

Einführung in die Optimierung für Hörer aller Fachbereiche

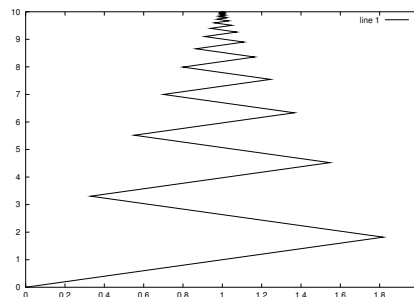
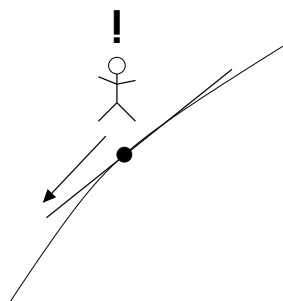
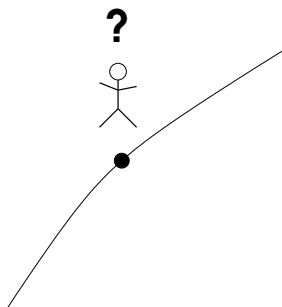
Vorlesung, zuerst gehalten im Sommersemester 2003

Tomas Sauer

Lehrstuhl für Numerische Mathematik
Justus–Liebig–Universität Gießen
Heinrich–Buff–Ring 44
D-35392 Gießen

Version 1.1

Letzte Änderung: 18.4.2005



Die Wissenschaften sind nicht wie Minerva, welche vollständig bewaffnet dem Haupte Jupiters entsprang. Sie sind die Töchter der Zeit und bilden sich langsam, zuerst durch Sammlung der Methoden, welche die Erfahrung angibt, und später durch Entdeckung der Principien, die aus der Combination der Methoden sich folgern lassen.

Die Greise, welche man ihrer Erfahrung wegen zum Bett der Kranken berief und die aus Mitleiden die Wunden verbanden, waren die ersten Aerzte.

Die ägyptischen Schäfer, welche die Beobachtung machten, dass einzelne Sterne nach einer gewissen Umlaufzeit wieder zu demselben Punkte der Himmels zurückkehrten, waren die ersten Astronomen.

Der Erste, der durch Zeichen jenes einfache Verhältniss $2 \times 2 = 4$ ausdrückte, erfand die Mathematik, jene mächtige Wissenschaft, welche wirklich den Menschen auf den Thron der Welt erhob.

J. A. Brillat-Savarin, *Physiologie des Geschmacks*

Inhaltsverzeichnis

0

1	Optimierung ist überall	4
1.1	Die allgemeine mathematische Formulierung	4
1.2	Lineare Optimierungsprobleme	6
1.3	Ganzzahlprogrammierung	9
1.4	Quadratische Programme	11
1.5	Kombinatorische Optimierung	14
1.6	Spieltheorie	16
2	Minima, Ableitungen, Analysis	22
2.1	Ableitungen	22
2.2	Minima, Maxima, Extrema	29
2.3	Zweite Ableitungen und Parabeln	33
2.4	Paraboloide und warum alles doch nicht so einfach ist	37
2.5	Randextrema	44
3	Abstiegsverfahren	48
3.1	Grundidee aller Verfahren	48
3.2	Abstiegsrichtungen	50
3.3	Schrittweitensteuerung	58
3.4	Trust Regions	60
3.5	Zusammenfassung	61
4	Zurück zur linearen Optimierung	63
4.1	Der Simplexalgorithmus	63
4.2	Die Zweiphasenmethode	69
5	Modellierung von Optimierungsproblemen	81
5.1	Das Diät–Problem	81
5.2	Transportprobleme	84
5.3	Zuordnungsprobleme	86
5.4	Flußin Netzwerken	89
5.5	Spieltheorie und die zwei Phasen	92

Don't panic!

D. Adams, *Hitchhiker's guide to the galaxy*

Optimierung ist überall

1

Eigentlich ist es klar, daß Optimierung eine der häufigsten Formen der angewandten Mathematik ist, ja sein muß, denn normalerweise versucht man ja, im Geschäftsleben wie auch im privaten Bereich oder auch bei Auseinandersetzungen ein *optimales* Resultat zu erzielen – wobei natürlich oftmals gewisse Randbedingungen einzuhalten sind. Und ob man nun einen Nutzen zu maximieren versucht oder einen Schaden so gering wie möglich halten will, das ist ohnehin kein Unterschied, mathematisch gesehen sogar nichts anderes als ein Vorzeichen.

1.1 Die allgemeine mathematische Formulierung

Um ein Optimierungsproblem mathematisch modellieren zu können, brauchen wir immer zwei Dinge:

1. Eine *zulässige* Teilmenge der *Parametermenge* X . Diese Menge X legt ganz allgemein fest, wie wir den Gang der Dinge beeinflussen können. Will man beispielsweise den Ertrag eines Aktienfonds optimieren, dann würde D angeben, wieviele Aktien von jeder individuellen Firma gehalten werden; läuft das ganze dynamisch, dann hätte man sogar noch eine Dimension, nämlich die Zeit.

Nun ist aber nicht unbedingt die ganze Menge X zugelassen, sondern möglicherweise nur eine Teilmenge $D \subseteq X$, die durch möglicherweise durch mehr oder weniger komplexe *Nebenbedingungen* bestimmt wird. Um zu unserem Aktienbeispiel zurückzukehren: Neben der offensichtlichen Einschränkung durch die Endlichkeit des eigenen Kapitals¹ müsste man bei einer realistischen Modellierung z.B. auch unverkäufliche Aktien oder kartellrechtliche Beschränkungen mit einbeziehen.

2. Eine *Zielfunktion* $F : X \rightarrow \mathbb{R}$, die angibt, welchen Nutzen wir aus einer bestimmten Parameterkonfiguration ziehen oder welchen Schaden wir in diesem Fall erleiden.

Das Ziel der Optimierung besteht nun “ganz einfach” darin, ein *optimales* $x^* \in D$ zu finden, so daß

$$F(x^*) = \min_{x \in D} F(x) \quad \text{oder} \quad F(x^*) = \max_{x \in D} F(x) \quad (1.1)$$

¹Ein Aspekt, an den fast jede(r) beinahe täglich erinnert wird.

ist. Die Wahl von “min” oder “max” hängt davon ab, ob die Zielfunktion nun Kosten oder Gewinn beschreibt.

Bemerkung 1.1 Es würde eigentlich genügen, sich in (1.1) entweder auf min oder auf max zu beschränken, was wir für die Herleitung unserer Verfahren später auch tun werden. Da nämlich die Multiplikation einer Ungleichung mit -1 das Ungleichungszeichen umdreht², gilt

$$\begin{aligned} F(x^*) = \min_{x \in D} F(x) &\Leftrightarrow F(x^*) \leq F(x) \Leftrightarrow -F(x^*) \geq -F(x) \\ &\Leftrightarrow (-F)(x^*) = \max_{x \in D} (-F)(x) \end{aligned}$$

und damit ist das Minimum von F gerade das Maximum von $-F$.

Sind wir also ruhig mal optimistisch und nehmen an, wir wollen *maximieren*, dann lautet das Optimierungsproblem

Man finde $x^* \in D$, so daß

$$F(x^*) \geq F(x) \quad \text{für alle } x \in D.$$

Was einem zur Lösung dieses Problems sofort einfällt, ist die *Hau-Ruck-Methode*, nämlich hinreichend häufiges Ausprobieren.

Algorithmus 1.2 Man wähle Punkte $x_1, \dots, x_n \in D$ geeignet³ und wählt x^* so, daß

$$F(x^*) = \max_{j=1, \dots, n} F(x_j).$$

Der Vorteil der Methode ist, daß sie eigentlich für alle beliebigen Zielfunktionen F funktioniert, was die Nachteile angeht, ist sie hingegen etwas großzügiger:

- es gibt eigentlich keine Garantie gibt, daß man auf diese Art und Weise einem Maximum auch nur nahekommt – man denke nur an die Funktion $F(x) = \sin x$ und die⁴ Wahl von sogar *unendlich vielen* Punkte $x_j = \frac{2j+1}{2}\pi$, wo $\sin x_j = -1$ ist, was so weit vom Maximum $+1$ entfernt ist, wie man’s sich nur vorstellen kann.
- Tatsächlich besteht eine der “besten” Strategien sogar darin, diese Punkte zuerst einmal *zufällig* zu wählen – man spricht dann von einer *Monte-Carlo-Methode*. Der Grund ist klar: bei zufälliger Punktwahl ist es sehr unwahrscheinlich, daß man die Punkte so ungeschickt ausfallen wie im Sinus-Beispiel.
- Wenn D unbeschränkt ist, z.B. $D = \mathbb{R}$, dann ist die Methode ohnehin etwas unpraktikabel, denn man kann in endlicher Zeit ja nur endlich viele Punkte bestimmen – ganz zu schweigen von dem endlichen Speicherplatz, den realistische Computer nur zur Verfügung stellen können.

²Wie man eigentlich in der Schule gelernt haben sollte.

³Normalerweise erst einmal so, daß sie hinreichend dicht liegen.

⁴Zugegebenermaßen ganz besonders ungeschickte.

Für eine mathematische Behandlung von Optimierungsproblemen klassifiziert man die Probleme daher nach gewissen Kriterien, die dann auch völlig unterschiedliche Methoden bedingen und zulassen. Ein paar davon sind

- Klassifikation nach dem Typ der Zielfunktion. Ist beispielsweise $x = (x_1, \dots, x_n) \in \mathbb{R}^n$ ein Vektor, dann heißt die Zielfunktion *linear*, wenn

$$F(x) = c^T x = c_1 x_1 + \dots + c_n x_n,$$

oder *quadratisch*, wenn⁵

$$F(x) = x^T A x + c^T x = \sum_{j,k=1}^n a_{jk} x_j x_k + \sum_{j=1}^n c_j x_j.$$

Ansonsten spricht man generell von *nichtlinearen* Optimierungsproblemen.

- Klassifizierung nach dem Parameterraum X . Kann man ein $x \in X$ beliebig gut durch $y \neq x$ annähern, dann spricht man von einem *kontinuierlichen* Optimierungsproblem, andernfalls von einem *diskreten* Optimierungsproblem. Typische Beispiele sind $X = \mathbb{R}$ und $X = \mathbb{Z}$, also *reelle Zahlen*⁶ und *ganze Zahlen*, also $\dots, -1, 0, 1, \dots$. Bei kontinuierlichen Problemen kann man Methoden der Analysis verwenden, die sich bei diskreten Problemen aber zumeist verbieten. Übrigens gibt es auch “Mischprobleme”! Die Zielfunktion kann durchaus von einigen Parametern kontinuierlich, von anderen hingegen diskret abhängen – beispielsweise bei der Optimierung des Musikgenusses mit einer Stereoanlage. Der Equalizer hat kontinuierliche Schieberegler, die Auswahl der CD hingegen ist eine diskrete Größe, die nur endlich viele Werte annehmen kann.
- Klassifizierung nach der zulässigen Menge D . Diese kann *beschränkt* sein oder nicht; auch hier ist es wieder möglich, daß in einigen Parametern Beschränkungen existieren oder nicht.

Das erscheint alles ziemlich abstrakt und nicht besonders realitätsbezogen. Daher sehen wir uns einige typische Sorten von Optimierungsproblemen einmal an halbwegs realistischen Beispielen an – bei der Gelegenheit erhalten wir dann auch Information über die Lösungsmethoden, die uns dabei zur Verfügung stehen.

1.2 Lineare Optimierungsprobleme

Bei einem *linearen* Optimierungsproblem betrachtet man, wie ja schon gesagt, eine Zielfunktion der Form

$$F(x) = c^T x = \sum_{j=1}^n c_j x_j, \quad c, x \in \mathbb{R}^n. \quad (1.2)$$

⁵Und hier brauchen wir leider (?) mathematische Notation.

⁶Was das wirklich ist, ist gar nicht so einfach zu beschreiben!

Im allereinfachsten Fall, $n = 1$, ist also $F(x) = cx$ und diese Funktion ist *unbeschränkt* bezüglich x . Nehmen wir an, daß $c > 0$ ist, dann wächst diese Funktion über alle Grenzen, wenn x nur größer und größer wird. Um einen Maximalwert zu bekommen, muß also der zulässige Bereich D *beschränkt* sein. Praktischerweise verwendet man dazu ein *Intervall* $I = [a, b]$, das man auch durch die beiden Nebenbedingungen

$$x \geq a \quad \text{und} \quad x \leq b$$

ausdrücken kann. Diese Nebenbedingungen sind übrigens ebenfalls *lineare* Funktionen in x und können auch etwas “mathematischer” als

$$\begin{bmatrix} a \\ -b \end{bmatrix} \leq \begin{bmatrix} x \\ -x \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} x$$

geschrieben werden, also in der Form $Ax \geq b$, wobei “ \geq ” hier *komponentenweise* zu verstehen⁷ ist. Jetzt lösen wir aber unser “Problem”, indem wir einen Blick auf Abb 1.2 werfen und Ma-

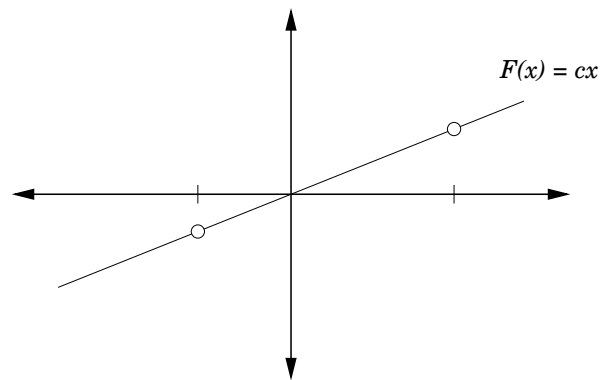


Abbildung 1.1: Eine lineare Funktion nimmt auf einem Intervall Maximum und Minimum immer am Rand an.

ximum wie Minimum der Funktion finden, indem wir einfach an einem der beiden Endpunkte nachsehen.

Übung 1.1 Zeigen Sie, daß Nebenbedingungen der Form $x \geq a_1, \dots, x \geq a_m$ und $x \leq b_1, \dots, x \leq b_n$ immer zu einem Intervall als zulässiger Menge führen. \diamond

Trotzdem gibt uns – wie immer – unser einfaches Beispiel bereits eine erste Idee, wie so ein allgemeines lineares Optimierungsproblem oder *lineares Programm* aussehen wird, nämlich

Minimiere $c^T x$ unter der Nebenbedingung $Ax \geq b$, wobei A eine $m \times n$ -Matrix ist und $b \in \mathbb{R}^m$ sowie $c, x \in \mathbb{R}^n$ liegen.

⁷Dies führt zu einer sogenannten *Halbordnung*, bei der von zwei verschiedenen Elementen nicht unbedingt eines das kleinere sein muss.

Jetzt aber ein “realistisches” Beispiel für ein lineares Programm, das sogar aus einem “richtigen” Mathematikbuch [13] stammt

Beispiel 1.3 (*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 Gewinn⁸, wenn die folgenden Parameter vorliegen:

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

Jetzt wollen wir dieses Problem aber einmal mathematisch formulieren, oder, wie man heute so schön sagt “modellieren”. Dazu sei x die Anzahl der produzierten Damenschuhe, y die Anzahl der produzierten Herrenschuhe. Dann ergibt sich der erzielte Gewinn als

$$F(x, y) = 16x + 32y = [16 \quad 32] \begin{bmatrix} x \\ y \end{bmatrix} \quad \Rightarrow \quad c = \begin{bmatrix} 16 \\ 32 \end{bmatrix}$$

und die Nebenbedingungen sind

$$20x + 10y \leq 8000, \quad 4x + 5y \leq 2000, \quad 6x + 15y \leq 4500. \quad (1.3)$$

Aber das ist noch nicht alles! Schließen wir nämlich negative Produktion⁹ aus, dann erhalten wir die zusätzlichen Forderungen, daß $x \geq 0$ und $y \geq 0$ sein müssen, was uns die Nebenbedingungsmatrix¹⁰

$$\underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -20 & -10 \\ -4 & -5 \\ -6 & -15 \end{bmatrix}}_{=:A} \begin{bmatrix} x \\ y \end{bmatrix} \geq \underbrace{\begin{bmatrix} 0 \\ 0 \\ -8000 \\ -2000 \\ -4500 \end{bmatrix}}_{=:b}$$

liefert. Wie löst man sowas? Nun, in zwei Variablen ist das sogar noch relativ einfach und das sehen wir uns jetzt mal an. Dazu bemerken wir zuerst, daß die Gleichung $ax + by = c$ einer Geraden in der Ebene entspricht. Auf einer Seite der Geraden¹¹ liegen die Punkte mit $\geq c$, auf der anderen die mit $\leq c$, die Ungleichungsbengungen liefern somit sogenannte *Halbräume*, siehe Abb. 1.2. Schneidet man diese nun – und genau das passiert ja, wenn mehrere Nebenbedingungen *gleichzeitig* zu erfüllen sind – dann erhält man ein sogenanntes *konvexes Polyeder*. Das ist in Abb. 1.3 zu sehen. Nun können wir das Problem nämlich sehr einfach *graphisch* lösen,

⁸Unter der (realistischen ?) Annahme, daß alle Schuhe verkauft werden können.

⁹Das wäre dann Ankauf von Schuhen von anderen Herstellern unter Freigabe von Produktionskapazitäten – ersteres wäre möglich, letzteres aber leider nicht!

¹⁰Um überall “ \geq ” oder “ \leq ” zu haben, müssen wir einen Teil der Ungleichungen mit -1 multiplizieren, wobei wir uns für (1.3) entscheiden wollen, die Wahl ist aber vollkommen willkürlich!

¹¹Auf welcher genau, das hängt von der Orientierung der Geraden ab, also von den Vorzeichen von a und b .

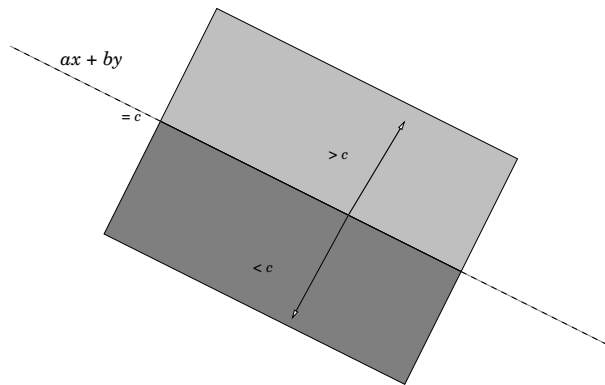


Abbildung 1.2: Die beiden Halbräume, die durch eine Gerade festgelegt werden. Entlang der Pfeile wird die Ungleichung immer “stärker” erfüllt.

indem wir uns erinnern, daß die Zielfunktion ja ebenfalls linear ist, das heißt, ihre *Niveaulinien*, auf denen die Zielfunktion konstant ist, sind Geraden. Diese Geraden verschieben wir nun immer so, daß sie einen höheren Wert annehmen. Solange es gemeinsame Punkte des zulässigen Bereichs und dieser Geraden gibt, sind das zulässige Lösungen des Optimierungsproblems mit immer besseren Werten, siehe Abb. 1.4.

Wie lange kann man das machen? Ganz einfach: bis die Zielfunktionsgerade bei jeder noch so kleinen Bewegung aus dem Polyeder herausgeschoben würde, also bis man in einer Ecke des Polyeders gelandet ist. Diese Ecke entspricht dann einer eindeutigen Lösung des Optimierungsproblems und der Wert der Zielfunktion dort ist maximal.

Übung 1.2 Kann es passieren, daß dieser Prozess *nicht* in einer Ecke landet? Wenn ja, was bedeutet das geometrisch und wie ist das für unser Optimierungsproblem zu interpretieren? \diamond

So einfach kann man tatsächlich alle linearen Optimierungsprobleme mit zwei Variablen graphisch lösen. Was aber, wenn wir es mit mehr als zwei Variablen zu tun haben, denn dann sind wir ziemlich schnell am Ende unserer Anschauung. Nun, dafür gibt es ein formales Verfahren, das wir im Verlauf dieser Vorlesung noch kennenlernen werden, nämlich den *Simplexalgorithmus*, der sich sozusagen von Ecke zu Ecke vorarbeitet, bis er das Optimalergebnis erreicht hat.

1.3 Ganzzahlprogrammierung

So schön und ansprechend die Sache mit der graphischen Lösung von linearen Optimierungsproblemen auch ist und so schön sich diese Idee auch in einen formalen Algorithmus packen lässt – es gibt ein Problem und zwar ein ziemlich fundamentales: Diese Methoden sind *kontinuierlich*! Wenn wir eine Gerade herumschieben, dann müssen wir beliebig kleine Schrittweiten zur Verfügung haben, sonst klappt das Ganze nicht so richtig. Außerdem gibt es keine Garantie, daß die Ecke, in der wir am Ende landen werden, überhaupt zu einem ganzzahligen Wert

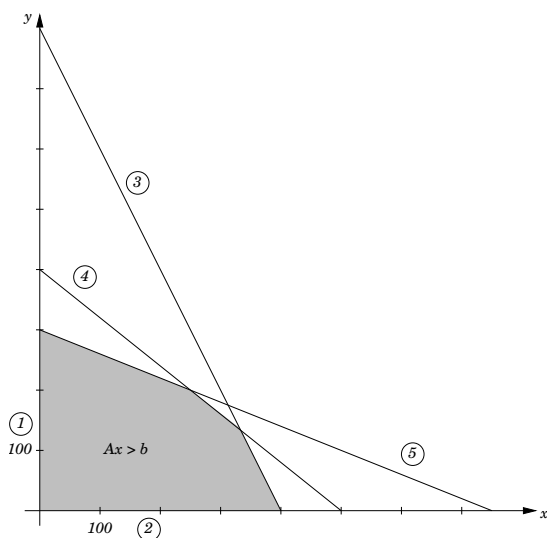


Abbildung 1.3: Der zulässige Bereich für Beispiel 1.3. Die Nummern der jeweiligen Nebenbedingungen entsprechen den Zeilen in der Matrix.

gehört, was sie im “Normalfall” auch nicht tun wird¹². Wenn man beispielsweise nur den Gewinn für Damenschuhe etwas höher ansetzt, so daß die Niveaulinien in Abb. 1.4 nicht so “flach” verlaufen, dann landet man möglicherweise dort und die Optimallösung bedeutet, daß die Firma Bruchteile von Schuhen herstellen¹³ und verkaufen¹⁴ muss. Doch nun ein Beispiel, wo es definitiv zu Problemen kommt.

Beispiel 1.4 (*Ganzzahliges Transportproblem*¹⁵) 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.

Dieses Beispiel hat die *kontinuierliche* Optimallösung $x = 5\frac{1}{2}$ und $y = 3$, die *ganzzahlige* oder *diskrete* Optimallösung, also

$$(x, y) \in \mathbb{Z}^2 \quad \text{so daß} \quad F(x, y) \geq F(x', y'), \quad (x', y') \in D \cap \mathbb{Z}^2$$

ist hingegen $x = y = 4$, siehe Abb. 1.5. Die mathematischen Methoden zur Lösung der Ganz-

¹²Insofern war Beispiel 1.3 auch besonders “geschickt” gewählt.

¹³Was noch gehen könnte

¹⁴Hier wird’s langsam schwierig!

¹⁵Aus [3, S. 359–360].

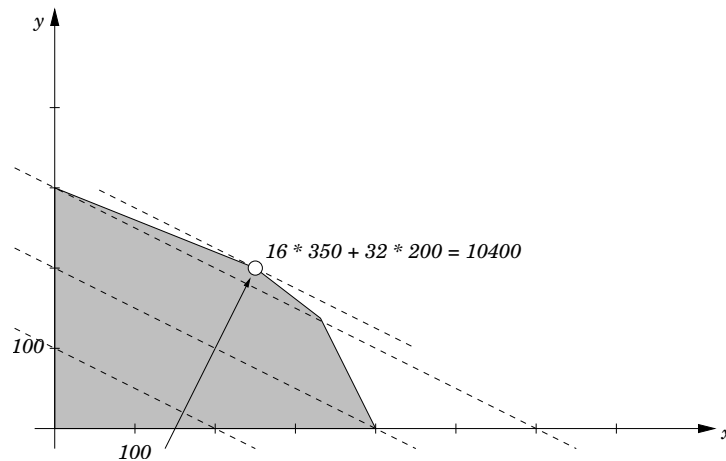


Abbildung 1.4: Verschieben der Zielfunktion bis der zulässige Bereich verlassen wird.

zahlprogrammierung (“*Integer programming*”) überschreiten übrigens bei weitem den Umfang dieser Vorlesung – da werden ziemlich niveauvolle Methoden aus der Computeralgebra eingesetzt, siehe [3], sogar so niveauvoll, daß es über den Umfang einer mathematischen Grundvorlesung hinausgeht.

1.4 Quadratische Programme

Die nächste Klasse von Optimierungsproblemen befaßt sich mit *quadratischen* Zielfunktionen, also mit Zielfunktionen der Form

$$F(x) = ax^2 + bx, \quad x \in \mathbb{R},$$

bzw.¹⁶

$$F(x) = \sum_{j,k=1}^n a_{jk} x_j x_k + \sum_{j=1}^n b_j x_j, \quad x \in \mathbb{R}^n.$$

Solche Probleme tauchen relativ natürlich bei der Regressionsanalyse auf.

Beispiel 1.5 (Konstante Regression) Gegeben sind Punkte Meßwerte y_1, \dots, y_n , gesucht ist der “Mittelwert” der Meßwerte, der im quadratischen Mittel am wenigsten von den Meßwerten abweicht¹⁷, also derjenige Wert y , so daß

$$F(y) = (y - y_1)^2 + \dots + (y - y_n)^2 = \min$$

wird.

¹⁶Auch die Polynome $x_j x_k$ bezeichnet man als (*homogene*) quadratische Polynome!

¹⁷In statistischer Sprechweise wollen wir also die Varianz minimieren.

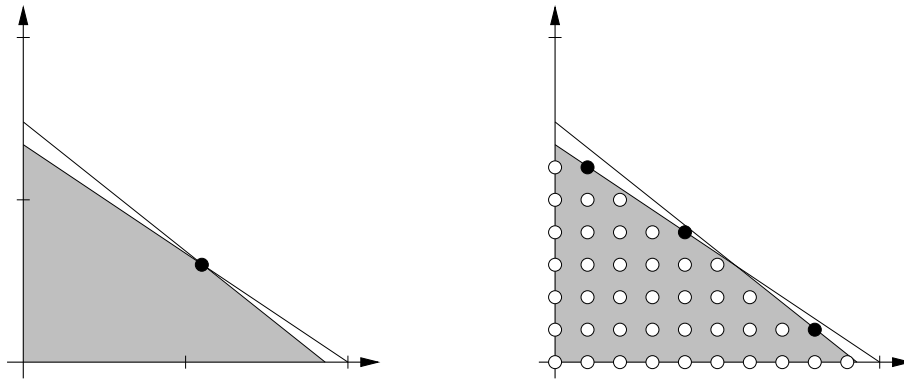


Abbildung 1.5: Links der kontinuierliche zulässige Bereich für Beispiel 1.4 mit der (kontinuierlichen) Optimallösung $[5\frac{1}{2}, 3]^T$, rechts die ganzzahligen Punkte im zulässigen Bereich, die schwarz markierten liegen gerade so auf dem Rand.

Die Schreibweise “Mittelwert” in Beispiel 1.5 hätten wir uns eigentlich sparen können, denn die Minimallösung y ist tatsächlich der Mittelwert

$$y^* = \frac{1}{n} (y_1 + \dots + y_n).$$

Aber wie kommt man drauf? In der Tat wird uns die Herleitung dieses Ergebnisses auch sagen, wie wir das Problem allgemein angehen können. Dazu berechnen wir

$$\begin{aligned} F(y) &= (y^2 - 2y y_1 + y_1^2) + \dots + (y^2 - 2y y_n + y_n^2) \\ &= n y^2 - 2y (y_1 + \dots + y_n) + (y_1^2 + \dots + y_n^2) \end{aligned}$$

Damit ist $F(y)$ eine Funktion der Form $ay^2 + by + c$ mit $a > 0$, also eine *nach oben geöffnete Parabel* wie in Abb. 1.6 ist. So eine Parabel hat nun *genau ein* Minimum und das kriegen wir wie in der Schule, indem wir die Ableitung gleich Null setzen, also

$$0 = F'(y) = 2ny - 2(y_1 + \dots + y_n) \quad \Longleftrightarrow \quad y = \frac{1}{n} (y_1 + \dots + y_n)$$

erhalten.

Na gut, dieses Problem war ja auch ziemlich einfach, um nicht zu sagen zu einfach, also suchen wir uns was interessanteres.

Beispiel 1.6 (Lineare Regression) Zu Wertepaaren $(x_1, y_1), \dots, (x_n, y_n)$ sucht man eine (Regressions-)Gerade der Form $\ell(x) = ax + b$, so daß die Standardabweichung

$$F(a, b) = (\ell(x_1) - y_1)^2 + \dots + (\ell(x_n) - y_n)^2 \quad (1.4)$$

minimiert wird. Typische Anwendungen dieser Art bestehen in der Annäherung von Meßdaten, die “eigentlich” auf einer Geraden liegen sollten, dies aber aufgrund von Störungen nicht so richtig tun, siehe Abb. 1.7.

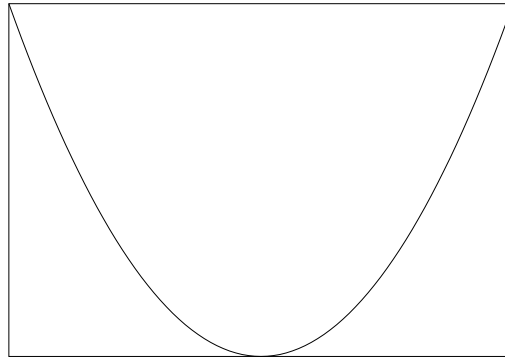


Abbildung 1.6: Eine Parabel – was für eine Überraschung!

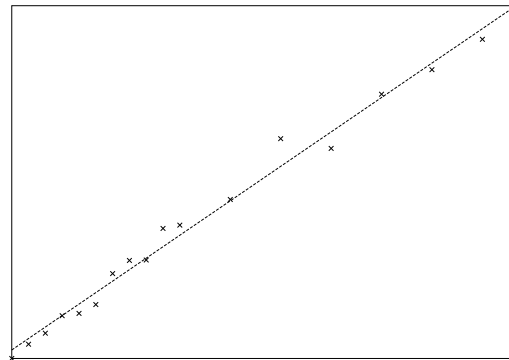
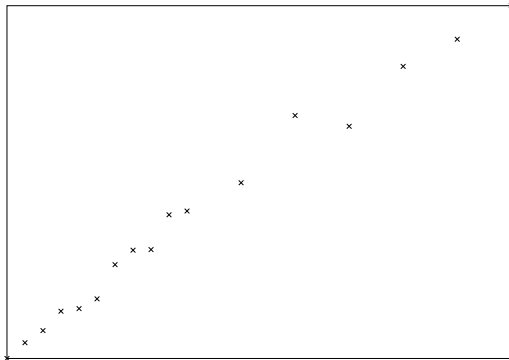


Abbildung 1.7: Zufällig (um maximal 16%) gestörte Daten auf einer Geraden (links) und die Lösungsgerade des Minimierungsproblems (rechts).

Schauen wir mal, ob und wie wir Werte a, b finden können, die den Ausdruck (1.4) minimieren. Dazu sehen wir erst einmal, daß die individuellen Ausdrücke der Summe als

$$\begin{aligned} (\ell(x_j) - y_j)^2 &= \ell(x_j)^2 - 2\ell(x_j)y_j + y_j^2 = (ax_j + b)^2 - 2(ax_j + b)y_j + y_j^2 \\ &= a^2x_j^2 + 2abx_j + b^2 - 2ax_jy_j - 2by_j + y_j^2. \end{aligned}$$

Aber Achtung: Die Variablen hier sind a und b ! Die Funktion, die hier in Abhängigkeit von a und b minimiert werden soll, ist jetzt ein sogenanntes *Paraboloid*, siehe Abb. 1.8. Genauso wie die Parabel hat dieses Gebilde nur ein eindeutiges Minimum, das man wieder durch “Ableiten” finden kann¹⁸, was aber ein bißchen mehr Theorie voraussetzt. Auf alle Fälle führt dies zu einem *linearen Gleichungssystem*. Wie man diese erhält und warum man sie mit etwas Vorsicht behandeln und mit der richtigen Methode lösen muß, das ist in [10, Kapitel 5] beschrieben.

¹⁸Wie das geht, das werden wir noch lernen – hoffe ich zumindest.

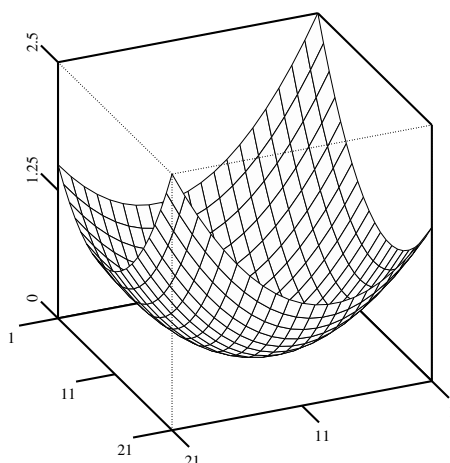


Abbildung 1.8: Ein einfaches, nach oben geöffnetes Paraboloid, das man erhält, wenn man eine Parabel wie in Abb. 1.6 um ihr Minimum rotieren lässt. Auch hier hat man wieder genau ein Minimum.

Trotzdem werden wir sehen, daß quadratische Probleme den Prototyp eines Optimierungsproblems darstellen, zumindest lokal. Die Idee dabei ist, daß eine Funktion sich *lokal* durch ihren Funktionswert, die Steigung und die Krümmung beschreiben läßt und daß dabei die Krümmung dem “Parabelanteil” entspricht. Und die ist in einer Variablen entweder

- nach oben geöffnet, und das liefert ein *Minimum*, oder
- nach unten geöffnet, was zu einem *Maximum* führt.

1.5 Kombinatorische Optimierung

Kombinatorische Optimierungsprobleme leben normalerweise nicht in der Welt der Analysis, sondern auf den “Standardobjekten” der diskreten Mathematik wie Graphen und Bäumen. Die Bestimmung optimaler Verfahren zur Lösung dieser Probleme hat dann oftmals einen Bezug zu Standardverfahren aus der Informatik, mit denen diese Datenstrukturen behandelt werden.

Beispiel 1.7 (Routenplanung) Eine Straßenkarte liegt vor als ein Liste von Wegpunkten¹⁹ und eine Liste von Straßen, die zwei Wegpunkte, beispielsweise X und Y verbinden und deren Länge $D(X, Y)$ angegeben ist.

Aufgabe: Für zwei Wegpunkte A und B finde man den kürzesten Weg von A nach B .

Das mathematische Modell für dieses Problem ist ein *Graph*²⁰, dessen *Ecken* die Wegpunkte und dessen *Kanten* die Verbindungsstraßen sind. *Gewichtet* sind die Kanten dann mit der

¹⁹Das können Städte, aber auch markante Straßenkreuzungen oder Autobahnabfahrten sein

²⁰Weswegen eine Vorlesung über diskrete Strukturen, die sich mit derartigen Objekten und Standardmethoden zu ihrer Behandlung auseinandersetzt, auch jede Menge Sinn im Rahmen eines Aufbaustudiums

Entfernung²¹ von einem Wegpunkt zum anderen. Eine Methode, die kürzeste Entfernung von A nach B zu finden, ist dann das folgende Verfahren.

Algorithmus 1.8 (Kürzeste Entfernung) *Generiert Liste \mathcal{L} von Wegpunkten und Zahl $d \geq 0$.*

1. Initialisierung: Setze $\mathcal{L} = \{A\}$ und $d = 0$.

2. Nächster Wegpunkt:

(a) Für alle Wegpunkte X in der Liste \mathcal{L} bestimme den nächsten Wegpunkt Y_X , der durch eine Straße mit X verbunden ist und noch nicht in \mathcal{L} liegt.

(b) Unter all diesen Y_X (mit "Parameter" X) wähle denjenigen, für den

$$D(A, X) + D(X, Y_X)$$

am kleinsten ist.

(c) Füge diesen Wegpunkt Y_X zur Liste hinzu und setze

$$d = D(A, X) + D(X, Y_X).$$

3. Wiederhole 2 so lange, bis $B \in \mathcal{L}$ ist.

Der Trick bei diesem Verfahren ist einfach: Man kann sich überlegen, daß zu jedem Zeitpunkt die Liste \mathcal{L} alle Wegpunkte enthält, deren Entfernung von A höchstens d ist – immer vorausgesetzt, man verwendet den kürzesten Weg. Will man nun auch noch wissen, wie man fahren muß, als nicht nur die Entfernung sondern auch den Weg kennen, dann muß man den Algorithmus etwas abändern und einen *Baum* generieren, in dem auch noch festgehalten wird, wie man zu den jeweiligen Einträgen in der Liste kommt.

Aber einen haben wir noch . . .

Beispiel 1.9 (Fahrender Händler) *Ein fahrender Händler hat in seinem Lager verschiedene Objekte verschiedensten Gewichts, beispielsweise Kämmе, Messer, Knöpfe oder Rasierklіngen²². Sein Transportkarren trägt aber nur ein bestimmtes Gewicht. Wie ist der Karren zu beladen, damit der Erlös optimal wird.*

Beispiel 1.9 ist ein typisches Beispiel für ein sogenanntes *Rucksackproblem*, englisch *knapsack problem*, bei denen eine *endliche* Zahl von Objekten²³, von denen man einen "Kosten-" und einen "Nutzenwert" kennt, so in einen "Rucksack" gepackt werden soll, daß die Kosten einen bestimmten Maximalwert nicht überschreiten und der Nutzen maximiert wird. Auch wenn es zur Lösung derartiger Probleme Methoden gibt, siehe z.B. [14], gehört es doch wie auch der berühmte "Travelling Salesman" zur Klasse der sogenannten *NP-vollständigen* Probleme, die mit wachsender Parameterzahl ziemlich schnell praktisch unlösbar werden.

²¹Oder mit der Fahrzeit, wenn man noch die Höchstgeschwindigkeit einstellen kann, die man auf Autobahnen und Landstraßen fahren möchte. Man könnte auch Stauwahrscheinlichkeit und dergleichen einbeziehen.

²²Die Auswahl ist zu 100% willkürlich.

²³Davon können einige identisch sein, müssen aber nicht.

1.6 Spieltheorie

Bisher haben wir eigentlich nur Optimierungsprobleme betrachtet, bei denen das Ergebnis *direkt* von der Parameterwahl abhängt. Es gibt aber auch Probleme, bei denen “auf der anderen Seite” jemand sitzt, der ebenfalls seinen Nutzen optimieren möchte und der zumeist andere Ziele hat als man selbst. Die mathematische Behandlung solcher Konflikte ist die *Spieltheorie*, die von John von Neumann²⁴ und dem Ökonom Oskar Morgenstern begründet wurde.

Machen wir es uns einfach und betrachten wir Spiele für *zwei* Personen, die noch am einfachsten darzustellen sind. Jeder der beiden Personen, nennen wir sie beispielsweise **Aragorn** und **Boromir**, stehen bei dem Spiel Strategien A_1, \dots, A_m bzw. B_1, \dots, B_n zur Verfügung. Bei einem Spiel mit “mehreren Zügen” entsprechen diese Strategien Entscheidungen der Form

... wenn in den vorherigen Zügen dies und jenes passiert ist, dann führe in diesem Zug diese Aktion aus ...

für eine Schachpartie wären also m und n astronomische Zahlen. Das Ergebnis der Strategien kann man dann in einer *Auszahlungsmatrix*, also eigentlich einer Tabelle, darstellen, in der angegeben ist, was der Gewinn der jeweilige Person im Falle einer bestimmten Strategie sein wird, also

	B_1	\dots	B_n
A_1	(x_{11}, y_{11})	\dots	(x_{1n}, y_{1n})
\vdots	\vdots	\ddots	\vdots
A_m	(x_{m1}, y_{m1})	\dots	(x_{mn}, y_{mn})

Diese Auszahlungsmatrix ist zwar *theoretisch* ein sehr nettes Konzept, mit dem sich im Prinzip alles erledigen läßt, in der Praxis ist es aber ein äußerst schwieriges Problem, diese Auszahlungswerte realistisch festzulegen²⁵

Das Spiel heißt *Nullsummenspiel*, wenn das was einer gewinnt vom anderen bezahlt werden muß, wenn also $x_{jk} = -y_{jk}$ ist. Nullsummenspiele sind immer “Konfliktspiele”, Nichtnullsummenspiele können hingegen zu kooperativen Strategien führen.

Beispiel 1.10 *Aragorn und Boromir treffen auf einen Ork. Wenn sie ihn beide angreifen, werden sie ohne weiteres mit ihm fertig und gewinnen (Gewinn 1), wenn einer allein den Ork angreift, dann wird er zwar gewinnen, braucht aber mehr Zeit dafür (Gewinn 0.5) und wer nichts tut, der gewinnt oder verliert nicht. Die Auszahlungsmatrix hat also die Form*

	<i>angreifen</i>	<i>zuschauen</i>
<i>angreifen</i>	(1, 1)	(0.5, 0)
<i>zuschauen</i>	(0, 0.5)	(0, 0)

und die optimale Strategie ist somit das kooperative “Auf ihn mit Gebrüll”.

²⁴Ungarischstämmiger Mathematiker, eine der herausragenden Gestalten des 20. Jahrhunderts.

²⁵So liegt auch heute die Qualität eines Schachprogramms weniger in der Fähigkeit, möglichst viele Züge im Voraus zu berechnen, sondern in immer ausgeklügelteren Methoden, Spielsituationen zu bewerten.

Abgesehen davon, daß Nullsummenspiele schon wegen der “Konfliktsituation”, die in ihnen enthalten ist, “interessanter” sind, sind sie auch einfacher darzustellen: Da das was der eine gewinnt auch das ist, was der andere verliert, brauchen wir nur einen Wert in die Auszahlungsmatrix zu schreiben, keine Paare.

Wie findet man aber jetzt eine optimale Strategie? Nun, jeder Spieler wird davon ausgehen, daß sein Gegner so viel wie möglich gewinnen oder zumindest so wenig wie möglich verlieren möchte. Deswegen wird Aragorn bei jeder seiner möglichen Strategien nachsehen, was herauskommt, wenn Boromir die beste Gegenstrategie wählt, das heißt, er wird in jeder *Zeile* der Auszahlungsmatrix nach dem kleinsten Wert suchen und sich diesen merken:

	B_1	\cdots	B_n	
A_1	x_{11}	\cdots	x_{1n}	$\min_k x_{1k}$
\vdots	\vdots	\ddots	\vdots	\vdots
A_m	x_{m1}	\cdots	x_{mn}	$\min_k x_{mk}$

Und jetzt schlägt Aragorn zurück: Er wählt seine Strategie so, daß dieses Minimum rechts neben der Tabelle maximiert wird, denn dann kann Boromir machen, was er will, das Ergebnis für Aragorn wird höchstens besser. Der Auszahlungswert ist dann

$$\max_{j=1,\dots,m} \min_{k=1,\dots,n} x_{jk}. \quad (1.5)$$

Jetzt betrachten wir das Spiel mal von der anderen Seite, also aus der Perspektive von Boromir: Er wird annehmen, daß Aragorn seine Strategie so wählt, daß so viel wie möglich herauskommt und deswegen in allen *Spalten* nach dem *Maximum* suchen,

	B_1	\cdots	B_n
A_1	x_{11}	\cdots	x_{1n}
\vdots	\vdots	\ddots	\vdots
A_m	x_{m1}	\cdots	x_{mn}
	$\max_j x_{j1}$	\cdots	$\max_j x_{jn}$

und nun die Spalte so wählen, daß dieser Wert *minimiert* wird, was zum Auszahlungswert

$$\min_{k=1,\dots,n} \max_{j=1,\dots,m} x_{jk} \quad (1.6)$$

Stimmen nun (1.5) und (1.6) überein, dann spricht man von einem *Sattelpunkt* und das Spiel ist *determiniert*: Weicht nämlich einer der beiden Spieler von seiner optimalen Strategie ab während der andere diese beibehält, dann wird sich sein Ergebnis nur verschlechtern.

Beispiel 1.11 (Sattelpunkt) *Aragorn und ein Ork stehen sich gegenüber, beide haben die Option anzugreifen oder zu fliehen. Da Aragorn stärker ist, wird er gewinnen, wenn beide angreifen, würde er fliehen, könnte der Ork ihn in den Rücken treffen. Flieht der Ork, hat er zumindest eine*

Chance, Aragorn zu entkommen, fliehen beide, dann leidet das Image des Helden darunter. Die Auszahlungsmatrix (aus der Sicht von Aragorn) ist daher

	angreifen	fliehen	
angreifen	5	1*	1*
fliehen	-2	-1	-2
	5	1*	

wobei die “optimalen” Werte mit “*” gekennzeichnet sind. Das Spiel hat offensichtlich einen Sattelpunkt: Aragorn greift an und der Ork ergreift die Flucht.

Was wir bisher gesehen haben waren Beispiele für eine *reine Strategie*, bei der die Spieler immer dasselbe tun und bei der das Ergebnis des Spiels eigentlich von vornherein feststeht. Solche reinen Strategien sind bei Sattelpunkten auch tatsächlich angebracht und optimal. Aber: Haben denn alle Spiele einen Sattelpunkt? Die Antwort sehen wir im nächsten Beispiel.

Beispiel 1.12 (Stein, Schere, Papier) Aragorn und Boromir nutzen eine Rast in Moria, um das bekannte Kinderspiel “Stein, Schere, Papier” zu spielen, bei dem die Spieler gleichzeitig mit ihrer Hand entweder einen Stein (Faust), eine Schere (Zeige- und Mittelfinger abgespreizt) oder ein Blatt Papier (flache Hand) darstellen. Da der Stein die Schere stumpf macht, die Schere Papier schneidet und Papier den Stein einwickeln kann, erhält man die Auszahlungsmatrix

	Stein	Schere	Papier	
Stein	0	1	-1	-1
Schere	-1	0	1	-1
Papier	1	-1	0	-1
	1	1	1	

und es ist weit und breit kein Sattelpunkt in Sicht.

Was macht man nun bei sattelpunktfreien Spielen? Dann packt halt Aragorn die Würfel aus und wendet sich sogenannten *gemischten Strategien* zu, bei denen er einfach seine Strategien mit bestimmten Wahrscheinlichkeiten auswählt²⁶, ein Zufallsexperiment durchführt (“Würfeln”) und je nach Ergebnis dieses Zufallsexperiments seine Strategie auswählt. Außerdem geht er davon aus, daß auch Boromir eine gemischte Strategie verwendet²⁷. Schauen wir uns das mal ganz einfach im allgemeinen Fall eines Spiels mit jeweils zwei Strategien an:

		B_1	B_2
		q	$1 - q$
A_1	p	a	b
A_2	$1 - p$	c	d

²⁶Jede reine Strategie ist eine gemischte Strategie, bei der die Wahrscheinlichkeiten als 1 und 0 gewählt werden.

²⁷Das ist keine Einschränkung, wir brauchen ja nur die relativen Häufigkeiten der Strategiewahl als Wahrscheinlichkeiten zu interpretieren – und mal ehrlich, wer weiss schon, was im Kopf eines Helden aus Gondor vorgeht?

Die *erwartete Auszahlung* ist dann

$$\begin{aligned} x(p, q) &= pqa + p(1-q)b + (1-p)qc + (1-p)(1-q)d \\ &= q(pa - pb + (1-p)c - (1-p)d) + pb + (1-p)d \\ &= q(c - d + p(a - b - c + d)) + pb + (1-p)d. \end{aligned}$$

Und hier bekommt Aragorn große Augen: Wählt er nämlich

$$p = \frac{d - c}{a - b - c + d} \iff c - d + p(a - b - c + d) = 0, \quad (1.7)$$

dann hängt die erwartete Auszahlung nicht mehr davon ab, was Boromir macht. Und das ist die optimale *gemischte Strategie*: Schalte einfach den Einfluß des Gegenspielers aus. Gemischte Strategien kann man sogar nutzen, um Optimierungsprobleme zu lösen. Dazu ein Beispiel aus [16].

Beispiel 1.13 (Der Regenschirmverkäufer) *Nach Einführung der Demokratie in Mitteleuropa müssen sich auch Könige neue Betätigungsfelder suchen. Aragorn entscheidet sich, Regenschirme und Sonnenbrillen²⁸ zu verkaufen. An einem regnerischen Tag kann er 500 Regenschirme verkaufen, an einem sonnigen Tag 100 Regenschirme und 1000 Sonnenbrillen. Regenschirme kosten .5 Goldstücke und werden für ein Goldstück verkauft, Sonnenbrillen kosten .2 Goldstücke und bringen .5. Er hat 250 Goldstücke zur Verfügung²⁹, die er investieren kann – was er nicht verkauft ist verloren. Was ist die optimale Strategie?*

Zuerst stellen wir einmal fest, daß es zwei extremale Strategien gibt: “Kaufe für Regen” (also 500 Schirme) oder “Kaufe für Sonnenschein” (also 100 Schirme und 1000 Sonnenbrillen). Gehen wir außerdem davon aus, daß es unsportlich ist, Gandalf um kleine Wettertricks zu bitten, dann erhalten wir die folgende Auszahlungsmatrix³⁰

	Regen	Sonne
Regen	250	-150
Sonne	-150	350

die offensichtlich³¹ keinen Sattelpunkt hat. Die optimale *wetterunabhängige* Strategie ist dann

$$p = \frac{350 - (-150)}{250 - (-150) - (-150) + 350} = \frac{500}{900} = \frac{5}{9},$$

das heißt, mit Wahrscheinlichkeit $\frac{5}{9}$ wird er für Regen einkaufen, mit Wahrscheinlichkeit $\frac{4}{9}$ für Sonnenschein. Sein erwarteter Gewinn ist dann

$$\begin{array}{cc} \text{Regen} & \text{Sonne} \\ \frac{5}{9} \times 250 - \frac{4}{9} \times 150 = 72.222 & \frac{5}{9} \times (-150) + \frac{4}{9} \times 350 = 72.222 \end{array}$$

²⁸So verkauft man bei jedem was.

²⁹Offenbar sind auch Könige nicht mehr das, was sie mal waren.

³⁰Der zweite Spieler ist jetzt das Wetter.

³¹Übungsaufgabe!

und somit wirklich vom Wetter unabhängig.

Übung 1.3 Was ist die optimale gemischte Strategie für das Spiel “Stein, Schere, Papier” aus Beispiel 1.12. \diamond

Im allgemeinen können wir aber nicht erwarten, daß es immer gemischte Strategien gibt, die den Gegner “neutralisieren”, zumindest was den Erwartungswert angeht.

Beispiel 1.14 *Vielen Leuten ist “Stein, Schere, Papier” zu langweilig und sie erweitern das Spiel um die Wahl “Brunnen”, wobei Stein und Schere in den Brunnen fallen, Papier hingegen den Brunnen abdeckt. Die Auszahlungsmatrix ist also*

	Stein	Schere	Papier	Brunnen
Stein	0	1	-1	-1
Schere	-1	0	1	-1
Papier	1	-1	0	1
Brunnen	1	1	-1	0

Man beachte, daß das Spiel nun asymmetrisch wird, da es zwei Strategien gibt, die zweimal gewinnen und einmal verlieren, nämlich Papier und Brunnen, während bei Stein und Schere die “Bilanz” negativ ist.

Würden wir nun für einen der Spieler³² eine Strategie suchen, die den anderen Spieler neutralisiert, dann müsste

$$x(p, q) = [q_1 q_2 q_3 q_4] \begin{bmatrix} 0 & 1 & -1 & -1 \\ -1 & 0 & 1 & -1 \\ 1 & -1 & 0 & 1 \\ 1 & 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix} =: q^T A p$$

von q unabhängig sein, was dann der Fall ist, wenn

$$A p = 0 \quad \text{oder} \quad A p = c \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad c \neq 0,$$

ist, denn schließlich haben wir es bei q ja mit Wahrscheinlichkeiten zu tun, die sich zu 1 addieren. Lösen wir das lineare Gleichungssystem nun nach p , dann erhalten wir allerdings die beiden Lösungen

$$p = 0 \quad \text{und} \quad p = \frac{1}{4} \begin{bmatrix} 3 \\ -1 \\ 1 \\ -3 \end{bmatrix}$$

³²Bezüglich der Spieler ist das Spiel immer noch symmetrisch!

und entweder summieren sich die Einträge von p nicht zu 1 oder sie haben positives und negatives Vorzeichen! Keine der beiden "Lösungen" ist also eine Wahrscheinlichkeitsverteilung und somit liefert der Ansatz, den Gegner auszuschalten, auch keine gemischte Strategie.

Daß es trotzdem immer eine optimale gemischte Strategie gibt, ist eine nichttriviale Aussage der Spieltheorie [8], aber wie man die bestimmt, das werden wir später sehen!

You may call it 'nonsense' if you like, [...] but I've heard nonsense, compared with which that would be as sensible as a dictionary.

L. Carroll, *Through the looking glass*

Minima, Ableitungen, Analysis

2

In diesem Kapitel wollen wir uns einmal ein paar ganz allgemeine mathematische Konzepte ansehen, mit deren Hilfe wir Minima mehr oder weniger gut beschreiben und identifizieren können – insbesondere sind das auch Hilfsmittel, die von mathematischer Software verwendet werden.

2.1 Ableitungen

Ableitungen von Funktionen sind “Dinge”, die man eigentlich aus dem Schulunterricht kennen sollte, aber was meistens hängengeblieben ist, das sind Halb“wahrheiten” der Form

Das ist das, was aus x^n ein nx^{n-1} und aus dem Sinus einen Cosinus gemacht hat und immer für diese bescheuerten Kurvendiskussionen gebraucht wurde.

Naja, eigentlich ist eine Ableitung schon etwas anderes, sogar etwas sehr nützliches, und zwar eine *lokale Linearisierung* einer Funktion. Man könnte auch sagen, *Information erster Ordnung* über die Funktion. Nehmen wir also mal an, wir hätten eine Funktion f zu untersuchen, und zwar eine richtig fiese, etwa von der Form

$$f(x) = \frac{\sin(3x + 7) e^{27x^2}}{\log(x^2 + 1) + \sqrt{|x| + 1}}.$$

Mit so einer Funktion kann man nicht wirklich viel anfangen und wir müssen uns etwas anderes einfallen lassen, um aus ihr *relevante* Information zu extrahieren. Nehmen wir außerdem an, daß uns die Funktion nur in der Nähe einer Stelle x_0 interessiert. Dann ist natürlich die erste Information, die wir verwenden können der Wert $f(x_0)$ und als Näherungswert in einer Umgebung³³ von x_0 verwenden wir einfach die *konstante Funktion*

$$g_0(x) = f(x_0), \quad \text{unabhängig von } x.$$

³³Im Moment soll eine Umgebung einfach der Bereich $[x_0 - \varepsilon, x_0 + \varepsilon]$, wobei ε in der Analysis per Konvention immer für eine *positive* aber *sehr kleine* Zahl steht.

Dies ist eine Näherung *nullter Ordnung* und eine Funktion heißt *stetig*, wenn diese Näherung beliebig gut wird, solange man nur genügend nahe bei x_0 ist³⁴. Stellt sich natürlich die Frage: “geht das denn nicht besser?” Klar, wenn wir für die Funktion g , die wir als lokale Näherung mehr Freiheitsgrade zulassen, zum Beispiel, indem wir die Funktion g als *Gerade* der Form

$$g(x) = ax + b = a(x - x_0) + \underbrace{(ax_0 + b)}_{=:c} = a(x - x_0) + c.$$

Der kleine “Kunstgriff”, von x auf $x - x_0$ zu wechseln ist so banal wie hilfreich: Aus der Forderung, daß g auch den Funktionswert an der Stelle x_0 reproduzieren soll³⁵, erhalten wir, daß

$$f(x_0) = g(x_0) = a(x - x_0) + c \quad \implies \quad c = f(x_0),$$

also ist

$$g(x) = a(x - x_0) + f(x_0). \quad (2.1)$$

Bleibt also noch die Bestimmung von a . Wenn wir nun wieder nahe genug an x_0 herangehen, dann wird der erste Term immer kleiner und wir haben wieder eine lokale Näherung an den Funktionswert. Aber: Das hätten wir mit unserer Näherung g_0 auch haben können, dafür brauchen wir keinen zusätzlichen Parameter a , noch viel schlimmer, das geht mit *jedem* Parameter a . Nun ist aber für kleine Werte von h

$$\begin{aligned} g(x_0 + h) - f(x_0 + h) &= a(x_0 + h - x_0) + f(x_0) - f(x_0 + h) \\ &= ah - [f(x_0 + h) - f(x_0)] \end{aligned}$$

und wir möchten, daß dieser Fehler *von erster Ordnung* klein wird, daß also die Näherung $g(x_0 + h) - f(x_0 + h)$ *schneller* als h Null wird³⁶. Diese Forderung nach einer “optimalen” linearen Näherung ist in Abb. 2.1 dargestellt – es gibt zwar jede Menge Geraden durch den Punkt x_0 , aber nur eine von ihnen ist wirklich optimal! Nicht jede Funktion läßt sich auf diese Art und Weise *linearisieren*, siehe Beispiel 2.1, aber wenn es geht, dann nennt man die Funktion *differenzierbar* und es muß

$$0 \leftarrow \frac{1}{h} [g(x_0 + h) - f(x_0 + h)] = a - \frac{f(x_0 + h) - f(x_0)}{h},$$

also

$$a = \lim_{h \rightarrow 0} \frac{f(x_0 + h) - f(x_0)}{h} =: f'(x_0)$$

gelten – das ist der allseits beliebte *Differenzenquotient*, der vielleicht sogar noch aus Schulzeiten bekannt ist.

³⁴Die exakte Definition von Stetigkeit und Konvergenz schenken wir uns hier!

³⁵Was für eine Näherungsfunktion wäre das denn sonst?

³⁶Die “erste Ordnung” kommt daher, daß der Fehler schneller als h^1 verschwinden soll; dies ist konsistent mit unserem Begriff der “nullten Ordnung”, bei dem die Abweichung schneller als $h^0 = 1$ Null werden sollte – das ist nichts anderes als die einfache Forderung, daß der Wert immer kleiner werden soll.

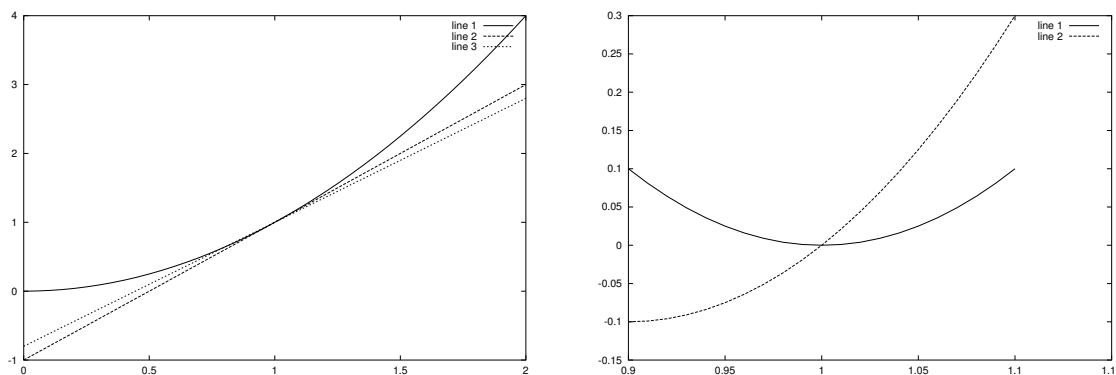


Abbildung 2.1: Zwei lokale lineare Näherungen an die Parabel $f(x) = x^2$ an der Stelle $x_0 = 1$. Da beide durch $(1, 1)$ gehen sind sie Näherungen nullter Ordnung, aber das Bild rechts, in dem die *Abweichung*, bereits dividiert durch h , für $h \in [-.1, .1]$ geplottet ist zeigt, daß nur für die “optimale” Gerade erster Ordnung dieser Fehler wirklich schneller als h Null wird..

Beispiel 2.1 (Wurzel an $x = 0$) Wir setzen

$$f(x) = \sqrt{|x|} = \begin{cases} \sqrt{x}, & x \geq 0, \\ \sqrt{-x}, & x < 0, \end{cases}$$

und $x_0 = 0$, dann ist

$$f(x_0 + h) - f(x_0) = \sqrt{|h|}$$

und ganz egal wie wir a wählen ist für $h > 0$ ³⁷

$$ah - \sqrt{|h|} = a\sqrt{h^2} - \sqrt{h} = \underbrace{\sqrt{h}}_{\rightarrow -1} (a\sqrt{h} - 1),$$

was viel langsamer als h , nämlich nur mit der Geschwindigkeit \sqrt{h} gegen Null geht. Betrachten wir den Differenzenquotienten an $x_0 = 0$, dann erhalten wir da den Wert $1/\sqrt{|h|}$, was für $h \rightarrow 0$ ja leider gegen ∞ davonläuft.

Übung 2.1 Zur Erinnerung: Die Ableitung der Funktion x^n ist nx^{n-1} . Aber wie kommt man drauf?

1. Zeigen Sie, daß für $f(x) = x^2$ die Identität

$$f(x+h) - f(x) = 2hx + h^2$$

gilt.

³⁷Für $h < 0$ geht's im wesentlichen genauso.

2. Zeigen Sie, daß

$$(x + h)^n = x^n + nhx^{n-1} + h^2(\dots)$$

ist, indem Sie es erst für $n = 1$ ausrechnen und dann die Tatsache ausnutzen, daß

$$(x + h)^n = (x + h)(x + h)^{n-1} = (x + h)(x^{n-1} + (n-1)hx + h(\dots))$$

gilt. Diese Vorgehensweise bezeichnet man als *vollständige Induktion*.

3. Leiten Sie nun die Ableitungsformel über den Differenzenquotienten her.

◇

Wir fassen zusammen:

Eine Funktion f heißt differenzierbar an x_0 , wenn sie lokal³⁸ *effizient* linearisierbar, wenn es also eine lineare Funktion

$$g_1(x) := a(x - x_0) + f(x_0)$$

gibt, so daß $f(x_0 + h) - g_1(x_0 + h)$ schneller als h Null wird. Es gibt höchstens einen solchen Wert a , den wir mit $f'(x_0)$ und dieser Wert ergibt sich als *Grenzwert*³⁹ der Differenzenquotienten

$$\frac{f(x_0 + h) - f(x_0)}{h}, \quad h \neq 0.$$

Außerdem hat die Sichtweise der Ableitung als effiziente lineare Näherung noch einen weiteren Vorteil: Wir können sie auch auf mehrere Variablen übertragen⁴⁰! Für einen Vektor $x = [x_1 \cdots x_n]^T$ hat eine lineare Funktion $g(x)$ die Form

$$g(x) = a^T(x - x_0) + c, \quad c = f(x_0),$$

die Ableitung f' ist jetzt also ein *Vektor* – was für eine Überraschung. Für einen Vektor $y \in \mathbb{R}^n$, möglicherweise so normiert, daß er Länge 1 hat⁴¹ und einen *Skalierungsparameter* $h \in \mathbb{R}$ heißt “effiziente Annäherung” also, daß

$$\begin{aligned} g(x_0 + hy) - f(x_0 + hy) &= a^T(x_0 + hy - x_0) + f(x_0) - f(x_0 + hy) \\ &= h a^T y - [f(x_0 + hy) - f(x_0)] \end{aligned}$$

³⁸Schließlich interessiert uns ja nur die Qualität der Näherung wenn h immer kleiner wird.

³⁹Mal ganz ehrlich – wer weiß wirklich, was ein Grenzwert ist? Mit solchen Begriffen wird gerne jongliert, ohne zu wissen, was sie eigentlich bedeuten und sowas kann zu geradezu katastrophalen Mißverständnissen führen, siehe [17].

⁴⁰Was wir für unsere Paraboloiden aus der linearen Regression, Beispiel 1.6, die ja eine Funktion in den beiden Variablen a und b ist, auch dringend brauchen.

⁴¹Das ist in keiner Weise notwendig, manchmal aber ganz hilfreich. Wobei natürlich die Frage bleibt, was dann wieder die Länge eines Vektors ist.

schneller als h null wird, daß also

$$0 \leftarrow \frac{1}{h} [g(x_0 + hy) - f(x_0 + hy)] = a^T y - \frac{f(x_0 + hy) - f(x_0)}{h} \quad (2.2)$$

Das Konzept in (2.2) ist das der *Richtungsableitung*: Der Vektor y ist die Richtung und durch den Skalierungsparameter h haben wir es nur mit einer ganz normalen, *eindimensionalen* Ableitung zu tun. Dieses Prinzip ist in Abb 2.2 grafisch dargestellt. Aber es bleibt immer noch die

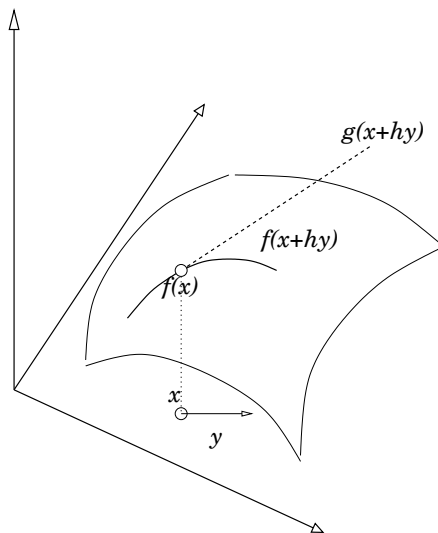


Abbildung 2.2: Das Prinzip der Richtungsableitung an einem Punkt x . Die Funktion $h \mapsto f(x + hy)$ beschreibt eine Kurve auf der Fläche, deren Parameter entlang y läuft und dabei positive wie negative Werte annehmen kann. Die lineare Näherung ist dann die eindimensionale Tangente an diese Kurve.

Frage, wie man den “Ableitungsvektor” a bestimmen kann, denn bisher haben wir die *unendlich vielen* Forderungen⁴²

$$a^T y = \lim_{h \rightarrow 0} \frac{f(x_0 + hy) - f(x_0)}{h}, \quad y \in \mathbb{R}^n \quad (2.3)$$

ist. Hier aber können wir es uns sehr einfach machen, indem wir für y die n *kanonischen Einheitsvektoren*⁴³

$$e_1 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad e_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \dots, \quad e_n = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

⁴²Immer vorausgesetzt, wir wüssten, was ein Grenzwert alias *Limes* alias “lim” ist!

⁴³Diese sind auch die Spaltenvektoren der Einheitsmatrix, also $I = [e_1 \cdots e_n]$.

verwenden, denn dann ist

$$a_j = a^T e_j = \lim_{h \rightarrow 0} \frac{f(x_0 + h e_j) - f(x_0)}{h} =: \frac{\partial}{\partial x_j} f, \quad j = 1, \dots, n. \quad (2.4)$$

Das Symbol $\partial/\partial x_j$ bezeichnet die *partielle Ableitung* von f nach der Variablen x_j und lässt sich ganz einfach mit Schulmathematik bestimmen⁴⁴:

Man betrachtet die Funktion $f(x_1, \dots, x_n)$ als eine Funktion in der *einen Variablen* x_j mit den *Parametern* $x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n$ und bestimmt deren univariate Ableitung.

Beispiel 2.2 Betrachten wir doch einmal die Funktion

$$F(a, b) = a^2 \left(\sum_{j=1}^n x_j^2 \right) + 2ab \left(\sum_{j=1}^n x_j \right) + b^2 - 2a \left(\sum_{j=1}^n x_j y_j \right) - 2b \left(\sum_{j=1}^n y_j \right) + \left(\sum_{j=1}^n y_j^2 \right)$$

aus Beispiel 1.6, der linearen Regression. Bei der partiellen Ableitung nach a sind alle Terme, in denen a nicht auftaucht, Konstanten und fallen daher weg, weswegen wir

$$\frac{\partial}{\partial a} F(a, b) = 2a \left(\sum_{j=1}^n x_j^2 \right) + 2b \left(\sum_{j=1}^n x_j \right) - 2 \left(\sum_{j=1}^n x_j y_j \right)$$

und entsprechend

$$\frac{\partial}{\partial b} F(a, b) = 2a \left(\sum_{j=1}^n x_j \right) + 2b - 2 \left(\sum_{j=1}^n y_j \right)$$

ist.

Definition 2.3 Den aus den partiellen Ableitungen gebildeten Vektor

$$\nabla f(x) = \begin{bmatrix} \frac{\partial}{\partial x_1} f \\ \vdots \\ \frac{\partial}{\partial x_n} f \end{bmatrix}$$

bezeichnet man als *Gradienten* von f und die *Richtungsableitung*⁴⁵ schreibt man auch als

$$D_y f(x) = y^T \nabla f = \lim_{h \rightarrow 0} \frac{f(x + hy) - f(x)}{h}$$

⁴⁴Und das ist wirklich absolut ganz und gar nichts geheimnisvolles dabei!

⁴⁵Was hier steht ist nicht vollständig korrekt. Es gibt Funktionen, bei denen zwar partielle Ableitungen existieren, aber keine Richtungsableitungen, oder bei denen $D_y f$ existiert, aber nicht gleich $y^T \nabla f$ ist. Solche Funktionen sind aber eher akademische kleine Spielzeuge für Mathematiker, von denen man besser die Finger lässt und die in der "Realität" nicht auftreten. Und wenn sie es tun, dann hat man halt ein Problem ...

Übung 2.2 Bestimmen Sie den Gradienten der linearen Funktion

$$\ell(x) = c^T x + d.$$

◇

Der Gradient gibt uns eine wichtige Information, er sagt uns nämlich, in welche Richtung die Funktion f an der Stelle x_0 am steilsten ansteigt! Um genau zu sein:

Wenn wir an der Stelle x_0 auf der “optimalen” Näherung erster Ordnung in Richtung des Gradienten gehen, dann gewinnen wir am meisten.

Das sieht man übrigens sehr einfach: ist y ein beliebiger Vektor, der Fairness halber so normiert, daß er dieselbe Länge wie der Gradient hat⁴⁶, dann ist

$$\begin{aligned} g_1(x_0 + y) - g_1(x_0) &= \nabla^T f(x_0)(x_0 + y - x_0) + f(x_0) - \underbrace{\nabla^T f(x_0)(x_0 - x_0)}_{=0} - f(x_0) \\ &= \nabla^T f(x_0) y. \end{aligned}$$

Nun zerlegen wir y in einen Anteil *parallel* zu ∇f und einen Anteil, der orthogonal dazu ist, also

$$y = \lambda \nabla f + y^\perp, \quad \nabla^T f y^\perp = 0,$$

wobei $|\lambda| \leq 1$ ist⁴⁷ dann ist

$$g_1(x_0 + y) - g_1(x_0) = \lambda \nabla^T f \nabla f = \lambda \|\nabla f\|_2^2$$

und, wie man auch in Abb. 2.3 sieht, fällt der Gewinn am größten aus, wenn $\lambda = 1$, also $y = \nabla f$ ist. Am weitesten nach unten kommt man hingegen, wenn man $y = -\nabla f$ wählt. Aber **Achtung:**

Der “optimale” Gewinn bezieht sich nicht auf die eigentliche Funktion, sondern nur auf die lineare Näherung! Das heißt, daß die steilste Anstiegsrichtung nur in einem “unendlich kleinen” Bereich um den Punkt x_0 herum zu guten Ergebnissen führen kann.

Diese “Lokalitätsgeschichte” ist ein generelles Problem der Optimierung: abgesehen davon, daß man “gute” Aufstiegsrichtungen wählen muß⁴⁸, ist ein weiteres “Kunststück” die Bestimmung der Schrittweite, also wie weit man in diese Richtung voranschreiten soll. Dieser Wert muß klein genug sein, um noch die Lokalität der linearen Näherung nutzen zu können und gleichzeitig so groß, daß man trotzdem etwas gewinnt, wenn man in diese Richtung geht⁴⁹

⁴⁶Und hier nehmen wir die euklidische Länge, also den “normalen” Abstand in Räumen, die man sich vorstellen kann wie \mathbb{R}^2 oder \mathbb{R}^3 .

⁴⁷Die Quadrate der euklidischen Längen zweier *orthogonaler* Vektoren addieren sich einfach – das ist der allseits beliebte Satz des Pythagoras!

⁴⁸Und erstaunlicherweise ist der Gradient gar keine so gute Strategie!

⁴⁹Mit diesem Dilemma, das entscheidend für die Funktion von Optimierungsverfahren ist, werden wir uns im Rahmen dieser Vorlesung nicht weiter beschäftigen; bei “fertigen” Optimierungsroutinen in mathematischer Software sollte die Schrittweitensteuerung mehr oder weniger gut implementiert sein.

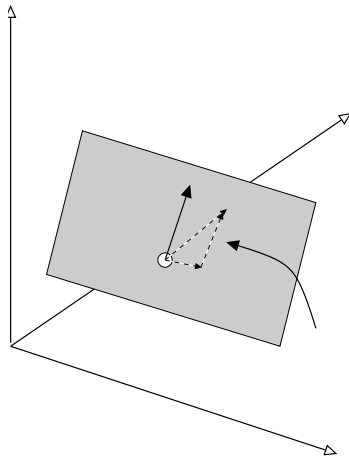


Abbildung 2.3: Zerlegung eines Vektors als Summe von Vektoren, die parallel bzw. orthogonal zum Gradienten sind. Entlang letzterem ist die lineare Funktion aber konstant und er spielt somit keinen Rolle. Der parallele Anteil ist aber *kürzer* als der Gradient selbst und daher fällt auch der Gewinn bei dieser “Aufstiegsrichtung” geringer aus.

2.2 Minima, Maxima, Extrema

Als nächstes wollen wir uns ansehen, wie uns der Gradient hilft, Minimal- und Maximalstellen⁵⁰ einer Funktion $F : \mathbb{R}^n \rightarrow \mathbb{R}$ bzw. $F : D \rightarrow \mathbb{R}^n$ zu erkennen. Dabei ist es sehr wohl von Bedeutung, wie die Menge D beschaffen ist.

Beispiel 2.4 Die univariate Funktion $F(x) = x$ hat auf ganz \mathbb{R} kein Maximum, auf der Menge $D = [a, b]$ hingegen das Maximum b .

Aber auch ohne Beschränkungen macht es durchaus Sinn, zwei Typen von Extrema zu unterscheiden, und zwar *lokale* und *globale* Extrema. Ein *lokales* Extremum zeichnet sich dadurch aus, daß es eine ganze Umgebung eines Punktes gibt, in der die Funktion nur größere (im Falle eines lokalen Minimums) bzw. kleinere (im Falle eines lokalen Maximums) Werte annimmt. Dabei ist es wichtig, sich darüber klarzuwerden, was eine Umgebung ist. Dabei machen wir es uns eher einfach.

Definition 2.5 Eine Umgebung eines Punktes $x \in \mathbb{R}^n$ besteht aus allen Punkten, die von diesem Punkt einen gewissen (euklidischen⁵¹ Maximalabstand haben. Ein Punkt heißt x innerer Punkt der Menge D , wenn es eine Umgebung des Punktes gibt, die ganz im Bereich D liegt, andernfalls nennt man x einen Randpunkt.

⁵⁰Und auf die sind wir ja im Rahmen dieser Vorlesung aus.

⁵¹Das ist in allen “vorstellbaren” Räumen genau das, was man sich so unter einem Abstand vorstellt!

Ein Umgebung ist also eine kleine, ja sogar *beliebig kleine*⁵², Kugel um den Punkt x herum. Wichtig ist, daß man in so einer Umgebung in kleines Stückchen in *jede* beliebige Richtung gehen kann und daß die Länge dieses Stückchens von der Richtung *unabhängig* ist.

Bei einem Intervall $[a, b]$ kennt man das ja: Die inneren Punkte sind all diejenigen x , die $a < x < b$ erfüllen, denn dann bilden alle Punkte aus $[x - \varepsilon, x + \varepsilon]$, $\varepsilon < x - a$ und $\varepsilon < b - x$ eine dieser magischen Umgebungen. Bei Gebilden im \mathbb{R}^s oder \mathbb{R}^3 kann das schon viel wilder sein. Naiv könnte man sagen, ein Randpunkt einer Menge würde sich dadurch auszeichnen, daß es eine Richtung gibt, mit der man immer aus der Menge herausfällt, und wenn man noch so kleine Schritte macht. Das stimmt aber nicht! Was man wirklich braucht ist, daß es zu jeder Schrittweite, und sei sie noch so klein, eine Richtung gibt, die mit dieser Schrittweite außerhalb ladet. Abb. 2.4 zeigt ein paar Beispiele für solche topologische “Perversionen”, die aber

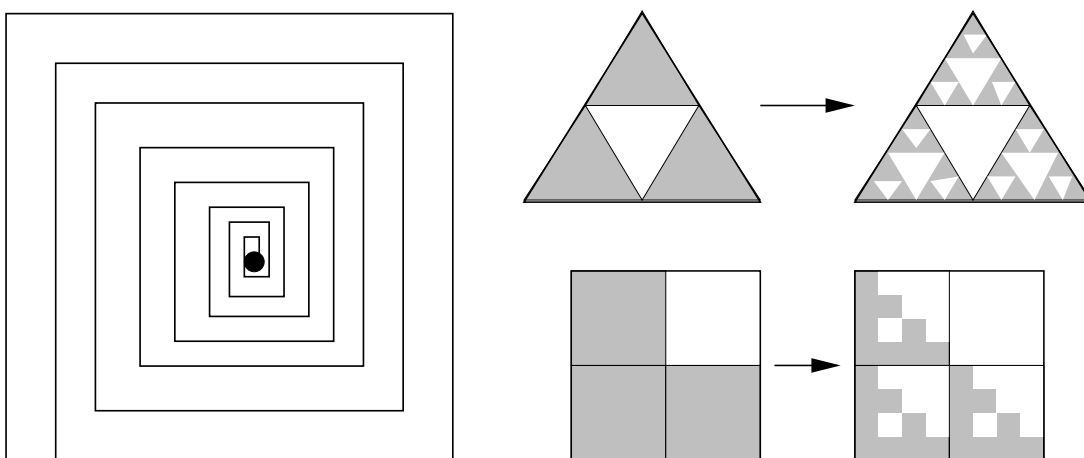


Abbildung 2.4: *Links*: “Todesspirale”, die sich gegen den Punkt in der Mitte windet und dabei immer dünner wird (die Menge braucht nicht ausschließlich aus einer Geraden zu bestehen, sondern kann durchaus “Ausdehnung”, also innere Punkte, haben. In jeder Richtung und beliebiger Nähe findet man einen Punkt in der Menge und trotzdem ist der Mittelpunkt ein Randpunkt.

Rechts: Zwei *Fraktale*, bei denen aus dem Quadrat die rechte obere Ecke und aus dem Dreieck das mittlere Dreieck entfernt werden, was man dann bei allen übriggebliebenen Teilen wieder macht. Die Mengen, die man so erhält, bestehen dann nur aus Rand.

zugegebenermaßen eher für Mathematiker als für “normale” Menschen interessant sind. Wir wenden uns lieber unserem Thema zu, nämlich den Extrema! Mit Hilfe des Umgebungsbe-griff können wir nun sagen, was unter einem lokalen Extremum zu verstehen ist:

Ein Punkt x heißt *lokales Minimum* einer Funktion f , wenn es eine Umgebung dieses Punktes gibt, in der die Funktion keinen größeren Wert annimmt⁵³. Ein lokales

⁵²Sie darf aber nicht nur aus dem Punkt x bestehen, der Maximalabstand, um den es hier geht, muss **strikt positiv** sein.

⁵³Die Definition eines lokalen Maximums läßt sich daraus leicht ableiten!

Extremum muß also *immer* ein innerer Punkt sein.

Und nun das “Credo” der Jagd nach Minima und Maxima⁵⁴:

Das Maximum bzw. Minimum einer Funktion f findet man unter den *lokalen* Maxima bzw. Minima und den Randpunkten.

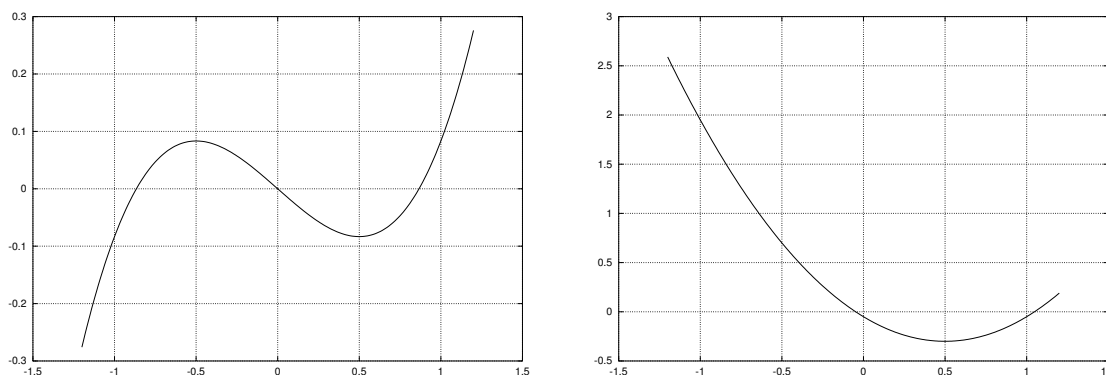


Abbildung 2.5: Lokale und globale Minima und Maxima. Die Funktion links hat ein *lokales* Maximum an $x = -\frac{1}{2}$ und ein lokales Minimum an $\frac{1}{2}$, die *globalen* Extrema werden aber am Rand angenommen. Der Parabelast rechts hingegen hat ein gleichzeitig lokales und globales Minimum, *kein* lokales Maximum, aber ein globales Maximum am linken Rand.

Wie man in Abb. 2.5 erkennen kann, kann das mit den lokalen und globalen Extrema schon in einer Variablen recht amüsant werden. Nun haben wir aber alles beisammen, um die *lokalen* Extrema einer Funktion in n Variablen beschreiben zu können.

Nehmen wir an, an einer Stelle x_0 habe die *differenzierbare* Funktion f ein lokales Minimum, das heißt es gibt so eine magische Umgebung U , so daß für alle $x \in U$ die Minimalitätsbedingung $f(x_0) \leq f(x)$ erfüllt ist. Wir können das aber auch anders sagen: Für alle normierten Richtungen y mit Länge 1 und alle *hinreichend kleinen* aber positiven Werte von h ist

$$f(x_0 + hy) \leq f(x_0), \quad \text{also} \quad 0 \leq \frac{f(x_0 + hy) - f(x_0)}{h}$$

und somit ergibt sich, wenn h nun langsam Null wird **ohne** dabei ins Negative abzugleiten, für die Richtungsableitung

$$D_y f(x_0) = \lim_{h \rightarrow 0} \frac{f(x_0 + hy) - f(x_0)}{h} \geq 0.$$

So, jetzt machen wir dasselbe für $-h$ anstelle von h . Weil x_0 ein lokales Minimum ist und $x_0 - hy$ ebenfalls in der Umgebung liegt, erhalten wir, daß

$$f(x_0 - hy) \leq f(x_0)$$

⁵⁴Was man als eine Art Extremsport, genauer als eine Art Extremdenksport ansehen könnte.

und somit⁵⁵

$$0 \geq \lim_{h \rightarrow 0} \frac{f(x_0 - hy) - f(x_0)}{-h} \rightarrow D_y f(x_0).$$

Zusammen liefert uns das dann nichts anderes als

$$0 \leq D_y f(x_0) \leq 0 \quad \implies \quad D_y f(x_0) = 0$$

und zwar für *alle* denkbaren Richtungen y . Damit muß aber der Gradient, also die “gesamte” Ableitung von f an der Stelle x_0 gleich Null sein!

Satz 2.6 Ist x_0 ein lokales Extremum der Funktion f , dann ist $\nabla f(x_0) = 0$.

Abb. 2.5 und Schulmathematik nähren ja bereits den Verdacht: An den lokalen Extremalstellen muß die Ableitung der Funktion den Wert Null haben, muß die “optimale Näherungsgerade” Steigung Null haben. Und das ist in zwei Variablen auch nicht anders, da ist die “Tangentialebene” an ein Extremum ein “Blatt Papier”, das sich waagrecht auf den Funktionsgraphen legen lässt. Das kann man in Abb. 2.6 ganz gut visuell verifizieren⁵⁶. Allerdings sollte man nie ver-

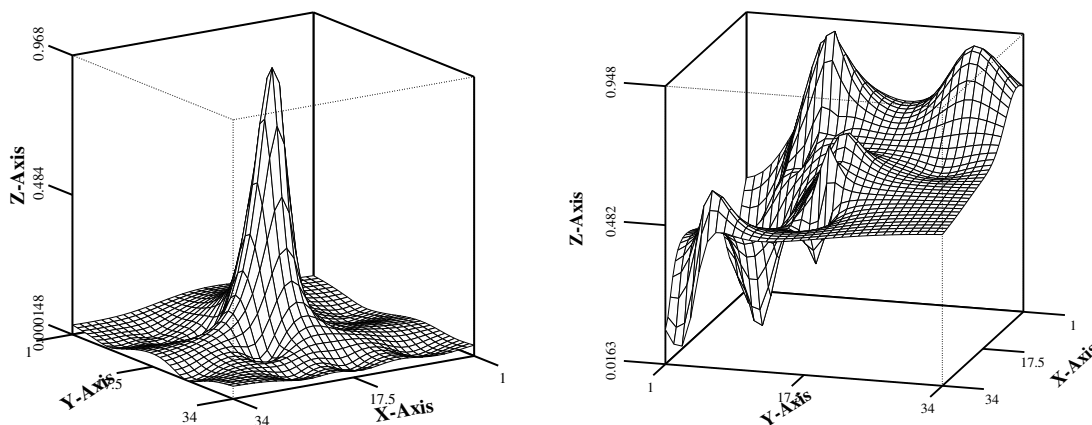


Abbildung 2.6: Zwei Beispiele für lokale und globale Extrema von Funktionen in zwei Variablen aus [11]. Man sieht gerade auch rechts jede Menge von “Flat Spots”, die auf lokale Extrema hindeuten.

gessen, daß zwar jedes lokale Extremum⁵⁷ einen verschwindenden Gradienten (oder eben eine verschwindende Ableitung) bedingt, daß aber die Umkehrung **nicht** stimmt, daß also nicht jeder Punkt, an dem die Ableitung Null ist auch wirklich ein lokales Extremum sein muß. Die Funktion auf der rechten Seite von Abb. 2.7 ist ein sogenannter *Sattelpunkt*, der physialisch eine

⁵⁵Die Division durch $-h$ dreht das Ungleichungszeichen um!

⁵⁶Oder auch nur ganz einfach sehen.

⁵⁷“Lokal” ist hier wichtig – ein Extremum am Rand hat selten verschwindende Ableitung, denn da kann man normalerweise den $h/ - h$ -Trick einfach nicht machen, wenn man in einer von den beiden Richtungen dauernd aus dem Bereich D herausfällt!

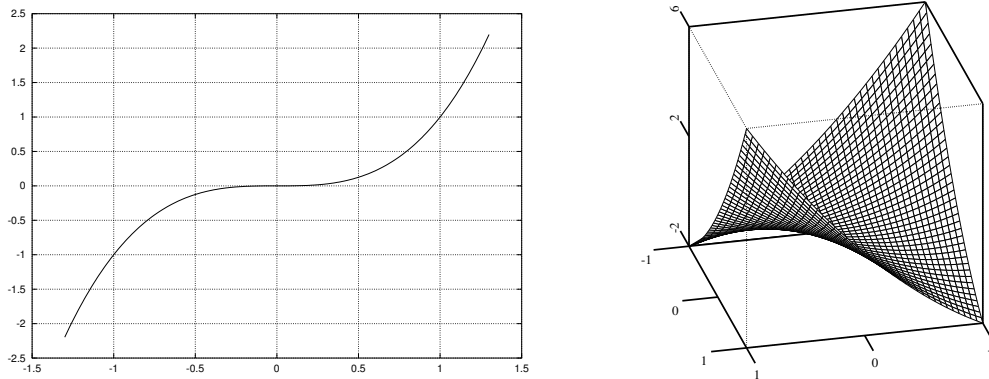


Abbildung 2.7: Links: Die Parabel $f(x) = x^3$, deren Ableitung $f'(x) = 3x^2$ an $x = 0$ verschwindet, wo man aber weit und breit kein lokales Extremum, sondern nur einen sogenannten *Wendepunkt* findet.

Rechts: Ein *Sattelpunkt* an der Stelle $[0, 0]^T$. In eine Richtung steigt die Funktion strikt an, in die andere fällt sie ab. Trotzdem ist der Gradient an dieser Stelle Null.

labiler Gleichgewichtspunkt ist, zumindest in die “Abfallrichtung”: Legt man einen Ball in den Sattelpunkt, so wird er, wenn er einen Impuls rein in Richtung des Aufstiegs erhält, wieder in diesen Punkt zurückfallen, wenn der Impuls⁵⁸ aber auch nur einen kleinen Anteil in Richtung des Abstiegs enthält, dann geht es bergab, und zwar rapide. Daß Sattelpunkte in der Spieltheorie stabilisierend wirken⁵⁹, liegt an der einfachen Tatsache, daß ja hier einer der Spieler den Ball bergauf, der andere bergab bewegen möchte und sich so die Anstrengungen egalieren.

Aber zurück zu unserem eigentlichen Problem, nämlich dem Identifizieren von lokalen Extrema. Wir fassen mal kurz zusammen, was wir bisher herausgefunden haben:

Ein lokales Extremum zeichnet sich dadurch aus, daß dort der Gradient verschwindet. Aber nicht jeder Punkt, an dem der Gradient verschwindet ist ein lokales Extremum und der Gradient sagt uns absolut nichts, ob es sich um ein Maximum oder Minimum handelt.

Die Antwort auf dieses Problem kennt man vielleicht noch aus der Schule: *zweite Ableitungen* und das seltsame Konzept der positiven und negativen Krümmung. Aber wir wollen’s natürlich wieder etwas anders machen, nämlich mit effektiver Annäherung durch Funktionen, die gerade ein lokales Extremum haben, und das sind Parabeln oder Paraboloide.

2.3 Zweite Ableitungen und Parabeln

Daß eine *Parabel*, also eine quadratische Funktion der Form

$$f(x) = ax^2 + bx + c, \quad a \neq 0,$$

⁵⁸Der ist ein Vektor!

⁵⁹Das mag ja bei dem labilen Gleichgewicht erst einmal paradox erscheinen.

gerade ein Extremum, also Maximum oder Minimum, hat, ist zwar “optisch” klar, spätestens

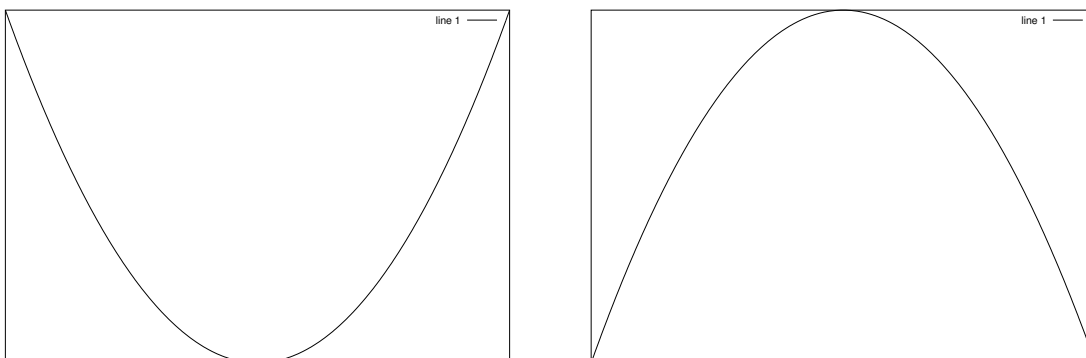


Abbildung 2.8: Eine nach oben und eine nach unten geöffnete Parabel, also etwas, das man durchaus schon einmal im Schulunterricht gesehen haben könnte.

nach einem Blick auf Abb. 2.8, aber wie kommt man drauf? Nun, um ein (lokales) Extremum zu sein⁶⁰, muß ja x die Forderung

$$0 = f'(x) = 2ax + b \quad \implies \quad x = \frac{b}{2a}$$

erfüllt sein – mit anderen Worten, es gibt *höchstens* ein lokales Extremum. Und jetzt kommt die allseits beliebte Fallunterscheidung:

- Ist $a > 0$, dann wird $f(x)$ immer größer werden, wenn x immer größer oder immer kleiner (negativ!) wird, denn dann dominiert irgendwann der Term ax^2 den Rest von x . Einen größten Wert kann f also nicht haben, aber einen *kleinsten* Wert muß es irgendwo annehmen, also ein Minimum, das ein globales, aber somit auch ein lokales Minimum ist. Und nachdem wir nur einen Kandidaten für diese Minimalstelle haben, ist die Wahl schnell entschieden.
- Ist $a < 0$, dann wird $f(x)$ für $x \rightarrow \pm\infty$ ganz analog unter alle Grenzen fallen, sich also gegen $-\infty$ bewegen und muß daher ein Maximum an der Stelle $x = b/2a$ besitzen.

Den “Leitkoeffizienten” a unserer Parabel können wir auch geometrisch als *Krümmung* der Parabel betrachten und zwar kann diese

- *positiv*, also “nach links”⁶¹ gekrümmt bzw. *konvex* sein, wenn $a > 0$ ist oder
- *negativ*, also “nach rechts” gekrümmt, bzw. *konkav* sein, wenn $a < 0$ ist.

⁶⁰Und nachdem wir keine Einschränkungen auf Intervalle betrachten, sondern uns die Parabel “überall” auf ganz \mathbb{R} ansehen, können wir also “Randeffekte” vergessen.

⁶¹Wenn man mit einem kleinen Auto auf der Parabel von $-\infty$ nach $+\infty$ fährt.

Was passiert, wenn keiner von den beiden Fällen eintritt, wenn also $a = 0$ ist? Nun, dann ist unsere “Parabel” ja in Wirklichkeit eine *lineare Funktion* der Form $f(x) = bx + c$ und stellt somit eine Gerade dar, die weder nach links noch nach rechts sondern einfach gar nicht gekrümmt ist. Aber eines noch: Die Krümmung einer Parabel ist überall gleich, also unabhängig vom Punkt x ! Aber sie sagt uns was über Minima und Maxima:

Die Parabel besitzt ein lokales und globales Minimum, wenn sie positiv gekrümmt ist und ein Maximum, wenn sie negativ gekrümmt ist.

Es ist anschaulich naheliegend, daß eine Funktion an einer Stelle x (genau)⁶² dann ein Minimum hat, wenn die Ableitung den Wert Null hat⁶³ und sie positiv gekrümmt ist, siehe Abb. 2.9.

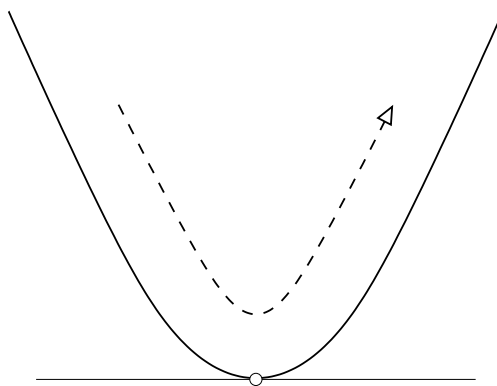


Abbildung 2.9: Ein Minimum liegt vor, wenn die Funktion waagerechte Tangente hat (das wissen wir schon) und positiv gekrümmt ist, denn dann kommt man immer “von oben” in den Punkt und verläßt ihn auch wieder “nach oben”.

Umgekehrt hat man es mit einem Maximum zu tun, wenn an einer Stelle mit verschwindender Ableitung auf eine negative Krümmung stößt, und das ganz unabhängig davon, ob es sich bei dieser Funktion um eine Parabel handelt oder nicht. Aber wie übertragen wir nun den intuitiven Begriff der *Krümmung* auf eine beliebigen Funktion f ? Ganz einfach – genauso, wie wir es mit dem Begriff der Ableitung als Steigung auch gemacht haben:

Die *Krümmung* einer Funktion f an einer Stelle x_0 ist die Krümmung derjenigen Parabel, die die beste *lokale* Näherung der Funktion f unter allen Parabeln darstellt.

Und natürlich sind wir nicht dumm und schreiben unsere Parabel wieder in den “lokalen Koordinaten” um x_0 als

$$g(x) = a (x - x_0)^2 + b (x - x_0) + c,$$

⁶²Leider nicht ganz genau.

⁶³Also eine waagerechte Tangente hat

wobei wir b und c natürlich bereits kennen, denn das sind die Parameter der optimalen linearen Funktion, also⁶⁴

$$b = f'(x_0), \quad a = f(x_0).$$

Und dann schauen wir uns halt den Fehler an, den wir auf diese Art und Weise machen, also

$$f(x_0 + h) - g(x_0) = f(x_0 + h) - ah^2 - bh - f(x_0)$$

Jetzt setzen wir für b die Näherung

$$b \sim \frac{f(x_0 + h/2) - f(x_0)}{h/2} = 2 \frac{f(x_0 + h/2) - f(x_0)}{h}$$

ein⁶⁵ und erhalten, daß

$$f(x_0 + h) - g(x_0) = f(x_0 + h) - 2f(x_0 + h/2) + f(x_0) - ah^2,$$

was nun von zweiter Ordnung, also schneller als h^2 gegen Null geht, wenn der Grenzwert⁶⁶

$$a = \lim_{h \rightarrow 0} \frac{f(x_0 + h) - 2f(x_0 + h/2) + f(x_0)}{h^2}$$

existiert. Dieser Grenzwert ist im wesentlichen die zweite Ableitung von f an der Stelle x_0 , also das, was man erhält, wenn man die Funktion $f''(x)$ noch einmal ableitet. Ganz korrekt ist aber der Wert

$$a = \frac{1}{2} f''(x_0).$$

Jetzt können wir ja das Geheimnis lüften: Die optimale lokale Annäherung durch ein *Polynom vom Grad n* , also einem Ausdruck der Form $a_n x^n + \dots + a_1 x + a_0$ erhält man als

$$p(x) = \sum_{k=0}^n \frac{f^{(k)}(x_0)}{k!} (x - x_0)^k. \quad (2.5)$$

Das ist die sogenannte *Taylorformel*, die für eine n -mal stetig differenzierbare Funktion eine lokale Näherung der Ordnung n liefert, die schneller als h^n Null wird!

Für uns ist aber die folgende Entdeckung wichtig:

Die Frage, ob an einer Stelle x mit waagerechter Tangente ein lokales Minimum oder Maximum vorliegt, können wir dadurch entscheiden, daß wir uns die Krümmung, also die zweite Ableitung von f an dieser Stelle ansehen. Ist diese positiv, dann haben wir ein Minimum gefunden, ist diese negativ, dann haben wir ein Maximum erreicht.

⁶⁴Wir nehmen hier einfach mal an, daß die Funktion "brav" genug ist, so daß wir sie ableiten können.

⁶⁵Warum wir hier $h/2$ und nicht h nehmen ist eine berechtigte Frage. Die Antwort wäre allerdings etwas langatmiger.

⁶⁶Immer noch in einem rein intuitiven Sinne!

Bleibt die Frage, was los ist, wenn wir die zweite Ableitung den Wert Null hat – ganz einfach: Dann ist die Frage nach Maximum oder Minimum mit der vorliegenden Information schlichtweg **nicht entscheidbar**.

Beispiel 2.7 Wir betrachten die drei Funktionen

$$f_1(x) = x^3, \quad f_2(x) = x^4, \quad f_3(x) = -x^4,$$

die alle die Eigenschaft $f'(0) = f''(0) = 0$ haben⁶⁷. Und dennoch hat f_1 weder Minimum noch Maximum an 0, f_2 ein lokales und globales Minimum und f_3 ein lokal-globales Maximum, siehe Abb. 2.10.

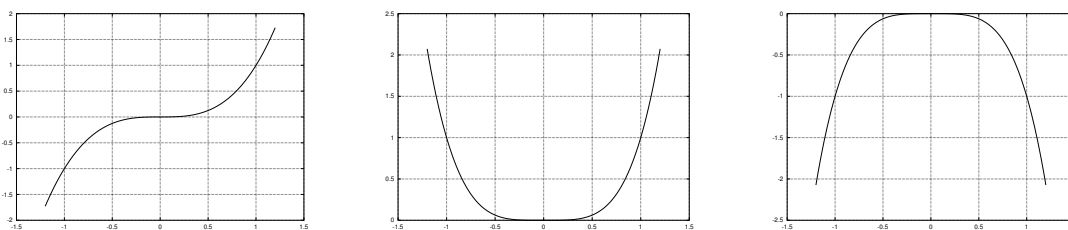


Abbildung 2.10: Drei Funktionen, die alle an der Stelle $x = 0$ waagerechte Tangente und Nullkrümmung haben.

2.4 Paraboloiden und warum alles doch nicht so einfach ist

So, jetzt haben wir also einen Krümmungsbegriff für Funktionen in einer Variablen, der uns obendrein noch mit dem Konzept der zweiten Ableitung, und zwar nicht als formalem Prozess, sondern als Idee der Krümmung, vertraut gemacht hat. Aber das hilft uns leider auch nicht so viel, wenn wir uns mit mehr als einer Variablen herumschlagen müssen, was leider das tägliche Brot der Optimierung ist. Aber da Krümmungen mit Parabeln zu tun haben, macht es sicherlich Sinn, sich mit den Gegenstücken zu Parabeln herumschlagen, nämlich den Funktionen der Form

$$f(x_1, \dots, x_n) = \sum_{j,k=1}^n a_{jk} x_j x_k + \sum_{j=1}^n b_j x_j + c.$$

Als kleiner ‘‘Realitätstest’’: Ist $n = 1$, dann sind die Summen keine Summen und unsere Funktion hat wieder die altbekannte einfache Form $f(x_1) = a_{11} x_1^2 + b_1 x_1 + c$. Nun erinnern wir uns an die Rechenregeln für Matrizen und Vektoren⁶⁸ und erhalten, daß

$$\sum_{j,k=1}^n a_{jk} x_j x_k = \sum_{j=1}^n x_j \sum_{k=1}^n a_{jk} x_k = \sum_{j=1}^n x_j (Ax)_j = x^T Ax,$$

⁶⁷Wer’s nicht glaubt, rechnet dies entweder von Hand oder mit einem geeigneten Programm aus.

⁶⁸Beispielsweise aus [10] oder aus der Schulzeit.

weswegen unsere quadratische Funktion sich viel kompakter in der Form

$$f(x) = x^T Ax + b^T x + c, \quad x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, \quad A = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}, \quad (2.6)$$

schreiben. Das ist tatsächlich weniger schlimm, als es aussieht – schließlich ist ja alles nur Notation. Erst mal eine Bemerkung zu dieser magischen Matrix A : Da ja $x_k x_j = x_j x_k$ ist, können wir die beiden Terme a_{jk} und a_{kj} beliebig zusammenfassen, denn der Faktor vor der Kombination⁶⁹ ist ja $a_{jk} + a_{kj}$, weswegen wir ohne weiteres auch annehmen können, daß $a_{jk} = a_{kj}$ ist, oder, wie wir Mathematiker sagen:

Die Matrix A ist *symmetrisch*.

Symmetrie bedeutet hier, daß man die Matrix A an ihrer Diagonale “spiegeln” oder *transponieren* kann, ohne daß die Matrix sich ändert:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{12} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{n-1,n} \\ a_{1n} & \dots & a_{n-1,n} & a_{nn} \end{bmatrix},$$

siehe auch Abb. 2.11. Nur wenn wir fordern, daß A symmetrisch ist, ist die Matrix A wirklich

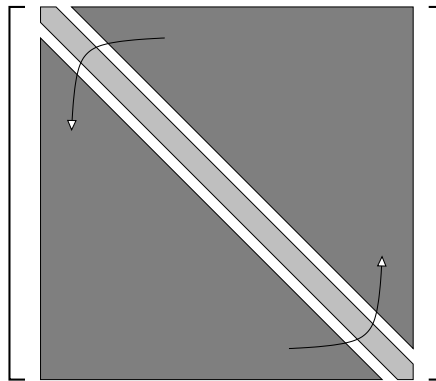


Abbildung 2.11: Schematische Darstellung einer symmetrischen Matrix. Man kann das “obere” oder “rechte” Dreieck entlang der Diagonalen “herunterklappen” (also spiegeln) und erhält so das “untere” bzw. “linke” Dreieck. Und das funktioniert natürlich auch ganz analog “von unten nach oben”.

eine eindeutige Darstellung von f , denn ansonsten könnten wir immer irgendeinen Wert von a_{jk} abziehen, solange wir nur denselben Wert wieder zu a_{kj} hinzuaddieren. Also:

⁶⁹Man beachte, daß hier $j < k$ gefordert wurde und damit die beiden Varianten $x_j x_k$ und $x_k x_j$ zu einem Faktor verschmolzen wurden.

Eine quadratische Funktion hat die Form $f(x) = x^T Ax + b^T x + c$, wobei A eine *symmetrische* Matrix ist.

Man könnte versucht sein, hier schon von einem Paraboloid zu sprechen, aber leider gehören Paraboloid nur zu *speziellen* Matrizen A . Doch bevor wir weitermachen, überprüfen wir zuerst wieder, ob die Verallgemeinerung auch wieder den Fall $n = 1$ einschließt. Und tatsächlich ist eine 1×1 Matrix A nichts anderes als eine Zahl und somit natürlich symmetrisch und es ist $x^T Ax = Ax^2$, denn bei der Multiplikation von Zahlen⁷⁰ darf man die Reihenfolge beliebig vertauschen und durcheinanderwerfen.

Im eindimensionalen Fall war ja der Koeffizient A einer quadratischen Funktion nichts anderes als deren zweite Ableitung, na ja, nicht ganz, sondern nur bis auf den Faktor $\frac{1}{2}$, denn schließlich ist ja die zweite Ableitung von ax^2 nichts anderes als $2a$. Was ist nun die zweite Ableitung einer Funktion in n Variablen und warum ist diese eine Matrix? Sie ist deswegen eine Matrix, und zwar eine *symmetrische* weil sie den “quadratischen” Koeffizienten der quadratischen Funktion beschreibt, die die Funktion f an der vorgegebenen Stelle optimal annähert. Man kann aber auch auf rein formalem Weg zu einer Matrix kommen. Dazu erinnern wir uns, daß die “Ableitung” einer n -dimensionalen Funktion (also deren optimale Linearisierung) der *Gradient* ist, also die *vektorwertige*⁷¹ Funktion

$$\nabla f(x) = \begin{bmatrix} \frac{\partial}{\partial x_1} f(x) \\ \vdots \\ \frac{\partial}{\partial x_n} f(x) \end{bmatrix}.$$

Und von der müssen wir, als zweite Ableitung nochmal den Gradienten bilden und dabei können wir entweder die partiellen Ableitungen des Gradienten als ewiglangen Vektor übereinanderschreiben oder aber diese übersichtlich nebeneinander in einem quadratischen Schema⁷² anordnen, das heißt, wir erhalten entweder

$$\nabla^2 f(x) = \begin{bmatrix} \frac{\partial}{\partial x_1} \nabla f(x) \\ \vdots \\ \frac{\partial}{\partial x_n} \nabla f(x) \end{bmatrix}$$

oder

$$\nabla^2 f(x) = \begin{bmatrix} \frac{\partial}{\partial x_1} \nabla f(x) & \dots & \frac{\partial}{\partial x_n} \nabla f(x) \end{bmatrix} = \begin{bmatrix} \frac{\partial^2}{\partial x_1^2} & \dots & \frac{\partial^2}{\partial x_1 \partial x_n} f(x) \\ \vdots & \ddots & \vdots \\ \frac{\partial^2}{\partial x_1 \partial x_n} f(x) & \dots & \frac{\partial^2}{\partial x_n^2} \end{bmatrix}$$

⁷⁰Nicht aber von Matrizen!

⁷¹Viele Leute sind erst einmal schockiert vom Konzept der vektorwertigen Funktionen oder *Vektorfelder* zumal die physikalischen Beispiele, die oft und gerne gebracht werden (Magnetfelder) auch nicht so ganz ohne sind. Dabei sind wir von vektorwertigen Funktionen umgeben: Fährt man beispielsweise im Auto, so zeigt das Armaturenbrett (je nach Fabrikat natürlich) zu jedem Zeitpunkt Geschwindigkeit, Drehzahl, gefahrene Kilometer, Uhrzeit, Außentemperatur und die diversen aktuellen Fehlfunktionen des Fahrzeugs an, also deutlich mehr als einen Wert.

⁷²Hat da jemand *Matrix* gerufen?

und diese Matrix ist symmetrisch, da die partiellen Ableitungen *kommutieren*, d.h.,

$$\frac{\partial^2}{\partial x_j \partial x_k} = \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_k} = \frac{\partial}{\partial x_k} \frac{\partial}{\partial x_j} = \frac{\partial^2}{\partial x_k \partial x_j}$$

ist⁷³ Damit haben wir aber unser Konzept der zweiten Ableitung gefunden!

Beispiel 2.8 Als nächsten Realitätscheck sehen wir uns mal an, was die zweite Ableitung der quadratischen Funktion

$$x^T A x = \sum_{j=1}^n \sum_{k=1}^k a_{jk} x_j x_k$$

ist. Bei der Bestimmung der partiellen Ableitung nach x_ℓ ist alles irrelevant bis auf die drei Terme

$$a_{\ell\ell} x_\ell^2, \quad a_{\ell k} x_\ell x_k, \quad a_{j\ell} x_j x_\ell$$

und die partielle Ableitung ist die Summe dieser Terme über alle j und k , also, wegen der Symmetrie

$$2a_{\ell\ell} x_\ell + \sum_{k \neq \ell} a_{\ell k} x_k + \sum_{j \neq \ell} a_{j\ell} x_j = 2a_{\ell\ell} x_\ell + 2 \sum_{j \neq \ell} a_{j\ell} x_j = 2 \sum_{j=1}^n a_{j\ell} x_j.$$

Leiten wir das nun nach x_m ab, dann erhalten wir nur noch $a_{m\ell} = a_{\ell m}$, wieder unter Ausnutzung der Symmetrie. Also: Für $f(x) = x^T A x$ ist

$$\nabla^2 f(x) = \left[\frac{\partial}{\partial x_j \partial x_k} f(x) : j, k = 1, \dots, n \right] = 2 [a_{jk} : j, k = 1, \dots, n] = 2 A. \quad (2.7)$$

Definition 2.9 Die Matrix $\nabla^2 f$ mit den zweiten partiellen Ableitungen in (2.7) bezeichnet man auch als Hesse-Matrix von A .

Übung 2.3 (Für Leute, die sich mal mathematisch betätigen möchten)

Zeigen Sie, daß für $q(x) = x^T A x$ die Identität $\nabla q(x) = A x$ gilt. \diamond

Und kaum haben wir eine zweite Ableitung, also $\nabla^2 f$, können wir auch schon mutig werden und anfangen, von *Krümmung* zu reden, insbesondere von positiver und negativer Krümmung.

Aber: Was bitte ist eine *positive Matrix*? Vielleicht eine Matrix, in der alle Einträge > 0 sind? Die Antwort ist "Ja". Oder was ganz was anderes? Die Antwort ist ebenfalls "Ja". Tatsächlich hängt es sehr stark vom Kontext ab, was man unter einer positiven Matrix versteht; in der Stochastik bezeichnet man damit oftmals Matrizen mit positiven Einträgen (normalerweise Wahrscheinlichkeiten), im Kontext der Optimierung aber oftmals etwas anderes. Und was anderes, das werden wir gleich sehen.

⁷³OK, um der Wahrheit die Ehre zu geben: Sie tun's nicht immer, aber zumindest lassen sich zweite Ableitungen für *zweimal stetig differenzierbare* Funktionen vertauschen und alle Funktionen, die diese Voraussetzungen nicht erfüllen sind sowieso Teufelswerk.

Dazu betrachten wir zu einer Matrix A nochmal die quadratische Funktion oder *quadratische Form*⁷⁴ $f(x) = x^T A x$ und sagen, daß A

- *positiv* oder *positiv definit* ist, wenn für alle $x \neq 0$ die Ungleichung $f(x) > 0$ gilt,
- *negativ* oder *negativ definit* ist, wenn für alle $x \neq 0$ stets $f(x) < 0$ ist,
- *indefinit*, wenn es von Null verschiedene x -Werte gibt, an denen $f(x) > 0$ ist, aber auch solche, an denen $f(x) < 0$ gilt.

Beispiel 2.10 *Im Fall $n = 1$ sind positive und negative Definitheit natürlich nichts anderes als Positivität bzw. Negativität der Zahl A , denn Ax^2 ist für alle $x \neq 0$ genau dann positiv, wenn $A > 0$ ist und genau dann negativ, wenn $A < 0$ ist. Der indefinite Fall entspricht dann gerade der "Entartung" $A = 0$.*

Eine positive Matrix muß nun nicht unbedingt positive Einträge haben! Ein ganz einfaches Beispiel ist

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

Trotzdem kann man feststellen, ob eine Matrix positiv, negativ oder auch nicht ist, indem man ihre *Eigenwerte* betrachtet. Was ein Eigenwert einer Matrix ist braucht uns eigentlich nicht zu interessieren⁷⁵, denn wir können solche Eigenwerte relativ komfortabel und stabil⁷⁶ vom Computer berechnen lassen und zwar mit Matlab bzw. Octave durch den Befehl `eig(A)`.

Beispiel 2.11 *Sehen wir uns doch einmal an, wie das für unsere Matrix A von oben aussieht und starten wir unser "virtuelles Octave":*

```
octave> A = [ 2 -1; -1 2 ]
```

```
A =
```

```
  2  -1
 -1  2
```

```
octave> eig(A)
```

⁷⁴Es ist faszinierend, daß die Theorie der quadratischen Formen wesentlich älter ist als die Theorie der Matrizen und daß es einiges an Überlegung und mathematischer Entwicklung brauchte, um die Gemeinsamkeiten und Unterschiede zwischen den beiden Konzepten zu ergründen.

⁷⁵Aber eine Fußnote sollte es uns schon wert sein: Ein *Eigenwert* einer Matrix A ist eine Zahl λ , zu der es einen Vektor $x \neq 0$, den sogenannten *Eigenvektor* gibt, so daß $Ax = \lambda x$ ist. Was also Multiplikation mit der Matrix angeht, "... kümmern sich Eigenvektoren um ihren eigenen Kram ..." – "... they mind their own business ...", wie es in [6] so schön heißt. Kennt man die Eigenwerte und Eigenvektoren einer Matrix, dann hat man sie im wesentlichen im Griff und kann ihr Verhalten sehr viel einfacher beschreiben, deswegen ist die numerische Bestimmung von Eigenwerten auch eine der "Königsdisciplinen der numerischen linearen Algebra, also der numerischen Matrizenrechnung.

⁷⁶Die unangenehme Überraschung, daß Matrizen mit reellen Einträgen plötzlich mit komplexen Eigenwerten um sich werfen bleibt uns bei unseren *symmetrischen* Matrizen glücklicherweise erspart.

ans =

1
3

Die Eigenwerte dieser Matrix sind also 1 und 3.

Satz 2.12 Eine symmetrische Matrix ist

- positiv definit, wenn alle Eigenwerte positiv sind,
- negativ definit, wenn alle Eigenwerte negativ sind,
- indefinit, wenn es mindestens einen positiven und mindestens einen negativen Eigenwert gibt.

Zusammen mit einem verschwindenden Gradienten beschreiben positive Hesse-Matrizen lokale Minima, negative Hesse-Matrizen lokale Maxima und indefinite Matrizen Sattelpunkte.

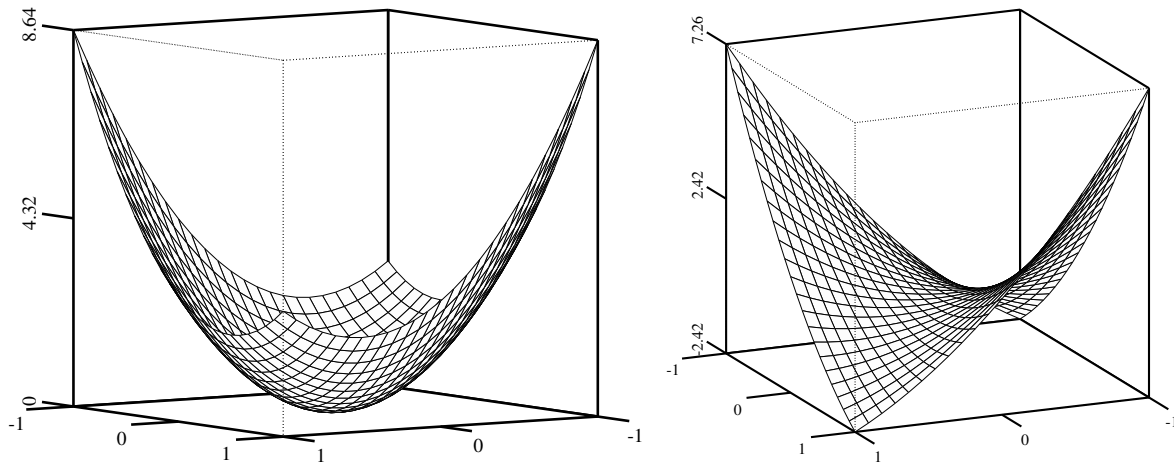


Abbildung 2.12: Zwei Beispiele für quadratische Funktionen.

Links: Das Paraboloid zur Matrix $\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$, die die Eigenwerte 1, 3 hat. Man sieht deutlich das Minimum. Wegen der unterschiedlichen Eigenwerte ist der Anstieg übrigens *nicht* in alle Richtungen gleich stark.

Mitte: Die quadratische Funktion, die zur Matrix $\begin{bmatrix} 1 & -2 \\ -2 & 1 \end{bmatrix}$ mit den Eigenwerten $-1, 3$ gehört. Diese Konfiguration liefert einen Sattelpunkt.

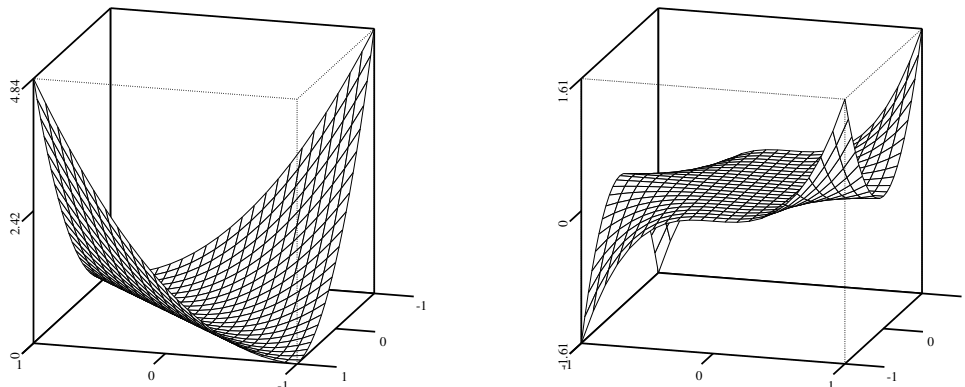


Abbildung 2.13: Noch zwei Beispiele, diesmal wie Nulleigenwerte Verwirrung stiften können.

Links: Die positiv *semidefinite* Matrix $\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$ mit den Eigenwerten 0 und 2 liefert eine Art “Parabeln, die an einer Schnur aufgreift sind”. Wir haben hier unendlich viele Maxima, nämlich alle Punkte entlang der Richtung $[-1, 1]^T$.

Rechts: Hier haben wir es mit der Funktion $f(x, y) = x^3y^2$ zu tun, deren Hessmatrix an 0 den einfachen Wert $\begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix}$ hat und damit die Eigenwerte 0 und 2. Und obwohl die Matrix *nichtnegativ*⁷⁷ ist, haben wir **kein** lokales Extremum an der Stelle 0.

Wir haben in diesem Satz wieder *Positivität* gefordert, was oftmals auch als *strikte positive/negative Definitheit* bezeichnet wird, denn nur dann bekommt man “richtige” Maxima und Minima bzw. “richtige” Sattelpunkte, bei denen in einer Richtung ein Anstieg, in der anderen ein Abstieg vorliegt, siehe Abb. 2.12.

Ein Eigenwert Null bei einer Hesse-Matrix ist in der Tat nochmals ein Sonderfall! Er bedeutet nämlich, in Analogie zum Fall $f''(x) = 0$, daß es mindestens eine Richtung gibt, in der über das Verhalten der Funktion allein mit Hilfe der Ableitungsinformationen **keine** Aussage machen kann. Ist unser f nur eine quadratische Funktion, dann geht das noch, denn dann erstreckt sich das Minimum halt entlang einer ganzen Gerade, aber bei Funktionen höherer Ordnung kann eigentlich alles passieren, von Minima über Maxima und Sattelpunkte bis hin zu “lokalen Plateaus”, die den “Wendepunkten” der Funktion $f(x) = x^3$ entsprechen. Zwei Beispiele für dieses Phänomen sind in Abb. 2.13 dargestellt.

Bemerkung 2.13 (Nur für Mathematik-Interessierte) *Natürlich stellt sich nun die Frage, ob es überhaupt eine Möglichkeit gibt, mit Informationen über die Funktion f an der Stelle x herauszufinden, ob an dieser Stelle ein Extremum vorliegt. Nun, was wir schon wissen, ist, daß $f'(x) = 0$ sein muß. Wäre nun $f''(x)$ von Null verschieden, dann ist die Frage auch beantwortet,*

andernfalls ist die Antwort vertagt und wir sehen uns dann halt die dritte Ableitung $f'''(x)$ an. Hier läuft es nun wieder genau umgekehrt zu f'' :

- Ist $f'''(x) \neq 0$, dann ist x keine Extremalstelle,
- ist $f'''(x) = 0$, dann hat x zumindest das Potential, eine Extremalstelle zu sein.

Und wer entscheidet in letzterem Fall? Richtig, die vierte Ableitung. Und wenn die wieder verschwinden sollte, dann sieht man sich eben das Paar $f^{(5)}$ und $f^{(6)}$ an und so weiter. Wie man auf sowas kommt? Man schaut sich die Taylorformel (2.5) genau an.

2.5 Randextrema

Jetzt wo wir unsere Extrema so halbwegs bestimmen und identifizieren können, sind wir fast so weit, daß wir uns an *Verfahren* machen können, mit deren Hilfe wir diese Extrema auch wirklich so halbwegs lokalisieren können. In der Tat erinnern wir uns an die "goldene Regel"

Die Extrema finden wir, indem wir alle *inneren* Punkte x mit $\nabla f(x) = 0$ und alle Randpunkte untersuchen.

Das mit den Randpunkten war in einer Variablen ziemlich banal: Der zulässige Bereich war einfach ein Intervall, das hatte zwei Endpunkte und diese Werte hat man sich dann halt noch zusätzlich angesehen. Nur ist das in zwei und mehr Variablen nicht so einfach, denn da kann der Rand eine Kurve oder schlimmeres sein und aus *unendlich vielen* Punkten bestehen, so daß man da nochmals ein separates Optimierungsproblem lösen muß.

Beispiel 2.14 (Randextrema sind nicht so einfach) Wir wollen die Funktion

$$F(x, y) = x^3 + x^2 + y^2 - 1$$

auf dem Einheitskreis

$$\mathbb{D} = \{(x, y) : x^2 + y^2 \leq 1\}$$

maximieren. Dazu betrachten wir alle Stellen mit verschwindendem Gradienten, also alle Punkte, die

$$0 = \begin{bmatrix} \frac{\partial F}{\partial x}(x, y) \\ \frac{\partial F}{\partial y}(x, y) \end{bmatrix} = \begin{bmatrix} 3x^2 + 2x \\ 2y \end{bmatrix} \implies (x, y) = \begin{cases} (0, 0) \\ (-\frac{2}{3}, 0) \end{cases}$$

erfüllen. Beide Punkte liegen im Inneren des Einheitskreises, also befragen wir unser Hesse-Orakel, indem wir

$$\begin{bmatrix} \frac{\partial^2 F}{\partial^2 x}(x, y) & \frac{\partial^2 F}{\partial x \partial y}(x, y) \\ \frac{\partial^2 F}{\partial x \partial y}(x, y) & \frac{\partial^2 F}{\partial^2 y}(x, y) \end{bmatrix} = \begin{bmatrix} 6x + 2 & 0 \\ 0 & 2y \end{bmatrix}$$

bestimmen, was uns sagt, daß an $(0, 0)$ ein Minimum und an $(-\frac{2}{3}, 0)$ ein unentscheidbarer Punkt⁷⁸ vorliegt. Aber jetzt brauchen wir noch das Randextremum, wir müssen also $F(x, y)$

⁷⁸Es wird ein Wendepunkt sein.

unter der Nebenbedingung $x^2 + y^2 = 1$ minimieren. Na gut, das Beispiel ist so gewählt, daß es klappt. Setzen wir nämlich die Nebenbedingung in die Gleichung von F ein und parametrisieren wir die Variablen als $x = \sin \varphi$ und $y = \cos \varphi$, dann ist

$$F(x, y) = x^3 + \underbrace{x^2 + y^2}_{=1} - 1 = x^3 = (\sin \varphi)^3,$$

dann erhalten wir eine Funktion in einer Variablen und deren Extrema bekommen wir durch⁷⁹

$$0 = \frac{d}{d\varphi} (\sin \varphi)^3 = 3 (\sin \varphi)^2 \cos \varphi \quad \Longrightarrow \quad \varphi = 0, \frac{1}{2}\pi, \pi, \frac{3}{2}\pi,$$

was den Punkten $(\pm 1, 0)$ und $(0, \pm 1)$ entspricht. Wir haben also insgesamt sechs Kandidaten für alle Extrema, Minima wie Maxima, und brauchen jetzt nur noch nachzusehen, welchen Wert f an diesen Punkten hat:

(x, y)	$(0, 0)$	$(-\frac{2}{3}, 0)$	$(-1, 0)$	$(1, 0)$	$(0, -1)$	$(0, 1)$
$F(x, y)$	-1	$\frac{4}{27}$	-1	1	0	0

und siehe da:

- Wir haben ein “globales” Maximum an $(1, 0)$
- und ein lokales Minimum an $(0, 0)$ sowie ein Randminimum an $(-1, 0)$.
- Der kritische Punkt⁸⁰ $(-\frac{2}{3}, 0)$ ist für die Maximumsuche völlig irrelevant.

Und in der Tat unterstützt die Anschauung auch das theoretische Resultat, siehe Abb. 2.14.

Man kann nun solche *Extrema unter Nebenbedingungen* auch tatsächlich theoretisch untersuchen, was zum Konzept der *Lagrange-Multiplikatoren* bzw. den sogenannten *Kuhn-Tucker-Bedingungen* führt. Ersteres ist die Sprechweise der Analysis, letzteres der Optimierer. Diese Bedingungen haben noch eine relativ einfache Interpretation bezüglich der Ableitungen: Wir erinnern uns, daß an einem *inneren* Punkt die Richtungsableitung in *alle* Richtungen an einem Minimum ≥ 0 sein mußte – und da wir von einem inneren Punkt aus auch in alle Richtungen gehen können, mußte der Gradient verschwinden. An einem “gutartigen”⁸¹ Randpunkt hingegen muß die Ableitung nichtnegativ sein für alle Richtungen, die *in den zulässigen Bereich hineinzeigen!* Und wegen der Lokalität der ganzen Geschichte reicht es für diese “hineinzeigenden”, zulässigen Richtungen, daß man ein beliebig kurzes Stückchen in diese Richtung gehen kann – das führt dazu, daß dieser sogenannte *Richtungskegel* im “Normalfall”⁸² der “innenliegende” Halbraum ist, der durch die Tangente abgeschnitten wird.

⁷⁹Entweder man erinnert sich an die Kettenregel oder verwendet ein Computeralgebra-Programm wie MuPAD, siehe [15].

⁸⁰So nennt man Stellen, an denen der Gradient verschwindet.

⁸¹Was das ist, das führt nun wirklich zu weit, aber ein Beispiel für einen nicht-gutartigen Punkt ist der Endpunkt der Spirale aus Abb. 2.4.

⁸²Das heißt dann, wenn der Randpunkt keine Ecke ist.

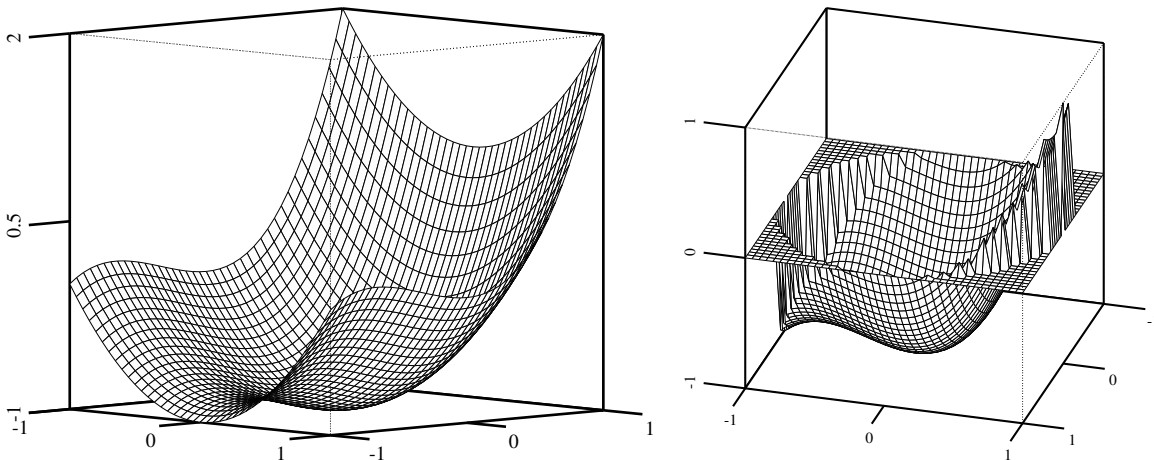


Abbildung 2.14: Die Funktion $F(x, y)$ aus Beispiel 2.14, einmal auf dem Quadrat (links) und einmal auf dem Einheitskreis (rechts), wobei außerhalb des Kreises die Funktion einfach auf Null gesetzt wurde. Man sieht recht gut die lokalen und globalen Extrema. .

Wir wollen uns damit aber **nicht** weiter auseinandersetzen⁸³, sondern uns einen einfachen Trick ansehen, wie wir Nebenbedingungen ganz einfach loswerden können, nämlich als *Straf-* oder *Barriereterme*. Nehmen wir an, wir wollen das Optimierungsproblem

$$\max F(x), \quad G(x) = \begin{bmatrix} g_1(x) \\ \vdots \\ g_m(x) \end{bmatrix} \geq 0, \quad x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

lösen, wobei die Nebenbedingungen *implizit* durch den Vektor $G(x)$ gegeben sind.

Übung 2.4 Formulieren Sie den Nebenbedingungsvektor für den Einheitskreis

$$\{(x, y) : x^2 + y^2 = 1\}$$

und die *Kreisscheibe*

$$\{(x, y) : x^2 + y^2 \leq 1\}.$$

◇

Der Trick ist nun ganz einfach: Wir schlagen die Nebenbedingungen einfach zur Zielfunktion, betrachten also beispielsweise zu einem Parameter⁸⁴ $\alpha > 0$

$$F(x) + \alpha (g_1^-(x) + \cdots + g_m^-(x)), \quad \alpha > 0, \quad (2.8)$$

⁸³Das führt wirklich zu weit und ist auch nicht in einer netten Fußnote unterzubringen.

⁸⁴Dieser Parameter regelt die Gewichtung zwischen Zielfunktion und Strafe.

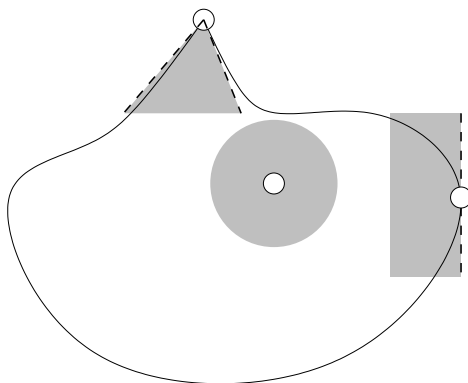


Abbildung 2.15: Drei Punkte und die zugehörigen, nach innen zeigenden Richtungen. Bei einem inneren Punkt sind es alle Richtungen (das ist der Kreis), bei einem Randpunkt mit “glattem” Rand ist der Richtungsbereich durch die Tangente festgelegt, bei einem Eckpunkt ist es der “Öffnungsbereich” der Ecke.

bzw.⁸⁵

$$F(x) + \alpha \left[(g_1^-(x))^2 + \cdots + (g_m^-(x))^2 \right], \quad \alpha > 0, \quad (2.9)$$

wobei

$$f^- = \begin{cases} -f(x), & f(x) < 0, \\ 0, & f(x) \geq 0. \end{cases}$$

Wenn nun Nebenbedingungen verletzt sind, wenn also das eine oder andere $g_j(x) < 0$ ist, dann wird dieser Wert abgezogen und macht so, für hinreichend großes α jedes Maximum außerhalb des zulässigen Bereichs kaputt. Eine *Barriere* ist ein Strafterm, der am Rand den Wert $-\infty$ annimmt, über den die Funktion also nicht “wegklettern” kann. Ein typisches Beispiel ist

$$F(x) + \alpha (\log g_1(x) + \cdots + \log g_m(x)), \quad \alpha > 0$$

was zum Rand hin den Wert $-\infty$ annimmt. Das Gute an der Sache ist, daß die Logarithmusfunktion bis auf ihr Verhalten an 0 eigentlich eine ganz brave Funktion ist und so verhältnismäßig wenig Schwierigkeiten macht. Um auch Randextrema zu erwischen, läßt man das Optimierungsverfahren für verschiedene Werte von α laufen, die immer man immer kleiner werden läßt. Mit etwas Glück erwischt man dann so alle Extrema einer Funktion.

Und hier nochmal für’s Poesiealbum:

Man kann ein Optimierungsproblem mit Nebenbedingungen durch Einführung *geeigneter* Straf- oder Barriereterme immer in ein nebenbedingungsfreies Optimierungsproblem umwandeln.

⁸⁵Der Vorteil der Quadrate in (2.9) liegt darin, daß für differenzierbares f die Funktion f^- nur stetig ist, die Funktion $(f^-)^2$ hingegen differenzierbar und daß man somit Verfahren anwenden kann, die erste Ableitungen verwenden. Und das tun eigentlich alle Abstiegsverfahren, die wir in Kapitel 3 diskutieren werden.

That's the reason they're called lessons, [. . .] because they lessen from day to day.

L. Carroll, *Alice's adventures in wonderland*

Abstiegsverfahren

3

Nach dem langen und zugegebenermaßen abstrakten Theoriekapitel über die Natur von Minima und Maxima können wir uns jetzt relativ unproblematisch an die Konstruktion von “allgemeinen” Optimierungsverfahren machen. Diese Verfahren werden alle eines gemeinsam haben: Sie werden, ausgehend von einem *Startwert* $x^{(0)}$ eine Folge $x^{(k)}$ von Punkten berechnen, die hoffentlich gegen ein Extremum wandern, wobei wir uns hier auf ein *Minimum* festlegen wollen.

3.1 Grundidee aller Verfahren

Alle hier vorgestellten Verfahren gehen von einer sehr einfachen und wirklich intuitiven Idee aus, nämlich der Verwendung *lokaler Information*: Es ist, als ob jemand, der an einem Punkt der Funktion steht, zu einem tiefergelegenen Punkt kommen möchte, aber die Funktion nur in einem unendlich kleinen Bereich einsehen kann⁸⁶. Wie in Abb. 3.1 macht man dann von der



Abbildung 3.1: Idee der Abstiegsverfahren in “Flatland”: Aus der *lokalen* Information “Ableitung” wird bestimmt, in welche Richtung es “bergab” geht und in diese Richtung wird dann weitergegangen.

Ableitung Gebrauch und geht in eine Richtung weiter, in der die optimale lineare Näherung, also die Tangente, abfällt. Etwas formaler: Wir werden nach dem Schema

$$x^{(k+1)} = x^{(k)} + \alpha y, \quad k = 0, 1, 2, \dots$$

vorgehen und in jedem Schritt eine

Abstiegsrichtung $y = y^{(k)}$, in der die optimale lineare Annäherung⁸⁷ der Funktion F abfällt und eine

Schrittweite $\alpha = \alpha_k$, die so gewählt werden sollte, daß $F(x^{(k+1)}) < F(x^{(k)})$ ist

bestimmen. Damit bekommen wir eine Folge von Punkten $x^{(k)}$ zu immer kleineren Funktionswerten

$$F(x^{(0)}) > F(x^{(1)}) > F(x^{(2)}) > \dots > F(x^{(k)}),$$

die hoffentlich irgendwann bei einem Minimum landen werden – vorausgesetzt, die Funktion besitzt so etwas überhaupt. Daß diese Vorgehensweise zwar einfach und einleuchtend klingt,

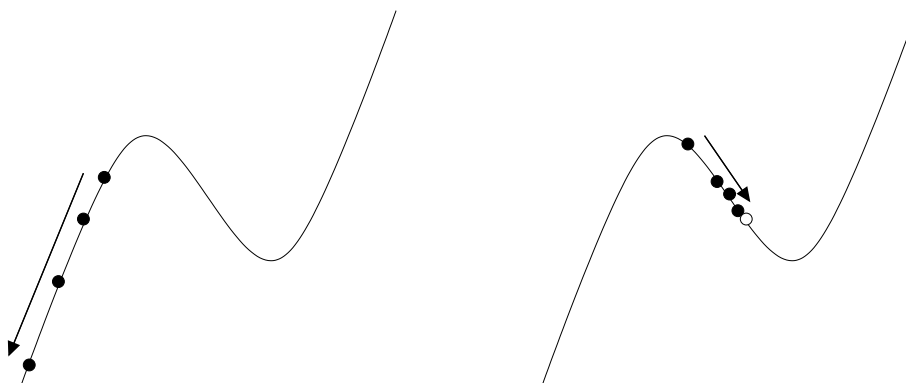


Abbildung 3.2: Zwei Dinge, die bei Abstiegsverfahren schiefgehen können.

Links: Wir steigen in einer Richtung ab, in der die Funktion **kein** Minimum besitzt. Selbst wenn man mit jedem Schritt einiges an Verkleinerung gewinnt, bringt das gar nichts, denn das Minimum ist $-\infty$.

Rechts: Der Abstieg erfolgt in Richtung eines Minimums, aber die Schrittweiten werden immer kleiner, so daß das Verfahren nicht am Minimum, sondern vorher “stationär” wird, also einfach nicht mehr weiterkommt.

es aber bei weitem nicht ist, zeigt Abb. 3.2. Ein Minimum finden wir nur, wenn es sowas auch gibt und wenn die Schrittweiten nicht dergestalt zu klein werden, daß der Prozess vor dem Minimum effektiv zum Halten kommt.

⁸⁶Zum Beispiel weil starker Nebel herrscht.

⁸⁷Mit anderen Worten: die lineare Funktion, die die Ableitung definiert bzw. die Tangente an der Stelle $x^{(0)}$.

Übung 3.1 Wie stellt man fest, daß man sich vor einem Minimum “festgelaufen” hat? \diamond

Die nächste Frage ist, wann die Iterationen beendet werden sollen. Hier gibt es mehrere Möglichkeiten:

- Wenn die Punkte sich nicht mehr signifikant verändern, also $x^{(k+1)} - x^{(k)}$ “längenmäßig” klein wird.
- Wenn sich die Werte nicht mehr signifikant, also wenn $|F(x^{(k+1)}) - F(x^{(k)})|$ klein wird.
- Wenn man an einem “Minimumskandidaten” mit “waagerechter Tangente” angekommen ist, also wenn $|F'(x^{(k)})|$ klein bzw. $\nabla F(x^{(k)})$ “praktisch” der Nullvektor ist⁸⁸.

Ein weiterer fundamentaler Aspekt von Abstiegsverfahren, über den man sich im klaren sein muß, ist, daß sie nur imstande sind *lokale* Minima zu finden: Um von einem lokalen Minimum zum nächsten zu kommen, muß man immer erst einmal einen “Berg” überqueren.

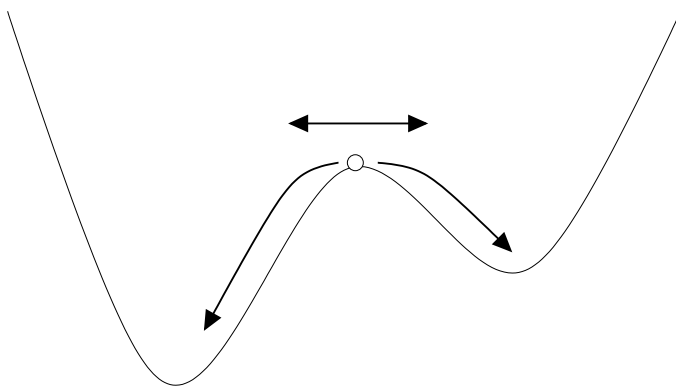


Abbildung 3.3: Wenn man in dem lokalen Maximum startet, gibt es gar keinen Abstieg, verschiebt man den Startpunkt beliebig wenig nach links, so landet ein Abstiegsverfahren beim linken lokalen Minimum, verschiebt man ihn etwas nach rechts, so erreicht man entsprechend das rechte lokale Minimum.

Und selbst bei welchem lokalen Extremum man landet, das kann auf ziemlich subtile Art und Weise vom Startpunkt abhängen wie Abb. 3.3 zeigt.

3.2 Abstiegsrichtungen

Für $n = 1$ ist die Sache mit der Abstiegsrichtung ja ziemlich einfach: Ist die Ableitung $F'(x^{(k)})$ an der Stelle $x^{(k)}$ positiv, dann geht es in positiver Richtung mit $x^{(k+1)} > x^{(k)}$ weiter, ist sie

⁸⁸Dies zu entscheiden ist im übrigen gar nicht so einfach.

negativ, dann muß man die negative Richtung, also $x^{(k+1)} < x^{(k)}$ wählen, und verschwindet die Ableitung, dann hat man einen Minimumskandidaten.

Übung 3.2 Kann es sein, daß an der Stelle $x^{(k)}$ ein Maximum vorliegt? Die Antwort ist “ja” und hat was mit schlechten Schrittweiten zu tun! \diamond

In zwei und mehr Variablen ist die Sache dann schon komplizierter und wir werden sehen, daß es durchaus mehrere gute Abstiegsrichtungen geben kann. Um das zu illustrieren betrachten wir immer das *Modellproblem* einer *quadratischen* Funktion

$$F(x) = \frac{1}{2} x^T A x + b^T x + c, \quad (3.1)$$

an der wir alle Phänomene schon ganz gut sehen können und die einen sehr einfachen Gradienten hat.

Übung 3.3 Zeigen Sie, daß für die quadratische Funktion aus (3.1) den Gradienten $\nabla F(x) = Ax - b$. Wie könnte man also Kandidaten für Extrema dieser Funktion berechnen? \diamond

Außerdem haben quadratische Funktionen den Vorteil, daß wir die optimale oder *exakte* Schrittweite, auch tatsächlich berechnen können. Diese Schrittweite ergibt sich, indem wir für eine beliebige Richtung y die quadratische Funktion

$$f(t) = F(x^{(k)} + t y), \quad t \in \mathbb{R},$$

in der *einen* Variablen t betrachten und dann mit unseren “normalen” Methoden

Ableiten und die Ableitung Null setzen

den Wert t zur Extremalstelle berechnen⁸⁹, was uns die Formel

$$t = t^{(k)} = \frac{(Ax^{(k)} - b)^T y}{y^T A y} \quad (3.2)$$

liefert. Und das ist nun definitiv das Beste, was wir mit der Richtung y erreichen können, denn es ist das globale Extremum der univariaten Parabel $f(t)$ – ob das allerdings ein Maximum oder ein Minimum ist, das hängt wieder ganz von A ab. Diese Vorgehensweise, die man als *exakte Schrittweitensteuerung* bezeichnet, ist natürlich bei “allgemeinen” Funktionen nicht möglich, denn da ist ja die zu minimierende univariate Funktion an sich schon ein recht komplexes Gebilde.

Aber es gibt noch einen weiteren Punkt: Jede hinreichend glatte Funktion läßt sich ja *lokal* bis auf einen Fehlerterm der Größenordnung h^3 (was schon eine ganze Menge ist) durch eine Parabel annähern – das war ja gerade das Wesen der zweiten Ableitung. Damit beschreibt das Modellproblem aber auch die Funktion recht gut, sofern wir A als zweite Ableitung $\nabla^2 F$ von F wählen.

Jetzt werden wir uns ein paar Abstiegsrichtungen und die zugehörigen Probleme ansehen – und zwar im wesentlichen in Form eines Computerexperiments (ohne zu viel Theorie dahinter), die Octave-Programme sind, zum “Selbstexperimentieren” im Skript enthalten.

```

## SteepDesc.m (Optmierung fuer Hoerer aller Fachbereiche)
## -----
## Steilster Abstieg,
## Eingabe:
##   A      Matrix (symmetrisch und besser positiv definit)
##   b      Linearer Term
##   tol    Toleranz

function [x,it] = SteepDesc( A,b,tol )
    n = length( A ); x = zeros( n,1 );      # Startwert x = 0
    it = 0; r = b;                          # Initialisierung

    while ( norm( r ) > tol )
        it = it+1;
        x = x + (r' * r) / (r' * A * r) * r; # Richtungsminimum
        r = b - A*x;                          # - neuer Gradient
    end
%endfunction

```

Programm 3.1 SteepDesc.m: Verfahren des steilsten Abstiegs.

Die erste Wahl einer Abstiegsrichtung ist die eigentlich einleuchtendste: Man wählt die Richtung, in der es am steilsten nach unten geht, und das ist

$$y^{(k)} = -\nabla F(x^{(k)}). \quad (3.3)$$

Das zugehörige Verfahren findet sich in Programm 3.1. Sehen wir uns doch mal an, was da passiert, indem wir uns eine zufällige 3×3 -Matrix holen, diese symmetrisch und positiv definit machen und dann nach dem Minimum suchen lassen:

```

octave> A = rand(3); b = rand( 3,1);
octave> [x,it] = SteepDesc ( A'*A, b, 10^(-8) )
x =

    6.7252
   -1.1921
   -5.6541

it = 618

```

⁸⁹Im Moment ist noch nicht ausgeschlossen, daß diese ein Maximum ist, dann haben wir ein Problem. Aber das können wir ja unserer quadratischen Funktion in einer Variablen ansehen, ob sie eine nach oben oder eine nach unten geöffnete Parabel ist.

Führen wir nun mehrere Experimente durch, so stellen wir ziemlich schnell fest, daß die Anzahl der Iterationsschritte *extrem* stark variieren kann, je nachdem, wie die Matrix A eben gerade so zufällig ausfällt. Das schreit natürlich nach ein bisschen mehr Systematik, wenn wir dieses Phänomen verstehen wollen! Zu diesem Zweck ziehen wir uns auf $n = 1$ zurück und setzen $b = [1, 1]^T$ – mit $b = 0$ würde der Startwert $x = 0$ immer sofort das Minimum liefern, was etwas langweilig wäre⁹⁰. Die Matrix A wählen wir nun als Diagonalmatrix und zwar als

$$A = \begin{bmatrix} 1 & 0 \\ 0 & \alpha \end{bmatrix}, \quad \alpha > 0,$$

dann erhalten wir über den Aufruf

```
octave> a = 1; A = [ 1 0; 0 a ]; b = [ 1;1 ];
octave> [x,it] = SteepDesc ( A, b, 10^(-8) )
```

die folgende Tabelle:

α	10^{-3}	10^{-2}	10^{-1}	1	10	10^2	10^3
Ergebnis	$[1, 10^3]^T$	$[1, 10^2]^T$	$[1, 10]^T$	$[1, 1]^T$	$[1, 10^{-1}]^T$	$[1, 10^{-2}]^T$	$[1, 10^{-3}]^T$
# It.	9384	939	94	1	94	939	9384

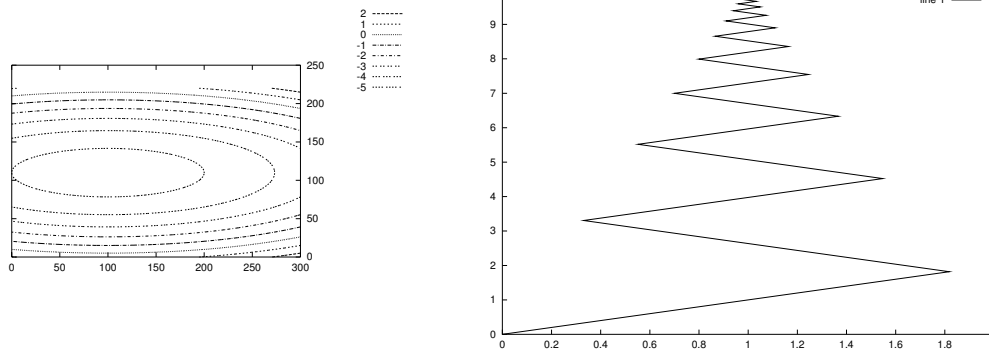


Abbildung 3.4: Der Konturplot der Parabel und das Verhalten des steilsten Abstiegs – dieser führt leider gerade nicht in Richtung des Minimums, sondern fast senkrecht dazu und so verbringt das Verfahren den größten Teil seiner Zeit damit, im Zickzack zu laufen.

Je größer und je kleiner unser α also wird, desto schlechter wird das Verhalten des Abstiegsverfahrens. Und was da passiert, das sieht man eigentlich schon ganz gut für $\alpha = 0.1$, und zwar in Abbildung 3.4. Und das Problem muß irgendetwas damit zu tun haben, daß die Ellipsen

⁹⁰So läuft übrigens schlechtes Testen eines Verfahrens ab: Man startet mit Werten für die das Verfahren gar nicht scheitern kann und wundert sich dann, warum es bei realistischen Daten realistische Probleme gibt. Manchmal ist so ein Zufallszahlengenerator wirklich keine schlechte Sache.

in den ‘‘Hohenlinien’’ mehr und mehr zusammengedruckelt werden, denn je groer bzw. kleiner α wird, desto starker wird dieser Effekt und desto steiler werden die Winkel, in denen im Zickzack gelaufen wird. Und dieses ‘‘Quetschverhaltis’’ kann man angeben: Es ist das Verhaltis ρ zwischen dem kleinsten und dem groten *Eigenwert* der Matrix A . Und tatsachlich kann man zeigen, da

$$\|x^{(k)} - x\| \leq (1 - \rho)^k \|x^{(0)} - x\|$$

gilt, da also die Konvergenz des Verfahrens beliebig *langsam* wird, je kleiner ρ ist.

ubung 3.4 Experimentieren Sie mit dem Verfahren des steilsten Abstiegs und versuchen Sie, die Anzahl der Iterationen aus den Eigenwerten vorherzusagen. \diamond

Das mit den ‘‘gequetschten’’ Ellipsen ist eine Art universelles Prinzip, nur konnen die im Gegensatz zum obigen Beispiel auch noch gedreht sein. Ein Beispiel ist die symmetrische Matrix⁹¹

$$A = \begin{bmatrix} 0.90888 & 0.28679 \\ 0.28679 & 0.10017 \end{bmatrix} \quad (3.4)$$

mit den Eigenwerten, erhaltlich via $\text{eig}(A)$, mit den Werten 0.0087958 und 1.0002582. Die Konturlinien sind in Abb. 3.5 zu sehen.

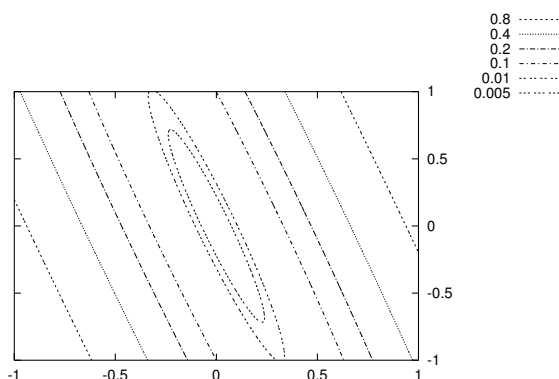


Abbildung 3.5: Hohenlinien der quadratischen Funktion zur Matrix aus (3.4).

Der Quotient 113.72 der Eigenwerte von A lat dann auch nichts gutes ahnen und in der Tat benotigt das Verfahren mit $b = [1, 1]^T$ solide 693 Iterationsschritte. Wie ungunstig der steilste Abstieg tatsachlich ist, sieht man ubrigens bei einem kleinen Experiment: Verwendet

⁹¹Erhalten als symmetrisierte positiv definite Version $A = A' * A$ einer zufalligen Matrix $A = \text{rand}(2)$, daher die ‘‘krummen’’ Zahlen.

```

## SteepRandDesc.m (Optmierung fuer Hoerer aller Fachbereiche)
## -----
## Steilster Abstieg mit zufaellieger Variation
## Eingabe:
##   A      Matrix (symmetrisch und besser positiv definit)
##   b      Linearer Term
##   tol    Toleranz
##   p      Gewichtung des Zufalls

function [x,it] = SteepRandDesc( A,b,tol,p )
    n = length( A ); x = zeros( n,1 );      # Startwert x = 0
    it = 0; r = b;                          # Initialisierung

    while ( norm( r ) > tol )
        it = it+1;
        r = (1-p)*r + p * norm(r) * rand( size(r) );
        x = x + (r' * r) / (r' * A * r) * r; # Richtungsminimum
        r = b - A*x;                          # - neuer Gradient
    end
%endfunction

```

Programm 3.2 `SteepRandDesc.m`: Zufällig modifizierten steilster Abstieg – kommt erstaunlicherweise nicht so leicht ins Schleudern.

man anstelle des negativen Gradienten eine *zufällig* gestörte Richtung der Form

$$y = (1 - \alpha)(-\nabla F(x)) + \alpha \|\nabla F(x)\| y', \quad y' \in \{0, 1\}^n,$$

wobei y' ein zufälliger Vektor ist, dann erhält man (im Mittel⁹²) die folgenden Ergebnisse

α	10^{-5}	10^{-4}	10^{-3}	10^{-2}	10^{-1}
# It.	690	669	450	90	130

Das ideale Ergebnis scheint also ungefähr bei 10^{-2} erreicht zu werden, weniger Zufall “bringt nichts”, mehr Zufall wird zu unorganisiert und läuft nach nirgendwo. Der Zufall selbst sorgt durch die leichte Störung der Abstiegsrichtung dafür, daß es zwar nicht so steil nach unten geht, wie es gehen könnte, dafür kann man aber wesentlich größere Schritte machen und kommt so letztendlich um einiges schneller voran. Für die, die vielleicht mal selbst experimentieren wollen, ist die Routine in Programm 3.2 abgedruckt.

Wenn man bei den Höhenlinien genau hinsieht, dann stellt man fest, daß man durch drehen des Koordinatensystems und geeignete Umskalierung einer Achse (entweder x - oder y -Achse⁹³) die Ellipsen in Kreise umformen kann. Diese Änderung der “Metrik” kann man aber

⁹²Für jeden Wert von α wurden zur Matrix A drei Versuche durchgeführt und “über den Daumen” gemittelt.

⁹³Der Skalierungsmaßstab ist dabei nichts anderes als das Verhältnis der Eigenwerte

auch implizit machen, also ohne die Matrix A zu verändern, was für unser Modellproblem (3.1) die Abstiegsrichtungen

$$p_{k+1} = r - \frac{r^T A p_k}{p_k^T A p_k} p_k, \quad r = b - A x^{(k)} \quad (3.5)$$

und die Iteration

$$x^{(k+1)} = x^{(k)} - \frac{r^T p_{k+1}}{p_{k+1}^T A p_{k+1}} p_{k+1}$$

liefert. Das ist die *Methode der konjugierten Gradienten*, die, wenn man es “mathematisch genau” betrachtet, bereits nach n Schritten das Minimum der quadratischen Funktion gefunden haben sollte, was es aber wegen der allgegenwärtigen Rundungsfehler nicht tut! Erstaunlicherweise braucht man aber nur weiterzuitern, um dann irgendwann doch bei der Lösung anzukommen, die Rundungsfehler werden durch hinreichend viel “Durchhaltevermögen” dann schon ausgemerzt. Tatsächlich haben sich die konjugierten Gradienten sogar als eines der wichtigsten Verfahren zum Lösen linearer Gleichungssysteme $Ax = b$ mit (dünnbesetzten) symmetrischen und positiv definiten Matrizen durchgesetzt⁹⁴.

Nun hat die Bestimmung der konjugierten⁹⁵ Richtungen in (3.5) aber für allgemeine Funktionen F einen ganz gewaltigen Schönheitsfehler: Zur ihrer Bestimmung braucht man (anscheinend) die Hesse-Matrix $A = \nabla^2 F(x^{(k)})$, was einen eigentlich nicht zu rechtfertigenden Aufwand darstellt. Mit ein bißchen Rechnerei kann man das aber dann doch umgehen und erhält die Abstiegsrichtungen

$$y^{(k)} = \nabla F(x^{(k)}) - \frac{\|\nabla F(x^{(k)})\|_2^2}{\|\nabla F(x^{(k-1)})\|_2^2} y^{(k-1)}, \quad (3.6)$$

die nur noch der Gradienten an $x^{(k)}$ und den im vorherigen Schritt berechneten Gradienten an $x^{(k-1)}$ verwendet, also nur noch Information über die *erste* Ableitung.

Aber einen haben wir noch! Dafür erinnern wir uns an die Tatsache, daß sich ein lokales Extremum dadurch auszeichnet, daß an dieser Stelle die Ableitung bzw. der Gradient den Wert Null hat, also verschwindet. Und da fragt man sich natürlich schon, warum man diesen Punkt nicht “einfach” sucht, indem man das normalerweise *nichtlineare* Gleichungssystem

$$F'(x) = 0 \quad \text{bzw.} \quad \nabla F(x) = 0$$

löst. Aus der Numerik, [10] kennen wir hierzu ein, wenn nicht “das” Verfahren, nämlich das *Newton-Verfahren*, das auf der Iteration

$$x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})}$$

⁹⁴Denn das Minimum von $\frac{1}{2}x^T A x - b^T x$ wird ja gerade an der Lösung von $Ax = b$ angenommen, so daß man wahlweise lineare Gleichungssysteme durch Optimieren oder umgekehrt lösen kann.

⁹⁵Ach ja: *konjugiert* ist eine Art von Orthogonalität und bedeutet, daß $p_{k+1}^T A p_k = 0$ sein muß.

basiert. Zum Finden eines Extremums suchen wir aber eine Nullstelle von $f = F'$ und erhalten so eine Iteration der Form

$$x^{(k+1)} = x^{(k)} - \frac{f'(x^{(k)})}{f''(x^{(k)})} = x^{(k)} - [f''(x^{(k)})]^{-1} f'(x^{(k)}). \quad (3.7)$$

Das zweite Gleichheitszeichen in (3.7) steht dabei nur für eine triviale Umformung, bei der man das Symbol $a^{-1} = 1/a$ verwendet hat. Allerdings können wir diese Umformung auch in mehreren Variablen verwenden, indem wir

$$x^{(k+1)} = x^{(k)} - [\nabla^2 F(x^{(k)})]^{-1} \nabla F(x^{(k)}) \quad (3.8)$$

setzen – und siehe da: Es passt alles zusammen. Durch einen glücklichen Zufall⁹⁶ ist ja $\nabla^2 F$ eine $n \times n$ -Matrix und ∇F ein n -Vektor und die können wir angemessen multiplizieren. Das liefert uns dann auch die sogenannte *Newton-Richtung*

$$y^{(k)} = [\nabla^2 F(x^{(k)})]^{-1} \nabla F(x^{(k)}) \quad \text{bzw.} \quad \nabla^2 F(x^{(k)}) y^{(k)} = \nabla F(x^{(k)}). \quad (3.9)$$

Normalerweise ist es numerisch einfacher und besser, keine inversen Matrizen zu berechnen wie auf der linken Seite von (3.9), sondern $y^{(k)}$ durch Lösung des linearen Gleichungssystems auf der rechten Seite zu bestimmen, zumal ja $\nabla^2 F$ auch noch eine *symmetrische* Matrix ist.

Allerdings kommt die Bestimmung der Newton-Richtung nun *nicht* mehr ohne Bestimmung der zweiten Ableitung, also der Hesse-Matrix aus. Man kann das zwar näherungsweise in einem gewissen Rahmen umgehen, das führt dann zum sogenannten *Broyden-Verfahren*, aber wenn man die exakte Newton-Richtung will, dann führt an zweiten Ableitungen leider kein Weg vorbei.

Zusammenfassend haben wir also jetzt drei Möglichkeiten kennengelernt, um Abstiegsrichtungen zu berechnen:

- Steilster Abstieg,
- Konjugierte Gradienten,
- Newton-Richtung.

Mit jeder von diesen Richtungen, die im Normalfall sehr unterschiedlich ausfallen werden, versucht man, dem Minimum nun mit einem Iterationsverfahren der Form

$$x^{(k+1)} = x^{(k)} + \alpha_k y^{(k)}, \quad k = 0, 1, 2, \dots \quad (3.10)$$

auf den Leib zu rücken, wobei $x^{(0)}$ ein “frei gewählter” Startwert ist, um die Wahl der *Schrittweite* α_k werden wir uns gleich noch kümmern. Dieses Verfahren iteriert man nun so lange, bis eine oder mehrere passende Abbruchbedingungen erfüllt sind.

⁹⁶Natürlich gibt es in der Mathematik keine Zufälle (außer in der Wahrscheinlichkeitsrechnung), entweder es paßt oder es paßt nicht.

3.3 Schrittweitensteuerung

Das zentrale Problem dabei besteht darin, die Schrittweite einerseits so groß zu wählen, daß man eine signifikante Verbesserung erhält⁹⁷, andererseits aber klein genug, um nicht “über das Ziel hinauszuschießen”. Natürlich wäre die optimale Wahl immer noch die *exakte* Schrittweitensteuerung, bei der man α_k als die Lösung t^* des *univariate* Optimierungsproblems

$$\min_t F(x^{(k)} + t y^{(k)})$$

wählt. Das aber ist im Normalfall zu schwierig, deswegen sucht man *einfacher* zu ermittelnde Schrittweiten, die aber gewissen Bedingungen zu gehorchen haben. Um diese Bedingungen hinzuschreiben, verzichten wir jetzt auf den Index k – das spart nur Schreibarbeit und macht die Sache lesbar.

Die erste (und wesentliche) Bedingung an die Schrittweite α ist die sogenannte *Armijo-Bedingung*, die fordert, daß

$$F(x + \alpha y) - F(x) \leq C_1 \alpha y^T \nabla F(x) = C_1 \alpha D_y F(x), \quad (3.11)$$

wobei $0 < C_1 < 1$ eine passende Konstante ist⁹⁸. Wenn wir uns daran erinnern, daß ja y eine Abstiegsrichtung ist, daß also $0 > y^T \nabla F(x) = D_y F(x)$ gilt, dann besagt die Forderung (3.11), daß α so zu wählen ist, daß der Abstieg *proportional* zu dem ist, was die Richtungsableitung vorhersagt, siehe Abb. 3.6.

Nun kann es Schrittweiten geben, die die Bedingung (3.11) erfüllen, aber die zu einem Punkt $x' = x + \alpha y$ führen, an denen es noch steiler in Richtung y bergab geht, als an x selbst, d.h. es wäre $D_y F(x') < D_y F(x)$. So ein Punkt ist natürlich noch nicht der Weisheit letzter Schluß, denn von einem solchen Punkt aus kann man ja nochmals ein Stückchen nach rechts gehen. Deshalb fordert man bei den *Wolfe-* oder *Powell-Bedingungen*⁹⁹ zusätzlich zu (3.11), daß

$$y^T \nabla F(x + \alpha y) \geq C_2 y^T \nabla F(x), \quad C_1 < C_2 < 1, \quad (3.12)$$

erfüllt sein soll, daß es also an x' *weniger steil* nach unten geht als an x – oder aber nach oben, das ist immer erlaubt. Wie das bei unserem Beispiel aus Abb. 3.6 aussieht, das zeigt Abb. 3.7. Und schließlich gibt es noch die sogenannten *starken Wolfe-Bedingungen*, bei denen auch zu starker Anstieg verboten wird, und bei denen (3.12) durch die Bedingung

$$|y^T \nabla F(x + \alpha y)| \leq C_2 |y^T \nabla F(x)| \quad (3.13)$$

ersetzt werden – das Bild dazu schenken wir uns, es würde einfach nur im “mittleren” Bereich von Abb. 3.7 ein weiteres Stückchen herausfallen.

Jetzt haben wir also drei Typen von Forderungen an die Schrittweiten, aber was uns natürlich aus praktischer Sicht interessiert, sind die folgenden Fragen:

⁹⁷Man kann beweisen, daß man für jede Abstiegsrichtung, genauer, für jede Richtung, die nicht auf den Gradienten senkrecht steht, eine Verbesserung erhält, sofern die Schrittweite (im Absolutbetrag) nur *klein* genug gewählt wird. Daß diese Verbesserung dann auch nur sehr marginal ausfallen wird, das ist halt die Kehrseite der Medaille.

⁹⁸Diese Konstante kann man frei wählen und sie bestimmt, wie pingelig ($C_1 \sim 1$) oder großzügig ($C_1 \sim 0$) die Schrittweitenauswahl sein soll.

⁹⁹Die Namensgebung ist je nach Buch, also je nach “politischer” Zugehörigkeit.

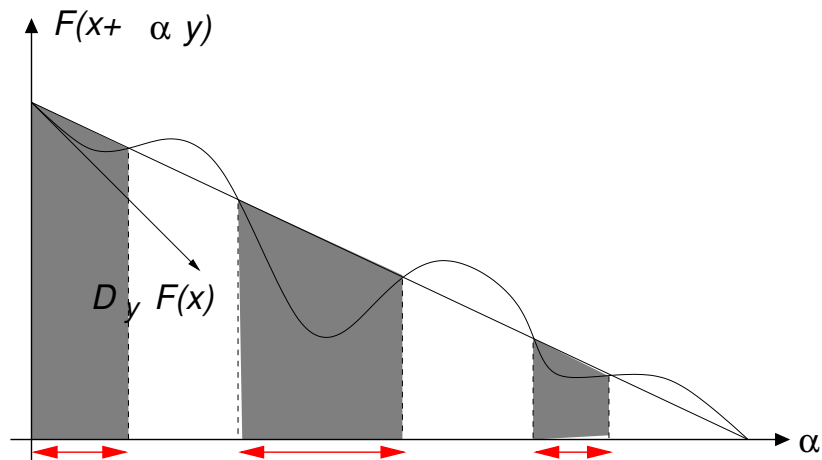


Abbildung 3.6: Graphische Darstellung der Armijo-Bedingung aus (3.11) mit einem Wert von $C_1 \sim \frac{1}{2}$ und drei zulässigen Schrittweitenbereichen. Die Schrittweite ist immer dann zulässig, wenn die Funktion unterhalb der Geraden liegt.

- Gibt es immer Schrittweiten, die diese Anforderungen erfüllen?
- Wenn ja, wie kann man diese Schrittweiten bestimmen?

Die Antwort auf die erste Frage ist “ja”, solange nur F nicht entlang der Richtung y gegen $-\infty$ läuft und die Antwort auf die zweite Frage ist das folgende “Verfahren” zur Bestimmung einer Schrittweite α , die sogar der *starken Wolfe*-Bedingung genügt.

1. Starte mit $\alpha_0 = 0$ und beliebigem¹⁰⁰ $\alpha_1 > 0$.
2. Für $k = 2, 3, \dots$ berechne $\alpha_k = \rho \alpha_{k-1}$ mit einem Wert $\rho > 1$ bis eine der folgenden drei Situationen eintritt:
 - (a) Die Schrittweite passt, d.h. α_k erfüllt (3.11) und (3.13).
 - (b) Die Armijo-Bedingung ist verletzt:

$$F(x + \alpha_k y) - F(x) > C_1 \alpha_k D_y F(x).$$
 - (c) Die Richtung ist eine Aufstiegsrichtung geworden, d.h. $D_y F(x + \alpha_k y) > 0$.
 - (d) Der letzte Punkt war eine bessere Näherung, d.h. $F(x + \alpha_{k-1} y) < F(x + \alpha_k y)$.
3. Für irgendein k bricht dieses Verfahren ab, weil einer der vier obigen Fälle eingetreten ist¹⁰¹ und entweder ist Fall 2a) eingetreten und α_k ist eine gute Schrittweite, oder aber es liegt eine gute Schrittweite zwischen α_{k-1} und α_k , und die kann man dann durch ein Bisektionsverfahren, siehe [10], finden.

¹⁰⁰Gut geratenem!

¹⁰¹Passiert das schon für $k = 1$, dann war α_1 entweder besonders gut oder besonders schlecht geraten und man ist entweder fertig oder muß sich ein neues α_1 ausdenken.

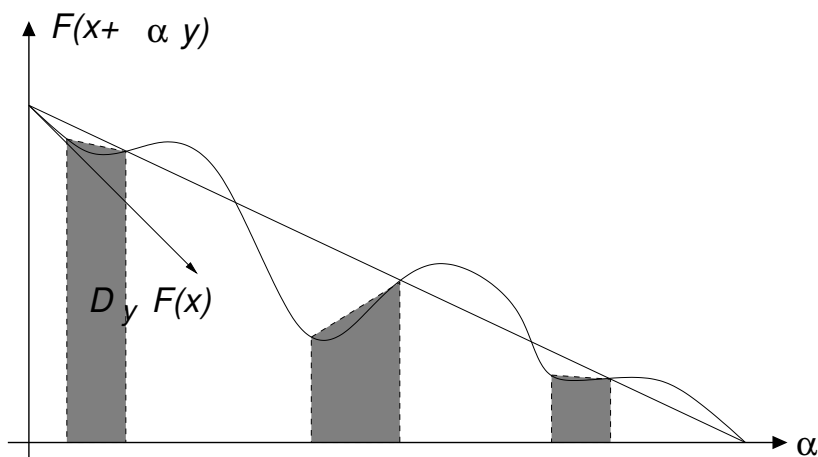


Abbildung 3.7: Die Wolfe-Bedingungen, also (3.11) **und** (3.12) lassen nur noch die hier abgebildeten Schrittweiten zu. Stellen, an denen die Funktion zu stark abfällt, werden ausgenommen.

3.4 Trust Regions

Einen Typ von Verfahren zur allgemeinen, nichtlinearen Optimierung sollten wir uns zumindest noch kurz ansehen, nämlich die sogenannten *Trust-Region-Verfahren*, die beispielsweise auch im recht etablierten Programmpaket *LANCELOT*, siehe [2], integriert sind. Dazu nähert man die zu optimierende Funktion wieder durch ein einfaches, normalerweise quadratisches, Modell an und legt einen Bereich fest, auf dem man diesem Modell “vertraut”, die sogenannte *Trust Region*, normalerweise ein Kreis mit Mittelpunkt x^* und Radius r . Das quadratische Modell $Q(x)$ kann man auf verschiedene Arten wählen:

Taylor-Reihe: Man setzt, wie wir es ja auch schon getan haben, die quadratische Funktion als

$$Q(x) = F(x^*) + (x - x^*)^T \nabla F(x^*) + (x - x^*)^T \nabla^2 F(x^*) (x - x^*)^T$$

an, nimmt also die “lokal beste” quadratische Näherung an x^* . Nachteil: Man braucht wieder einmal zweite Ableitungen!

Interpolation: Man wählt Q so, daß Q und F an bestimmten Punkten¹⁰² übereinstimmen, daß also Q die Funktion F an diesen Punkten *interpoliert*. Hier braucht man nur noch Punktauswertungen von F .

“Regression”: Man wählt Q so, daß das quadratische Mittel

$$\sum_{j=1}^N [Q(x_j) - F(x_j)]^2$$

¹⁰²Die Anzahl der Punkte hängt von der Dimension n ab und die Punkte dürfen nicht zu ungeschickt gewählt werden.

an Punkten x_1, \dots, x_N minimiert wird. Bei diesem Ansatz kann man *mehr* Punkte verwenden als bei der Interpolation und die Punkte können liegen wie sie wollen.

Das Trust-Region-Verfahren geht nun folgendermaßen vor:

1. Bestimme aus dem *Modell* Q eine Abstiegsrichtung y und eine Schrittweite α , die aber so klein sein muß, daß $x + \alpha y$ immer noch *in* der Trust-Region liegt.
2. Vergleiche den *realen Gewinn* $F(x) - F(x + \alpha y)$ mit dem “*Modellgewinn*” $Q(x) - Q(x + \alpha y)$, über den Quotienten

$$\eta = \frac{F(x) - F(x + \alpha y)}{Q(x) - Q(x + \alpha y)}.$$

- Ist η klein oder gar negativ, dann hat das Modell nichts getaugt und die Trust-Region, genauer gesagt, deren Radius ρ , wird verkleinert.
 - Ist η “moderat”, dann wird x durch $x + \alpha y$ ersetzt, die Trust-Region bleibt unverändert.
 - Ist η groß, dann erstzt man ebenfalls x durch $x + \alpha y$, hat aber das Modell anscheinend unterschätzt und vergrößert deswegen zusätzlich den Radius der Trust-Region.
3. Nun fährt man mit Schritt 1) fort.

Tatsächlich kann man *beweisen*, daß solche Verfahren tatsächlich imstande sind, Minima zu finden, allerdings muß man die Unterscheidung zwischen “großen”, “moderaten” und “kleinen” Werten von η und die Vergrößerung und Verkleinerung von ρ mit ein klein wenig Sorgfalt behandeln.

3.5 Zusammenfassung

Diese Schnelleinführung in das (Un-)Wesen der Abstiegsverfahren war natürlich in keinsten Weise vollständig oder erschöpfend und es gibt hier eine Vielzahl von Verfahren, die die unterschiedlichsten¹⁰³ Namen tragen. Trotzdem basieren sie aber ziemlich alle auf dem iterativen Vorgehen mit Bestimmung von Abstiegsrichtungen und Schrittweiten und dem Ziel, einen *stationären* Punkt, also einen Punkt mit “waagerechter Tangente” zu finden. All diese Verfahren haben einiges gemeinsam:

- Sie sind nur *lokal konvergent*, das heißt, sie finden im allgemeinen nur dann ein Extremum, wenn man hinreichend nahe bei einem solchen startet. Damit ist die Wahl des Startpunkts ein extrem kritischer Prozess¹⁰⁴, dem man deswegen bei der Modellbildung gar nicht zu viel Aufmerksamkeit widmen kann.
- Wenn der Startwert schlecht gewählt ist, dann kann so ein Verfahren *divergieren*¹⁰⁵ oder sich an nicht-extremalen Stellen festlaufen oder in Kreisen laufen oder sich sonstwie unmöglich benehmen.

¹⁰³Normalerweise auch mehrere verschiedene Namen für dasselbe Verfahren.

¹⁰⁴Fast wichtiger als die Auswahl des Verfahrens.

¹⁰⁵Die Punkte laufen irgendwo ins Unendliche.

- Selbst wenn so ein Verfahren konvergiert, konvergiert es nur gegen ein *lokales* Extremum, das noch lange nicht die globale Optimallösung sein muss. Diese zu finden ist eine ganz andere Geschichte, die im wesentlichen auf Heuristiken und weniger auf Mathematik basiert, siehe z.B. [1, 4].
- Auch das Extremum kann sehr subtil vom Startwert abhängen, siehe zur Erinnerung Abb. 3.3.

Fazit: Auch wenn die Verfahren noch so ausgetestet und “bewiesenermaßen” zuverlässig sind, gibt es doch einige prinzipielle Probleme, die immer wieder durchschlagen können. Daher ist es bei der nichtlinearen Optimierung auch absolut essentiell, einen guten Startwert zu haben.

Es ist fast nicht möglich, etwas Gutes zu schreiben, ohne daß man sich dabei jemanden oder auch eine gewisse Auswahl von Menschen denkt, die man anredet. Es erleichtert wenigstens den Vortrag sehr in tausend Fällen gegen einen.

G. Chr. Lichtenberg

Zurück zur linearen Optimierung

4

Jetzt kommen wir endlich zur lange versprochenen *numerischen* Behandlung des linearen Optimierungsproblems. Wir haben in Beispiel 1.3, insbesondere in Abb 1.4, ja schon gesehen, wie man das Problem *graphisch* angehen kann, aber das klappt halt aus offensichtlichen Gründen nur für zwei Variable. Unser lineares Optimierungsproblem in diesem Kapitel soll immer von der Form

$$\min c^T x, \quad Ax \leq b, \quad x \geq 0, \quad (4.1)$$

sein, wobei die Vektoren b, c und die Matrix A einfach “passende” Größe haben sollen. Man beachte allerdings, daß hier in der Matrix A nur noch diejenigen Nebenbedingungen codiert werden, die *zusätzlich* zu $x \geq 0$ einzuhalten sind.

4.1 Der Simplexalgorithmus

Der *Simplexalgorithmus*, der 1947 von G. B. Dantzig eingeführt wurde, operiert formal auf dem sogenannten *Simplextableau*, in dem die Variablen und die Matrix folgendermaßen aufgetragen sind:

	x_1	\dots	x_n	
y_1	a_{11}	\dots	a_{1n}	b_1
\vdots	\vdots	\ddots	\vdots	\vdots
y_m	a_{m1}	\dots	a_{mn}	b_m
	c_1	\dots	c_n	z

Schauen wir uns zuerst einmal die Einträge in dem Tableau an:

- a_{11}, \dots, a_{mn} sind die Einträge der Matrix A , die die zusätzlichen Nebenbedingungen beschreibt.
- b_1, \dots, b_m sind die Komponenten des Nebenbedingungsvektors b .

- c_1, \dots, c_m sind die Komponenten des Zielfunktionsvektors c , der durch $c^T x$ die zu minimierende Zielfunktion bestimmt.
- x_1, \dots, x_n sind die *Basisvariablen* des Schemas, also diejenigen Parameter deren Wert wir für die Optimallösung bestimmen müssen.
- y_1, \dots, y_m sind die *abhängigen Variablen*, deren Wert durch $y = Ab - b$ festgelegt ist.
- z ist schließlich der Wert der Zielfunktion, den man erhält, wenn man die Variablen in der "Kopfzeile" auf Null setzt. Diese Variable wird am Anfang auf Null gesetzt.

Jetzt müssen wir eine Annahme machen, nämlich, daß $b \geq 0$ ist. Das bedeutet wiederum nichts anderes, als daß $A0 = 0 \leq b$ ist, daß also der Punkt $x = 0$ *zulässig*¹⁰⁶ ist, eine Annahme, die für die Schuhproduktion aus Beispiel 1.3 erfüllt ist.

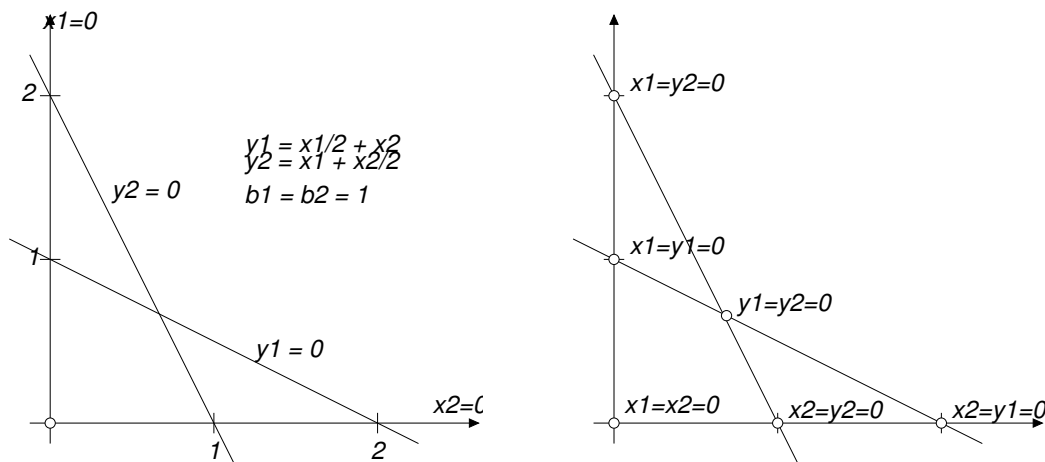


Abbildung 4.1: *Links*: Die Basis- und abhängigen Variablen für $A = \begin{bmatrix} 1 & 1/2 \\ 1/2 & 1 \end{bmatrix}$. Man sieht, daß jede Bedingung der Form $x_j = 0$ bzw. $y_j = 0$ einer Linien entspricht, die den zulässigen Bereich begrenzen. Außerdem ist $x = 0$ hier eine zulässige Ecke.

Rechts: Die Ecken. Jede Ecke ist dadurch ausgezeichnet, daß zwei Variablen, und hier sind Basisvariable wie abhängige möglich, den Wert Null haben.

Die geometrische Bedeutung der Basisvariablen und der abhängigen Variablen ist in Abb 4.1 dargestellt. Wenn man eine dieser Variablen gleich Null setzt, so erhält man eine der Geraden¹⁰⁷, die den zulässigen Bereich beranden und man erhält auch alle Gerade auf diese Art und Weise. Die *Ecken* hingegen – und das sind ja, wie wir aus der graphischen Lösung von Beispiel 1.3¹⁰⁸

¹⁰⁶Ein Punkt oder eine "Konfiguration" x heißt *zulässig*, wenn die Nebenbedingungen $Ax \leq b$ **und** $x \geq 0$ erfüllt sind.

¹⁰⁷Ist $n > 2$ so sind die Gegenstücke sogenannte *Hyperebenen*, das sind $n - 1$ -dimensionale Gebilde!

¹⁰⁸Siehe insbesondere Abb. 1.4.

wissen, die Punkte, an denen das Extremum angenommen wird – erhalten wir, indem wir *zwei*¹⁰⁹ Variablen gleich Null setzen. Allerdings erhalten wir so mehr als nur die Ecken des zulässigen Bereichs, wie man ebenfalls in Abb. 4.1 sehr schön sieht: Nur die Ecken

$$x_1 = x_2 = 0, \quad x_1 = y_1 = 0, \quad x_2 = y_2 = 0, \quad y_1 = y_2 = 0$$

sind auch Ecken des zulässigen Bereichs, die Ecken $x_1 = y_2 = 0$ und $x_2 = y_1 = 0$ liegen dagegen zu weit draussen.

Die Idee des Simplexalgorithmus ist nun ganz einfach:

Durch Vertauschen von Basisvariablen und abhängigen Variablen wechselt man von der Ecke $x_1 = \dots = x_n = 0$ zu einer Nachbarecke, an der die Zielfunktion einen kleineren Wert¹¹⁰ annimmt und formt das Tableau so um, daß die ausgetauschte Basisvariable als abhängige Variable geschrieben wird.

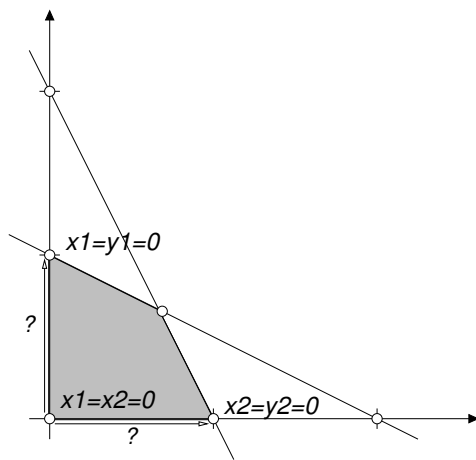


Abbildung 4.2: Übergang von der Ecke $x = 0$ zu einer der beiden Nachbarecken des zulässigen Bereichs – je nachdem, an welcher von beiden der Wert der Zielfunktion kleiner ist.

Diese Idee ist in Abb 4.2 dargestellt. Was dabei passiert, sieht man in der folgenden Tabelle:

Ecke	Bedingung	Vertauschung
Ausgangsecke	$x_1 = x_2 = 0$	keine
“Oben”	$x_1 = y_1 = 0$	$x_2 \leftrightarrow y_1$
“Rechts”	$x_2 = y_2 = 0$	$x_1 \leftrightarrow y_2$

¹⁰⁹Im allgemeinen Fall n .

¹¹⁰Nicht vergessen: Unsere Normalform (4.1) verlangt von uns, nach einem *Minimum* zu suchen!

Und je nach durchgeführter Vertauschung müssen wir dann entweder x_2 in Abhängigkeit von x_1 und y_1 oder x_1 in Abhängigkeit von x_2 und y_2 ausdrücken. Soviel zur Idee, jetzt aber zum Verfahren. Dazu verwenden wir unser Tableau

	u_1	\dots	u_n	
v_1	a_{11}	\dots	a_{1n}	b_1
\vdots	\vdots	\ddots	\vdots	\vdots
v_m	a_{m1}	\dots	a_{mn}	b_m
	c_1	\dots	c_n	z

wobei in u und v die Symbole $x_1, \dots, x_n, y_1, \dots, y_m$ eingetragen sind¹¹¹. Der Algorithmus läuft nun in den folgenden Schritten ab:

1. *Bestimmung der "Pivotspalte"*: Wähle ein k zwischen 1 und n , so daß $c_k < 0$ ist¹¹².
Ist dies nicht möglich, dann haben wir die Lösungsecke gefunden und $-z$ ist der optimale Wert. → **Stop**.
2. *Bestimmung der "Pivotzeile"*: Für alle ℓ zwischen 1 und m mit $a_{\ell k} > 0$ bestimme die Hilfsgrößen¹¹³

$$h_\ell = \frac{b_\ell}{a_{\ell k}} \geq 0$$

und wähle j so, daß $h_j = \min h_\ell$ ist.

Sollte diese Bestimmung nicht möglich sein, weil alle $a_{\ell k} \leq 0$ sind, dann ist die Zielfunktion unbeschränkt und es gibt keine Lösung. → **Stop**.

3. *Austausch*: Vertausche die beiden Symbole u_k und v_j .
4. *Update des Tableaus*:

- (a) *Pivotelement*: $a_{jk} \leftarrow 1/a_{jk}$.
- (b) *Pivotspalte*: $\ell = 1, \dots, m, \ell \neq j$,

$$\begin{aligned} a_{\ell k} &\leftarrow -a_{\ell k}/a_{jk} \\ c_\ell &\leftarrow -c_\ell/a_{jk} \end{aligned}$$

- (c) *Pivotzeile*: $\ell = 1, \dots, n, \ell \neq k$,

$$\begin{aligned} a_{j\ell} &\leftarrow a_{j\ell}/a_{jk} \\ b_\ell &\leftarrow b_\ell/a_{jk} \end{aligned}$$

¹¹¹Am Anfang beginnt man mit $u = x$ und $v = y$, aber mit jedem Vertauschungsschritt werden diese natürlich durcheinandergemischt. Trotzdem ist eine Buchführung über diese Vertauschungen unerlässlich, denn am Schluß will man ja nicht nur den Wert der Optimallösung, sondern auch die zugehörigen x -Werte wissen.

¹¹²Sind mehrere c_k negativ, dann kann man die Spalte so wählen, daß c_k so klein wie möglich wird.

¹¹³Hier schlägt die Annahme durch, daß $b \geq 0$ ist!

(d) Alle übrigen Elemente $a_{pq}, p \neq j, q \neq \ell$, nach der ‘‘Rechtecksregel’’:

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

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

oder, in einfacher Form

$$\begin{matrix} a_{jk} & \dots & r \\ \vdots & \ddots & \vdots \\ s & \dots & t \end{matrix} \implies t \leftarrow t - \frac{rs}{a_{jk}}$$

5. Fahre mit 1) fort, bis eine der beiden Abbruchbedingungen erreicht ist.

Bemerkung 4.1 (Simplexverfahren)

1. Nachdem ein Optimierungsproblem entweder unbeschränkt und damit unlösbar ist oder nicht, muß der Abbruch ‘‘mangels Lösung’’ immer bereits im ersten Schritt erfolgen. Spätere Abbrüche können also nur daher kommen, daß die Optimallösung gefunden wurde.
2. Die Wahl der Pivotzeile in Schritt 2) hat ebenfalls eine geometrische Interpretation: Hat man sich einmal für eine Pivotspalte, also eine auszutauschende Basisvariable entschieden, so muß man diese gegen die erste abhängige Variable austauschen, mit sich $x_k = 0$ schneidet. Hat man sich beispielsweise in Abb. 4.2 dafür entschlossen, nach rechts zu gehen, so sagt uns Abb. 4.1, daß wir x_1 gegen y_1 und nicht gegen y_2 austauschen müssen, denn der Schnitt von $x_2 = 0$ mit $y_2 = 0$ kommt früher als der mit $y_1 = 0$. Genau dieses ‘‘früher’’ realisiert die Minimumssuche in Schritt 2).

Nun hat so ein Polyeder aber nur endlich viele Ecken, nämlich maximal $\binom{n+m}{n}$ und da der Simplexalgorithmus in jedem Schritt zu einer *besseren* Ecke weitergeht, wenn er nicht wegen Optimalität terminiert, muß er irgendwann eine Optimallösung finden. Das können wir wie folgt formulieren.

Satz 4.2 Ist das Optimierungsproblem (4.1) lösbar und ist die Startecke $x = 0$ eine zulässige Ecke, dann berechnet das Simplexverfahren die Lösung in endlich vielen Schritten.

Beispiel 4.3 Jetzt wird es aber höchste Zeit für ein Beispiel und was wäre da geeigneter als unsere Schuhfabrik aus Beispiel 1.3. Das zugehörige Anfangstableau ist dann

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

Wegen “min” und “ \leq ” der Normalform (4.1) mussten wir hierbei ein paar Vorzeichen umdrehen. Suchen wir nun nach der Pivotspalte; wegen der Einträge -16 und -32 könnten wir beide Spalten wählen, da aber in der zweiten Spalte der kleiner Wert steht, entscheiden wir uns für diese, also $k = 2$. Die Hilfswerte h_ℓ sind nun der Vektor

$$h = \begin{bmatrix} 8000/10 \\ 2000/5 \\ 4500/15 \end{bmatrix} = \begin{bmatrix} 800 \\ 400 \\ \boxed{300} \end{bmatrix} \implies j = 3.$$

Nach Vertauschen der Symbole erhalten wir in Pivotzeile und -spalte die folgenden Einträge:

	x_1	y_3	
y_1	*	$-\frac{2}{3}$	*
y_2	*	$-\frac{1}{3}$	*
x_2	$\frac{2}{5}$	$\frac{1}{15}$	300
	*	$\frac{32}{15}$	*

und die anderen Einträge bestimmen sich nach der “Rechtecksregel” als

	x_1	y_3	
y_1	16	$-\frac{2}{3}$	5000
y_2	2	$-\frac{1}{3}$	500
x_2	$\frac{2}{5}$	$\frac{1}{15}$	300
	$-\frac{16}{5}$	$\frac{32}{15}$	9600

Im zweiten Schritt ist die Pivotspalte $k = 1$ leicht gefunden und es ist

$$h = \begin{bmatrix} 312.5 \\ \boxed{250} \\ 750 \end{bmatrix} \implies \begin{array}{|c|c|c|c|} \hline & y_2 & y_3 & \\ \hline y_1 & -8 & 2 & 1000 \\ \hline \boxed{x_1} & \frac{1}{2} & -\frac{1}{6} & \boxed{250} \\ \hline \boxed{x_2} & -\frac{1}{5} & \frac{2}{15} & \boxed{200} \\ \hline & \frac{8}{5} & \frac{40}{15} & \boxed{10400} \\ \hline \end{array}$$

womit das Verfahren terminiert – in der unteren Zeile stehen nur positive Werte – und wovon wir die Optimalkonfiguration [250, 200] sowie den Optimalwert¹¹⁴ 10400 ablesen können.

¹¹⁴Wir haben $-c^T x$ minimiert und die Lösung davon ist -10400 (Vorzeichen beachten)! Wenn wir also $c^T x$ maximieren wollen, dann heben sich die Vorzeichen auf und wir erhalten genau den gewünschten Wert.

4.2 Die Zweiphasenmethode

Ein Problem oder eine Schwäche hat unser Simplexalgorithmus allerdings noch: Was tun, wenn der Punkt $x = 0$ **keine** zulässige¹¹⁵ Ecke ist, wenn also die Forderung $b \geq 0$ **nicht** erfüllt ist? Solche Situationen treten typischerweise dann auf, wenn Dinge von A nach B bewegt werden müssen und daher die “untätige” Lösung $x = 0$ nicht zulässig ist.

Beispiel 4.4 (Transportproblem) *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 Kilometerzahl¹¹⁶?

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
$A \rightarrow X$	x
$A \rightarrow Y$	y
$A \rightarrow Z$	$18 - x - y$
$B \rightarrow X$	$11 - x$
$B \rightarrow Y$	$10 - y$
$B \rightarrow Z$	$x + y - 9$

und alle diese Größen müssen selbstverständlich positiv sein, also erhalten wir neben dem offensichtlichen $x, y \geq 0$ noch

$$\begin{bmatrix} 18 \\ 11 \\ 10 \\ -9 \end{bmatrix} \geq \begin{bmatrix} x + y \\ x \\ y \\ -x - y \end{bmatrix}.$$

Die Gesamtzahl der gefahrenen Kilometer ist dann

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

¹¹⁵Wegen der immanenten Nebenbedingung $x \geq 0$ ist $x = 0$ immer eine Ecke in dem Sinne, daß n Variablen den Wert Null haben-

¹¹⁶Auch hier handelt es sich eigentlich wieder um ein Problem aus der *Ganzzahloptimierung*, aber wieder einmal wird, rein zufällig, die kontinuierliche Optimallösung ganzzahlig sein.

und dieser Wert muß unter den obigen Nebenbedingungen minimiert werden. Unsere Normalform für das Optimierungsproblem lautet also

$$\min 229 - x - 3y, \quad \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \leq \begin{bmatrix} 18 \\ 11 \\ 10 \\ -9 \end{bmatrix}$$

bestimmen; die letzte Zeile der Nebenbedingungen, die nichts anderes als $x + y \geq 9$ bedeutet, 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. 4.3 grafisch dargestellt.

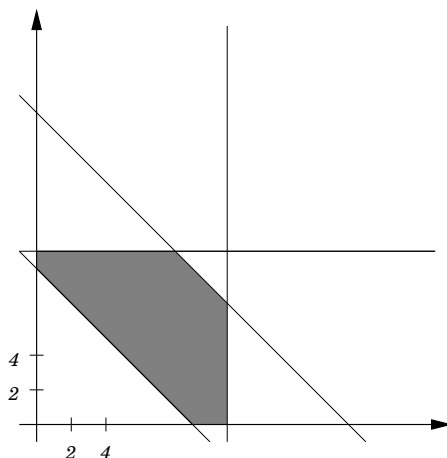


Abbildung 4.3: Der zulässige Bereich für das Transportproblem aus Beispiel 4.4; der Nullpunkt ist offensichtlich “abgeschnitten” worden.

Beispiel 4.5 (Stein, Schere, Papier) Wir wollen jetzt versuchen¹¹⁷, die optimale gemischte Strategie für das Spiel “Stein, Schere, Papier” aus Beispiel 1.12 bestimmen. Im ersten Schritt übersetzen wir es in ein Optimierungsproblem. Die Nebenbedingungen ergeben sich als

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ 2 & -1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \iff \begin{bmatrix} 1 & 1 & 1 \\ -1 & -1 & -1 \\ 1 & -2 & 1 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \\ -2 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \leq \begin{bmatrix} 1 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

¹¹⁷Siehe auch Übung 1.3.

und die Nebenbedingung $x \geq 0$ versteht sich ja für Wahrscheinlichkeiten von selbst. Auch hier ist $x = 0$ keine zulässige Ecke mehr¹¹⁸! Die Zielfunktion $-x_1 + x_2$ ergibt sich aus der letzten Spalte der Gewinnmatrix von Stein, Schere, Papier und da sie maximiert werden soll, unser Schema aber für Minimierungsprobleme gebaut ist, müssen wir also deren negativen Wert, $x_1 - x_2$, minimieren.

Ist also $b^* = \min_j b_j < 0$, dann ist $x = 0$ keine zulässige Ecke mehr und in diesem Fall können wir unseren Simplexalgorithmus erst einmal vergessen. Trotzdem gibt es einen Ausweg: Da $x = 0$ ja auf jeden Fall eine Ecke ist, Zulässigkeit hin oder her, verwendet man eine Variante des Simplexalgorithmus, die ebenfalls von Ecke zu Ecke springt – jetzt aber so, daß man nach endlich vielen Schritten bei einer zulässigen Ecke landet.

Zu diesem Zweck führt man eine zusätzliche “künstliche” Variable x_0 ein und betrachtet das (Hilfs-)Optimierungsproblem

$$\min z'(x) := -x_0 - b^*, \quad \left[\begin{array}{c|c} 1 & \\ \vdots & A \\ 1 & \end{array} \right] \left[\begin{array}{c} x_0 \\ x \end{array} \right] \leq \left[\begin{array}{c} b_1 - b^* \\ \vdots \\ b_m - b^* \end{array} \right], \quad (4.2)$$

oftmals als *Phase 1* des Simplexalgorithmus bezeichnet. Der Trick von (4.2) beruht auf zwei Beobachtungen:

1. Der Punkt $x_0 = 0, x = 0$ ist eine zulässige Ecke, da $b_j - b^* \geq 0$ ist – wir können also unseren “normalen” Simplexalgorithmus verwenden, um dieses Problem zu lösen.
2. Ist die Hilfszielfunktion $z'(x) = -x_0 + b^*$ für irgendwelche im Simplexalgorithmus bestimmten Werte von x_0 und x einmal *negativ*, also $-x_0 - b^* \leq 0$, dann ist $x_0 \geq -b^*$ und da wegen der Nebenbedingung von (4.2)

$$\left[\begin{array}{c} b_1 - b^* \\ \vdots \\ b_m - b^* \end{array} \right] \geq \left[\begin{array}{c|c} 1 & \\ \vdots & A \\ 1 & \end{array} \right] \left[\begin{array}{c} x_0 \\ x \end{array} \right] = Ax + \left[\begin{array}{c} x_0 \\ \vdots \\ x_0 \end{array} \right] \geq Ax - \left[\begin{array}{c} b^* \\ \vdots \\ b^* \end{array} \right]$$

gilt, also $Ax \leq b$, ist der so erhaltene Wert von x eine zulässige Ecke des *Originalproblems*, mit der wir das “eigentliche” Simplexverfahren, die *Phase 2*, starten können.

Auf eines müssen wir allerdings noch achten: Auch wenn wir für die Bestimmung der Pivotzeile die Hilfs-Zielfunktion $-x_0 - b^*$ aus (4.2) verwenden, müssen wir trotzdem auch die Original-Zielfunktion $z(x) = c^T x$ bei jedem Austauschschritt mit umformen.

Beispiel 4.6 (Transportproblem; Phase I) Für unser Transportproblem erhalten wir die erweiterte Form für das Simplextableau zum Auffinden der Startecke mit unten angefügter Hilfs-

¹¹⁸Und zwar gerade deswegen, weil sich die Wahrscheinlichkeiten sich zu 1 summieren.

Zielfunktion ist nun¹¹⁹

	x_0	x_1	x_2	
y_1	1	1	1	27
y_2	1	1	0	20
y_3	1	0	1	19
y_4	1	-1	-1	0
	0	-1	-3	-229
z'	-1	0	0	-9

was sich, gemäß unserer Regeln in

	x_0	x_1	x_2	
y_1	1	1	1	27
y_2	1	1	0	20
y_3	1	0	1	19
y_4	1	-1	-1	0
	0	-1	-3	-229
z'	-1	0	0	-9

→

	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
	0	-1	-3	-229
z'	1	-1	-1	-9

→

	y_4	y_2	x_2	
y_1	0	-1	1	7
x_1	$-\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	10
y_3	$-\frac{1}{2}$	$-\frac{1}{2}$	$\frac{3}{2}$	9
x_0	$\frac{1}{2}$	$\frac{1}{2}$	$-\frac{1}{2}$	10
	$-\frac{1}{2}$	$\frac{1}{2}$	$-\frac{3}{2}$	-219
z'	$\frac{1}{2}$	$\frac{1}{2}$	$-\frac{1}{2}$	1

umformt. Das letzte Tableau erfüllt die Abbruchbedingung und wir haben die Startecke $x_1 = 10$, $x_2 = 0$ gefunden.

Trotzdem stehen wir jetzt nochmals vor einem Problem, allerdings einem nicht allzu schweren: Das Tableau passt nicht, hat eine Spalte zu viel, was ja auch klar ist wegen der künstlichen Variablen x_0 . Nachdem die jetzt, genau wie die Hilfszielfunktion z' , ihre Schuldigkeit getan hat, solltet wir sie natürlich auch wieder loswerden. Für die Hilfszielfunktion ist dies ganz besonders einfach, wir brauchen nur die angehängte Zeile zu löschen, aber um die Variable x_0 loswerden zu können, darf sie keine abhängige Nicht-Basis-Variable sein. Dazu führt man einen weiteren Austauschschritt durch, der x_0 zur Basisvariablen macht, also in die obere Zeile plziert, und verwirft dann einfach die Spalte mit x_0 . Dazu verwendet man einfach eine Spalte, in der die Matrix einen *positiven* Eintrag hat.

Aber Halt! Dabei würden wir einen Fehler machen! Denn unsere Nebenbedingungen waren ja modifiziert, wir haben ja den Vektor

$$-b^* \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

zu unseren Nebenbedingungen hinzugezählt. Wenn wir uns dazu nochmals (4.2), dann ist das nichts anderes, als die zur Variablen x_0 gehörige *Spalte*. Und diese Korrektur müssen wir nun in unserem *modifizierten* Problem, bei dem die Rollen einiger Basis- und abhängiger Variablen vertauscht wurden, eben wieder rückgängig machen. Und das tun wir, indem wir das $-b^*$ -fache der zur x_0 gehörigen Spalte von den Randbedingungen *subtrahieren*.

¹¹⁹Wir minimieren hier keine *lineare* Funktion der Form $c^T x$, sondern die *affine* Funktion $x_0 - b^*$, so daß in der Ecke rechts unten der *negative* (den sonst hätten wir ja keine Zweiphasenmethode gebraucht) Wert b^* steht.

Beispiel 4.7 (Transportproblem; Übergang zu Phase II) Zuerst halten wir fest, daß x_0 den Wert 10 haben muß. Um die Variable x_0 auszutauschen, suchen wir nach einem positiven Wert in der zugehörigen Zeile und entscheiden uns beispielsweise für die erste Spalte mit y_4 . Das liefert dann den Austausch

	y_4	y_2	x_2	
y_1	0	-1	1	7
x_1	$-\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	10
y_3	$-\frac{1}{2}$	$-\frac{1}{2}$	$\frac{3}{2}$	9
x_0	$\frac{1}{2}$	$\frac{1}{2}$	$-\frac{1}{2}$	10
	$-\frac{1}{2}$	$\frac{1}{2}$	$-\frac{5}{2}$	-219
z'	$\frac{1}{2}$	$\frac{1}{2}$	$-\frac{1}{2}$	1

→

	x_0	y_2	x_2	
y_1	0	-1	1	7
x_1	1	1	0	20
y_3	1	0	1	19
y_4	2	1	-1	20
	1	1	-3	-209
z'	$-\frac{1}{2}$	*	*	*

was uns nach Update der rechten Spalte¹²⁰ unter Berücksichtigung von $b^* = -9$ und Wegwerfen der x_0 -Spalte sowie der überflüssigen Zeile das Tableau

	y_2	x_2	
y_1	-1	1	7
x_1	1	0	11
y_3	0	1	10
y_4	1	-1	2
	1	-3	-210

liefert, das wir dann mit unserem normalen Simplexalgorithmus, also der Phase II, behandeln können.

Übung 4.1 Bestimmen Sie die Optimallösung für das Transportproblem. ◇

Aber spätestens jetzt wird's langweilig! Wir haben zwar nun ein Verfahren zur Hand, mit dessen Hilfe wir jede Menge von linearen Optimierungsproblemen durchrechnen können, aber gerade sowas ist so eine stupide Tätigkeit, daß man sie gerne einem Computer überläßt. Zu diesem Zweck haben wir eine Octave-Routine `SimSimplex` bzw `SimSimplexv` (mit Ausgabe der Tableaus) zur Verfügung, die diesen Job für uns erledigen.

Beispiel 4.8 (Stein, Schere, Papier; die Antwort) Hier lassen wir nun den Computer die Arbeit für uns machen und geben die Parameter folgendermaßen ein (siehe Beispiel 4.5):

```
octave> A = [ 1 1 1; -1 -1 -1; 1 -2 1; -1 2 -1; 2 -1 -1; -2 1 1 ]
A =
```

```

1   1   1
-1  -1  -1
1  -2   1
```

¹²⁰Natürlich behandeln wir die Zielfunktion auch gleich mit.

```

## SimSimplex.m (Optmierung fuer Hoerer aller Fachbereiche)
## -----
## Simplexalgorithmus
## Eingabe:
##   A   Matrix
##   b   Rechte Seite (Spalte)
##   c   Zielfunktion (Spalte)
## Ergebnis:
##   x   Loesungsparameter
##   z   Loesungswert

function [x,z] = SimSimplex( A,b,c )
    [m,n] = size( A );

    ## Fuehre Phase 1 aus, wenn noetig

    if ( min( b ) < 0 )
        T = SimPhase1( A,b,c );
    else
        T = [ 0, (1:n), 0; (-1:-1:-m)', A, b; 0, c', 0 ];
    end

    ## Simplex-Iteration

    T = SimPhase2( T );

    x = zeros( n,1 );
    for j = 2:m+1
        k = T( j,1 );
        if ( k > 0 )
            x(k) = T( j,n+2 );
        end
    end
    z = -T( m+2,n+2 );

```

Programm 4.1 SimSimplex.m: Das "Startprogramm" für den Simplexalgorithmus.

```

## SimPhase1.m (Optmierung fuer Hoerer aller Fachbereiche)
## -----
## Phase1 fuer Simplexalgorithmus
## Eingabe:
##   A,b,c      Parameter des SA

function T = SimPhase1( A,b,c )
    [m,n] = size( A );
    bb = min( b );

    ## Erweitertes Tableau

    T = [
        0, (0:n), 0;
        (-1:-1:-m)', ones( m,1 ), A, b .- bb;
        0, 0, c', 0;
        0, -1, zeros( 1,n ), bb
    ];

    M = m+3; N = n+3; k = 1; kl = 1;
    while ( T( M,N ) < 0 && k > 0 )
        [j,k] = SimPivot( T( 2:M, 2:N ), 1 );
        if ( k > 0 )
            ## Austausch
            t = T( 1,k+1 ); T( 1,k+1 ) = T( j+1,1 ); T( j+1,1 ) = t;
            T( 2:M,2:N ) = SimAustausch( T( 2:M,2:N ), j,k );
            kl = k;
        end
    end
end

```

Programm 4.2 SimPhase1t1.m: Phase I des Simplexalgorithmus, Teil 1: Erweitern und Optimieren

```

## Tausche Variable 0 zurueck, wenn noetig, und zwar mit
## LETZTEM k

xx = ( T( 2:M-2,1 ) == 0 );
if ( sum(xx) > 0 )
    j = find( xx );
    x0Val = T( j+1,N );
    k = k1;
    ## Austausch
    t = T( 1,k+1 ); T( 1,k+1 ) = T( j+1,1 ); T( j+1,1 ) = t;

    T( 2:M,2:N ) = SimAustausch( T( 2:M,2:N ), j,k );
end

## Finde Spalte zu x0
k = find( T( 1,2:N-1 ) == 0 );

## Modifiziere Randbedingungen
T( 2:M-1,N ) = T( 2:M-1,N ) - x0Val * T( 2:M-1,k+1 );

## Tausche Spalte nach hinten
t = T( :,N-1 ); T( :,N-1 ) = T( :,k+1 ); T( :,k+1 ) = t;

## Loesche vorletzte Spalte und letzte Zeile
T = [ T( 1:M-1,1:N-2), T( 1:M-1,N ) ];

```

Programm 4.3 SimPhase1t2.m: Phase I des Simplexalgorithmus, Teil 2: Entfernen der Variablen x_0 .

```
## SimPhase2.m (Optmierung fuer Hoerer aller Fachbereiche)
## -----
## Phase2 fuer Simplexalgorithmus
## Eingabe:
##   T      Tableau

function T = SimPhase2( T )
    [m,n] = size( T );
    j=1; k=1;

    while( k > 0 )
        [j,k] = SimPivot( T( 2:m,2:n ), 2 );
        if ( k > 0 )
            ## Austausch
            t = T( 1,k+1 ); T( 1,k+1 ) = T( j+1,1 ); T( j+1,1 ) = t;

            T( 2:m,2:n ) = SimAustausch( T( 2:m,2:n ), j,k );
        end
    end
end
```

Programm 4.4 SimPhase2.m: Phase II des Simplexalgorithmus – die einfache, “klassische” Version.

```

## SimPivot.m (Optmierung fuer Hoerer aller Fachbereiche)
## -----
## Pivotsuche fuer Simplexalgorithmus
## Eingabe:
##   A      Tableau
##   pha     Pase: 1/2

function [j,k] = SimPivot( A,pha )
    [m,n] = size( A );
    sm = m + pha - 3; sn = n-1; ## Suchbereich

    [c,k] = min( A( m,1:n-1) );

    if ( c >= 0 )                ## Fertig
        j = 0;k = -1;
        return;
    end

    y = A( 1:sm, k );
    y = y .* (y > 0);           ## Nur positive Eintraege

    if ( y == 0 )                ## Unbeschraenkt
        disp( '**** Unbeschraenkt ****' );
        j = -1; k = -1;
        return;
    end

    ## Alle "unnoetigen" Eintraege in b = 0

    b = A ( 1:sm, n ) ./ ( y + (y == 0) ) .* (y > 0);
    b = b + ( max( abs(b) ) + 1 ) * ( y == 0 );

    [bmin,j] = min( b );

```

Programm 4.5 SimPivot.m: Ermittlung des Pivotelements.

```
## SimAustausch.m (Optmierung fuer Hoerer aller Fachbereiche)
## -----
## Austauschschritt fuer Simplexalgorithmus
## Eingabe:
##   A      Tableau
##   j,k    Koordinaten

function B = SimAustausch( A,j,k )
    piv = A( j,k );
    pzeil = A( j,: );
    pspal = A( :,k );

    ## Rechtecksregel - gnadenlos vektorisiert

    B = A - pspal * pzeil / piv;

    ## Pivotteil

    B( j,: ) = pzeil / piv;
    B( :,k ) = -pspal / piv;
    B( j,k ) = 1/piv;
```

Programm 4.6 `SimAustausch.m`: Der Austauschschritt des Simplexverfahrens. Man beachte, daß man hier sehr schön vektorisieren und so den Austauschschritt effizient berechnen kann.

$$\begin{array}{ccc} -1 & 2 & -1 \\ 2 & -1 & -1 \\ -2 & 1 & 1 \end{array}$$

```
octave> b = [ 1 -1 0 0 0 0 ]'; c = [ 1 -1 0 ]';
```

Dann starten wir die Routine mit

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

```
x =
```

```
0.33333
0.33333
0.33333
```

```
z = -0
```

was uns die optimale Wahrscheinlichkeitsverteilung $\left[\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right]$ und den zu erwartenden Gewinn, nämlich 0 liefert. Mit anderen Worten: Die optimale Strategie wählt Stein, Schere und Papier mit gleicher Wahrscheinlichkeit¹²¹ und der zu erwartende Optimalgewinn ist 0 – das Spiel ist fair, was bei Nullsummenspielen immer bedeutet, daß der zu erwartende Gewinn 0 sein muß.

Da haben wir also jetzt mit jeder Menge Theorie die erwartete Lösung bekommen. Tatsächlich stellt sich aber heraus, daß $x = \left[\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right]$ die *einzige* Lösung des Nebenbedingungssystems

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

ist, so daß der Optimierungsanteil bei diesem Spiel eher klein ist.

¹²¹Naheliegend, denn das Spiel ist ja absolut symmetrisch in den drei Möglichkeiten.

As far as the laws of mathematics refer to reality, they are not certain; and as far as they are certain, they do not refer to reality.

Albert Einstein

Modellierung von Optimierungsproblemen

5

In diesem Abschnitt wollen wir uns ein paar “typische” Optimierungsprobleme in Form einer “Textaufgabe” ansehen und sie dann in eine Form bringen, in der wir sie mit unserer “Software” aus dem letzten Kapitel lösen. Es geht also nun nicht mehr um den mechanischen Berechnungsprozess für die Optimallösung¹²², sondern um die Frage, wie wir ein realistisches, na gut, ein irgendwie realitätsnahes Problem auf “mathematische” Art formulieren können. Die Beispiele stammen aus [5]¹²³.

5.1 Das Diät–Problem

Beginnen wir erst einmal mit einem ganz typischen, einfachen Problem, ganz ähnlich zur “Schuhfabrik”.

Beispiel 5.1 (Frühstücksplanung) *Eine Hausfrau versucht für Ihre Familie ein optimales Frühstück zusammenzustellen. Dafür stehen ihr¹²⁴ zwei verschiedene Typen von Getreideflocken¹²⁵, nämlich Crunchies und Krispies zur Verfügung, die zwei Spurenelemente, Thiamin¹²⁶ und Niacin¹²⁷ 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:*

¹²²Denn der ist so stupide, daß wir ihn getrost einem Computer überlassen können.

¹²³Eine sehr empfehlenswerte, anschauliche und auch noch, wie es sich für Dover–Reprints gehört, preiswerte Einführung in die lineare Optimierung.

¹²⁴Ja, das Beispiel stammt aus den USA.

¹²⁵Auf gut neudeutsch auch als “Cerealien” bezeichnet – das kommt davon, wenn’s an Zerebralien mangelt.

¹²⁶Synonym für Vitamin B_1 , siehe [9].

¹²⁷Synonym für Nicotinsäure, die Nebenwirkungen in [9] liest man besser nicht.

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

Das Problem ist klar: Was ist die optimale 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 (4.1) die Ungleichungen als “ \leq ” geschrieben haben will, erhalten wir also das Optimierungsproblem¹²⁸

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

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
```

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 nicht. Das sieht man ja auch an der Problemstellung selbst, denn die einfachste Möglichkeit, die Nebenbedingungen

¹²⁸Von jetzt an schreiben wir die allgegenwärtige Randbedingung $x \geq 0$ **nicht** mehr explizit hin.

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 Cent¹²⁹. Eine andere Möglichkeit bestünde darin, den Preis zu beschränken, also beispielsweise nur Frühstücke 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 wieder in Schwierigkeiten:

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

    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.

¹²⁹Und enthält im übrigen rund 756 Kalorien, also fast doppelt so viel wie gewünscht. Vielleicht sollte man nochmal ein grundlegend anderes Frühstück in Betracht ziehen ...

5.2 Transportprobleme

Transportprobleme sind immer vom Typ wie in Beispiel 4.4: Ressourcen müssen von Ausgangspunkten zu Zielpunkten transportiert werden, wobei *alle* Ausgangspunkte mit *allen* Zielpunkten als verbunden angenommen werden. Die Entfernungen (oder Kosten) von einer Ausgangspunkt 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.

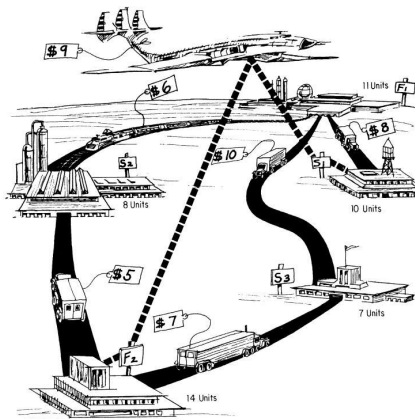


Abbildung 5.1: Kühlschränke und deren Transportwege. Aus [5].

Beispiel 5.2 (Kühlschränke) Eine Firma stellt in zwei Fabriken, F_1 und F_2 , Kühlschränke her, die in den Läden¹³⁰ S_1, S_2, 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	

¹³⁰“Shop”.

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

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

In unserem Beispiel haben wir also die Variablen

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

auch in einen Vektor angeordnet, indem wir die sogenannte *lexikographische*¹³¹ Ordnung verwenden. Unser Beispiel liefert nun die Nebenbedingungen

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

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:

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

Und ab geht's in den Computer:

```
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

¹³¹Indizes werden angeordnet wie im Lexikon: zuerst ordnet man nach dem ersten Eintrag, dann nach dem zweiten und so weiter.

```

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 5.3 (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 [5, S. 20] sind wie folgt:*

	<i>MacDill</i>	<i>March</i>	<i>Davis-Monthan</i>	<i>McConnell</i>	<i>Pinecastle</i>	
<i>Oklahoma City</i>	938	1030	824	136	995	8
<i>Macon</i>	346	1818	1416	806	296	5
<i>Columbus</i>	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.

5.3 Zuordnungsprobleme

Zuordnungsprobleme versuchen Ressourcen und Aufgaben so einander zuzuordnen, daß ein vorgegebener Nutzen maximiert wird. Eine *Zuordnung* zweier Mengen¹³² ist eine "Funktion", die jedem Element der einen Menge¹³³ *eindeutig* ein Element der anderen Menge¹³⁴ zuordnet. Alternativ ist eine *Zuordnungstabelle* eine quadratische Matrix, die in jeder Spalte und jeder Zeile *genau eine* Eins stehen hat.

¹³²Die gleichviele Elemente enthalten müssen.

¹³³Also jeder Ressource.

¹³⁴Also eine Aufgabe.

```

## TransMat.m (Optimierung 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 5.1 TransMat.m: Generierung von Matrizen zum Transportproblem nach dem einfachen Schema.

Beispiel 5.4 (Personal und Fähigkeiten) *Einer Militäreinheit¹³⁵ 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.

Auch hier beschreibt wieder x_{jk} , in welchem Maße Soldat Nummer j Job Nummer k ausübt¹³⁶ und die Nebenbedingungen sind

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

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!

¹³⁵Nicht meine Erfindung, sondern aus [5, S. 56–61]!

¹³⁶Man kann zeigen, daß bei der Optimallösung $x_{jk} = 1$ sein muß, das liegt an der Struktur des Problems.



Abbildung 5.2: Die drei Mitarbeiter und ihre Fähigkeiten. Aus [5]

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
    0
    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.

5.4 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 Netzwerks¹³⁷ dargestellt, siehe Abb. 5.3. Wie

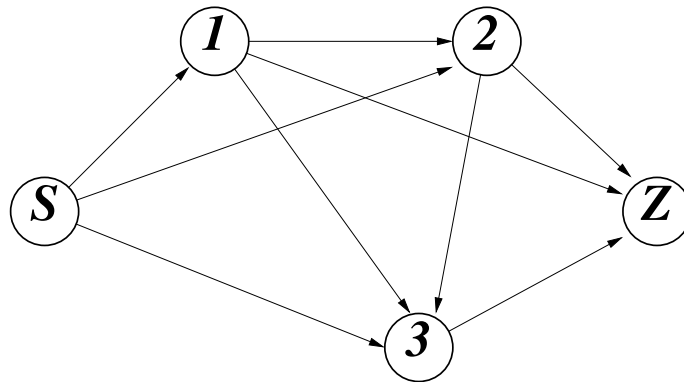


Abbildung 5.3: Die Verbindungen im Netzwerk – wohin kann man von wo aus kommen. Der *gerichtete* Graph bedeutet, daß es keine Schleifen gibt.

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

Beispiel 5.5 (Maximaler Transport oder “Fluß” im Netzwerk) Das Netzwerk aus Abb. 5.3 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 Verkehrsmittel¹³⁸ wie in Abb. 5.4 dargestellt sind, und wie muß man die Fahrgäste auf die einzelnen Verkehrsmittel verteilen?

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 Nebenbe-

¹³⁷In der Sprache der diskreten Mathematik: ein gerichteter Graph.

¹³⁸Sagen wir in der Einheit “100 Fahrgäste”.

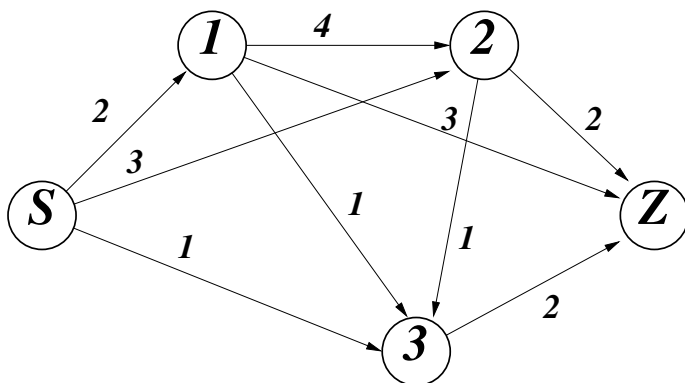


Abbildung 5.4: Die Kapazitäten der einzelnen Kanten des Netzwerk aus Abb. 5.3.

dingungen:

$$\begin{array}{rcl}
 x_{01} & & \leq 2 \\
 x_{02} & & \leq 3 \\
 x_{03} & & \leq 1 \\
 x_{12} & & \leq 4 \\
 x_{13} & & \leq 1 \\
 x_{14} & & \leq 3 \\
 x_{23} & & \leq 1 \\
 x_{24} & & \leq 2 \\
 x_{34} & & \leq 2
 \end{array} \tag{5.1}$$

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

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

schreiben können: Die Summe über die x_{jk} ist je gerade die Menge, die in den Knoten k hineingeführt 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. 5.4 die Nebenbedingungen

$$\begin{array}{rccccrcr}
 x_{01} & & -x_{12} & -x_{13} & -x_{14} & & = 0 \\
 x_{02} & & +x_{12} & & & -x_{23} & -x_{24} & = 0 \\
 x_{03} & & & +x_{13} & & +x_{23} & & -x_{34} = 0
 \end{array} \tag{5.2}$$

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

$$\begin{array}{rccccrcr}
 -x_{01} & -x_{02} & -x_{03} & & & +t & = 0 \\
 & & & x_{14} & +x_{24} & +x_{34} & -t & = 0
 \end{array} \tag{5.3}$$

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 (5.2) und eq:NetFlowNB3, 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 (5.1) 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
```

Die Lösung ist in Abb. 5.5 dargestellt. Man sieht ihr ein typisches Phänomen von Optimallösungen von Netzwerkproblemen an: Alle Kanten, die aus S herausführen, sind *saturiert*, also voll belegt.

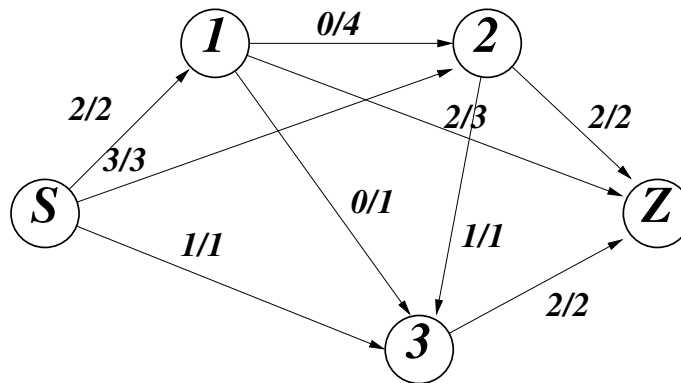


Abbildung 5.5: Fluß der Optimallösung im Vergleich zu deren Kapazität.

Man kann auch allgemeinere Netzwerkprobleme auf diese Art und Weise angehen, indem man für n Knoten (Start und Ziel mitgezählt!) normalerweise eine *Verbindungsmatrix*

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

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 vorgeschrieben¹³⁹. 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

$$\sum_{j=1}^n v_{jk} x_{jk} \geq \sum_{j=1}^n v_{kj} x_{kj}, \quad k = 1, \dots, n.$$

Übrigens kann man da auch großzügiger sein, indem man lediglich

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

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

5.5 Spieltheorie und die zwei Phasen

Auch für die Lösung von spieltheoretischen Problemen ist der Simplexalgorithmus sehr hilfreich. Dabei basieren die Überlegungen auf einer Gleichgewichtsaussage [7] aus dem Jahre 1928, dem sogenannte *Minimax-Theorem*.

¹³⁹Man denke nur an Einbahnstraßen.

Satz 5.6 Es gibt immer optimale gemischte Strategien p^*, q^* für die beiden Spieler, so daß

$$v := x(p^*, q^*) = \max_p \min_q x(p, q) = \min_q \max_p x(p, q). \quad (5.4)$$

Dieser Satz ist wirklich bemerkenswert da, wie wir ja bei den Sattelpunkten der Spiele schon schmerzlich merken mussten, Maximierungs- und Minimierungsprozesse nicht 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 nachweisen¹⁴⁰, daß

$$\min_q x(p, q) = \min_{k=1, \dots, n} \sum_{j=1}^m x_{jk} p_j \quad \text{und} \quad \max_p x(p, q) = \max_{j=1, \dots, m} \sum_{k=1}^n x_{jk} q_k \quad (5.5)$$

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

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

wobei wir

$$X := \begin{bmatrix} x_{jk} & j = 1, \dots, m \\ & k = 1, \dots, n \end{bmatrix}$$

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

$$X^T p^* \geq v \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}, \quad Xq^* \leq v \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}, \quad (5.6)$$

also

$$\left[\begin{array}{c|c|c} X^T & 0 & \begin{matrix} -1 \\ \vdots \\ -1 \end{matrix} \\ \hline 0 & -X & \begin{matrix} 1 \\ \vdots \\ 1 \end{matrix} \end{array} \right] \begin{bmatrix} p^* \\ q^* \\ v \end{bmatrix} \geq 0. \quad (5.7)$$

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

$$\sum_{j=1}^m p_j^* = 1 \quad \Rightarrow \quad \begin{bmatrix} 1 & \dots & 1 \\ -1 & \dots & -1 \end{bmatrix} p^* \geq \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

¹⁴⁰Das Zauberwort heißt *Konvexität*, ansonsten siehe [8] oder [12].

und

$$\sum_{k=1}^n q_k^* = 1 \quad \Rightarrow \quad \begin{bmatrix} 1 & \dots & 1 \\ -1 & \dots & -1 \end{bmatrix} q^* \geq \begin{bmatrix} 1 \\ -1 \end{bmatrix}.$$

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

$$\left[\begin{array}{ccc|ccc} -X^T & & & & & 1 \\ & & & & & \vdots \\ & & & & & 1 \\ \hline & & & & & -1 \\ 0 & & X & & & \vdots \\ & & & & & -1 \\ \hline -1 & \dots & -1 & & & \\ 1 & \dots & 1 & & & 0 \\ \hline & & & -1 & \dots & -1 \\ & & & 1 & \dots & 1 \\ & & & & & 0 \end{array} \right] \begin{bmatrix} p^* \\ q^* \\ v \end{bmatrix} \leq \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \hline 0 \\ \vdots \\ 0 \\ \hline -1 \\ 1 \\ \hline -1 \\ 1 \end{bmatrix}. \quad (5.8)$$

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* (5.8) erfüllen müssen und daß alles, was das Ungleichungssystem erfüllt, auch Optimalstrategie ist¹⁴¹, wir müssen also “nur” eine Lösung von (5.8) 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 5.2 und Programm 5.3 angegeben, das erste stellt die Nebenbedingungsmatrix auf, das andere verwendet Phase I und extrahiert die Werte der Variablen.

Beispiel 5.7 (Stein, Schere, Papier formal) Zurück zu unserem klassischen Beispiel. Hier können wir die Nebenbedingungen sogar noch explizit angeben, nämlich

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

und mit Octave erhalten wir das Ergebnis

¹⁴¹Die optimale Strategie muß nicht eindeutig sein und tatsächlich gibt es immer unendlich viele Optimalstrategien sobald nur zwei voneinander verschiedene Optimalstrategien existieren!

```

## GameStratBed.m (Optmierung fuer Hoerer aller Fachbereiche)
## -----
## 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 5.2 GameStratBed.m: Bestimmung der Nebenbedingungsmatrix für die optimalen Strategien nach (5.8).

```

## GameOptStrat.m (Optmierung fuer Hoerer aller Fachbereiche)
## -----
## 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 = SimPhaseI( 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
end

```

Programm 5.3 GameOptStrat.m: Bestimmung der optimalen Strategie und des Wertes unter Verwendung von Phase I.

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

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

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.

*Uns ist in alten mæren
wunders viel geseit
von Helden lobebæren
von grôzer arebeit*

Das Nibelungenlied

Literatur

5

- [1] P. Ablay, *Optimieren mit Evolutionsstrategien*, Computer-Anwendungen, Spektrum der Wissenschaft: Verständliche Forschung, Spektrum-Verlag, 1989, pp. 162–174.
- [2] 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.
- [3] D. Cox, J. Little, and D. O’Shea, *Using algebraic geometry*, Graduate Texts in Mathematics, vol. 185, Springer Verlag, 1998.
- [4] G. Dueck, T. Scheuer, and H.-M. Wallmeier, *Toleranzschwelle und Sintflut: neue Ideen zur Optimierung*, Spektrum der Wissenschaft **1993/3** (1993), 42–51.
- [5] S. I. Gass, *An illustrated guide to linear programming*, McGraw-Hill, 1970, Republished by Dover, 1990.
- [6] R. W. Hamming, *Digital filters*, Prentice-Hall, 1989, Republished by Dover Publications, 1998.
- [7] J. von Neumann, *Zur Theorie der Gesellschaftsspiele*, Math. Annalen **100** (1928), 295–320.
- [8] J. von Neumann and O. Morgenstern, *Theory of games and economic behavior*, sixth paperback printing, 1990 ed., Princeton University Press, 1944.
- [9] Pschyrembel, *Klinisches wörterbuch*, 257 ed., Walter de Gruyter & Co, 1994.
- [10] T. Sauer, *Einführung in die Numerische Mathematik für Hörer aller Fachbereiche*, Vorlesungsskript, Justus-Liebig-Universität Gießen, 2000, <http://www.math.uni-giessen.de/tomas.sauer>.

- [11] ———, *Interpolation in geographischen Informationssystemen. Mathematischer Hintergrund*, Vorlesungsskript, Justus–Liebig–Universität Gießen, 2002, <http://www.math.uni-giessen.de/tomas.sauer>.
- [12] ———, *Spieltheorie*, Vorlesungsskript, Justus–Liebig–Universität Gießen, 2005, <http://www.math.uni-giessen.de/tomas.sauer>.
- [13] H. R. Schwarz, *Numerische Mathematik*, B. G. Teubner, Stuttgart, 1988.
- [14] A. Steger, *Diskrete Strukturen 1. Kombinatorik – Graphentheorie – algebra*, Springer, 2001.
- [15] The MuPAD Group, *MuPAD*, <http://www.mupad.org>, 2003.
- [16] J.D. Williams, *The complete strategist. being a primer on the theory of games on strategy*, Dover Publications, 1986, Reprint. Originally Mc–Graw–Hill, 1966.
- [17] M. Woitschach, *Gödel, götzen und computer. eine kritik der unreinen vernunft*, Horst Pöller Verlag, Stuttgart, 1986.

- Abbruchbedingungen, 51
- Ableitung
 - dritte, 43
 - erste, 58
 - partielle, 26, 27
 - Richtungs-, 25, 26, 27, 31, 45, 60
 - totale, *siehe* Gradient 27
 - zweite, 33, 40, 40, 59, 62
- Abstieg
 - steilster, 53
- Anstieg
 - steilster, 27, 29
- Barriereterm, 46, 47
- Bedingung
 - Armijo-, 60, 62
 - Powell-, 60
 - Wolfe-, 60
 - starke, 61
- Diätproblem, 84
- Differenzenquotient, 23
- Ecke, 66
 - benachbarte, 67
 - Lösungs-, 68
 - zulässige, 66, 71–73
- Eigenvektor, 41
- Eigenwert, 41, 42, 54, 55
- Einheitsvektoren, 26
- Erhaltung, 93
- Extremum
 - globales, 29, 31, 35, 52, 64
 - lokales, 29, 30, 31–35, 37, 42, 51, 64
 - unter Nebenbedingungen, 45
- Flat Spot, 32
- Funktion
 - differenzierbare, 23, 31
 - konkave, 35
 - konstante, 22
 - konvexe, 35
 - lineare, 35
 - quadratische, 34, 38, 38, 52, 62
 - stetige, 22
 - vektorwertige, 39
- Gleichungssystem
 - lineares, 59
 - nichtlineares, 58
- Gradient, 27, 32, 33, 39, 45, 53, 58
- Graph, 15, 92
- Grenzwert, 25
- Halbraum, 8
- Hyperebene, 66
- Information
 - lokale, 49
- Interpolation, 62
- Kalorien, 84
- Konvergenz
 - lokale, 63
- Krümmung, 33, 35–37, 41
 - negative, 35, 41
 - positive, 35, 41
- Kuhn–Tucker–Bedingungen, 45
- Lagrange–Multiplikatoren, 45
- LANCELOT, 62
- Linearisierung, 22, 23
- Lokalität, 29
- Matrix
 - Auszahlungs-, 16

- Diagonal-, 54
- Gewinn-, 73
- Hesse-, 40, 42, 58, 59
- indefinite, 41, 42
- inverse, 59
- Kosten-, 87
- negativ definite, 41, 42
- negative, 41
- positiv definite, 41, 42, 53
- positive, 41, 41
- symmetrische, 38, 39, 42, 53, 59
- transponierte, 38
- Transport-, 90
- Verbindungs-, 95
- zufällige, 53
- Maximum, 14
 - globales, 45
- Menge
 - zulässige, 4
- Minimum, 14
 - lokales, 30, 45
 - Rand-, 45
- Modellproblem, 52, 56, 62
- MuPAD, 45
- Nebenbedingung, 4
- Nebenbedingungen
 - implizite, 46
- Netzwerk, 92
- Netzwerkfluß, 92
- Niacin, 84
- NP-vollständig, 16
- Nullstelle, 58
- Nullsummenspiel, 16
- Optimierungsproblem, 5
 - diskretes, 6
 - erweitertes, 73
 - ganzzahliges, 71
 - kontinuierliches, 6
 - lineares, 6, 65
 - Normalform, 65
 - nichtlineares, 6
 - unbeschränktes, 85
 - univariates, 59
- Ordnung
 - lexikographische, 88
- Parabel, 34, 35, 52
- Paraboloid, 38
- Pivotspalte, 68
- Pivotzeile, 68, 69
- Polyeder, 9
- Problem
 - Transport-, 71
- Programm
 - ganzzahliges, 9
 - kombinatorisches, 14
 - lineares, 7
 - quadratisches, 12
- Punkt
 - innerer, 29, 43
 - kritischer, 45
 - Rand-, 29–31, 43
 - zulässiger, 66
- quadratische Form, 41
- Randextremum, 45
- Rechtecksregel, 69
- Regenschirme, 19
- Regression, 12
 - lineare, 12
- Richtung
 - Abstiegs-, 50, 51, 53, 59
 - Anstiegs-, 27
 - Aufstiegs-, 28, 29, 62
 - konjugierte Gradienten, 56
 - Newton-, 58, 59
 - zufällige, 55
 - zulässige, 46
- Richtungsableitung, *siehe* Ableitung 25
- Richtungskegel, 46
- Rucksackproblem, 15
- Sattelpunkt, 18, 33, 42, 43
- Schrittweite, 50, 59, 61
 - exakte, 52, 59

- Schuhfabrik, 8
- Simplextableau, 65, 68, 74
 - erweitertes, 73
- Spieltheorie, 16
- Startwert, 49, 54, 64
- Strafterm, 46, 47
- Strategie, 16
 - gemischte, 19, 19, 21, 72
 - kooperative, 17
 - reine, 18
- Straßenkarte, 14

- Tangentialebene, 32
- Taylorformel, 36, 43, 62
- Textaufgabe, 84
- Thianin, 84
- Transportproblem, 87, 91
 - ganzzahliges, 10
- Trust Region, 62

- Umgebung, 22, 29
- Ungleichung, 5

- Variable
 - abhängige, 66, 69
 - Basis-, 66, 69, 74
 - zusätzliche, 94
- Verfahren
 - Abstiegs-, 47
 - Broyden-, 59
 - grafisches, 8
 - Hau-Ruck, 5
 - konjugierte Gradienten, 56
 - Monte-Carlo, 5
 - Newton-, 58
 - Simplex-, 9, 65, 67, 69
 - Phase 1, 73
 - Trust-Region-, 62, 63

- Wendepunkt, 33

- Zielfunktion, 4
 - lineare, 6
 - quadratische, 6
- Zuordnungsproblem, 90