Definition gefordert wird, bei uns im Falle $n = 5$ aber nur zufällig erreicht wurde.

Appendix C: Pathologisches Beispiel zu Satz 4.2.8

Zum Abschluß wollen wir noch ein Beispiel angeben, das zeigt, daß es nicht selbstverständlich ist, daß ein Punkt, der in allen Richtungen ein lokales Minimum ist, auch insgesamt ein lokales Minimum ist. Um dies zu erreichen müssen wir uns allerding in einen unendlich dimensionalen Vektorraum begeben. Sei

$$V := \left\{ (a_1, a_2, \ldots) \mid \sum_{i=1}^{\infty} |a_i| < \infty \right\}$$

der Vektorraum der absolut konvergenten Reihen. Dann ist die Zweinorm gegeben durch

$$\|(a_1, a_2, \ldots)\|_2 = \sqrt{\sum_{i=1}^{\infty} a_i^2} < \sum_{i=1}^{\infty} |a_i| < \infty.$$

Von den Axiomen einer Norm ist nur die Dreiecksungleichung nicht sofort klar. Hierfür berechnen wir:

$$\begin{aligned}
\left(\sqrt{\sum_{i=1}^{\infty} a_i^2} + \sqrt{\sum_{i=1}^{\infty} b_i^2} \right)^2 &= \sum_{i=1}^{\infty} a_i^2 + \sum_{i=1}^{\infty} b_i^2 + 2\sqrt{\left(\sum_{i=1}^{\infty} a_i^2\right)\left(\sum_{j=1}^{\infty} b_j^2\right)} \\
&\overset{\sqrt{}\text{konkav}}{\geq} \sum_{i=1}^{\infty} a_i^2 + \sum_{i=1}^{\infty} b_i^2 + 2\left(\sum_{i,j=1}^{\infty} a_i b_j\right) \\
&= \left(\sum_{i=1}^{\infty} a_i + b_i\right)^2 .
\end{aligned}$$

Für alle $k \in \mathbb{N}$ definieren wir

$$f_k(a) = \frac{a_k^2}{k^2} - a_k^4.$$

Sei nun $f : V \to \mathbb{R}$ definiert durch

$$f(a) = \sum_{i=1}^{\infty} f_i(a).$$

Mit a und $\sum_{i=1}^{\infty} \frac{1}{i^2}$ ist dann auch $f(a)$ konvergent. Man überzeuge sich, daß f auch stetig ist.

Wir behaupten nun 0 ist ein relatives Minimum der Funktion $f(td)$ für jede Richtung $d = (d_1, d_2, \ldots)$, aber kein relatives Minimum von f.

Sei also d vorgegeben.

$$\begin{aligned} f(0 + td) &= \sum_{i=1}^{\infty} \frac{t^2 d_i^2}{i^2} - t^4 d_i^4 \\ &= t^2 \left(\left(\sum_{i=1}^{\infty} \frac{d_i^2}{i^2} \right) - t^2 \sum_{i=1}^{\infty} d_i^4 \right) > 0, \end{aligned}$$

für $|t|$ klein genug. Wegen $f(0) = 0$ ist 0 also ein lokales Minimum von $f(td)$.

Sei nun $\varepsilon > 0$ vorgegeben und $k > \frac{2}{\varepsilon}$. Sei $a = (0, 0, \ldots, \frac{2}{k}, 0, \ldots)$, wobei der Nichtnulleintrag an k-ter Stelle liegt. Dann ist $\|a\| < \varepsilon$, also $a \in U_\varepsilon(a)$ und

$$f(a) = \frac{4}{k^4} - \frac{16}{k^4} < 0 = f(0).$$

Also ist 0 kein lokales Minimum von f.

Index

$A_{\cdot i}, A_{j\cdot}$, 5
2^M, 5
$C^k(U)$, 59
$M \triangle N$, 5
$M+N, M-N, \alpha N, M^\perp$, 5
Q-konjugiert, 89
Q-orthogonal, 89
$R^{m\times n}$, 5
$U_\varepsilon(P)$, 6
$\forall, \exists$, 4
$\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}$, 4
e_i, 5

zulassige Richtung, 61

Abstiegsrichtung, 61
affine Hülle, 38
affine Kombination, 38
al-Khwarizmi, 3
Algorithmus, 3

Basis, 42
Basislösung, 42
Basiswechsel, 46
Berechenbarkeit, 13
Bland's rule, 47

Datenfehler, 10
duales Programm, 39

Ecke, 42
Eigenvektor, 21
Eigenwert, 21
entartete Ecke, 47
entarteter Pivot, 47

Fehler, absoluter, 9
Fehler, relativer, 9
Fibonaccisuche, 77
Fletcher-Reeves, 93
Frobeniusmatrix, 18

Gauseliminationsschritt, 15
Gauß-Jordan-Elimination, 20
Gaußelimination, 15
Gleitkommazahlen, 10
Goldener Schnitt-Suche, 80
größter Fortschritt, 47
Gradient, 59

Hauptachsentransformation, 22
hermitesch, 21
Hessematrix, 59

Jacobische, 59

Komplexität, 11
Kondition, 29
konische Hülle, 38
konische Kombination, 38
konjugiert, 89
konvex, 74
konvexe Hülle, 37, 38
Konvexkombination, 38
Koordinatenabstiegsmethode, 81
Kurve, 57

Landau-Symbole, 12
Laufzeitverhalten, 11
lineare Hülle, 38
Lineares Optimierungsproblem, 33

Linearkombination, 38

Mannigfaltigkeit, 65
Mantisse, 10
Maschinengenauigkeit, 10
Maschinenzahlen, 10
Minimum, globales, 56
Minimum, lokales, 56
Minimum, relatives, 56

Nichtbasis, 42
Norm, 26
normierter Vektorraum, 26

obere Dreiecksmatrix, 16
orthogonal, 21

partielle Ableitung, 59
Permutationsmatrix, 17
Phase I, 51
Pivotelement, 15
Pivotregel, 46
Pivotsuche, 20
Polyeder, 35
positiv semidefinit, 64

quadratisches Problem, 74

reduzierte Kosten, 43
Richtungsableitung, 59
runden, 9

Simplexalgorithmus, 45
Simplextableau, 46
Spacer step, 93
Spaltensummennorm, 27
Spektralnorm, 27
Standardform eines Linearen Optimierungsproblems, 33
Startbasis, 45, 46
steilster Anstieg, 47
stetig differenzierbar, 59
strikt konvex, 74
strikt unimodal, 74

Tangentialvektor, 57

unimodal, 74
untere Dreiecksmatrix, 16

Verfahrensfehler, 10

Weg, 57
worst-case Komplexitat, 11

Zeilensummennorm, 27
zulässige Basis, 42
zulässige Basislösung, 42
zyklisches Abstiegsverfahren, 82