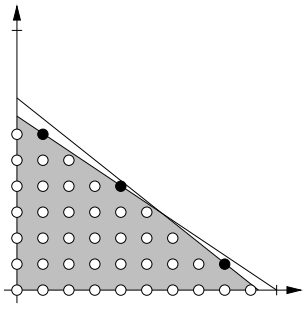
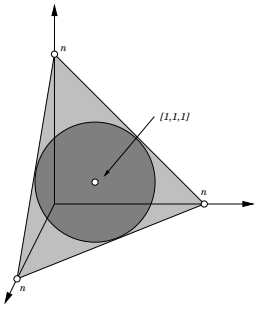


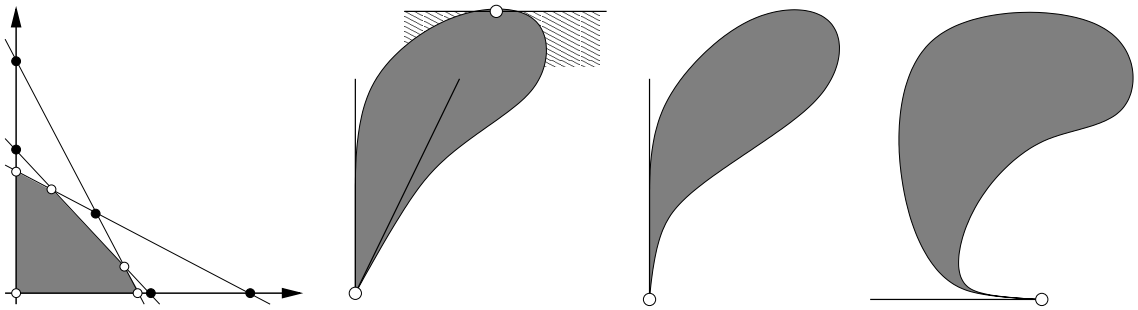


Optimierung

Vorlesung, zuerst gehalten im Sommersemester 2002

Tomas Sauer

Version 2.1
Letzte Änderung: 27.4.2010



Statt einer Leerseite . . . | 0

Nach Ansicht der Leute ist die Erkenntnis, die aus der Erfahrung kommt, mechanisch, wissenschaftlich ist die, die im Geist entsteht und in ihm endet. Aber mir will es eher scheinen, daß die Wissenschaften eitel und irrig sind, die nicht aus der Erfahrung, der Mutter aller Gewißheit, entstanden und nicht in einer gewissen Erfahrung münden!

Leonardo da Vinci

0 Inhaltsverzeichnis

Erst die natürlichen Betrachtungen gemacht, ehe die subtilen kommen, und immer vor allen Dingen erst versucht, ob etwas ganz simpel und natürlich werden könne.

G. Chr. Lichtenberg

1 Optimierung – Grundlagen und Beispiele

Eigentlich ist Optimierung (in "voller" Allgemeinheit) ein ziemlich einfaches Problem, nämlich das Auffinden eines Extremums:

Zu einer Funktion $F : D \to \mathbb{R}$ und $D' \subset D$ finde man ein $x^* \in D'$, so daß

$$F(x^*) \leq F(x) \qquad \textit{oder} \qquad F(x^*) \geq F(x)$$

für alle $x \in D'$.

Zuerst einmal sollte man bemerken, daß es egal ist, ob man die *Zielfunktion* F minimiert oder maximiert, denn man kann die Suche nach einem Maximum von F immer auch als Suche nach einem Minimum von $-F$ auffassen. Je nachdem, was uns gerade genehm ist, können wir daher bei einer *Normalform* des Optimierungsproblems annehmen, daß wir eine Zielfunktion nur maximieren oder nur minimieren wollen.
Weitere generelle Vereinfachungen kann man allerdings weder in der Theorie noch in der Praxis machen:

1. Die Funktion F kann, je nach Problemstellung differenzierbar, stetig oder auch unstetig sein; und selbst wenn F differenzierbar sein sollte, ist es noch lange nicht klar, ob und wie man diese Ableitung auch wirklich bestimmen kann.

2. Die *Auswertung* der Funktion F, das heißt, die Berechnung des Wertes $F(x)$ für ein $x \in D$, kann *teuer* oder *billig* sein. Das kann sich auf Rechenzeit beziehen, denn manchmal kann $F(x)$ nur durch aufwendige Simulationen berechnet werden, oder aber auch auf "echte" Unkosten, wenn die zur Bestimmung von $F(x)$ reale Experimente oder Messungen nötig sind.

3. Der *zulässige Bereich* D' kann ganz D umfassen, insbesondere ist $D' = D = \mathbb{R}^n$ möglich, oder er kann eine echte, "dünne" oder kompakte Teilmenge von D sein.

4. Insbesondere kann D' *implizit* gegeben sein, das heißt, man kennt lediglich eine Funktion $g : D \to \mathbb{R}^m$, so daß
$$D' = \{x \in D \ : \ g(x) = 0\}\,.$$

All diese Situationen verlangen natürlich nach unterschiedlichen Methoden, wenn man die Lösung,

- das Optimum,
- ein Optimum,
- einen nahezu optimalen Wert,

praktisch bestimmen will. Mit derartigen Verfahren, die natürlich wesentlich von der Struktur des Optimierungsproblems abhängen, soll sich diese Vorlesung beschäftigen – **das** Black–Box–Verfahren schlechthin, das jedes Optimierungsproblem löst, kann und wird es nicht geben.
Sehen wir uns nun aber zuerst einmal ein paar Beispiele für Optimierungsprobleme an.

1.1 Lineare und Ganzzahlprogrammierung

Der "einfachste" Fall eines Optimierungsproblems liegt vor, wenn die Zielfunktion $F : \mathbb{R}^n \to \mathbb{R}$ eine *lineare* Funktion der Form

$$F(x) = v^T x, \qquad x \in \mathbb{R}^n,$$

für ein gegebenes $v \in \mathbb{R}^n$ ist. Da eine nichttriviale lineare Funktion auf ganz $\mathbb{R}^n$ alle Werte zwischen $\pm\infty$ annimmt, sind bei derartigen Optimierungsproblemen Nebenbedingungen unvermeidbar, wenn man vernünftige Aussagen machen will. Besonders schön sind dabei natürlich *lineare* Nebenbedingungen der Form

$$a_j^T x \geq b_j, \qquad j = 1, \ldots, N,$$

die man dann schön in der Matrixform $Ax \geq b$ schreiben kann.

Beispiel 1.1 *(Einkauf chemischer Rohstoffe[1])*
Eine Chemiefirma benötigt zwei Chemikalien A und B zur Herstellung ihres Produkts, und zwar mindestens 3t von Stoff A und 4t von Stoff B. Allerdings sind diese beiden Rohstoffe nicht in reiner Form erhältlich, sondern lediglich die beiden Rohstoffe X und Y die A und B enthalten, und zwar wie folgt:

Rohstoff	*Anteil A*	*Anteil B*	*Kosten*
X	*60 %*	*40 %*	*300*
Y	*30 %*	*50 %*	*200*

[1] Dieses Beispiel stammt (in allgemeinerer Form) aus [16, S. 501], die grafische Lösung dort ist ein sehenswertes Kunstwerk.

Hierbei bezeichnet "Kosten" die Summe aus Einkaufspreis und den Aufwendungen für die Gewinnung der Rohstoffe. Was ist nun die günstigste Einkaufspolitik?

Nun, dieses Problem ist relativ einfach zu lösen! Seien nämlich x, y die gekauften Mengen der Rohstoffe X und Y, dann ergibt sich das Optimierungsproblem

$$\min\ 300x + 200y, \qquad \begin{bmatrix} .6 & .3 \\ .4 & .5 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \geq \begin{bmatrix} 3 \\ 4 \end{bmatrix}. \tag{1.1}$$

Jetzt hätten wir fast was vergessen: Da wir keine negativen Mengen einkaufen können[2], müssen wir auch noch $x, y \geq 0$ fordern. Der *zulässige Bereich* für dieses Optimierungsproblem ist in Abb 1.1 dargestellt. Um das Optimierungsproblem (grafisch) zu lösen betrachten wir, daß die

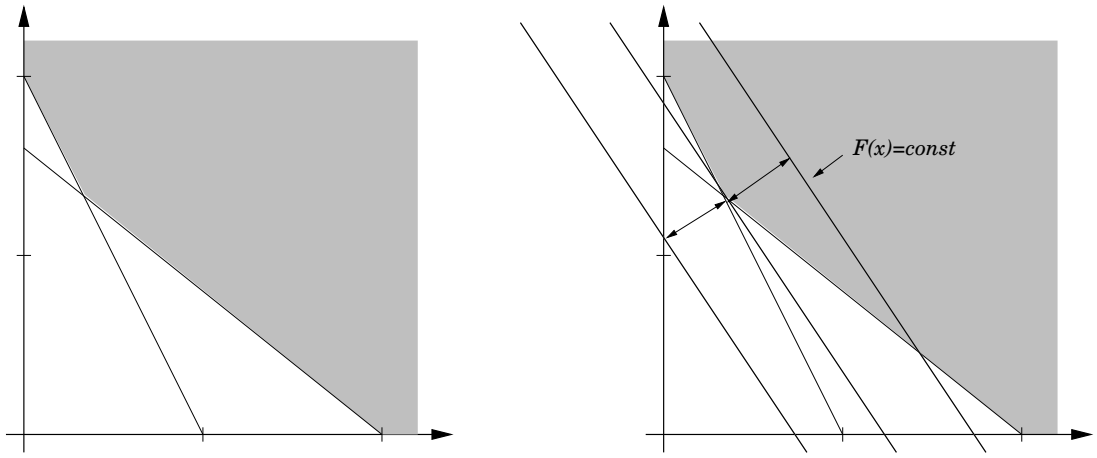


Abbildung 1.1: Der zulässige Bereich (links) und die grafische Lösung (rechts) des Optimierungsproblems aus Beispiel 1.1. Bei der grafischen Lösung verschiebt man die Gerade $F(x) = c$, $c \in \mathbb{R}$, so lange, bis sie den zulässigen Bereich gerade noch berührt – das ist dann offensichtlich der Minimalwert.

Kosten jeweils auf der Gerade $F(x) = c$ den konstanten Wert c haben; verschiebt man also die Gerade nach "links unten" so erhält man günstigere Ergebnisse – natürlich nur, solange die Gerade auch den zulässigen Bereich schneidet, denn ansonsten würde man zwar billig einkaufen, könnte aber nicht produzieren[3]. Also "schieben" wir solange, bis die Gerade den zulässigen Bereich gerade noch berührt und haben die optimale Lösung, nämlich denjenigen Punkt $[x, y]^T$, der sich als

$$\begin{bmatrix} .6 & .3 \\ .4 & .5 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \end{bmatrix} \qquad \Longrightarrow \qquad \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1\frac{2}{3} \\ 6\frac{2}{3} \end{bmatrix}$$

[2]Das ist übrigens keine generelle Annahme für Optimierungsprobleme: In der Finanzmathematik sind sogenannte *Leerverkäufe*, also Verkauf von Aktien, die man gar nicht hat, durchaus nicht verboten.

[3]Bemerkungen über moderne Managementkonzepte sollen an dieser Stelle nicht gemacht werden.

ergibt.

Das alles ist schön und gut, solange nur zwei Parameter zu optimieren sind – dann kann man sich sehr einfach auf grafische Lösungsverfahren zurückziehen. Es kann aber natürlich vorkommen, daß die Anzahl der Parameter sehr viel größer wird.

Beispiel 1.2 *(Transportproblem[4]) Ein Konzern hat m Fabriken, die pro Tag a_j Tonnen einer Chemikalie herstellen, $j = 1, \ldots, m$, und n Verkaufsstellen, die pro Tag einen Mindestbedarf von b_k Tonnen der Chemikalie, $k = 1, \ldots, n$, haben. Der Transport einer Tonne der Chemikalie von Fabrik j zu Verkaufsstelle k kostet c_{jk} Euro. Wie erhält man eine kostenoptimale Versorgung der Verkaufsstellen?*

Hier haben wir es offenbar mit einer ganzen Menge, genauer gesagt mn, Parametern x_{jk}, $j = 1, \ldots, m$, $k = 1, \ldots, n$, zu tun und das Optimierungsproblem lautet

$$\min \sum_{j=1}^{m} \sum_{k=1}^{n} c_{jk}\, x_{jk}$$

unter den Nebenbedingungen

$$\sum_{j=1}^{m} x_{jk} \geq b_k, \quad k = 1, \ldots, n, \qquad \sum_{k=1}^{n} x_{jk} \leq a_k, \quad j = 1, \ldots, m.$$

Und da tut sich grafisch nicht mehr viel ...

Es war nicht ganz zufällig, daß wir in den beiden vorhergegangenen Beispielen von Chemikalien gesprochen haben – man kann nämlich davon ausgehen, daß man die in beliebigen Bruchteilen hin- und herverschieben kann. Das wird anders, wenn es sich um "gequantelte" Objekte handelt, die nicht beliebig unterteilt werden können, denn dann muß man nach *ganzzahligen* Lösungen suchen und landet bei der *Ganzzahlprogrammierung* auch als *"Integer programming"* bekannt

Beispiel 1.3 *(Ganzzahliges Transportproblem[5]) Eine Transportfirma transportiert zwei verschiedene Typen, A und B von Paletten, die unterschiedliche Größe und Gewicht haben und unterschiedlich bezahlt werden:*

Typ	*Größe (cbm)*	*Gewicht (kg)*	*Bezahlung*
A	*2*	*400*	*11*
B	*3*	*500*	*15*

Ein Transportfahrzeug hat eine Zuladung von 3700 kg und ein Ladevolumen von 20 cbm. Was ist die optimale Beladung.

Ganz genau wie vorher können wir unser Optimierungsproblem in der Form

$$\max\ 11x + 15y, \qquad \begin{bmatrix} 2 & 3 \\ 400 & 500 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \leq \begin{bmatrix} 20 \\ 3700 \end{bmatrix}, \qquad x, y \geq 0,$$

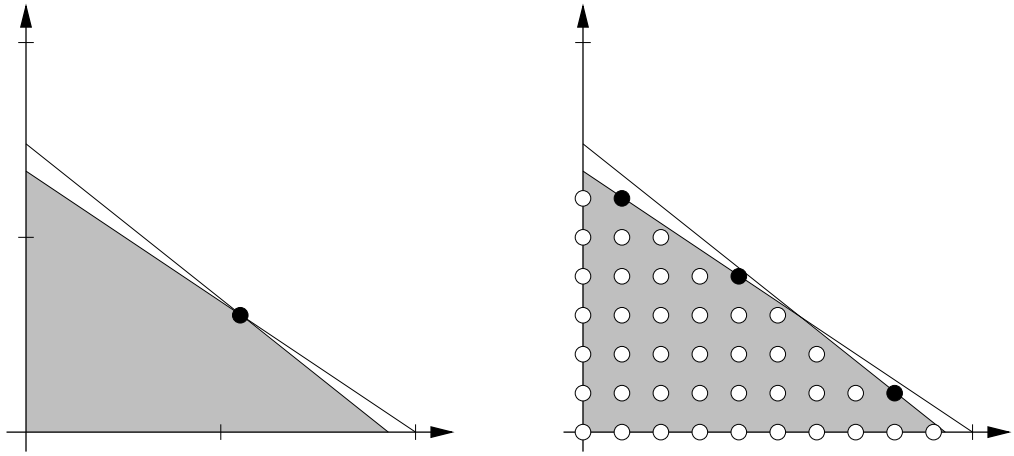


Abbildung 1.2: Links der kontinuierliche zulässige Bereich für Beispiel 1.3 mit der (kontinuierlichen) Optimallösung $\left[5\frac{1}{2}, 3\right]^T$, rechts die ganzzahligen Punkte im zulässigen Bereich, die schwarz markierten liegen gerade so auf dem Rand.

hinschreiben, nur interessieren wir uns jetzt nur noch für die *ganzzahligen* Lösungen und die unter diesen optimale, nämlich $x = y = 4$. Was unseren "graphischen" Ansatz angeht, wird ja eigentlich alles viel leichter, denn man muß halt nun nur noch die Gerade so verschieben, daß sie durch einen der ganzzahligen Punkte im zulässigen Bereich geht und sich da den größten Wert aussuchen. Aber man sieht schon an diesem einfachen Beispiel, daß das ein extrem genaues Arbeiten nötig ist.
Der zweite Ansatz wäre ein schlichtes Ausprobieren, man schreibt ein kleines Programm, das alle zulässigen ganzen Zahlen durchprobiert und so den Optimalwert ermittelt. Das funktioniert noch bei kleinen Beispielen, wird aber schnell hoffnungslos, wenn man die Anzahl der Parameter erhöht. Tatsächlich verwendet ein methodisches Vorgehen algebraische Geometrie, Gröbnerbasen für torische Ideale und Eliminationsideale, siehe [11, Chapter 8] oder [28].

1.2 Quadratische Programmierung

Ein weiterer einfacher Fall ist die Situation, daß F ein *quadratisches* Polynom ist, das heißt,

$$F(x) = x^T A x + b^T x, \qquad x \in \mathbb{R}^n,$$

am besten noch mit einer *symmetrischen, positiv definiten*[6] Matrix $A \in \mathbb{R}^{n \times n}$. Dann hat man es nämlich mit genau einem Minimum zu tun: da

$$\lim_{|x| \to \infty} F(x) = \infty$$

[4] Aus [36, S. 4], aber in leicht verallgemeinerter Form.

[5] Aus [11, S. 359–360].

[6] Positiv definit bedeutet hier *strikt* positiv definit, also $x^T A x > 0$ für $x \neq 0$. *Achtung:* Diese Terminologie ist *nicht* eindeutig in der Literatur.

und, für $j = 1, \ldots, n$,

$$\begin{aligned}\frac{\partial}{\partial x_j} F(x) &= \frac{\partial}{\partial x_j}\left(\sum_{k=1}^{n} a_{kk} x_k^2 + 2 \sum_{1 \le k < \ell \le n} a_{k\ell}\, x_k x_\ell + \sum_{k=1}^{n} b_k\, x_k\right) \\ &= 2a_{jj} x_j + 2\sum_{k \neq j} a_{jk} x_k + b_j = 2\,(Ax)_j + b_j,\end{aligned}$$

haben wir ein Extremum an x^* wenn

$$0 = \nabla F\,(x^*) =: \left[\frac{\partial F}{\partial x_j}\,(x^*) \;:\; j = 1, \ldots, n\right],$$

also wenn $0 = 2Ax^* + b$ oder

$$x^* = -\frac{1}{2} A^{-1}\, b,$$

vorausgesetzt, A ist invertierbar, was sicher der Fall ist, wenn A positiv definit ist. Mindestens ein Minimum muß es aber geben, also ist x^* das eindeutige Minimum – wie man das berechnet, und zwar effizient, das ist dann wieder eine andere Frage.

Beispiel 1.4 *(Portfolio–Optimierung[7])*
Gegeben seien n Investitionsmöglichkeiten mit "Return" oder Auszahlung r_j, $j = 1, \ldots, n$. Diese Auszahlungen betrachtet man als Zufallsvariable, von denen jeweils der Erwartungswert $\mu_j = E\,[r_j]$ und die Varianz $\sigma_j^2 = E\left[(r_j - \mu_j)^2\right]$, $j = 1, \ldots, n$, bekannt sind. Ein Portfolio[8] *besteht nun aus x_j Anteilen der Investition Nummer j, $j = 1, \ldots, n$, der Einfachheit so normiert, daß $x_1 + \cdots + x_n = 1$. Die Auszahlung des Portfolio ist dann*

$$R = \sum_{j=1}^{n} r_j\, x_j$$

und die erwartete Auszahlung

$$E[R] = E\left[\sum_{j=1}^{n} r_j\, x_j\right] = \sum_{j=1}^{n} \mu_j\, x_j = \mu^T x.$$

Mit Hilfe der Kovarianzmatrix

$$K := \left[\frac{E\,[(r_j - \mu_j)\,(r_k - \mu_k)]}{\sigma_j\, \sigma_k} \;:\; j, k = 1, \ldots, n\right]$$

der r_j ist dann

$$\sigma_R := E\left[(R - E[R])^2\right] = x^T G x,$$

[7] Aus [36, S. 216].
[8] Der ganz einfachen Form.

wobei

$$G = \Sigma^T K \Sigma = \left[E\left[(r_j - \mu_j)\,(r_k - \mu_k) \right] \;:\; j,k = 1,\ldots,n \right]$$

und

$$\Sigma = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{bmatrix}.$$

Nun sieht man σ_R als das Risiko *des Portfolios an und wählt, für einen Parameter $\kappa \in [0,\infty)$, der die "Risikofreude" des Anlegers widerspiegeln soll*[9]*, die Anteile der Investitionen als Lösung des quadratischen Optimierungsproblems*

$$\max\, \mu^T x - \kappa\, x^T G x, \qquad \sum_{j=1}^{n} x_j = 1, \quad x_j \geq 0.$$

Beispiel 1.5 (Glättende Funktionen und Lerntheorie) *Gegeben seien nun Punkte x_j, $j = 0,\ldots,N$, und vorgeschriebene oder gemessene Werte y_j, $j = 0,\ldots,N$, und man möchte gerne aus diesen diskreten oder "abgetasteten" Werten eine Funktion rekonstruieren, oder auch ein zugrundeliegendes "Bildungsgesetz". Der erste Ansatz, der einem sofort in den Sinn kommt, wäre* Interpolation*, also die Bestimmung einer Funktion f mit $f(x_j) = y_j$, $j = 0,\ldots,N$, siehe z.B. [29, 31, 41]. Das ist an sich nichts schlimmes, wirft aber sofort die folgenden Probleme auf*

- *Es gibt unendlich viele Funktionen, die das Interpolationsproblem lösen, das Problem ist also* schlechtgestellt*. Man kann dem Problem dadurch begegnen, daß man f als*

$$f = \sum_{j=0}^{N} c_j\, f_j, \qquad \boldsymbol{c} = (c_0,\ldots,c_N) \in \mathbb{R}^{N+1}$$

  ansetzt und den $N+1$–dimensionalen Funktionenraum der f_j passend wählt[10]*.*

- *Die Werte y_j sind oftmals nicht genau, sondern fehlerbehaftet, z.B. bei Meßwerten, und dieses* Rauschen *kann durchaus beträchtliche Ausmaße haben.*

Vor allem das zweite Problem sorgt dafür, daß Interpolanten in vielen Fällen nicht wirklich geeignet sind, um Funktionen zu konstruieren, weswegen man sich mit einem anderen Ansatz behilft: Man wählt nicht $N+1$ Funktionen, sondern $M+1$ Funktionen, wobei M mit N nichts zu tun haben muß, es kann größer oder kleiner als N sein, ja selbst $M = N$ ist nicht verboten und sucht zuerst mal nach der Lösung des Minimierungsproblems

$$\min \left\{ \sum_{j=0}^{N} |f(x_j) - y_j|^2 \;:\; f = \sum_{j=0}^{M} c_j f_j \right\}. \tag{1.2}$$

[9] Je kleiner κ, desto weniger wird das "Risiko", das sich in der Varianz versteckt, in Betracht gezogen und desto mehr zählt die Auszahlung, genauer gesagt, die erwartete Auszahlung.

[10] Möglicherweise in Abhängigkeit von den x_j, aber warum auch nicht? Niemand erwartet an dieser Stelle die ultimative Universallösung.

Da

$$\begin{bmatrix} f(x_0) \\ \vdots \\ f(x_N) \end{bmatrix} = \begin{bmatrix} c_0 f_0(x_0) + \cdots + c_M f_M(x_0) \\ \vdots \\ c_0 f_0(x_N) + \cdots + c_M f_M(x_N) \end{bmatrix} = \begin{bmatrix} f_0(x_0) & \dots & f_M(x_0) \\ \vdots & \ddots & \vdots \\ f_0(x_N) & \dots & f_M(x_N) \end{bmatrix} \begin{bmatrix} c_0 \\ \vdots \\ c_M \end{bmatrix},$$

oder, in Kurzform, $f(\boldsymbol{x}) = \boldsymbol{F}\boldsymbol{c}$*, ist die zu minimierende Funktion*

$$\|\boldsymbol{F}\boldsymbol{c} - \boldsymbol{y}\|_2^2 = (\boldsymbol{F}\boldsymbol{c} - \boldsymbol{y})^T (\boldsymbol{F}\boldsymbol{c} - \boldsymbol{y}) = \boldsymbol{c}^T \boldsymbol{F}^T \boldsymbol{F}\boldsymbol{c} - 2\boldsymbol{y}^T \boldsymbol{F}\boldsymbol{c} + \boldsymbol{y}^T \boldsymbol{y},$$

was bekanntlich[11] *durch die Lösung des Gleichungssystems*

$$\boldsymbol{F}^T \boldsymbol{F}\boldsymbol{c} = \boldsymbol{F}^T \boldsymbol{y} \tag{1.3}$$

minimiert wird. OK, wo ist nun das Problem? Die Lösungsfunktion f *wird immer noch versuchen, den vorgegebenen Werten so gut zu folgen, wie es geht, insbesondere wird sie an allen Punkten interpolieren, wenn es ihr mglich ist, denn dann ist der Fehler Null und besser geht es ohnehin nicht. Und damit ist unser Rauschproblem wieder da, Rauschen in den Daten* y_j *kann durchaus wieder in der Funktion landen und zu Oszillationen führen. Um das zu unterbinden,* verbieten *wir einfach der Funktion* f *das Oszillieren, indem wir zum Minimierungsproblem ein* Glättefunktional *hinzufügen, das zu viel Oszillation bestraft, beispielsweise das sehr beliebte*[12]

$$\begin{aligned} \int |f''(x)|^2 \, dx &= \int \left| \sum_{j=0}^{M} c_j f_j''(x) \right|^2 dx = \int \sum_{j,k=0}^{M} c_j c_k f_j''(x) f_k''(x) \, dx \\ &= \int \boldsymbol{c}^T \begin{bmatrix} f_0''(x) f_0''(x) & \dots & f_0''(x) f_M''(x) \\ \vdots & \ddots & \vdots \\ f_M''(x) f_0''(x) & \dots & f_M''(x) f_M''(x) \end{bmatrix} \boldsymbol{c} = \boldsymbol{c}^T \boldsymbol{F}' \boldsymbol{c}, \end{aligned}$$

wobei $\boldsymbol{F}'$ *das Integral über die Matrix ist. Für einen Parameter* $\lambda \geq 0$ *erhalten wir dann das modifizierte Optimierungsproblem*

$$\min_{\boldsymbol{c}} \|\boldsymbol{F}\boldsymbol{c} - \boldsymbol{y}\|_2^2 + \lambda \boldsymbol{c}^T \boldsymbol{F}' \boldsymbol{c},$$

dessen Lösung über das Gleichungssystem

$$\left(\boldsymbol{F}^T \boldsymbol{F} + \lambda \boldsymbol{F}'\right) \boldsymbol{c} = \boldsymbol{F}^T \boldsymbol{y} \tag{1.4}$$

gefunden wird. Diese Lösungsfunktion versucht nun, in Abhängigkeit vom Parameter λ*, der die "Prioritäten" fixiert, die vorgegebenen Werte gut zu approximieren, aber eben nicht um jeden Preis, sondern auf möglichst glatte Art und Weise.*

Das Verfahren ist alt, um nicht zu sagen klassisch, und beinhaltet den guten alten "smoothing spline" aus Kurvenapproximation und Statistik, aber ganz genauso moderne Ansätze wie Lerntheorie.

[11] Das geht wie in der Schule: Ableiten und gleich Null setzen.

[12] Die Integralgrenzen lassen wir hier bewußt weg, das ist "Detailkram" und uns geht es ja hier schließlich um das Prinzip!

1.3 Platinen und Handlungsreisende

Ein klassisches Optimierungsproblem aus dem Bereich der "kombinatorischen" Optimierung ist das *Travelling Salesman Problem* (TSP), bei dem ein Handlungsreisender eine Tour entwickeln muß, bei der jede von n Städten mindestens einmal angefahren werden muß und bei der der gesamte zurückgelegte Weg minimiert werden muß.

Bezeichnet d_{jk}, $j, k = 1, \ldots, n$, den Abstand[13], zwischen den Städten Nummer j und k, dann besteht eine Formulierung des Travelling Salesman Problem in der Bestimmung eines Vektors $J = (j_1, \ldots, j_N) \in \{1, \ldots, n\}^N$, der den Wert

$$d(I) = \sum_{k=1}^{N-1} d_{j_k, j_{k+1}}$$

unter der Nebenbedingung

$$\{1, \ldots, n\} = \{j_1, \ldots, j_N\},$$

daß jede Stadt *mindestens einmal* besucht wird, minimiert. Diese Nebenbedingung bedeutet, daß $N = n$ ist, daß also jede Stadt einmal besucht wird, oder daß $N = n + 1$ und $j_1 = j_N$ ist, was einer Rundreise entspricht; man könnte auch noch $j_1 = j_N = j^*$ fordern, was heißt, daß der Handlungsreisende genau in der Stadt mit Index j^* – dem Firmensitz zum Beispiel – mit seiner Tour beginnen und enden muß.

Auch wenn der Handlungsreisende ein Klassiker ist, ist er doch nicht wirklich das praktische, realistische Problem. Trotzdem gibt es Varianten davon, die in praktischen Anwendungen ganz natürlich auftreten.

Beispiel 1.6 *(Bohrlöcher in Platinen) In einer Platine sind für die Bestückung mit ICs oder Halbleitern Löcher zu bohren, und zwar nicht 10 oder 20, sondern Größenordnungen von mehreren Hundert oder sogar Tausend Löchern. Man bestimme den kürzesten oder schnellsten*[14] *Weg für den Bohrroboter. Ein ähnliches Problem taucht auch bei der späteren Bestückung der Platine und beim Anbringen der Lötpunkte auf.*

Ein weiteres Beispiel für Varianten des TSP ist die Produktionsplanung, bei der Prozesse optimal auf verschiedenen Ressourcen verteilt werden sollen – dabei kann es sich um Produktionsprozesse und Maschinen, Flugrouten und Flugzeuge, oder auch um Vorlesungen und Räume, Dozenten und Studenten bei der Stundenplanoptimierung handeln. Eine nette und untechnische Übersicht ist [17], siehe auch [1].

Der "Reiz" des Travelling Salesman besteht darin, daß die *Komplexität* $K(n)$ des Problems, also die Anzahl der Rechenoperationen, die nötig sind, um die optimale Lösung zu berechnen, nicht polynomial in n beschränkt werden kann, das heißt, es ist

$$\lim_{n \to \infty} \frac{K(n)}{|p(n)|} = \infty$$

[13] Oder die Reisezeit, z.B. in der "DB–Metrik", oder die Reisekosten, oder eine gewichtete Mischung aus all dem ...

[14] Nicht immer ist der kürzeste Weg auch der schnellste, wenn man Beschleunigung und Abbremsen in Betracht zieht. Man beachte aber, daß die Weglänge eine sehr *einfache* Zielfunktion darstellt, die benötigte Zeit aber auf hochgradig *komplizierte* Art von der Reihenfolge der Punkte abhängen kann.

für jedes Polynom p. Damit ist es für *realistische* Werte von n ziemlich hoffnungslos *die* exakte Optimallösung berechnen zu wollen; überraschenderweise gibt es aber (heuristische) Verfahren, die gute bis sehr gute Lösungen[15] sehr schnell bestimmen. Man kann vielleicht noch nicht mal beweisen, daß sie immer funktionieren, aber sie tun es trotzdem.

1.4 Noch drei Beispiele

Beispiel 1.7 *(Virtuelles Scharfstellen des menschlichen Auges[16]) Bei der Untersuchung von Sehstörungen, die durch Schädigungen der Hornhaut (z.B. Kratzer, Narben, aber auch Verkrümmungen) verursacht sind, ist eine wichtige Größe die* Punktetrennung*, das heißt, inwieweit der Patient imstande ist, zwei punktförmige Lichtquellen zu unterscheiden. Hierbei wird die Hornhautvorderfläche mit einem sogenannten* Videokeratoskop *vermessen, wohingegen die weiteren brechenden Flächen des Auges wie Hornhauthinterfläche und Linse numerisch simuliert werden. Um allerdings "gute" Ergebnisse zu erhalten, muß das Auge scharfgestellt werden, wozu man gewisse Parameter (z.B. Öffnungswinkel der Linse) passend wählen muß – witzig ist dabei, daß das Auge durch eine "falsche" Fokussierung Schädigungen der Hornhaut kompensieren kann und im Normalfall auch wird. Zur Einstellung dieser Parameter wählt man eine punktförmige Lichtquelle in der vorgegebenen Entfernung und berechnet für N Lichtstrahlen und eine vorgegebene "Einstellung" des Auges die* Varianz *der Auftreffpunkte $P_j \in \mathbb{R}^3$, $j = 1, \ldots, N$ auf der Netzhaut:*

$$V^2 = \frac{1}{N-1} \sum_{j=1}^{N} \|P_j - \mu\|_2^2, \qquad \mu = \frac{1}{N} \sum_{j=1}^{N} P_j.$$

Dann verändert man iterativ[17] die Parameter, bis man die optimale Fokussierung erreicht hat.

Beispiel 1.8 *(Parameterbestimmung für elektronische Schwingkreise) Zur Erzeugung von elektromagnetischen Schwingungen, beispielsweise in Mobiltelefonen, wird die Frequenz eines elektronischen Schwingkreises durch Modifikation einer Vielzahl (30 und mehr ist hier keine Seltenheit) von Parametern beeinflußt (Schalter, regelbare Widerstände und Kondensatoren). Der Schwingkreis selbst ist durch einen Chip realisiert, dessen Verhalten in Abhängigkeit von den Eingangsparametern äußerst komplex ist und nicht mehr berechnet, sondern lediglich simuliert werden kann.*
Mathematisch hat man es also beispielsweise mit einer Funktion $F : D \to \mathbb{R}$, $D \subset \mathbb{R}^{30}$ zu tun, für die man zu einer vorgegebenen Frequenz ω das Optimierungsproblem

$$\min |F(x) - \omega|, \qquad x \in D,$$

lösen möchte. Problematisch wird dieses Problem aus mehreren Gründen:

1. *Die Funktion F ist nicht differenzierbar, oftmals nicht einmal stetig, weil der Chip beim Erreichen bestimmter Schwellenwerte "umschalten" kann.*

[15] Normalerweise nur um wenige Prozent schlechter als die Optimallösung!
[16] Siehe [32].
[17] Das Stichwort heißt *Newtonverfahren*, das werden wir auch später wohl noch kennenlernen.

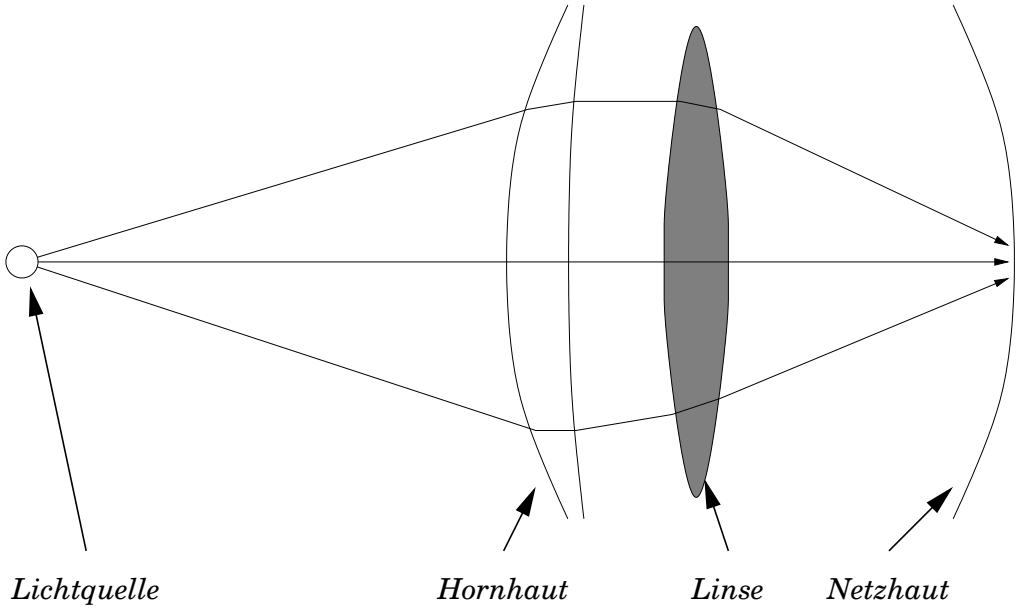


Abbildung 1.3: Stark vereinfachtes Modell des menschlichen Auges. Lichtstrahlen werden sowohl an der Hornhautvorder- wie -rückfläche sowie der Linse gebrochen (die Brechungsindizes der "Materialien" gegen Luft und Kammerwasser sind glücklicherweise bekannt), bevor sie die gekrümmte Hornhaut erreichen. Das "richtige" Modell ist *dreidimensional* und muß außerdem berücksichtigen, daß die Sehkraft der Netzhaut mit zunehmendem Abstand vom Mittelpunkt rapide abnimmt. Das *Sehzentrum* des menschlichen Auges ist lediglich ein stecknadelkopfgroßer Punkt.

2. *Manche der Parameter sind kontinuierlich (Widerstände, Kondensatoren), manche diskret (Schalter).*

3. *Jede Auswertung der Funktion F dauert mehrere Minuten, weil für jede Auswertung ein Simulationsprozess durchgeführt werden muß.*

Beispiel 1.9 *(Bahngeschwindigkeit für hydraulische Roboter) Ein hydraulischer*[18] *Zwei–Arm–Roboter, siehe Abb. 1.4 soll eine vorgegebene Bahn abfahren, die in als Kontrollpolygon eines kubischen Splines mit einfachen Knoten*[19] *gegeben ist; daß die Bahnkurve C^2 ist, ist übrigens eine natürliche physikalische Nebenbedingung, da die Ventilöffnung der Hydraulik, die praktisch die Winkelbeschleunigung des Roboters ist, nur kontinuierlich verändert werden kann. Die wählbaren Parameter sind die* Knoten *des Splines, das heißt, die "Umschaltpunkte" der Steuerung, die Nebenbedinungen sind erstens, daß die Reihenfolge der Knoten beibehalten wird*

[18] Der Vorteil von hydraulischen Robotern besteht darin, daß sie keine Motoren mit sich "herumschleppen" müssen und deswegen ein sehr gutes Verhältnis zwischen Nutzlast und Eigengewicht haben.

[19] Was man ja in der Numerik–Vorlesung lernt oder lernen sollte, siehe [41].

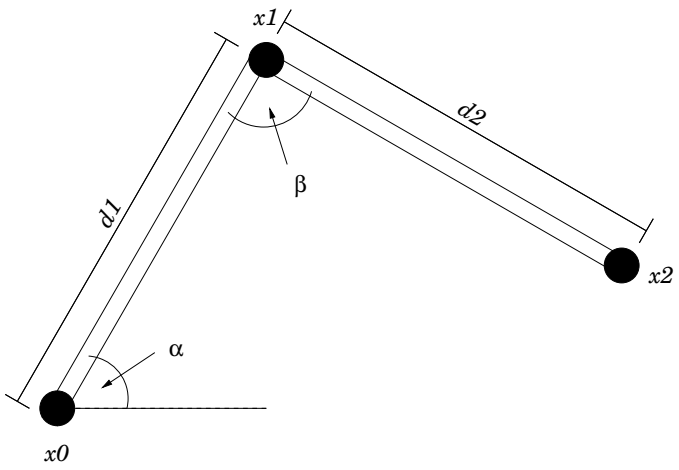


Abbildung 1.4: Ein Roboter mit zwei Gelenken, der Endpunkt des Arms kann in einem Teilbereich des $\mathbb{R}^2$ geführt werden.

und zweitens, daß die zweite Ableitung der Splinekurve im Absolutbetrag beschränkt bleibt, denn aus technischen Gründen kann die Winkelgeschwindigkeit nicht beliebig groß werden. Unser Optimierungsproblem nimmt also die Form

$$\min\, t_n - t_0, \qquad t_0 \le t_1 \le \cdots \le t_n, \qquad |s''(t)| \le M, \quad t \in [t_0, t_n]$$

an, allerdings ist ein "allgemeines" Optimierungsverfahren (z.B. von `Matlab`*) extrem aufwendig. In [27] wurde ein einfaches Verfahren entwickelt, das die analytischen Eigenschaften kubischer Splines ausnutzt und in wesentlich kürzerer Zeit nahezu optimale Lösungen findet. Durch Entfernen überflüssiger Knoten (wenn s auf einem Teilbereich linear ist, braucht man da keine inneren Knoten) kann man außerdem das starke "Wackeln" eliminieren, das die Optimallösung auszeichnet.*

. . . the calculations, be it remembered, of the hard-headed, strong handed, exemplary working men . . .

P. Smyth, *The Great Pyramid*

2 Lineare Optimierungsprobleme

Das Ziel von Optimierungsverfahren besteht ja, wie schon erwähnt, darin, eine Zielfunktion unter vorgegebenen Nebenbedingungen zu maximieren oder zu minimieren. Sind sowohl die Zielfunktion, wie auch die Nebenbedingungen *linear*[20], dann spricht man überraschenderweise von einem *linearen Optimierungsproblem.* Ein solches Optimierungsproblem läßt sich immer schreiben als

$$\begin{array}{rcl} c^T x & = & \max, \\ Ax & \geq & b, \end{array} \qquad A \in \mathbb{R}^{m\times n}, \quad c, x \in \mathbb{R}^n, \quad b \in \mathbb{R}^m, \tag{2.1}$$

wobei für zwei Vektoren $x, y \in \mathbb{R}^m$ die Halbordnung[21] $x \leq y$ bedeutet, daß $x_j \leq y_j$, $j = 1, \ldots, m$. Fangen wir mit den Nebenbedingungen an: Eine allgemeine *lineare* Nebenbedingung an x hätte die Form

$$a^T x \geq b \qquad \text{oder} \qquad a^T x \leq b \quad \Longleftrightarrow \quad (-a)^T x \geq (-b)\,,$$

das heißt, wir können immer von Nebenbedingungen der Form $Ax \geq b$ ausgehen. Ähnliches gilt für die Zielfunktion: Würde man $c^T x$ *minimieren* wollen, so kann man genauso gut $(-c)^T x$ maximieren. Durch Einführung sogenannter *"Schlupfvariablen"* $x_{n+1}, \ldots, x_{n+m}$ kann man die Ungleichungen[22] $a_j^T x \geq b$ auf die äquivalente Form

$$a_j^T x - x_{n+j} = b, \quad x_{n+j} \geq 0, \qquad j = 1, \ldots, m,$$

bringen und erhält so durch passende Erweiterung von A, b und c die äquivalente Normalform eines linearen Optimierungsproblems

$$\begin{array}{rcll} c^T x & = & \max, & \\ Ax & = & b, & I \subset \{1, \ldots, n\}. \\ x_j & \geq & 0, \qquad j \in I, & \end{array} \tag{2.2}$$

[20] Eigentlich natürlich affin.

[21] Zur Erinnerung: Halbordnung heißt, daß für $x \neq y$ nicht notwendig eine der beiden Beziehungen $x < y$ oder $x > y$ gelten muß.

[22] Hier und im Rest dieses Kapitels seien a_j^T, $j = 1, \ldots, m$, die Zeilenvektoren der Matrix A.

Übung 2.1 Wie muß man A, b und c aus (2.1) erweitern, so daß (2.2) eine äquivalente Formulierung ist. ♢

Wie wir sehen werden, haben (lineare) Optimierungsprobleme eine Menge mit *konvexer Analysis*, siehe z.B. [39], zu tun – wobei es durchaus so ist, daß sich die Gebiete gegenseitig beeinflußt und motiviert haben und daß sich Resultate des einen Gebiets auch im anderen Gebiet als hilfreich erwiesen haben.

Bevor wir uns ein bißchen die mathematische Theorie ansehen, befassen wir uns erst einmal mit einem "realistischen" Beispiel aus [44, Beispiel 2.1, S. 55].

Beispiel 2.1 *(Produktionsproblem einer Schuhfabrik)*
Eine Schuhfabrik stellt Damen- und Herrenschuhe her, die unterschiedliche Forderungen an Herstellungszeit, Maschinenlaufzeit und Lederbedarf stellen – Ressourcen, die natürlich gewissen Einschränkungen unterliegen. Welche Produktionskombination erzielt den höchsten Gewinn[23]*, wenn die folgenden Parameter vorliegen:*

	Damenschuh	*Herrenschuh*	*Verfügbar*
Herstellungszeit	20	10	8000
Maschinenzeit	4	5	2000
Leder	6	15	4500
Gewinn	16	32	

Aber im Ernst – die mathematische Formulierung dieses Problems ist, in der "Normalform" (2.1)

$$c = \begin{bmatrix} 16 \\ 32 \end{bmatrix}, \qquad A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -20 & -10 \\ -4 & -5 \\ -6 & -15 \end{bmatrix}, \qquad b = \begin{bmatrix} 0 \\ 0 \\ -8000 \\ -2000 \\ -4500 \end{bmatrix},$$

wobei die ersten beiden Zeilen von A und b die Tatsache wiedergeben, daß man keine negative Anzahl von Schuhen produzieren kann.

2.1 Zulässige Punkte

Als erstes sehen wir uns einmal den Bereich an, in dem wir unsere Lösungen suchen, das heißt, die Menge[24]

$$F := F(A, b) := \{x \in \mathbb{R}^n \ : \ Ax \geq b\} \subset \mathbb{R}^n,$$

die *zulässige Menge* oder der *zulässige Bereich* für das Optimierungsproblem.

Definition 2.2 *Eine Menge $\Omega \subset \mathbb{R}^n$ heißt konvex, wenn sie mit je zwei Punkten auch ihre Verbindungsstrecke enthält, das heißt, wenn*

$$x, y \in \Omega \qquad \Longrightarrow \qquad [x, y] := \{(1-\alpha)\ x + \alpha\ y \ : \ \alpha \in [0, 1]\} \subset \Omega. \tag{2.3}$$

[23] Unter der (realistischen ?) Annahme, daß alle Schuhe verkauft werden können.
[24] Das "F" steht für die englische Bezeichnung "feasible".

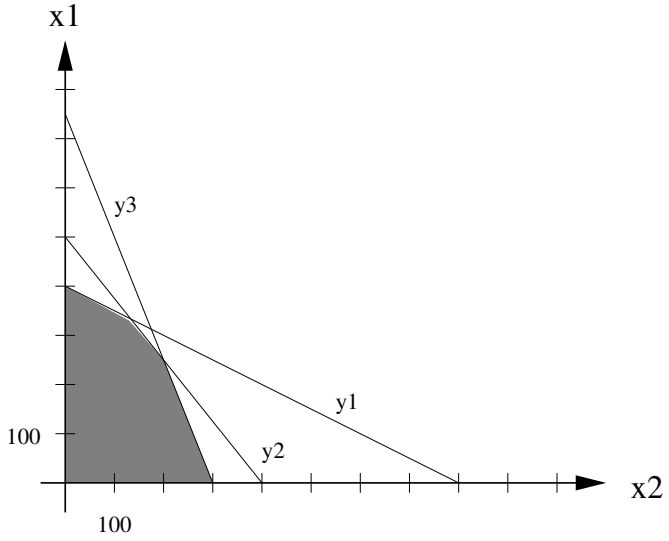


Abbildung 2.1: Zulässige Menge für das Optimierungsproblem aus Beispiel 2.1.

Lemma 2.3 *Für jedes $A \in \mathbb{R}^{m\times n}$ und $b \in \mathbb{R}^m$ ist die Menge $F(A,b)$ konvex.*

Beweis: Der (einfache) Beweis ist eine Konsequenz der beiden folgenden Tatsachen:

1. *Halbräume sind konvex*: Ist $a^T x \geq b$ und $a^T y \geq b$, so gilt auch
$$a^T\left((1-\alpha)x+\alpha y\right) = (1-\alpha)\underbrace{a^T x}_{\geq b} + \alpha\underbrace{a^T y}_{\geq b} \geq (1-\alpha+\alpha)b = b.$$

2. *Der Schnitt zweier konvexer Mengen ist konvex:* Sind Ω, Ω' konvex, dann ist, für $x, y \in \Omega \cap \Omega'$,
$$(1-\alpha)x+\alpha y \in \left\{ \begin{array}{c} \Omega \\ \Omega' \end{array} \right. , \quad \alpha \in [0,1] \qquad \Longrightarrow \qquad [x,y] \subset \Omega \cap \Omega'.$$

□

Den Durchschnitt einer endlichen Anzahl von Halbräumen im $\mathbb{R}^n$ bezeichnet man als *konvexes Polyeder*, insbesondere ist also F ein konvexes Polyeder. Die *Ecken* dieses Polyeders sind diejenigen Punkte, die man nicht als Konvexkombination anderer Punkte des Polyeders schreiben kann. Formal heißt das, daß x eine Ecke ist, wenn

$$x \in (y,y')\,, \quad y,y' \in F \qquad \Longleftrightarrow \qquad x = y = y'.$$

Dabei bezeichnet[25]

$$(y,y') = \{\alpha\, y + (1-\alpha)\, y' \;:\; \alpha \in (0,1)\}$$

[25]Das sieht nun sehr stark wie ein *offenes* Intervall, was es aber nur ist, wenn $y \neq y'$ ist, ansonsten ist es einpunktig, abgeschlossen und **nicht** offen. Daß sich "offen" und "abgeschlossen" nicht notwendigerweise gegenseitig ausschließen ist ja hoffentlich bekannt.

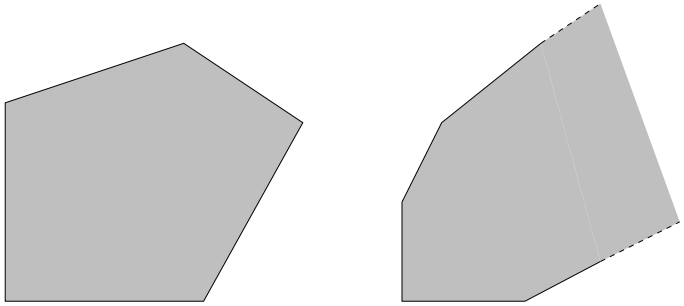


Abbildung 2.2: Ein endliches und ein unendliches konvexes Polyeder.

das *relative Innere* der Strecke $[y, y']$.
Die Ecken von $F(A, b)$ kann man nun mittels Linearer Algebra erhalten: Dazu wählt man eine Indexmenge $J \subset \{1, \ldots, m\}$, $\#J = n$ aus und betrachtet die Teilmatrix und den Teilvektor

$$A_J = \left[a_j^T \ : \ j \in J\right] \quad \in \mathbb{R}^{n\times n} \qquad \text{und} \qquad b_J = [b_j \ : \ j \in J] \quad \in \mathbb{R}^n.$$

Lemma 2.4 *Sei $F = F(A, b)$ das konvexe Polyeder der zulässigen Punkte für das lineare Optimierungsproblem (2.1). Dann ist ein Punkt $x \in F$ genau dann ein Eckpunkt von F, wenn es eine Indexmenge $J \subset \{1, \ldots, m\}$, $\#J = n$, gibt, so daß $A_J x = b_J$ und* $\det A_J \neq 0$.

Beweis: Wir beginnen mit $\Longleftarrow$. Ist $\det A_J \neq 0$, dann ist $x = A_J^{-1} b_J$ ein Randpunkt des Polyeders[26]; wäre außerdem $x = (1-\alpha)y + \alpha y'$ für $y, y' \in F$ und $\alpha \in (0, 1)$, dann wäre

$$b_J = A_J x = A_J\left((1-\alpha)y + \alpha y'\right) = (1-\alpha)\underbrace{A_J y}_{\geq b_J} + \alpha \underbrace{A_J y'}_{\geq b_J},$$

weswegen $A_J y = A_J y' = b_J$, also $y = y' = x$ sein muß. Damit ist x aber ein Eckpunkt.
Umgekehrt ist jeder Eckpunkt x des konvexen Polyeders ein Randpunkt und liegt damit auf dem Rand mindestens eines Halbraums, erfüllt also $a_j^T x = b_j$ für mindestens ein $j \in \{1, \ldots, m\}$. Setzen wir

$$J := J(x) := \left\{1 \leq j \leq m \ : \ a_j^T x = b_j\right\} \subset \{1, \ldots, m\},$$

dann muß $A_J x = b_J$ sein; hat A_J Rang n, dann ist nach der obigen Argumentation x tatsächlich ein Eckpunkt, ansonsten gibt es einen mindestens eindimensionalen Teilraum $Y \subset \mathbb{R}^n$, so daß $A_J y = 0$, also $A_J(x + y) = b_J$ für alle $y \in Y$. Da alle anderen Ungleichungen strikt gelten, also

$$a_j^T x > b_j, \qquad j \notin J,$$

gilt, gibt es ein $\epsilon > 0$, so daß

$$\{x + y \ : \ y \in Y \, \|y\| \leq \epsilon\} \subset F,$$

[26]Nach Voraussetzung ist $x \in F$, erfüllt also die anderen Ungleichungen ebenfalls, das heißt $A_K x \leq b_K$, $K = \{1, \ldots, m\} \setminus J$.

und in dieser Menge kann man x konvex kombinieren. □

Allerdings: Diese Charakterisierung von Eckpunkten über Indexmengen J der Mächtigkeit n gilt nur, wenn x zum Polyeder gehört! Diese Information wurde im Beweis ja auch weidlich ausgenutzt.

Bemerkung 2.5 *Die Existenz einer Indexmenge J, so daß $\det A_J \neq 0$ ist, hat ja eine einfache geometrische Interpretation: Die Hyperebenen*

$$H_j := \left\{ x \in \mathbb{R}^n \ : \ a_j^T x = b_j \right\}, \qquad j \in J,$$

schneiden sich in dem eindeutigen Punkt $x_J := A_J^{-1} b_J$, andernfalls ist der Schnitt, je nach "rechter Seite" b_J leer oder ein nichttrivialer linearer Raum. Und im Normalfall liegen die meisten solchen Schnittpunkte eben nicht *im Polyeder $F(A, b)$, siehe Abb. 2.3.*

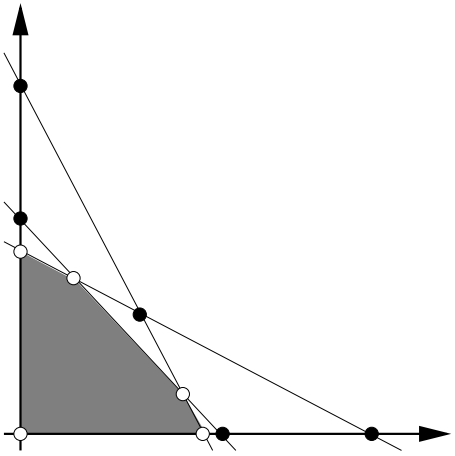


Abbildung 2.3: Alle Schnittpunkte der Nebenbedingungen eines einfachen linearen Optimierungsproblems – einige sind Ecken, einige nicht. Die zugehörige Nebenbedingungsmatrix $A \in \mathbb{R}^{5\times 2}$ hat übrigens die "Standard"–Eigenschaft, daß jede 2×2–Teilmatrix von A invertierbar ist.

Korollar 2.6 *Das konvexe Polyeder $F(A, b)$ hat höchstens $\binom{m}{n}$ Ecken.*

Beweis: Zu jeder Ecke gehören n linear unabhängige Zeilen von A. Da A insgesamt $m \geq n$ Zeilen hat, ist die Anzahl solcher Konfigurationen höchstens $\binom{m}{n}$. □

Jedes *endliche* konvexe Polyeder ist die konvexe Hülle seiner Eckpunkte[27]; dazu erinnern wir uns, daß die *konvexe Hülle* einer Menge $\Omega \subset \mathbb{R}^n$ definiert ist als die kleinste konvexe Menge,

[27] Das bedarf natürlich eines Beweises, aber den schenken wir uns hier erst einmal.

die Ω enthält, also als diejenige Menge, die man erhält, wenn man beliebige endliche *Konvexkombinationen* von Punkten aus Ω bildet. Eine derartige Konvexkombination von $x_1, \ldots, x_m \in \mathbb{R}^n$ ist, als Verallgemeinerung von (2.3), ein Ausdruck der Form

$$\sum_{j=1}^{m} \alpha_j \, x_j, \qquad \alpha_j \geq 0, \quad \sum_{j=1}^{m} \alpha_j = 1,$$

und wir setzen

$$[x_1, \ldots, x_m] = \left\{ \sum_{j=1}^{m} \alpha_j \, x_j \ : \ \alpha_j \geq 0, \sum_{j=1}^{m} \alpha_j = 1 \right\}. \tag{2.4}$$

Die konvexe Hülle von Ω lässt sich dann als

$$[\Omega] := \{ [x_1, \ldots, x_m] \ : \ x_1, \ldots, x_m \in \Omega, \, m \in \mathbb{N} \} \, .$$

beschreiben. Man kann $[\Omega]$ aber auch anders interpretieren, nämlich als Abschluß von Ω unter der Operation "Verbindungslinie" aus (2.3). Dazu betrachtet man die Folge

$$\Omega_0 = \Omega, \qquad \Omega_{j+1} = \Omega_j \cup \{ [x, y] \ : \ x, y \in \Omega_j \} \, ,$$

und sieht leicht, daß

$$[\Omega] = \lim_{j \to \infty} \Omega_j. \tag{2.5}$$

Übung 2.2 Beweisen Sie (2.5). ♢

Um die konvexe Hülle einer endlichen Menge $X \subset \mathbb{R}^n$ etwas handlicher schreiben zu können, ordnen wir ihre Elemente als Spaltenvektoren einer Matrix X an[28] und erhalten, daß

$$[X] = \{ Xu \ : \ u \in \Delta_{\#X} \} \, , \qquad \Delta_n = \left\{ u \in \mathbb{R}^n \ : \ u_j \geq 0, \, \sum_{j=1}^{n} u_j = 1 \right\}. \tag{2.6}$$

Die Menge Δ_n ist das n–dimensionale *Einheitssimplex* – weswegen man die konvexe Menge $[X]$ auch als *Simplex* bezeichnet.

In der Folge wollen wir nun immer annehmen, daß die zulässigen Punkte $F(A, b)$ ein *endliches* Polyeder bilden, das heißt, es gibt $N > 0$, so daß $F(A, b) \subseteq [-N, N]^n$, denn dann ist auch $F(A, b) = [X]$, wobei X die Eckenmenge von $F(A, b)$ ist. Das sollten wir auch beweisen.

Proposition 2.7 *Ist $F(A, b)$ ein* endliches *Polyeder und X die zugehörige Eckenmenge, dann ist*[29] $F(A, b) \subseteq [X]$.

[28] Dies ermöglicht die Verwendung von *"Multisets"*, das sind "Mengen", in denen Vielfachheiten der Elemente erfasst und berücksichtigt werden können.

[29] Die Umkehrung, $[X] \subseteq F(A, b)$ ist klar, denn es ist $X \subset F(A, b)$ und da $F(A, b)$ konvex ist, muss die konvexe Hülle $[X]$ als *kleinste* konvexe Obermenge von X ebenfalls in $F(A, b)$ enthalten sein. Wir behaupten und beweisen also in dieser Proposition nur den "interessanten" Teil der Aussage $F(A, b) = [X]$.

Beweis: Wir bemerken zuerst, daß jede Seite

$$F_J(A,b) = \{x \in F(A,b) \ : \ A_J x = b_J\}, \qquad \emptyset \neq J \subset \{1,\ldots,m\},$$

ebenfalls ein Polyeder ist und zwar eines von der Dimension $\leq n - \#J$, denn schließlich liegt dieses Polyeder ja im Schnitt von $\#J$ Hyperräumen[30].

Sei nun $x \in F(A,b)$, dann ist x entweder eine Ecke und die Aussage der Proposition ist trivialerweise erfüllt, oder es gibt $y \neq y' \in F(A,b)$ und $\alpha \in (0,1)$, so daß $x = \alpha\, y + (1-\alpha) y'$. Die Gerade $\ell : t \mapsto \ell(t) = t\, y + (1-t)\, y'$, $t \in \mathbb{R}$, durch y und y' hat nun die Eigenschaft, daß

$$\ell \cap F(A,b) = \{\ell(t) \ : \ t \in [t_0,t_1]\} = \{\alpha\, \ell\,(t_0) + (1-\alpha)\ (t_1)\}\,.$$

Hier haben wir die Beschränktheit des Polyeders verwendet, denn sonst könnte im schlimmsten Fall die Gerade komplett im Inneren des Polyeders verlaufen. Die beiden Punkte $\ell\left(t_{0/1}\right)$ sind Randpunkte von $F(A,b)$, gehören also zu Seiten des Polyeder und sind so – per Induktion[31] – Konvexkombinationen von Ecken. Da aber x auch eine Konvexkombination dieser beiden Punkte ist, gilt $x \in [X]$. □

2.2 Konvexe Funktionen

Auf konvexen Polyedern besonders einfach zu optimieren sind *konvexe Funktionen*.

Definition 2.8 *Eine Funktion $f : \Omega \subset \mathbb{R}^n \to \mathbb{R}$ heißt* konvex*, wenn für alle $x, y \in \Omega$ und alle $\alpha \in [0,1]$*

$$f\left((1-\alpha)x + \alpha y\right) \leq (1-\alpha)\ f(x) + \alpha\ f(y). \tag{2.7}$$

Bemerkung 2.9 *(Konvexität)*

1. *Eine Funktion f heißt* konkav*, wenn $-f$ konvex ist. Alle Aussagen über Maxima konvexer Funktionen ergeben sofort auch Aussagen über* Minima *konkaver Funktionen.*

2. *Mit der Notation (2.6) kann man Konvexität einer Funktion auch als*

$$f\,(Xu) \leq \sum_{j=1}^{N} u_j f\,(x_j)\,, \qquad u \in \Delta_N,\ X = [x_1 \cdots x_N] \in \mathbb{R}^{n\times N}, \tag{2.8}$$

   beschreiben.

3. *Für affine Funktionen der Form $x \mapsto a^T x + b$ gilt in (2.8) Gleichheit, das heißt, affine Funktionen sind konvex und konkav.*

[30] Wobei wir redundante Nebenbedingungen ausschließen wollen – also den Fall, daß A zwei oder mehr identische oder abhängige Zeilen enthält.

[31] Der Induktionsanfang ist einfach: Eindimensionale Seiten sind Intervalle, also sicherlich Konvexkombinationen ihrer Ecken, der Endpunkte des Intervalls.

Das folgende Resultat sagt uns, wo wir nach den Optimallösungen suchen müssen – in den Ecken des konvexen Polyeders.

Satz 2.10 *Eine konvexe Funktion nimmt auf einem endlichen konvexen Polyeder ihr Maximum in einer der Ecken an.*

Beweis: Sei $X \in \mathbb{R}^{n\times N}$ die Eckenmenge des konvexen Polyeders, das heißt, jeder Punkt $x \in [X]$ hat die Form $x = Xu$, $u = u(x) \in \Delta_N$. Wegen der Konvexität von f ist dann

$$f(x) = f(Xu) \leq \sum_{j=1}^{N} u_j f(x_j) \leq \underbrace{\left(\sum_{j=1}^{N} u_j\right)}_{=1} \max_{j=1,\ldots,N} f(x_j) \leq \max_{j=1,\ldots,N} f(x_j).$$

□

Dieser Satz gäbe eine Möglichkeit, das Optimierungsproblem zu lösen: Man braucht ja "nur" die Ecken von F zu bestimmen, sich die Zielfunktionen an diesen anzusehen und wird unter diesen endlich vielen Werten auch das Maximum finden. Trotzdem ist das mit unseren bisherigen Mitteln nicht praktikabel, denn das Auffinden der Ecken, das heißt, die Bestimmung von n linear unabhängigen Zeilen in der Matrix A, dauert einige Zeit. Außerdem entspricht ja nicht für jede Indexmenge J so daß $\det A_J \neq 0$ die Lösung von $A_J x = b_J$ einer Ecke von F – es kann und wird, wie in Bemerkung 2.5 gezeigt, in vielen Fällen passieren, daß $(Ax)_j < b_j$ für ein $j \in \{1,\ldots,m\} \setminus J$ gilt und damit $x \notin F$ ist.

2.3 Der Simplexalgorithmus

Der Simplexalgorithmus, der auf Dantzig[32] zurückgeht, siehe [12], stellt wohl eines der wichtigsten numerischen Verfahren dar[33] So schreibt Laszlo Lovasz[34] 1980:

> *If one would take statistics about which mathematical problem is using up most of the computer time in the world, then ... the answer would probably be linear programming.*

Dantzig selbst bemerkt

> *The tremendous power of the simplex method is a constant surprise to me.*

Das Verfahren nutzt ebenfalls die Tatsache aus, daß die *lineare* und damit sowohl konvexe als auch konkave Zielfunktion ihr Extremum in einer Ecke des konvexen Polyeders F annehmen muß. Anstatt nun alle Ecken des Polyeders der Reihe nach abzusuchen, wird bei diesem Verfahren zu einer bekannten Ecke eine "Nachbarecke" bestimmt, an der die Zielfunktion einen

[32] George Dantzig, 1918–2005, entwickelte dieses "mechanisierte" Planungsverfahren unter dem Namen "linear programming" 1947 für die U.S. Air Force. Später arbeitete er für die bekannte *RAND corporation* und wurde 1966 Professor für Operations Research in Stanford. Ein Nachruf auf ihn erschien 2005 sogar im *Time Magazine*.

[33] Nummer 1 ist fraglos die schnelle Fouriertransformation von Cooley und Tukey [10].

[34] Wer auch immer das ist.

größeren Wert annimmt – auf diese Weise hofft man, sich relativ schnell und systematisch bis zum Maximum vorzuarbeiten – das muß natürlich nicht unbedingt rasend schnell funktionieren: Man kann Beispiele angeben, bei denen der Simplexalgorithmus alle Ecken ablaufen muß, bevor er das Optimum erreicht und insofern nicht schneller als "systematisches" Suchen ist. Trotzdem funktioniert das Ganze, denn wenn sich die Zielfunktion beim Übergang zu allen Nachbarecken verkleinert, dann dann tut sie das für keine andere Ecke.

Um richtig konkrete Aussagen machen zu können, müssen wir aber zuerst einmal formal klarstellen, was eine "Nachbarecke" eigentlich ist. Betrachtet man eine Ecke x des konvexen Polyeders F mit Eckenmenge X, dann können wir wieder

$$J = J(x) = \left\{ j \in \{1, \ldots, m\} \ : \ a_j^T x = b_j \right\}$$

mit $\#J \geq n$ einführen; geometrisch ist x ja gerade der Schnittpunkt der Hyperebenen

$$\left\{ y \ : \ a_j^T y = b \right\}, \qquad j \in J.$$

Die *Nachbarecken* von x sind nun gerade die Elemente von X, die auf mindestens einer der Hyperebenen durch x liegen, also

$$V_x := \{ y \in X \setminus \{x\} \ : \ \# \left(J(x) \cap J(y) \right) \geq n - 1 \},$$

das heißt, mindestens eine der Ungleichungen in $A_J y \geq b_J$ muß zur Gleichheit werden.

Proposition 2.11 *Sei x^* eine Ecke des konvexen Polyeders F und seien x_j, $j \in J$, die Nachbarecken von x^*. Ist*

$$c^T x_j < c^T x^*, \qquad j \in J,$$

dann ist $c^T x < c^T x^*$ für alle $x \in F$.

Beweis: Da der *konvexe Kegel*

$$C_{x^*} := x^* + \left\{ \sum_{j \in J} \lambda_j \ (x_j - x^*) \ : \ \lambda_j \geq 0 \right\},$$

der von x und V_x aufgespannt wird, das konvexe Polyeder F enthält, können wir jedes $x \in F \setminus \{x^*\}$ als

$$x = x^* + \sum_{j \in J} \lambda_j \ (x_j - x^*), \qquad \lambda \neq 0,$$

schreiben und erhalten somit, daß

$$c^T x = c^T x^* + \sum_{j \in J} \lambda_j \ \underbrace{\left(c^T x_j - c^T x^* \right)}_{<0} < c^T x^*,$$

wie behauptet. □

Bemerkung 2.12 *Ersetzen wir die strikte Ungleichung "$<$" in Proposition 2.11 durch "$\leq$", dann gilt die Aussage immer noch! Mit anderen Worten: Wir haben* ein *Maximum erreicht, wenn wir uns beim Übergang zu Nachbarecken nicht verbessern können, und wir haben* das *Maximum erreicht, wenn wir uns beim Übergang zu Nachbarecken nur verschlechtern können*[35]

Um uns das Leben einfacher zu machen, nehmen wir erst einmal an, daß der Nullpunkt eine zulässige Ecke von F ist, und zwar dergestalt, daß

$$A = \begin{bmatrix} I_n \\ \widetilde{A} \end{bmatrix}, \qquad b = \begin{bmatrix} 0 \\ \widetilde{b} \end{bmatrix}, \qquad \widetilde{A} \in \mathbb{R}^{m-n \times n}, \quad 0 \geq \widetilde{b} \in \mathbb{R}^{m-n}. \tag{2.9}$$

In diese Form von Nebenbedingungen, die wir auch als

$$\widetilde{A}x \geq \widetilde{b}, \qquad x \geq 0, \tag{2.10}$$

schreiben können, läßt sich unser Originalproblem immer transformieren: Ist 0 eine Ecke, so gibt es (unter Umständen nach Permutation der Zeilen) eine invertierbare Matrix B, so daß

$$\begin{bmatrix} 0 \\ \widetilde{b} \end{bmatrix} = b \leq Ax = \begin{bmatrix} B \\ C \end{bmatrix} x = \begin{bmatrix} I \\ CB^{-1} \end{bmatrix} Bx := \begin{bmatrix} I \\ CB^{-1} \end{bmatrix} x'$$

mit dem "Taschenspielertrick" $x' := Bx$. Die entsprechende Zielfunktion ist dann

$$c^T x = c^T B^{-1} x' = \left(B^{-T} c\right)^T x' =: c'^T x,$$

und so lässt sich jedes Optimierungsproblem mit einer Ecke an $x = 0$ in die Form (2.9) bzw. (2.10) darstellen:

$$\max c'^T x, \qquad A'x \geq 0, \quad x \geq 0, \qquad \text{mit} \qquad c' = B^{-T} c, \quad A' = CB^{-1}. \tag{2.11}$$

Die Vorteile von (2.10) liegen auf der Hand: Wir brauchen jetzt in unserem Optimierungsproblem n Zeilen weniger zu betrachten und haben eine Normalform mit der sich sehr einfach rechnen läßt, da die Suche nach Nachbarecken jetzt entlang der Koordinatenachsen verläuft.

Die "Startecke" $x^{(0)} = 0$ ist in dieser Konfiguration dann wie gesagt durch die Indexmenge $J = \{1, \ldots, n\}$ charakterisiert. Außerdem verwenden wir die Bezeichnung

$$y := \widetilde{A}x - \widetilde{b},$$

also

$$\begin{bmatrix} x \\ y \end{bmatrix} = Ax - b, \qquad x, y \geq 0. \tag{2.12}$$

Damit haben die Hyperebenen, die $F(A, b)$ beranden, die einfache Form

$$\{\, x_j = 0 \,\}, \quad j = 1, \ldots, n, \qquad \text{und} \qquad \{\, y_j = 0 \,\}, \quad j = n+1, \ldots, m.$$

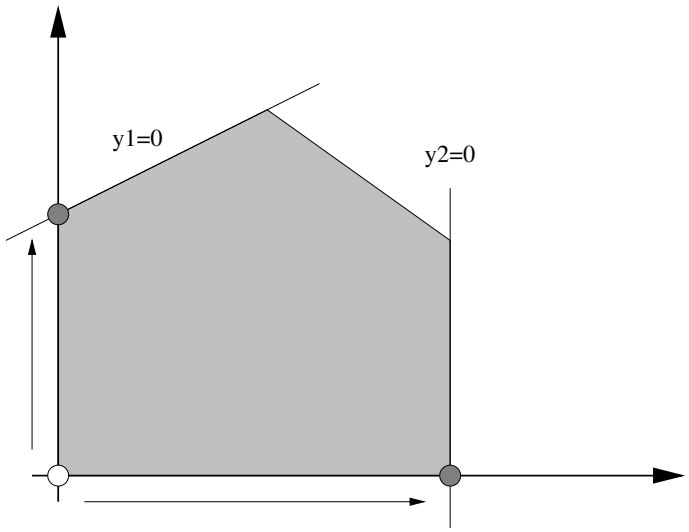


Abbildung 2.4: Benachbarte Ecken des Ursprungs.

Bevor wir uns an den formalen Teil machen, wollen wir uns erst einmal die "Strategie" ansehen: Wie in Abb. 2.4 zu sehen ist, hat die zulässige Ecke $x^{(0)} = 0$ zwei Nachbarecken, nämlich die Ecken

$$\{\, x_1 = 0 \,\} \cap \{\, y_1 = 0 \,\} \qquad \text{und} \qquad \{\, x_2 = 0 \,\} \cap \{\, y_2 = 0 \,\}\,.$$

Unter diesen beiden Ecken, die wir erhalten, indem wir entweder x_2 durch y_1 oder x_1 durch y_2 ersetzen, wählen wir jetzt natürlich diejenige, an der die Zielfunktion den größeren Wert annimmt.

Um den *Austauschschritt* durchzuführen, der die formalen Variablen x_j und y_k vertauscht, fixieren wir $j \in \{1, \ldots, n\}$ und $k \in \{1, \ldots, m-n\}$ und bemerken zuerst einmal, daß wir die Gleichung

$$y_k = a_{n+k}^T x - b_{n+k}$$

genau dann nach x_j auflösen können, wenn $a_{n+k,j} \neq 0$ ist und dann gilt

$$x_j = \frac{1}{a_{n+k,j}} \left(y_k - \sum_{\ell \neq j} a_{n+k,\ell}\, x_\ell + b_{n+k} \right). \tag{2.13}$$

[35] Also fast wie in der Realität.

In Matrixform heißt dies, daß

$$x = \underbrace{\begin{bmatrix} 1 & & & & & & & \\ & \ddots & & & & & & \\ & & 1 & & & & & \\ -\frac{a_{n+k,1}}{a_{n+k,j}} & \cdots & -\frac{a_{n+k,j-1}}{a_{n+k,j}} & \frac{1}{a_{n+k,j}} & -\frac{a_{n+k,j+1}}{a_{n+k,j}} & \cdots & -\frac{a_{n+k,n}}{a_{n+k,j}} \\ & & & & 1 & & \\ & & & & & \ddots & \\ & & & & & & 1 \end{bmatrix}}_{=:B} \underbrace{\begin{bmatrix} x_1 \\ \vdots \\ x_{j-1} \\ y_k \\ x_{j+1} \\ \vdots \\ x_n \end{bmatrix}}_{=:z} + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \frac{b_{n+k}}{a_{n+k,j}} \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

also

$$x = Bz + \frac{b_{n+k}}{a_{n+k,j}}\, e_j. \tag{2.14}$$

Setzt man das in (2.12) ein, dann ergibt sich

$$\begin{aligned} \begin{bmatrix} x \\ y \end{bmatrix} &= Ax - b = \begin{bmatrix} I_n \\ \widetilde{A} \end{bmatrix} \left(Bz + \frac{b_{n+k}}{a_{n+k,j}}\, e_j \right) - \begin{bmatrix} 0 \\ \widetilde{b} \end{bmatrix} = \underbrace{\begin{bmatrix} B \\ \widetilde{A}\, B \end{bmatrix}}_{=:C} z + \underbrace{\frac{b_{n+k}}{a_{n+k,j}}\, Ae_j}_{=:-\widehat{b}} - b \\ &= Cz - \left(b + \widehat{b} \right), \end{aligned} \tag{2.15}$$

wobei

$$\widehat{b} = -\frac{b_{n+k}}{a_{n+k,j}} \begin{bmatrix} a_{1,j} \\ \vdots \\ a_{m,j} \end{bmatrix} = -\frac{b_{n+k}}{a_{n+k,j}} \left(\begin{bmatrix} 0 \\ \vdots \\ 0 \\ a_{n+1,j} \\ \vdots \\ a_{m,j} \end{bmatrix} + e_j \right).$$

Da $y_k = (Cz)_{n+k}$, ist die $(n+k)$–te Zeile von C, also c_{n+k}^T, von besonders einfacher Form:

$$c_{n+k}^T = e_j^T.$$

Vertauschen wir also die Zeilen j und $n+k$ in (2.15), so können wir (2.15) unter Verwendung der Notation

$$x^{(1)} := z = \begin{bmatrix} x_1 \\ \vdots \\ x_{j-1} \\ y_k \\ x_{j+1} \\ \vdots \\ x_n \end{bmatrix} \qquad \text{und} \qquad y^{(1)} := \begin{bmatrix} y_1 \\ \vdots \\ y_{k-1} \\ x_j \\ y_{k+1} \\ \vdots \\ y_{m-n} \end{bmatrix}$$

umschreiben in

$$\begin{bmatrix} x^{(1)} \\ y^{(1)} \end{bmatrix} = A^{(1)}x^{(1)} - b^{(1)} = \begin{bmatrix} I_n \\ \widetilde{A}^{(1)} \end{bmatrix} x^{(1)} - b^{(1)}, \tag{2.16}$$

wobei[36] $b^{(1)} = b + \widehat{b}$. Dann sind natürlich die beiden Nebenbedingungen, das heißt die Ungleichungssysteme $Ax \geq b$ und $A^{(1)}x^{(1)} \geq b^{(1)}$ äquivalent, oder, anders gesagt, die Polyeder $F(A,b)$ und $F\left(A^{(1)}, b^{(1)}\right)$ sind gleich.

Wenn wir es jetzt noch schaffen, daß der Nullpunkt wieder eine zulässige Ecke des konvexen Polyeders $F\left(A^{(1)}, b^{(1)}\right)$ ist, das heißt, daß

$$0 = A^{(1)}0 \geq b^{(1)} \qquad \Longrightarrow \qquad b^{(1)} \leq 0,$$

dann haben wir, via Austausch, tatsächlich den Schritt von einer Ecke zur einer benachbarten Ecke geschafft, und zwar so, daß diese "neue" Ecke wieder der Nullpunkt ist. Dazu bemerken wir zuerst, daß[37]

$$b_j^{(1)} = b_{n+k} + \widehat{b}_{n+k} = b_{n+k} - \frac{b_{n+k}}{a_{n+k,j}}\, a_{n+k,j} = 0$$

und, trivialerweise, $b_\ell^{(1)} = 0$, $\ell \in \{1,\ldots,n\} \setminus \{j\}$, und $x^{(1)} = 0$ ist zumindest schon einmal ein *Kandidat* für eine Ecke des Polyeders – allerdings muß dieser Punkt auch zulässig sein, um wirklich eine Ecke darzustellen. Dies ist nun genau dann der Fall, wenn $b^{(1)} \leq 0$ ist, also wenn

$$0 \geq b_{n+k}^{(1)} = \underbrace{b_j}_{=0} - \frac{b_{n+k}}{a_{n+k,j}}\, \underbrace{a_{jj}}_{=1} = -\frac{b_{n+k}}{a_{n+k,j}} \tag{2.17}$$

und

$$0 \geq b_{n+\ell}^{(1)} = b_{n+\ell} - \frac{a_{n+\ell,j}}{a_{n+k,j}}\, b_{n+k}, \qquad \ell = 1,\ldots,m-n, \quad \ell \neq k. \tag{2.18}$$

Aus (2.17) und der Forderung $b \leq 0$ folgt, daß[38]

$$a_{n+k,j} < 0. \tag{2.19}$$

Die Bedingung (2.18) läßt sich hingegen zuerst einmal in

$$\frac{a_{n+\ell,j}}{a_{n+k,j}}\, b_{n+k} \geq b_{n+\ell}, \qquad \ell = 1,\ldots,m-n,\ \ell \neq k,$$

umformen, was immer erfüllt ist, wenn $a_{n+\ell,j} \geq 0$ ist. Im anderen Fall erhalten wir, daß

$$\frac{b_{n+k}}{a_{n+k,j}} \leq \frac{b_{n+\ell}}{a_{n+\ell,j}}, \qquad \text{falls} \quad a_{n+\ell,j} < 0, \tag{2.20}$$

sein muß. Also haben wir die folgende Regel zur Bestimmung von k (für ein gegebenes $j \in \{1,\ldots,n\}$):

[36] Nach Vertauschung der Komponenenten j und $n+k$ von b!

[37] Nicht vergessen: Die Zeilen j und $n+k$ wurden vertauscht!

[38] Genaugenommen könnte $a_{n+k,j}$ machen, was es will, wenn $b_{n+k} = 0$ ist; da aber immer durch eine beliebig kleine "zulässige" Störung $b_{n+k} < 0$ erreicht werden kann, können wir dies hier auch annehmen.

Die "Pivotzeile" $n+k$ ist so zu bestimmen, daß $a_{n+k,j} < 0$ und daß der (positive) Quotient

$$\frac{b_{n+\ell}}{a_{n+\ell,j}}, \qquad a_{n+\ell,j} < 0, \qquad \ell = 1, \ldots, m-n,$$

minimiert wird, also

$$\frac{b_{n+k}}{a_{n+k,j}} = \min\left\{ \frac{b_{n+\ell}}{a_{n+\ell,j}} \ : \ a_{n+\ell,j} < 0,\ \ell = 1, \ldots, m-n \right\}. \tag{2.21}$$

Bleibt also noch die Frage, wie man diese ominöse Spalte j (also die auszutauschende Variable) wählt. Hier kommt jetzt die *Vergrößerung* der Zielfunktion ins Spiel. Zu diesem Zweck setzen wir (2.13) in die Zielfunktion

$$c^T x = \sum_{\ell \neq j} c_\ell \, x_\ell + \frac{c_j}{a_{n+k,j}} \left(y_k - \sum_{\ell \neq j} a_{n+k,\ell} \, x_\ell + b_{n+k} \right),$$

ein, das heißt,

$$c^T x = \frac{c_j}{a_{n+k,j}} \, y_k + \sum_{\ell \neq j} \left(c_\ell - \frac{a_{n+k,\ell}}{a_{n+k,j}} \, c_j \right) x_\ell + \frac{c_j}{a_{n+k,j}} \, b_{n+k} =: {c^{(1)}}^T \, x^{(1)} + d^{(1)},$$

setzen $x^{(1)} = 0$ und erhalten eine Verbesserung (oder zumindest keine Verschlechterung) gegenüber dem Ausgangswert, falls

$$0 \leq d^{(1)} = \frac{c_j}{a_{n+k,j}} \, b_{n+k}, \tag{2.22}$$

also wenn $c_j \geq 0$ ist, was zu der folgenden Regel führt:

Die Spalte j ist so zu bestimmen, daß $c_j > 0$ ist.

Diese beiden Auswahlregeln für j und k können erfüllt werden, solange

1. die Matrix $\widetilde{A}$ *negative* Werte enthält,
2. eine Spalte von $\widetilde{A}$ existiert, die einen negativen Wert enthält und einem nichtnegativen Eintrag von c entspricht.

Ist eine dieser beiden Bedingungen verletzt, so hat dies jeweils eine Bedeutung:

1. Ist $\widetilde{A} \geq 0$, dann ist $Ax \geq 0$ wann immer $x \geq 0$ und gibt es auch nur ein zulässiges $x^* \geq 0$, dann ist für alle $x \geq 0$ auch

$$A\,(x^* + x) = \underbrace{Ax^*}_{\geq b} + \underbrace{Ax}_{\geq 0} \geq b,$$

und somit ist $F(A,b) \subseteq x^* + \mathbb{R}^n_+$. Mit anderen Worten: *Das Polyeder $F(A,b)$ ist unbeschränkt und ist auch nur ein $c_j > 0$, dann ist auch die Zielfunktion unbeschränkt. Ansonsten ist ihr Maximum ohnehin trivialerweise* 0.

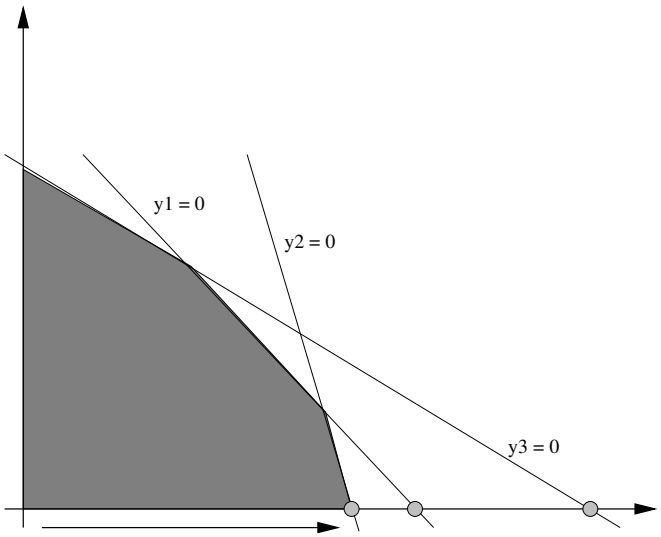


Abbildung 2.5: Geometrische Interpretation der Bedingungen für die Wahl der Austauschparameter. Unter allen "guten" Schnittpunkten wird derjenige gewählt, der am nächsten bei der Ecke $(0, 0)$ liegt.

2. Ist $c_j \leq 0$ für alle Spalten von A, die negative Werte enthalten, dann wird beim Übergang zu allen benachbarten Ecken die Zielfunktion nur verkleinert, also: *Wir haben ein Maximum gefunden.* Damit kann nach Proposition 2.11 und Bemerkung 2.12 der Algorithmus beendet werden.

Noch kurz zur *geometrischen* Interpretation der ersten Bedingung. Die Bedingung $a_{n+k,j} < 0$ bedeutet, daß man nur nach Hyperebenen $y_k = 0$ sucht, die "vernünftig" im Oktanten $x \geq 0$ liegen. Die Minimumsbedingung hingegen ist dafür zuständig, daß unter allen Schnitten der Geraden $x_1 = \cdots = x_{j-1} = x_{j+1} = \cdots = x_n = 0$ mit derartigen Hyperebenen diejenige gewählt wird, die "als erste" erreicht wird.

Bleibt uns noch die *Bestimmung* der Optimallösung, was aber jetzt einfach ist: Das Problem wurde ja nach r Schritten so modifiziert, daß $x^{(r)} = 0$ die *extremale Ecke* des Problems

$$\max x^T c^{(r)}, \qquad A^{(r)}x \geq b^{(r)}$$

ist. Das heißt aber nach (2.9), daß

$$\begin{bmatrix} x^{(r)} \\ y^{(r)} \end{bmatrix} = A^{(r)}0 - b^{(r)} = -b^{(r)} = \begin{bmatrix} 0 \\ \widetilde{b}^{(r)} \end{bmatrix},$$

also

$$x^{(r)} = 0, \qquad y^{(r)} = -\widetilde{b}^{(r)}.$$

Und jetzt müssen wir nur noch nachschauen, in welche Komponenten von $x^{(r)}$ und $y^{(r)}$ die Variablen $x_1, \ldots, x_n$ getauscht wurden.

2.4 Die Implementierung

Als nächstes wollen wir den eben hergeleiteten Simplexalgorithmus in `Matlab`/`Octave` implementieren und versuchen, damit unser Beispiel vom Anfang zu lösen. Dazu "vergessen" wir die Einheitsmatrix "oben" in A, schreiben m für $m-n$ und setzen

$$\begin{aligned} c^{(0)} &= (c,0) = \left(c, d^{(0)}\right) \quad \in \mathbb{R}^{n+1}, \\ A^{(0)} &= \widetilde{A} \quad \in \mathbb{R}^{m\times n}, \\ b^{(0)} &= \widetilde{b} \quad \in \mathbb{R}^{m}. \end{aligned}$$

Ein Austauschschritt für die Matrix $A^{(r)}$ hat nun, für gegebene $j \in \{1\ldots,n\}$ und $k \in \{1,\ldots,m\}$ die folgende Gestalt:

1. Setze

$$c_\ell^{(r+1)} = \begin{cases} \dfrac{c_j^{(r)}}{a_{kj}^{(r)}}, & \ell = j, \\ c_\ell^{(r)} - \dfrac{a_{k\ell}^{(r)}}{a_{kj}^{(r)}}\, c_j^{(r)}, & \ell \in \{1,\ldots,n\}\setminus\{j\}, \\ c_{n+1}^{(r)} + \dfrac{c_j}{a_{kj}}\, b_k^{(r)}, & \ell = n+1. \end{cases} \tag{2.23}$$

2. Setze

$$b_\ell^{(r+1)} = \begin{cases} -\dfrac{b_k^{(r)}}{a_{kj}^{(r)}}, & \ell = k, \\ b_\ell^{(r)} - \dfrac{a_{\ell j}^{(r)}}{a_{kj}^{(r)}}\, b_k^{(r)}, & \ell \neq k, \end{cases} \qquad \ell = 1,\ldots,m. \tag{2.24}$$

3. Setze

$$B = \begin{bmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ -\frac{a_{k1}}{a_{kj}} & \cdots & -\frac{a_{k,j-1}}{a_{kj}} & \frac{1}{a_{kj}} & -\frac{a_{k,j+1}}{a_{kj}} & \cdots & -\frac{a_{kn}}{a_{kj}} \\ & & & & 1 & & \\ & & & & & \ddots & \\ & & & & & & 1 \end{bmatrix} \in \mathbb{R}^{n\times n} \tag{2.25}$$

4. Berechne die Matrix $A^{(r)}B$, ersetze deren k–te Zeile durch die k–te Zeile von B und nenne diese Matrix $A^{(r+1)}$.

```
%%
%% Austausch.m
%% Austauschschritt fuer Simplexverfahren
%% Daten:
%%   j  Spaltenindex
%%   k  Zeilenindex
%%   A  Matrix des Problems
%%   b  Nebenbedingungen
%%   c  Zielfunktion

function [AA,bb,cc] = Austausch( j,k,A,b,c )
  [m,n] = size(A);
  a = A(k,j); cj = c( j ); bk = b( k );
  cc = zeros( n+1,1 );

  %% Update von c
  cc(1:n) = c(1:n) - (cj /  a) * A( k,: )';
  cc(j) = cj / a;
  cc(n+1) = c(n+1) + cj / a * b(k);

  %% Update von b
  bb = b - bk / a * A( :,j );
  bb(k) = -bk / a;

  %% Update von A
  B = eye( n );
  B( j,: ) = (-1 / a) * A( k, : );
  B(j,j) = 1 / a;
  AA = A * B;
  AA( k,: ) = B( j,: );

%endfunction
```

Programm 2.1 `Austausch.m`: Ein Austauschschritt.

Diese Operationen sind in `Austausch.m` implementiert. Die Reihenfolge oben ist absichtlich so gewählt, da die Berechnung von $c^{(r+1)}$ sowohl $b^{(r)}$, $c^{(r)}$ wie auch $A^{(r)}$ benötigt, die Bestimmung von $b^{(r+1)}$ lediglich $b^{(r)}$ und $A^{(r)}$ und $A^{(r+1)}$ schließlich aus $A^{(r)}$ berechnet werden kann, das heißt, in dieser Reihenfolge können die Variablen überschrieben werden.

Bemerkung 2.13 *Man kann sich die schematischen Regeln des Simplexalgorithmus recht einfach merken*[39]*:*

1. *Dividiere die Pivotzeile durch das Pivotelement.*
2. *Dividiere die Pivotspalt durch das Negative des Pivotelements.*
3. *Für alle anderen Elemente verwende die "Rechtecksregel":*

	$\dots$	u_k	$\dots$	u_q	$\dots$	
$\vdots$	$\ddots$	$\vdots$		$\vdots$		$\vdots$
v_j	$\dots$	a_{jk}	$\dots$	a_{jq}	$\dots$	b_j
$\vdots$		$\vdots$	$\ddots$	$\vdots$		$\vdots$
v_p	$\dots$	a_{pk}	$\dots$	$\boxed{a_{pq}}$	$\dots$	b_p
$\vdots$		$\vdots$		$\vdots$	$\ddots$	$\vdots$
	$\dots$	c_k	$\dots$	c_q	$\dots$	

$$a_{pq} \leftarrow a_{pq} - \frac{a_{pk}\, a_{jq}}{a_{jk}}$$

4. *Ersetze das Pivotelement durch seinen Reziprokwert.*

Will man den Algorithmus mit Überschreiben realisieren[40]*, dann muss man natürlich mit Schritt 3 beginnen.*

Zur Bestimmung der Indizes j, k, für die die Vertauschung durchgeführt werden soll, wird die Matrix $A^{(r)}$ spaltenweise durchgegangen. Sobald eine Spalte gefunden wurde, die ein negatives Element enthält[41], wird der zugehörige Eintrag in $c^{(r)}$ geprüft. Ist dieser ≥ 0, so verschlechtert der Übergang zur Nachbarecke das Ergebnis nicht, ist er < 0, so wird die Spalte verworfen. Findet man keine passende Spalte, dann ist die Lösung optimal.

Bleibt noch der Simplexalgorithmus selbst. Um auch die Parameter $x_1, \dots, x_n$ der Optimallösung angeben zu können, müssen wir Buch führen, welche Gleichungen miteinander ausgetauscht wurden. Das wird durch zwei Vektoren `xVec` und `yVec` erledigt, die angeben, welche der Parameter als Variablen (`xVec`) und welche als affine Funktionen (`yVec`) fungieren; ein positiver Eintrag j bedeutet hierbei die Variable x_j, ein negativer $-k$ die Variable y_k. Da die Optimallösung ja durch $x^{(r)} = 0$ gegeben ist, erhalten diejenigen x_j, die zu affinen Funktionen geworden sind, den Wert der entsprechenden Komponente von $b^{(r)}$, die Variablen geblieben sind, werden hingegen $= 0$ gesetzt.

[39] Wenn man den Simplexalgorithmus denn unbedingt manuell durchführen möchte, was in Anbetracht von Programmen wie `Octave` und der Verfügbarkeit exzellenter Implementierungen wie `lpsolve` eigentlich Zeitvergeudung ist.

[40] Beispielsweise an einer Tafel mit Auswischen.

[41] Hierbei gehen wir immer davon aus, daß $A^{(0)}$ "vernünftig" gewählt war, also mindestens einen negativen Wert enthalten hat – damit ist der zulässige Bereich nicht total unbeschränkt.

```
%% auxFindjk.m (Optimierung)
%% Auffinden passender Werte j,k
%% Daten:
%%   A  Matrix des Problems
%%   b  Rechte Seite
%%   c  Zielfunktion

function [j,k] = auxFindjk( A,b,c )
  if min( min( A ) ) > 0                 %% kleinster Eintrag
    disp( "*** Unbeschraenkt ***" );
    j = 0; k = 0;
    return;
  end

  cA = find( min(A) < 0 );               %% Spalten mit neg. Eintrag
  cc = find( c( cA ) > 0 );              %% dort c > 0
  if length( cc != 0 )
    j = cA( cc( 1 ) );                   %% Erste Spalte - warum nicht?
    cA = find( A( :,j ) < 0 );           %% Negative Eintraege in Spalte
    [m,k] = min( b( cA ) ./ A( cA,j ) ); %% Lokalisiere Minimum
    k = cA( k );                         %% k ist entsprechende Spalte
  else
    j = 0; k = 0;
  end
```

Programm 2.2 `auxFindjk.m`: Suche nach den Indizes j, k.

```
%% Simplex.m (Optimierung)
%% Simplexverfahren
%% Daten:
%%   A  Matrix des Problems
%%   b  Nebenbedingungen
%%   c  Zielfunktion

function [x,opt] = Simplex( A,b,c )
  StepNum = 0; c = [ c; 0 ];
  [m,n] = size( A );
  xVec = ( 1:n ); yVec = ( -1:-1:-m );   %% +/- fuer x/y

  disp( [[A,b];c'] );                    %% Zeige Simplextableau

  do
    StepNum = StepNum + 1;
    [j,k] = auxFindjk( A,b,c );          %% Finde Indizes

    if j != 0
      disp( [StepNum,j,k] );             %% Zeige SchrittNr und Austausch
      t = xVec( j ); xVec( j ) = yVec( k ); yVec( k ) = t;
      [A,b,c] = Austausch( j,k,A,b,c );
      disp( [[A,b];c'] );                %% Zeige Simplextableau
    end
  until ( j == 0 )

  x = auxGenx( n,yVec,b );
  opt = c(n+1);
%endfunction
```

Programm 2.3 `Simplex.m`: Der Simplexalgorithmus für ein Problem, das den Nullpunkt als zulässige Ecke hat.

```
%%
%% auxGenx.m
%% Simplexverfahren
%% Daten:
%%   yVec  Vector der Gleichungsnummern; Eintraege > 0 entsprechen Variablen
%%   b     Rechte Seite

function x = auxGenx( n,yVec,b )
  m = length( yVec );
  x = zeros( n,1 );

  for k = 1:m
    if yVec( k ) > 0
      x( yVec(k) ) = -b( k );
    end
  end

%endfunction
```

Programm 2.4 `auxGenx.m`: Bestimmung des Lösungsvektors x für das Optimierungsproblem aus den Austauschinformationen.

Beispiel 2.14 *Sehen wir uns nochmals Beispiel 2.1 an. Die für unsere Simplexmethode relevanten Parameter können wir nun wie folgt in einer Tabelle, dem* "Simplextableau"*, darstellen:*

	x_1	$\dots$	x_n	
y_1	a_{11}	$\dots$	a_{1n}	b_1
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
y_n	a_{n1}	$\dots$	a_{nn}	b_n
	c_1	$\dots$	c_n	c_{n+1}

,

in unserem Fall also

	x_1	x_2	
y_1	-20	-10	-8000
y_2	-4	-5	-2000
y_3	-6	-15	-4500
	16	32	0

.

Die einzelnen Schritte des Simplexalgorithmus liefern dann

$(1,1)$ $\longrightarrow$

	y_1	x_2	
x_1	-0.05	-0.5	-400
y_2	0.2	-3	-400
y_3	0.3	-12	-2100
	-0.8	24	6400

$(2,2)$ $\longrightarrow$

	y_1	y_2	
x_1	-0.0833	0.1667	-333.33
x_2	0.0667	-0.333	-133.33
y_3	-0.5	4	-500
	0.8	-8	9600

$(1,3)$ $\longrightarrow$

	y_3	y_2	
x_1	.1667	-0.5	-250
x_2	-0.133	0.2	-200
y_1	-2	8	-1000
	-1.6	-1.6	10400

$$\Longrightarrow \qquad x = \begin{bmatrix} 250 \\ 200 \end{bmatrix}, \ \max = 10400.$$

Ein ziemlich wichtiger Implementierungsparameter ist die Bestimmung der *Pivotspalte* j, von der wir bisher nur gefordert haben, daß $c_j^{(r)} > 0$ sein soll. In der Tat gibt es die verschiedensten Strategien, diese Spalte auszuwählen:

1. Man wählt das kleinste j, so daß $c_j > 0$ ist. Diese Methode ist in Algorithmus 2.2 beschrieben.

2. Man sucht sich eine Spalte j aus, wo c_j *maximal* wird, also

$$c_j \geq c_\ell, \qquad \ell = 1, \dots, n.$$

   Dies ist wohl auch das "Originalverfahren" bei Dantzig und in Algorithmus 2.5 implementiert.

3. Zu jeder Spalte j mit $c_j > 0$ bestimmt man die zugehörige Zeile k, bildet den Wert

$$d_j = \frac{c_j}{a_{kj}} b_k \geq 0,$$

```
%% auxFindjk3.m (Optimierung)
%% Auffinden passender Werte j,k
%%     mit Pivotsuche
%% Daten:
%%   A  Matrix des Problems
%%   b  Rechte Seite
%%   c  Zielfunktion

function [j,k] = auxFindjk3( A,b,c )
  if min( min( A ) ) > 0                 %% kleinster Eintrag
    disp( "*** Unbeschraenkt ***" );
    j = 0; k = 0;
    return;
  end

  cA = find( min(A) < 0 );               %% Spalten mit neg. Eintrag
  [cmx,j] = max( c( cA ) );              %% Groesster Wert und wo
  if ( cmx > 0 )
    j = cA( j );                         %% Groesster c-Wert!
    cA = find( A( :,j ) < 0 );           %% Negative Eintraege in Spalte
    [m,k] = min( b( cA ) ./ A( cA,j ) ); %% Lokalisiere Minimum
    k = cA( k );                         %% k ist entsprechende Spalte
  else
    j = 0; k = 0;
  end
```

Programm 2.5 `auxFindjk3.m`: "Pivotsuche" für die Zielspalte; man wählt die Spalte so, daß c_j maximiert wird.

um den die Zielfunktion verbessert wird und wählt j dann so, daß

$$d_j = \max\{d_\ell \,:\, \ell = 1, \ldots, n,\, c_\ell > 0\}\,.$$

Diese Strategie, die einer *Totalpivotsuche* entspricht, ist in Algorithmus 2.6 realisiert. Es zeigt sich, daß diese Pivotstrategie wirklich Vorteile haben kann, insbesondere in Extremfällen wie Beispiel 2.22.

```
%% auxFindjk.m (Optimierung)
%% Auffinden passender Werte j,k
%%     mit Totalpivotsuche, erste Spalte von c
%% Daten:
%%   A  Matrix des Problems
%%   b  Rechte Seite
%%   c  Zielfunktion

function [j,k] = auxFindjk4( A,b,c )
  if min( min( A ) ) > 0                 %% kleinster Eintrag
    disp( "*** Unbeschraenkt ***" );
    j = 0; k = 0;
    return;
  end

  cA = find( min(A) < 0 );               %% Spalten mit neg. Eintrag
  cc = find( c( 1,cA ) > 0 );            %% dort c > 0
  d = zeros( 2,length( cc )+1 );         %% Wenigstens eine Null :-)
  for l = 1:length( cc )
    j = cA( cc( l ) );                   %% Erste Spalte - warum nicht?
    cA = find( A( :,j ) < 0 );           %% Negative Eintraege in Spalte
    [m,k] = min( b( cA ) ./ A( cA,j ) ); %% Lokalisiere Minimum
    k = cA( k );                         %% k ist entsprechende Zeile
    d(l) = [ c(1,l)*b(k)/A(k,l), k ];    %% Ergebnisse
  end
  [ mx,l ] = max( d(1,:) );
  if ( mx > 0 )
    j = cA( cc(l) ); k = d(2,l);
  else
    j = 0; k = 0;
  end
```

Programm 2.6 `auxFindjk4.m`: Totalpivotsuche zur Bestimmung der Zielspalte und -zeile; Der Rechenaufwand zur Bestimmung von `j` und `k` wird bei diesem Verfahren natürlich maximal, dafür hofft man aber natürlich, daß man so die Anzahl der Austauschschritte drastisch vermindern kann.

2.5 Degenerierung und andere Ärgernisse

Unangenehm wird es aber, wenn die *Pivotzeile* nicht eindeutig ist, das heißt, wenn es mindestens *zwei* Indizes $k \neq k' \in \{1, \ldots, m-n\}$ gibt, so daß

$$\frac{b_k^{(r)}}{a_{k,j}^{(r)}} = \frac{b_{k'}^{(r)}}{a_{k',j}^{(r)}} = \min\left\{ \frac{b_\ell^{(r)}}{a_{\ell,j}^{(r)}} \;:\; a_{\ell,j}^{(r)} < 0,\, \ell = 1, \ldots, m-n \right\}. \tag{2.26}$$

Führt man nun einen Austauschschritt mit dem Paar (j,k) durch, dann folgt aus (2.24), daß

$$b_{k'}^{(r+1)} = b_{k'}^{(r)} - a_{k'j}^{(r)} \frac{b_k^{(r)}}{a_{kj}^{(r)}} = b_{k'}^{(r)} - a_{k'j}^{(r)} \frac{b_{k'}^{(r)}}{a_{k'j}^{(r)}} = 0.$$

Das heißt aber, daß die zur Ecke $x^{(r+1)} = 0$ gehörige Indexmenge $J^{(r+1)}$ mindestens $n+1$ Einträge hat – die ersten, "virtuellen" n Zeilen, die zur Einheitsmatrix gehören und die Zeile k'. Die Ecke $x^{(r+1)} = 0$ ist also der Durchschnitt von mindestens $n+1$ Hyperebenen[42]. Diese Situation bezeichnet man als *Degenerierung*. Degenerierungen können dazu führen, daß der Simplexalgorithmus stationär wird, das heißt, auf einer nicht optimalen Seite "im Kreis läuft".

Beispiel 2.15 *Wir betrachten das folgende Beispiel aus [44, S. 69]:*

$$\widetilde{A} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ -1 & 0 & -1 \\ 0 & -1 & -1 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \end{bmatrix}, \qquad \widetilde{b} = \begin{bmatrix} -2 \\ -2 \\ -3 \\ -3 \\ -1 \\ -1 \end{bmatrix}, \qquad c = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}.$$

Die geometrische Interpretation des zugehörigen Polyeders ist ein Quader mit aufgesetzter Pyramide, siehe Abb. 2.6. Und in der Tat sorgt die obere Ecke für Schwierigkeiten beim Simplexalgorithmus: Wir erhalten die Folge

$(3,5)$ $\longrightarrow$

	x_1	x_2	y_5	
y_1	-1	0	0	-2
y_2	0	-1	0	-2
y_3	-2	0	1	-2
y_4	-1	-1	1	-2
x_3	1	0	-1	-1
y_6	-1	1	1	$\mathbf{0}$
	4	2	-3	3

$(1,6)$ $\longrightarrow$

	y_6	x_2	y_5	
y_1	1	-1	-1	-2
y_2	0	-1	0	-2
y_3	2	-2	-1	-2
y_4	1	-2	0	-2
x_3	-1	1	0	-1
x_1	-1	1	1	$\mathbf{0}$
	-4	6	1	3

[42] Der "normale", also generische Fall im $\mathbb{R}^n$ ist, daß sich gerade n Hyperebenen, nicht mehr und nicht weniger, in einem Punkt schneiden.

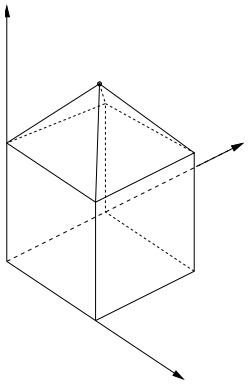


Abbildung 2.6: Das Polyeder $F(A,b)$ zu Beispiel 2.15. Die obere Ecke ist degeneriert, da schneiden sich 4 Seiten des Polyeders.

$$
\begin{array}{c|c|ccc|c}
 & & y_6 & y_3 & y_5 & \\ \hline
 & y_1 & 0 & \frac{1}{2} & -\frac{1}{2} & -1 \\
 & y_2 & -1 & \frac{1}{2} & \frac{1}{2} & -1 \\
(2,3) & x_2 & 1 & -\frac{1}{2} & -\frac{1}{2} & -1 \\
\longrightarrow & y_4 & -1 & 1 & 1 & \mathbf{0} \\
 & x_3 & 0 & -\frac{1}{2} & -\frac{1}{2} & -2 \\
 & x_1 & 0 & -\frac{1}{2} & \frac{1}{2} & -1 \\ \hline
 & & 2 & -3 & -2 & 9
\end{array}
\qquad
\begin{array}{c|c|ccc|c}
 & & y_4 & y_3 & y_5 & \\ \hline
 & y_1 & 0 & \frac{1}{2} & -\frac{1}{2} & -1 \\
 & y_2 & 1 & -\frac{1}{2} & -\frac{1}{2} & -1 \\
(1,4) & x_2 & -1 & \frac{1}{2} & \frac{1}{2} & -1 \\
\longrightarrow & y_6 & -1 & 1 & 1 & \mathbf{0} \\
 & x_3 & 0 & -\frac{1}{2} & -\frac{1}{2} & -2 \\
 & x_1 & 0 & -\frac{1}{2} & \frac{1}{2} & -1 \\ \hline
 & & -2 & -1 & \mathbf{0} & 9
\end{array}
$$

und müssen uns nun überlegen, was die "fetten" Nullen bedeuten:

1. *Die Nullen in der* Spalte *auf der rechten Seite zeigen uns an, daß wir uns in einer entarteten, degenerierten, Ecke befinden, was dazu führen kann, daß der Simplexalgrorithmus diese Ecke nicht verläßt. In der Tat nimmt er den Weg*

$$
\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \overset{(3,5)}{\longrightarrow} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \overset{(1,6)}{\longrightarrow} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \overset{(2,3)}{\longrightarrow} \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix} \overset{(1,4)}{\longrightarrow} \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix},
$$

   bleibt also mehrmals *in derselben Ecke!*

2. *Die Null in der Zeile unter dem Simplextableau zeigt hingegen an, daß die Optimallösung* nicht *eindeutig ist – es gibt eine Nachbarecke, an der die Zielfunktion denselben Wert annimmt.*

Ein solches Verhalten ist der einfachste Falle eines *Zyklus*, der sich natürlich auch über mehrere Punkte erstrecken kann und der dazu führen kann, daß der Simplexalgorithmus *nie* terminiert;

solche Beispiele gibt es, wenngleich sie ziemlich konstruiert sind, aber laut [36] tauchen sie doch recht häufig in der Realität auf, und zwar bei Problemen, die aus der Ganzzahlprogrammierung stammen.

Die Entfernung dieser Zyklen durch geeignete Störungen des Optimierungsproblems ist ein wichtiges Detail bei der praktischen Realisierung des Simplexalgorithmus, siehe [36, S. 389–391].

Beispiel 2.16 *Verwendet man die Pivotmethode aus Algorithmus 2.6, dann taucht keiner der Zyklen aus Beispiel 2.15 auf, denn hier hat man die Situation, daß man von jeder Ecke aus die Zielfunktion verbessern kann. Die Folge der Simplextableaus ist dann*

	x_1	y_2	x_3	
y_1	-1	0	0	-2
x_2	0	-1	0	-2
y_3	-1	0	-1	-3
y_4	0	1	-1	-1
y_5	1	0	-1	-1
y_6	0	-1	-1	-3
	1	-2	3	4

$(2,2)$ $\longrightarrow$

	x_1	y_2	y_4	
y_1	-1	0	0	-2
x_2	0	-1	0	-2
y_3	-1	-1	1	-2
x_3	0	1	-1	-1
y_5	1	-1	1	$\mathbf{0}$
y_6	0	-2	1	-2
	1	1	-3	7

$(3,4)$ $\longrightarrow$

	y_1	y_2	y_4	
x_1	-1	0	0	-2
x_2	0	-1	0	-2
y_3	1	-1	1	$\mathbf{0}$
x_3	0	1	-1	-1
y_5	-1	-1	1	-2
y_6	0	-2	1	-2
	-1	1	-3	9

$(1,1)$ $\longrightarrow$

Den Übergang von der Ecke $[2,2,1]^T$ *nach* $[1,1,2]$ *vermeidet diese Version des Simplexalgorithmus, weil* keine *Verbesserung mehr möglich ist, und so terminiert der Simplexalgorithmus, obwohl es eine Spalte gibt, in der* $c_j > 0$ *ist!*

Übung 2.3 Zeigen Sie: Ist eine $n-1$–dimensionale Seitenfläche X von F eine *Niveaufläche* von $c^T\cdot$, d.h. $c^Tx = c^Tx'$, $x, x' \in X$, dann nimmt die Zielfunktion auf X ihr Maximum oder ihr Minimum an. ♢

Ein anderes "Ärgernis" sind *freie Variablen*. Bisher haben wir immer angenommen, daß die automatischen Nebenbedingungen $x \geq 0$ gelten. Solange alle Variablen einseitig beschränkt sind, das heißt, Forderungen der Form

$$x_j \geq \xi_j \qquad \text{oder} \qquad x_j \leq \xi_j \qquad j = 1, \ldots, n,$$

vorliegen, können wir diese immer durch $x \geq 0$ ausdrücken, indem wir x_j durch $\pm(x_j - \xi_j)$ mit passendem Vorzeichen[43] ersetzen. Das resultierende, *äquivalente* Optimierungsproblem hat dann die gewünschte Form (2.1).

[43] Um, wenn nötig, das Ungleichungszeichen umzudrehen.

Hingegen heißt x_j *freie Variable*, wenn es keine solche *direkte* Beschränkung an x_j gibt, sondern sich der zulässige x_j–Bereich nur *indirekt* aus den Nebenbedingungen $Ax \geq b$ ergibt. Solche Variablen stören natürlich, wenn man ein Verfahren verwenden will, bei dem der Nullpunkt eine zulässige Ecke sein soll[44], weswegen man zuerst einmal eine freie Variable austauscht. Die Zeile, die *nach* dem Austausch mit y_k zu der freien Variablen x_j gehört,

$$x_j = \left(A^{(1)}x^{(1)}\right)_{n+k} - b^{(1)}_{n+k}$$

kann man zwar mitführen (z.B. um am Schluß den Wert von x_j zu ermitteln), aber sie muß *redundant*[45] sein, denn ansonsten läge ja plötzlich doch eine Beschränkung an x_j vor – für die *Rechnung* kann man sie aber getrost vergessen.

Trotzdem, wenn wir schon austauschen, dann tun wir das natürlich am besten so, daß nach dem Austauschschritt $b^{(1)} \leq 0$ ist[46], denn dann ist nämlich 0 eine zulässige Ecke und unser Mechanismus aus dem vorigen Kapitel greift wieder. Außerdem soll der Austausch natürlich die Zielfunktion vergrößern. Dazu müssen wir eine Fallunterscheidung bezüglich c_j machen[47]:

$c_j \geq 0$: Hier liegt gerade die Situation vor, wie im "normalen" Austauschschritt des Simplexalgorithmus, wir können also k so wählen, daß

$$\frac{b^{(r)}_k}{a^{(r)}_{k,j}} = \min\left\{ \frac{b^{(r)}_k}{a^{(r)}_{k,j}} \ : \ a^{(r)}_{\ell,j} < 0 \right\}.$$

$c_j < 0$: Jetzt wird's etwas interessanter. Damit (2.22) erfüllt ist, muß jetzt $a^{(r)}_{k,j} \geq 0$ sein. Damit ist (2.18) nun immer erfüllt, wenn $a_{n+\ell,j} = a^{(r)}_{\ell,j} < 0$ ist und "kritisch" wird nur der Fall, daß $a_{n+\ell,j} > 0$ ist. Und jetzt muß man halt maximieren[48]: Man wählt k so, daß

$$\frac{b^{(r)}_k}{a^{(r)}_{k,j}} = \max\left\{ \frac{b^{(r)}_k}{a^{(r)}_{k,j}} \ : \ a^{(r)}_{\ell,j} > 0 \right\}.$$

Nach der Elimination aller freien Variablen (und der entsprechenden Nebenbedingungen) hat also unser Optimierungsproblem dann die Normalform (2.9).

2.6 Auffinden einer Startecke

Bisher haben wir uns in einer Hinsicht das Leben sehr leicht gemacht: Wir haben angenommen, daß der Nullpunkt eine zulässige Ecke war. Nun, eine *Ecke* ist der Nullpunkt immer, wenn das

[44] Denn dazu bräuchten wir ja sowas wie $x_j = 0$.

[45] Das heißt, es kann keine zulässige Ecke x geben, zu deren Bestimmungsmenge $J(x)$ die Gleichung $x_j = 0$ gehören *muß*.

[46] Natürlich unter Vernachlässigung der k–ten Komponente und unter der Voraussetzung, daß $b \leq 0$ war.

[47] Nicht vergessen: Die Spalte können wir nicht wählen, denn sie ist ja dadurch vorgegeben, daß x_j eine *freie* Variable ist.

[48] Was wieder nichts anderes heißt, als den *Betrag* dieses Quotienten zu minimieren.

Optimierungsproblem in der Form (2.9) vorliegt[49], wenn also

$$A = \begin{bmatrix} I_n \\ \widetilde{A} \end{bmatrix} \qquad \text{und} \qquad b = \begin{bmatrix} 0 \\ \widetilde{b} \end{bmatrix} \tag{2.27}$$

ist, nur mit der *Zulässigkeit* kann es unter Umständen hapern.

Beispiel 2.17 *(Transportproblem)*[50]
In den Rangierbahnhöfen A und B stehen 18 bzw. 12 leere Waggons, in den Bahnhöfen X, Y und Z werden 11, 10 und 9 Waggons benötigt. Die Distanzen zwischen den Bahnhöfen betragen

	X	*Y*	*Z*
A	*5*	*4*	*9*
B	*7*	*8*	*10*

Welche Verteilung der Waggons minimiert die gefahrene Kilometerzahl[51]*?*
Um dieses Problem mathematisch darzustellen, sei x die Anzahl der Wagen, die von A nach X fahren und y die Anzahl der Wagen, die von A nach y fahren. Dann lassen sich alle Wagenbewegungen durch x und y ausdrücken und zwar

Strecke	*# Wagen*
$A \to X$	x
$A \to Y$	y
$A \to Z$	$18 - x - y$
$B \to X$	$11 - x$
$B \to Y$	$10 - y$
$B \to Z$	$x + y - 9$

und alle diese Größen müssen selbstverständlich positiv sein. Damit müssen wir den Wert

$$5x + 4y + 9\,(18 - x - y) + 7\,(11 - x) + 8\,(10 - y) + 10\,(x + y - 9) = -x - 3y + 229$$

unter den obigen Nebenbedingungen minimieren, unsere Normalform für das Optimierungsproblem lautet also

$$\max\, x + 3y - 229, \qquad \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & -1 \\ -1 & 0 \\ 0 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \\ -18 \\ -11 \\ -10 \\ 9 \end{bmatrix}$$

bestimmen; die letzte Zeile der Nebenbedingungen sorgt nun dafür, daß $[x, y] = 0$ zwar eine Ecke, aber keine zulässige *Ecke ist – wenn man keine Wagen bewegt, dann kommt halt auch nichts in X, Y oder Z an. Die Nebenbedingungen sind in Abb. 2.7 grafisch dargestellt.*

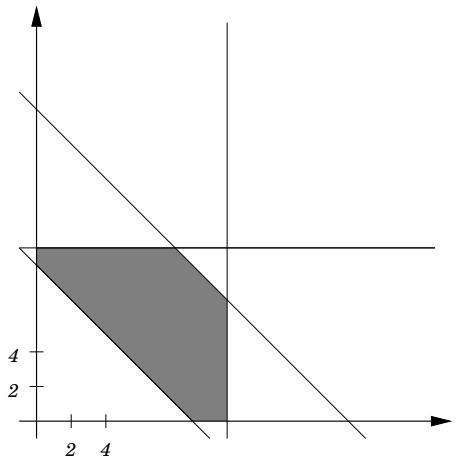


Abbildung 2.7: Der zulässige Bereich für das Transportproblem aus Beispiel 2.17; der Nullpunkt ist offensichtlich "abgeschnitten" worden.

Wenn nun

$$b^* := \max_{j=1,\dots,n} b_j \tag{2.28}$$

positiv ist, dann[52] ist der der Nullpunkt *keine* zulässige Ecke des zulässigen Bereichs mehr und unser bisheriger Simplexalgorithmus ist nicht anwendbar. In diesem Fall behilft man sich mit der sogenannten *Zweiphasenmethode*, bei der man zuerst einmal, und zwar wieder mit dem Simplexalgorithmus, ein Optimierungsproblem "löst", um eine Startecke zu finden. Das sehen wir uns nun an.

Dazu setzen wir $1_m = [1, \dots, 1]^T \in \mathbb{R}^m$ und betrachten das erweiterte "Hilfsproblem"

$$\max -(x_0 + b^*)\,, \qquad \underbrace{\begin{bmatrix} 0_n & I_n \\ 1_{m-n} & \widetilde{A} \end{bmatrix}}_{=:\widehat{A}} \underbrace{\begin{bmatrix} x_0 \\ x \end{bmatrix}}_{=:\widehat{x}} \geq \underbrace{b - b^* \begin{bmatrix} 0_n \\ 1_{m-n} \end{bmatrix}}_{=:\widehat{b}}. \tag{2.29}$$

Die Zielfunktion $x_0 + b^*$, beschreibt die "Unzulässigkeit" des Optimierungsproblems: Wäre $b^* < 0$, dann wäre diese Funktion an der Stelle $x_0 = 0$ negativ und alles wäre in Ordnung; ist hingegen $b^* > 0$ – was ja der Fall ist, den wir hier untersuchen wollen – dann hat besteht an $x_0 = 0$ eine echte Unzulässigkeit durch diesen positiven Wert. Da wir diesen Defekt verkleinern, besser noch: minimieren, wollen, müssen wir in (2.29) also die *negative* Unzulässigkeit maximieren, der Normalform wegen. Unser Ziel ist es also, die Unzulässigkeit des Optimierungsproblems zu minimieren und es dadurch zulässig zu bekommen.

[49] Und in diese Form können wir es ja immer durch die Eliminierung eventueller freier Variablen bringen.

[50] Aus [44, Beispiel 2.2, S. 57], ein Spezialfall von Beispiel 1.2

[51] Auch hier handelt es sich eigentlich wieder um ein Problem aus der *Ganzzahloptimierung*, aber wieder einmal wird, rein zufällig, die kontinuierliche Optimallösung ganzzahlig sein.

[52] Und nur dann!

Da in (2.29) nun $\widehat{b} \leq 0$ ist, ist der Punkt $\widehat{x} = 0$ eine zulässiger Punkt, allerdings ist x_0 hier eine freie Variable – aber die können (und müssen) wir ja, wie schon gesehen, austauschen.

Nach endlich vielen Schritten mit unserem erweiterten Problem finden wir dann eine Optimallösung $\widehat{x}^*$ von (2.29) mit zugehörigen Werten x_0^* und x^*, die wir aus den entsprechenden Zeilen und Spalten des Simplextableaus bestimmen können. Setzen wir x_0^* und x^* in (2.29) ein, so erhalten wir, daß

$$Ax^* + \begin{bmatrix} 0_n \\ 1_{m-n} \end{bmatrix} x_0^* = \widehat{A} \begin{bmatrix} x_0^* \\ x^* \end{bmatrix} \geq \widehat{b} = b - b^* \begin{bmatrix} 0_n \\ 1_{m-n} \end{bmatrix},$$

also

$$Ax^* \geq b - (x_0^* + b^*) \begin{bmatrix} 0_n \\ 1_{m-n} \end{bmatrix}. \tag{2.30}$$

Die rechte Seite von (2.30) ist nun genau dann $\geq b$, wenn $x_0 + b^* \leq 0$ ist, also wenn $x_0^* \leq -b^*$ ist. In diesem Fall haben wir einen zulässigen Punkt gefunden; erfüllt umgekehrt x die Bedingung $Ax \geq b$, dann erfüllt jeder Punkt der Form $[x_0, x]$ mit $x_0 \leq -b^*$ aber auch (2.29). Das können wir folgendermaßen zusammenfassen.

Lemma 2.18 *Es gibt genau dann einen zulässigen Punkt x mit $Ax \geq b$, wenn es einen Punkt $\widehat{x} = [x_0, x]$ gibt, so daß $\widehat{A}\widehat{x} \geq \widehat{b}$ und $x_0 \leq -b^*$.*

Ist also bei unserer Optimallßung $x_0^* > -b^*$, dann kann es auch keinen zulässigen Punkt des Ausgangsproblems geben und dieses Optimierungsproblem wäre unsinnig, genauer, der dazu gehörende zulässige Bereich wäre leer. Das kann durchaus passieren, beispielsweise bei Transportproblemen im Sinne von Beispiel 2.17, bei denen mehr an den Zielen ankommen soll als in den Ausgangspunkten bereitsteht.

Ein abschliessender Blick auf (2.29) zeigt, daß unsere Bedingung an x_0^*, also $-x_0 \geq b^*$, nichts anderes bedeutet, als daß die Zielfunktion $-x_0 - b^*$ aus (2.29) *nichtnegativ* werden soll. Wir müssen also nicht mal unbedingt bis zum Optimum suchen, es genügt, denn Simplexalgorithmus so lange auf das Hilfsproblem anzuwenden, bis der Wert der (Hilfs–)Zielfunktion nichtnegativ ist. Wir fassen zusammen.

Lemma 2.19 *Ist $x_0^* \leq -b^*$, dann ist der Punkt x^* ist eine zulässige Ecke des Polyeders $F\left(A, \widehat{b}\right)$, wobei*

$$\widehat{b} = b - (x_0^* + b^*) \begin{bmatrix} 0_n \\ 1_{m-n} \end{bmatrix}.$$

Beweis: Daß x^* zulässig ist, wenn $x_0^* \leq -b^*$ ist, das wissen wir ja schon. Da außerdem $\widehat{x}^*$ eine Ecke von F^* ist[53] gibt es eine Menge J, $\#J = n + 1$, so daß

$$[1_m\, A]_J \begin{bmatrix} x_0^* \\ x^* \end{bmatrix} = \widehat{b}_J, \qquad \Longrightarrow \qquad A_J\, x^* = b_J^*$$

[53] F^* ist, wie man sich leicht vorstellen kann, der zulässige Bereich des "Hilfsproblems".

und da $A_J \in \mathbb{R}^{n+1\times n}$ Rang n hat gibt es eine Teilmenge $J' \subset J$, $\#J' = n$, so daß

$$\det A_{J'} \neq 0 \qquad \text{und} \qquad A_{J'}\, x^* = b^*_{J'},$$

also ist x^* eine Ecke. □

Korollar 2.20 *Eine zulässige Ecke $\widehat{x} = \left[x_0, x^T\right]^T$ von (2.29) enthält genau dann eine Ecke x unseres Ausgangsproblems, wenn $x_0 \leq -b^*$.*

Und damit haben wir unsere Startecke aufgespürt: Verpassen wir nun der *freien* Variablen den Wert $x_0 = -\max b_j$, dann ist x^* eine Ecke von (2.30), insbesondere ist $Ax^* \geq b$, also ist x^* eine zulässige Ecke und damit die Startecke, von der aus wir loslegen können.

Zur praktischen Realisierung betrachten wir wieder die "abgeschnittene" Form

$$y = \left[1_m\, \widetilde{A}\right] \begin{bmatrix} x_0 \\ x \end{bmatrix} - \left(\widetilde{b} - b^*\, 1_m\right) =: A^{(-1)} \begin{bmatrix} x_0 \\ x \end{bmatrix} - b^{(-1)},$$

mit der Hilfs–Zielfunktion[54]

$$f\,(x_0, x) = -x_0 - b^*,$$

und der Original–Zielfunktion

$$c\,(x_0, x) = c^T x$$

tauschen die freie Variable x_0 aus und erhalten, nach eventueller Zeilenvertauschung

$$\begin{bmatrix} x_0 \\ y^{(0)} \end{bmatrix} = A^{(0)} x^{(0)} - b^{(0)}, \qquad y^{(0)} \in \mathbb{R}^{m-n-1},\ A^{(0)} \in \mathbb{R}^{m-n,n+1},\ b^{(0)} \in \mathbb{R}^{m-n}, \tag{2.31}$$

sowie einen Vektor $c^{(0)}$, so daß $c(x) = \left(c^{(0)}\right)^T x^{(0)}$. Beim nun folgende Simplexalgorithmus, bei dem zwar bezüglich f minimiert, aber $c^{(r)}$ stets mitbestimmt werden muß, darf die Zeile, die zu x_0 gehört *nicht* mehr ausgetauscht werden[55], und so erhalten wir nach endlich vielen Schritten das Tableau

$$\begin{bmatrix} x_0 \\ y^{(r)} \end{bmatrix} = A^{(r)} x^{(r)} - b^{(r)},$$

aus dem wir den Wert von x_0 ablesen können; außerdem sagt uns die Zielfunktion, ob unsere Kante so gefundene Kante überhaupt zulässig ist, denn das ist genau der Fall, wenn $f\left(x_0, x^{(r)}\right) \geq 0$ ist. Nehmen wir an, das wäre der Fall, dann müssen wir nur noch x_0 loswerden, was wir dadurch erreichen, daß wir es mit der zuletzt "eingetauschten" Spalte vertauschen, also einen

[54] Weil man es gar nicht oft genug sagen kann: Für die eigentliche Optimierung des Hilfsproblems bräuchten wir den konstanten Term $-b^*$ nicht, aber da die Positivität dieser Zielfunktion bereits ein Abbruchkriterium ist, lohnt es sich.

[55] Schließlich könnte man die Zeilen, die zu freien Variablen gehören, sogar aus dem Simplextableau entfernen.

Austauschschritt mit dem Paar $(1, j_r)$ durchführen, was uns, nach einer *Spaltenvertauschung* die Darstellung

$$\begin{aligned}\left[y^{(r+1)}\right] &= A^{(r+1)} \begin{bmatrix} x_0 \\ x^{(r+1)} \end{bmatrix} - b^{(r+1)} = [a\, B] \begin{bmatrix} x_0 \\ x^{(r+1)} \end{bmatrix} - b^{(r+1)} \\ &= Bx^{(r+1)} + \left(x_0 a - b^{(r+1)}\right), \qquad B \in \mathbb{R}^{m\times n},\, a \in \mathbb{R}^m,\end{aligned}$$

liefert. Setzen wir da nun $x_0 = -\max b_j$ ein, dann haben wir die modifizierte Form unseres "Originalproblems", bei der $x^{(r+1)} = 0$ eine zulässige Ecke ist und können endlich den "normalen" Simplexalgorithmus anwerfen.

Um uns das "Mitführen" der "echten" Zielfunktion c, das heißt die Bestimmung von $c^{(r)}$ etwas leichter zu machen, bemerken wir, daß wir die Austauschregel (2.23) mit der Matrix B aus (2.25) auch als

$$c^{(r+1)} = B\, c^{(r)}$$

schreiben können, ein Update von $c^{(r)}_{n+1}$ ist nicht unbedingt nötig, weil wir die Zielfunktion ja so nur um eine Konstante abändern. Das heißt aber, daß wir die Zielfunktion c lediglich als "tote" Zeile unseres Hilfsproblems mitzuführen brauchen, also als Zeile, die nicht zur Auswahl der Pivotzeilen zugelassen ist.

Beispiel 2.21 *(Transportproblem aus Beispiel 2.17)*
Zuerst transformiert der erste Austauschschritt das erweiterte Tableau folgendermaßen:

	x_0	x_1	x_2	
y_1	1	−1	−1	−27
x_2	1	−1	0	−20
y_3	1	0	−1	−19
y_4	1	1	−1	0
c	0	1	3	−229
	−1	0	0	−9

$(1,4)$ $\longrightarrow$

	y_4	x_1	x_2	
y_1	1	−2	−2	−27
y_2	1	−2	−1	−20
y_3	1	−1	−2	−19
x_0	1	−1	−1	0
c	0	1	3	−229
	−1	1	1	−9

Die beiden unteren Zeilen sind nun tabu und das Simplexverfahren liefert jetzt das Tableau

$(2,2)$ $\longrightarrow$

	y_4	y_2	x_2	
y_1	0	1	−1	−7
x_1	0.5	−0.5	−0.5	−10
y_3	0.5	0.5	−1.5	−9
x_0	0.5	0.5	−0.5	10
c	0.5	−0.5	2.5	−229
	−0.5	−0.5	0.5	−9

Die Ecke die wir so gefunden haben ist also der Punkt $[10, 0]^T$. *Nach Vertauschen von* y_2 *und* x_0 *und den oben beschriebenen Umformungen läuft dann der "normale" Simplexalgorithmus*

```
%% SimplexStep1.m
%% Simplexverfahren, erster Schritt
%% Daten:        A,b,c wie gehabt
%% Ergebnise:    Modifiziertes Problem mit zul. Ecke 0

function [AA,bb,cc,xVec,yVec] = SimplexStep1( A,b,c )
  [m,n] = size( A );

  AA = [ [ ones( m,1 ), A ]; [0;c(1:n)]' ];        % Erweitere Problem
  [bmax,k] = max( b );
  bb = [ b - bmax * ones( m,1 ) ; c(n+1) ] ;
  cc = [ -1; zeros( n,1 ); -bmax ];

  [AA,bb,cc] = Austausch( 1,k,AA,bb,cc );          % Eliminiere freie Variable
  t = A( k,: ); A( k,: ) = A( m,: ); A( m,: ) = t; % Zeile ans Ende
  t = b( k ); b( k ) = b( m ); b( m ) = t;

  [AA,bb,cc,xVec,yVec,jj] = SimplexStep1a( AA,bb,cc,m,n );  % Simplexverfahren

  t = xVec( jj ); xVec( jj ) = yVec( m ); yVec( m ) = t; % Ruecktausch
  [AA,bb,cc] = Austausch( jj,m,AA,bb,cc );
  t = AA( :,jj ); AA( :,jj ) = AA( :,n+1 ); AA( :,n+1 ) = t;
  t = xVec( jj ); xVec( jj ) = xVec( n+1 ); xVec( n+1 ) = t;

  cc = [ AA( m+1,1:n )'; bb(m+1) + bmax ];
  bb = bb( 1:m )  + bmax * AA( 1:m,n+1 );
  AA = AA( 1:m,1:n );
  xVec = xVec(1:n); yVec = yVec(1:m);
%endfunction
```

Programm 2.7 `SimplexStep1.m`: Erster Schritt der Zweiphasenmethode, gibt "korrigiertes" Schema und Tabelle der Variablen zurück.

```
%% SimplexStep1.m
%% Simplexverfahren, erster Schritt, eigentliches Simplexverfahren
%% Daten:        A,b,c wie gehabt
%% Ergebnise:    Modifiziertes Problem mit zul. Ecke 0

function [AA,bb,cc,xVec,yVec,jj] = SimplexStep1a( AA,bb,cc,m,n )
  xVec = [ -1,( 1:n ) ]; yVec = [( -2:-1:-m ),0];  % Indiziere Variablen%
  jj = 1; HaveOpt = 0; StepNum = 0;                % Initialisierung

  disp( [[AA,bb];cc'] );        % Zeige Simplextableau

  while HaveOpt == 0
    StepNum = StepNum + 1;
    [j,k] = auxFindjk3( AA( 1:m-1,: ),bb( 1:m-1 ) ,cc ); % Finde Indizes

    if j == 0                     % Nix gefunden - Optimum
      HaveOpt = 1;
    elseif cc(n+2) >= 0            % Zielfunktion gross genug - Optimum
      HaveOpt = 1;
    else
      disp( [StepNum,j,k] );     % Zeige SchrittNr und Austausch
      t = xVec( j ); xVec( j ) = yVec( k ); yVec( k ) = t;
      [AA,bb,cc] = Austausch( j,k,AA,bb,cc );
      jj = j;
      disp( [[AA,bb];cc'] );        % Zeige Simplextableau
    end
  end

%endfunction
```

Programm 2.8 `SimplexStep1a.m`: Der eigentliche Simplexalgorithmus für Algorithmus 2.7. Der Grund für die Zerlegung war im wesentlichen, daß der Programmtext sonst nicht auf eine Seite gepasst hätte.

```
%%
%% SimplexPerm.m
%% Simplexverfahren mit bereits vertauschten Variablen
%% Daten:
%%   A  Matrix des Problems
%%   b  Nebenbedingungen
%%   c  Zielfunktion

function [x,opt] = SimplexPerm( A,b,c,xVec,yVec )
  HaveOpt = 0; StepNum = 0;
  [m,n] = size( A );

  disp( [[A,b];c'] );            % Zeige Simplextableau

  while HaveOpt == 0
    StepNum = StepNum + 1;
    [j,k] = auxFindjk3( A,b,c ); % Finde Indizes

    if j == 0
      HaveOpt = 1;               % Nix gefunden - Optimum!
    else
      disp( [StepNum,j,k] );     % Zeige SchrittNr und Austausch
      t = xVec( j ); xVec( j ) = yVec( k ); yVec( k ) = t;
      [A,b,c] = Austausch( j,k,A,b,c );
      disp( [[A,b];c'] );        % Zeige Simplextableau
    end
  end

  x = auxGenx( n,yVec,b );
  opt = c(n+1);
%endfunction
```

Programm 2.9 `SimplexPerm.m`: Simplexverfahren für bereits "vorvertauschte" Variable.

```
%%
%% SimplexTwoStep.m
%% Simplexverfahren, Zweischritt
%% Daten:
%%   A  Matrix des Problems
%%   b  Nebenbedingungen
%%   c  Kostenvektor
%% Ergebnise:
%%   Optimum und zug. Ecke

function [x,opt] = SimplexTwoStep( A,b,c )
  [m,n] = size( A );

  if max(b) > 0    % 0 ist KEINE zulaessige Ecke
    [A,b,c,xvec,yvec] = SimplexStep1( A,b,c );
  end

  [x,opt] = SimplexPerm( A,b,c,xvec,yvec );
%endfunction
```

Programm 2.10 `SimplexTwoStep.m`: Allgemeines Verfahren für lineare Programme, verwendet Zweischrittverfahren wenn nötig.

folgendermaßen ab:

	y_4	x_2	
y_1	-1	0	-9
x_1	1	-1	-9
y_3	0	-1	-10
y_2	-1	1	-2
	1	2	-220

$\overset{(2,2)}{\longrightarrow} * \overset{(1,3)}{\longrightarrow} * \overset{(2,1)}{\longrightarrow}$

	y_3	y_1	
x_1	1	-1	-8
x_2	-1	0	-10
y_4	0	-1	-9
y_2	-1	1	-3
	-2	-1	-191

und die Optimallösung ist also $[x, y] = [8, 10]$.

2.7 Kleine Komplexitätsbetrachtungen

Wie viele Schritte braucht nun so ein Simplexalgorithmus, um sein Ziel zu erreichen? Laut [45] reichen "im Normalfall" so etwa $2m$ bis $3m$ Austauschschritte aus, um das Optimum zu bestimmen. Allerdings sehen die "worst case"–Szenarien etwas übler aus. Hier ein Beispiel aus [45], das von Minty und Klee [33] stammt.

Beispiel 2.22 *Das Optimierungsproblem*

$$\begin{aligned} \sum_{k=1}^{n} 10^{n-k} x_k &= \max, \\ 2 \sum_{k=1}^{j-1} 10^{j-k} x_k + x_k &\leq 100^{j-1}, \qquad j = 1, \ldots, n \\ x &\geq 0, \end{aligned}$$

das in unserer Notation durch die Normalform

$$A = \begin{bmatrix} 1 & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & \ddots & & \\ & & & & 1 & \\ -1 & & & & & \\ -20 & -1 & & & & \\ -200 & -20 & -1 & & & \\ \vdots & \vdots & & \ddots & & \\ -2 \times 10^{n-1} & -2 \times 10^{n-1} & \ldots & -20 & -1 \end{bmatrix} \quad x \geq \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ -1 \\ -100 \\ -10000 \\ \vdots \\ -100^{n-1} \end{bmatrix}$$

dargestellt wird, benötigt $2^n - 1$ *Austauschschritte, wenn man mit der Startecke* $x^{(0)} = 0$ *beginnt.*

Man kann sich dieses Beispiel sehr schön experimentell ansehen und erkennt dabei auch, welche Bedeutung eine gute Pivotstrategie haben. Mit der Methode von Algorithmus 2.5, also

der "koordinatenweisen" Pivotisierung benötigt das Verfahren `Simplex3` immer die vollen $2^n - 1$ Austauschschritte, während das Verfahren `Simplex4`, das auf der "raffinierteren" Pivotstrategie von Algorithmus 2.6 basiert, nach gerade mal *einem* Schritt am Ziel ist.

Wenn nun also Algorithmus 2.6 so eine tolle Pivotregel ergibt, warum verwendet man sie dann nicht auch? Ganz einfach: im "normalen" Simplexalgorithmus muß man pro Rechenschritt $O(n)$ Vergleichsoperationen durchführen, um das maximale c_j zu finden und dann $O(m-n)$ Rechenoperationen, um die Pivotzeile zu bestimmen; bei der "Totalpivotsuche" müßte man im schlimmsten Fall $O(m-n)$ Rechenoperationen für *jede* der n Spalten durchführen, also $O\left(n(m-n)\right)$ Rechen- und ebensoviele Vergleichsoperationen. Das heißt, daß der Aufwand pro Schritt des Simplexalgorithmus von $O(m)$ auf $O\left(n(m-n)\right)$, für große Werte von m und n nicht gerade erstrebenswert.

To isolate mathematics from the practical demands of the sciences is to invite the sterility of a cow shut away from the bulls.

P. Chebyshev

Lineare Optimierung – Beispiele und Anwendungen

3

Bevor wir uns wieder in das Vergnügen der Theorie stürzen, wollen wir uns zuerst einmal ein paar Beispiele ansehen, in denen wir den Simplexalgorithmus *anwenden* können. Interessant wird hierbei vor allem werden, wie wir die Probleme in einer Form aufbereiten können, daß wir sie in unsere `Octave`–Programme stecken und von diesen lösen lassen können. Hochgestochen gesprochen befassen wir uns also nun mit der *Modellierung* von Optimierungsproblemen. Die Beispiele stammen übrigens aus [20][56].

Allerdings, und sei es nur, um Langeweile zu vermeiden, verwenden wir jetzt eine andere Normalform für unsere linearen Optimierungsprobleme, nämlich beinahe[57] wie in (2.11) die "algorithmische Normalform"

$$\min_x c^T x, \qquad Ax \leq b, \qquad x \geq 0. \tag{3.1}$$

Der Grund dafür ist relativ profan: Wir verwenden einfach etwas andere `Octave`–Funktionen[58] für den Simplexalgorithmus, die jetzt nur noch "richtige" Ausgaben liefern. Aber natürlich lassen sich ja alle Normalformen durch etwas Vorzeichenspielerei ineinander überführen.

3.1 Das Diät–Problem

Beginnen wir erst einmal mit einem ganz typischen, einfachen Problem, ganz änlich zur "Schuhfabrik".

Beispiel 3.1 (Frühstücksplanung) *Eine Hausfrau versucht für Ihre Familie ein optimales Frühstück zusammenzustellen. Dafür stehen ihr*[59] *zwei verschiedene Typen von Getreideflocken*[60]*, nämlich*

[56] Eine sehr empfehlenswerte, anschauliche und auch noch, wie es sich für Dover–Reprints gehört, *preiswerte* Einführung in die lineare Optimierung.

[57] "Beinahe" deswegen weil wir jetzt minimieren und nicht maximieren.

[58] Die wir jetzt aber nicht mehr extra abdrucken wollen.

[59] Ja, das Beispiel stammt aus den USA.

[60] Auf gut neudeutsch auch als "Cerealien" bezeichnet – das kommt davon, wenn's an Zerebralien mangelt.

Crunchies *und* Krispies *zur Verfügung, die zwei Spurenelemente,* Thiamin[61] *und* Niacin[62] *in unterschiedlicher Anzahl enthalten, unterschiedlichen Brennwert in Kalorien liefern und natürlich unterschiedlich teuer sind. Das ideale Frühstück versorgt die Familie mit einem gewissen Mindestmaß an "Vitaminen" und Kalorien und ist dabei natürlich möglichst billig. Die genauen Werte sind in der folgenden Tabelle aufgelistet:*

	Crunchies	*Krispies*	*Benötigt*
Thiamin (in mg)	*0.10*	*0.25*	*1*
Niacin (in mg)	*1.00*	*0.25*	*5*
Kalorien	*110*	*120*	*400*
Preis	*3.8*	*4.2*	

Das Problem ist klar: Was ist die optimate Diät, die diese Randbedingungen erfüllt?

Nun, dieses Problem ist, was die Modellierung angeht, noch richtig einfach, denn wir müssen nur die Bedingungen in Ungleichungsform hinschreiben. Seien dazu x_1 die Menge der verwendeten Crunchies und x_2 die Menge an Krispies, dann erhalten wir die Ungleichungen

$$\begin{aligned} 0.1\,x_1 + .25\,x_2 &\geq 1 \\ x_1 + .25\,x_2 &\geq 5 \\ 110\,x_1 + 120\,x_2 &\geq 400 \end{aligned}$$

und zu minimieren sind die Kosten $3.8\,x_1 + 4.2\,x_2$. Nachdem unsere Normalform aus (3.1) die Ungleichungen als "$\leq$" geschrieben haben will, erhalten wir also das Optimierungsproblem[63]

$$\min 3.8\,x_1 + 4.2\,x_2, \qquad \begin{bmatrix} -0.1 & -0.25 \\ -1 & -0.25 \\ -110 & -120 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} -1 \\ -5 \\ -400 \end{bmatrix},$$

was ein klarer Fall für die Zweiphasenmethode ist. Also ist alles klar, oder? Geben wir das Problem in den Rechner ein, dann erhalten wir mit der Eingabe

```
octave> A = [ -.1 -.25; -1 -.25; -110 -120]; b = [-1 -5 -400]';
octave> c = [ 3.8 4.2]';
octave> [x,opt] = SimSimplex( A,b,c )
```

die etwas überraschende Ausgabe

```
**** Unbeschraenkt ****
x =

  5.00000
  0.00000

opt = 19.000
```

[61] Synonym für Vitamin B_1, siehe [38].

[62] Synonym für Nicotinsäure, die Nebenwirkungen in [38] liest man besser nicht.

[63] Von jetzt an schreiben wir die allgegenwärtige Randbedingung $x \geq 0$ **nicht** mehr explizit hin.

die noch nicht einmal zulässig ist, denn die erste Nebenbedingung ist nicht erfüllt. Allerdings sehen wir ja auch an der Ausgabe, wo die Schwierigkeiten herkommen: Das Optimierungsproblem ist *unbeschränkt*, und da funktioniert unser Simplexalgorithmus halt nun einmal nicht[64]. Das sieht man ja auch an der Problemstellung selbst, denn die einfachste Möglichkeit, die Nebenbedingungen zu erfüllen besteht einfach darin, eine Packung von jeder Sorte in sich hineinzustopfen, und wenn's nicht reicht, dann halt noch eine und so weiter. Damit wir unsere Methode anwenden können, müssen wir also das Problem künstlich beschränken. Eine Möglichkeit besteht darin, x_1 und x_2 *individuell* zu beschränken, indem man nachsieht, aus welcher Menge das "kleinste" Frühstück aus Crunchies bestehen muß (nämlich $x_1 = 10$) und wieviele Krispies man mindestens essen muß, um alle "Nährstoffe" aufzunehmen (das ist $x_2 = 20$). Dann können wir die Nebenbedingungen $x_1 \leq 10$ und $x_2 \leq 20$ hinzufügen und erhalten

```
octave> AA = [ A; [ 1 0; 0 1] ]; bb = [ b; 10; 20 ];
octave> [x,opt] = SimSimplex( AA,bb,c )
x =

  4.4444
  2.2222

opt = 26.222
```

das optimale Frühstück kostet also 26.2222 Cent[65]. Eine andere Möglichkeit bestünde darin, den Preis zu beschränken, also beispielsweise nur Früstücke für weniger als einen Dollar:

```
octave> AA = [ A; [ 3.8 4.2 ]  ]; bb = [ b; 100 ];
octave> [x,opt] = SimSimplex( AA,bb,c )
x =

  4.4444
  2.2222

opt = 26.222
```

Und siehe da: Das Ergebnis ist wieder richtig. Wird man hingegen zu knauserig, dann kommt man erneut in Schwierigkeiten:

```
octave> AA = [ A; [ 3.8 4.2 ]  ]; bb = [ b; 15 ];
octave> [x,opt] = SimSimplex( AA,bb,c )
x =
```

[64] Nur um das nochmal klarzustellen: Das Minimum existiert natürlich mit und ohne Beschränkung, nur das **Verfahren** funktioniert nicht!

[65] Und enthält im übrigen rund 756 Kalorien, also fast doppelt so viel wie gewünscht. Vielleicht sollte man dochmal ein grundlegend anderes Frühstück in Betracht ziehen ...

```
  5
  0

opt = 19
```

aber an der Tatsache, daß die zusätzliche Nebenbedingung durch die berechnete Optimallösung verletzt wurde, sieht man schon, daß irgendwas faul sein muß.

3.2 Transportprobleme

Transportprobleme sind immer vom Typ wie in Beispiel 2.17: Ressourcen müssen von Ausgangspunkten zu Zielpunkten transportiert werden, wobei *alle* Ausgangspunkte mit *allen* Zielpunkten als verbunden angenommen werden. Die Entfernungen (oder Kosten) von eine Ausgangs- zu einem Zielpunkt sowie die in den Ausgangspunkten vorrätigen und die in den Zielpunkten benötigten Ressourcen sind typischerweise in einer Matrix aufgelistet:

	Z_1	$\dots$	Z_n	
A_1	a_{11}	$\dots$	a_{1n}	a_1
$\vdots$	$\vdots$	$\ddots$	$\vdots$	
A_m	a_{m1}	$\dots$	a_{mn}	a_m
	z_1	$\dots$	z_n	

Dabei bezeichnet $a_{11}, \dots, a_{mn}$ die *Kostenmatrix*, $a_1, \dots, a_m$ die vorhandenen und $z_1, \dots, z_n$ die benötigten Ressourcen. Damit das Problem überhaupt lösbar ist, muß natürlich

$$a_1 + \dots + a_m \geq z_1 + \dots z_n$$

sein.

Beispiel 3.2 (Kühlschränke) *Eine Firma stellt in zwei Fabriken, F_1 und F_2, Kühlschränke her, die in den Läden*[66] *S_1, S_3, S_3 verkauft werden sollten. Die Kosten-/Ressourcen–Matrix ist wie folgt:*

	S_1	S_2	S_3	
F_1	8	6	10	11
F_2	9	5	7	14
	10	8	7	

Wie ist der optimale Transport?

Transportprobleme zeichnen sich dadurch aus, daß man sehr viele Variablen hat, die man am zweckmäßigsten *doppelt* indiziert, nämlich als x_{jk}, wobei x_{jk} die Menge bezeichnet, die vom Ausgangspunkt j zum Zielpunkt k transportiert wird. Die Gesamtkosten sind dann immer

$$\sum_{j=1}^{m} \sum_{k=1}^{n} a_{jk} x_{jk}.$$

[66] "Shop".

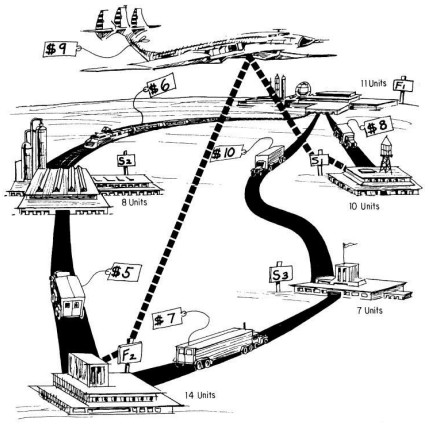


Abbildung 3.1: Kühlschränke und deren Transportwege. Aus [20].

In unserem Beispiel haben wir also die Variablen

$$x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{23} \qquad \Longrightarrow \qquad x = [x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{23}]^T$$

auch in einen Vektor angeordnet, indem wir die sogenannte *lexikographische*[67] Ordnung verwenden. Unser Beispiel liefert nun die Nebenbedingungen

$$\begin{array}{rrrrrrcl}
x_{11} & +x_{12} & +x_{13} & & & & \leq & 11 \\
 & & & x_{21} & +x_{22} & +x_{23} & \leq & 14 \\
x_{11} & & & +x_{21} & & & \geq & 10 \\
 & x_{12} & & & +x_{22} & & \geq & 8 \\
 & & x_{13} & & & +x_{23} & \geq & 7
\end{array}$$

Die ersten beiden Ungleichungen sind die Beschränkungen an die Ressourcen, die anderen drei betreffen das Minimum, das an den Zielpunkten ankommen soll. Damit können wir uns auch schon wieder ans Modellieren machen: Nachdem wir noch ein paar unpassende Vorzeichen umgedreht haben, erhalten wir die folgenden Parameter:

$$A = \begin{bmatrix}
1 & 1 & 1 & & & \\
 & & & 1 & 1 & 1 \\
-1 & & & -1 & & \\
 & -1 & & & -1 & \\
 & & -1 & & & -1
\end{bmatrix}, \qquad b = \begin{bmatrix} 11 \\ 14 \\ -10 \\ -8 \\ -7 \end{bmatrix}, \qquad c = \begin{bmatrix} 8 \\ 6 \\ 10 \\ 9 \\ 5 \\ 7 \end{bmatrix}$$

Und ab geht's in den Computer:

[67] Indizes werden angeordnet wir im Lexikon: zuerst ordnet man nach dem ersten Eintrag, dann nach dem zweiten und so weiter.

```
octave> A = [ 1 1 1 0 0 0; 0 0 0 1 1 1; -1 0 0 -1 0 0; 0 -1 0 0 -1 0;
    0 0 -1 0 0 -1];
octave> b = [ 11 14 -10 -8 -7 ]'; c = [ 8 6 10 9 5 7 ]';
octave> [x,opt] = SimSimplex(A,b,c)
x =

  10.00000
   1.00000
   0.00000
   0.00000
   7.00000
   7.00000

opt = 170
```

Was man sieht ist, daß bei Transportproblemen zwar die Anzahl der Variablen dramatisch steigt (man hat mn Variablen bei einer $m \times n$ Kostenmatrix), daß man dafür aber sehr einfach strukturierte Matrizen hat: In den ersten n Zeilen stehen n verschobene Zeilen von je m Einsen und darunter n nebeneinandergestellte, mit -1 multiplizierte Einheitsmatrizen.

Beispiel 3.3 (Noch ein Transportproblem) *Ausrüstungsgegenstände sollen von drei Basen auf fünf andere Basen verteilt werden, wobei die zurückgelegte Gesamtdistanz minimiert werden soll. Die Vorgaben, wie in [20, S. 20] sind wie folgt:*

	MacDill	*March*	*Davis-Monthan*	*McConnell*	*Pinecastle*	
Oklahoma City	*938*	*1030*	*824*	*136*	*995*	*8*
Macon	*346*	*1818*	*1416*	*806*	*296*	*5*
Columbus	*905*	*1795*	*1590*	*716*	*854*	*8*
	3	*5*	*5*	*5*	*3*	

Das sind jetzt also solide 15 *Variable und da wird's langsam heftig.*

Es wäre jetzt schon ziemlich eklig, diese Matrix noch von Hand einzugeben, weswegen wir ein kleines `Octave`–Programm namens `TransMat` verwenden, das die Strukturmatrix automatisch generiert. Und dann brauchen wir nur noch unsere Werte einzutippen,

```
octave> A = TransMat( 3,5 ); b = [ 8 5 8 -3 -5 -5 -5 -3 ]';
octave> c = [ 938 1030 824 136 995 346 1818 1416 806 296 905 1795
   1590 716 854 ]';
```

um auf die Optimallösung mit 16384 Kilometern zu kommen.

Übung 3.1 Zeigen Sie, daß Transportprobleme mit ganzzahliger Kostenmatrix immer auch ganzzahlige Lösungen haben. ♢

```
%%  TransMat.m (Optmierung fuer Hoerer aller Fachbereiche)
%%  ------------------------------------------------------
%%  Matrix zu Transportproblem
%%  Eingabe:
%%    m,n    Dimension

function A = TransMat( m,n )
  A0 = zeros( m, m*n );
  A1 = [ ones( 1,n ); zeros( m-1,n ) ];
  for j = 0:m-1
    A0( 1:m, j*n+1:(j+1)*n ) = shift( A1, j );
  end
  A1 = repmat( -eye( n ), 1,m );
  A = [ A0; A1 ];
```

Programm 3.1 `TransMat.m`: Generierung von Matrizen zum Transportproblem nach dem einfachen Schema.

3.3 Zuordnungsprobleme

Zuordnungsprobleme versuchen Ressourcen und Aufgaben so einander zuzuordnen, daß ein vorgegebener Nutzen maximiert wird. Eine *Zuordnung* zweier Mengen[68] ist eine "Funktion", die jedem Element der einen Menge[69] *eindeutig* ein Element der anderen Menge[70] zuordnet. Alternativ ist eine *Zuordnungstabelle* eine quadratische Matrix, die in jeder Spalte und jeder Zeile *genau eine* Eins stehen hat.

Beispiel 3.4 (Personal und Fähigkeiten) *Einer Militäreinheit*[71] *stehen drei neue Mitarbeiter, Able, Baker und Charlie, zur Verfügung, die für drei Aufgaben eingesetzt werden können, nämlich am Schreibtisch, am Funkgerät oder am Computer. In vorhergehenden Tests wurden ihre Fähigkeiten wie folgt ermittelt:*

	Funk	*Computer*	*Schreibtisch*
Able	*5*	*4*	*7*
Baker	*6*	*7*	*3*
Charlie	*8*	*11*	*2*

Wie setzt man die drei Soldaten so ein, daß ein möglichst hoher Wert erreicht wird.

[68] Die gleichviele Elemente enthalten müssen.
[69] Also jeder Ressource.
[70] Also eine Aufgabe.
[71] Nicht meine Erfindung, sondern aus [20, S. 56–61]!

Abbildung 3.2: Die drei Mitarbeiter und ihre Fähigkeiten. Aus [20]

Auch hier beschreibt wieder x_{jk}, in welchem Maße Soldat Nummer j Job Nummer k ausübt[72] und die Nebenbedingungen sind

$$\begin{array}{rrrrrrrrrcl}
x_{11} & +x_{12} & +x_{13} & & & & & & & = & 1 \\
 & & & x_{21} & +x_{22} & +x_{23} & & & & = & 1 \\
 & & & & & & x_{31} & +x_{32} & +x_{33} & = & 1 \\
x_{11} & & & +x_{21} & & & +x_{31} & & & = & 1 \\
 & x_{12} & & & +x_{22} & & & +x_{32} & & = & 1 \\
 & & x_{13} & & & +x_{23} & & & +x_{32} & = & 1
\end{array}$$

Kommt uns irgendwie bekannt vor, oder? Wenn wir das nämlich in Ungleichungen umschreiben, dann steht da bis auf die rechte Seite nicht anderes als ein "gedoppeltes" Transportproblem! Und dafür haben wir ja schon unsere Routinen. Aber nicht vergessen: Da wir *maximieren* wollen, müssen wir die Zielfunktion mit -1 multiplizieren. Also:

```
octave> A = TransMat( 3,3 ); b = [ ones( 3,1 ) ; -ones( 3,1 ) ];
octave> A = [ A; -A ]; b = [ b; -b ];
octave> c = -[ 5 4 7 6 7 3 8 11 2 ]';
octave> [x,opt] = SimSimplex( A,b,c )
x =

  0
  0
  1
  1
  0
```

[72] Man kann zeigen, daß bei der Optimallösung $x_{jk} = 1$ sein muß, das liegt an der Struktur des Problems.

```
  0
  0
  1
  0

opt = -24
```

und die Lösung "Able am Schreibtisch, Baker am Funkgerät und Charlie am Computer" ist ja auch das, worauf man ohne Computer hätte kommen können.

3.4 Fluß in Netzwerken

Die letzte Problemklasse sieht schon richtig "fortgeschritten", um nicht zu sagen professionell aus. Es geht darum, auf verschlungenen Wegen möglichst viel von A nach B zu transportieren. Diese verschlungenen Wege werden in Form eines Netzwerks[73] dargestellt, siehe Abb. 3.3. Wie

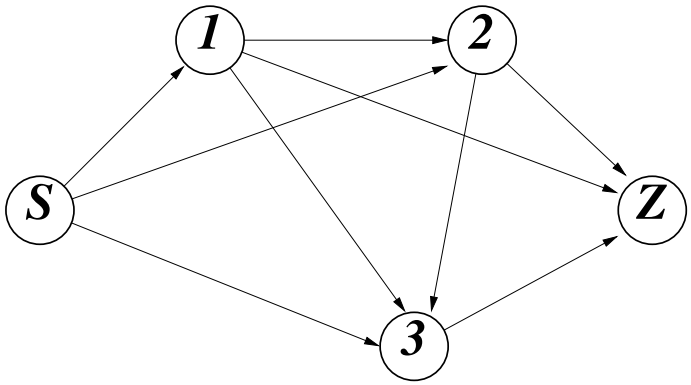


Abbildung 3.3: Die Verbindungen im Netzwerk – wohin kann man von wo aus kommen. Der *gerichtete* Graph bedeutet, daß es keine Schleifen gibt.

man sieht, gibt es nun viele verschiedene Möglichkeiten vom Startpunkt "S" zum Zielpunkt "Z" zu gelangen, beispielsweise den Weg $S \to 1 \to 2 \to Z$ oder $S \to 1 \to 3 \to Z$ und so weiter.

Beispiel 3.5 (Maximaler Transport oder "Fluß" im Netzwerk) *Das Netzwerk aus Abb. 3.3 stelle alle Möglichkeiten dar, mit öffentlichen Verkehrsmitteln von S nach Z zu gelangen, wobei $1, 2, 3$ die Umsteigepunkte seien. Wieviele Fahrgäste kann man maximal von S nach Z bringen, wenn die Kapazitäten der Verkehrsmittel*[74] *wie in Abb. 3.4 dargestellt sind, und wie muß man die Fahrgäste auf die einzelnen Verkehrsmittel verteilen?*

[73] In der Sprache der diskreten Mathematik: ein gerichteter Graph.
[74] Sagen wir in der Einheit "100 Fahrgäste".

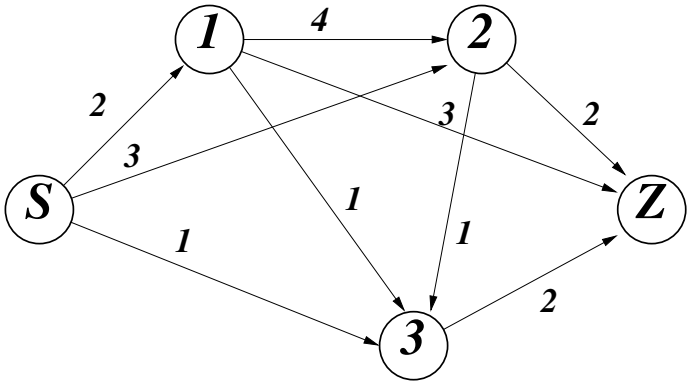


Abbildung 3.4: Die Kapazitäten der einzelnen Kanten des Netzwerk aus Abb. 3.3.

Wieder bezeichnen wir mit x_{jk} die Anzahl der Passagiere, die von Knoten j nach Knoten k fahren, wobei Knoten 0 der Startpunkt und Knoten 4 der Zielpunkt ist. Damit werden die Kapazitätsbeschränkungen, ohne daß wir irgendwie nachdenken müssen, sofort zu Nebenbedingungen:

$$\begin{array}{rcl}
x_{01} & \leq & 2 \\
x_{02} & \leq & 3 \\
x_{03} & \leq & 1 \\
x_{12} & \leq & 4 \\
x_{13} & \leq & 1 \\
x_{14} & \leq & 3 \\
x_{23} & \leq & 1 \\
x_{24} & \leq & 2 \\
x_{34} & \leq & 2
\end{array} \tag{3.2}$$

Das war der einfache Teil. Was wir außerdem noch fordern müssen, ist, daß niemand an einem Umsteigepunkt vergessen wird und dort verhungern muß, daß also alles, was in einen Knoten *hineinfließt*, auch wieder *herausfließen* muß, was wir mathematisch als

$$\sum_j x_{jk} = \sum_j x_{kj}, \qquad \forall k$$

schreiben können: Die Summe über die x_{jk} ist je gerade die Menge, die in den Knoten k hineintransportiert wird und die Summe über die x_{kj} die Menge, die von Knoten k in andere Knoten weitergeleitet wird. In unserem Beispiel entnehmen wir Abb. 3.4 die Nebenbedingungen

$$\begin{array}{rcl}
x_{01} - x_{12} - x_{13} - x_{14} & = & 0 \\
x_{02} + x_{12} - x_{23} - x_{24} & = & 0 \\
x_{03} + x_{13} + x_{23} - x_{34} & = & 0
\end{array} \tag{3.3}$$

und wie wir die in Ungleichungsbedingungen umwandeln, das wissen wir ja schon. Bleibt noch, daß das was wir in das System reinstecken, also was aus S "hinausfließt", auch in Z ankommen

muß. Nennen wir diesen Wert t, dann erhalten wir schließlich noch die beiden Nebenbedingungen

$$\begin{array}{rrrrrrrrr} -x_{01} & -x_{02} & -x_{03} & & & & +t & = & 0 \\ & & & x_{14} & +x_{24} & +x_{34} & -t & = & 0 \end{array} \tag{3.4}$$

Und was ist unser Ziel? Wir wollen ja den *Gesamtfluß* maximieren, also nichts anderes als den Wert t, der gerade in unserer Nebenbedingung aufgetaucht ist. Dazu müssen wir also t als *zusätzliche* Variable einführen und haben unser Problem fertig modelliert. Jetzt müssen wir es nur noch computergerecht aufbereiten, wobei wir t als zusätzliche, zehnte Variable ansetzen. Das tun wir geschickterweise zuerst für die Nebenbedingungen (3.3) und (3.4), denn die können wir dann mit umgedrehtem Vorzeichen übereinanderstapeln:

```
octave> A = [ 1 0 0 -1 -1 -1 0 0 0 0;
              0 1 0 1 0 0 -1 -1 0 0;
              0 0 1 0 1 0 1 0 -1 0;
              -1 -1 -1 0 0 0 0 0 0 1;
              0 0 0 0 0 1 0 1 1 -1 ];
octave> A = [ A; -A ];
```

Dann fügen wir noch die Nebenbedingungen aus (3.2) hinzu

```
octave> A = [ A; [ eye(9), zeros( 9,1 ) ] ];
```

setzen unsere rechte Seite und die Zielfunktion an, wobei wir beachten müssen, daß wir *maximieren* wollen, also als Zielfunktion $-t$ setzen sollten,

```
octave> b = [ zeros( 1,10 ), [ 2 3 1 4 1 3 1 2 2 ] ]';
octave> c = [ zeros( 1,9 ), -1 ]';
```

und erhalten die Optimallösung als

```
octave> [x,opt] = SimSimplex( A,b,c )
x =

  2
  3
  1
  0
  0
  2
  1
  2
  2
  6

opt = -6
```

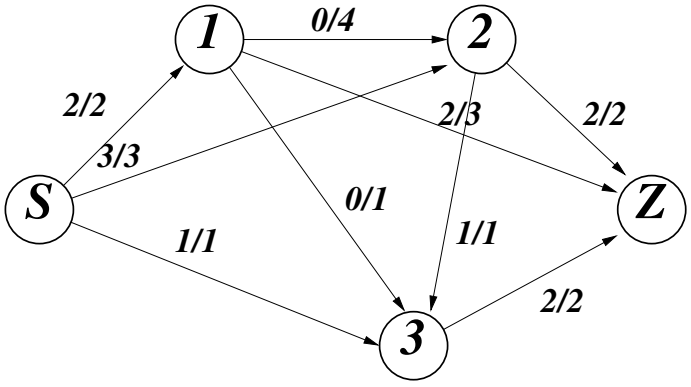


Abbildung 3.5: Fluß der Optimallösung im Vergleich zu deren Kapazität.

Die Lösung ist in Abb. 3.5 dargestellt. Man sieht ihr ein typisches Phänomen von Optimallösungen von Netzwerkproblemen an: Alle Kanten, die aus S herausführen, sind *saturiert*, also voll belegt.
Man kann auch allgemeinere Netzwerkprobleme auf diese Art und Weise angehen, indem man für n Knoten (Start und Ziel mitgezählt!) normalerweise eine *Verbindungsmatrix*

$$V = [v_{jk} \ : \ j, k = 1, \ldots, n]$$

verwendet, in der nur Nullen und Einsen stehen – und zwar $v_{jk} = 1$, wenn es zwischen den Knoten j und k eine Verbindung gibt und Null, wenn diese Verbindung nicht existiert. Dabei ist $v_{jk} = v_{kj} = 1$ zwar möglich, aber nicht zwingend vorgeschrieben[75]. Die Nebenbedingungen ergeben sich dann wieder aus den Kapazitäten der Kanten und aus der "Erhaltungseigenschaft", daß alles, was in einen Knoten fließt, auch wieder rausmuss: In formaler Schreibweise heißt das

$$\sum_{j=1}^{n} v_{jk}\, x_{jk} = \sum_{j=1}^{n} v_{kj}\, x_{kj}, \qquad k = 1, \ldots, n.$$

Übrigens kann man da auch großzügiger sein, indem man lediglich

$$\sum_{j=1}^{n} v_{jk}\, x_{jk} \geq \sum_{j=1}^{n} v_{kj}\, x_{kj}, \qquad k = 1, \ldots, n$$

fordert – jetzt darf in jedem Knoten auch was versickern. Und dann steckt man nur noch t in das System und minimiert die Zielfunktion $z(x, t) = -t$.

3.5 Spieltheorie und die zwei Phasen

Auch für die Lösung von spieltheoretischen Problemen ist der Simplexalgorithmus überraschend hilfreich. Dazu brauchen wir aber zuerst zwei Begriffe, nämlich den der *Auszahlungsmatrix* und

[75] Man denke nur an Einbahnstraßen.

den der *Strategie*, zumindest in einem *Zweipersonen–Nullsummenspiel*. Hierbei spielen[76] zwei Spieler gegeneinander, und was der eine gewinnt, ist genau das, was der andere verliert, das ist das Nullsummenspiel[77]. Nehmen wir weiterhin an, daß es für das Spiel nur endlich viele verschiedene Verläufe gibt[78], dann kann sich jeder Spieler von vornherein für jeden möglichen Ablauf sein Verhalten überlegen und und kommt so zu einer großen aber immer noch endlichen Menge von Strategien.

Beispiel 3.6 *Im guten alten "Stein, Schere, Papier" jeder der beiden Spieler in jeder Runde drei Strategien, nämlich Stein, Schere oder Papier. Hier ist die Aktion sogar* unabhängig *vom Verhalten des Gegners, von dem man ja normalerweise nichts weiß. Sowas nennt man ein* Spiel mit unvollständiger Information.

Bei einem Spiel wählen also beide Spieler jeweils eine Strategie aus und diese Auswahl determiniert bereits das Spiel, das heißt, jetzt kommt irgendjemand und sagt uns, wieviel der eine Spieler gewonnen und damit der andere Spieler verloren hat. Hat also Spieler 1 sagen wir m Strategien zur Auswahl und Spieler 2 entsprechend n Strategien, dann gibt es bei Verwendung der Strategien j und k eine Auszahlung $x(j,k) = x_{jk}$ und die Matrix $X \in \mathbb{R}^{m\times n}$ ist die *Auszahlungsmatrix* des Spieles, in der sozusagen die Spielregeln codiert sind.

Beispiel 3.7 *Die Auszahlungsmatrix (aus Sicht von Spieler* 1*) zu "Stein, Schere, Papier" ist*

$$X = \begin{bmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix}$$

Nun ist die Auswahl von solchen *reinen* Strategien aber nicht flexibel genug, weswegen jeder Spieler seine Strategien mit einer gewissen Wahrscheinlichkeit versieht: Spieler 1 spielt also die erste Strategie mit Wahrscheinlichkeit p_1, die zweite mit Wahrscheinlichkeit p_2, und so weiter. Die so erhaltenen *gemischten* Strategien sind also Wahrscheinlichkeitsvektoren $p = (p_1, \ldots, p_m)$ und $q = (q_1, \ldots, q_n)$ und die zu erwartende Auszahlung ist

$$x(p,q) = \sum_{j=1}^{m}\sum_{k=1}^{n} x_{jk}\, p_j\, q_k = p^T X q.$$

Die Ziele der Spieler sind nun unterschiedlich: Spieler 1 will p so wählen, daß $x(p,q)$ *maximal* wird, Spieler 2 hingegen sucht sich sein q so, daß der Wert *minimal* wird. Spieltheoretiker sind nun aber konservativ und so besteht die beste Strategie für Spieler 1 darin, den garantierten Minimalgewinn zu maximieren, also

$$\max_p \min_q\ x(p,q)$$

[76] Wen wird es bei diesem Namen überraschen.

[77] Daß dieser Begriff heutzutage in Politik und Wirtschaft oft anders verwendet wird, nämlich für eine Situtation, in der man am Ende ohne Gewinn oder Verlust dasteht, zeigt nur, daß man in diesen Kreisen von Spieltheorie keine Ahnung hat, dafür aber von viersilbigen Worten zu beeindrucken ist.

[78] Was absolut keine Einschränkung und nur realistisch ist.

zu bestimmen, während Spieler 2 den garantierten Maximalgewinn minimieren will, sich also für

$$\min_q \max_p \; x(p,q)$$

interessiert. Für allgemeine Funktionen f gilt nun lediglich

$$\min_y \max_x \; f(x,y) \geq \max_x \min_y \; f(x,y),$$

was keinen der beiden Spieler so richtig zufriedenstellt, aber im Falle der spieltheoretischen Auszahlungsfunktion für gemischte Strategien gibt es glücklicherweise eine Gleichgewichtsaussage [34] aus dem Jahre 1928, das sogenannte *Minimax–Theorem*, das uns sagt, daß beide Spieler eigentlich dasselbe wollen.

Satz 3.8 (Minimax–Theorem) *Es gibt immer optimale gemischte Strateien p^*, q^* für die beiden Spieler, so daß*

$$v := x\left(p^*, q^*\right) = \max_p \min_q \; x(p,q) = \min_q \max_p \; x(p,q). \tag{3.5}$$

Dieser Satz ist wirklich bemerkenswert da "normalerweise" Maximierungs- und Minimierungsprozesse nicht so einfach vertauscht werden können. Der Wert v, der Erwartungswert bei optimaler Spielweise beider Spieler, wird als der *Wert des Spieles* bezeichnet und ein Spiel heißt *fair*, wenn $v = 0$ ist.

Außerdem kann man nachweisen[79], daß

$$\min_q x\left(p,q\right) = \min_{k=1,\ldots,n} \sum_{j=1}^{m} x_{jk}\, p_j \qquad \text{und} \qquad \max_p x\left(p,q\right) = \max_{j=1,\ldots,m} \sum_{k=1}^{n} x_{jk}\, q_k \tag{3.6}$$

gilt, und somit können wir unser Problem der optimalen gemischten Strategien endlich mathematisieren: Der Wert v des Spieles ist die garantierte erwartete Mindestauszahlung für Spieler 1, so lange er nur die optimale Strategie p^* spielt, also

$$v \leq x\left(p^*, q\right) \qquad \Rightarrow \qquad v \leq \min_q x\left(p^*, q\right) = \min_{k=1,\ldots,n} \sum_{j=1}^{m} x_{jk}\, p_j^* = \min_{k=1,\ldots,n} \left(X^T p^*\right)_k,$$

wobei wir

$$X := \left[x_{jk} \; : \; \begin{array}{rcl} j & = & 1,\ldots,m \\ k & = & 1,\ldots,n \end{array} \right]$$

setzen. Dasselbe Spiel mit der anderen Ungleichung, $v \geq x\left(p, q^*\right)$, also die Aussage, daß Spieler 2 durch optimale Strategiewahl die erwartete Auszahlung für Spieler 1 unter v halten kann, liefert uns, daß $\left(Xq^*\right)_j \leq v$ sein muß, $j = 1,\ldots,m$. Mit anderen Worten: Wir erhalten die Bedingungen

$$X^T p^* \geq v \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}, \qquad Xq^* \leq v \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}, \tag{3.7}$$

[79] Das Zauberwort heißt wieder einmal *Konvexität*, ansonsten siehe [35] oder [43].

also

$$\left[\begin{array}{c|c|c} & & -1 \\ X^T & 0 & \vdots \\ & & -1 \\ \hline & & 1 \\ 0 & -X & \vdots \\ & & 1 \end{array}\right] \left[\begin{array}{c} p^* \\ q^* \\ v \end{array}\right] \geq 0. \tag{3.8}$$

Und das sieht doch jetzt schon ziemlich stark nach einer Nebenbedingungsmenge für ein Optimierungsproblem aus! Allerdings, ganz fertig sind wir noch nicht, denn es fehlen noch die Nebenbedingungen

$$\sum_{j=1}^{m} p_j^* = 1 \qquad \Rightarrow \qquad \left[\begin{array}{ccc} 1 & \dots & 1 \\ -1 & \dots & -1 \end{array}\right] p^* \geq \left[\begin{array}{c} 1 \\ -1 \end{array}\right]$$

und

$$\sum_{k=1}^{n} q_k^* = 1 \qquad \Rightarrow \qquad \left[\begin{array}{ccc} 1 & \dots & 1 \\ -1 & \dots & -1 \end{array}\right] q^* \geq \left[\begin{array}{c} 1 \\ -1 \end{array}\right].$$

Fassen wir all das zusammen und bringen wir es in die Normalform $Ax \leq b$, indem wir mit -1 multiplizieren, dann erhalten wir schließlich

$$\left[\begin{array}{ccc|ccc|c} & & & & & & 1 \\ & -X^T & & & 0 & & \vdots \\ & & & & & & 1 \\ \hline & & & & & & -1 \\ & 0 & & & X & & \vdots \\ & & & & & & -1 \\ \hline -1 & \dots & -1 & & & & \\ 1 & \dots & 1 & & 0 & & 0 \\ \hline & 0 & & -1 & \dots & -1 & \\ & & & 1 & \dots & 1 & 0 \end{array}\right] \left[\begin{array}{c} p^* \\ q^* \\ v \end{array}\right] \leq \left[\begin{array}{c} 0 \\ \vdots \\ 0 \\ \hline 0 \\ \vdots \\ 0 \\ \hline -1 \\ 1 \\ \hline -1 \\ 1 \end{array}\right]. \tag{3.9}$$

Gut, wir haben also Nebenbedingungen, aber wo bitte ist nun die Zielfunktion? Aber Moment mal – wer hat denn jemals behauptet, wir bräuchten eine? Was uns unsere Theorie sagt ist, daß die optimalen Strategien das *Ungleichungssystem* (3.9) erfüllen müssen und daß alles, was das Ungleichungssystem erfüllt, auch Optimalstrategie ist[80], wir müssen also "nur" eine Lösung von (3.9) finden. Aber das ist ein Problem, mit dem wir uns bereits herumgeschlagen haben, nämlich das Auffinden einer Startecke beim Simplexalgorithmus, also die gute alte Phase I des Zweiphasenalgorithmus! Die beiden `Octave`–Routinen, die diesen Job erledigen, sind in Programm 3.2 und Programm 3.3 angegeben, das erste stellt die Nebenbedingungsmatrix auf, das andere verwendet Phase I und extrahiert die Werte der Variablen.

[80] Die optimale Strategie muß nicht eindeutig sein und tatsächlich gibt es immer unendlich viele Optimalstrategien sobald nur zwei voneinander verschiedene Optimalstrategien existieren!

```
%%  GameStratBed.m (Optmierung)
%%  ---------------------------------------------------
%%  Ungleichungssystem fuer Spielmatrix X von der Form
%%    Ax <= b
%%  Eingabe:
%%    X    Auszahlungsmatrix des Spiels
%%  Ausgabe:
%%    A    Nebenbedingungsmatrix
%%    b    rechte Seite

function [A,b] = GameStratBed( X )
  [m,n] = size( X );

  A = [
       -X', zeros( n,n ), ones( n,1 );
       zeros( m,m ), X, -ones( m,1 );
       -ones( 1,m ), zeros( 1,n ), 0;
       ones( 1,m ), zeros( 1,n ), 0;
       zeros( 1,n ), -ones( 1,m ), 0;
       zeros( 1,n ), ones( 1,m ), 0
       ];

  b = [
       zeros( m+n,1 ); -1; 1; -1; 1
       ];
```

Programm 3.2 `GameStratBed.m`: Bestimmung der Nebenbedingungsmatrix für die optimalen Strategien nach (3.9).

```
%%  GameOptStrat.m (Optmierung)
%%  ------------------------------------------------------
%%  Optimale gemischte Strategien
%%  Eingabe:
%%    X    Auszahlungsmatrix des Spiels
%%  Ausgabe:
%%    p    Strategie fuer Spieler 1
%%    q    Strategie fuer Spieler 2
%%    v    Wert des Spiels

function [p,q,v] = GameOptStrat( X )
  [m,n] = size( X );

  %% Setup und Phase I
  [A,b] = GameStratBed( X );c = zeros( m+n+1,1 );
  T = SimPhase1( A,b,c );

  %% Extrahiere p,q,v
  p = zeros( m,1 ); q = zeros( n,1 ); v = 0;
  for j = 2:m+n+5
    k = T( j,1 );
    if k <= 0
      continue;
    elseif k <= m
      p( k ) = T( j,m+n+3 );
    elseif k <= m+n
      q( k-m ) = T( j,m+n+3 );
    else
      v = T( j,m+n+3 );
    end
  end
```

Programm 3.3 `GameOptStrat.m`: Bestimmung der optimalen Strategie und des Wertes unter Verwendung von Phase I.

Beispiel 3.9 (Stein, Schere, Papier formal) *Zurück zu unserem klassischen Beispiel. Hier können wir die Nebenbedingungen sogar noch explizit angeben, nämlich*

$$\left[\begin{array}{ccc|ccc|c} 0 & 1 & -1 & & & & 1 \\ -1 & 0 & 1 & & & & 1 \\ 1 & -1 & 0 & & & & 1 \\ \hline & & & 0 & 1 & -1 & -1 \\ & & & -1 & 0 & 1 & -1 \\ & & & 1 & -1 & 0 & -1 \\ \hline -1 & -1 & -1 & & & & 0 \\ 1 & 1 & 1 & & & & 0 \\ \hline & & & -1 & -1 & -1 & 0 \\ & & & 1 & 1 & 1 & 0 \end{array}\right] \left[\begin{array}{c} p_1 \\ p_2 \\ p_3 \\ q_1 \\ q_2 \\ q_3 \\ v \end{array}\right] \leq \left[\begin{array}{c} 0 \\ 0 \\ 0 \\ \hline 0 \\ 0 \\ 0 \\ \hline -1 \\ 1 \\ \hline -1 \\ 1 \end{array}\right],$$

und mit Octave erhalten wir das Ergebnis

```
octave> [p,q,v] = GameOptStrat( [ 0 1 -1; -1 0 1; 1 -1 0 ] )
p =

  0.33333
  0.33333
  0.33333

q =

  0.33333
  0.33333
  0.33333

v = 0
```

und damit diejenigen optimalen gemischten Strategien, mit denen wir irgendwie schon gerechnet haben. Ach ja: Fair ist das Spiel auch.

Beispiel 3.10 (Stein, Schere, Papier, Brunnen) *Jetzt wirds ein bißchen interessanter, denn nun haben wir es mit der Auzahlungsmatrix*

$$X = \left[\begin{array}{cccc} 0 & 1 & -1 & -1 \\ -1 & 0 & 1 & -1 \\ 1 & -1 & 0 & 1 \\ 1 & 1 & -1 & 0 \end{array}\right]$$

zu tun, für die wir noch keine Lösung kennen. Fragen wir also unser Orakel:

```
octave> [p,q,v] = GameOptStrat( [ 0 1 -1 -1; -1 0 1 -1;
```

```
> 1 -1 0 1; 1 1 -1 0 ] )
p =

  0.00000
  0.33333
  0.33333
  0.33333

q =

  0.00000
  0.33333
  0.33333
  0.33333

v = 0
```

und eine optimale Strategie besteht darin, den Stein zu vermeiden! Daß das Spiel fair ist, das leuchtet schon eher ein. Wenn wir unsere Matrix X mal ein wenig partitionieren, dann sieht man, warum man den Stein besser weglässt: Die 3×3–Matrix "unten rechts"

$$X = \left[\begin{array}{c|ccc} 0 & 1 & -1 & -1 \\ \hline -1 & 0 & 1 & -1 \\ 1 & -1 & 0 & 1 \\ 1 & 1 & -1 & 0 \end{array}\right]$$

zeigt uns nämlich, daß "Schere, Papier, Brunnen" nichts anderes als ein umbenanntes "Stein, Schere, Papier" ist.

Manchmal sieht man aber auch erst mit Hilfe der Optimalstrategien, ob ein Spiel fair ist oder nicht. Hier noch ein nettes Beispiel.

Beispiel 3.11 (Skin Game) *Das "Skin Game" wird in [21], siehe auch [43] als ein Jahrmarktsspiel*[81] *vorgestellt, das aus der Sicht des Jahrmarktsbesuchers (also des "Kunden") die Auszahlungsmatrix*

$$X = \left[\begin{array}{ccc} -1 & 1 & -2 \\ 1 & -1 & 1 \\ 2 & -1 & 0 \end{array}\right]$$

hat und auf den ersten Blick eigentlich recht fair[82] *aussieht. Nur der Vollständigkeit halber: Ein Spiel heißt* fair *, wenn bei optimaler Spielweise beider Spieler die zu erwartende Auszahlung Null ist. Und das ist das "Skin Game" gerade nicht, denn in der Tat liefert unser Verfahren das folgenden Ergebnis*

[81] Mit Karten, die gleiche Farbe haben können oder nicht ...

[82] Schiefsymmetrische Matrizen, also Matrizen mit $X^T = -X$ führen zu symmetrischen Spielen, die immer fair sind, siehe [43]. Und diese Matrix ist fast schiefsymmetrisch – aber eben nur fast ...

```
octave> X = [ -1 1 -2; 1 -1 1; 2 -1 0 ]; [p,q,v] = GameOptStrat( X )
p =

  0.33333
  0.66667
  0.00000

q =

  0.33333
  0.66667
  0.00000

v = 0
```

Also spielt unser schlauer Kirmesbesucher fleißig die Strategie $\left[\frac{1}{3}, \frac{2}{3}, 0\right]$*, und wenn er am Abend mit leeren Taschen nach Hause kommt, dann war es halt ein wahrscheinlichkeitstheoretischer Ausreißer, gemeinhin auch als "Pech" bezeichnet. Oder?*

Wir müssen eigentlich nur `Octave` zu Rate ziehen, um zu sehen, daß unser Kirmesbesucher bei Verwendung seine scheinbaren Optimalstrategie solide abgezockt wird:

```
octave> p'*X*[0;.6;.4]
ans = -0.20000
```

Bei geeigneter Gegenstrategie verliert er 0.2 Einheiten pro Spiel! Wieso?

Beispiel 3.12 (Skin game, andere Perspektive) *Sehen wir uns doch jetzt einmal die Auszahlungsmatrix*

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

des "Skin[83] *Game" aus der Sicht des Jahrmarktbudenbesitzers an, dann erscheint das Spiel auf den ersten Blick immer noch recht fair, ist es aber nicht, wie wir jetzt einfach nachprüfen können:*

```
octave> A = [ 1 -1 -2; -1 1 1 ; 2 -1 0 ];
octave> [p,q,v] = GameOptStrat( A )
p =

  0.00000
  0.60000
  0.40000
```

[83] Der Name kommt von *to skin*, die Haut abziehen.

```
q =

   0.40000
   0.60000
   0.00000

v = 0.20000
```

Das Spiel hat den Wert $\frac{1}{5}$ aus der Sicht von Spieler 1, und die Optimalstrategie zeigt uns warum: Wenn Spieler 1 die erste Strategie vermeidet, dann fällt der große Verlust schon mal raus, und für Spieler 2 ist dann die zweite Strategie immer besser als die dritte – der Gewinn von Spieler 1 ist da auf keinen Fall größer. Und schon bleibt nur noch das Spiel mit der Auszahlungsmatrix

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

übrig, dem man die Unfairness schon eher ansieht.

Wo liegt das Problem? Ganz einfach: Der Simplexalgorithmus sucht nach einem zulässingen Vektor $[p, q, v] \geq 0$ und wenn es den nicht gibt, dann kommt halt irgendein Blödsinn raus!
Fazit: Um eine sichere Optimalstrategie zu bestimmen, muß man immer die zulässigen Punkte für X und $-X^T$ berechnen.
Wenn jemand das alles nun hochinteressant findet: Mehr zu Spielen und deren Theorie findet sich beispielsweise in [43].

Bin des Professortons nun satt,
will wieder einmal den Teufel spielen.

Goethe, *Urfaust*

4 Innere–Punkte–Methoden

Einerseits ist die Strategie des Simplex–Algorithmus ja ganz logisch: Nachdem das Maximum in einer Ecke angenommen wird, klappern wir diese eben so lange ab, bis wir an der optimalen Ecke angekommen sind. Andererseits hat das auch einen Nachteil, denn auf diese Art und Weise marschiert man ja immer "außen" am zulässigen Bereich entlang, und erreicht so sein Ziel daher immer auf einem Umweg, denn der "direkte Weg" würde normalerweise quer durch das konvexe Polyeder führen. Deshalb interessiert man sich inzwischen für Verfahren, die auf der Suche nach dem Extremum den zulässigen Bereich durchqueren – solche Verfahren bezeichnet man dann als *innere–Punkte–Methoden* und sie zeichnen sich dadurch aus, daß sie ausgehend von einem inneren Punkt des zulässigen Bereichs[84] eine Folge von inneren Punkten konstruieren, die gegen die Extremalecke konvergiert.

Für dieses Kapitel nehmen wir an, daß das lineare Optimierungsproblem stets in der Normalform

$$\min c^T x, \qquad Ax = b, \quad x \geq 0, \qquad x \in \mathbb{R}^n, \tag{4.1}$$

mit $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, die man ja durch Einführung von *Schlupfvariablen* immer erreichen kann. Die Idee der Schlupfvariablen ist wieder einmal sehr einfach und basiert auf der fast trivialen Äquivalenz

$$a^T x \geq b \qquad \Leftrightarrow \qquad a^T x = b + t, \quad t \geq 0.$$

Damit erhalten wir

$$Ax \geq b, \quad x \geq 0 \qquad \Leftrightarrow \qquad Ax = b + y, \quad x, y \geq 0,$$

oder eben

$$[A - I] \begin{bmatrix} x \\ y \end{bmatrix} = b, \quad \begin{bmatrix} x \\ y \end{bmatrix} \geq 0, \tag{4.2}$$

und die hinzugefügten Variablen y bezeichnet man als Schlupfvariablen. Die Normalform (4.1) ist natürlich nur dann sinnvoll, wenn der Rang von A kleiner als n ist, also wird vernünftigerweise $m \leq n$ sein.

Um die Verfahren herleiten und verstehen zu können, müssen wir jetzt etwas tiefer in die Theorie, genauer in die konvexe Analysis, einsteigen.

[84] Also gerade *keinem* Randpunkt und insbesondere keiner Ecke!

4.1 Dualität

Die erste Beobachtung ist, daß dem *Minimierungsproblem* (4.1) ein *Maximierungsproblem*, das sogenannte *duale Problem*, zugeordnet ist, und zwar so, daß die Optimallösungen der beiden Probleme übereinstimmen.

Definition 4.1 *Das* duale Problem *zu (4.1) ist definiert als die Optimierungsaufgabe*

$$\max\, b^T y, \qquad A^T y \le c, \qquad y \in \mathbb{R}^m. \tag{4.3}$$

Dabei bezeichnen b und c in (4.1) und (4.3) dieselben *Vektoren.*

Bemerkung 4.2 *Mit $x \ge 0$ und $c \ge A^T y$ ist*

$$c^T x \ge \left(A^T y\right)^T x = y^T A x = y^T b,$$

also ist auch

$$\min_{x \in F_x} c^T x =: c^T x^* \ge b^T y^* := \max_{y \in F_y} b^T y,$$

wobei die zulässigen Bereiche

$$F_x = \{x \in \mathbb{R}^n \;:\; Ax = b,\, x \ge 0\} \qquad \textit{und} \qquad F_y = \left\{y \in \mathbb{R}^m \;:\; A^T y \le c\right\}$$

beide *nichtleer sein sollen. Für $x \in F_x$ und $y \in F_y$ bezeichnen wir die Größe*

$$0 \le g(x,y) := c^T x - b^T y$$

als Dualitätslücke[85] *zwischen x und y.*

Satz 4.3 (Starker Dualitätssatz) *Sind $x^* \in F_x \ne \emptyset$ und $y^* \in F_y \ne \emptyset$ Optimallösungen von (4.1) und (4.3), dann ist*

$$c^T x^* = b^T y^*. \tag{4.4}$$

Mit anderen Worten: Bei den Optimallösungen gibt es keine Dualitätslücke.

Bemerkung 4.4 *Eigentlich ist Satz 4.3 nur die "schwache" Version der starken Dualität. Es gilt nämlich außerdem, daß $F_x = \emptyset$ genau dann der Fall ist, wenn entweder $F_y = \emptyset$ oder $b^T y$ auf $F_y \ne \emptyset$ nach oben unbeschränkt ist. Umgekehrt bedeutet auch $F_y = \emptyset$, daß entweder $F_x = \emptyset$ oder $c^T x$ nach unten unbeschränkt ist.*
Diese Bemerkung ist durchaus nützlich: Kann man nämlich dem dualen Problem ansehen, daß es keine zulässigen Punkte enthält, dann weiß man auch, daß man sich mit dem primalen Problem gar nicht abzugeben braucht, denn eine Optimallösung kann es ja nicht geben.

[85] *"Duality gap".*

Beweis von Satz 4.3: Sei die Optimallösung x^* eine *nichtentartete*[86] Ecke von F_x, d.h., es gibt

$$J \subset \{1, \ldots, n\}, \qquad \#J = m, \qquad K := \{1, \ldots, n\} \setminus J,$$

so daß die Matrix A'_J, die von den durch J indizierten *Spalten*[87] von A gebildet wird, invertierbar ist und daß

$$A'_J x^*_J = b \quad \Rightarrow \quad x^*_J = (A'_J)^{-1} b, \qquad x^*_K = 0$$

Da, für beliebiges $x \in F_x$

$$Ax = A'_J\, x_J + A'_K\, x_K = [A'_J\, A'_K] \begin{bmatrix} x_J \\ x_K \end{bmatrix} = b \qquad \Longrightarrow \qquad x_J = (A'_J)^{-1} (b - A'_K\, x_K),$$

ist

$$\begin{aligned} c^T x &= c_J^T x_J + c_K^T x_K = c_J^T \underbrace{(A'_J)^{-1} b}_{=x^*_J} + \underbrace{\left(c_K - (A'_K)^T (A'_J)^{-T} c_J\right)^T}_{d_K^T} x_K \\ &= c_J^T x^*_J + c_K^T \underbrace{x^*_K}_{=0} + d_K^T\, x_K = c^T x^* + d_K^T\, x_K, \end{aligned}$$

und also ist $d_K^T\, x_K = c^T x - c^T x^* \geq 0$ und da das für alle $x_K \geq 0$ gelten muß, ist auch $d_K \geq 0$.

Nun setzen wir

$$y^* = (A'_J)^{-T} c_J$$

und erhalten, daß

$$A^T y^* = \begin{bmatrix} (A'_J)^T \\ (A'_K)^T \end{bmatrix} (A'_J)^{-T} c_J = \begin{bmatrix} c_J \\ (A'_K)^T (A'_J)^{-T} c_J \end{bmatrix} = \begin{bmatrix} c_J \\ c_K - d_K \end{bmatrix} \leq \begin{bmatrix} c_J \\ c_K \end{bmatrix} = c,$$

also ist y^* eine zulässige Ecke[88], und es gilt

$$b^T y^* = b^T (A'_J)^{-T} c_J = c_J^T \underbrace{(A'_J)^{-1} b}_{=x^*_J} = c_J^T\, x^*_J = c^T x^*$$

was zu beweisen war. □

Übung 4.1 ("Kolmogoroff[89]–Kriterium")

[86] Durch beliebig kleine Störungen von A oder auch von b kann man immer erreichen, daß *alle* Ecken nichtentartet sind, was den Beweis nur um ein ε–Argument bereichern würde: Man zeigt, daß die Aussage für alle hinreichend kleinen $\varepsilon > 0$ gültig ist und lässt dann $\varepsilon \to 0$ gehen ...

[87] Um den Unterschied zur bisherigen Notation, wo A_J ja eine *Zeilenauswahl* darstellte, klarzumachen wird "′" verwendet.

[88] In den "ersten" m Komponenten herrscht ja Gleichheit!

[89] Andrey Nikolaevich Kolmogoroff (oder "Kolmogorov"), 1903–1987, trug wesentlich zu den Grundlagen der Wahrscheinlichkeitstheorie, aber auch zu Approximationstheorie, Topologie, Funktionalanalysis, Geometrie und so einigem mehr bei. Mit anderen Worten: einer der ganz, ganz großen Mathematiker des 20. Jahrhunderts!

Zeigen Sie, daß x^* genau dann Optimallösung von (4.1) ist, wenn

$$0 \geq (Ay - c)^T (x - x^*), \qquad x \in F_x,\, y \in F_y$$

gilt.
Hinweis: x^* ist offensichtlich Optimallösung genau dann, wenn $0 \geq c^T x^* - c^T x$ für alle $x \in F_x$. Formen Sie diesen Ausdruck geeignet um. ♢

Als nächstes eine Aussage, in der die Frage, wann $F_x \neq \emptyset$ ist über die dualen Nebenbedingungen charakterisiert wird, siehe [49, Theorem 1.9, S. 17] oder [45, A2.1.4, S. 40]

Satz 4.5 ("Farkas–Lemma") *Für $A \in \mathbb{R}^{m\times n}$ und $b \in \mathbb{R}^m$ ist $F_x \neq \emptyset$ genau dann, wenn*

$$A^T y \geq 0 \qquad \Longrightarrow \qquad b^T y \geq 0. \tag{4.5}$$

Beweis: Die Richtung "⇒" ist einfach: Ist $x \in F_x$, also $Ax = b$ sowie $x \geq 0$, und ist $A^T y \geq 0$, dann ist

$$b^T y = (Ax)^T y = \underbrace{x^T}_{\geq 0} \underbrace{A^T y}_{\geq 0} \geq 0.$$

Für die Umkehrung "⇐" bemerken wir zuerst, daß $F_x = \emptyset$ bedeutet, daß der Vektor $b \in \mathbb{R}^m$ nicht zu dem konvexen Kegel

$$K_A := A\,\mathbb{R}^n_+ = \{Ax \;:\; x \geq 0\} \tag{4.6}$$

gehört. Dieser konvexe Kegel ist der Durchschnitt seiner berandenden Halbräume, das heißt,

$$K_A = \bigcap_{j=1}^{N} \left\{ y \in \mathbb{R}^m \;:\; v_j^T y \geq 0 \right\},$$

wobei die v_j, $j = 1, \ldots, N$, die *Normalenvektoren* auf die *linearen*[90] Halbräume sind, die auf einer Seite der von $m-1$ Spaltenvektoren von A definierten[91] Hyperebenen liegen, siehe Übung 4.2. Sei also jetzt (4.5) erfüllt und nehmen wir an, es gäbe kein $x \geq 0$, so daß $Ax = b$ ist, da heißt $b \notin K_A$. Nach unserer obigen Bemerkung muß es also ein $v \in \{v_1, \ldots, v_N\}$ geben, so daß $v^T b < 0$ ist, woraus mit (4.5) (rückwärts gelesen) folgt, daß auch $A^T v \not\geq 0$ ist, daß es also mindestens ein $k \in \{1, \ldots, n\}$ gibt, so daß $\left(A^T v\right)_k < 0$ ist. Setzen wir nun $x = e_k$, dann ist $Ax \in K_A$ und somit

$$0 \leq v^T Ax = \left(A^T v\right)^T x = e_k^T \left(A^T v\right) = \left(A^T v\right)_k < 0,$$

was ein offensichtlicher Widerspruch ist. Also muss $b \in K_A$ gelten. □

[90] Daher auch, und das ist wichtig, die Beschreibung der Halbräume als $v_j^T y \geq 0$, wir haben hier keinen affinen oder "Verschiebungs"–Anteil!

[91] Aber nicht jede Auswahl von $m-1$ Spalten liefert natürlich eine Randhyperebene, manche dieser Halbräume könnten ja durchaus redundant sein.

Übung 4.2 Zeigen Sie: zu jedem konvexen Kegel K_A, $A \in \mathbb{R}^{m\times n}$, wie in (4.6) gibt es Vektoren $v_1, \ldots, v_N$, so daß

$$K_A = \bigcap_{j=1}^{N} H_j, \qquad H_j = \left\{ y \ : \ v_j^T y \geq 0 \right\}.$$

♢

4.2 Kegel und Multiplikatoren

Das wesentliche Resultat dieses Kapitels wird eine Variante der Lagrangeschen[92] Multiplikatorenformel, die ja bekanntlich Aussagen über Extrema unter Nebenbedingungen macht, was ja im Kontext der Optimierung nicht zu abwegig erscheinen sollte. Um diese Resultate angeben und beweisen zu können brauchen wir aber ein bißchen mehr Terminologie.

Definition 4.6 *1. Eine Menge $K \subset \mathbb{R}^n$ heißt* Kegel *mit Spitze x, wenn*

$$y \in K \qquad \Longrightarrow \qquad x + \alpha\,(y - x) \in K, \quad \alpha \in \mathbb{R}_+.$$

2. Sei $M \subset \mathbb{R}^n$ und $x \in M$. Der abgeschlossene Tangentialkegel *an M in x ist definiert als*

$$T(M,x) := \bigcap_{\varepsilon>0} \overline{\{(y-x)\ \mathbb{R}_+ \ : \ y \in M,\ \|y-x\| \leq \varepsilon\}}$$

3. Für $M \subset \mathbb{R}^n$ heißt

$$M' = \left\{ y \in \mathbb{R}^n \ : \ y^T M \geq 0 \right\}$$

positiver Normalenkegel *an M*.

Übung 4.3 Zeigen Sie:

1. $T(M,x)$ und M' sind Kegel. Was ist die Spitze der beiden Kegel?

2. Für $y \in \mathbb{R}^n$ gilt $y \in T(M,x)$ genau dann, wenn es Folgen $x_k \in M$ und $\alpha_k \in \mathbb{R}_+$ gibt, so daß

$$\lim_{k\to\infty} x_k = x \qquad \text{und} \qquad \lim_{k\to\infty} \alpha_k\,(x_k - x) = y.$$

♢

[92] Joseph–Louis Lagrange, 1736–1813, italienisch–französischer Mathematiker (geboren in Turin, das zu dieser Zeit aber zu Sardinien–Piemont gehörte; der Name seines Vaters ist Giuseppe Francesco Lodovico Lagrangia), Beiträge zur Wahrscheinlichkeitstheorie, Variationsrechnung und mathematischen Physik.

Beispiel 4.7 *Sehen wir uns doch einmal den Tangentialkegel von*

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

für vorgegebene Matrix $A \in \mathbb{R}^{m\times n}$ und $b \in \mathbb{R}^m$ an. Eine ε–Umgebung von x in F_x besteht also aus allen Punkten $z \in \mathbb{R}^n$, so daß $Az = b$, $z \geq 0$ und $\|z - x\| \leq \varepsilon$. Der Vektor $y := z - x$ hat dann die Eigenschaft, daß

$$Ay = Az - Ax = b - b = 0$$

und

$$y_j = \underbrace{z_j}_{\geq 0} - x_j \in \begin{cases} \mathbb{R}, & x_j > 0, \\ \mathbb{R}_+, & x_j = 0, \end{cases} \qquad j = 1, \ldots, n.$$

Der Tangentialkegel ist also

$$T(F_x, x) = \{y \in \mathbb{R}^n \ : \ Ay = 0,\ y_j \geq 0 \text{ falls } x_j = 0\}.$$

Bemerkung 4.8

1. *Die Idee des Tangentialkegels besteht darin, sich "beliebig kleine" Umgebungen des Punktes x in M anzusehen und die Strahlen in alle Richtungen zusammenzufassen, in die man so gehen kann. Den Abschluss verwendet man, um wirklich "auf Nummer sicher" zu gehen – es hat eher mit komplexeren Bereichen als unseren konvexen Polyedern zu tun.*

2. *Ist $x \in M^\circ$ ein* innerer *Punkt von M, so ist offensichtlich $T(M, x) = \mathbb{R}^n$.*

3. *Bei zweidimensionalen konvexen Polyedern, wo es ja nur drei Typen von Punkten, nämlich innere Punkte, Randpunkte und Eckpunkte, gibt, kann man sich die Tangentialkegel einfach vorstellen.*

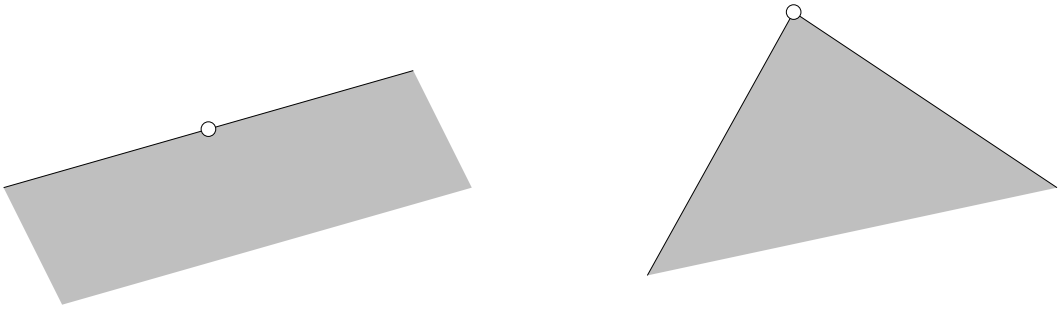


Abbildung 4.1: Die Tangentialkegel an einer Kante und einer Ecke eines konvexen Polyeders.

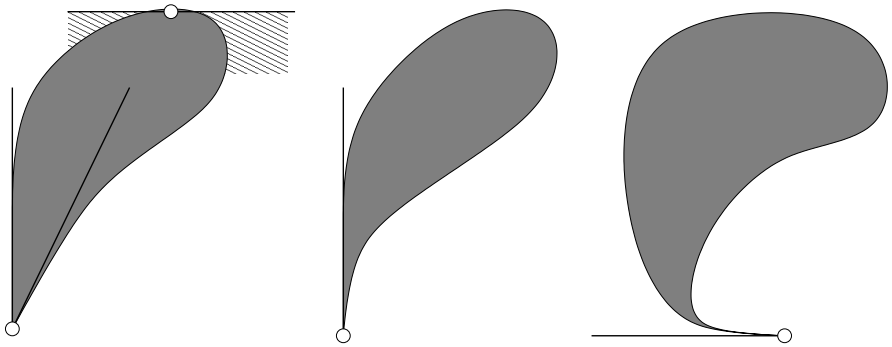


Abbildung 4.2: Beispiele für etwas allgemeinere Tangentialkegel zu Rand- und Eckpunkten. Ist der Rand des Bereichs an einer Stelle differenzierbar, dann besteht der Tangentialkegel aus allen Punkten auf "einer Seite" der Tangente. An Punkten mit Singularitäten der Randkurve kann der Tangentialkegel sogar zu einer Gerade degenerieren (mitte). Im dritten Beispiel erhält man den Tangentialkegel wirklich nur als Abschluß des Grenzwerts.

Definition 4.9 *Sei $f \in C^1(\mathbb{R}^n)$ eine stetig differenzierbare*[93] *Funktion, dann ist der* Gradient $\nabla f : \mathbb{R}^n \to \mathbb{R}^n$ *definiert als*

$$\nabla f = \left[\frac{\partial f}{\partial x_j} \ : \ j = 1, \ldots, n \right].$$

Die Richtungsableitung *von f in Richtung $y \in \mathbb{R}^n$ ist dann der Ausdruck*

$$D_y f = y^T \nabla f.$$

Mit der so bereitgestellten Notation können wir jetzt die notwendige Bedingung für die Existenz eines lokalen Minimums auf "etwas andere" Art hinschreiben.

Proposition 4.10 *Ist der Punkt $x \in M \subset \mathbb{R}^n$ ein lokales Minimum der Funktion $f \in C^1(M)$, dann gilt*

$$\nabla f(x) \in T(M, x)'. \tag{4.7}$$

Bemerkung 4.11 *Ist $T(M, x) = \mathbb{R}^n$, das heißt, der Punkt x ist ein "innerer Punkt", man kann von x aus innerhalb von M in alle Richtungen gehen, dann ist $T(M, x)' = \{0\}$ und wir erhalten in (4.7) das klassische Kriterium aus der Analysis für das Vorliegen eines Extremums: Die Ableitung, also in diesem Fall der Gradient, verschwindet!*

[93] Mit den Details totaler Differenzierbarkeit, stetiger Differenzierbarkeit, partieller Differenzierbarkeit und stetiger partieller Differenzierbarkeit wollen wir uns in dieser Vorlesung nicht herumschlagen. Trotzdem ist klar, daß sie bei hinreichender Allgemeinheit eine wichtige Rolle spielen würden.

Beweis von Proposition 4.10: Sei $y \in T(M,x)$. Nach Übung 4.3[94] gibt es Folgen $M \ni x_k \to x$ und $\alpha_k \in \mathbb{R}_+$, $k \in \mathbb{N}$, mit $\alpha_k (x_k - x) \to y$. Das heißt insbesondere, daß $\alpha_k \to \infty$ oder $\alpha_k^{-1} \to 0$.
Da $x_k \to x$ und da x ein lokales Minimum ist, gilt also, für hinreichend großes $k \in \mathbb{N}$, daß

$$\begin{aligned} 0 &\leq f(x_k) - f(x) = f(x + (x_k - x)) - f(x) = f\left(x + \alpha_k^{-1}(\alpha_k(x_k - x))\right) - f(x) \\ &=: f\left(x + \alpha_k^{-1} y_k\right) - f(x), \end{aligned}$$

also auch

$$0 \leq \frac{f\left(x + \alpha_k^{-1} y_k\right) - f(x)}{\alpha_k^{-1}} \to D_y f = y^T \nabla f(x),$$

da $y_k \to y$ und da f stetig differenzierbar ist. Wir haben also gezeigt, daß unter der Annahme eines Minimums

$$y \in T(M,x) \qquad \Longrightarrow \qquad y^T \nabla f(x) \geq 0 \qquad \Longrightarrow \qquad \nabla f(x)\, T(M,x) \geq 0$$

gilt, was nichts anderes als (4.7) ist. □

So noch ein klein wenig Notation und wir können uns an unseren "Hauptsatz" dieses Kapitels machen, in dem wir einen sehr allgemeinen zulässigen Bereich, nämlich

$$\Omega = \Omega(g,h) = \{x \in \mathbb{R}^n \ : \ g(x) = 0,\ h(x) \geq 0\}$$

betrachten, wobei $g : \mathbb{R}^n \to \mathbb{R}^p$ und $h : \mathbb{R}^n \to \mathbb{R}^q$ zumindest mal *differenzierbare* Funktionen[95] sein sollen.

Definition 4.12 (Aktive Nebenbedingungen und linearisierende Kegel)

1. *Für $x \in \Omega$ bezeichnet*

$$J(x) = \{1 \leq j \leq q \ : \ h_j(x) = 0\} \subseteq \{1, \ldots, q\}$$

die aktiven Nebenbedingungen*.*

2. *Der* linearisierende Kegel *der Nebenbedingungen g, h ist definiert als*

$$L(x) = L(x,g,h) = \left\{y \in \mathbb{R}^n \ : \ y^T \nabla g_j = 0,\ y^T \nabla h_k \geq 0,\ j = 1, \ldots, p,\ k \in J(x)\right\}.$$

Übung 4.4 Zeigen Sie, daß $T(\Omega, x) \subseteq L(x)$. (Hinweis: Taylorformel) ♢

Satz 4.13 *Es sei $x \in \Omega$ eine Minimalstelle von $f \in C^1(\Omega)$ und es sei*

$$L(x)' = T(\Omega, x)'. \tag{4.8}$$

Dann existieren Vektoren $\lambda \in \mathbb{R}^p$ und $\mu \in \mathbb{R}^q_+$, so daß

$$\nabla f(x) - \nabla g(x)\, \lambda - \nabla h(x)\, \mu = 0 \tag{4.9}$$

$$\mu^T h(x) = 0 \tag{4.10}$$

[94] Ja, jetzt wird's gemein! Wer volle Gewissheit haben will, muß selbst was tun!
[95] Wer weiss noch, wie eine differenzierbare Funktion von $\mathbb{R}^n$ nach $\mathbb{R}^p$ wirklich definiert ist?

Hierbei ist für eine Funktion $g = (g_1, \ldots, g_p)$ der Gradient als

$$\nabla g = [\nabla g_1 \cdots \nabla g_p] = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \cdots & \frac{\partial g_p}{\partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_1}{\partial x_n} & \cdots & \frac{\partial g_p}{\partial x_n} \end{bmatrix}$$

definiert. Das sieht anders aus, als man es zumeist aus Analysisbüchern kennt, ist dafür aber konsistent mit unserer Notation hier, in der die Gradienten skalarer Funktionen als *Spaltenvektoren* geschrieben werden. Bevor wir uns an den (gar nicht mal so schweren) Beweis dieses Satzes machen, erst einmal ein paar Bemerkungen und Konsequenzen daraus.

Bemerkung 4.14 (Multiplikatoren) *1. Die geometrische Bedingung (4.8) an die Randbedingungen, die die Menge Ω festlegen (und nicht unbedingt an die Menge Ω selbst!) wird laut [45] als* Bedingung von Guingard *bezeichnet, die dieser in [23] angegeben haben soll*[96]*. Was das allerdings für Bereiche sind, die (4.8)* nicht *erfüllen – wer weiß.*

2. *Da $X \subseteq Y \Rightarrow Y' \subseteq X'$, folgt mittels Übung 4.4, daß $L(x)' \subseteq T(\Omega, x)'$; was man also eigentlich in (4.8) fordert, ist daß $L(x)' \supseteq T(\Omega, x)'$. Der einfachste Fall läge sicherlich vor, wenn $L(x) = T(\Omega, x)$, aber dem ist halt leider nicht immer so, siehe Übung 4.5.*

3. *Die Vektoren λ und μ spielen die Rolle der wohlbekannten* Lagrange–Multiplikatoren*, weswegen wir sie auch in Zukunft als* Multiplikatoren *bezeichnen werden. Allerdings hat die spezielle Struktur der Nebenbedingungen auch Auswirkungen auf die Multiplikatoren: man kann λ als Vektor mit* nichtnegativen *Einträgen wählen!*

4. *In Optimiererkreisen werden die Bedingungen (4.9) und (4.10) auch als* Kuhn–Tucker–Bedingungen *bezeichnet.*

5. *Da $\mu \geq 0$ und $h(x) \geq 0$ – schließlich ist ja $x \in \Omega$ – bedeutet (4.10), daß die Träger der beiden Vektoren* disjunkt *sein müssen, das heißt $\mu_j > 0 \Rightarrow h_j(x) = 0$ und $h_j(x) > 0 \Rightarrow \mu_j = 0$.*

Übung 4.5 Bestimmen Sie für $n = 2$, $p = 0$, $q = 3$ und

$$h(x) = \begin{bmatrix} (1-x)^3 - y \\ x \\ y \end{bmatrix}$$

die Kegel $L(e_1)$ und $T(\Omega, e_1)$, wobei $e_1 = [1, 0]^T$ natürlich der erste Einheitsvektor ist. ♢

[96] Es ist ja nur Hörensagen, und es gibt durchaus genug Beispiele, die nahelegen, nicht alles zu glauben, was in Büchern steht.

Korollar 4.15 (Der lineare Fall) *Es sei $A \in \mathbb{R}^{m\times n}$ und $x \in F_x \neq \emptyset$ eine Minimalstelle von $f(x) = c^T x$. Dann gibt es Vektoren $\lambda \in \mathbb{R}^m$ und $\mu \in \mathbb{R}^n_+$, so daß*

$$\begin{aligned} c - A^T\lambda - \mu &= 0 && (4.11)\\ \mu^T x &= 0. && (4.12) \end{aligned}$$

Beweis: Wir betrachten also den Spezialfall, daß $f(x) = c^T x$, $g(x) = Ax - b$ und $h(x) = x$; damit sind die Gradienten

$$\nabla f = c, \qquad \nabla g = A^T, \qquad \nabla h = I_n \tag{4.13}$$

allesamt konstante Funktionen.
Zuerst weisen wir nach, daß F_x die Bedingung (4.8) erfüllt. Da

$$L(x) = \{z \in \mathbb{R}^n \ : \ Az = 0,\ z_j \geq 0, j \in J(x)\} = T\left(F_x, x\right), \tag{4.14}$$

siehe Beispiel 4.7, ist natürlich auch $L(x)' = T\left(F_x, x\right)'$. Damit können wir Satz 4.13 anwenden und erhalten (4.11) und (4.12), indem wir (4.13) in (4.9) und (4.10) einsetzen. □

Bemerkung 4.16 *Die einfache Beobachtung (4.14) kann man auch als die Tatsache interpretieren, daß die linearisierenden Nebenbedingungen zu linearen Nebenbedingung wieder die linearen Nebenbedingungen sind, was nun wieder nicht allzu überraschend klingt.*

Nun schließlich noch zum Beweis von Satz 4.13, der, im Gegensatz zu den Sandard–Beweisen für die Lagrange–Multiplikatoren[97] sogar wohltuend kurz und einfach ist.
Beweis von Satz 4.13: Sei also $x \in \Omega$ eine Minimalstelle von f. Nach Proposition 4.10 und der Annahme (4.8) heißt das, daß

$$\nabla f(x) \in T\left(\Omega, x\right)' = L\left(x\right)', \qquad \Longrightarrow \qquad z^T \nabla f(x) \geq 0, \quad z \in L(x).$$

Definieren wir $A \in \mathbb{R}^m$, $m := 2p + \#J(x)$ als

$$\begin{aligned} A \ &:= \ A(x) = [\nabla g(x), -\nabla g(x), [\nabla h_j(x) \ : \ j \in J(x)]] \\ &= \ \begin{bmatrix} \frac{\partial g_1(x)}{\partial x_1} & \cdots & \frac{\partial g_p(x)}{\partial x_1} & -\frac{\partial g_1(x)}{\partial x_1} & \cdots & -\frac{\partial g_p(x)}{\partial x_1} & \frac{\partial h_k(x)}{\partial x_1} & \cdots & \frac{\partial h_{k'}(x)}{\partial x_1} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_1(x)}{\partial x_n} & \cdots & \frac{\partial g_p(x)}{\partial x_n} & -\frac{\partial g_1(x)}{\partial x_n} & \cdots & -\frac{\partial g_p(x)}{\partial x_n} & \frac{\partial h_k(x)}{\partial x_n} & \cdots & \frac{\partial h_{k'}(x)}{\partial x_n} \end{bmatrix}, \end{aligned}$$

dann ist[98] nach der Definition von $L(x)$

$$z \in L(x) \qquad \Longleftrightarrow \qquad A^T z \geq 0,$$

und wir können $\nabla f(x) \in L(x)'$ umschreiben in

$$A^T z \geq 0 \qquad \Longrightarrow \qquad z^T \nabla f(x) \geq 0. \tag{4.15}$$

[97] Siehe z.B. [25, 174.1, S. 320].

[98] Unter Verwendung der tiefliegenden Erkenntnis, daß $x = 0$ genau dann, wenn $[x, -x]^T \geq 0$.

Nach dem Farkas–Lemma, Satz 4.5, heißt dies aber, daß die Menge

$$\{\gamma \in \mathbb{R}^m \ : \ A\gamma = \nabla f(x),\, \gamma \geq 0\}$$

nichtleer ist und somit gibt es ein

$$\gamma = \left[\gamma_j^{(1)}, \gamma_j^{(2)}, \gamma_k^{(3)} \ : \ j = 1, \ldots, p,\, k \in J(x)\right] \in \mathbb{R}_+^{2p+\#J(x)},$$

so daß

$$\begin{aligned} \nabla f(x) &= \sum_{j=1}^{p} \gamma_j^{(1)} \nabla g_j(x) - \sum_{j=1}^{p} \gamma_j^{(2)} \nabla g_j(x) + \sum_{j \in J(x)} \gamma_j^{(3)} \nabla h_j(x) \\ &= \sum_{j=1}^{p} \underbrace{\left(\gamma_j^{(1)} - \gamma_j^{(2)}\right)}_{=:\lambda_j} \nabla g_j(x) + \sum_{j \in J(x)} \underbrace{\gamma_j^{(3)}}_{=:\mu_j} \nabla h_j(x), \end{aligned}$$

und mit $\mu_j = 0$, $j \in \{1, \ldots, q\} \setminus J(x)$, ergibt sich (4.9). Die andere Folgerung, (4.10), sieht man sofort, wenn man das innere Produkt ausschreibt und sich an die Definition von $J(x)$ erinnert:

$$\mu^T h(x) = \sum_{j \in J(x)} \mu_j \underbrace{h_j(x)}_{=0} + \sum_{j \notin J(x)} \underbrace{\mu_j}_{=0} h_j(x) = 0.$$

□

4.3 Affine Skalierung

Nach diesen Vorarbeiten können wir nun endlich unser erstes Verfahren herleiten, und zwar die *affine Skalierung* von Barnes [3], die laut [45] auf ein wesentlich älteres Verfahren von Dikin [15] zurückgeht[99]. Die Idee besteht darin, das Optimierungsproblem (4.1) nicht *global* auf F_x, sondern nur in einer Umgebung eines Punktes $x^{(r)}$ zu lösen, das lokale Optimum als $x^{(r+1)}$ zu wählen und sich hoffentlich auf diese Art und Weise dem *globalen* Optimum hinreichend schnell hinreichend genau anzunähern.
Allerdings benötigen wir dazu eine Voraussetzung:

> *Das primale wie auch das duale Problem sollen keine* entarteten *Ecken besitzen, d.h.,*
>
> $$\#\{j \ : \ x_j > 0\} \geq m, \qquad x \in F_x, \tag{4.16}$$
>
> *und*
>
> $$\#\left\{j \ : \ \left(A^T y\right)_j = c_j\right\} \leq m, \qquad y \in F_y, \tag{4.17}$$
>
> *und A soll Rang m haben.*

[99] Die Arbeit [15] ist in den "Doklady" erschienen, wo Resultate normalerweise nur *vorgestellt*, nicht aber notwendigerweise bewiesen werden.

Auch diese Forderungen sind nur für "singuläre" Probleme nicht erfüllt und können durch Störungen oder "positive" Rundungseffekte erreicht werden.

Zur Konstruktion einer Folge von inneren Punkten wählt man einen Parameter $0 < \rho < 1$ und betrachtet zu $x \in F_x$, $x > 0$, das Hilfsproblem

$$\min_y c^T y, \qquad Ay = b, \quad \sum_{j=1}^{n} \frac{(y_j - x_j)^2}{x_j^2} \leq \rho^2, \tag{4.18}$$

wobei die nichtlineare Nebenbedingung dafür sorgt, daß wir im Inneren von F_x bleiben, also auch $y > 0$ ist. Wäre nämlich $y_k \leq 0$ für ein $k \in \{1, \ldots, n\}$, dann wäre

$$\sum_{j=1}^{n} \frac{(y_j - x_j)^2}{x_j^2} \geq \frac{(y_k - x_k)^2}{x_k^2} \geq 1 > \rho^2.$$

Wir definieren nun die Menge $\Omega \subset \mathbb{R}^n$ durch die Randbedingungen[100]

$$0 = g(y) = Ay - b, \qquad 0 \leq h(y) = \rho^2 - \sum_{j=1}^{n} \frac{(y_j - x_j)^2}{x_j^2}.$$

Damit ist Ω der Schnitt der Hyperebene $Ay = b$ mit dem Ellipsoid mit Mittelpunkt x und Halbachsen ρx_j, $j = 1, \ldots, n$, siehe Abb. 4.3.

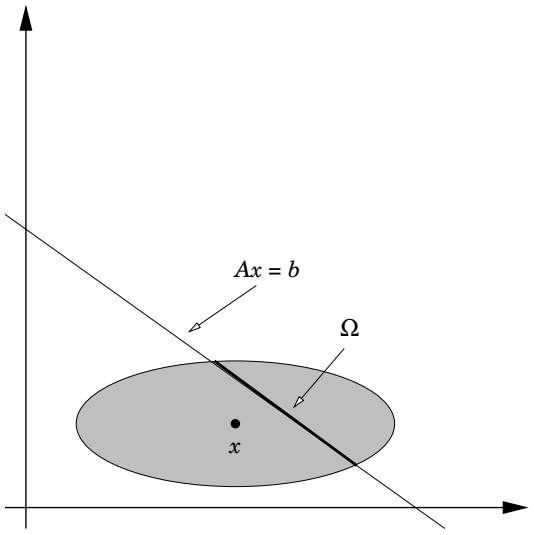


Abbildung 4.3: Der zulässige Bereich Ω als Schnitt der Ellipse um x mit Halbachsen $\rho\, x$ und der Hyperebene $Ax = b$.

[100] Nicht vergessen: wir halten x fest und optimieren bezüglich y!

Wir wollen jetzt Satz 4.13 anwenden, um das lokale Minimum von $c^T y$ auf Ω zu finden[101], wozu wir erst einmal die Guingard–Bedingung (4.8) nachweisen müssen. Nun hat Ω für y mit $h(y) > 0$ den Tangentialkegel

$$T(\Omega, y) = \left\{ z \ : \ A^T z = 0 \right\},$$

ist hingegen $h(y) = 0$, so ist für jedes $y' \in \Omega$

$$\begin{aligned} 0 \ &\leq \ h(y') = h(y') - \underbrace{h(y)}_{=0} = \int_0^1 \left(D_{y'-y} h\right)(y + t(y'-y)) \, dt \\ &= \ \int_0^1 (y'-y)^T \nabla h\,(y + t(y'-y))\, dt, \end{aligned}$$

also auch

$$0 \leq \frac{1}{\|y'-y\|} \int_0^1 (y'-y)^T \nabla h\,(y + t(y'-y))\, dt,$$

und für $\|y'-y\| \to 0$ liefert das die Elemente des Tangentialkegels für $h(y) = 0$ als

$$T(\Omega, y) = \left\{ z \ : \ A^T z = 0, \ z^T \nabla h \geq 0 \right\},$$

was, wie man leicht sieht, wieder nichts anderes als $L(y)$ ist. Also ist die Bedingung (4.8) von Satz 4.13 erfüllt und die gesuchte (und ja auch vorhandene) Minimalstelle y^* liefert die Existenz von $\lambda \in \mathbb{R}^m$ und $\mu \in \mathbb{R}_+$, so daß

$$c - A^T\lambda - 2\mu\, X^{-2}(x-y) \ = \ 0, \tag{4.19}$$

$$\mu \left(\rho^2 - (y-x)^T X^{-2} (y-x)\right) \ = \ 0, \tag{4.20}$$

wobei

$$X = \begin{bmatrix} x_1 & & \\ & \ddots & \\ & & x_n \end{bmatrix} \qquad \Longrightarrow \qquad \begin{cases} h(y) = \rho^2 - (y-x)^T X^{-2}(y-x), \\ \nabla h(y) = 2X^{-2}(x-y). \end{cases}$$

Die Lösung $\mu = 0$, die (4.20) so einfach machen würde ist nicht zulässig, denn das ergäbe, eingesetzt in (4.19), daß $A^T\lambda = c$ wäre und somit hätte F_y eine degenerierte Ecke. Also muß $\mu > 0$ sein und wir erhalten aus (4.20) die erste Forderung an y, nämlich, daß

$$\rho^2 = (y-x)^T X^{-2}(y-x) = \left(X^{-1}(y-x)\right)^T \left(X^{-1}(y-x)\right) = \left\|X^{-1}(y-x)\right\|_2^2;$$

anders gesagt: y liegt am *Rand* der Ellipse. Multiplizieren wir außerdem (4.19) von links mit $(x-y)^T$, dann erhalten wir, nach geeigneter Umformung, daß

$$(x-y)^T \left(c - A^T\lambda\right) = 2\mu (x-y)^T X^{-2}(x-y) = 2\mu\rho^2$$

[101] Für etwas mehr als pure mathematische Ästhetik sollte der Satz schon gut sein!

und somit

$$\mu = \frac{1}{2\rho^2}\Big((x-y)^T c - \underbrace{(x-y)^T A^T}_{=b^T-b^T=0}\,\lambda\Big) = \frac{c^T(x-y)}{2\rho^2} = \frac{c^T(x-y)}{2\left\|X^{-1}(y-x)\right\|_2^2}$$

also

$$c^T y = c^T x - 2\rho^2\,\mu < c^T x, \tag{4.21}$$

der Wert der Zielfunktion wird durch die Lösung unseres Hilfsproblems definitiv *verkleinert*.

Bleibt also noch die Bestimmung von y. Dazu multiplizieren wir (4.19) von links mit AX^2 und erhalten, daß

$$0 = AX^2c - AX^2A^T\lambda - 2\mu\,\underbrace{A(x-y)}_{=0} = AX^2c - (AX)\,(AX)^T\,\lambda \tag{4.22}$$

Da A den Maximalrang m hat, ist die Matrix $(AX)(AX)^T$ symmetrisch und positiv definit[102] und demnach invertierbar, was es uns ermöglicht, den Multiplikator λ als

$$\lambda = \left(AX^2A^T\right)^{-1} AX^2c \tag{4.23}$$

zu bestimmen. Damit bekommen wir aber auch μ: Formen wir nämlich

$$2\mu X^{-1}(x-y) = X\left(c - A^T\lambda\right)$$

um und nehmen auf beiden Seiten die euklidische Norm, dann ist

$$2\mu = \frac{\left\|X\left(c-A^T\lambda\right)\right\|_2}{\left\|X^{-1}(y-x)\right\|_2} = \frac{\left\|X\left(c-A^T\lambda\right)\right\|_2}{\rho}$$

Das setzen wir nun alles nochmal in (4.19) ein und erhalten so, daß

$$y = x + \frac{1}{2\mu}\,X^2\left(c-A^T\lambda\right) = x + \rho\,\frac{X^2\left(c-A^T\lambda\right)}{\left\|X\left(c-A^T\lambda\right)\right\|_2}. \tag{4.24}$$

Wer will kann jetzt noch (4.23) einsetzen, aber schöner oder gar übersichtlicher wird dadurch auch nichts mehr.

Übung 4.6 Zeigen Sie, daß aus (4.24) und den übrigen Voraussetzungen $Ay = b$ folgt. ♢

Fassen wir also zusammen.

Algorithmus 4.17 (Affine Skalierung)
Gegeben: $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ *und innerer Punkt* $x^{(0)}$ *mit* $Ax^{(0)} = b$, $x^{(0)} > 0$.

1. *Finde inneren Punkt* $x^{(0)}$ *mit* $Ax^{(0)} = b$, $x^{(0)} > 0$.
2. *Wähle* $\rho \in (0,1)$

[102] Also *strikt* positiv definit!

3. *Für* $r = 0, 1, 2, \ldots$

   (a) *Setze*

$$X = \begin{bmatrix} x_1^{(r)} & & \\ & \ddots & \\ & & x_n^{(r)} \end{bmatrix}.$$

   (b) *Berechne* $B = XA^T$ *und*

$$\lambda = \left(B^T B\right)^{-1} BXc$$

   (c) *Berechne*

$$x^{(r+1)} = x^{(r)} + \rho \frac{X^2 \left(c - A^T\lambda\right)}{\left\|X\left(c - A^T\lambda\right)\right\|_2}.$$

Bemerkung 4.18 *Schreibt man (4.22) als*

$$(AX)(AX)^T\lambda = (AX)Xc$$

um, dann stellt diese Gleichung die sogenannten Normalengleichungen *zum* Least–squares–problem

$$\min_y \left\|Xc - (AX)^T y\right\|_2 = \min_y \left\|X\left(c - A^T y\right)\right\|_2$$

und der Vektor $X\left(c - A^T\lambda\right)$ *stellt das* Residuum *dieses Minimierungsproblems dar; beide Werte, die Lösung wie auch der Fehler, können sehr stabil über ein* QR*–Verfahren bestimmt werden, siehe z.B. [42], vor allem aber [22] oder [26]: Dazu bestimmt man eine orthogonale*[103] *Matrix* $Q \in \mathbb{R}^{m\times m}$*, so daß*

$$QXA^T = \begin{bmatrix} R \\ 0_{n-m,m} \end{bmatrix}, \qquad R = \begin{bmatrix} * & \ldots & * \\ & \ddots & \vdots \\ & & * \end{bmatrix} \in \mathbb{R}^{m\times m},$$

und bestimmt λ *als Lösung des einfachen Dreieckssystems* $R\lambda = Q^T Xc$*. Das führt zu Algorithmus 4.19.*

Algorithmus 4.19 (Affine Skalierung, QR**–Version)**
Gegeben: $A \in \mathbb{R}^{m\times n}$*,* $b \in \mathbb{R}^m$*,* $c \in \mathbb{R}^n$ *und innerer Punkt* $x^{(0)}$ *mit* $Ax^{(0)} = b$*,* $x^{(0)} > 0$*.*

1. *Finde inneren Punkt* $x^{(0)}$ *mit* $Ax^{(0)} = b$*,* $x^{(0)} > 0$*.*

2. *Wähle* $\rho \in (0, 1)$

3. *Für* $r = 0, 1, 2, \ldots$

[103] Zur Erinnerung: eine reelle Matrix $Q \in \mathbb{R}^{n\times n}$ heißt *orthogonal*, wenn $Q^T Q = QQ^T = I$ ist.

(a) Setze

$$X = \begin{bmatrix} x_1^{(r)} & & \\ & \ddots & \\ & & x_n^{(r)} \end{bmatrix}.$$

(b) Berechne $B = AX$ und Q, R als QR–Zerlegung[104] *von B und λ als Lösung von*

$$R\lambda = Q^T X c$$

(c) Setze

$$y = X\left(c - A^T\lambda\right).$$

(d) Berechne

$$x^{(r+1)} = x^{(r)} + \rho\,\frac{Xy}{\|y\|_2}.$$

Man kann nun sogar *beweisen*, daß dieses Verfahren tatsächlich gegen eine Lösung des Optimierungsproblems konvergiert.

Satz 4.20 *Besitzen das lineare Optimierungsproblem und sein duales keine entarteten Ecken, dann konvergiert die Folge*

$$x^{(r+1)} := x^{(r)} + \rho\, X^{(r)}\frac{X^{(r)}\left(c - A^T\lambda^{(r)}\right)}{\|X^{(r)}\left(c - A^T\lambda^{(r)}\right)\|_2}, \qquad X^{(r)} = \begin{bmatrix} x_1^{(r)} & & \\ & \ddots & \\ & & x_n^{(r)} \end{bmatrix} \tag{4.25}$$

für jeden Startwert $x^{(0)} \in F_x^\circ$ gegen eine Optimallösung x^* und es gilt für $r \in \mathbb{N}_0$

$$0 \le c^T x^{(r+1)} - c^T x^* \le \beta_r\left(c^T x^{(r)} - c^T x^*\right), \qquad \beta_r = 1 - \frac{\rho}{\sqrt{n-m+\varepsilon_r}}, \quad \varepsilon_r \to 0. \tag{4.26}$$

Bemerkung 4.21 *Die Abschätzung (4.26) liefert eine Aussage über die* Approximationsordnung *oder* Konvergenzordnung *des Verfahrens: Mit $\beta = \max_r \beta_r$ ist ja*

$$\begin{aligned} c^T x^{(r+1)} - c^T x^* &\le \beta\left(c^T x^{(r)} - c^T x^*\right) \le \beta^2\left(c^T x^{(r-1)} - c^T x^*\right) \le \cdots \\ &\le \beta^{r+1}\left(c^T x^{(0)} - c^T x^*\right), \end{aligned}$$

wir haben es also mit exponentieller Konvergenz, aber sogenannter linearer Konvergenzordnung *zu tun*[105]

Beweis: Der Beweis ist etwas länglich und gliedert sich grob in drei Teile:

[104] Die QR–Zerlegung ist ein Standardbestandteil jeder Software für numerische Lineare Algebra, insbesondere von `Matlab` und `Octave`, siehe auch [2].

[105] Die Terminologie ist da leider nicht so richtig einheitlich.

1. Wir werden zuerst zeigen, daß die Folge $x^{(r)}$ immer gegen einen Grenzwert konvergiert. Dazu brauchen wir die Nichtentartungsbedingungen an F_x und F_y.

2. Dann zeigen wir, daß dieser Grenzwert tatsächlich eine Optimallösung ist, wozu wir ein Dualitätsargument verwenden werden.

3. Schließlich müssen wir noch die Konvergenzordnung (4.26) nachweisen – das ist im wesentlichen eine technische Abschätzung.

Beginnen wir mit der Konvergenz. Da $x^{(r)}$ eine Folge in dem kompakten Polyeder F_x ist, muß zumindest eine Teilfolge konvergieren, oder die Folge einen Häufungspunkt besitzen, den wir einmal x^∞ nennen wollen.
Wegen (4.21) ist für $r \in \mathbb{N}_0$

$$\begin{aligned} c^T x^* &\leq c^T x^{(r+1)} \leq c^T x^{(r)} - 2\rho^2 \mu^{(r)} = c^T x^{(r)} - 2\rho \left\| X^{(r)} \left(c - A^T \lambda^{(r)}\right) \right\|_2 \\ &= c^T x^{(r-1)} - 2\rho \left\| X^{(r-1)} \left(c - A^T \lambda^{(r-1)}\right) \right\|_2 - 2\rho \left\| X^{(r)} \left(c - A^T \lambda^{(r)}\right) \right\|_2 \\ &= c^T x^{(0)} - 2\rho \sum_{j=0}^{r} \left\| X^{(j)} \left(c - A^T \lambda^{(j)}\right) \right\|_2 \end{aligned}$$

und daher

$$\sum_{j=0}^{r} \left\| X^{(j)} \left(c - A^T \lambda^{(j)}\right) \right\|_2 \leq \frac{1}{2\rho} \left(c^T x^{(0)} - c^T x^*\right), \qquad r \in \mathbb{N}_0, \tag{4.27}$$

die Reihe auf der linken Seite konvergiert also und es folgt, daß

$$\lim_{r\to\infty} X^{(r)} \left(c - A^T \lambda^{(r)}\right) = 0. \tag{4.28}$$

Da das duale Problem nichtentartet ist gibt es eine Konstante $C > 0$, die nur vom Optimierungsproblem abhängt[106] und zu jedem r eine Indexmenge J mit $\#J \geq n - m$, so daß

$$\left(c - A^T \lambda^{(r)}\right)_J > C\, 1_J;$$

nach eventuellem Übergang zu einer Teilfolge[107], die gegen x^∞ konvergiert, gilt dies dann für *eine* Indexmenge J unabhängig von r. Wegen (4.28) ist dann

$$\lim_{r\to\infty} X_J^{(r)} = \lim_{r\to\infty} x_J^{(r)} = x_J^\infty = 0,$$

die Folge konvergiert also gegen eine Ecke und, wieder wegen der Nichtentartung von F_x, muß $\#J = n - m$ und

$$\lim_{r\to\infty} x_K^{(r)} = x_K^\infty > 0, \qquad K = \{1, \ldots, n\} \setminus J,$$

[106] Siehe [45, S. 269]; im wesentlichen hat es damit zu tun, daß zu jedem $y \in F_y$ mindestens $n-m$ Komponenten von $c - A^T y$ *strikt* positiv sein müssen und daß F_x ein *kompaktes* Polyeder ist.

[107] Es gibt nur endlich viele solcher Mengen, also muß mindestens eine für *unendlich viele* Werte von r auftreten, und eine solche greifen wir heraus und gehen zur entsprechenden Teilfolge über.

sein. Insbesondere ist der Häufungspunkt x^∞ eine Ecke. Wir müssen aber noch zeigen, daß auch wirklich *die ganze* Folge gegen x^∞ konvergiert. Dazu schauen wir uns nochmals (4.24) an und formen es in

$$x^{(r+1)} - x^{(r)} = \rho\, X^{(r)} \frac{X^{(r)}\left(c - A^T\lambda^{(r)}\right)}{\left\| X^{(r)}\left(c - A^T\lambda\right)^{(r)} \right\|_2} = \rho X^{(r)} y, \qquad \|y\|_\infty \le \|y\|_2 = 1,$$

um, woraus

$$\left|x^{(r)}\right| - \left|x^{(r+1)}\right| \le \rho \left|x^{(r)}\right| \qquad \Longrightarrow \qquad \left|x^{(r+1)}\right| \ge (1-\rho)\left|x^{(r)}\right|$$

folgt, was uns liefert, daß $x_J^{(r)} \to 0$ für *alle* r, nicht nur für die Teilfolge, und damit, daß

$$x^\infty = \lim_{r\to\infty} x^{(r)}.$$

Damit ist also Punkt 1), die Konvergenz, erledigt.

Als nächstes weisen wir nach, daß x^∞ Optimallösung ist. Da die Folge der $x^{(r)}$ konvergiert existiert natürlich auch die Diagonalmatrix

$$X^\infty := \lim_{r\to\infty} X^{(r)}, \qquad X_J^\infty = 0, \quad X_K^\infty > 0$$

und hat offensichtlich Rang $\#K = m$. Damit müssen, zumindest für hinreichend große Werte von r, auch die Matrizen $X^{(r)}$ Rang $\ge m$ haben, womit $AX^{(r)}$ Rang m hat[108] und damit die Matrizen

$$\left(\left(AX^{(r)}\right)\left(AX^{(r)}\right)^T\right)^{-1} A\left(X^{(r)}\right)^2$$

existieren und und, da $AX^\infty = A'_K X_K^\infty$, gegen

$$\left(\left(A'_K X_K^\infty\right)\left(A_K X_K^\infty\right)^T\right)^{-1} A'_K \left(X^\infty\right)^2 =: \left[B'_K, B'_J\right], \qquad B'_J = 0_{m,m-n}$$

konvergieren. Damit existiert

$$\begin{aligned}
\lambda^\infty &= \lim_{r\to\infty} \left(\left(AX^{(r)}\right)\left(AX^{(r)}\right)^T\right)^{-1} A\left(X^{(r)}\right)^2 c \\
&= \left(\left(A'_K X_K^\infty\right)\left(A'_K X_K^\infty\right)^T\right)^{-1} A'_K \left(X_K^\infty\right)^2 c_K \\
&= \left(A'_K X_K^\infty X_K^{\infty T} A_K'^T\right)^{-1} A'_K X_K^{\infty 2} c_K = A_K'^{-T} \left(\operatorname{diag} x_K\right)^{-2} {A'_K}^{-1} A'_K \left(\operatorname{diag} x_K\right)^2 c_K \\
&= {A'_K}^{-T} c_K.
\end{aligned}$$

Somit ist

$$b^T\lambda^\infty = b^T {A'_K}^{-T} c_K = c_K^T \left({A'_K}^{-1} b\right) = c_K^T x_K^\infty = c^T x^\infty$$

[108] Hier verwenden wir nochmal, daß es keine entarteten Ecken gibt! Jeder Rangdefekt würde nämlich eine entartete Ecke liefern.

und nach unserem Dualitätssatz, Satz 4.3 ist $x^* = x^\infty$ eine Optimallösung. Den Nachweise der Zulässigkeit von λ^∞ sparen wir uns hier und verweisen nur auf [45, S. 271] – und damit ist 2) auch schon erledigt.

Für den Beweis der Konvergenzordnung betrachten wir schließlich

$$\begin{aligned} 0 \le c^T x^{(r)} - c^T x^* &= c^T x^{(r)} - \underbrace{b^T}_{=Ax^{(r)}} \lambda^{(r)} + \underbrace{b^T}_{=Ax^*} \lambda^{(r)} - c^T x^* = \left(c - A^T\lambda^{(r)}\right)^T \left(x^{(r)} - x^*\right) \\ &= \left(X^{(r)}\left(c - A^T\lambda^{(r)}\right)\right)^T \left(X^{(r)}\right)^{-1}\left(x^{(r)} - x^*\right) \\ &\le \left\|X^{(r)}\left(c - A^T\lambda^{(r)}\right)\right\|_2 \left\|\left(X^{(r)}\right)^{-1}\left(x^{(r)} - x^*\right)\right\|_2 \\ &\le \frac{1}{\rho}\left(c^T x^{(r)} - c^T x^{(r+1)}\right) \left\|\left(X^{(r)}\right)^{-1}\left(x^{(r)} - x^*\right)\right\|_2 \end{aligned}$$

Da

$$\left\|\left(X^{(r)}\right)^{-1} x^{(r)}\right\|_2^2 = \sum_{j=1}^{n} \frac{\left(x_j^{(r)} - x_j^*\right)^2}{\left(x_j^{(r)}\right)^2} = \sum_{j\in J^*} \frac{\left(x_j^{(r)} - x_j^*\right)^2}{\left(x_j^{(r)}\right)^2} + \sum_{j\in K^*} \frac{\left(x_j^{(r)} - x_j^*\right)^2}{\left(x_j^{(r)}\right)^2},$$

wobei

$$J^* := \left\{j \ : \ x_j^* = 0\right\}, \qquad K^* := \left\{j \ : \ x_j^* \neq 0\right\},$$

und somit

$$\sum_{j\in J^*} \frac{\left(x_j^{(r)} - x_j^*\right)^2}{\left(x_j^{(r)}\right)^2} = \sum_{j\in J^*} \frac{\left(x_j^{(r)}\right)^2}{\left(x_j^{(r)}\right)^2} = \#J^* \le n - m,$$

ist also

$$c^T x^{(r)} - c^T x^* \le \left(c^T x^{(r)} - c^T x^{(r+1)}\right) \underbrace{\frac{1}{\rho}\left(n - m + \sum_{j\in K^*}\left(\frac{x_j^{(r)} - x_j^*}{x_j^{(r)}}\right)^2\right)^{1/2}}_{=\frac{1}{\rho}(n-m+\varepsilon_r)=:\gamma_r}$$

mit $\varepsilon_r \to 0$ wegen der Konvergenz der $x^{(r)} \to x^*$ und weil die positiven Terme im Nenner nach unten beschränkt sind[109]. Also ist

$$\frac{1}{\gamma}\left(c^T x^{(r)} - c^T x^*\right) \le c^T x^{(r)} - c^T x^* + c^T x^* - c^T x^{(r+1)}$$

und somit

$$c^T x^{(r+1)} - c^T x^* \le \underbrace{\left(1 - \frac{1}{\gamma_r}\right)}_{=:\beta_r}\left(c^T x^{(r)} - c^T x^*\right),$$

wie behauptet. □

[109] Ihr Grenzwert ist ja strikt positiv!

4.4 Projektive Skalierung

Ein weiteres innere–Punkte–Verfahren zur linearen Optimierung ist die *projektive Skalierung* von Karmarkar [30], laut dem Titel von [3] in gewissem Sinne sogar die "Vorlage" zur affinen Skalierung. Um das Karmarkar–Verfahren am besten darstellen zu können, gönnen wir uns mal wieder eine neue Normalform, die man auch als *Karmarkar–Normalform*[110] des lineare Optimierungsproblems bezeichnet:

$$\min_x c^T x, \qquad Ax = 0, \quad x \geq 0, \quad 1_n^T x = n, \tag{4.29}$$

mit $A \in \mathbb{R}^{m \times n}$, $m < n$. Hierbei ist wieder $1_n = [1 \ldots 1]^T \in \mathbb{R}^n$. Der zulässige Bereich ist also der Schnitt

$$\underbrace{\{x \ : \ Ax = 0\}}_{=\ker A} \cap \underbrace{(n\,\Delta_n)}_{=:\Sigma_n},$$

wobei Δ_n wieder das Einheitssimplex aus (2.6) ist, siehe Abb. 4.4. Alternativ können wir die

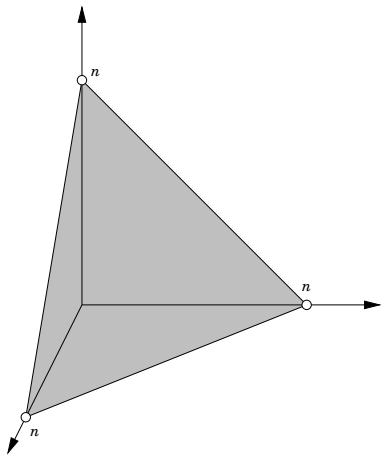


Abbildung 4.4: Das Simplex $n\,\Delta_n$ für $n = 3$. Dieses baryzentrische Standardsimplex wird mit $\ker A$ geschnitten.

Karmarkar–Normalform also als

$$\min_x c^T x, \qquad x \in \ker A \cap \Sigma_n \tag{4.30}$$

schreiben. Man kann jedes Optimierungsproblem auch in die Karmarkar–Normalform (4.29) bringen, indem man das primale und das duale Problem in ein großes Optimierungsproblem kombiniert, siehe [47, Satz 4.7, S. 138–139]. Da aber ein solches Problem nur künstlich aufgebläht ist, verwendet man "in der Praxis" ein modifiziertes Verfahren, das sich direkt auf die

[110] Wen wundert's?

"kompaktere" primale oder duale Form des linearen Optimierungsproblems anwenden läßt, siehe [45, S. 280].

Für die Herleitung machen wir noch weitere Annahmen, nämlich zuerst einmal die Nichtentartungsannahme, daß A vollen Rang hat, außerdem, daß $A1_n = 0$ ist[111] und daß $\min_x c^T x = 0$ ist.

Sei jetzt also wieder $x \in \Sigma_n$, $x > 0$, ein zulässiger innerer Punkt. Um dem Verfahrensnamen gerecht zu werden betrachtet man jetzt die *projektive* Transformation

$$T_x : \Sigma_n \to \Sigma_n, \qquad T_x(y) := \frac{n}{1_n^T X^{-1} y} X^{-1} y$$

mit der Eigenschaft, daß $T_x(x) = 1_n$ ist. Daß mit $y \geq 0$ auch $T_x(y) \geq 0$ gilt, ist offensichtlich, da $X^{-1} \geq 0$ ist. Außerdem ist

$$1_n^T T_x(y) = n \frac{1_n^T X^{-1} y}{1_n^T X^{-1} y} = n,$$

also ist $T_x(y) \in \Sigma_n$ für $y \in \Sigma_n$. Die Inverse zu T_x ist

$$T_x^{-1}(y) = \frac{n}{x^T y} X y.$$

Denn in der Tat ist

$$\frac{n}{x^T T_x(y)} X T_x(y) = \frac{n}{x^T X^{-1} y} X X^{-1} y = \frac{n}{1_n^T y} y = y,$$

solange $y \in \Sigma_n$.

Übung 4.7 Zeigen Sie, daß mit den obigen Definitionen auch $T_x\left(T_x^{-1}(y)\right) = y$ für alle $y \in \Sigma_n$ gilt. ♢

Ersetzen wir jetzt x in der Normalform (4.29) durch $z = T_x^{-1}(y)$, das mit y ja auch ganz Σ_n durchläuft, dann erhalten wir das zu (4.30) äquivalente Problem

$$\min_y c^T T_x^{-1}(y) = n \frac{c^T X y}{x^T y}, \qquad \underbrace{0 = Az = \frac{n}{x^T y} AXy, \quad y \in \Sigma_n}_{y \in \ker AX \cap \Sigma_n}. \tag{4.31}$$

Dabei ist $T_x(x) = 1_n$, unser Ausgangspunkt x wird also in den Schwerpunkt des Simplex transformiert. Unter der Annahme $\min_x c^T x = 0$ können wir den Nenner vernachlässigen und erhalten so das weiter vereinfachte Minimierungsproblem

$$\min_y c^T X y, \qquad y \in \ker AX \cap \Sigma_n. \tag{4.32}$$

Jetzt ist natürlich (4.32) nur eine äquivalente Umformung des Ausgangsproblems (4.29) und es wäre schon eine große Überraschung, wenn das plötzlich einfacher zu lösen wäre als das "Originalproblem". Aus diesem Grund ziehen wir uns wieder auf eine Lokalisierung des Problems zurück. Dazu betrachten wir die Menge

$$K_r := \left\{ y \in \mathbb{R}^n \ : \ 1_n^T y = n, \ \|1_n - y\|_2 \leq r \right\}, \qquad 0 < r.$$

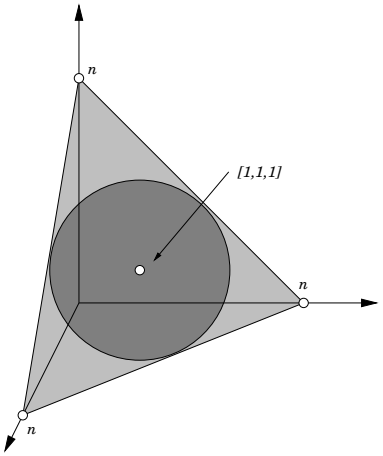


Abbildung 4.5: Der Bereich K_r für den "Extremalwert" $r = \sqrt{\frac{n}{n-1}}$.

siehe Abb. 4.5. Mit $R := \sqrt{\frac{n}{n-1}}$ ist dann

$$K_R \subset \Sigma_n \subset K_{\sqrt{n(n-1)}}, \tag{4.33}$$

denn aus

$$\|1_n - y\|_2^2 = \sum_{j=1}^{n} (1-y_j)^2 = n - 2\underbrace{\sum_{j=1}^{n} y_j}_{=n} + \sum_{j=1}^{n} y_j^2 = -n + \|y\|_2^2$$

folgt für jedes $y \in \Sigma_n$, daß

$$\|1_n - y\|_2^2 \leq \|y\|_1^2 - n = \left(1^T y\right)^2 - n = n^2 - n = n(n-1),$$

was die rechte Inklusion liefert. Wäre außerdem $y \in K_r$ für ein r, und $y_j < 0$ für ein j, dann ist

$$\begin{aligned} r^2 &\geq \|1_n - y\|_2^2 = \underbrace{(1-y_j)^2}_{>1} + \sum_{k\neq j} (1-y_k)^2 > 1 + \sum_{k\neq j} (1-y_k)^2 \\ &\geq 1 + \frac{1}{n-1} \underbrace{\left(\sum_{k\neq j} (1-y_k)\right)^2}_{=(-(1-y_j))^2>1} > 1 + \frac{1}{n-1} = \frac{n}{n-1}, \end{aligned}$$

also bedeutet $y \in K_r \setminus \Sigma_n$, daß $r > \sqrt{\frac{n}{n-1}}$ sein muß, was die linke Inklusion beweist.

Nun können wir wieder unseren Verfahrensparameter $\rho \in (0,1)$ wählen und das *lokale* Optimierungsproblem

$$\min_y c^T X y, \qquad y \in \ker AX \cap K_{\rho R} \tag{4.34}$$

[111] Das heißt, daß der *Schwerpunkt* von Σ_n zulässig ist.

lösen, dessen Lösung wir wieder in geschlossener Form werden angeben können. Wie das geht und warum das funktioniert, das zeigt das folgende Lemma.

Lemma 4.22 *Sei $x \in \ker A \cap \Sigma_n$, $x > 0$, ein nichtoptimaler zulässiger Punkt, das heißt, $c^T x > 0$.*

1. *Die Matrix*
$$B := \begin{bmatrix} AX \\ 1_n^T \end{bmatrix} \in \mathbb{R}^{m+1 \times n}$$
hat Rang $m+1$.

2. *Der Vektor*
$$v = \left(I - B^T \left(BB^T\right)^{-1} B\right) Xc \tag{4.35}$$
erfüllt $v \neq 0$.

3. *Der Vektor*
$$y := 1_n - \rho \, R \, \frac{v}{\|v\|_2}$$
ist die eindeutige Lösung des Optimierungsproblems (4.34).

4. *Es ist*
$$(Xc)^T y \leq \left(1 - \frac{\rho}{n-1}\right) c^T x. \tag{4.36}$$

Bemerkung 4.23 *1. Die "Verbesserung" der Zielfunktion in (4.36) sieht zuerst einmal etwas merkwürdig aus, denn es ist nicht $c^T y$ verringert worden sondern $c^T Xy$. Schreibt man aber x auf der rechten Seite von (4.36) als $x = X1_n$, dann ist wird (4.36) zu*

$$(Xc)^T y \leq \left(1 - \frac{\rho}{n-1}\right) (Xc)^T 1_n, \tag{4.37}$$

und da wir ja das Problem so transformiert haben, daß der Schwerpunkt unser Ausgangspunkt war, haben wir also doch wieder etwas gewonnen.

2. *Auch hier ist die Korrekturrichtung v aus (4.35) wieder die Lösung eines Least–Squares–Problems: Schreiben wir nämlich (4.35) in*
$$v = Xc - B^T \underbrace{\left(BB^T\right)^{-1} BXc}_{=:w} = Xc - B^T w,$$
mit
$$BB^T w = BXc, \qquad \Longrightarrow \qquad \|Xc - Bw\| = \min_u \|Xc - Bu\|,$$
also lässt sich auch die Korrekturrichtung v wieder als Residuum eines Minimierungsproblems interpretieren und (effektiv) bestimmen.

Beweis: Um 1) zu beweisen, nehmen wir an, es gäbe ein $0 \neq z = [\widetilde{z}, z_{m+1}]^T \in \mathbb{R}^{m+1}$, so daß $z^T B = 0$ – dann wäre der Rang ja gerade $\leq m$. In diesem Fall wäre

$$0 = B^T z = \left[XA^T, 1_n\right] \begin{bmatrix} \widetilde{z} \\ z_{m+1} \end{bmatrix} = XA^T\widetilde{z} + z_{m+1}1_n$$

Multiplikation mit A von links liefert, da nach Annahme $A1_n = 0$, daß

$$AXA^T\widetilde{z} = 0$$

sein muß; das aber widerspricht der Tatsache, daß A Rang m hat, denn dann hätte auch die Matrix[112] $A\sqrt{X}$ Rang m und

$$AXA^T = A\sqrt{X}\left(A\sqrt{X}\right)^T$$

wäre strikt positiv definit.

Auch 2) beweisen wir per Widerspruch. Wäre nämlich $v = 0$, dann wäre

$$Xc = B^T\left(BB^T\right)^{-1}BXc =: B^Tz = XA^T\widetilde{z} + z_{m+1}1_n \quad \Longrightarrow \quad c = A^T\widetilde{z} + z_{m+1}X^{-1}1_n$$

Dann wäre aber für die Optimallösung x^*

$$0 = \min_x c^Tx = c^Tx^* = \widetilde{z}^T\underbrace{Ax^*}_{=0} + z_{m+1}\underbrace{1_n^TX^{-1}x^*}_{>0},$$

also $z_{m+1} = 0$ und somit $c = A^T\widetilde{z}$. Dann ist aber auch für unser nichtoptimales x

$$c^Tx = \widetilde{z}^T\underbrace{Ax}_{=0} = 0,$$

im Widerspruch zur Annahme.

Für 3) bemerken wir zuerst, daß

$$Bv = \Big(B - \underbrace{BB^T\left(BB^T\right)^{-1}}_{=I}B\Big)Xc = (B-B)\,Xc = 0$$

und daher ist

$$AX\,(1_n - v) = Ax - AXv = \underbrace{Ax}_{=0} - \begin{bmatrix} I_m & 0 \\ 0 & 0 \end{bmatrix}\underbrace{Bv}_{=0} = 0 \qquad \Longrightarrow \qquad 1_n - v \in \ker AX,$$

sowie $1_n^Tv = 0$ und, trivialerweise,

$$\left\|1_n - \left(1_n - \rho R\,\frac{v}{\|v\|_2}\right)\right\|_2 = \rho R \qquad \Longrightarrow \qquad y := 1_n - \rho R\,\frac{v}{\|v\|_2} \in K_{\rho R}.$$

[112]Die Bedeutung der Notation $\sqrt{X}$ ist hoffentlich klar: Es ist diejenige Matrix mit Diagonalelementen $\sqrt{x_j}$, $j = 1, \ldots, n$. Da $x > 0$ ist gibt es da keine Probleme.

Da für beliebiges $z \in \ker AX \cap K_{\rho R}$

$$B\,(y-z) = \begin{bmatrix} AXy - AXz \\ 1_n^T y - 1_n^T z \end{bmatrix} = \begin{bmatrix} 0 \\ n-n \end{bmatrix} = 0$$

ist, erhalten wir, daß

$$\begin{aligned}(Xc)^T\,(y-z) &= \left(v + B^T\left(BB^T\right)^{-1}BXc\right)^T (y-z) \\ &= v^T\,(y-z) + c^T XB^T\left(BB^T\right)^{-T}\underbrace{B\,(y-z)}_{=0} = v^T\left(1_n - \rho R\frac{v}{\|v\|_2} - z\right) \\ &= v^T\,(1_n - z) - \rho R\,\underbrace{\frac{v^T v}{\|v\|_2}}_{=\|v\|_2} \le \|v\|_2\,\underbrace{\|1_n - z\|}_{\le \rho R} - \rho R\|v\|_2 \le 0,\end{aligned}$$

also $c^T Xy \le c^T Xz$. Darüberhinaus gilt in der Cauchy[113]–Schwarz[114]–Ungleichung, die wir hier verwendet haben, genau dann Gleichheit, wenn $1_n - z = \lambda v$ für ein $\lambda \in \mathbb{R}$ und um generell Gleichheit zu bekommen, muß $\lambda = \pm\frac{\rho R}{\|v\|_2}$ sein; da aber $\lambda = -\frac{\rho R}{\|v\|_2}$ aus dem Simplex hinausführen würde, gilt also Gleichheit genau dann, wenn $z = y$ ist. Damit ist auch die Eindeutigkeit bewiesen.

Für 4) bezeichne schließlich y die Lösung des "lokalen" Optimierungsproblems (4.34) und y^* die Lösung des "globalen" Optimierungsproblems (4.32). Da $y^* \in \Sigma_n$, ist nach (4.33) $\|1_n - y^*\|_2 \le \sqrt{n(n-1)}$ und somit ist

$$z := 1_n + \frac{\rho R}{\sqrt{n(n-1)}}\,(y^* - 1_n) \qquad \Longrightarrow \qquad \|1_n - z\|_2 = \frac{\rho R}{\sqrt{n(n-1)}}\,\|1_n - y^*\|_2 \le \rho R,$$

und da $AXz = 0$ ist, ergibt sich, nochmals mit (4.33), daß $z \in \ker AX \cap K_{\rho R}$. Daher ist

$$\begin{aligned}(Xc)^T y &\le (Xc)^T z = (Xc)^T\left(1_n + \frac{\rho R}{\sqrt{n(n-1)}}\,(y^* - 1_n)\right) \\ &= c^T x\left(1 - \frac{\rho R}{\sqrt{n(n-1)}}\right) + \frac{\rho R}{\sqrt{n(n-1)}}\underbrace{c^T y^*}_{=0} = \left(1 - \frac{\rho}{n-1}\right)c^T x,\end{aligned}$$

[113] Augustin Louis Cauchy, 1789–1857, aufgewachsen in den Wirren der französischen Revolution. Abel sagte 1826 über ihn:

> *Cauchy is mad and there is nothing that can be done about him, although, right now, he is the only one who knows how mathematics should be done.*

Auch sonst erwies sich Cauchy als schwieriger Charakter.

[114] Hermann Amandus Schwarz, 1843–1921, ein Schüler von Weierstraß, Professuren in Zürich (ETH) und Göttingen, bevor er sich 1892 auf der Weierstraß–Nachfolge in Berlin "zur Ruhe setzte" (die Bemerkung stammt von Bieberbach). Arbeitete unter anderem an der Theorie der Minimalflächen und ist durch die Cauchy–Schwarz–Ungleichung "verewigt".

da $R = \sqrt{\frac{n}{n-1}}$. □

Haben wir also das Minimierungsproblem bezüglich y gelöst, so setzen wir schließlich als neuen Wert

$$x' := T_x^{-1}(y) = \frac{n}{x^T y}\, Xy = \frac{n}{1_n^T Xy}\, Xy$$

und erhalten so die "lokale" Optimallösung x' von (4.30). Würden wir bei dieser Zuweisung durch Null dividieren, dann müßte, wegen $x > 0$ ja $y = 0$ sein, aber das ist auch kein Punkt in Σ_n. Das alles läßt sich im folgenden Algorithmus zusammenfassen.

Algorithmus 4.24 *(Karmarkar)*
Gegeben: $x^{(0)} \in \ker A \cap \Sigma_n$, $x^{(0)} > 0$.

1. *Wähle*[115] $\rho \in (0, 1)$.
2. *Für* $r = 0, 1, 2, \ldots$
   (a) *Setze*
   $$B = \begin{bmatrix} AX^{(r)} \\ 1_n^T \end{bmatrix}, \qquad X^{(r)} = \operatorname{diag} x_r.$$
   (b) *Setze*
   $$v = \left(I - B^T\left(BB^T\right)B\right) X^{(r)} c \qquad \textit{und} \qquad y = 1_n - \rho R \frac{v}{\|v\|_2}.$$
   (c) *Setze*
   $$x^{(r+1)} = \frac{n}{y^T x^{(r)}}\, X^{(r)} y.$$

Ergebnis: *Optimallösung*

$$x^\infty = \lim_{r\to\infty} x^{(r)}.$$

4.5 Auffinden eines Startpunkts

Um ein Problem haben wir uns bisher bei unseren Innere–Punkte–Verfahren noch geschickt herumgemogelt, nämlich um die Bestimmung eines zulässigen *inneren* Punktes. Eine Möglichkeit bestünde sicherlich darin, eine zulässige *Ecke* zu bestimmen, beispielsweise mit der Zweiphasenmethode des Simplexalgorithmus[116], wir wollen uns hier aber eine andere Methode ansehen, die uns gleichzeitig noch ein (letztes) Lösungsverfahren für lineare Optimierungsprobleme geben wird, und zwar ein sogenanntes *"Primal–Dual–Verfahren"*. Die Idee besteht darin, das primale[117] Optimierungsproblem

$$\min_x c^T x, \qquad Ax = b, \quad x \geq 0, \tag{4.38}$$

[115] In [45] wird $\rho \in \left(0, \frac{2}{3}\right)$ gewählt.
[116] Genauer gesagt, der Phase 1 davon.
[117] Wir kehren jetzt wieder zur Terminologie vom Anfang des Kapitels zurück, Karmarkar war sozusagen nur ein Ausrutscher.

und das mit Schlupfvariablen $s \in \mathbb{R}^n_+$ versehene duale Optimierungsproblem[118]

$$\max_y b^T y, \qquad A^T y + s = c, \quad s \geq 0, \tag{4.39}$$

in *ein* Problem

$$\min_{x,y} c^T x - b^T y = [b, c] \begin{bmatrix} -y \\ x \end{bmatrix}, \qquad \underbrace{\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \end{bmatrix}}_{\in \mathbb{R}^{m+n \times m+2n}} \underbrace{\begin{bmatrix} x \\ y \\ s \end{bmatrix}}_{\in \mathbb{R}^{m+2n}} = \underbrace{\begin{bmatrix} b \\ c \end{bmatrix}}_{\in \mathbb{R}^{m+n}}, \quad \begin{bmatrix} x \\ s \end{bmatrix} \geq 0, \tag{4.40}$$

"doppelter Größe" zusammenzufassen.

Übung 4.8 Bestimmen Sie das duale Problem zu (4.40). ♢

Und solche "primal–dualen" Optimierungsprobleme sind weit weniger gekünstelt und redundant, als es zuerst aussehen mag!

Um eine Lösung dieses Optimierungsproblems zu bekommen schreiben wir noch schnell die Zielfunktion als

$$f(x) = [c, -b, 0] \begin{bmatrix} x \\ y \\ s \end{bmatrix}$$

und die Nebenbedingung $\begin{bmatrix} x \\ s \end{bmatrix} \geq 0$ in

$$\begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} x \\ y \\ s \end{bmatrix} \geq 0$$

um und verwenden Satz 4.13 bzw. Korollar 4.15, daß $[x, y, z]^T$ eine Optimallösung ist, wenn es $\lambda \in \mathbb{R}^{m+n}$ und $\mu \in \mathbb{R}^{2m+n}_+$ gibt, so daß

$$\begin{bmatrix} c \\ -b \\ 0 \end{bmatrix} - \begin{bmatrix} A^T & 0 \\ 0 & A \\ 0 & I \end{bmatrix} \lambda - \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & I \end{bmatrix} \mu = 0, \qquad \mu^T \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} x \\ y \\ s \end{bmatrix} = 0. \tag{4.41}$$

Zerlegen wir μ in $\mu = [\mu_1, \mu_2, \mu_3]$, $\mu_1 \in \mathbb{R}^m_+$, $\mu_2, \mu_3 \in \mathbb{R}^n_+$, dann ergibt

1. die mittlere Zeile, daß $\mu_2 = 0$,
2. die untere Zeile, daß $\lambda = [\lambda_1, -\mu_3]$, $\lambda_1 \in \mathbb{R}^m$, ist.

Mit $\mu_3 := \xi$, $\lambda_1 = \eta$ und $\mu_1 := \sigma$ erhalten wir dann aus (4.41), daß

$$\begin{aligned} A^T\eta + \sigma &= c, \\ A\,\xi &= b, \qquad \sigma, \xi \geq 0, \\ \sigma^T x + \xi^T s &= 0, \end{aligned} \tag{4.42}$$

[118] Hierbei handelt es sich um keine "neue" Normalform, lediglich um eine Umformung der "alten".

ist. Das heißt, daß $[\xi, \eta, \sigma]^T$ ebenfalls ein zulässiger Punkt sein muß und daß die Wahl $[\xi, \eta, \sigma]^T = [x, y, s]^T$ also genau dann optimal ist, wenn[119] $x_j s_j = 0$ ist, $j = 1, \ldots, n$. Genauer kann man folgendes sagen.

Proposition 4.25 *Der Punkt $[x, y, s]^T$ ist genau dann Optimallösung von (4.40), wenn er die* Karush–Kuhn–Tucker–Bedingungen[120]

$$\begin{array}{rcll} A^T y + s & = & c, & \\ Ax & = & b, & \\ x_j\, s_j & = & 0, & j = 1, \ldots, n \\ [x, s]^T & \geq & 0, & \end{array} \tag{4.43}$$

erfüllt.

Beweis: Eine Richtung haben wir schon gezeigt: Wenn $[x, y, s]^T$ (4.43) erfüllt, dann ist $[\xi, \eta, \sigma]^T = [x, y, s]^T$ der gesuchte Multiplikatorenvektor. Umgekehrt ist

$$x^T s = x^T \left(c - A^T y\right) = x^T c - \underbrace{(Ax)^T}_{=b^T} y = c^T x - b^T y$$

für zulässige x, y genau dann $= 0$, wenn x und y optimale Primal- und Duallösungen sind – mit anderen Worten: jede Optimallösung erfüllt die Bedingung (4.43). □

Ein Tripel $[x, y, s]^T$ ist nach Proposition 4.25 also genau dann Optimallösung von (4.40), wenn

$$0 = F(x, y, s) = \begin{bmatrix} A^T y + s - c \\ Ax - b \\ XS1_n \end{bmatrix}, \qquad F : \mathbb{R}^{m+2n} \to \mathbb{R}^{m+2n}, \qquad \begin{array}{l} X = \operatorname{diag} x, \\ S = \operatorname{diag} s, \end{array} \tag{4.44}$$

ist. Und dieses Problem kann man jetzt mit dem *Newton*[121]*–Verfahren*[122] angehen und lösen. Zu einem vorgegebenen Startwert $\left[x^{(0)}, y^{(0)}, s^{(0)}\right]^T$ berechnet man hierbei

$$\begin{bmatrix} x^{(r+1)} \\ y^{(r+1)} \\ s^{(r+1)} \end{bmatrix} = \begin{bmatrix} x^{(r)} \\ y^{(r)} \\ s^{(r)} \end{bmatrix} - \begin{bmatrix} \xi \\ \eta \\ \sigma \end{bmatrix}, \qquad J[F]\left(x^{(r)}, y^{(r)}, s^{(r)}\right) \begin{bmatrix} \xi \\ \eta \\ \sigma \end{bmatrix} = F\left(x^{(r)}, y^{(r)}, s^{(r)}\right),$$

[119] Nicht vergessen: $s, x \in \mathbb{R}^n_+$, das innere Produkt ist also genau dann $= 0$, wenn alle Summanden, die darin auftauchen $= 0$ sind.

[120] Der Name stammt aus [36].

[121] Sir Isaac Newton, 1643–1727, über seine wissenschaftlichen Beiträge braucht man wohl nichts mehr zu sagen (außer ihm ist nur Mandelbrot mit Äpfeln berühmt geworden). Er zog sich 1693 nach seinem zweiten Nervenzusammenbruch aus der Wissenschaft zurück und wurde "Master of Mint" (Mint = "Münze"). Außerdem gab es heftige Kontroversen zwischen ihm und Leibniz, wer der Erfinder der Analysis wäre, was sogar zur Einsetzung einer Kommission mit Taylor als Vorsitzendem führte; Newton schrieb sogar das offizielle Gutachten für diese Kommission, aus naheliegenden Gründen aber nicht unter seinem eigenen Namen.

[122] Siehe z.B. aber nicht nur [42].

wobei, mit $z = [x, y, s]^T$,

$$J[F] = \left[\frac{\partial F_j}{\partial z_k} \left(x^{(r)} \right) \ : \ j, k = 1, \ldots, m + 2n \right] = \begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S^{(r)} & 0 & X^{(r)} \end{bmatrix}$$

die *Jacobi*[123]*–Matrix* von F ist. Ist außerdem $[x, y, z]^T$ zulässig, dann ist das zu lösende Gleichungssystem also lediglich

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \xi \\ \eta \\ \sigma \end{bmatrix} = \begin{bmatrix} A^T y + s - c \\ Ax - b \\ XS1_n \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ XS1_n \end{bmatrix}. \tag{4.45}$$

So, das ist nun also ein anderes Verfahren, um das Optimierungsproblem zu lösen (es bewegt sich von zulässigen inneren Punkten[124] zu zulässigen inneren Punkten auf die Nullstelle zu), solange man die modifizierte Form

$$\begin{bmatrix} x^{(r+1)} \\ y^{(r+1)} \\ s^{(r+1)} \end{bmatrix} = \begin{bmatrix} x^{(r)} \\ y^{(r)} \\ s^{(r)} \end{bmatrix} - \alpha \begin{bmatrix} \xi \\ \eta \\ \sigma \end{bmatrix}, \qquad \alpha \in \mathbb{R},$$

der Newton–Iteration verwendet, wobei α so klein gewählt wird daß $x_j^{(r+1)} s_j^{(r+1)} > 0$, $j = 1, \ldots, n$, ist.

Aber wie bekommen wir nun einen Startwert? Ganz einfach: wir wählen $x^{(0)}, s^{(0)} > 0$, $y^{(0)}$ beliebig und verwenden die Iteration

$$\begin{bmatrix} x' \\ y' \\ s' \end{bmatrix} = \begin{bmatrix} x \\ y \\ s \end{bmatrix} - \alpha \begin{bmatrix} \xi \\ \eta \\ \sigma \end{bmatrix}, \qquad \begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \xi \\ \eta \\ \sigma \end{bmatrix} = \begin{bmatrix} A^T y + s - c \\ Ax - b \\ XS1_n \end{bmatrix},$$

wobei $\alpha > 0$ wieder so gewählt wird, daß $x' > 0$ und $s' > 0$ ist. Kann man irgendwann $\alpha = 1$ wählen (was der Fall ist, wenn $\xi < x$ und $\sigma < s$ ist), dann ist der Punkt $[x', y', s']^T$ zulässig und wir haben gewonnen!

[123] Carl Gustav Jacob Jacobi, 1804–1851, Zeitgenosse von Gauß mit einer Vielzahl von Beiträgen zur reinen wie zur angewandten Mathematik. Wurde 1832 zum Professor ernannt – nach einer *vierstündigen* Disputation in *Latein*. Jacobi publizierte mindestens in Deutsch, Französisch und Latein (wie übrigens auch Gauß).

[124] Die sich durch $XS1_n > 0$ auszeichnen.

Will man weitergehen, so befaßt man beides unter das Sein, dann unter das, was das Sein verleiht. Von hier aus kann man auf analytischem Wege wieder abwärts steigen [...]

Plotin, *Enneaden*, Band 1

5 Abstiegsverfahren für nichtlineare Optimierung

Jetzt ist es langsam an der Zeit, sich mit *beliebigen*[125] Optimierungsproblemen herumzuschlagen, also mit Problemen der Form

$$\min_{x\in\mathbb{R}^n} f(x), \qquad f:\mathbb{R}^n\to\mathbb{R}.$$

Solche Optimierungsprobleme bezeichnet man als *unrestringiert*, denn der zulässige Bereich ist jetzt nicht mehr eingeschränkt. Allerdings ist das kein Verlust, sondern höchstens eine Verallgemeinerung. Wäre nämlich $D\subset\mathbb{R}^n$ ein kompakter zulässiger Bereich und wäre f wenigstens stetig[126] auf $\mathbb{R}^n$, dann kann man beispielsweise

$$G := \min_{x\in D} f(x) \qquad \text{und} \qquad g := f\,\chi_D + (1-\chi_D)\,(|G|+1)$$

setzen, dann nimmt g sein Minimum nur in D an und dort ist es das Minimum von f. Anders gesagt: jedes *restringierte* Optimierungsproblem läßt sich ganz einfach in ein unrestringiertes umschreiben. Wie man das "richtig" macht, werden wir später in Kapitel 7 sehen.

Einen ganz klaren Vorteil hat die fehlende Nebenbedingung natürlich sofort: Man kann jetzt auf die Bedingungen an den Tangentialkegel aus Proposition 4.10 verzichten.

5.1 Notwendige und hinreichende Kriterien für Minima

Wiederholen wir doch nochmal schnell ein paar Kriterium für die Existenz eines (lokalen) Minimums von f.

Definition 5.1 *Eine Funktion $f\in C(\mathbb{R}^n)$ heißt* richtungsdifferenzierbar *in einem Punkt $x\in\mathbb{R}^n$, wenn für alle $y\in\mathbb{R}^n$ die Grenzwerte*

$$D_y f(x) = \lim_{h\to 0^+} \frac{f(x+hy)-f(x)}{h}$$

[125] Naja, so ganz beliebig können sie nicht sein, ohne ein paar Annahmen geht grundsätzlich nichts.

[126] "Prinzipiell" muss man bei den meisten Verfahren sogar die eine oder andere Form von Differenzierbarkeit annehmen.

existieren.

Richtungsdifferenzierbarkeit ist *schwächer* als Differenzierbarkeit. So ist der Kegel $f(x) = \|x\|_2$ an der Stelle $x = 0$ richtungsdifferenzierbar, weil für jedes y ja $\|0 + hy\| - \|0\| = h\|y\|$ und somit als $D_y f = \|y\|$ ist, aber wegen der Spitze eben nicht differenzierbar. Ist hingegen $f \in C^1(\mathbb{R}^n)$, dann ist natürlich

$$D_y f = (\nabla f)^T \, y. \tag{5.1}$$

Schließlich brauchen wir noch die zweite Ableitung von f, die sogenannte *Hesse–Matrix*, die man in der Optimierungsliteratur meist als

$$\nabla^2 f = \left[\frac{\partial^2 f}{\partial x_j \, \partial x_k} \; : \; j,k = 1,\ldots,n \right]$$

schreibt. Mit deren Hilfe kann man nun Minima sehr einfach und klassisch charakterisieren.

Proposition 5.2 *Sei $f \in C^2(\mathbb{R}^n)$.*

1. *Ist x ein lokales Minimum von f, dann ist $\nabla f(x) = 0$ und $\nabla^2 f(x)$ positiv semidefinit.*
2. *Ist $\nabla f(x) = 0$ und $\nabla^2 f(x) > 0$, dann ist x ein striktes lokales Minimum, d.h. es gibt eine Umgebung D von x, so daß $f(x) < f(x')$, $x' \in D$.*

Beweis: [127] Man verwendet die Taylor[128]–Entwicklung

$$f(x+ty) = f(x) + t\,(\nabla f(x))^T \, y + \frac{t^2}{2} y^T \left(\nabla^2 f(\xi)\right) y, \qquad \xi \in [x, ty],$$

und die Stetigkeit von ∇f und $\nabla^2 f$. Ist nämlich x ein Minimum, so ist *für alle Richtungen* $y \in \mathbb{R}^n$

$$0 \le \frac{f(x+ty) - f(x)}{t} \quad \to \quad (\nabla f(x))^T \, y,$$

was nur mit $\nabla f(x) = 0$ zu erfüllen ist. Damit ist aber dann

$$\frac{t^2}{2} y^T (\nabla f(\xi))\, y = f(x+ty) - f(x) \ge 0$$

für alle hinreichend kleinen t, das heißt $y^T (\nabla f(\xi))\, y \ge 0$ und mit $t \to 0$ konvergiert ja $\xi \to x$. Die zweite Aussage folgt direkt aus der Taylorentwicklung und der Tatsache, daß die *strikte* positive Definitheit von $\nabla^2 f$ an der Stelle x die *strikte* positive Definitheit von $\nabla^2 f$ in einer ganzen Umgebung D von x impliziert. □

[127] Sozusagen zum "Aufwärmen".

[128] Brook Taylor, 1685–1731, war Mitglied einer 1712 eingesetzten Kommission, die darüber zu entscheiden hatte, ob Newton oder Leibniz, die Analysis "erfunden" hätte. "Seine" berühmten Taylor–Reihen wurden, laut Taylor selbst, durch eine Bemerkung von Machin über "Sir Isaac Newton's series" in *Child's Coffeehouse* motiviert (nicht vergessen, das war etwa 1710 und da gab es weder Starbuck's noch Coffee Bay).

5.2 Nochmals Konvexität

Trotzdem sind Differenzierbarkeit und vor allem zweimalige Differenzierbarkeit von f schon *starke* Forderungen. Und eigentlich braucht man sie ja auch gar nicht immer um Minima zu beschreiben.

Definition 5.3 *Für $x \in \mathbb{R}^n$ und eine in x richtungsdifferenzierbare Funktion f bezeichnen wir mit*

$$G[f,x] \ : \ \mathbb{R}^n \to \mathbb{R}, \qquad G[f,x](y) := D_y f(x)$$

die Gateaux–Variation *von f an der Stelle x.*

Übung 5.1 Zeigen Sie: Die Funktion $f(x) = \|x\|_\infty$ ist an der Stelle $x = 0$ richtungsdifferenzierbar, aber ihre Gateaux–Variation ist dort unstetig. Zumindest, wenn $n > 1$ ist. ♢

Proposition 5.4 *Sei $f \in C(\mathbb{R}^n)$* richtungsdifferenzierbar *in x. Ist x ein lokales Minimum von f, dann ist*

$$D_y f(x) \geq 0, \qquad y \in \mathbb{R}^n. \tag{5.2}$$

Beweis: Auch noch ganz einfach! Ist x ein lokales Minimum, dann ist $f(x) \leq f(x+ty)$ für alle $y \in \mathbb{R}^n$ und alle hinreichend kleinen[129] $t > 0$. Also ist auch

$$0 \leq \frac{f(x+ty) - f(x)}{t} \quad \to \quad D_y f.$$

□

Zur Erinnerung: Eine Funktion f heißt *konvex*, wenn

$$f(\alpha x + (1-\alpha)x') \leq \alpha\, f(x) + (1-\alpha) f(x'), \qquad x, x' \in \mathbb{R}^n, \quad \alpha \in [0,1].$$

Lokale Minima x konvexer Funktionen sind immer *globale* Minima! Wäre nämlich $f(x') < f(x)$ für irgendein $x' \in \mathbb{R}^n$, dann ist für $\alpha \in (0,1)$

$$f(\alpha x + (1-\alpha)x') \leq \alpha\, f(x) + (1-\alpha) f(x') < \alpha\, f(x) + (1-\alpha) f(x) = f(x),$$

was für $\alpha \to 1$ gegen $f(x)$ konvergiert. Außerdem sind konvexe Funktionen in gewissem Sinne "differenzierbar".

Proposition 5.5 *Sei $f \in C(\mathbb{R}^n)$ konvex. Dann ist f an jeder Stelle $x \in \mathbb{R}^n$ richtungsdifferenzierbar und die Gateaux–Variation von f in x ist*

1. positiv homogen*, d.h.,*

$$G[f,x](\alpha\cdot) = \alpha\, G[f,x](\cdot), \qquad \alpha > 0.$$

[129] Zeigen Sie: t hängt von y ab. Naja, eigentlich eher von $\|y\|$.

2. sublinear, *d.h.*

$$G[f,x](y+y') \leq G[f,x](y) + G[f,x](y').$$

Beweis: Zu $x, y \in \mathbb{R}^n$ setzen wir

$$\phi(t) = \frac{f(x+ty)-f(x)}{t}, \qquad t \in (0,1],$$

und betrachten $\lim_{t\to 0^+}\phi(t)$. Mit $x = \frac{x+ty}{t+1} + \frac{t(x-y)}{t+1}$ und daher

$$f(x) \leq \frac{1}{t+1} f(x+ty) + \frac{t}{t+1} f(x-y)$$

und

$$\frac{t}{t+1} f(x) \leq \underbrace{\frac{1}{t+1}(f(x+ty)-f(x))}_{=\frac{t}{t+1}\phi(t)} + \frac{t}{t+1} f(x-y)$$

erhalten wir zuerst einmal, daß ϕ auf $(0,1]$ unabhängig von t durch $f(x)-f(x-y)$ nach unten beschränkt ist. Ist außerdem $0 < s \leq t \leq 1$, dann ist

$$\underbrace{f(x+sy)-f(x)}_{=s\,\phi(s)} = f\underbrace{\left(\frac{s}{t}(x+ty)+\frac{t-s}{t}x\right)}_{=x} - f(x) \leq \underbrace{\frac{s}{t}f(x+ty) - \frac{s}{t}f(x)}_{s\,\phi(t)},$$

also $\phi(s) \leq \phi(t)$ und da ϕ einen monoton steigende Funktion ist, die nach unten beschränkt ist, muß

$$D_y f(x) = \lim_{t\to 0^+}\phi(t)$$

existieren. Außerdem ist[130]

$$D_{\alpha y} f(x) = \lim_{t\to 0^+} \frac{f(x+t(\alpha y))-f(x)}{t} = \lim_{t\to 0^+} \alpha\,\frac{f(x+(\alpha t)y)-f(x)}{\alpha t} = \alpha\, D_y f(x)$$

sowie

$$\begin{aligned} D_{y+y'} f(x) &= \lim_{t\to 0^+} \frac{f(x+t(y+y'))-f(x)}{t} \leq \lim_{t\to 0^+} \frac{\frac{1}{2}f(x+2ty)+\frac{1}{2}f(x+2ty')-f(x)}{t} \\ &= \lim_{t\to 0^+}\left(\frac{f(x+2ty)-f(x)}{2t} + \frac{f(x+2ty')-f(x)}{2t}\right) = D_y f(x) + D_{y'} f(x). \end{aligned}$$

□

Übung 5.2 Zeigen Sie: Ist $f \in C(\mathbb{R}^n)$ konvex, dann ist auch $G[f,x]$ konvex, $x \in \mathbb{R}^n$. ♢

Wir sehen also, daß konvexe Funktionen in der Optimierung wieder einmal eine ganz ausgezeichnete Rolle spielen.

[130] Und hier braucht man noch nicht einmal Konvexität.

5.3 Abstiegsverfahren – die allgemeine Idee

Wir betrachten jetzt zuerst einmal "glatte" Minimierungsprobleme und zwar nehmen wir jetzt an, daß

1. für vorgegebenes $x \in \mathbb{R}^n$ die *Niveaumenge*

$$F := \{x' \in \mathbb{R}^n \ : \ f(x') \leq f(x)\}$$

   kompakt ist.

2. die Zielfunktion auf einer offenen Umgebung D von F stetig differenzierbar ist.

3. der Gradient ∇f auf F *Lipschitz–stetig* ist, das heißt, es gibt $\gamma_f > 0$, so daß

$$\|\nabla f(x) - \nabla f(x')\|_2 \leq \gamma_f \|x - x'\|_2$$

Diese Voraussetzungen sind in gewissem Sinne Minimalvoraussetzungen, um "vernünftige" Iterationsverfahren "basteln" zu können. Die Differenzierbarkeit brauchen wir, um eine Richtung zu finden, in der wir uns verbessern können, die Kompaktheit von F_x gibt uns die Hoffnung, irgendwann zumindest bei einem lokalen Minimum anzukommen und die Lipschitz–Stetigkeit sorgt dafür, daß f "praktisch C^2" ist und so nicht allzuviel Unsinn anstellt.

Definition 5.6 *Ein Punkt $x \in \mathbb{R}^n$ heißt* stationärer Punkt *von f, wenn $\nabla f(x) = 0$ ist.*

Nun zu unserem Verfahren. Ist x kein stationärer Punkt, d.h. ist $\nabla f(x) \neq 0$, dann gibt es Richtungen, so daß $D_y f(x) < 0$ ist, solche Richtungen bezeichnet man als *Abstiegsrichtungen* und solche Richtungen haben das Potential, daß $f(x + ty)$ für hinreichend kleine Werte von t kleiner als $f(x)$ wird. Und tatsächlich kann man das auch beweisen.

Lemma 5.7 *Sei $x \in F$ und $y \in \mathbb{R}^n$ so gewählt, daß $D_y f(x) < 0$ ist. Dann gibt es einen Wert $t > 0$, so daß*

$$f(x + ty) < f(x).$$

Beweis: Wir geben fast sogar ein quantitatives Resultat, indem wir zeigen, daß

$$f(x + ty) - f(x) \leq t\left(D_y f(x) + t\frac{\gamma}{2}\|y\|_2^2\right). \tag{5.3}$$

Das folgt wieder mal aus einer Taylor–Entwicklung

$$\begin{aligned} f(x + ty) - f(x) &= t\, D_y f(x) + \int_0^t \underbrace{(D_y f(x + sy) - D_y f(x))}_{=(\nabla f(x+sy) - \nabla f(x))^T y} ds \\ &\leq t\, D_y f(x) + \gamma_f \int_0^t \|sy\|_2 \, \|y\|_2 \, ds = t\, D_y f(x) + t^2 \frac{\gamma}{2}\|y\|_2^2 \end{aligned}$$

Wählt man nun in (5.3)

$$0 < t < \frac{2\,|D_y f(x)|}{\gamma\ \|y\|_2^2},$$

dann ist die rechte Seite von (5.3), was die Behauptung beweist. □

Wir haben uns also mit dem folgenden (Doppel-)Problem auseinanderzusetzen: Wie wählt man zu gegebenem x

1. eine *Abstiegsrichtung* y
2. eine *Schrittweite* t

so daß

$$f(x + ty) < f(x)$$

ist. Wie man dann den Prozess zum Konvergieren bekommt, das ist dabei noch gar nicht mal die Frage.

5.4 Abstiegsrichtungen – der naive Ansatz

Die erste Idee, die man haben könnte, besteht darin, als Abstiegs*richtung* y, $\|y\|_2 = 1$, diejenige Richtung zu wählen, für die $D_y f(x)$ *minimal* wird, und das ist natürlich der Wert

$$y = -\frac{\nabla f(x)}{\|\nabla f(x)\|_2}. \tag{5.4}$$

Und die Schrittweite könnte man ja so wählen, daß man das Minimum auf der *ganzen* Geraden $x + ty$, $t \in \mathbb{R}$, bestimmt. Dieses Minimum zeichnet sich dadurch aus, daß

$$0 = \frac{d}{dt} f\,(x + ty) = D_y f(x + ty) = (\nabla f(x + ty))^T\ y.$$

Wählt man nun das kleinste positive t mit dieser Eigenschaft, dann muß es zu einem Minimum gehören, da y ja eine Abstiegsrichtung ist. Diese Nullstelle von $\phi(t) := D_y f(x + ty)$, $t \in \mathbb{R}$, könnte man dann mit einem Newton–Verfahren, das mit dem Punkt x, also mit $t = 0$, gestartet wird, ermitteln[131]. Das geht gut, solange $\phi(0) \geq 0$ ist, da dann, wegen $\phi'(0) < 0$,

$$t_1 = 0 - \frac{\phi(0)}{\phi'(0)} > 0$$

ist und wir zumindest schon mal in der richtigen Richtung anfangen. Diese Schrittweitenwahl bezeichnet man als *exakte Schrittweite*. Und zumindest in einem Fall kann man die exakte Schrittweite auch einfach berechnen.

[131] Oder zu ermitteln versuchen.

Beispiel 5.8 *Ist*

$$f(x) = \frac{1}{2}x^T Ax - b^T x + c, \qquad A \in \mathbb{R}^{n\times n},\, b \in \mathbb{R}^n, c \in \mathbb{R}, \tag{5.5}$$

mit einer symmetrischen, positiv definiten Matrix A, dann ist

$$\nabla f(x) = Ax - b \qquad \Longrightarrow \qquad y = b - Ax.$$

Da

$$f(x+ty) = \frac{1}{2}(x+ty)^T A(x+ty) - b^T(x+ty) = \underbrace{x^T Ax - b^T x}_{=f(x)} + \frac{t^2}{2}\, y^T Ay + t\, x^T Ay - t\, b^T y,$$

erhalten wir somit die Bedingung

$$0 = \frac{d}{dt} f(x+ty) = ty^T Ay + y^T(Ax - b)$$

also

$$t = \frac{y^T(b-Ax)}{y^T Ay} = \frac{\|Ax-b\|_2^2}{(b-Ax)^T A(b-Ax)}.$$

Dieses Beispiel ist nicht ganz unbedeutend, ganz im Gegenteil: Ist $f \in C^2(\mathbb{R}^n)$ und sind wir nahe genug an einem *strikten* lokalen Minimum, dann lässt sich f – Taylor sei Dank – in einer hinreichend kleinen Umgebung immer durch so eine quadratische Parabel annähern. Schreiben wir nämlich

$$f(x) \sim \frac{1}{2}\,(x-x^*)^T\,\nabla^2 f\,(x^*)\,(x-x^*) + (x-x^*)^T\,\nabla f\,(x^*) + f\,(x^*)\,, \tag{5.6}$$

dann sind wir, zumindest in einer gewissen Umgebung von x^*, in genau der quadratischen Situation. Und ist x^* ein *striktes* lokales Minimum, dann ist ja, nach Proposition 5.2, auch die von Haus aus symmetrische Matrix $\nabla^2 f\,(x^*)$ strikt positiv definit. Wir könnten also zu einem Startwert $x^{(0)}$ eine Folge von Näherungslösungen *iterativ* über

$$x^{(j+1)} = x^{(j)} - \frac{\nabla^T f\left(x^{(j)}\right)\nabla f\left(x^{(j)}\right)}{\nabla^T f\,(x^{(j)})\,\nabla^2 f\,(x^{(j)})\,\nabla f\,(x^{(j)})}\,\nabla f\left(x^{(j)}\right), \qquad j \in \mathbb{N}_0,$$

bestimmen. Nun ist aber das Verfahren des steilsten Abstiegs aber leider nicht immer das Mittel der Wahl – naiv geht's eben nicht immer. Denn leider kann es sehr schnell passieren, daß dieses Verfahren beliebig langsam, das heißt numerisch gar nicht, konvergiert. Dazu setzen wir

$$f(x) = \frac{1}{2}\,(Ax-b)^T\,A^{-1}\,(Ax-b) = \frac{1}{2}x^T Ax - b^T x + b^T A^{-1} b,$$

sorgen also dafür, daß der Minimalwert gerade Null ist, und erhalten das folgende Resultat.

Lemma 5.9 *Es seien $\lambda_1 \leq \ldots \leq \lambda_n$ die Eigenwerte von A. Dann ist*

$$f\left(x^{(j+1)}\right) \leq \left(1 - \frac{\lambda_1}{\lambda_n}\right) f\left(x^{(j)}\right), \qquad j \in \mathbb{N}_0. \tag{5.7}$$

Beweis: Für $j \in \mathbb{N}_0$ und $y = b - Ax^{(j)}$ ist

$$\begin{aligned}
f\left(x^{(j+1)}\right) &= f\left(x^{(j)}\right) + \frac{1}{2}\left(\frac{y^T y}{y^T A y}\right)^2 y^T A y + \frac{y^T y}{y^T A y} y^T \underbrace{\left(Ax^{(j)} - b\right)}_{=-y} \\
&= f\left(x^{(j)}\right) - \frac{1}{2}\frac{\left(y^T y\right)^2}{y^T A y} = f\left(x^{(j)}\right) - \frac{1}{2}\frac{\|y\|_2^4}{y^T A y} \leq f\left(x^{(j)}\right) - \frac{1}{2}\frac{\|y\|_2^4}{\lambda_n \|y\|_2^2} \\
&= f\left(x^{(j)}\right) - \frac{1}{2}\frac{y^T y}{\lambda_n} = \frac{1}{2} y^T \left(A^{-1} - \frac{1}{\lambda_n} I\right) y \leq \frac{1}{2} y^T \left(A^{-1} - \frac{\lambda_1}{\lambda_n} A^{-1}\right) y \\
&= \left(1 - \frac{\lambda_1}{\lambda_n}\right) f\left(x^{(j)}\right).
\end{aligned}$$

Der letzte Schritt ist wegen $x^T A^{-1} x \leq \lambda_1^{-1} x^T x$ richtig[132]. □

Übung 5.3 Zeigen Sie: Ist $A \in \mathbb{R}^{n \times n}$ symmetrisch und positiv definit und ist λ der größte Eigenwert von A, dann ist

$$x^T A x \leq \lambda \|x\|_2^2, \qquad x \in \mathbb{R}^n.$$

♢

Dieses Verhalten kann man auch praktisch in `Matlab` sehen: man startet mit einer zufälligen $n \times n$–Matrix `A = rand( n )` und startet die Methode des steilsten Abstiegs mit einer Matrix der Form

```
A' * A + t eyes( n )
```

für verschiedene Werte von `t`> 0. Je größer `t` ist, desto besser wird die Methode funktionieren und konvergieren, für `t` $= 0$ hingegen kann man normalerweise nicht mehr von Konvergenz sprechen (die Matrix wird fast singulär). Warum das so ist und was die geometrische Interpretation ist, sieht man einfach am folgenden Beispiel.

Beispiel 5.10 *Wir betrachten*

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 10^{-3} \end{bmatrix} \qquad \textit{und} \qquad b = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

Die Lösung ist natürlich $x = [1, 1000]^T$. Für Vektoren $x = [x_1, x_2]^T$ mit moderatem x_2 ist dann

$$y = -\nabla f(x) = b - Ax = \begin{bmatrix} 1 - x_1 \\ 1 - 10^{-3} x_2 \end{bmatrix} \approx \begin{bmatrix} 1 - x_1 \\ 1 \end{bmatrix}$$

[132] Der größte Eigenwert von A^{-1} ist λ_1^{-1}.

und

$$t := \frac{y^T y}{y^T A y} \sim \frac{1 + (1 - x_1)^2}{(1 - x_1)^2}.$$

Ist nun $x_1 \sim 0$ *oder* $x_1 \sim 2$*, dann ist* $\alpha \sim 2$ *und damit ist*

$$x + ty \sim \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + 2 \begin{bmatrix} 1 - x_1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 - x_1 \\ x_2 + 2 \end{bmatrix}$$

Starten wir also mit $x^{(0)} = 0$*, so werden die Werte* $x_1^{(j)}$ *anfangs zwischen* 0 *und* 2 *pendeln*[133] *und die Werte* $x_2^{(j)} \sim 2j$ *sein. Kommt dann das Verfahren richtig "in Fahrt" (also gegen Ende des Verfahrens), dann wird auch die Konvergenz deutlich schneller.*
Geometrisch bedeutet dies, daß sich das Verfahren an "flachgedrückten" Ellipsen entlanghangelt, siehe Abb. 5.1.

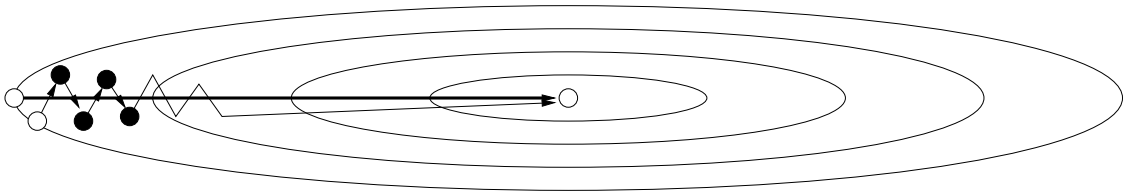


Abbildung 5.1: Verfahren des steilsten Abstiegs für Beispiel 5.10 (nicht maßstabsgetreu). Beachte: mit dem "richtigen" Startwert (nicht ganz zufällig ein Eigenvektor von A) würde das Verfahren nach *einem* Schritt erfolgreich terminieren.

Übung 5.4 Implementieren Sie das Verfahren des steilsten Abstiegs für Probleme der Form (5.5) in `Matlab`. ♢

Es gibt noch ein interessantes Experiment: Verwendet man eine modifizierte Version des Verfahrens des steilsten Abstiegs, bei der der negative Gradient um etwa 5 % *zufällig* gestört wird, dann erhält man Konvergenz in signifikant *weniger* Schritten.

5.5 Abstiegsrichtungen – konjugierte Gradienten

Wie das Beispiel mit den "plattgedrückten" Ellipsen zeigt, sind die steilsten Abstiegsrichtungen nicht unbedingt optimal, man muß offensichtlich Terme zweiter Ordnung berücksichtigen, die das Problem "verzerren" können.

Definition 5.11 *Sei* $A \in \mathbb{R}^{n \times n}$ *eine symmetrische, positiv definite Matrix. Zwei Vektoren* x, y *heißen* konjugiert *bezüglich* A*, wenn* $x^T A y = 0$ *ist. Entsprechend heißt eine endliche Menge* $X \subset \mathbb{R}^n$ *konjugiert, wenn*

$$x^T A x' = 0, \qquad x \neq x' \in X.$$

[133] In Wirklichkeit sind sie stets etwas größer als 0 und etwas kleiner als 2 und dieses "etwas" wächst.

Übung 5.5 Sei $A \in \mathbb{R}^{n\times n}$ eine symmetrische, positiv definite Matrix. Zeigen Sie, daß

$$\langle x, y\rangle_A := x^T A y \qquad \text{und} \qquad \|x\|_A := \langle x, x\rangle_A^{1/2}$$

ein Skalarprodukt und eine Norm definieren. ♢

Bemerkung 5.12 *1. Unter Berücksichtigung von Übung 5.5 bedeutet Konjugiertheit von Vektoren also eigentlich nichts anderes als* Orthogonalität*, nur eben nicht bezüglich des* Standard–Skalarprodukts $\langle x, y\rangle = x^T y$*, sondern bezüglich des Skalarprodukts* $\langle\cdot,\cdot\rangle_A$.

2. *Da orthogonale Vektoren immer* linear unabhängig *sind und da Konjugiertheit nichts anderes als Orthogonalität ist, sind also auch konjugierte Vektoren immer linear unabhängig.*

Diese Beobachtungen legen es nahe, unser Optimierungsproblem zur Abwechslung mal als ein *Approximationsproblem* aufzufassen – eigentlich sind ja Approximationsproblem auch nur spezielle Optimierungsprobleme. Was uns hier interessiert ist das folgende Problem:

Sei $X \subset \mathbb{R}^n$ ein Unterraum[134] *mit $\dim X = m \le n$ und sei $\langle\cdot,\cdot\rangle$ ein Skalarprodukt*[135]*, sowie $\|\cdot\|$ die dadurch induzierte Norm. Zu $y \in \mathbb{R}^n$ bestimme man $x^* \in X$, so daß*

$$\|y - x^*\| = \min_{x\in X} \|y - x\|\,. \tag{5.8}$$

Die Lösung ist recht einfach – Normen, die von Skalarprodukten herrühren sind sehr stark mit Orthogonalität verknüpft, womit man die Lösungen explizit angeben kann[136]

Lemma 5.13 *Sei $X \subset \mathbb{R}^n$ mit $\dim X = m$. Dann ist für jedes $y \in \mathbb{R}^n$*

$$\|y - x^*\| = \min_{x\in X} \|y - x\| \qquad \Longleftrightarrow \qquad \langle y - x^*, X\rangle = 0. \tag{5.9}$$

Ist darüberhinaus $\{x_1, \ldots, x_m\}$ eine Orthonormalbasis[137] *von X, dann ist*

$$x^* = \sum_{j=1}^{m} \langle y, x_j\rangle\, x_j \tag{5.10}$$

die Minmallösung von (5.8).

[134] Selbstverständlich kann man das Problem auch auf unendlichdimensionalen (Funktionen)–Räumen betrachten.

[135] Also eine symmetrische, definite Bilinearform, oder, was hier dasselbe ist, eine nichtentartete Sesquilinearform, siehe z.B. [5, S. 314].

[136] Das macht dann auch die *Approximation in Hilberträumen* zu einem vergleichsweise leichten Problem, beispielsweise hat man keine Probleme mit Existenz und *Eindeutigkeit* von sogenannten *Bestapproximationen*.

[137] Und eine solche kann man ja über das Gram–Schmidt–Verfahren immer konstruieren.

Bemerkung 5.14 *Bevor wir an den Beweis von Lemma 5.13 herangehen, sollten wir uns erst einmal klarmachen, warum es so hilfreich für unser Problem ist und warum die konjugierten Vektoren auftauchen. Das ist aber einfach: Die konjugierten Vektoren sind ja gerade* orthogonal *bezüglich des Skalarprodukts $\langle\cdot,\cdot\rangle_A$, so daß wir mit ihrer Hilfe ohne große Probleme die Minimallösungen von (5.8) bestimmen können.*

Beweis von Lemma 5.13: Sei $\{x_1,\ldots,x_n\}$ eine Orthonormalbasis von X und sei $y\in\mathbb{R}^n$. Dann ist, für jede Wahl von Zahlen $a_1,\ldots,a_m\in\mathbb{R}$,

$$\begin{aligned}
\left\| y-\sum_{j=1}^m \langle y,x_j\rangle\, x_j\right\|^2 &= \left\langle y-\sum_{j=1}^m \langle y,x_j\rangle\, x, y-\sum_{j=1}^m \langle y,x_j\rangle\, x_j\right\rangle\\
&= \langle y,y\rangle - 2\sum_{j=1}^m \langle y,x_j\rangle\langle y,x_j\rangle + \sum_{j,k=1}^m \langle y,x_j\rangle\langle y,x_k\rangle \underbrace{\langle x_j,x_k\rangle}_{\delta_{jk}}\\
&= \|y\|^2 - \sum_{j=1}^n \langle y,x_j\rangle^2 \le \|y\|^2 - \sum_{j=1}^m \langle y,x_j\rangle^2 + \sum_{j=1}^m (a_j-\langle y,x_j\rangle)^2\\
&= \|y\|_2^2 - 2\sum_{j=1}^m a_j\,\langle y,x_j\rangle + \sum_{j=1}^m a_j^2\langle x_j,x_j\rangle = \left\| y-\sum_{j=1}^m a_j\,x_j\right\|^2,
\end{aligned}$$

weswegen die Bestapproximation gerade der Fall $a_j=\langle y,x_j\rangle$, $j=1,\ldots,m$ ist. Und das ist für $k=1,\ldots,m$ äquivalent zu

$$\left\langle y-\sum_{j=1}^m \langle y,x_j\rangle\, x_j, x_k\right\rangle = \langle y,x_k\rangle - \sum_{j=1}^m \langle y,x_j\rangle\langle x_j,x_k\rangle = \langle y,x_k\rangle - \langle y,x_k\rangle = 0.$$

□

Erinnern wir uns kurz daran, daß sich die eindeutige Minimalstelle von $f(x)=\frac{1}{2}x^TAx-b^Tx$ ja dadurch auszeichnet, daß

$$0=\nabla f(x)=Ax-b \qquad\Longleftrightarrow\qquad Ax=b \quad\text{oder}\quad x=A^{-1}b.$$

Wären jetzt also $p_1,\ldots,p_n\in\mathbb{R}^n$ eine Orthogonalbasis[138] bezüglich $\langle\cdot,\cdot\rangle_A$, dann besteht die Idee darin, ausgehend von $x^{(0)}$ die Werte $x^{(k)}=x^{(0)}+a_1\,p_1+\cdots+a_k\,p_k$ so zu wählen, daß

$$\left\|x^{(k)}-A^{-1}b\right\| = \min\left\{\left\|x-A^{-1}b\right\| \;:\; x\in x^{(0)}+\text{span }\{p_1,\ldots,p_k\}\right\}$$

oder

$$\left\|\xi^{(k)}-\left(x^{(0)}-A^{-1}b\right)\right\| = \min\left\{\left\|\xi-\left(x^{(0)}-A^{-1}b\right)\right\| \;:\; \xi\in\text{span }\{p_1,\ldots,p_k\}\right\}$$

[138] Wir verzichten aus den diversesten Gründen auf die Normierung $\langle p_j,p_j\rangle=1$, $j=1,\ldots,n$.

ist – wir lösen also unser Gleichungssystem $Ax = b$, indem wir sukzessive minimieren, denn das geht nach Lemma 5.13 ja ganz einfach: mit $y = x^{(0)} - A^{-1}b$ ist

$$\begin{aligned}\xi^{(k)} &= \sum_{j=1}^{k} \frac{\langle x^{(0)} - A^{-1}b, p_j\rangle_A}{\langle p_j, p_j\rangle_A}\, p_j = \sum_{j=1}^{k-1} \frac{\langle x^{(0)} - A^{-1}b, p_j\rangle_A}{\langle p_j, p_j\rangle_A}\, p_j + \frac{\langle x^{(0)} - A^{-1}b, p_k\rangle_A}{\langle p_k, p_k\rangle_A}\, p_k \\ &= \xi^{(k-1)} + \frac{\langle x^{(0)} - A^{-1}b, p_k\rangle_A}{\langle p_k, p_k\rangle_A}\, p_k,\end{aligned}$$

also

$$x^{(k)} = x^{(0)} - \xi^{(k)} = \underbrace{x^{(0)} - \xi^{(k-1)}}_{=x^{(k-1)}} - \frac{\langle x^{(0)} - A^{-1}b, p_k\rangle_A}{\langle p_k, p_k\rangle_A}\, p_k. =: x^{(k-1)} + \alpha_k\, p_k.$$

In unserer "Optimierungsterminologie" verwenden wir jetzt also für den Übergang von $x^{(k-1)}$ zu $x^{(k)}$ die *Abstiegsrichtung* p_k und die *Schrittweite*

$$\begin{aligned}\alpha_k &= -\frac{\langle x^{(0)} - A^{-1}b, p_k\rangle_A}{\langle p_k, p_k\rangle_A} = -\frac{\langle x^{(k-1)} - A^{-1}b, p_k\rangle_A}{\langle p_k, p_k\rangle_A} = -\frac{p_k^T\left(Ax^{(k-1)} - b\right)}{p_k^T A p_k} \\ &= -\frac{p_k^T r_{k-1}}{p_k^T A p_k}, \qquad r_{k-1} := Ax^{(k-1)} - b\end{aligned}$$

da $x^{(k-1)} - x^{(0)} \in \text{span}\,\{p_1, \ldots, p_{k-1}\} \perp_A p_k$. Hätten wir also eine Basis aus konjugierten Vektoren, dann wären wir fertig. Bleibt also nur die Frage, wie man diese Basis bestimmt. Dazu nehmen wir an, wir hätten schon einen Punkt $x^{(k)}$ und konjugierte Richtungen $p_1, \ldots, p_k$ bestimmt und suchen nun eine konjugierte Richtung, in die wir uns verbessern könnten. Ein erster Versuch für eine Abstiegsrichtung wäre natürlich nun wieder der *steilste* Abstieg $-r_k = b - Ax^{(k)}$. Führt diese Richtung zu keiner Verbesserung, dann haben wir unser Minimum gefunden, ergibt sich hingegen eine Verbesserung, dann ist

$$r_k \notin \text{span}\,\{p_j \;:\; j = 1, \ldots, k\},$$

aber leider auch (noch) nicht konjugiert. Auch kein Problem, dann setzen wir eben[139]

$$p_{k+1} := r_k - \sum_{j=1}^{k} \frac{\langle r_k, p_j\rangle_A}{\langle p_j, p_j\rangle_A}\, p_j, \tag{5.11}$$

und erhalten so, daß $\langle p_{k+1}, P_k\rangle_A = 0$, also unsere konjugierte Richtung. In Wirklichkeit ist das aber sogar noch einfacher.

Lemma 5.15 *Die Vektoren $p_0, \ldots, p_n$, generiert durch die Vorschrift*

$$p_{k+1} := r_k - \frac{\langle r_k, p_k\rangle_A}{\langle p_k, p_k\rangle_A}\, p_k, \qquad p_0 := 0, \tag{5.12}$$

sind konjugiert.

[139] Wenn das jemandem bekannt vorkommen sollte – stimmt! Das ist die Vorgehensweise wie beim aus der Linearen Algebra bekannten *Gram-Schmidt-Verfahren*.

Beweis: Wir definieren die Räume

$$P_k := \text{span } \{p_1, \ldots, p_k\}$$

und behaupten zuerst einmal, daß

$$P_k = \text{span } \{r_j \ : \ j = 0, \ldots, k-1\} = \text{span } \left\{A^j\, r_0 \ : \ j = 0, \ldots, k-1\right\}, \tag{5.13}$$

was man durch Induktion über k nachweist. Der Fall $k = 0$ ist die triviale Feststellung, daß $p_1 = r_0 = A^0 r_0$ und für den Schritt $k \to k+1$ verwenden wir die Rekursionsformel

$$\begin{aligned} r_k &= Ax^{(k)} - b = A\left(x^{(k-1)} + \alpha_k p_k\right) - b = Ax^{(k-1)} - b + \alpha_k\, Ap_k \\ &= \underbrace{r_{k-1}}_{\in P_k} - \frac{p_k^T r_{k-1}}{p_k^T A p_k}\ \underbrace{Ap_k}_{\in AP_k}, \end{aligned} \tag{5.14}$$

die direkt aus der Definition der $x^{(k)}$ und damit der r_k folgt und die uns zusammen mit der Induktionshypothese liefert, daß $r_k \in P_k + AP_k$, also, nach (5.12), auch $p_{k+1} \in P_k + AP_k$ und daher ist

$$P_{k+1} \subseteq \text{span } \left\{A^j\, r_0 \ : \ j = 0, \ldots, k\right\},$$

woraus (5.13) aus einfachen Dimensionsgründen folgt – schließlich sind die Vektoren $p_1, \ldots, p_{k+1}$ ja linear unabhängig. Wegen Lemma 5.13, genauer, wegen (5.9), ist

$$r_k^T P_k = 0 \tag{5.15}$$

und da $P_k \supset AP_{k-1}$ erhalten wir auch, daß

$$0 = r_k^T AP_{k-1} = \langle r_k, P_{k-1} \rangle_A. \tag{5.16}$$

□

Das fassen wir jetzt einmal in einem Algorithmus zusammen.

Algorithmus 5.16 *(Lineares CG–Verfahren*[140]*)*
Gegeben: *Symmetrische, positiv definite Matrix* $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$.

1. *Wähle* beliebigen *Startwert* $x^{(0)} \in \mathbb{R}^n$.
2. *Setze* $p_0 = 0$.
3. *Für* $k = 1, 2, \ldots$
   (a) *Setze*
   $$r := Ax^{(k-1)} - b.$$

[140]Das englische Schlagwort ist *"Conjugate Gradients"*, ein Verfahren, das auf Hestenes & Stiefel [24] zurückgeht.

(b) Setze

$$p_k := r - \frac{r^T A p_{k-1}}{p_{k-1}^T A p_{k-1}}\, p_{k-1}.$$

(c) Setze

$$x^{(k)} = x^{(k-1)} - \frac{r^T p_k}{p_k^T A p_k}\, p_k.$$

Ergebnis: *Folge $x^{(k)}$ die gegen das Minimum von f konvergiert.*

Übung 5.6 Zeigen Sie: Das CG–Verfahren terminiert spätestens nach n Schritten, das heißt, $x^{(n)} = x^{(n+1)} = x^{(n+2)} = \cdots$ ♢

Bemerkung 5.17 *Mathematisch–theoretisch terminiert das CG–Verfahren nach n Schritten mit einem Minimum von $f(x) = \frac{1}{2}x^T Ax - b^T x$, oder, äquivalent, mit einer Lösung von $Ax = b$. In der numerischen Praxis ist dem aber leider* nicht *so – allerdings, und hier ist die Überraschung, hat es sich gezeigt, daß die konjugierten Gradienten ein sehr gutes und stabiles* Iterationsverfahren *liefern, wenn man nicht mit dem n–ten Schritt aufhört, sondern einfach weiteriteriert.*

Jetzt aber zurück zu unserem allgemeinen Optimierungsproblem. Ist $f \in C^2(\mathbb{R}^n)$, so könnten wir die CG–Bedingungen "wörtlich" als

$$\begin{aligned} p_k &= \nabla f\left(x^{(k-1)}\right) - \frac{p_{k-1}^T \nabla^2 f\left(x^{(k-1)}\right)\, \nabla f\left(x^{(k-1)}\right)}{p_{k-1}^T \nabla^2 f\left(x^{(k-1)}\right) p_{k-1}}\, p_{k-1} \\ x^{(k)} &= x^{(k-1)} - \frac{p_k^T \nabla f\left(x^{(k-1)}\right)}{p_{k-1}^T \nabla^2 f\left(x^{(k-1)}\right) p_{k-1}}\, p_k \end{aligned}$$

umschreiben. Allerdings ist das noch nicht so ganz das, was wir wollen, denn

1. Wir brauchen hier überall die zweite Ableitung, und die muß erst einmal existieren und berechenbar sein; unserer generellen Annahmen waren ja "nur" Differenzierbarkeit und Lipschitz–Stetigkeit der Ableitung.

2. Die Schrittweite, die in der Berechnungsvorschrift für $x^{(k)}$ auftaucht, ist eine *exakte* Schrittweite, die ohnehin nur für den Fall einer quadratischen Zielfunktion Sinn macht. Im nichtlinearen "Normalfall" braucht man hier sowieso etwas anderes.

Anders gesagt: die konjugierten Gradienten spielen eigentlich zur Bestimmung der Abstiegsrichtung eine Rolle, aber solange zweite Ableitungen darin auftauchen, bleibt ihr Nutzen beschränkt. Doch das kann man glücklicherweise ändern! Wegen (5.12) und (5.15) ist nämlich

$$\begin{aligned} r_{k-1}^T p_k &= r_{k-1}^T \left(r_{k-1} - \frac{\langle r_{k-1}, p_{k-1}\rangle_A}{\langle p_{k-1}, p_{k-1}\rangle_A}\, p_{k-1} \right) = r_{k-1}^T r_{k-1} - \frac{\langle r_{k-1}, p_{k-1}\rangle_A}{\langle p_{k-1}, p_{k-1}\rangle_A} \underbrace{r_{k-1}^T p_{k-1}}_{=0} \\ &= r_{k-1}^T r_{k-1}, \end{aligned}$$

also

$$\alpha_k = -\frac{r_{k-1}^T r_{k-1}}{\langle p_k, p_k \rangle_A}$$

und da, nach (5.14) $Ap_k = \alpha_k^{-1}(r_k - r_{k-1})$ gilt, ist, wieder mit (5.15),

$$\begin{aligned} \langle r_k, p_k \rangle_A &= r_k^T A p_k = \alpha_k^{-1} r_k^T (r_k - r_{k-1}) = -\frac{\langle p_k, p_k \rangle_A}{r_{k-1}^T r_{k-1}} \left(r_k^T r_k - \underbrace{r_k^T r_{k-1}}_{=0} \right) \\ &= \langle p_k, p_k \rangle_A \frac{r_k^T r_k}{r_{k-1}^T r_{k-1}} \end{aligned}$$

und somit, als direkte Folgerung aus (5.12)

$$p_{k+1} = r_k - \frac{r_k^T r_k}{r_{k-1}^T r_{k-1}} p_k. \tag{5.17}$$

In dieser Darstellung[141] tauchen nun *keine* zweiten Ableitungen mehr auf, und wir können sie direkt in

$$p_{k+1} = \nabla f\left(x^{(k)}\right) - \frac{\nabla^T f\left(x^{(k)}\right) \nabla f\left(x^{(k)}\right)}{\nabla^T f\left(x^{(k-1)}\right) \nabla f\left(x^{(k-1)}\right)} p_k \tag{5.18}$$

umschreiben. Damit haben wir also eine "vernünftige" Wahl der Abstiegsrichtung[142] gefunden – allerdings ist nicht garantiert, daß diese konjugierten Richtungen auch wirklich Abstiegsrichtungen darstellen! Das ist nur der Fall, wenn auch die Schrittweitensteuerung geeignet ist. Trotzdem hat die Wahl (5.18) laut [36] sogar einen Namen, nämlich *Fletcher–Reeves–Methode*, siehe [18]. Eine andere Methode, die *Pollak–Ribière–Methode* bestimmt die neue Suchrichtung[143] als

$$p_{k+1} = \nabla f\left(x^{(k)}\right) - \frac{\nabla^T f\left(x^{(k)}\right) \left(\nabla f\left(x^{(k)}\right) - \nabla f\left(x^{(k-1)}\right)\right)}{\nabla^T f\left(x^{(k-1)}\right) \nabla f\left(x^{(k-1)}\right)} p_k \tag{5.19}$$

und ist nach der Tabelle in [36, S. 124] wesentlich effektiver und braucht in vielen Fällen nur die Hälfte der Iterationen oder weniger[144] als die Fletcher–Reeves–Methode. Aber um das zu verstehen, brauchen wir etwas mehr Information über die Schrittweitensteuerung.

5.6 Wahl der Schrittweite

Bleibt noch das zweite Problem, nämlich die Bestimmung der Schrittweite. Dabei möchte man gerne zwei Fliegen mit einer Klappe schlagen, nämlich:

1. Die Schrittweite soll so *klein* sein, daß eine Verbesserung erzielt wird (siehe Lemma 5.7).

[141] Die nebenbei noch effektiver und numerisch stabiler ist.

[142] Auch wenn das Sprichwort "Runter kommen sie immer" durchaus seine Gültigkeit behält, haben wir ja wohl doch gesehen, daß das "wie" eine ganz gewaltige Rolle spielen kann und wird.

[143] Nachdem Abstieg nicht sicher gewährleistet werden kann, erscheint mir dieser Begriff angemessener.

[144] Das Minimum liegt bei $\frac{1}{5}$!

2. Die Schrittweite soll so *groß* sein, daß der neue Punkt $x^{(k+1)}$ sich möglichst signifikant von $x^{(k)}$ unterscheidet, denn sonst könnte das Verfahren ja "numerisch stationär" werden.

Die "beste" Wahl wäre natürlich die *exakte* Schrittweite, das heißt für einen Punkt $x \in \mathbb{R}^n$ und eine Abstiegsrichtung y suchen wir nach der ersten Nullstelle von

$$\phi(t) = \frac{d}{dt} f\left(x + ty\right) = D_y f\left(x + ty\right).$$

Die Bestimmung dieser Nullstelle ist erstens schwierig[145] und zweitens aufwendig und kann i.a. nicht mit endlich vielen Schritten durchgeführt werden. Iterationen im Inneren von Iterationen sind immer eine äußerst problematische Angelegenheit, denn deren Verhalten beeinflusst ja auch ganz massiv die "äußere" Iteration. Deswegen verzichtet man auf die Exaktheit zugunsten der Effizienz und verwendet sogenannte *inexakte* Methoden zur Schrittweitensteuerung. Dabei fordert man nur noch, daß die Schrittweite bestimmte Bedingungen erfüllen muß. Dazu wählt man Konstanten $0 < c_1 < c_2 < 1$ und fordert, daß die Schrittweite α eines der beiden folgenden Kriterien erfüllt:

1. Die *Armijo–Bedingung*

$$f\left(x + \alpha y\right) - f\left(x\right) \leq \alpha\, c_1\, y^T \nabla f(x) \tag{5.20}$$

   fordert, daß die Verbesserung[146], die man erzielt, in etwa linear mit der Richtungsableitung geht.

2. Die *Wolfe–Bedingungen*[147] oder auch *Powell–Bedingungen*[148]

$$\begin{aligned} f\left(x + \alpha y\right) - f\left(x\right) &\leq \alpha\, c_1\, y^T \nabla f(x) \\ y^T \nabla f\left(x + \alpha y\right) &\geq c_2\, y^T \nabla f(x) \end{aligned} \tag{5.21}$$

   verlangen außerdem, daß wir bis zu einem Punkt marschieren, an dem die Funktion weniger stark abfällt – so weit sollte man schon mindestens gehen.

3. Die *starken Wolfe–Bedingungen*

$$\begin{aligned} f\left(x + \alpha y\right) - f\left(x\right) &\leq \alpha\, c_1\, y^T \nabla f(x) \\ \left|y^T \nabla f\left(x + \alpha y\right)\right| &\leq c_2\, \left|y^T \nabla f(x)\right| \end{aligned} \tag{5.22}$$

   verlangen schließlich, daß man nicht bis zu einem Punkt marschiert, an dem es zu steil "nach oben" geht.

[145] Im allgemeinen gibt es kein Verfahren, die *nächstgelegene* Nullstelle einer Funktion zu ermitteln, weder Newton, noch Bisektion oder Regula Falsi haben hier eine Chance. In Spezialfällen geht das natürlich schon, beispielsweise, wenn ϕ konvex ist, aber wer will schon *dritte* Ableitungen von f berechnen.

[146] Denn die linke Seite ist negativ!

[147] Nach [48].

[148] Nach [37].

Die erste Frage, die man sich natürlich stellt, ist, ob sich diese Forderungen überhaupt erfüllen lassen.

Lemma 5.18 *Ist* $f \in C^1(\mathbb{R}^n)$, $D_y f(x) < 0$ *und ist*

$$\inf\{\phi(t) := f(x+ty) \ : \ t \in \mathbb{R}_+\} > -\infty,$$

dann gibt es für alle $0 < c_1 < c_2 < 1$ *Werte von* $\alpha \in \mathbb{R}_+$*, die (5.21) bzw. (5.22) erfüllen*[149].

Beweis: Es sei $\ell(t) = f(x) + t\, c_1 D_y f(x)$, $t \in \mathbb{R}_+$, die lineare Funktion, die ϕ und ϕ' an der Stelle 0 interpoliert. Da $\ell(t) \to -\infty$ für $t \to \infty$ und da $\phi > M$ für ein $M \in \mathbb{R}$, gibt es einen *kleinsten* Wert $t' > 0$, so daß

$$f(x) + t'\, c_1 D_y f(x) = \ell(t') = \phi(t') = f(x+t'y)$$

und da $c_1 < 1$ ist, muß $\ell(t) \geq \phi(t)$ für alle $t \leq t'$ sein, also

$$f(x+\alpha y) - f(x) \leq \alpha\, c_1 y^T \nabla f(x), \qquad \alpha \in (0, t')$$

was nichts anderes als (5.20) ist. Nach dem Zwischenwertsatz existiert außerdem ein $t^* \in (0,t')$, so daß

$$\phi(t') - \phi(0) = (t'-0)\ \phi'(t^*)$$

ist, also

$$t'\, y^T \nabla f(x+t^* y) = f(x+t'y) - f(x) = t' \underbrace{c_1}_{<c_2} \underbrace{y^T \nabla f(x)}_{<0} > t'\, c_2\, y^T \nabla f(x).$$

Kürzen wir nun $t' > 0$, dann erhalten wir in einer Umgebung von t^* die zweite Bedingung von (5.21), aber auch von (5.22), denn die linke Seite der letzten Ungleichungskette ist ja auch negativ. □

Der Beweis von Lemma 5.18 sagt uns nicht nur, daß solche Werte von α, solche "schönen" Schrittweiten, immer existieren, er gibt uns auch ein Rezept, wie man sie berechnet! Und zwar machen wir das in zwei Schritten:

1. Wir setzen $\alpha_0 = 0$, starten mit einem (geratenen) Wert $\alpha_1 > 0$ und vergrößern ihn (z.B. durch Multiplikation mit $\rho > 1$) so lange, bis für den so erhaltenen Wert α_k eine der folgenden Bedingungen erfüllt ist.

   (a) Die Schrittweite passt:

   $$\phi(\alpha_k) - \phi(0) \leq \alpha_k\, c_1\, \phi'(0) \qquad \text{und} \qquad |\phi'(\alpha_k)| \leq c_2\, |\phi'(0)|\,.$$

   (b) Die Armijo Bedingung ist verletzt:

   $$\phi(\alpha_k) - \phi(0) > \alpha_k\, c_1\, \phi'(0).$$

[149] Und damit natürlich auch (5.20).

(c) Unsere Richtung y ist zur Aufstiegsrichtung mutiert:

$$\phi'(\alpha_k) \geq 0.$$

(d) Der letzte Punkt war besser:

$$\phi(\alpha_k) \geq \phi(\alpha_{k-1}).$$

Passiert das schon für $k = 1$, dann war α_1 idiotisch gewählt und wir halbieren α_1 so lange, bis $\phi(\alpha_1) < \phi(0)$.

Dieses Verfahren bricht irgendwann für ein k ab. Nehmen wir mal an, der erste Fall wäre nicht eingetreten, dann Setzen wir $\alpha_- = \alpha_{k-1}$, $\alpha_+ = \alpha_k$.

2. Im Intervall (α_-, α_+) liegen dann zulässige Schrittweiten, die (5.22) erfüllen und die man über ein geeignetes Bisektionsverfahren finden kann.

Der Vorteil der Wolfe- oder Powell–Bedingungen liegt nun darin, daß man mit ihnen tatsächlich auch etwas über die Konvergenz des Abstiegsverfahrens sagen kann.

Satz 5.19 *Es sei $f \in C^1(\mathbb{R}^n)$ mit Lipschitz–stetigem Gradienten*[150] *nach unten beschränkt:*

$$\inf\{f(x) \ : \ x \in \mathbb{R}^n\} > -\infty.$$

Bildet man zu einem Startwert $x^{(0)}$ die Folge

$$x^{(k+1)} = x^{(k)} + \alpha_k\, y^{(k)}, \qquad k \in \mathbb{N}_0,$$

so daß $D_{y^{(k)}} f\left(x^{(k)}\right) < 0$ und wählt man die α_k so, daß sie die Bedingung (5.21) erfüllen, dann gilt für jeden Startwert $x^{(0)}$, daß

$$\sum_{k\in\mathbb{N}_0} \cos^2\theta_k \left\|\nabla f\left(x^{(k)}\right)\right\|_2^2 < \infty, \tag{5.23}$$

wobei

$$\cos\theta_k := -\frac{\nabla^T f\left(x^{(k)}\right)\, y^{(k)}}{\|\nabla^T f\left(x^{(k)}\right)\|\ \|y^{(k)}\|}$$

Korollar 5.20 *Unter den Voraussetzungen von Satz 5.19 ist*

$$\lim_{k\to\infty} \cos\theta_k \left\|\nabla f\left(x^{(k)}\right)\right\|_2 = 0. \tag{5.24}$$

[150] Das sind also unsere "Standardbedingungen" aus dem Anfang dieses Kapitels.

Bemerkung 5.21 *Leider ist (5.24) noch nicht ganz das, was wir wollen, denn wir erhalten nicht die Konvergenz der Gradienten gegen* 0*, das heißt, die Konvergenz der* $x^{(k)}$ *gegen ein lokales Minimum, sondern wir haben das Problem, daß die Winkel* θ_k *mit ins Spiel kommen. Schaffen wir es allerdings, zu gewährleisten, daß*

$$\inf_{k\in\mathbb{N}_0} |\cos\theta_k| > 0,$$

sind also die Abstiegsrichtungen hinreichend nichtorthogonal *zu den Gradienten, dann sieht die Sache anders aus.*

Beweis von Satz 5.19: Unter Verwendung der Abkürzung $\nabla f_k := \nabla f\left(x^{(k)}\right)$ erhalten wir aus (5.21) und der Lipschitz–Stetigkeit des Gradienten, daß

$$(c_2-1)\,\nabla^T f_k\, y^{(k)} \le \left(\nabla f_{k+1}-\nabla f_k\right)^T y^{(k)} \le \gamma \underbrace{\left\|x^{(k+1)}-x^{(k)}\right\|_2}_{=\alpha_k\,\|y_k\|_2} \left\|y^{(k)}\right\|_2 = \alpha_k\,\gamma\left\|y^{(k)}\right\|_2^2,$$

also

$$\alpha_k \ge \frac{c_2-1}{\gamma}\,\frac{\nabla^T f_k\, y^{(k)}}{\|y^{(k)}\|_2^2} > 0,$$

letzteres, da $c_2 < 1$ und $\nabla^T f_k\, y^{(k)} < 0$. Setzen wir das in die erste Ungleichung von (5.21) ein, dann ergibt sich, daß

$$\begin{aligned} f\left(x^{(k+1)}\right) &= f\left(x^{(k)}+\alpha_k\, y^{(k)}\right) \le f\left(x^{(k)}\right) - \frac{c_1\,(1-c_2)}{\gamma} \underbrace{\frac{\left(\nabla^T f_k\, y^{(k)}\right)^2}{\|y^{(k)}\|_2^2}}_{=\cos^2\theta_k\,\left\|\nabla f\left(x^{(k)}\right)\right\|_2^2} \\ &= f\left(x^{(k)}\right) - \frac{c_1\,(1-c_2)}{\gamma}\left(\cos^2\theta_k\,\left\|\nabla f\left(x^{(k)}\right)\right\|_2^2\right) \\ &= f\left(x^{(0)}\right) - \frac{c_1\,(1-c_2)}{\gamma}\sum_{j=0}^{k}\cos^2\theta_k\,\left\|\nabla f\left(x^{(k)}\right)\right\|_2^2 \end{aligned}$$

und da f nach unten beschränkt ist folgt (5.23). □

5.7 Nochmal konjugierte Gradienten

Man kann zeigen[151], daß mit den starken Wolfe-Bedingungen (5.22) und $0 < c_1 < c_2 < \frac{1}{2}$ das Verfahren von Fletcher & Reeves "konvergiert", genauer, daß

$$\liminf_{k\to\infty}\left\|\nabla f\left(x^{(k)}\right)\right\|_2 = 0$$

[151] Siehe [36, Theorem 5.8, S. 128].

ist. Trotzdem kann es zu Schwierigkeiten kommen, die bei Pollak–Ribière nicht auftreten. Dazu bemerkt man, daß für $c_2 < \frac{1}{2}$ und der Iterationsvorschrift (5.18) die Abschätzungen[152]

$$\underbrace{\frac{1-2c_2}{1-c_2}}_{=:a_1>0} \leq -\frac{\nabla^T f\left(x^{(k)}\right) y^{(k)}}{\|\nabla^T f\left(x^{(k)}\right)\|_2^2} \leq \underbrace{\frac{1}{1-c_2}}_{=:a_2>0}$$

gelten, also

$$a_1 \frac{\left\|\nabla^T f\left(x^{(k)}\right)\right\|_2}{\left\|y^{(k)}\right\|_2} \leq \cos\theta_k \leq a_2 \frac{\left\|\nabla^T f\left(x^{(k)}\right)\right\|_2}{\left\|y^{(k)}\right\|_2}.$$

Damit heißt der "schlechte" Fall $\cos\theta_k \sim 0$, daß $\left\|\nabla f\left(x^{(k)}\right)\right\|_2 \ll \left\|y^{(k)}\right\|_2$; dann ist aber auch, wie im Beweis von Satz 5.19, $x^{(k+1)} \sim x^{(k)}$, also auch $\nabla f\left(x^{(k+1)}\right) \sim \nabla f\left(x^{(k)}\right)$ und damit $\alpha_{k+1} \sim 1$ und somit, nach (5.18),

$$y^{(k+1)} \sim \nabla f\left(x^{(k+1)}\right) + y^{(k)} \sim y^{(k)},$$

der Algorithmus läuft sich also fest! Und hier ist der Vorteil von (5.19): Sind bei zwei aufeinanderfolgenden Iterationsschritten die Gradienten (nahezu) gleich, dann wird die konjugierte Richtung verworfen und das Verfahren mit dem steilsten Abstieg neu gestartet – in den meisten Fällen eine gute Wahl.

Zum Abschluß aber noch ein nettes theoretisches Ergebnis ohne Beweis.

Satz 5.22 *Es gibt eine Funktion $f \in C^2(\mathbb{R}^3)$ und einen Startwert $x^{(0)} \in \mathbb{R}^3$, so daß für die Pollak–Ribière–Methode mit* exakter *Schrittweitenbestimmung*

$$\inf\left\{\left\|\nabla f\left(x^{(k)}\right)\right\|_2 \,:\, k \in \mathbb{N}_0\right\} > 0$$

ist.

[152] Siehe [36, S 125].

Noch hat der Name Philosophie bei den Engländern allgemein diese Bestimmung, Newton hat fortdauernd den Ruhm des größten Philosophen; bis in die Preiskurante der Instrumentenmacher herab heißen diejenigen Instrumente, die nicht unter eine besondere Rubrik magnetischen, elektrischen Apparats gebracht werden, die Thermometer, Barometer usf. philosophische Instrumente; freilich sollte nicht eine Zusammensetzung von Holz, Eisen usf., sondern allein das Denken das Instrument der Philosophie genannt werden.

Georg Wilhelm Friderich Hegel,
Enzyklopädie der philosophischen Wissenschaften im Grundrisse

6 Newton–Verfahren und Variationen

Da sich lokale Extrema x^* einer Funktion $f \in C^1(\mathbb{R}^n)$ ja dadurch auszeichnen, daß $\nabla f(x^*) = 0$, können wir also unser Optimierungsproblem auch als die Suche nach einer Nullstelle von $F(x) = \nabla f(x)$, $F : \mathbb{R}^n \to \mathbb{R}^n$ auffassen. Zur Erinnerung: diesen Ansatz haben wir ja schon bei den primal–dualen Optimierungsproblemen, siehe (4.44) auf Seite 102, verwendet. Dazu zuerst einmal der nötige Formalismus.

Definition 6.1 *Sei $F = [F_j \ : \ j = 1, \ldots, n] \in C(\mathbb{R}^n)^n$ ein Vektorfeld. Die* Jacobimatrix *zu F ist definiert als*

$$F' := J[F] := \left[\frac{\partial F_j}{\partial x_k} \ : \ j,k = 1, \ldots, n \right], \qquad F \in C^1(\mathbb{R}^n)^n.$$

Besonders einfach ist die Sache natürlich, wenn $F = \nabla f$, $f \in C^2(\mathbb{R}^n)$, denn dann ist

$$F'_{jk} = \frac{\partial}{\partial x_k} F_j = \frac{\partial}{\partial x_k} \frac{\partial f}{\partial x_j} = \frac{\partial^2 f}{\partial x_j \partial x_k} = \nabla^2_{jk} f, \qquad j,k = 1, \ldots, n,$$

also

$$F' = J[F] = J[\nabla f] = \nabla^2 f. \tag{6.1}$$

6.1 Das Newton–Verfahren und das Broyden–Verfahren

Zu einer Funktion $F \in C^1(\mathbb{R})$ erzeugt das Newton–Verfahren vermittels der Iterationsvorschrift

$$x^{(k+1)} = x^{(k)} - \frac{F\left(x^{(k)}\right)}{F'\left(x^{(k)}\right)}, \qquad k \in \mathbb{N}_0, \tag{6.2}$$

eine Folge $\left\{x^{(k)} \,:\, k \in \mathbb{N}\right\}$ von Punkten, die für einen günstig gewählten Startwert $x^{(0)}$ auch tatsächlich gegen eine *einfache* Nullstelle x^* mit

$$F\left(x^*\right) = 0, \qquad F'\left(x^*\right) \neq 0,$$

konvergiert. Für $F \in C^1\left(\mathbb{R}^n\right)^n$ verwendet man hingegen die Iteration

$$x^{(k+1)} = x^{(k)} - \left(F'\right)^{-1}\left(x^{(k)}\right) F\left(x^{(k)}\right), \qquad k \in \mathbb{N}_0, \tag{6.3}$$

allerdings ist jetzt der Begriff der "einfachen" Nullstelle etwas präziser zu fassen, es muß nämlich

$$F\left(x^*\right) = 0, \qquad \det F'\left(x^*\right) \neq 0$$

sein, die Matrix $F'\left(x^*\right)$ muß also *invertierbar* sein. Das führt dann auch dazu, daß für alle $0 \neq y \in \mathbb{R}^n$ die *vektorwertige* Richtungsableitung $D_y F = F'\, y$ von Null verschieden ist. Eine "typische" allgemeine Konvergenzaussage für das Newton–Verfahren sieht dann wie folgt aus[153].

Satz 6.2 *Ist $F \in C^2\left(\mathbb{R}^n\right)^n$ und ist $x^* \in \mathbb{R}^n$ eine einfache Nullstelle von F, das heißt,*

$$F\left(x^*\right) = 0 \qquad \textit{und} \qquad \det\, J[F]\left(x^*\right) \neq 0, \tag{6.4}$$

dann gibt es eine offene Menge $U \subset \mathbb{R}^n$, $x^* \in U$, so daß

$$x^{(0)} \in U \qquad \Longrightarrow \qquad \lim_{k\to\infty} x^{(k)} = x^*.$$

Nur zur Erinnerung: Iterationsverfahren, die für Startwerte konvergieren, die hinreichend nahe bei der gesuchten Lösung liegen (und dann aber auch gegen diese "naheliegende" Lösung!), bezeichnet man als *lokal konvergent*.

Neben den offensichtlichen Schwierigkeiten der nur lokalen Konvergenz gibt es noch ein weiteres Problem, das die praktische Anwendung des Newton–Verfahrens schwierig macht: die Bestimmung der Jacobimatrix $J[F]$! Schließlich kann man nicht unbedingt davon ausgehen, daß die Ableitungen aller Komponenten von F auch wirklich so einfach verfügbar sind. Möglichkeiten zur praktischen Bestimmung der Jacobimatrix wären

[153] Aus [42], dort findet sich auch der Beweis, der auf Fixpunktiterationen und dem Banachschen Fixpunktsatz beruht.

Automatische Differentiation: (siehe z.B. [36, Chapter 7.2]) Funktionen werden intern als Kombination elementarer Funktionen dargestellt, deren Ableitungen bei der Auswertung mitberechnet werden:

$$f = gh \qquad \Longrightarrow \qquad \nabla f = g\nabla h + h\nabla g$$

oder

$$f = g\left(h_1, \ldots, h_m\right) \qquad \Longrightarrow \qquad \nabla f = \nabla g\, J[H], \quad H = \left[h_j \;:\; j = 1, \ldots, m\right].$$

Solche Schemata lassen sich sehr gut in C++ implementieren.

Numerische Differentiation: Man erhält die näherungsweisen Ableitungen durch Differenzenquotienten oder durch Differentiation von Interpolationspolynomen. So kann man für Punkte $\{x_k \in \mathbb{R}^n \;:\; k = 1, \ldots, N\}$ ein Polynom p bestimmen, so daß $f\left(x_j\right) = p\left(x_j\right)$ und dann $\nabla p(x)$ bestimmen. Der Differenzenquotient ist hierbei nur der Spezialfall $N = 2$. Allerdings ist das mit der Interpolation in mehreren Variablen nicht mehr so ganz einfach, siehe [19].

Trotzdem, beide Methoden zur Bestimmung von Ableitungen, insbesondere von höheren Ableitungen sind aufwendig und numerisch nicht immer stabil. Deswegen versucht man, die Bestimmung von Gradienten so weit es geht zu vermeiden. Das führt zu einer wichtigen Variante des Newton–Verfahrens, zum sogenannten *Broyden–Verfahren*, [6]. Dabei wird $F'\left(x^{(k)}\right)$ durch eine Matrix B_k angenähert und anstatt dann im nächsten Schritt die Matrix $F'\left(x^{(k+1)}\right)$ zu bestimmen, bestimmt man einen *näherungsweisen* "Update" B_{k+1}, der nur von der Richtung $y^{(k)}$ und den Werten von F abhängt. Außerdem spendiert man sich wieder eine Schrittweitensteuerung $\alpha_k \in \mathbb{R}_+$. Insgesamt sieht das Ganze dann folgendermaßen aus:

$$\begin{aligned} y &= B_k^{-1}\, F\left(x^{(k)}\right) \\ x^{(k+1)} &= x^{(k)} - \alpha_k\, y, \qquad d = F\left(x^{(k+1)}\right) - F\left(x^{(k)}\right) \\ B_{k+1} &= B_k + \frac{1}{y^T y}\left(d - B_k y\right)\, y^T. \end{aligned} \tag{6.5}$$

Die Transpositionszeichen in der Update–Regel für B_{k+1} sind hierbei schon korrekt: Der Übergang von B_k zu B_{k+1} erfolgt durch Addition einer Matrix der Form zy^T, also einer Matrix *vom Rang* 1. Laut [46, S. 248] führt man die Iterationen nach der Regel (6.6) aus, wenn die Schrittweite α_k, die durch (näherungsweise, numerische) Lösung des Minimierungsproblems[154]

$$\left\|F\left(x^{(k)} - \alpha_k\, y\right)\right\|_2^2 = \min_{t \geq 0} \left\|F\left(x^{(k)} - t\, y\right)\right\|_2^2$$

ermittelt wurde, die Bedingung $\frac{1}{2} \leq \alpha_k \leq 1$ erfüllt, ansonsten berechnet man zähneknirschend $B_{k+1} \sim F'\left(x^{(k+1)}\right)$, entweder numerisch oder mit automatischer Differentiation.
Der Grund für die Iterationsvorschrift (6.5) liegt im folgenden Modellproblem, ein Resultat das auf Broyden [6] zurückgeht.

[154] Preisfrage: Warum steht hier das Quadrat? Wer es immer noch nicht verstanden hat: *Gehe zurück zum Anfang des Skripts, gehe nicht über Los, ziehe keinen Schein ein.*

Proposition 6.3 *Sei $F(x) = Ax + b$, $A \in \mathbb{R}^{n\times n}$ und $B \in \mathbb{R}^{n\times n}$ eine beliebige Matrix. Für beliebige $x, x' \in \mathbb{R}^n$ sei $y := x - x'$ und $d := F(x) - F(x') = Ay$*

$$B' := B + \frac{1}{y^T y}\,(d - By)\,y^T.$$

Dann ist

$$\|B' - A\|_2 \le \|B - A\|_2\,.$$

Beweis: Da $d = Ay$, ist

$$B' = B + (A - B)\,\frac{yy^T}{y^T y} = A\frac{yy^T}{y^T y} + B\left(I - \frac{yy^T}{y^T y}\right),$$

also

$$B' - A = (B - A)\,\left(I - \frac{yy^T}{y^T y}\right).$$

Schreiben wir nun einen beliebigen Vektor $u \in \mathbb{R}^n$ als $u = y + z$, $y \perp z$, dann ist

$$(B' - A)\,u = (B - A)\left(z - \frac{1}{y^T y}y\underbrace{y^T z}_{=0}\right) + (B - A)\overbrace{\left(y - \underbrace{y\frac{1}{y^T y}y^T y}_{=1}\right)}^{=0} = (B - A)\,z$$

und da $\|u\|_2^2 = \|y\|_2^2 + \|z\|_2^2$ wegen der Orthogonalität, ergibt sich, daß

$$\|(B' - A)\,u\|_2 = \|(B - A)\,z\|_2 \le \|B - A\|_2\,\|z\|_2 \le \|B - A\|_2\,\|u\|_2\,,$$

also $\|B' - A\|_2 \le \|B - A\|_2$. □

6.2 Das Newton–Verfahren zur Minimumsbestimmung

Beschäftigen wir uns aber nun mit unserem "Spezialfall" $F = \nabla f$ und der Iterationsvorschrift

$$x^{(k+1)} = x^{(k)} - \left(\nabla^2 f\left(x^{(k)}\right)\right)^{-1}\,\nabla f\left(x^{(k)}\right) =: x^{(k)} + \alpha_k\,y^{(k)}, \qquad \alpha_k = -1, \tag{6.6}$$

wobei man die wie in (6.6) gewählte Richtung $y^{(k)}$ als *Newton–Richtung* bezeichnet. Die Newton–Richtung muß übrigens keine Abstiegsrichtung sein. Trotzdem kann man nun Aussagen über (lokale) Konvergenz und Konvergenzgeschwindigkeit machen.

Satz 6.4 *Sei $f \in C^2(\mathbb{R}^n)$ und $\nabla^2 f$ Lipschitz–stetig in einer Umgebung eines* strikten *Minimums x^* von f. Dann gibt es eine Umgebung U von x^*, so daß die Iteration (6.6) für alle $x^{(0)} \in U$*

1. *gegen x^* konvergiert:*

$$\lim_{k\to\infty} x^{(k)} = x^*. \tag{6.7}$$

2. quadratisch *konvergiert:*

$$\sup_{k\in\mathbb{N}_0} \frac{\left\|x^{(k+1)} - x^*\right\|}{\left\|x^{(k)} - x^*\right\|^2} < \infty. \tag{6.8}$$

3. *quadratisch gegen Null konvergente Gradienten liefert:*

$$\sup_{k\in\mathbb{N}_0} \frac{\left\|\nabla f\left(x^{(k+1)}\right)\right\|}{\left\|\nabla f\left(x^{(k)}\right)\right\|^2} < \infty. \tag{6.9}$$

Beweis: Wir setzen wieder $\nabla f_k = \nabla f\left(x^{(k)}\right)$ und entsprechend auch $\nabla^2 f_k$ und ∇f_*. Nach (6.6) ist

$$\begin{aligned}
x^{(k+1)} - x^* &= x^{(k)} - x^* - \left(\nabla^2 f_k\right)^{-1} \nabla f_k \\
&= \left(\nabla^2 f_k\right)^{-1} \left(\nabla^2 f_k \left(x^{(k)} - x^*\right) - \nabla f_k + \underbrace{\nabla f_*}_{=0}\right) \\
&= \left(\nabla^2 f_k\right)^{-1} \left(\nabla^2 f_k \left(x^{(k)} - x^*\right) - \int_0^1 \nabla^2 f\left(x^* + t\left(x^{(k)} - x^*\right)\right)\left(x^{(k)} - x^*\right)\, dt\right) \\
&= \left(\nabla^2 f_k\right)^{-1} \left(\nabla^2 f_k - \int_0^1 \nabla^2 f\left(x^* + t\left(x^{(k)} - x^*\right)\right)\, dt\right)\left(x^{(k)} - x^*\right)
\end{aligned}$$

Nun gibt es ein $\delta > 0$ und ein $C > 0$, so daß für alle $x \in \mathbb{R}^n$ mit $\|x - x^*\| \le \delta$ die Abschätzungen

$$\left\|\nabla^2 f(x)\right\| \le C\left\|\nabla^2 f_*\right\| \qquad \text{und} \qquad \left\|\nabla^2 f(x)^{-1}\right\| \le C\left\|\nabla^2 f_*^{-1}\right\|$$

gelten. Angenommen, $\left\|x^{(k)} - x^*\right\| \le \delta$, dann ist

$$\begin{aligned}
&\left\|\nabla^2 f_k - \int_0^1 \nabla^2 f\left(x^* + t\left(x^{(k)} - x^*\right)\right)\, dt\right\| \\
&\quad= \left\|\int_0^1 \nabla^2 f_k - \nabla^2 f\left(x^* + t\left(x^{(k)} - x^*\right)\right)\, dt\right\| \\
&\quad\le \int_0^1 \underbrace{\left\|\nabla^2 f_k - \nabla^2 f\left(x^* + t\left(x^{(k)} - x^*\right)\right)\right\|}_{\le\gamma\left\|x^{(k)} - x^*\right\|}\, dt \le \gamma\left\|x^{(k)} - x^*\right\|,
\end{aligned}$$

also

$$\begin{aligned}
\left\|x^{(k+1)} - x^*\right\| &\le \left\|\nabla^2 f_k^{-1}\right\| \left\|\nabla^2 f_k - \int_0^1 \nabla^2 f_t\, dt\right\| \left\|x^{(k)} - x^*\right\| \\
&\le C\gamma \left\|\nabla^2 f_*^{-1}\right\| \left\|x^{(k)} - x^*\right\|^2.
\end{aligned}$$

Ist nun[155] $\left\|x^{(k)} - x^*\right\| \le \left(C\gamma \left\|\nabla^2 f_*^{-1}\right\|\right)^{-1}$, dann gilt das auch für $x^{(k+1)}$ und die Iteration bleibt in der "guten" Umgebung, woraus (6.8) und somit auch (6.7) folgen. Unter Verwendung

[155] Hier nehmen wir an, daß die Konstante $C\gamma \left\|\nabla^2 f_*^{-1}\right\| \ge 1$ ist, ansonsten wäre alles nur noch einfacher.

von $y^{(k)} = x^{(k+1)} - x^{(k)} = -\nabla^2 f_k^{-1} \nabla f_k$ ergibt sich dann auch

$$\begin{aligned}
\|\nabla f_{k+1}\| &= \left\|\nabla f_{k+1} - \nabla f_k + \nabla^2 f_k \left(\nabla^2 f_k^{-1} \nabla f_k\right)\right\| \\
&= \left\| \int_0^1 \nabla^2 f\left(x^{(k)} + t y^{(k)}\right) y^{(k)}\, dt - \nabla^2 f_k\, y^{(k)} \right\| \\
&\leq \left\| \int_0^1 \nabla^2 f\left(x^{(k)} + t y^{(k)}\right) - \nabla^2 f_k\, dt \right\| \left\|y^{(k)}\right\| \leq \gamma \left\|y^{(k)}\right\|^2 \leq \gamma \left\|\nabla^2 f_k^{-1}\right\|^2 \|\nabla f_k\|^2 \\
&\leq \gamma\, C^2 \left\|\nabla^2 f_*^{-1}\right\|^2 \|\nabla f_k\|^2,
\end{aligned}$$

was (6.9) liefert. □

6.3 Quasi–Newton–Verfahren

Wie wir also in Satz 6.4 gesehen haben, ist das Newton–Verfahren ein "gutes" Verfahren in dem Sinn, daß es lokal und *schnell* gegen eine lokale Minimalstelle konvergiert, was uns natürlich nicht von der Schwierigkeit befreit, so einen Startwert zu finden. Was in solchen Fällen oftmals hilft, ist ein sogenanntes *Hybridverfahren*, bei dem Abstiegsmethode und Newton–Verfahren im Wechsel ausgeführt werden (beispielsweise ein Schritt Abstieg, dann mehrere Schritte Newton), in der Hoffnung, daß das Abstiegsverfahren dafür sorgt, daß man nahe genug an das Minimum kommt, bis dann letztendlich das Newton–Verfahren "greift". Solche Verfahren sind zwar heuristisch naheliegend und auch gut motiviert, aber mathematisch nicht so schön zu untersuchen.

Wir wollen hier wieder die Idee des Broyden–Verfahrens ins Spiel bringen, also die Frage, wie man die Berechnung von $\nabla^2 f_k$ nach Möglichkeit vermeiden kann. Dazu betrachten wir im k–ten Schritt an der Stelle $x^{(k)}$ das *quadratische Modell*

$$\underbrace{f\left(x^{(k)} + y\right)}_{} \sim \underbrace{f\left(x^{(k)}\right)}_{f_k} + \nabla^T f_k\, y + \frac{1}{2} y^T \nabla^2 f_k\, y \sim f_k + \nabla^T f_k\, y + \frac{1}{2} y^T B_k y =: \widehat{f}_k(y), \quad (6.10)$$

wobei $B_k \in \mathbb{R}^{n\times n}$ eine *symmetrische, positiv definite*[156] Matrix und Näherung von $\nabla^2 f_k$ sein soll. Dann setzen wir wieder einmal

$$x^{(k+1)} = x^{(k)} + \alpha_k\, y^{(k)}, \qquad y^{(k)} = -B_k^{-1} \nabla f_k, \qquad \alpha_k \in \mathbb{R}_+,$$

und stellen uns anhand des "Modells"

$$\widehat{f}_{k+1}(y) = f_{k+1} + \nabla^T f_{k+1}\, y + \frac{1}{2} y^T B_{k+1} y$$

die Frage, wie man nun B_{k+1} wählen sollte. Erinnern wir uns daran, daß wir eine Nullstelle von ∇f berechnen wollen, dann könnten wir beispielsweise fordern, daß die lineare Näherung B_{k+1} der Ableitung $\nabla^2 f$ von ∇f zumindest die *Sekantenbedingung*

$$B_{k+1}\left(x^{(k+1)} - x^{(k)}\right) = \nabla f_{k+1} - \nabla f_k \quad (6.11)$$

[156] Wir nehmen also an, wir wären schon nahe genug an einem *strikten* lokalen Minimum.

erfüllt. Für $n = 1$ ist das das altbekannte *Sekantenverfahren*[157] für f', für $n > 1$ reicht das aber natürlich nicht aus, um die Matrix B_{k+1} komplett festzulegen. Multiplikation von links mit $\left(x^{(k+1)} - x^{(k)}\right)^T$ ergibt, zusammen mit der positiven Definitheit von B_{k+1}, daß die Sekantenbedingung nur dann erfüllbar ist, wenn die *Krümmungsbedingung*

$$\left(x^{(k+1)} - x^{(k)}\right)^T \left(\nabla f_{k+1} - \nabla f_k\right) > 0 \tag{6.12}$$

erfüllt ist, was sich mit der "richtigen" Schrittweitensteuerung erreichen läßt. In der Tat liefert die zweite Wolfe–Bedingung aus (5.21) mit $y = \alpha_k^{-1}\left(x^{(k+1)} - x^{(k)}\right)$, daß

$$\begin{aligned} 0 &\leq \alpha_k^{-1}\left(x^{(k+1)} - x^{(k)}\right)^T \left(\nabla f_{k+1} - c_2 \nabla f_k\right) \\ &= \frac{c_2}{\alpha_k}\left(x^{(k+1)} - x^{(k)}\right)^T \left(\nabla f_{k+1} - \nabla f_k\right) + \frac{1-c_2}{\alpha_k} \underbrace{\left(x^{(k+1)} - x^{(k)}\right)^T \nabla f_{k+1}}_{\leq 0} \\ &\leq \frac{c_2}{\alpha_k}\left(x^{(k+1)} - x^{(k)}\right)^T \left(\nabla f_{k+1} - \nabla f_k\right), \end{aligned}$$

woraus (6.12) folgt, da $y^{(k)}$ eine Abstiegsrichtung[158] und somit auch $\alpha_k > 0$ ist. Nun bestimmt die Sekantenbedingung (6.11) aber natürlich die Matrix B_{k+1} nicht vollständig, weswegen man sie wieder einmal als Lösung eines Minimierungsproblems definieren kann, beispielsweise

$$B_{k+1} = \min_B \|B - B_k\|, \qquad B = B^T, \quad B\left(x^{(k+1)} - x^{(k)}\right) = \nabla f_{k+1} - \nabla f_k, \tag{6.13}$$

wobei $\|\cdot\|$ eine *beliebige* Matrixnorm sein kann – und in der Tat liefert verschiedene Normen auch verschiedene *Quasi–Newton–Verfahren*, wie man diese Familie von Iterationsverfahren auch nennt. Eine beliebte Wahl sind *gewichtete Frobenius*[159]*–Normen* der Form

$$\|A\|_W = \left\|W^{1/2} A W^{1/2}\right\|_F, \qquad \|A\|_F^2 = \text{trace}\left(A^T A\right) = \sum_{j,k=1}^{n} a_{jk}^2, \tag{6.14}$$

wobei die "Gewichtsmatrix" $W \in \mathbb{R}^{n\times n}$ symmetrisch und positiv semidefinit sein muß, denn dann ist die (positive) Wurzel $W^{1/2}$ wohldefiniert als diejenige symmetrische, positiv definite Matrix B, die $B^2 = W$ erfüllt.

Übung 6.1 Zeigen Sie:

[157] Zumal alle 1×1–Matrizen immer und automatisch symmetrisch sind.

[158] Schließlich ist ja $B_k y^{(k)} = \nabla f_k$ und im quadratischen Fall ist das sogar der *direkte* Weg zum Minimum.

[159] Ferdinand Georg Frobenius, 1849–1917, promovierte bei Weierstrass und wurde 1874 *ohne Habilitation* in Berlin zum Professor ernannt. Wichtige Beiträge zur Darstellungstheorie von Gruppen (hat ja auch einiges mit Matrizen zu tun), insbesondere Entwicklung der Charakteren–Theorie. Eine interessante Bemerkung über Frobenius ist:

> *For Frobenius, conceptual argumentation played a somewhat secondary role. Although he argued in a comparatively abstract setting, abstraction was not an end in itself.*

Darüberhinaus konnte er den "neuen mathematischen Stil" aus Göttingen (verkörpert durch Klein und Lie) ganz und gar nicht ausstehen ...

1. Zu jeder symmetrischen positiv semidefiniten Matrix W gibt es eine eindeutige symmetrische positiv semidefinite Matrix $B = W^{1/2}$, so daß $W = B^2$ ist.

2. Für $A \in \mathbb{R}^{n\times n}$ ist
$$\text{trace}\left(A^T A\right) = \sum_{j,k=1}^{n} a_{jk}^2.$$

♢

Bei "geeigneter" Wahl von W (als "gemittelte" Hessematrix) so daß $W^{-1}\xi_k = \eta_k$, mit $\xi_k := x^{(k+1)} - x^{(k)}$ und $\eta_k := \nabla f_{k+1} - \nabla f_k$, ergibt sich dann die folgende Update–Regel aus [36, S. 196]
$$B_{k+1} = \left(I - \frac{\eta_k\,\xi_k^T}{\eta_k^T\xi_k}\right) B_k \left(I - \frac{\xi_k\,\eta_k^T}{\eta_k^T\xi_k}\right) + \frac{\eta_k\eta_k^T}{\eta_k^T\xi_k}, \tag{6.15}$$
die 1959 von Davidon vorgeschlagen [13, 14], aber vor allem von Fletcher und Powell (unabhängig) untersucht und popularisiert wurde, weswegen man sie als *DFP–Methode* bezeichnet.

Anstelle mit B_k zu rechnen und in jedem Iterationsschritt das Gleichungssystem $B_k y^{(k)} = -\nabla f_k$ lösen zu müssen, kann man auch *direkt* mit $B_k^{-1} := H_k$ rechnen und nun das Minimierungsproblem
$$H_{k+1} = \min_H \|H - H_k\|, \qquad H = H^T, \quad H\left(\nabla f_{k+1} - \nabla f_k\right) = x^{(k+1)} - x^{(k)}, \tag{6.16}$$
lösen, was zur Update–Regel
$$H_{k+1} = \left(I - \frac{\xi_k\,\eta_k^T}{\eta_k^T\xi_k}\right) H_k \left(I - \frac{\eta_k\,\xi_k^T}{\eta_k^T\xi_k}\right) + \frac{\xi_k\xi_k^T}{\eta_k^T\xi_k} \tag{6.17}$$
führt, in der, wegen des Übergangs zur Inversen, die Rollen von ξ_k und η_k vertauscht sind. Das damit verbundene Verfahren bezeichnet man nach seinen "Vätern" Broyden, Fletcher, Goldfarb und Shanno[160] als *BFGS–Verfahren*.

Man kann nun auch, ausgehend von (6.15) eine "inverse" Regel für die Updates der entsprechenden H_k im DFP–Verfahren, beziehungsweise, ausgehend von (6.17), eine "primäre" Regel zur Bestimmung von B_{k+1} aus B_k für das BFGS–Verfahren aufstellen. Das geht ganz einfach unter Verwendung des folgenden Resultats.

Lemma 6.5 *("Sherman–Morrison–Woodbury–Formel")*[161]
Für eine nichtsinguläre Matrix $A \in \mathbb{R}^{n\times n}$ und $x, y \in \mathbb{R}^n$ ist
$$\left(A + xy^T\right)^{-1} = A^{-1} - \frac{A^{-1}xy^TA^{-1}}{1 + y^TA^{-1}x}. \tag{6.18}$$

[160] Das Literaturverzeichnis von [36] legt nahe, daß sie das Verfahren nicht gemeinsam sondern aufeinander aufbauend oder in Konkurrenz oder unabhängig oder wie auch immer entwickelt haben.

[161] Normalerweise bin ich sehr skeptisch bei allem, was den Namen von mehr als zwei Personen trägt ...

Übung 6.2 Beweisen Sie (6.18) (durch Ausmultiplizieren)[162] und charakterisieren Sie, wann $A + xy^T$ invertierbar ist. ♢

Übung 6.3 Zeigen Sie, daß die zu (6.17) äquivalente Update–Regel sich als

$$H_{k+1} = H_k - \frac{H_k \xi_k \, \xi_k^T H_k^T}{\xi_k^T H_k \xi_k} + \frac{\eta_k \eta_k^T}{\xi_k^T \eta_k} \tag{6.19}$$

schreiben läßt. ♢

Wir wollen uns nun aber lieber mit *Konvergenzeigenschaften* des BFGS–Verfahrens befassen, die man unter gewissen (lokalen) Bedingungen auch tatsächlich beweisen kann.

Satz 6.6 *Sei $f \in C^2(\mathbb{R}^n)$ und sei $x^{(0)}$ so gewählt, daß*

$$\Omega := \left\{ x \in \mathbb{R}^n \ : \ f(x) \leq f\left(x^{(0)}\right) \right\}$$

konvex ist und es Konstanten $0 < m < M$ gibt, so daß

$$m \left\| y \right\|_2^2 \leq y^T \nabla^2 f(x) \, y \leq M \left\| y \right\|_2^2, \qquad x \in \Omega. \tag{6.20}$$

Dann konvergiert die Folge

$$x^{(k+1)} := x^{(k)} + \alpha_k \, y^{(k)}, \qquad y^{(k)} = -H_k \nabla f_k, \qquad k \in \mathbb{N}_0,$$

unter Beachtung der Wolfe– oder Powell–Bedingungen und unter Verwendung der Update–Regel (6.17) gegen ein Minimum x^* von f.

Bemerkung 6.7 *Die Bedingung (6.20) bedeutet, daß die Funktion f auf der gesamten Niveaumenge* gleichmäßig strikt *konvex (oder auch* stark konvex*) ist. Das ist natürlich ziemlich viel verlangt und eine Bedingung für ein* striktes lokales *Minimum, das, wenn man den Einstiegswert niedrig genug wählt, dann aber auch gefunden wird.*

Für den Beweis von Satz 6.6 brauchen wir zuerst eine kleine Hilfsaussage aus der linearen Algebra.

Lemma 6.8 *Seien $x, y, u, v \in \mathbb{R}^n$. Dann ist*

$$\det\left(I + uv^T + xy^T\right) = \left(1 + u^T v\right)\left(1 + x^T y\right) - \left(v^T x\right)\left(u^T y\right) \tag{6.21}$$

Beweis: Beginnen wir mit dem Fall $x = 0$ oder $y = 0$. Dazu seien $w_2, \ldots, w_n$ linear unabhängige Vektoren, die senkrecht auf v stehen, dann ist

$$\left(I + uv^T\right) w_j = w_j + u \underbrace{v^T w_j}_{=0} = w_j =: \lambda_j w_j, \qquad j = 2, \ldots, n,$$

[162] Drei Namen und ein fast trivialer Beweis.

sowie

$$\left(I + uv^T\right) u = u + \left(v^T u\right) u = \left(1 + u^T v\right) u =: \lambda_1 u,$$

womit wie alle Eigenwerte und Eigenvektoren identifiziert haben und da die Determinante das Produkt der Eigenvektoren ist, erhalten wir, daß

$$\det\left(I + uv^T\right) = \prod_{j=1}^{n} \lambda_j = 1 + u^T v.$$

Nun nehmen wir an, daß v und y linear unabhängig sind, denn ansonsten könnten wir das auf den einfachen Fall zurückführen, den wir gerade erledigt haben. Nun wählen wir $w_3, \ldots, w_n$ senkrecht zu v und y, was uns sofort

$$\left(I + uv^T + xy^T\right)\, w_j = w_j, \qquad j = 3, \ldots, n$$

liefert und da

$$\begin{aligned} \left(I + uv^T + xy^T\right) x &= \left(1 + x^T y\right) x + \left(v^T x\right) u \\ \left(I + uv^T + xy^T\right) u &= \left(u^T y\right) x + \left(1 + v^T u\right) u \end{aligned}$$

ist, ergibt sich für die ersten beiden Eigenwerte λ_1 und λ_2, daß

$$\lambda_1\, \lambda_2 = \det \begin{bmatrix} 1 + x^T y & v^T x \\ u^T y & 1 + u^T v \end{bmatrix} = \left(1 + x^T y\right)\left(1 + u^T v\right) - \left(v^T x\right)\left(u^T y\right),$$

woraus (6.21) unmittelbar folgt. □

Beweis von Satz 6.6: Wie die Verwendung der Wolfe–Bedingungen ja nahelegt, wollen wir Satz 5.19 verwenden – zu diesem Zweck müssen wir aber die Winkel zwischen Gradienten und Abstiegsrichtungen in den Griff bekommen.

Da

$$\eta_k = \nabla f_{k+1} - \nabla f_k = \int_0^1 \nabla\left(\nabla f\right) \underbrace{\left(x^{(k)} + t\xi_k\right)}_{=:x_t} \xi_k\, dt = \int_0^1 \nabla^2 f\left(x_t\right)\, \xi_k\, dt =: G_k\, \xi_k,$$

ist wegen der Annahme (6.20)

$$\xi_k^T \eta_k = \int_0^1 \underbrace{\xi_k^T\, \nabla^2 f\left(x_t\right)\, \xi_k}_{\geq m\|\xi_k\|_2^2}\, dt \geq m\, \|\xi_k\|_2^2 \qquad \Longrightarrow \qquad m_k := \frac{\xi_k^T \eta_k}{\xi_k^T \xi_k} \geq m,$$

sowie

$$M_k := \frac{\eta_k^T \eta_k}{\eta_k^T \xi_k} = \frac{\xi^T G_k^2 \xi}{\xi^T G_k \xi} = \frac{\left(\sqrt{G_k}\xi\right)^T G_k \left(\sqrt{G_k}\xi\right)}{\left(\sqrt{G_k}\xi\right)^T \left(\sqrt{G_k}\xi\right)} = \frac{1}{\|z\|_2^2} \int_0^1 z^T \nabla^2 f\left(x_t\right)\, z\, dt \leq M.$$

Schreiben wir (6.19) in

$$B_{k+1} = B_k - \frac{(B_k\xi_k)\ (B_k\xi_k)^T}{\xi_k^T B_k \xi_k} + \frac{\eta_k \eta_k^T}{\xi_k^T \eta_k} = B_k \left(I - \frac{\xi_k\ (B_k\xi_k)^T}{\xi_k^T B_k \xi_k^T} + \frac{B_k^{-1}\eta_k \eta_k^T}{\xi_k^T \eta_k} \right) \tag{6.22}$$

um und berücksichtigen wir, daß trace $\left(xx^T\right) = \|x\|_2^2$ ist, dann erhalten wir, daß

$$\text{trace } B_{k+1} = \text{trace } B_k - \frac{\|B_k\xi_k\|_2^2}{\xi_k^T B_k \xi_k} + \frac{\|\eta_k\|_2^2}{\xi_k^T \eta_k} = \text{trace } B_k - \frac{\|B_k\xi_k\|_2^2}{\xi_k^T B_k \xi_k^T} + M_k. \tag{6.23}$$

Die zweite Identität in (6.22) und Lemma 6.8 liefern außerdem, daß

$$\begin{aligned} \det B_{k+1} &= \det B \left(\underbrace{\left(1 - \frac{\xi_k^T B_k \xi_k}{\xi_k^T B_k \xi_k} \right)}_{=0} \left(1 + \frac{\eta_k^T B_k^{-1} \eta_k}{\xi_k^T \eta_k} \right) - \frac{-\xi_k^T \eta_k}{\xi_k^T B_k \xi_k} \frac{\xi_k^T B_k B_k^{-1} \eta_k}{\xi_k^T \eta_k} \right) \\ &= \frac{\xi_k^T \eta_k}{\xi_k^T B_k \xi_k} \det B_k \end{aligned} \tag{6.24}$$

Schreiben wir θ_k für den Winkel zwischen ξ_k und $B_k\xi_k$, also

$$\cos\theta_k := \frac{\xi_k^T\ B_k\ \xi_k}{\|\xi_k\|_2 \|B_k\xi_k\|_2},$$

so erhalten wir für den zweiten Term auf der rechten Seite von (6.23), daß

$$\frac{\|B_k\xi_k\|_2^2}{\xi_k^T B_k \xi_k} = \frac{\|B_k\xi_k\|_2^2\ \|\xi_k\|_2^2}{\left(\xi_k^T\ B_k\ \xi_k\right)^2} \frac{\xi_k^T B_k \xi_k}{\|\xi_k\|_2^2} = \frac{\xi_k^T B_k \xi_k}{\|\xi_k\|_2^2 \cos^2\theta_k} =: \frac{\beta_k}{\cos^2\theta_k}, \tag{6.25}$$

wobei $\beta_k = \|\xi_k\|_2^{-2} \left(\xi_k^T B_k \xi_k\right)$ eine Zahl ist, die nach unten durch den kleinsten[163] Eigenwert von B_k und nach oben durch den größten Eigenwert von B_k beschränkt ist. Damit liefert auch (6.24), daß

$$\det B_{k+1} = \frac{1}{\beta_k} \underbrace{\frac{\xi_k^T\ \eta_k}{\xi_k^T\ \xi_k}}_{=m_k} \det B_k = \frac{m_k}{\beta_k} \det B_k. \tag{6.26}$$

Zu einer symmetrischen, positiv (semi-)definiten Matrix $B \in \mathbb{R}^{n\times n}$ mit Eigenwerten $0 \le \lambda_1 \le \cdots \le \lambda_n$ betrachten wir nun die Funktion

$$\psi(B) := \text{trace } B - \log\det B = \sum_{j=1}^n \lambda_j - \log\left(\prod_{j=1}^n \lambda_j \right) = \sum_{j=1}^n (\lambda_j - \log\lambda_j) > 0,$$

[163] Aber immer noch positiven!

da $\log t < t$ ist für $t > 0$. Mit (6.23), (6.24), (6.25) und (6.26) sowie Übung 6.4 erhalten wir somit, daß

$$\begin{aligned}
0 \;<\; & \psi(B_{k+1}) = \operatorname{trace} B_k - \frac{\beta_k}{\cos^2\theta_k} + M_k - \log\det B_k - \log\frac{m_k}{\beta_k} \\
= \; & \psi(B_k) - \frac{\beta_k}{\cos^2\theta_k} + M_k - \log m_k + \log\beta_k \\
= \; & \psi(B_k) + (M_k - \log m_k - 1) + \left(1 - \frac{\beta_k}{\cos^2\theta_k} + \log\frac{\beta_k}{\cos^2\theta_k}\right) + \log\cos^2\theta_k \\
\leq \; & \psi(B_k) + (M - \log m - 1) + \underbrace{\left(1 - \frac{\beta_k}{\cos^2\theta_k} + \log\frac{\beta_k}{\cos^2\theta_k}\right)}_{\leq 0} + \log\cos^2\theta_k \\
\leq \; & \psi(B_k) + (M - \log m - 1) + \log\cos^2\theta_k \\
\leq \; & \psi(B_{k-1}) + 2\,(M - \log m - 1) + \log\cos^2\theta_{k-1} + \log\cos^2\theta_k \\
\vdots \; & \\
\leq \; & \psi(B_1) + k\,(M - \log m - 1) + \sum_{j=1}^{k} \log\cos^2\theta_j,
\end{aligned}$$

und indem wir die Schranken m und M hinreichend klein bzw. groß wählen, können wir ohne Einschränkung annehmen, daß $M - \log m - 1 > 0$ ist.

Und damit bekommen wir schließlich unsere Winkel θ_k in den Griff: Wäre nämlich

$$\lim_{k\to\infty} \cos\theta_k = 0 \qquad \Longrightarrow \qquad \lim_{k\to\infty} \cos^2\theta_k = 0 \qquad \Longrightarrow \qquad \lim_{k\to\infty} \log\cos^2\theta_k = -\infty,$$

also auch

$$\lim_{k\to\infty} \sum_{j=1}^{k} \big(\log\cos^2\theta_j + (M - \log m - 1)\big) = -\infty,$$

dann erhielten wir den Widerspruch

$$0 \leq \lim_{k\to\infty} \psi(B_{k+1}) = \psi(B_1) + \underbrace{\lim_{k\to\infty} \sum_{j=1}^{k} \big(\log\cos^2\theta_j + (M - \log m - 1)\big)}_{=-\infty} = -\infty.$$

Also gibt es zumindest eine Teilfolge $x^{(k_j)}$, $j \in \mathbb{N}_0$, die gegen ein Minimum konvergiert, aber wegen der *starken* Konvexität der Funktion f bleibt dann auch der gesamten Folge nichts anderes übrig, als zu konvergieren. □

Übung 6.4 Zeigen Sie: Für jedes $t > 0$ gilt $\log t \leq t - 1$ mit Gleichheit genau dann, wenn $t = 1$ ist. ♢

Bemerkung 6.9 *1. Nach [36] läßt sich dieser Beweis für BFGS mit Powell–Schrittweiten*[164] *auf eine ganze Klasse von Verfahren, die sogenannte* Broyden–Klasse *ausdehnen, funktioniert aber* nicht *für das DFP–Verfahren.*

2. Man kann auch zeigen[165]*, daß das BFGS–Verfahren* superlinear *konvergiert, wenn die zweite Ableitung Lipschitz–stetig ist. Genauer: es ist*

$$\lim_{k\to\infty} \frac{\left\|x^{(k+1)} - x^*\right\|}{\left\|x^{(k)} - x^*\right\|} = 0.$$

[164] Oder "Wolfe–Schrittweiten".

[165] Mit noch etwas sorgfältigerer Rechnerei, siehe [36, S. 214–218].

Strafen heißt, absichtlich ein Übel zuzufügen. Wer in diesem Sinne strafen will, muß sich eines höheren Auftrags zuversichtlich bewußt sein.

Gustav Radbruch

7 Strafterme und Barrieren

Wir kehren jetzt nochmal zu der in Kapitel 5 bereits erwähnten Idee zurück, *restringierte* Optimierungsprobleme dadurch zu behandeln, daß man sie in ein oder mehrere *unrestringierte* Approximationsprobleme umwandelt, bei denen die "Verletzung" der Nebenbedingungen als "Bestandteil" der Zielfunktion aufgefasst wird. Dabei betrachten wir das restringierte Optimierungsproblem

$$\min f(x), \qquad g(x) = 0, \qquad g \ : \ \mathbb{R}^n \to \mathbb{R}^m \tag{7.1}$$

bzw., wenn wir auch Ungleichungsbedingungen zulassen wollen,

$$\min f(x), \qquad g(x) = 0, \quad h(x) \geq 0, \qquad g, h \ : \ \mathbb{R}^n \to \mathbb{R}^m, \tag{7.2}$$

mit den *stetigen* Nebenbedingungsfunktionen[166] g und h für die Gleichheits- und Ungleichungsbedingungen. Um nicht in Existenznöte bezüglich des Minimums zu kommen nehmen wir außerdem an, daß der zulässige Bereich

$$\{x \in \mathbb{R}^n \ : \ g(x) = 0\} \qquad \text{bzw.} \qquad \{x \in \mathbb{R}^n \ : \ g(x) = 0,\, h(x) \geq 0\}$$

kompakt sein soll.
Die Idee hinter den Straftermen und Barrieren besteht nun darin, anstelle von f eine Funktion $x \mapsto f_\Phi(x) := f(x) + \Phi\left(g(x), h(x)\right)$ zu minimieren, wobei man natürlich $\Phi \ : \ \mathbb{R}^{2m} \to \mathbb{R}$ so wählen sollte, daß f_Φ einfach zu berechnen und einfach zu minimieren ist.

7.1 Quadratische Strafterme

Für $f \ : \ \mathbb{R}^n \to \mathbb{R}$ und Nebenbedingungen g und h definiert man die *quadratische Straffunktion*[167] $Q \ : \ \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}$ als

$$Q\left(x, \mu\right) = f(x) + \frac{1}{2\mu} \left\| g(x) \right\|_2^2 = f(x) + \frac{1}{2\mu} \sum_{j=1}^{m} g_j^2(x) \tag{7.3}$$

[166] Daß beide als Wert m–Vektoren haben, ist keine Einschränkung! Wir brauchen den Restriktionstyp, der aus weniger Nebenbedingungen besteht, durch Einführung von Bedingungen des Typs "$0 = 0$" oder "$0 \geq 0$", also durch $g_j \equiv 0$ oder $h_j \equiv 0$ aufzufüllen.

[167] Englisch: *"penalty function"*.

beziehungsweise

$$Q(x,\mu) = f(x) + \frac{1}{2\mu}\left(\|g(x)\|_2^2 + \|h_-(x)\|_2^2\right) = f(x) + \frac{1}{2\mu}\sum_{j=1}^{m} g_j^2(x) + (h_j)_-^2(x), \qquad (7.4)$$

wobei

$$y_- = \frac{1}{2}(y - |y|) = \begin{cases} 0, & y \geq 0, \\ y, & y < 0. \end{cases}$$

Allerdings hat (7.4) im Vergleich zu (7.3) einen wesentlichen Nachteil: Die Funktion $x \mapsto x_-$ ist nicht mehr differenzierbar und so kann die Differenzierbarkeitsordnung von Q *geringer* sein als die von f, g und h.

Zur Bestimmung des restringierten Minimums wählt man nun eine positive Nullfolge $\mu_k \in \mathbb{R}_0, k \in \mathbb{N}, \mu_k \to 0, k \to \infty$ und minimiert[168] $Q(x,\mu_k)$ bezüglich x, bis man eine Näherungslösung $x^{(k)}$ gefunden hat, die

$$\left\|\nabla_x Q\left(\mu_k, x^{(k)}\right)\right\|_2 \leq \tau_k \qquad (7.5)$$

für vorgegebene Toleranzen $\tau_k > 0$, $k \in \mathbb{N}_0$, erfüllt. Je kleiner nun μ_k wird, desto weniger darf die Näherungslösung die Nebenbedingungen verletzen und so besteht die Hoffnung, daß $x^{(k)} \to x^*$ für $k \to \infty$, wobei

1. x^* eine *Minimalstelle* von f ist.
2. x^* ein (näherungsweise) zulässiger Punkt ist:

$$g(x^*) \sim 0 \qquad \text{und} \qquad h(x^*) \geq -\varepsilon, \quad \varepsilon \sim 0.$$

Und diese Hoffnung besteht zu Recht.

Proposition 7.1 *Sei μ_k, $k \in \mathbb{N}$, eine positive Nullfolge und seien $x^{(k)}$, $k \in \mathbb{N}$, die Minimallösungen von $Q\left(x^{(k)}, \mu_k\right)$ aus (7.4). Dann ist jeder Häufungspunkt x^* der Folge $x^{(k)}$ eine Lösung von (7.1).*

Beweis: Sei $\widehat{x}$ eine globale Lösung von (7.2), das heißt,

$$f(\widehat{x}) \leq f(x), \qquad x \in Z_g \cap Z_{h_-}, \qquad Z_\varphi := \{x' \in \mathbb{R}^n \ : \ \varphi(x') = 0\}.$$

Insbesondere ist also $\widehat{x}$ ein *zulässiger* Punkt, der $g(\widehat{x}) = h_-(\widehat{x}) = 0$ erfüllt. Nach der Definition der $x^{(k)}$ als Minimallösungen des modifizierten Problems ist außerdem für $k \in \mathbb{N}$

$$\begin{aligned} f\left(x^{(k)}\right) \ &+ \ \frac{1}{2\mu_k}\left(\left\|g\left(x^{(k)}\right)\right\|_2^2 + \left\|h_-\left(x^{(k)}\right)\right\|_2^2\right) = Q\left(x^{(k)}, \mu\right) \leq Q(\widehat{x}, \mu) \\ &= \ f(\widehat{x}) + \frac{1}{2\mu_k}\Big(\underbrace{\|g(\widehat{x})\|_2^2}_{=0} + \underbrace{\|h_-(\widehat{x})\|_2^2}_{=0}\Big) = f(\widehat{x}), \end{aligned}$$

[168] Mit einem der Verfahren zur unrestringierten Optimierung aus den vorherigen Kapiteln. Oder natürlich mit etwas besserem ...

also

$$\left\|g\left(x^{(k)}\right)\right\|_2^2 + \left\|h_-\left(x^{(k)}\right)\right\|_2^2 \leq 2\mu_k \left(f\left(\widehat{x}\right) - f\left(x^{(k)}\right)\right). \tag{7.6}$$

Sei nun x^* ein Häufungspunkt, d.h., es gibt eine Folge k_j, so daß

$$x^* = \lim_{j\to\infty} x^{(k_j)}.$$

Wegen der Stetigkeit von g und h, der Stetigkeit der Norm und (7.6) ergibt siche

$$\begin{aligned}\left\|g\left(x^*\right)\right\|_2^2 + \left\|h_-\left(x^*\right)\right\|_2^2 &= \lim_{j\to\infty}\left(\left\|g\left(x^{(k_j)}\right)\right\|_2^2 + \left\|h_-\left(x^{(k_j)}\right)\right\|_2^2\right)\\ &\leq \lim_{j\to\infty} 2\underbrace{\mu_{k_j}}_{\to 0}\underbrace{\left(f\left(\widehat{x}\right) - f\left(x^{(k_j)}\right)\right)}_{\to f(\widehat{x})-f(x^*)} = 0,\end{aligned}$$

weswegen $x^* \in Z_g$ ein *zulässiger* Punkt ist. Und da

$$\begin{aligned}f\left(x^*\right) &= \lim_{j\to\infty} f\left(x^{(k_j)}\right) \leq \lim_{j\to\infty} f\left(x^{(k_j)}\right) + \frac{1}{2\mu_{k_j}}\left(\left\|g\left(x^{(k_j)}\right)\right\|_2^2 + \left\|h_-\left(x^{(k_j)}\right)\right\|_2^2\right)\\ &\leq f\left(\widehat{x}\right)\end{aligned}$$

ist, bleibt x^* auch gar nichts anderes übrig, als Minimallösung zu sein. □

Auch für die "Penalty–Methode" kann man wieder die Konvergenz gegen einen Punkt mit "Gradient Null" beweisen, allerdings müssen wir jetzt wieder die Nebenbedingungen berücksichtigen. Und das erinnert uns deutlich an die verallgemeinerten *Lagrange–Multiplikatoren* aus Satz 4.13, deren Existenz uns eine *notwendige*[169] Bedingung für die Existenz eines lokalen Minimums geliefert hat – ganz genau wie die Forderung $\nabla f = 0$. Trotzdem kann man für die "Penalty–Methode" einen Konvergenzbeweis führen, was wir allerdings nur für Probleme der Form (7.1), also ohne Verwendung von Ungleichungsnebenbedingungen tun werden.

Satz 7.2 *Sei $g \in C^1\left(\mathbb{R}^n\right)$ und seien μ_k und τ_k, $k \in \mathbb{N}$, positive Nullfolgen und sei x^* ein Häufungspunkt der Folge $x^{(k)}$, die (7.5) erfüllt. Sind die Gradienten $\nabla g_j\left(x^*\right)$, $j = 1, \ldots, m$, linear unabhängig, dann gibt es einen Vektor $\lambda \in \mathbb{R}^n$, so daß*

$$\nabla f\left(x^*\right) - \nabla g\left(x^*\right)\lambda = 0 \tag{7.7}$$

ist, und es ist

$$\lambda = \lim_{j\to\infty} -\frac{g\left(x^{(k_j)}\right)}{\mu_{k_j}}, \qquad x^* = \lim_{j\to\infty} x^{(k_j)}. \tag{7.8}$$

Bemerkung 7.3 *1. Die Forderung, daß die Gradienten $\nabla g_j : \mathbb{R}^n \to \mathbb{R}^n$ linear unabhängig sind, setzt natürlich voraus, daß $m \leq n$ ist; zu viele Nebenbedingungen dürfen wir also nicht haben.*

[169] Und nicht unbedingt hinreichende!

2. *Daß auch die τ_k eine Nullfolge bilden müssen, ist ziemlich naheliegend, denn ansonsten sind die Lösungen $x^{(k)}$ auch keine hinreichend guten Minimallösungen des modifizierten Problems im k–ten Schritt.*

3. *Die Gleichung (7.7), also der Lagrange–Multiplikator, ist nichts anderes als (4.9) aus Satz 4.13. Da (4.10) trivialerweise erfüllt ist – schließlich gibt es ja $h = 0$ – ist also die einzige Bedingung aus Satz 4.13, um die wir uns herumgemogelt haben, die Bedingung (4.8) an die Kegel.*

Beweis: Bildet man von (7.3) den Gradient bezüglich x, dann ergibt sich

$$\nabla_x Q\,(x,\mu) = \nabla f(x) + \frac{1}{\mu}\sum_{j=1}^{m} g_j(x)\,\nabla g_j(x), \tag{7.9}$$

was zusammen mit (7.5) die Bedingung

$$\begin{aligned}\tau_k &\geq \left\|\nabla_x Q\left(x^{(k)},\mu_k\right)\right\| = \left\|\nabla f\left(x^{(k)}\right) + \frac{1}{\mu_k}\sum_{j=1}^{m} g_j\left(x^{(k)}\right)\,\nabla g_j\left(x^{(k)}\right)\right\| \\ &\geq \frac{1}{\mu_k}\left\|\sum_{j=1}^{m} g_j\left(x^{(k)}\right)\,\nabla g_j\left(x^{(k)}\right)\right\| - \left\|\nabla f\left(x^{(k)}\right)\right\|,\end{aligned}$$

also

$$\left\|\sum_{j=1}^{m} g_j\left(x^{(k)}\right)\,\nabla g_j\left(x^{(k)}\right)\right\| \leq \mu_k\left(\tau_k + \left\|\nabla f\left(x^{(k)}\right)\right\|\right) \tag{7.10}$$

liefert. Wegen der Stetigkeit aller beteiligten Größen ist somit

$$\begin{aligned}\left\|\sum_{j=1}^{m} g_j\left(x^*\right)\,\nabla g_j\left(x^*\right)\right\| &= \lim_{j\to\infty}\left\|\sum_{j=1}^{m} g_j\left(x^{(k_j)}\right)\,\nabla g_j\left(x^{(k_j)}\right)\right\| \\ &\leq \lim_{j\to\infty}\underbrace{\mu_{k_j}}_{\to 0}\Big(\underbrace{\tau_k}_{\to 0} + \underbrace{\left\|\nabla f\left(x^{(k_j)}\right)\right\|}_{\to\nabla f(x^*)}\Big) = 0\end{aligned}$$

und die lineare Unabhängigkeit der $\nabla g_j\left(x^*\right)$ liefert, daß

$$g_j\left(x^*\right) = 0, \qquad j = 1,\ldots,m, \tag{7.11}$$

und somit ist x^* zulässig. Mit $\lambda^{(k)} = -\frac{1}{\mu_k} g\left(x^{(k)}\right) \in \mathbb{R}^m$ ergibt sich aus (7.9), daß

$$\nabla_x Q\left(x^{(k)},\mu_k\right) = \nabla f_k - \nabla g_k\,\lambda^{(k)}, \qquad \nabla g_k := \left[\nabla g_j\left(x^{(k)}\right)\;:\; j = 1,\ldots,m\right] \in \mathbb{R}^{n\times m},$$

und da

$$\left\|\nabla f_k - \nabla g_k\,\lambda^{(k)}\right\| = \left\|\nabla_x Q\left(x^{(k)},\mu_k\right)\right\| \leq \tau_k \to 0$$

ist, dann müssen wir aus Stetigkeitsgründen zeigen, daß $\lambda^* = \lim_{k\to\infty} \lambda^{(k)}$ existiert. Da $\nabla g\,(x^*)$ vollen Rang m hat, gilt dies für hinreichend großes k auch für ∇g_k und deswegen ist $\nabla^T g_k \nabla g_k \in \mathbb{R}^{m\times m}$ invertierbar, wenn nur k hinreichend groß gewählt ist. Nun ist dann

$$\nabla g_k\,\lambda^{(k)} = \nabla f_k - \nabla_x Q\left(x^{(k)},\mu_k\right) \quad \Longrightarrow \quad \nabla^T g_k \nabla g_k\,\lambda^{(k)} = \nabla^T g_k\left(\nabla f_k - \nabla_x Q\left(x^{(k)},\mu_k\right)\right)$$

und somit, für hinreichend großes k,

$$\lambda^{(k)} = \underbrace{\left(\nabla^T g_k \nabla g_k\right)^{-1}}_{\to(\nabla^T g_*\nabla g_*)^{-1}}\ \underbrace{\nabla^T g_k}_{\to\nabla^T g_*}\Big(\underbrace{\nabla f_k}_{\to\nabla f_*} - \underbrace{\nabla_x Q\left(x^{(k)},\mu_k\right)}_{\to 0}\Big), \tag{7.12}$$

also

$$\lambda^* = \lim_{j\to\infty} \lambda^{(k_j)} = \left(\nabla^T g_* \nabla g_*\right)^{-1}\,\nabla^T g_*\,\nabla f_*,$$

was einen wohldefinierten Multiplikator ergibt. □

Allerdings gibt es ein kleines Problem, und zwar ein numerisches Problem bei der Bestimmung der näherungsweisen Minima $x^{(k)}$. Dazu nehmen wir der Einfachheit an, daß $h \equiv 0$ ist, daß also die Nebenbedingungen ausschließlich in Gleichungsform vorliegen, und bilden einmal die Hessematrix

$$\begin{aligned}
\nabla_x^2 Q\,(x,\mu) &= \nabla_x\left(\nabla f(x) + \frac{1}{\mu}\sum_{j=1}^m g_j(x)\,\nabla g_j(x)\right) = \nabla^2 f(x) + \frac{1}{\mu}\sum_{j=1}^m \nabla\left(g_j(x)\,\nabla g_j(x)\right)\\
&= \nabla^2 f(x) + \frac{1}{\mu}\sum_{j=1}^m \left(\nabla g_j(x)\,\nabla^T g_j(x) + g_j(x)\,\nabla^2 g_j(x)\right)\\
&= \nabla^2 f(x) + \frac{1}{\mu}\nabla g(x)\nabla^T g(x) + \sum_{j=1}^m \frac{g_j(x)}{\mu}\nabla^2 g_j(x),
\end{aligned} \tag{7.13}$$

die ja bei den meisten Verfahren eine ziemlich entscheidende Rolle gespielt hat. Sei nun x^μ die Optimallösung für ein vorgegebenes $\mu > 0$ und $\lambda = -g\,(x^\mu)\,/\mu$, dann ist für $x \sim x^\mu$

$$\begin{aligned}
A_\mu &:= \nabla_x^2 Q\,(x,\mu) = \nabla^2 f(x) + \frac{1}{\mu}\nabla g(x)\nabla^T g(x) - \sum_{j=1}^m \lambda_j \nabla^2 g_j(x)\\
&= \nabla^2\left(f - \lambda^T g\right)(x) + \frac{1}{\mu}\nabla g(x)\nabla^T g(x).
\end{aligned}$$

Da für eine "vernünftige" Konvergenz $m \le n$ sein muß, siehe Satz 7.2, und im Normalfall sogar $m < n$ sein wird, hat die symmetrische Matrix A_μ gerade $n-m$ Eigenvektoren und Eigenwerte η_j, $j = m+1,\ldots,n$, die nicht von μ abhängen, nämlich diejenigen Vektoren, die zu $\left(\nabla^T g(x)\right)^\perp \subset \mathbb{R}^n$ gehören, und m Eigenvektoren zu Eigenwerten der Form $\eta_j = \eta_j'/\mu$, $j = 1,\ldots,m$, die für $\mu \to 0$ beliebig groß werden können, für $\mu \to 0$ sind die Hessematrizen also beliebig schlecht konditioniert! Und das hat natürlich Auswirkungen, wenn man Gleichungssysteme der Form

$$\nabla_x^2 Q\left(x^{(k)},\mu\right) y^{(k)} = -\nabla_x Q\left(x^{(k)},\mu\right),$$

beispielsweise beim Newton–Verfahren, lösen will. Glücklicherweise ist das aber beim Newton–Verfahren nun gerade wieder nicht so schlimm: setzen wir nämlich

$$z := \mu^{-1} \nabla^T g(x)\, y$$

und verwenden wir (7.13), dann erhalten wir das äquivalente Gleichungssystem

$$\begin{aligned} \left(\nabla^2 f(x) + \sum_{j=1}^{m} \frac{g_j(x)}{\mu} \nabla^2 g_j(x) \right) y + \nabla g(x)\, z &= -\nabla_x Q\,(x, \mu) \\ \nabla^T g(x)\, y - \mu\, z &= 0, \end{aligned}$$

also

$$\begin{bmatrix} \nabla^2 f(x) + \sum_{j=1}^{m} \frac{g_j(x)}{\mu} \nabla^2 g_j(x) & \nabla g(x) \\ \nabla^T g(x) & -\mu I \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} = \begin{bmatrix} -\nabla_x Q\,(x, \mu) \\ 0 \end{bmatrix}$$

und diese Matrix ist nun wieder gut konditioniert und somit das Gleichungssystem numerisch stabil lösbar.

7.2 Logarithmische Barrieren

Barrieren sind eine gute Methode für restringierte Optimierungsprobleme, die nur durch *Ungleichungen* beschränkt sind, also Probleme der Form

$$\min f(x), \qquad h(x) \geq 0, \qquad h : \mathbb{R}^n \to \mathbb{R}^m. \tag{7.14}$$

mit mindestens stetigem h. Dabei setzen wir

$$\Omega := \{x \in \mathbb{R}^n \ : \ h(x) \geq 0\} \qquad \text{sowie} \qquad \Omega^* := \{x \in \mathbb{R}^n \ : \ h(x) > 0\}$$

und nehmen an, daß $\Omega^* \neq \emptyset$, daß es also "richtige" *innere Punkte* von Ω gibt.

Übung 7.1 Zeigen Sie: $\Omega^* \subseteq \Omega^\circ$, aber Gleichheit gilt im allgemeinen nicht. ♢

Definition 7.4 *Sei $\Omega \subseteq \mathbb{R}^n$ wie oben. Eine Funktion $\phi \ : \ \Omega \to \mathbb{R}$ heißt* Distanzfunktion *für Ω, wenn*

1. *$\phi(x) > 0$, $x \in \Omega^*$.*
2. *$\phi(x) = 0$, $x \in \partial^*\Omega := \Omega \setminus \Omega^*$.*
3. *für $x, x' \in \Omega$*

$$h(x) > h\,(x') \qquad \Longrightarrow \qquad \phi(x) > \phi\,(x')$$

gilt.

Die letzte Bedingung in der obigen Definition bedeutet, daß ϕ den Abstand vom Rand im Sinne der Nebenbedingungen misst und daß ϕ umso größer ist, je "besser" die Nebenbedingungen erfüllt sind.

Beispiel 7.5 *Die "natürliche" Distanzfunktion zu der Nebenbedingungsfunktion h ist die Funktion*

$$\phi = \phi_h := \prod_{j=1}^{m} h_j.$$

Eine *Barrierefunktion* für Ω ist nun eine möglichst glatte Funktion $\psi : \Omega^* \to \mathbb{R}$ mit der Eigenschaft, daß

$$\psi(x) < \infty, \quad x \in \Omega^* \qquad \text{und} \qquad \lim_{x \to \delta^* \Omega} \psi(x) = \infty.$$

Verwendet man nämlich so eine Barrierefunktion als Strafterm, dann wird bei der Minimumssuche für $f+\psi$, ausgehend von einem Startwert $x^{(0)} \in \Omega^*$ der *strikt zulässige* Bereich Ω^* nie verlassen werden, insbesondere, wenn man ψ außerhalb von Ω "glatt" mit $\psi = \infty$ fortsetzt.

Beispiel 7.6 *Die natürliche Barrierefunktion zu einer Distanzfunktion ϕ ist $\psi = -\log \phi$, also insbesondere*

$$\psi_h = -\log \phi_h = -\log \prod_{j=1}^{m} h_j = -\sum_{j=1}^{m} \log h_j$$

Ist außerdem

$$\alpha := \sup_{x \in \Omega} \phi(x) < \infty,$$

dann kann man ϕ durch $\alpha^{-1}\phi$ ersetzen und die zugehörige Barrierefunktion $\psi = \log \alpha - \log \phi$ wäre sogar nichtnegativ.

Wie vorher mit den quadratischen Straftermen betrachtet man auch jetzt wieder ein modifiziertes Optimierungsproblem, nämlich

$$\min_x P(x, \gamma) = f(x) - \gamma \log \phi_h(x) = f(x) - \gamma \sum_{j=1}^{m} \log h_j, \qquad \gamma > 0, \tag{7.15}$$

und läßt dann γ schön langsam gegen Null gehen. Dabei erzeugt man *immer* eine Folge von *strikten inneren* Punkten, denn die Funktion ϕ_h nimmt ja nur auf Ω^* endliche Werte an[170]. Die Vorgehensweise ist nun wieder wie vorher bei den Straftermen.

Algorithmus 7.7 Gegeben: *Funktion $f \in C^1(\mathbb{R}^n)$, Nebenbedingungen $h \in C^1(\mathbb{R}^n)^m$.*

1. *Wähle $\gamma_1, \tau_1 \in \mathbb{R}_+$.*
2. *Für $k = 1, 2 \ldots$*

[170] Unter Verwendung der Konvention $\log t = -\infty$ für $t < 0$.

(a) Bestimme $x^{(k)} \in \mathbb{R}^n$, so daß

$$\left\|\nabla_x P\left(x^{(k)}, \gamma_k\right)\right\| \leq \tau_k. \tag{7.16}$$

(b) Wähle

$$\gamma_{k+1} \in (0, \gamma_k), \qquad \tau_{k+1} \in (0, \tau_k).$$

Ergebnis: *Folge $x^{(k)}$, die (hoffentlich) gegen ein Minimum konvergiert.*

Eine Konvergenzanalyse solcher Barrierefunktionen ist haarig und aufwendig, so daß wir sie uns schenken. Allerdings sieht man ganz gut, *warum* die Sache so problematisch ist. Sehen wir uns nämlich die notwendige Voraussetzung für ein Minimum von $P(x, \gamma)$ an, also

$$0 = \nabla_x P(x, \gamma) = \nabla f(x) - \gamma \sum_{j=1}^{n} \nabla \log h_j(x) = \nabla f(x) - \sum_{j=1}^{n} \frac{\gamma}{h_j(x)} \nabla h_j(x),$$

dann ist, weil $x \in \Omega^*$ ist,

$$\mu := \mu^\gamma = \left[\frac{\gamma}{h_j(x)} \ : \ j = 1, \ldots, m\right] \in \mathbb{R}^m_+$$

ein guter Kandidat für den "Ungleichungsmultiplikator" aus Satz 4.13, denn schließlich ist ja

$$\nabla f(x) - \underbrace{[\nabla h_j \ : \ j = 1, \ldots, m]}_{=\nabla h} \mu = 0;$$

Allerdings folgt aus der Definition von μ^γ, daß

$$\mu^T h(x) = \sum_{j=1}^{m} h_j(x) \frac{\gamma}{h_j(x)} = m\,\gamma$$

und damit ist die Bedingung (4.10) aus Satz 4.13 leider nicht erfüllt. Na gut, wenn $\gamma \to 0$ geht, dann wird das besser und besser, aber dann muß halt auch

$$\mu_j^* = \lim_{\gamma \to 0} \mu_j^\gamma = \frac{\gamma}{h_j\left(x^\gamma\right)}, \qquad j = 1, \ldots, m.$$

existieren. Das ist kein wirkliches Problem, wenn $x^* = \lim x^\gamma$ in Ω^* liegt, aber wenn das Minimum an einem Randpunkt angenommen wird, dann braucht man weitere Bedingungen an die Nebenbedingungen und die Funktion f, um Konvergenz beweisen zu können.

7.3 Erweiterte Lagrange–Multiplikatoren

Als letztes Beispiel betrachten wir eine Methode, die sich in praktischen Anwendungen besonders gut bewährt hat, nämlich die *augmented Lagrangian*, was man als "ergänzte Lagrange–Multiplikatoren" oder "erweiterte Lagrange–Multiplikatoren" übersetzen könnte. Auch wenn

man Ungleichungsnebenbedingungen in diesen Rahmen integrieren könnte, wollen wir[171] uns nur auf Gleichungen beschränken, also ein Optimierungsproblem der Form

$$\min f(x), \qquad g(x) = 0, \tag{7.17}$$

zu lösen versuchen. Die "Hilfsfunktion", die wir jetzt betrachten wollen, hat die Form

$$L(x, \lambda, \mu) = f(x) - \lambda^T g(x) + \frac{1}{2\mu} \|g(x)\|_2^2, \tag{7.18}$$

wobei man sich unter λ eine Näherung für den Lagrange–Multiplikator vorzustellen hat – man "mischt" also sozusagen Lagrange–Multiplikatoren mit quadratischen Straftermen. Der Name "erweiterte Lagrange–Funktion" stammt übrigens daher, daß man die Funktion $L(x, \lambda) = f(x) - \lambda^T g(x)$ auch manchmal als *Lagrange–Funktion* bezeichnet[172]. Mit den schon wohlbekannten Rechnungen ergibt sich dann sofort, daß

$$\nabla_x L(x, \lambda, \mu) = \nabla f(x) - \nabla g(x)\, \lambda + \frac{1}{\mu} \nabla g(x)\, g(x), \tag{7.19}$$

und ist x^* nun eine Minimallösung von (7.18), insbesondere also $x^* \in Z_g$, dann ist

$$0 = \nabla_x L(x^*, \lambda, \mu) = \nabla f(x^*) - \nabla g(x^*)\ \lambda + \frac{1}{\mu} \nabla g(x^*)\ \underbrace{g(x^*)}_{=0} = \nabla f(x^*) - \nabla g(x^*)\ \lambda,$$

λ wäre also ein Lagrange–Multiplikator für f. Haben wir hingegen einen unzulässigen, näherungsweisen Minimalwert $\widehat{x}$ des unrestringierten Hilfsproblems (7.18) gefunden, dann ist also, unter Verwendung von (7.19),

$$0 \sim \nabla_x L(\widehat{x}, \lambda, \mu) = \nabla f(\widehat{x}) - \nabla g(\widehat{x}) \left(\lambda - \frac{1}{\mu} g(\widehat{x}) \right),$$

also ist

$$\widehat{\lambda} := \lambda - \frac{g(\widehat{x})}{\mu}$$

eine gute Schätzung für den Lagrange–Multiplikator. Und das können wir auch schon wieder als Basis für ein iteratives Verfahren nehmen.

Algorithmus 7.8 Gegeben: *Zielfunktion* $f \in C^1(\mathbb{R}^n)$ *und Gleichungsnebenbedingungen* $g \in C^1(\mathbb{R}^n)^m$.

1. *Wähle*

$$\lambda^1 \in \mathbb{R}^m, \qquad \tau_1, \mu_1 \in \mathbb{R}_+$$

[171] Schon der Einfachheit halber, ansonsten siehe [36, S. 516–518].

[172] Um die Verwirrung zu komplettieren: Im Zusammenhang mit der *Lagrange–Interpolation*, das ist, im Gegensatz zur *Hermite–Interpolation*, bei der auch Ableitungen interpoliert werden, die Interpolation von Funktionswerten an vorgegebenen Stellen, verwendet man den Begriff "Lagrange–Funktion" gerne für eine Funktion, die an einem der Interpolationspunkte den Wert 1, an allen anderen Interpolationspunkten aber den Wert 0 hat.

2. *Für* $k = 1, 2, \ldots$

   (a) *Bestimme* $x^{(k)} \in \mathbb{R}^n$*, so daß*

   $$\left\| \nabla_x L_x \left(x^{(k)}, \lambda^k, \mu_k \right) \right\| \leq \tau_k.$$

   (b) *Setze*

   $$\lambda^{k+1} = \lambda^k - \frac{g\left(x^{(k)}\right)}{\mu_k}$$

   (c) *Wähle*

   $$\mu_{k+1} \in (0, \mu_k), \qquad \tau_{k+1} \in (0, \tau_k).$$

Im Gegensatz zu den "einfachen" quadratischen Straftermen besteht der Reiz dieser Methode darin, daß man μ nicht beliebig verkleinern muß, sondern daß es einen Wert $\bar{\mu}$ gibt, so daß man für alle $\mu < \bar{\mu}$ bei einem lokalen Minimum landet – das läßt auf ein sinnvolles Terminieren des Verfahrens nach endlich vielen Schritten hoffen.

Satz 7.9 *Für* $f \in C^2(\mathbb{R}^n)$ *und* $g \in C^2(\mathbb{R}^n)^m$ *sei* $x^* \in Z_g$ *eine lokale Lösung von (7.17) und* λ^* *der zugehörige Multiplikator. Außerdem seien die Spalten von* $\nabla g(x^*)$ *linear unabhängig*[173] *und es sei*

$$y^T \nabla_x^2 L(x^*, \lambda^*)\, y := y^T \nabla_x^2 \left(f - g^T \lambda^* \right)(x^*)\; y > 0, \qquad \nabla^T g(x^*)\; y = 0, \quad y \neq 0. \tag{7.20}$$

Dann gibt es einen Wert $\bar{\mu} > 0$*, so daß für alle* $\mu < \bar{\mu}$ *der Punkt* x^* *ein striktes lokales Minimum von* $L(\cdot, \lambda^*, \mu)$ *ist.*

Bemerkung 7.10 *Die Bedingung (7.20) ist eine* hinreichende Bedingung *zweiter Ordnung für das Vorliegen eines Minimums unter Nebenbedingungen. Für Details siehe [36, Theorem 12.6, S. 345].*

Definition 7.11 *Sei* $A \in \mathbb{R}^{n \times n}$ *eine symmetrische Matrix. Wir schreiben* $A \geq 0$ *wenn* A *positiv semidefinit ist und* $A > 0$*, wenn* A *strikt positiv definit ist.*

Beweis: Wir werden zeigen, daß

$$\nabla_x L(x^*, \lambda^*, \mu) = 0 \qquad \text{und} \qquad \nabla_x^2 L(x^*, \lambda^*, \mu) > 0 \tag{7.21}$$

ist; nach Proposition 5.2 ist dann x^* ein striktes lokales Minimum von $L(\cdot, \lambda^*, \mu)$.

Der erste Teil von (7.21) ist einfach: Da $x^* \in Z_g$ und da λ^* als Lagrange–Multiplikator die Bedingung $\nabla f(x^*) - \nabla g(x^*)\; \lambda^* = 0$ erfüllt, ist nach (7.19)

$$\nabla_x L(x^*, \lambda^*, \mu) = \underbrace{\nabla f(x^*) - \nabla g(x^*)\; \lambda^*}_{=0} + \frac{1}{\mu} \nabla g(x^*)\; \underbrace{g(x^*)}_{=0} = 0$$

[173] Wie in Satz 7.2, das heißt also auch wieder, daß $m \leq n$ ist.

und zwar sogar *unabhängig* von μ.

Interessanter wird natürlich die Sache mit der zweiten Ableitung. Da

$$\begin{aligned}\nabla_x^2 L(x,\lambda,\mu) &= \nabla_x\left(\nabla f(x) - \nabla g(x)\left(\lambda - \frac{1}{\mu}\, g(x)\right)\right)\\ &= \nabla^2 f(x) - \sum_{j=1}^{m} \lambda_j\, \nabla^2 g_j(x) + \frac{1}{\mu}\sum_{j=1}^{m}\left(g_j(x)\,\nabla^2 g_j(x) + \nabla g_j(x)\,\nabla^T g_j(x)\right)\\ &= \nabla_x^2 L(x,\lambda) + \frac{1}{\mu}\sum_{j=1}^{m}\left(g_j(x)\,\nabla^2 g_j(x) + \nabla g_j(x)\,\nabla^T g_j(x)\right),\end{aligned}$$

ist wegen $x^* \in Z_g$ dann

$$\nabla_x^2 L(x^*,\lambda^*,\mu) = \nabla_x^2 L(x^*,\lambda^*) + \frac{1}{\mu}\sum_{j=1}^{m} \nabla g_j(x^*)\ \nabla^T g_j(x^*) = \nabla_x^2 L(x^*,\lambda^*) + \frac{\nabla g_* \nabla^T g_*}{\mu}. \tag{7.22}$$

Da[174]

$$\mathbb{R}^n = \ker \nabla^T g^* \oplus \nabla g_* \mathbb{R}^m$$

ist, können wir also nun ein beliebiges $0 \neq y \in \mathbb{R}^n$ als

$$y = v + \nabla g_*\, w = v + w^*, \qquad \nabla^T g^*\, v = 0, \qquad v \in \mathbb{R}^n,\ w \in \mathbb{R}^m,$$

schreiben und es ist, mit (7.22)

$$\begin{aligned} y^T\, \nabla_x^2 L(x^*,\lambda^*,\mu)\ y &= (v + \nabla g_*\, w)^T \left(\nabla_x^2 L(x^*,\lambda^*) + \frac{\nabla g_* \nabla^T g_*}{\mu}\right)(v + \nabla g_*\, w)\\ &= v^T\, \nabla_x^2 L(x^*,\lambda^*)\ v + 2v^T \nabla_x^2 L(x^*,\lambda^*)\ \nabla g_*\, w + \nabla^T g_*\, w\, \nabla_x^2 L(x^*,\lambda^*)\ \nabla g_*\, w\\ &\quad + \frac{1}{\mu}\Big(\underbrace{v^T \nabla g^*}_{=0}\ \underbrace{\nabla^T g^* v}_{=0} + 2\underbrace{v^T \nabla g^*}_{=0}\nabla^T g^* \nabla g^*\, w + w^T \nabla^T g^* \nabla g^* \nabla^T g^* \nabla g^*\, w\Big)\\ &= v^T\, \nabla_x^2 L(x^*,\lambda^*)\ v + 2v^T \nabla_x^2 L(x^*,\lambda^*)\ w^* + w^{*T}\, \nabla_x^2 L(x^*,\lambda^*)\ w^* + \frac{\left\|\nabla^T g_* \nabla g_*\, w\right\|_2^2}{\mu}\end{aligned}$$

Nach der Voraussetzung (7.20) ist nun

$$v^T\, \nabla_x^2 L(x^*,\lambda^*)\ v \geq A\, \|v\|_2^2, \qquad A > 0,$$

sowie

$$\begin{aligned} v^T \nabla_x^2 L(x^*,\lambda^*)\ w^* &\geq -\left|v^T \nabla_x^2 L(x^*,\lambda^*)\ w^*\right| \geq -\|v\|_2 \underbrace{\left\|\nabla_x^2 L(x^*,\lambda^*)\, \nabla g_*\right\|_2}_{=:B} \|w\|_2\\ &\geq -B\, \|v\|_2\, \|w\|_2, \qquad B > 0,\end{aligned}$$

[174] Sollte aus der linearen Algebra bekannt sein!

und

$$w^{*T}\,\nabla_x^2 L\,(x^*,\lambda^*)\;w^* \geq -\underbrace{\left\|\nabla^T g_*\,\nabla_x^2 L\,(x^*,\lambda^*)\;\nabla g_*\right\|_2}_{=:C}\|w\|_2^2 = -C\,\|w\|_2^2\,, \qquad C>0.$$

Da die Matrix ∇g_* den Maximalrang m hat ist außerdem $\nabla^T g_* \nabla g_*$ strikt positiv definit, weswegen

$$\left\|\nabla^T g_* \nabla g_*\,w\right\|_2^2 \geq D\;\|w\|_2^2\,, \qquad D>0,$$

ist. Somit ist

$$\begin{aligned} y^T\,\nabla_x^2 L\,(x^*,\lambda^*,\mu)\;y &\geq A\left(\|v\|_2^2 - 2\frac{B}{A}\,\|v\|_2\,\|w\|_2 + \frac{B^2}{A^2}\,\|w\|_2^2\right) + \|w\|_2^2\left(\frac{D}{\mu} - C - \frac{B^2}{A}\right) \\ &= \underbrace{A\left(\|v\|_2^2 - \frac{B}{A}\;\|v\|_2^2\right)^2}_{\geq 0} + \|w\|_2^2\left(\frac{D}{\mu} - C - \frac{B^2}{A}\right), \end{aligned}$$

was ≥ 0 ist, sobald

$$\mu < \bar{\mu} := \frac{D}{C + B^2/A}$$

ist. Außerdem gilt für jedes solche $\mu < \bar{\mu}$, daß

$$y^T\,\nabla_x^2 L\,(x^*,\lambda^*,\mu)\;y = 0 \qquad \Longleftrightarrow \qquad v=0, \quad w=0.$$

□

Bei der "echten" praktischen Implementierung im Optimierungspaket LANCELOT von Conn, Gould und Toint [8] betrachtet man dann "nur" lokalisierte Probleme der Form

$$\min f(x), \qquad g(x)=0, \quad a \leq x \leq b, \quad a,b \in \mathbb{R}^n,$$

die mit einem geeigneten Iterationsverfahren und Updateregeln für die Multiplikatoren, Toleranzen und Strafparameter (das μ) behandelt werden. Siehe [36, S. 522–523].

Denn viel größres Vertrauen muß immer erwecken, was selber
Unabhängig von andrem den Irrtum
schlägt mit der Wahrheit.

Lukrez, *Über die Natur der Dinge*

8 Trust–Region–Verfahren

In diesem Kapitel befassen wir uns mit einer anderen Familie von Methoden zur unrestringierten Optimierung, bei der ein *quadratisches Modell* der Zielfunktion optimiert wird, aber nur Schritweiten innerhalb eines Bereiches zugelassen werden, auf der das quadratische Modell die Zielfunktion auch "zuverlässig" annähert, der sogenannten *Trust Region*.

8.1 Quadratische Modelle und wem man wo wie vertraut

Wir nähern wieder die Zielfunktion f lokal um $x \in \mathbb{R}^n$ durch das quadratische Modell

$$f(x+y) \sim q(y) = f + g^T y + \frac{1}{2} y^T B y, \qquad f \in \mathbb{R},\, g \in \mathbb{R}^n,\, B \in \mathbb{R}^{n\times n}, \quad B^T = B, \tag{8.1}$$

bzw.

$$f\left(x^{(k)} + y\right) \sim q_k(y) = f_k + g_k^T y + \frac{1}{2} y^T B_k y$$

wenn wir uns iterative Verfahren basteln wollen. Dieses quadratische Modell kann man auf die verschiedensten Arten erhalten:

1. Durch *exakte* Kenntnis von $f \in C^2(\mathbb{R}^n)$ und die Taylorformel, das heißt, man setzt in (8.1)
$$f = f(x), \qquad g = \nabla f(x), \qquad B = \nabla^2 f(x).$$

2. Durch *polynomiale Interpolation* von f. Kennt man f an $\binom{n+2}{2} = \dim \Pi_2$ Stellen $\mathscr{X} \subset \mathbb{R}^n$, dann kann man[175] ein quadratisches Polynom bestimmen $q \in \Pi_2$, das an diesen Stellen interpoliert,
$$q(x) = f(x), \qquad x \in \mathscr{X},$$
und dieses Polynom als Modell verwenden.

[175] Hoffentlich ...

3. Durch *Least–Squares–Approximation* von f. Kennt man f an *mindestens* $\dim \Pi_2$ Stellen $\mathscr{X} \subset \mathbb{R}^n$, dann sucht man ein Polynom $q \in \Pi_2$, so daß

$$\sum_{x \in \mathscr{X}} (q(x) - f(x))^2 = \min_{q' \in \Pi_2} \sum_{x \in \mathscr{X}} (q'(x) - f(x))^2$$

Die beiden letzten Ansätze haben den Vorteil, daß sie nicht nur für differenzierbare oder zweimal differenzierbare Funktionen verwendet werden können, sondern wir nur die Möglichkeit haben müssen, die Funktion f an gewissen Punkten auszuwerten.

Bemerkung 8.1 *So einfach ist es aber leider doch wieder nicht mit der Erzeugung eines quadratischen Modells durch Interpolation. Es gibt da einiges an Problemen:*

1. *Im Gegensatz zum univariaten Fall spielt die Geometrie der Punkte in $\mathscr{X}$ eine Rolle bereits bei der Frage nach der (eindeutigen)* Lösbarkeit *des Interpolationsproblems. So ist es beispielsweise in zwei Variablen nicht möglich, einen quadratischen Interpolanten an $\dim \Pi_2 = 6$ Punkte zu finden, wenn diese alle auf dem Einheitskreis liegen, denn dann verschwindet ja das quadratische Polynom $x^2 + y^2 - 1$ an all diesen Punkten.*

2. *Auch die Frage inwieweit so ein Interpolationspolynom überhaupt eine gute Näherung an f darstellt, also Fehlerabschätzungen der Form $\|f - q\| \leq \ldots$ hängen selbst für hinreichende oft differenzierbares f von der Geometrie der Punkte ab, beispielsweise vom Quotienten aus Umkreis- und Inkreisradius, siehe [7], und das kann beliebig schlecht werden.*

3. *Auch algorithmisch ist die Polynominterpolation nicht so ganz einfach, für effiziente und stabile Implementierungen muß man sich schon ein bißchen was überlegen, siehe z.B. [40, 4].*

4. *Mehr Information über Trust–Region–Verfahren unter Verwendung polynomialer Interpolation findet sich in [9].*

Aber zurück zur Optimierung! Bei einem "Trust–Region–Verfahren" erzeugt man zusätzlich zu einer Folge $x^{(k)} \in \mathbb{R}^n$ von Punkten eine Folge $r_k > 0$ von Radien; die *Trust Region* $T_k = B\left(x^{(k)}, r_k\right)$ ist dann der Kreis vom Radius r_k um $x^{(k)}$ und dieser Radius wird die Schrittweitensteuerung beeinflussen.
Zuerst einmal bestimmt man jetzt $y^{(k)}$ als Optimalstelle des quadratischen Modells innerhalb der Trust Region, also als Lösung von

$$\min_y q_k(y), \qquad \|y\| \leq r_k.$$

Dann überprüft man, inwieweit das quadratische Modell wirklich zutreffend war, indem man den Quotienten

$$\rho_k := \frac{f\left(x^{(k)}\right) - f\left(x^{(k)} + y^{(k)}\right)}{q_k(0) - q_k\left(y^{(k)}\right)}$$

bestimmt. Der Nenner von ρ_k ist wegen der Definition von $y^{(k)}$ als Minimalstelle übrigens immer positiv; wäre also $\rho_k < 0$, dann muß $f\left(x^{(k)}\right) < f\left(x^{(k)} + y^{(k)}\right)$ und das Modell kann nur sehr schlecht sein: Die Minimalstelle von q entspricht noch nicht einmal einer Verbesserung von f. Ist hingegen das quadratische Modell exakt, also $f(x+y) = q(y)$, dann ergibt sich natürlich $\rho_k = 1$. Und nun entscheidet man anhand von ρ_k:

1. Ist ρ_k klein und insbesondere negativ, dann verkleinert man die Trust Region und versucht es nochmals[176] mit $x^{(k+1)} = x^{(k)}$.

2. Ist ρ_k groß und hat man für $y^{(k)}$ die zulässige Maximalschrittweite r_k voll ausgenutzt, dann vergrößert man den Radius der Trust Region und setzt $x^{(k+1)} = x^{(k)} + y^{(k)}$.

3. Ist ρ_k "durchwachsen", dann beläßt man den Radius wie er ist[177] und setzt wieder $x^{(k+1)} = x^{(k)} + y^{(k)}$.

Natürlich muß man die Frage was groß und was klein ist und wie man vergrößert und verkleinert spezifizieren. In [36] heißt das $\rho_k < \frac{1}{4}$ für "klein", und dann wird $r_{k+1} = \frac{1}{4} r_k$ gesetzt und $\rho > \frac{3}{4}$ für "groß", wobei der Radius allerdings "nur" verdoppelt wird. Außerdem kann man noch einen Maximalradius r^* vorgeben, der nie überschritten werden darf, was zur Regel $r_{k+1} = \min\left(2r_k, r^*\right)$ führt.

8.2 Wahl der Richtung

Als erstes wollen wir uns zwei Methoden zur schnellen näherungsweisen Bestimmung der Minimallösung des "Modellproblems" ansehen und damit der "Fortschrittsrichtung" $y^{(k)}$.

Definition 8.2 *Der* Cauchy–Punkt y_c *ist definiert als* $y_c := \tau^* y^*$, *wobei* y^*, τ^* *Lösungen der (sequentiellen) Optimierungsprobleme*

$$\min_{y \in \mathbb{R}^n} f + g^T y = q(0) + \nabla^T q(0)\, y, \quad \|y\|_2 \le r, \qquad \min_{\tau \in \mathbb{R}_+} q\left(\tau\, y^*\right), \quad \|\tau y^*\|_2 \le r.$$

Wählt man $x^{(k+1)} = x^{(k)} + y^{(k)}$, wobei $y^{(k)}$ Cauchy–Punkt bezüglich g_k ist, dann ergibt das zwar ein Trust–Region–Verfahren, bei dem ein "vernünftiger" Abstieg gewählt ist, siehe Lemma 8.7, aber da

$$y^{(k)} = r_k \frac{g}{\|g\|_2} = r_k \frac{\nabla q(0)}{\|\nabla q(0)\|_2}$$

ist, erhalten wir, bis auf die Schrittweitensteuerung, eine Variante des steilsten Abstiegs und von dem wissen wir ja, siehe Lemma 5.9 und Beispiel 5.10, daß er nicht so grandios funktioniert.

"Vernünftiger" wäre es mit Sicherheit, wie beim Newton–Verfahren die Richtung

$$y = \left(\nabla^2 q\right)^{-1} \nabla q(0) = B^{-1} g$$

[176] Das quadratische Modell bleibt dabei unverändert. Man könnte natürlich hier auch einen "Modell–Update" in Betracht ziehen, bei dem z.B. neue, nähere Interpolationspunkte gewählt werden.

[177] Es funktioniert ja so halbwegs.

zu wählen[178], oder zumindest diese Größe bei der Richtungsbestimmung in Betracht zu ziehen. Zu diesem Zweck sehen wir uns einmal an, wie die Lösung y^r des Minimierungsproblems

$$\min_y q(y) = f + g^T y + \frac{1}{2} y^T B y, \qquad \|y\|_2 \leq r, \tag{8.2}$$

eigentlich aussieht. Ist $r = \infty$, betrachten wir also das *unrestringierte* Problem, so kennen wir die Lösung: $y^\infty = -B^{-1}g$, siehe Beispiel 5.8. Das heißt aber, daß $y^r = y^\infty = -B^{-1}g$ so lange $r \geq \|y^\infty\|_2$ ist. Andererseits liefert uns aber die Taylorformel, genauer, die Tatsache, daß der quadratische Anteil nur mit der Größenordnung $\|y\|_2^2$, der lineare Anteil aber von der Größenordnung $\|y\|_2$ beiträgt, daß

$$y^0 := \lim_{r \to 0} y^r = -g = -\nabla q(0).$$

ist. Da nutzt man für die sogenannte *Dogleg*[179]*–Methode*, bei der man in der steilsten Abstiegsrichtung y^0 aus dem Punkt 0 "herausfährt", aber dafür sorgt, daß man in der unrestringierten Optimallösung y^r "ankommt". Dazu kombinieren wir die Richtungsvektoren des steilsten Abstiegs und der Newton–Richtung

$$y^0 := -\frac{g^T g}{g^T B g}\, g \qquad \text{und} \qquad y^1 := -B^{-1} g$$

in eine stückweise lineare Funktion

$$y(t) := \begin{cases} t\, y^0, & 0 \leq t \leq 1 \\ (2-t)\, y^0 + (t-1)\; y^1, & 1 \leq t \leq 2, \end{cases} \qquad t \in [0,2]$$

die die Eigenschaft hat, daß $y(0) = y^0$ und $y(2) = y^1$. Und dann suchen wir das Minimum entlang dieses geknickten Streckenzugs, welches immer eindeutig bestimmt ist.

Proposition 8.3 *Ist B positiv definit und ist $\|y^1\| \geq r$, dann gibt es genau einen Wert $t \in [0,2]$, so daß $\|y(t)\|_2 = r$ und für genau diesen Wert ist die Funktion $q\,(y(t))$ minimal unter der Nebenbedingung $\|y(t)\|_2 \leq r$.*

Beweis: Wir beweisen sogar viel mehr, wir zeigen nämlich, daß

$$t < t' \qquad \Longrightarrow \qquad \begin{cases} \|y(t)\|_2 \leq \|y(t')\|_2 \\ q\,(y(t)) \geq q\,(y(t')) \end{cases} \tag{8.3}$$

Daß (8.3) für $t \in [0,1]$ gilt, liegt an der Wahl von y^0: Das Minimum von $q\,(t\, y^0)$ wird ja gerade für $t = g^T g / g^T B g$ angenommen. Interessant ist also nur der Fall $t \in [1,2]$; schreiben wir $t = 1 + s$, $s \in [0,1]$, dann ist

$$\begin{aligned} \|y(t)\|_2^2 &= \|y(1+s)\|_2^2 = \left\|y^0 + s\left(y^1 - y^0\right)\right\|_2^2 \\ &= \left\|y^0\right\|_2^2 + 2s\left(y^1 - y^0\right)^T y^0 + s^2 \left\|y^1 - y^0\right\|_2^2 \end{aligned}$$

[178] Denn mit dieser Richtung, die $q(0)$ mit dem Minimum verbindet, wird das quadratische Optimierungsproblem in einem Iterationsschritt *global* gelöst.

[179] Encyclopedia Britannica: "**dog-leg** a thing that bends sharply, in particular a sharp bend in a road or route."

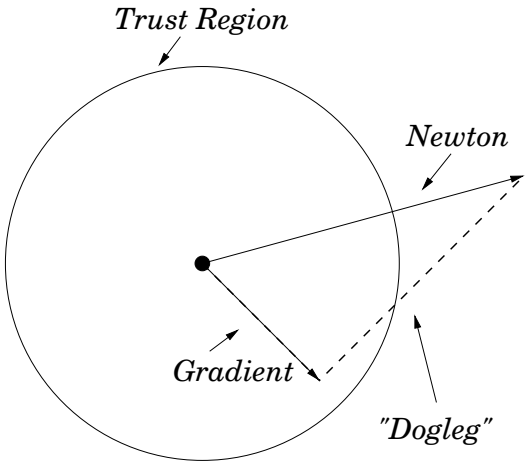


Abbildung 8.1: Der Pfad des "Dogleg"–Verfahrens und die Trust Region.

und daher

$$\begin{aligned}\frac{d}{ds}\frac{1}{2}\left\|y(1+s)\right\|_2^2 &= \left(y^1-y^0\right)^T y^0 + s\left\|y^1-y^0\right\|_2^2 \geq \left(y^1-y^0\right)^T y^0 \\ &= \frac{g^Tg}{g^TBg}\,g^T\left(B^{-1}g-\frac{g^Tg}{g^TBg}\,g\right) = \underbrace{\frac{g^Tg}{g^TBg}}_{\geq 0}\underbrace{g^TB^{-1}g}_{\geq 0}\left(1-\frac{g^Tg}{g^TBg}\,\frac{g^Tg}{g^TB^{-1}g}\right).\end{aligned}$$

Seien $0 < \lambda_1 \leq \cdots \leq \lambda_n$ die Eigenwerte von B, $b_1, \ldots, b_n$ die orthonormalen Eigenvektoren dazu und

$$g = \sum_{j=1}^{n} g_j b_j,$$

dann ist

$$g^TBg = \sum_{j,k=1}^{n} g_j g_k \underbrace{b_j^TBb_k}_{=\lambda_k\delta_{jk}} = \sum_{j=1}^{n}\lambda_j g_j^2 \qquad \text{und} \qquad g^TB^{-1}g = \sum_{j=1}^{n}\lambda_j^{-1}g_j^2$$

und so erhalten wir, da für $a, b > 0$

$$\frac{a^2+b^2}{ab} = \frac{(a-b)^2+2ab}{ab} \geq \frac{2ab}{ab} = 2,$$

erhalten wir

$$\left(g^TBg\right)\left(g^TB^{-1}g\right) = \sum_{j,k=1}^{n}\frac{\lambda_j}{\lambda_k}g_j^2\,g_k^2 = \sum_{j=1}^{n} g_j^4 + \sum_{1\leq j<k\leq n}\underbrace{\left(\frac{\lambda_j}{\lambda_k}+\frac{\lambda_k}{\lambda_j}\right)}_{=\frac{\lambda_j^2+\lambda_k^2}{\lambda_j\,\lambda_k}\geq 2} g_j^2\,g_k^2$$

$$\geq \sum_{j=1}^{n} g_j^4 + \sum_{1\leq j<k\leq n} 2g_j^2\, g_k^2 = \sum_{j,k=1}^{n} g_j^2 g_k^2 = \|g\|_2^4,$$

und so ist $\frac{d}{ds}\left\|y(1+s)\right\|_2^2 \geq 0$, was den ersten Teil von (8.3) liefert.
Für den zweiten Teil von (8.3) betrachten wir $y(1+s) = y^0 + s\left(y^1 - y^0\right)$ und

$$\begin{aligned}
&\frac{d}{ds} q\left(y(1+s)\right) \\
&= \frac{d}{ds}\left(f + g^T\left(y^0 + s\left(y^1 - y^0\right)\right) + \frac{1}{2}\left(y^0 + s\left(y^1 - y^0\right)\right)^T B\left(y^0 + s\left(y^1 - y^0\right)\right)\right) \\
&= g^T\left(y^1 - y^0\right) + \left(y^1 - y^0\right)^T B y^0 + s\,\underbrace{\left(y^1 - y^0\right)^T B\left(y^1 - y^0\right)}_{>0} \\
&\leq \left(y^1 - y^0\right)^T\left(g + B y^0\right) + \left(y^1 - y^0\right)^T B\left(y^1 - y^0\right) \\
&= \left(y^1 - y^0\right)^T\left(g + B y^1\right) = \left(y^1 - y^0\right)^T\left(g - BB^{-1}g\right) = 0,
\end{aligned}$$

weswegen wir ständig abfallende Werte erzeugen. □

Der Beweis zeigt: das Minimum auf dem "Dogleg"–Pfad wird genau dort angenommen, wo dieser Pfad die Trust Region verläßt!

8.3 Exakte Lösungen des quadratischen Problems

Cauchy–Punkte und Dogleg–Methode sind nette, aber recht heuristische Ansätze, um eine näherungsweise Optimallösung des quadratischen "Modellproblems" (8.2) zu bestimmen. Besser wäre es aber doch sicherlich, mit der *exakten* Lösung zu arbeiten. Und die kann man zumindest beschreiben.

Satz 8.4 *Ein Vektor $y^* \in \mathbb{R}^n$ ist genau dann Lösung von (8.2), wenn $\|y^*\| \leq r$ und es eine Zahl $\lambda \geq 0$ gibt, so daß*

$$\begin{aligned}
\left(B + \lambda I\right) y^* &= -g, && (8.4)\\
\lambda\left(r - \|y^*\|_2\right) &= 0, && (8.5)\\
\left(B + \lambda I\right) &\geq 0. && (8.6)
\end{aligned}$$

Als Hilfsmittel ein bißchen Analysis quadratischer Funktionen.

Lemma 8.5 *Sei $B \in \mathbb{R}^{n\times n}$ symmetrisch und $q(y) = g^T y + \frac{1}{2} y^T B y$. Dann gilt:*

1. *q besitzt genau dann ein* globales *Minimum, wenn B positiv semidefinit ist und $g \in B\,\mathbb{R}^n$.*
2. *q hat genau dann ein* eindeutiges *globales Minimum, wenn B positiv definit ist.*
3. *Ist B positiv semidefinit, dann ist jede Lösung y von $By = g$ ein globales Minimum von q.*

Beweis: 1): Hat q ein Minimum y^*, dann muß

$$0 = \nabla q\left(y^*\right) = g + By^* \qquad \text{und} \qquad 0 \leq \nabla^2 q\left(y^*\right) = B$$

weswegen $g \in B\,\mathbb{R}^n$ liegen und $B \geq 0$ gelten muß. Für die Umkehrung wählen wir ein $v \in \mathbb{R}^n$, so daß $g = -Bv$ – nach den Voraussetzungen $g \in B\,\mathbb{R}^n$ muß das ja funktionieren. Dann ist, für beliebiges $w \in \mathbb{R}^n$

$$\begin{aligned} q\left(v+w\right) &= g^T\left(v+w\right) + \frac{1}{2}\left(v+w\right)^T B\left(v+w\right) \\ &= g^T v + g^T w + \frac{1}{2} v^T B v + \underbrace{v^T B}_{=g^T} w + \frac{1}{2}\underbrace{v^T B v}_{\geq 0} \geq \underbrace{g^T v \frac{1}{2} v^T B v}_{=v} + g^T w - g^T w \\ &= q(v), \end{aligned}$$

womit v ein Minimum sein muß, was 3) im Übrigen gleich miterledigt.
2): Ist y^* ein striktes Minimum, so muß nach Proposition 5.2 $0 < \nabla^2 q\left(y^*\right) = B$ sein und umgekehrt ist für eine strikt positiv definite Matrix B ja $B\,\mathbb{R}^n = \mathbb{R}^n$ und in obiger Rechnung gilt die strikte Ungleichung. □

Beweis von Satz 8.4: Ohne Einschränkung nehmen wir an, daß $q(0) = 0$ ist – für die Suche nach dem Minimum ist der konstante Term irrelevant.
Wir beginnen mit "⇐" und nehmen an, es existiere ein $\lambda \geq 0$, das (8.4)–(8.6) erfüllt. Zusammen mit Lemma 8.5 ergeben (8.4) und (8.6), daß y^* ein globales Minimum der Funktion

$$q_\lambda(y) := g^T y + \frac{1}{2} y^T\left(B + \lambda I\right) y = \underbrace{g^T y + \frac{1}{2} y^T B y}_{=q(y)} + \frac{\lambda\,\|y\|_2^2}{2}$$

ist, es gilt also für alle $y \in \mathbb{R}^n$, daß

$$q\left(y^*\right) + \frac{\lambda}{2}\left\|y^*\right\|_2^2 \leq q\left(y\right) + \frac{\lambda}{2}\left\|y\right\|_2^2, \tag{8.7}$$

also für alle y mit $\|y\|_2 \leq r$

$$\begin{aligned} q\left(y\right) &\geq q\left(y^*\right) + \frac{\lambda}{2}\left(\left\|y^*\right\|_2^2 - \left\|y\right\|_2^2\right) = q\left(y^*\right) + \frac{\lambda}{2}\left(\left\|y^*\right\|_2^2 - r + r - \left\|y\right\|_2^2\right) \\ &= q\left(y^*\right) + \frac{1}{2}\underbrace{\lambda\left(\left\|y^*\right\|_2^2 - r^2\right)}_{=0} + \frac{1}{2}\underbrace{\lambda\left(r^2 - \left\|y\right\|_2^2\right)}_{\geq 0} \geq q\left(y^*\right), \end{aligned}$$

weswegen y^* eine Lösung von (8.2) sein muß.
Für die Umkehrung, "⇒", sei y^* eine Lösung von (8.2). Ist $\left\|y^*\right\|_2 < r$, dann muß nach (8.5) $\lambda = 0$ sein und dann ergibt sich (8.4) aus $0 = \nabla q\left(y^*\right) = g + By^*$ sowie (8.6) aus $0 \leq \nabla^2 q\left(y^*\right) = B$.

Interessant wird es also, wenn $\|y^*\|_2 = r$ ist, dann müssen wir die Existenz des ominösen $\lambda > 0$ nachweisen. Nun ist aber y^* auch eine Lösung des restringierten Optimierungsproblems

$$\min_y q(y), \qquad \underbrace{\frac{1}{2}\left(\|y\|_2^2 - r^2\right)}_{=:g(y)} = 0,$$

und nach unserem Multiplikatoren–Satz 4.13 muß es ein $\lambda \in \mathbb{R}$ geben, so daß[180]

$$0 = \nabla q\left(y^*\right) + \nabla g\left(y^*\right)\lambda = g + By^* + \lambda y^* = g + \left(B + \lambda I\right)\, y^*,$$

also muß $g = -\left(B + \lambda I\right) y^*$ gelten. Unter Verwendung von (8.7) gilt somit wegen der Minimalität von y^*

$$\|y\|_2 = r \qquad \Longrightarrow \qquad q(y) \geq q\left(y^*\right) + \frac{\lambda}{2}\left(\|y^*\|_2^2 - \|y\|_2^2\right),$$

also

$$\begin{aligned}
0 &\leq q(y) - q\left(y^*\right) - \frac{\lambda}{2}\left(\|y^*\|_2^2 - \|y\|_2^2\right) \\
&= g^T y + \frac{1}{2} y^T B y - g^T y^* - \frac{1}{2} y^{*T} B y^* - \frac{1}{2} y^{*T}\left(\lambda I\right) y^* + \frac{1}{2} y^T\left(\lambda I\right) y \\
&= g^T\left(y - y^*\right) + \frac{1}{2}\left(y - y^*\right)^T\left(B + \lambda I\right)\left(y - y^*\right) + y^T\left(B + \lambda I\right) y^* - y^{*T}\left(B + \lambda I\right) y^* \\
&= -\left(y - y^*\right)^T\left(B + \lambda I\right) y^* + \frac{1}{2}\left(y - y^*\right)^T\left(B + \lambda I\right)\left(y - y^*\right) + \left(y - y^*\right)^T\left(B + \lambda I\right) y^* \\
&= \frac{1}{2}\left(y - y^*\right)^T\left(B + \lambda I\right)\left(y - y^*\right),
\end{aligned}$$

also ist $\left(B + \lambda I\right)$ positiv semidefinit, weil die Vektoren

$$\left\{ \pm \frac{y - y^*}{\|y - y^*\|_2} \;:\; \|y\|_2 = r \right\}$$

eine dichte Teilmenge der Einheitskugel bilden. Bleibt zu zeigen, daß $\lambda \geq 0$ ist. Da (8.4) und (8.6) erfüllt sind, sagt uns Lemma 8.5, 3), daß y^* ein globales Minimum von

$$q_\lambda(y) = g^T y + \frac{1}{2} y^T\left(B + \lambda I\right) y$$

ist. Könnten wir nun nur $\lambda < 0$ wählen, dann liefert uns wieder einmal (8.7), daß für alle $y \in \mathbb{R}^n$, $\|y\|_2 > r = \|y^*\|_2$, daß

$$q(y) \geq q\left(y^*\right) + \underbrace{\overbrace{\frac{\lambda}{2}}^{<0}\ \overbrace{\left(\|y^*\|_2^2 - \|y\|_2^2\right)}^{<0}}_{>0} > q\left(y^*\right),$$

[180] Und hier ersetzen wir das λ in (4.9) durch $-\lambda$.

und da y^* schon das Minimum auf $\{y \,:\, \|y\| \le r\}$ war, ist also y^* ein *globales* Minimum von q. Nach Lemma 8.5, 1), wäre dann aber $g = -By^*$ und B wäre positiv semidefinit und wir könnten, im Widerspruch zu unserer Annahme, eben doch $\lambda = 0$ wählen. □

Jetzt können wir also mit Hilfe von Satz 8.4 das "lokalisierte" Optimierungsproblem (8.2) in Angriff nehmen:

1. Wir bestimmen zuerst y als Lösung von[181] $By = -g$ und testen, ob $\|y\| < r$. Wenn ja, dann können wir nach (8.5) $\lambda = 0$ wählen, und y ist die gesuchte Lösung, außerdem ist B positiv semidefinit.

2. Ansonsten müssen wir einen Wert $\lambda > 0$ bestimmen, so daß $B + \lambda I$ positiv semindefinit, besser (strikt) positiv definit, ist und dann $(B + \lambda I)\, y(\lambda) = -g$ lösen. Allerdings, und das macht die Sache interessant, muß gleichzeitig $\|y(\lambda)\| = r$ gelten.

Schauen wir uns also mal an, warum es so ein λ immer geben muß. Da B eine *symmetrische* Matrix ist, gibt es eine *orthogonale* Matrix $Q \in \mathbb{R}^{n\times n}$, $Q^TQ = QQ^T = I$, mit orthogonalen Spaltenvektoren $q_j \in \mathbb{R}^n$, $j = 1, \ldots, n$, so daß

$$Q^T B Q = \Lambda = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix}, \qquad \lambda_1 \le \cdots \le \lambda_n,$$

wobei nicht unbedingt $\lambda_1 \ge 0$ gelten muß – wir haben *nicht* vorausgesetzt, daß $B \ge 0$ sein soll! Ist nun $\lambda > 0$ so gewählt, daß

$$B + \lambda I = Q\Lambda Q^T + \lambda Q Q^T = Q\,(\Lambda + \lambda I)\, Q^T$$

invertierbar ist[182], dann ist, mit $Q = \sum q_j\, e_j^T$,

$$\begin{aligned} y(\lambda) &= -\,(B + \lambda I)^{-1}\, g = -Q\,(\Lambda + \lambda I)^{-1}\, Q^T g \\ &= -\left(\sum_{j=1}^n q_j\, e_j^T\right)(\Lambda + \lambda I)^{-1} \left(\sum_{k=1}^n q_k\, e_k^T\right)^T g \\ &= sum_{j,k=1}^n q_j\, \underbrace{e_j^T \begin{bmatrix} (\lambda_1+\lambda)^{-1} & & \\ & \ddots & \\ & & (\lambda_n+\lambda)^{-1} \end{bmatrix} e_k}_{\delta_{jk}(\lambda_j+\lambda)^{-1}}\, q_k^T g \\ &= -\sum_{j=1}^n \frac{q_j\, q_j^T}{\lambda_j + \lambda}\, g = -\sum_{j=1}^n \frac{q_j^T\, g}{\lambda_j + \lambda}\, q_j \end{aligned}$$

[181] Hier können wir unser "Lieblingsverfahren" verwenden. Symmetrische Matrizen, vor allem dann, wenn sie auch noch positiv semidefinit sind, sind ja dankbare Kandidaten für die Cholesky–Zerlegung und für das iterative Gauß–Seidel–Verfahren.

[182] Was, unabhängig von Λ für alle $\lambda \in \mathbb{R}$, abgesehen von endlich vielen Ausnahmen, gilt.

was uns, wegen der Orthogonalität der q_j

$$\|y(\lambda)\|_2^2 = \sum_{j,k=1}^{n} \frac{q_j^T g}{\lambda_j + \lambda} \frac{q_k^T g}{\lambda_k + \lambda} \underbrace{q_j^T q_k}_{=\delta_{jk}} = \sum_{j=1}^{n} \frac{\left(q_j^T g\right)^2}{\left(\lambda_j + \lambda\right)^2} \tag{8.8}$$

liefert. Ist nun $q_1^T g \neq 0$, dann hat diese Funktion eine Singularität an $\lambda = -\lambda_1$, ist aber für $\lambda \in (-\lambda_1, \infty)$ wohldefiniert und erfüllt $\|y(\lambda)\|_2 \to 0$ für $\lambda \to \infty$. Mit anderen Worten:

Ist $q_1^T g \neq 0$, dann gibt es ein $\lambda \in (-\lambda_1, \infty)$, so daß $\|y(\lambda)\|_2 = r$.

Was aber passiert, wenn $q_1^T g = 0$ ist? Auch kein Beinbruch, denn dann beginnt eben erst bei $j = 2$, oder, wenn allgemein $q_1^T g = \cdots = q_k^T g = 0$, eben bei $j = k + 1$ und wir finden dann halt ein $\lambda \in (-\lambda_{k+1}, \infty)$, das die gewünschte Eigenschaft hat. Allerdings: Die positive Semidefinitheit setzt immer noch voraus, daß $\lambda \geq -\lambda_1$ ist.
Zur Berechnung von λ könnte (und wird) man nun wieder das Newton–Verfahren verwenden, um eine Nullstelle von $F(\lambda) = \|y(\lambda)\|_2 - r$ zu berechnen. Dabei taucht aber ein kleines Problem auf: Für $\lambda \sim -\lambda_1$ ist $F(\lambda) \sim (\lambda + \lambda_1)^{-1}$, was uns noch nicht einmal lokale Konvergenz des Newton–Verfahrens garantiert, denn F' und F'' sind in einer Umgebung von $-\lambda_1$ unbeschränkt. Das ist aber halb so wild, dann betrachten wir eben

$$F(\lambda) = \frac{1}{\|y(\lambda)\|} - \frac{1}{r}, \tag{8.9}$$

die sich um $-\lambda_1$ wie $\lambda + \lambda_1 + C$ verhält – also wesentlich anständiger. Für die Newton–Iteration

$$\lambda_{k+1} = \lambda_k - \frac{F(\lambda_k)}{F'(\lambda_k)}, \qquad k \in \mathbb{N}_0,$$

brauchen wir also die Ableitung F', die sich mit Hilfe von (8.8) als

$$\begin{aligned} F'(\lambda) &= \frac{d}{d\lambda} \left(\|y(\lambda)\|_2^2\right)^{-1/2} = -\frac{1}{2} \left(\|y(\lambda)\|_2^2\right)^{-3/2} \frac{d}{d\lambda} \|y(\lambda)\|_2^2 \\ &= -\frac{1}{2} \left(\|y(\lambda)\|_2^2\right)^{-3/2} \frac{d}{d\lambda} \left(\sum_{j=1}^{n} \frac{\left(q_j^T g\right)^2}{\left(\lambda_j + \lambda\right)^2} \right) = -\frac{1}{2} \left(\|y(\lambda)\|_2^2\right)^{-3/2} \left(\sum_{j=1}^{n} -2 \frac{\left(q_j^T g\right)^2}{\left(\lambda_j + \lambda\right)^3} \right) \\ &= \|y(\lambda)\|_2^{-3} \left(\sum_{j=1}^{n} \frac{\left(q_j^T g\right)^2}{\left(\lambda_j + \lambda\right)^3} \right) \end{aligned}$$

bestimmen läßt. Für den zweiten Term sei $B + \lambda I = G_\lambda^T G_\lambda$ die Cholesky–Zerlegung von $B + \lambda I$ und wir setzen $z(\lambda) = G_\lambda^{-T} y(\lambda)$. Dann ist

$$\begin{aligned} \|z(\lambda)\|_2^2 &= \left(G_\lambda^{-T} y(\lambda)\right)^T \left(G_\lambda^{-T} y(\lambda)\right) = y^T \underbrace{G_\lambda^{-1} G_\lambda^{-T}}_{=(B+\lambda I)^{-1}} y \\ &= \left(-(B + \lambda I)^{-1} g\right)^T (B + \lambda I)^{-1} \left(-(B + \lambda I)^{-1} g\right) = g^T (B + \lambda I)^{-3} g \end{aligned}$$

$$
\begin{aligned}
&= g^T Q \underbrace{\left(Q^T \left(B+\lambda I\right)^{-1} Q\right)^3}_{=(\Lambda+\lambda I)^{-3}} Q^T g \\
&= \left(\sum_{j=1}^{n} q_j^T g\, e_j\right)^T \begin{bmatrix} (\lambda_1+\lambda)^{-3} & & \\ & \ddots & \\ & & (\lambda_n+\lambda)^{-3} \end{bmatrix} \left(\sum_{j=1}^{n} q_j^T g\, e_j\right) \\
&= \sum_{j=1}^{n} \frac{\left(q_j^T g\right)^2}{\left(\lambda_j+\lambda\right)^3}
\end{aligned}
$$

und somit

$$
\frac{F(\lambda)}{F'(\lambda)} = \left(\frac{1}{\|y(\lambda)\|_2} - \frac{1}{r}\right) \frac{\|y(\lambda)\|_2^3}{\|z(\lambda)\|_2^2} = \left(\frac{\|y(\lambda)\|_2}{\|z(\lambda)\|_2}\right)^2 \frac{r-\|y(\lambda)\|_2}{r},
$$

was sich nun ganz gut als Newton–Iteration verwenden läßt:

$$
\lambda_{k+1} = \lambda_k - \left(\frac{\|y\left(\lambda_k\right)\|_2}{\|z\left(\lambda_k\right)\|_2}\right)^2 \frac{r-\|y\left(\lambda\right)\|_2}{r}, \qquad k \in \mathbb{N}_0. \tag{8.10}
$$

Um dafür zu sorgen, daß $B+\lambda_k I$ auch immer positiv definit ist, empfiehlt es sich, mit *großen* Startwerten von λ zu beginnen, indem man z.B.

$$
\lambda_0 \geq \rho(B) := \max\left\{|\mu| \ : \ \ker\left(B-\mu I\right) \neq \{0\},\, \mu \in\right\}
$$

wählt, wobei $\rho(B)$ den *Spektralradius* der Matrix B bezeichnet; für diesen Wert gibt es Abschätzungen, die man mit verhältnismäßig geringem Aufwand berechnen kann.

8.4 Konvergenz von Trust–Region–Verfahren

Wir zeigen nun, daß die Trust–Region–Verfahren unter bestimmten Voraussetzungen tatsächlich gegen eine Minimallösung konvergieren. Dazu nehmen wir zuerst an, daß wir den linearen Teil des quadratischen Modells *exakt* wählen, d.h., $f_k = f\left(x^{(k)}\right)$ und $g_k = \nabla f\left(x^{(k)}\right)$, also

$$
q_k(y) = f\left(x^{(k)}\right) + \nabla^T f\left(x^{(k)}\right)\, y + \frac{1}{2} y^T B_k y. \tag{8.11}
$$

Außerdem seien $0 < \rho_- < \rho_+ < 1$ die Schwellenwerte für ρ_k, nach denen entschieden wird, ob ρ_k als "groß" ($> \rho_+$) oder als "klein" ($< \rho_-$) angesehen wird.

Satz 8.6 *Ist $f \in C^1\left(\mathbb{R}^n\right)$ nach unten beschränkt und gibt es eine Konstante $\beta > 0$, so daß $\|B_k\|_2 \leq \beta$, $k \in \mathbb{N}_0$, dann gilt für das Trust–Region–Verfahren mit den exakten Lösungen von (8.2), daß*

$$
\liminf_{k\to\infty} \left\|\nabla f\left(x^{(k)}\right)\right\| = 0. \tag{8.12}
$$

Zuerst halten wir fest, daß Cauchy–Punkte und alles, was "besser" als diese ist, für eine "spürbare" Verbesserung des Modells sorgen.

Lemma 8.7 *Für das quadratische Modell (8.11) und den zugehörigen Cauchy–Punkt $y = y_c(r)$ gilt*

$$q_k(0) - q_k(y) \geq \frac{1}{2} \|\nabla f(x)\|_2 \min\left(r, \frac{\|\nabla f(x)\|_2}{\|B\|_2}\right) \tag{8.13}$$

Da "Dogleg" und exakte Lösung ja nur zu Verbesserung des Ergebnisses führen, das man mit Hilfe des Cauchy–Punkts gewinnt, erhalten wir unmittelbar das folgende Resultat.

Korollar 8.8 *Die Abschätzung (8.13) gilt auch für $y(r)$, das mit Hilfe der Dogleg–Methode oder durch exakte Lösung des Modellproblems erhalten wird.*

Beweis von Lemma 8.7: Wir schreiben $g = \nabla f(x)$, was uns die Richtung $y^* = -r\frac{g}{\|g\|_2}$ liefert. Dann ist

$$q(\tau y^*) - q(0) = -\tau\, r\|g\|_2 + \tau^2 \frac{r^2}{\|g\|_2^2} g^T B g.$$

Dieser Ausdruck ist monoton fallend in τ falls $g^T B g \leq 0$ ist — in diesem Fall wählen wir $\tau^* = 1$ — und eine konvexe quadratische Funktion in τ wenn $g^T B g > 0$ ist. In diesem zweiten Fall wird das globale Minimum für

$$\tau^* = \frac{r\|g\|_2}{2} \left(r\frac{g^T B g}{\|g\|_2^2}\right)^{-1} = \frac{1}{r}\frac{\|\nabla f(x)\|_2^3}{g^T B g} \tag{8.14}$$

oder $\tau^* = 1$ angenommen – je nachdem welcher der beiden Werte eher kommt. Und genau diese drei Fälle müssen wir jetzt (natürlich) auch unterscheiden.

1. Ist $g^T B g \leq 0$, also $\tau^* = 1$, dann ist

$$\begin{aligned} q(y_c(r)) - q(0) &= -r\|g\|_2 + \frac{1}{2}\frac{r^2}{\|g\|_2^2}\underbrace{g^T B g}_{\leq 0} \leq -r\|g\|_2 \leq -\frac{1}{2} r\|g\|_2 \\ &\leq -\frac{1}{2}\|\nabla f(x)\|_2 \min\left(r, \frac{\|\nabla f(x)\|_2}{\|B\|_2}\right) \end{aligned}$$

2. Ist $g^T B g > 0$ und erfüllt das τ^* aus (8.14) die Bedingung $\tau^* \leq 1$, dann ist

$$\begin{aligned} q(y_c(r)) - q(0) &= -(r\,\tau^*)\,\|g\|_2 + \frac{1}{2}\frac{(r\,\tau^*)^2}{\|g\|_2^2} g^T B g \\ &= -\frac{\|g\|_2^3}{g^T B g}\|g\|_2 + \frac{1}{2}\frac{\|g\|_2^6}{(g^T B g)^2}\frac{g^T B g}{\|g\|^2} = -\frac{1}{2}\frac{\|g\|_2^4}{g^T B g} \leq -\frac{1}{2}\frac{\|g\|_2^4}{\|B\|_2\|g\|_2^2} = -\frac{1}{2}\frac{\|g\|_2^2}{\|B\|_2} \\ &\leq -\frac{1}{2}\|\nabla f(x)\| \min\left(r, \frac{\|\nabla f(x)\|_2}{\|B\|_2}\right) \end{aligned}$$

3. Ist schließlich $g^T Bg > 0$, aber erfüllt das τ^* aus (8.14) die Bedingung $\tau^* > 1$, das heißt also, daß $g^T Bg < \|g\|_2^3/r$, dann müssen wir $\tau^* = 1$ wählen und erhalten, daß

$$\begin{aligned} q\left(y_c(r)\right) - q(0) &= -r\|g\|_2 + \frac{1}{2}\frac{r^2}{\|g\|_2^2} g^T Bg < -r\|g\|_2 + \frac{1}{2}\frac{r^2}{\|g\|_2^2}\,\frac{\|g\|_2^3}{r} \\ &= -r\|g\|_2 + \frac{1}{2}r\|g\|_2 = -\frac{1}{2}r\|g\|_2 \leq -\frac{1}{2}\left\|\nabla f(x)\right\| \min\left(r, \frac{\|\nabla f(x)\|_2}{\|B\|_2}\right). \end{aligned}$$

In jedem dieser Fälle folgt (8.13) durch Multiplikation mit -1. □

Beweis von Satz 8.6: Wir bemerken zuerst, daß wegen $q_k(0) = f\left(x^{(k)}\right)$

$$\begin{aligned} \left|\frac{q_k\left(y^{(k)}\right) - f\left(x^{(k)} + y^{(k)}\right)}{q_k(0) - q_k\left(y^{(k)}\right)}\right| &= \left|\frac{q_k\left(y^{(k)}\right) - q_k(0) + f\left(x^{(k)}\right) - f\left(x^{(k)} + y^{(k)}\right)}{q_k(0) - q_k\left(y^{(k)}\right)}\right| \\ &= \left|\frac{f\left(x^{(k)}\right) - f\left(x^{(k)} + y^{(k)}\right)}{q_k(0) - q_k\left(y^{(k)}\right)} - \frac{q_k(0) - q_k\left(y^{(k)}\right)}{q_k(0) - q_k\left(y^{(k)}\right)}\right| \\ &= \left|\rho_k - 1\right|. \end{aligned} \tag{8.15}$$

Nach der Taylorformel ist außerdem

$$f\left(x^{(k)} + y^{(k)}\right) = \underbrace{f\left(x^{(k)}\right) + \nabla^T f\left(x^{(k)}\right) y^{(k)}}_{= q_k\left(y^{(k)}\right) - \frac{1}{2} y^{(k)T} B_k y^{(k)}} + \int_0^1 \left(\nabla f\left(x^{(k)} + t y^{(k)}\right) - \nabla f\left(x^{(k)}\right)\right)^T y^{(k)}\, dt,$$

weswegen sich

$$\begin{aligned} &\left|q_k\left(y^{(k)}\right) - f\left(x^{(k)} + y^{(k)}\right)\right| \\ &= \left|\frac{1}{2} y^{(k)T} B_k y^{(k)} - \int_0^1 \left(\nabla f\left(x^{(k)} + t y^{(k)}\right) - \nabla f\left(x^{(k)}\right)\right)^T y^{(k)}\, dt\right| \\ &\leq \left|\frac{1}{2} y^{(k)T} B_k y^{(k)}\right| + \left|\int_0^1 \left(\nabla f\left(x^{(k)} + t y^{(k)}\right) - \nabla f\left(x^{(k)}\right)\right)^T y^{(k)}\, dt\right| \\ &\leq \frac{\beta}{2}\left\|y^{(k)}\right\|_2^2 + \omega\left(\nabla f, \left\|y^{(k)}\right\|_2\right)\left\|y^{(k)}\right\|_2 \qquad (8.16) \\ &= \left\|y^{(k)}\right\|_2 \left(\frac{\beta}{2}\left\|y^{(k)}\right\|_2 + \omega\left(\nabla f, \left\|y^{(k)}\right\|_2\right)\right) \leq r_k\left(\frac{\beta}{2} r_k + \omega\left(\nabla f, r_k\right)\right) \qquad (8.17) \end{aligned}$$

ergibt, wobei der *Stetigkeitsmodul* $\omega(F, \delta)$ für $F \in C\left(\mathbb{R}^n\right)^n$ als

$$\omega\left(F, \delta\right) = \sup_x \sup_{\|d\| < \delta} \left\|F\left(x + d\right) - F(x)\right\|_2$$

definiert ist und $\omega(F, \delta) \to 0$ für $\delta \to 0$ erfüllt.

Nach all den Vorbemerkungen nehmen wir nun an, daß (8.12) *nicht* erfüllt wäre, das heißt, es gibt ein $\varepsilon > 0$, so daß $\|\nabla f_k\| \geq \varepsilon$, $k \in \mathbb{N}_0$. Nach Lemma 8.7 ist dann auch der Nenner von (8.15) nach unten beschränkt, denn es ist

$$q_k(0) - q_k\left(y^{(k)}\right) \geq \frac{1}{2}\,\|\nabla f_k\|_2\,\min\left(r_k, \frac{\|\nabla f_k\|_2}{\|B\|}\right) \geq \frac{\varepsilon}{2}\min\left(r_k, \frac{\varepsilon}{\beta}\right). \tag{8.18}$$

Setzen wir nun (8.18) und (8.17) in (8.15) ein, dann erhalten wir, daß

$$|\rho_k - 1| \leq \frac{r_k\left(r_k\,\beta + 2\omega\left(\nabla f, r\right)\right)}{\varepsilon\,\min\left(r_k, \varepsilon/\beta\right)} \tag{8.19}$$

Ist nun $r_k < \epsilon/\beta$, dann wird (8.19) zu

$$|\rho_k - 1| \leq \frac{r_k\beta + 2\omega\left(\nabla f, r\right)}{\varepsilon}$$

und es gibt eine Schranke $\bar{r}$, so daß für jedes $r < \bar{r}$ die Ungleichung $|\rho_k - 1| < |1 - \rho_+|$ erfüllt ist, was dazu führen würde, daß $r_{k+1} > r_k$ ist[183]. Das heißt aber, daß eine Verkleinerung der Trust Region nur dann eintreten kann, wenn $r_k > \bar{r}$ ist, dann aber mit Sicherheit wieder vergrößert werden muß. Sei $0 < \gamma < 1$ dieser Verkleinerungsfaktor[184], dann erhalten wir, daß

$$r_k \geq \min\left(r_0, \gamma\,\bar{r}\right), \qquad k \in \mathbb{N}_0, \tag{8.20}$$

die Radien der Trust Regions sind also nach unten beschränkt. Insbesondere bedeutet (8.20), daß unendlich oft $\rho_k > \rho_-$ gelten muß, denn sonst würde ja für $k \to \infty$ die Folge $r_k \to 0$ konvergieren. Nehmen wir also an, daß, nach eventuellem Übergang zu einer Teilfolge, $\rho_k > \rho_-$, $k \in \mathbb{N}_0$, gilt, dann erhalten wir, daß

$$f\left(x^{(k)}\right) - f\left(x^{(k+1)}\right) \geq \rho_-\left(q_k(0) - q_k\left(y^{(k)}\right)\right) \geq \frac{\rho_-\,\varepsilon}{2}\,\min\left(r_k, \frac{\varepsilon}{\beta}\right)$$

und somit, weil f nach unten beschränkt ist, gibt es ein $C > 0$ so daß

$$\begin{aligned} C \;&>\; f\left(x^{(0)}\right) - f\left(x^{(k+1)}\right) = \sum_{j=0}^{k}\left(f\left(x^{(j)}\right) - f\left(x^{(j+1)}\right)\right) \geq \sum_{j=0}^{k}\frac{\rho_-\,\varepsilon}{2}\,\min\left(r_j, \frac{\varepsilon}{\beta}\right) \\ &=\; \frac{\rho_-\,\varepsilon}{2}\sum_{r_j < \varepsilon/\beta} r_j + \frac{\rho_-\,\varepsilon^2}{2\beta}\,\#\left\{j \;:\; j \leq k,\, r_j \geq \frac{\varepsilon}{\beta}\right\}, \end{aligned}$$

was für *alle* $k \in \mathbb{N}_0$ gelten muß. Mit $k \to \infty$ erhalten wir somit, daß

$$\#\left\{j \;:\; r_j \geq \frac{\varepsilon}{\beta}\right\} < \infty$$

und somit

$$\sum_{j=0}^{\infty} r_j < \infty \qquad \Longrightarrow \qquad \lim_{j\to\infty} r_j = 0$$

ist, was den langersehnten Widerspruch zu (8.20) liefert, weswegen (8.12) eben doch erfüllt sein muß. □

[183] Beispielsweise, indem man dann, wie auf Seite 151, $r_{k+1} = 2r_k$ wählt.

[184] Im Beispiel auf Seite 151 war dies $\gamma = \frac{1}{4}$.

Uns ist in alten mæren
wunders viel geseit
von Helden lobebæren
von grôzer arebeit

Das Nibelungenlied

8 Literatur

[1] P. Ablay, *Optimieren mit Evolutionsstragien*, Computer–Anwendungen, Spektrum der Wissenschaft: Verständliche Forschung, Spektrum–Verlag, 1989, pp. 162–174.

[2] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen, *LAPACK user's guide*, second ed., SIAM, 1995.

[3] E. R. Barnes, *A variation of Karmarkar's algorithm for solving linear programming problems*, Math. Prog. **36** (1986), 174–182.

[4] C. de Boor, *Computational aspects of multivariate polynomial interpolation: Indexing the coefficients*, Advances Comput. Math. **12** (2000), 289–301.

[5] E. Brieskorn, *Lineare Algebra und Analytische Geometrie II*, Vieweg, 1985.

[6] C. G. Broyden, *A class of methods for solving nonlinear simulaneous equations*, Math. Comp. **19** (1965), 577–593.

[7] P. G. Ciarlet and P. A. Raviart, *General Lagrange and Hermite interpolation in $\mathbb{R}^n$ with applications to finite element methods*, Arch. Rational Mech. Anal. **46** (1972), 178–199.

[8] A. R. Conn, N. I. M Gould, and Ph. L. Toint, *LANCELOT: a FORTRAN package for large–scale nonlinear optimization*, Springer Series in Computational Mathematics, vol. 17, Springer–Verlag, 1992.

[9] A. R. Conn, K. Scheinfeld, and Ph. L. Toint, *On the convergence of derivative–free methods for unconstrained optimization*, Approximation Theory and Optimization – Tributes to M. J. D. Powell (M. D. Buhmann and A. Iserles, eds.), Cambridge University Press, 1997, pp. 83–108.

[10] J. W. Cooley and J. W. Tukey, *An algorithm for machine calculation of complex Fourier series*, Math. Comp. **19** (1965), 297–301.

[11] D. Cox, J. Little, and D. O'Shea, *Using algebraic geometry*, Graduate Texts in Mathematics, vol. 185, Springer Verlag, 1998.

[12] G. B. Dantzig, *Linear programming and extensions*, Pinceton University Press, 1963.

[13] W. C. Davidon, *Variable metric method for minimization*, Tech. Report ANL-5990, Argonne National Laboratory, Argonne, Il, 1959.

[14] ———, *Variable metric method for minimization*, SIAM J. Optimization **1** (1991), 1–17.

[15] I. I. Dikin, *Iterative solution of problems of linear and quadratic programming*, Soviet Math. Doklady **8** (1967), 674–675.

[16] Duden, *Rechnen und Mathematik. das Lexikon für Schule und Praxis*, 3. ed., Bibliographisches Institut Mannheim/Wien/Zürich, 1969.

[17] G. Dueck, T. Scheuer, and H.-M. Wallmeier, *Toleranzschwelle und Sintflut: neue Ideen zur Optimierung*, Spektrum der Wissenschaft **1993/3** (1993), 42–51.

[18] R. Fletcher and C. M. Reeves, *Function minimization by conjugate gradients*, Computer Journal **7** (1964), 149–154.

[19] M. Gasca and T. Sauer, *Polynomial interpolation in several variables*, Advances Comput. Math. **12** (2000), 377–410, to appear.

[20] S. I. Gass, *An illustrated guide to linear programming*, McGraw–Hill, 1970, Republished by Dover, 1990.

[21] ———, *An illustrated guide to linear programming*, McGraw–Hill, 1970, Republished by Dover 1990.

[22] G. Golub and C. F. van Loan, *Matrix computations*, 3rd ed., The Johns Hopkins University Press, 1996.

[23] M. Guingard, *Generalized Kuhn–Tucker conditions for mathematical programmin in a Banach space*, SIAM J. Control **7** (1969), 232–241.

[24] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, Journal of Research of the National Bureau of Standards **49** (1952), 409–436.

[25] H. Heuser, *Lehrbuch der Analysis. Teil 2*, 2. ed., B. G. Teubner, 1983.

[26] N. J. Higham, *Accuracy and stability of numerical algorithms*, SIAM, 1996.

[27] W. Hoffmann, *B–splinekurven zur zeitoptimalen Robotersteuerung*, Master's thesis, Universität Erlangen, 2001, Zulassungsarbeit.

[28] S. Hoşten and R. Thomas, *Gröbner bases and integer programming*, Gröbner bases and applications (B. Buchberger and F. Winkler, eds.), Cambridge University Press, 1998, pp. 144–158.

[29] E. Isaacson and H. B. Keller, *Analysis of Numerical Methods*, John Wiley & Sons, 1966.

[30] N. Karmarkar, *A new polyomial–time algorithm for linear programming*, Combinatorica **4** (1984), 373–395.

[31] K. S. Kunz, *Numerical Analysis*, McGraw-Hill Book Company, 1957.

[32] A. Langenbucher, T. Sauer, G. J. van der Heyd, A. Viestenz, and B. Seitz, *Raytracing von Hornhauttopographiedaten zur Ermittlung der optischen Abbildungsqualität des Auges*, Klin. Monatsb. Ausgenheilkd. **220** (2003), 1–12.

[33] G. J. Minty and V. Klee, *How good is the simplex algorithm*, Inequalities – III (O. Shisha, ed.), Academic Press, 1972.

[34] J. von Neumann, *Zur Theorie der Gesellschaftsspiele*, Math. Annalen **100** (1928), 295–320.

[35] J. von Neumann and O. Morgenstern, *Theory of games and economic behavior*, sixth paperback printing, 1990 ed., Princeton University Press, 1944.

[36] J. Nocedal and S. J Wright, *Numerical optimization*, Springer Series in Operations Research, Springer, 1999.

[37] M. J. D. Powell, *Some global convergence properties of a variable matric algorithm for minimization without exact line searches*, SIAM–AMS Proceedings 9: Nonlinear Programming (1976), 53–72.

[38] Pschyrembel, *Klinisches wörterbuch*, 257 ed., Walter de Gruyter & Co, 1994.

[39] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, 1970.

[40] T. Sauer, *Computational aspects of multivariate polynomial interpolation*, Advances Comput. Math. **3** (1995), no. 3, 219–238.

[41] _____, *Numerische Mathematik I*, Vorlesungsskript, Friedrich–Alexander–Universität Erlangen–Nürnberg, Justus–Liebig–Universität Gießen, 2000, `http://www.math.uni-giessen.de/tomas.sauer`.

[42] _____, *Numerische Mathematik II*, Vorlesungsskript, Friedrich–Alexander–Universität Erlangen–Nürnberg, Justus–Liebig–Universität Gießen, 2000, `http://www.math.uni-giessen.de/tomas.sauer`.

[43] _____, *Spieltheorie*, Vorlesungsskript, Justus–Liebig–Universität Gießen, 2005, `http://www.math.uni-giessen.de/tomas.sauer`.

[44] H. R. Schwarz, *Numerische Mathematik*, B. G. Teubner, Stuttgart, 1988.

[45] P. Spellucci, *Numerische Verfahren der nichtlinearen Optimierung*, Internationale Schriftenreihe zu Numerischen Mathematik, Birkhäuser, 1993.

[46] J. Stoer, *Einführung in die Numerische Mathematik I*, 4 ed., Heidelberger Taschenbücher, Springer Verlag, 1983.

[47] J. Werner, *Numerische Mathematik 2. Eigenwertaufgaben, lineare Optimierungsaufgaben, unrestringierte Optimierungsaufgaben*, Vieweg, 1992.

[48] P. Wolfe, *Convergence conditions for ascent methods*, SIAM Review **11** (1969), 220–228.

[49] Yinyu Ye, *Interior points algorithms. Theory and analysis*, John Wiley & Sons, 1997.

8 Index