14 Lineare Optimierung

Bei diesem Abschnitt handelt es sich im wesentlichen um das Kapitel 6 aus Opfer [1994] unter Weglassung der Aufgaben.

14.1 Aufgabenstellung

Bereits in der Schule werden Aufgaben behandelt, bei denen es darum geht, Maximal- oder Minimalstellen gewisser Funktionen zu finden. Entweder soll einfach nur ein Extremum (oder alle Extrema) gefunden werden, z. B. das Maximum von $f(x) = 10 + 3x - 7x^2 + 5x^3$, oder es sind zusätzlich noch Nebenbedingungen zu beachten. Ein typisches Beispiel dafür ist die Bestimmung der Abmessungen einer Konservendose, die bei vorgebenem Inhalt die kleinste Oberfläche hat. Aufgaben dieses Typs heißen *Optimierungsaufgaben*. In diesem Kapitel soll es um eine spezielle Aufgabenform gehen. Wir wollen Optimierungsaufgaben untersuchen, bei denen die zu optimierende Funktion linear von den unbekannten Größen abhängt und bei denen alle Nebenbedingungen als lineare Gleichungen oder Ungleichungen geschrieben werden können. Derartige Aufgaben heißen *lineare Optimierungsaufgaben*. Aufgaben dieses Typs sind bei Benutzung von Rechenanlagen verhältnismäßig einfach zu lösen und gleichzeitig – etwa bei der Planung wirtschaftlicher Aktivitäten in einem Betrieb – weitverbreitet. Typischerweise haben diese Aufgaben eine große Anzahl von Unbekannten und, um die Realität gut zu erfassen, auch eine große Anzahl von einzuhaltenden Nebenbedingungen. Daher spielen für derartige Aufgaben Computer-orientierte Lösungstechniken eine große Rolle.

Beispiel 14.1. *Ernährungsplan*
Bei dem vorzustellenden Problem handelt es sich vermutlich um das älteste und am meisten strapazierte lineare Optimierungsproblem in der mathematischen Literatur (Stigler [1945]). Dieses Problem wird wegen einer mangelhaften Übersetzung von englisch *diet* in deutsch *Diät* in der deutschen Literatur mißverständlich meistens als *Diätproblem* eingeführt: Aus einem Angebot von n *Nahrungsmitteln* (Brot, Fleisch, Gemüse, Obst, ...) ist ein Essen zusammenzustellen, das möglichst billig ist, trotzdem aber Mindestgehalte bestimmter m *Grundsubstanzen* (Eiweiß, Fett, Kohlehydrate, Vitamine, Mineralstoffe, ...) enthält. Ein konkretes Beispiel findet man in Varga [1991, S. 98–100].

Diese Aufgabe läßt sich folgendermaßen mathematisch formulieren: Bezeichne für $i = 1, 2, \ldots, m; j = 1, 2, \ldots, n$

x_j: die unbekannte zu verwendende Menge vom j-ten Nahrungsmittel,

a_{ij}: die Menge der i-ten Grundsubstanz im j-ten Nahrungsmittel,

b_i: die Menge der i-ten Grundsubstanz, die das Menü mindestens enthalten muß,

c_j: den Preis für das j-te Nahrungsmittel,

wobei sich alle Angaben auf eine feste Mengeneinheit beziehen. Dann lautet die Aufgabe: Man finde Zahlen $x_1, x_2, \ldots, x_n$, so daß die Gesamtkosten $S := \sum_{j=1}^{n} c_j x_j$ minimal ausfallen unter Einhaltung der *Nebenbedingungen*

$$\begin{array}{rll} \sum_{j=1}^n a_{ij}x_j & \geq b_i & \text{für} \quad i = 1, \ldots, m, \\ x_j & \geq 0 & \text{für} \quad j = 1, \ldots, n. \end{array}$$

In diesem Fall hat die Optimierungsaufgabe also die formale Gestalt: Finde ein $\mathbf{x} \in \mathbb{R}^n$ mit

$$\left.\begin{array}{lrcl} & \mathbf{c}^{\mathrm{T}}\mathbf{x} & \stackrel{!}{=} & \min \\ \text{unter} & \mathbf{Ax} & \geq & \mathbf{b}, \\ & \mathbf{x} & \geq & \mathbf{0}, \end{array}\right\} (**)$$

wobei $\mathbf{c} \in \mathbb{R}^n,\ \mathbf{b} \in \mathbb{R}^m,\ \mathbf{A} \in \mathbb{R}^{m \times n}$ gegeben sind und das Zeichen $\geq$ zwischen Vektoren $\mathbf{y}$, $\mathbf{z}$ bedeutet: $\mathbf{z} \geq \mathbf{y} \iff z_j \geq y_j$ für alle j. Das Zeichen $\stackrel{!}{=}$ min bedeutet, daß der links davon stehende Ausdruck minimiert werden soll.

Bei einem beliebigen Optimierungsproblem heißt die zu minimierende (oder zu maximierende) Funktion *Zielfunktion*, hier also $\mathbf{c}^{\mathrm{T}}\mathbf{x}$. Jeder Vektor, der alle Nebenbedingungen einhält, heißt *zulässiger Vektor*, in $(**)$ also jedes $\mathbf{x} \in \mathbb{R}^n$ mit $\mathbf{Ax} \geq \mathbf{b}$ und $\mathbf{x} \geq \mathbf{0}$. Die Menge aller zulässigen Vektoren heißt *zulässige Menge* oder *zulässiger Bereich* der Optimierungsaufgabe. Ein zulässiges $\mathbf{x}$ heißt *optimal*, wenn $\mathbf{c}^{\mathrm{T}}\mathbf{x} \leq \mathbf{c}^{\mathrm{T}}\mathbf{y}$ für alle zulässigen Vektoren $\mathbf{y} \in \mathbb{R}^n$ gilt. Jedes optimale $\mathbf{x}$ ist also eine *Lösung* der Aufgabe $(**)$.

Je nach Aufgabenstellung kann man auch auf andere Schreibweisen kommen. Da es unser Ziel ist, ein Lösungsverfahren für alle linearen Optimierungsaufgaben zu finden, soll hier kurz zusammengestellt werden, wie sich die verschiedenen Schreibweisen ineinander überführen lassen:

i) Eine Maximierungsaufgabe wird zu einer Minimierungsaufgabe durch Übergang zum Negativen der Zielfunktion:

$$\mathbf{c}^{\mathrm{T}}\mathbf{x} = \max \quad \iff \quad -\mathbf{c}^{\mathrm{T}}\mathbf{x} = \min$$

(beidesmal unter denselben Nebenbedingungen)

ii) Eine Ungleichung

$$a_{i1}x_1 + \cdots + a_{in}x_n \geq b_i$$

innerhalb der Nebenbedingungen kann durch Einführen einer *Schlupfvariablen* $y_i \geq 0$ in die Gleichung

$$a_{i1}x_1 + \cdots + a_{in}x_n - y_i = b_i$$

überführt werden.

iii) Eine Gleichung

$$a_{i1}x_1 + \cdots + a_{in}x_n = b_i$$

innerhalb der Nebenbedingungen kann durch zwei Ungleichungen

$$a_{i1}x_1 + \cdots + a_{in}x_n \geq b_i,$$

$$-a_{i1}x_1 - \cdots - a_{in}x_n \geq -b_i$$

ersetzt werden.

iv) Eine Komponente x_j von $\mathbf{x}$, für die keine Vorzeichenbedingung besteht, kann durch den Ausdruck $x_j = x_j^+ - x_j^-$ mit $x_j^+, x_j^- \geq 0$ ersetzt werden.

v) Eine „$\leq$“-Ungleichung kann durch Multiplikation mit -1 in eine „$\geq$“-Ungleichung überführt werden.

Jede Optimierungsaufgabe mit linearer Zielfunktion und linearen Nebenbedingungen kann also auf die folgende Form gebracht werden:

$$\left.\begin{array}{lrcl} & \mathbf{c}^{\mathrm{T}}\mathbf{x} & \stackrel{!}{=} & \min \\ \text{unter} & \mathbf{A}\mathbf{x} & = & \mathbf{b}, \\ & \mathbf{x} & \geq & \mathbf{0}. \end{array}\right\} (*)$$

In dieser Form wollen wir die Aufgabe jetzt weiter untersuchen. Die Form $(*)$ heißt meistens *kanonische Form*, die Form $(**)$ *Standardform* der *linearen Optimierungsaufgabe.*

Hinweis: Die Überführung einer vorgelegten Aufgabe in eine andere Form kann eine Vergrößerung der Anzahl der Nebenbedingungen (bei iii)) oder eine Vergrößerung der Komponentenzahl von $\mathbf{x}$ (bei ii) und iv)) mit sich bringen. Im letzteren Fall muß auch der Vektor $\mathbf{c}$ entsprechend mehr Komponenten erhalten. Für jede Schlupfvariable erhält $\mathbf{c}$ eine Komponente mit dem Wert Null; bei iv) tritt x_j^+ an die Stelle von x_j und für x_j^- erhält $\mathbf{c}$ eine Komponente mit dem Wert $-c_j$.

Beispiel 14.2. *Umwandlung von Nebenbedingungen*
Die Aufgabe

$$\begin{array}{lrrcl} & -2x_1 & +3x_2 & \stackrel{!}{=} & \max \\ \text{unter} & x_1 & +x_2 & \geq & 5, \\ & -x_1 & +x_2 & \leq & 7, \\ & x_1 & & \leq & 10, \\ & x_1 & & & \text{nicht vorzeichenbeschränkt}, \\ & & x_2 & \geq & 0, \end{array}$$

wird so zu

$$\begin{array}{lrcrcrcrrrcr} & 2x_1^+ & - & 2x_1^- & - & 3x_2 & + & 0y_1 + & 0y_2 & +0y_3 & \stackrel{!}{=} & \min \\ \text{unter} & x_1^+ & - & x_1^- & + & x_2 & - & y_1 & & & = & 5, \\ & x_1^+ & - & x_1^- & - & x_2 & & & -y_2 & & = & -7, \\ & -x_1^+ & + & x_1^- & & & & & & -y_3 & = & -10, \\ & x_1^+ & & & & & & & & & \geq & 0, \\ & & & x_1^- & & & & & & & \geq & 0, \\ & & & & & x_2 & & & & & \geq & 0, \\ & & & & & & & y_1 & & & \geq & 0, \\ & & & & & & & & y_2 & & \geq & 0, \\ & & & & & & & & & y_3 & \geq & 0. \end{array}$$

Probleme in Standardform mit nur zwei Unbekannten x_1, x_2 wie in Beispiel 14.2 angegeben, lassen sich auf einfache Weise graphisch lösen. Jede Ungleichung beschreibt eine Halbebene im $\mathbb{R}^2$, alle Ungleichungen zusammen beschreiben daher den zulässigen Bereich (wenn er nicht leer ist) als eine konvexe, geradlinig berandete Menge im $\mathbb{R}^2$. Ist $\mathbf{c}^{\mathrm{T}}\mathbf{x}$ die Zielfunktion, so sind $\mathbf{c}^{\mathrm{T}}\mathbf{x} = \mathrm{const}$ Geraden, die für verschiedene Konstanten alle parallel sind. Durch Parallelverschiebung dieser Geraden über dem gezeichneten zulässigen Bereich solange bis nur ein Randpunkt oder oder ein geradliniges Stück der Berandung auf der Geraden liegt, kann man die Lösung finden. Diese Lösungstechnik, die nur für Probleme mit zwei Variablen angewendet werden kann, heißt *graphische Methode* zur Lösung des Optimierungsproblems.

14.2 Basisvektoren

Wir untersuchen von jetzt an die folgende Aufgabe: Man finde einen Vektor $\mathbf{x} \in \mathbb{R}^n$, so daß bei gegebenen $\mathbf{c} \in \mathbb{R}^n$, $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{A} \in \mathbb{R}^{m\times n}$

$$\left.\begin{array}{lrcl} & \mathbf{c}^{\mathrm{T}}\mathbf{x} & \stackrel{!}{=} & \min \\ \text{unter} \qquad & \mathbf{A}\mathbf{x} & = & \mathbf{b}, \\ & \mathbf{x} & \geq & \mathbf{0}. \end{array}\right\} (*)$$

Wir treffen die folgenden generellen Voraussetzungen.

Voraussetzungen 14.3. Für die Matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ gelte $m < n$ und $\mathrm{Rang}\mathbf{A} = m$.

Diese Voraussetzungen über $\mathbf{A}$ stellen sicher, daß das Gleichungssystem $\mathbf{A}\mathbf{x}{=}\mathbf{b}$ keine überzähligen Gleichungen enthält und lösbar ist. Die Voraussetzungen werden für die theoretischen Untersuchungen gebraucht. Tatsächlich „merkt“ der zu entwickelnde Algorithmus, ob die Voraussetzung Rang $\mathbf{A} = m$ erfüllt ist. Ein wesentlicher Schritt zur Lösung des gestellten Problems ist die Tatsache, daß der Lösungsvektor nur noch in einer endlichen Menge von zulässigen Vektoren gesucht werden muß. Dazu benötigen wir die auf den ersten Blick etwas unanschaulich aussehende Definition.

Definition 14.4. Seien die Voraussetzungen 14.3 erfüllt. Eine m-elementige Indexmenge $I \subset \{1, 2, \ldots, n\}$ heißt *Basisindexmenge*, wenn die Spalten $\mathbf{a}^j$ von $\mathbf{A}$ mit den Nummern $j \in I$ eine Basis des $\mathbb{R}^m$ bilden, also linear unabhängig sind. Ein zulässiger Vektor $\mathbf{x} = (x_1, x_2, \ldots, x_n)^{\mathrm{T}}$ (der also $\mathbf{A}\mathbf{x} = \mathbf{b}$ und $\mathbf{x} \geq \mathbf{0}$ erfüllt) heißt *Basisvektor*, wenn eine Basisindexmenge I existiert mit $x_j = 0$ für $j \notin I$.

Ein zulässiger Vektor $\mathbf{x}$ ist z. B. ein Basisvektor, wenn die zu positiven Komponenten von $\mathbf{x}$ gehörenden Spalten von $\mathbf{A}$ linear unabhängig sind. Wie man dann die dazugehörige Basisindexmenge konstruiert, wird am Beginn des Beweises zu Satz 14.5 näher ausgeführt. Zu einer Basisindexmenge gibt es also höchstens einen Basisvektor. Geometrisch handelt es sich bei den Basisvektoren um die *Ecken* des zulässigen Bereichs. Wir wollen diese Deutung aber um kurz zu bleiben, nicht weiter verfolgen. Die Wichtigkeit des neuen Begriffs *Basisvektor* kommt im folgenden Satz zum Ausdruck.

Satz 14.5. Sei unter den Voraussetzungen 14.3 das Problem $(*)$ lösbar. Dann gibt es einen Basisvektor, der $(*)$ löst.

Beweis: Sei $\mathbf{x}^* = (x_1^*, x_2^*, \ldots, x_n^*)^{\mathrm{T}}$ eine Lösung von $(*)$ und $I := \{i : x_i^* > 0\}$. Mit $\mathbf{a}^j$, $j = 1, 2, \ldots, n$ bezeichnen wir die Spaltenvektoren der Matrix $\mathbf{A}$. Fall a: Seien die Spalten $\mathbf{a}^j$, $j \in I$ linear unabhängig. Ist $\#I = m$,[1] so ist $\mathbf{x}^*$ bereits Basisvektor zur Basisindexmenge I. Sind die Spalten unabhängig, aber $\#I < m$, so kann man (nach einem Satz aus der linearen Algebra) wegen Rang $\mathbf{A} = m$ die unabhängigen Spalten $\mathbf{a}^j$, $j \in I$ zu einer Basis (aus m Spalten) ergänzen. Vergrößert man die Indexmenge I entsprechend, so ist $\mathbf{x}^*$ Basisvektor zu dieser vergrößerten Basisindexmenge. In diesen Überlegungen ist auch der Fall $I = \emptyset$, d. h. $x_i^* = 0$ für alle $i \in \{1, 2, \ldots, n\}$ eingeschlossen. In diesem besonderen Fall definiert jede Basisindexmenge einen Basisvektor. Fall b: Seien die Spalten $\mathbf{a}^j$, $j \in I$ linear abhängig. Dann hat $\mathbf{Ay} = \mathbf{0}$ eine Lösung mit $y_j = 0$ für alle $j \notin I$ und $y_i \neq 0$ für mindestens ein $i \in I$. D. h., die Zahl $\mu := \min_{i \in I, y_i \neq 0} x_i^*/|y_i|$ ist positiv und aus ihrer Definition folgt $x_i - \mu|y_i| \geq 0$ für alle $i \in I$ (sogar für alle $i \in \{1, 2, \ldots, n\}$). Für $y_i > 0$ folgt daraus $x_i^* + \mu y_i \geq x_i^* - \mu y_i \geq 0$, für $y_i < 0$ folgt daraus $x_i^* - \mu y_i \geq x_i^* + \mu y_i = x_i^* - \mu|y_i| \geq 0$. Setzen wir $\mathbf{u}^{(1)} = \mathbf{x}^* + \mu\mathbf{y}$, $\mathbf{u}^{(2)} = \mathbf{x}^* - \mu\mathbf{y}$, so gilt $\mathbf{u}^{(1)} \geq \mathbf{0}$, $\mathbf{u}^{(2)} \geq \mathbf{0}$, $\mathbf{Au}^{(1)} = \mathbf{Ax}^* + \mu\mathbf{Ay} = \mathbf{Ax}^* = \mathbf{b}$, $\mathbf{Au}^{(2)} = \mathbf{Ax}^* - \mu\mathbf{Ay} = \mathbf{b}$, und beide Vektoren sind also zulässig. Weiter ist $\mathbf{c}^{\mathrm{T}}\mathbf{u}^{(1)} = \mathbf{c}^{\mathrm{T}}\mathbf{x}^* + \mu\mathbf{c}^{\mathrm{T}}\mathbf{y} \geq \mathbf{c}^{\mathrm{T}}\mathbf{x}^*$ und $\mathbf{c}^{\mathrm{T}}\mathbf{u}^{(2)} = \mathbf{c}^{\mathrm{T}}\mathbf{x}^* - \mu\mathbf{c}^{\mathrm{T}}\mathbf{y} \geq \mathbf{c}^{\mathrm{T}}\mathbf{x}^*$ wegen der Minimalität von $\mathbf{c}^{\mathrm{T}}\mathbf{x}^*$. Wir erhalten $\mathbf{c}^{\mathrm{T}}\mathbf{y} \geq 0$ und $-\mathbf{c}^{\mathrm{T}}\mathbf{y} \geq 0$ und daher $\mathbf{c}^{\mathrm{T}}\mathbf{y} = 0$. D. h., $\mathbf{u}^{(1)}$ und $\mathbf{u}^{(2)}$ sind auch Lösungen des Ausgangsproblems. Einer der Vektoren $\mathbf{u}^{(1)}$, $\mathbf{u}^{(2)}$ hat mehr Nullkomponenten als $\mathbf{x}^*$. Mit diesem Vektor wiederholen wir die obige Schlußweise und wir erhalten schließlich einen Basisvektor als Lösung. □

Es gibt genau $\binom{n}{m} := \dfrac{n!}{m!(n-m)!}$ Basisindexmengen. Da jede Basisindexmenge aber höchstens einen Basisvektor definiert, ist die Anzahl der Basisvektoren ebenfalls durch $\binom{n}{m}$ beschränkt. Indem wir willkürlich $n - m$ Komponenten von $\mathbf{x}$ Null setzen und die restlichen in eindeutiger Weise aus dem Gleichungssystem $\mathbf{Ax}{=}\mathbf{b}$ berechnen, könnten wir im Prinzip alle Lösungskandidaten und damit nach Satz 14.5 auch die Lösung des gestellten Problems bestimmen. Die Anzahl der zu bestimmenden Vektoren ist aber fast immer prohibitiv zu groß. Daher benötigen wir ein systematisches Verfahren, z. B ein solches das nur Basisvektoren in Betracht zieht, die den Wert der Zielfunktion verkleinern.

14.3 Das Simplexverfahren

Angenommen, wir kennen einen Basisvektor $\mathbf{x}^{(0)}$ der vorgegebenen linearen Optimierungsaufgabe $(*)$. Die entsprechende m-elementige dazugehörende Basisindexmenge sei I. Von George Dantzig stammt ein Verfahren (ca. 1951), hieraus einen Basisvektor $\mathbf{x}^{(1)}$ zu berechnen, für den $\mathbf{c}^{\mathrm{T}}\mathbf{x}^{(0)} \geq \mathbf{c}^{\mathrm{T}}\mathbf{x}^{(1)}$ gilt. Eine neue (dann zu $\mathbf{x}^{(1)}$ gehörende) Basis wollen wir dadurch finden, daß wir eine bisher nicht zur Basis gehörende Spalte von $\mathbf{A}$ gegen eine Spalte aus der Basis austauschen. Entsprechend muß auch der Austausch in der Basisindexmenge I vorgenommen werden. Das Auffinden eines Basisvektors $\mathbf{x}^{(0)}$ kann durch Lösen eines Hilfsproblems erfolgen. Das besprechen wir im nächsten Abschnitt.

Mit $\mathbf{a}^j$, $j = 1, 2, \ldots, n$ bezeichnen wir nach wie vor die Spalten von $\mathbf{A}$. Sei $\overline{\mathbf{x}}$ irgendein zulässiger Vektor. Wie können wir die Zielfunktionswerte $\mathbf{c}^{\mathrm{T}}\overline{\mathbf{x}}$ und $\mathbf{c}^{\mathrm{T}}\mathbf{x}^{(0)}$ vergleichen?

[1] Das Zeichen $\#\{\cdots\}$ heißt Anzahl der Elemente von $\{\cdots\}$.

Jeder der Spaltenvektoren $\mathbf{a}^j$ von $\mathbf{A}$ läßt sich eindeutig als Linearkombination

$$\mathbf{a}^j = \sum_{i\in I} d_{ij}\mathbf{a}^i, \quad j = 1, 2, \ldots, n \tag{14.1}$$

der Spaltenvektoren aus der Basis darstellen. Für $j \in I$ ist daher insbesondere[3]

$$d_{ij} = \delta_{ij} = \begin{cases} 1 & \text{für } i = j, \\ 0 & \text{sonst.} \end{cases} \tag{14.2}$$

Da sowohl $\mathbf{x}^{(0)}$ als auch $\overline{\mathbf{x}}$ zulässig sind, gilt:

$$\sum_{i\in I} x_i^{(0)}\mathbf{a}^i = \mathbf{b} = \sum_{j=1}^{n} \overline{x}_j\ \mathbf{a}^j = \sum_{j=1}^{n} \overline{x}_j\ (\sum_{i\in I} d_{ij}\mathbf{a}^i) = \sum_{i\in I}(\sum_{j=1}^{n} d_{ij}\overline{x}_j\ )\mathbf{a}^i.$$

Da die $\mathbf{a}^i, i \in I$ linear unabhängig sind, ist die Darstellung von $\mathbf{b}$ eindeutig, also ist:

$$\begin{aligned} \text{für alle } i \in I: \quad x_i^{(0)} &= \sum_{j=1}^{n} d_{ij}\overline{x}_j = \sum_{j\in I} d_{ij}\overline{x}_j + \sum_{j\notin I} d_{ij}\overline{x}_j\ . \\ \text{Also gilt für } i \in I: \quad x_i^{(0)} &= \overline{x}_i + \sum_{j\notin I} d_{ij}\overline{x}_j\ , \\ \text{oder} \qquad \overline{x}_i &= x_i^{(0)} - \sum_{j\notin I} d_{ij}\overline{x}_j\ . \end{aligned}$$

Für den Zielfunktionswert $\mathbf{c}^{\mathrm{T}}\overline{\mathbf{x}}$ gilt dann:

$$\begin{aligned} \mathbf{c}^{\mathrm{T}}\overline{\mathbf{x}} = \sum_{j=1}^{n} c_j\overline{x}_j &= \sum_{i\in I} c_i\overline{x}_i + \sum_{j\notin I} c_j\overline{x}_j \\ &= \sum_{i\in I} c_i(x_i^{(0)} - \sum_{j\notin I} d_{ij}\overline{x}_j) + \sum_{j\notin I} c_j\overline{x}_j \\ &= \sum_{i\in I} c_i x_i^{(0)} + \sum_{j\notin I} \overline{x}_j\ (c_j - \sum_{i\in I} d_{ij}c_i) \\ &= \mathbf{c}^{\mathrm{T}}\mathbf{x}^{(0)} + \sum_{j\notin I} \overline{x}_j\ (c_j - \sum_{i\in I} d_{ij}c_i). \end{aligned} \tag{14.3}$$

Wir setzen

$$s_j = \sum_{i\in I} d_{ij}c_i \text{ und } t_j = c_j - s_j \quad \text{für alle } j = 1, 2, \ldots, n. \tag{14.4}$$

Satz 14.6. a) Genau dann ist $\mathbf{c}^{\mathrm{T}}\overline{\mathbf{x}}\ < \mathbf{c}^{\mathrm{T}}\mathbf{x}^{(0)}$, wenn $\sum_{j\notin I} t_j\overline{x}_j\ < 0$. b) Ist $t_j \geq 0$ für alle $j \notin I$, so ist $\mathbf{x}^{(0)}$ Lösung von $(*)$.

Beweis: a) folgt direkt aus (14.3) in der Form $\mathbf{c}^{\mathrm{T}}\overline{\mathbf{x}}\ = \mathbf{c}^{\mathrm{T}}\mathbf{x}^{(0)} + \sum_{j\notin I} t_j\overline{x}_j$. b) Unter den getroffenen Voraussetzungen ist $\sum_{j\notin I} t_j\overline{x}_j\ \geq 0$, also $\mathbf{c}^{\mathrm{T}}\overline{\mathbf{x}}\ \geq \mathbf{c}^{\mathrm{T}}\mathbf{x}^{(0)}$ für jedes zulässige $\overline{\mathbf{x}}$. Also ist $\mathbf{x}^{(0)}$ Lösung. □

[3] Die hier in (14.2) eingeführte Größe δ_{ij} heißt auch *Kronecker-Symbol*.

Ist also $\mathbf{x}^{(0)}$ keine Lösung, so existiert ein $r \notin I$ mit $t_r < 0$.

Satz 14.7. Existiere ein Index $r \notin I$ mit $t_r < 0$. a) Für jedes $\delta \in \mathbb{R}$ löst $\mathbf{x}(\delta) \in \mathbb{R}^n$ mit

$$x_j(\delta) = \begin{cases} x_j^{(0)} - \delta d_{jr} & \text{für } j \in I, \\ \delta & \text{für } j = r, \\ 0 & \text{sonst,} \end{cases} \tag{14.5}$$

das Gleichungssystem $\mathbf{A}\mathbf{x}(\delta) = \mathbf{b}$ und $\mathbf{c}^{\mathrm{T}}\mathbf{x}(\delta) = \mathbf{c}^{\mathrm{T}}\mathbf{x}^{(0)} + \delta t_r$. b) Ist $d_{ir} \le 0$ für alle $i \in I$, so hat die Aufgabe $(*)$ keine Lösung.

Beweis: a) Die j-te Spalte von $\mathbf{A}$ bezeichnen wir wieder mit $\mathbf{a}^j$, $j = 1, 2, \ldots, n$. Dann ist

$$\begin{aligned} \mathbf{A}\mathbf{x}(\delta) &= \sum_{j=1}^{n} x_j(\delta)\mathbf{a}^j = \sum_{j \in I} x_j(\delta)\mathbf{a}^j + \delta\mathbf{a}^r = \sum_{j \in I}(x_j^{(0)} - \delta d_{jr})\mathbf{a}^j + \delta\mathbf{a}^r \\ &= \mathbf{A}\mathbf{x}^{(0)} + \delta(\mathbf{a}^r - \sum_{j \in I} d_{jr}\mathbf{a}^j) = \mathbf{A}\mathbf{x}^{(0)} = \mathbf{b}. \end{aligned}$$

Weiter ist (man vgl. auch (14.3))

$$\mathbf{c}^{\mathrm{T}}\mathbf{x}(\delta) = \sum_{j \in I} c_j x_j(\delta) + c_r\delta = \mathbf{c}^{\mathrm{T}}\mathbf{x}^{(0)} + \delta(c_r - \sum_{j \in I} d_{jr}c_j) = \mathbf{c}^{\mathrm{T}}\mathbf{x}^{(0)} + \delta t_r.$$

b) Sind alle $d_{jr} \le 0$ für $j \in I$, so ist $\mathbf{x}(\delta)$ für jedes $\delta \ge 0$ zulässig und $\mathbf{c}^{\mathrm{T}}\mathbf{x}(\delta)$ kann daher beliebig kleine Werte annehmen, somit ist die Zielfunktion nach unten unbeschränkt und das Problem $(*)$ unlösbar. $\square$

Satz 14.8. Sei $\mathbf{x}^{(0)}$ Basisvektor zur Basisindexmenge I. Existiere ein $r \notin I$ mit $t_r < 0$, sei $d_{ir} > 0$ für mindestens ein $i \in I$, und sei

$$\tilde{\delta} := \min_{i \in I} \left\{ \frac{x_i^{(0)}}{d_{ir}} : d_{ir} > 0 \right\} = \frac{x_s^{(0)}}{d_{sr}}. \tag{14.6}$$

Dann gilt mit $\mathbf{x}(\tilde{\delta})$ nach (14.5) und $\tilde{\delta}$ aus (14.6):

i) $\mathbf{x}(\tilde{\delta})$ ist zulässig,

ii) $\mathbf{x}(\tilde{\delta})$ ist Basisvektor zur Basisindexmenge $J = (I \cup \{r\}) \setminus \{s\}$,

iii) $\mathbf{c}^{\mathrm{T}}\mathbf{x}(\tilde{\delta}) \le \mathbf{c}^{\mathrm{T}}\mathbf{x}^{(0)}$.

Beweis: i) Wegen Satz 14.7a) müssen wir nur $\mathbf{x}(\tilde{\delta}) \ge \mathbf{0}$ zeigen. Die Wahl von $\tilde{\delta}$ nach (14.6) impliziert aber gerade $\mathbf{x}(\tilde{\delta}) \ge \mathbf{0}$. ii) Mit $\mathbf{a}^j$ bezeichnen wir wieder die Spalten von $\mathbf{A}$, $j = 1, 2, \ldots, n$. Sei $\sum_{j \in J} e_j\mathbf{a}^j = \mathbf{0}$. Wir müssen $e_j = 0$ für alle $j \in J$ zeigen. Es ist

$$\begin{aligned} \textstyle\sum_{j \in J} e_j\mathbf{a}^j &= \textstyle\sum_{j \in I\setminus\{s\}} e_j\mathbf{a}^j + e_r\mathbf{a}^r \\ &= \textstyle\sum_{j \in I\setminus\{s\}} e_j\mathbf{a}^j + e_r\sum_{j \in I} d_{jr}\mathbf{a}^j \\ &= \textstyle\sum_{j \in I\setminus\{s\}}(e_j + e_r d_{jr})\mathbf{a}^j + e_r d_{sr}\mathbf{a}^s. \end{aligned}$$

Aus der linearen Unabhängigkeit der $\mathbf{a}^i, i \in I$ folgt $e_i + e_r d_{ir} = 0$ für $i \in I \setminus \{s\}$ und $e_r d_{sr} = 0$. Wegen $d_{sr} > 0$ folgt $e_r = 0$ und somit auch $e_i = 0$ für $i \in I \setminus \{s\}$. Also $e_j = 0$ für alle $j \in J$. iii) folgt wegen $\tilde{\delta} \geq 0$ und $t_r < 0$ aus Satz 14.7a). □

Es ist zweckmäßig, die Koeffizienten d_{ij} zu einer Koeffizientenmatrix $\mathbf{D} = (d_{ij})$, $i \in I$, $j = 1, 2, \ldots, n$, der sogenannten *Tableaumatrix* zusammenzufassen. Das in (14.6) vorkommende Element d_{sr} heißt *Pivotelement*, s heißt *Pivotzeile*, r heißt *Pivotspalte*. Beim Übergang von I zu J wird $s \in I$ gegen $r \notin I$ ausgetauscht.

Im vorigen Satz kann der Fall $\mathbf{c}^{\mathrm{T}}\mathbf{x}(\tilde{\delta}) = \mathbf{c}^{\mathrm{T}}\mathbf{x}^{(0)}$ eintreten, d. h., $\mathbf{x}(\tilde{\delta})$ verbessert den Wert der Zielfunktion nicht. Betrachtet man noch einmal die Darstellung von $\mathbf{c}^{\mathrm{T}}\mathbf{x}(\tilde{\delta})$ aus dem Beweis von Satz 14.7, so ist dieser Fall offensichtlich nur möglich, wenn $\tilde{\delta} = 0$ ist.

Definition 14.9. Wir nennen einen Basisvektor $\mathbf{x}^{(0)}$ zur Basisindexmenge I *entartet*, wenn es einen Index $i \in I$ mit $x_i^{(0)} = 0$ gibt.

Bei nicht entartetem Basisvektor $\mathbf{x}^{(0)}$ ist also stets $\tilde{\delta} > 0$. Bei entartetem Basisvektor können beide Fälle $\tilde{\delta} = 0$ und $\tilde{\delta} > 0$ vorkommen. Es kann eine Folge $\mathbf{x}^{(k)}, \mathbf{x}^{(k+1)}, \ldots, \mathbf{x}^{(k+l)}, \mathbf{x}^{(k)}$ von übereinstimmenden entarteten Basisvektoren geben, bei der schließlich die Basisindexmenge von $\mathbf{x}^{(k)}$ erneut auftritt. Eine derartige Folge nennt man einen *Zyklus*. In diesem Fall tritt man also auf der Stelle und kommt nicht zu einer Lösung. Man kann jedoch durch geeignete Maßnahmen derartige Zyklen vermeiden. Man konsultiere etwa D. Gale: [1960, S. 123–128]. Beispiele von Zyklen findet man in S. I. Gass: [1969, Ch. 7]

In der Praxis kommen Zyklen jedoch sehr selten vor, so daß man in aller Regel auf besondere Maßnahmen zur Vermeidung von Zyklen verzichten kann.

Korollar 14.10. Das Optimierungsproblem ist genau dann unlösbar, wenn entweder (a) die zulässige Menge leer ist, oder (b) die Zielfunktion nach unten unbeschränkt ist.

Beweis: Ist die zulässige Menge nicht leer, so haben wir in den Sätzen 14.6, 14.7 und 14.8 alle Möglichkeiten für Lösbarkeit und Unlösbarkeit diskutiert. Für Unlösbarkeit bleibt nur der in Satz 14.7 angegebene Fall der Unbeschränktheit der Zielfunktion. Der Fall, daß man zyklisch immer wieder dieselben Basisvektoren erzeugt, kann nach den obigen Bemerkungen ausgeschlossen werden. □

Definition 14.11. Der Übergang von einem Basisvektor $\mathbf{x}^{(k)}$ zu einem neuen $\mathbf{x}^{(k+1)}$ nach dem angegebenen Rechenschema heißt *Simplexverfahren*.

Die zulässige Menge $\{\mathbf{x} : \mathbf{A}\mathbf{x} \geq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$ ist leer oder ein Polyeder im $\mathbb{R}^n$ und die Basisvektoren $\mathbf{x}^{(k)}, \mathbf{x}^{(k+1)}$ können als *benachbarte Ecken* dieses Polyeders aufgefaßt werden, die bei Entartung allerdings zusammenfallen. In diesem Fall sind nur die Basisindexmengen, die $\mathbf{x}^{(k)}, \mathbf{x}^{(k+1)}$ definieren, verschieden.

Satz 14.12. Sind alle Basisvektoren $\mathbf{x}^{(k)}, k = 0, 1, \ldots$, die nach dem Simplexverfahren erzeugt werden, nicht entartet, so bricht das Verfahren nach endlich vielen Schritten ab. Im letzten Schritt wird entweder die Lösung geliefert, oder die Aussage, daß das Problem wegen unbeschränkter Zielfunktion unlösbar ist.

Beweis: Nach den Voraussetzungen sind alle $\mathbf{x}^{(k)}$ verschieden. Da es nur endlich viele Basisvektoren gibt, muß das Verfahren abbrechen. □

Ist der zulässige Bereich ein n-dimensionaler Würfel, so haben Klee & Minty [1971] gezeigt, daß bei geeignet gewählter Zielfunktion das Simplexverfahren alle 2^n Ecken des

Würfels durchlaufen kann. Eine lange gehegte Vermutung, die sich aus vielen Experimenten erhärtet hatte (z. B.QUANDT & KUHN [1964]), daß nämlich die Anzahl der Iterationen beim Simplexverfahren durch ein Polynom in m, n eingeschränkt werden kann, wurde damit widerlegt.

14.4 Praktische Durchführung

Wir beschreiben *einen* Iterationsschritt des *Simplexverfahrens*. Dazu sei eine m-elementige Basisindexmenge I, die zu einem Basisvektor $\mathbf{x}$ gehört, bekannt. Den Basisvektor $\mathbf{x}$ braucht man an dieser Stelle noch nicht zu kennen. Wir definieren zu dieser Basisindexmenge I die $(m \times m)$-Matrix

$$\mathbf{B} = (\mathbf{a}^i), \quad i \in I, \quad \mathbf{a}^i = i\text{-te Spalte von } \mathbf{A}, \tag{14.7}$$

die definitionsgemäß regulär ist. Sei $\mathbf{D} = (d_{ij})$, $i \in I$, $j = 1, 2, \ldots, n$ die Tableaumatrix, die früher durch

$$\mathbf{a}^j = \sum_{i \in I} d_{ij} \mathbf{a}^i, \quad j = 1, 2, \ldots, n, \text{ mit } d_{ij} = \delta_{ij} \text{ für } j \in I \tag{14.8}$$

definiert wurde. Es gilt also wegen (14.7) und (14.8) $\mathbf{A} = \mathbf{BD}$, oder in äquivalenter Form

$$\mathbf{A}^{\mathrm{T}} = \mathbf{D}^{\mathrm{T}} \mathbf{B}^{\mathrm{T}}. \tag{14.9}$$

Wir setzen in Übereinstimmung mit (14.4), S. 148

$$\begin{aligned} &\mathbf{s} = (s_1, s_2, \ldots, s_n)^{\mathrm{T}}, \quad s_j = \textstyle\sum_{i \in I} d_{ij} c_i, \;\; j = 1, 2, \ldots, n, \\ &\mathbf{c} = (c_i), \; i \in I. \end{aligned} \tag{14.10}$$

Dann ist wegen (14.10)

$$\mathbf{s} = \mathbf{D}^{\mathrm{T}} \mathbf{c}. \tag{14.11}$$

Wir benutzen zum Rechnen jetzt nur die in (14.7) vorhandene Information, nämlich I. Der Vektor $\mathbf{s}$ kann aus (14.9) und (14.11) ohne Benutzung von $\mathbf{D}$ ausgerechnet werden. Multiplizieren wir nämlich (14.9) mit einem beliebigen Vektor $\mathbf{y} \in \mathbb{R}^m$, so erhalten wir $\mathbf{A}^{\mathrm{T}}\mathbf{y} = \mathbf{D}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{y}$. Bestimmen wir jetzt $\mathbf{y}$ so, daß $\mathbf{B}^{\mathrm{T}}\mathbf{y} = \mathbf{c}$ ist, so ist $\mathbf{A}^{\mathrm{T}}\mathbf{y} = \mathbf{D}^{\mathrm{T}}\mathbf{c} = \mathbf{s}$, d. h., um $\mathbf{s}$ zu berechnen, lösen wir das lineare Gleichungssystem

$$\boxed{\mathbf{B}^{\mathrm{T}} \mathbf{y} = \mathbf{c}} \tag{14.12}$$

und bestimmen $\mathbf{s}$ nach der Formel

$$\mathbf{s} = \mathbf{A}^{\mathrm{T}} \mathbf{y}. \tag{14.13}$$

Die Pivotspalte r erhält man daraus, sofern

$$t_r = c_r - s_r = \min_{j \notin I} (c_j - s_j) < 0; \tag{14.14}$$

man vgl. dazu Satz 14.6, S. 148 und Satz 14.7, S. 149. Die r-te Spalte $\mathbf{d}^r$ von $\mathbf{D}$, die in (14.6) benötigt wird, ergibt sich aus $\mathbf{BD}{=}\mathbf{A}$ zu

$$\boxed{\mathbf{B}\mathbf{d}^r = \mathbf{a}^r,} \tag{14.15}$$

wobei $\mathbf{a}^r$, wie üblich, die r-te Spalte von $\mathbf{A}$ bezeichnet. Gleichzeitig kann man den entsprechenden Basisvektor $\mathbf{x}$ mittels

$$\boxed{\mathbf{Bx}{=}\mathbf{b}} \tag{14.16}$$

ausrechnen. d. h., man berechne (14.15),(14.16) simultan, in der Form $\mathbf{B}(\mathbf{d}^r\,\mathbf{x}) = (\mathbf{a}^r\,\mathbf{b})$. Sind $\mathbf{d}^r = (d_{ir})$ und $\mathbf{x}$ bekannt, so kann nach der Formel (14.6) die Pivotzeile s bestimmt werden und damit auch die neue Basisindexmenge

$$J = (I \cup \{r\}) \setminus \{s\}. \tag{14.17}$$

Die fehlenden Komponenten von $\mathbf{x}$ müssen am Schluß durch Null ergänzt werden.

Dieses Verfahren, nämlich die Lösung der drei eingerahmten Gleichungssysteme ist für große m aufwendig, für kleine m (bei schnellen Rechnern etwa ≤ 50) durchaus passabel. Eine Verringerung des Rechenaufwandes kann erreicht werden, wenn einmalig in jedem Schritt eine Zerlegung von $\mathbf{B}$ (etwa nach dem Gaußschen Eliminationsverfahren) hergestellt und entsprechend umgerechnet wird. Haben wir eine Zerlegung der Form

$$\mathbf{FB} = \mathbf{R}, \qquad \mathbf{R} \text{ ist rechte Dreiecksmatrix, } \mathbf{F} \text{ regulär} \tag{14.18}$$

hergestellt, so kann man die auftretenden Gleichungssysteme vom Typ

$$\text{a)}\quad \mathbf{B}^{\mathrm{T}}\mathbf{y} = \mathbf{c}, \qquad \text{b)}\quad \mathbf{Bx} = \mathbf{b} \tag{14.19}$$

folgendermaßen lösen:

$$\text{a)}\quad \mathbf{R}^{\mathrm{T}}\mathbf{z} = \mathbf{c}, \quad \mathbf{y} = \mathbf{F}^{\mathrm{T}}\mathbf{z}, \qquad \text{b)}\quad \mathbf{Rx} = \mathbf{Fb}. \tag{14.20}$$

Fall a) kann also durch Vorwärtseinsetzen mit anschließender Matrix-Vektor-Multiplikation, b) durch Rückwärtseinsetzen gelöst werden. Die obige Zerlegung liegt bei der sog. **QR**-Zerlegung vor, bei der $\mathbf{Q}$ eine *orthogonale Matrix* ist, d. h. $\mathbf{Q}^{\mathrm{T}}\mathbf{Q} = \mathbf{I}$. Hat man speziell die beim Gaußschen Eliminationsverfahren entstehende Zerlegung $\mathbf{B} = \mathbf{LR}$, $\mathbf{L}$ = linke Dreiecksmatrix, so lauten a) und b) entsprechend

$$\text{a)}\quad \mathbf{R}^{\mathrm{T}}\mathbf{z} = \mathbf{c}, \quad \mathbf{L}^{\mathrm{T}}\mathbf{y} = \mathbf{z}, \qquad \text{b)}\quad \mathbf{Lw} = \mathbf{b}, \quad \mathbf{Rx} = \mathbf{w}, . \tag{14.21}$$

Eine weitere Verringerung der Anzahl der Operationen kann erreicht werden, wenn man berücksichtigt, daß zwei aufeinanderfolgende Matrizen $\mathbf{B}$, $\mathbf{B}'$, nämlich

$$\mathbf{B} = (\mathbf{a}^i),\ i \in I,\ \mathbf{B}' = (\mathbf{a}^j),\ j \in J = I \cup \{r\} \setminus \{s\} \tag{14.22}$$

sich nur in einer Spalte unterscheiden, so daß auch die entsprechenden Zerlegungen $\mathbf{FB} = \mathbf{R}$, $\mathbf{F}'\mathbf{B}' = \mathbf{R}'$ relativ leicht auseinander ausgerechnet werden können. Techniken zur

Zerlegung von $\mathbf{B}'$ unter Benutzung der Zerlegung von $\mathbf{B}$ werden *Modifikationstechniken* (Glashoff-Gustafson [1978, S. 165]) genannt. Darauf wird in Abschnitt 14.5, S. 159 eingegangen.

Kann man für das Problem $(*)$ keinen Basisvektor finden, so kann die Lösung eines *Hilfsproblems* Abhilfe schaffen.

Definition 14.13. Sei $\mathbf{e} = (1, 1, \ldots, 1)^{\mathrm{T}} \in \mathbb{R}^m$. Das folgende Problem $(***)$ nennen wir *Hilfsproblem der linearen Optimierung*.

$$\text{unter den Nebenbedingungen} \quad \left.\begin{array}{ll} \mathbf{e}^{\mathrm{T}}\mathbf{y} & \stackrel{!}{=} \min \\ \mathbf{A}\mathbf{x} + \mathbf{y} & = \mathbf{b}, \\ \mathbf{z} = \binom{\mathbf{x}}{\mathbf{y}} & \geq \mathbf{0}. \end{array}\right\} (***)$$

Satz 14.14. Sei das Problem $(*)$ unter der nichteinschränkenden Voraussetzung $\mathbf{b} \geq \mathbf{0}$ zu lösen. Für das obige Hilfsproblem $(***)$ gilt dann:
(a) Der Vektor $\mathbf{z} = \binom{\mathbf{0}}{\mathbf{b}}$ ist ein Basisvektor zur Basisindexmenge $I = \{n+1, n+2, \ldots, n+m\}$.
(b) Existiert eine Lösung $\mathbf{z} = \binom{\mathbf{x}}{\mathbf{y}}$ mit $\mathbf{e}^{\mathrm{T}}\mathbf{y} > 0$, so hat das Ausgangsproblem $(*)$ keine zulässigen Vektoren, ist also unlösbar.
(c) Ist $\mathbf{z} = \binom{\mathbf{x}}{\mathbf{y}}$ eine Lösung von $(***)$ mit $\mathbf{e}^{\mathrm{T}}\mathbf{y} = 0$, so ist $\mathbf{x}$ zulässig für $(*)$.

Beweis: (a) Da $\mathbf{A}\mathbf{x} + \mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{I}\mathbf{y}$ ($\mathbf{I} = (m \times m)$-Einheitsmatrix), trifft die Definition eines Basisvektors direkt auf $\mathbf{z}$ zu. (b) Sei $\hat{\mathbf{x}}$ zulässig für $(*)$. Dann ist $\hat{\mathbf{z}} = \binom{\hat{\mathbf{x}}}{\mathbf{0}}$ auch zulässig für $(***)$, und daher ist die Zielfunktion nach oben durch Null beschränkt, ein Widerspruch zu $\mathbf{e}^{\mathrm{T}}\mathbf{y} > 0$, d. h., es gibt in diesm Fall kein zulässiges $\hat{\mathbf{x}}$ für $(*)$. (c) Aus den Voraussetzungen folgt $\mathbf{y} = \mathbf{0}$, also ist $\mathbf{x}$ zulässig für $(*)$.

Bemerkungen. (i) Wir haben mit (b) auch den Fall, daß die zulässige Menge von $(*)$ leer ist, behandelt. (ii) Die im Falle (c) des vorigen Satzes ausgerechnete Basisindexmenge I für $(***)$ ist noch nicht notwendig eine Basisindexmenge für $(*)$, oder in anderen Worten, $\mathbf{x}$ ist noch nicht notwendig ein Basisvektor für $(*)$. Hier gibt es jetzt zwei Möglichkeiten: Es gelingt durch Fortsetzung des Simplexverfahrens für $(***)$ (ohne weitere Verkleinerung der Zielfunktion) eine Basisindexmenge für $(*)$ zu konstruieren oder das gelingt nicht. Im zweiten Fall bedeutet das, daß der Rang von $\mathbf{A}$ nicht maximal war. An dieser Stelle wird das also bemerkt. Um weiterzukommen, muß man überzählige Zeilen aus dem Gleichungssystem $\mathbf{A}\mathbf{x} = \mathbf{b}$ entfernen. Man vergleiche dazu die Literatur, z. B. Collatz & Wetterling, [1971, S. 36–38].

Die Lösung von $(*)$ über das Problem $(***)$ nennt man auch *Zweiphasenmethode* im Gegensatz zur *Einphasenmethode*, bei der im Problem $(***)$ die Zielfunktion durch

$$S_1\mathbf{c}^{\mathrm{T}}\mathbf{x} + S_2\mathbf{e}^{\mathrm{T}}\mathbf{y} \tag{14.23}$$

mit geeigneten Konstanten $S_1, S_2 \geq 0$ ersetzt wird. Wenn man mit dieser Methode arbeiten will, muß man zeigen, daß die Lösungen unter geeigneten Annahmen (großer Quotient S_2/S_1) nicht von den gewählten Zahlen S_1, S_2 abhängen. Die Wahl $S_1 = 0$ führt auf die hier verwendete Zweiphasenmethode.

Der Nachteil der hier beschriebenen Zweiphasenmethode ist, daß nach Beendigung der Rechnungen für $(***)$ das Simplexverfahren für $(*)$ neu gestartet werden muß. Da im Hilfsproblem die eigentliche Zielfunktion gar nicht vorkommt, kann man auch nicht erwarten, daß der ausgerechnete zulässige Vektor ein besonders guter Startvektor für $(*)$ ist. Die zur Lösung eines linearen Optimierungsproblems notwendigen Rechenschritte sind in der Übersicht 14.19, S. 158 schematisch zusammengefaßt.

Beispiel 14.15. *Lösung eines linearen Optimierungsproblems*
Wir lösen $\mathbf{c}^{\mathrm{T}}\mathbf{x} \overset{!}{=} \min$, $\mathbf{A}\mathbf{x} = \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$ mit $\mathbf{c}^{\mathrm{T}} = (-120, -360, 0, 0, 0)$ und

$$\mathbf{A} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 30 & 60 & 0 & 1 & 0 \\ 2 & 10 & 0 & 0 & 1 \end{pmatrix}, \ \mathbf{b} = \begin{pmatrix} 1200 \\ 42000 \\ 5200 \end{pmatrix}.$$

Erster Schritt: $I = \{3, 4, 5\}$, $\mathbf{B} = \mathbf{I} =$ Einheitsmatrix, $\mathbf{c} = (c_3, c_4, c_5)^{\mathrm{T}} = (0, 0, 0)^{\mathrm{T}}$. Also hat nach (14.12), S. 151 das System $\mathbf{B}^{\mathrm{T}}\mathbf{y} = \mathbf{c}$ die Lösung $\mathbf{y} = \mathbf{0}$ und somit $\mathbf{s} = \mathbf{0}$ und $t_2 = \min_{j=1,2}(c_j - 0) = -360$, d. h. $r = 2$. Aus (14.15), S. 152 folgt $\mathbf{B}\mathbf{d} = \mathbf{a}^2 = (1, 60, 10)^{\mathrm{T}}$, d. h. $\mathbf{d} = \mathbf{a}^2$ und aus (14.16), S. 152 folgt $\mathbf{x} = \mathbf{b} = (1200, 42000, 5200)^{\mathrm{T}}$. Nach (14.6), S. 149 erhält man $\tilde{\delta} = \min_{j \in I}\left\{\frac{x_j}{d_j} : d_j > 0\right\} = x_5/d_5 = 520$, und somit $s = 5$.

Zweiter Schritt: $I = \{3, 4, 5\} \cup \{r\} \setminus \{s\} = \{3, 4, 2\}$, $\mathbf{B} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 60 \\ 0 & 0 & 10 \end{pmatrix}$, . $\mathbf{c} = (0, 0, -360)^{\mathrm{T}}$, $\mathbf{B}^{\mathrm{T}}\mathbf{y} = \mathbf{c}$ hat die Lösung $\mathbf{y} = (0, 0, -36)^{\mathrm{T}}$ und daraus folgt $\mathbf{s} = \mathbf{A}^{\mathrm{T}}\mathbf{y} = (-72, -360, 0, 0, -36)^{\mathrm{T}}$ und $t_1 = \min_{j=1,5}(c_j - s_j) = -48$, $r = 1$. Aus $\mathbf{B}\mathbf{d} = \mathbf{a}^1 = (1, 30, 2)^{\mathrm{T}}$ folgt $\mathbf{d} = (4/5, 18, 1/5)^{\mathrm{T}}$, und aus $\mathbf{B}\mathbf{x} = \mathbf{b}$ folgt $\mathbf{x} = (680, 10800, 520)^{\mathrm{T}}$ und $\tilde{\delta} = \min_{j \in I}\left\{\frac{x_j}{d_j} : d_j > 0\right\}$ $= x_4/d_4 = 600$, also $s = 4$.

Dritter Schritt: $I = \{3, 1, 2\}$, $\mathbf{B} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 30 & 60 \\ 0 & 2 & 10 \end{pmatrix}$, $\mathbf{c} = (0, -120, -360)^{\mathrm{T}}$, $\mathbf{B}^{\mathrm{T}}\mathbf{y} = \mathbf{c}$ hat die Lösung $\mathbf{y} = (0, -8/3, -20)^{\mathrm{T}}$ und daraus folgt $\mathbf{s} = \mathbf{A}^{\mathrm{T}}\mathbf{y} = (-120, -360, 0, -8/3, -20)^{\mathrm{T}}$ und alle $t_j \geq 0$, $j = 4, 5$, d. h., wir sind bei der Lösung angekommen, die sich aus $\mathbf{B}\mathbf{x} = \mathbf{b}$ zu $\mathbf{x} = (x_3, x_1, x_2)^{\mathrm{T}} = (200, 600, 400)^{\mathrm{T}}$ berechnet. Damit ist $\tilde{\mathbf{x}} = (600, 400, 200, 0, 0)^{\mathrm{T}}$ die endgültige Lösung und $\mathbf{c}^{\mathrm{T}}\tilde{\mathbf{x}} = -216000$.

Man beachte, daß die Basisindexmengen I zweckmäßigerweise nicht nach der Größe der Indizes geordnet werden sollten. Ein einfaches Musterprogramm 14.16 in matlab-Sprache ist unten angegeben.

Programm 14.16. Simplexprogramm nach dem angegeben Schema

```
%Es wird ein Schritt des Simplexverfahrens programmiert, ausgehend von

%  c'x=min, Ax=b, x>=0, A eine m x n Matrix mit m < n und Rang A = m

%bei bekannter Basisindexmenge I, d. h. B := A(I) besteht aus m linear
%unabh\"angigen Spalten und Bx=b hat eine zul\"assige L\"osung. Das Ergebnis
%ist entweder eine neue Basisindexmenge J, oder das Problem hat wegen
%unbeschr\"ankter Zielfunktion keine L\"osung. Das Programm hat die Form
%
%       function [wert,x,J,info]=simplex(A,b,c,I)
```

```
%
%In info gibt es die folgenden Informationen:
%info=-1: Problem hat keine L\"osung, Rechnung beenden,
%info=0 : I definiert noch keine L\"osung, Rechnung fortsetzen,
%info=1 : I definiert eine L\"osung, Rechnung beenden.
%Ist die L\"osung erreicht (bei info=1), so enth\"alt x die L\"osung in
%der urspr\"unglichen Numerierung.
%Programm: Gerhard Opfer, 12. Januar 1999
%============================================================================
function [wert,x,J,info]=simplex(A,b,c,I);
%wert=0;x=[];J=I; %rein formal, keine Bedeutung
[m,n]=size(A);
if m>=n error('Matrix im Simplexprogramm hat falsche Dimension');end;
%============================================================================
%                                           Erstes Gleichungssystem l\"osen
B=A(:,I);  y=B'\c(I);  svektor=A'*y;  notI=1:n;  notI(I)=[];
t=c-svektor; t=t(notI);
[mi,i]=min(t);                            %i kann mehrere Indizes enthalten
%============================================================================
                                     %Pr\"ufung, ob L\"osung bereits erreicht
if mi >= 0, info=1; x=B\b; J=I; wert=c(I)'*x;         %L\"osung ist erreicht
  xx=zeros(n,1); xx(I')=x; xx(notI')=[0;0];x=xx;
%============================================================================
  else                                          %L\"osung ist nicht erreicht
  r=notI(i(1));  %Pivotspalte  %Zweites und drittes Gleichungssystem l\"osen
  a=A(:,r);      %r-te Spalte von A
  dx=B\[a,b];d=dx(:,1);x=dx(:,2);
%============================================================================
                                      %Pr\"ufung, ob L\"osung nicht existiert
  if max(d)<=0,info=-1; x=[]; return, end;        %L\"osung existiert nicht
%============================================================================
                                                               %weitermachen
  P=find(d>0);
  [delta,s]=min(x(P)./d(P));              %s kann mehrere Indizes enthalten
  zeile=I(P(s(1)));
  p=find(I==zeile); J=I; J(p)=r;              %Vertauschung ohne Sortierung
  info=0;  wert=c(I)'*x;
  xx=zeros(n,1); xx(I')=x; xx(notI')=zeros(n-m,1); x=xx;
end; %if
%============================================================================
```

Ein Beispielprogramm, das graphisch interpretiert werden kann und auch größere Probleme zuläßt ist im folgenden Programm 14.18 angegeben. Die Nebenbedingungen sind dabei Halbebenen, die durch Tangenten an eine gegebene Ellipse begrenzt werden. Das unten angegebene Bild 14.17 wurde damit gerechnet.

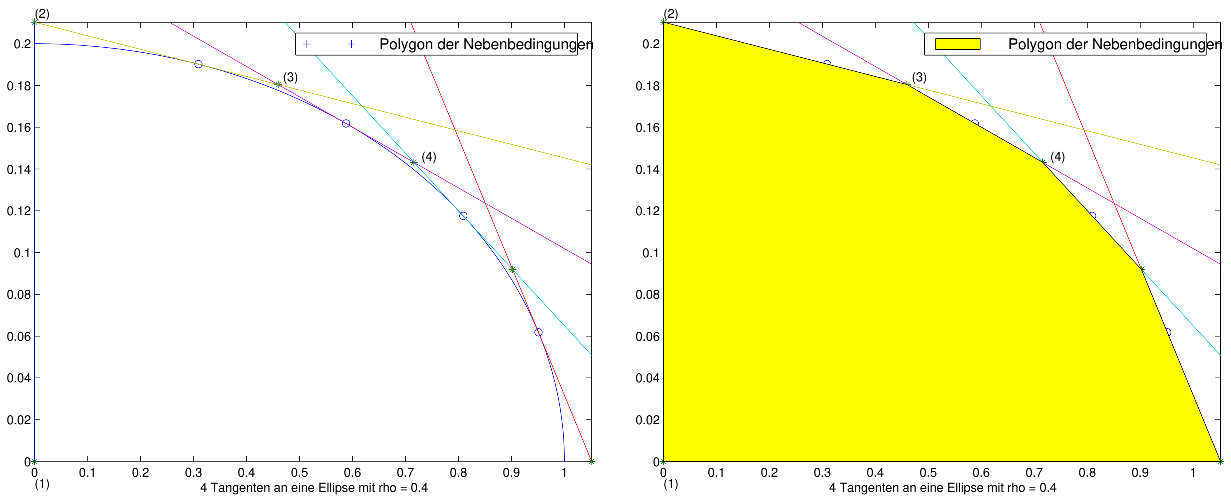


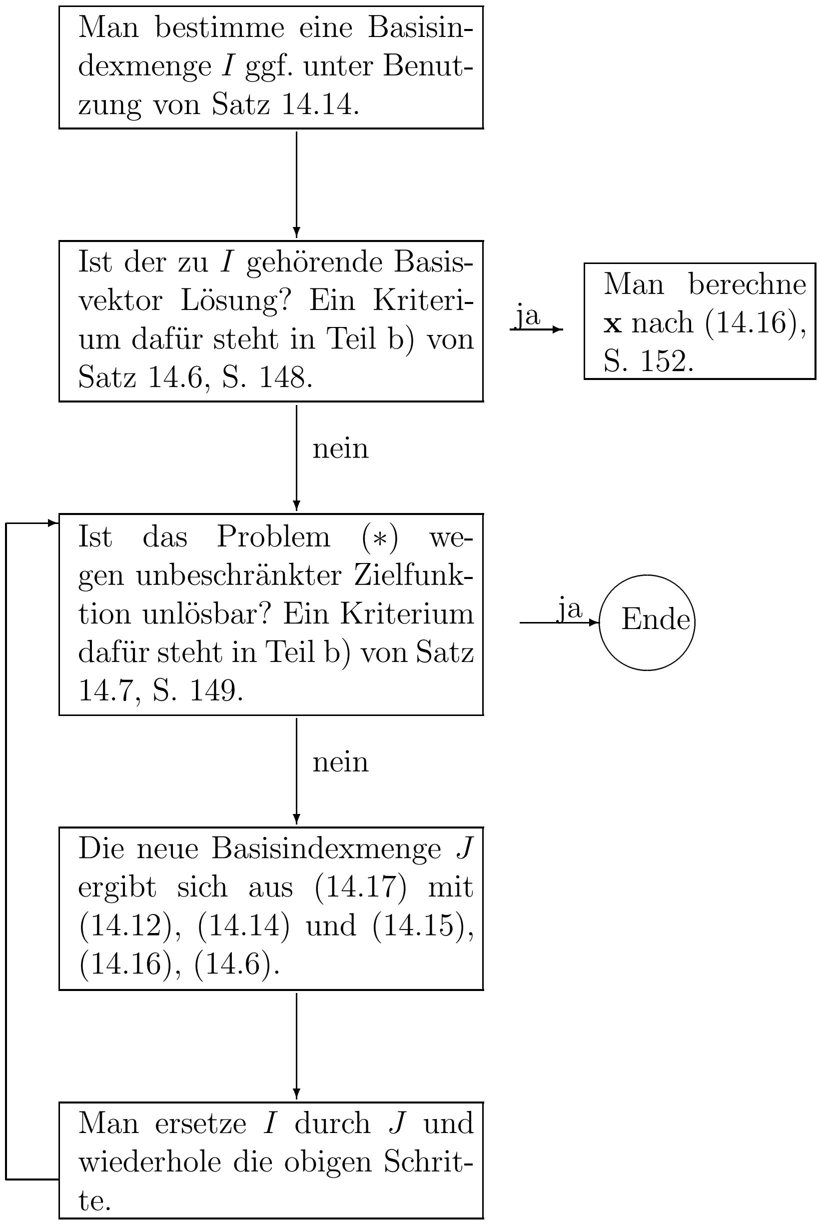
Abb. 14.17. Test des Simplexverfahrens, links Konstruktion, rechts zulässiger Bereich

Programm 14.18. Simplexprogramm nach dem angegeben Schema

```
%Es werden Nebenbedingungen f\"ur ein LP-Programm erzeugt, die
%aus Tangenten an eine Ellipse bestehen. Die Ellipse hat die Form

%      E(t):=(1-rho)z(t)+rho/z(t), rho in [0, 0.5[,  z(t)=exp(it).

%Diese Ellipse hat die Halbachsen a=1 und b=1-2rho. Ist
%(x_0,y_0), ein Punkt auf der Ellipse, aber kein Punkt auf einer der Achsen,
%so hat die Tangente die Form

%                       x*x_0+y*y_0/(1-2rho)^2=1.

%Diese Tangente durchst\"o\ss t die beiden Achsen an den Stellen

%              (x1,y1) = (1/x_0,0); (x2,y2) = (0,(1-2rho)^2/y_0).

%Zwei aufeinander folgende Nebenbedingungen
%x*x_0+y*y_0/(1-2rho)^2=1
%x*u_0+y*v_0/(1-2rho)^2=1 schneiden sich im Punkt

%           (xs,ys)= ((y0-v0)/Nenner,(1-2*rho)^2*(u0-x0)/Nenner)

%                        mit Nenner=u0*y0-x0*v0.
%Das sind Ecken des zul\"assigen Bereichs. F\"ugt man noch die Ecken
%(0, 0), (x1(1), 0), (0, y2(letzte Komponente)) hinzu, erh\"alt man
%alle Ecken.                       Programm: Gerhard Opfer, 22. Januar 1999.
%===========================================================================
%            Vorspann, kann entfallen, dann m\"ussen ell, rho besetzt werden
function [wert,x]=simplextest(wieviele_Punkte,Form_der_Ellipse);
ell=wieviele_Punkte; if ell<=0, ell=1; end; if ell > 100, ell=100; end;
if nargin==1, rho=0; else
  if Form_der_Ellipse=='Kreis',rho=0;   end;
  if Form_der_Ellipse=='fl_El',rho=0.4; end;
  if Form_der_Ellipse=='rd_El',rho=0.1; end;
end; %if nargin
%===========================================================================
%ell=1;                           %ell ausgew\"ahlte Punkte auf der Ellipse
```

```
%rho=0.4;%rho=0;   %rho=0 gibt Einheitskreis, ansonsten Ellipsen bis rho<0.5
T0=linspace(pi/(2*(ell+1)),ell*pi/(2*(ell+1)),ell);
x0=cos(T0);
y0=sin(T0)*(1-2*rho);
%============================================================================
Nenner=ones(1,ell-1);                     %Schnittpunkte benachbarter Tangenten
Nenner=x0(2:ell).*y0(1:ell-1)-x0(1:ell-1).*y0(2:ell);
xs=(y0(1:ell-1)-y0(2:ell))./Nenner;
ys=(1-2*rho)^2*(x0(2:ell)-x0(1:ell-1))./Nenner;
%============================================================================
                              %Durchsto\ss punkte der Tangenten durch die Achsen
x1=1./x0;y2=(1-2*rho)^2./y0;
%============================================================================
%Hinzunahme der \"ubrigen Ecken des zul\"assigen Bereichs
xs=[x1(1) xs];ys=[0 ys];xs=[0 xs];ys=[0 ys];xs=[xs 0];ys=[ys y2(ell)];
%xs,ys enthalten jetzt die Koordinaten aller ell+2 Ecken.
%============================================================================
%       Besetzung des LP-Programms Ax=b, c'x=min, x>=0 und I=Basisindexmenge
%                              A enth\"alt die Koeffizienten der Tangente, s. o.
A=[x0',y0'/(1-2*rho)^2];
A=[A,eye(ell)];I=3:ell+2;
c=zeros(ell+2,1);c(1)=-1;c(2)=-1/(1-2*rho);      %Also max x_1+x_2/(1-2rho)
b=ones(ell,1);
I=3:ell+2;
%============================================================================
%                                                 Ausf\"uhren des LP-Programms
info=0;zeler=0;                        %und Zeichnen der ausgerechneten Punkte
while info==0 & zeler <= ell+3
  zeler=zeler+1;
  if zeler==1
    disp(['Beginn von Simplexiteration Nr. ',int2str(zeler)]); else
    disp(['                            Nr. ',int2str(zeler)]);
  end; %if
  [wert,x,I,info]=weiter14simplex(A,b,c,I);
  plot(x(1),x(2),'+'); hold on
  if zeler==1
    text(x(1),x(2)-y2(ell)/20,'(1)');
    else
    text(1.02*x(1),1.02*x(2),['(',int2str(zeler),')']);
  end; %if
end; %while
%============================================================================
t=linspace(0,pi/2);                 %Zeichnen der Ellipse im ersten Quadranten
z=exp(i*t);
E=(1-rho)*z+rho./z;
plot(E);                                                  %k\"onnte auch entfallen
%============================================================================
                                              %Zeichnen des zul\"assigen Bereichs
plot(x0,y0,'o',xs,ys,'*',[x1' zeros(ell,1)]',[zeros(ell,1) y2']',...
     [0 max(x1)],[0 0],[0 0],[0 max(y2)]);
disp('Der zulässige Bereich wird gleich vergrößert gezeigt');
pause(5);                                  %Bevor der zul. Bereich gefüllt wird,
                                        %ein kurzer Blick auf die Konstruktion
h=fill(xs,ys,'r');             %hier kann man auch mit anderen Farben spielen
axis equal     %Ohne diesen Befehl, sind alle Bereiche i. w. Viertelkreise
axis([0  x1(1)  0  y2(ell)]);
legend(h,'Polygon der Nebenbedingungen');
xlabel([int2str(ell),' Tangenten an eine Ellipse mit rho = ',num2str(rho)]);
print -dpsc zulBer14.ps
hold off
%============================================================================

%Testergebnisse:
```

```
%Alle Basisvektoren im Fall ell=2, rho=0.25
%   not in I        X(I)             x_1, x_2
%-----------------------------------------------------
%    1   2          1         1        (0, 0)      (Nr.1)
%    1   4     0.5774    0.4226        (0, 0.5774) (Nr.4)
%    2   3     1.1547    0.4226   (1.1547, 0)      (Nr 2)
%    3   4     0.7321    0.3660   (0.7321, 0.3660) (Nr 3)
```



Übersicht 14.19. Rechenschritte für das Simplexverfahren

14.5 Modifikationstechniken

Sei $\mathbf{B} = (\mathbf{b}^1, \ldots, \mathbf{b}^m)$ eine $m \times m$-Matrix mit den Spalten $\mathbf{b}^1, \ldots, \mathbf{b}^m$ und

$$\mathbf{F}\mathbf{B} \;=\; \mathbf{R}, \quad \mathbf{R} \text{ rechte Dreiecksmatrix}, \quad \mathbf{F} \text{ regulär.} \tag{14.24}$$

sei eine bekannte Zerlegung, die man z. B. durch Gauß-Elimination (LR-Zerlegung) oder mit dem noch zu besprechenden Householder-Verfahren (QR-Zerlegung) gewinnen kann. Wir setzen $\mathbf{R} = (\mathbf{r}^1, \ldots, \mathbf{r}^m)$, d. h., $\mathbf{r}^1, \ldots, \mathbf{r}^m$ bezeichnen die Spalten der rechten Matrix $\mathbf{R}$. Sei nun $\mathbf{b}$ ein weiterer vorgegebener Vektor und $\mathbf{B}'$ die Matrix, die aus $\mathbf{B}$ durch Streichen von $\mathbf{b}^p$ und Hinzufügen von $\mathbf{b}$ als letzter Spalte entsteht, wobei $p \in \{1, \ldots, m\}$ eine vorgegebene Spaltennummer ist: $\mathbf{B}' := (\mathbf{b}^1, \ldots, \mathbf{b}^{p-1}, \mathbf{b}^{p+1}, \ldots, \mathbf{b}^m, \mathbf{b})$. Gesucht ist eine (14.24) entsprechende Zerlegung für $\mathbf{B}'$, also

$$\mathbf{F}'\mathbf{B}' = \mathbf{R}', \quad \mathbf{R}' \ \text{ rechte Dreiecksmatrix}, \quad \mathbf{F}' \ \text{ regulär,} \tag{14.25}$$

und zwar möglichst einfach durch Umrechnen von (14.24) in (14.25). Wir beschreiben dazu zwei Methoden. Sehen wir uns zunächst die Matrix $\mathbf{H} := \mathbf{F}\mathbf{B}'$ an:

$$\begin{aligned}
\mathbf{H} \;&=\; (\mathbf{r}^1, \ldots, \mathbf{r}^{p-1}, \mathbf{r}^{p+1}, \ldots, \mathbf{r}^m, \mathbf{F}\mathbf{b}) \\
&=\; \begin{array}{c}
\left(\begin{array}{ccccccccc}
\times & \times & \ldots & \times & \times & \ldots & & & \times \\
 & \times & \ldots & \times & \times & \ldots & & & \times \\
 & & \ddots & \vdots & \vdots & & & & \vdots \\
 & & & \times & \times & & & & \vdots \\
 & & & & \times & & & & \vdots \\
 & & & & \times & \times & & & \vdots \\
 & 0 & & & & \times & \ddots & & \vdots \\
 & & & & & & \ddots & & \vdots \\
 & & & & & & & \times & \times
\end{array}\right) . \quad \leftarrow p\text{-te Zeile} \\
\uparrow \\
p\text{-te Spalte}
\end{array}
\end{aligned}$$

Die Matrix $\mathbf{H}$ enthält als letzte Spalte den Vektor $\mathbf{h} := \mathbf{F}\mathbf{b}$, davor die Spalten von $\mathbf{R}$ mit Ausnahme der p-ten Spalte. Matrizen, die wie $\mathbf{H}$ nur noch in der Subdiagonalen und darüber nicht verschwindende Einträge haben, nennt man *(obere) Hessenbergmatrizen.* Die $m - p$, i. a. von 0 verschiedenen Matrixelemente unterhalb der Diagonalen von $\mathbf{H}$ können nun leicht durch Gauß-Eliminationsschritte mit Spaltenpivotsuche zu Null gemacht werden; dabei braucht man nur eine Vertauschung *benachbarter* Zeilen in Betracht zu ziehen. Einem solchen Eliminationsschritt entspricht eine Multiplikation von links mit einer Matrix $\mathbf{G}_i$, die entweder von der Form

$$
\begin{array}{cl}
\begin{pmatrix}
1 & & & & & & & 0 \\
 & \ddots & & & & & & \\
 & & 1 & & & & & \\
 & & & 1 & 0 & & & \\
 & & & -\ell_{i+1} & 1 & & & \\
 & & & & & 1 & & \\
 & & & & & & \ddots & \\
0 & & & & & & & 1
\end{pmatrix}
&
\begin{array}{l}
\\ \\ \\ \leftarrow \quad i \\ \leftarrow \quad i+1 \\ \\ \\ \\
\end{array}
\end{array}
$$

(falls kein Zeilentausch stattfindet) oder von der Form

$$
\begin{array}{cl}
\begin{pmatrix}
1 & & & & & & & 0 \\
 & \ddots & & & & & & \\
 & & 1 & & & & & \\
 & & & 0 & 1 & & & \\
 & & & 1 & -\ell_{i+1} & & & \\
 & & & & & 1 & & \\
 & & & & & & \ddots & \\
0 & & & & & & & 1
\end{pmatrix}
&
\begin{array}{l}
\\ \\ \\ \leftarrow \quad i \\ \leftarrow \quad i+1 \\ \\ \\ \\
\end{array}
\end{array}
$$

(falls ein Zeilentausch stattfindet) ist. Aufgrund der Pivotsuche ist $|\ell_{i+1}| \leq 1$. Nach $m-p$ Eliminationsschritten hat man

$$\mathbf{G}_{m-1} \cdots \mathbf{G}_p \mathbf{H} \;=:\; \mathbf{R}' \tag{14.26}$$

mit einer rechten Dreiecksmatrix $\mathbf{R}'$ erreicht. Setzt man hierin $\mathbf{H} = \mathbf{F}\mathbf{B}'$ ein, so erhält man mit

$$\mathbf{G}_{m-1} \cdots \mathbf{G}_p \, \mathbf{F} \;=:\; \mathbf{F}' \tag{14.27}$$

eine Zerlegung der gewünschten Form (14.25).

Für die numerische Rechnung kann man aus (14.26) und (14.27) ablesen, daß man $\mathbf{R}'$ und $\mathbf{F}'$ dadurch erhält, daß man die Eliminationsschritte *simultan* auf $\mathbf{H}$ und $\mathbf{F}$ anwendet. Während das direkte Ausrechnen einer Zerlegung (14.25) durch Gauß-Elimination ca. $c \cdot m^3$ Multiplikationen und Divisionen kostet, verbraucht die beschriebene Modifizierung von (14.24) lediglich ca. $(m-p)^2$ Operationen.

Wir geben noch eine weitere Methode an, bei der eine möglichst einfache Matrix $\mathbf{T}$ gefunden werden soll, so daß bei unveränderten Bezeichnungen das Matrixprodukt

$$\tilde{\mathbf{H}} := \mathbf{T}\mathbf{H} = \mathbf{T}\mathbf{F}\mathbf{B}' \tag{14.28}$$

die Gestalt

$$\begin{array}{cl}
\begin{pmatrix}
\times & \times & \dots & \times & \times & \dots & & & \times \\
 & \times & \dots & \times & \times & \dots & & & \times \\
 & & \ddots & \vdots & \vdots & & & & \vdots \\
 & & & \times & \times & & & & \vdots \\
 & & & & 0 & 0 & \dots & 0 & \times \\
 & & & & \times & \times & & & \vdots \\
 & 0 & & & & \times & \ddots & & \vdots \\
 & & & & & & \ddots & & \vdots \\
 & & & & & & & \times & \times
\end{pmatrix} & \begin{array}{l} \\ \\ \\ \\ \leftarrow \quad p\text{-te Zeile} \\ \\ \\ \\ \\ \end{array} \\
\begin{array}{c} \uparrow \\ p\text{-te Spalte} \end{array} &
\end{array} \tag{14.29}$$

hat. Die endgültige Form erhält man dann durch Verschieben der p-ten Zeile nach hinten und Vorschieben der nachfolgenden Zeilen.

Als Transformation $\mathbf{T}$ benutzen wir eine *elementare Zeilenmatrix* der Form

$$\mathbf{T} = \mathbf{I} - \mathbf{e_p}\mathbf{v}^{\mathrm{T}}. \tag{14.30}$$

Dabei ist $\mathbf{e_p}$ der p-te Einheitsvektor und $\mathbf{v}^{\mathrm{T}} = (0, 0, \dots, 0, v_{p+1}, v_{p+2}, \dots, v_m)$ ein Vektor, dessen Komponenten wir durch Vorwärtseinsetzen aus dem Gleichungssystem

$$\mathbf{R}^{\mathrm{T}}\mathbf{v} = \mathbf{r} \tag{14.31}$$

bestimmen, wobei $\mathbf{r}^{\mathrm{T}} = (0, 0, \dots, 0, r_{p,p+1}, r_{p,p+2}, \dots, r_{pm})$ aus der bekannten p-ten Zeile der rechten Matrix $\mathbf{R}$ abgelesen wird. Man beachte, daß in $\mathbf{r}$ an Stelle des Diagonalelements von $\mathbf{R}$ Null steht. Damit erhalten wir dann

$$\tilde{\mathbf{H}} = \mathbf{TH} = (\mathbf{I} - \mathbf{e_p}\mathbf{v}^{\mathrm{T}})\mathbf{H} = \mathbf{H} - \mathbf{e_p}\mathbf{v}^{\mathrm{T}}\mathbf{H}, \tag{14.32}$$

wobei die Matrix $\mathbf{e_p}\mathbf{v}^{\mathrm{T}}\mathbf{H}$ nur in der p-ten Zeile von Null verschiedene Einträge besitzt, die wegen (14.31) mit Ausnahme des letzten Elements alle mit den entsprechenden Elementen von $\mathbf{H}$ übereinstimmen. Damit haben wir die gewünschte Form hergestellt, die abgesehen von den erwähnten Änderungen in der p-ten Zeile $\mathbf{H}$ total unverändert läßt. Wollen wir also die neue Form (14.29) herstellen, ersetzen wir in $\mathbf{H}$ alle Elemente der p-ten Zeile bis auf das letzte durch Null, und ersetzen das letzte Element durch

$$h_{pm} = h_p - \mathbf{v}^{\mathrm{T}}\mathbf{h}, \tag{14.33}$$

wobei $\mathbf{h} = \mathbf{Fb} = (h_1, h_2, \dots, h_m)^{\mathrm{T}}$ die letzte Spalte von $\mathbf{H}$ und $\mathbf{v}$ die Lösung von (14.31) bedeutet. Bis auf die angegebenen Zeilenvertauschungen wird $\mathbf{F}$ wegen (14.28) nach der Formel

$$\tilde{\mathbf{F}} := \mathbf{TF} = (\mathbf{I} - \mathbf{e_p}\mathbf{v}^{\mathrm{T}})\mathbf{F} = \mathbf{F} - \mathbf{e_p}\mathbf{v}^{\mathrm{T}}\mathbf{F} \tag{14.34}$$

umgerechnet. Die Matrix $\mathbf{e_p v}^{\mathrm{T}}$ (eine Dyade) enthält höchstens in der Zeile p ab Spalte $p+1$ von Null verschiedene Einträge, also ist

$$\tilde{\mathbf{F}}_{jk} = \mathbf{F}_{jk} \text{ für alle Zeilen } j \neq p, \text{ und } \tilde{\mathbf{F}}_{pk} = \mathbf{F}_{pk} - (\mathbf{v}^{\mathrm{T}}\mathbf{F})_k, \text{ für alle } k = 1, 2, \ldots, m, \tag{14.35}$$

wobei die unteren Indizes Matrix- bzw. Vektorelemente bezeichnen. Die beiden Verfahren lassen sich algorithmisch so fassen:

Schritt 0: Gegeben Matrizen $\mathbf{B}$, $\mathbf{F}$, $\mathbf{R}$ mit (14.24), Vektor $\mathbf{b}$, Spaltennummer p.

Schritt 1: Man berechne $\mathbf{h} = \mathbf{Fb}$ (m^2 Op.).

Schritt 2: Man bilde die Matrix $\mathbf{H}$. Hier verzweigen die beiden Algorithmen. Für den ersten Algorithmus ist der mit **A3** bezeichnete Schritt auszuführen, für den zweiten die mit **B** bezeichneten Schritte.

Schritt A3: Man eliminiere mit dem Gaußschen Verfahren in der rechteckigen Matrix $(\mathbf{H}, \mathbf{F})$ die Subdiagonalelemente von $\mathbf{H}$ unter Einschaltung einer Spaltenpivotsuche ($(m-p)(m+1+(m+1-p)/2)$ Op.). Man erhält die gewünschte Matrix $(\mathbf{H}', \mathbf{F}')$. Hier endet der erste Algorithmus.

Schritt B3: Man löse $\mathbf{R}^{\mathrm{T}}\mathbf{v} = \mathbf{r} = p$-te Zeile von $\mathbf{R}$, Diagonalelement Null ($(m-p)(m-p+1)/2$ Op.).

Schritt B4: Dazu setze man in $\tilde{\mathbf{H}}$ die p-te Zeile Null und berechne das letzte Element der p-ten Zeile nach (14.33) ($m-p$ Op.).

Schritt B5: Man bilde $\tilde{\mathbf{F}}$ nach (14.35) ($m(m-p)$ Op.).

Schritt B6: Man schiebe in den Matrizen $\tilde{\mathbf{H}}$, $\tilde{\mathbf{F}}$ die p-te Zeile nach hinten und alle nachfolgenden Zeilen eine Zeile nach oben. Man erhält die gewünschten Matrizen $\mathbf{H}'$, $\mathbf{F}'$. Hier endet der zweite Algorithmus.

Der Vorteil dieser zweiten Methode ist, daß relativ wenige Elemente verändert werden. Das ist insbesondere für *dünn besetzte Matrizen* wichtig (das sind solche mit vielen Nullen), bei denen dann die vorhandenen Nullen zum großen Teil erhalten bleiben. Hinweise zu den Modifikationstechniken findet man in den Büchern von Fletcher[1987, Ch. 8.5], Nazareth[1987, Ch. 6].

Beispiel 14.20. *Modifizierung nach den beiden angegebenen Methoden*
Wir verwenden die Zerlegung $\mathbf{FB} = \mathbf{R}$ mit $m = 4$ und wählen $\mathbf{b} = (1, 2, 3, 4)^{\mathrm{T}}$ und $p = 1$.

$$\mathbf{F} = \begin{pmatrix} -5.2792e-02 & 3.9594e-01 & -7.9188e-01 & 4.6193e-01 \\ 8.5203e-01 & -3.9783e-01 & -6.0391e-02 & 3.3485e-01 \\ 5.1999e-01 & 6.6415e-01 & -1.5782e-02 & -5.3690e-01 \\ 2.9394e-02 & 4.9382e-01 & 6.0748e-01 & 6.2148e-01 \end{pmatrix},$$

$$\mathbf{B} = \begin{pmatrix} 16 & 120 & 240 & -140 \\ -120 & 1200 & -2700 & 1680 \\ 240 & -2700 & 6480 & -4200 \\ -140 & 1680 & -4200 & 2800 \end{pmatrix},$$

$$\mathbf{R} = \begin{pmatrix} -3.0308e+02 & 3.3829e+03 & -8.1532e+03 & 5.2918e+03 \\ 0 & 3.5044e+02 & -5.1906e+02 & 4.0357e+02 \\ 0 & 0 & 4.8430e+02 & -3.9406e+02 \\ 0 & 0 & 0 & 1.4227e+01 \end{pmatrix}.$$

Die erste Methode liefert $\mathbf{F}'\mathbf{B}' = \mathbf{R}'$ mit

$$\mathbf{F}' = \begin{pmatrix} -5.2792e-02 & 3.9594e-01 & -7.9188e-01 & 4.6193e-01 \\ 5.1999e-01 & 6.6415e-01 & -1.5782e-02 & -5.3690e-01 \\ 5.0796e-01 & -8.8530e-01 & 3.2250e-02 & 6.4790e-01 \\ -3.0695e-02 & 5.9855e-01 & 6.0367e-01 & 5.4484e-01 \end{pmatrix},$$

$$\mathbf{B}' = \begin{pmatrix} 120 & 240 & -140 & 1 \\ 1200 & -2700 & 1680 & 2 \\ -2700 & 6480 & -4200 & 3 \\ 1680 & -4200 & 2800 & 4 \end{pmatrix},$$

$$\mathbf{R}' = \begin{pmatrix} 3.3829e+03 & -8.1532e+03 & 5.2918e+03 & 2.1117e-01 \\ 0 & 4.8430e+02 & -3.9406e+02 & -3.4665e-01 \\ 0 & 0 & 1.2027e+02 & 1.4257e+00 \\ 0 & 0 & 0 & 5.1568e+00 \end{pmatrix}.$$

Die zweite Methode liefert $\mathbf{F}'\mathbf{B}' = \mathbf{R}'$ mit

$$\mathbf{F}' = \begin{pmatrix} 8.5203e-01 & -3.9783e-01 & -6.0391e-02 & 3.3485e-01 \\ 5.1999e-01 & 6.6415e-01 & -1.5782e-02 & -5.3690e-01 \\ 2.9394e-02 & 4.9382e-01 & 6.0748e-01 & 6.2148e-01 \\ -2.5048e+00 & 4.8843e+01 & 4.9261e+01 & 4.4460e+01 \end{pmatrix},$$

$$\mathbf{B}' = \begin{pmatrix} 120 & 240 & -140 & 1 \\ 1200 & -2700 & 1680 & 2 \\ -2700 & 6480 & -4200 & 3 \\ 1680 & -4200 & 2800 & 4 \end{pmatrix},$$

$$\mathbf{R}' = \begin{pmatrix} 3.5044e+02 & -5.1906e+02 & 4.0357e+02 & 1.2146e+00 \\ 0 & 4.8430e+02 & -3.9406e+02 & -3.4665e-01 \\ 0 & 0 & 1.4227e+01 & 5.3254e+00 \\ 0 & 0 & 0 & 4.2080e+02 \end{pmatrix}.$$

Wendet man beide Verfahren auf zufällige Matrizen mit Ordnungen bis 100 an, so liefert Methode 2 gegenüber Methode 1 leicht größere Fehler gemessen an $\max_{jk} |(\mathbf{F}'\mathbf{B}'-\mathbf{R}')_{jk}|$. Bei Matrizen mit großer Kondition jedoch, wie Hilbertmatrizen ab Ordnung 5, liefert die zweite Methode deutlich größere Fehler. Gerechnet wurden die obigen Beispiele und die erwähnten Tests mit den beiden angegebenen Programmen 14.21,14.22.

Programm 14.21. matlab-Programm für Modifizierung nach Methode 1

```
%Sei eine Zerlegung
%
%                F*B=R, R rechte Dreiecksmatrix gegeben.
%
%Die p-te Spalte von B soll weggelassen werden,
%und in die letzte Spalte von B soll daf\"ur ein neuer Vektor b
%eingesetzt werden. Die so ver\"anderte Matrix hei\ss e Bstrich. Der Aufruf
%
%      function [Fstrich,Bstrich,Rstrich]=mod1(F,B,R,b,p);
%
%berechnet Fstrich, Rstrich mit
%
%   Fstrich*Bstrich=Rstrich, Rstrich rechte Dreiecksmatrix
%
%unter Verwendung der obigen Zerlegung.
%Programm Gerhard Opfer, 10. Januar 1999
%Lit.: G. Opfer, Numerische Mathematik f\"ur Anf\"anger, 2. Aufl.,
%Vieweg, Braunschweig & Wiesbaden, 1994, S. 173/174.
%==============================================================================
function [Fstrich,Bstrich,Rstrich]=mod1(F,B,R,b,p);
[m,n]=size(B);
if m~=n | p>m | p<1, error('mod1 falsche Parameterwahl');end;
%1. Berechnung von Bstrich
%
Bstrich=B;Bstrich(:,p)=[];Bstrich=[Bstrich b];
%2. Die beiden Matrizen H und F werden zu einer Matrix HF zusammengefa\ss t.
%
h=F*b; HF=R;HF(:,p)=[]; HF=[HF,h,F];
%3. Elimination und Berechnung von Rstrich unf Fstrich
%
for i=p:m-1
  if abs(HF(i,i))>abs(HF(i+1,i)) %nicht vertauschen
    q=HF(i+1,i)/HF(i,i);
    HF(i+1,i+1:2*m)=HF(i+1,i+1:2*m)-HF(i,i+1:2*m)*q;
    HF(i+1,i)=0; else           %vertauschen
    hilfs=HF(i,:);
    HF(i,:)=HF(i+1,:);
    HF(i+1,:)=hilfs;            %Vertauschung zu Ende
    q=HF(i+1,i)/HF(i,i);        %jetzt wie oben
    HF(i+1,i+1:2*m)=HF(i+1,i+1:2*m)-HF(i,i+1:2*m)*q;
    HF(i+1,i)=0;
  end; %if
end; %for
%4. Rstrich, Fstrich besetzen
%
Rstrich=HF(:,1:m);  Fstrich=HF(:,m+1:2*m);
return  %Schlu\ss
%==============================================================================
```

Programm 14.22. matlab-Programm für Modifizierung nach Methode 2

```
%Sei eine Zerlegung
%
%                F*B=R, R rechte Dreiecksmatrix gegeben.
%
%Die p-te Spalte von B soll weggelassen werden,
%und in die letzte Spalte von B soll daf\"ur ein neuer Vektor b
%eingesetzt werden. Die so ver\"anderte Matrix hei\ss e Bstrich. Der Aufruf
```

```
%
%      function [Fstrich,Bstrich,Rstrich]=mod2(F,B,R,b,p);
%
%berechnet Fstrich, Rstrich mit
%
%   Fstrich*Bstrich=Rstrich, Rstrich rechte Dreiecksmatrix
%
%unter Verwendung der obigen Zerlegung.
%Programm Gerhard Opfer, 10. Januar 1999
%Lit.: G. Opfer, Numerische Mathematik f\"ur Anf\"anger, 2. Aufl.,
%Vieweg, Braunschweig & Wiesbaden, 1994, S. 175/176.
%Bei schlecht konditionierten Problemen (z. B. Hilbert-Matrizen ab Ordnung 5)
%liefert diese Methode deutlich schlechtere Zerlegungen als die Methode 1
%gemessen an max(max(abs(Fstrich*Bstrich-Rstrich))).
%==============================================================================
function [Fstrich,Bstrich,Rstrich]=mod2(F,B,R,b,p);
[m,n]=size(B);
if m~=n | p>m | p<1, error('mod2 falsche Parameterwahl');end;
%1. Berechnung von Bstrich
%
Bstrich=B; Bstrich(:,p)=[];Bstrich=[Bstrich b];  %wird nicht gebraucht
%2. H wird bestimmt
%
h=F*b; H=R; H(:,p)=[]; H=[H,h];
%3. Einige Hilfsgr\"o\ss en werden bestimmt
%
r=R(p,:)'; r(p)=0;   %r-te Zeile von R, Diagonalelement = 0.
v=R'\r;              %k\"urzer durch Vorw\"artseinsetzen
%4. Hschlange wird bestimmt
Hschlange=H;
Hschlange(p,:)=zeros(1,m);
Hschlange(p,m)=h(p)-v'*h;  %Fehler im Buch!
%5. Fschlange wird bestimmt: nur p-te Zeile von F \"andern
%
Fschlange=F;Fschlange(p,:)=Fschlange(p,:)-v'*F; %v'*F ist Zeile!
%6. Die p-te Zeile von Hschlange, Fschlange mu\ss\ eliminiert
%   und dann hinten angeh\"angt werden.
%
HF=[Hschlange Fschlange];
hilfs=HF(p,:);
HF(p,:)=[];     %p-te Zeile weg
HF=[HF;hilfs];  %hinten wieder dran
Rstrich=HF(:,1:m);Fstrich=HF(:,m+1:2*m);
return %Schlu\ss
%==============================================================================
```