



METHODEN DER OPTIMIERUNG UND OPTIMALEN STEUERUNG

Wintersemester 2010/2011

Prof. Dr.-Ing. Knut Graichen

Institut für Mess-, Regel- und Mikrotechnik
Fakultät für Ingenieurwissenschaften und Informatik
Universität Ulm

“Cum enim mundi universi fabrica sit perfectissima atque a Creatore sapientissimo absoluta, nihil omnino in mundo contingit, in quo non maximi minimive ratio quaequam eluceat; quamobrem dubium prorsus est nullum, quin omnes mundi effectus ex causis finalibus ope methodi maximorum et minimorum aequè feliciter determinari queant, atque ex ipsis causis efficientibus.”

“Da nämlich der Bau der gesamten Welt höchst vollkommen und vom weisesten Schöpfer vollendet wurde, geschieht überhaupt nichts in der Welt, bei dem nicht das Verhältnis eines Maximums oder Minimums hervorleuchtet; deshalb gibt es durchaus keinen Zweifel, dass alle Wirkungen der Welt, die auf Finalursachen beruhen, mit Hilfe der Methode der Maxima und Minima gleich erfolgreich bestimmt werden können wie aus den bewirkten Ursachen selbst.”

Leonhard Euler, *Variationsrechnung*, 1744.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Statische Optimierungsprobleme	1
1.1.1	Mathematische Formulierung	2
1.1.2	Beispiele	2
1.2	Dynamische Optimierungsprobleme	6
1.2.1	Mathematische Formulierung	6
1.2.2	Beispiele	6
2	Grundlagen der Optimierung	11
2.1	Lokale/globale Minima und zulässige Menge	11
2.2	Gradient und Hessematrix	13
2.3	Konvexität	15
2.3.1	Konvexe Mengen	15
2.3.2	Konvexe Funktionen	15
2.3.3	Konvexe Optimierungsprobleme	18
3	Statische Optimierung ohne Beschränkungen	21
3.1	Notwendige Optimalitätsbedingungen	21
3.2	Hinreichende Optimalitätsbedingungen	23
3.3	Numerische Lösung mit Liniensuchverfahren	24
3.3.1	Allgemeiner Algorithmus	24
3.3.2	Wahl der Schrittweite	26
3.3.2.1	Abstiegs- und Krümmungsbedingungen	26
3.3.2.2	Backtracking-Verfahren	28
3.3.2.3	Intervallschachtelungsverfahren (“Goldener Schnitt”)	28
3.3.3	Wahl der Suchrichtung	30
3.3.3.1	Gradientenverfahren	30
3.3.3.2	Konjugierte-Gradienten-Verfahren	32
3.3.3.3	Newton-Verfahren	32

3.3.3.4	Quasi-Newton-Verfahren	34
3.4	Weitere numerische Verfahren	35
3.4.1	Methode der Vertrauensbereiche	35
3.4.2	Direkte Suchverfahren	37
3.5	Beispiel: Rosenbrock's "Bananenfunktion"	40
4	Statische Optimierung mit Beschränkungen	45
4.1	Einführende Beispiele	45
4.1.1	Eine Gleichungsbeschränkung	45
4.1.2	Eine Ungleichungsbeschränkung	47
4.1.3	Zwei Ungleichungsbeschränkungen	49
4.2	Optimalitätsbedingungen	50
4.2.1	Beschränkungsqualifikation	50
4.2.2	Optimalitätsbedingungen 1. Ordnung	51
4.2.3	Optimalitätsbedingungen 2. Ordnung	53
4.2.4	Interpretation der Lagrange-Multiplikatoren	55
4.3	Lineare Optimierung	56
4.3.1	Standardform-Darstellung	56
4.3.2	Optimalitätsbedingungen	57
4.3.3	Simplex-Algorithmus	57
4.3.3.1	Iterationsphase	59
4.3.3.2	Initialisierungsphase	62
4.3.4	Interior-Point-Verfahren	63
4.3.4.1	Prinzip des Verfahrens	63
4.3.4.2	Zusammenhang mit Barrieremethoden	65
4.3.5	Beispiel: Transportnetz	66
4.4	Quadratische Optimierung	67
4.4.1	Optimalitätsbedingungen	68
4.4.2	Gleichungsbeschränkte quadratische Probleme	69
4.4.3	Aktive-Restriktionen-Verfahren	69
4.4.4	Weitere numerische Verfahren	73
4.5	Nichtlineare Optimierung	74
4.5.1	Sequentielle quadratische Programmierung	74
4.5.1.1	Allgemeiner Algorithmus	74
4.5.1.2	Wahl der Schrittweite	75
4.5.1.3	Beispiel: Optimale Auslegung von Reglerparametern	76

4.5.2	Weitere numerische Verfahren	78
4.6	Software-Übersicht	80
5	Dynamische Optimierung	82
5.1	Allgemeine Aufgabenstellung	82
5.2	Einführung in die Variationsrechnung	84
5.2.1	Euler-Lagrange-Gleichungen	84
5.2.2	Beispiel: Galileo's hängende Kette	87
5.3	Unbeschränkte Optimalsteuerungsprobleme	89
5.3.1	Anwendung der Variationsrechnung	89
5.3.1.1	Berücksichtigung der Systemgleichungen	89
5.3.1.2	Berücksichtigung einer freien Endzeit	91
5.3.2	Optimalitätsbedingungen	92
5.3.3	Beispiel zur Betrachtung einer freien Endzeit	95
5.3.4	Allgemeine Vorgehensweise und singulärer Fall	98
5.3.5	Interpretation der adjungierten Variablen	99
5.3.6	Erweiterung auf allgemeine Randbedingungen	101
5.4	Optimierung linearer Systeme mit quadratischem Kostenfunktional	102
5.4.1	Riccati-Differentialgleichung	102
5.4.2	Beispiel: 2FHG-Regelung zum Aufschwingen des Einfachpendels	103
5.4.3	Algebraische Riccati-Gleichung (Sonderfall $t_f \rightarrow \infty$)	108
5.5	Optimalsteuerungsprobleme mit Eingangsbeschränkungen	110
5.5.1	Pontryagin's Maximumprinzip	110
5.5.2	Allgemeine Vorgehensweise	112
5.5.3	Minimierung der Hamilton-Funktion	113
5.5.3.1	Kostenfunktional mit verbrauchsoptimalem Anteil	113
5.5.3.2	Kostenfunktional mit energieoptimalem Anteil	114
5.5.3.3	Zeitoptimales Kostenfunktional	115
5.5.4	Beispiel: Doppelintegrator	116
5.5.5	Zustandsabhängige Eingangsbeschränkungen	119
6	Numerische Lösung von dynamischen Optimierungsproblemen	121
6.1	Indirekte Verfahren: Lösung der Optimalitätsbedingungen	121
6.1.1	Diskretisierungsverfahren	121
6.1.2	Schießverfahren	123
6.1.3	Berücksichtigung allgemeiner Endbedingungen und freier Endzeit	124

6.1.4	Gradientenverfahren	125
6.2	Direkte Verfahren: Reduktion auf statische Optimierungsprobleme	127
6.2.1	Teildiskretisierung	128
6.2.2	Volldiskretisierung	129
6.3	Beispiel: Navigationsproblem in der Schifffahrt	131
6.3.1	Indirekter Lösungsweg	131
6.3.1.1	Realisierung eines Richtungswechsels	131
6.3.1.2	Diskretisierungsverfahren	132
6.3.1.3	Schießverfahren	134
6.3.2	Direkter Lösungsweg	135
6.3.2.1	Realisierung eines Ausweichmanövers	135
6.3.2.2	Volldiskretisierung mit AMPL/IPOPT	136
6.4	Software-Übersicht	139

Anhang – Lösungen zu den Aufgaben (nur Masterskript)

A.1	Aufgaben in Kapitel 2	1
A.2	Aufgaben in Kapitel 3	3
A.3	Aufgaben in Kapitel 4	7
A.4	Aufgaben in Kapitel 5	14
A.5	Aufgaben in Kapitel 6	18

Kapitel 1

Einleitung

Unter dem Begriff der Optimierung wird im Allgemeinen das Suchen einer optimalen oder (zumindest guten) Lösung zu einem Problem verstanden. Das Feld der Optimierung ist in vielen Feldern von Industrie und Wirtschaft vertreten. Einige Beispiele sind

- Wirtschaft und Finanzen: z.B. Portfolio–Optimierung oder Unternehmensentwicklung (Maximierung des Gewinns durch optimale Verteilung von Investitionen, Ausgaben, Anlagen, etc.).
- Prozess- und Produktoptimierung: z.B. Ablaufplanung von Prozessen, Minimierung von Herstellungskosten, energieeffizienter Betrieb.
- Regelungstechnik allgemein: z.B. Berechnung optimaler Bahnen in der Robotik, Entwurf optimaler Steuerungen und Regelungen, Parameterschätzung, optimale Reglerauslegung.

Die mathematische Formulierung dieser Probleme führt auf ein Optimierungsproblem. Generell wird dabei zwischen *statischen* und *dynamischen Optimierungsproblemen* unterschieden, deren Unterschied im Folgenden erläutert werden soll.

- *Statisches Optimierungsproblem*: Minimierung einer Funktion mit Optimierungsvariablen, die Elemente des Euklidischen Raumes sind.
- *Dynamisches Optimierungsproblem*: Minimierung eines Funktionals, bei dem die Optimierungsvariablen Elemente des Hilbert–Raumes sind (z.B. Zeitfunktionen).

In diesem Kapitel soll zunächst an einigen Beispielen ein Eindruck sowie der prinzipielle Unterschied zwischen der statischen und dynamischen Optimierung vermittelt werden.

1.1 Statische Optimierungsprobleme

Unter einem *statischen Optimierungsproblem* wird das Minimieren einer Funktion $f(\mathbf{x})$ unter Berücksichtigung gewisser Nebenbedingungen verstanden, bei dem die Optimierungsvariablen Elemente $\mathbf{x}$ des Euklidischen Raumes $\mathbb{R}^n$ sind.

1.1.1 Mathematische Formulierung

Die Standardformulierung eines statischen Optimierungsproblems wird wie folgt angegeben:

$$\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) \quad \text{Kostenfunktion} \quad (1.1)$$

$$\text{u.B.v. } g_i(\mathbf{x}) = 0, \quad i = 1, \dots, p \quad \text{Gleichungsbeschränkungen} \quad (1.2)$$

$$h_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, q \quad \text{Ungleichungsbeschränkungen} \quad (1.3)$$

Ist ein Optimierungsproblem ohne die Ungleichungs- und Gleichungsbeschränkungen (1.3), (1.2) gegeben, spricht man von einem *unbeschränkten Optimierungsproblem*. Im allgemeinen Fall, d.h. unter Berücksichtigung der Nebenbedingungen (1.2)–(1.3), handelt es sich um ein *beschränktes Optimierungsproblem*.

Im Laufe der Jahrzehnte hat sich die Formulierung als Minimierungsproblem (1.1) standardisiert. Analog dazu kann ein Maximierungsproblem ebenfalls in der Notation (1.1) geschrieben werden:

$$\max_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) = \min_{\mathbf{x} \in \mathbb{R}^n} -f(\mathbf{x}).$$

Neben der Bezeichnung statische Optimierung werden häufig auch die Begriffe *Mathematische Programmierung* oder *Endlich-Dimensionale Optimierung* verwendet.

Die Bezeichnung “Programmierung” ist historisch begründet. Die ersten Anwendungen der Optimierung waren militärischer Natur, bei denen es um Aktionspläne ging:

“It is a word used by the military for a plan, a schedule, or in general for a program of actions. This is exactly what it means for computers: a program of actions for the computer to execute.”

George Danzig

Begründer der Linearen Programmierung

Unterschieden werden bei statischen Optimierungsproblemen häufig folgende Klassen:

- *Lineare Programmierung*: Die Kostenfunktion und Beschränkungen sind linear.
- *Quadratische Programmierung*: Die Kostenfunktion ist quadratisch, während die Beschränkungen linear sind.
- *Nichtlineare Programmierung*: Die Kostenfunktion oder mindestens eine Beschränkung ist nichtlinear.
- *Integer-Programmierung*: Alle Variablen sind diskret.
- *Mixed-Integer-Programmierung*: Kontinuierliche und diskrete Variablen treten auf.

1.1.2 Beispiele

Insbesondere die *lineare Programmierung* wird häufig bei wirtschaftlichen Fragestellungen, wie Produktions-, Planungs- oder Investitionsproblemen, eingesetzt. Das folgende Beispiel ist ein stark vereinfachtes Beispiel einer Portfolio-Optimierung.

Beispiel 1.1 (Portfolio-Optimierung) Ein Anleger möchte 10.000 Euro gewinnbringend investieren und hat die Auswahl zwischen drei Aktienfonds mit unterschiedlicher Gewinnerwartungen und Risikoeinstufung:

Fonds	erwarteter Gewinn/Jahr	Risikoeinstufung
Fonds A	10%	4
Fonds B	7%	2
Fonds C	4%	1

Der Anleger möchte nach einem Jahr mindestens 600 Euro Gewinn erzielen. Andererseits möchte er sein Geld eher konservativ anlegen, d.h. er möchte mindestens 4.000 Euro in Fonds C investieren und das Risiko minimieren. Wie muss der Anleger die 10.000 Euro verteilen, damit diese Kriterien erfüllt werden?

Zunächst werden die Optimierungsvariablen x_1, x_2, x_3 eingeführt, die den prozentualen Anteil der investierten 10.000 Euro an den jeweiligen Fonds A, B, C kennzeichnen. Dabei kann x_3 durch die Beziehung

$$x_3 = 1 - x_1 - x_2$$

ersetzt werden. Der geforderte Mindestgewinn von 600 Euro lässt sich als die Beschränkung

$$10.000[0.1x_1 + 0.07x_2 + 0.04(1 - x_1 - x_2)] \geq 600 \quad \Rightarrow \quad 6x_1 + 3x_2 \geq 2 \quad (1.4)$$

ausdrücken. Die Mindestanlage von 4.000 Euro in Fonds C führt zu

$$10.000(1 - x_1 - x_2) \geq 4.000 \quad \Rightarrow \quad x_1 + x_2 \leq 0.6 \quad (1.5)$$

Des Weiteren muss $x_1, x_2, x_3 \geq 0$ gelten. Das Ziel ist die Minimierung des Anlagerisikos, was sich durch die Funktion

$$f(\mathbf{x}) = 4x_1 + 2x_2 + (1 - x_1 - x_2) = 1 + 3x_1 + x_2 \quad (1.6)$$

ausdrücken lässt. Somit kann das statische Optimierungsproblem in der Form

$$\min_{\mathbf{x} \in \mathbb{R}^2} \quad f(\mathbf{x}) = 1 + 3x_1 + x_2 \quad (1.7a)$$

$$\text{u.B.v.} \quad 6x_1 + 3x_2 \geq 2 \quad (1.7b)$$

$$x_1 + x_2 \leq 0.6 \quad (1.7c)$$

$$x_1 + x_2 \leq 1 \quad (1.7d)$$

$$x_1, x_2 \geq 0 \quad (1.7e)$$

geschrieben werden. Abbildung 1.1 stellt die einzelnen Beschränkungen sowie den zulässigen Bereich grafisch dar. Aus dem Verlauf der Höhenlinien $f(\mathbf{x}) = \text{konst.}$ der Kostenfunktion (1.7a) ist direkt ersichtlich, dass der Punkt $\mathbf{x}^*$ des zulässigen Bereiches mit dem niedrigsten Wert von $f(\mathbf{x})$ an der Ecke $\mathbf{x}^*$ liegt. Somit ergibt sich für die optimale Verteilung der 10.000 Euro auf die einzelnen Fonds

$$x_1^* = \frac{1}{15}, \quad x_2^* = \frac{8}{15}, \quad x_3^* = \frac{6}{15}. \quad (1.8)$$

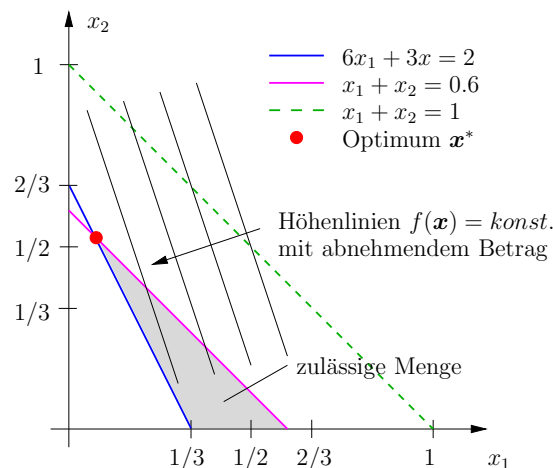


Abbildung 1.1: Veranschaulichung der Portfolio-Optimierung in Beispiel 1.1.

Das folgende (akademische) Beispiel der quadratischen Programmierung soll den Einfluss von Beschränkungen auf eine optimale Lösung verdeutlichen.

Beispiel 1.2 Betrachtet wird das (zunächst) unbeschränkte Problem

$$\min_{\mathbf{x} \in \mathbb{R}^2} f(\mathbf{x}) = (x_1 - 2)^2 + (x_2 - 1)^2. \quad (1.9)$$

Die Höhenlinien $f(\mathbf{x}) = \text{konst.}$ der Funktion $f(\mathbf{x})$ sind in Abbildung 1.2 in Abhängigkeit der beiden Optimierungsvariablen $\mathbf{x} = [x_1, x_2]^T$ dargestellt. Es ist direkt ersichtlich, dass das Minimum $f(\mathbf{x}^*) = 0$ an der Stelle $\mathbf{x}^* = [2, 1]^T$ auftritt.

Um den Einfluss verschiedener Beschränkungen zu untersuchen, wird zunächst eine zusätzliche Gleichungsbeschränkung der Form (1.2) betrachtet:

$$g(\mathbf{x}) = x_2 - 2x_1 = 0. \quad (1.10)$$

Die Gleichungsbeschränkung entspricht einer algebraischen Zwangsbedingung, wodurch lediglich noch eine Optimierungsvariable frei wählbar ist. Geometrisch interpretiert bedeutet dies, dass eine mögliche Lösung auf der Geraden liegen muss, die durch (1.10) definiert wird, siehe Abbildung 1.3. Die optimale Lösung liegt dabei auf dem Berührungspunkt der Geraden $g(\mathbf{x}) = 0$ mit der Höhenlinie $f(\mathbf{x}) = 1.8$.

Anstelle der Gleichungsbeschränkung (1.10) wird nun die Ungleichungsbeschränkung

$$h_1(\mathbf{x}) = x_1 + x_2 - 2 \leq 0 \quad (1.11)$$

betrachtet, wodurch sich die Menge der zulässigen Punkte $\mathbf{x} = [x_1, x_2]^T$ auf die Region unterhalb der Geraden $h_1(\mathbf{x}) = 0$ beschränkt (Abbildung 1.4). Das Optimum $f(\mathbf{x}^*) = 0.5$ an der Stelle $\mathbf{x}^* = [1.5, 0.5]^T$ befindet sich an der Grenze des zulässigen Bereiches und liegt, wie im vorherigen Szenario, auf einer Höhenlinie, die die Gerade $h_1(\mathbf{x}) = 0$ tangential berührt.

Zusätzlich zur ersten Ungleichungsbeschränkung (1.11) soll eine weitere Ungleichung der Form

$$h_2(\mathbf{x}) = x_1^2 - x_2 \leq 0, \quad (1.12)$$

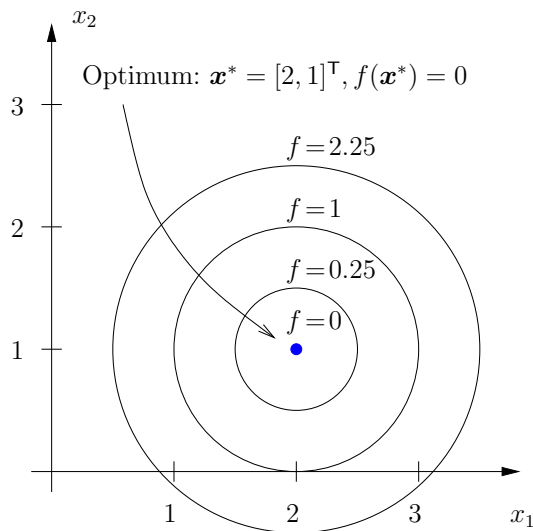


Abbildung 1.2: Geometrische Darstellung des unbeschränkten OP (1.9).

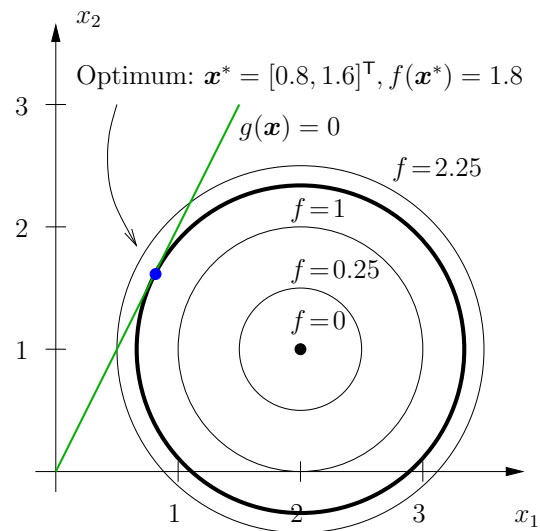


Abbildung 1.3: Geometrische Darstellung des beschränkten OP (1.9), (1.10).

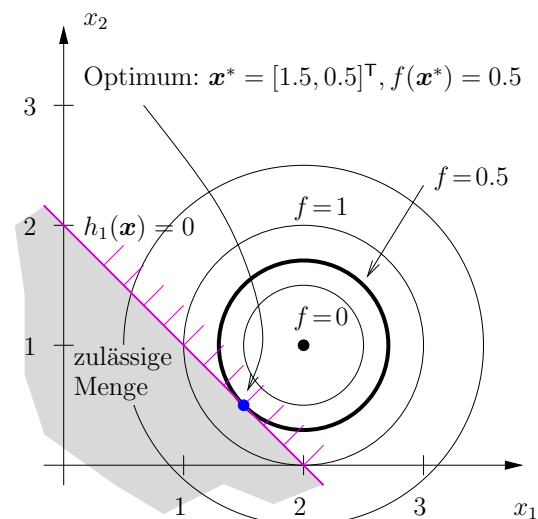


Abbildung 1.4: Geometrische Darstellung des beschränkten OP (1.9), (1.11).

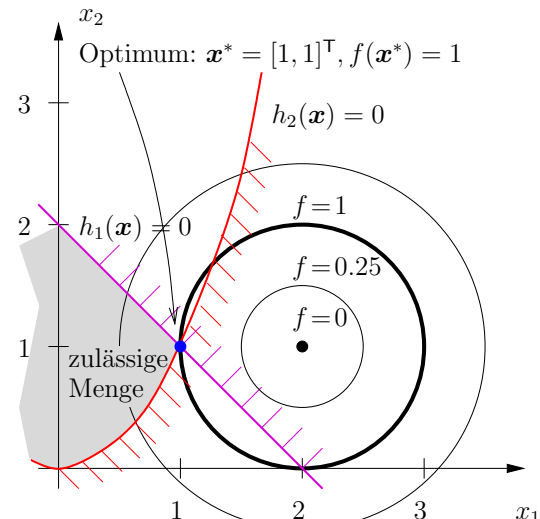


Abbildung 1.5: Geometrische Darstellung des beschränkten OP (1.9), (1.11), (1.12).

betrachtet werden, durch die sich die Menge der zulässigen Punkte weiter verkleinert (Abbildung 1.5). Der optimale Punkt $\mathbf{x} = [1, 1]^T$ mit dem Minimum $f(\mathbf{x}^*) = 1$ liegt nun im Schnittpunkt der Kurven $h_1(\mathbf{x}) = 0$ und $h_2(\mathbf{x}) = 0$, d.h. beide Beschränkungen (1.11) und (1.12) sind aktiv.

Das obige Beispiel 1.2 verdeutlicht den Einfluss von Gleichungs- und Ungleichungsbeschränkungen auf die Lösung (und Lösbarkeit) des Optimierungsproblems (1.1)–(1.3). Für eine systematische Untersuchung von statischen OP sowie Verfahren zur numerischen Lösung von statischen OP sei auf die hinteren Kapitel verwiesen.

1.2 Dynamische Optimierungsprobleme

Bei den Problemstellungen der statischen Optimierung im vorherigen Abschnitt 1.1 stellen die Optimierungsvariablen $\mathbf{x}$ Elemente aus dem Euklidischen Raum $\mathbb{R}^n$ dar. Bei der dynamischen Optimierung handelt es sich darum, Funktionen einer unabhängigen Variablen zu bestimmen. Da es sich bei der unabhängigen Variablen meistens um die Zeit t handelt, wird in diesem Zusammenhang von *dynamischer Optimierung* gesprochen.

1.2.1 Mathematische Formulierung

Die generelle Struktur eines dynamischen Optimierungsproblems lautet

$$\min_{\mathbf{u}(\cdot)} \quad J(\mathbf{u}) = \varphi(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} l(\mathbf{x}(t), \mathbf{u}(t), t) dt \quad \text{Kostenfunktional} \quad (1.13)$$

$$\text{u.B.v.} \quad \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t), \quad \mathbf{x}(t_0) = \mathbf{x}_0 \quad \text{Systemdynamik \& AB} \quad (1.14)$$

$$\mathbf{g}(\mathbf{x}(t_f), t_f) = \mathbf{0} \quad \text{Endbedingungen (EB)} \quad (1.15)$$

$$h_i(\mathbf{x}, \mathbf{u}) \leq 0, \quad i = 1, \dots, q \quad \text{Ungleichungsbeschr.} \quad (1.16)$$

Dabei stellt $\mathbf{u} \in \mathbb{R}^m$ die Eingangsgröße des nichtlinearen Systems (1.14) mit dem Zustand $\mathbf{x} \in \mathbb{R}^n$ dar. Zusätzlich zu den Anfangsbedingungen in (1.14) sind häufig Endbedingungen der Form (1.15) gegeben, um z.B. einen gewünschten Zustand $\mathbf{x}_f$ zur Endzeit t_f zu erreichen (also $\mathbf{g}(\mathbf{x}(t_f), t_f) = \mathbf{x}(t_f) - \mathbf{x}_f$). Ein häufiges Problem in der Praxis ist das Auftreten von Ungleichungsbeschränkungen (1.16), die z.B. die Begrenzung einer Stellgröße oder Sicherheitsschranken eines Zustandes darstellen können.

Die Problemstellung der dynamischen Optimierung besteht nun darin, eine Eingangstrajektorie $\mathbf{u}(t)$, $t \in [t_0, t_f]$ derart zu finden, dass die Zustandstrajektorie $\mathbf{x}(t)$, $t \in [t_0, t_f]$ des dynamischen Systems (1.14), die Endbedingungen (1.15) und Beschränkungen (1.16) erfüllt und gleichzeitig das Kostenfunktional (1.13) minimiert wird. Abhängig davon, ob t_f vorgegeben oder unbekannt ist, spricht man von einer *festen* oder *freien Endzeit* t_f .

Neben der Bezeichnung dynamische Optimierung werden häufig auch die Begriffe *Unendlich-Dimensionale Optimierung*, *Optimalsteuerungsproblem* oder *Dynamische Programmierung* verwendet. Im Folgenden sind einige Beispiele angegeben, um die Problem- und Aufgabenstellung der dynamischen Optimierung zu erläutern.

1.2.2 Beispiele

Beispiel 1.3 (Inverses Pendel) *Ein klassisches Problem in der Regelungstechnik ist das inverse Pendel, das an einem Wagen drehbar befestigt ist. Als Beispielpblem soll*

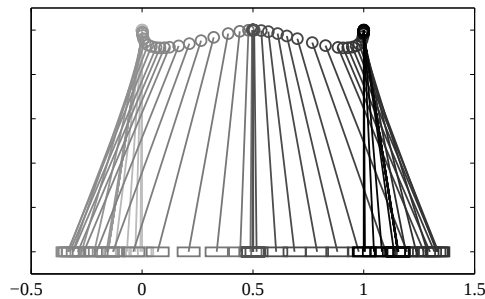


Abbildung 1.6: Momentaufnahmen des Versetzens des inversen Pendels.

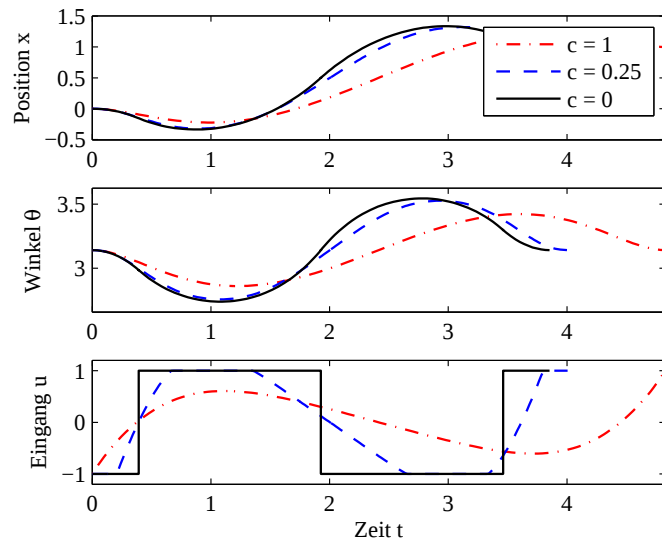


Abbildung 1.7: Optimale Trajektorien für das Versetzen des inversen Pendels.

das seitliche Versetzen des Pendels betrachtet werden:

$$\min_{u(\cdot)} \quad J(u) = \int_0^{t_f} 1 + c u^2 \, dt, \quad (1.17)$$

$$\text{u.B.v.} \quad \begin{bmatrix} 1 & \varepsilon \cos \theta \\ \cos \theta & 1 \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \varepsilon \dot{\theta}^2 \sin \theta + u \\ -\sin \theta \end{bmatrix}, \quad \varepsilon = m/(M + m) \quad (1.18)$$

$$\mathbf{x}(0) = [0, 0, 0, 0]^T, \quad \mathbf{x}(t_f) = [1, 0, 0, 0]^T, \quad (1.19)$$

$$0 \leq u \leq 1. \quad (1.20)$$

Die vereinfachten Bewegungsgleichungen (1.18) für die Zustände $\mathbf{x} = [x, \dot{x}, \theta, \dot{\theta}]^T$ sind normiert. Der Eingang u stellt die am Wagen angreifende Kraft dar und ist durch (1.20) beschränkt. Abbildung 1.6 zeigt exemplarisch das seitliche Versetzen des Pendels, um die Bewegung zu verdeutlichen.

Das Kostenfunktional (1.17) und somit der Charakter des Optimierungsproblems hängt von dem Parameter c ab. Für $c = 0$ ergibt sich die Aufgabe, die Endzeit t_f zu minimieren:

$$J(u) = \int_0^{t_f} 1 \, dt = t_f. \quad (1.21)$$

Für $c > 0$ wird der Eingang u im Kostenfunktional und somit der Aspekt der Energieoptimalität mit berücksichtigt.

Abbildung 1.7 zeigt die optimalen Trajektorien für die Werte $c = 0$, $c = 0.25$ und $c = 1$. Für $c = 0$ weist der Eingang u ein Bang-bang-Verhalten auf, während für $c > 0$ die Steueramplituden kleiner werden und die Versatzzeit t_f zunimmt.

Das inverse Pendel ist ein gutes Beispiel um zu verdeutlichen, dass nicht zu jedem Optimierungsproblem eine Lösung existiert, insbesondere wenn die Endzeit t_f nicht festgelegt ist. Wie in Abbildung 1.7 dargestellt ist, vergrößert sich die Endzeit t_f bei zunehmender

Gewichtung von u^2 im Vergleich zum zeitoptimalen Anteil in dem Kostenfunktional (1.17). Wenn reine Energieoptimalität gefordert würde, d.h.

$$J(u) = \int_0^{t_f} u^2 dt, \quad (1.22)$$

hätte das Optimierungsproblem keine Lösung, da das Versetzen des Pendels dann unendlich langsam mit $t_f \rightarrow \infty$ ablaufen würde.

Beispiel 1.4 (Goddard–Rakete [2, 1]) Ein klassisches Optimierungsproblem aus der Raumfahrt ist die Maximierung der Flughöhe einer Rakete unter dem Einfluss von Luftreibung und Erdbeschleunigung. Dieses Problem wurde von dem amerikanischen Raketenpionier Robert H. Goddard im Jahr 1919 aufgestellt und kann wie folgt (normiert) geschrieben werden:

$$\min_{u(\cdot)} -h(t_f) \quad (1.23)$$

$$\text{u.B.v.} \quad \dot{h} = v, \quad \dot{v} = \frac{u - D(h, v)}{m} - \frac{1}{h^2}, \quad \dot{m} = -\frac{u}{c}, \quad (1.24)$$

$$h(0) = 1, \quad v(0) = 0, \quad m(0) = 1, \quad m(t_f) = 0.6, \quad (1.25)$$

$$0 \leq u \leq 3.5. \quad (1.26)$$

Die Zustandsgrößen sind die Flughöhe h , die Geschwindigkeit v und die Masse m der Rakete. Die Luftreibung $D(h, v)$ hängt über die Funktion

$$D(h, v) = D_0 v^2 \exp[\beta(1 - h)] \quad (1.27)$$

von den Zuständen h und v ab. Die Randbedingungen in (1.25) umfassen die normierten Anfangsbedingungen sowie die Endbedingung für $m(t_f)$, die dem Leergewicht der Rakete ohne Treibstoff entspricht. Der Eingang des Systems ist der Schub u , der innerhalb der Beschränkungen (1.26) liegen muss.

In Abbildung 1.8 sind die optimalen Trajektorien für die Goddard–Rakete dargestellt.¹ Der Schub $u(t)$ ist am Anfang maximal und weist dann einen parabelförmigen Verlauf auf, bevor der Treibstoff verbraucht ist. Dieses Verhalten wird durch den Luftwiderstand $D(h, v)$ hervorgerufen, der mit zunehmender Höhe abnimmt. Es ist somit “optimaler”, im Falle eines hohen Luftwiderstandes nicht mit vollem Schub zu fliegen.

Beispiel 1.5 (Ökonomisches Modell [4, 3]) Ein weiterer Anwendungszweig der dynamischen Optimierung sind wirtschaftliche Prozesse. Das folgende Beispiel beschreibt das Verhalten eines typischen Konsumenten, der Konsum, Freizeit und Bildung über die Lebensdauer maximieren will. Der Bildungsgrad B und das Kapital K eines durchschnittlichen Konsumenten lassen sich durch folgendes Modell beschreiben:

$$\dot{B} = \overbrace{B^\varepsilon u_2 u_3}^{\text{Weiterbildung}} - \overbrace{\delta B}^{\text{Vergessen}}, \quad B(0) = B_0 \quad (1.28)$$

$$\dot{K} = \underbrace{iK}_{\text{Verzinsung}} + \underbrace{Bu_2 g(u_3)}_{\text{Einkommen}} - \underbrace{u_1}_{\text{Konsum}}, \quad K(0) = K_0 \quad (1.29)$$

¹ Die verwendeten Parameterwerte sind $c = 0.5$, $D_0 = 310$, $\beta = 500$.

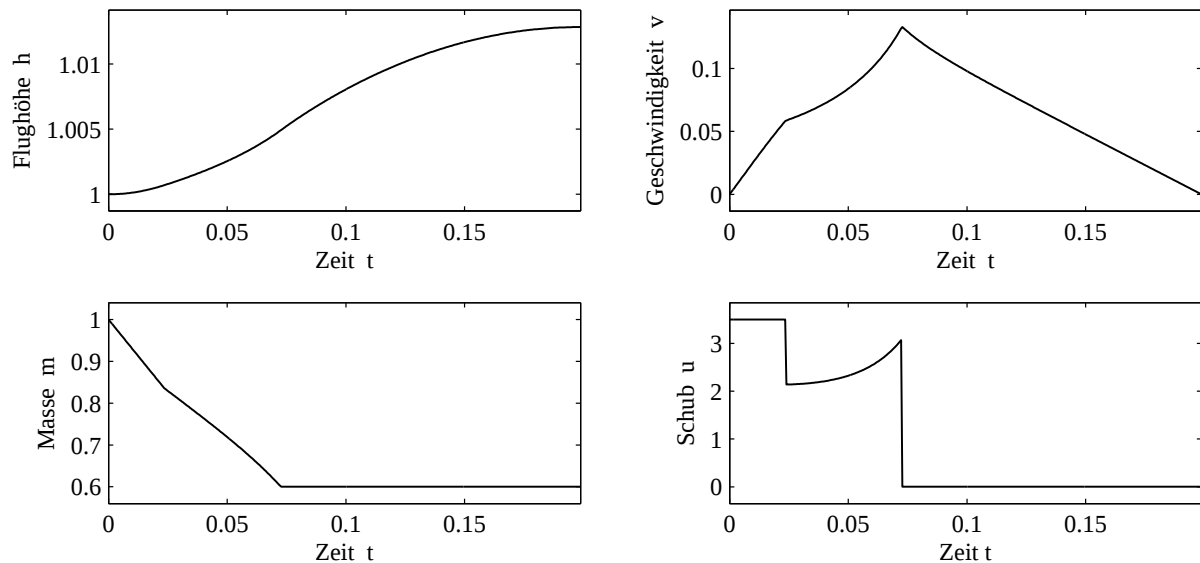


Abbildung 1.8: Trajektorien für die Goddard–Rakete in Beispiel 1.4.

Die Eingangsgrößen sind der Konsum u_1 , der Anteil der Arbeitszeit an der Gesamtzeit u_2 sowie der Anteil der Fortbildungszeit an der Arbeitszeit u_3 . Die Eingänge unterliegen den Beschränkungen

$$u_1 > 0, \quad 0 \leq u_2 \leq 1, \quad 0 \leq u_3 < 1. \quad (1.30)$$

Das Optimierungsziel des Konsumenten ist die Maximierung von Konsum, Freizeit und Bildung über die Lebensdauer von $t_f = 75$ Jahren, was in dem folgenden (zu minimierenden) Kostenfunktional ausgedrückt ist:

$$J(\mathbf{u}) = -K^\kappa(t_f) - \int_{t_0}^{t_f} U(t, u_1, u_2, B) e^{-\rho t} dt. \quad (1.31)$$

Der Integralanteil

$$U(t, u_1, u_2, B) = \alpha_0 u_1^\alpha + \beta_0 (1 - u_2)^\beta + \gamma_0 t B^\gamma \quad (1.32)$$

gewichtet dabei den Konsum u_1 , die Freizeit $1 - u_2$ und den Bildungsgrad B , während der Endwert $-K^\kappa(t_f)$ in (1.31) zusätzlich das Vererbungskapital berücksichtigt.

Die optimalen Zeitverläufe des Bildungsgrades $B(t)$ und des Kapitals $K(t)$ sind in Abbildung 1.9 dargestellt.² Die ersten 17 Jahren stellen die Lernphase dar (d.h. $u_3 = 1$). Daraufhin folgt eine lange Arbeitsphase von 34 Jahren mit einem hohen Maß an Weiterbildung, bevor in den nächsten 10 Jahren (52.–61. Lebensjahr) eine reine Arbeitsphase mit zusätzlich reduzierter Arbeitszeit u_2 stattfindet. Ab dem 62. Lebensjahr setzt der Ruhestand ein. Der Bildungsgrad B ist besonders hoch im Alter von 30–60 Jahren. Das Kapital K ist negativ während der ersten Lebenshälfte, was der Aufnahme eines Kredites entspricht. Im Laufe des Lebens wird dies aber durch das steigende Einkommen kompensiert.

²Die verwendeten Parameterwerte sind $\alpha = -1$, $\alpha_0 = -1$, $\beta = -0.5$, $\beta_0 = -1$, $\gamma = 0.2$, $\gamma_0 = 5$, $\kappa = 0.2$, $\rho = 0.01$, $\varepsilon = 0.35$, $\delta = 0.01$, $i = 0.04$, $H_0 = 1$, $K_0 = 30$. Die Funktion $g(u_3)$ in (1.29) ist durch die Parabel $g(u_3) = 1 - (1 - a)u_3 - au_3^2$ gegeben.

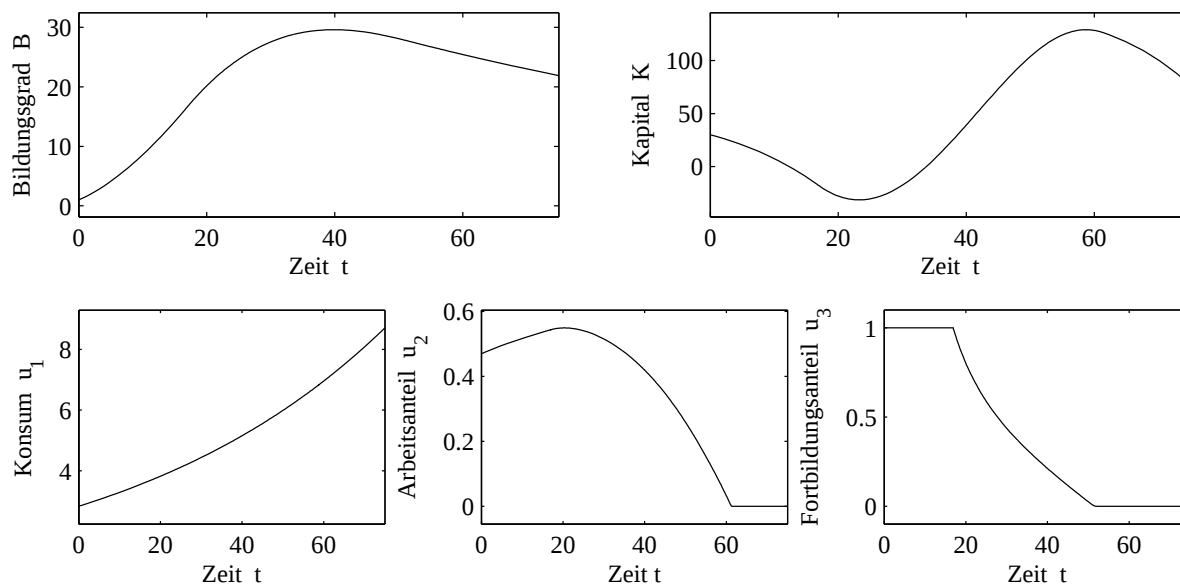


Abbildung 1.9: Optimale Trajektorien für das Konsumentenverhalten in Beispiel 1.5.

Literatur

- [1] A.E. Bryson. *Dynamic Optimization*. Addison–Wesley, Menlo Park, CA, 1999.
- [2] R.H. Goddard. A method for reaching extreme altitudes. *Smithsonian Miscellaneous Collections* 71, 1919.
- [3] H.J. Oberle und R. Rosendahl. Numerical computation of a singular–state subarc in an economic optimal control model. *Optimal Control Applications and Methods*, 27:211–235, 2006.
- [4] K. Pohmer. *Mikroökonomische Theorie der personellen Einkommens– und Vermögensverteilung*. Studies in Contemporary Economics, Band 16. Springer, 1985.

Kapitel 2

Grundlagen der Optimierung

In diesem Kapitel werden zunächst einige Begriffe und Eigenschaften statischer Optimierungsprobleme

$$\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) \quad (2.1)$$

$$\text{u.B.v. } g_i(\mathbf{x}) = 0, \quad i = 1, \dots, p \quad (2.2)$$

$$h_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, q \quad (2.3)$$

betrachtet, auf die in den folgenden Kapiteln Bezug genommen wird.

2.1 Lokale/globale Minima und zulässige Menge

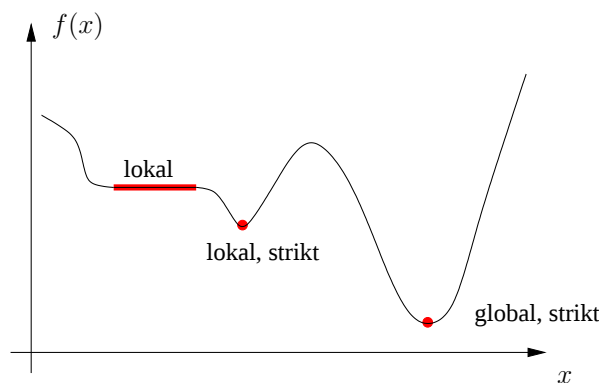
Definition 2.1 (Globale und lokale Minima) Die Funktion $f(\mathbf{x})$ mit $f : \mathcal{X} \rightarrow \mathbb{R}$ und $\mathcal{X} \subseteq \mathbb{R}^n$ besitzt an der Stelle $\mathbf{x}^* \in \mathcal{X}$

- a) ein lokales Minimum, falls $f(\mathbf{x}^*) \leq f(\mathbf{x})$ für alle $\mathbf{x}$ in einer hinreichend kleinen Umgebung von $\mathbf{x}^*$,
- b) ein striktes lokales Minimum, falls $f(\mathbf{x}^*) < f(\mathbf{x})$ für alle $\mathbf{x}$ in einer hinreichend kleinen Umgebung von $\mathbf{x}^*$,
- c) ein globales (absolutes) Minimum, falls $f(\mathbf{x}^*) \leq f(\mathbf{x})$ für alle $\mathbf{x} \in \mathcal{X}$,
- c) ein eindeutiges globales Minimum, falls $f(\mathbf{x}^*) < f(\mathbf{x})$ für alle $\mathbf{x} \in \mathcal{X}$.

Aus der Definition ergibt sich, dass ein globales Minimum stets auch ein lokales Minimum ist. Die unterschiedlichen Arten von Minima sind in Abbildung 2.1 skizziert.

Definition 2.2 (Zulässige Menge) Die zulässige Menge $\mathcal{X}_{\text{ad}}$ (Englisch: admissible set) des Optimierungsproblems (2.1)–(2.3) ist

$$\mathcal{X}_{\text{ad}} = \left\{ \mathbf{x} \in \mathbb{R}^n : \begin{array}{ll} g_i(\mathbf{x}) = 0, & i = 1, \dots, p \\ h_i(\mathbf{x}) \leq 0, & i = 1, \dots, q \end{array} \right\}.$$

Abbildung 2.1: Verschiedene Minima einer Funktion $f(x)$ mit $x \in \mathbb{R}$.

Alle Punkte $\mathbf{x} \in \mathcal{X}_{\text{ad}}$ werden als *zulässige Punkte* bezeichnet und erfüllen die Nebenbedingungen $g_i(\mathbf{x}) = 0$ und $h_i(\mathbf{x}) \leq 0$. Somit lässt sich das statische OP (2.1)–(2.3) auch in der äquivalenten Form

$$\min_{\mathbf{x} \in \mathcal{X}_{\text{ad}}} f(\mathbf{x}) \quad (2.4)$$

angeben. Im Falle von unbeschränkten Problemen gilt $\mathcal{X}_{\text{ad}} = \mathbb{R}^n$.

Beispiel 2.1 Im eingangs erwähnten Beispiel 1.2 mit den beiden Ungleichungsbeschränkungen (1.11) und (1.12) ist der zulässige Bereich durch

$$\mathcal{X}_{\text{ad}} = \{ \mathbf{x} \in \mathbb{R}^n : h_1(\mathbf{x}) = x_1 + x_2 - 2 \leq 0, \quad h_2(\mathbf{x}) = x_1^2 - x_2 \leq 0 \}$$

definiert. Dies entspricht der zwischen der Geraden $h_1(\mathbf{x}) = 0$ und der Parabel $h_2(\mathbf{x}) = 0$ eingeschlossenen Fläche, siehe Abbildung 1.5.

Es ist direkt ersichtlich, dass $\mathcal{X}_{\text{ad}}$ nicht die leere Menge sein darf, da das Optimierungsproblem (2.4) ansonsten keine Lösung besitzt. Eine weitere notwendige Bedingung für $\mathcal{X}_{\text{ad}}$ kann aus den Gleichungsbedingungen (2.2) abgeleitet werden, da sich durch die algebraischen Restriktionen $g_i(\mathbf{x}) = 0$ die Anzahl der freien Optimierungsvariablen $\mathbf{x} \in \mathbb{R}^n$ auf $n - p$ reduziert. Somit darf die Anzahl p der (unabhängigen) Gleichungsbedingungen (2.2) nicht größer als die Anzahl der Optimierungsvariablen $\mathbf{x} \in \mathbb{R}^n$ sein, da die zulässige Menge $\mathcal{X}_{\text{ad}}$ ansonsten leer wäre.

Satz 2.1 (Weierstrass) Sei $f : \mathcal{X}_{\text{ad}} \rightarrow \mathbb{R}$ stetig und $\mathcal{X}_{\text{ad}}$ eine kompakte nichtleere Menge. Dann besitzt das Problem

$$\min_{\mathbf{x} \in \mathcal{X}_{\text{ad}}} f(\mathbf{x}) \quad (2.5)$$

an mindestens einem Punkt $\mathbf{x}^* \in \mathcal{X}_{\text{ad}}$ ein Minimum.

Beispiel 2.2 Betrachtet wird eine erweiterte Version des beschränkten Optimierungsproblem aus Beispiel 1.2 bzw. Abbildung 1.5:

$$\min_{\mathbf{x} \in \mathbb{R}^2} (x_1 - 2)^2 + (x_2 - 1)^2 \quad (2.6)$$

$$\text{u.B.v.} \quad x_1 + x_2 - 2 \leq 0 \quad (2.7)$$

$$x_1^2 - x_2 \leq 0, \quad x_1 \geq 0. \quad (2.8)$$

Die Kostenfunktion ist stetig und die zulässige Menge $\mathcal{X}_{\text{ad}}$ ist nicht-leer und kompakt. Somit folgt aus Satz 2.1 die Existenz einer Lösung $\mathbf{x}^*$.

Der Satz von Weierstrass gilt nicht für Probleme, die auf einer *offenen, nicht kompakten Menge* definiert sind (z.B. $\mathcal{X}_{\text{ad}} = \mathbb{R}^n$). Ein Beispiel ist

$$\min_{x \in (0, \infty)} \frac{1}{x}. \quad (2.9)$$

Obwohl $f(x) = 1/x$ auf dem offenen Intervall $x \in (0, \infty)$ stetig ist, existiert kein Minimum. Allerdings kann die Existenz einer Lösung eines unbeschränkten Optimierungsproblems garantiert werden, wenn die Funktion $f(\mathbf{x})$ die Bedingung $\lim_{\|\mathbf{x}\| \rightarrow \infty} f(\mathbf{x}) = \infty$ erfüllt.

2.2 Gradient und Hessematrix

Die Berechnung von Ableitungen erster und zweiter Ordnung einer Kostenfunktion $f(\mathbf{x})$ ist von fundamentaler Bedeutung in der Optimierung. Da im Falle von unstetigen Funktionen oder unstetigen Ableitungen häufig Probleme auftreten (sowohl numerischer als auch theoretischer Natur), wird oft angenommen, dass alle Funktionen eines Optimierungsproblems stetig und hinreichend oft differenzierbar sind.

Definition 2.3 (Gradient) Sei $f : \mathcal{X} \rightarrow \mathbb{R}$ stetig differenzierbar. Dann bezeichnet

$$\nabla f(\mathbf{x}) = \frac{\partial f}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix} \quad (2.10)$$

den Gradienten (also die 1. partielle Ableitung) von $f(\mathbf{x})$ an der Stelle $\mathbf{x} = [x_1, \dots, x_n]^\top$.

Der negative Gradient ist von besonderer Bedeutung zur Lösung eines Optimierungsproblems, da er die sogenannte *maximale Abstiegsrichtung* einer Kostenfunktion $f(\mathbf{x})$ an der Stelle $\mathbf{x}$ angibt, siehe Abbildung 2.2. Diese Eigenschaft wird im nächsten Kapitel näher untersucht.

Definition 2.4 (Hessematrix) Sei $f : \mathcal{X} \rightarrow \mathbb{R}$ zweimal stetig differenzierbar. Dann bezeichnet

$$\nabla^2 f(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix} \quad (2.11)$$

die Hessematrix (also die 2. partielle Ableitung) von $f(\mathbf{x})$ an der Stelle $\mathbf{x} = [x_1, \dots, x_n]^\top$.

Im Falle von ein-dimensionalen Funktionen $f(x)$ wird die ∇ -Notation normalerweise durch $f'(x)$ und $f''(x)$ ersetzt.

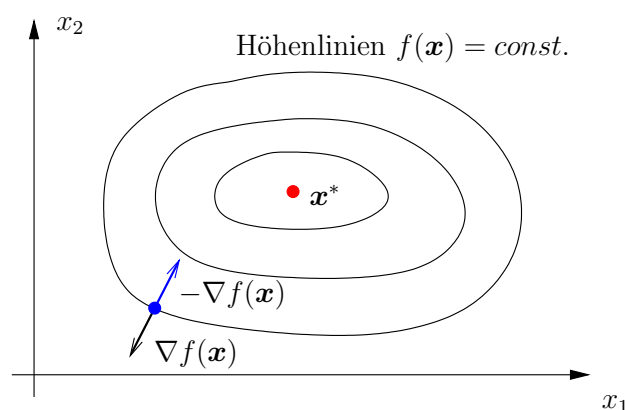


Abbildung 2.2: Maximale Abstiegsrichtung $-\nabla f(\mathbf{x})$ für eine Funktion $f(\mathbf{x})$ mit zwei Variablen $\mathbf{x} = [x_1, x_2]^T$.

Nach dem Satz von Schwarz folgt aus der Stetigkeit der 2. partiellen Ableitungen Kommutativität, d.h.

$$\frac{\partial^2 f}{\partial x_i \partial x_j} = \frac{\partial^2 f}{\partial x_j \partial x_i}.$$

Somit ist die Hessematrix symmetrisch ($\nabla^2 f(\mathbf{x}) = \nabla^2 f(\mathbf{x})^T$) und hat stets reelle Eigenwerte. In der Optimierung ist oft von Bedeutung, ob Hessematrizen positiv (semi-)definit sind. Diese Eigenschaft kann wie folgt untersucht werden:

Satz 2.2 (Definitheit) Die Definitheit einer symmetrischen Matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ lässt sich durch folgende Bedingungen charakterisieren:

Matrix $\mathbf{A}$ ist	(1): für alle $\mathbf{p} \in \mathbb{R}^n$ mit $\mathbf{p} \neq \mathbf{0}$ gilt	(2): alle n Eigen- werte λ_i sind	(3): für alle n Haupt- unterdet. D_i gilt
positiv semi-definit:	$\mathbf{p}^T \mathbf{A} \mathbf{p} \geq 0$	reell und ≥ 0	$D_i \geq 0$
positiv definit:	$\mathbf{p}^T \mathbf{A} \mathbf{p} > 0$	reell und > 0	$D_i > 0$
negativ semi-definit:	$\mathbf{p}^T \mathbf{A} \mathbf{p} \leq 0$	reell und ≤ 0	$(-1)^i D_i \leq 0$
negativ definit:	$\mathbf{p}^T \mathbf{A} \mathbf{p} < 0$	reell und < 0	$(-1)^i D_i < 0$

Die Eigenwerte λ_i , $i = 1, \dots, n$ der Matrix $\mathbf{A}$ sind die Lösungen der Gleichung

$$\det(\lambda \mathbf{I} - \mathbf{A}) = 0,$$

wobei $\mathbf{I}$ die Einheitsmatrix der Dimension n darstellt. Die Hauptunterdeterminanten D_i sind die Determinanten der linken oberen Untermatrizen von $\mathbf{A}$:

$$D_1 = |a_{11}|, \quad D_2 = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \cdots D_n = \begin{vmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{vmatrix},$$

wobei die Elemente a_{ij} die Elemente der Matrix $\mathbf{A}$ darstellen. Um die Definitheit einer symmetrischen Matrix $\mathbf{A}$ zu bestimmen, reicht es eine der drei Bedingungen (1)–(3) auszuwerten, da jede für sich notwendig und hinreichend ist. Das Kriterium (3) wird auch *Sylvester-Kriterium* genannt. Zur Anwendung der drei Kriterien zur Bestimmung der Definitheit einer Matrix sei auf Aufgabe 2.1 verwiesen.

2.3 Konvexität

Die Eigenschaft der Konvexität ist von fundamentaler Bedeutung in der Optimierung und führt im Allgemeinen zu einer einfacheren numerischen Lösung des Optimierungsproblems. Zudem ist Konvexität notwendig, um eine Aussage über eine eindeutige Lösung eines Optimierungsproblems treffen zu können. Der Begriff *konvex* kann sowohl auf Mengen als auch auf Funktionen und Optimierungsprobleme angewendet werden.

2.3.1 Konvexe Mengen

Definition 2.5 (Konvexe Menge) Eine Menge $\mathcal{X} \subseteq \mathbb{R}^n$ ist konvex, falls die folgende Bedingung für beliebige Punkte $\mathbf{x}, \mathbf{y} \in \mathcal{X}$ erfüllt ist:

$$\mathbf{z} = k \mathbf{x} + (1 - k) \mathbf{y} \in \mathcal{X} \quad \text{für alle } k \in [0, 1]. \quad (2.12)$$

Eine geometrische Interpretation dieser Definition ist, dass eine Menge $\mathcal{X} \subseteq \mathbb{R}^n$ genau dann konvex ist, falls die Verbindungslinie zwischen zwei beliebigen Punkten $\mathbf{x}, \mathbf{y} \in \mathcal{X}$ komplett in $\mathcal{X}$ enthalten ist. Abbildung 2.3 zeigt einige Beispiele konvexer und nicht-konvexer Mengen.

Eine bekannte Eigenschaft konvexer Mengen ist, dass ihre Schnittmenge ebenfalls eine konvexe Menge darstellt. Diese Eigenschaft ist von Bedeutung zur Charakterisierung der Konvexität der zulässigen Menge $\mathcal{X}_{\text{ad}}$ von Optimierungsproblemen (siehe Satz 2.4).

2.3.2 Konvexe Funktionen

Definition 2.6 (Konvexe und konkave Funktionen) Sei $\mathcal{X} \subseteq \mathbb{R}^n$ eine konvexe Menge. Eine Funktion $f : \mathcal{X} \rightarrow \mathbb{R}$ ist konvex auf $\mathcal{X}$, falls die folgende Bedingung für beliebige Punkte $\mathbf{x}, \mathbf{y} \in \mathcal{X}$ und $\mathbf{z} = k \mathbf{x} + (1 - k) \mathbf{y}$ erfüllt ist:

$$f(\mathbf{z}) \leq k f(\mathbf{x}) + (1 - k) f(\mathbf{y}) \quad \forall k \in [0, 1]. \quad (2.13)$$

Die Funktion f ist strikt konvex, falls

$$f(\mathbf{z}) < k f(\mathbf{x}) + (1 - k) f(\mathbf{y}) \quad \forall k \in (0, 1). \quad (2.14)$$

Die Funktion ist f (strikt) konkav, falls $-f$ (strikt) konvex ist.

Die Definition 2.6 kann wie folgt geometrisch interpretiert werden: eine Funktion f ist konvex (konkav), falls alle Funktionswerte $f(\mathbf{z})$ mit $\mathbf{z} = k \mathbf{x} + (1 - k) \mathbf{y}$ unterhalb (oberhalb) der Verbindungslinie zwischen $f(\mathbf{x})$ und $f(\mathbf{y})$ liegen. Abbildung 2.4 zeigt einige Beispiele konvexer und konkaver Funktionen. Es ist direkt ersichtlich, dass lineare Funktionen sowohl konkav als auch konvex sind.

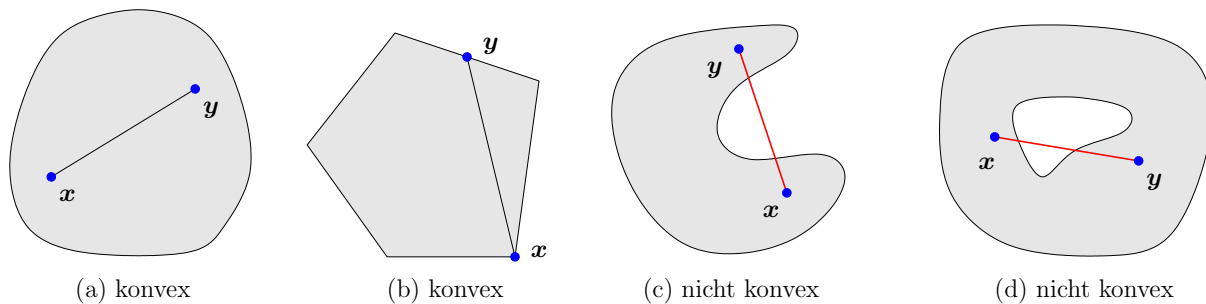


Abbildung 2.3: Beispiele von konvexen Mengen.

Konvexe Funktionen besitzen einige interessante Eigenschaften, die im Folgenden dargestellt sind:

a) Die Summenfunktion

$$f(\mathbf{x}) = \sum_{i=1}^k a_i f_i(\mathbf{x}) \quad (2.15)$$

von konvexen Funktionen $f_i(\mathbf{x})$, $i = 1, \dots, k$ mit den Koeffizienten $a_i \geq 0$, $i = 1, \dots, k$ ist konvex.

b) Ist die Funktion $f(\mathbf{x})$ konvex, so ist die Menge

$$\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^n : f(\mathbf{x}) \leq k\} \quad (2.16)$$

mit $k > 0$ stets konvex, siehe Abbildung 2.5.

c) Eine stetig differenzierbare Funktion $f(\mathbf{x})$ ist konvex auf $\mathcal{X}$, wenn für alle $\mathbf{x}, \mathbf{y} \in \mathcal{X}$ die Ungleichung

$$f(\mathbf{y}) \geq f(\mathbf{x}) + (\mathbf{y} - \mathbf{x})^\top \nabla f(\mathbf{x}) \quad (2.17)$$

erfüllt ist. Dies kann als eine Alternative zu Definition 2.6 angesehen werden und ist in Abbildung 2.6 veranschaulicht. Die geometrische Interpretation der Ungleichung (2.17) ist, dass an jedem Punkt $\mathbf{x}$ einer konvexen Funktion $f(\mathbf{x})$ eine sogenannte *stützende Hyperebene* (skalärer Fall: *stützende Tangente*) existieren muss, oberhalb der $f(\mathbf{x})$ verläuft.

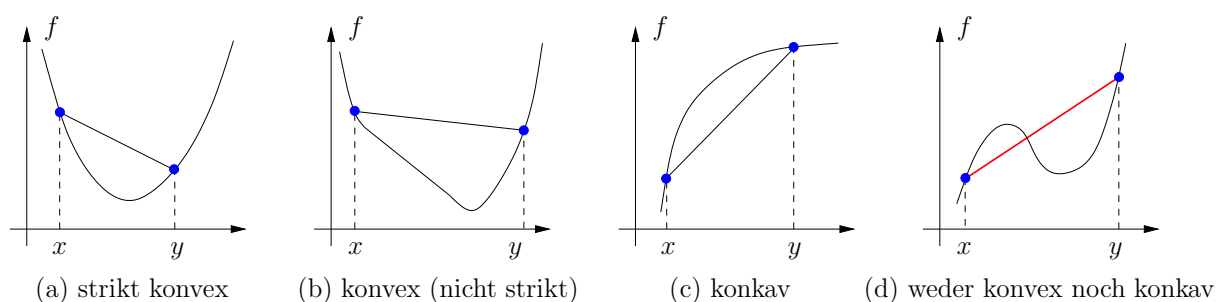


Abbildung 2.4: Beispiele von konvexen und konkaven Funktionen.

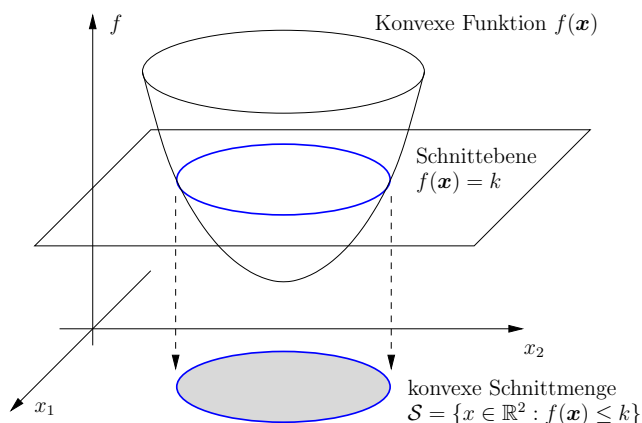


Abbildung 2.5: Konvexe Menge $\mathcal{S}$, die durch den Schnitt einer konvexen Funktion $f(\mathbf{x})$ mit der Ebene $f(\mathbf{x}) = \text{konst.}$ entsteht.

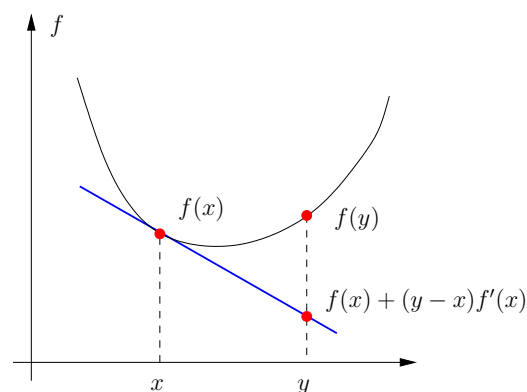


Abbildung 2.6: Stützende Tangente einer konvexen Funktion $f(x)$.

Beispiel 2.3 Gegeben ist die Funktion

$$\min_{\mathbf{x} \in \mathbb{R}^2} f(\mathbf{x}) = -\frac{100}{(x_1 - 1)^2 + (x_2 - 1)^2 + 1} - \frac{200}{(x_1 + 1)^2 + (x_2 + 2)^2 + 1}. \quad (2.18)$$

die in Abbildung 2.7 dargestellt ist. Es ist leicht zu sehen, dass die Funktion (2.18) auf $\mathbb{R}^2$ nicht konvex ist. Dies ist auch anhand der Höhenlinien in Abbildung 2.7 ersichtlich, die die Schnittmengen (2.16) definieren: es ist offensichtlich, dass Schnittmengen existieren, die nicht konvex sind.

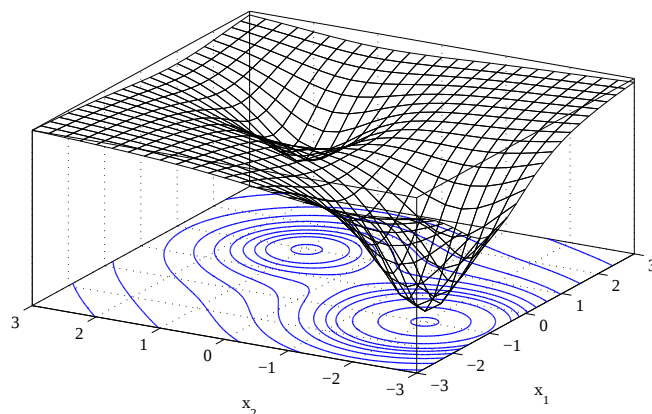


Abbildung 2.7: Profil und Höhenlinien der Funktion (2.18).

Der folgende Satz charakterisiert die Konvexität einer Funktion anhand der Definitheit ihrer Hessematrix (siehe auch Satz 2.2):

Satz 2.3 Sei $\mathcal{X} \subseteq \mathbb{R}^n$ konvex und $f : \mathcal{X} \rightarrow \mathbb{R}$ zweimal stetig differenzierbar. Dann gilt:

- a) f ist konvex, falls $\nabla^2 f(\mathbf{x})$ für alle $\mathbf{x} \in \mathcal{X}$ positiv semi-definit ist,
- b) f ist strikt konvex, falls $\nabla^2 f(\mathbf{x})$ für alle $\mathbf{x} \in \mathcal{X}$ positiv definit ist,
- c) f ist konkav, falls $\nabla^2 f(\mathbf{x})$ für alle $\mathbf{x} \in \mathcal{X}$ negativ semi-definit ist,
- d) f ist strikt konkav, falls $\nabla^2 f(\mathbf{x})$ für alle $\mathbf{x} \in \mathcal{X}$ negativ definit ist.

An dieser Stelle sei allerdings angemerkt, dass die strikte Konvexität einer Funktion $f(\mathbf{x})$ nicht die positiv Definitheit der Hessematrix gewährleistet. Ein Beispiel ist die strikt konvexe Funktion $f(x) = x^4$ mit der zweiten Ableitung $f''(0) = 0$ am Punkt $x = 0$.

Aufgabe 2.1 Untersuchen Sie, ob die folgenden Funktionen auf Ihrem Definitionsbereich konvex sind, indem Sie die Definitheit der Hessematrix $\nabla^2 f(\mathbf{x})$ mittels der Kriterien (1)–(3) in Satz 2.2 überprüfen:

$$f(\mathbf{x}) = x_1^4 + x_1^2 - 2x_1x_2 + x_2^2, \quad \mathbf{x} \in \mathbb{R}^2 \quad (2.19)$$

$$f(\mathbf{x}) = x_2 + \frac{1}{x_1x_2}, \quad x_{1,2} > 0. \quad (2.20)$$

Aufgabe 2.2 Untersuchen Sie, ob die folgenden Funktionen auf Ihrem Definitionsbereich konvex bzw. konkav sind und stellen Sie die Funktionen grafisch (z.B. unter MATLAB) dar:

$$f(x) = e^x - 1, \quad x \in \mathbb{R} \quad (2.21)$$

$$f(\mathbf{x}) = x_1^\alpha x_2^{1-\alpha}, \quad x_{1,2} > 0, \quad \alpha \in [0, 1]. \quad (2.22)$$

2.3.3 Konvexe Optimierungsprobleme

Der Begriff der Konvexität von Mengen und Funktionen lässt sich ebenfalls auf Optimierungsprobleme erweitern:

Definition 2.7 Das statische Optimierungsproblem

$$\min_{\mathbf{x} \in \mathcal{X}_{\text{ad}}} f(\mathbf{x})$$

ist (strikt) konvex, falls die zulässige Menge $\mathcal{X}_{\text{ad}}$ konvex und die Kostenfunktion f (strikt) konvex auf $\mathcal{X}_{\text{ad}}$ ist.

Die Konvexität einer zulässigen Menge $\mathcal{X}_{\text{ad}}$ kann wie folgt untersucht werden:

Satz 2.4 Falls die Funktionen $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, p$ linear und die Funktionen $h_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, q$ konvex sind, dann ist die zulässige Menge

$$\mathcal{X}_{\text{ad}} = \left\{ \mathbf{x} \in \mathbb{R}^n : \begin{array}{ll} g_i(\mathbf{x}) = 0, & i = 1, \dots, p \\ h_i(\mathbf{x}) \leq 0, & i = 1, \dots, q \end{array} \right\}.$$

konvex.

Dieser Satz ergibt sich aus der bereits erwähnten Eigenschaft, dass der Schnitt zweier konvexer Mengen wiederum eine konvexe Menge ergibt. Für lineare (und somit konvexe) Gleichungsbedingungen $g_i(\mathbf{x}) = 0$ und konvexe Ungleichungsnebenbedingungen $h_i(\mathbf{x}) \leq 0$

sind die zugehörigen Untermengen stets konvex (siehe auch Punkt b) auf Seite 16 und Abbildung 2.5). Somit ist die gesamte Schnittmenge, also der zulässige Bereich $\mathcal{X}_{\text{ad}}$, ebenfalls eine konvexe Menge.

Mit den vorherigen Ergebnissen können nun hinreichende Bedingungen für die Konvexität von statischen Optimierungsproblemen angegeben werden:

Satz 2.5 *Das statische Optimierungsproblem*

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^n} \quad & f(\mathbf{x}) \\ \text{u.B.v.} \quad & g_i(\mathbf{x}) = 0, \quad i = 1, \dots, p \\ & h_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, q \end{aligned}$$

ist (strikt) konvex, falls

- a) die Funktionen $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, p$ linear sind,
- b) die Funktionen $h_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, q$ konvex sind,
- c) die Kostenfunktion f (strikt) konvex auf $\mathcal{X}_{\text{ad}} \subseteq \mathbb{R}^n$ ist.

Einige wichtige Eigenschaften konvexer Optimierungsprobleme sind:

- a) Jedes lokale Minimum ist ein globales Minimum.
- b) Falls ein striktes lokales Minimum existiert, dann ist dieses eindeutig und global.
- c) Ein strikt konvexes Optimierungsproblem kann nicht mehr als eine Lösung besitzen: falls ein lokales Minimum existiert, ist dieses das eindeutige globale Minimum.

Es sei an dieser Stelle angemerkt, dass strikte Konvexität eines Optimierungsproblems nicht die Existenz einer Lösung impliziert. Als Beispiel dient wieder das Problem (2.9)

$$\min_{x \in (0, \infty)} \frac{1}{x}. \quad (2.23)$$

Die erste und zweite Ableitung der Funktion $f(x) = \frac{1}{x}$ lauten $f'(x) = -\frac{1}{x^2}$ und $f''(x) = \frac{2}{x^3} > 0$. Somit ist $f''(x) > 0$ für alle $x \in \mathcal{X} = (0, \infty)$ und das Optimierungsproblem (2.23) ist strikt konvex. Allerdings ist direkt ersichtlich, dass kein Minimum existiert.¹

Aus den oben angegebenen Eigenschaften konvexer Optimierungsprobleme ist ersichtlich, dass Konvexität die Analyse von Optimierungsproblemen erheblich vereinfacht. Die meisten theoretischen Untersuchungen von Optimierungsproblemen und Konvergenzeigenschaften numerischer Verfahren basieren auf dieser Annahme. Da die meisten numerischen Verfahren zu einem lokalen Minimum konvergieren, impliziert dies im Falle von (strikter) Konvexität automatisch eine (eindeutige) globale Lösung.

¹ Es sei an dieser Stelle auch daran erinnert, dass der Satz 2.1 von Weierstrass nur die Existenz einer Lösung garantiert, wenn die zulässige Menge $\mathcal{X}$ (bzw. $\mathcal{X}_{\text{ad}}$) kompakt ist.

Beispiel 2.4 *Betrachtet wird das Problem aus Beispiel 2.2. Die Kostenfunktion (2.6) und die Beschränkungen (2.7), (2.8) sind konvex. Somit ist das Optimierungsproblem (2.6)–(2.8) konvex und die lokale Lösung $\mathbf{x}^* = [1, 1]^\top$ (siehe Abbildung 1.5) ist die globale Lösung.*

Literatur

- [1] S. Boyd und L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [2] J. Nocedal und S.J. Wright. *Numerical Optimization*. Springer, 2006.
- [3] M. Papageorgiou. *Optimierung*. Oldenbourg Verlag, 1991.

Kapitel 3

Statische Optimierung ohne Beschränkungen

In diesem Kapitel werden unbeschränkte statische Optimierungsprobleme der Form

$$\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) \quad (3.1)$$

betrachtet sowie notwendige und hinreichende Optimalitätsbedingungen hergeleitet. Im Anschluss werden einige bekannte numerische Verfahren vorgestellt, um einen Eindruck zu vermitteln, wie unbeschränkte Optimierungsprobleme numerisch gelöst werden können.

3.1 Notwendige Optimalitätsbedingungen

Die Optimalitätsbedingungen für unbeschränkte Probleme (3.1) dienen einerseits zum Auffinden von (lokalen) Extremwerten $\mathbf{x}^*$ als auch zum Überprüfen, ob es sich tatsächlich um ein Minimum handelt, dass also

$$f(\mathbf{x}) \geq f(\mathbf{x}^*) \quad (3.2)$$

für alle $\mathbf{x}$ in einer hinreichend kleinen Umgebung von $\mathbf{x}^*$ (siehe Definition 2.1) gilt.

Unter der Annahme, dass $f(\mathbf{x})$ mindestens zweimal stetig differenzierbar ist, lässt sich $f(\mathbf{x})$ an der Stelle $\mathbf{x}^*$ in eine Taylorreihe entwickeln:

$$f(\mathbf{x}^* + \delta\mathbf{x}) = f(\mathbf{x}^*) + \nabla f(\mathbf{x}^*)^\top \delta\mathbf{x} + \frac{1}{2} \delta\mathbf{x}^\top \nabla^2 f(\mathbf{x}^*) \delta\mathbf{x} + \mathcal{O}(\|\delta\mathbf{x}\|^3). \quad (3.3)$$

Falls der Punkt $\mathbf{x}^*$ ein lokales Minimum darstellt, muss (3.2) mit $\mathbf{x} = \mathbf{x}^* + \delta\mathbf{x}$ für alle hinreichend kleinen $\delta\mathbf{x}$ erfüllt sein. Einsetzen von (3.3) in (3.2) führt auf die Ungleichung

$$\nabla f(\mathbf{x}^*)^\top \delta\mathbf{x} + \frac{1}{2} \delta\mathbf{x}^\top \nabla^2 f(\mathbf{x}^*) \delta\mathbf{x} + \mathcal{O}(\|\delta\mathbf{x}\|^3) \geq 0. \quad (3.4)$$

Für hinreichend kleine $\delta\mathbf{x}$ ist der erste Term $\nabla f(\mathbf{x}^*)^\top \delta\mathbf{x}$ dominant und bestimmt das Vorzeichen der gesamten linken Seite in (3.4). Es muss also $\nabla f(\mathbf{x}^*)^\top \delta\mathbf{x} \geq 0$ erfüllt sein.

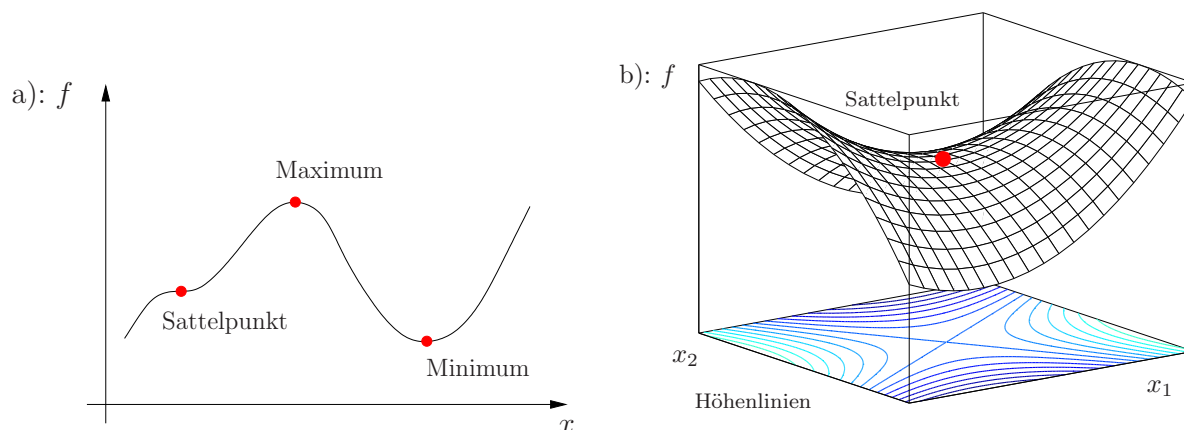


Abbildung 3.1: Beispiele von stationären Punkten im ein- und zwei-dimensionalen Fall.

Da $\mathbf{x}^*$ nur ein lokales Minimum ist, wenn die Ungleichung (3.4) für alle (hinreichend kleinen) $\delta\mathbf{x}$ erfüllt ist, ergibt sich direkt, dass

$$\nabla f(\mathbf{x}^*) = \mathbf{0} \quad (3.5)$$

gelten muss. Allerdings sagt diese Bedingung 1. Ordnung lediglich aus, dass $\mathbf{x}^*$ ein Extrempunkt von f ist, da $\nabla f(\mathbf{x}^*) = \mathbf{0}$ auch von Maxima oder Sattelpunkten erfüllt wird, wie in Abbildung 3.1 dargestellt ist. In der Optimierung spricht man in diesem Zusammenhang oft von einem *stationären Punkt*.

Eine strengere Bedingung 2. Ordnung für ein lokales Minimum $\mathbf{x}^*$ ergibt sich aus (3.4):

$$\delta\mathbf{x}^\top \nabla^2 f(\mathbf{x}^*) \delta\mathbf{x} \geq 0. \quad (3.6)$$

Da diese Bedingung für alle hinreichend kleinen $\delta\mathbf{x}$ in einer Umgebung von $\mathbf{x}^*$ gelten soll, muss die Hessematrix $\nabla^2 f(\mathbf{x}^*)$ folglich positiv semi-definit sein, wie sich aus Satz 2.2 leicht ablesen lässt. Die notwendigen Bedingungen für ein lokales Minimum sind im folgenden Satz zusammengefasst:

Satz 3.1 (Notwendige Optimalitätsbedingungen)

Sei $f: \mathbb{R}^n \rightarrow \mathbb{R}$ eine zweimal stetig differenzierbare Funktion und $\mathbf{x}^*$ ein lokales Minimum von f . Dann gelten die folgenden Optimalitätsbedingungen 1. und 2. Ordnung:

$$\nabla f(\mathbf{x}^*) = \mathbf{0} \quad (3.7)$$

$$\nabla^2 f(\mathbf{x}^*) \geq \mathbf{0} \quad (\text{positiv semi-definit}). \quad (3.8)$$

Allerdings stellen diese Optimalitätsbedingungen keine hinreichenden Bedingungen dar und sind somit nicht umkehrbar. Ein Beispiel ist die Funktion $f(x) = x^3$, die an der Stelle $x^* = 0$ einen Extrempunkt besitzt, wie man anhand der ersten Ableitung $f'(x) = 3x^2$ leicht erkennen kann. Obwohl die zweite Ableitung $f''(x^*) = 3x^* = 0$ positiv semi-definit ist, ist $x^* = 0$ kein Minimum, sondern ein Sattelpunkt (siehe Abbildung 3.1).

3.2 Hinreichende Optimalitätsbedingungen

Eine hinreichende Bedingung, die die Existenz eines lokalen Minimums $\mathbf{x}^*$ garantiert, kann aus der Ungleichung (3.4) abgelesen werden. Falls $\nabla f(\mathbf{x}^*) = \mathbf{0}$ gilt und der quadratische Term in (3.4) strikt positiv ist, d.h.

$$\delta \mathbf{x}^\top \nabla^2 f(\mathbf{x}^*) \delta \mathbf{x} > 0, \quad (3.9)$$

dann ist garantiert, dass für alle hinreichend kleinen $\delta \mathbf{x}$ die Ungleichung (3.4) strikt erfüllt und somit $\mathbf{x}^*$ ein lokales Minimum darstellt. Man spricht in diesem Fall von *hinreichenden Optimalitätsbedingungen*, die im folgenden Satz zusammengefasst sind:

Satz 3.2 (Hinreichende Optimalitätsbedingungen)

Sei $f : \mathbb{R}^n \rightarrow \mathbb{R}$ eine zweimal stetig differenzierbare Funktion und $\mathbf{x}^* \in \mathcal{X}$. Falls $f(\mathbf{x}^*)$ die Bedingungen

$$\nabla f(\mathbf{x}^*) = \mathbf{0} \quad (3.10)$$

$$\nabla^2 f(\mathbf{x}^*) > \mathbf{0} \quad (\text{positiv definit}) \quad (3.11)$$

erfüllt, dann ist $\mathbf{x}^*$ ein striktes lokales Minimum von f .

An dieser Stelle sei angemerkt, dass ein striktes Minimum $\mathbf{x}^*$ nicht automatisch die Positiv-Definitheit der Hessematrix $\nabla^2 f(\mathbf{x}^*)$ impliziert. Ein Beispiel ist die Funktion $f(x) = x^4$, die offensichtlich ein striktes Minimum an der Stelle $x^* = 0$ besitzt. Allerdings verschwindet die zweite Ableitung $f''(x) = 12x^2$ an der Stelle $x^* = 0$ und ist somit nicht positiv definit.

Die Optimalitätsbedingungen erster Ordnung, $\nabla f(\mathbf{x}^*) = \mathbf{0}$, stellen ein nichtlineares Gleichungssystem der Ordnung n dar, das gelöst werden kann, um einen stationären Punkt $\mathbf{x}^*$ zu bestimmen. Die Positiv-Definitheit der Hessematrix $\nabla^2 f(\mathbf{x}^*)$ kann dann mit Hilfe der Methoden in Satz 2.2 untersucht werden, um zu überprüfen, ob $\mathbf{x}^*$ ein striktes Minimum ist.

Beispiel 3.1 Für das Optimierungsproblem

$$\min_{\mathbf{x} \in \mathbb{R}^2} x_1^2 + ax_2^2 - x_1x_2, \quad (3.12)$$

sollen die stationären Werte $\mathbf{x}^*$ in Abhängigkeit des Parameters a charakterisiert werden. Der Gradient und die Hessematrix von $f(\mathbf{x})$ ergeben sich zu

$$\nabla f(\mathbf{x}) = \begin{bmatrix} 2x_1 - x_2 \\ 2ax_2 - x_1 \end{bmatrix}, \quad \nabla^2 f(\mathbf{x}) = \begin{bmatrix} 2 & -1 \\ -1 & 2a \end{bmatrix}. \quad (3.13)$$

Aus $\nabla f(\mathbf{x}^*) = \mathbf{0}$ folgt $\mathbf{x}^* = [0, 0]^\top$ als einziger stationärer Punkt. Die Definitheit der Hessematrix $\nabla^2 f(\mathbf{x})$ an der Stelle $\mathbf{x}^*$ lässt sich mit Hilfe der Hauptunterdeterminanten (Sylvesterkriterium, siehe (3) in Satz 2.2) untersuchen:

$$D_1 = 2, \quad D_2 = 4a - 1. \quad (3.14)$$

Somit ist $\nabla^2 f(\mathbf{x}^*)$ positiv definit für $a > 1/4$ und $\mathbf{x}^* = [0, 0]^\top$ stellt ein striktes Minimum dar. Für $a < 1/4$ ist $D_1 > 0$ und $D_2 < 0$ und $\nabla^2 f(\mathbf{x})$ somit indefinit. In diesem Fall ist $\mathbf{x}^* = [0, 0]^\top$ ein Sattelpunkt, wie er in Abbildung 3.1b) für $a = -1$ dargestellt ist.

Falls die Kostenfunktion $f(\mathbf{x})$ konvex (bzw. strikt konvex) ist, lassen sich stärkere Aussagen treffen als in den Sätzen 3.1–3.2, da in diesem Fall die Hessematrix $\nabla^2 f(\mathbf{x})$ stets positiv semi-definit (bzw. positiv definit) ist:

Satz 3.3 (Hinreichende Optimalitätsbedingungen für konvexe Funktionen)

Sei $f : \mathbb{R}^n \rightarrow \mathbb{R}$ eine stetig differenzierbare und (strikt) konvexe Funktion. Falls $\nabla f(\mathbf{x}^*) = \mathbf{0}$, dann ist $\mathbf{x}^*$ ein (eindeutiges) globales Minimum der Funktion f .

3.3 Numerische Lösung mit Liniensuchverfahren

Da die Stationaritätsbedingung $\nabla f(\mathbf{x}^*) = \mathbf{0}$ nur in seltenen Fällen analytisch ausgewertet werden können, ist man im Allgemeinen auf numerische Verfahren angewiesen, um einen stationären Punkt $\mathbf{x}^*$ zu finden. Eines der bekanntesten numerischen Verfahren ist die *Liniensuche* (Englisch: *line search*). In diesem Abschnitt wird ein Überblick über die bekanntesten Liniensuchverfahren zur Lösung von unbeschränkten statischen Optimierungsproblemen gegeben. Für detaillierte Informationen sei auf die zitierten Textbücher am Ende des Kapitels verwiesen.

3.3.1 Allgemeiner Algorithmus

Die meisten numerischen Verfahren zur Lösung von unbeschränkten Optimierungsproblemen versuchen, ausgehend von einer suboptimalen Startlösung $\mathbf{x}^0$ mit $f(\mathbf{x}^0) > f(\mathbf{x}^*)$, sich iterativ einem Minimum $\mathbf{x}^*$ anzunähern. Dabei wird in jedem Schritt k gefordert, dass

$$f(\mathbf{x}^{k+1}) < f(\mathbf{x}^k) \quad \forall k \in \mathbb{N}_0^+ \quad (3.15)$$

erfüllt ist, um im Grenzfall $k \rightarrow \infty$ gegen ein Minimum zu konvergieren:

$$\lim_{k \rightarrow \infty} \mathbf{x}^k = \mathbf{x}^*. \quad (3.16)$$

Liniensuchverfahren verfolgen dabei prinzipiell die Strategie, zunächst eine *Suchrichtung* (oder *Abstiegsrichtung*) $\mathbf{s}^k$ zu bestimmen, entlang der die Ungleichung (3.15) erfüllt ist.

Um dies zu verdeutlichen, wird die Kostenfunktion $f(\mathbf{x})$ an der Stelle $\mathbf{x}^k$ in eine Taylorreihe entwickelt:

$$f(\mathbf{x}^k + \mathbf{s}) = f(\mathbf{x}^k) + \nabla f(\mathbf{x}^k)^\top \mathbf{s} + \mathcal{O}(\|\mathbf{s}\|^2). \quad (3.17)$$

Für hinreichend kleine $\mathbf{s}$ kann der Restterm $\mathcal{O}(\|\mathbf{s}\|^2)$ vernachlässigt werden und man erkennt sofort, dass an dem Punkt $\mathbf{x}^k$ eine geeignete Suchrichtung $\mathbf{s}^k$ die *Abstiegsbedingung*

$$\nabla f(\mathbf{x}^k)^\top \mathbf{s}^k < 0 \quad (3.18)$$

erfüllen muss, damit $f(\mathbf{x}^k + \mathbf{s}^k) < f(\mathbf{x}^k)$ gilt. Verschiedene Methoden existieren zur Bestimmung der Suchrichtung $\mathbf{s}^k$. Die gebräuchlichsten Verfahren sind in Abschnitt (3.3.3) vorgestellt.

In einem zweiten Schritt muss eine geeignete *Schrittweite* α^k bestimmt werden, um festzulegen, wie weit in Suchrichtung $\mathbf{s}^k$ bis zum nächsten Punkt

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k \mathbf{s}^k \quad (3.19)$$

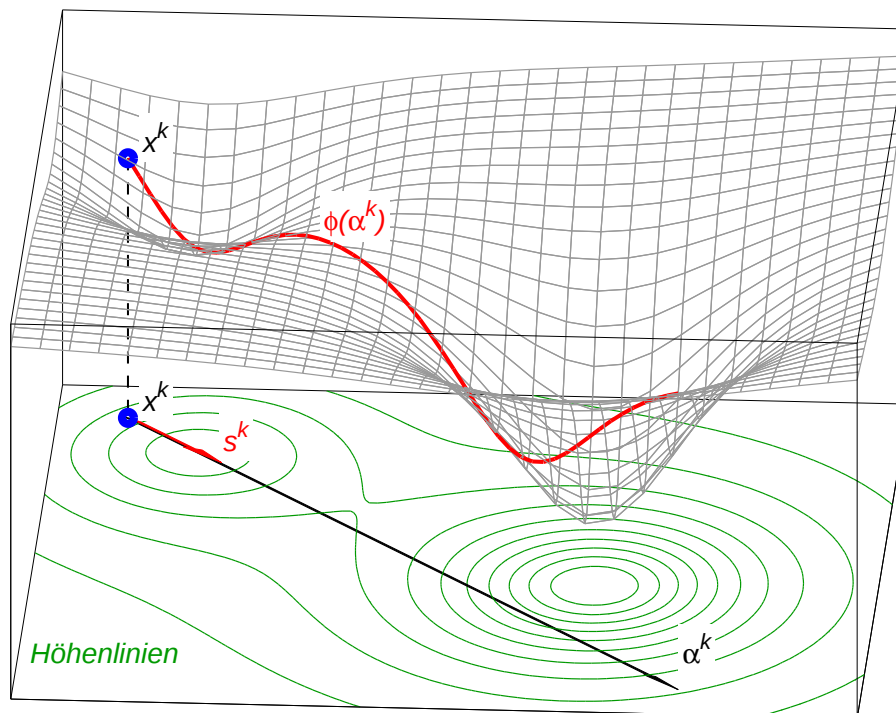


Abbildung 3.2: Veranschaulichung des Liniensuchproblems (3.20).

vorangeschritten werden soll. Die optimale Schrittweite α^k stellt die Lösung des unterlagerten (skalaren) Optimierungsproblems

$$\min_{\alpha^k > 0} \phi(\alpha^k) \quad \text{mit} \quad \phi(\alpha^k) = f(\mathbf{x}^k + \alpha^k \mathbf{s}^k) \quad (3.20)$$

dar, was dem maximal möglichen Abstieg entlang der Suchrichtung $\mathbf{s}^k$ entspricht. Abbildung 3.2 veranschaulicht das Prinzip der Liniensuche für eine (nicht konvexe) Kostenfunktion $f(\mathbf{x})$ mit $\mathbf{x} \in \mathbb{R}^2$ und einer gegebenen Suchrichtung $\mathbf{s}^k$.

Im Allgemeinen wird (3.20) nicht exakt gelöst, sondern anhand geeigneter Methoden approximiert (siehe Abschnitt 3.3.2). Das gewählte α^k definiert dann die Schrittweite in der k -ten Iteration und bestimmt den nächsten Punkt (3.19). Der grundsätzliche Algorithmus von Liniensuchverfahren ist in Tabelle 3.1 dargestellt.

Initialisierung: $\mathbf{x}^0$ (Startwert)
 $k \leftarrow 0$ (Iterationsindex)
 $\varepsilon_x, \varepsilon_f$ (Abbruchkriterien)

repeat

Wahl der Suchrichtung $\mathbf{s}^k$
 Wahl der Schrittweite α^k
 $\mathbf{x}^{k+1} \leftarrow \mathbf{x}^k + \alpha^k \mathbf{s}^k$
 $k \leftarrow k + 1$

until $\|\mathbf{x}^{k+1} - \mathbf{x}^k\| \leq \varepsilon_x$ **or** $\|f(\mathbf{x}^{k+1}) - f(\mathbf{x}^k)\| \leq \varepsilon_f$

Tabelle 3.1: Allgemeine Struktur von Liniensuchverfahren.

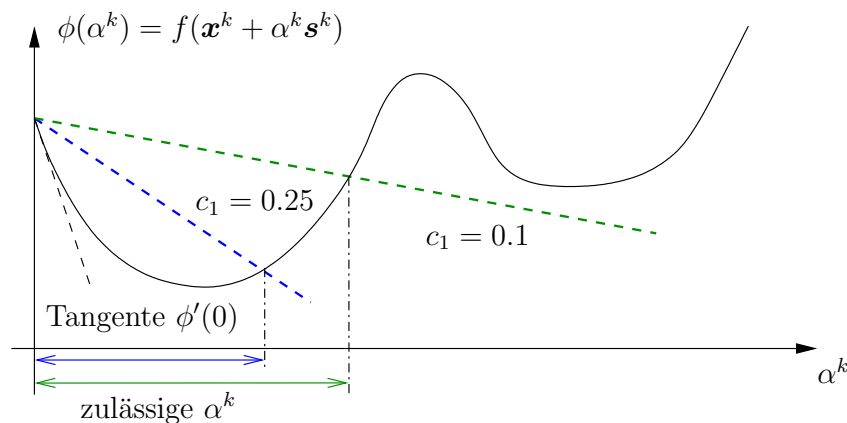


Abbildung 3.3: Veranschaulichung der Abstiegsbedingung (3.21).

3.3.2 Wahl der Schrittweite

Die Berechnung einer geeigneten Schrittweite α^k stellt einen Kompromiss zwischen Rechenzeit und Genauigkeit dar. Die optimale Lösung bestünde darin, das skalare Optimierungsproblem (3.20) exakt oder zumindest hinreichend genau zu lösen, was jedoch in der Praxis zu viele Auswertungen der Kostenfunktion f und eventuell des Gradienten ∇f benötigen würde.

Die meisten numerischen Verfahren verwenden deshalb eine suboptimale Liniensuche, um eine Schrittweite α^k zu bestimmen, die zu einer adäquaten Reduktion der Kostenfunktion f im nächsten Schritt $\mathbf{x}^{k+1}$ führt.

3.3.2.1 Abstiegs- und Krümmungsbedingungen

Ein erstes Kriterium für einen hinreichenden Abstieg in der Kostenfunktion f ist die Ungleichung

$$\underbrace{f(\mathbf{x}^k + \alpha^k \mathbf{s}^k)}_{= \phi(\alpha^k)} \leq \underbrace{f(\mathbf{x}^k)}_{= \phi(0)} + c_1 \alpha^k \underbrace{\nabla f(\mathbf{x}^k)^\top \mathbf{s}^k}_{= \phi'(0)} \quad (3.21)$$

in Abhängigkeit einer Konstanten $c_1 \in (0, 1)$. Gelegentlich wird die Abstiegsbedingung auch *Armijo-Bedingung* genannt. Anschaulich interpretiert bedeutet die Ungleichung (3.21), dass der Abstieg in f proportional zur Schrittweite α^k und der Richtungsableitung $\phi'(0) = \nabla f(\mathbf{x}^k)^\top \mathbf{s}^k$ sein sollte.

Abbildung 3.3 illustriert diesen Zusammenhang. Die rechte Seite der Ungleichung (3.21) stellt eine Gerade mit der negativen Steigung $c_1 \phi'(0)$ dar, aber aufgrund der Konstanten $c_1 \in (0, 1)$ liegt die Gerade für hinreichend kleine Werte von α^k über dem Funktionswert $\phi(\alpha^k)$. Die Abstiegsbedingung (3.21) verlangt folglich, dass Werte von α^k nur zulässig sind, wenn ihr Funktionswert $\phi(\alpha^k)$ unterhalb dieser Geraden liegt. In der Praxis wird c_1 häufig sehr klein gewählt (z.B. 10^{-4}), damit die Abstiegsbedingung nicht zu restriktiv ist.

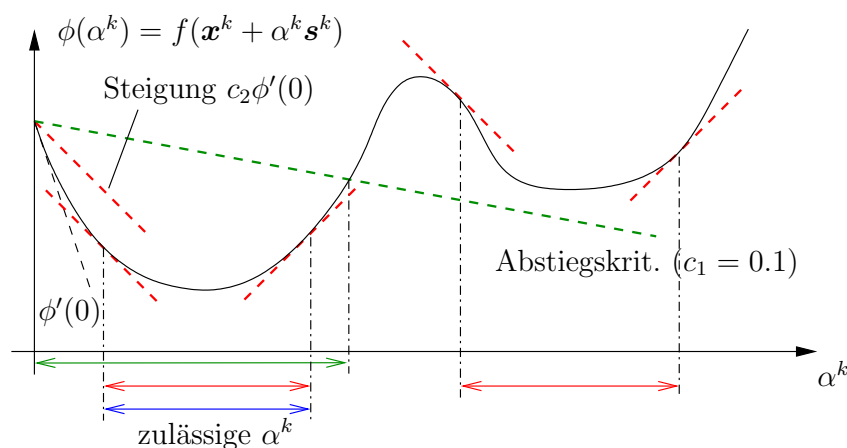


Abbildung 3.4: Veranschaulichung der Abstiegs- und Krümmungsbedingungen (Wolfe-Bedingungen).

Die Abstiegsbedingung (3.21) alleine ist oft nicht genug, um einen nennenswerten Abstieg in der Kostenfunktion f zu gewährleisten, da sie für hinreichend kleine Schrittweiten α^k stets erfüllt ist (siehe Abbildung 3.3). Unnötig kleine Werte von α^k können durch ein zweites zusätzliches Kriterium, die sogenannte *Krümmungsbedingung* vermieden werden ¹

$$\left| \underbrace{\nabla f(\mathbf{x}^k + \alpha^k \mathbf{s}^k)^\top \mathbf{s}^k}_{= \phi'(\alpha^k)} \right| \leq c_2 \left| \underbrace{\nabla f(\mathbf{x}^k)^\top \mathbf{s}^k}_{= \phi'(0)} \right|. \quad (3.22)$$

Die Konstante $c_2 \in (c_1, 1)$ wird dabei in Abhängigkeit der ersten Konstante c_1 aus (3.21) gewählt. Da die linke Seite der Krümmungsbedingung den Gradienten $\phi'(\alpha^k)$ darstellt, bedeutet das Kriterium also, dass der Betrag der Steigung $\phi'(\alpha^k)$ an der Stelle α^k kleiner oder gleich dem c_2 -fachen Betrag der Anfangssteigung $\phi'(0)$ sein muss, siehe Abbildung 3.4. Zusammen mit der Abstiegsbedingung (3.21) ergibt sich somit eine Menge an zulässigen Schrittweiten α^k .

Die Formulierung der Krümmungsbedingung macht durchaus Sinn, da im Falle einer stark negativen Steigung $\phi'(\alpha^k)$ davon auszugehen ist, dass ϕ (und damit f) weiter reduziert werden kann, wenn man sich weiter entlang der α^k -Achse bewegt. Andererseits wird durch die Betragsbedingung vermieden, dass die Steigung $\phi'(\alpha^k)$ zu groß wird und man somit zu weit von einem lokalen Minimum von ϕ entfernt ist.

Die Abstiegs- und Krümmungsbedingungen (3.21)–(3.22) werden auch *Wolfe-Bedingungen* genannt und können in den meisten Liniensuchverfahren eingesetzt werden. Typischerweise wird c_2 zu 0.9 gewählt, wenn die Suchrichtung $\mathbf{s}^k$ durch ein Newton- oder Quasi-Newton-Verfahren bestimmt wird, bzw. zu 0.1 im Falle von Konjugierten-Gradienten-Verfahren (siehe Abschnitt 3.3.3).

Es lässt sich zeigen, dass für stetig differenzierbare und von unten beschränkte Funktionen f stets ein Intervall existiert, in dem die Wolfe-Bedingungen erfüllt sind.

¹ Genau genommen handelt es sich um eine *strenge* Krümmungsbedingung, da eine schwächere Bedingung der Form (3.22) auch ohne Betragsfunktionen angegeben werden kann.

3.3.2.2 Backtracking–Verfahren

Wie bereits erwähnt wurde, ist die Abstiegsbedingung (3.21) allein nicht ausreichend, um einen hinreichend schnellen Abstieg in der Kostenfunktion f zu gewährleisten. Wenn allerdings die Kandidaten für die Schrittweite α^k geeignet gewählt werden, kann die Krümmungsbedingung (3.22) vernachlässigt werden und die Liniensuche anhand der Abstiegsbedingung (3.21) abgebrochen werden. Ein geeignetes Verfahren ist das *Backtracking*–Verfahren, das (in der einfachsten Form) durch den Algorithmus in Tabelle 3.2 beschrieben wird.

Das Vorgehen kann anhand von Abbildung 3.3 erklärt werden. Wenn der Startwert $\bar{\alpha}$ weit auf der rechten Seite liegt, wird α solange um den Faktor ρ reduziert, bis $\phi(\alpha) = f(\mathbf{x}^k + \alpha \mathbf{s}^k)$ unterhalb einer der Geraden (mit $c = 0.1$ oder 0.25) liegt. Die sukzessive Reduktion um einen Faktor ρ stellt sicher, dass α^k klein genug ist, um die Abstiegsbedingung (3.21) zu erfüllen. Andererseits wird durch die sukzessive Reduktion um den Faktor ρ sichergestellt, dass α^k *nicht zu klein* wird, was zu einer Stagnation des Verfahrens führen könnte.

Das Backtracking–Verfahren wird häufig in Zusammenhang mit Newton–Verfahren eingesetzt und die Startschrittweite zu $\bar{\alpha} = 1$ gewählt. Für Quasi–Newton– oder Konjugierte–Gradienten–Verfahren (siehe Abschnitt 3.3.3) ist dieses Verfahren allerdings weniger gut geeignet.

Initialisierung: $\alpha \leftarrow \bar{\alpha} > 0$ (Startwert)
 $\rho \in (0, 1)$ (Backtracking–Parameter)
 c (Abstiegsbedingung)

repeat

$\alpha \leftarrow \rho \alpha$

until $f(\mathbf{x}^k + \alpha \mathbf{s}^k) \leq f(\mathbf{x}^k) + c \alpha \nabla f(\mathbf{x}^k)^T \mathbf{s}^k$

$\alpha^k \leftarrow \alpha$ (Schrittweite)

Tabelle 3.2: Backtracking–Verfahren.

3.3.2.3 Intervallschachtelungsverfahren (“Goldener Schnitt”)

Ein alternatives Verfahren zur Bestimmung der Schrittweite α^k ist das *Intervallschachtelungsverfahren*, das eine konvergierende Folge von Intervallschachtelungen erzeugt, um das Minimum von $\phi(\alpha)$ einzugrenzen.

Zunächst muss ein Intervall $[\alpha_l, \alpha_r]$ gefunden werden, in dem die Funktion $\phi(\alpha^k)$ ein Minimum aufweist, siehe Abbildung 3.5. Dies kann z.B. dadurch erreicht werden, dass mit einem hinreichend kleinen α_l gestartet wird und α_r (ausgehend von α_l) sukzessive vergrößert wird, bis der Funktionswert $\phi(\alpha_r)$ anfängt zuzunehmen.

Anschließend werden zwei innere Werte $\alpha_l < \hat{\alpha}_l < \hat{\alpha}_r < \alpha_r$ durch die linearen Beziehungen

$$\hat{\alpha}_l = \alpha_l + (1 - a)(\alpha_r - \alpha_l), \quad \hat{\alpha}_r = \alpha_l + a(\alpha_r - \alpha_l) \quad (3.23)$$

mit dem Parameter $a \in (\frac{1}{2}, 1)$ bestimmt und die Funktionswerte $\phi(\hat{\alpha}_l)$ und $\phi(\hat{\alpha}_r)$ berechnet. Falls $\phi(\hat{\alpha}_l) < \phi(\hat{\alpha}_r)$ gilt, so liegt das Minimum offenbar im Intervall $[\alpha_l, \hat{\alpha}_r]$ und der

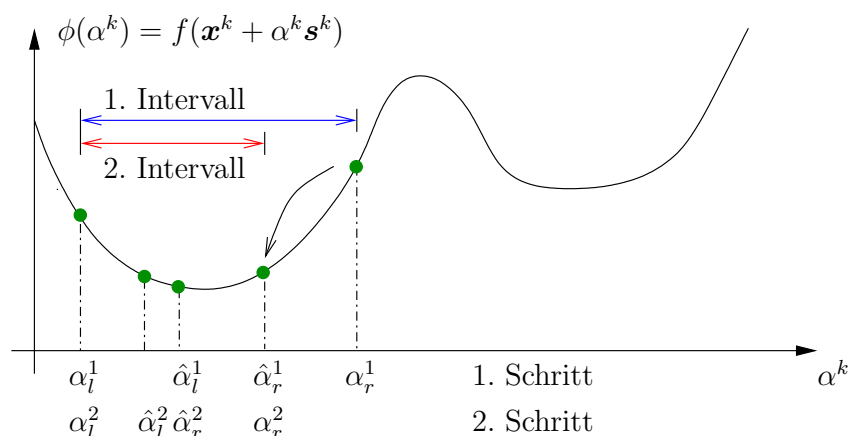


Abbildung 3.5: Veranschaulichung der Intervallhalbierungsmethode.

äußere rechte Punkt α_r wird verworfen (siehe “1. Schritt” in Abbildung 3.5). Falls andererseits $\phi(\hat{\alpha}_l) > \phi(\hat{\alpha}_r)$ gilt, so muss das Minimum im Intervall $[\hat{\alpha}_l, \alpha_r]$ liegen und der linke äußere Punkt α_l wird verworfen. Im nächsten Iterationsschritt wird somit das verkürzte Intervall $[\alpha_l, \hat{\alpha}_r]$ bzw. $[\hat{\alpha}_l, \alpha_r]$ betrachtet und die Intervallschachtelung erneut angewendet.

Wenn der Parameter a zu

$$a = \frac{\sqrt{5} - 1}{2} \approx 0.618 \quad (3.24)$$

gewählt wird, ist lediglich eine weitere Funktionsauswertung notwendig, um einen neuen inneren Punkt zu bestimmen (siehe “2. Schritt”, $\hat{\alpha}_r^2 = \hat{\alpha}_l^1$, in Abbildung 3.5). Aus diesem Grund wird diese Wahl von a als der *Goldene Schnitt* bezeichnet. Pro Schritt wird das ursprüngliche Intervall somit um ca. 38% reduziert. Tabelle 3.3 fasst den Algorithmus zusammen.

Initialisierung:	α_l, α_r	(Startintervall mit innerem Minimum)
	$a = 0.618$	(Goldener Schnitt-Parameter)
	$\varepsilon_\alpha, \varepsilon_\phi$	(Abbruchskriterien)
	$\hat{\alpha}_l \leftarrow \alpha_l + (1 - a)(\alpha_r - \alpha_l)$	(innere Punkte)
	$\hat{\alpha}_r \leftarrow \alpha_l + a(\alpha_r - \alpha_l)$	
repeat		
	if $\phi(\hat{\alpha}_l) < \phi(\hat{\alpha}_r)$ do	
	$\alpha_r \leftarrow \hat{\alpha}_r$	
	$\hat{\alpha}_r \leftarrow \hat{\alpha}_l$	
	$\hat{\alpha}_l \leftarrow \alpha_l + (1 - a)(\alpha_r - \alpha_l)$	
	else (d.h. $\phi(\hat{\alpha}_l) \geq \phi(\hat{\alpha}_r)$)	
	$\alpha_l \leftarrow \hat{\alpha}_l$	
	$\hat{\alpha}_l \leftarrow \hat{\alpha}_r$	
	$\hat{\alpha}_r \leftarrow \alpha_l + a(\alpha_r - \alpha_l)$	
	end	
until	$ \alpha_r - \alpha_l \leq \varepsilon_\alpha$ or $ \phi(\alpha_r) - \phi(\alpha_l) \leq \varepsilon_\phi$	

Tabelle 3.3: Intervallschachtelungsverfahren (“Goldener Schnitt”).

Abschließend kann α^k entweder aus der Mittelung $\alpha^k = (\alpha_l + \alpha_r)/2$ oder aus einer quadratischen Interpolation zwischen den kleinsten drei der vier Funktionswerte ϕ von $(\alpha_l, \hat{\alpha}_l, \hat{\alpha}_r, \alpha_r)$ gewonnen werden.

Das Intervallschachtelungsverfahren ist ein einfaches und robustes Verfahren, das allerdings meistens mehr Iterationen benötigt als andere Verfahren, um eine geeignete Schrittweite α^k zu bestimmen.

3.3.3 Wahl der Suchrichtung

Zusätzlich zur Schrittweite α^k ist die Bestimmung der Suchrichtung $\mathbf{s}^k$ von elementarer Bedeutung, um den nächsten Iterationspunkt

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k \mathbf{s}^k \quad (3.25)$$

mit einem möglichst großen Abstieg in der Kostenfunktion f zu bestimmen. Dabei existieren unterschiedliche Verfahren zur Bestimmung der Suchrichtung $\mathbf{s}^k$, die sich hinsichtlich Genauigkeit, Konvergenzgeschwindigkeit und Rechenaufwand unterscheiden. Dieser Abschnitt stellt einige der bekanntesten Verfahren erster und zweiter Ordnung vor, also Verfahren, die Informationen über den Gradienten ∇f , bzw. zusätzlich noch über die Hessematrix $\nabla^2 f$ verwenden. Für Verfahren nullter Ordnung, die lediglich die Kostenfunktion f an sich verwenden, sei auf Abschnitt 3.4.2 verwiesen.

3.3.3.1 Gradientenverfahren

Die einfachste Methode, um eine Suchrichtung zu bestimmen, besteht darin, den negativen Gradienten an der Stelle $\mathbf{x}^k$ zu verwenden,

$$\mathbf{s}^k = -\nabla f(\mathbf{x}^k), \quad (3.26)$$

weshalb dieses Verfahren auch *Gradientenverfahren* genannt wird. Es ist leicht zu verifizieren, dass die Ungleichung (3.18) für nicht-stationäre Punkte ($\nabla f(\mathbf{x}^k) \neq \mathbf{0}$) erfüllt ist:

$$-\nabla f(\mathbf{x}^k)^T \nabla f(\mathbf{x}^k) = -\|\nabla f(\mathbf{x}^k)\|^2 < 0. \quad (3.27)$$

Da der negative Gradient die Ungleichung (3.18) minimiert, spricht man von der Richtung des *steilsten Abstiegs* (Englisch: *steepest descent*). Der Gradient steht dabei orthogonal zu der Höhenlinie $f(\mathbf{x}^k) = \text{konst.}$, die durch den Punkt $\mathbf{x}^k$ verläuft (Abbildungen 3.6 und 3.7).

Wenn ein Problem gut konditioniert ist (z.B. $f(\mathbf{x}) = x_1^2 + x_2^2$), zeigt der negative Gradient in die Richtung des Minimums $\mathbf{x}^*$, siehe Abbildung 3.6. In diesem Fall konvergiert das Gradientenverfahren schnell zur optimalen Lösung. Wenn andererseits das Problem schlecht konditioniert ist und der negative Gradient nicht in die Nähe des Minimums $\mathbf{x}^*$ weist, konvergiert das Gradientenverfahren sehr langsam, wie in Abbildung 3.7 dargestellt ist. Dies wird auch aus dem folgendem Satz deutlich:

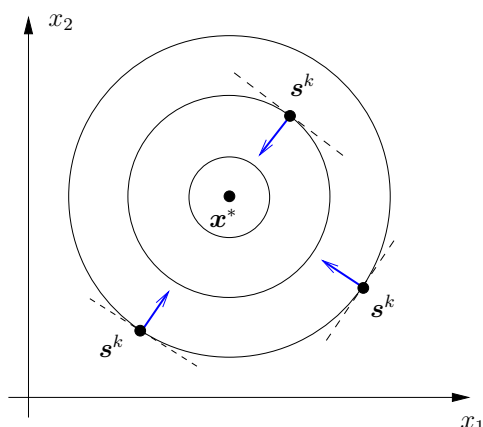


Abbildung 3.6: Beispiel eines gut konditionierten Problems für das Gradientenverfahren.

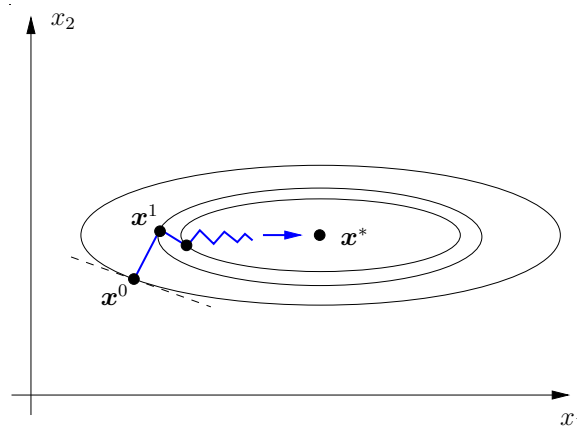


Abbildung 3.7: Beispiel eines schlecht konditionierten Problems für das Gradientenverfahren.

Satz 3.4 (Konvergenzrate des Gradientenverfahrens)

Für quadratische Funktionen $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top \mathbf{Q}\mathbf{x} + \mathbf{b}^\top \mathbf{x} + \mathbf{c}$ mit positiv definiter Matrix $\mathbf{Q}$ konvergiert das Gradientenverfahren bei exakter Lösung des Liniensuchproblems (3.20) gemäß

$$f(\mathbf{x}^{k+1}) - f(\mathbf{x}^*) \leq \left(\frac{\kappa - 1}{\kappa + 1} \right)^2 [f(\mathbf{x}^k) - f(\mathbf{x}^*)], \quad \kappa = \frac{\lambda_{\max}}{\lambda_{\min}}, \quad (3.28)$$

wobei $\lambda_{\min}$ und $\lambda_{\max}$ der kleinste und größte Eigenwert von $\mathbf{Q}$ sind.

Die Zahl κ stellt die spektrale Konditionszahl der Matrix $\mathbf{Q}$ dar und beeinflusst direkt die Konvergenzgeschwindigkeit, wie in der folgenden Aufgabe untersucht werden soll.

Aufgabe 3.1 Untersuchen Sie die Konvergenzrate (3.28) des Gradientenverfahrens für das quadratische Problem

$$\min_{\mathbf{x} \in \mathbb{R}^2} f(\mathbf{x}) = \frac{1}{2} \left(\frac{x_1 - x_1^*}{a} \right)^2 + \frac{1}{2} (x_2 - x_2^*)^2 \quad (3.29)$$

mit den Werten $a = 1$ und $a = 5$ (siehe Abbildung 3.6 und 3.7).

Die Vor- und Nachteile des Gradientenverfahrens sind im Folgenden zusammengefasst:

- + größerer Konvergenzbereich als beim Newton-Verfahren
($\nabla^2 f(\mathbf{x}^k) > \mathbf{0}$ nicht erforderlich)
- + einfaches Verfahren
- langsame Konvergenz bei schlecht konditionierten / schlecht skalierten Problemen
- begrenzte Genauigkeit (Verfahren erster Ordnung)

3.3.3.2 Konjugierte-Gradienten-Verfahren

Das *Konjugierte-Gradienten-Verfahren* ist eine bedeutende Alternative zum herkömmlichen Gradientenverfahren, da der Rechenaufwand nicht viel höher ist, die Effizienz und Konvergenzeigenschaften aber deutlich besser sind als beim Gradientenverfahren.

Das Konjugierte-Gradienten-Verfahren nützt Informationen des vorherigen Schrittes aus, um die Suchrichtung $\mathbf{s}^k$ zu bestimmen:

$$\mathbf{s}^0 = -\nabla f(\mathbf{x}^0) \quad (3.30a)$$

$$\mathbf{s}^k = -\nabla f(\mathbf{x}^k) + \beta^k \mathbf{s}^{k-1}. \quad (3.30b)$$

Der Start des Verfahrens erfolgt mit dem steilsten Abstieg $-\nabla f(\mathbf{x}^0)$, während bei allen weiteren Schritten gemäß (3.30b) ein Korrekturterm in Abhängigkeit der vorherigen Suchrichtung $\mathbf{s}^{k-1}$ dazukommt. Der Faktor β^k wird durch die Beziehung

$$\beta^k = \frac{\|\nabla f(\mathbf{x}^k)\|^2}{\|\nabla f(\mathbf{x}^{k-1})\|^2} \quad (3.31)$$

aktualisiert, die von *Fletcher* und *Reeves* vorgeschlagen wurde. Weitere Formeln sind in den am Ende des Kapitels aufgelisteten Textbüchern angegeben.

Konjugierte-Gradienten-Verfahren verzichten prinzipiell auf aufwendige Matrixoperationen und sind somit gut auf große Probleme anwendbar, bei denen auf Speichereffizienz geachtet werden muss. Des Weiteren kann gezeigt werden, dass bei quadratischen Problemen der Form

$$f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^\top \mathbf{Q} \mathbf{x} + \mathbf{b}^\top \mathbf{x} + c, \quad \mathbf{x} \in \mathbb{R}^n$$

mit positiv definiter Hessematrix $\mathbf{Q}$ und exakter Lösung des Liniensuchproblems (3.20) das Konjugierte-Gradienten-Verfahren nach höchstens n Iterationen konvergiert.

Aufgabe 3.2 Verifizieren Sie, dass das Konjugierte-Gradienten-Verfahren tatsächlich eine Abstiegsrichtung $\mathbf{s}^k$ liefert, die die Ungleichung (3.18) erfüllt.

3.3.3.3 Newton-Verfahren

Ein Sonderfall des Liniensuchverfahrens liegt vor, wenn die Suchrichtung $\mathbf{s}^k$ so gewählt wird, dass die Iterationsvorschrift $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k \mathbf{s}^k$ dem Newton-Verfahren entspricht. In diesem Fall spricht man von der *Newton-Richtung* $\mathbf{s}^k$.

Zunächst wird die Kostenfunktion $f(\mathbf{x}) = f(\mathbf{x}^k + \mathbf{s})$ an der Stelle $\mathbf{x}^k$ durch ein quadratisches Modell

$$m^k(\mathbf{s}) = f(\mathbf{x}^k) + \nabla f(\mathbf{x}^k)^\top \mathbf{s} + \frac{1}{2} \mathbf{s}^\top \nabla^2 f(\mathbf{x}^k) \mathbf{s} \quad (3.32)$$

approximiert, für das das Minimum $\mathbf{s}$ gesucht wird. Durch Ableitung von (3.32) nach $\mathbf{s}^k$ ergibt sich

$$\nabla f(\mathbf{x}^k) + \nabla^2 f(\mathbf{x}^k) \mathbf{s} = \mathbf{0}. \quad (3.33)$$

Unter der Annahme, dass $\nabla^2 f(\mathbf{x}^k)$ positiv definit ist, lautet die Newton-Richtung $\mathbf{s}^k = \mathbf{s}$ an der Stelle $\mathbf{x}^k$ somit

$$\mathbf{s}^k = -[\nabla^2 f(\mathbf{x}^k)]^{-1} \nabla f(\mathbf{x}^k). \quad (3.34)$$

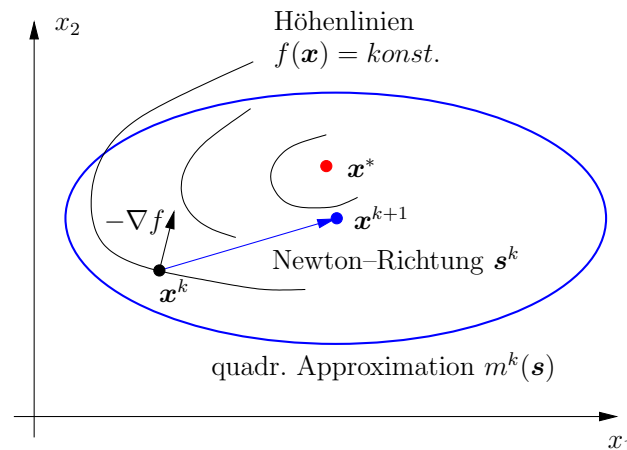


Abbildung 3.8: Veranschaulichung der Newton-Richtung $\mathbf{s}^k$ im Vergleich zur steilsten Abstiegsrichtung $-\nabla f(\mathbf{x}^k)$.

Des Weiteren ist leicht zu überprüfen, dass die Ungleichung (3.18)

$$\nabla f(\mathbf{x}^k)^\top \mathbf{s}^k = -\nabla f(\mathbf{x}^k)^\top [\nabla^2 f(\mathbf{x}^k)]^{-1} \nabla f(\mathbf{x}^k) < 0. \quad (3.35)$$

stets erfüllt ist, da für positiv definite $\nabla^2 f(\mathbf{x}^k)$ die Inverse ebenfalls positiv definit ist.

Mit Hilfe der Newton-Richtung (3.34) ergibt sich der neue Iterationspunkt $\mathbf{x}^{k+1} = \mathbf{x}^k + \mathbf{s}^k$, was einer “natürlichen” Schrittweite $\alpha^k = 1$ für die Iterationsvorschrift (3.25) des Liniensuchverfahren entspricht. Die meisten Varianten des Newton-Verfahrens verwenden deshalb standardmäßig eine Schrittweite von $\alpha^k = 1$, die lediglich reduziert wird, wenn der Abstieg in der Kostenfunktion $f(\mathbf{x}^{k+1})$ unterhalb einer Schranke liegt.

Das Newton-Verfahren ist ein Verfahren zweiter Ordnung, da die Approximation (3.32) die Fehlerordnung $\mathcal{O}(\|\mathbf{s}\|^3)$ besitzt. Es ist zuverlässig und auch für hoch-dimensionale Systeme gut geeignet, wenn die quadratische Approximation der Funktion f an der Stelle $\mathbf{x}^k$ hinreichend genau ist. Abbildung 3.8 illustriert die quadratische Approximationsgüte des Newton-Verfahrens im Vergleich zur steilsten Abstiegsrichtung $-\nabla f$ des Gradientenverfahrens.

Aufgabe 3.3 Zeigen Sie, dass das Newton-Verfahren für quadratische Probleme

$$\min_{\mathbf{x} \in \mathbb{R}^n} \frac{1}{2} \mathbf{x}^\top \mathbf{Q} \mathbf{x} + \mathbf{b}^\top \mathbf{x} + c \quad (\mathbf{Q} \text{ positiv definit}) \quad (3.36)$$

unabhängig vom Startpunkt $\mathbf{x}^0$ innerhalb von einem Iterationsschritt konvergiert.

Die Vor- und Nachteile des Newton-Verfahrens sind im Folgenden zusammengefasst:

- + quadratische Konvergenz in der Nähe von $\mathbf{x}^*$ (innerhalb weniger Iterationen)
- + hohe Genauigkeit (Verfahren zweiter Ordnung)
- nur lokale Konvergenz ($\nabla^2 f(\mathbf{x}) > \mathbf{0}$ im Allg. nicht für alle $\mathbf{x}$ gewährleistet)
- aufwendige Berechnung der Hessematrix $\nabla^2 f$, insbesondere bei großen Problemen

Um globale Konvergenz des Newton–Verfahrens zu erreichen, gibt es verschiedene Ansätze, die die Positiv–Definitheit der Hessematrix gewährleisten (z.B. durch Modifikationen), auf die an dieser Stelle aber nicht näher eingegangen wird.

Die aufwendige Berechnung der Hessematrix $\nabla^2 f$ kann durch Approximationen (z.B. finite Differenzen) oder Quasi–Newton–Verfahren vermieden werden, wie im nächsten Abschnitt beschrieben wird.

3.3.3.4 Quasi–Newton–Verfahren

Quasi–Newton–Verfahren stellen eine attraktive Alternative zum reinen Newton–Verfahren dar, da sie die aufwendige Berechnung der Hessematrix $\nabla^2 f$ und ihrer Inversen vermeiden. Statt dessen wird die Hessematrix approximiert und während der einzelnen Iterationen aktualisiert. Dazu werden Informationen über die Gradienten an den einzelnen Iterationspunkten ausgenutzt, die Informationen über die zweite Ableitung entlang der Suchrichtung liefern.

Um dieses Vorgehen zu illustrieren, wird der Gradient ∇f an der Stelle $\mathbf{x}^k$ durch ein lineares Modell approximiert:

$$\nabla f(\mathbf{x}^k + \mathbf{s}) \approx \nabla f(\mathbf{x}^k) + \nabla^2 f(\mathbf{x}^k) \mathbf{s}. \quad (3.37)$$

Für hinreichend kleine $\mathbf{s} = \mathbf{x}^{k-1} - \mathbf{x}^k$ ergibt sich somit eine Approximationsgleichung für den Gradienten am vorherigen Iterationspunkt $\mathbf{x}^{k-1}$:

$$\nabla f(\mathbf{x}^{k-1}) \approx \nabla f(\mathbf{x}^k) + \nabla^2 f(\mathbf{x}^k)(\mathbf{x}^{k-1} - \mathbf{x}^k). \quad (3.38)$$

Aus diesem Zusammenhang wird ersichtlich, dass die Hessematrix an dem Punkt $\mathbf{x}^k$ durch eine Matrix $\mathbf{H}^k \approx \nabla^2 f(\mathbf{x}^k)$ angenähert werden kann, die die folgende sogenannte *Sekantengleichung* erfüllen muss:

$$\mathbf{H}^k(\mathbf{x}^k - \mathbf{x}^{k-1}) = \nabla f(\mathbf{x}^k) - \nabla f(\mathbf{x}^{k-1}). \quad (3.39)$$

Da es sich hierbei um ein unterbestimmtes Problem handelt ($\mathbf{H}^k$ hat n^2 Elemente, aber (3.39) nur n Gleichungen), werden typischerweise weitere Bedingungen an $\mathbf{H}^k$ gestellt, z.B. Symmetrie und minimale Differenz zwischen $\mathbf{H}^k$ und $\mathbf{H}^{k-1}$. Diese zusätzliche Bedingungen lassen sich zusammen mit (3.39) in einem Optimierungsproblem für $\mathbf{H}^k$ formulieren, dass analytisch gelöst werden kann.

Eine der bekanntesten Formeln zur Aktualisierung von $\mathbf{H}^k$ ist die sogenannte *BFGS–Formel* (nach *Broyden, Fletcher, Goldfarb, Shanno*)

$$\mathbf{H}^k = \mathbf{H}^{k-1} - \frac{\mathbf{H}^{k-1} \mathbf{d}^k (\mathbf{d}^k)^\top \mathbf{H}^{k-1}}{(\mathbf{d}^k)^\top \mathbf{H}^{k-1} \mathbf{d}^k} + \frac{\mathbf{y}^k (\mathbf{y}^k)^\top}{(\mathbf{d}^k)^\top \mathbf{y}^k} \quad (3.40)$$

mit den Abkürzungen

$$\mathbf{d}^k = \mathbf{x}^k - \mathbf{x}^{k-1}, \quad \mathbf{y}^k = \nabla f(\mathbf{x}^k) - \nabla f(\mathbf{x}^{k-1}). \quad (3.41)$$

Die Initialisierung des Quasi–Newton–Verfahrens an der Stelle $\mathbf{x}^0$ erfolgt über eine positiv definite Schätzung $\mathbf{H}^0 \approx \nabla^2 f(\mathbf{x}^0)$ oder im einfachsten Fall mit Hilfe der Einheitsmatrix $\mathbf{H}^0 = \mathbf{I}$.

Da ein Newton–Verfahren die inverse Hessematrix benötigt, um die Suchrichtung $\mathbf{s}^k$ zu berechnen, siehe (3.34), ist es oft sinnvoller, anstelle von $\mathbf{H}^k$ eine Approximation der inversen Hessematrix $\mathbf{B}^k = (\mathbf{H}^k)^{-1}$ zu bestimmen. Die zu der BFGS–Formel (3.40) äquivalente Formel lautet

$$\mathbf{B}^k = [\mathbf{I} - \rho^k \mathbf{d}^k (\mathbf{y}^k)^\top] \mathbf{B}^{k-1} [\mathbf{I} - \rho^k \mathbf{y}^k (\mathbf{d}^k)^\top] + \rho^k \mathbf{d}^k (\mathbf{d}^k)^\top, \quad \rho^k = \frac{1}{(\mathbf{y}^k)^\top \mathbf{d}^k}. \quad (3.42)$$

Somit ergibt sich analog zur Newton–Richtung (3.34) die Suchrichtung

$$\mathbf{s}^k = -\mathbf{B}^k \nabla f(\mathbf{x}^k). \quad (3.43)$$

Quasi–Newton–Verfahren zeichnen sich durch eine schnelle Konvergenz aus, erreichen aber in der Regel nicht die Konvergenzgeschwindigkeit des reinen Newton–Verfahrens. Dafür sind die Vektor–Matrizen–Multiplikationen zur Berechnung von $\mathbf{H}^k$ bzw. $\mathbf{B}^k$ deutlich schneller auszuführen als die Berechnung der Hessematrix $\nabla^2 f$ und ihrer Inversen in (3.34).

Ein weiterer Vorteil gegenüber dem Newton–Verfahren ist, dass Singularitätsprobleme theoretisch nicht auftreten können, da zumindest bei exakter Lösung des Liniensuchproblems (3.20) die Positiv–Definitheit von $\mathbf{H}^k$ bzw. $\mathbf{B}^k$ gewährleistet ist.

Ähnlich wie bei den Konjugierten–Gradienten–Verfahren kann gezeigt werden, dass Quasi–Newton–Verfahren bei quadratischen Problemen

$$f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^\top \mathbf{Q} \mathbf{x} + \mathbf{b}^\top \mathbf{x} + c, \quad \mathbf{x} \in \mathbb{R}^n$$

mit positiv definiter Hessematrix $\mathbf{Q}$ und exakter Lösung des Liniensuchproblems (3.20) nach höchstens n Iterationen konvergieren.

3.4 Weitere numerische Verfahren

In Folgenden werden weitere Verfahren zur numerischen Lösung von unbeschränkten Optimierungsproblemen vorgestellt. Die *Methode der Vertrauensbereiche* besitzt Ähnlichkeit zu den Liniensuchverfahren im vorherigen Abschnitt, während *direkte Suchverfahren* ohne die Auswertung von Gradienten auskommen.

3.4.1 Methode der Vertrauensbereiche

Im Gegensatz zu den Liniensuchverfahren in Abschnitt 3.3 konstruiert die *Methode der Vertrauensbereiche* (Englisch: *trust region method*) ein lokales vereinfachtes Modell $m^k(\mathbf{s}) \approx f(\mathbf{x}^k)$ der Kostenfunktion $f(\mathbf{x}^k)$ an der Stelle $\mathbf{x}^k$, dass anstelle des ursprünglichen Optimierungsproblems (3.1) gelöst wird:

$$\min_{\|\mathbf{s}\| \leq \Delta^k} m^k(\mathbf{s}) \quad \text{mit} \quad m^k(\mathbf{s}) \approx f(\mathbf{x}^k) \quad \text{für} \quad \|\mathbf{s}\| \leq \Delta^k. \quad (3.44)$$

Da das Modell $m^k(\mathbf{s})$ in der Regel nur lokal (für hinreichend kleine $\mathbf{s}$) genau ist, wird die Suche nach dem Minimum von $m^k(\mathbf{s})$ auf einen *Vertrauensbereich* $\|\mathbf{s}\| \leq \Delta^k$ beschränkt.

Das Näherungsmodell m^k wird im Allgemeinen durch eine quadratische Funktion der Form

$$m^k(\mathbf{s}) = \frac{1}{2} \mathbf{s}^\top \mathbf{H}^k \mathbf{s} + \nabla f(\mathbf{x}^k)^\top \mathbf{s} + f(\mathbf{x}^k). \quad (3.45)$$

angesetzt, wobei die Matrix $\mathbf{H}^k$ entweder durch die Hessematrix $\nabla^2 f(\mathbf{x}^k)$ oder eine Approximation davon gegeben ist. Die Lösung $\mathbf{s}^k$ des neuen Optimierungsproblems (3.44) führt zu einem “Kandidaten” für den nächsten Punkt $\mathbf{x}^{k+1} = \mathbf{x}^k + \mathbf{s}^k$. Falls der entsprechende Funktionswert $f(\mathbf{x}^{k+1})$ nicht zu einer hinreichenden Reduktion des vorherigen Kostenfunktionswertes $f(\mathbf{x}^k)$ führt, wird der Vertrauensbereich Δ^k verkleinert und (3.45) erneut gelöst.

Während jeder dieser Iterationen mit abnehmendem Vertrauensbereich Δ^k wird der Abstand zwischen $\mathbf{x}^k$ und dem neuen Kandidaten $\mathbf{x}^{k+1}$ kleiner werden und $\mathbf{s}^k$ in der Regel in eine andere Richtung weisen. Somit unterscheidet sich die Methode der Vertrauensbereiche von den bisherigen Liniensuchverfahren in der Art und Weise, wie die Richtung und Schrittweite zum nächsten Iterationspunkt gewählt wird:

- Liniensuchverfahren fixieren die Suchrichtung $\mathbf{s}^k$ und bestimmen dann eine geeignete Schrittweite α^k .
- In der Methode der Vertrauensbereiche wird zunächst ein maximaler Vertrauensbereich Δ^k gewählt und dann *simultan* die geeignete Richtung und Abstand (also $\mathbf{s}^k$) zum nächsten Iterationspunkt $\mathbf{x}^{k+1}$ bestimmt. Gegebenenfalls wird der Vertrauensbereich verkleinert und die Suche wiederholt.

Der Vertrauensbereich Δ^k wird in jedem Iterationsschritt in Abhängigkeit der Übereinstimmung des Modells m^k und der Kostenfunktion f angepasst. Dazu wird das Verhältnis

$$\rho^k(\mathbf{s}^k) = \frac{f(\mathbf{x}^k) - f(\mathbf{x}^k + \mathbf{s}^k)}{m^k(\mathbf{0}) - m^k(\mathbf{s}^k)} \quad (3.46)$$

eingeführt, das ein Maß für die *Modellgüte* darstellt. Der Zählerterm bezeichnet den tatsächlichen Abstieg in der Kostenfunktion, während der Nennerterm den prädizierten Abstieg wiedergibt. Für $\rho^k \approx 1$ besteht eine gute Übereinstimmung zwischen Modell und Realität in dem momentanen Iterationsschritt, so dass der Vertrauensbereich Δ^k im nächsten Schritt vergrößert werden kann. Falls stattdessen $\rho^k \ll 1$ gilt, sollte Δ^k im nächsten Schritt reduziert werden. Der grundlegende Algorithmus für die Methode der Vertrauensbereiche ist in Tabelle 3.4 angegeben.

Initialisierung: $\bar{\Delta}, \Delta^0 \in (0, \bar{\Delta})$ (Vertrauensbereich: Grenz- & Startwert)
 $\eta \in [0, \frac{1}{4})$ (Parameter)
 $k \leftarrow 0$ (Iterationsindex)
 $\varepsilon_x, \varepsilon_f$ (Abbruchkriterien)

repeat

$m^k(s) \leftarrow (3.45)$ (Modell)
 $s^k \leftarrow \arg \min \{m^k(s) : \|s\| \leq \Delta^k\}$ (evtl. approximativ gelöst)
 $\rho^k \leftarrow (3.46)$ (Modellgüte)

if $\rho^k < \frac{1}{4}$ **do**
 $\Delta^{k+1} \leftarrow \frac{1}{4}\Delta^k$ (Reduktion)
else if $\rho^k > \frac{3}{4}$ **and** $\|s^k\| = \Delta^k$ **do**
 $\Delta^{k+1} \leftarrow \min\{2\Delta^k, \bar{\Delta}\}$ (Vergrößerung)
else
 $\Delta^{k+1} \leftarrow \Delta^k$
end if

if $\rho^k > \eta$ **do**
 $x^{k+1} \leftarrow x^k + s^k$ (nächster Schritt)
 $H^{k+1} \leftarrow H^k + \dots$ (Aktualisierung der Hessematrix)
else
 $x^{k+1} \leftarrow x^k$ (Schritt mit $\Delta^{k+1} < \Delta^k$ wiederholen)
end if

$k \leftarrow k + 1$

until $\|x^k - x^{k-1}\| \leq \varepsilon_x$ **or** $|f(x^k) - f(x^{k-1})| \leq \varepsilon_f$

Tabelle 3.4: Methode der Vertrauensbereiche.

3.4.2 Direkte Suchverfahren

Die vorhergegangenen sogenannten *ableitungsbehafteten* Lösungsverfahren verwendeten Informationen der Gradienten ∇f (und der Hessematrix $\nabla^2 f$ falls verfügbar), um mittels einer Iterationsvorschrift einen neuen Punkt x^{k+1} zu bestimmen, der einen hinreichenden Abstieg in der Kostenfunktion $f(x^{k+1}) < f(x^k)$ ermöglicht.

Allerdings sind in manchen praktischen Fällen die partiellen Ableitungen nicht verfügbar bzw. können nicht approximiert oder mit vertretbarem Aufwand berechnet werden, da das betrachtete Problem zu komplex bzw. nicht stetig differenzierbar ist. Abhilfe verschaffen in diesem Fall sogenannte *direkte* oder *ableitungsfreie Suchverfahren*, die mit Hilfe von Stichproben eine Reihe von Funktionswerten berechnen, um daraus einen neuen Iterationspunkt x^{k+1} zu bestimmen.

Eines der bekanntesten und gleichzeitig einfachsten Verfahren in der nichtlinearen Optimierung ist das *Simplex-Verfahren* nach Nelder und Mead.² Der Algorithmus basiert auf einem *Simplex* im n -dimensionalen Parameterraum, das aus $n + 1$ Punkten $x^{k,i}$, $i = 1, \dots, n + 1$ aufgespannt wird (k ist der Iterationsindex). Für $n = 1$ ist dies eine Gerade,

² Es sei an dieser Stelle angemerkt, dass sich das Simplex-Verfahren nach Nelder und Mead grundsätzlich vom Simplex-Algorithmus in der Linearen Programmierung unterscheidet, der von G. Danzig eingeführt wurde!

für $n = 2$ ein Dreieck, usw.

Für die $n + 1$ Punkte $\mathbf{x}^{k,i}$ des Simplex werden die Funktionswerte $f(\mathbf{x}^{k,i})$ berechnet und der Größe nach umsortiert, so dass sich die aufsteigende Folge

$$f(\mathbf{x}^{k,1}) \leq f(\mathbf{x}^{k,2}) \leq \dots \leq f(\mathbf{x}^{k,n+1}) \quad (3.47)$$

ergibt. Der Algorithmus ersetzt dann den “schlechtesten” Punkt $\mathbf{x}^{k,n+1}$ durch einen neuen Punkt nach der Geradengleichung

$$g(a) = \bar{\mathbf{x}}^k + a(\bar{\mathbf{x}}^k - \mathbf{x}^{k,n+1}) \quad (3.48)$$

in Abhängigkeit eines Parameters a . Der Punkt $\bar{\mathbf{x}}^k$ stellt den Mittelpunkt (bzw. Schwerpunkt) des restlichen Simplex dar:

$$\bar{\mathbf{x}}^k = \frac{1}{n} \sum_{i=1}^n \mathbf{x}^{k,i}. \quad (3.49)$$

Das Simplex-Verfahren verwendet verschiedene Abbildungen mittels der Geradengleichung (3.48), um ein geeignetes neues Simplex zu finden: *Reflexion*, *Expansion*, *innere/äußere Kontraktion* und *Schrumpfung*.

- a) **Reflexion** (siehe Abbildung 3.9a):
Bestimme den Reflexionspunkt $\mathbf{x}_r^k = g(1)$ und $f_r^k = f(\mathbf{x}_r^k)$.
Falls $f(\mathbf{x}^{k,1}) \leq f_r^k < f(\mathbf{x}^{k,n})$, setze $\mathbf{x}^{k,n+1} = \mathbf{x}_r^k$.
- b) **Expansion** (siehe Abbildung 3.9b):
Falls $f_r^k < f(\mathbf{x}^{k,1})$, berechne $\mathbf{x}_e^k = g(2)$ und $f_e^k = f(\mathbf{x}_e^k)$.
Falls $f_e^k < f_r^k$, setze $\mathbf{x}^{k,n+1} = \mathbf{x}_e^k$. Andernfalls setze $\mathbf{x}^{k,n+1} = \mathbf{x}_r^k$.
- c) **äußere Kontraktion** (siehe Abbildung 3.9c):
Falls $f(\mathbf{x}^{k,n}) \leq f_r^k < f(\mathbf{x}^{k,n+1})$, berechne $\mathbf{x}_c^k = g(1/2)$ und $f_c^k = f(\mathbf{x}_c^k)$.
Falls $f_c^k \leq f_r^k$, setze $\mathbf{x}^{k,n+1} = \mathbf{x}_c^k$. Andernfalls gehe zu Schritt e).
- d) **innere Kontraktion** (siehe Abbildung 3.9d):
Falls $f_r^k \geq f(\mathbf{x}^{k,n+1})$, berechne $\mathbf{x}_c^k = g(-1/2)$ und $f_c^k = f(\mathbf{x}_c^k)$.
Falls $f_c^k < f(\mathbf{x}^{k,n+1})$, setze $\mathbf{x}^{k,n+1} = \mathbf{x}_c^k$. Andernfalls gehe zu Schritt e).
- e) **Schrumpfung** (siehe Abbildung 3.9e):
Komprimiere den Simplex um den “besten” Punkt $\mathbf{x}^{k,1}$:
 $\mathbf{x}^{k,i} = \mathbf{x}^{k,1} - \frac{1}{2}(\mathbf{x}^{k,i} - \mathbf{x}^{k,1})$, $i = 2, \dots, n + 1$.

Der Algorithmus und die einzelnen Simplexoperationen sind in Tabelle 3.5 und in Abbildung 3.9 dargestellt. Das Simplex-Verfahren wird iterativ fortgesetzt, bis ein Konvergenzkriterium erfüllt ist. Der Simplex wandert dabei in Richtung des Optimums und zieht sich zusammen. Allerdings ist die Konvergenz nicht garantiert und es kann vorkommen, dass das Simplex-Verfahren zu einem nicht-optimalen Punkt konvergiert.³ In der Praxis führt das Simplex-Verfahren aber häufig zu guten Ergebnissen, wenn auch nur mit mäßig schneller Konvergenz. Aufgrund der Robustheit und des einfachen Algorithmus ist es ein gebräuchliches Optimierungsverfahren.

³ In diesem Fall kann versucht werden, den Algorithmus mit einem neuen Simplex an der letzten Stelle neu zu starten.

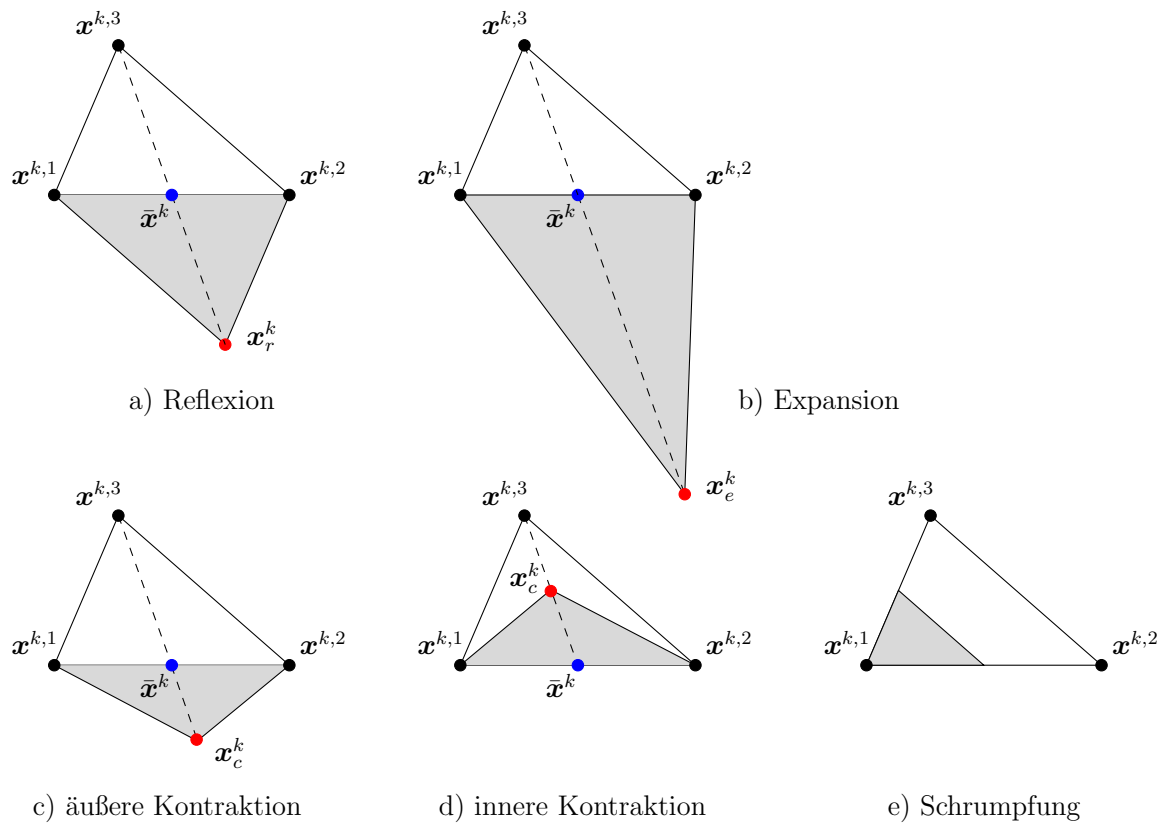


Abbildung 3.9: Operationen des Simplex-Verfahrens nach Nelder und Mead.

Initialisierung: $k \leftarrow 0$ (Iterationsindex)
 ε (Abbruchkriterium)

repeat

Sortierung (3.47)

$\bar{x}^k \leftarrow (3.49)$ (Mittelpunkt)

Austausch von $x^{k,n+1}$ (Reflexion/Expansion/Kontraktion)
bzw. von $x^{k,i}$, $i = 2, \dots, n+1$ (Schrumpfung)

$k \leftarrow k+1$

until $f(x^{k,n+1}) - f(x^{k,1}) \leq \varepsilon$

Tabelle 3.5: Simplex-Verfahren nach Nelder und Mead.

3.5 Beispiel: Rosenbrock's "Bananenfunktion"

Ein bekanntes Beispiel in der Optimierung ist das *Rosenbrock*-Problem

$$\min_{\mathbf{x} \in \mathbb{R}^2} f(\mathbf{x}) \quad \text{mit} \quad f(\mathbf{x}) = 100(x_2 - x_1^2)^2 + (x_1 - 1)^2. \quad (3.50)$$

Abbildung 3.10 zeigt das Profil und die Höhenlinien der Funktion, die auch als *Bananenfunktion* bezeichnet wird. Das Rosenbrock-Problem soll als Beispiel verwendet werden, um die Konvergenzeigenschaften der behandelten Verfahren numerisch mit Hilfe von MATLAB zu untersuchen.

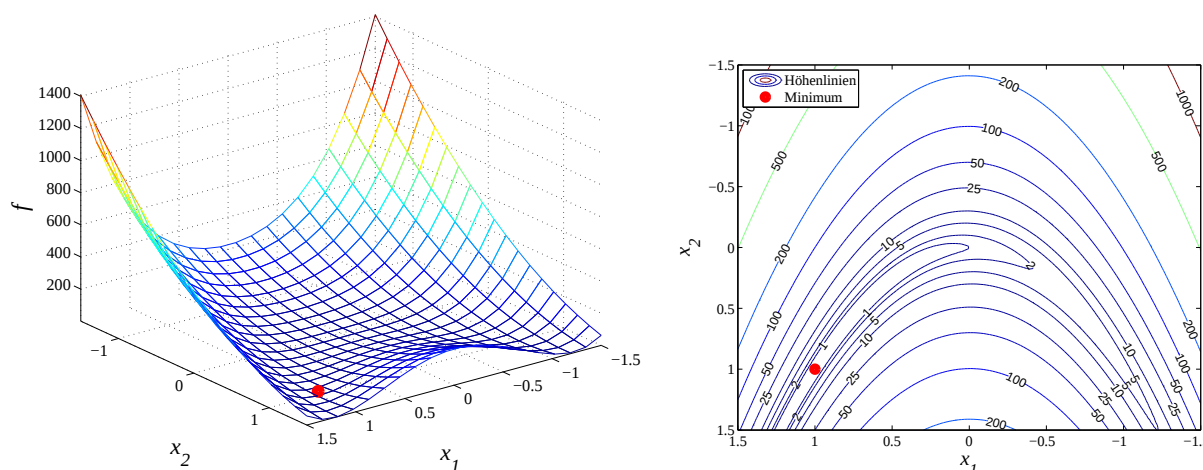


Abbildung 3.10: Profil und Höhenlinien von Rosenbrock's Bananenfunktion.

Aufgabe 3.4 Verifizieren Sie, dass der Punkt $\mathbf{x}^* = [1, 1]^T$ ein Minimum darstellt. Ist das Minimum $\mathbf{x}^*$ global und eindeutig? Sind die Funktion $f(\mathbf{x})$ und das Optimierungsproblem (3.50) konvex?

Zur Lösung von unbeschränkten Optimierungsproblemen stellt die *Optimization Toolbox* von MATLAB die folgenden Funktionen zur Verfügung

- **fminunc:** Liniensuche: Gradientenverfahren, Quasi-Newton-Verfahren
Methode der Vertrauensbereiche: Newton-Verfahren
- **fminsearch:** Simplex-Verfahren nach Nelder-Mead.

Eine empfehlenswerte Alternative ist die frei zugängliche MATLAB-Funktion **minFunc** [5], die eine große Auswahl an Liniensuchverfahren bietet. Tabelle 3.6 zeigt einige Vergleichsdaten für die numerische Lösung des Rosenbrock-Problems (ausgehend vom Startwert $\mathbf{x}^0 = [-1, -1]^T$), die mit Hilfe von **fminunc**, **fminsearch** und **minFunc** berechnet wurden.

Abbildung 3.11 stellt zusätzlich die Iterationsverläufe für die Verfahren dar, die unter **fminunc** und **fminsearch** implementiert sind. Der Vollständigkeit halber ist in Abbildung 3.13 am Ende des Kapitels der MATLAB-Code für das Rosenbrock-Problem (3.50) angegeben, um zu verdeutlichen, wie die einzelnen Optimierungsverfahren mit **fminunc** und **fminsearch** angesprochen werden können.

Verfahren	Funktion	Iter.	$f(\mathbf{x}^*)$	$\nabla f(\mathbf{x}^*)$	Funktionsaufrufe:		
					$f(\mathbf{x})$	$\nabla f(\mathbf{x})$	$\nabla^2 f(\mathbf{x})$
LS: Gradientenverfahren	fminunc	57	0.1232	1.1978	200	200	–
LS: Konj.-Grad.-Verf.	minFunc	31	$4.1 \cdot 10^{-17}$	$6.9 \cdot 10^{-9}$	91	90	–
LS: Newton-Verfahren	minFunc	21	$3.5 \cdot 10^{-17}$	$3.5 \cdot 10^{-8}$	29	28	24
LS: Quasi-Newton (BFGS)	fminunc	23	$5.4 \cdot 10^{-12}$	$9.2 \cdot 10^{-6}$	29	29	–
VB: Newton-Verfahren	fminunc	25	$2.2 \cdot 10^{-18}$	$2.1 \cdot 10^{-8}$	26	26	26
Nelder-Mead Simplex-Verf.	fminsearch	67	$5.3 \cdot 10^{-10}$	–	125	–	–

Tabelle 3.6: Vergleich der numerischen Verfahren für das Rosenbrock-Problem (LS=Linienuche, VB=Methode der Vertrauensbereiche).

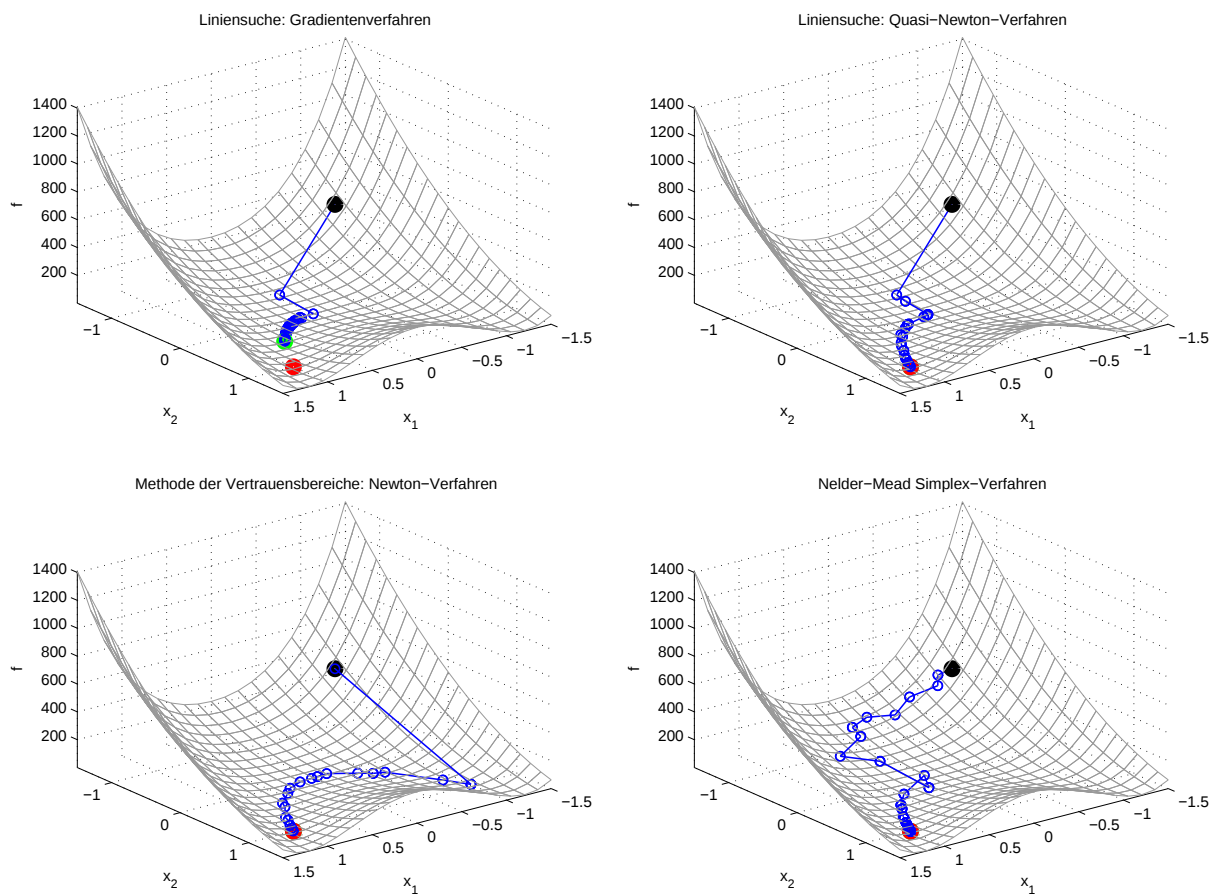


Abbildung 3.11: Rosenbrock's Bananenfunktion: Vergleich der numerischen Verfahren mit fminunc und fminsearch.

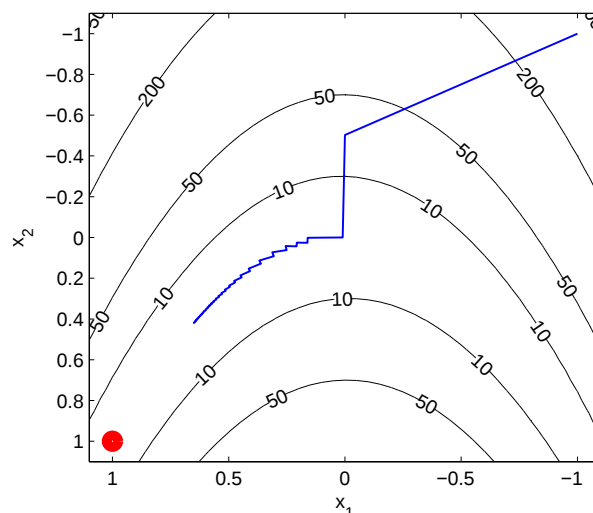


Abbildung 3.12: Darstellung der Iterationen des Gradientenverfahrens.

Beim Gradientenverfahren fällt die langsame Konvergenz auf, da auch nach dem Erreichen der maximalen Anzahl an Funktionsauswertungen (200) das Minimum noch nicht erreicht ist. In Abbildung 3.12 ist der Iterationsverlauf des Gradientenverfahrens noch einmal detailliert über die Höhenlinien der Rosenbrock-Funktion (3.50) dargestellt. Erkennbar ist, dass sich das Gradientenverfahren an der maximalen Abstiegsrichtung orientiert, die senkrecht zu der jeweiligen Höhenlinie verläuft. In Richtung des Minimums werden die Iterationsschritte immer kleiner. Die niedrige Konvergenzgeschwindigkeit wurde bereits in Abbildung 3.7 veranschaulicht und soll in der folgenden Aufgabe näher untersucht werden:

Aufgabe 3.5 Berechnen Sie für das Minimum $\mathbf{x}^* = [1, 1]^T$ des Rosenbrock-Problems (3.50) die Konvergenzrate des Gradientenverfahrens gemäß Satz 3.4.

Das Konvergenzverhalten des Quasi-Newton-Verfahrens in Abbildung 3.11 ist wesentlich besser als beim Gradientenverfahren. Das Newton-Verfahren (Methode der Vertrauensbereiche) in Abbildung 3.11 startet zunächst in die "falsche" Richtung, was durch die quadratische Approximation (3.45) am Startpunkt $\mathbf{x}^0 = [-1, 1]^T$ zu erklären ist, dessen Minimum in der Nähe von $\mathbf{x} \approx [-1, 1]^T$ liegt. Allerdings sind die einzelnen Schritte entlang des Tales der Rosenbrock-Funktion deutlich größer, da das Newton-Verfahren die Hessematrix explizit verwendet und nicht auf eine Approximation angewiesen ist wie im Fall des Quasi-Newton-Verfahrens.

Zusätzlich sind in Tabelle 3.6 und Abbildung 3.11 die Ergebnisse für das Simplex-Verfahren von Nelder-Mead angegeben, die mit der MATLAB-Funktion `fminsearch` erzielt wurden. Allerdings bietet die Grafikausgabe unter `fminsearch` nicht die Möglichkeit, die einzelnen Simplexe darzustellen. In der nächsten Aufgabe soll das Simplex-Verfahren deshalb näher untersucht werden, um einen Eindruck von den Simplex-Operationen und der Robustheit des Verfahrens zu erhalten.

Aufgabe 3.6 Schreiben Sie eine MATLAB-Funktion, die das Rosenbrock-Problem (3.50) mit Hilfe des Simplex-Verfahrens nach Nelder-Mead numerisch löst (siehe Abschnitt 3.4.2). Konstruieren Sie den ersten Simplex mit dem Eckpunkten

$$\mathbf{x}^{0,1} = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \quad \mathbf{x}^{0,2} = \mathbf{x}^{0,1} + s \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \mathbf{x}^{0,3} = \mathbf{x}^{0,1} + s \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (3.51)$$

in Abhängigkeit der Seitenlänge $s = 1$. Verwenden Sie für die Abbruchbedingung in Tabelle 3.5 die Schranke $\varepsilon = 10^{-9}$ und vergleichen Sie Ihre Ergebnisse mit den `fminsearch`-Ergebnissen in Tabelle 3.6. Stellen Sie die Simplexe aus den einzelnen Iterationen grafisch dar. Untersuchen Sie die Robustheit und das Konvergenzverhalten des Simplex-Verfahrens für verschiedene Seitenlängen s des Startsimplex und unterschiedliche Startpunkte $\mathbf{x}^{0,1}$.

Aufgabe 3.7 (Zusatzaufgabe) Schreiben Sie eine MATLAB-Funktion, die das Rosenbrock-Problem (3.50) mit Hilfe des Newton-Verfahrens (Linienuche) löst, siehe Abschnitt 3.3.3.3. Verwenden Sie zur Bestimmung der Schrittweite α^k das Backtracking-Verfahren mit dem Startwert $\alpha = 1$ und der Armijo-Bedingung (3.21)

$$f(\mathbf{x}^k + \alpha^k \mathbf{s}^k) \leq f(\mathbf{x}^k) + c_1 \alpha^k \nabla f(\mathbf{x}^k)^\top \mathbf{s}^k \quad (3.52)$$

als Abbruchkriterium für die Schrittweitenbestimmung. Vergleichen Sie die Konvergenzergebnisse mit den Werten in Tabelle 3.6. Stellen Sie die einzelnen Iterationen grafisch dar und variieren Sie die Startpunkte $\mathbf{x}^0$.

Literatur

- [1] S. Boyd und L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [2] C.T. Kelley. *Iterative Methods for Optimization*. Society for Industrial and Applied Mathematics (SIAM), 1999.
- [3] J. Nocedal und S.J. Wright. *Numerical Optimization*. Springer, 2006.
- [4] M. Papageorgiou. *Optimierung*. Oldenbourg Verlag, 1991.
- [5] Mark Schmidt. `minFunc` (MATLAB-Funktion zur Lösung unbeschränkter Optimierungsprobleme). University of British Columbia, Vancouver, Kanada. Erhältlich unter <http://www.cs.ubc.ca/~schmidtm/Software/minFunc.html>.

```

function Xopt = rosenbrock_problem(Xinit,methodQ)
% -----
% Xinit:   Startpunkt
% methodQ: 1 - Liniensuche: Gradientenverfahren
%           2 - Liniensuche: Quasi-Newton
%           3 - Methode der Vertrauensbereiche: Newton-Verfahren
%           4 - Nelder-Mead Simplex-Verfahren
global old
old = [Xinit; rosenbrock(Xinit)];
opt = optimset('Display','iter','PlotFcns',@plot_iterates); % Optionen für Ausgabe

switch methodQ
case 1, % Liniensuche: Gradientenverfahren
    opt = optimset(opt,'LargeScale','off','HessUpdate','steepdesc','GradObj','on');
case 2, % Liniensuche: Quasi-Newton
    opt = optimset(opt,'LargeScale','off','HessUpdate','bfgs','GradObj','on');
case 3, % Methode der Vertrauensbereiche: Newton-Verfahren
    opt = optimset(opt,'LargeScale','on','Hessian','on','GradObj','on');
end

if methodQ<4, Xopt = fminunc(@rosenbrock,Xinit,opt); % numerische Lösung mit
else Xopt = fminsearch(@rosenbrock,Xinit,opt); % fminunc oder finsearch
end

function [f, grad, H] = rosenbrock(x)
% -----
grad = {}; H = {};
f = 100*(x(2)-x(1)^2)^2 + (x(1)-1)^2; % Rosenbrock-Funktion
if nargin>1, % falls Gradient angefordert wird
    grad = [ -400*(x(2)-x(1)^2)*x(1)+2*(x(1)-1);
             200*(x(2)-x(1)^2) ];
end
if nargin>2, % falls Hessematrix angefordert wird
    H = [ -400*(x(2)-3*x(1)^2)+2, -400*x(1);
          -400*x(1), 200 ];
end

function stop = plot_iterates(x,info,state)
% -----
global old
f = rosenbrock(x);
switch state % Grafische Ausgabe:
case 'init', % Initialisierung
    plot_surface(x,f);
case 'iter', % Iterationen
    plot3([old(1),x(1)],[old(2),x(2)],[old(3),f],'b-o','LineWidth',1);
case 'done', % nach letzter Iteration
    plot3(x(1),x(2),f,'go','LineWidth',5);
end
stop = false; % kein Abbruchkriterium
old = [x;f];

function plot_surface(x,f)
% -----
[X1,X2] = meshgrid(-1.5:0.15:1.5); % 3D-Profil von
F = 100*(X2-X1.^2).^2 + (X1-1).^2; % Rosenbrock-Funktion
h = surf(X1,X2,F,'EdgeColor',0.6*[1,1,1],'FaceColor','none');
hold on; axis tight;
plot3(x(1),x(2),f,'ko','LineWidth',5); % Startpunkt
plot3(1,1,0,'ro','LineWidth',5); % optimale Lösung
xlabel('x_1'); ylabel('x_2'); zlabel('f')
set(gcf,'ToolBar','figure'); % figure settings
set(gca,'Xdir','reverse','Ydir','reverse');

```

Abbildung 3.13: MATLAB-Code für das Rosenbrock-Problem (fminunc, fminsearch).

Kapitel 4

Statische Optimierung mit Beschränkungen

In diesem Kapitel werden beschränkte Optimierungsprobleme

$$\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) \quad (4.1a)$$

$$\text{u.B.v. } g_i(\mathbf{x}) = 0, \quad i = 1, \dots, p \quad (4.1b)$$

$$h_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, q \quad (4.1c)$$

mit den Gleichungs- und Ungleichungsbeschränkungen (4.1b), (4.1c) betrachtet. Die notwendigen und hinreichenden Optimalitätsbedingungen des unbeschränkten Falls (Kapitel 3) müssen dafür entsprechend erweitert werden.

4.1 Einführende Beispiele

Zur Veranschaulichung wird Beispiel 1.2 aus Kapitel 1 verwendet und die einzelnen Beschränkungen (1.10)–(1.12) sukzessive betrachtet, um die Erweiterungen gegenüber dem unbeschränkten Fall sowie die Unterschiede zwischen Gleichungs- und Ungleichungsbeschränkungen herzuleiten.

4.1.1 Eine Gleichungsbeschränkung

Beispiel 4.1 *Betrachtet wird das Optimierungsproblem (1.9) mit der Gleichungsbeschränkung (1.10):*

$$\min_{\mathbf{x} \in \mathbb{R}^2} f(\mathbf{x}) = (x_1 - 2)^2 + (x_2 - 1)^2 \quad (4.2a)$$

$$\text{u.B.v. } g(\mathbf{x}) = x_2 - 2x_1 = 0. \quad (4.2b)$$

Abbildung 4.1 stellt die Gerade $g(\mathbf{x}) = 0$ und Höhenlinien von $f(\mathbf{x})$ dar. Da optimale Punkte $\mathbf{x}^*$ auf der Geraden $g(\mathbf{x}) = 0$ liegen müssen, existieren z.B. für die Höhenlinie $f(\mathbf{x}) = 4$ zwei Schnittpunkte. Es ist direkt ersichtlich, dass für Höhenlinien mit $f(\mathbf{x}) < 4$

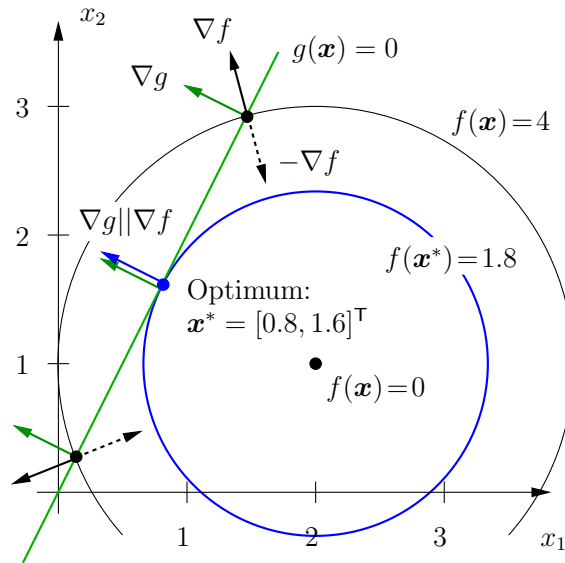


Abbildung 4.1: Veranschaulichung von Beispiel 4.1 mit einer Gleichungsbeschränkung.

die Schnittpunkte dichter zusammen wandern und schließlich zum Minimum

$$\mathbf{x}^* = [0.8, 1.6]^T, \quad f(\mathbf{x}^*) = 1.8 \quad (4.3)$$

führen. Eine wichtige Rolle spielen dabei die Gradienten der Funktionen $f(\mathbf{x})$ und $g(\mathbf{x})$

$$\nabla f(\mathbf{x}) = \begin{bmatrix} 2(x_1 - 2) \\ 2(x_2 - 1) \end{bmatrix}, \quad \nabla g(\mathbf{x}) = \begin{bmatrix} -2 \\ 1 \end{bmatrix}. \quad (4.4)$$

Bei allen nicht-optimalen Punkten auf der Gerade $g(\mathbf{x}) = 0$ ist ∇f nicht kollinear zu ∇g , d.h. es existiert eine Komponente von $-\nabla f$ (maximale Abstiegsrichtung) in Richtung der Geraden $g(\mathbf{x}) = 0$, die zu einer Abnahme der Kostenfunktion f unter Einhaltung der Gleichungsbeschränkung $g(\mathbf{x}) = 0$ führt. Am optimalen Punkt $\mathbf{x}^*$ sind dagegen die Gradientenvektoren

$$\nabla f(\mathbf{x}^*) = \begin{bmatrix} -2.4 \\ 1.2 \end{bmatrix}, \quad \nabla g(\mathbf{x}^*) = \begin{bmatrix} -2 \\ 1 \end{bmatrix} \quad (4.5)$$

kollinear, d.h., es gilt $\nabla f(\mathbf{x}^*) \parallel \nabla g(\mathbf{x}^*)$ bzw.

$$0 = \nabla f(\mathbf{x}^*) + \lambda^* \nabla g(\mathbf{x}^*) \quad \text{mit} \quad \lambda^* = -1.2. \quad (4.6)$$

Das Ergebnis des Beispiels kann mit Hilfe der *Lagrange-Funktion*

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \lambda g(\mathbf{x}) \quad (4.7)$$

verallgemeinert werden. Dabei wird die Gleichungsbeschränkung mit Hilfe eines *Lagrange-Multiplikators* λ zu der Kostenfunktion $f(\mathbf{x})$ addiert. Die Kollinearität der Gradienten ∇f und ∇g an einem optimalen Punkt $\mathbf{x}^*$ ergibt sich dann aus der *Stationaritätsbedingung*

$$\nabla_{\mathbf{x}} L(\mathbf{x}^*, \lambda^*) = \nabla f(\mathbf{x}^*) + \lambda^* \nabla g(\mathbf{x}^*) = 0. \quad (4.8)$$

Zusammen mit der Gleichungsbeschränkung $g(\mathbf{x}^*) = 0$ erhält man ein Gleichungssystem der Ordnung $n + 1$ für die $n + 1$ Unbekannten $\mathbf{x}^* \in \mathbb{R}^n$ und $\lambda^* \in \mathbb{R}$. Allerdings sei angemerkt, dass es sich bei (4.8) lediglich um eine *notwendige* und keine *hinreichende* Bedingung für Optimalität handelt.

Im Falle einer aktiven Beschränkung kann $h(\mathbf{x}) = 0$ als Gleichungsbeschränkung betrachtet werden, d.h. die Gradienten ∇f und ∇h müssen im optimalen Punkt $\mathbf{x}^*$ kollinear sein. Dies kann mit Hilfe der Lagrange-Funktion (4.10) durch

$$\nabla_{\mathbf{x}} L(\mathbf{x}^*, \mu^*) = \nabla f(\mathbf{x}^*) + \mu^* \nabla h(\mathbf{x}^*) = 0 \quad (4.12)$$

ausgedrückt werden. Allerdings spielt an dieser Stelle das Vorzeichen des Lagrange-Multiplikators μ^* eine wichtige Rolle. Um dies zu untersuchen, wird die Funktion $h(\mathbf{x})$ in eine Taylorreihe 1. Ordnung entwickelt:

$$h(\mathbf{x} + \mathbf{s}) = h(\mathbf{x}) + \nabla h(\mathbf{x})^\top \mathbf{s} + \mathcal{O}(\|\mathbf{s}\|^2).$$

Im Falle einer aktiven Beschränkung $h(\mathbf{x}) = 0$ muss eine (hinreichend kleine) zulässige Richtung $\mathbf{s}$ die Ungleichung $h(\mathbf{x} + \mathbf{s}) \leq 0$ erfüllen, d.h.

$$\nabla h(\mathbf{x})^\top \mathbf{s} \leq 0. \quad (4.13)$$

Analog dazu wird die Kostenfunktion $f(\mathbf{x})$ in eine Taylorreihe entwickelt

$$f(\mathbf{x} + \mathbf{s}) = f(\mathbf{x}) + \nabla f(\mathbf{x})^\top \mathbf{s} + \mathcal{O}(\|\mathbf{s}\|^2).$$

Wenn der Punkt $\mathbf{x}$ kein Minimum $\mathbf{x}^*$ darstellt, muss es eine Abstiegsrichtung $\mathbf{s}$ mit $f(\mathbf{x} + \mathbf{s}) < f(\mathbf{x})$ geben, d.h.

$$\nabla f(\mathbf{x})^\top \mathbf{s} < 0. \quad (4.14)$$

Der Umkehrschluss bedeutet, dass falls $\mathbf{x}^*$ ein Minimum am Rand $h(\mathbf{x}^*) = 0$ darstellt, darf es keine Richtung $\mathbf{s}$ geben, die *beide* Bedingungen (4.13), (4.14) erfüllt. Dieser Umstand ist in Abbildung 4.3 für einen nicht-optimalen Punkt $\mathbf{x}$ veranschaulicht.

Aus Abbildung 4.3 ist ersichtlich, dass es nur dann keine zulässige Richtung $\mathbf{s}$ gibt, die beide Ungleichungen (4.13), (4.14) erfüllt, wenn $-\nabla f$ und ∇h in die gleiche Richtung weisen, d.h.

$$-\nabla f(\mathbf{x}^*) = \mu^* \nabla h(\mathbf{x}^*) \quad \text{mit} \quad \mu^* \geq 0. \quad (4.15)$$

Mit anderen Worten: falls (4.15) gilt, existiert keine zulässige Richtung $\mathbf{s}$, die zu einer Verringerung der Kostenfunktion $f(\mathbf{x}^* + \mathbf{s})$ führt, ohne die aktive Beschränkung $h(\mathbf{x}^*) = 0$ zu verletzen.

Das Vorzeichen von μ^* ist wichtig in diesem Zusammenhang und kann anhand von Abbildung 4.3 erklärt werden. Falls $\mu^* < 0$ wäre, würden ∇f und ∇h in die gleiche Richtung weisen, und eine ganze Halbebene an zulässigen Richtungen $\mathbf{s}$ würde entstehen, die zu einer Abnahme von f führen würde, während die Beschränkung $h(\mathbf{x}) \leq 0$ eingehalten wäre.

Die Bedingungen für ein Minimum an der Stelle $\mathbf{x}^*$ lassen sich wie folgt zusammenfassen:

$$\exists \mu^* \geq 0 : \quad \nabla_{\mathbf{x}} L(\mathbf{x}^*, \mu^*) = 0, \quad \mu^* h(\mathbf{x}^*) = 0 \quad (4.16)$$

Die Gleichung $\mu^* h(\mathbf{x}^*) = 0$ drückt in einer kompakten Schreibweise die Bedingung $\mu^* = 0$ im Falle einer inaktiven Beschränkung $h(\mathbf{x}^*) < 0$ aus und wird als *Komplementaritätsbedingung* bezeichnet.

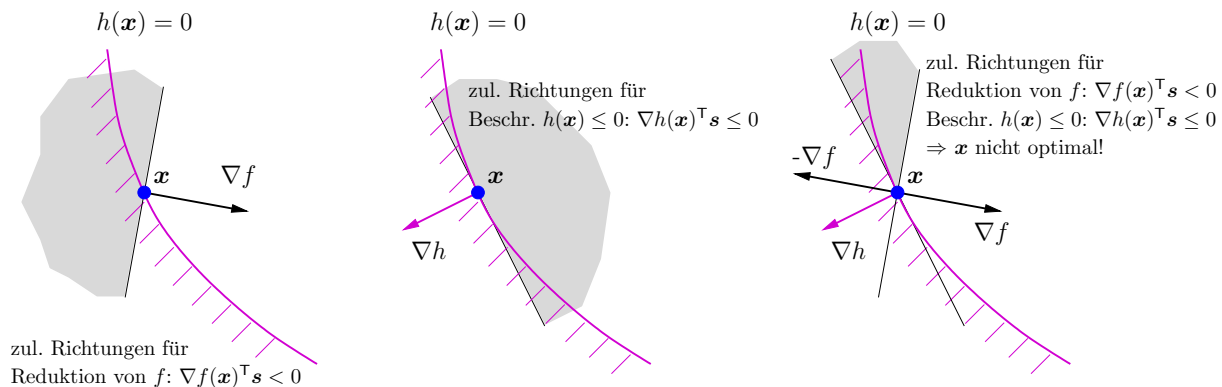


Abbildung 4.3: Geometrische Interpretation der Bedingungen (4.13) und (4.14) für eine aktive Ungleichungsbeschränkung im $\mathbb{R}^2$.

4.1.3 Zwei Ungleichungsbeschränkungen

Im Folgenden sollen zwei Ungleichungsbeschränkungen betrachtet werden:

$$\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) \quad (4.17a)$$

$$\text{u.B.v. } h_1(\mathbf{x}) \leq 0 \quad (4.17b)$$

$$h_2(\mathbf{x}) \leq 0. \quad (4.17c)$$

Analog zum vorhergehenden Fall mit einer Beschränkung muss an Punkten $\mathbf{x}$ an denen beide Beschränkungen aktiv sind, d.h. $h_1(\mathbf{x}) = 0$ und $h_2(\mathbf{x}) = 0$ gilt, eine zulässige Richtung $\mathbf{s}$ die Bedingung (4.13)

$$\nabla h_1(\mathbf{x})^\top \mathbf{s} \leq 0, \quad \nabla h_2(\mathbf{x})^\top \mathbf{s} \leq 0 \quad (4.18)$$

erfüllen. Die Bedingungen (4.18) lassen sich für $\mathbf{x} \in \mathbb{R}^2$ anschaulich erklären, da in diesem Fall zulässige Richtungen $\mathbf{s}$ in dem Kegel liegen müssen, der durch die Tangenten an die Kurven $h_1(\mathbf{x}) = 0$ und $h_2(\mathbf{x}) = 0$ aufgespannt wird, siehe Abbildung 4.4.

Falls der Punkt $\mathbf{x}$ kein Minimum darstellt, muss es Richtungen $\mathbf{s}$ geben, die die Abstiegsbedingung (4.14)

$$\nabla f(\mathbf{x})^\top \mathbf{s} < 0 \quad (4.19)$$

erfüllen. Im Umkehrschluss darf an einem Minimum $\mathbf{x}^*$ keine Richtung $\mathbf{s}$ existieren, die (4.19) und (4.18) erfüllt. Dies ist genau dann der Fall, wenn der negative Gradient $-\nabla f(\mathbf{x}^*)$ in den Kegel, der durch die Gradienten $\nabla h_1(\mathbf{x}^*)$ und $\nabla h_2(\mathbf{x}^*)$ aufgespannt wird. Abbildung 4.4c) veranschaulicht diesen Zusammenhang für $\mathbf{x} \in \mathbb{R}^2$.

Somit muss der negative Gradient $-\nabla f(\mathbf{x}^*)$ an einem Minimum $\mathbf{x}^*$ mit zwei aktiven Ungleichungsbeschränkungen eine *positive* Linearkombination der Gradienten der Beschränkungsfunktionen sein:

$$\exists \mu_1^*, \mu_2^* \geq 0 : \quad -\nabla f(\mathbf{x}^*) = \mu_1^* \nabla h_1(\mathbf{x}^*) + \mu_2^* \nabla h_2(\mathbf{x}^*). \quad (4.20)$$

Analog zu den vorherigen Abschnitten können die notwendigen Bedingungen für ein lokales Minimum $\mathbf{x}^*$ wieder mit der Lagrange-Funktion

$$L(\mathbf{x}, \boldsymbol{\mu}) = f(\mathbf{x}) + \mu_1 h_1(\mathbf{x}) + \mu_2 h_2(\mathbf{x}), \quad \boldsymbol{\mu} = [\mu_1, \mu_2]^\top \quad (4.21)$$

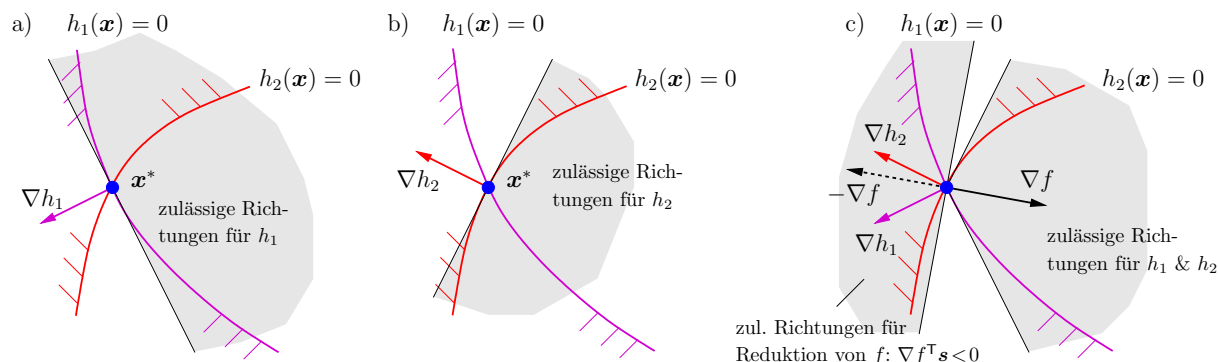


Abbildung 4.4: Geometrische Interpretation der Bedingungen (4.18) und (4.19) für zwei aktive Ungleichungsbeschränkungen im $\mathbb{R}^2$.

angegeben werden:

$$\exists \mu_1^*, \mu_2^* \geq 0 : \quad \nabla_{\mathbf{x}} L(\mathbf{x}^*, \boldsymbol{\mu}^*) = 0, \quad \mu_1^* h_1(\mathbf{x}^*) = 0, \quad \mu_2^* h_2(\mathbf{x}^*) = 0. \quad (4.22)$$

Die beiden Komplementaritätsbedingungen in (4.22) sind erfüllt, falls entweder die entsprechende Beschränkung $h_i(\mathbf{x}^*) \leq 0$ aktiv oder der Lagrange-Multiplikator $\mu_i^* = 0$ ist.

Aufgabe 4.1 Als Erweiterung von Beispiel 4.2 wird die zweite Ungleichungsbeschränkung (1.12) betrachtet

$$\min_{\mathbf{x} \in \mathbb{R}^2} \quad f(\mathbf{x}) = (x_1 - 2)^2 + (x_2 - 1)^2 \quad (4.23a)$$

$$\text{u.B.v.} \quad h_1(\mathbf{x}) = x_1 + x_2 - 2 \leq 0 \quad (4.23b)$$

$$h_2(\mathbf{x}) = x_1^2 - x_2 \leq 0. \quad (4.23c)$$

Berechnen Sie für den optimalen Punkt $\mathbf{x}^* = [1, 1]^T$ die Lagrange-Multiplikatoren μ_1^* , μ_2^* und überprüfen Sie die Vorzeichenbedingung (4.20) grafisch.

4.2 Optimalitätsbedingungen

Die Ergebnisse aus dem vorherigen Abschnitt lassen sich auf den allgemeinen Fall

$$\min_{\mathbf{x} \in \mathbb{R}^n} \quad f(\mathbf{x}) \quad (4.24a)$$

$$\text{u.B.v.} \quad g_i(\mathbf{x}) = 0, \quad i = 1, \dots, p \quad (4.24b)$$

$$h_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, q \quad (4.24c)$$

mit p Gleichungs- und q Ungleichungsbeschränkungen übertragen.

4.2.1 Beschränkungsqualifikation

Aus dem betrachteten Beispiel mit der Gleichungsbeschränkung $g(\mathbf{x}) = 0$ (Abschnitt 4.1.1) ist ersichtlich, dass die $n + 1$ Gleichungen

$$g(\mathbf{x}) = 0, \quad \nabla L(\mathbf{x}, \lambda) = \nabla f(\mathbf{x}) + \lambda \nabla g(\mathbf{x}) = 0$$

zusammen betrachtet werden müssen, um den Punkt $\mathbf{x} \in \mathbb{R}^n$ und $\lambda \in \mathbb{R}$ bestimmen zu können. Das Gleiche gilt auch im Falle einer aktiven Ungleichungsbeschränkung $h(\mathbf{x}) = 0$, um $\mathbf{x} \in \mathbb{R}^n$ und $\mu \in \mathbb{R}$ aus den Gleichungen

$$h(\mathbf{x}) = 0, \quad \nabla L(\mathbf{x}, \lambda, \mu) = \nabla f(\mathbf{x}) + \mu \nabla h(\mathbf{x}) = 0$$

zu erhalten. Für das allgemeine Optimierungsproblem (4.24) mit p Gleichungs- und q Ungleichungsbeschränkungen muss deshalb eine zusätzliche Annahme für aktive Beschränkungen getroffen werden. Dafür wird die folgende Definition verwendet:

Definition 4.1 (Menge aktiver Ungleichungsbeschränkungen) Die Menge aktiver Ungleichungsbeschränkungen des Optimierungsproblems (4.24) an einem Punkt $\mathbf{x}$ ist definiert durch

$$\mathcal{A}(\mathbf{x}) = \{i \in \{1, \dots, q\} \mid h_i(\mathbf{x}) = 0\}. \quad (4.25)$$

Die sogenannte *Beschränkungsqualifikation* (Englisch: *constraint qualification*) fordert, dass die Gradienten der aktiven Ungleichungsbeschränkungen und der Gleichungsbeschränkungen linear unabhängig sind

$$\text{Rang} \begin{bmatrix} \nabla \mathbf{g}(\mathbf{x}) \\ \nabla \mathbf{h}_{\mathcal{A}(\mathbf{x})}(\mathbf{x}) \end{bmatrix} = p + \dim\{\mathcal{A}(\mathbf{x})\}. \quad (4.26)$$

Die vektorielle Gradientennotation $\nabla \mathbf{g}(\mathbf{x})$ bedeutet $\nabla \mathbf{g}(\mathbf{x}) = [\nabla g_1(\mathbf{x}), \dots, \nabla g_p(\mathbf{x})]^\top$. Analog dazu ist der zweite Term $\nabla \mathbf{h}_{\mathcal{A}(\mathbf{x})}(\mathbf{x})$ zu verstehen, wobei die Vektorfunktion $\mathbf{h}_{\mathcal{A}(\mathbf{x})}$ die im Punkt $\mathbf{x}$ aktiven Ungleichungsbeschränkungen darstellt:

$$\mathbf{h}_{\mathcal{A}(\mathbf{x})}(\mathbf{x}) = [h_i(\mathbf{x})]_{i \in \mathcal{A}(\mathbf{x})} \quad (4.27)$$

Die Rangbedingung (4.26) besagt, dass maximal $n - p$ Ungleichungsbeschränkungen am gleichen Punkt $\mathbf{x}$ aktiv sein dürfen. Wäre dies nicht der Fall, würde das Gleichungssystem bestehend aus $\mathbf{g}(\mathbf{x}) = 0$ und $\mathbf{h}_{\mathcal{A}(\mathbf{x})}(\mathbf{x}) = 0$ mehr als n Gleichungen umfassen und wäre somit überbestimmt.

Die Beschränkungsqualifikation (4.26) betrifft die lineare Unabhängigkeit der aktiven Beschränkungen und ist die gebräuchlichste Bedingung dieser Art in der Optimierung. Allerdings sei angemerkt, dass (4.26) nicht notwendig, sondern nur *hinreichend* ist und in manchen Fällen zu konservativ sein kann. Weitere Beschränkungsqualifikationen können in der Literatur gefunden werden.

4.2.2 Optimalitätsbedingungen 1. Ordnung

Definition 4.2 (Lagrange-Funktion) Die Lagrange-Funktion für das Optimierungsproblem (4.24) ist

$$L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\mathbf{x}) + \sum_{i=1}^p \lambda_i g_i(\mathbf{x}) + \sum_{i=1}^q \mu_i h_i(\mathbf{x}) \quad (4.28)$$

mit den Lagrange-Multiplikatoren $\boldsymbol{\lambda} = [\lambda_1, \dots, \lambda_p]^\top$ und $\boldsymbol{\mu} = [\mu_1, \dots, \mu_q]^\top$.

Mit Hilfe der Lagrange-Funktion $L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu})$ können die notwendigen Optimalitätsbedingungen erster Ordnung für das Optimierungsproblem (4.24) angegeben werden:

Satz 4.1 (Notwendige Optimalitätsbedingungen 1. Ordnung)

Sei $\mathbf{x}^*$ ein lokales Minimum des Optimierungsproblems (4.24) und sei die Beschränkungsqualifikation (4.26) am Punkt $\mathbf{x}^*$ erfüllt. Falls die Funktionen f , g_i und h_i stetig differenzierbar sind, existieren Lagrange-Multiplikatoren $\boldsymbol{\lambda}^* = [\lambda_1^*, \dots, \lambda_p^*]^\top$ und $\boldsymbol{\mu}^* = [\mu_1^*, \dots, \mu_q^*]^\top$, die die folgenden Bedingungen erfüllen:

$$\nabla_{\mathbf{x}} L(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = 0 \quad (4.29a)$$

$$g_i(\mathbf{x}^*) = 0, \quad i = 1, \dots, p \quad (4.29b)$$

$$h_i(\mathbf{x}^*) \leq 0, \quad i = 1, \dots, q \quad (4.29c)$$

$$\mu_i^* \geq 0, \quad i = 1, \dots, q \quad (4.29d)$$

$$\mu_i^* h_i(\mathbf{x}^*) = 0, \quad i = 1, \dots, q. \quad (4.29e)$$

Die Bedingungen (4.29) werden als *Karush-Kuhn-Tucker-Bedingungen* bzw. KKT-Bedingungen bezeichnet. Die *Komplementaritätsbedingung* impliziert, dass die i -te Ungleichungsbeschränkung entweder aktiv ist, d.h. $h_i(\mathbf{x}^*) = 0$ oder $\mu_i^* = 0$ gilt.

Die Optimalitätsbedingungen in Satz (4.1) sind nur gültig, wenn die Beschränkungsqualifikation (4.26) am Minimum $\mathbf{x}^*$ erfüllt ist. Falls dies nicht der Fall ist, kann es sein, dass die KKT-Bedingungen (4.29) nicht anwendbar sind, wie im nächsten Beispiel gezeigt wird.

Beispiel 4.3 Betrachtet wird das Problem

$$\min_{\mathbf{x} \in \mathbb{R}^2} \quad -x_1 \quad (4.30a)$$

$$\text{u.B.v.} \quad h_1(\mathbf{x}) = x_1^3 - x_2 \leq 0 \quad (4.30b)$$

$$h_2(\mathbf{x}) = x_1^3 + x_2 \leq 0. \quad (4.30c)$$

Abbildung 4.5 zeigt die zulässige Menge $\mathcal{X}_{\text{ad}}$ und den optimalen Punkt $\mathbf{x}^* = [0, 0]^\top$, an dem beide Beschränkungen aktiv sind. Die Gradienten der Kostenfunktion f und der Beschränkungsfunktionen an diesem Punkt $\mathbf{x}^*$ lauten

$$\nabla f(\mathbf{x}^*) = \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \quad \nabla h_1(\mathbf{x}^*) = \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \quad \nabla h_2(\mathbf{x}^*) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \quad (4.31)$$

womit sich die KKT-Bedingung (4.29a)

$$\nabla L_{\mathbf{x}}(\mathbf{x}^*, \boldsymbol{\mu}^*) = \begin{pmatrix} -1 \\ 0 \end{pmatrix} + \mu_1^* \begin{pmatrix} 0 \\ -1 \end{pmatrix} + \mu_2^* \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \mathbf{0} \quad (4.32)$$

ergibt. Allerdings besitzt diese Gleichung keine Lösung, da die Beschränkungsqualifikation (4.26) am optimalen Punkt $\mathbf{x}^*$ verletzt ist.

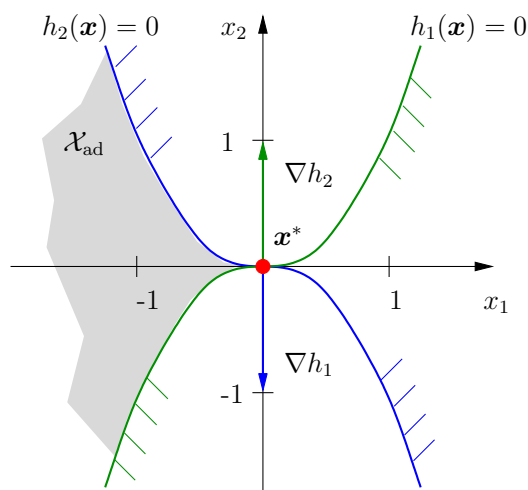


Abbildung 4.5: Veranschaulichung von Problem (4.30) in Beispiel 4.3.

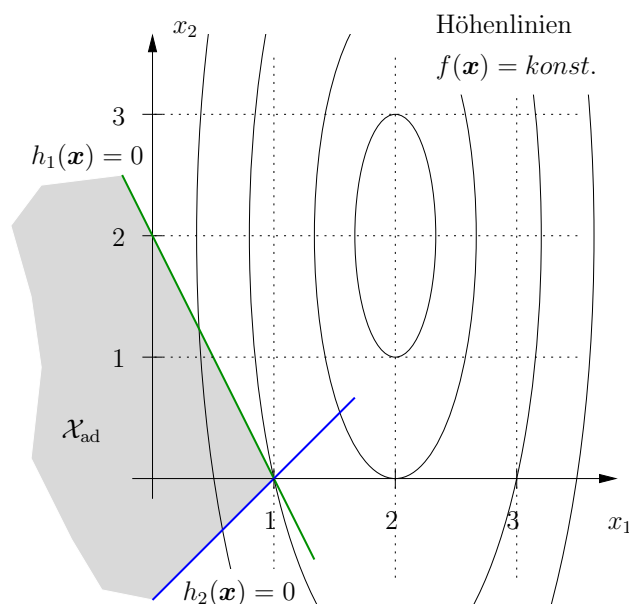


Abbildung 4.6: Höhenlinien und zulässige Menge in Aufgabe 4.2.

Aufgabe 4.2 Gegeben sei das Optimierungsproblem

$$\min_{\mathbf{x} \in \mathbb{R}^n} \quad f(\mathbf{x}) = (x_1 - 2)^2 + \left(\frac{x_2 - 2}{3} \right)^2 \quad (4.33a)$$

$$\text{u.B.v.} \quad h_1(\mathbf{x}) = 2x_1 + x_2 - 2 \leq 0 \quad (4.33b)$$

$$h_2(\mathbf{x}) = x_1 - x_2 - 1 \leq 0. \quad (4.33c)$$

Bestimmen Sie die optimale Lösung $\mathbf{x}^*$, indem Sie die KKT-Bedingungen (4.29) für die unterschiedlichen Fälle an aktiven/inaktiven Beschränkungen untersuchen und in Abbildung 4.6 eintragen.

Aufgabe 4.3 Das folgende Optimierungsproblem ist in Abhängigkeit des Parameters s gegeben:

$$\min_{\mathbf{x} \in \mathbb{R}^n} \quad \left(x_1 - \frac{3}{2} \right)^2 + (x_2 - s)^4 \quad \text{u.B.v.} \quad \begin{bmatrix} 1 - x_1 - x_2 \\ 1 - x_1 + x_2 \\ 1 + x_1 - x_2 \\ 1 + x_1 + x_2 \end{bmatrix} \geq 0. \quad (4.34a)$$

Für welche Werte von s erfüllt der Punkt $\mathbf{x}^* = [1, 0]^\top$ die KKT-Bedingungen (4.29)?

4.2.3 Optimalitätsbedingungen 2. Ordnung

Satz 4.2 (Notwendige Optimalitätsbedingungen 2. Ordnung)

Sei $\mathbf{x}^*$ ein lokales Minimum des Optimierungsproblems (4.24) und sei die Beschränkungsqualifikation (4.26) am Punkt $\mathbf{x}^*$ erfüllt, so dass die KKT-Bedingungen (4.29) mit zugehörigen Lagrange-Multiplikatoren $\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*$ gelten. Falls f , g_i und h_i zweimal stetig differenzierbar sind, dann gilt zusätzlich

$$\mathbf{s}^\top \nabla_{\mathbf{x}\mathbf{x}}^2 L(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \mathbf{s} \geq 0 \quad \forall \mathbf{s} \in \mathcal{C}(\mathbf{x}^*, \boldsymbol{\mu}^*) \quad (4.35)$$

mit

$$\begin{aligned} \mathcal{C}(\mathbf{x}^*, \boldsymbol{\mu}^*) = \{ \mathbf{s} \in \mathbb{R}^n : & \nabla g_i(\mathbf{x}^*)^\top \mathbf{s} = 0, \quad i = 1, \dots, p \\ & \nabla h_i(\mathbf{x}^*)^\top \mathbf{s} = 0, \quad i \in \mathcal{A}(\mathbf{x}^*), \mu_i^* > 0 \\ & \nabla h_i(\mathbf{x}^*)^\top \mathbf{s} \leq 0, \quad i \in \mathcal{A}(\mathbf{x}^*), \mu_i^* = 0 \}. \end{aligned} \quad (4.36)$$

Die Bedingung (4.35) hat Ähnlichkeit mit dem unbeschränkten Fall (Satz 3.1), bei dem die Hessematrix $\nabla^2 f(\mathbf{x}^*)$ am Punkt $\mathbf{x}^*$ positiv semi-definit sein musste, d.h. $\mathbf{s}^\top \nabla^2 f(\mathbf{x}^*) \mathbf{s} \geq 0$ für alle $\mathbf{s} \in \mathbb{R}^n$. Im hier betrachteten beschränkten Fall beinhaltet die Menge $\mathcal{C}(\mathbf{x}^*, \boldsymbol{\mu}^*)$ nur zulässige Richtungen $\mathbf{s}$, für die die Beschränkungen (4.24b), (4.24c) eingehalten werden. Dabei werden nur Richtungen $\mathbf{s}$ betrachtet, bei denen aus den Informationen der ersten Ableitungen nicht hervorgeht, ob sie zu einer Abnahme oder Zunahme der Kostenfunktion $f(\mathbf{x}^* + \mathbf{s})$ führen.¹

Satz 4.3 (Hinreichende Optimalitätsbedingungen)

Sei $\mathbf{x}^*$ ein zulässiger Punkt des Optimierungsproblems (4.24) mit zugehörigen Lagrange-Multiplikatoren $\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*$, so dass die KKT-Bedingungen (4.29) erfüllt sind. Falls f , g_i und h_i zweimal stetig differenzierbar sind und

$$\mathbf{s}^\top \nabla_{xx}^2 L(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \mathbf{s} > 0 \quad \forall \mathbf{s} \in \mathcal{C}(\mathbf{x}^*, \boldsymbol{\mu}^*), \mathbf{s} \neq \mathbf{0} \quad (4.37)$$

mit $\mathcal{C}(\mathbf{x}^*, \boldsymbol{\mu}^*)$ in (4.36) gilt, dann ist $\mathbf{x}^*$ ein striktes lokales Minimum.

Falls die Hessematrix $\nabla_{xx}^2 L$ positiv definit ist, d.h.

$$\mathbf{s}^\top \nabla_{xx}^2 L(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \mathbf{s} > 0 \quad \forall \mathbf{s} \in \mathbb{R}^n, \mathbf{s} \neq \mathbf{0} \quad (4.38)$$

gilt, dann ist die hinreichende Bedingung (4.37) stets erfüllt. Dies ist vor allem bei linearen und quadratischen Problemen der Fall.

Im Allgemeinen kann bei konvexen Problemen auf die Auswertung der Optimalitätsbedingungen 2. Ordnung verzichtet werden, da in diesem Fall die Erfüllung der KKT-Bedingungen (4.29) notwendig und hinreichend ist, wie im folgenden Satz gezeigt wird:

Satz 4.4 (Hinreichende Optimalitätsbedingungen für konvexe Probleme)

Seien $f : \mathbb{R}^n \rightarrow \mathbb{R}$ und $h_i : \mathbb{R}^n \rightarrow \mathbb{R}$ konvexe und stetig differenzierbare Funktionen und sei $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$ stetig differenzierbar und linear. Falls ein Punkt $\mathbf{x}^*$ mit zugehörigen Lagrange-Multiplikatoren $\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*$ die KKT-Bedingungen (4.29) erfüllt, dann ist $\mathbf{x}^*$ ein globales Minimum des Optimierungsproblems (4.24).

Aufgabe 4.4 (Fortsetzung von Aufgabe 4.2) Überprüfen Sie die hinreichenden Optimalitätsbedingungen nach Satz 4.3 und Satz 4.4 für das Problem (4.33) aus Aufgabe 4.2.

¹Für aktive Ungleichungsbeschränkungen mit $\mu_i^* > 0$ führen alle zulässigen Richtungen mit $\nabla h_i(\mathbf{x}^*)^\top \mathbf{s} < 0$ zu einer Zunahme der Kostenfunktion $f(\mathbf{x}^* + \mathbf{s})$, siehe Abschnitt 4.2.4.

4.2.4 Interpretation der Lagrange–Multiplikatoren

Die Bedeutung der Lagrange–Multiplikatoren ist durch die einführenden Beispiele in Abschnitt 4.1 illustriert worden. Neben der mathematischen Formulierung existiert auch eine anschauliche Interpretation, die im Folgenden für den Fall einer Gleichungsbeschränkung

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^n} \quad & f(\mathbf{x}) \\ \text{u.B.v.} \quad & g(\mathbf{x}) = 0 \end{aligned}$$

untersucht werden soll. Für eine optimale Lösung $\mathbf{x}^*$ dieses Problems muss gelten

$$\nabla L(\mathbf{x}^*, \lambda^*) = \nabla f(\mathbf{x}^*) + \lambda^* \nabla g(\mathbf{x}^*) = \mathbf{0}. \quad (4.39)$$

Um die Beziehung zwischen den Lagrange–Multiplikatoren und der optimalen Lösung $\mathbf{x}^*$ zu untersuchen, wird eine kleine Modifikation der Beschränkungen

$$g(\mathbf{x}) = \varepsilon \quad \Rightarrow \quad \bar{g}(\mathbf{x}(\varepsilon), \varepsilon) = g(\mathbf{x}(\varepsilon)) - \varepsilon = 0 \quad (4.40)$$

betrachtet. Die Variablen $\mathbf{x}(\varepsilon)$ hängen nun von ε ab, wobei sich für $\varepsilon = 0$ die ursprüngliche Lösung $\mathbf{x}^* = \mathbf{x}(0)$ des obigen Problems ergibt. Da $\bar{g}(\mathbf{x}(\varepsilon), \varepsilon) = 0$ für alle ε gelten muss, folgt

$$\frac{\partial \bar{g}}{\partial \varepsilon} + \nabla_{\mathbf{x}} \bar{g}(\mathbf{x}(\varepsilon), \varepsilon)^\top \frac{d\mathbf{x}}{d\varepsilon} = 0 \quad \Rightarrow \quad \nabla_{\mathbf{x}} \bar{g}(\mathbf{x}(\varepsilon), \varepsilon)^\top \frac{d\mathbf{x}}{d\varepsilon} = 1 \quad (4.41)$$

mit $\partial \bar{g} / \partial \varepsilon = -1$. Von besonderem Interesse ist die Auswirkung von ε auf den minimalen Kostenwert $f(\mathbf{x}^*)$, die mit Hilfe der Bezeichnung $\bar{f}(\varepsilon) = f(\mathbf{x}(\varepsilon))$ wie folgt ausgedrückt werden kann:

$$\left. \frac{d\bar{f}}{d\varepsilon} \right|_{\varepsilon=0} = \nabla f(\mathbf{x}^*)^\top \left. \frac{d\mathbf{x}}{d\varepsilon} \right|_{\varepsilon=0} \stackrel{(4.39)}{=} -\lambda^* \nabla g(\mathbf{x}^*)^\top \left. \frac{d\mathbf{x}}{d\varepsilon} \right|_{\varepsilon=0} \stackrel{(4.41)}{=} -\lambda^*. \quad (4.42)$$

Der Lagrange–Multiplikator λ^* ist also ein *direktes Maß der Veränderung (Sensitivität)* der Kostenfunktion am Optimum bezogen auf eine Veränderung der Gleichungsbeschränkung $g(\mathbf{x}^*) = 0$.

Diese Betrachtung lässt sich analog für eine aktive Ungleichungsbeschränkung durchführen, d.h.

$$h(\mathbf{x}) \leq \varepsilon \quad \Rightarrow \quad \left. \frac{d\bar{f}}{d\varepsilon} \right|_{\varepsilon=0} = -\mu^*, \quad (4.43)$$

wobei μ^* den entsprechenden (optimalen) Lagrange–Multiplikator für den optimalen Punkt $\mathbf{x}^*$ mit $h(\mathbf{x}^*) = 0$ darstellt. Die Interpretation ist dabei die gleiche wie im Fall der Gleichungsbeschränkung: der Lagrange–Multiplikator μ^* einer aktiven Ungleichungsbeschränkung $h(\mathbf{x}) \leq 0$ am Optimum $\mathbf{x}^*$ spiegelt die Sensitivität der Kostenfunktion $f(\mathbf{x}^*)$ bzgl. dieser Beschränkung wider.

Mit anderen Worten: ein großer Wert von λ^* oder μ^* weist darauf hin, dass die “Lockerung” einer sehr restriktiven Beschränkung zu einer deutlichen Reduktion der Kostenfunktion f führen kann.

Die Untersuchungen sind direkt auf den allgemeinen Fall (4.24) mit p Gleichungs– und q Ungleichungsbeschränkungen erweiterbar. Die Gleichung (4.43) verdeutlicht auch die

Vorzeichenbedingung (4.29d) der KKT-Bedingungen. Falls $\mu^* < 0$ wäre, würde dies bedeuten, dass

$$\left. \frac{d\bar{f}}{d\varepsilon} \right|_{\varepsilon=0} > 0$$

wäre und somit eine Reduktion der Kostenfunktion $f(\mathbf{x}^*)$ durch Verlassen der aktiven Ungleichungsbeschränkung (für $\varepsilon < 0$) möglich wäre. Da in diesem Fall $\mathbf{x}^*$ nicht die optimale Lösung wäre, muss folglich $\mu^* \geq 0$ gelten.

Aufgabe 4.5 (Fortsetzung von Aufgabe 4.2) Interpretieren Sie den Einfluß der Lagrange-Multiplikatoren grafisch für das Problem (4.33) aus Aufgabe 4.2.

4.3 Lineare Optimierung

Die lineare Optimierung (auch *lineare Programmierung* genannt) ist ein Sonderfall der nichtlinearen Optimierung, bei der die Kostenfunktion und die Beschränkungen linear sind. Dieser Spezialfall stellt die für praktische Anwendungen am weitesten verbreitete Optimierungsaufgabe dar. Ihre vielfältige Anwendung z.B. bei wirtschaftlichen Fragestellungen oder Produktions- und Transportproblemen (siehe Abschnitt 4.3.5) verdankt die lineare Programmierung ihrer Einfachheit als auch dem Vorhandensein zuverlässiger numerischer Lösungsverfahren.

4.3.1 Standardform-Darstellung

Die Standardform eines linearen Optimierungsproblems ist

$$\min_{\mathbf{x} \in \mathbb{R}^n} \quad \mathbf{c}^\top \mathbf{x} \quad (4.44a)$$

$$\text{u.B.v.} \quad \mathbf{Ax} = \mathbf{b}, \quad \mathbf{A} \in \mathbb{R}^{p \times n}, \quad \mathbf{b} \in \mathbb{R}^p \quad (4.44b)$$

$$\mathbf{x} \geq \mathbf{0}, \quad (4.44c)$$

die die Basis für die meisten numerischen Verfahren der linearen Optimierung darstellt. Jedes lineare Optimierungsproblem lässt sich anhand von einfachen Methoden in diese Standardform bringen. Ist z.B. das Problem

$$\min_{\mathbf{x} \in \mathbb{R}^n} \quad \mathbf{c}^\top \mathbf{x} \quad (4.45a)$$

$$\text{u.B.v.} \quad \mathbf{Ax} \leq \mathbf{b}, \quad \mathbf{A} \in \mathbb{R}^{q \times n}, \quad \mathbf{b} \in \mathbb{R}^q \quad (4.45b)$$

gegeben, so können die Ungleichungsbeschränkungen (4.45b) mit Hilfe von *Schlupfvariablen* $\mathbf{z}$ in Gleichungsbeschränkungen und zusätzliche Ungleichungen transformiert werden

$$\mathbf{Ax} \leq \mathbf{b} \quad \Rightarrow \quad \mathbf{Ax} + \mathbf{z} = \mathbf{b}, \quad \mathbf{z} \geq \mathbf{0}. \quad (4.46)$$

Da die Variablen $\mathbf{x}$ bisher nicht notwendigerweise die Ungleichung (4.44c) erfüllen, wird $\mathbf{x}$ in nicht-negative und nicht-positive Anteile aufgespalten:

$$\mathbf{x} = \mathbf{x}^+ - \mathbf{x}^-, \quad \mathbf{x}^-, \mathbf{x}^+ \geq \mathbf{0}. \quad (4.47)$$

Somit kann das Problem (4.45) in die Standardform (4.44)

$$\min_{\bar{\mathbf{x}} \in \mathbb{R}^{2n+q}} [\mathbf{c}^\top, -\mathbf{c}^\top, \mathbf{0}^\top] \bar{\mathbf{x}} \quad (4.48a)$$

$$\text{u.B.v.} \quad [\mathbf{A}, -\mathbf{A}, \mathbf{I}] \bar{\mathbf{x}} = \mathbf{b}, \quad \bar{\mathbf{x}} = \begin{bmatrix} \mathbf{x}^+ \\ \mathbf{x}^- \\ \mathbf{z} \end{bmatrix} \quad (4.48b)$$

$$\bar{\mathbf{x}} \geq 0 \quad (4.48c)$$

mit dem erweiterten Variablenvektor $\bar{\mathbf{x}}$ transformiert werden. Im Weiteren wird angenommen, dass das Optimierungsproblem bereits in der Standardform (4.44) vorliegt.

Aufgabe 4.6 Transformieren Sie das folgende lineare Problem in die Standardform (4.44):

$$\min_{\mathbf{x} \in \mathbb{R}^3} -x_1 + 2x_2 + x_3 \quad (4.49a)$$

$$\text{u.B.v.} \quad x_1 + x_3 = 1 \quad (4.49b)$$

$$-x_1 + x_2 + x_3 \leq 1 \quad (4.49c)$$

$$x_2, x_3 \geq 0 \quad (4.49d)$$

4.3.2 Optimalitätsbedingungen

Mit der Lagrange-Funktion

$$L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \mathbf{c}^\top \mathbf{x} + \boldsymbol{\lambda}^\top (\mathbf{A}\mathbf{x} - \mathbf{b}) + \boldsymbol{\mu}^\top (-\mathbf{x})$$

$$\nabla_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \mathbf{c} + \mathbf{A}^\top \boldsymbol{\lambda} - \boldsymbol{\mu}$$

und $\boldsymbol{\lambda} = [\lambda_1, \dots, \lambda_p]^\top$, $\boldsymbol{\mu} = [\mu_1, \dots, \mu_n]^\top$ ergeben sich die Optimalitätsbedingungen für das lineare Problem (4.44) als Sonderfall der KKT-Bedingungen (4.29):

$$\mathbf{c} + \mathbf{A}^\top \boldsymbol{\lambda}^* - \boldsymbol{\mu}^* = \mathbf{0} \quad (4.50a)$$

$$\mathbf{A}\mathbf{x}^* - \mathbf{b} = \mathbf{0} \quad (4.50b)$$

$$\mathbf{x}^* \geq \mathbf{0} \quad (4.50c)$$

$$\mu_i^* \geq 0, \quad i = 1, \dots, n \quad (4.50d)$$

$$\mu_i^*(-x_i^*) = \mu_i^* x_i^* = 0, \quad i = 1, \dots, n. \quad (4.50e)$$

Da das lineare Problem (4.44) konvex ist (siehe Satz 2.5 und 4.4), stellt eine Lösung der Optimalitätsbedingungen (4.50) ein globales Minimum dar. Somit ist auch die Auswertung der Optimalitätsbedingungen 2. Ordnung nicht notwendig. Diese liefern ohnehin keine zusätzlichen Informationen, da die Hessematrix $\nabla_{\mathbf{x}\mathbf{x}} L = \mathbf{0}$ ist.

4.3.3 Simplex-Algorithmus

Der *Simplex-Algorithmus* wurde 1948 von *George B. Danzig* entwickelt und markierte (zusammen mit dem Aufkommen elektronischer Computer) den Startpunkt der modernen numerischen Optimierung. Der Simplex-Algorithmus nützt die geometrischen Eigenschaften von linearen Optimierungsproblemen aus, dass die zulässige Menge

$$\mathcal{X}_{\text{ad}} = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\} \quad (4.51)$$

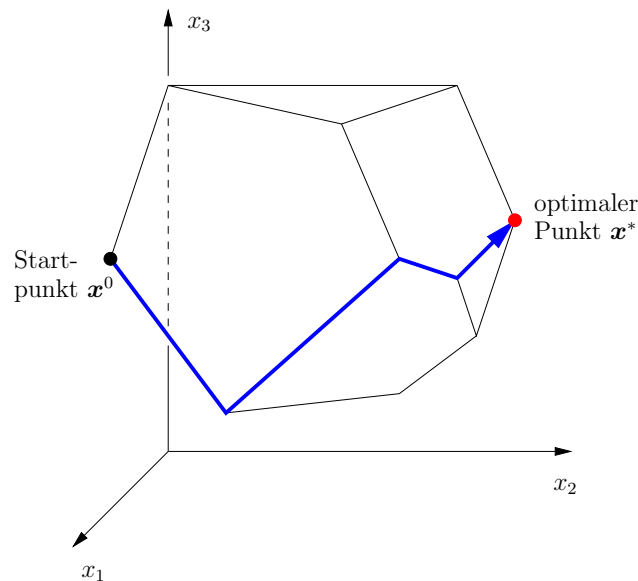


Abbildung 4.7: Veranschaulichung des Simplex-Algorithmus.

stets einen konvexen *Polyeder* mit endlich vielen Eckpunkten bildet (Abbildung 4.7). Des Weiteren kann gezeigt werden, dass die optimale Lösung $\mathbf{x}^*$ (falls sie existiert) immer auf dem Rand von $\mathcal{X}_{\text{ad}}$ liegt. Falls die Lösung $\mathbf{x}^*$ eindeutig ist, liegt sie auf einer Ecke des Polyeders. Das Simplex-Verfahren nutzt diese Eigenschaft aus, indem es von einer beliebigen Ecke des zulässigen Polyeders entlang seiner Kanten zur optimalen Lösung $\mathbf{x}^*$ läuft, siehe Abbildung 4.7.

Definition 4.3 (Notationen des Simplex-Algorithmus)

Basis und Basismatrix: Sei $\mathcal{B}$ eine Untermenge von $\{1, \dots, n\}$ mit p Elementen. Falls die $p \times p$ -Matrix $\mathbf{B} = [\mathbf{A}_i]_{i \in \mathcal{B}}$ regulär ist, dann ist $\mathcal{B}$ eine Basis und $\mathbf{B}$ die Basismatrix.²

Basisvariablen: $\mathbf{x}_B = [x_i]_{i \in \mathcal{B}} = \mathbf{B}^{-1} \mathbf{b}$

Nichtbasisvariablen: $\mathbf{x}_N = [x_i]_{i \in \mathcal{N}} = \mathbf{0}$ mit $\mathcal{N} = \{1, \dots, n\} \setminus \mathcal{B}$

Zulässige Basislösung: $\mathbf{x}$ mit $\mathbf{x}_B = \mathbf{B}^{-1} \mathbf{b} \geq 0$ und $\mathbf{x}_N = \mathbf{0}$.

Nichtdegenerierte

zulässige Basislösung: $\mathbf{x}$ mit $\mathbf{x}_B = \mathbf{B}^{-1} \mathbf{b} > 0$ und $\mathbf{x}_N = \mathbf{0}$.

Da die Gleichungsbeschränkungen (4.44b) mit $p < n$ ein unterbestimmtes Gleichungssystem darstellen, können p Elemente $\mathbf{x}_B$ berechnet werden (die Basisvariablen), während $n - p$ Elemente $\mathbf{x}_N = \mathbf{0}$ (die Nichtbasisvariablen) zu Null gesetzt werden.

Unter der Annahme, dass die $p \times n$ -Matrix $\mathbf{A}$ zeilenregulär ist, ist es stets möglich, eine Basis $\mathcal{B}$ zu bestimmen, so dass die zugehörige Basismatrix $\mathbf{B} = [\mathbf{A}_i]_{i \in \mathcal{B}}$ regulär ist. Somit können die Gleichungsbeschränkungen (4.44b) in $\mathbf{x}_B$ und $\mathbf{x}_N$ aufgespalten werden

$$\mathbf{b} = \mathbf{B} \mathbf{x}_B + \mathbf{N} \mathbf{x}_N, \quad \mathbf{N} = [\mathbf{A}_i]_{i \in \mathcal{N}}, \quad (4.52)$$

² Der Vektor $\mathbf{A}_i$ bezeichnet die i -te Spalte der Matrix $\mathbf{A}$.

um $\mathbf{x}_B$ mit Hilfe der regulären Basismatrix $\mathbf{B}$ zu berechnen:

$$\mathbf{x}_B = \mathbf{B}^{-1}(\mathbf{b} - \mathbf{N}\mathbf{x}_N) = \mathbf{B}^{-1}\mathbf{b} \quad \text{mit} \quad \mathbf{x}_N = \mathbf{0}. \quad (4.53)$$

Falls $\mathbf{x}_B \geq \mathbf{0}$ gilt, so stellen die Basisvariablen $\mathbf{x}_B$ und $\mathbf{x}_N = \mathbf{0}$ eine (zulässige) Basislösung dar. Die Initialisierungsphase des Simplex-Verfahrens besteht darin, eine Basislösung mit $\mathbf{x}_B \geq \mathbf{0}$ zu finden (siehe Abschnitt 4.3.3.2). Ist dieser Startpunkt $\mathbf{x}$ einmal bestimmt, liefern die nachfolgenden Iterationen des Simplex-Verfahrens stets zulässige Basislösungen.

Beispiel 4.4 *Betrachtet wird das Beispiel*

$$\min_{\mathbf{x} \in \mathbb{R}^4} \quad \mathbf{c}^\top \mathbf{x}, \quad \mathbf{c}^\top = [1, 2, 3, 4] \quad (4.54a)$$

$$\text{u.B.v. } \mathbf{A}\mathbf{x} = \mathbf{b}, \quad \mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & -3 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \quad (4.54b)$$

$$\mathbf{x} \geq \mathbf{0} \quad (4.54c)$$

In der folgenden Tabelle werden einige mögliche Basen $\mathcal{B}$ betrachtet. Im letzten Fall ergibt sich keine zulässige Basislösung. Keine Basis für das betrachtete Beispiel wäre $\{1, 3\}$, da in diesem Fall $\mathbf{B}$ singulär wäre.

Basis $\mathcal{B}$	Basismatrix $\mathbf{B}$	$\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b}$	$\mathbf{x}_N = \mathbf{0}$	zul. Basislösung?
$\mathcal{B} = \{1, 2\}$	$\mathbf{B} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$	$x_1 = 0.5$ $x_2 = 0.5$	$x_3 = 0$ $x_4 = 0$	ja
$\mathcal{B} = \{1, 4\}$	$\mathbf{B} = \begin{bmatrix} 1 & 1 \\ 1 & -3 \end{bmatrix}$	$x_1 = 0.875$ $x_4 = 0.125$	$x_2 = 0$ $x_3 = 0$	ja
$\mathcal{B} = \{2, 4\}$	$\mathbf{B} = \begin{bmatrix} 1 & 1 \\ 0 & -3 \end{bmatrix}$	$x_2 = 7/6$ $x_4 = -1/6$	$x_1 = 0$ $x_3 = 0$	nein

4.3.3.1 Iterationsphase

Das prinzipielle Vorgehen des Simplex-Algorithmus besteht im iterativen Überprüfen der Optimalitätsbedingungen (4.50) und dem Bestimmen einer neuen zulässigen Basislösung $\mathbf{x}^+$ mit $f(\mathbf{x}^+) < f(\mathbf{x})$.

Überprüfung der Optimalitätsbedingungen (4.50)

Eine zulässige Basislösung (4.53), d.h. $\mathbf{x}_B \geq \mathbf{0}$ und $\mathbf{x}_N = \mathbf{0}$, erfüllt bereits die KKT-Bedingungen (4.50b) und (4.50c). Als nächstes wird die Stationaritätsbedingung (4.50a) betrachtet und bezüglich der Basis $\mathcal{B}$ und $\mathcal{N}$ aufgespalten:

$$\begin{bmatrix} \mathbf{c}_B \\ \mathbf{c}_N \end{bmatrix} + \begin{bmatrix} \mathbf{B}^\top \\ \mathbf{N}^\top \end{bmatrix} \boldsymbol{\lambda} = \begin{bmatrix} \boldsymbol{\mu}_B \\ \boldsymbol{\mu}_N \end{bmatrix}. \quad (4.55)$$

Um die Komplementaritätsbedingung (4.50e) zu erfüllen, wird $\boldsymbol{\mu}_B = \mathbf{0}$ gesetzt. Für $\boldsymbol{\mu}_N$ ist (4.50e) bereits erfüllt, da $\mathbf{x}_N = \mathbf{0}$ gilt. Die in (4.55) verbleibenden Vektoren $\boldsymbol{\lambda}$ und $\boldsymbol{\mu}_N$ berechnen sich zu

$$\boldsymbol{\lambda} = -(\mathbf{B}^\top)^{-1} \mathbf{c}_B, \quad \boldsymbol{\mu}_N = \mathbf{c}_N + \mathbf{N}^\top \boldsymbol{\lambda}. \quad (4.56)$$

Die letzte verbleibende KKT-Bedingung ist $\boldsymbol{\mu}_N \geq \mathbf{0}$. Wenn diese Bedingung durch die Gleichungen (4.56) erfüllt ist, ist die Basislösung $(\mathbf{x}_B, \mathbf{x}_N)$ die optimale Lösung $\mathbf{x}^*$ und der Algorithmus terminiert.

Wahl einer neuen Basisvariablen

Falls ein oder mehrere $\mu_i, i \in \mathcal{N}$ negativ sein sollten, dann stellt $\mathbf{x}$ nicht die optimale Lösung dar. In diesem Fall konstruiert das Simplex-Verfahren eine neue zulässige Basislösung $(\mathbf{x}_B^+, \mathbf{x}_N^+)$ durch Vertauschen zweier Elemente aus $\mathbf{x}_B$ und $\mathbf{x}_N$ mit den Indices $r \in \mathcal{B}$ und $s \in \mathcal{N}$. Dazu wird $x_r^+ = 0$ gesetzt und $x_s^+ \geq 0$ zugelassen, d.h.

$$\mathbf{x}_B^+ = [\dots, x_r^+ = 0, \dots]^T, \quad \mathbf{x}_N^+ = [\dots, x_s^+ \geq 0, \dots]^T. \quad (4.57)$$

Zunächst muss ein Element x_s mit $s \in \mathcal{N}$ und $\mu_s < 0$ ausgewählt werden, dass ausgetauscht werden soll. Naheliegender ist das Element mit dem negativsten μ_i zu nehmen, also das Element aus $\mathbf{x}_N$, dass die Optimalitätsbedingung $\boldsymbol{\mu}_N \geq \mathbf{0}$ am stärksten verletzt:

$$\mu_s = \min_{i \in \mathcal{N} | \mu_i < 0} \mu_i \quad \text{mit zugehörigem Index } s \in \mathcal{N} \quad (4.58)$$

Wahl einer neuen Nichtbasisvariablen

Im vorherigen Schritt wurde das Element $s \in \mathcal{N}$ ausgewählt, das gegen ein Element $r \in \mathcal{B}$ ausgetauscht werden soll. Um den Index r der neuen Nichtbasisvariablen x_r^+ aus den bisherigen Basisvariablen $\mathbf{x}_B^+$ auszuwählen, muss der Wert von x_s^+ geeignet bestimmt werden. Einen Hinweis darauf liefert die Gleichungsbeschränkung (4.52), die für $\mathbf{x}_B^+$ und $\mathbf{x}_N^+$ erfüllt sein muss. Mit (4.53) ergibt sich für die neuen Basisvariablen

$$\begin{aligned} \mathbf{x}_B^+ &= \mathbf{B}^{-1}(\mathbf{b} - \mathbf{N}\mathbf{x}_N^+) \\ &= \mathbf{x}_B - \mathbf{B}^{-1}\mathbf{A}_s x_s^+ \quad \text{mit} \quad \mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b} \\ &= \mathbf{x}_B - \mathbf{d} x_s^+ \quad \mathbf{d} = \mathbf{B}^{-1}\mathbf{A}_s, \end{aligned} \quad (4.59)$$

wobei $\mathbf{A}_s$ die s -te Spalte der Matrix $\mathbf{A}$ bezeichnet. Der Vektor $\mathbf{d}$ stellt die Veränderungsrichtung der bisherigen Basisvariablen $\mathbf{x}_B$ bei einer Vergrößerung von x_s^+ dar.

Gilt insbesondere $d_i > 0$ für ein Element von $\mathbf{d}$, so führt eine Vergrößerung von x_s^+ zu einer Verkleinerung des entsprechenden Elements $x_{B,i}^+$ von $\mathbf{x}_B^+$. Somit darf x_s^+ nur soweit vergrößert werden, bis ein Element von $\mathbf{x}_B^+$ an die Grenze der Beschränkung $\mathbf{x}_B^+ \geq \mathbf{0}$ gelangt. Gemäß (4.59) wird aber eine Basisvariable $x_{B,i}^+$ zu Null, wenn $x_s^+ = x_{B,i}^+/d_i$ gilt. Somit kann x_s^+ mittels

$$x_s^+ = \min_{i | d_i > 0} \frac{x_{B,i}}{d_i} \quad \text{mit zugehörigem Index } r \in \mathcal{B} \quad (4.60)$$

bestimmt werden. Der Index r kennzeichnet die Basisvariable x_r^+ , die als Erstes die Grenze der Beschränkung $\mathbf{x}_B^+ \geq \mathbf{0}$ erreicht. Mit x_s^+ ergibt sich aus (4.59) der neue Basispunkt $\mathbf{x}_B^+$ in (4.57), wobei die Position von $x_r^+ = 0$ im Vektor $\mathbf{x}_B^+$ genau der Index i ist, der das Minimierungsproblem (4.60) löst.

Falls $\mathbf{d} \leq \mathbf{0}$ gilt, kann x_s^+ beliebig groß gewählt werden, ohne dass die Beschränkung $\mathbf{x}_B^+ \geq \mathbf{0}$ verletzt wird. In diesem Fall ist der zulässige Bereich $\mathcal{X}_{\text{ad}}$ des Problems unbeschränkt und es existiert keine Lösung im Endlichen.

Gegeben (durch Initialisierung oder vorherigen Schritt):

$$\mathcal{B}, \mathcal{N}, \mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b} \geq \mathbf{0}, \mathbf{x}_N = \mathbf{0}$$

KKT-Bedingungen:

$$\boldsymbol{\lambda} \leftarrow -(\mathbf{B}^\top)^{-1}\mathbf{c}_B \quad \text{Lagrange-Multiplikatoren}$$

$$\boldsymbol{\mu}_N \leftarrow \mathbf{c}_N + \mathbf{N}^\top \boldsymbol{\lambda}$$

if $\boldsymbol{\mu}_N \geq \mathbf{0}$ **then**

return $\mathbf{x}_B^* = \mathbf{x}_B, \mathbf{x}_N^* = \mathbf{x}_N$ Optimale Lösung

end

Neue zulässiger Basislösung:

Index $s \in \mathcal{N}$ mit $\mu_s < 0$ aus (4.58)

Richtung $\mathbf{d} = \mathbf{B}^{-1}\mathbf{A}_s$

if $\mathbf{d} \leq \mathbf{0}$ **then**

stop unbeschränktes Problem

end

x_s^+ und Index r aus (4.60)

$$\mathbf{x}_B^+ = \mathbf{x}_B - \mathbf{d}x_s^+$$

Neue Basisvariablen

$$\mathbf{x}_N^+ = [0, \dots, 0, x_s^+, 0, \dots, 0]^\top$$

Neue Nichtbasisvariablen

Neue Basis:

$$\mathcal{B} \leftarrow \mathcal{B} \setminus \{r\} \cup \{s\}, \quad \mathcal{N} \leftarrow \{1, \dots, n\} \setminus \mathcal{B}$$

Tabelle 4.1: Einzelner Schritt des Simplex-Algorithmus.

Da der neue zulässige Punkt (4.57) nun festgelegt ist, werden für die nächste Simplex-Iteration die Basis $\mathcal{B}$ entsprechend aktualisiert und die Indices $r \in \mathcal{B}$ und $s \in \mathcal{N}$ vertauscht. Tabelle 4.1 fasst die Operationen eines Simplex-Schrittes zusammen.

Geometrisch interpretiert bedeutet die Vergrößerung der Nichtbasisvariablen x_s^+ eine geradlinige Bewegung entlang einer Randkante des zulässigen Bereiches $\mathcal{X}_{\text{ad}}$ bis zum nächsten Randpunkt, siehe Abbildung 4.7. Dieser neue Punkt kann wieder auf Optimalität überprüft werden, bevor gegebenenfalls wieder eine neue Iteration gestartet wird. Im nichtdegenerierten Fall (siehe Definition 4.3) kann gezeigt werden, dass jeder Schritt des Simplex-Algorithmus zu einer Reduktion der Kostenfunktion führt und der Algorithmus in einer endlichen Anzahl von Schritten zum optimalen Punkt $\mathbf{x}^*$ konvergiert.

Eine Ausnahme ist der oben bereits erwähnte Fall, dass der Vektor $\mathbf{d}$ keine positiven Elemente besitzt. In diesem Fall ist (4.60) nicht lösbar und das lineare Optimierungsproblem (4.44) besitzt keine endliche Lösung.

Ein rechentechnischer Nachteil des Simplex-Verfahrens besteht darin, dass die Anzahl der Iterationen im “worst case” gleich der Anzahl der Randpunkte des zulässigen Bereiches ist, die wiederum exponentiell mit der Variablendimension $\mathbf{x} \in \mathbb{R}^n$ anwächst. Allerdings zeigt sich in den meisten praktischen Anwendungen, dass das Simplex-Verfahren wesentlich schneller konvergiert als die “worst case”-Abschätzung.

Beispiel 4.5 Für das Problem (4.54) aus Beispiel 4.4 wird das Simplex-Verfahren mit

$$\mathcal{B} = \{1, 2\}, \quad \mathbf{B} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{x}_B = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}, \quad \mathbf{c}_B = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$$\mathcal{N} = \{3, 4\}, \quad \mathbf{N} = \begin{bmatrix} 1 & 1 \\ 1 & -3 \end{bmatrix}, \quad \mathbf{x}_N = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \mathbf{c}_N = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

initialisiert. Die Auswertung der KKT-Bedingungen gemäß (4.56)

$$\boldsymbol{\lambda} = -(\mathbf{B}^\top)^{-1} \mathbf{c}_B = \begin{bmatrix} -2 \\ 1 \end{bmatrix}, \quad \boldsymbol{\mu}_N = \mathbf{c}_N + \mathbf{N}^\top \boldsymbol{\lambda} = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$$

zeigt, dass dieser Punkt nicht das Minimum darstellt ($\mu_{N,2} = \mu_4 < 0$). Ein Schritt des Simplex-Algorithmus führt zu

$$(4.58): \quad s = 4, \quad \mu_s = -1, \quad \mathbf{d} = \mathbf{B}^{-1} \mathbf{A}_4 = \begin{bmatrix} 0 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ -3 \end{bmatrix} = \begin{bmatrix} -3 \\ 4 \end{bmatrix}$$

$$(4.60): \quad r = 2, \quad x_s^+ = \frac{x_{B,2}}{d_2} = 0.125$$

$$(4.59): \quad \mathbf{x}_B^+ = \mathbf{x}_B - \mathbf{d} x_s^+ = \begin{bmatrix} 0.875 \\ 0 \end{bmatrix} = \begin{bmatrix} x_1^+ \\ x_2^+ \end{bmatrix}, \quad \mathbf{x}_N^+ = \begin{bmatrix} 0 \\ x_s^+ \end{bmatrix} = \begin{bmatrix} 0 \\ 0.125 \end{bmatrix} = \begin{bmatrix} x_3^+ \\ x_4^+ \end{bmatrix}.$$

Für den nächsten Simplexschritt werden die Elemente aus $\mathcal{B}$ und $\mathcal{N}$ mit den Indices $r \in \mathcal{B}$ und $s \in \mathcal{N}$ vertauscht, womit sich die neue zulässige Lösung

$$\mathcal{B} = \{1, 4\}, \quad \mathbf{B} = \begin{bmatrix} 1 & 1 \\ 1 & -3 \end{bmatrix}, \quad \mathbf{x}_B = \begin{bmatrix} 0.875 \\ 0.125 \end{bmatrix}, \quad \mathbf{c}_B = \begin{bmatrix} 1 \\ 4 \end{bmatrix}$$

$$\mathcal{N} = \{3, 2\}, \quad \mathbf{N} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{x}_N = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \mathbf{c}_N = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$$

ergibt. Optional können die Einträge in $\mathbf{N}$, $\mathbf{x}_B$ und $\mathbf{c}_N$ vertauscht werden, um die Indices in $\mathcal{N}$ aufsteigend zu ordnen. Die Überprüfung der KKT-Bedingungen mittels (4.56) für den neuen Punkt ergibt

$$\boldsymbol{\lambda} = -(\mathbf{B}^\top)^{-1} \mathbf{c}_B = \begin{bmatrix} -1.75 \\ 0.75 \end{bmatrix}, \quad \boldsymbol{\mu}_N = \mathbf{c}_N + \mathbf{N}^\top \boldsymbol{\lambda} = \begin{bmatrix} 2 \\ 0.25 \end{bmatrix} > \mathbf{0}.$$

Da beide Lagrange-Multiplikatoren $\boldsymbol{\mu}_N = [\mu_{N,1}, \mu_{N,2}]^\top = [\mu_3, \mu_2]^\top$ positiv sind, ist der Punkt $\mathbf{x}^* = [0.875, 0, 0, 0.125]^\top$ die optimale Lösung.

4.3.3.2 Initialisierungsphase

Falls (anders als im vorherigen Beispiel) eine zulässige Startlösung für das Simplex-Verfahren nicht leicht gefunden werden kann, kann das ursprüngliche Problem (4.44) durch einen Trick umgeschrieben werden, indem zusätzliche Variablen $\mathbf{z}$ eingeführt werden:

$$\min_{(\mathbf{x}, \mathbf{z}) \in \mathbb{R}^{n+p}} \sum_{i=1}^p z_i \tag{4.61a}$$

$$\text{u.B.v.} \quad \mathbf{A}\mathbf{x} + \mathbf{E}\mathbf{z} = \mathbf{b} \tag{4.61b}$$

$$\mathbf{x} \geq \mathbf{0}, \quad \mathbf{z} \geq \mathbf{0}. \tag{4.61c}$$

Die Matrix $\mathbf{E} = \text{diag}\{E_{ii}\} \in \mathbb{R}^{p \times p}$ wird in Abhängigkeit der Vorzeichen des Vektors $\mathbf{b}$ angesetzt

$$E_{ii} = \begin{cases} 1 & \text{falls } b_i \geq 0 \\ -1 & \text{falls } b_i < 0. \end{cases} \quad (4.62)$$

Eine zulässige Basislösung für dieses Problem ist offensichtlich

$$\mathbf{x} = \mathbf{0}, \quad z_i = |b_i|, \quad i = 1, \dots, p, \quad (4.63)$$

wodurch das modifizierte Problem (4.61) mit dem Simplex-Verfahren gelöst werden kann.

Findet man eine optimale Lösung mit $\mathbf{x}^*$ und $\mathbf{z}^* = \mathbf{0}$, so ist $\mathbf{x}^*$ eine zulässige Startlösung für das Originalproblem (4.44). Die Basis $\mathcal{B}^*$ und Basismatrix $\mathbf{B}^*$ der letzten Iteration kann für den ersten Simplex-Schritt des Originalproblems (4.44) verwendet werden. Falls andererseits nicht alle Elemente z_i^* des optimalen Vektors $\mathbf{z}^*$ identisch Null sind, so besitzt das Originalproblem (4.44) keine Lösung.

4.3.4 Interior-Point-Verfahren

Ein Nachteil des Simplex-Verfahrens ist, dass die maximale Anzahl an Rechenoperationen exponentiell mit der Problemgröße anwachsen kann, obwohl dies in der Praxis kaum der Fall ist. Eine Alternative sind *Interior-Point-Verfahren*, die maßgeblich auf *Karmarkar's Projektionsalgorithmus* aus dem Jahr 1984 basieren. Diese Verfahren zeichnen sich durch einen Rechenaufwand aus, der maximal polynomial von der Systemordnung n abhängt.

4.3.4.1 Prinzip des Verfahrens

Im Prinzip wäre es möglich, die KKT-Bedingungen (4.50) mit Hilfe eines Newton-Verfahrens direkt zu lösen, um einen optimalen Punkt $\mathbf{x}^*$ zu bestimmen. Im Allgemeinen wird dieses Vorgehen aber eine sehr langsame Konvergenz besitzen, da die Schrittweite des Newton-Verfahrens so klein gewählt werden müsste, dass in jedem Iterationsschritt die Beschränkungen $x_i \geq 0$ und $\mu_i \geq 0$ eingehalten werden.

Die Grundidee der Interior-Point-Verfahren besteht darin, die Komplementaritätsbedingung (4.50e) durch einen Parameter $x_i \mu_i = \tau$ aufzuweichen. Vektoriell geschrieben ergibt sich somit das nichtlineare Gleichungssystem

$$\mathbf{F}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \begin{bmatrix} \mathbf{c} + \mathbf{A}^\top \boldsymbol{\lambda} - \boldsymbol{\mu} \\ \mathbf{A} \mathbf{x} - \mathbf{b} \\ \mathbf{X} \mathbf{M} \mathbf{e} \end{bmatrix} - \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \tau \mathbf{e} \end{bmatrix} = \mathbf{0}, \quad (\mathbf{x}, \boldsymbol{\mu}) \geq \mathbf{0} \quad (4.64)$$

mit den Matrizen $\mathbf{X} = \text{diag}(\mathbf{x})$, $\mathbf{M} = \text{diag}(\boldsymbol{\mu})$ und $\mathbf{e} = [1, \dots, 1]^\top$. Es kann gezeigt werden, dass sich die optimale Lösung $(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ für $\tau \rightarrow 0$ ergibt:

$$\lim_{\tau \rightarrow 0} (\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = (\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*). \quad (4.65)$$

In den meisten praktischen Algorithmen die (4.64) iterativ lösen, wird τ mit Hilfe des sogenannten *Dualitätsmaßes*

$$\bar{\mu} = \frac{1}{n} \sum_{i=1}^n x_i \mu_i = \frac{1}{n} \mathbf{x}^\top \boldsymbol{\mu} \quad (4.66)$$

in jedem Iterationsschritt k wie folgt angesetzt:

$$\tau^k = \sigma^k \bar{\mu}^k, \quad \sigma^k \in [0, 1]. \quad (4.67)$$

Der Parameter σ ist dabei der Reduktionsfaktor, um den das Dualitätsmaß $\bar{\mu}^k$ im jeweiligen Schritt k verkleinert werden soll.

Die Iterationsvorschrift zum Lösen des nichtlinearen Gleichungssystems (4.64), (4.67) lautet

$$\begin{bmatrix} \mathbf{x}^{k+1} \\ \boldsymbol{\lambda}^{k+1} \\ \boldsymbol{\mu}^{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{x}^k \\ \boldsymbol{\lambda}^k \\ \boldsymbol{\mu}^k \end{bmatrix} + \alpha^k \mathbf{s}^k \quad \text{mit} \quad \mathbf{s}^k = \begin{bmatrix} \Delta \mathbf{x}^k \\ \Delta \boldsymbol{\lambda}^k \\ \Delta \boldsymbol{\mu}^k \end{bmatrix}. \quad (4.68)$$

Die Suchrichtung $\mathbf{s}^k$ berechnet sich mit Hilfe des Newton-Verfahrens (siehe auch Abschnitt 3.3.3.3) aus dem Gleichungssystem³

$$\underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{A}^\top & -\mathbf{I} \\ \mathbf{A} & \mathbf{0} & \mathbf{0} \\ \mathbf{M}^k & \mathbf{0} & \mathbf{X}^k \end{bmatrix}}_{= \nabla F(\mathbf{x}^k, \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k)} \begin{bmatrix} \Delta \mathbf{x}^k \\ \Delta \boldsymbol{\lambda}^k \\ \Delta \boldsymbol{\mu}^k \end{bmatrix} = \underbrace{\begin{bmatrix} -\mathbf{c} - \mathbf{A}^\top \boldsymbol{\lambda}^k + \boldsymbol{\mu}^k \\ -\mathbf{A} \mathbf{x}^k + \mathbf{b} \\ -\mathbf{X}^k \mathbf{M}^k \mathbf{e} + \sigma^k \bar{\mu}^k \mathbf{e} \end{bmatrix}}_{= -F(\mathbf{x}^k, \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k)}. \quad (4.69)$$

mit $\bar{\mu}^k$ aus (4.66). Darüberhinaus muss ein geeignetes Liniensuchverfahren die Schrittweite α^k in (4.68) so bestimmen, dass die strikten Beschränkungen

$$\mathbf{x}^k > \mathbf{0}, \quad \boldsymbol{\mu}^k > \mathbf{0} \quad (4.70)$$

in jedem Schritt k eingehalten werden. Aus diesem Zusammenhang entstammt auch der Name *Interior-Point-Verfahren*. Einen Einfluss auf die Schrittweite α^k besitzt auch der Reduktionsfaktor $\sigma^k \in [0, 1]$. Für größere Werte von σ^k (gegen 1 gehend) ist es in der Regel möglich, einen größeren Schritt α^k in die durch (4.69) definierte Newton-Richtung $\mathbf{s}^k$ zu gehen, bevor die Beschränkungen $(\mathbf{x}^k, \boldsymbol{\mu}^k) > \mathbf{0}$ verletzt werden.

Es existieren mehrere unterschiedlich komplexe Verfahren zur Wahl von α^k und σ^k , auf die an dieser Stelle nicht näher eingegangen werden soll (z.B. Kurzschrift-, Langschrittsverfahren, Prädiktor-Korrektorverfahren). Eine große Rolle spielt des Weiteren die Wahl der Startlösung $(\mathbf{x}^0, \boldsymbol{\lambda}^0, \boldsymbol{\mu}^0)$, die das Gleichungssystem (4.64) und die Beschränkungen $(\mathbf{x}^0, \boldsymbol{\mu}^0) > \mathbf{0}$ ausreichend genau erfüllen sollte.

Interior-Point-Verfahren sind auch für große Probleme gut einsetzbar. In diesem Fall gibt es effiziente Algorithmen, die die meistens dünn-besetzte Struktur des linearen Gleichungssystems (4.69) ausnutzen. Mehr Informationen zu diesen Punkten findet man z.B. in [1, 4].

³ Gleichung (4.69) ergibt sich aus der Approximation von (4.64) mittels des *linearen Modells* $m_F^k(\mathbf{s}) = \mathbf{F}(\mathbf{x}^k, \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k) + \nabla \mathbf{F}(\mathbf{x}^k, \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k) \mathbf{s}$ und der Forderung $m_F^k(\mathbf{s}) = 0$. Die Bestimmung der Newton-Richtung $\mathbf{s}$ ist analog zum Vorgehen in Abschnitt 3.3.3.3. Dort wurde zunächst ein *quadratisches Modell* (3.32) aufgestellt. Die Bestimmung des Minimums gemäß $\nabla m^k(\mathbf{s}) = 0$ führte dabei ebenfalls zu einem linearen Modell in $\mathbf{s}$.

4.3.4.2 Zusammenhang mit Barrieremethoden

Interior-Point-Verfahren sind eng verwandt mit Barrieremethoden, bei denen die Beschränkungen $\mathbf{x} \geq \mathbf{0}$ durch logarithmische Strafterme zur Kostenfunktion addiert werden:

$$\min_{\mathbf{x} \in \mathbb{R}^n} \quad \mathbf{c}^\top \mathbf{x} - \tau \sum_{i=1}^n \log x_i, \quad \tau > 0 \quad (4.71a)$$

$$\text{u.B.v.} \quad \mathbf{Ax} = \mathbf{b}. \quad (4.71b)$$

Da die Kostenfunktion (4.71a) aufgrund der log-Terme gegen Unendlich geht, wenn eine Variable x_i zu Null wird, muss eine Lösung des Problems (4.71) die strikte Beschränkung $\mathbf{x} > \mathbf{0}$ erfüllen. Der Strafparameter τ beeinflusst die Form der Barrierenfunktionen, siehe Abbildung 4.8. Für $\tau \rightarrow 0$ nähern die log-Terme ein diskontinuierliches Verhalten an, während die Bestrafung für innere Punkte $\mathbf{x} > \mathbf{0}$ immer geringer wird.

Mit der Lagrange-Funktion

$$L(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{c}^\top \mathbf{x} - \tau \sum_{i=1}^n \log x_i + \boldsymbol{\lambda}^\top (\mathbf{Ax} - \mathbf{b})$$

lauten die KKT-Bedingungen (4.50a), (4.50b) für das Barriereproblem

$$\begin{aligned} c_i - \frac{\tau}{x_i} + \mathbf{A}_i^\top \boldsymbol{\lambda} &= 0, \quad i = 1, \dots, n \\ \mathbf{Ax} &= \mathbf{b}, \end{aligned} \quad (4.72)$$

wobei $\mathbf{A}_i$ die i -te Spalte der Matrix $\mathbf{A}$ darstellt. Mit der Einführung der neuen Variablen $\mu_i = \tau/x_i$ ergibt sich das gleiche Gleichungssystem wie in (4.64):

$$\mathbf{F}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \begin{bmatrix} \mathbf{c} + \mathbf{A}^\top \boldsymbol{\lambda} - \boldsymbol{\mu} \\ \mathbf{Ax} - \mathbf{b} \\ \mathbf{XMe} \end{bmatrix} - \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \tau \mathbf{e} \end{bmatrix} = \mathbf{0}, \quad (\mathbf{x}, \boldsymbol{\mu}) \geq \mathbf{0}$$

Die Optimalitätsbedingungen für Interior-Point-Verfahren und logarithmische Barrieremethoden sind also (über die Einführung der neuen Variablen $\mu_i = \tau/x_i$) äquivalent.

Reine Barrieremethoden wie in (4.71) sind in der Regel numerisch sensibel, da die Terme in den KKT-Bedingung (4.72) sehr groß werden können, wenn man im Grenzfalle $\tau \rightarrow 0$ nahe an die Beschränkungen $\mathbf{x} \geq \mathbf{0}$ gelangt. Die Formulierung der Interior-Point-Verfahren ist in der Regel vorzuziehen, da gezeigt werden kann, dass das Gleichungssystem (4.64) eine bessere Konditionierung aufweist als die Optimalitätsbedingungen (4.72) von Barrieremethoden.

Das Verhalten der logarithmischen Barrierenfunktionen kann jedoch verwendet werden, um die Interior-Point-Verfahren zu veranschaulichen. Jeder Wert $\tau^k \geq 0$ definiert eine Lösung $(\mathbf{x}^k, \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k)$. Die Menge aller Lösungen für verschiedene τ beschreibt den sogenannten *zentralen Pfad*, der für eine Sequenz $\{\tau^k\}$ mit $\tau^{k+1} < \tau^k$ im Grenzfalle $\lim_{k \rightarrow \infty} \tau^k = 0$ zur optimalen Lösung $(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ des ursprünglichen Problems (4.44) führt, siehe Abbildung 4.9. Da τ dem Parameter (4.67) entspricht, weisen Interior-Point-Verfahren das gleiche Verhalten auf und nähern die optimale Lösung über den zentralen Pfad an.

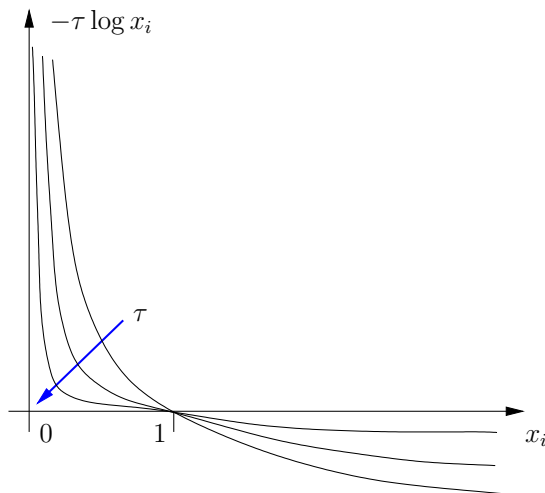


Abbildung 4.8: Logarithmische Barrierefunktion für verschiedene Werte von τ .

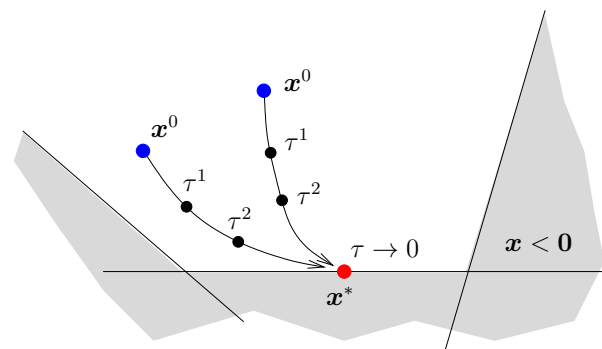


Abbildung 4.9: Zentraler Pfad, der durch τ definiert wird.

4.3.5 Beispiel: Transportnetz

Das in zwei Raffinerien R_1 und R_2 hergestellte Benzin soll über das in der Abbildung 4.10 dargestellte Transportnetz bestehend aus vier Pipelines $l_1, \dots, l_4$ zu den Verbrauchszentren Z_1, Z_2 transportiert werden.⁴ Die Verbrauchszentren Z_1, Z_2 haben die konstanten Verbrauchsrate V_1, V_2 . Die Transportkosten innerhalb einer Leitung seien direkt proportional zu dem Durchfluss x_i der Leitung mit dem Proportionalitätsfaktor K_i .

Gesucht wird die optimale Verteilung der Durchflüsse x_i zur Minimierung der Transportkosten unter Gewährleistung der Verbrauchsrate V_1, V_2 und der Einhaltung der maximalen Durchflusskapazitäten C_i der Leitungen. Des Weiteren soll berücksichtigt werden, dass die Raffinerie R_1 eine maximale Produktionsrate P_1 besitzt.

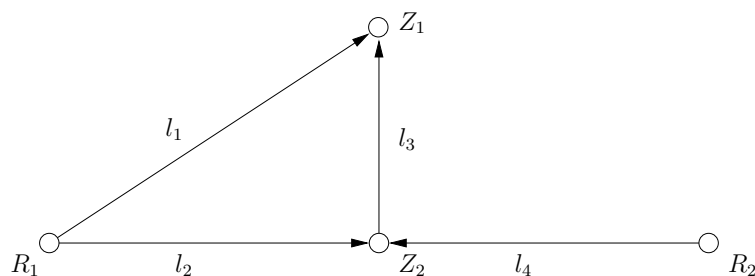


Abbildung 4.10: Transportnetz des Raffinerien-Beispiels.

Für die Parameter werden folgende Werte betrachtet:

$$\begin{aligned} K_1 &= 40, & C_1 &= 30, & V_1 &= 50, & P_1 &= 45 \\ K_2 &= 20, & C_2 &= 20, & V_2 &= 40 \\ K_3 &= 10, & C_3 &= 30 \\ K_4 &= 10, & C_4 &= 60. \end{aligned}$$

⁴ Dies ist eine modifizierte Version des Raffinerien-Beispiels aus [2].

Aufgabe 4.7 Formulieren Sie das Transportproblem als Optimierungsproblem. Reduzieren Sie anschließend das Problem auf ein zweidimensionales Optimierungsproblem durch Einsetzen der Gleichungsbeschränkungen.

Aufgabe 4.8 Stellen Sie den zulässigen Bereich und die Höhenlinien der Kostenfunktion grafisch dar und ermitteln Sie auf diese Weise die optimale Lösung.

Die Optimization Toolbox von MATLAB stellt die Funktion

$$\text{linprog}(\mathbf{f}, \mathbf{A}, \mathbf{b}, \mathbf{Aeq}, \mathbf{beq}, \mathbf{LB}, \mathbf{UB}, \mathbf{X0}, \text{options})$$

zur Lösung von linearen Optimierungsproblemen der Form

$$\min_{\mathbf{x} \in \mathbb{R}^n} \mathbf{c}^\top \mathbf{x} \quad \text{u.B.v.} \quad \mathbf{A}_{\text{eq}} \mathbf{x} = \mathbf{b}_{\text{eq}}, \quad \mathbf{A} \mathbf{x} \leq \mathbf{b}, \quad \mathbf{LB} \leq \mathbf{x} \leq \mathbf{UB}$$

zur Verfügung. Das Simplex-Verfahren bzw. das Interior-Point-Verfahren können über die folgenden Optionen eingestellt werden:⁵

Simplex-Verfahren: options = optimset('LargeScale','off','Simplex','on')

Interior-Point-Verfahren: options = optimset('LargeScale','on')

Aufgabe 4.9 Lösen Sie das Transportproblem unter MATLAB (siehe Abbildung 4.11) und vergleichen Sie das Simplex-Verfahren mit dem Interior-Point-Verfahren hinsichtlich Genauigkeit und Anzahl an Iterationen.

4.4 Quadratische Optimierung

Ein weiterer Sonderfall in der statischen Optimierung ist die quadratische Optimierung (auch *quadratische Programmierung* genannt), zu der Optimierungsprobleme mit quadratischer Kostenfunktion und linearen Beschränkungen gehören:

$$\min_{\mathbf{x} \in \mathbb{R}^n} \quad \frac{1}{2} \mathbf{x}^\top \mathbf{G} \mathbf{x} + \mathbf{c}^\top \mathbf{x}, \quad \mathbf{G} \in \mathbb{R}^{n \times n} \quad (4.73a)$$

$$\text{u.B.v.} \quad \mathbf{A} \mathbf{x} = \mathbf{b}, \quad \mathbf{A} \in \mathbb{R}^{p \times n}, \quad \mathbf{b} \in \mathbb{R}^p \quad (4.73b)$$

$$\mathbf{C} \mathbf{x} \leq \mathbf{d}, \quad \mathbf{C} \in \mathbb{R}^{q \times n}, \quad \mathbf{d} \in \mathbb{R}^q. \quad (4.73c)$$

Ebenso wie die lineare Optimierung (Abschnitt 4.3) ist diese Problemklasse von besonderer Bedeutung. Ein Beispiel ist die Portfolio-Optimierung, bei der ein Optimum zwischen Ertragsmaximierung und Risikominimierung (Kovarianz) erzielt werden soll. Darüber hinaus basieren viele numerische Verfahren zur Lösung von allgemeinen nichtlinearen Optimierungsproblemen (Abschnitt 4.5) auf einer sequentiellen Lösung von quadratischen Problemen.

⁵ Der Startwert $\mathbf{X0}$ wird beim Aufruf von `linprog` ignoriert, falls das Simplex- oder Interior-Point-Verfahren verwendet wird, da bei beiden Verfahren automatisch ein Startwert berechnet wird.

```

function transportnetz(methodQ)
% -----
% methodQ: 1 - Simplex-Verfahren
%           2 - Interior-Point-Verfahren

opt = optimset('Display','iter');           % Optionen für Ausgabe

switch methodQ
    case 1,                                % Simplex-Verfahren
        opt = optimset(opt,'LargeScale','off','Simplex','on');
    case 2,                                % Interior-Point-Verfahren
        opt = optimset(opt,'LargeScale','on');
end

K = [40; 20; 10; 10];                     % Parameter
C = [30; 20; 30; 60];                     % für Transportproblem
V = [50; 40];

xopt = [20;10;30;60];                     % optimaler Punkt

c = K;                                     % Kostenvektor
Aeq = [1,0,1,0; 0,1,-1,1]; beq = V;        % Gleichungsbeschränkungen
LB = [0;0;0;0]; UB = C;                   % Beschränkungen von x
A = [1,1,0,0]; b = 45;                     % allg. Beschränkung Ax <= b

% Lösung mit linprog
% -----
[x,f] = linprog(c,A,b,Aeq,beq,LB,UB,[],opt);
fprintf('\nBerechnete Lösung: [%g, %g, %g, %g] mit f(x) = %g',x,f);
fprintf('\nAbstand zum Optimum: [%2.2g, %2.2g, %2.2g, %2.2g]\n',x-xopt);

```

Abbildung 4.11: MATLAB-Code für das Transportnetz-Problem (linprog).

4.4.1 Optimalitätsbedingungen

Die allgemein nichtlinearen KKT-Bedingungen (4.29) vereinfachen sich strukturell für quadratische Probleme der Form (4.73). Mit Hilfe der Lagrange-Funktion

$$L(x, \lambda, \mu) = \frac{1}{2} x^T G x + c^T x + \lambda^T (Ax - b) + \mu^T (Cx - d)$$

und den Lagrange-Multiplikatoren $\lambda \in \mathbb{R}^p$, $\mu \in \mathbb{R}^q$ lauten die KKT-Bedingungen für eine optimale Lösung

$$Gx^* + c + A^T \lambda^* + C^T \mu^* = 0 \quad (4.74a)$$

$$Ax^* - b = 0 \quad (4.74b)$$

$$Cx^* - d \leq 0 \quad (4.74c)$$

$$\mu^* \geq 0 \quad (4.74d)$$

$$\mu^{*\top} (Cx^* - d) = 0. \quad (4.74e)$$

Falls G positiv semi-definit ist, spricht man von einem *konvexen* quadratischen Problem. Im konvexen Fall ist eine Lösung x^* (falls sie existiert) immer eine globale Lösung des quadratischen Problems und die KKT-Bedingungen (4.29) sind notwendig und hinreichend für die Existenz von x^* . Nicht-konvexe quadratische Probleme sind im Allgemeinen schwieriger zu handhaben, da mehrere stationäre Punkte und Minima auftreten können.

4.4.2 Gleichungsbeschränkte quadratische Probleme

Von besonderem Interesse sind gleichungsbeschränkte quadratische Probleme

$$\min_{\mathbf{x} \in \mathbb{R}^n} \quad \frac{1}{2} \mathbf{x}^\top \mathbf{G} \mathbf{x} + \mathbf{c}^\top \mathbf{x} \quad (4.75a)$$

$$\text{u.B.v.} \quad \mathbf{A} \mathbf{x} = \mathbf{b}, \quad \mathbf{A} \in \mathbb{R}^{p \times n}, \quad \mathbf{b} \in \mathbb{R}^p, \quad (4.75b)$$

für die sich die KKT-Bedingungen (4.74) auf das lineare Gleichungssystem

$$\begin{bmatrix} \mathbf{G} & \mathbf{A}^\top \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}^* \\ \boldsymbol{\lambda}^* \end{bmatrix} = \begin{bmatrix} -\mathbf{c} \\ \mathbf{b} \end{bmatrix} \quad (4.76)$$

reduzieren. Das Gleichungssystem (4.76) umfasst $n + p$ Gleichungen für die $n + p$ Unbekannten $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$. Die Blockmatrix in (4.76) wird *Karush–Kuhn–Tucker–Matrix* (kurz: *KKT–Matrix*) genannt.

Es existieren unterschiedliche Verfahren zur Lösung von (4.76), die an dieser Stelle nur kurz skizziert werden sollen.⁶

- *Direkte Verfahren* lösen das Gleichungssystem (4.76) mit Hilfe von Faktorisierungen, z.B. das *Nullraum–Verfahren*. Diese Verfahren werden vorrangig bei Problemen kleiner und mittlerer Dimension eingesetzt, da die Berechnung der Faktorisierungen mit zunehmender Dimension recht aufwändig werden kann.
- *Iterative Verfahren* sind für große Problemdimensionen besonders gut geeignet. Beispiele dafür sind bestimmte Klassen von *Konjugierte–Gradienten–Verfahren*.

Gleichungsbeschränkte quadratische Probleme sind von großer Bedeutung in der Optimierung, da das zu lösende Gleichungssystem (4.76) linear ist und mittels der oben erwähnten Verfahren effizient gelöst werden kann.

Des Weiteren basieren ungleichungsbeschränkte quadratische Probleme sowie generell nichtlineare Probleme oft auf der (sequentiellen) Lösung von gleichungsbeschränkten Unterproblemen der Form (4.75).

4.4.3 Aktive–Restriktionen–Verfahren

Eine bekannte Methode zur Lösung von allgemeinen quadratischen Problemen (4.73) ist das *Aktive–Restriktionen–Verfahren*, das (ähnlich wie das Simplex–Verfahren in der Linearen Optimierung) aktive Beschränkungen in jeder Iteration identifiziert und ein gleichungsbeschränktes Problem der Form (4.75) mit einer kleineren Anzahl von freien Optimierungsvariablen löst.

Die Grundidee des Aktive–Restriktionen–Verfahrens besteht darin, in jeder Iteration k eine Anzahl von aktiven Ungleichungsbeschränkungen $\mathbf{C} \mathbf{x} \leq \mathbf{d}$ zu bestimmen, die als *Arbeitsmenge* $\mathcal{A}^k$ bezeichnet wird:

$$\mathbf{C}_i^\top \mathbf{x}^k = d_i, \quad i \in \mathcal{A}^k.$$

⁶Mehr Informationen findet man z.B. in [1].

Dabei stellt $\mathbf{C}_i^\top$ die i -te Zeile der Beschränkungsmatrix $\mathbf{C}$ dar. Die Menge $\mathcal{A}^k$ berücksichtigt nicht notwendigerweise alle im Punkt $\mathbf{x}^k$ aktiven Beschränkungen, sondern wird im Laufe der Iterationen vervollständigt, bis der optimale Punkt $\mathbf{x}^*$ mit der Menge der aktiven Beschränkungen (siehe Definition 4.1)

$$\mathcal{A}(\mathbf{x}^*) = \{i \in \{1, \dots, q\} \mid \mathbf{C}_i^\top \mathbf{x}^* = d_i\}$$

erreicht ist. Im Allgemeinen wird angenommen, dass in jedem Iterationspunkt $\mathbf{x}^k$ die Beschränkungsqualifikation (4.26) erfüllt ist, d.h.

$$\text{Rang} \begin{bmatrix} \mathbf{A} \\ [\mathbf{C}_i^\top]_{i \in \mathcal{A}^k} \end{bmatrix} = p + |\mathcal{A}^k| \quad \forall k. \quad (4.77)$$

Des Weiteren wird angenommen, dass die Matrix $\mathbf{G}$ positiv definit ist.

Das Aktive–Restriktionen–Verfahren überprüft in jeder Iteration k , ob der Punkt $\mathbf{x}^k$ mit der Arbeitsmenge $\mathcal{A}^k$ die optimale Lösung des quadratischen Problems darstellt. Wenn dies nicht der Fall ist, wird ein Schritt $\mathbf{s}$ berechnet, indem ein gleichungsbeschränktes Problem gelöst wird, bei dem alle aktiven Ungleichungen entsprechend der Arbeitsmenge $\mathcal{A}^k$ als Gleichungsbeschränkungen betrachtet werden. Aus dem Originalproblem (4.73) ergibt sich somit durch Ersetzen von $\mathbf{x} = \mathbf{x}^k + \mathbf{s}$ und der momentanen Arbeitsmenge $\mathcal{A}^k$ das gleichungsbeschränkte quadratische Problem⁷

$$\min_{\mathbf{s} \in \mathbb{R}^n} \quad \frac{1}{2} \mathbf{s}^\top \mathbf{G} \mathbf{s} + (\mathbf{G} \mathbf{x}^k + \mathbf{c})^\top \mathbf{s} \quad (4.78a)$$

$$\text{u.B.v.} \quad \mathbf{A} \mathbf{s} = \mathbf{0}, \quad \mathbf{C}_i^\top \mathbf{s} = 0, \quad i \in \mathcal{A}^k \quad (4.78b)$$

mit der Lösung $\mathbf{s}^k$. Eine wichtige Eigenschaft dieser Formulierung ist, dass jede aktive Beschränkung $i \in \mathcal{A}^k$ aktiv bleibt und nicht verletzt wird, da

$$\mathbf{C}_i^\top (\mathbf{x}^k + \alpha \mathbf{s}^k) = \mathbf{C}_i^\top \mathbf{x}^k = d_i, \quad \forall i \in \mathcal{A}^k \quad \forall \alpha \geq 0. \quad (4.79)$$

Somit sind alle aktiven Beschränkungen in $\mathcal{A}^k$, die in $\mathbf{x}^k$ aktiv sind, ebenfalls für $\mathbf{x}^k + \alpha \mathbf{s}^k$ für alle Schrittweiten α aktiv. Da angenommen wurde, dass $\mathbf{G}$ positiv definit ist, kann (4.78) mit einem der in Abschnitt 4.4.2 erwähnten Verfahren gelöst werden.

Falls für die Lösung $\mathbf{s}^k \neq \mathbf{0}$ gilt und $\mathbf{x}^k + \mathbf{s}^k$ ein zulässiger Punkt im Hinblick auf die Beschränkungen $\mathbf{C}(\mathbf{x}^k + \mathbf{s}^k) \leq \mathbf{d}$ ist, wird $\mathbf{x}^{k+1} = \mathbf{x}^k + \mathbf{s}^k$ als neuer Iterationspunkt verwendet. Wenn $\mathbf{x}^k + \mathbf{s}^k$ nicht zulässig ist, wird

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k \mathbf{s}^k \quad (4.80)$$

gesetzt, wobei die Schrittweite $\alpha^k < 1$ soweit verkleinert wird, bis alle Beschränkungen erfüllt sind. Um α^k geeignet zu bestimmen, müssen lediglich die Beschränkungen betrachtet werden, die nicht in $\mathcal{A}^k$ enthalten sind, da alle aktiven Beschränkungen $i \in \mathcal{A}^k$ für alle α^k erfüllt bleiben, siehe (4.79).

Von Interesse sind die Beschränkungen $i \notin \mathcal{A}^k$, für die $\mathbf{C}_i^\top \mathbf{s}^k > 0$ gilt. In diesem Fall wird die ursprüngliche Beschränkung $\mathbf{C}_i^\top (\mathbf{x}^k + \alpha^k \mathbf{s}^k) \leq d_i$ nur erfüllt sein, wenn

$$\alpha^k \leq \frac{d_i - \mathbf{C}_i^\top \mathbf{x}^k}{\mathbf{C}_i^\top \mathbf{s}^k} \quad (4.81)$$

⁷ Beim Ersetzen von $\mathbf{x} = \mathbf{x}^k + \mathbf{s}$ in der Kostenfunktion (4.73a) entsteht ein zusätzlicher Term $\frac{1}{2} (\mathbf{x}^k)^\top \mathbf{G} \mathbf{x}^k + \mathbf{c}^\top \mathbf{x}^k$, der aber unabhängig von $\mathbf{s}$ ist und somit in der neuen Kostenfunktion (4.78a) vernachlässigt werden kann.

gilt. Die größtmögliche Schrittweite α^k lässt sich wie folgt bestimmen:

$$\alpha^k = \min \left\{ 1, \min_{i \notin \mathcal{A}^k, \mathbf{C}_i^\top \mathbf{s}^k > 0} \frac{d_i - \mathbf{C}_i^\top \mathbf{x}^k}{\mathbf{C}_i^\top \mathbf{s}^k} \right\}. \quad (4.82)$$

Falls $\alpha^k < 1$ aus (4.82) folgt und somit eine aktiv werdende Beschränkung außerhalb der Arbeitsmenge $\mathcal{A}^k$ die maximale Schrittweite $\alpha^k = 1$ verhindert, wird eine neue Arbeitsmenge $\mathcal{A}^{k+1}$ gebildet, indem der Index i der neuen aktiven Ungleichungsbeschränkung zu $\mathcal{A}^k$ hinzugefügt wird.

Dieses Verfahren wird iterativ wiederholt, d.h. aktive Beschränkungen sukzessive zur Arbeitsmenge $\mathcal{A}^k$ hinzugefügt, bis ein Punkt $\mathbf{x}^k$ die quadratische Kostenfunktion (4.78a) minimiert. In diesem Fall muss $\mathbf{s}^k = \mathbf{0}$ gelten. Damit die KKT-Bedingungen (4.74) an diesem Punkt $\mathbf{x}^k$ tatsächlich erfüllt sind, müssen alle Lagrange-Multiplikatoren μ_i^k , $i \in \mathcal{A}^k$, die den aktiven Beschränkungen in der Arbeitsmenge $\mathcal{A}^k$ entsprechen, nicht-negativ sein, d.h. $\mu_i^k \geq 0$, $i \in \mathcal{A}^k$. Ist dies der Fall, stellt $\mathbf{x}^k$ die optimale Lösung $\mathbf{x}^*$ dar. Da $\mathbf{G}$ positiv definit ist, ist $\mathbf{x}^* = \mathbf{x}^k$ auch die globale Lösung.

Falls andererseits mindestens einer der Lagrange-Multiplikatoren μ_i^k , $i \in \mathcal{A}^k$ negativ ist, bedeutet dies, dass die Kostenfunktion (4.73a) weiter reduziert werden kann, wenn eine der aktiven Beschränkungen in $\mathcal{A}^k$ fallen gelassen wird (siehe Abschnitt 4.1.2 und 4.1.3). Die Reduktion von $\mathcal{A}^k$ um ein Element $i \in \mathcal{A}^k$ mit $\mu_i^k < 0$ und die anschließende erneute Lösung des Problems (4.78) erzeugt eine zulässige neue Richtung $\mathbf{s}^{k+1}$.

Der grundlegende Algorithmus des Aktive-Restriktionen-Verfahrens ist in Tabelle 4.2 zusammengefasst. Es kann gezeigt werden, dass das Aktive-Restriktionen-Verfahren für strikt konvexe quadratische Probleme in einer endlichen Anzahl von Iterationen konvergiert.

Beispiel 4.6 ([1]) Betrachtet wird das Beispiel

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^2} \quad & (x_1 - 1)^2 + (x_2 - 2.5)^2 \\ \text{u.B.v.} \quad & 1) \quad -x_1 + 2x_2 - 2 \leq 0 \\ & 2) \quad x_1 + 2x_2 - 6 \leq 0 \\ & 3) \quad x_1 - 2x_2 - 2 \leq 0 \\ & 4) \quad x_1 \geq 0 \\ & 5) \quad x_2 \geq 0. \end{aligned}$$

Der zulässige Bereich $\mathcal{X}_{\text{ad}}$ und die Höhenlinien der Kostenfunktion sind in Abbildung 4.12 dargestellt. Die Beschränkungen werden im Folgenden mit den Indices 1 bis 5 entsprechend ihrer Reihenfolge gekennzeichnet.

Start: Als Startwert für das Aktive-Beschränkungen-Verfahren wird $\mathbf{x}^0 = [2, 0]^\top$ gewählt. Da die Beschränkungen 3 und 5 in $\mathbf{x}^0$ aktiv sind, wird die Arbeitsmenge zu $\mathcal{A}^0 = \{3, 5\}$ gesetzt. Die Lösung des unterlagerten Problems (4.78) führt zu $\mathbf{s}^0 = \mathbf{0}$. Die Auswertung der KKT-Bedingung (4.74a) mit $\mu_i^0 = 0$, $i = 1, 2, 4$ führt auf die Lagrange-Multiplikatoren $\mu_3 = -2$ und $\mu_5 = -1$ der aktiven Beschränkungen 3 und 5.

Iteration 1: Beschränkung 3 wird aus der Arbeitsmenge $\mathcal{A}^0$ entfernt, da $\mu_3^0 < \mu_5^0$. Somit ergibt sich $\mathcal{A}^1 = \{5\}$. Die Lösung von (4.78) lautet $\mathbf{s}^1 = [-1, 0]^\top$. Die Schrittweite

Initialisierung:	$\mathbf{x}^0$ & $\mathcal{A}^0$	Startpunkt & Arbeitsmenge
for $k = 0, 1, 2, \dots$	do	
	$\mathbf{s}^k$ & $\mu_i^k, i \in \mathcal{A}^k \leftarrow (4.78) \text{ \& } (4.74)$	gleichungsbeschr. quadr. Problem (4.78)
	if $\mathbf{s}^k = \mathbf{0}$	then Minimum gefunden ($\mathbf{s}^k = \mathbf{0}$)?
		if $\mu_i^k \geq 0 \forall i \in \mathcal{A}^k$
		then return $\mathbf{x}^* = \mathbf{x}^k$
		optimale Lösung
		else
		$i^k \leftarrow \arg \min_{\mathcal{A}^k} \mu_i^k$
		$\mathcal{A}^{k+1} \leftarrow \mathcal{A}^k \setminus \{i^k\}, \mathbf{x}^{k+1} \leftarrow \mathbf{x}^k$
		Deaktivieren aktiver Beschränkung i^k
		end
	else	Minimum noch nicht gefunden ($\mathbf{s}^k \neq \mathbf{0}$)
		α^k & $i^k \leftarrow (4.82)$
		Schrittweitenbestimmung
		if $\alpha^k < 1$
		then $\mathcal{A}^{k+1} \leftarrow \mathcal{A}^k \cup \{i^k\}$
		Aktivieren neuer Beschränkung i^k
		else
		$\mathcal{A}^{k+1} \leftarrow \mathcal{A}^k$
		end
		$\mathbf{x}^{k+1} \leftarrow \mathbf{x}^k + \alpha^k \mathbf{s}^k$
		nächster Iterationspunkt
	end	
end		

Tabelle 4.2: Aktive-Restriktionen-Verfahren.

$\alpha^k = 1$ folgt aus (4.82). Somit ist ein “voller” Schritt $\mathbf{x}^2 = \mathbf{x}^1 + \mathbf{s}^1 = [1, 0]^\top$ ohne das Auftreten einer weiteren aktiven Beschränkung möglich, so dass $\mathcal{A}^2 = \mathcal{A}^1$ gesetzt wird, siehe Abbildung 4.12.

Iteration 2: Die Lösung von (4.78) ergibt $\mathbf{s}^2 = \mathbf{0}$. Die Auswertung der KKT-Bedingung (4.74a) liefert $\mu_5^2 = -5$ für die einzige aktive Beschränkung in $\mathcal{A}^2 = \{5\}$. Gemäß dem Algorithmus in Tabelle 4.2 wird 5 aus $\mathcal{A}^2$ entfernt, was zur leeren Menge $\mathcal{A}^3 = \{\}$ führt.

Iteration 3: Die Lösung des jetzt unbeschränkten Problems (4.78) führt auf $\mathbf{s}^3 = [0, 2.5]^\top$ mit der Schrittweite $\alpha^3 = 0.6$ gemäß (4.82). Somit ergibt sich der neue Iterationspunkt $\mathbf{x}^4 = [1, 1.5]^\top$. Da die (jetzt aktive) Beschränkung 1 einen vollen Schritt blockierte, wird $\mathcal{A}^4 = \{1\}$ gesetzt.

Iteration 4: Jetzt ergibt sich $\mathbf{s}^4 = [0.4, 0.2]^\top$ mit $\alpha^4 = 1$. Da keine neue Beschränkung aktiviert wird, bleibt die Arbeitsmenge $\mathcal{A}^5 = \mathcal{A}^4 = \{1\}$ unverändert und es ergibt sich $\mathbf{x}^5 = [1.4, 1.7]^\top$, siehe Abbildung 4.12.

Iteration 5: Im letzten Schritt liefert (4.78) $\mathbf{s}^5 = \mathbf{0}$ mit dem Lagrange-Multiplikator $\mu_5^5 = 0.8 > 0$ aus (4.74a). Somit stellt $\mathbf{x}^5 = \mathbf{x}^*$ die optimale Lösung dar und der Algorithmus terminiert.

Aufgabe 4.10 Verifizieren Sie die Lösung des quadratischen Problems in Beispiel 4.6 mit Hilfe der MATLAB-Funktion `quadprog` aus der Optimization Toolbox (siehe auch Abbildung 4.13).

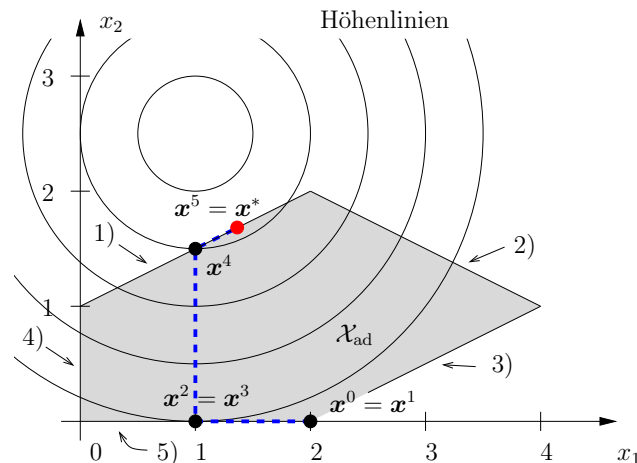


Abbildung 4.12: Iterationen des Aktiven-Beschränkungen-Verfahrens in Beispiel 4.6.

```

% Matlab-Lösung eines quadratischen Beispielpblems
% -----
opt = optimset('LargeScale','off');

G = [2,0; 0,2];
c = [-2;-5];
C = [-1,2; 1,2; 1,-2; -1,0; 0,-1];
d = [ 2; 6; 2; 0; 0];

x0 = [2.0;0.0];

[x,f] = quadprog(G,c,C,d,[],[],[],[],x0,opt);
fprintf('\nBerechnete Lösung: [%g, %g] mit f(x) = %g\n',x,f);

```

Abbildung 4.13: MATLAB-Code zu Aufgabe 4.10 (quadprog).

4.4.4 Weitere numerische Verfahren

Das Aktive-Restriktionen-Verfahren ist gut geeignet für Probleme kleiner bis mittlerer Dimension, kann aber eine langsame Konvergenz bei hoch-dimensionalen QPs aufweisen, da die Arbeitsmenge $\mathcal{A}^k$ nur jeweils um ein Element reduziert oder erweitert wird. Alternativ sind *Projizierte-Gradienten-Verfahren* für große Probleme gut geeignet, da sie eine schnelle Änderung der Arbeitsmenge $\mathcal{A}^k$ erlauben.

Eine weitere Verfahrensklasse zur Lösung von quadratischen Problemen sind *Interior-Point-Verfahren*, wie sie bereits in der Linearen Programmierung vorgestellt wurden (Abschnitt 4.3.4). Durch die Einführung von *Schlupfvariablen* können die Ungleichungsbeschränkungen (4.73c) dabei wieder auf die Standardform (4.44b)–(4.44c) gebracht werden.

Weitere Informationen zu Projizierte-Gradienten-Verfahren und Interior-Point-Methoden im Rahmen der Quadratischen Optimierung finden sich in den Büchern am Ende des Kapitels.

4.5 Nichtlineare Optimierung

Nach den betrachteten Sonderfällen der Linearen und Quadratischen Optimierung soll nun auf Lösungsstrategien für allgemein nichtlineare Probleme (4.1) eingegangen werden, die hier der Vollständigkeit halber wiederholt werden:

$$\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) \quad (4.83a)$$

$$\text{u.B.v. } g_i(\mathbf{x}) = 0, \quad i = 1, \dots, p \quad (4.83b)$$

$$h_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, q. \quad (4.83c)$$

4.5.1 Sequentielle quadratische Programmierung

Eine effiziente Methode zur Lösung nichtlinearer Probleme ist die *Sequentielle Quadratische Programmierung*, die eine Folge von linear-quadratischen Approximationen des beschränkten Problems (4.83) löst. Dieses Verfahren greift somit auf Methoden aus der Quadratischen Programmierung zurück und ist für kleine und große Probleme gut geeignet.

4.5.1.1 Allgemeiner Algorithmus

SQP-Verfahren betrachten in jedem Schritt k das am Iterationspunkt $\mathbf{x}^k$ linearisierte Problem⁸

$$\min_{\mathbf{s} \in \mathbb{R}^n} \frac{1}{2} \mathbf{s}^\top \mathbf{H}^k \mathbf{s} + \nabla f(\mathbf{x}^k)^\top \mathbf{s} \quad (4.84a)$$

$$\text{u.B.v. } \nabla g_i(\mathbf{x}^k)^\top \mathbf{s} + g_i(\mathbf{x}^k) = 0, \quad i = 1, \dots, p \quad (4.84b)$$

$$\nabla h_i(\mathbf{x}^k)^\top \mathbf{s} + h_i(\mathbf{x}^k) \leq 0, \quad i = 1, \dots, q \quad (4.84c)$$

wobei $\mathbf{H}^k \approx \nabla^2 f(\mathbf{x}^k)$ eine Approximation der Hessematrix der Kostenfunktion f am Punkt $\mathbf{x}^k$ darstellt. Dieses quadratische Programm kann mit den in Abschnitt 4.4 beschriebenen Verfahren gelöst werden. Der neue Iterationspunkt ergibt sich dabei zu $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k \mathbf{s}^k$, wobei $\mathbf{s}^k$ die Lösung von (4.84) darstellt.

Die Schrittweite α^k muss geeignet gewählt werden, damit der neue Punkt $\mathbf{x}^{k+1}$ einen hinreichenden Abstieg in der Kostenfunktion $f(\mathbf{x}^{k+1}) < f(\mathbf{x}^k)$ bewirkt, andererseits aber auch die Ungleichungsbeschränkungen (4.83c) eingehalten werden. Dieser Punkt wird in Abschnitt 4.5.1.2 näher untersucht. Der grundsätzliche Algorithmus eines SQP-Verfahrens ist in Tabelle 4.3 dargestellt.

Praktische Verfahren zur Lösung von (4.84) können prinzipiell in folgende Klassen unterteilt werden:

- *EQP*-Verfahren (Englisch: *equality-constrained QP*) wählen in jedem Iterationspunkt $\mathbf{x}^k$ eine Arbeitsmenge an aktiven Beschränkungen und reduzieren (4.84) auf

⁸ Die Linearisierung der Kostenfunktion $f(\mathbf{x})$ liefert im Allgemeinen $\frac{1}{2} \mathbf{s}^\top \mathbf{H}^k \mathbf{s} + \nabla f(\mathbf{x}^k)^\top \mathbf{s} + f(\mathbf{x}^k)$. Da der letzte Term $f(\mathbf{x}^k)$ unabhängig ist von $\mathbf{s}$, kann er in (4.84a) vernachlässigt werden.

Initialisierung: $k \leftarrow 0$

Iterationszähler

 $\mathbf{x}^0$

Startpunkt

 $\mathbf{H}^0$

Startapproximation der Hessematrix (pos. def.)

repeat $\mathbf{s}^k \leftarrow$ Lösung von (4.84) $\alpha^k \leftarrow$ z.B. Armijo–Bed. (3.21) Wahl von Schrittweite α^k für Abstieg in Straffkt.
 $P(\mathbf{x}^* + \alpha^k \mathbf{s}^k, \sigma) < P(\mathbf{x}^*, \sigma)$, z.B. mit (4.85) $\mathbf{x}^{k+1} \leftarrow \mathbf{x}^k + \alpha^k \mathbf{s}^k$

nächster Iterationspunkt

 $\mathbf{H}^{k+1} \leftarrow \mathbf{H}^k + \dots$

Quasi-Newton Update der Hessematrix

 $k \leftarrow k + 1$ **until** $\|\nabla_{\mathbf{x}} L(\mathbf{x}^k, \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k)\| \leq \varepsilon$ oder anderes Abbruchkriterium

Tabelle 4.3: Struktur von SQP–Verfahren.

ein gleichungsbeschränktes Problem, wobei alle Beschränkungen, die sich nicht in der Arbeitsmenge befinden, verworfen werden. Die Arbeitsmenge wird in jedem neuen Schritt aktualisiert. Dieses Verfahren hat den Vorteil, dass die Lösung gleichungsbeschränkter quadratischer Probleme für hoch-dimensionale Probleme weniger Rechenaufwand benötigt als die Lösung des allgemeinen Problems (4.84).

- *IQP*–Verfahren (Englisch: *inequality-constrained QP*) verwenden die Menge der aktiven Ungleichungsbeschränkungen von (4.84) als Schätzung der aktiven Beschränkungsmenge $\mathcal{A}(\mathbf{x}^*)$ an der optimalen Lösung $\mathbf{x}^*$. Falls diese Menge $\mathcal{A}(\mathbf{x}^*)$ korrekt bestimmt werden kann, verhält sich das SQP–Verfahren wie ein Newton–Verfahren für gleichungsbeschränkte Probleme und konvergiert schnell zum optimalen Punkt $\mathbf{x}^*$. Dieses Verfahren hat sich als sehr erfolgreich in der Praxis erwiesen. Der Nachteil ist, dass die Lösung von (4.84) bei großen Problemen sehr rechenaufwändig werden kann. Andererseits können in der Nähe des Optimums Informationen aus den vorhergehenden Iterationen verwendet werden (*Warmstart-Strategie*), um eine gute Schätzung für die Lösung in der momentanen Iteration zu erhalten.

4.5.1.2 Wahl der Schrittweite

Wie bereits erwähnt wurde, ist es nicht ausreichend, die Schrittweite α^k so zu wählen, dass der Abstieg in der Kostenfunktion $f(\mathbf{x}^k + \alpha^k \mathbf{s}^k) < f(\mathbf{x}^k)$ hinreichend groß ist, da auch das Einhalten der Beschränkungen $g_i(\mathbf{x}) = 0$ und $h_i(\mathbf{x}) \leq 0$ berücksichtigt werden muss. Im Allgemeinen wird dazu eine *Strafffunktion* angesetzt, die beide Aspekte berücksichtigt. Ein Beispiel dafür ist die sogenannte *l_1 –Strafffunktion*

$$P(\mathbf{x}, \sigma) = f(\mathbf{x}) + \sigma \left[\sum_{i=1}^p |g_i(\mathbf{x})| + \sum_{i=1}^q \max\{0, h_i(\mathbf{x})\} \right], \quad (4.85)$$

die die Eigenschaft besitzt, dass jede Lösung $\mathbf{x}^*$ des nichtlinearen Problems (4.83) ebenfalls ein minimierender Punkt von P ist, unter der Voraussetzung, dass $(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ die

hinreichenden Optimalitätsbedingungen (4.37) erfüllt und der Strafparameter σ die Ungleichungen $\sigma > |\lambda_i^*|$, $i = 1, \dots, p$ und $\sigma > |\mu_i^*|$, $i = 1, \dots, q$ erfüllt. Man spricht in diesem Zusammenhang von einer *exakten Straffunktion*.

Die Schrittweite α^k kann nun mit einem der in Abschnitt 3.3.2 vorgestellten Liniensuchverfahren geeignet bestimmt werden (z.B. mittels der Armijo-Bedingung (3.21)), um einen hinreichenden Abstieg in der Straffunktion $P(\mathbf{x}^k + \alpha^k \mathbf{s}^k, \sigma) < P(\mathbf{x}^k, \sigma)$ zu erreichen.

4.5.1.3 Beispiel: Optimale Auslegung von Reglerparametern

Als Beispiel soll die Optimierung von Reglerparametern für ein totzeitbehaftetes zeitdiskretes Streckenmodell

$$x(k) = ax(k-1) + bu(k-10) \quad (4.86)$$

mit den Streckenparametern $a = b = 0.5$ betrachtet werden [2]. Das Regelgesetz sei gegeben durch

$$u(k) = c_1 u(k-1) + c_2 e(k) + c_3 e(k-1), \quad (4.87)$$

wobei $e(k) = 1 - x(k)$ den Regelfehler auf einen Einheitssprung darstellt. Die Reglerparameter sollen so eingestellt werden, dass die (zeitdiskrete) *betragslineare Regelfläche*

$$f(\mathbf{c}) = \sum_{k=1}^N |e(k)| \quad (4.88)$$

bzw. die *zeitgewichtete betragslineare Regelfläche*

$$f(\mathbf{c}) = \sum_{k=1}^N \frac{k}{N} |e(k)| \quad (4.89)$$

minimiert wird, bei der zeitlich spätere Abweichungen stärker ins Gewicht fallen. Darüber hinaus soll das Überschwingen von $x(k)$ begrenzt werden.

Die Optimization Toolbox von MATLAB stellt zur Lösung von nichtlinearen Optimierungsproblemen die Funktion `fmincon` zur Verfügung. Das SQP-Verfahren kann mit Hilfe der Option `optimset('LargeScale','off')` ausgewählt werden.

Der MATLAB-Code zur Bestimmung der optimalen Reglerparameter mittels `fmincon` ist in Abbildung 4.14 zusammengefasst.⁹ Dabei wird die Begrenzung des Überschwingens als nichtlineare Beschränkung implementiert. Als Startwert für die Reglerparameter wurde $\mathbf{c}^0 = [1, 0.5, -0.5]^T$ gewählt.

Die optimierten Sprungantworten für die Fälle (4.88) und (4.89) mit jeweils 10% und 0% Überschwingen und die dazugehörigen Reglerparameter sind in den Abbildungen 4.15, 4.16 und in Tabelle 4.4 dargestellt.

In allen Fällen liefert die Optimierung den Wert $c_1 = 1$ zurück. Der Regler (4.87) besitzt somit integrierendes Verhalten. Des Weiteren ist in Abbildung 4.16 zu erkennen, dass die zeitliche Gewichtung in der Kostenfunktion (4.89) eine Möglichkeit darstellt, um das Schwingungsverhalten nach Erreichen des Sollwertes zu dämpfen.

⁹Der geschlossene Regelkreis (4.86), (4.87) ist als z -Übertragungsfunktion implementiert. Die Sprungantwort wird mit Hilfe der Funktion `step` berechnet.

```

function reglerparameter(caseQ)
% -----
global x cold
switch caseQ
    case 1, p.costQ=1; p.ue=0.1; % betragslineare Regelfläche, max. Überschw. 10%
    case 2, p.costQ=1; p.ue=0; % betragslineare Regelfläche, max. Überschw. 0%
    case 3, p.costQ=2; p.ue=0.1; % zeitgew. betragslin. Regelfläche, max. Überschw. 10%
    case 4, p.costQ=2; p.ue=0; % zeitgew. betragslin. Regelfläche, max. Überschw. 0%
end

p.N = 70; % Anzahl der Abtastschritte
p.a = 0.5; p.b = 0.5; % Streckenparameter
c0 = [1;0.5;-0.5]; % Startwert

opt = optimset('LargeScale','off', ... % SQP-Verfahren für fmincon
               'Display','iter','MaxFunEvals',500);
[c,f] = fmincon(@kostenfkt,c0,[],[],[],[],[],[],@beschr,opt,p);
fprintf('\nOptimale Parameter c=[%2.3g,%2.3g,%2.3g] mit f(c)=%2.3g\n',c,f);

x = sprungantwort(c,p); % optimierte Sprungantwort

figure(1); clf % Darstellung der Sprungantwort
h(1) = stairs(ones(1,p.N),'r--'); hold on
h(2) = stairs(x,'k');
h(3) = stairs(1-x,'m'); xlabel('Zeitschritte');
axis([0, p.N, -0.1, 1.1+p.ue]);
legend(h,'Sollwert','Zustand x','Regelabw. e',4)

function x = sprungantwort(c,p)
% -----
z = tf('z'); % z-Transformation:
G0 = (c(2)+c(3)*z^(-1))/(1-c(1)*z^(-1)) * p.b/(z^10-p.a*z^9); % Gregler * Gstrecke
x = step( G0/(1+G0), p.N ); % Sprungantwort

function f = kostenfkt(c,p)
% -----
global x cold
x = sprungantwort(c,p); % Sprungantwort
if p.costQ==1, f = sum( abs(1-x) ); % Kostenfunktion
else f = 1/p.N*sum( [1:p.N]*abs(1-x) ); end
cold = c;

function [C,Ceq] = beschr(c,p)
% -----
global x cold
if any(c~=cold), x=sprungantwort(c,p); end % falls neue c-Werte vorliegen
C = max(x-(1+p.ue)); Ceq = []; % nichtlin. Ungleichungsbeschr.

```

Abbildung 4.14: MATLAB-Code zur Bestimmung der optimalen Reglerparameter (fmincon).

Kostenfkt.	max. Überschw.	c_1^*	c_2^*	c_3^*	$f(c^*)$
(4.88)	10%	1.000	0.438	-0.368	14.6
(4.88)	0%	1.000	0.454	-0.388	14.9
(4.89)	10%	1.000	0.367	-0.297	1.93
(4.89)	0%	1.000	0.400	-0.333	2.06

Tabelle 4.4: Optimale Reglerparameter für den Regelkreis (4.86)–(4.87).

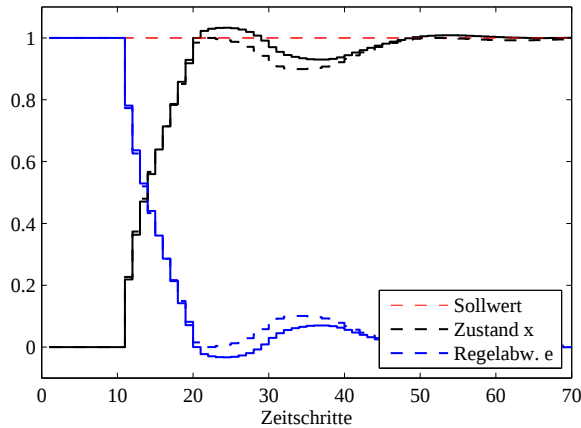


Abbildung 4.15: Sprungantwort des optimierten Regelkreises mit Kostenfunktion (4.88) (Überschwingen 10%: —, 0%: - -).

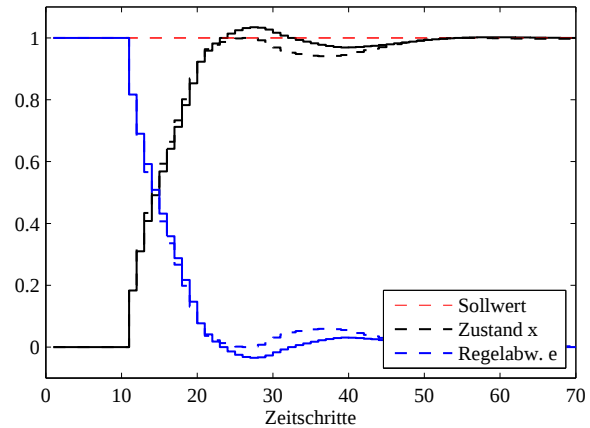


Abbildung 4.16: Sprungantwort des optimierten Regelkreises mit Kostenfunktion (4.89) (Überschwingen 10%: —, 0%: - -).

4.5.2 Weitere numerische Verfahren

Ein grundsätzlich anderes Vorgehen als bei SQP-Verfahren besteht darin, das ursprüngliche Problem (4.83) durch ein oder mehrere unbeschränkte Probleme zu ersetzen, die sukzessive numerisch gelöst werden. Im Folgenden wird nur kurz auf diese Verfahren eingegangen. Für weitere Informationen sei auf die am Ende des Kapitels angegebenen Bücher verwiesen.

Methode der äußeren Straffunktionen

Im einfachsten Fall kann das ursprüngliche beschränkte Problem (4.83) durch ein neues unbeschränktes Problem ersetzt werden, indem die Beschränkungen (4.83b) und (4.83c) als Strafterme zur Kostenfunktion addiert werden:

$$\min_{\mathbf{x} \in \mathbb{R}^n} P(\mathbf{x}, \tau) = f(\mathbf{x}) + \frac{1}{\tau} \left[\sum_{i=1}^p \phi(g_i(\mathbf{x})) + \sum_{i=1}^q \psi(h_i(\mathbf{x})) \right]. \quad (4.90)$$

Die Funktionen ϕ und ψ stellen *äußere Straffunktionen* dar, die Beschränkungen von außen bestrafen und somit ein Verletzen der Beschränkungsgrenzen erlauben.

Die Straffunktion ϕ für die Gleichungsbeschränkungen $g_i(\mathbf{x}) = 0$ wird gewöhnlich als Potenzfunktion $\phi(g_i(\mathbf{x})) = |g_i(\mathbf{x})|^r$ mit $r \geq 1$ angesetzt, während ψ typischerweise wie folgt definiert ist

$$\psi(h_i(\mathbf{x})) = (\max\{0, h_i(\mathbf{x})\})^r, \quad r \geq 1. \quad (4.91)$$

Abbildung 4.17a verdeutlicht das Verhalten von ψ und den Einfluss des Strafparameters τ .

Die Lösung $\mathbf{x}(\tau)$ des neuen unbeschränkten Problems (4.90) wird in der Regel von der optimalen Lösung $\mathbf{x}^*$ des Originalproblems (4.83) abweichen, da der Strafterm in (4.90) die ursprüngliche Kostenfunktion (4.83a) verfälscht.

In der Praxis wird daher das Problem (4.90) sukzessive für eine abnehmende Sequenz $\{\tau^k\}$ von Strafparametern gelöst, was einer zunehmenden Bestrafung der Beschränkungsverletzung entspricht (Abbildung 4.17a). Unter gewissen Bedingungen kann gezeigt werden,

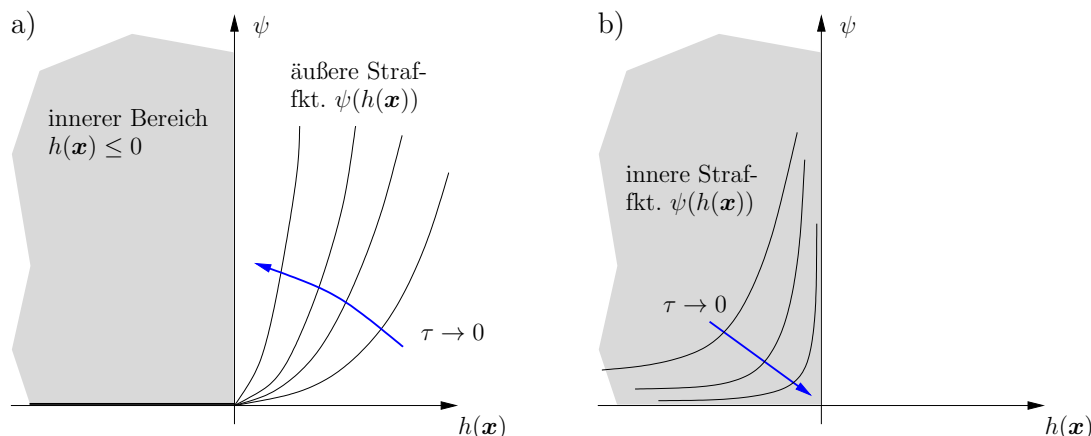


Abbildung 4.17: Veranschaulichung von äußeren und inneren Straffunktionen $\psi(h(\mathbf{x}))$ für eine Ungleichungsbeschränkung $h(\mathbf{x}) \leq 0$.

dass sich für eine Sequenz $\{\tau^k\}$ mit $\lim_{k \rightarrow \infty} \tau^k = 0$ die optimale Lösung im Limit ergibt, d.h.

$$\lim_{k \rightarrow \infty} \mathbf{x}(\tau^k) = \mathbf{x}^*. \quad (4.92)$$

Ein bekannter Nachteil des Verfahrens ist allerdings, dass das Problem (4.90) für $\tau \rightarrow 0$ zunehmend schlecht konditioniert wird, was eine genaue Approximation der optimalen Lösung $\mathbf{x}^*$ in der Praxis schwierig macht.

Methode der inneren Straffunktionen (Barrieremethode)

Innere Straffunktionen oder *Barrierefunktionen* bestrafen die Annäherung an die Beschränkungsgrenze und gehen gegen Unendlich, wenn die Grenze $h_i(\mathbf{x}) = 0$ berührt wird, wie in den Abbildungen 4.17b und 4.8 zu erkennen ist. Beispiele von Barrierefunktionen sind

$$\psi(h_i(\mathbf{x})) = \frac{-1}{h_i(\mathbf{x})} \quad \text{oder} \quad \psi(h_i(\mathbf{x})) = -\ln(-h_i(\mathbf{x})). \quad (4.93)$$

Da Barrierefunktionen nur im Inneren des zulässigen Bereiches definiert sind, können nur Ungleichungsbeschränkungen (4.83) berücksichtigt werden:

$$\min_{\mathbf{x} \in \mathbb{R}^n} P(\mathbf{x}, \tau) = f(\mathbf{x}) + \tau \sum_{i=1}^q \psi(h_i(\mathbf{x})). \quad (4.94)$$

Durch sukzessive Reduktion des Straffparameters τ reduziert sich die Bestrafung des zulässigen Bereiches für $h_i(\mathbf{x}) < 0$, während $\psi \rightarrow \infty$ bei Berührung der Beschränkungsgrenze $h_i(\mathbf{x}) \rightarrow 0$ erhalten bleibt (Abbildung 4.17b). Wie im Fall der äußeren Straffunktionen kann (unter gewissen Umständen) gezeigt werden, dass für eine abnehmende Sequenz $\{\tau^k\}$ mit $\lim_{k \rightarrow \infty} \tau^k = 0$ die optimale Lösung $\mathbf{x}^*$ im Limit (4.92) erreicht wird.

Bei Barrieremethoden stellt sich insbesondere das Problem, eine zulässige Startlösung $\mathbf{x}^0$ zu bestimmen, die innerhalb der Schranken $h_i(\mathbf{x}) \leq 0$ liegt.

Interior-Point-Verfahren

Interior-Point-Verfahren sind bereits im Rahmen der Linearen Programmierung in Abschnitt 4.3.4 vorgestellt worden. Ihre Erweiterung auf nichtlineare Probleme soll deshalb nur kurz angerissen werden.

Interior-Point-Verfahren basieren auf der Idee, die strikte Komplementaritätsbedingung (4.29e) durch einen Parameter τ zu modifizieren, d.h. $\mu_i^* h_i(\mathbf{x}^*) = \tau$, um die numerische Lösung der KKT-Bedingungen (4.29) zu erleichtern. Wie bereits in Abschnitt 4.3.4.2 gezeigt wurde, besteht ein Zusammenhang zwischen Interior-Point-Verfahren und der logarithmischen Barrierefunktion in (4.93). Allerdings sind Interior-Point-Verfahren numerisch besser konditioniert und gehören zu den leistungsfähigsten Verfahren der nichtlinearen Optimierung.

Für weitergehende Informationen sei insbesondere auf den sehr guten Übersichtsaufsatz [3] und die Bücher [4, 1] verwiesen.

4.6 Software-Übersicht

Im Folgenden ist eine Auswahl an Software zur Lösung von statischen Optimierungsproblemen zusammengestellt.

Lineare Optimierung

- linprog: MATLAB Optimization Toolbox (kostenpflichtig)
- CPLEX (kostenpflichtig)
<http://www.ilog.com/products/cplex>
- GLPK: “GNU Linear Programming Kit” (frei zugänglich)
<http://www.gnu.org/software/glpk>
- lp_solve: Mixed-Integer Lineare Optimierung (frei zugänglich)
<http://lpsolve.sourceforge.net>

Quadratische Optimierung

- quadprog: MATLAB Optimization Toolbox (kostenpflichtig)
- CPLEX (kostenpflichtig)
<http://www.ilog.com/products/cplex>
- OOQP (frei zugänglich)
<http://pages.cs.wisc.edu/~swright/ooqp>
- LOQO (kostenpflichtig)
<http://www.princeton.edu/~rvdb>

Nichtlineare Optimierung

- fmincon: MATLAB Optimization Toolbox (kostenpflichtig)
- LOQO (kostenpflichtig)
<http://www.princeton.edu/~rvdb>
- MINOS (kostenpflichtig)
http://www.sbsi-sol-optimize.com/asp/sol_product_minos.htm
- SNOPT (kostenpflichtig, aber Studentenversion frei zugänglich)
<http://www.scicomp.ucsd.edu/~peg>
- DONLP2 (frei zugänglich)
<ftp://ftp.mathematik.tu-darmstadt.de/pub/departement/software/opti>
- IPOPT (frei zugänglich)
<https://projects.coin-or.org/Ipopt>

Modellierungssprachen

Die meisten der oben angegebenen Optimierer unterstützen eine Anbindung an MATLAB (z.B. `lp_solve`, SNOPT, DONLP2, IPOPT) oder an eine der Modellierungssprachen AMPL (z.B. LOQO, GLPK, IPOPT) oder GAMS (z.B. MINOS). Diese Sprachen bieten eine symbolorientierte Syntax zum Formulieren von Optimierungsproblemen:

- AMPL: “A Mathematical Programming Language”
<http://www.ampl.com>
- GAMS: “General Algebraic Modeling System”
<http://www.gams.com>

Literatur

- [1] J. Nocedal und S.J. Wright. *Numerical Optimization*. Springer, 2006.
- [2] M. Papageorgiou. *Optimierung*. Oldenbourg Verlag, 1991.
- [3] M.H. Wright. Interior methods for constrained optimization. *Acta Numerica*, 1:341–407, 1992. <http://eprints.kfupm.edu.sa/46694/>.
- [4] S.J. Wright. *Primal–Dual Interior–Point Methods*. Society for Industrial and Applied Mathematics (SIAM), 1997.

Kapitel 5

Dynamische Optimierung

In den bisherigen Kapiteln ging es um die Minimierung einer Funktion bzgl. eines Parametervektors $\mathbf{x}$ im Euklidischen Raum $\mathbb{R}^n$. Im Folgenden sollen nun dynamische Optimierungsprobleme betrachtet werden, bei denen eine Funktion einer unabhängigen Variablen (in der Regel eine Zeitfunktion) gesucht wird, die ein *Kostenfunktional* minimiert. Die häufigste Anwendung findet man bei dynamischen Systemen für die eine optimale Eingangstrajektorie $u^*(t)$ gesucht wird. Man spricht in diesem Zusammenhang auch von einem *Optimalsteuerungsproblem*.

5.1 Allgemeine Aufgabenstellung

Eine recht allgemeine Struktur von Optimalsteuerungsproblemen lautet

$$\min_{\mathbf{u}(\cdot)} \quad J(\mathbf{u}) = \varphi(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} l(\mathbf{x}(t), \mathbf{u}(t), t) dt \quad (5.1a)$$

$$\text{u.B.v.} \quad \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t), \quad \mathbf{x}(t_0) = \mathbf{x}_0 \quad (5.1b)$$

$$\mathbf{g}(\mathbf{x}(t_f), t_f) = \mathbf{0} \quad (5.1c)$$

$$\mathbf{h}(\mathbf{x}(t), \mathbf{u}(t), t) \leq \mathbf{0} \quad \forall t \in [t_0, t_f]. \quad (5.1d)$$

Die Endbedingungen (5.1c), die Art des Kostenfunktionals (5.1a), die Beschränkungen (5.1d) als auch die Endzeit t_f haben Einfluss auf die Charakteristik des Optimierungsproblems. Im Folgenden wird stets angenommen, dass alle auftretenden Funktionen stetig differenzierbar in ihren Argumenten sind.

Endbedingungen (5.1c)

Die Gleichungen (5.1b)–(5.1c) für sich genommen stellen ein *Steuerungsproblem* dar, da für ein (im Allgemeinen) nichtlineares System mit Zustand $\mathbf{x} \in \mathbb{R}^n$ und Eingang $\mathbf{u} \in \mathbb{R}^m$ eine Steuertrajektorie $\mathbf{u}(t)$, $t \in [t_0, t_f]$ gesucht wird, die das System vom Anfangszustand $\mathbf{x}(t_0) = \mathbf{x}_0$ zu einer gewünschten Endfunktion (5.1c) zur Endzeit t_f überführt. Die allgemeine Form (5.1c) von Endbedingungen kommt in der Praxis eher selten vor.¹ Wesentlich

¹Ein Beispiel einer nichtlinearen Endbedingung ist die Optimierung der Flugbahn einer Trägerrakete, die einen Satelliten auf eine geostationären Umlaufbahn um die Erde bringen soll.

häufiger sind partielle Endbedingungen der Form

$$x_i(t_f) = x_{f,i}, \quad i \in \mathcal{I}_f. \quad (5.2)$$

Die Menge $\mathcal{I}_f$ umfasst die Indices der bei t_f fixierten Zustände x_i . Wenn der vollständige Endzustand vorgegeben ist, d.h. $\mathbf{x}(t_f) = \mathbf{x}_f$, so gilt $\mathcal{I}_f = \{1, \dots, n\}$. Der andere Grenzfall liegt vor, wenn $\mathcal{I}_f = \{\}$ ist. Man spricht in diesem Fall von einem *freien Endzustand*.

Im Folgenden werden vorrangig die partiellen Endbedingungen (5.2) betrachtet. Auf die allgemeinen Endbedingungen (5.1c) wird in Abschnitt 5.3.6 eingegangen.

Kostenfunktional (5.1a)

Aus den in der Regel unendlich vielen Steuerfunktionen $\mathbf{u}(t)$, die das System (5.1b) mit gegebenen Anfangsbedingungen zu (5.1c) bzw. (5.2) überführen, wird eine *optimale Steuerung* $\mathbf{u}^*(t)$ gesucht, die das Kostenfunktional (5.1a) minimiert. Bei der allgemeinen Form (5.1a) wird zwischen der *Endzustandsbewertung* bzw. dem *Mayer-Term* $\varphi(\mathbf{x}(t_f), t_f)$ und dem *Integralanteil* $l(\mathbf{x}, \mathbf{u}, t)$ unterschieden. Generell kann das Kostenfunktional somit folgende Formen aufweisen

$$J(\mathbf{u}) = \begin{cases} \varphi(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} l(\mathbf{x}(t), \mathbf{u}(t), t) dt & \text{Bolza-Form} \\ \int_{t_0}^{t_f} l(\mathbf{x}(t), \mathbf{u}(t), t) dt & \text{Lagrange-Form} \\ \varphi(\mathbf{x}(t_f), t_f) & \text{Mayer-Form.} \end{cases} \quad (5.3)$$

Die Bolza- und Lagrange-Darstellungen können stets in die Mayer-Form gebracht werden, indem der Integralanteil als neuer Zustand

$$\dot{x}_{n+1} = l(\mathbf{x}, \mathbf{u}, t), \quad x_{n+1}(t_0) = 0 \quad (5.4)$$

eingeführt und $x_{n+1}(t_f)$ zu $\varphi(\mathbf{x}(t_f), t_f)$ hinzugefügt wird. Die wichtigsten Beispiele für ein Integralkriterium sind

$J(\mathbf{u}) = \frac{1}{2} \int_{t_0}^{t_f} \mathbf{x}^\top(t) \mathbf{Q} \mathbf{x}(t) dt$	quadratische Zustandsgewichtung ($\mathbf{Q}$ symmetrisch & $\mathbf{Q} > \mathbf{0}$)
$J(\mathbf{u}) = \frac{1}{2} \int_{t_0}^{t_f} \mathbf{u}^\top(t) \mathbf{R} \mathbf{u}(t) dt$	energieoptimales Kriterium ($\mathbf{R}$ symmetrisch & $\mathbf{R} > \mathbf{0}$)
$J(\mathbf{u}) = \frac{1}{2} \int_{t_0}^{t_f} \mathbf{x}^\top(t) \mathbf{Q} \mathbf{x}(t) + \mathbf{u}^\top(t) \mathbf{R} \mathbf{u}(t) dt$	allgemeines quadratisches Kriterium ($\mathbf{Q}, \mathbf{R}$ symmetrisch & $\mathbf{Q} \geq \mathbf{0}, \mathbf{R} > \mathbf{0}$)
$J(\mathbf{u}) = \int_{t_0}^{t_f} 1 dt = t_f - t_0$	zeitoptimales Kriterium
$J(\mathbf{u}) = \int_{t_0}^{t_f} 1 + \mathbf{u}^\top(t) \mathbf{R} \mathbf{u}(t) dt$	Kombination Zeit-/Energieoptimalität

Im Zusammenhang mit quadratischen Termen der Form $\mathbf{u}^\top \mathbf{R} \mathbf{u}$ wird gewöhnlich der Begriff *Energieoptimalität* verwendet, da der zu minimierende Ausdruck häufig im Zusammenhang mit der Energie eines elektrischen Signals steht.

Beispiel 5.1 In Kapitel 1 wurden unterschiedliche Kostenfunktionale betrachtet. Im Falle der Goddard-Rakete (Beispiel 1.4) stellt $\varphi(\mathbf{x}(t_f), t_f) = -h(t_f)$ die Maximierung der Flughöhe dar, während im Konsumenten-Beispiel 1.5 $\varphi(\mathbf{x}(t_f), t_f) = -K^\kappa(t_f)$ das Vererbungskapital des Konsumenten berücksichtigt. Die Kombination aus zeit- und energieoptimalen Anteilen im Integralkriterium wurde beim inversen Pendel in Beispiel 1.3 betrachtet.

Endzeit t_f

Die Endzeit (oder Optimierungshorizont) t_f kann entweder vorgegeben oder frei sein. Falls die Endzeit frei ist, muss t_f als Teil der Lösung des Optimierungsproblems (5.1) bestimmt werden. Die Endzeit t_f spielt in diesem Zusammenhang eine wichtige Rolle und entscheidet oft, ob ein Problem lösbar ist oder nicht.

Beschränkungen (5.1d)

Beschränkungen sind in vielen praktischen Problemen vorhanden. Die nichtlineare Form (5.1d) ist eine sehr allgemeine Schreibweise, die in den meisten Fällen auf Beschränkungen der Eingänge $\mathbf{u}$

$$\mathbf{u}(t) \in U \subseteq \mathbb{R}^m \quad \forall t \in [t_0, t_f] \quad (5.5)$$

und/oder der Zustände

$$\mathbf{x}(t) \in X \subseteq \mathbb{R}^n \quad \forall t \in [t_0, t_f] \quad (5.6)$$

reduziert werden kann. Der in der Praxis am häufigsten auftretende Fall sind Eingangsbeschränkungen, wie z.B. Spannungsbegrenzungen oder Maximalkräfte eines Aktors.

Der Inhalt dieses Kapitels beschränkt sich größtenteils auf die Betrachtung der Eingangsbeschränkungen (5.5). Die strenge mathematische Berücksichtigung von Zustandsbeschränkungen (5.6) bzw. von allgemeinen Beschränkungen (5.1d) ist erheblich schwieriger als der eingangsbeschränkte Fall und wird in Abschnitt 5.5.5 lediglich angedeutet.

5.2 Einführung in die Variationsrechnung

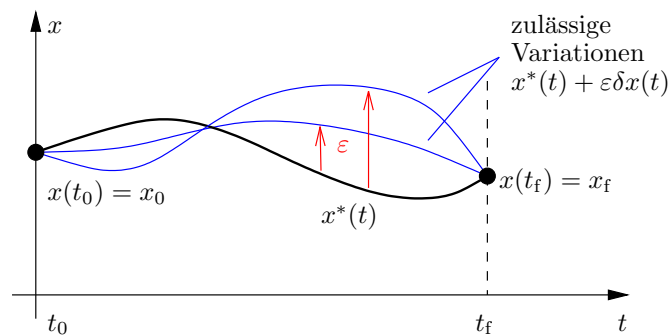
Die Betrachtung von dynamischen Optimierungsproblemen ist naturgemäß komplexer im Vergleich zu statischen Problemen, bei denen die Optimierungsvariablen von endlicher Anzahl im $\mathbb{R}^n$ waren. Eine elegante Methodik, mit der dynamische Optimierungsprobleme angegangen werden können, ist die *Variationsrechnung*, die Ende des 17. und Anfang des 18. Jahrhunderts entwickelt wurde. Die bekanntesten Namen, die mit der Variationsrechnung in Verbindung gebracht werden, sind *Bernoulli*, *Euler* und *Lagrange*.

Um eine Einführung in die Variationsrechnung zu geben, werden zunächst allgemeine Funktionale betrachtet, bevor die Ergebnisse auf unbeschränkte Optimalsteuerungsprobleme erweitert werden (Abschnitt 5.3).

5.2.1 Euler–Lagrange–Gleichungen

Betrachtet wird ein Funktional der Form

$$J(\mathbf{x}) = \int_{t_0}^{t_f} \phi(\mathbf{x}(t), \dot{\mathbf{x}}(t), t) dt. \quad (5.7)$$

Abbildung 5.1: Zulässige Variation um eine optimale Lösung $x^*(t)$.

Die Endzeit t_f ist fixiert, ebenso wie die Anfangs- und Endbedingungen

$$\mathbf{x}(t_0) = \mathbf{x}_0, \quad \mathbf{x}(t_f) = \mathbf{x}_f. \quad (5.8)$$

Es wird angenommen, dass eine optimale Lösung $\mathbf{x}^*(t) \in \mathcal{C}^2$ (d.h. zweimal stetig differenzierbar) existiert, die das Funktional (5.7) minimiert.

Bei der Variationsrechnung wird ausgehend von einer optimalen Lösung $\mathbf{x}^*(t)$, $t \in [t_0, t_f]$ eine kleine Variation $\delta\mathbf{x}(t)$ betrachtet und die entsprechende Variation δJ des Kostenfunktional untersucht. Zunächst wird die nichtoptimale Zeitfunktion

$$\mathbf{x}(t) = \mathbf{x}^*(t) + \varepsilon\delta\mathbf{x}(t), \quad t \in [t_0, t_f] \quad (5.9)$$

mit dem Parameter ε angesetzt, wobei $\delta\mathbf{x}(t)$ eine beliebige *zulässige Variation* mit homogenen Randbedingungen $\delta\mathbf{x}(t_0) = \mathbf{0}$ und $\delta\mathbf{x}(t_f) = \mathbf{0}$ darstellt, so dass die variierte Funktion (5.9) die gegebenen Randbedingungen (5.8) erfüllt. Abbildung 5.1 stellt ein Beispiel von $\delta\mathbf{x}(t)$ und den Einfluss des Parameters ε im eindimensionalen Fall dar.

Die variierte Größe $\mathbf{x}(t)$ und deren Zeitableitung $\dot{\mathbf{x}}(t)$ werden jetzt in das Kostenfunktional (5.7) eingesetzt

$$J(\mathbf{x}^* + \varepsilon\delta\mathbf{x}) = \int_{t_0}^{t_f} \phi(\mathbf{x}^*(t) + \varepsilon\delta\mathbf{x}(t), \dot{\mathbf{x}}^*(t) + \varepsilon\delta\dot{\mathbf{x}}(t), t) dt. \quad (5.10)$$

Betrachtet man die Funktionen $\delta\mathbf{x}(t)$ und $\delta\dot{\mathbf{x}}(t)$ als festgelegt, so hängt J in (5.10) allein von ε ab. Aus der Stationaritätsbedingung für gewöhnliche Extrema (siehe Kapitel 3.1) folgt, dass eine notwendige Bedingung für ein (lokales) Extremum das Verschwinden der ersten Ableitung ist, also $dJ(\mathbf{x}^* + \varepsilon\delta\mathbf{x})/d\varepsilon = 0$ gelten muss.

Im hier betrachteten Fall ist aber bekannt, dass – unabhängig von der Wahl von $\delta\mathbf{x}(t)$ – diese Nullstelle der Ableitung bei $\varepsilon = 0$ liegen muss, da $\mathbf{x}^*(t)$ laut Definition eine optimale Lösung darstellt, so dass die rechte Seite von (5.10) ihr Minimum für $\varepsilon = 0$ annimmt. Es muss also gelten

$$\delta J = \left. \frac{dJ(\mathbf{x}^* + \varepsilon\delta\mathbf{x})}{d\varepsilon} \right|_{\varepsilon=0} = 0, \quad (5.11)$$

d.h. die *erste Variation von J* muss Null sein. Explizites Differenzieren von (5.10) nach ε und anschließendes Nullsetzen von ε ergibt

$$\delta J = \int_{t_0}^{t_f} \left(\frac{\partial \phi}{\partial \mathbf{x}} \right)^\top \delta\mathbf{x} + \left(\frac{\partial \phi}{\partial \dot{\mathbf{x}}} \right)^\top \delta\dot{\mathbf{x}} dt, \quad (5.12)$$

wobei die optimalen Größen $\mathbf{x}^*(t)$ und $\dot{\mathbf{x}}^*(t)$ als Argumente der partiellen Ableitungen der Einfachheit halber weggelassen wurden.

Aus (5.12) ist ersichtlich, dass sich die Variation δJ des Kostenfunktional in Abhängigkeit der Variationen $\delta \mathbf{x}(t)$ und $\delta \dot{\mathbf{x}}(t)$ ausdrücken lässt. Da $\delta \dot{\mathbf{x}}(t)$ als Zeitableitung von $\delta \mathbf{x}(t)$ aber nicht unabhängig von $\delta \mathbf{x}(t)$ variiert werden kann, wird der zweite Term in (5.12) partiell integriert²

$$\int_{t_0}^{t_f} \left(\frac{\partial \phi}{\partial \dot{\mathbf{x}}} \right)^T \delta \dot{\mathbf{x}} dt = \left[\left(\frac{\partial \phi}{\partial \dot{\mathbf{x}}} \right)^T \delta \mathbf{x} \right]_{t_0}^{t_f} - \int_{t_0}^{t_f} \frac{d}{dt} \left(\frac{\partial \phi}{\partial \dot{\mathbf{x}}} \right)^T \delta \mathbf{x} dt. \quad (5.13)$$

Da nur zulässige Variationen mit $\delta \mathbf{x}(t_0) = \delta \mathbf{x}(t_f) = \mathbf{0}$ betrachtet werden, verschwinden die Randterme in (5.13) und für δJ in (5.12) ergibt sich

$$\delta J = \int_{t_0}^{t_f} \left[\frac{\partial \phi}{\partial \mathbf{x}} - \frac{d}{dt} \left(\frac{\partial \phi}{\partial \dot{\mathbf{x}}} \right) \right]^T \delta \mathbf{x} dt \quad (5.14)$$

in Abhängigkeit von $\delta \mathbf{x}(t)$. An dieser Stelle kommt das *Fundamentallemma der Variationsrechnung* ins Spiel.³

Satz 5.1 (Fundamentallemma der Variationsrechnung) Sei $\mathbf{h} : [t_0, t_f] \rightarrow \mathbb{R}^n$ eine stetige Funktion. Falls für alle stetigen Funktionen $\delta \mathbf{x} : [t_0, t_f] \rightarrow \mathbb{R}^n$ die Bedingung

$$\int_{t_0}^{t_f} \mathbf{h}(t)^T \delta \mathbf{x}(t) dt = 0 \quad (5.15)$$

gilt, so ist $\mathbf{h}(t) = \mathbf{0}$ für alle $t \in [t_0, t_f]$.

Gemäß (5.11) muss die Variation δJ für alle zulässigen Variationen $\delta \mathbf{x}(t)$ Null sein, damit $\mathbf{x}^*(t)$ tatsächlich eine optimale Lösung darstellt. Somit folgt mit Hilfe des Fundamentallemmas der Variationsrechnung aus (5.14)

$$\boxed{\frac{\partial \phi}{\partial \mathbf{x}} - \frac{d}{dt} \left(\frac{\partial \phi}{\partial \dot{\mathbf{x}}} \right) = \mathbf{0}.} \quad (5.16)$$

Die Differentialgleichung (5.16) wird als *Euler–Lagrange–Gleichung* bezeichnet und stellt eine *notwendige Bedingung* für eine optimale Funktion $\mathbf{x}^*(t)$ dar, die das Funktional (5.7) unter Berücksichtigung der Randbedingungen (5.8) minimiert.

Über die *zweite Variation* von J

$$\delta^2 J = \frac{d^2 J(\mathbf{x}^* + \varepsilon \delta \mathbf{x})}{d\varepsilon^2} \Big|_{\varepsilon=0} \geq 0 \quad (5.17)$$

kann gezeigt werden, dass die Hessematrix

$$\frac{\partial^2 \phi}{\partial \dot{\mathbf{x}}^2}(\mathbf{x}^*(t), \dot{\mathbf{x}}^*(t), t) \quad \text{positiv semi-definit} \quad \forall t \in [t_0, t_f] \quad (5.18)$$

² Zur Erinnerung: $\int \mathbf{u}^T \dot{\mathbf{v}} dt = \mathbf{u}^T \mathbf{v} - \int \dot{\mathbf{u}}^T \mathbf{v} dt$

³ Der Beweis des Fundamentallemmas kann z.B. in [1] nachgelesen werden.

sein muss. Diese Bedingung ist als *Legendre-Bedingung* bekannt und stellt eine notwendige Bedingung 2. Ordnung für eine optimale Lösung $\mathbf{x}^*(t)$, $t \in [t_0, t_f]$ zusätzlich zur Euler-Lagrange-Gleichung (5.16) dar. Die Herleitung von (5.18) erfolgt analog zum Vorgehen bei der ersten Variation (5.11) mittels partieller Integration und wird an dieser Stelle nicht weiter vertieft.

Zwei Sonderfälle der Euler-Lagrange-Gleichung (5.16) sind

- ϕ hängt nicht explizit von $\mathbf{x}$ ab, d.h. $\phi = \phi(\dot{\mathbf{x}}, t)$. In diesem Fall liefert eine erste Integration von (5.16)

$$\frac{\partial \phi}{\partial \dot{\mathbf{x}}} = \text{konst.} \quad (5.19)$$

- ϕ hängt nicht explizit von t ab, d.h. $\phi = \phi(\mathbf{x}, \dot{\mathbf{x}})$. In diesem Fall kann gezeigt werden [10], dass sich die Euler-Lagrange-Gleichung (5.16) wie folgt vereinfacht:

$$\phi - \dot{\mathbf{x}}^\top \frac{\partial \phi}{\partial \dot{\mathbf{x}}} = \text{konst.} \quad (5.20)$$

Aufgabe 5.1 Bestimmen Sie den minimalen Abstand zwischen zwei Punkten A und B in der (x, y) -Ebene. Stellen Sie dazu das zu minimierende Kostenfunktional in Abhängigkeit der Bogenlänge s auf und verwenden Sie die Relation

$$ds = \sqrt{dx^2 + dy^2} = \sqrt{1 + y'(x)^2} dx. \quad (5.21)$$

5.2.2 Beispiel: Galileo's hängende Kette

Galileo formulierte im Jahr 1638 die Frage, welche Form eine an beiden Enden aufgehängte Kette im Schwerfeld der Erde besitzt. Galileo nahm dabei fälschlicherweise an, dass die Kette eine Parabel beschreiben würde. Das Problem wurde um 1690 von *Jakob und Johannes Bernoulli* und von *Leibnitz* mit Hilfe der Variationsrechnung gelöst.

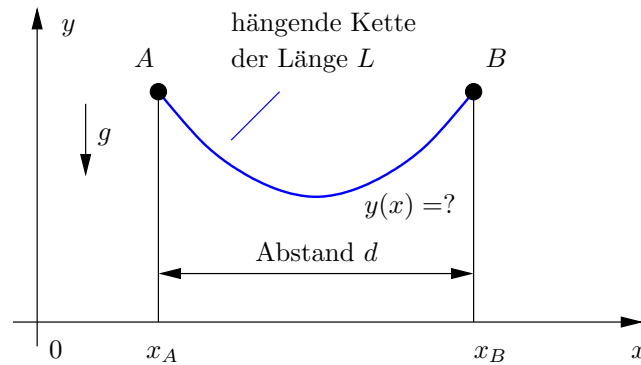
Betrachtet wird eine Kette der Länge L , deren Enden in den Punkten A und B aufgehängt sind, siehe Abbildung 5.2. Aus dem Hamilton-Prinzip in der Technischen Mechanik folgt, dass die potentielle Energie der Kette im statischen Gleichgewicht minimal sein muss. Mit Hilfe der Bogenlänge s und der Relation (5.21) ergibt sich die potentielle Energie zu

$$V = \rho g \int_0^L y ds = \rho g \int_{x_A}^{x_B} y \sqrt{1 + y'(x)^2} dx, \quad (5.22)$$

wobei g die Erdbeschleunigung und ρ die Masse pro Seillänge darstellt. Im Hinblick auf die Minimierung der potentiellen Energie V kann der konstante Faktor ρg vernachlässigt werden, womit sich das Kostenfunktional und die Randbedingungen

$$J(y) = \int_{x_A}^{x_B} y \sqrt{1 + y'(x)^2} dx, \quad y(x_A) = y_A, \quad y(x_B) = y_B \quad (5.23)$$

ergeben. Da in $\phi(y, y') = y \sqrt{1 + y'(x)^2}$ die Ortsvariable x nicht explizit auftritt, ergibt

Abbildung 5.2: Galileo's Problem: Bestimmung der Form $y(x)$ einer hängenden Kette.

sich für die Euler–Lagrange–Gleichung (5.16) der Spezialfall (5.20), d.h. (ohne Argumente)

$$\begin{aligned}\phi(y, y') - y' \frac{\partial \phi}{\partial y'} &= y \sqrt{1 + y'^2} - \frac{y y'^2}{\sqrt{1 + y'^2}} \\ &= \frac{y}{\sqrt{1 + y'^2}} = c_1 = \text{konst.}\end{aligned}\quad (5.24)$$

Unter der Annahme $c_1 \neq 0$ kann die letzte Gleichung in die Form

$$y' = \sqrt{\frac{y^2}{c_1^2} - 1} \quad (5.25)$$

gebracht werden. Unter Berücksichtigung von $y' = \frac{dy}{dx}$ wird x zunächst mittels Integration als Funktion von y ausgedrückt

$$x = \int \frac{1}{\sqrt{\frac{y^2}{c_1^2} - 1}} dy = c_1 \ln \left(\frac{y + \sqrt{y^2 - c_1^2}}{c_1} \right) + c_2,$$

wobei c_2 eine Integrationskonstante darstellt. Durch Umformung und Quadrierung,

$$\left(c_1 e^{(x-c_2)/c_1} - y \right)^2 = y^2 - c_1^2,$$

kann y in folgender Form geschrieben werden:

$$y = \frac{1}{2} c_1 \frac{e^{2(x-c_2)/c_1} + 1}{e^{(x-c_2)/c_1}} = \frac{1}{2} c_1 \left(e^{(x-c_2)/c_1} + e^{-(x-c_2)/c_1} \right).$$

Mit $\cosh b = \frac{1}{2}(e^b + e^{-b})$ ergibt sich also die Kettenlinie (Englisch: *catenary*)

$$y(x) = c_1 \cosh \left(\frac{x - c_2}{c_1} \right). \quad (5.26)$$

Gegebenenfalls kann noch eine dritte Konstante c_3 zu (5.26) addiert werden, um die Kettenlinie für eine in den Punkten A und B aufgehängte Kette durch (numerische) Lösung der Gleichungen

$$y(x_A) = y_A, \quad y(x_B) = y_B, \quad \int_{x_A}^{x_B} \sqrt{1 + y'(x)^2} dx = L \quad (5.27)$$

in Abhängigkeit der Kettenlänge L zu berechnen. Abbildung 5.3 zeigt die Kettenlinie $y(x)$ mit der Länge $L = 1.5$ für verschiedene Aufhängepunkte im Vergleich zu der von Galileo vermuteten Parabel.

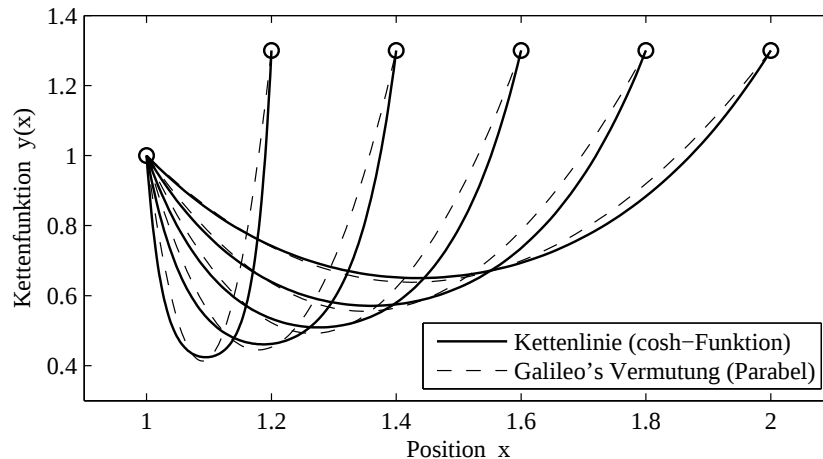


Abbildung 5.3: Hängende Kette der Länge $L = 1.5$ für verschiedene Aufhängepunkte.

5.3 Unbeschränkte Optimalsteuerungsprobleme

In diesem Abschnitt wird die Variationsrechnung auf unbeschränkte Optimalsteuerungsprobleme

$$\min_{\mathbf{u}(\cdot)} \quad J(\mathbf{u}) = \varphi(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} l(\mathbf{x}(t), \mathbf{u}(t), t) dt \quad (5.28a)$$

$$\text{u.B.v.} \quad \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t), \quad \mathbf{x}(t_0) = \mathbf{x}_0 \quad (5.28b)$$

$$x_i(t_f) = x_{f,i}, \quad i \in \mathcal{I}_f \quad (5.28c)$$

angewendet, um notwendige Bedingungen für eine optimale Lösung herzuleiten.

5.3.1 Anwendung der Variationsrechnung

Der bisher betrachtete Fall der Variationsrechnung wird zunächst sukzessive erweitert, um die Systemgleichungen (5.28b), die partiellen Endbedingungen (5.28c) und eine eventuell freie Endzeit t_f zu berücksichtigen.

5.3.1.1 Berücksichtigung der Systemgleichungen

Die Systemgleichungen (5.28b) können als (dynamische) Gleichungsbeschränkungen betrachtet werden, die (ähnlich dem Vorgehen bei beschränkten statischen Optimierungsproblemen) anhand von Lagrange-Multiplikatoren im Kostenfunktional berücksichtigt werden

$$\bar{J} = \varphi(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} l(\mathbf{x}(t), \mathbf{u}(t), t) + \boldsymbol{\lambda}^\top (\mathbf{f}(\mathbf{x}, \mathbf{u}, t) - \dot{\mathbf{x}}) dt. \quad (5.29)$$

Im Gegensatz zum endlich-dimensionalen Fall stellt $\boldsymbol{\lambda}(t), t \in [t_0, t_f]$ nun eine Zeitfunktion dar.

Die Endzeit t_f sei zunächst fest, d.h. $\varphi = \varphi(\mathbf{x}(t_f))$. Zur Betrachtung der Variation $\delta \bar{J}$ des Kostenfunktionals werden Variationen im Zustand $\mathbf{x}(t) = \mathbf{x}^*(t) + \varepsilon \delta \mathbf{x}(t)$ und Eingang

$\mathbf{u}(t) = \mathbf{u}^*(t) + \varepsilon \delta \mathbf{u}(t)$ um eine optimale Trajektorie $\mathbf{x}^*(t)$, $\mathbf{u}^*(t)$, $t \in [t_0, t_f]$ angesetzt

$$\begin{aligned} \bar{J} &= \varphi(\mathbf{x}^*(t_f) + \varepsilon \delta \mathbf{x}(t_f)) \\ &+ \int_{t_0}^{t_f} l(\mathbf{x}^* + \varepsilon \delta \mathbf{x}, \mathbf{u}^* + \varepsilon \delta \mathbf{u}, t) + \boldsymbol{\lambda}^\top [\mathbf{f}(\mathbf{x}^* + \varepsilon \delta \mathbf{x}, \mathbf{u}^* + \varepsilon \delta \mathbf{u}, t) - (\dot{\mathbf{x}}^* + \varepsilon \delta \dot{\mathbf{x}})] dt. \end{aligned} \quad (5.30)$$

und die Variation $\delta \bar{J} = \frac{d\bar{J}}{d\varepsilon} \Big|_{\varepsilon=0}$ betrachtet (ohne Argumente und Sternindex)

$$\delta \bar{J} = \left(\frac{\partial \varphi}{\partial \mathbf{x}} \right)_{t_f}^\top \delta \mathbf{x}(t_f) + \int_{t_0}^{t_f} \left[\frac{\partial l}{\partial \mathbf{x}} + \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right)^\top \boldsymbol{\lambda} \right]^\top \delta \mathbf{x} + \left[\frac{\partial l}{\partial \mathbf{u}} + \left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right)^\top \boldsymbol{\lambda} \right]^\top \delta \mathbf{u} - \boldsymbol{\lambda}^\top \delta \dot{\mathbf{x}} dt. \quad (5.31)$$

Partielle Integration der Variation $\delta \dot{\mathbf{x}}(t)$ liefert

$$\delta \bar{J} = \left(\frac{\partial \varphi}{\partial \mathbf{x}} \right)_{t_f}^\top \delta \mathbf{x}(t_f) - [\boldsymbol{\lambda}^\top \delta \mathbf{x}]_{t_0}^{t_f} + \int_{t_0}^{t_f} \left[\frac{\partial l}{\partial \mathbf{x}} + \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right)^\top \boldsymbol{\lambda} + \dot{\boldsymbol{\lambda}} \right]^\top \delta \mathbf{x} + \left[\frac{\partial l}{\partial \mathbf{u}} + \left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right)^\top \boldsymbol{\lambda} \right]^\top \delta \mathbf{u} dt.$$

Zusätzlich kann der Restterm vereinfacht werden

$$[\boldsymbol{\lambda}^\top \delta \mathbf{x}]_{t_0}^{t_f} = \boldsymbol{\lambda}(t_f)^\top \delta \mathbf{x}(t_f) - \boldsymbol{\lambda}(t_0)^\top \delta \mathbf{x}(t_0) = \boldsymbol{\lambda}(t_f)^\top \delta \mathbf{x}(t_f),$$

da für zulässige Variationen $\delta \mathbf{x}(t_0) = \mathbf{0}$ gelten muss, um die Anfangsbedingungen $\mathbf{x}(t_0) = \mathbf{x}_0$ in (5.28b) zu erfüllen.

Damit $\mathbf{x}^*(t)$, $\mathbf{u}^*(t)$ tatsächlich eine optimale Lösung des Optimalsteuerungsproblems (5.28) darstellt, muss $\delta \bar{J} = 0$ für alle zulässigen Variationen $\delta \mathbf{x}(t)$ und $\delta \mathbf{u}(t)$ erfüllt sein. Es müssen also die folgenden Gleichungen gelten

$$\dot{\boldsymbol{\lambda}} = -\frac{\partial l}{\partial \mathbf{x}} - \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right)^\top \boldsymbol{\lambda}, \quad t \in (t_0, t_f) \quad (5.32a)$$

$$\mathbf{0} = \frac{\partial l}{\partial \mathbf{u}} + \left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right)^\top \boldsymbol{\lambda}, \quad t \in (t_0, t_f) \quad (5.32b)$$

$$0 = \left(\frac{\partial \varphi}{\partial \mathbf{x}} - \boldsymbol{\lambda} \right)_{t=t_f}^\top \delta \mathbf{x}(t_f) = \sum_{i=1}^n \left(\frac{\partial \varphi}{\partial x_i} - \lambda_i \right)_{t=t_f} \delta x_i(t_f). \quad (5.32c)$$

Das heißt, es muss ein zu $\mathbf{x}^*(t)$, $\mathbf{u}^*(t)$ gehörender Lagrange-Multiplikator $\boldsymbol{\lambda}(t)$, $t \in [t_0, t_f]$ existieren, der die Dgl. (5.32a), die algebraische Zwangsbedingung (5.32b) und die Endbedingung (5.32c) für alle zulässigen Endvariationen $\delta \mathbf{x}(t_f)$ erfüllt.

Da partielle Endbedingungen (5.28c) betrachtet werden, lohnt es sich, die Bedingung (5.32c) ebenfalls komponentenweise anzuschauen. Für die Zustände x_i , $i \in \mathcal{I}_f$, die am Endzeitpunkt $t = t_f$ fixiert sind, siehe (5.28c), muss eine zulässige Variation $\delta x_i(t_f) = 0$ erfüllen, während für alle freien Endzustände $x_i(t_f)$, $i \notin \mathcal{I}_f$ die Endvariation beliebig sein kann. Es muss also gelten

$$\lambda_i(t_f) = \frac{\partial \varphi}{\partial x_i} \Big|_{t=t_f}, \quad i \notin \mathcal{I}_f, \quad (5.33)$$

damit die *Transversalitätsbedingung* (5.32c) für alle zulässigen Endvariationen erfüllt ist.

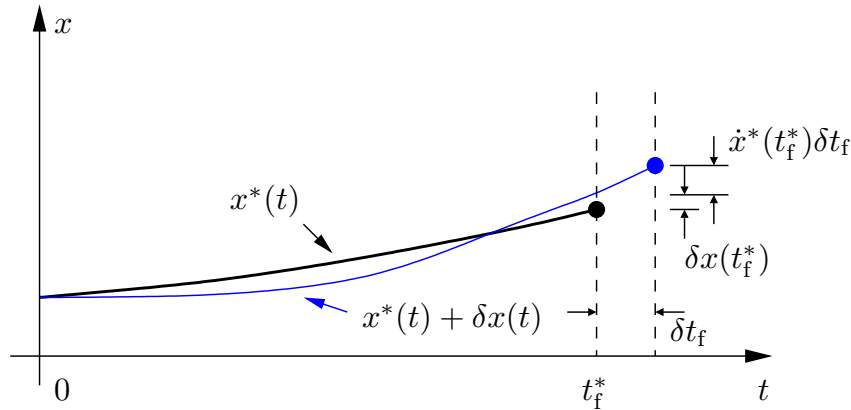


Abbildung 5.4: Totale Variation $\Delta \mathbf{x}_f = \delta \mathbf{x}(t_f^*) + \dot{\mathbf{x}}^*(t_f^*) \delta t_f$ des Endzustandes für $x \in \mathbb{R}$.

5.3.1.2 Berücksichtigung einer freien Endzeit

Falls die Endzeit t_f des Optimierungsproblems (5.1) frei ist, wird die Herleitung der Variation $\delta \bar{J}$ etwas komplizierter, da zusätzlich zu den Variationen $\delta \mathbf{x}(t)$ und $\delta \mathbf{u}(t)$ noch die Variation $t_f = t_f^* + \varepsilon \delta t_f$ um die optimale Endzeit t_f^* berücksichtigt werden muss. Die Variation $\delta \bar{J}$ bzgl. $\delta \mathbf{x}(t)$, $\delta \mathbf{u}(t)$ und δt_f ist eine Erweiterung von (5.31) und lautet (ohne Argumente und Sternindex)

$$\begin{aligned} \delta \bar{J} = & \left(\frac{\partial \varphi}{\partial \mathbf{x}} \right)_{t_f}^\top \left(\overbrace{\delta \mathbf{x}(t_f) + \dot{\mathbf{x}}(t_f) \delta t_f}^{= \Delta \mathbf{x}_f} \right) + \left(\frac{\partial \varphi}{\partial t} \right)_{t_f} \delta t_f + \left(l + \boldsymbol{\lambda}^\top (\mathbf{f} - \dot{\mathbf{x}}) \right)_{t_f} \delta t_f \\ & + \int_{t_0}^{t_f} \left[\frac{\partial l}{\partial \mathbf{x}} + \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right)^\top \boldsymbol{\lambda} \right]^\top \delta \mathbf{x} + \left[\frac{\partial l}{\partial \mathbf{u}} + \left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right)^\top \boldsymbol{\lambda} \right]^\top \delta \mathbf{u} - \boldsymbol{\lambda}^\top \delta \dot{\mathbf{x}} \, dt. \quad (5.34) \end{aligned}$$

Die *totale Variation* $\Delta \mathbf{x}_f$ (genau genommen müsste es $\Delta \mathbf{x}_f = \delta \mathbf{x}(t_f^*) + \dot{\mathbf{x}}^*(t_f^*) \delta t_f$ heißen) berücksichtigt den zusätzlichen Einfluss der Zeitvariation δt_f auf die Änderung des Endzustandes, siehe Abbildung 5.4. Der Term $(l + \boldsymbol{\lambda}^\top (\mathbf{f} - \dot{\mathbf{x}}))_{t_f} \delta t_f$ in (5.34) stellt die Variation des Integralwertes bei einer kleinen Variation δt_f der Endzeit dar.

Partielle Integration von (5.34) mit $\boldsymbol{\lambda}(t_0)^\top \delta \mathbf{x}(t_0) = 0$ liefert

$$\begin{aligned} \delta \bar{J} = & \left(\frac{\partial \varphi}{\partial \mathbf{x}} \right)_{t_f}^\top \Delta \mathbf{x}_f - \boldsymbol{\lambda}(t_f)^\top \left(\overbrace{\delta \mathbf{x}(t_f) + \dot{\mathbf{x}}(t_f) \delta t_f}^{= \Delta \mathbf{x}_f} \right) + \left(\frac{\partial \varphi}{\partial t} + l + \boldsymbol{\lambda}^\top \mathbf{f} \right)_{t_f} \delta t_f \\ & + \int_{t_0}^{t_f} \left[\frac{\partial l}{\partial \mathbf{x}} + \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right)^\top \boldsymbol{\lambda} + \dot{\boldsymbol{\lambda}} \right]^\top \delta \mathbf{x} + \left[\frac{\partial l}{\partial \mathbf{u}} + \left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right)^\top \boldsymbol{\lambda} \right]^\top \delta \mathbf{u} \, dt. \end{aligned}$$

Falls $\mathbf{x}^*(t)$, $\mathbf{u}^*(t)$ mit der Endzeit t_f^* eine optimale Lösung darstellt, muss $\delta \bar{J} = 0$ sein für

alle zulässigen Variationen $\delta \mathbf{x}(t)$, $\delta \mathbf{u}(t)$ und δt_f . Es muss also gelten

$$\dot{\boldsymbol{\lambda}} = -\frac{\partial l}{\partial \mathbf{x}} - \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)^\top \boldsymbol{\lambda}, \quad t \in (t_0, t_f) \quad (5.35a)$$

$$\mathbf{0} = \frac{\partial l}{\partial \mathbf{u}} + \left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right)^\top \boldsymbol{\lambda}, \quad t \in (t_0, t_f) \quad (5.35b)$$

$$0 = \left(\frac{\partial \varphi}{\partial \mathbf{x}} - \boldsymbol{\lambda}\right)_{t=t_f}^\top \Delta \mathbf{x}_f \quad (5.35c)$$

$$0 = \left(\frac{\partial \varphi}{\partial t} + l(\mathbf{x}, \mathbf{u}, t) + \boldsymbol{\lambda}^\top \mathbf{f}(\mathbf{x}, \mathbf{u}, t)\right)_{t=t_f}. \quad (5.35d)$$

Analog zum vorherigen Abschnitt ist (5.35c) für die Endbedingungen (5.33) erfüllt. Zusätzlich kommt aufgrund der freien Endzeit t_f die Endbedingung (5.35d) hinzu.

5.3.2 Optimalitätsbedingungen

Die im letzten Abschnitt hergeleiteten Beziehungen stellen notwendige Bedingungen dar, die eine optimale Lösung $\mathbf{x}^*(t)$, $\mathbf{u}^*(t)$ des Optimalsteuerungsproblems (5.28) erfüllen muss. Mit Hilfe der sogenannten *Hamilton-Funktion* können diese Beziehungen in einer kompakteren Weise geschrieben werden.

Definition 5.1 (Hamilton-Funktion) Die *Hamilton-Funktion* für das Optimalsteuerungsproblem (5.28) ist

$$H(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}, t) = l(\mathbf{x}, \mathbf{u}, t) + \boldsymbol{\lambda}^\top \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \quad (5.36)$$

mit den *Lagrange-Multiplikatoren* $\boldsymbol{\lambda}(t) \in \mathbb{R}^n$.

Die notwendigen Optimalitätsbedingungen 1. Ordnung für das Optimalsteuerungsproblem (5.28) lauten (ohne Sternindex)

$$\left. \begin{aligned} \dot{\mathbf{x}} &= \frac{\partial H}{\partial \boldsymbol{\lambda}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \\ \dot{\boldsymbol{\lambda}} &= -\frac{\partial H}{\partial \mathbf{x}} = -\frac{\partial l}{\partial \mathbf{x}} - \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)^\top \boldsymbol{\lambda} \\ \mathbf{0} &= \frac{\partial H}{\partial \mathbf{u}} = \frac{\partial l}{\partial \mathbf{u}} + \left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right)^\top \boldsymbol{\lambda} \end{aligned} \right\} t \in (t_0, t_f) \quad (5.37)$$

$$\left. \begin{aligned} \dot{\boldsymbol{\lambda}} &= -\frac{\partial H}{\partial \mathbf{x}} = -\frac{\partial l}{\partial \mathbf{x}} - \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)^\top \boldsymbol{\lambda} \\ \mathbf{0} &= \frac{\partial H}{\partial \mathbf{u}} = \frac{\partial l}{\partial \mathbf{u}} + \left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right)^\top \boldsymbol{\lambda} \end{aligned} \right\} t \in (t_0, t_f) \quad (5.38)$$

$$\left. \begin{aligned} \dot{\boldsymbol{\lambda}} &= -\frac{\partial H}{\partial \mathbf{x}} = -\frac{\partial l}{\partial \mathbf{x}} - \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)^\top \boldsymbol{\lambda} \\ \mathbf{0} &= \frac{\partial H}{\partial \mathbf{u}} = \frac{\partial l}{\partial \mathbf{u}} + \left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right)^\top \boldsymbol{\lambda} \end{aligned} \right\} t \in (t_0, t_f) \quad (5.39)$$

mit den gegebenen Randbedingungen in (5.1b), (5.1c)

$$\boxed{\mathbf{x}(t_0) = \mathbf{x}_0} \quad (5.40a)$$

$$\boxed{x_i(t_f) = x_{f,i}, \quad i \in \mathcal{I}_f} \quad (5.40b)$$

und den Transversalitätsbedingungen

$$\boxed{\lambda_i(t_f) = \left. \frac{\partial \varphi}{\partial x_i} \right|_{t=t_f} \quad \text{für } i \notin \mathcal{I}_f} \quad (5.41)$$

$$\boxed{H(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}, t) \big|_{t=t_f} = -\left. \frac{\partial \varphi}{\partial t} \right|_{t=t_f} \quad \text{falls } t_f \text{ frei.}} \quad (5.42)$$

Zusätzlich kann die Legendre-Bedingung (5.18) für das Optimalsteuerungsproblem (5.1) wie folgt angegeben werden

$$\frac{\partial^2 H}{\partial \mathbf{u}^2} \quad \text{positiv semi-definit} \quad \forall t \in [t_0, t_f], \quad (5.43)$$

die eine notwendige Optimalitätsbedingung 2. Ordnung für eine optimale Lösung $\mathbf{x}^*(t)$, $\mathbf{u}^*(t)$ darstellt.

Die Differentialgleichungen (5.37)–(5.38) werden als *kanonische Gleichungen* oder als *Hamilton-Gleichungen* bezeichnet, während (5.38) für sich genommen das *adjungierte System* mit dem *adjungierten Zustand* $\boldsymbol{\lambda}$ darstellt.

Die Optimalitätsbedingungen definieren eine *Zwei-Punkt-Randwertaufgabe* (RWA) für die optimalen Zustände $\mathbf{x}^*(t)$, $\boldsymbol{\lambda}^*(t)$ und die optimale Steuerung $\mathbf{u}^*(t)$. Falls t_f fest ist, umfasst die RWA (5.37)–(5.41) $2n$ Gleichungen mit $2n$ Randbedingungen. Falls t_f frei ist, kommt zusätzlich die Transversalitätsbedingung (5.42) hinzu, um die optimale Endzeit t_f^* bestimmen zu können. Die Endbedingungen (5.40b), (5.41) beinhalten insbesondere die folgenden Sonderfälle:

- Fester Endzustand $\mathbf{x}(t_f) = \mathbf{x}_f$, d.h. $\mathcal{I}_f = \{1, \dots, n\}$:
Der komplette Zustand $\mathbf{x}$ ist am Anfang und Ende des Zeitintervalls $[t_0, t_f]$ fixiert, während der adjungierte Endzustand $\boldsymbol{\lambda}(t_f)$ frei ist.
- Freier Endzustand $\mathbf{x}(t_f)$, d.h. $\mathcal{I}_f = \{\}$:
In diesem Fall ist der komplette adjungierte Endzustand (5.41) fixiert, d.h. $\boldsymbol{\lambda}(t_f) = \frac{\partial \varphi}{\partial \mathbf{x}}|_{t_f}$, und die Randbedingungen teilen sich in Anfangs- und Endbedingungen für $\mathbf{x}(t_0)$ und $\boldsymbol{\lambda}(t_f)$ auf.
- Freie Endzeit t_f mit Mayer-Term $\varphi = 0$ oder $\varphi = \varphi(\mathbf{x}(t_f))$:
In diesem Fall ist die Transversalitätsbedingung (5.42) homogen, d.h. $H|_{t=t_f} = 0$.

Eine besondere Eigenschaft der Hamilton-Funktion (5.36) ist ihr Verhalten entlang einer optimalen Lösung:

$$\begin{aligned} \frac{d}{dt} H &= \frac{\partial H}{\partial t} + \underbrace{\left(\frac{\partial H}{\partial \mathbf{x}} \right)^\top \dot{\mathbf{x}}}_{= \left(\frac{\partial H}{\partial \mathbf{x}} \right)^\top \mathbf{f}} + \underbrace{\left(\frac{\partial H}{\partial \mathbf{u}} \right)^\top \dot{\mathbf{u}}}_{= \mathbf{0}} + \underbrace{\left(\frac{\partial H}{\partial \boldsymbol{\lambda}} \right)^\top \dot{\boldsymbol{\lambda}}}_{= -\mathbf{f}^\top \frac{\partial H}{\partial \mathbf{x}}} = \frac{\partial H}{\partial t}. \end{aligned} \quad (5.44)$$

Für zeitinvariante Probleme (5.28), bei denen weder die Kostenfunktion l noch die Systemfunktion $\mathbf{f}$ explizit von t abhängen, ist die Hamilton-Funktion H also entlang einer optimalen Trajektorie *konstant*.

Aufgabe 5.2 (feste Endzeit & Endbedingung) Betrachtet wird das Problem

$$\begin{aligned} \min_{u(\cdot)} \quad & \int_0^1 \frac{1}{2} u^2 + \frac{a}{2} x^2 \, dt, \quad a > 0 \\ \text{u.B.v.} \quad & \dot{x} = u, \quad x(0) = 1, \quad x(1) = 0. \end{aligned}$$

Zeigen Sie, dass die Lösung

$$x^*(t) = \frac{1}{1 - e^{2\sqrt{a}}} \left(e^{\sqrt{a}t} - e^{\sqrt{a}(2-t)} \right), \quad u^*(t) = \frac{\sqrt{a}}{1 - e^{2\sqrt{a}}} \left(e^{\sqrt{a}t} + e^{\sqrt{a}(2-t)} \right) \quad (5.45)$$

lautet und interpretieren Sie die Ergebnisse in Abbildung 5.5, die für verschiedene Parameterwerte a dargestellt sind.

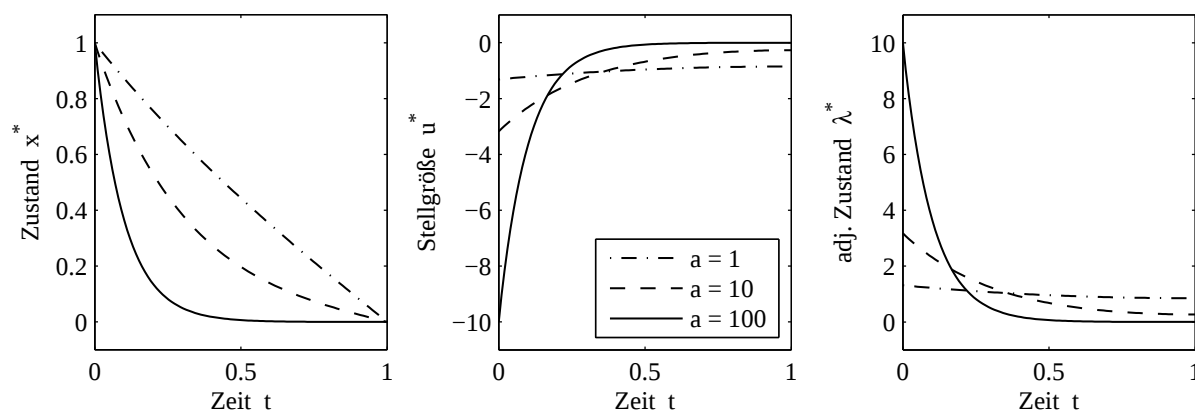


Abbildung 5.5: Optimale Trajektorien in Aufgabe 5.2.

Aufgabe 5.3 (feste Endzeit & partielle Endbedingung) Betrachtet wird

$$\begin{aligned} \min_{u(\cdot)} \quad & \frac{a}{2} x_2(1)^2 + \int_0^1 \frac{1}{2} u^2 dt, \quad a \geq 0 \\ \text{u.B.v.} \quad & \dot{x}_1 = x_2, \quad x_1(0) = 1, \quad x_1(1) = 0 \\ & \dot{x}_2 = u, \quad x_2(0) = 0 \end{aligned}$$

Zeigen Sie, dass sich für den (freien) Endzustand $x_2^*(1) = -6/(4+a)$ in Abhängigkeit des Parameters $a \geq 0$ ergibt und dass die optimale Lösung

$$x_1^*(t) = \frac{2(1+a)}{4+a} t^3 - \frac{3(2+a)}{a+4} t^2 + 1, \quad u^*(t) = \frac{12(1+a)}{4+a} t - \frac{6(2+a)}{a+4} \quad (5.46)$$

lautet. Interpretieren Sie die Ergebnisse in Abbildung 5.6.

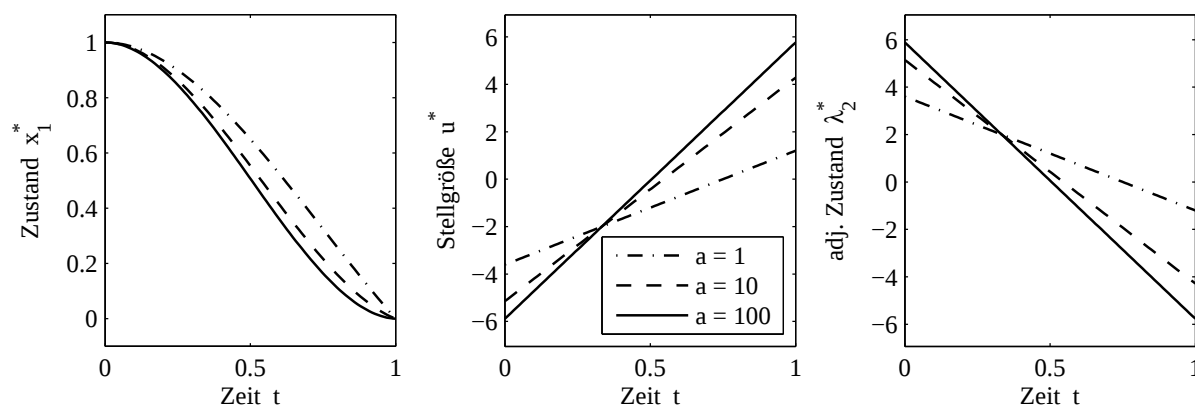
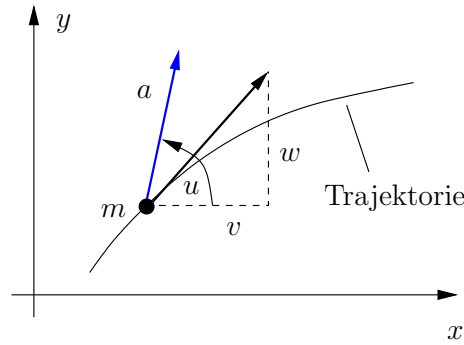


Abbildung 5.6: Optimale Trajektorien in Aufgabe 5.3.

Abbildung 5.7: Bewegung eines Partikels der Masse m in der (x, y) -Ebene.

5.3.3 Beispiel zur Betrachtung einer freien Endzeit

Betrachtet wird ein Partikel der Masse m in der (x, y) -Ebene, auf das eine konstante Kraft $F = ma$ wirkt. Die Stellgröße u des Problems ist der Winkel zwischen der Schubrichtung und der x -Achse, siehe Abbildung 5.7. Ziel ist es, das Partikel in minimaler Zeit $[t_0 = 0, t_f]$ zu einem Zielpunkt (x_f, y_f) zu steuern. Unter der Annahme, dass außer dem Schub keine weiteren Kräfte auftreten, kann das Optimalsteuerungsproblem wie folgt formuliert werden

$$\min_{u(\cdot)} \quad t_f \quad (5.47a)$$

$$\text{u.B.v.} \quad \dot{x} = v, \quad x(0) = x_0, \quad x(t_f) = x_f \quad (5.47b)$$

$$\dot{v} = a \cos(u), \quad v(0) = v_0 \quad (5.47c)$$

$$\dot{y} = w, \quad y(0) = y_0, \quad y(t_f) = y_f \quad (5.47d)$$

$$\dot{w} = a \sin(u), \quad w(0) = w_0. \quad (5.47e)$$

Der Endzustand ist nur für die Position (x, y) vorgegeben und nicht für die Geschwindigkeiten, so dass sich in Analogie zu den partiellen Endbedingungen (5.40b) die Menge $\mathcal{I}_f = \{x, y\}$ ergibt. Die Hamilton-Funktion (5.36) für das Problem ist

$$H(\mathbf{x}, u, \boldsymbol{\lambda}) = \lambda_x v + \lambda_v a \cos(u) + \lambda_y w + \lambda_w a \sin(u) \quad (5.48)$$

mit $\mathbf{x} = [x, v, y, w]^T$ und den adjungierten Zuständen $\boldsymbol{\lambda} = [\lambda_x, \lambda_v, \lambda_y, \lambda_w]^T$. Das adjungierte System (5.38) mit den Endbedingungen (5.41) lautet

$$\dot{\lambda}_x = -\frac{\partial H}{\partial x} = 0 \quad (5.49a)$$

$$\dot{\lambda}_v = -\frac{\partial H}{\partial v} = -\lambda_x, \quad \lambda_v(t_f) = 0 \quad (5.49b)$$

$$\dot{\lambda}_y = -\frac{\partial H}{\partial y} = 0 \quad (5.49c)$$

$$\dot{\lambda}_w = -\frac{\partial H}{\partial w} = -\lambda_y, \quad \lambda_w(t_f) = 0, \quad (5.49d)$$

woraus direkt

$$\lambda_x = \text{konst.}, \quad \lambda_v = \lambda_x(t_f - t), \quad \lambda_y = \text{konst.}, \quad \lambda_w = \lambda_y(t_f - t) \quad (5.50)$$

folgt. Des Weiteren muss (5.39) gelten, d.h.

$$\frac{\partial H}{\partial u} = -\lambda_v a \sin(u) + \lambda_w a \cos(u) = 0,$$

womit sich u wie folgt berechnen lässt

$$\begin{aligned} \tan(u) &= \frac{\lambda_w}{\lambda_v} \stackrel{(5.50)}{=} \frac{\lambda_y(t_f - t)}{\lambda_x(t_f - t)} = \frac{\lambda_y}{\lambda_x} = \text{konst.} \\ \Rightarrow u &= \arctan\left(\frac{\lambda_y}{\lambda_x}\right) \quad \text{falls} \quad -\frac{\pi}{2} < u < \frac{\pi}{2}. \end{aligned} \quad (5.51)$$

Die optimale Steuerung ist also auf dem gesamten Zeitintervall $[t_0 = 0, t_f]$ konstant und die Zustände $x(t)$ und $y(t)$ können durch zweifache Integration der Differentialgleichungen (5.47c), (5.47e) und Einsetzen der Anfangsbedingungen bestimmt werden ⁴

$$x(t) = g_x(\lambda_x, \lambda_y, t) = x_0 + v_0 t + \frac{1}{2} a \cos(u) t^2, \quad \cos(u) \stackrel{(5.51)}{=} \frac{1}{\sqrt{1 + \lambda_y^2 / \lambda_x^2}} \quad (5.52a)$$

$$v(t) = g_v(\lambda_x, \lambda_y, t) = v_0 + a \cos(u) t \quad (5.52b)$$

$$y(t) = g_y(\lambda_x, \lambda_y, t) = y_0 + w_0 t + \frac{1}{2} a \sin(u) t^2, \quad \sin(u) \stackrel{(5.51)}{=} \frac{\lambda_y}{\lambda_x \sqrt{1 + \lambda_y^2 / \lambda_x^2}} \quad (5.52c)$$

$$w(t) = g_w(\lambda_x, \lambda_y, t) = w_0 + a \sin(u) t. \quad (5.52d)$$

Da die Endzeit t_f frei ist, muss zusätzlich die Transversalitätsbedingung (5.42) gelten, wobei sich die Hamilton-Funktion (5.48) aufgrund der Endbedingungen $\lambda_v(t_f) = \lambda_w(t_f) = 0$ entsprechend vereinfacht

$$H(\mathbf{x}, u, \boldsymbol{\lambda})|_{t=t_f} = -1 \quad \Rightarrow \quad \lambda_x g_v(\lambda_x, \lambda_y, t_f) + \lambda_y g_w(\lambda_x, \lambda_y, t_f) = -1. \quad (5.53)$$

Die gewünschten Endbedingungen $x(t_f) = x_f$, $y(t_f) = y_f$ und die Endzeit t_f müssen aus den Gleichungen (5.52a), (5.52c), (5.53) berechnet werden, die ein Gleichungssystem für die drei verbleibenden Unbekannten $(\lambda_x, \lambda_y, t_f)$ darstellen

$$\begin{bmatrix} g_x(\lambda_x, \lambda_y, t_f) \\ g_y(\lambda_x, \lambda_y, t_f) \\ \lambda_x g_v(\lambda_x, \lambda_y, t_f) + \lambda_y g_w(\lambda_x, \lambda_y, t_f) \end{bmatrix} - \begin{bmatrix} x_f \\ y_f \\ -1 \end{bmatrix} = \mathbf{0} \quad (5.54)$$

welches auf numerischem Wege gelöst werden kann.

Eine geeignete MATLAB-Funktion zur Lösung von nichtlinearen Gleichungen ist **fsolve** aus der Optimization Toolbox. Die Funktion **fsolve** verwendet standardmäßig die Methode der Vertrauensbereiche (siehe Abschnitt 3.4.1), um ein Gleichungssystem in Residuenform $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ als Minimierungsproblem in $\mathbf{x}$ zu lösen.

Als Beispiel ist in Abbildung 5.8 der MATLAB-Code dargestellt, wie **fsolve** zur Lösung von (5.54) verwendet werden kann. Der gewünschte Endpunkt (x_f, y_f) wird beim Aufruf der Funktion **particle(xf,yf)** übergeben, wobei angenommen wird, dass das Partikel am Punkt $(x_0, y_0) = (0, 0)$ mit der Geschwindigkeit $(v_0, w_0) = (0, 1)$ in vertikale Richtung startet. Abbildung 5.9 stellt die optimalen Bahnen $x^*(t)$, $y^*(t)$ des Partikels in der (x, y) -Ebene für verschiedene Endpunkte (x_f, y_f) dar. Die Pfeile zeigen die (konstante) Richtung $u = \arctan(\lambda_y / \lambda_x)$ der angreifenden Kraft ma an.

⁴ Trigonometrische Beziehungen: $\sin(\arctan(b)) = b / \sqrt{1 + b^2}$, $\cos(\arctan(b)) = 1 / \sqrt{1 + b^2}$.

```

function [t,x,y,p] = particle(xf,yf)
% -----
% (xf,yf): gewünschter Endpunkt
% (t,x,y): Trajektorien des Partikels
% p:      Parameterstruktur

p.a = 1;                                % Parameter
p.x0=0; p.v0=0; p.y0=0; p.w0=1;         % Anfangsbedingungen
p.xf=xf; p.yf=yf;                       % Endbedingungen (Übergabe aus Funktionsaufruf)

opt = optimset('Display','iter');        % Optionen
X0 = [-1,0,1];                          % Startwert
Xopt = fsolve(@eqns,X0,opt,p);           % Numerische Lösung mit fsolve
p.lx=Xopt(1); p.ly=Xopt(2); p.tf=Xopt(3); % Lösung

t = linspace(0,p.tf,100);               % Trajektorien
x = xfct(p.lx,p.ly,t,p);
y = yfct(p.lx,p.ly,t,p);
% -----
function res = eqns(X,p)                 % Gleichungen in Residuenform
lx=X(1); ly=X(2); tf=X(3);
res = [ xfct(lx,ly,tf,p) - p.xf;
        yfct(lx,ly,tf,p) - p.yf;
        lx*vfct(lx,ly,tf,p) + ly*wfct(lx,ly,tf,p) + 1 ];
% -----
function x = xfct(lx,ly,t,p)             % Funktionen für x und v
cosu = 1/sqrt(1+(ly/lx)^2);
x = p.x0 + p.v0*t + p.a/2*cosu*t.^2;    % 't.^2' steht für komponentenweise Auswertung
function v = vfct(lx,ly,t,p)
cosu = 1/sqrt(1+(ly/lx)^2);
v = p.v0 + p.a*cosu*t;
% -----
function y = yfct(lx,ly,t,p)             % Funktionen für y und w
sinu = ly/(lx*sqrt(1+(ly/lx)^2));
y = p.y0 + p.w0*t + p.a/2*sinu*t.^2;    % 't.^2' steht für komponentenweise Auswertung
function w = wfct(lx,ly,t,p)
sinu = ly/(lx*sqrt(1+(ly/lx)^2));
w = p.w0 + p.a*sinu*t;

```

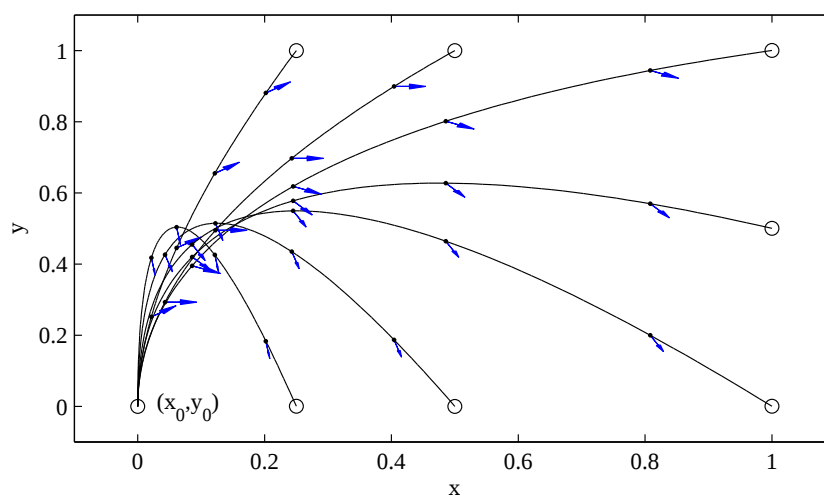
Abbildung 5.8: MATLAB-Code für das Partikel-Problem unter Verwendung von `fsolve`.

Abbildung 5.9: Zeitoptimale Steuerung eines Partikels zu verschiedenen Endpunkten.

5.3.4 Allgemeine Vorgehensweise und singulärer Fall

Die vorherigen Beispiele illustrierten, wie die Optimalitätsbedingungen (5.37)–(5.42) bei einfachen Problemen weitgehend analytisch gelöst werden können. Eine generelle Vorgehensweise zur Berechnung einer optimalen Steuerung mittels der Optimalitätsbedingungen gliedert sich in die folgenden Schritte:

1. Aufstellen der Hamilton-Funktion $H(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}, t) = l(\mathbf{x}, \mathbf{u}, t) + \boldsymbol{\lambda}^\top \mathbf{f}(\mathbf{x}, \mathbf{u}, t)$
2. Berechnung von $\mathbf{u}$ aus (5.39), so dass sich $\mathbf{u}$ als Funktion von $(\mathbf{x}, \boldsymbol{\lambda}, t)$ ergibt:

$$\mathbf{u} = \boldsymbol{\psi}(\mathbf{x}, \boldsymbol{\lambda}, t). \quad (5.55)$$

3. Einsetzen der Funktion (5.55) in die kanonischen Gleichungen (5.37)–(5.38) führt auf das Randwertproblem

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \boldsymbol{\psi}(\mathbf{x}, \boldsymbol{\lambda}, t), t), \quad \dot{\boldsymbol{\lambda}} = -H_{\mathbf{x}}(\mathbf{x}, \boldsymbol{\psi}(\mathbf{x}, \boldsymbol{\lambda}, t), \boldsymbol{\lambda}, t), \quad \text{RBen} \quad (5.40), (5.41)$$

mit $H_{\mathbf{x}} = \frac{\partial H}{\partial \mathbf{x}}$, das nun unabhängig von der Steuergröße $\mathbf{u}$ ist. Falls die Endzeit t_f frei ist, kommt noch die Endbedingung (5.42) hinzu.

4. Die Lösung des Randwertproblems liefert $\mathbf{x}^*(t)$, $\boldsymbol{\lambda}^*(t)$, $t \in [t_0, t_f]$, womit sich die Steuertrajektorie $\mathbf{u}^*(t)$ aus (5.55) ergibt. In den meisten Fällen muss das Randwertproblem numerisch gelöst werden (siehe letztes Kapitel).

Es sei angemerkt, dass die Lösung der kanonischen Gleichungen nicht automatisch die Optimalität der resultierenden Trajektorien $\mathbf{x}^*(t)$, $\mathbf{u}^*(t)$ impliziert, da es sich lediglich um notwendige Bedingungen handelt. Aus der Sicht des Anwenders wird aber häufig angenommen, dass eine (eindeutige) optimale Lösung für ein gegebenes Problem existiert, und somit die Auswertung der notwendigen Optimalitätsbedingungen ausreichend ist.

Falls $\mathbf{u}$ in der Systemfunktion $\mathbf{f}(\mathbf{x}, \mathbf{u})$ oder im Integralanteil $l(\mathbf{x}, \mathbf{u}, t)$ des Kostenfunktional (5.28a) nichtlinear eingeht, muss die Funktion (5.55) in der Regel numerisch bestimmt werden. In vielen praktischen Fällen kann dies aber umgangen werden:

Beispiel 5.2 (Eingangsaffines Systeme & energieoptimaler Kostenanteil)

Das Aufstellen der Funktion (5.55) ist besonders einfach für eingangsaffine Systeme

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) = \mathbf{f}_0(\mathbf{x}) + \sum_{i=1}^m \mathbf{f}_i(\mathbf{x}) u_i \quad (5.56)$$

mit energieoptimalen Anteil im Kostenfunktional

$$J(\mathbf{u}) = \varphi(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} l_0(\mathbf{x}) + \frac{1}{2} \sum_{i=1}^m r_i u_i^2 dt, \quad r_i > 0. \quad (5.57)$$

In diesem Fall berechnet sich u_i explizit aus der Bedingung $\frac{\partial H}{\partial u_i} = r_i u_i + \boldsymbol{\lambda}^\top \mathbf{f}_i(\mathbf{x}) = 0$:

$$u_i = \psi_i(\mathbf{x}, \boldsymbol{\lambda}) = -\frac{1}{r_i} \boldsymbol{\lambda}^\top \mathbf{f}_i(\mathbf{x}), \quad i = 1, \dots, m.$$

Problematisch ist der Fall, wenn ein oder mehrere Elemente u_i von $\mathbf{u}$ nicht aus der Stationaritätsbedingung (5.39) berechnet werden können. Dies lässt sich am einfachsten am Beispiel des eingangsaffinen Systems (5.56) mit skalarer Steuergröße u ($m = 1$) erklären. Falls im Kostenfunktional (5.28a) der Integralanteil l unabhängig von u ist, d.h. $l = l(\mathbf{x}, t)$, lautet die Hamilton-Funktion und die erste Ableitung nach u

$$H(\mathbf{x}, u, \boldsymbol{\lambda}, t) = l(\mathbf{x}, t) + \boldsymbol{\lambda}^\top [\mathbf{f}_0(\mathbf{x}) + \mathbf{f}_1(\mathbf{x})u], \quad H_u(\mathbf{x}, \boldsymbol{\lambda}) = \boldsymbol{\lambda}^\top \mathbf{f}_1(\mathbf{x}) = 0$$

mit $H_u = \frac{\partial H}{\partial u}$. Man spricht in diesem Fall von einem *singulären Problem*, da u_i nicht aus der Beziehung $\frac{\partial H}{\partial u} = 0$ bestimmt werden kann.

Beim singulären Fall orientiert sich das Vorgehen an der Tatsache, dass für eine optimale Lösung $\mathbf{x}^*(t)$, $\boldsymbol{\lambda}^*(t)$, $u^*(t)$ die Stationaritätsbedingung $H_u(\mathbf{x}^*(t), \boldsymbol{\lambda}^*(t)) = 0$ nach wie vor gelten muss, auch wenn sie keine Informationen über die optimale Steuerung liefert. Folglich müssen die Zeitableitungen $\frac{d^r}{dt^r} H_u(\cdot)$ ebenfalls verschwinden. Von besonderem Interesse ist dabei die erste Zeitableitung, bei der u explizit erscheint, d.h.

$$\frac{d^r}{dt^r} H_u(\mathbf{x}^*(t), \boldsymbol{\lambda}^*(t)) = 0 \quad \text{mit} \quad \frac{\partial}{\partial u} \left(\frac{d^r}{dt^r} H_u(\cdot) \right) \neq 0.$$

Die Variable r stellt den *Grad der Singularität* dar. Die Gleichung $\frac{d^r}{dt^r} H_u(\cdot) = 0$ kann nun verwendet werden, um $u^*(t)$ zu berechnen.

Es ist offensichtlich, dass die Behandlung von singulären Problemen recht komplex und vor allem problemspezifisch ist. In der Praxis wird dieser Fall häufig umgangen, indem ein *Regularisierungsterm* $\sum_{i=1}^m r_i u_i^2$, siehe (5.57), mit hinreichend kleinen Gewichtung $r_i > 0$ zur Kostenfunktion hinzugefügt wird. Damit ist sicher gestellt, dass u_i in den einzelnen Stationaritätsbedingungen $\frac{\partial H}{\partial u_i} = 0$ nicht verschwindet.

5.3.5 Interpretation der adjungierten Variablen

Im Falle der statischen Optimierung wurde in Abschnitt 4.2.4 gezeigt, dass die Lagrange-Multiplikatoren λ_i und μ_i ein Maß für die Sensitivität der Kostenfunktion f am Optimum bezogen auf kleine Änderungen der jeweiligen Beschränkungen darstellen.

Eine ähnliche Interpretation lässt sich bei Optimalsteuerungsproblemen für die adjungierten Variablen $\boldsymbol{\lambda}^*(t)$ herleiten. Um dies zu betrachten, wird das folgende Optimierungsproblem ohne Endbedingungen, ohne Endgewichtung und mit fester Endzeit t_f betrachtet:⁵

$$\min_{\mathbf{u}(\cdot)} \int_{t_0}^{t_f} l(\mathbf{x}, \mathbf{u}, t) dt \quad (5.58a)$$

$$\text{u.B.v.} \quad \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t), \quad \mathbf{x}(t_0) = \mathbf{x}_0. \quad (5.58b)$$

Die optimale Lösung sei $\mathbf{x}^*(t)$, $\mathbf{u}^*(t)$ und stetig auf $[t_0, t_f]$. Des Weiteren wird angenommen, dass die Funktionen f und l stetig differenzierbar sind.

Im Folgenden werden kleine Abweichungen $\boldsymbol{\varepsilon}$ von den Anfangsbedingungen $\mathbf{x}(t_0) = \mathbf{x}_0 + \boldsymbol{\varepsilon}$ betrachtet, für die die zugehörigen optimalen Trajektorien mit $\tilde{\mathbf{x}}(t; \boldsymbol{\varepsilon})$ und $\tilde{\mathbf{u}}(t; \boldsymbol{\varepsilon})$ bezeichnet werden. Es ist direkt ersichtlich, dass

$$\tilde{\mathbf{x}}(t; \mathbf{0}) = \mathbf{x}^*(t), \quad \tilde{\mathbf{u}}(t; \mathbf{0}) = \mathbf{u}^*(t), \quad t \in [t_0, t_f] \quad (5.59)$$

⁵Dieser Abschnitt orientiert sich an den entsprechenden Herleitungen in [7, 3].

gilt. Da die Systemgleichungen (5.58b) für die optimale Lösung $\tilde{\mathbf{x}}(t; \boldsymbol{\varepsilon})$, $\tilde{\mathbf{u}}(t; \boldsymbol{\varepsilon})$ erfüllt sind, d.h.

$$\dot{\tilde{\mathbf{x}}}(t; \boldsymbol{\varepsilon}) = \mathbf{f}(\tilde{\mathbf{x}}(t; \boldsymbol{\varepsilon}), \tilde{\mathbf{u}}(t; \boldsymbol{\varepsilon}), t), \quad \tilde{\mathbf{x}}(t_0; \boldsymbol{\varepsilon}) = \mathbf{x}_0 + \boldsymbol{\varepsilon}, \quad (5.60)$$

können sie mit Hilfe von adjungierten Variablen zur Kostenfunktion in (5.58a) addiert werden

$$\begin{aligned} V(\tilde{\mathbf{x}}(t_0; \boldsymbol{\varepsilon}), t_0) &= \int_{t_0}^{t_f} l(\tilde{\mathbf{x}}(t; \boldsymbol{\varepsilon}), \tilde{\mathbf{u}}(t; \boldsymbol{\varepsilon}), t) dt \\ &= \int_{t_0}^{t_f} l(\tilde{\mathbf{x}}(t; \boldsymbol{\varepsilon}), \tilde{\mathbf{u}}(t; \boldsymbol{\varepsilon}), t) + \boldsymbol{\lambda}^*(t)^\top [\mathbf{f}(\tilde{\mathbf{x}}(t; \boldsymbol{\varepsilon}), \tilde{\mathbf{u}}(t; \boldsymbol{\varepsilon}), t) - \dot{\tilde{\mathbf{x}}}(t; \boldsymbol{\varepsilon})] dt. \end{aligned}$$

Dabei stellen $\boldsymbol{\lambda}^*(t)$ die zur nominellen optimalen Lösung $\mathbf{x}^*(t)$, $\mathbf{u}^*(t)$ gehörenden adjungierten Zustände dar. Unter der Annahme, dass $\tilde{\mathbf{x}}(t; \boldsymbol{\varepsilon})$ und $\tilde{\mathbf{u}}(t; \boldsymbol{\varepsilon})$ bzgl. $\boldsymbol{\varepsilon}$ stetig differenzierbar sind, folgt mittels partieller Integration und anschließendem Differenzieren nach $\boldsymbol{\varepsilon}$ (ohne Argumente)

$$\begin{aligned} \frac{d}{d\boldsymbol{\varepsilon}} V(\tilde{\mathbf{x}}(t_0; \boldsymbol{\varepsilon}), t_0) &= -[\tilde{\mathbf{x}}_\boldsymbol{\varepsilon}^\top \boldsymbol{\lambda}^*]_{t_0}^{t_f} \\ &\quad + \int_{t_0}^{t_f} \tilde{\mathbf{x}}_\boldsymbol{\varepsilon}^\top \left[\left(\frac{\partial l}{\partial \mathbf{x}} \right)_{\tilde{\mathbf{x}}, \tilde{\mathbf{u}}} + \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right)_{\tilde{\mathbf{x}}, \tilde{\mathbf{u}}}^\top \boldsymbol{\lambda}^* + \dot{\boldsymbol{\lambda}}^* \right] + \tilde{\mathbf{u}}_\boldsymbol{\varepsilon}^\top \left[\left(\frac{\partial l}{\partial \mathbf{u}} \right)_{\tilde{\mathbf{x}}, \tilde{\mathbf{u}}} + \left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right)_{\tilde{\mathbf{x}}, \tilde{\mathbf{u}}}^\top \boldsymbol{\lambda}^* \right] dt \end{aligned}$$

wobei die partiellen Ableitungen an der Stelle $\tilde{\mathbf{x}}(t; \boldsymbol{\varepsilon})$, $\tilde{\mathbf{u}}(t; \boldsymbol{\varepsilon})$ ausgewertet sind. Die Sensitivitäten $\tilde{\mathbf{x}}_\boldsymbol{\varepsilon}(t; \boldsymbol{\varepsilon}) = \frac{\partial \tilde{\mathbf{x}}}{\partial \boldsymbol{\varepsilon}}$ und $\tilde{\mathbf{u}}_\boldsymbol{\varepsilon}(t; \boldsymbol{\varepsilon}) = \frac{\partial \tilde{\mathbf{u}}}{\partial \boldsymbol{\varepsilon}}$ stellen Matrizen dar. Somit ist die obige Gleichung eine Vektorgleichung, in der das Integral elementweise zu verstehen ist.

Für $\boldsymbol{\varepsilon} = \mathbf{0}$ ergibt sich die nominelle optimale Lösung (5.59), womit die vorherige Gleichung wie folgt geschrieben werden kann:

$$\begin{aligned} \frac{d}{d\boldsymbol{\varepsilon}} V(\tilde{\mathbf{x}}(t_0; \boldsymbol{\varepsilon}), t_0) \Big|_{\boldsymbol{\varepsilon}=\mathbf{0}} &= -\tilde{\mathbf{x}}_\boldsymbol{\varepsilon}(t_f; \mathbf{0})^\top \boldsymbol{\lambda}^*(t_f) + \tilde{\mathbf{x}}_\boldsymbol{\varepsilon}(t_0; \mathbf{0})^\top \boldsymbol{\lambda}^*(t_0) \\ &\quad + \int_{t_0}^{t_f} \tilde{\mathbf{x}}_\boldsymbol{\varepsilon}(t; \mathbf{0})^\top \left[\left(\frac{\partial l}{\partial \mathbf{x}} \right)_{\mathbf{x}^*, \mathbf{u}^*} + \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right)_{\mathbf{x}^*, \mathbf{u}^*}^\top \boldsymbol{\lambda}^* + \dot{\boldsymbol{\lambda}}^* \right] + \tilde{\mathbf{u}}_\boldsymbol{\varepsilon}(t; \mathbf{0})^\top \left[\left(\frac{\partial l}{\partial \mathbf{u}} \right)_{\mathbf{x}^*, \mathbf{u}^*} + \left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right)_{\mathbf{x}^*, \mathbf{u}^*}^\top \boldsymbol{\lambda}^* \right] dt. \end{aligned}$$

Da das Tripel $(\mathbf{x}^*, \mathbf{u}^*, \boldsymbol{\lambda}^*)$ aber die notwendigen Optimalitätsbedingungen (5.38)–(5.39) erfüllt und $\boldsymbol{\lambda}^*(t_f) = \mathbf{0}$ ist, vereinfacht sich diese Gleichung zu

$$\frac{d}{d\boldsymbol{\varepsilon}} V(\tilde{\mathbf{x}}(t_0; \boldsymbol{\varepsilon}), t_0) \Big|_{\boldsymbol{\varepsilon}=\mathbf{0}} = \tilde{\mathbf{x}}_\boldsymbol{\varepsilon}(t_0; \mathbf{0})^\top \boldsymbol{\lambda}^*(t_0) \quad \Rightarrow \quad V_\mathbf{x}(\mathbf{x}_0, t_0) = \boldsymbol{\lambda}^*(t_0) \quad (5.61)$$

mit $V_\mathbf{x} = \frac{\partial V}{\partial \mathbf{x}}$. In der letzten Beziehung wurde ausgenutzt, dass an der Stelle $t = t_0$ die Sensitivität $\tilde{\mathbf{x}}_\boldsymbol{\varepsilon}(t_0; \boldsymbol{\varepsilon})$ genau die Einheitsmatrix $\mathbf{I}$ der Ordnung n ist, wie aus (5.60) leicht ersichtlich ist.

Aus (5.61) erkennt man, dass die adjungierten Variablen $\boldsymbol{\lambda}^*(t_0)$ zur Anfangszeit $t = t_0$ ein Maß für die Sensitivität des Kostenfunktionalen bezüglich Änderungen in den Anfangsbedingungen $\mathbf{x}_0$ darstellen. Dieser Zusammenhang ist in den Abbildungen 5.5, 5.6 der Aufgaben 5.2 und 5.3 gut erkennbar.

Der bisher gefundene Zusammenhang für die Startzeit $t = t_0$ kann leicht auf das Gesamtintervall $[t_0, t_f]$ erweitert werden, indem das *Optimalitätsprinzip* zu Hilfe gezogen wird.

Satz 5.2 (Optimalitätsprinzip) Falls $\mathbf{u}^*(t)$, $t \in [t_0, t_f]$ eine optimale Steuerung des Problems (5.58) mit der zugehörigen Zustandstrajektorie $\mathbf{x}^*(t)$ ist, so ist $\mathbf{u}^*(t)$, $t \in [t_1, t_f]$ für jedes Restintervall $t_1 \in [t_0, t_f]$ die optimale Steuerung für das Problem

$$\min_{\mathbf{u}(\cdot)} \int_{t_1}^{t_f} l(\mathbf{x}, \mathbf{u}, t) dt \quad (5.62a)$$

$$\text{u.B.v. } \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t), \quad \mathbf{x}(t_1) = \mathbf{x}^*(t_1). \quad (5.62b)$$

Aufgrund des Optimalitätsprinzips kann die Herleitung der Beziehung (5.61) auf das “abgeschnittene” Problem (5.62) übertragen werden, d.h. $V_{\mathbf{x}}(\mathbf{x}^*(t_1), t_1) = \boldsymbol{\lambda}^*(t_1)$. Da aber t_1 auf dem Intervall $[t_0, t_f]$ frei gewählt werden kann, gilt folglich

$$\boxed{V_{\mathbf{x}}(\mathbf{x}^*(t), t) = \boldsymbol{\lambda}^*(t), \quad t \in [t_0, t_f].} \quad (5.63)$$

Die adjungierten Zustände $\boldsymbol{\lambda}^*(t)$ sind also ein Maß dafür, wie stark sich das Kostenfunktional bei einer kleinen Störung des Zustandes $\mathbf{x}^*(t)$ zur Zeit t ändern würde, wenn die Steuerung auf dem Restintervall $[t, t_f]$ optimal wäre.

5.3.6 Erweiterung auf allgemeine Randbedingungen

Der bisher betrachtete Fall der partiellen Endbedingungen ist die in der Praxis am häufigsten vorkommende Form von Endbedingungen und spiegelt die Dualität zwischen den Endbedingungen (5.40b) und (5.41) für die Systemzustände $\mathbf{x}$ und die adjungierten Zustände $\boldsymbol{\lambda}$ wider.

Die Herleitung der Optimalitätsbedingungen (Abschnitt 5.3.2) über die Variationsrechnung erlaubt ebenfalls die Berücksichtigung von allgemeinen Endbedingungen

$$\mathbf{g}(\mathbf{x}(t_f), t_f) = \mathbf{0}, \quad \mathbf{g} : \mathbb{R}^n \times \mathbb{R}_+ \rightarrow \mathbb{R}^r. \quad (5.64)$$

Analog zum Vorgehen in Abschnitt 5.3.1.1 werden die Endbedingungen (5.64) über zusätzliche Lagrange-Multiplikatoren $\boldsymbol{\nu} \in \mathbb{R}^r$ zu dem erweiterten Kostenfunktional (5.29) hinzugefügt

$$\bar{J} = \varphi(\mathbf{x}(t_f)) + \boldsymbol{\nu}^\top \mathbf{g}(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} l(\mathbf{x}(t), \mathbf{u}(t), t) + \boldsymbol{\lambda}^\top (\mathbf{f}(\mathbf{x}, \mathbf{u}, t) - \dot{\mathbf{x}}) dt. \quad (5.65)$$

Durch Anwendung der Variationsrechnung lässt sich zeigen, dass sich die Transversalitätsbedingungen (5.41), (5.42) durch den zusätzlichen Term $\boldsymbol{\nu}^\top \mathbf{g}(\mathbf{x}(t_f), t_f)$ entsprechend erweitern:

$$\boldsymbol{\lambda}(t_f) = \left[\frac{\partial \varphi}{\partial \mathbf{x}} + \left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}} \right)^\top \boldsymbol{\nu} \right]_{t=t_f} \quad (5.66)$$

$$H(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}, t)|_{t=t_f} = - \left[\frac{\partial \varphi}{\partial t} + \left(\frac{\partial \mathbf{g}}{\partial t} \right)^\top \boldsymbol{\nu} \right]_{t=t_f} \quad \text{falls } t_f \text{ frei.} \quad (5.67)$$

Im Falle einer festen Endzeit umfasst das Randwertproblem der Optimalitätsbedingungen die $2n$ Differentialgleichungen (5.37), (5.38), $2n+r$ Randbedingungen (5.40a), (5.64), (5.66) und die m algebraischen Gleichungen (5.39) für die $2n$ Zustände $\mathbf{x}(t)$, $\boldsymbol{\lambda}(t)$ und die r (konstanten) Multiplikatoren $\boldsymbol{\nu}$. Falls die Endzeit frei ist, kommt noch die Transversalitätsbedingung (5.67) zur Bestimmung von t_f hinzu.

5.4 Optimierung linearer Systeme mit quadratischem Kostenfunktional

Ein großes Anwendungsfeld der Optimalsteuerungstheorie sind lineare Systeme mit quadratischem Kostenfunktional

$$\min_{\mathbf{u}(\cdot)} J(\mathbf{u}) = \frac{1}{2} \mathbf{x}(t_f)^\top \mathbf{S} \mathbf{x}(t_f) + \frac{1}{2} \int_{t_0}^{t_f} \mathbf{x}(t)^\top \mathbf{Q} \mathbf{x}(t) + \mathbf{u}(t)^\top \mathbf{R} \mathbf{u}(t) dt \quad (5.68a)$$

$$\text{u.B.v. } \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \quad \mathbf{x}(t_0) = \mathbf{x}_0. \quad (5.68b)$$

Der Endzustand $\mathbf{x}(t_f)$ sei frei und die Gewichtungsmatrizen $\mathbf{S}$, $\mathbf{Q}$ und $\mathbf{R}$ seien positiv definit.⁶ Es wird der Einfachheit halber zunächst angenommen, dass die Systemmatrizen $\mathbf{A}$ und $\mathbf{B}$ sowie die Gewichtungsmatrizen $\mathbf{Q}$, $\mathbf{R}$, $\mathbf{S}$ zeitinvariant sind. Allerdings sind die Betrachtungen in diesem Abschnitt direkt auf den zeitvarianten Fall übertragbar.

Das Optimalsteuerungsproblem (5.68) kann wie folgt interpretiert werden: Bringe das System (5.68b) im Zeitintervall $[t_0, t_f]$ ausgehend von $\mathbf{x}(t_0) = \mathbf{x}_0$ möglichst nahe an den Ursprung $\mathbf{x}(t_f) \approx \mathbf{0}$ unter Verwendung von “akzeptablen” Steuer- und Zustandstrajektorien $\mathbf{u}(t)$ und $\mathbf{x}(t)$. Die Gewichtungsmatrizen $\mathbf{Q}$, $\mathbf{R}$ und $\mathbf{S}$ bestimmen dabei das Zustands- und Eingangsverhalten des Systems (siehe Aufgaben 5.2 und 5.3).

5.4.1 Riccati-Differentialgleichung

Im Folgenden sollen die im vorherigen Kapitel hergeleiteten Optimalitätsbedingungen auf das linear-quadratische Problem angewendet werden. Mit der Hamilton-Funktion

$$H(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}, t) = \frac{1}{2} (\mathbf{x}^\top \mathbf{Q} \mathbf{x} + \mathbf{u}^\top \mathbf{R} \mathbf{u}) + \boldsymbol{\lambda}^\top (\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u})$$

lauten die Optimalitätsbedingungen (5.37)–(5.41)

$$\dot{\mathbf{x}} = \frac{\partial H}{\partial \boldsymbol{\lambda}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \quad \mathbf{x}(t_0) = \mathbf{x}_0 \quad (5.69a)$$

$$\dot{\boldsymbol{\lambda}} = -\frac{\partial H}{\partial \mathbf{x}} = -\mathbf{Q}\mathbf{x} - \mathbf{A}^\top \boldsymbol{\lambda}, \quad \boldsymbol{\lambda}(t_f) = \mathbf{S}\mathbf{x}(t_f) \quad (5.69b)$$

$$\mathbf{0} = \frac{\partial H}{\partial \mathbf{u}} = \mathbf{R}\mathbf{u} + \mathbf{B}^\top \boldsymbol{\lambda}. \quad (5.69c)$$

Aus (5.69c) folgt die optimale Steuerung

$$\mathbf{u}^*(t) = -\mathbf{R}^{-1} \mathbf{B}^\top \boldsymbol{\lambda}^*(t), \quad (5.70)$$

womit sich die Differentialgleichungen (5.69a), (5.69b) wie folgt schreiben lassen

$$\begin{bmatrix} \dot{\mathbf{x}}^* \\ \dot{\boldsymbol{\lambda}}^* \end{bmatrix} = \begin{bmatrix} \mathbf{A} & -\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^\top \\ -\mathbf{Q} & -\mathbf{A}^\top \end{bmatrix} \begin{bmatrix} \mathbf{x}^* \\ \boldsymbol{\lambda}^* \end{bmatrix}, \quad \begin{array}{l} \mathbf{x}^*(t_0) = \mathbf{x}_0 \\ \boldsymbol{\lambda}^*(t_f) = \mathbf{S}\mathbf{x}^*(t_f) \end{array}. \quad (5.71)$$

Es handelt sich also um eine Zwei-Punkt-Randwertaufgabe, die nicht ohne Weiteres per Integration gelöst werden kann. Es kann allerdings mit Hilfe der zu (5.71) gehörenden

⁶Für $\mathbf{S}$ und $\mathbf{Q}$ würde Positiv-Semidefinitheit ausreichen.

Transitionsmatrix gezeigt werden, dass $\mathbf{x}^*(t)$ und $\boldsymbol{\lambda}^*(t)$ über eine zeitvariante Matrix $\mathbf{P}(t)$ mit der Endbedingung $\mathbf{P}(t_f) = \mathbf{S}$ verknüpft sind, d.h.

$$\boldsymbol{\lambda}^*(t) = \mathbf{P}(t)\mathbf{x}^*(t), \quad t \in [t_0, t_f]. \quad (5.72)$$

Durch Differenzieren dieser Beziehung, $\dot{\boldsymbol{\lambda}}^* = \dot{\mathbf{P}}\mathbf{x}^* + \mathbf{P}\dot{\mathbf{x}}^*$, und Einsetzen in die Differentialgleichungen (5.71) ergibt sich

$$\left[\dot{\mathbf{P}} + \mathbf{P}\mathbf{A} + \mathbf{A}^\top \mathbf{P} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^\top \mathbf{P} + \mathbf{Q} \right] \mathbf{x}^*(t) = \mathbf{0}.$$

Damit diese Beziehung erfüllt ist, ist es nicht nur *hinreichend*, dass der Klammerinhalt verschwindet, sondern auch *notwendig*, da die Gleichung für *alle* möglichen Anfangswerte $\mathbf{x}_0$ und damit für alle möglichen optimalen Trajektorien $\mathbf{x}^*(t)$ gelten muss.

Somit erhält man für $\mathbf{P}(t)$ die sogenannte *Matrix-Riccati-Differentialgleichung*

$$\dot{\mathbf{P}} = -\mathbf{P}\mathbf{A} - \mathbf{A}^\top \mathbf{P} + \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^\top \mathbf{P} - \mathbf{Q}, \quad \mathbf{P}(t_f) = \mathbf{S}. \quad (5.73)$$

Charakteristisch ist, dass die Riccati-Gleichung *nichtlinear* ist, da $\mathbf{P}(t)$ im vorletzten Term quadratisch auftritt. Einige Anmerkungen zur Riccati-Dgl. sind im Folgenden zusammengetragen:

- Bei der Lösung der Riccati-Dgl. (5.73) geht man aufgrund der nicht vorhandenen Anfangsbedingung zweckmäßigerweise von $t = t_f$ aus und integriert *rückwärts* in der Zeit bis $t = t_0$.
- $\mathbf{P}(t)$ ist positiv semi-definit und symmetrisch. Somit ist die Riccati-Dgl. der Ordnung n^2 aufgrund der Symmetrie nur $n(n+1)/2$ (untereinander verknüpften) Differentialgleichungen äquivalent.
- Im Falle von zeitvarianten Systemmatrizen $\mathbf{A}(t)$, $\mathbf{B}(t)$ und/oder Gewichtungsmatrizen $\mathbf{Q}(t)$, $\mathbf{R}(t)$ ändert sich nichts an der Herleitung.

Durch Bestimmung von $\mathbf{P}(t)$ können die optimalen Trajektorien $\mathbf{x}^*(t)$ und $\boldsymbol{\lambda}^*(t)$ aus (5.71) und (5.72) durch Vorwärtsintegration mit der zusätzliche Anfangsbedingung $\boldsymbol{\lambda}^*(t_0) = \mathbf{P}(t_0)\mathbf{x}^*(t_0)$ gewonnen werden. Anschließend ergibt sich die optimale Steuerung $\mathbf{u}^*(t)$ aus (5.70). Von besonderem Interesse ist die Riccati-Dgl. aber vor allem im Rahmen der *optimalen Regelung*, da $\mathbf{P}(t)$ direkt im optimalen Regelgesetz

$$\mathbf{u}(t) = -\mathbf{K}(t)\mathbf{x}(t) \quad \text{mit} \quad \mathbf{K}(t) = \mathbf{R}^{-1}\mathbf{B}^\top \mathbf{P}(t) \quad (5.74)$$

und den rückgeführten Zuständen $\mathbf{x}(t)$ verwendet werden kann. Die Matrix $\mathbf{K}(t)$ stellt dabei die zeitvariante *Rückführmatrix* dar.

5.4.2 Beispiel: Zwei-Freiheitsgrade-Regelung zum Aufschwingen des Einfachpendels

Ein Anwendungsgebiet für die Riccati-Differentialgleichung ist die sogenannte *Zwei-Freiheitsgrade-Regelungsstruktur* (2FHG), die in der Praxis häufig zum Entwurf von Folge-regelungen eingesetzt wird, siehe Abbildung 5.10. Die Idee der 2FHG-Struktur ist die

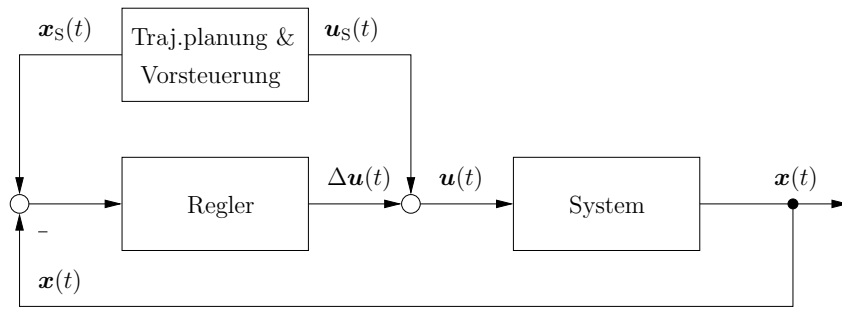


Abbildung 5.10: Regelungsstruktur mit zwei Freiheitsgraden.

Entkopplung von Führungs- und Störungsverhalten, indem eine Steuertrajektorie $\mathbf{u}_S(t)$ auf das System geschaltet wird, die das System im nominellen Fall exakt auf einer gewünschten Zustandstrajektorie $\mathbf{x}_S(t)$ führt. Die Regelung stabilisiert das System entlang von $\mathbf{x}_S(t)$ und ist für die Ausregelung von Modellfehlern und Störungen zuständig.

Beim 2FHG-Entwurf kann das betrachtete System $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$ um die Referenztrajektorie linearisiert werden, was zu einem linearen zeitvarianten System

$$\Delta \dot{\mathbf{x}} = \mathbf{A}(t)\Delta \mathbf{x} + \mathbf{B}(t)\Delta \mathbf{u} \quad \text{mit} \quad \mathbf{A}(t) = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{x}_S(t), \mathbf{u}_S(t)}, \quad \mathbf{B}(t) = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right|_{\mathbf{x}_S(t), \mathbf{u}_S(t)} \quad (5.75)$$

mit $\Delta \mathbf{x} = \mathbf{x} - \mathbf{x}_S$ und $\Delta \mathbf{u} = \mathbf{u} - \mathbf{u}_S$ führt, das für den Reglerentwurf herangezogen werden kann. Die Riccati-Differentialgleichung kann an dieser Stelle verwendet werden, um ein optimales Rückführgesetz der Form (5.74) zu berechnen.

Im Folgenden soll dieses Vorgehen und die Lösung der Riccati-Dgl. (5.73) am Beispiel des in Abbildung 5.11 dargestellten Einfachpendels demonstriert werden. Ein einfaches nichtlineares Modell für die Wagenposition y und den Pendelwinkel ϕ ist ⁷

$$\ddot{y} = u, \quad \ddot{\phi} = \frac{3}{2l} [g \sin(\phi) - \cos(\phi)u] \quad (5.76)$$

mit der Erdbeschleunigung g und der Pendellänge l . Als Eingang u dient die Beschleunigung des Wagens $\ddot{y}$.

Die betrachtete Folgeregelungsaufgabe ist das Aufschwingen des Pendels aus der unteren in die obere (invertierte) Ruhelage, d.h.

$$\left. \begin{array}{l} y(0) = 0, \quad \dot{y}(0) = 0 \\ \phi(0) = -\pi, \quad \dot{\phi}(0) = 0 \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} y(t_f) = 0, \quad \dot{y}(t_f) = 0 \\ \phi(t_f) = 0, \quad \dot{\phi}(t_f) = 0. \end{array} \right. \quad (5.77)$$

Eine mögliche Steuertrajektorie zum Aufschwingen eines Pendels mit der Länge $l = 1$ m in der Zeit $t_f = 2$ s ist

$$u_S(t) = \sum_{i=1}^5 c_i t^i, \quad t \in [0, t_f = 2 \text{ s}] \quad (5.78)$$

⁷Dieses Modell kann über den Lagrange-Formalismus unter Vernachlässigung von Reibung hergeleitet werden, wobei für das Pendel das Trägheitsmoment eines Stabes $J = \frac{1}{12}ml^2$ mit der Pendelmasse m angesetzt wurde. Da sich m aus den Gleichungen kürzt, ist die Pendeldynamik (5.76) unabhängig von der Masse des Pendels.

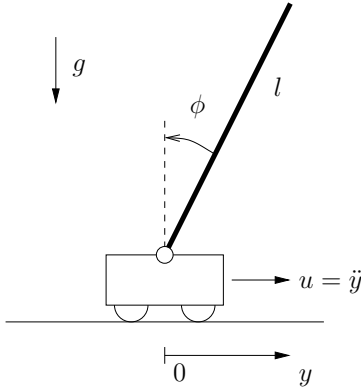


Abbildung 5.11: Aufbau des Pendels.

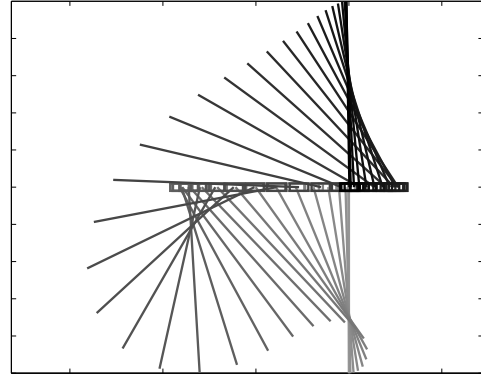


Abbildung 5.12: Momentaufnahmen des Aufschwingens.

mit dem Koeffizientenvektor $\mathbf{c} = [-135.128, 625.696, -888.598, 497.538, -96.3861]^\top$ (ohne Einheiten). Diese Steuerung wurde mit Hilfe der in [5] entwickelten Methodik numerisch berechnet, auf die an dieser Stelle aber nicht näher eingegangen werden soll. Abbildung 5.12 veranschaulicht die Aufschwingbewegung des Pendels. Die zugehörigen Zustandstrajektorien $\mathbf{x}_s(t)$ mit

$$\mathbf{x} = [y, \dot{y}, \phi, \dot{\phi}]^\top$$

können durch Integration der Bewegungsgleichungen (5.76) bestimmt werden und sind in Abbildung 5.14 dargestellt (rote Kurven).

Um zusätzlich zu der Steuerung (5.78) eine optimale Regelung zu entwerfen, wird das nichtlineare Modell (5.76) um die Solltrajektorien $\mathbf{x}_s(t)$ und $\mathbf{u}_s(t)$ linearisiert, was zu dem linearen zeitvarianten Modell (5.75)

$$\Delta \dot{\mathbf{x}} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{3}{2l}[g \cos \phi_s(t) + \sin \phi_s(t)u_s(t)] & 0 \end{bmatrix}}_{=\mathbf{A}(t)} \Delta \mathbf{x} + \underbrace{\begin{bmatrix} 0 \\ 1 \\ 0 \\ -\frac{3}{2l} \cos \phi_s(t) \end{bmatrix}}_{=\mathbf{b}(t)} \Delta u \quad (5.79)$$

mit $\mathbf{x}(t) = \mathbf{x}_s + \Delta \mathbf{x}(t)$ und $u(t) = u_s(t) + \Delta u(t)$ führt. Zur Auslegung des Riccati-Reglers wird das zu minimierende Kostenfunktional (5.68a)

$$J(\Delta u) = \frac{1}{2} \Delta \mathbf{x}(t_f)^\top \mathbf{S} \Delta \mathbf{x}(t_f) + \frac{1}{2} \int_0^{t_f} \Delta \mathbf{x}(t)^\top \mathbf{Q} \Delta \mathbf{x}(t) + R \Delta u(t)^2 dt, \quad t_f = 2s \quad (5.80)$$

mit den Diagonalmatrizen $\mathbf{S} = \mathbf{Q} = \text{diag}(1, 1, 1, 1)$ und $R = 0.01$ angesetzt. Der Eingang wird also im Vergleich zu den Zuständen eher gering bestraft, was einer schnellen Reglerauslegung entspricht.

Die Lösung der Riccati-Dgl. (5.73) muss numerisch erfolgen, um die Matrix $\mathbf{P}(t)$ auf dem Zeitintervall $t \in [0, t_f]$ zu erhalten. Das optimale Rückführgesetz für das Aufschwingen des Pendels ergibt sich nach (5.74) zu

$$\Delta u(t) = -\mathbf{k}(t)^\top \Delta \mathbf{x}(t) = \frac{1}{R} \mathbf{b}(t)^\top \mathbf{P}(t) [\mathbf{x}_s(t) - \mathbf{x}(t)]. \quad (5.81)$$

Dabei wird der Einfachheit halber angenommen, dass der komplette Zustand $\mathbf{x} = [y, \dot{y}, \phi, \dot{\phi}]^\top$ als Messinformation zur Verfügung steht. Wie aus Abbildung 5.10 ersichtlich ist, ergibt

sich die gesamte Stellgröße u als Kombination von Steuerung und Regelung

$$u(t) = u_S(t) + \Delta u(t). \quad (5.82)$$

Abbildung 5.13 zeigt den MATLAB-Code zur Lösung der Riccati-Dgl. (5.73), zur Berechnung des Rückführgesetzes (5.81) und zur Simulation des Aufschwingens mit einer Störung in den Anfangsbedingungen (5.77). Der Einfachheit halber wird die Riccati-Dgl. (5.73) für alle Elemente der Matrix $\mathbf{P}(t)$ numerisch integriert, obwohl diese (wie bereits erwähnt wurde) symmetrisch ist.

Die simulierten Trajektorien für das Aufschwingen in Abbildung 5.14 stellen den offenen Kreis (Simulation mit reiner Steuerung) und den geschlossenen Kreis (Steuerung und Regelung) dar. Gut erkennbar ist, dass das Pendel bei reiner Steuerung umfällt, während es im geschlossenen Kreis trotz der Anfangsstörung auf die Solltrajektorie $\mathbf{x}_S(t)$ gezwungen wird, was zu einer Korrektur der Steuerung $u_S(t)$ durch das Regelgesetz (5.81) führt.

Interessant ist ebenfalls, dass die Reglerverstärkungen $\mathbf{k}(t)$ in Abbildung 5.14 das Vorzeichen wechseln, was dadurch zu erklären ist, dass das Pendel von der stabilen unteren Ruhelage in die instabile obere Ruhelage überführt wird. Die Vorzeichenumkehr ist dabei ein Indikator dafür, dass das lineare zeitvariante System während dem Aufschwingen punktuell die Steuerbarkeit verliert. Bei der experimentellen Realisierung müsste deswegen überlegt werden, die Regelung in kritischen Zeitbereichen auszublenden und mit der Steuerung $u_S(t)$ zu durchfahren.

```

function vec = pendel
% -----
g = 9.81; l = 1;                                     % Pendelparameter
p.a1 = 3*g/(2*l); p.a2 = 3/(2*l); p.n=4; p.steps = 200;

p.tf = 2;                                             % Steuerung/Referenzgrößen
vec.t = linspace(0,p.tf,p.steps);                  % -----
c = [-135.128, 625.696, -888.598, 497.538, -96.3861]; % Eingangstrajektorie
vec.uS = c(1)*vec.t+c(2)*vec.t.^2+c(3)*vec.t.^3+c(4)*vec.t.^4+c(5)*vec.t.^5;
[tsim,xsim] = ode45(@pendel_sim,vec.t,[0;0;-pi;0],[],vec,p, 0); % Integration für Zustand
vec.xS = xsim';

p.Q = diag([1,1,1,1]); p.R = 0.01; p.S = diag([1,1,1,1]); % Gewichtungsmatrizen
S = reshape(p.S,p.n^2,1);                             % Matrix -> Vektor
[tsim,Psim] = ode45(@riccati_dgl,vec.t(end:-1:1),S,[],vec,p); % Riccati-Dgl.: Int. in Rückwärtszeit
vec.P = Psim(end:-1:1,:)' ;                            % -----

for i=1:length(vec.t),                                % zeitvariante Reglerverstärkungen
    [A,B] = matrizen(vec.t(i),vec,p);                % -----
    P = reshape(vec.P(:,i),p.n,p.n);
    vec.k(:,i) = inv(p.R)*B'*P;
end

X0 = [0;0;-pi+10*pi/180;2];
[tsim,x_ol] = ode45(@pendel_sim,vec.t,X0,[],vec,p, 0); % Simulation mit Anfangsstörung
[tsim,x_cl] = ode45(@pendel_sim,vec.t,X0,[],vec,p, 1); % offener Kreis (open-loop)
vec.x_ol = x_ol'; vec.x_cl = x_cl';                  % geschlossener Kreis (closed-loop)

for i=1:p.steps,                                     % Berechnung des Regelungsanteil
    vec.du(i) = vec.k(:,i)'*(vec.xS(:,i)-vec.x_cl(:,i));
end

function F = pendel_sim(t,x,vec,p,regelung)
% -----
y=x(1:2); phi=x(3:4);
uS = interp1(vec.t,vec.uS,t);
if regelung==0,                                     % Steuerung
    u = uS;                                           % Sim. im offenen Kreis
else                                                 % Sim. im geschlossenen Kreis
    xS = interp1(vec.t,vec.xS',t)';
    k = interp1(vec.t,vec.k',t);
    u = uS + k*(xS-x);
end
F = [ y(2); u; phi(2); p.a1*sin(phi(1))-p.a2*cos(phi(1))*u ]; % nichtlin. Pendelmodell

function Pp = riccati_dgl(t,Pvec,vec,p)
% -----
[A,B] = matrizen(t,vec,p);                          % zeitvar. Matrizen A und B
P = reshape(Pvec,p.n,p.n);                          % Vektor -> Matrix
Pp = -P*A - A'*P + P*B*inv(p.R)*B'*P - p.Q;          % Riccati-Dgl.
Pp = reshape(Pp,p.n^2,1);                            % Matrix -> Vektor

function [A,B] = matrizen(t,vec,p)
% -----
phiS = interp1(vec.t,vec.xS(3:4,:)',t)';             % momentane Referenzgrößen
uS = interp1(vec.t,vec.uS,t);
A = [0,1,0,0; 0,0,0,0; 0,0,0,1;                     % zeitvar. Matrizen A und B
      0,0,p.a1*cos(phiS(1))+p.a2*sin(phiS(1))*uS,0];
B = [0; 1; 0; -p.a2*cos(phiS(1))];

```

Abbildung 5.13: MATLAB-Code zur Berechnung der optimalen Rückführung (5.81) und zur Simulation des Aufschwingens des Pendels.

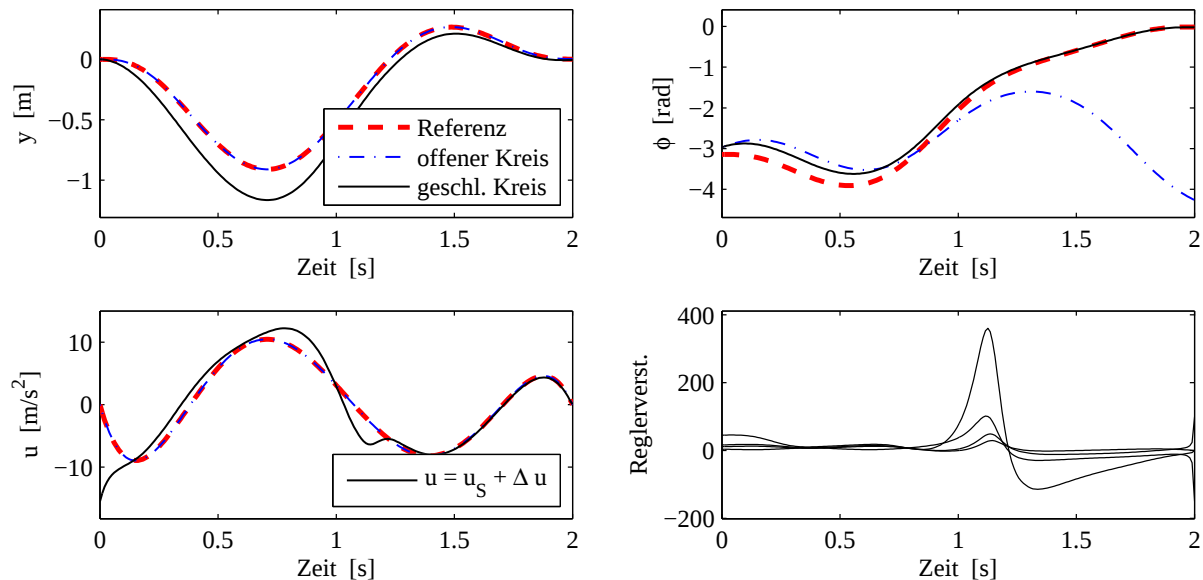


Abbildung 5.14: Simulationsergebnisse für das Aufschwingen des Einfachpendels.

5.4.3 Algebraische Riccati-Gleichung (Sonderfall $t_f \rightarrow \infty$)

Die Berechnung der optimalen Rückführung (5.74) wird besonders einfach, wenn ein unendliches Zeitintervall $[t_0, \infty)$ betrachtet wird. In diesem Fall macht die Endgewichtung $\mathbf{x}(t_f)^\top \mathbf{S} \mathbf{x}(t_f)$ keinen Sinn, da ohnehin $\lim_{t \rightarrow \infty} \mathbf{x}(t) = \mathbf{0}$ gelten muss, damit das Integral im Kostenfunktional (5.68a) existiert. Es wird also im Folgenden $\mathbf{S} = \mathbf{0}$ gesetzt. Des Weiteren wird angenommen, dass

- die Matrizen $\mathbf{A}$, $\mathbf{B}$, $\mathbf{Q}$ und $\mathbf{R}$ in (5.68) zeitinvariant sind,
- $\mathbf{Q}$ positiv semi-definit und $\mathbf{R}$ positiv definit ist,
- das System (5.68b) stabilisierbar ist,
- das Paar $[\mathbf{A}, \mathbf{C}]$ detektierbar ist, wobei die Matrix $\mathbf{C}$ aus der Zerlegung $\mathbf{C}^\top \mathbf{C} = \mathbf{Q}$ (z.B. mittels der Cholesky-Zerlegung) resultiert.⁸

Unter diesen Annahmen kann gezeigt werden, dass die Lösung der Riccati-Dgl. (5.73) in Rückwärtszeit gegen eine stationäre positiv semi-definite Matrix $\bar{\mathbf{P}}$ konvergiert, siehe z.B. [8, 9]. Dabei ist $\bar{\mathbf{P}}$ unabhängig von dem Randwert $\mathbf{P}(t_f)$ bei dem die Rückwärtsintegration gestartet wird, solange $\mathbf{P}(t_f)$ positiv semi-definit ist. Somit kann zur näherungsweisen Bestimmung von $\bar{\mathbf{P}}$ die Riccati-Dgl. (5.73) ausgehend von einem beliebigen positiv semi-definiten Endpunkt $\mathbf{P}(t_f)$, z.B. $\mathbf{P}(t_f) = \mathbf{0}$, rückwärts integriert werden, bis die zeitliche Änderung $\dot{\mathbf{P}}(t)$ unterhalb eines Toleranzkriteriums liegt.

Eine elegantere Möglichkeit $\bar{\mathbf{P}}$ zu bestimmen, ist die Lösung der sogenannten *algebraischen Riccati-Gleichung*

$$\bar{\mathbf{P}}\mathbf{A} + \mathbf{A}^\top \bar{\mathbf{P}} - \bar{\mathbf{P}}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^\top \bar{\mathbf{P}} + \mathbf{Q} = \mathbf{0}, \quad (5.83)$$

⁸Eine anschauliche Erklärung der letzten beiden Annahmen kann in [9] nachgelesen werden.

die sich direkt aus (5.73) mit $\dot{\mathbf{P}}(t) = \mathbf{0}$ ergibt. Da die Gleichung nichtlinear ist, können mehrere Lösungen existieren. Es kann aber unter den obigen Annahmen gezeigt werden, dass nur *eine positiv semi-definite* Lösung $\bar{\mathbf{P}}$ existiert. Zur Berechnung von $\bar{\mathbf{P}}$ aus der algebraischen Riccati-Gleichung (5.83) können verschiedene Verfahren verwendet werden (z.B. das Verfahren nach *Kalman* und *Englar*), auf die an dieser Stelle nicht eingegangen werden soll.

Mit Hilfe des Optimalitätsprinzips in Satz 5.2 kann gezeigt werden, dass das nun zeitinvariante Rückführgesetz (5.74)

$$\mathbf{u}(t) = -\bar{\mathbf{K}}\mathbf{x}(t) \quad \text{mit} \quad \bar{\mathbf{K}} = \mathbf{R}^{-1}\mathbf{B}^\top \bar{\mathbf{P}}$$

tatsächlich die optimale Lösung des Problems (5.68) auf dem unendlichen Intervall $[t_0, \infty)$ liefert. Die Ergebnisse sind im folgenden Satz zusammengefasst.

Satz 5.3 (LQ-Regler für lineare zeitinvariante Systeme) *Gegeben sei das stabilisierbare lineare zeitinvariante System*

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \quad \mathbf{x}(t_0) = \mathbf{x}_0$$

mit dem zu minimierenden Kostenfunktional

$$J(\mathbf{u}) = \frac{1}{2} \int_{t_0}^{\infty} \mathbf{x}(t)^\top \mathbf{Q} \mathbf{x}(t) + \mathbf{u}(t)^\top \mathbf{R} \mathbf{u}(t) dt, \quad (5.84)$$

wobei $\mathbf{Q}$ positiv semi-definit und $\mathbf{R}$ positiv definit sind und das Paar $[\mathbf{A}, \mathbf{C}]$ mit $\mathbf{Q} = \mathbf{C}^\top \mathbf{C}$ detektierbar ist. Dann ergibt sich das Minimum von (5.84) durch das optimale Rückführgesetz

$$\mathbf{u}(t) = -\bar{\mathbf{K}}\mathbf{x}(t) \quad \text{mit} \quad \bar{\mathbf{K}} = \mathbf{R}^{-1}\mathbf{B}^\top \bar{\mathbf{P}}, \quad (5.85)$$

wobei $\bar{\mathbf{P}}$ die eindeutige positiv semi-definite Lösung der algebraischen Riccati-Gleichung (5.83) ist. Des Weiteren besitzen sämtliche Eigenwerte der Matrix $(\mathbf{A} - \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^\top \bar{\mathbf{P}})$ negative Realteile, so dass der geschlossene Kreis

$$\dot{\mathbf{x}} = [\mathbf{A} - \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^\top \bar{\mathbf{P}}] \mathbf{x}, \quad \mathbf{x}(t_0) = \mathbf{x}_0$$

asymptotisch stabil ist.

Der optimale Zustandsregler (5.85) wird häufig auch als *Riccati-Regler* oder *LQ-Regler* (“linear-quadratischer Regler”) bezeichnet und wird in der Praxis häufig eingesetzt. Im Gegensatz zum Polvorgabe-Verfahren können die Eigenwerte des geschlossenen Kreises nicht direkt vorgegeben werden. Dafür besitzen die Einträge in den Gewichtungsmatrizen $\mathbf{Q}$ und $\mathbf{R}$ (zumindest in der Diagonalen) eine anschauliche Bedeutung für das Verhalten des geschlossenen Kreises.

Unter MATLAB kann die optimale Rückführmatrix $\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^\top \bar{\mathbf{P}}$ mit Hilfe der Befehle `lqr` bzw. `lqrd` oder `dlqr` berechnet werden.

5.5 Optimalsteuerungsprobleme mit Eingangsbeschränkungen

Bei den bisherigen Betrachtungen mittels der Variationsrechnung wurde vorausgesetzt, dass das Optimalsteuerungsproblem (5.28) unbeschränkt ist. Im Folgenden sollen Beschränkungen der Steuergrößen

$$\mathbf{u}(t) \in U \subseteq \mathbb{R}^m \quad \forall t \in [t_0, t_f]$$

betrachtet werden, wobei U eine Untermenge des $\mathbb{R}^m$ darstellt und zu jedem Zeitpunkt t eingehalten werden soll. Eingangsbeschränkungen kommen in der Praxis am häufigsten vor und lassen sich mittels *Pontryagin's Maximumprinzip* berücksichtigen. Zustandsbeschränkungen sind im Rahmen der dynamischen Optimierung wesentlich schwieriger zu handhaben und sollen an dieser Stelle nicht weiter vertieft werden.

5.5.1 Pontryagin's Maximumprinzip

Das Maximumprinzip wurde 1956 von *L.S. Pontryagin* als Hypothese formuliert und in den folgenden Jahren von ihm und seinen Mitarbeitern *Boltjanski*, *Gamkrelidse* und *Mischenko* für zunehmend allgemeinere Systemklassen bewiesen.⁹

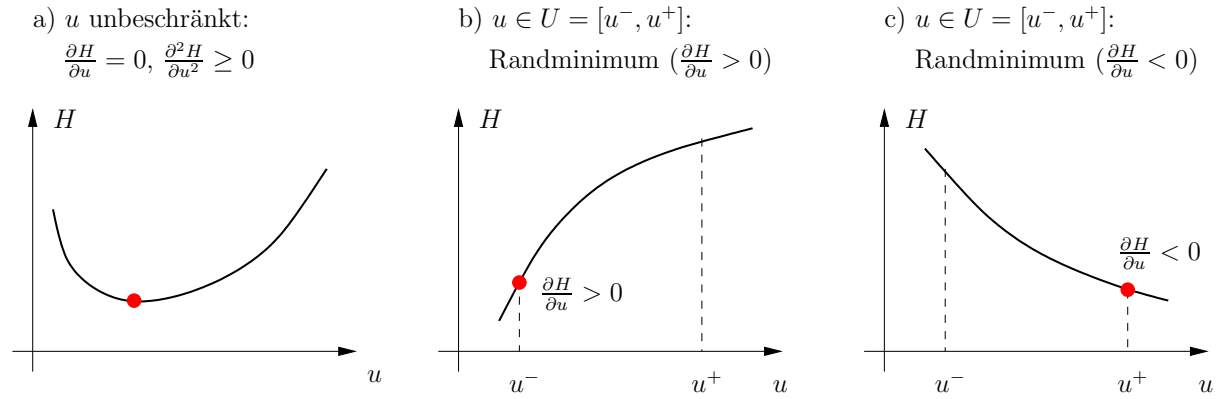
Pontryagin's Maximumprinzip stützt sich auf die Hamilton-Funktion und die kanonischen Gleichungen (5.69a), (5.69b). Allerdings verliert die Bedingung $\frac{\partial H}{\partial \mathbf{u}} = \mathbf{0}$ ihre Gültigkeit als notwendige Optimalitätsbedingung. Dieser Umstand ist in Abbildung 5.15 illustriert, wo für den Fall einer skalaren Eingangsgröße u die Hamilton-Funktion $H(\mathbf{x}^*, u, \boldsymbol{\lambda}^*, t)$ für feste (optimale) $\mathbf{x}^*(t)$, $\boldsymbol{\lambda}^*(t)$ zu einer festen Zeit t über u aufgetragen ist. Im unbeschränkten Fall a) stellen (5.39) und (5.43), also $\frac{\partial H}{\partial u} = 0$ und Positiv-Semi-Definitheit von $\frac{\partial^2 H}{\partial u^2}$, notwendige Bedingungen für ein *Minimum* der Hamilton-Funktion H zu jedem Zeitpunkt t dar. Falls u dagegen auf einem Intervall $U = [u^-, u^+]$ beschränkt ist, kann das Minimum auf dem Rand u^- oder u^+ liegen, wo nicht notwendigerweise $\frac{\partial H}{\partial u} = 0$ gilt, wie die Fälle b) und c) in Abbildung 5.15 zeigen.

Es ist also naheliegend, dass das Verschwinden der partiellen Ableitung von H nach $\mathbf{u}$ durch

$$H(\mathbf{x}^*, \mathbf{u}^*, \boldsymbol{\lambda}^*, t) = \min_{\mathbf{u} \in U} H(\mathbf{x}^*, \mathbf{u}, \boldsymbol{\lambda}^*, t)$$

zu ersetzen ist. Diese Bedingung stellt die Basis von Pontryagin's Maximumprinzip dar, das im Folgenden für autonome Systeme (d.h. l und $\mathbf{f}$ in (5.86) hängen nicht explizit von der Zeit t ab) angegeben wird.

⁹Das Maximumprinzip wurde ursprünglich für *Maximierungsprobleme* definiert, woher auch der Name stammt. Somit müsste es im hier betrachteten Fall korrekterweise *Minimumprinzip* heißen, wie es auch in der Literatur gelegentlich vorkommt. Angesichts seiner Bedeutung in der dynamischen Optimierung und der Verbindung mit dem Namen Pontryagin soll es im Folgenden weiterhin als *Maximumprinzip* bezeichnet werden.

Abbildung 5.15: Randminima im beschränkten Fall $u \in U = [u^-, u^+]$.

Satz 5.4 (Maximumprinzip für autonome Systeme) Gegeben sei das Optimalsteuerungsproblem

$$\min_{\mathbf{u}(\cdot)} \quad J(\mathbf{u}) = \int_{t_0}^{t_f} l(\mathbf{x}(t), \mathbf{u}(t)) dt \quad (5.86a)$$

$$\text{u.B.v.} \quad \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \quad (5.86b)$$

$$\mathbf{x}(t_0) = \mathbf{x}_0, \quad \mathbf{x}(t_f) = \mathbf{x}_f \quad (5.86c)$$

$$\mathbf{u}(t) \in U, \quad \forall t \in [t_0, t_f] \quad (5.86d)$$

mit freier Endzeit t_f . Die Funktionen l und $\mathbf{f}$ seien stetig in $(\mathbf{x}, \mathbf{u})$ und stetig differenzierbar in $\mathbf{x}$. Falls das Problem (5.86) eine optimale Lösung $\mathbf{x}^*(t)$, $\mathbf{u}^*(t)$, $t \in [t_0, t_f^*]$ besitzt, dann existiert eine nicht-negative Konstante $\lambda_0 \geq 0$ und n nicht-verschwindende stückweise stetig differenzierbare Zeitfunktionen $\boldsymbol{\lambda}^* = [\lambda_1^*, \dots, \lambda_n^*]^\top$, so dass Folgendes gilt:

a) $\mathbf{x}^*(t)$ und $\boldsymbol{\lambda}^*(t)$, $t \in [t_0, t_f^*]$ erfüllen die kanonischen Gleichungen

$$\dot{\mathbf{x}}^* = \frac{\partial H}{\partial \boldsymbol{\lambda}} = \mathbf{f}(\mathbf{x}^*, \mathbf{u}^*), \quad \dot{\boldsymbol{\lambda}}^* = -\frac{\partial H}{\partial \mathbf{x}} = -H_{\mathbf{x}}(\mathbf{x}^*, \mathbf{u}^*, \boldsymbol{\lambda}^*) \quad (5.87)$$

und die Randbedingungen (5.86c) mit der Hamilton-Funktion

$$H(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}, \lambda_0) = \lambda_0 l(\mathbf{x}, \mathbf{u}) + \boldsymbol{\lambda}^\top \mathbf{f}(\mathbf{x}, \mathbf{u}). \quad (5.88)$$

b) $H(\mathbf{x}^*(t), \mathbf{u}^*(t), \boldsymbol{\lambda}^*(t), \lambda_0^*)$ als Funktion von $\mathbf{u}$ besitzt ein absolutes Minimum an der Stelle $\mathbf{u}(t) = \mathbf{u}^*(t) \in U$ für alle $t \in [t_0, t_f^*]$:

$$H(\mathbf{x}^*(t), \mathbf{u}^*, \boldsymbol{\lambda}^*(t), \lambda_0^*) = \min_{\mathbf{u} \in U} H(\mathbf{x}^*(t), \mathbf{u}, \boldsymbol{\lambda}^*(t), \lambda_0^*) \quad \forall t \in [t_0, t_f^*]. \quad (5.89)$$

c) $H(\mathbf{x}^*(t), \mathbf{u}^*(t), \boldsymbol{\lambda}^*(t), \lambda_0^*)$ ist konstant für alle $t \in [t_0, t_f^*]$. Bei einer freien Endzeit t_f^* gilt außerdem die Transversalitätsbedingung

$$H(\mathbf{x}^*(t_f^*), \mathbf{u}^*(t_f^*), \boldsymbol{\lambda}^*(t_f^*), \lambda_0^*) = 0. \quad (5.90)$$

Das Maximumprinzip kann auf Probleme mit partiellen oder allgemeinen Endbedingungen erweitert werden, indem die Randbedingungen für die kanonischen Gleichungen in (5.87) analog zum unbeschränkten Fall durch (5.40b), (5.41) bzw. durch (5.64), (5.66) ersetzt werden.

Im nicht-autonomen Fall, d.h. $l = l(\mathbf{x}, \mathbf{u}, t)$ und $\mathbf{f} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t)$ ändert sich c) in Satz 5.4 dahingehend, dass H nicht mehr konstant ist, sondern die Differentialgleichung (5.44) erfüllen muss, d.h.

$$\frac{d}{dt}H = \lambda_0^* l_t(\mathbf{x}^*(t), \mathbf{u}^*(t), t) + \boldsymbol{\lambda}^*(t)^\top \mathbf{f}_t(\mathbf{x}^*(t), \mathbf{u}^*(t), t) \quad \forall t \in [t_0, t_f^*]$$

mit $l_t = \frac{\partial l}{\partial t}$ und $\mathbf{f}_t = \frac{\partial \mathbf{f}}{\partial t}$. Der zusätzliche Parameter λ_0 in der Hamilton-Funktion (5.88) berücksichtigt sogenannte *abnormale* Fälle von Optimalsteuerungsproblemen, für die $\lambda_0^* = 0$ gilt. Dieser Fall liegt z.B. vor, wenn die optimale Lösung unabhängig von der integrierten Kostenfunktion l in (5.86a) ist. Abnormale Probleme sind *nicht sinnvoll gestellt*, da ihre Lösung den Wert des zu minimierenden Kostenfunktional nicht beeinflusst.

Für sinnvoll gestellte Probleme gilt $\lambda_0^* > 0$. Es ist dabei gebräuchlich, $\lambda_0^* = 1$ zu setzen, womit sich die gleiche Hamilton-Funktion wie im unbeschränkten Fall ergibt.

Beispiel 5.3 (Abnormaler Fall) Gegeben sei das Problem

$$\begin{aligned} \min_{u(\cdot)} \quad & \int_0^1 l(x, u) dt \\ \text{u.B.v.} \quad & \dot{x} = u, \quad x(0) = 0, \quad x(1) = 1 \\ & u(t) \in [0, 1]. \end{aligned}$$

Es existiert nur eine Steuerung $u^*(t) = 1$, die den Zustand $x^*(t) = t$ von $x^*(0) = 0$ nach $x^*(1) = 1$ überführt. Somit ist die optimale Lösung unabhängig von der Wahl der Kostenfunktion $l(x, u)$ und es liegt ein abnormaler Fall vor.

5.5.2 Allgemeine Vorgehensweise

Die Anwendung des Maximumprinzips orientiert sich zum großen Teil am Vorgehen im unbeschränkten Fall (Abschnitt 5.3.4).¹⁰

1. Aufstellen der Hamilton-Funktion mit $\lambda_0 = 1$: $H(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}) = l(\mathbf{x}, \mathbf{u}) + \boldsymbol{\lambda}^\top \mathbf{f}(\mathbf{x}, \mathbf{u})$.
2. Lösung des Minimierungsproblems $\mathbf{u}^* = \arg \min_{\mathbf{u} \in U} H(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda})$ für jede mögliche Kombination $(\mathbf{x}, \boldsymbol{\lambda})$. Somit ergibt sich eine Funktion

$$\mathbf{u} = \boldsymbol{\psi}(\mathbf{x}, \boldsymbol{\lambda}) \tag{5.91}$$

in Abhängigkeit von $(\mathbf{x}, \boldsymbol{\lambda})$. Das Aufstellen dieser Funktion zur punktweisen Minimierung von H wird im Abschnitt 5.5.3 näher erläutert.

¹⁰In Analogie zu Satz 5.4 wird der autonome Fall betrachtet, was aber keine Einschränkung darstellt.

3. Einsetzen der Funktion (5.91) in die kanonischen Gleichungen (5.87) führt auf das Randwertproblem

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \psi(\mathbf{x}, \boldsymbol{\lambda})), \quad \dot{\boldsymbol{\lambda}} = -H_{\mathbf{x}}(\mathbf{x}, \psi(\mathbf{x}, \boldsymbol{\lambda}), \boldsymbol{\lambda}), \quad \text{RBen} \quad (5.86c).$$

Im Falle von partiellen oder nichtlinearen Randbedingungen werden $\mathbf{x}(t_0) = \mathbf{x}_0$ und $\mathbf{x}(t_f) = \mathbf{x}_f$ in (5.86c) entsprechend durch (5.40b), (5.41) bzw. (5.64), (5.66) ersetzt. Falls die Endzeit frei ist, kommt noch die Endbedingung (5.90) hinzu.

4. Die (numerische) Lösung des Randwertproblems liefert $\mathbf{x}^*(t)$, $\boldsymbol{\lambda}^*(t)$, womit sich die Steuertrajektorie $\mathbf{u}^*(t)$ aus (5.91) ergibt.

Analog zum unbeschränkten Fall in Abschnitt 5.3.4 sei erwähnt, dass das Maximumprinzip lediglich notwendige Bedingungen für eine optimale Lösung darstellt. Für bestimmte Aufgabenklassen kann das Maximumprinzip als hinreichend betrachtet werden, z.B. für die zeitoptimale Steuerung von zeitinvarianten linearen Systemen.

5.5.3 Minimierung der Hamilton-Funktion

Die Lösung des Minimierungsproblems (5.89) wird im Folgenden für einige praktisch relevante Fälle angegeben. Dabei wird davon ausgegangen, dass das bisher allgemein nichtlineare System (5.86b) in eingangsaffiner Form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) = \mathbf{f}_0(\mathbf{x}) + \sum_{i=1}^m \mathbf{f}_i(\mathbf{x}) u_i \quad (5.92)$$

vorliegt. Des Weiteren wird der gebräuchliche Fall betrachtet, dass alle Komponenten u_i des Eingangs $\mathbf{u} = [u_1, \dots, u_m]^\top$ beidseitig beschränkt sind, d.h.

$$\mathbf{u} \in U = [\mathbf{u}^-, \mathbf{u}^+] \quad \text{bzw.} \quad u_i \in [u_i^-, u_i^+], \quad i = 1, \dots, m. \quad (5.93)$$

5.5.3.1 Kostenfunktional mit verbrauchsoptimalem Anteil

Falls Verbrauchsoptimalität gewünscht wird, kann das Kostenfunktional in (5.86a) in der Form

$$J(\mathbf{u}) = \int_{t_0}^{t_f} l_0(\mathbf{x}) + \sum_{i=1}^m r_i |u_i| \, dt, \quad r_i > 0 \quad (5.94)$$

angesetzt werden. Mit $l_0(\mathbf{x}) = 0$ stellt die Minimierung der *Stellfläche* bzw. des *Verbrauchs* $\sum_{i=1}^m r_i |u_i|$ das Optimierungsziel dar. Für $l_0(\mathbf{x}) \neq 0$ kann der verbrauchsoptimale Anteil über die Koeffizienten $r_i > 0$ gewichtet werden.

Die Hamilton-Funktion lautet

$$H(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}) = l_0(\mathbf{x}) + \sum_{i=1}^m r_i |u_i| + \boldsymbol{\lambda}^\top \left(\mathbf{f}_0(\mathbf{x}) + \sum_{i=1}^m \mathbf{f}_i(\mathbf{x}) u_i \right)$$

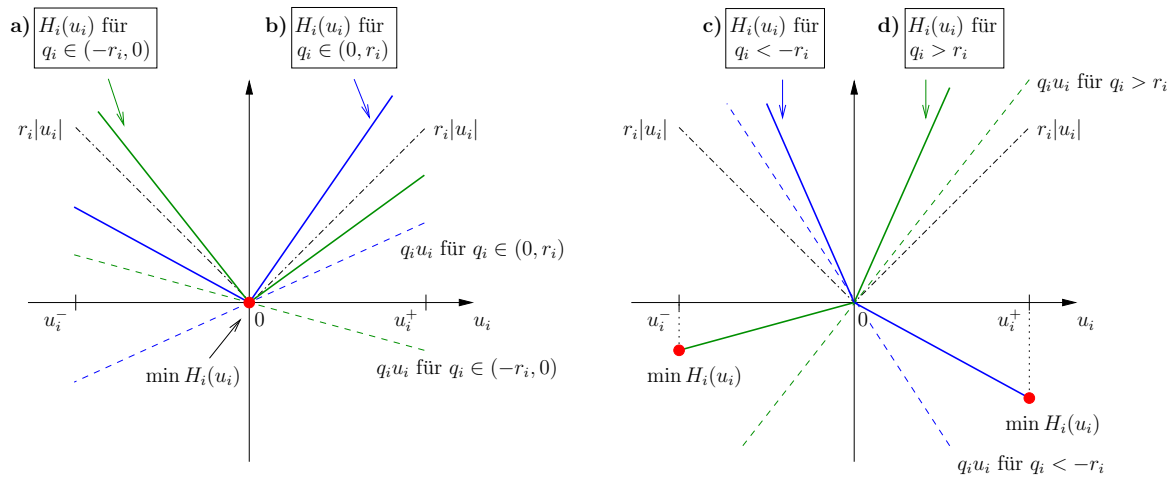


Abbildung 5.16: Veranschaulichung von (5.95) im verbrauchsoptimalen Fall.

Da der Anteil $l_0(\mathbf{x}) + \boldsymbol{\lambda}^\top \mathbf{f}_0(\mathbf{x})$ unabhängig von $\mathbf{u}$ ist, kann er im Minimierungsproblem $\min_{\mathbf{u} \in U} H$ vernachlässigt werden.¹¹ Zudem treten die Elemente u_i des Eingangs $\mathbf{u} = [u_1, \dots, u_m]^\top$ linear in H auf, so dass die Minimierung von H für u_i separat betrachtet werden kann

$$\min_{u_i \in [u_i^-, u_i^+]} H_i(u_i) = r_i |u_i| + q_i(\mathbf{x}, \boldsymbol{\lambda}) u_i \quad \text{mit} \quad q_i(\mathbf{x}, \boldsymbol{\lambda}) = \boldsymbol{\lambda}^\top \mathbf{f}_i(\mathbf{x}). \quad (5.95)$$

Der Vorfaktor $q_i(\mathbf{x}, \boldsymbol{\lambda})$ spielt die maßgebliche Rolle in diesem Problem. Abbildung 5.16 illustriert die unterschiedlichen Fälle a)–d), mit denen die optimale Steuerfunktion (5.91) komponentenweise durch die Fallunterscheidung¹²

$$u_i = \psi_i(\mathbf{x}, \boldsymbol{\lambda}) = \begin{cases} u_i^- & \text{falls } q_i(\mathbf{x}, \boldsymbol{\lambda}) \geq r_i \\ 0 & \text{falls } q_i(\mathbf{x}, \boldsymbol{\lambda}) \in (-r_i, r_i), \\ u_i^+ & \text{falls } q_i(\mathbf{x}, \boldsymbol{\lambda}) \leq -r_i \end{cases} \quad i = 1, \dots, m \quad (5.96)$$

bestimmt ist. Die Steuerung ist also *stückweise stetig*.

5.5.3.2 Kostenfunktional mit energieoptimalem Anteil

Falls Energieoptimalität berücksichtigt werden soll, lautet das Kostenfunktional (5.86a)

$$J(\mathbf{u}) = \int_{t_0}^{t_f} l_0(\mathbf{x}) + \frac{1}{2} \sum_{i=1}^m r_i u_i^2 dt, \quad r_i > 0. \quad (5.97)$$

Analog zum vorherigen Fall kann die Minimierung der Hamilton-Funktion elementweise betrachtet werden

$$\min_{u_i \in [u_i^-, u_i^+]} H_i(u_i) = \frac{1}{2} r_i u_i^2 + q_i(\mathbf{x}, \boldsymbol{\lambda}) u_i \quad \text{mit} \quad q_i(\mathbf{x}, \boldsymbol{\lambda}) = \boldsymbol{\lambda}^\top \mathbf{f}_i(\mathbf{x}). \quad (5.98)$$

¹¹Es sei an dieser Stelle noch einmal betont, dass die Minimierung von $H(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda})$ als Funktion von $\mathbf{u}$ für feste Größen $\mathbf{x}, \boldsymbol{\lambda}$ (genauer: für die *optimalen* $\mathbf{x}^*(t), \boldsymbol{\lambda}^*(t)$ zu jedem Zeitpunkt t) zu verstehen ist, wie es im Maximumprinzip (Satz 5.4) formuliert ist.

¹²An den Schwellpunkten $q_i(\mathbf{x}, \boldsymbol{\lambda}) = \pm r_i$ ist u_i nicht eindeutig definiert, wie aus Abbildung 5.16 abgeleitet werden kann.

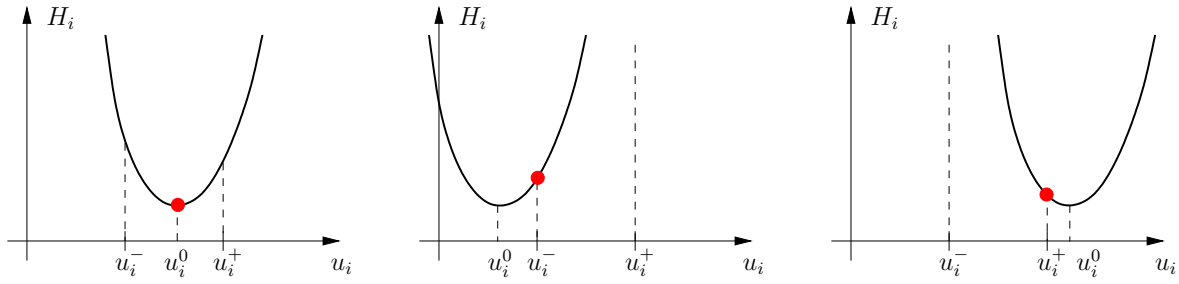


Abbildung 5.17: Veranschaulichung von (5.98) im energieoptimalen Fall.

Durch den quadratischen Term mit $r_i > 0$ hat die Funktion $H_i(u_i)$ stets ein unbeschränktes Minimum an der Stelle

$$u_i^0 = -\frac{1}{r_i} q_i(\mathbf{x}, \boldsymbol{\lambda}).$$

Dies folgt aus der Forderung $\frac{dH_i}{du_i} = r_i u_i + q_i(\mathbf{x}, \boldsymbol{\lambda}) = 0$. Falls u_i^0 innerhalb des zulässigen Intervalls $[u_i^-, u_i^+]$ liegt, ist die optimale Lösung von (5.98) durch $u_i^* = u_i^0$ gegeben. Falls u_i^0 außerhalb von $[u_i^-, u_i^+]$ liegt, so befindet sich das Minimum von $H_i(u_i)$ stets auf der entsprechenden Beschränkung u_i^- oder u_i^+ , da $H_i(u_i)$ für $u_i^0 < u_i^-$ (bzw. $u_i^0 > u_i^+$) streng monoton fallend (steigend) ist, siehe Abbildung 5.17. Somit ist die optimale Steuerfunktion (5.91) wie folgt definiert

$$u_i = \psi_i(\mathbf{x}, \boldsymbol{\lambda}) = \begin{cases} u_i^- & \text{falls } u_i^0 \leq u_i^- \\ u_i^0 & \text{falls } u_i^0 \in (u_i^-, u_i^+), \\ u_i^+ & \text{falls } u_i^0 \geq u_i^+ \end{cases}, \quad i = 1, \dots, m. \quad (5.99)$$

5.5.3.3 Zeitoptimales Kostenfunktional

Für zeitoptimale Probleme ist das Kostenfunktional (5.86a) durch

$$J(\mathbf{u}) = \int_{t_0}^{t_f} 1 \, dt = t_f - t_0 \quad (5.100)$$

definiert. Mit der Hamilton-Funktion

$$H(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}) = 1 + \boldsymbol{\lambda}^\top \left(\mathbf{f}_0(\mathbf{x}) + \sum_{i=1}^m \mathbf{f}_i(\mathbf{x}) u_i \right)$$

kann das Minimierungsproblem (5.89) wieder komponentenweise betrachtet werden

$$\min_{u_i \in [u_i^-, u_i^+]} H_i(u_i) = q_i(\mathbf{x}, \boldsymbol{\lambda}) u_i \quad \text{mit} \quad q_i(\mathbf{x}, \boldsymbol{\lambda}) = \boldsymbol{\lambda}^\top \mathbf{f}_i(\mathbf{x}). \quad (5.101)$$

In Abhängigkeit des Vorzeichens von $q_i(\mathbf{x}, \boldsymbol{\lambda})$ ergibt sich also

$$u_i = \psi_i(\mathbf{x}, \boldsymbol{\lambda}) = \begin{cases} u_i^- & \text{falls } q_i(\mathbf{x}, \boldsymbol{\lambda}) > 0 \\ u_i^+ & \text{falls } q_i(\mathbf{x}, \boldsymbol{\lambda}) < 0 \end{cases}, \quad i = 1, \dots, m. \quad (5.102)$$

Ein kritischer Fall liegt vor, falls $q_i(\mathbf{x}(t), \boldsymbol{\lambda}(t)) = 0$ auf einem nicht verschwindenden Subintervall $I_s \in [t_0, t_f]$ gilt. Die Hamilton-Funktion H ist somit *unabhängig von u_i* , so dass H für jeden beliebigen Wert von u_i trivialerweise minimal ist. Die Minimumsforderung (5.89) ist damit zwar erfüllt, liefert aber keine Informationen über die Wahl von u_i . Dieser *singuläre Fall* wurde bereits in Abschnitt 5.3.4 diskutiert.

In der Praxis wird der singuläre Fall oft durch einen zusätzlichen *Regularisierungsterm*

$$J(\mathbf{u}) = \int_{t_0}^{t_f} 1 + \frac{1}{2} \sum_{i=1}^m r_i u_i^2 dt$$

vermieden, wobei $r_i > 0$ hinreichend klein gewählt wird, um annähernd Zeitoptimalität zu erzielen (siehe auch Beispiel 1.3). Durch den Regularisierungsterm entspricht das Kostenfunktional dem Fall (5.97), so dass $u_i = \psi_i(\mathbf{x}, \boldsymbol{\lambda})$ mit (5.99) berechnet werden kann.

5.5.4 Beispiel: Doppelintegrator

Ein bekanntes Beispiel zur Veranschaulichung des Maximumprinzips ist die zeitminimale Überführung eines doppelt integrierenden Systems in den Ursprung $\mathbf{x}(t_f) = \mathbf{0}$. Das Optimalsteuerungsproblem mit dem Zustand $\mathbf{x} = [x_1, x_2]^T$ kann wie folgt formuliert werden

$$\min_{u(\cdot)} \int_{t_0=0}^{t_f} dt = t_f \quad (5.103a)$$

$$\text{u.B.v.} \quad \dot{x}_1 = x_2, \quad \dot{x}_2 = u \quad (5.103b)$$

$$\mathbf{x}(0) = \mathbf{x}_0, \quad \mathbf{x}(t_f) = \mathbf{0} \quad (5.103c)$$

$$|u(t)| \leq 1 \quad \forall t \in [0, t_f]. \quad (5.103d)$$

Mit der Hamilton-Funktion $H(\mathbf{x}, u, \boldsymbol{\lambda}) = 1 + \lambda_1 x_2 + \lambda_2 u$ ergeben sich die adjungierten Zustände $\boldsymbol{\lambda} = [\lambda_1, \lambda_2]^T$ aus (5.87) zu

$$\dot{\lambda}_1 = 0 \quad \Rightarrow \quad \lambda_1(t) = c_1 \quad (5.104a)$$

$$\dot{\lambda}_2 = -\lambda_1 \quad \lambda_2(t) = -c_1 t + c_2, \quad (5.104b)$$

wobei c_1 und c_2 Integrationskonstanten darstellen. Die Minimierungsbedingung (5.89) für die Hamilton-Funktion führt auf die optimale Steuerung

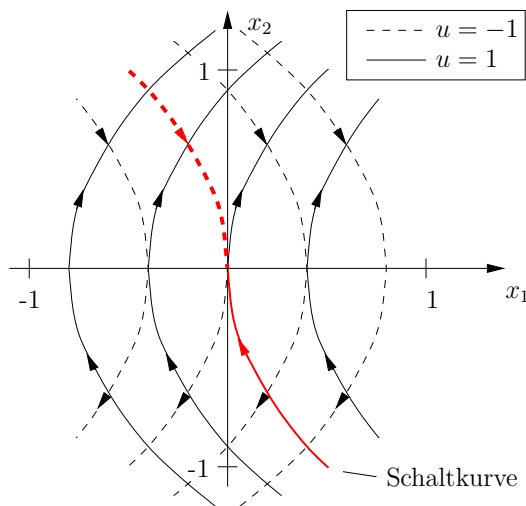
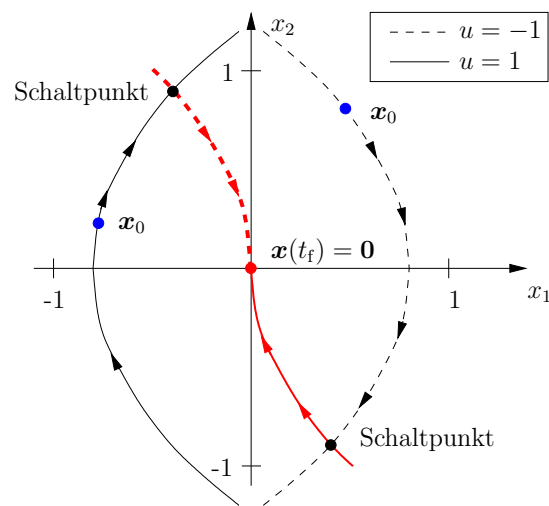
$$u(t) = \begin{cases} +1 & \text{falls } \lambda_2 < 0 \\ -1 & \text{falls } \lambda_2 > 0. \end{cases} \quad (5.105)$$

Der singuläre Fall, d.h. $\lambda_2(t) = 0$ auf einem nicht verschwindenden Subintervall $t \in I_s$, kann nicht auftreten, da dann aufgrund von (5.104b) $\lambda_2(t) = 0$ auf dem Gesamtintervall $[0, t_f]$ gelten müsste. Dies widerspricht aber der Transversalitätsbedingung (5.90) für die freie Endzeit t_f

$$H(\mathbf{x}, u, \boldsymbol{\lambda})|_{t=t_f} = 1 + \lambda_2(t_f) u(t_f) = 0.$$

Der Fall $\lambda_2(t) = 0$ kann also nur zu einem diskreten Zeitpunkt $t = t_s$ auftreten, was einem Umschalten von $u(t)$ zwischen -1 und $+1$ entspricht.¹³ Da $\lambda_2(t)$ in (5.104b) eine Gerade

¹³Es kann gezeigt werden, dass im Falle von steuerbaren linearen (zeitinvarianten) Systemen zeitoptimale Probleme stets nicht-singulär sind, siehe z.B. [9, 4].

Abbildung 5.18: Mögliche Trajektorien des Doppelintegrators in der (x_1, x_2) -Ebene.Abbildung 5.19: Optimale Umschaltung für verschiedene Anfangspunkte $\mathbf{x}_0$.

darstellt, gibt es *maximal einen* Umschaltzeitpunkt im Zeitintervall $[t_0, t_f]$, an dem $u(t)$ zwischen -1 und $+1$ wechselt. Somit existieren lediglich vier mögliche Schaltsequenzen $\{+1\}$, $\{-1\}$, $\{+1, -1\}$, $\{-1, +1\}$, die für eine optimale Lösung in Frage kommen. Da u stets auf einem Intervall konstant ist, stellen die Trajektorien in der (x_1, x_2) -Ebene Parabeln dar, wie in der folgenden Aufgabe gezeigt werden soll.

Aufgabe 5.4 Zeigen Sie, dass die Lösung von (5.103b) für $u(t) = \pm 1 = \text{konst.}$ die folgende Parabelgleichung erfüllt

$$x_1 = \frac{x_2^2}{2u} + c, \quad u = \pm 1.$$

Das Optimierungsziel ist das schnellstmögliche Erreichen des Ursprungs $\mathbf{x}(t_f) = \mathbf{0}$ ausgehend von einem beliebigen Anfangspunkt $\mathbf{x}(0) = \mathbf{x}_0$. Der Ursprung ist aber nur über die *Schaltkurve* $x_1 = x_2^2/2$ für $u = 1$ bzw. $x_1 = -x_2^2/2$ für $u = -1$ erreichbar, siehe Abbildung 5.18. Diese beiden Fälle können in der Funktion

$$S(x_2) = -\frac{1}{2}x_2|x_2|.$$

zusammengefasst werden. Angesichts der Tatsache, dass lediglich die Schaltsequenzen $\{+1\}$, $\{-1\}$, $\{+1, -1\}$, $\{-1, +1\}$ für u in Frage kommen, gibt es nur eine Möglichkeit, um $\mathbf{x}(0) = \mathbf{x}_0$ schnellstmöglich zum Ursprung $\mathbf{x}(t_f) = \mathbf{0}$ zu bringen:

- Falls $x_{1,0} = S(x_{2,0})$ gilt, ist keine Umschaltung notwendig, und $\mathbf{x}(t_f) = \mathbf{0}$ wird direkt über die Schaltkurve mit $u(t) = 1$ oder $u(t) = -1$ erreicht, siehe Abbildung 5.18.
- Falls $\mathbf{x}_0$ nicht auf der Schaltkurve liegt, d.h. $x_{1,0} < S(x_{2,0})$ oder $x_{1,0} > S(x_{2,0})$, ist genau eine Umschaltung notwendig, um die Schaltkurve zu erreichen und anschließend zum Ursprung zu laufen, siehe Abbildung 5.19.

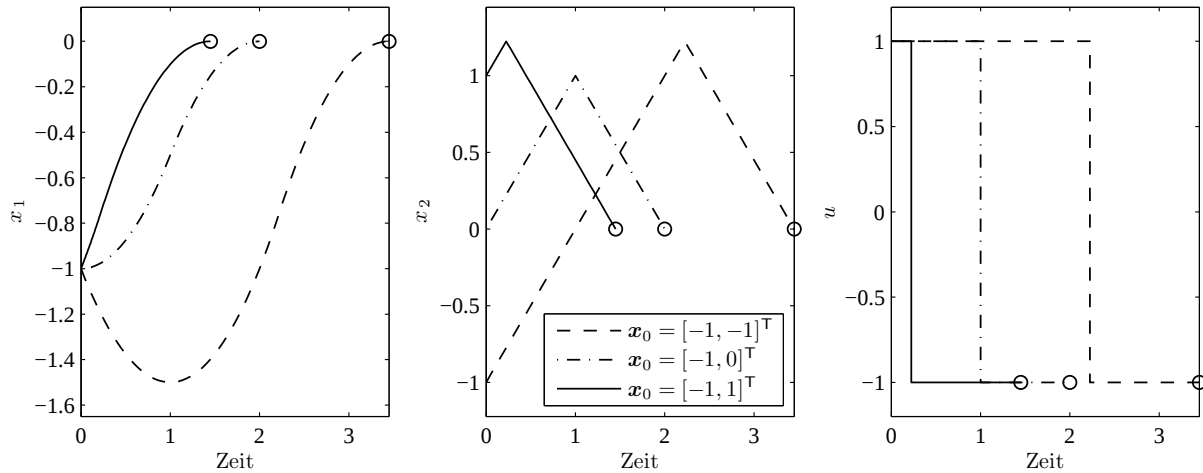


Abbildung 5.20: Zeitoptimale Trajektorien des Doppelintegrators für verschiedene Anfangswerte $\mathbf{x}_0$.

Das optimale Steuer- bzw. Regelgesetz lautet also

$$u(t) = \begin{cases} +1 & \text{falls } x_1 < S(x_2) \\ +1 & \text{falls } x_1 = S(x_2) \text{ und } x_1 > 0 \\ -1 & \text{falls } x_1 > S(x_2) \\ -1 & \text{falls } x_1 = S(x_2) \text{ und } x_1 < 0, \end{cases} \quad (5.106)$$

mit dessen Hilfe der Schaltzeitpunkt t_s und die minimale Endzeit t_f berechnet werden können.

Aufgabe 5.5 Verifizieren Sie, dass der optimale Umschaltzeitpunkt t_s und die Endzeit t_f fallabhängig wie folgt definiert sind:

$$t_s = \begin{cases} x_{2,0} + \sqrt{\frac{1}{2}x_{2,0}^2 + x_{1,0}} & \text{falls } x_{1,0} > S(x_{2,0}) \\ -x_{2,0} + \sqrt{\frac{1}{2}x_{2,0}^2 - x_{1,0}} & \text{falls } x_{1,0} < S(x_{2,0}) \end{cases} \quad (5.107)$$

$$t_f = \begin{cases} x_{2,0} + \sqrt{2x_{2,0}^2 + 4x_{1,0}} & \text{falls } x_{1,0} > S(x_{2,0}) \\ -x_{2,0} + \sqrt{2x_{2,0}^2 - 4x_{1,0}} & \text{falls } x_{1,0} < S(x_{2,0}) \\ |x_{2,0}| & \text{falls } x_{1,0} = S(x_{2,0}) \end{cases} \quad (5.108)$$

Abbildung 5.20 stellt die zeitoptimalen Trajektorien für den Doppelintegrator für verschiedene Anfangswerte $\mathbf{x}_0 = [x_{1,0}, x_{2,0}]^T$ dar.

Eine praktische Schreibweise für die fallabhängige Definition des Stellgesetzes (5.106) ist durch

$$u(t) = -\text{sign}\left(x_1 + \frac{1}{2}x_2|x_2|\right), \quad 0 = \text{sign}(0) \quad (5.109)$$

gegeben. In der Signum-Funktion ist der Sonderfall $x_1 = S(x_2)$ in (5.106) nicht berücksichtigt, was bedeutet, dass der Zustand $\mathbf{x}$ die Schaltkurve $S(x_2)$ um ein kleines Stück überquert, bevor die Umschaltung erfolgt. Dies stellt allerdings numerisch kein Problem dar, da der Zustand $\mathbf{x}$ nach dem Umschalten auf einer Trajektorie hinreichend nahe zur Schaltgeraden $S(x_2)$ verläuft.

Aufgabe 5.6 Implementieren Sie das doppelintegrierende System (5.103b) mit dem optimalen Regelgesetz (5.109) unter MATLAB/SIMULINK und verifizieren Sie die Ergebnisse in Abbildung 5.20 für verschiedene Anfangswerte $\mathbf{x}_0$. Verwenden Sie t_f gemäß (5.108) als Zeithorizont für die Simulation.

5.5.5 Zustandsabhängige Eingangsbeschränkungen

Das Maximumprinzip (siehe Satz 5.4) kann auf eine allgemeinere Klasse von Stellgrößenbeschränkungen

$$\mathbf{h}(\mathbf{x}(t), \mathbf{u}(t)) \leq \mathbf{0} \quad \forall t \in [t_0, t_f] \quad (5.110)$$

mit $\mathbf{h} : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^q$ erweitert werden. Da die Beschränkungen (5.110) allgemein vom Zustand $\mathbf{x}$ abhängen, ist die zulässige Menge

$$U(\mathbf{x}) = \{\mathbf{u} \in \mathbb{R}^m : \mathbf{h}(\mathbf{x}, \mathbf{u}) \leq \mathbf{0}\} \quad (5.111)$$

für die Steuergrößen $\mathbf{u}$ ebenfalls zustandsabhängig, d.h. es muss gelten $\mathbf{u}(t) \in U(\mathbf{x}(t)) \forall t \in [t_0, t_f]$. Es wird angenommen, dass alle Elemente von $\mathbf{h} = [h_1, \dots, h_q]^\top$ explizit von $\mathbf{u}$ abhängen und dass die Beschränkungsqualifikation¹⁴

$$\text{Rang} \left[\frac{\partial \mathbf{h}}{\partial \mathbf{u}}, \text{diag}(\mathbf{h}) \right] = q \quad (5.112)$$

erfüllt ist. Ähnlich dem Vorgehen bei statischen Optimierungsproblemen (Kapitel 4) werden die Ungleichungsbeschränkungen (5.110) mittels (zeitabhängiger) Lagrange-Multiplikatoren $\boldsymbol{\mu} = [\mu_1, \dots, \mu_q]^\top$ zur Hamilton-Funktion (5.88) addiert

$$H(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}, \lambda_0) = \lambda_0 l(\mathbf{x}, \mathbf{u}) + \boldsymbol{\lambda}^\top \mathbf{f}(\mathbf{x}, \mathbf{u}) + \boldsymbol{\mu}^\top \mathbf{h}(\mathbf{x}, \mathbf{u}). \quad (5.113)$$

Zusätzlich zu den Bedingungen (5.87)–(5.90) des Maximumprinzips, die nun für die modifizierte Hamilton-Funktion (5.113) und die (zustandsabhängige) zulässige Menge (5.111) gelten, kommen noch die Komplementaritäts- und Vorzeichenbedingungen

$$\mu_i^*(t) h_i(\mathbf{x}^*(t), \mathbf{u}^*(t)) = 0, \quad \mu_i^*(t) \geq 0 \quad \forall t \in [t_0, t_f] \quad (5.114)$$

als notwendige Optimalitätsbedingungen hinzu. Die Korrespondenz zu den KKT-Bedingungen (4.29d), (4.29e) im statischen Fall ist dabei offensichtlich.

Im Fall von reinen Zustandsbeschränkungen

$$\mathbf{x}(t) \in X \subseteq \mathbb{R}^n \quad \text{bzw.} \quad \mathbf{h}(\mathbf{x}(t)) \leq \mathbf{0} \quad \forall t \in [t_0, t_f]$$

mit $\mathbf{h} : \mathbb{R}^n \rightarrow \mathbb{R}^q$ werden die Optimalitätsbedingungen erheblich komplizierter und verlangen eine genaue Betrachtung von *beschränkten und unbeschränkten Bereichen*, d.h. Subintervalle von $[t_0, t_f]$, auf denen die Beschränkungen aktiv oder inaktiv sind. Des Weiteren existieren verschiedene Möglichkeiten, Zustandsbeschränkungen in der Hamilton-Funktion zu berücksichtigen, die zu unterschiedlichen Optimalitätsbedingungen führen. Für mehr Informationen zu diesem Thema sei auf den Übersichtsautsatz [6] und die Bücher [2, 9] verwiesen.

¹⁴Die Beschränkungsqualifikation (5.112) besagt, dass die Gradienten aller in einem Punkt $(\mathbf{x}, \mathbf{u})$ aktiven Beschränkungen $h_i(\mathbf{x}, \mathbf{u}) = 0$ linear unabhängig sein müssen.

Literatur

- [1] M. Athans und P.L. Falb. *Optimal Control*. McGraw–Hill, New York, 1966.
- [2] A.E. Bryson und Y.-C. Ho. *Applied Optimal Control*. John Wiley & Sons, New York, 1975.
- [3] B. Chachuat. Nonlinear and Dynamic Optimization: From Theory to Practice. <http://lawwww.epfl.ch/page4234.html>, 2007.
- [4] O. Föllinger. *Optimale Steuerung und Regelung*. Oldenbourg, München, 1994.
- [5] K. Graichen. *Feedforward Control Design for Finite–Time Transition Problems of Nonlinear Systems with Input and Output Constraints*. Dissertation, Universität Stuttgart. Shaker Verlag, Aachen, 2006. <http://elib.uni-stuttgart.de/opus/volltexte/2007/3004>.
- [6] R.F. Hartl, S.P. Sethi und R.G. Vickson. A survey of the maximum principles for optimal control problems with state constraints. *SIAM Review*, 37:181–218, 1995.
- [7] M.I. Kamien und N.L. Schwartz. *Dynamic Optimization: The Calculus of Variations and Optimal Control in Economics and Management*. North–Holland, Amsterdam, Amsterdam, 2nd edition edition, 1991.
- [8] H. Kwakernaak und R. Sivan. *Linear Optimal Control Systems*. Wiley–Interscience, New York, 1972.
- [9] M. Papageorgiou. *Optimierung*. Oldenbourg Verlag, 1991.
- [10] B. van Brunt. *The Calculus of Variations*. Springer, New York, 2004.

Kapitel 6

Numerische Lösung von dynamischen Optimierungsproblemen

In der Regel ist man zur Lösung von dynamischen Optimierungsproblemen auf numerische Verfahren angewiesen, wobei grundsätzlich zwischen zwei Verfahrensklassen unterschieden wird:

- *Indirekte Verfahren* basieren auf der Lösung der Optimalitätsbedingungen (5.37)–(5.42) bzw. (5.87)–(5.90) im eingangsbeschränkten Fall.
- *Direkte Verfahren* diskretisieren die Steuertrajektorie $\mathbf{u}(t)$ und reduzieren somit das dynamische (unendlich dimensionale) Optimierungsproblem auf ein statisches Problem, das mit den in den Kapitel 3 und 4 beschriebenen Verfahren gelöst werden kann.

Das folgende Kapitel stellt die gebräuchlichsten direkten und indirekten Verfahren vor und wendet diese anschließend auf ein Navigationsproblem in der Schifffahrt an.¹

6.1 Indirekte Verfahren: Lösung der Optimalitätsbedingungen

Zwei-Punkt-Randwertprobleme, wie sie aus den Optimalitätsbedingungen dynamischer Optimierungsprobleme resultieren, lassen sich mit einer Reihe von numerischen Verfahren lösen. Bekannte Verfahren sind *Diskretisierungs-*, *Schieß-* und *Gradientenverfahren*, die im Folgenden erläutert werden.

6.1.1 Diskretisierungsverfahren

Wie bereits mehrfach erwähnt wurde, stellen die Optimalitätsbedingungen für dynamische Optimierungsprobleme ein Zwei-Punkt-Randwertproblem dar. Insbesondere wenn sich

¹Dank gebührt an dieser Stelle Alexander Lutz und Dr. Eckhard Arnold vom Institut für Systemdynamik der Universität Stuttgart für Beiträge und Anregungen zu Abschnitt 6.3.

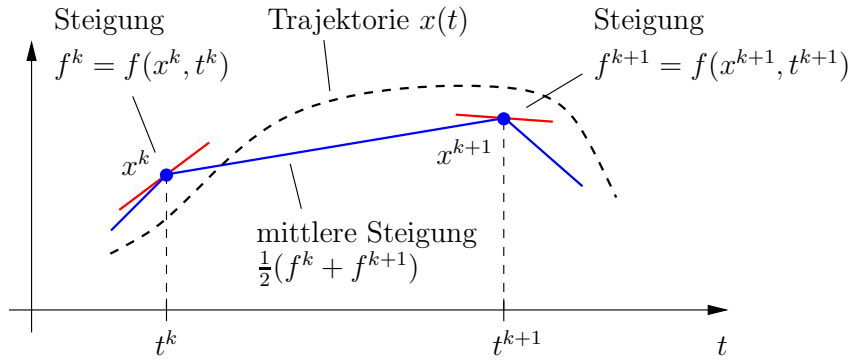


Abbildung 6.1: Veranschaulichung der Trapezregel.

die optimale Steuerung als Funktion $\mathbf{u} = \boldsymbol{\psi}(\mathbf{x}, \boldsymbol{\lambda})$ der Zustände $\mathbf{x}$ und $\boldsymbol{\lambda}$ ausdrücken lässt, können die kanonischen Gleichungen in die von $\mathbf{u}$ unabhängige Form (siehe Abschnitt 5.3.4 und 5.5.2)

$$\left. \begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \boldsymbol{\psi}(\mathbf{x}, \boldsymbol{\lambda}), t) \\ \dot{\boldsymbol{\lambda}} &= -H_{\mathbf{x}}(\mathbf{x}, \boldsymbol{\psi}(\mathbf{x}, \boldsymbol{\lambda}), \boldsymbol{\lambda}, t) \end{aligned} \right\} \dot{\bar{\mathbf{x}}} = \mathbf{F}(\bar{\mathbf{x}}, t), \quad \bar{\mathbf{x}} = \begin{bmatrix} \mathbf{x} \\ \boldsymbol{\lambda} \end{bmatrix} \quad (6.1)$$

überführt werden. Im Folgenden werden eine feste (vorgegebene) Endzeit t_f und die folgenden Anfangs- und (partiellen) Endbedingungen

$$\mathbf{x}(t_0) = \mathbf{x}_0, \quad \mathbf{G}(\bar{\mathbf{x}}(t_f)) = \begin{bmatrix} [x_i(t_f) - x_{f,i}]_{i \in \mathcal{I}_f} \\ [\lambda_i(t_f) - \frac{\partial \varphi}{\partial x_i} \Big|_{t_f}]_{i \notin \mathcal{I}_f} \end{bmatrix} = \mathbf{0} \quad (6.2)$$

mit $\mathbf{G} : \mathbb{R}^{2n} \rightarrow \mathbb{R}^n$ betrachtet, so dass $2n$ Randbedingungen für die $2n$ Differentialgleichungen zur Verfügung stehen. Die Behandlung von allgemeinen Endbedingungen und einer freien Endzeit t_f wird in Abschnitt 6.1.3 angesprochen.

Eine gebräuchliche Vorgehensweise zur Lösung von Zwei-Punkt-Randwertaufgaben ist die Diskretisierung des Zeitintervalls $[t_0, t_f]$

$$t_0 = t^0 < t^1 < \dots < t^N = t_f,$$

um die Lösung an den $N + 1$ Stützstellen t^k zu approximieren, d.h.

$$\bar{\mathbf{x}}^k \approx \bar{\mathbf{x}}(t^k), \quad k = 0, 1, \dots, N. \quad (6.3)$$

Eine bekannte Diskretisierungsformel ist z.B. die *Trapezregel*

$$\frac{\bar{\mathbf{x}}^{k+1} - \bar{\mathbf{x}}^k}{t^{k+1} - t^k} = \frac{1}{2} [\mathbf{F}(\bar{\mathbf{x}}^k, t^k) + \mathbf{F}(\bar{\mathbf{x}}^{k+1}, t^{k+1})], \quad k = 0, 1, \dots, N-1, \quad (6.4a)$$

die in Abbildung 6.1 veranschaulicht ist. Zusätzlich müssen die $2n$ Randbedingungen (6.2)

$$\mathbf{x}^0 = \mathbf{x}_0, \quad \mathbf{G}(\bar{\mathbf{x}}^N) = \mathbf{0} \quad (6.4b)$$

erfüllt sein. Die Bedingungen (6.4) stellen ein nichtlineares Gleichungssystem der Ordnung $2n(N + 1)$ für die $2n(N + 1)$ zu bestimmenden Variablen (6.3) dar.

Ein verwandtes Verfahren zur Lösung von Zwei-Punkt-Randwertaufgaben ist das *Kollokationsverfahren*, das den Zustand $\bar{\mathbf{x}}(t)$ auf jedem Intervall $[t^k, t^{k+1}]$ durch Basisfunktionen

(z.B. Polynome) annähert. Die Übereinstimmung mit den Differentialgleichungen (6.1) wird an zusätzlichen *Kollokationspunkten* innerhalb der Intervalle $[t^k, t^{k+1}]$ erzwungen und die Stetigkeit an den Intervallgrenzen t^k durch Stetigkeitsbedingungen gefordert. Somit ergibt sich ebenfalls ein nichtlineares Gleichungssystem für die Koeffizienten der Basisfunktionen. Für bestimmte Basisfunktionen und Kollokationspunkte ist das Kollokationsverfahren zu gewissen Diskretisierungsformeln äquivalent.

Einige Merkmale von Diskretisierungsverfahren sind im Folgenden zusammengetragen:

- Aufgrund der simultanen Betrachtung der Differentialgleichungen (6.1) und Randbedingungen (6.2) sind Diskretisierungsverfahren oft numerisch robuster als das Schießverfahren (Abschnitt 6.1.2).
- Trotzdem spielt die Startschätzung der adjungierten Variablen eine maßgebliche Rolle und entscheidet oft darüber, ob und wie schnell das Verfahren konvergiert.
- Der Implementierungsaufwand ist recht hoch. Die Jacobi-Matrix des Gleichungssystems (6.4) ist *dünn besetzt* und weist eine *Blockstruktur* auf. Diese strukturellen Eigenschaften können durch geeignete numerische Verfahren ausgenutzt werden.
- Die Anzahl der Stützstellen $(N + 1)$ hat maßgeblichen Einfluß auf die Lösung. Größere Werte von N erhöhen die Genauigkeit, vergrößern aber ebenfalls die Dimension des zu lösenden Gleichungssystems.
- Die Trapezregel besitzt die Verfahrensordnung 2, d.h. die numerische Lösung $\bar{\mathbf{x}}^k \approx \bar{\mathbf{x}}(t^k)$ stimmt bis zum quadratischen Term mit der Taylorreihe der exakten Lösung $\bar{\mathbf{x}}(t)$ an der Stelle $t = t^k$ überein. Alternativ können z.B. Runge-Kutta-Formeln mit höherer Verfahrensordnung verwendet werden.

6.1.2 Schießverfahren

Im Gegensatz zum Diskretisierungsverfahren, das ein großes Gleichungssystem für die Differentialgleichungen (6.1) und die Randbedingungen (6.2) löst, betrachtet das Schießverfahren das Anfangswertproblem

$$\dot{\bar{\mathbf{x}}} = \mathbf{F}(\bar{\mathbf{x}}, t), \quad \bar{\mathbf{x}}(t_0) = \begin{bmatrix} \mathbf{x}_0 \\ \boldsymbol{\lambda}_0 \end{bmatrix} \quad (6.5)$$

mit den (noch unbekannten) Anfangswerten $\boldsymbol{\lambda}_0$ der adjungierten Zustände $\boldsymbol{\lambda}$. Die Lösung von (6.5) kann formal in Abhängigkeit von $\boldsymbol{\lambda}_0$ wie folgt geschrieben werden

$$\bar{\mathbf{x}}(t; \boldsymbol{\lambda}_0) = \begin{bmatrix} \mathbf{x}_0 \\ \boldsymbol{\lambda}_0 \end{bmatrix} + \int_{t_0}^t \mathbf{F}(\bar{\mathbf{x}}(\tau; \boldsymbol{\lambda}_0), \tau) d\tau. \quad (6.6)$$

Um die Endbedingungen in (6.2) zu erfüllen, muss gelten

$$\mathbf{G}(\bar{\mathbf{x}}(t_f; \boldsymbol{\lambda}_0)) = \mathbf{0}, \quad (6.7)$$

was ein nichtlineares Gleichungssystem der Ordnung n für die Anfangsbedingungen $\boldsymbol{\lambda}_0 \in \mathbb{R}^n$ darstellt.

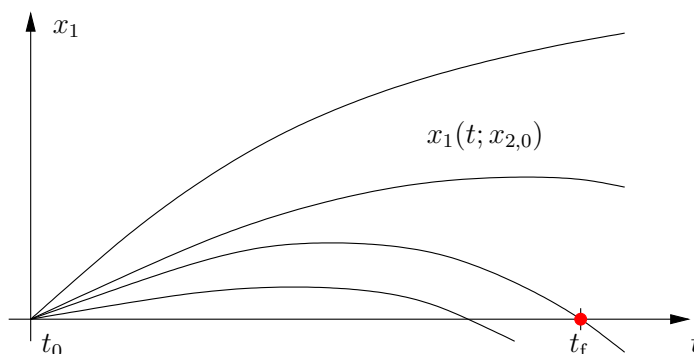


Abbildung 6.2: Veranschaulichung des Schießverfahrens (Beispiel 6.1).

Das Gleichungssystem (6.7) muss im Allgemeinen iterativ gelöst werden, z.B. mit Hilfe des Newton-Verfahrens. Das Anfangswertproblem (6.5) wird dazu unterlagert numerisch integriert, um die Trajektorie (6.6) zu bestimmen (siehe Abschnitt 6.3). Aufgrund der wiederholten (numerischen) Integration von (6.5) spricht man in diesem Zusammenhang vom *Schießverfahren*.

Beispiel 6.1 Zur Veranschaulichung des Schießverfahrens sei das Randwertproblem

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = f(x_1, x_2), \quad x_1(t_0) = 0, \quad x_1(t_f) = 0$$

gegeben. Wird das Randwertproblem als Anfangswertproblem betrachtet, so hängt die Lösung $x_1(t; x_{2,0})$ vom Startwert $x_2(t_0) = x_{2,0}$ des zweiten Zustandes ab. Der Wert $x_{2,0}$ entspricht der Anfangssteigung der Trajektorie $x_1(t; x_{2,0})$, die so gewählt werden muss, dass die Endbedingung $x_1(t_f; x_{2,0}) = 0$ erfüllt ist, siehe Abbildung 6.2.

Einige Anmerkungen zum Schießverfahren:

- Der Implementierungsaufwand ist aufgrund der sequentiellen Lösung des Anfangswertproblems (6.5) und der Endbedingungen (6.7) relativ gering.
- Gute Startwerte für λ_0 sind wichtig, da das Schießverfahren häufig sehr sensitiv bzgl. dieser Größen ist.
- Die Integration des Anfangswertproblems (6.5) kann numerisch kritisch sein, vor allem wenn das Zeitintervall $[t_0, t_f]$ vergleichsweise groß ist, da die kanonischen Gleichungen in der Regel instabile Moden besitzen.²
- Abhilfe verschaffen *Mehrfachschießverfahren*, die das Intervall $[t_0, t_f]$ in Unterintervalle unterteilen [1, 9].

6.1.3 Berücksichtigung allgemeiner Endbedingungen und freier Endzeit

In den vorherigen Abschnitten wurde zunächst von einer festen Endzeit t_f und partiellen Endbedingungen ausgegangen. Im Falle von allgemeinen Endbedingungen (5.64) wird die

²Im linearen Fall kann gezeigt werden, dass die Eigenwerte der kanonischen Gleichungen, siehe (5.71), symmetrisch zur imaginären Achse sind.

Funktion $\mathbf{G}$ in (6.2) durch

$$\mathbf{G}(\bar{\mathbf{x}}(t_f), \boldsymbol{\nu}) = \begin{bmatrix} \mathbf{g}(\mathbf{x}(t_f), t_f) \\ \boldsymbol{\lambda}(t_f) - \left[\frac{\partial \varphi}{\partial \mathbf{x}} + \left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}} \right)^\top \boldsymbol{\nu} \right]_{t=t_f} \end{bmatrix} = \mathbf{0} \quad (6.8)$$

mit $\mathbf{G} : \mathbb{R}^{2n} \times \mathbb{R}^r \rightarrow \mathbb{R}^{n+r}$ ersetzt. Die Lagrange-Multiplikatoren $\boldsymbol{\nu} \in \mathbb{R}^r$ stellen zusätzliche Parameter dar, die den zusätzlichen r Randbedingungen in der Funktion $\mathbf{G}$ Rechnung tragen.

Falls die Endzeit t_f frei ist, kann eine Zeittransformation

$$t = \delta \tau \quad \text{mit} \quad \tau \in [t_0, t_f^0], \quad t_f^0 = \text{konst.} \quad (6.9)$$

verwendet werden, um die kanonischen Gleichungen auf das feste Zeitintervall $[t_0, t_f^0]$ der neuen Zeitkoordinate τ zu transformieren. Der Parameter δ stellt dabei den Zeitstreckfaktor dar. Mit den neuen Zuständen

$$\tilde{\mathbf{x}}(\tau) := \begin{bmatrix} \mathbf{x}(\delta \tau) \\ \boldsymbol{\lambda}(\delta \tau) \end{bmatrix}, \quad \tau \in [t_0, t_f^0]$$

und der Notation $(\cdot)' = \frac{d}{d\tau}(\cdot)$ ergibt sich für die kanonischen Gleichungen (6.1)

$$\tilde{\mathbf{x}}' = \delta \mathbf{F}(\tilde{\mathbf{x}}, \delta \tau), \quad \tau \in (t_0, t_f^0). \quad (6.10)$$

Zusätzlich werden die Endbedingungen $\mathbf{G}(\cdot) = \mathbf{0}$ um die Transversalitätsbedingung (5.42) bzw. (5.67) erweitert.

Für das Diskretisierungs- bzw. Schießverfahren stellen die zusätzlichen Größen δ und/oder $\boldsymbol{\nu}$ weitere zu bestimmende Variablen dar, wobei die zusätzlichen Gleichungen in den Randbedingungen $\mathbf{G}(\cdot) = \mathbf{0}$ die gleiche Dimension besitzen. Im Folgenden sind die unterschiedlichen Fälle zusammengefasst:

Endzeit t_f	Endbed.	Zusätzliche Parameter		Anzahl Anfangs-/Endbed.	
		$\delta \in \mathbb{R}$	$\boldsymbol{\nu} \in \mathbb{R}^r$	$\mathbf{x}(t_0) = \mathbf{x}_0$	$\mathbf{G}(\cdot) = \mathbf{0}$
fest	partiell	–	–	n	n
fest	allgemein	–	✓	n	$n + r$
frei	partiell	✓	–	n	$n + 1$
frei	allgemein	✓	✓	n	$n + r + 1$

6.1.4 Gradientenverfahren

Ein weiteres klassisches Verfahren zur Lösung von Optimalsteuerungsproblemen ist das *Gradientenverfahren*, das Ähnlichkeit zum gleichnamigen Verfahren bei statischen Optimierungsproblemen (siehe Abschnitt 3.3.3.1) aufweist. Die hauptsächliche Anwendung findet das Gradientenverfahren bei Problemen mit fester Endzeit und freiem Endzustand, für die sich die Optimalitätsbedingungen (5.37)–(5.39) zu

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t), \quad \mathbf{x}(t_0) = \mathbf{x}_0 \quad (6.11)$$

$$\dot{\boldsymbol{\lambda}} = -H_{\mathbf{x}}(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}, t), \quad \boldsymbol{\lambda}(t_f) = \varphi_{\mathbf{x}}(\mathbf{x}(t_f)) \quad (6.12)$$

$$\mathbf{0} = H_{\mathbf{u}}(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}, t) \quad (6.13)$$

mit $H_{\mathbf{x}} = \frac{\partial H}{\partial \mathbf{x}}$, $H_{\mathbf{u}} = \frac{\partial H}{\partial \mathbf{u}}$ und $\varphi_{\mathbf{x}} = \frac{\partial \varphi}{\partial \mathbf{x}}$ ergeben.

Das Gradientenverfahren nutzt die Entkopplung der Randbedingungen aus, indem es in jedem Schritt j die kanonischen Gleichungen in Abhängigkeit einer Steuertrajektorie $\mathbf{u}^j(t)$, $t \in [t_0, t_f]$ sequentiell in Vorwärts- und Rückwärtszeit integriert, um die Zustandstrajektorien $\mathbf{x}^j(t)$ und $\boldsymbol{\lambda}^j(t)$ zu bestimmen. Im Allgemeinen wird diese Lösung die Stationaritätsbedingung (6.13) nicht erfüllen, so dass ein nicht verschwindender Gradient

$$\mathbf{g}^j(t) = H_{\mathbf{u}}(\mathbf{x}^j(t), \mathbf{u}^j(t), \boldsymbol{\lambda}^j(t), t), \quad t \in [t_0, t_f] \quad (6.14)$$

existiert. Ähnlich dem Gradientenverfahren bei statischen Optimierungsproblemen (siehe Abschnitt 3.3.3.1) kann der negative Gradient $-\mathbf{g}^j(t)$ verwendet werden, um einen Abstieg im Kostenfunktional

$$J(\mathbf{u}) = \varphi(\mathbf{x}(t_f)) + \int_{t_0}^{t_f} l(\mathbf{x}(t), \mathbf{u}(t), t) dt$$

zu erreichen. Tabelle 6.1 fasst den grundsätzlichen Algorithmus zusammen.

Initialisierung:	
$j \leftarrow 0$	Iterationszähler
ε	Abbruchkriterium
$\mathbf{u}^0(t)$	Startschätzung der Steuertrajektorie
$\mathbf{x}^0(t)$	Integration von (6.11) in Vorwärtszeit
repeat	
$\boldsymbol{\lambda}^j(t)$	Integration von (6.12) in Rückwärtszeit
$\mathbf{g}^j(t) \leftarrow (6.14)$	Berechnung des Gradienten
$\alpha^j \leftarrow \arg \min_{\alpha > 0} J(\mathbf{u}^j - \alpha \mathbf{g}^j)$	Linien suche (meist approximative Lösung)
$\mathbf{u}^{j+1}(t) = \mathbf{u}^j(t) - \alpha^j \mathbf{g}^j(t)$	neue Steuertrajektorie mit Schrittweite α^j
$\mathbf{x}^{j+1}(t)$	Integration von (6.11) in Vorwärtszeit
$j \leftarrow j + 1$	
until $ J(\mathbf{u}^{j+1}) - J(\mathbf{u}^j) \leq \varepsilon$	oder anderes Abbruchkriterium

Tabelle 6.1: Algorithmus des Gradientenverfahrens.

Das unterlagerte Liniensuchproblem wird in der Regel approximativ gelöst, z.B. mit Hilfe einer Armijo-Bedingung (siehe Abschnitt 3.3.2.1). In manchen Versionen des Gradientenverfahrens wird die Liniensuche durch eine feste Schrittweite α ersetzt, womit sich direkt $\mathbf{u}^{j+1}(t) = \mathbf{u}^j(t) - \alpha \mathbf{g}^j(t)$ ergibt. In diesem Fall muss allerdings α hinreichend klein gewählt werden, was oft zu einer langsameren Konvergenz führt.

Mehr Informationen zum Gradientenverfahren finden sich z.B. in [10, 4, 8]. Einige Anmerkungen sind im Folgenden zusammengetragen:

- Ein Vorteil des Gradientenverfahrens (vor allem bei allgemeinen nichtlinearen Problemen) ist, dass $\mathbf{u}$ nicht explizit aus $H_{\mathbf{u}} = \mathbf{0}$ bzw. $\min_{\mathbf{u} \in U} H$ berechnet werden muss (siehe Abschnitte 5.3.4 und 5.5.2).

- Stellgrößenbeschränkungen $\mathbf{u} \in U$ können durch *Projektion* auf die zulässige Menge U berücksichtigt werden.
- Das Gradientenverfahren ist in der Regel numerisch robuster als das Schießverfahren, da das adjungierte System (6.12) nicht in die instabile Vorwärtsrichtung integriert wird. Ein weiterer Pluspunkt (auch gegenüber Diskretisierungsverfahren) ist, dass die Startschätzung der adjungierten Variablen entfällt.
- Es tritt meist eine deutliche Reduktion des Kostenfunctionals am Anfang des Verfahrens auf (für kleine j). Die Konvergenz in der Nähe der optimalen Lösung ist aber recht langsam.
- Alternativ können *Konjugierte-Gradienten-Verfahren* oder *Gradientenverfahren 2. Ordnung* mit besseren Konvergenzeigenschaften verwendet werden.
- Das Gradientenverfahren kann auf partielle/allgemeine Randbedingungen und freie Endzeit erweitert werden, wodurch sich aber die Robustheit und Konvergenzeigenschaften oft deutlich verschlechtern.

6.2 Direkte Verfahren: Reduktion auf statische Optimierungsprobleme

Direkte Methoden beruhen auf der Parametrierung der Steuertrajektorie $\mathbf{u}(t)$, $t \in [t_0, t_f]$, um das Optimalsteuerungsproblem auf ein endlich dimensionales Problem zu reduzieren, das mit den bekannten Methoden der statischen Optimierung gelöst werden kann (siehe Kapitel 3 und 4). Im Folgenden sind einige Vorteile des direkten/indirekten Lösungswegs gegenübergestellt:

Direkte Verfahren:

- Herleitung der kanonischen Gleichungen ist nicht notwendig
- Zustandsbeschränkungen sind leichter zu berücksichtigen
- oft größerer Konvergenzbereich, da keine Startschätzung der adjungierten Zustände benötigt wird

Indirekte Verfahren:

- liefern Einsicht in die Struktur der optimalen Lösung
- hochgenaue Lösung (wichtig z.B. bei der Bahnplanung in der Raumfahrt)
- adjungierte Variablen können für Sensitivitätsanalyse und Reglerentwurf ("*neighbouring extremal control*") verwendet werden

Unterschieden wird bei direkten Verfahren zwischen *Teil-* oder *Volldiskretisierung*, die im Folgenden erläutert werden. Als Basis dafür dient ein allgemeines beschränktes Optimal-

steuerungsproblem der Form

$$\min_{\mathbf{u}(\cdot)} \quad J(\mathbf{u}) = \varphi(\mathbf{x}(t_f)) + \int_{t_0}^{t_f} l(\mathbf{x}(t), \mathbf{u}(t), t) dt \quad (6.15a)$$

$$\text{u.B.v.} \quad \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t), \quad \mathbf{x}(t_0) = \mathbf{x}_0 \quad (6.15b)$$

$$\mathbf{g}(\mathbf{x}(t_f)) = \mathbf{0} \quad (6.15c)$$

$$\mathbf{h}(\mathbf{x}(t), \mathbf{u}(t), t) \leq \mathbf{0} \quad \forall t \in [t_0, t_f]. \quad (6.15d)$$

Die Endzeit t_f sei der Einfachheit halber fest. Falls t_f frei ist, kann die Zeittransformation (6.9) verwendet werden, um das Problem wieder auf ein festes Intervall $[t_0, t_f^0]$ zu transformieren.

6.2.1 Teildiskretisierung

Bei direkten Lösungsverfahren wird zunächst das Zeitintervall $[t_0, t_f]$ diskretisiert

$$t_0 = t^0 < t^1 < \dots < t^N = t_f, \quad (6.16)$$

und die Steuergrößen $\mathbf{u}$ auf den N Subintervallen $[t^k, t^{k+1}]$ parametrisiert. Im einfachsten Fall wird $\mathbf{u}$ auf jedem Subintervall konstant angesetzt, d.h.

$$\mathbf{u}(t, \hat{\mathbf{u}}) = \mathbf{u}^k, \quad t \in [t^k, t^{k+1}] \quad \text{mit} \quad \hat{\mathbf{u}} = \begin{bmatrix} \mathbf{u}^0 \\ \vdots \\ \mathbf{u}^{N-1} \end{bmatrix} \in \mathbb{R}^{mN}. \quad (6.17)$$

Somit ergibt sich für die Steuertrajektorie $\mathbf{u}(t, \hat{\mathbf{u}})$ ein stückweise konstantes Profil in Abhängigkeit der mN Parameter $\hat{\mathbf{u}}$. Alternativ können z.B. stückweise lineare Ansätze für $\mathbf{u}(t, \hat{\mathbf{u}})$ verwendet werden, siehe Abbildung 6.3.

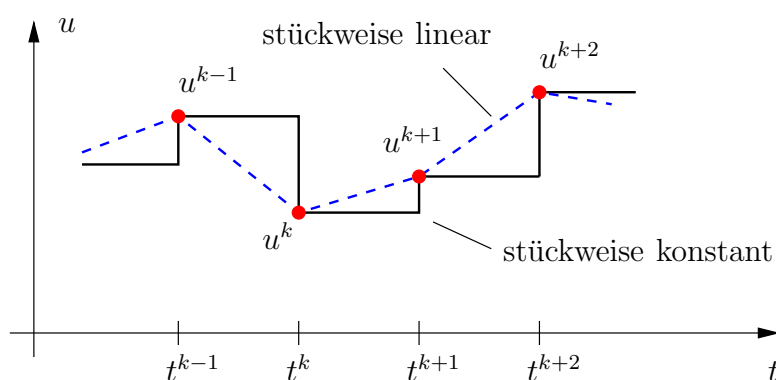


Abbildung 6.3: Stückweise konstante und stückweise lineare Parametrierung der Steuertrajektorie im skalaren Fall ($m = 1$).

Durch die Parametrierung der Steuertrajektorie $\mathbf{u}(t, \hat{\mathbf{u}})$ wird das Optimalsteuerungsproblem auf das statische (endlich-dimensionale) Optimierungsproblem

$$\min_{\hat{\mathbf{u}} \in \mathbb{R}^{mN}} \hat{J}(\hat{\mathbf{u}}) = \varphi(\mathbf{x}(t_f)) + \sum_{k=0}^{N-1} \int_{t^k}^{t^{k+1}} l(\mathbf{x}(t), \mathbf{u}(t, \hat{\mathbf{u}}), t) dt \quad (6.18a)$$

$$\text{u.B.v. } \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}(t, \hat{\mathbf{u}}), t), \quad t \in (t^k, t^{k+1}), \quad k = 0, 1, \dots, N-1 \quad (6.18b)$$

$$\mathbf{x}(t^0) = \mathbf{x}_0, \quad \mathbf{x}(t^k) = \mathbf{x}(t^{k-}), \quad k = 0, 1, \dots, N-1 \quad (6.18c)$$

$$\mathbf{g}(\mathbf{x}(t^N)) = \mathbf{0} \quad (6.18d)$$

$$\mathbf{h}(\mathbf{x}(t), \mathbf{u}(t, \hat{\mathbf{u}}), t) \leq \mathbf{0} \quad \forall t \in [t^k, t^{k+1}], \quad k = 0, 1, \dots, N-1 \quad (6.18e)$$

überführt. Zur Lösung des nichtlinearen Optimierungsproblems (6.18) in Abhängigkeit der Parameter $\hat{\mathbf{u}}$ muss die Differentialgleichung (6.18b) auf den einzelnen Subintervallen unterlagert numerisch integriert werden. Die Bedingungen (6.18c) gewährleisten dabei Stetigkeit des Zustandes $\mathbf{x}(t)$ über die Intervallgrenzen t^k hinweg.

Ein Problem in dieser Formulierung ist, dass die Ungleichungsbeschränkungen (6.18e) korrekterweise auf jedem Intervall $[t^k, t^{k+1}]$ und nicht nur an den Stützstellen t^k erfüllt sein sollten. In der Praxis wird (6.18e) häufig durch punktweise Ungleichungsbeschränkungen

$$\mathbf{h}(\mathbf{x}(t^{k,j}), \mathbf{u}(t^{k,j}, \hat{\mathbf{u}}), t^{k,j}) \leq \mathbf{0}$$

an einer Anzahl von inneren Punkten $t^{k,j} \in [t^k, t^{k+1}]$ ersetzt. Das Problem (6.18) kann dann mit herkömmlichen Methoden der nichtlinearen Optimierung, z.B. dem SQP-Verfahren, angegangen werden.

Einige Merkmale direkter Verfahren mit Teildiskretisierung sind:

- Falls die Bedingungen (6.18c) als Anfangsbedingungen und nicht als Gleichungsbeschränkungen im Optimierungsverfahren implementiert sind, erfüllt die Zustands-trajektorie $\mathbf{x}(t)$, $t \in [t_0, t_f]$ (im Rahmen des verwendeten Integrationsverfahrens) stets die Differentialgleichung (6.18b), d.h. auch während der einzelnen Iterationen des Optimierungsverfahrens.
- Die Berechnung der Gradienten der Kostenfunktion und Beschränkungen in (6.18) nach $\hat{\mathbf{u}}$ (z.B. über die Sensitivitätsdgl.) ist im Allgemeinen recht aufwändig.
- Aufgrund der separaten Integration der Systemgleichungen (6.18b) auf den einzelnen Intervallen spricht man bei der Teildiskretisierung oft von einem *direkten sequentiellen Verfahren*.

6.2.2 Volldiskretisierung

Im Gegensatz zur Teildiskretisierung werden bei der Volldiskretisierung neben der Steuertrajektorie ebenfalls die Differentialgleichungen (6.15b) an den Stützstellen (6.16) diskretisiert. In Analogie zu Abschnitt 6.1.1 kann dies z.B. mit Hilfe der Trapezregel (6.4) erfolgen, während $\mathbf{u}$ stückweise linear angesetzt wird:

$$\frac{\mathbf{x}^{k+1} - \mathbf{x}^k}{t^{k+1} - t^k} = \frac{1}{2} \left[\mathbf{f}(\mathbf{x}^k, \mathbf{u}^k, t^k) + \mathbf{f}(\mathbf{x}^{k+1}, \mathbf{u}^{k+1}, t^{k+1}) \right], \quad k = 0, 1, \dots, N-1.$$

Somit ergibt sich für das Optimalsteuerungsproblem (6.15) das nichtlineare Optimierungsproblem

$$\min_{\hat{\mathbf{x}}, \hat{\mathbf{u}}} \quad \hat{J}(\hat{\mathbf{x}}, \hat{\mathbf{u}}) = \varphi(\mathbf{x}^N) + \sum_{k=0}^{N-1} \frac{t^{k+1} - t^k}{2} \left[l(\mathbf{x}^k, \mathbf{u}^k, t^k) + l(\mathbf{x}^{k+1}, \mathbf{u}^{k+1}, t^{k+1}) \right] \quad (6.19a)$$

$$\text{u.B.v.} \quad \frac{\mathbf{x}^{k+1} - \mathbf{x}^k}{t^{k+1} - t^k} = \frac{1}{2} \left[\mathbf{f}(\mathbf{x}^k, \mathbf{u}^k, t^k) + \mathbf{f}(\mathbf{x}^{k+1}, \mathbf{u}^{k+1}, t^{k+1}) \right], \quad k = 0, 1, \dots, N-1 \quad (6.19b)$$

$$\mathbf{x}^0 = \mathbf{x}_0, \quad \mathbf{g}(\mathbf{x}^N) = \mathbf{0} \quad (6.19c)$$

$$\mathbf{h}(\mathbf{x}^k, \mathbf{u}^k, t^k) \leq \mathbf{0} \quad k = 0, 1, \dots, N. \quad (6.19d)$$

Im Optimierungsproblem werden die diskretisierten Systemgleichungen (6.19b) zusammen mit den Anfangs- und Endbedingungen (6.19c) als Gleichungsbeschränkungen behandelt. Die Optimierungsvariablen umfassen dabei die diskretisierten Eingangs- und Zustandsvariablen

$$\hat{\mathbf{u}} = \begin{bmatrix} \mathbf{u}^0 \\ \vdots \\ \mathbf{u}^N \end{bmatrix} \in \mathbb{R}^{m(N+1)}, \quad \hat{\mathbf{x}} = \begin{bmatrix} \mathbf{x}^0 \\ \vdots \\ \mathbf{x}^N \end{bmatrix} \in \mathbb{R}^{n(N+1)}. \quad (6.20)$$

Die Anzahl der Optimierungsvariablen, $(m+n)(N+1)$, ist somit zwar deutlich höher als bei der Teildiskretisierung, dafür liegen aber die Differentialgleichungen (6.15b) bereits als Gleichungsbeschränkungen vor und müssen nicht separat integriert werden.

Einige Merkmale direkter Verfahren mit Volldiskretisierung sind:

- Die Optimierungsvariablen $\hat{\mathbf{x}}$ erfüllen die diskretisierten Systemgleichungen (6.19b) nur bei der optimalen Lösung $\hat{\mathbf{x}}^*$ hinreichend genau.
- Die Berechnung der Gradienten der Kostenfunktion und Beschränkungen in (6.19) nach $(\hat{\mathbf{x}}, \hat{\mathbf{u}})$ ist einfacher als im Falle der Teildiskretisierung. Des Weiteren sind die Matrizen dünn besetzt und weisen eine Blockstruktur auf, die von speziellen numerischen Verfahren ausgenutzt werden kann.
- Im Gegensatz zu (6.18e) liegen die Ungleichungsbeschränkungen (6.19d) ebenfalls direkt in diskretisierter Form vor.
- Die Genauigkeit der volldiskretisierten optimalen Lösung hängt direkt von der Anzahl der Stützstellen $(N+1)$ ab. Im Gegensatz zur Teildiskretisierung beeinflusst N nicht nur wie fein die Steuerung diskretisiert wird, sondern ebenfalls wie genau $\hat{\mathbf{x}}$ die tatsächliche Lösung der Differentialgleichungen (6.15b) annähert.
- Aufgrund der Volldiskretisierung des Optimalsteuerungsproblems (6.15) spricht man in diesem Zusammenhang auch von *direkten simultanen Verfahren*.

6.3 Beispiel: Navigationsproblem in der Schifffahrt

Als Beispiel für die numerische Lösung von Optimalsteuerungsproblemen wird die Steuerung eines Binnenschiffes betrachtet. Die Basis dafür ist das folgende nichtlineare Modell, das in [3] hergeleitet wurde:

$$\left. \begin{aligned} \dot{x}_1 &= x_2, & x_1(0) &= 0 \\ \dot{x}_2 &= c_1 x_2 + c_2 u, & x_2(0) &= 0 \\ \dot{x}_3 &= c_3 v |x_3| x_3 + c_4 x_2, & x_3(0) &= 0 \end{aligned} \right\} \quad \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, u). \quad \begin{aligned} (6.21a) \\ (6.21b) \\ (6.21c) \end{aligned}$$

Die Bedeutung der auftretenden Variablen ist aus Abbildung 6.4 ersichtlich. Die Schiffparameter

$$c_1 = -0.26 \frac{1}{s}, \quad c_2 = 0.2 \frac{1}{s^2}, \quad c_3 = -1.87 \frac{1}{\text{rad m}}, \quad c_4 = 0.6 \frac{1}{s}$$

wurden an einem realen Binnenschiff identifiziert. Der Ruderwinkel u sei durch maximale Ausschläge

$$u \in [u^-, u^+] \quad (6.22)$$

begrenzt, für die im Folgenden unterschiedliche Werte betrachtet werden.

6.3.1 Indirekter Lösungsweg

Für die Anwendung des indirekten Diskretisierungs- und Schießverfahrens unter MATLAB wird als Optimalsteuerungsproblem ein Richtungswechsel des Schiffes betrachtet.

6.3.1.1 Realisierung eines Richtungswechsels

Es soll ein Richtungswechsel des Schiffes um 45° realisiert werden, was im zu minimierenden Kostenfunktional

$$J(u) = \frac{1}{2} \Delta \mathbf{x}(t_f)^\top \mathbf{S} \Delta \mathbf{x}(t_f) + \frac{1}{2} \int_0^{t_f} \Delta \mathbf{x}^\top \mathbf{Q} \Delta \mathbf{x} + r u^2 dt, \quad \Delta \mathbf{x} = \mathbf{x} - \mathbf{x}_f, \quad \mathbf{x}_f = \begin{bmatrix} \pi/4 \\ 0 \\ 0 \end{bmatrix} \quad (6.23)$$

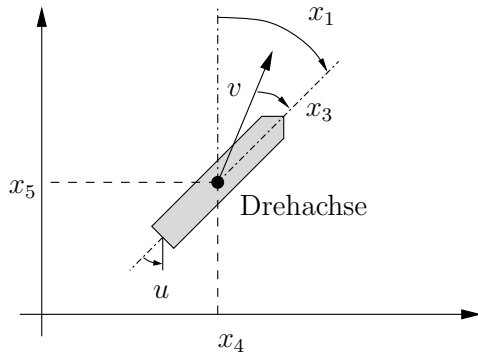
berücksichtigt wird. Die Endzeit und die Geschwindigkeit des Schiffes werden zu $t_f = 15$ s und $v = 3.5$ m/s gesetzt. Der Einfachheit halber werden die Gewichtungsmatrizen als Einheitsmatrizen $\mathbf{S} = \mathbf{Q} = \mathbf{I}$ angesetzt, während $r = 0.1$ einer relativ niedrigen Gewichtung des Ruderwinkels u entspricht.

Da die indirekten Lösungsverfahren auf der numerischen Lösung der Optimalitätsbedingungen basieren, wird zunächst die Hamilton-Funktion aufgestellt

$$H(\mathbf{x}, \boldsymbol{\lambda}, u) = \frac{1}{2} \Delta \mathbf{x}^\top \mathbf{Q} \Delta \mathbf{x} + \frac{r}{2} u^2 + \boldsymbol{\lambda}^\top \mathbf{f}(\mathbf{x}, u), \quad \mathbf{f}(\mathbf{x}, u) = \begin{bmatrix} x_2 \\ c_1 x_2 + c_2 u \\ c_3 v |x_3| x_3 + c_4 x_2 \end{bmatrix}.$$

Das adjungierte System (5.38) und die Randbedingungen (5.41) für die adjungierten Zustände $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \lambda_3]^\top$ lauten

$$\dot{\boldsymbol{\lambda}} = -H_{\mathbf{x}}(\mathbf{x}, u, \boldsymbol{\lambda}) = -\mathbf{Q}(\mathbf{x} - \mathbf{x}_f) - \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right)^\top \boldsymbol{\lambda}, \quad \boldsymbol{\lambda}(t_f) = \mathbf{S}(\mathbf{x}(t_f) - \mathbf{x}_f) \quad (6.24)$$



- x_1 : Vorausswinkel [rad]
- $\dot{x}_1 = x_2$: Drehgeschwindigkeit [rad/s]
- x_3 : Driftwinkel [rad]
- (x_4, x_5) : Position des Schiffes [m]
- u : Ruderwinkel [rad]
- v : Geschwindigkeit [m/s]

Abbildung 6.4: Größen zur Beschreibung der Schifffynamik.

mit $H_{\mathbf{x}} = \frac{\partial H}{\partial \mathbf{x}}$ und der Jacobimatrix

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & c_1 & 0 \\ 0 & c_4 & 2c_3v \operatorname{sign}(x_3)x_3 \end{bmatrix}.$$

Das System liegt in eingangsaffiner Form vor und das Kostenfunktional (6.23) besitzt einen energieoptimalen Anteil mit $r > 0$. Somit liegt der Sonderfall (5.99) des Maximumprinzips vor, so dass sich die optimale Steuerung

$$u = \psi(\mathbf{x}, \boldsymbol{\lambda}) = \begin{cases} u^- & \text{falls } u^0 \leq u^- \\ u^0 & \text{falls } u^0 \in (u^-, u^+), \\ u^+ & \text{falls } u^0 \geq u^+ \end{cases}, \quad \text{mit } u^0 = -\frac{1}{r}c_2\lambda_2. \quad (6.25)$$

ergibt. Die Steuerung u wird in den kanonischen Gleichungen (6.21) und (6.24) ersetzt,

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \psi(\mathbf{x}, \boldsymbol{\lambda})), \quad \dot{\boldsymbol{\lambda}} = -H_{\mathbf{x}}(\mathbf{x}, \psi(\mathbf{x}, \boldsymbol{\lambda}), \boldsymbol{\lambda}), \quad (6.26)$$

was die Basis für die Anwendung des (hier betrachteten) Diskretisierungs- und Schießverfahrens darstellt. Abbildung 6.5 fasst die MATLAB-Funktionen zur Bestimmung der kanonischen Gleichungen zusammen.

6.3.1.2 Diskretisierungsverfahren

Zur Lösung der kanonischen Gleichungen (6.26) mit den Randbedingungen

$$\mathbf{x}(0) = \mathbf{x}_0, \quad \boldsymbol{\lambda}(t_f) = \mathbf{S}(\mathbf{x}(t_f) - \mathbf{x}_f) \quad (6.27)$$

kann die MATLAB-Funktion **bvp4c** verwendet werden, die Zwei-Punkt-Randwertaufgaben mit Hilfe des Kollokationsverfahrens löst (siehe [11, 6] und “**help bvp4c**” unter MATLAB). Abbildung 6.6 stellt den MATLAB-Code zum Aufruf von **bvp4c** dar. Die wichtigsten Übergabeargumente an **bvp4c** sind die rechte Seite der Differentialgleichung (6.26) und die Randbedingungen (6.27) in Residuenform.

Für die Lösung des Randwertproblems wird eine Startschätzung **sol0** benötigt, die über die Funktion **bvpinit** erstellt werden kann. Das Zeitintervall $[0, t_f]$ wird mit 30 Stützstellen t^k initialisiert und die Startwerte der Zustände $\mathbf{x} = [x_1, x_2, x_3]^T$, $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \lambda_3]^T$

```

function p = schiff_param
% -----
p.c = [-0.26, 0.2, -1.87, 0.6];           % Schiffparameter
p.v = 3.5;                               % Geschwindigkeit
p.tf = 15;                               % Endzeit (Optimierungshorizont)

p.x0 = [0; 0; 0];                        % Anfangsbedingungen
p.xf = [45*pi/180; 0; 0];                % (gewünschte) Endbedingungen

p.r = 0.1;                               % Gewichtungen im Integralanteil
p.Q = diag([1,1,1]);                     % Gewichtung des Endzustands
p.S = diag([1,1,1]);

function f = schiff_kangln(t,X,p)
% -----
x = X(1:3); adj = X(4:6);
u = schiff_uopt(x,adj,p);
f = [ fsys(x,u,p);                       % kanonische Gleichungen
      -p.Q*(x-p.xf)-dfdx(x,adj,u,p)'*adj ];

function f = fsys(x,u,p)
% -----
f = [ x(2);                               % Schiffdynamik
      p.c(1)*x(2)+p.c(2)*u;
      p.c(3)*p.v*abs(x(3))*x(3)+p.c(4)*x(2) ];

function J = dfdx(x,adj,u,p)
% -----
J = [ 0, 1, 0;                             % Jacobimatrix
      0, p.c(1), 0;
      0, p.c(4), 2*p.c(3)*p.v*x(3)*sign(x(3)) ];

function u = schiff_uopt(x,adj,p)
% -----
u0 = -1/p.r*adj'*[0; p.c(2); 0];           % unbeschränkte Stellgröße
if u0 > p.umin & u0 < p.umax, u = u0;
elseif u0 <= p.umin, u = p.umin;
else u = p.umax;
end

```

Abbildung 6.5: MATLAB-Funktionen zur Berechnung der kanonischen Gleichungen für das Schifffproblem.

auf diesem Intervall als konstant angesetzt (siehe Abbildung 6.6 und “help bvpinit” unter MATLAB). Die letzten beiden Übergabeargumente sind Optionen für `bvp4c` und die Parameterstruktur `p`.

Abbildung 6.7 zeigt die optimalen Trajektorien für den 45° -Richtungswechsel des Schiffes für verschiedene Beschränkungen des Ruderwinkels $u \in [u^-, u^+]$. Offensichtlich ist die Beschränkung $u \in [-5, 5]^\circ$ zu restriktiv, um die 45° -Drehung in der festen Endzeit $t_f = 15$ s zu ermöglichen. Rechts in Abbildung 6.7 ist die Fortbewegung des Schiffes auf dem Fluss dargestellt, wobei sich die Position (x_4, x_5) des Schiffes (siehe Abbildung 6.4) aus den folgenden Differentialgleichungen durch Integration ergibt:

$$\dot{x}_4 = v \sin(x_1 - x_3), \quad \dot{x}_5 = v \cos(x_1 - x_3), \quad x_4(0) = x_5(0) = 0. \quad (6.28)$$

Aufgabe 6.1 Verifizieren Sie die Ergebnisse in Abbildung 6.7 unter MATLAB mit Hilfe von `bvp4c` (siehe Abbildungen 6.5 und 6.6). Untersuchen Sie numerisch den Einfluss der Gewichtungen im Kostenfunktional (6.23) auf den Verlauf der optimalen Trajektorien $\mathbf{x}^*(t)$, $t \in [0, t_f]$.

```

function [vec,sol,p] = schiff_rwa(umin,umax)
% -----
p = schiff_param;                                % Schiffparameter
p.umin = umin*pi/180;                            % Stellgrößenbeschränkungen
p.umax = umax*pi/180;                            % (Übergabe in Grad)

opt = bvpset('Stats','on','RelTol',1e-4);        % Optionen für bvp4c
sol0 = bvpinit(linspace(0,p.tf,30),[p.x0;zeros(3,1)]); % Startlösung
sol = bvp4c(@schiff_kangln, @schiff_rb, sol0, opt, p); % Aufruf von bvp4c

vec.t = linspace(0,p.tf,100);                    % optimale Lösung
vec.x = interp1(sol.x,sol.y(1:3,:),vec.t)';
vec.adj = interp1(sol.x,sol.y(4:6,:),vec.t)';
for i=1:length(vec.t),
    vec.u(:,i) = schiff_uopt(vec.x(:,i),vec.adj(:,i),p); % Steuerung
end

% -----
function res = schiff_rb(X0,Xf,p)                  % Randbedingungen
x0 = X0(1:3); xf = Xf(1:3); adjf = Xf(4:6);
res = [ x0 - p.x0;
        adjf - p.S*(xf-p.xf) ];

```

Abbildung 6.6: MATLAB-Code zur Berechnung des optimalen Richtungswechsels des Schiffes mittels des Kollokationsverfahrens (bvp4c).

6.3.1.3 Schießverfahren

Beim Schießverfahren werden die kanonischen Gleichungen (6.26) mit den Anfangsbedingungen $\mathbf{x}(0) = \mathbf{x}_0$ und den noch unbekannten Werten $\boldsymbol{\lambda}(0) = \boldsymbol{\lambda}_0$ integriert. Die sich dadurch ergebende Trajektorie der adjungierte Zustände $\boldsymbol{\lambda}(t; \boldsymbol{\lambda}_0)$ muss die Endbedingungen in (6.27) erfüllen, was zu der folgenden Gleichung in Residuenform führt

$$\boldsymbol{\lambda}(t_f; \boldsymbol{\lambda}_0) - \mathbf{S}(\mathbf{x}(t_f; \boldsymbol{\lambda}_0) - \mathbf{x}_f) = \mathbf{0}.$$

Die Endbedingung in Abhängigkeit von $\boldsymbol{\lambda}_0$ wird mit der MATLAB-Funktion `fsolve` aus der Optimization Toolbox gelöst (siehe auch Abschnitt 5.3.3), während die unterlagerte numerische Integration der Differentialgleichungen (6.26) mit einem ODE-Solver von MATLAB durchgeführt wird. Abbildung 6.8 fasst den entsprechenden MATLAB-Code zur Lösung des Schiffproblems zusammen.

Für die numerische Integration muss eine hohe Genauigkeit verwendet werden, da die Instabilität der kanonischen Gleichungen zu einer schlechten Konditionierung der numerischen Integration führt. Des Weiteren ist das Schießverfahren recht sensitiv hinsichtlich der Startwerte von $\boldsymbol{\lambda}_0$, die daher nahe genug an der optimalen Lösung liegen müssen.

Aufgabe 6.2 Überprüfen Sie die Ergebnisse in Abbildung 6.7 unter MATLAB mit Hilfe des Schießverfahrens (MATLAB-Code in Abbildung 6.5 und 6.8) für die Startwerte $\boldsymbol{\lambda}_0 = [-5, -10, 0.1]^T$. Untersuchen Sie die numerische Sensitivität des Schießverfahrens durch Variation von $\boldsymbol{\lambda}_0$ und unterschiedliche Toleranzgrenzen für die numerische Integration (siehe `help odeset` unter MATLAB). Lösen Sie des Weiteren die in Abbildung 6.7 betrachteten Fälle mit Hilfe des sogenannten Fortsetzungsverfahrens, indem Sie für den (einfach zu lösenden) Fall $u^\pm = \pm 5^\circ$ die Startschätzung $\boldsymbol{\lambda}_0 = \mathbf{0}$ ansetzen, anschließend die Schranken sukzessive auf $u^\pm = \pm 15^\circ$ erhöhen und die vorherigen Lösungen $\boldsymbol{\lambda}_0^*$ als neue Startschätzungen verwenden.

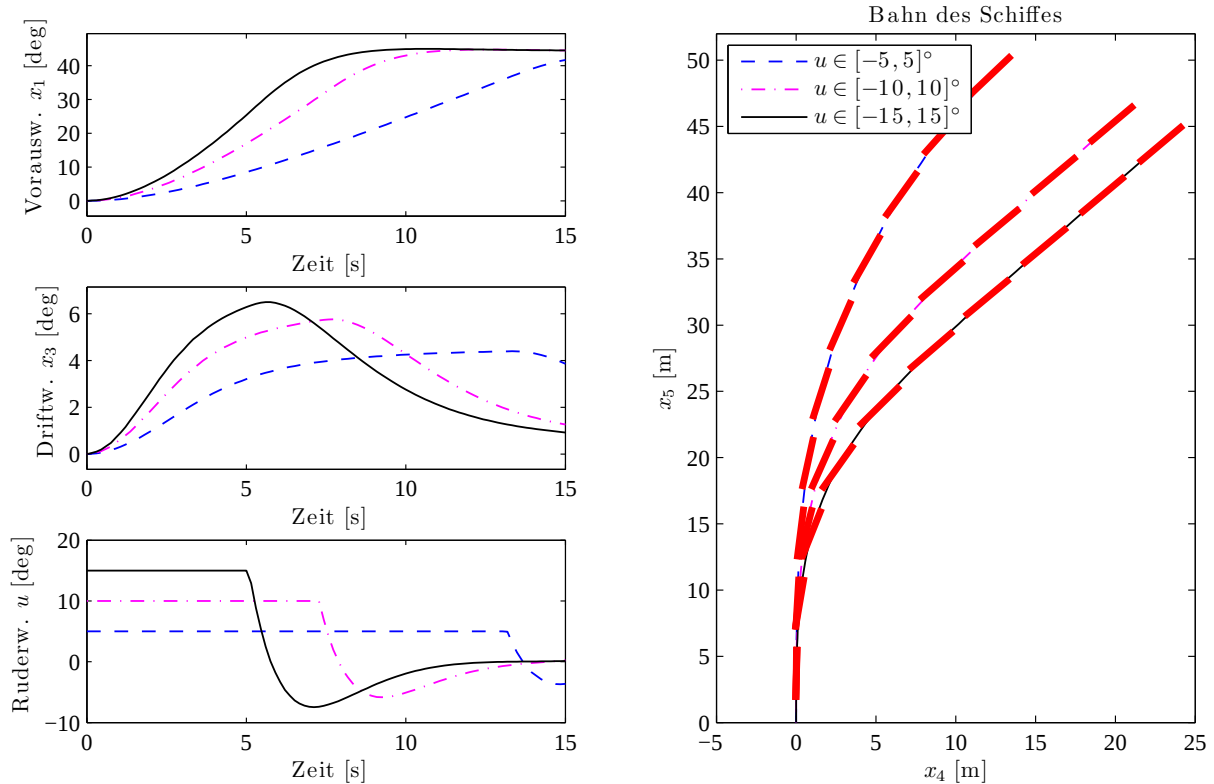


Abbildung 6.7: Optimale Trajektorien für den Richtungswechsel des Schiffes mit verschiedenen Beschränkungen des Ruderwinkels $u \in [u^-, u^+]$.

6.3.2 Direkter Lösungsweg

Beim direkten Lösungsweg wird lediglich die Volldiskretisierung (Abschnitt 6.2.2) betrachtet, da die Teildiskretisierung in Verbindung mit nichtlinearen Optimierern aufgrund der unterlagerten Integration in der Regel komplizierter ist.

6.3.2.1 Realisierung eines Ausweichmanövers

Als Erweiterung des Richtungswechsels soll ein Ausweichmanöver betrachtet werden, um z.B. ein entgegenkommendes Schiff zu passieren. Das Schiffmodell (6.21) wird dazu um die erste Differentialgleichung in (6.28) erweitert. Somit ergibt sich das Optimalsteuerungsproblem

$$\min_{u(\cdot)} \quad J(u) = \Delta \mathbf{x}(t_f)^\top \mathbf{S} \Delta \mathbf{x}(t_f) + \frac{1}{2} \int_0^{t_f} \Delta \mathbf{x}^\top \mathbf{Q} \Delta \mathbf{x} + r u^2 dt \quad (6.29a)$$

$$\text{u.B.v.} \quad \dot{x}_1 = x_2, \quad x_1(0) = 0 \quad (6.29b)$$

$$\dot{x}_2 = c_1 x_2 + c_2 u, \quad x_2(0) = 0 \quad (6.29c)$$

$$\dot{x}_3 = c_3 v |x_3| x_3 + c_4 x_2, \quad x_3(0) = 0 \quad (6.29d)$$

$$\dot{x}_4 = v \sin(x_1 - x_3), \quad x_4(0) = 0 \quad (6.29e)$$

$$u(t) \in [u^-, u^+] \quad \forall t \in [0, t_f]. \quad (6.29f)$$

```

function [adj0opt,vec,p] = schiff_schiess(umin,umax,adj0)
% -----
p = schiff_param;                                % Schiffparameter
p.umin = umin*pi/180;                            % Stellgrößenbeschränkungen
p.umax = umax*pi/180;                            % (Übergabe in Grad)

optoptim = optimset('Display','iter');
optode = odeset('RelTol',1e-13,'AbsTol',1e-14);    % hohe Integrationsgenauigkeit notwendig
adj0opt = fsolve(@schiff_res,adj0,optoptim,p,optode); % Aufruf von fsolve (Startlösung adj0)

vec.t = linspace(0,p.tf,100);
[t,X] = ode113(@schiff_kangln,vec.t,[p.x0;adj0opt],optode,p); % letzte Integration für optimale Lösung
vec.x = X(:,1:3)';
vec.adj = X(:,4:6)';
for i=1:length(vec.t),
    vec.u(:,i) = schiff_uopt(vec.x(:,i),vec.adj(:,i),p);    % optimale Steuerung
end

% -----
function res = schiff_res(adj0,p,optode)
X0 = [p.x0; adj0];
[t,X] = ode113(@schiff_kangln,[0,p.tf],X0,optode,p);    % Integration der kanonischen Gln.
xf = X(end,1:3)';
adjf = X(end,4:6)';
res = adjf - p.S*(xf-p.xf);                            % Endbedingungen in Residuenform

```

Abbildung 6.8: MATLAB-Code zur Berechnung des optimalen Richtungswechsels des Schiffes mittels des Schießverfahrens (`fsolve`).

Im Kostenfunktional (6.29a) geht die Abweichung $\Delta \mathbf{x} = \mathbf{x} - \mathbf{x}_f$ von der gewünschten Endposition ein. Da das Schiff um eine Strecke von 10 m in x_4 -Richtung (d.h. quer zur Fahrtrichtung, siehe Abbildung 6.4 und 6.6) ausweichen soll, wird

$$\mathbf{x}_f = [0, 0, 0, 10 \text{ m}]^T$$

gesetzt. Analog zum vorherigen Abschnitt werden die Gewichtungsmatrizen als Einheitsmatrizen $\mathbf{S} = \mathbf{Q} = \mathbf{I}$ angesetzt, während die Stellgröße mit $r = 0.1$ gewichtet wird.

6.3.2.2 Volldiskretisierung mit AMPL/IPOPT

Das Optimalsteuerungsproblem (6.29) wird mit Hilfe der Trapezregel volldiskretisiert. Das resultierende nichtlineare Optimierungsproblem (6.19) kann dann mit einem herkömmlichen nichtlinearen Optimierer numerisch gelöst werden. Im Folgenden wird dazu der Interior-Point-Solver IPOPT³ verwendet. Das diskretisierte Problem (6.19) wird mit Hilfe der Modellierungssprache AMPL⁴ unter IPOPT implementiert, die über eine eigene Syntax zur Beschreibung von nichtlinearen Optimierungsproblemen verfügt.

Abbildung 6.9 stellt den AMPL-Code des diskretisierten nichtlinearen Optimierungsproblems (6.19) für das betrachtete Ausweichmanöver des Schiffes (6.29) dar. Entsprechend der Struktur des nichtlinearen Problems (6.19) wird das diskretisierte Kostenfunktional (6.29a) durch “minimize” definiert, während die diskretisierten Systemgleichungen und Anfangsbedingungen (6.29b) durch “subject to” gekennzeichnet sind. Für eine ausführliche Beschreibung der Syntax von AMPL sei auf das Buch [5] und die zahlreichen Beschreibungen im Internet verwiesen.⁵

³Frei erhältlich unter <https://projects.coin-or.org/Ipopt>.

⁴Eine Studentenversion ist erhältlich unter <http://www.ampl.com>.

⁵Eine Auswahl an AMPL-Skripten findet man unter <http://www.princeton.edu/~rvdb/>.

```

param pi      := 3.14159265358979;
param c1      := -0.26;           # Schiffparameter
param c2      := 0.2;
param c3      := -1.87;
param c4      := 0.6;
param v       := 3.5;           # Geschwindigkeit
param umin    := -15*pi/180;     # Stellgrößenbeschränkung
param umax    := 15*pi/180;
param tf      := 20;           # Endzeit

set Nx := {1..4};
param x0{i in Nx}; let x0[1]:=0; let x0[2]:=0; let x0[3]:=0; let x0[4]:=0; # Anfangsbedingungen
param xf{i in Nx}; let xf[1]:=0; let xf[2]:=0; let xf[3]:=0; let xf[4]:=10; # Endbedingungen
param S {i in Nx}; let S[1]:=1; let S[2]:=1; let S[3]:=1; let S[4]:=1; # Endzustandsgew.
param Q {i in Nx}; let Q[1]:=1; let Q[2]:=1; let Q[3]:=1; let Q[4]:=1; # Integralgew.
param R:=10;

param N       := 59;           # Diskretisierungspunkte
param h       := tf/N;         # konstante Schrittweite h = t[i+1]-t[i]
set Nt       := {0..N} ;
set Nt1      := {0..N-1};

# Optimierungsvariablen
# -----
var u{Nt} <= umax , >= umin;    # Stellgrößenbeschränkung
var x{Nx, Nt};

# Diskretisiertes Kostenfunktional (Trapezregel)
# -----
minimize cost:
    sum{i in Nx}( S[i]*(x[i,N]-xf[i])^2 )/2 + # Endzustandsbewertung
    h/2 * sum{k in Nt1}( # Integralanteil
        sum{i in Nx}( Q[i]*( (x[i,k]-xf[i])^2 + (x[i,k+1]-xf[i])^2 )) + R*(u[k]^2 + u[k+1]^2) )/2 ;

# Diskretisierte Systemgleichungen (Trapezregel)
# -----
subject to x1_diskr {k in Nt1} :
    x[1,k+1]-x[1,k] = h*( x[2,k] + # f1(x[k], u[k])
                        x[2,k+1] )/2 ; # f1(x[k+1],u[k+1])
subject to x2_diskr {k in Nt1} :
    x[2,k+1]-x[2,k] = h*( c1*x[2,k]+c2*u[k] + # f2(x[k], u[k])
                        c1*x[2,k+1]+c2*u[k+1] )/2 ; # f2(x[k+1],u[k+1])
subject to x3_diskr {k in Nt1} :
    x[3,k+1]-x[3,k] = h*( c3*v*abs(x[3,k])*x[3,k]+c4*x[2,k] + # f3(x[k], u[k])
                        c3*v*abs(x[3,k+1])*x[3,k+1]+c4*x[2,k+1] )/2 ; # f3(x[k+1],u[k+1])
subject to x4_diskr {k in Nt1} :
    x[4,k+1]-x[4,k] = h*( v*sin(x[1,k]-x[3,k]) + # f4(x[k], u[k])
                        v*sin(x[1,k+1]-x[3,k+1]) )/2 ; # f4(x[k+1],u[k+1])

# Anfangsbedingungen
# -----
subject to x_init {i in Nx} : x[i,0] = x0[i] ;

# Loesung und Ausgabe
# -----
option solver ipopt;
solve;
display cost ;
printf {k in Nt}: "%10.5f %10.5f %10.5f %10.5f %10.5f %10.5f\n",
    k*h, x[1,k], x[2,k], x[3,k], x[4,k], u[k] > schiff_AMPL_sol.txt;
printf "Lösung wurde in 'schiff_AMPL_sol.txt' gespeichert.";

```

Abbildung 6.9: AMPL-Datei schiff_AMPL.mod zur Realisierung des Ausweichmanövers des Schiffes mit Hilfe von IPOPT.

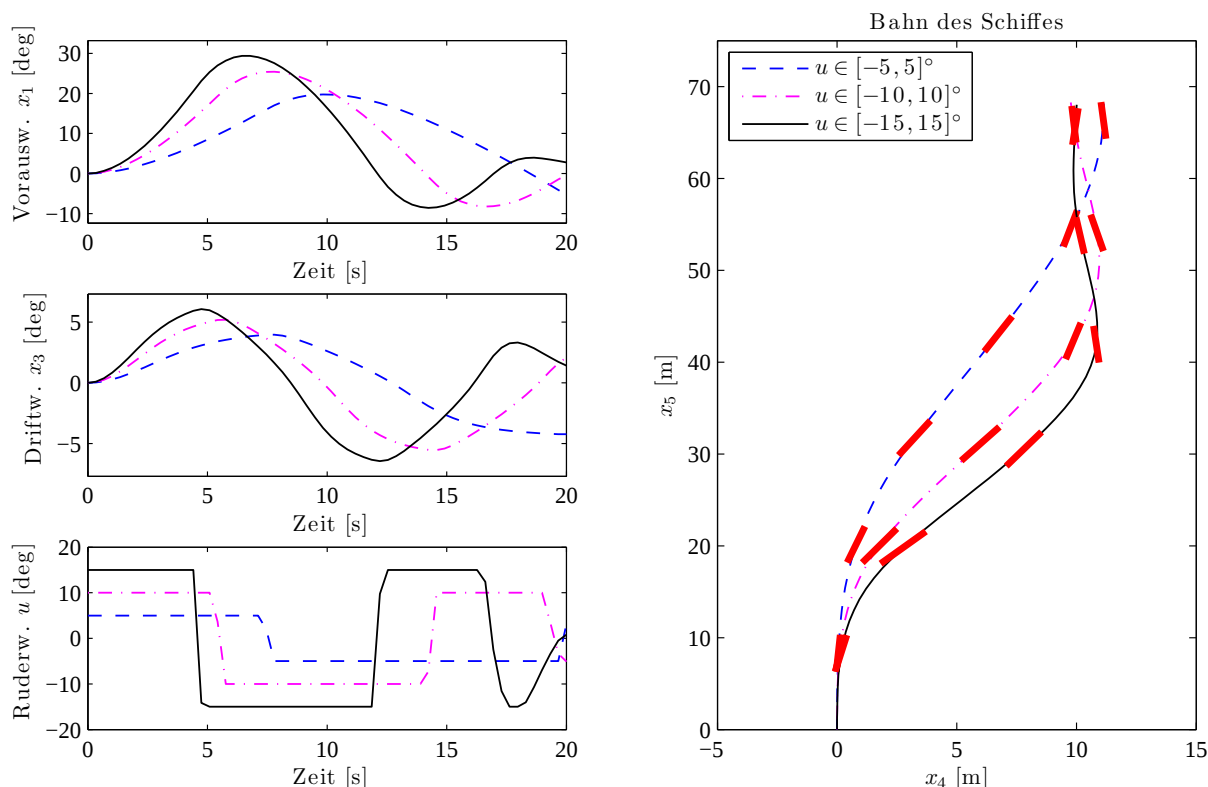


Abbildung 6.10: Ausweichmanöver des Schiffes mit verschiedenen Beschränkungen des Ruderwinkels $u \in [u^-, u^+]$ (berechnet mit AMPL/IPOPT).

Nach der Deklaration wird das Optimierungsproblem mit Hilfe des Befehls `solve` gelöst, wobei “`option solver ipopt`” AMPL mitteilt, dass IPOPT aufgerufen werden soll. Der “`printf`”-Befehl schreibt die Optimierungsvariablen, also die diskretisierte näherungsweise Lösung $\mathbf{x}^*(t)$, $u^*(t)$, $t \in [0, t_f]$ in die Datei `schiff_AMPL_sol.txt`. Die Daten können dann unter MATLAB mit Hilfe der Funktion `load` eingelesen werden.

Abbildung 6.10 zeigt die Ergebnisse von AMPL/IPOPT für das Ausweichmanöver des Schiffes mit der Endzeit $t_f = 20$ s und verschiedenen Ruderbeschränkungen $u \in [u^-, u^+]$. Die Steuerung weist ein recht aggressives Verhalten auf, da die Abweichung $x_4 - x_{f,4}$ vom gewünschten Versatz $x_{f,4} = 10$ m maßgeblichen Einfluß auf das Kostenfunktional (6.29a) besitzt. Ein langsamerer Übergang kann durch Erhöhung des Gewichtungsparameters r für die Stellgröße u erreicht werden.

Eine Alternative zur eigenen Installation von IPOPT oder einem anderen Optimierer ist der Internet-Dienst NEOS⁶ (*Network Enabled Optimization Server*), an den AMPL-Skripte gesendet werden können. NEOS bietet eine Auswahl an Solvern, unter denen ein geeigneter Optimierer ausgewählt werden kann. Die Lösung wird dann entweder über ein Web-Interface ausgegeben oder per Email zugesendet.

⁶<http://neos.mcs.anl.gov/>

6.4 Software–Übersicht

Neben der beschriebenen Möglichkeit, ein Optimalsteuerungsproblem eigenhändig zu diskretisieren und als nichtlineares Optimierungsproblem zu lösen, gibt es verschiedene Software–Pakete für dynamische Optimierungsprobleme, die großteils auf direkter Teil– oder Volldiskretisierung basieren. Eine kleine Auswahl an frei erhältlichen und kostenpflichtigen Programmen ist im Folgenden zusammengetragen:

- HQP (Huge Quadratic Programming)
<http://hqp.sourceforge.net/>
- MUSCOD II (MUltiple Shooting CODe for optimization)
<http://www.iwr.uni-heidelberg.de/~agbock/RESEARCH/muscod.php>
- NTG (Nonlinear Trajectory Generation), siehe auch [7]
http://www.cds.caltech.edu/~murray/software/2002a_ntg.html
- SOCS (Sparse Optimal Control Software), siehe auch [2]
<http://www.boeing.com/phantom/socs>
- TOMLAB/PROPT
<http://tomdyn.com>

Literatur

- [1] U.M. Ascher, R.M.M. Mattheij und R.D. Russell. *Numerical solution of boundary value problems of ordinary differential equations*. Prentice Hall, 1988.
- [2] J.T. Betts. *Practical Methods for Optimal Control Using Nonlinear Programming*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2001.
- [3] R. Bittner, A. Driescher und E.D. Gilles. Drift dynamics modeling for automatic track-keeping of inland vessels. In *10th Saint Petersburg International Conference on Integrated Navigation Systems*, Seiten 218–227, St. Petersburg (Rußland), 2003.
- [4] A.E. Bryson und Y.-C. Ho. *Applied Optimal Control*. John Wiley & Sons, New York, 1975.
- [5] R. Fourer, D.M. Gay und B.W. Kernighan. *AMPL: A modeling language for mathematical programming*. Duxbury, 2002.
- [6] J. Kierzenka und L.F. Shampine. A BVP solver based on residual control and the MATLAB PSE. *ACM Transactions on Mathematical Software*, 27:299–316, 2001.
- [7] M.B. Milam. *Real-Time Trajectory Generation for Constrained Dynamical Systems*. PhD thesis, California Institute of Technology, 2003.
- [8] M. Papageorgiou. *Optimierung*. Oldenbourg Verlag, 1991.

- [9] H.J. Pesch. Optimal and nearly optimal guidance by multiple shooting. In Centre d'Études Spatiales (CNES), editor, *Proc. Intern. Symp. "Mécanique Spatiale"*, Seiten 761–771, Toulouse, 6.–10.11.1989, 1990. Cepadues Editions. Erhältlich unter <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.54.8387>.
- [10] E. Polak. An historical survey of computational methods in optimal control. *SIAM Review*, 15:553–584, 1973.
- [11] L.F. Shampine, J. Kierzenka und M.W. Reichelt. Solving boundary value problems for ordinary differential equations in MATLAB with **bvp4c**. http://www.mathworks.com/bvp_tutorial, 2000.

A.1 Aufgaben in Kapitel 2

Lösung von Aufgabe 2.1 (Seite 18)

Kostenfunktion $f(\mathbf{x}) = x_1^4 + x_1^2 - 2x_1x_2 + x_2^2$	<code>plot_fcts_convexity(1)</code>
--	-------------------------------------

- Gradient $\nabla f(\mathbf{x}) = \begin{bmatrix} 4x_1^3 + 2x_1 - 2x_2 \\ -2x_1 + 2x_2 \end{bmatrix}$
- Hessematrix $\nabla^2 f(\mathbf{x}) = \begin{bmatrix} 12x_1^2 + 2 & -2 \\ -2 & 2 \end{bmatrix}$

Überprüfung der Definitheit von $\nabla^2 f(\mathbf{x})$ anhand der Kriterien (1)–(3) in Lemma 2.2:

- Kriterium (1):

$$\begin{aligned}
 \mathbf{p}^\top \nabla^2 f(\mathbf{x}) \mathbf{p} &= \begin{bmatrix} p_1 & p_2 \end{bmatrix} \begin{bmatrix} 12x_1^2 + 2 & -2 \\ -2 & 2 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} \\
 &= (12x_1^2 + 2)p_1^2 + 2p_2^2 - 4p_1p_2 \\
 &= 2(p_1 - p_2)^2 + 12x_1^2p_1^2 \\
 &\geq 0 \quad \forall \mathbf{x} \in \mathbb{R}^2, \quad \forall \mathbf{p} \in \mathbb{R}^2, \quad \mathbf{p} \neq \mathbf{0}.
 \end{aligned}$$

- Kriterium (2): $|\lambda \mathbf{I} - \mathbf{A}| = 0 \Rightarrow$ Eigenwerte $\lambda_{1,2}(\mathbf{x}) = 2 + 6x_1^2 \pm \sqrt{4 + 36x_1^4}$.
Mit $\sqrt{4 + 36x_1^2} \leq \sqrt{4 + 24x_1^2 + 36x_1^4} = 2 + 6x_1^2$ folgt

$$\begin{aligned}
 \lambda_1(\mathbf{x}) &= 2 + 6x_1^2 - \sqrt{4 + 36x_1^4} \geq 2 + 6x_1^2 - 2 - 6x_1^2 = 0. \\
 &\Rightarrow \lambda_{1,2}(\mathbf{x}) \geq 0 \quad \forall \mathbf{x} \in \mathbb{R}^2
 \end{aligned}$$

- Kriterium (3): Hauptunterdet. $D_1(\mathbf{x})$ und $D_2(\mathbf{x})$ der Hessematrix $\nabla^2 f(\mathbf{x})$:

$$D_1(\mathbf{x}) = 12x_1^2 + 2 > 0, \quad D_2(\mathbf{x}) = |\nabla^2 f(\mathbf{x})| = 24x_1^2 \geq 0 \quad \forall \mathbf{x} \in \mathcal{X}.$$

$\Rightarrow f(\mathbf{x})$ ist konvex auf dem gesamten Definitionsbereich $\mathcal{X} = \mathbb{R}^2$

- **Hinweis:** $f(\mathbf{x})$ ist sogar strikt konvex, siehe `plot_fcts_convexity(1)` $\rightarrow$ Beispiel dafür, dass strikte Konvexität nicht notwendigerweise positiv Definitheit gewährleistet!

Kostenfunktion $f(\mathbf{x}) = x_2 + \frac{1}{x_1 x_2}$	plot_fcts_convexity(2)
--	------------------------

- Gradient $\nabla f(\mathbf{x}) = \begin{bmatrix} \frac{-1}{x_1^2 x_2} \\ 1 - \frac{1}{x_1 x_2^2} \end{bmatrix}$
- Hessematrix $\nabla^2 f(\mathbf{x}) = \frac{1}{x_1 x_2} \begin{bmatrix} \frac{2}{x_1^2} & \frac{1}{x_1 x_2} \\ \frac{1}{x_1 x_2} & \frac{2}{x_2^2} \end{bmatrix}$

Überprüfung der Definitheit von $\nabla^2 f(\mathbf{x})$ anhand der Kriterien (1)–(3) in Lemma 2.2:

- Kriterium (1):

$$\begin{aligned} \mathbf{p}^\top \nabla^2 f(\mathbf{x}) \mathbf{p} &= \frac{1}{x_1 x_2} \begin{bmatrix} p_1 & p_2 \end{bmatrix} \begin{bmatrix} \frac{2}{x_1^2} & \frac{1}{x_1 x_2} \\ \frac{1}{x_1 x_2} & \frac{2}{x_2^2} \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} \\ &= \frac{1}{x_1 x_2} \left[\left(\frac{p_1}{x_1} + \frac{p_2}{x_2} \right)^2 + \left(\frac{p_1}{x_1} \right)^2 + \left(\frac{p_2}{x_2} \right)^2 \right] \\ &> 0 \quad \forall x_{1,2} > 0, \quad \forall \mathbf{p} \in \mathbb{R}^2, \quad \mathbf{p} \neq \mathbf{0}. \end{aligned}$$

- Kriterium (2): $|\lambda \mathbf{I} - \mathbf{A}| = 0 \Rightarrow$ Eigenwerte

$$\lambda_{1,2}(\mathbf{x}) = \frac{x_1^3 x_2 + x_1 x_2^3 \pm \sqrt{x_1^2 x_2^2 (x_1^4 - x_1^2 x_2^2 + x_2^4)}}{x_1^4 x_2^4}$$

Mit $\sqrt{x_1^2 x_2^2 (x_1^4 - x_1^2 x_2^2 + x_2^4)} < x_1^3 x_2 + x_1 x_2^3$ folgt, dass $\lambda_{1,2} > 0 \quad \forall x_{1,2} > 0$.

- Kriterium (3): Hauptunterdet. $D_1(\mathbf{x})$ und $D_2(\mathbf{x})$ der Hessematrix $\nabla^2 f(\mathbf{x})$:

$$D_1(\mathbf{x}) = \frac{2}{x_1^3 x_2} > 0, \quad D_2(\mathbf{x}) = |\nabla^2 f(\mathbf{x})| = \frac{3}{x_1^4 x_2^4} > 0 \quad \forall x_{1,2} > 0.$$

$\Rightarrow f(\mathbf{x})$ ist strikt konvex auf dem Definitionsbereich $x_{1,2} > 0$.

Lösung von Aufgabe 2.2 (Seite 18)

f(x) = e^x - 1, x ∈ ℝ :	plot_fcts_convexity(3)
-------------------------	------------------------

$$\nabla f(x) = e^x, \quad \nabla^2 f(x) = e^x > 0 \quad \forall x \in \mathbb{R} \quad \Rightarrow \quad f(x) \text{ ist strikt konvex}$$

f(x) = x_1^\alpha x_2^{1-\alpha}, x_{1,2} > 0, α ∈ [0, 1] :	plot_fcts_convexity(4,alpha)
---	------------------------------

$$\nabla f(\mathbf{x}) = \begin{bmatrix} \alpha x_1^{\alpha-1} x_2^{1-\alpha} \\ (1-\alpha) x_1^\alpha x_2^{-\alpha} \end{bmatrix}, \quad \nabla^2 f(\mathbf{x}) = \alpha(\alpha-1) \begin{bmatrix} x_1^{\alpha-2} x_2^{1-\alpha} & -x_1^{\alpha-1} x_2^{-\alpha} \\ -x_1^{\alpha-1} x_2^{-\alpha} & x_1^\alpha x_2^{-\alpha-1} \end{bmatrix}$$

Unterdeterminanten: $D_1 = \alpha(\alpha-1)x_1^{\alpha-2}x_2^{1-\alpha} < 0$ für $\alpha \in [0, 1]$, $D_2 = |\nabla^2 f(\mathbf{x})| = 0$

$\Rightarrow f(\mathbf{x})$ ist also konkav auf dem Definitionsbereich $x_{1,2} > 0$.

A.2 Aufgaben in Kapitel 3

Lösung von Aufgabe 3.1 (Seite 31)

- Gradient und Hessematrix: $\nabla f(\mathbf{x}) = \begin{bmatrix} \frac{x_1 - x_1^*}{a^2} \\ (x_2 - x_2^*) \end{bmatrix}$, $\nabla^2 f(\mathbf{x}) = \begin{bmatrix} \frac{1}{a^2} & 0 \\ 0 & 1 \end{bmatrix}$

$\Rightarrow$ Eigenwerte sind $\lambda_1 = \lambda_{\min} = 1/a^2$ und $\lambda_2 = \lambda_{\max} = 1$ für $a \in \{1, 5\}$

- Spektrale Konditionszahl $\kappa = \frac{\lambda_{\max}}{\lambda_{\min}} = a^2$

- Konvergenzrate $\left(\frac{\kappa - 1}{\kappa + 1}\right)^2 = \left(\frac{a^2 - 1}{a^2 + 1}\right)^2 = \begin{cases} 0 & \text{für } a = 1 \\ (24/26)^2 & \text{für } a = 5 \end{cases}$

$\Rightarrow a = 1$ (Kreisgleichung): Konvergenz in einer Iteration

$a = 5$ (stark elliptisches Problem): Konvergenzrate von nur ca. 0.85 pro Iteration (siehe Abb. 3.7)

Lösung von Aufgabe 3.2 (Seite 32)

- Ziel: Überprüfung von Abstiegsbed. (3.18): $\nabla f(\mathbf{x}^k)^\top \mathbf{s}^k \stackrel{?}{<} 0$
- mit $\mathbf{s}^k = -\nabla f(\mathbf{x}^k) + \beta^k \mathbf{s}^{k-1}$ aus (3.30b):

$$\nabla f(\mathbf{x}^k)^\top \mathbf{s}^k = -\nabla f(\mathbf{x}^k)^\top \nabla f(\mathbf{x}^k) + \beta^k \nabla f(\mathbf{x}^k)^\top \mathbf{s}^{k-1} \quad (\text{A.30})$$

- Erster Term $-||\nabla f(\mathbf{x}^k)||^2 < 0$ für alle $\mathbf{x}^k \neq \mathbf{x}^*$
- Zweiter Term $\beta^k \nabla f(\mathbf{x}^k)^\top \mathbf{s}^{k-1}$: Betrachtung des Liniensuchproblems (3.20)

$$\min_{\alpha^k > 0} \phi(\alpha^k) \quad \text{mit} \quad \phi(\alpha^k) = f(\mathbf{x}^k + \alpha^k \mathbf{s}^k) \quad (\text{A.31})$$

Für ein optimales α^k gilt $\phi'(\alpha^k) = \nabla f(\underbrace{\mathbf{x}^k + \alpha^k \mathbf{s}^k}_{=\mathbf{x}^{k+1}})^\top \mathbf{s}^k = 0$.

- Somit gilt auch $\nabla f(\mathbf{x}^k)^\top \mathbf{s}^{k-1} = 0$.
- (A.30) $\Rightarrow \nabla f(\mathbf{x}^k)^\top \mathbf{s}^k = -||\nabla f(\mathbf{x}^k)||^2 < 0$

$\Rightarrow \mathbf{s}^k$ ist eine gültige Abstiegsrichtung

Lösung von Aufgabe 3.3 (Seite 33)

- Quadratische Funktion $f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^\top \mathbf{Q} \mathbf{x} + \mathbf{b}^\top \mathbf{x} + c$
- Gradient und Hessematrix $\nabla f(\mathbf{x}) = \mathbf{Q} \mathbf{x} + \mathbf{b}$, $\nabla^2 f(\mathbf{x}) = \mathbf{Q}$

$\Rightarrow$ globales Minimum $\mathbf{x}^* = -\mathbf{Q}^{-1}\mathbf{b}$

- Start von $\mathbf{x}^0$ mit Newton-Richtung $\mathbf{s}^0 = -[\nabla^2 f(\mathbf{x}^0)]^{-1} \nabla f(\mathbf{x}^0) = -\mathbf{Q}^{-1}(\mathbf{Q}\mathbf{x}^0 + \mathbf{b})$:

$$\begin{aligned}\mathbf{x}^1 &= \mathbf{x}^0 + \mathbf{s}^0 \quad (\text{voller Newtonschritt } \alpha^0 = 1) \\ &= \mathbf{x}^0 - \mathbf{Q}^{-1}(\mathbf{Q}\mathbf{x}^0 + \mathbf{b}) \\ &= -\mathbf{Q}^{-1}\mathbf{b} = \mathbf{x}^*\end{aligned}\tag{A.32}$$

$\Rightarrow$ Konvergenz nach einem Schritt für alle $\mathbf{x}^0$.

Lösung von Aufgabe 3.4 (Seite 40)

- **Minimum** $\mathbf{x}^* = [1, 1]^\top$: Gradient und Hessematrix

$$\nabla f(\mathbf{x}) = \begin{bmatrix} -400(x_2 - x_1^2)x_1 + 2(x_1 - 1) \\ 200(x_2 - x_1^2) \end{bmatrix}\tag{A.33}$$

$$\nabla^2 f(\mathbf{x}) = \begin{bmatrix} -400(x_2 - 3x_1^2) + 2 & -400x_1 \\ -400x_1 & 200 \end{bmatrix}\tag{A.34}$$

- Stationaritätsbed. $\nabla f(\mathbf{x}^*) = 0 \Rightarrow x_2^* = x_1^{*2}$ und $2(x_1^* - 1) = 0 \Rightarrow \mathbf{x}^* = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

- Untersuchung der Hessematrix $\nabla^2 f(\mathbf{x}^*) = \begin{bmatrix} 802 & -400 \\ -400 & 200 \end{bmatrix}$:

Hauptunterdet. $D_1 = 802 > 0$, $D_2 = |\nabla^2 f(\mathbf{x}^*)| = 400 \Rightarrow \nabla^2 f(\mathbf{x}^*)$ ist positiv definit

$\Rightarrow \mathbf{x}^* = [1, 1]^\top$ ist ein striktes lokales Minimum

- $\mathbf{x}^*$ ist einzige Lösung von $\nabla f(\mathbf{x}^*) = 0 \Rightarrow$ eindeutiges globales Minimum
- **Konvexität:** wähle z.B. $\mathbf{x} = [1, 1]^\top$ und $\mathbf{y} = [-1, 1]^\top$

$\Rightarrow f(\mathbf{x})$ schneidet die Verbindungslinie zwischen $\mathbf{x}$ und $\mathbf{y}$

$\Rightarrow f(\mathbf{x})$ und somit das OP (3.50) sind nicht konvex

- Andere Erklärung: eine konvexe Funktion hat konvexe Untermengen
 $\mathcal{S} = \{\mathbf{x} \in \mathcal{X} : f(\mathbf{x}) \leq k\}$, was nicht erfüllt ist (siehe Höhenlinien in Abbildung 3.10).

Lösung von Aufgabe 3.5 (Seite 42)

- Eigenwerte von $\nabla^2 f(\mathbf{x}^*)$, $\mathbf{x}^* = [1, 1]^T$

$$|\lambda \mathbf{I} - \nabla^2 f(\mathbf{x}^*)| = (\lambda - 802)(\lambda - 200) - 160000 = 0 \quad (\text{A.35})$$

$$\Rightarrow \lambda_{1,2} = 501 \pm \sqrt{250601} \quad (\text{A.36})$$

$$\lambda_1 = \lambda_{\max} \approx 1001.6 \quad (\text{A.37})$$

$$\lambda_2 = \lambda_{\min} \approx 0.399 \quad (\text{A.38})$$

- approx. Konvergenzrate (Satz 3.4 gilt nur in Nähe von $\mathbf{x}^*$)

$$\left(\frac{\kappa - 1}{\kappa + 1}\right)^2 = \left(\frac{\lambda_{\max} - \lambda_{\min}}{\lambda_{\max} + \lambda_{\min}}\right)^2 \approx 0.9984, \quad (\text{A.39})$$

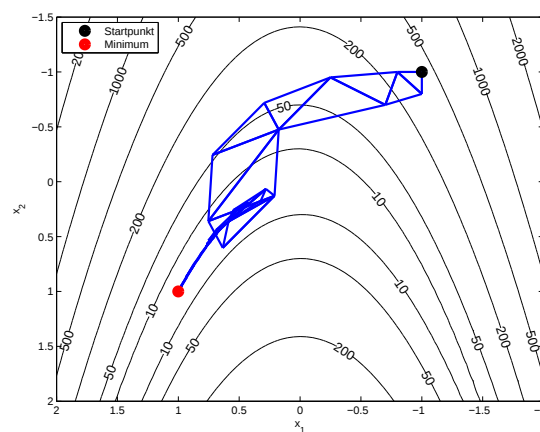
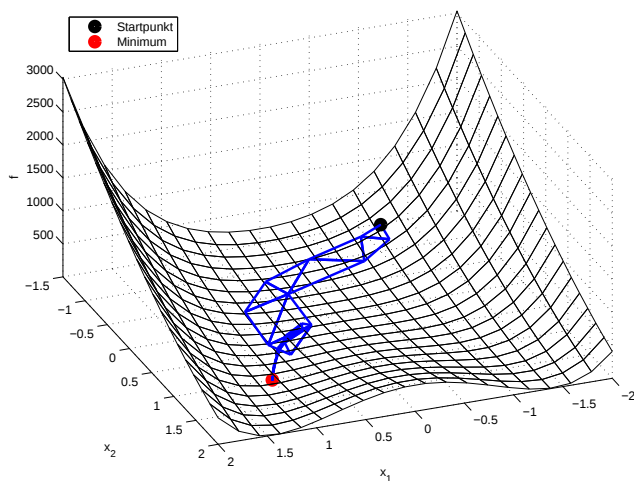
$\Rightarrow$ sehr langsame Konvergenz des Gradientenverfahrens

Lösung von Aufgabe 3.6 (Seite 43)

... siehe Matlab-file rosenbrock_simplex.m ...

s0	Iter.	N(f)	f(xmin)	f(xmax)	f(xmin)-f(xmax)
0.1	69	131	5.32792e-010	1.46524e-009	9.3245e-010
0.2	63	121	2.48917e-010	1.1487e-009	8.99783e-010
1.0	65	123	6.33426e-010	1.07261e-009	4.3918e-010
1.5	61	115	2.8336e-010	6.82877e-010	3.99517e-010

Anfängliche Simplex-Seitenlänge: 0.2



Anfängliche Simplex-Seitenlänge: 1.0

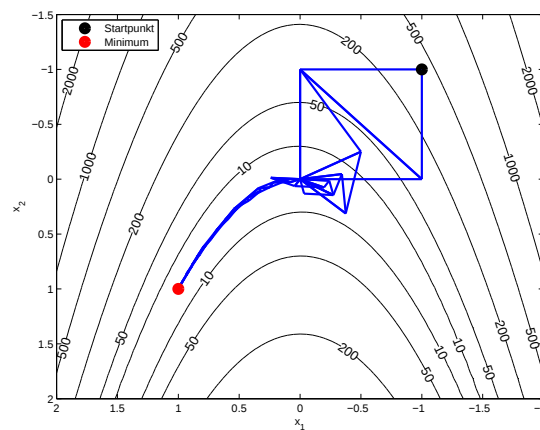
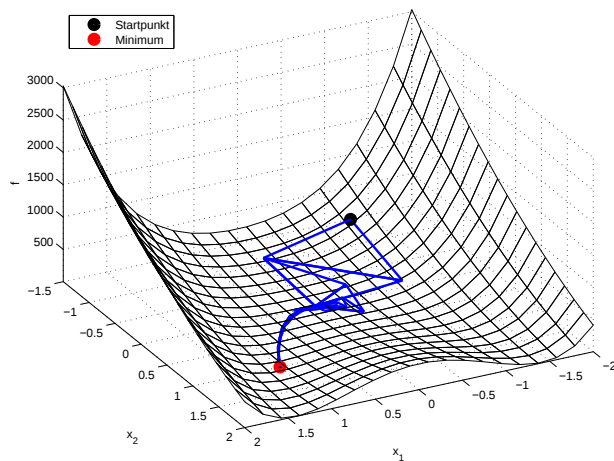


Abbildung A.11: Rosenbrock's Bananenfunktion: Simplex-Verfahren nach Nelder/Mead.

A.3 Aufgaben in Kapitel 4

Lösung von Aufgabe 4.1 (Seite 50)

- zulässige Menge $\mathcal{X}_{\text{ad}} = \{\mathbf{x} \in \mathbb{R}^2 : h_1(\mathbf{x}) \leq 0, h_2(\mathbf{x}) \leq 0\}$

- optimale Lösung $\mathbf{x}^* = [1, 1]^\top$ liegt auf der Ecke von $\mathcal{X}_{\text{ad}}$

$\Rightarrow$ beide Beschr. sind aktiv: $h_1(\mathbf{x}^*) = 0, h_2(\mathbf{x}^*) = 0$ (siehe Abbildung A.12)

$\Rightarrow$ Kompl.bed. $\mu_1^* h_1(\mathbf{x}^*) = 0$ und $\mu_2^* h_2(\mathbf{x}^*) = 0$ in (4.22) sind erfüllt

- Gradienten $\nabla f(\mathbf{x}) = \begin{bmatrix} 2(x_1 - 2) \\ 2(x_2 - 1) \end{bmatrix}$, $\nabla h_1(\mathbf{x}) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, $\nabla h_2(\mathbf{x}) = \begin{bmatrix} 2x_1 \\ -1 \end{bmatrix}$
- Lagrange-Bedingung $\nabla_{\mathbf{x}} L(\mathbf{x}^*, \boldsymbol{\mu}^*) = 0$ in (4.22)

$$\nabla_{\mathbf{x}} L(\mathbf{x}^*, \boldsymbol{\mu}^*) = \begin{bmatrix} -2 \\ 0 \end{bmatrix} + \mu_1^* \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \mu_2^* \begin{bmatrix} 2 \\ -1 \end{bmatrix} = \mathbf{0}$$

liefert $\mu_1^* = \frac{2}{3}, \mu_2^* = \frac{2}{3}$.

$\Rightarrow$ Notw. Optimalitätsbedingungen (4.22) sind für $\mathbf{x}^*$ erfüllt.

- Anschaulich: $-\nabla f(\mathbf{x}^*)$ liegt im Kegel, der durch $\nabla h_1(\mathbf{x}^*), \nabla h_2(\mathbf{x}^*)$ gebildet wird

- Zu Abschnitt 4.2.1: $\nabla \mathbf{h}_{\mathcal{A}(\mathbf{x}^*)}(\mathbf{x}^*) = \begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix}$ mit $|\cdot| = -1 - 2 = -3$

$\Rightarrow$ Beschränkungsqualifikation ist erfüllt

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^2} \quad & f(\mathbf{x}) = (x_1 - 2)^2 + (x_2 - 1)^2 \\ \text{u.B.v.} \quad & h_1(\mathbf{x}) = x_1 + x_2 - 2 \leq 0 \\ & h_2(\mathbf{x}) = x_1^2 - x_2 \leq 0 \end{aligned}$$

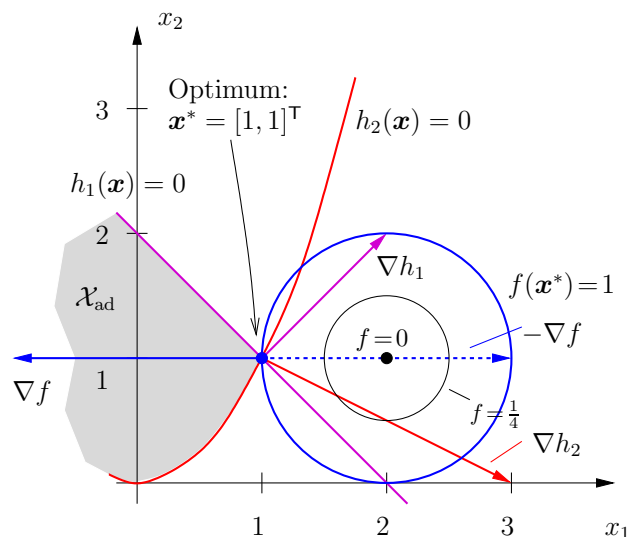


Abbildung A.12: Aufgabe 4.1 mit zwei Ungleichungsbeschränkungen.

Lösung von Aufgabe 4.2 (Seite 52)

- Lagrange-Funktion

$$L(\mathbf{x}, \boldsymbol{\mu}) = \overbrace{(x_1 - 2)^2 + \frac{1}{9}(x_2 - 2)^2}^{= f(\mathbf{x})} + \mu_1 \overbrace{(2x_1 + x_2 - 2)}^{= h_1(\mathbf{x})} + \mu_2 \overbrace{(x_1 - x_2 - 1)}^{= h_2(\mathbf{x})}$$

$$\nabla_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\mu}) = \begin{bmatrix} 2(x_1 - 2) \\ \frac{2}{9}(x_2 - 2) \end{bmatrix} + \mu_1 \begin{bmatrix} 2 \\ 1 \end{bmatrix} + \mu_2 \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

Fälle	Beschr.	KB (4.29e)	Gl.system	Lösung	stat. Punkt?
a) unbeschr.	$h_1(\mathbf{x}) < 0$ $h_2(\mathbf{x}) < 0$	$\mu_1 = 0$ $\mu_2 = 0$	$\nabla_{\mathbf{x}} L(\mathbf{x}) = \mathbf{0}$	$\mathbf{x} = [2, 2]^T$	nein ($\mathbf{x} \notin \mathcal{X}_{\text{ad}}$)
b) h_1 aktiv	$h_1(\mathbf{x}) = 0$ $h_2(\mathbf{x}) < 0$	$\mu_1 = ?$ $\mu_2 = 0$	$\begin{bmatrix} \nabla_{\mathbf{x}} L(\mathbf{x}, \mu_1) \\ h_1(\mathbf{x}) \end{bmatrix} = \mathbf{0}$	$\mathbf{x} = [\frac{18}{13}, -\frac{10}{13}]^T$ $\mu_1 = \frac{8}{13}$	nein ($\mathbf{x} \notin \mathcal{X}_{\text{ad}}$)
c) h_2 aktiv	$h_1(\mathbf{x}) < 0$ $h_2(\mathbf{x}) = 0$	$\mu_1 = 0$ $\mu_2 = ?$	$\begin{bmatrix} \nabla_{\mathbf{x}} L(\mathbf{x}, \mu_2) \\ h_2(\mathbf{x}) \end{bmatrix} = \mathbf{0}$	$\mathbf{x} = [\frac{21}{10}, \frac{11}{10}]^T$ $\mu_2 = -\frac{1}{5}$	nein ($\mathbf{x} \notin \mathcal{X}_{\text{ad}}, \mu_2 < 0$)
d) h_1, h_2 aktiv	$h_1(\mathbf{x}) = 0$ $h_2(\mathbf{x}) = 0$	$\mu_1 = ?$ $\mu_2 = ?$	$\begin{bmatrix} \nabla_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\mu}) \\ h_1(\mathbf{x}) \\ h_2(\mathbf{x}) \end{bmatrix} = \mathbf{0}$	$\mathbf{x} = [1, 0]^T$ $\boldsymbol{\mu} = [\frac{22}{27}, \frac{10}{27}]^T$	ja: $\mathbf{x}^* = \mathbf{x}$

- Beispiel Fall b):

$$\begin{cases} 2(x_1 - 1) + 2\mu_1 = 0 & (1) \\ \frac{2}{9}(x_2 - 2) + \mu_1 = 0 & (2) \\ 2x_1 + x_2 - 2 = 0 & (3) \end{cases}$$

$$\text{aus (3): } x_2 = 2 - 2x_1$$

$$\text{in (2): } \mu_1 = -\frac{2}{9}(2 - 2x_1 - 2) = \frac{4}{9}x_1$$

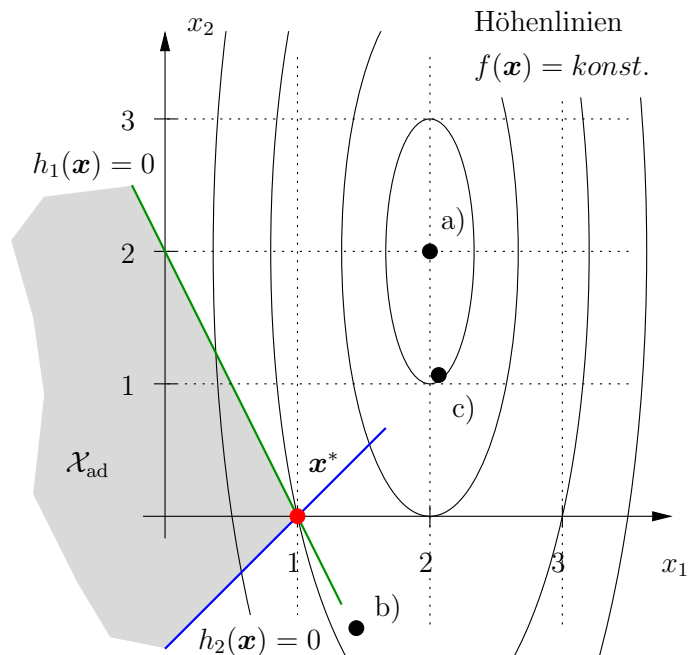
$$\text{in (1): } 2(x_1 - 2) + \frac{8}{9}x_1 = 0$$

$$\Rightarrow x_1 = \frac{18}{13}$$

$$x_2 = 2 - \frac{36}{13} = -\frac{10}{13}$$

$$\mu_1 = \frac{4}{9} \cdot \frac{18}{13} = \frac{8}{13}$$

- Veranschaulichung:



Lösung von Aufgabe 4.3 (Seite 53)

$$\bullet L(\mathbf{x}, \boldsymbol{\mu}) = (x_1 - \frac{3}{2})^2 + (x_2 - s)^4 + \underbrace{\mu_1(-1 + x_1 + x_2)}_{h_1(\mathbf{x})} + \underbrace{\mu_2(-1 + x_1 - x_2)}_{h_2(\mathbf{x})} + \underbrace{\mu_3(-1 - x_1 + x_2)}_{h_3(\mathbf{x})} + \underbrace{\mu_4(-1 - x_1 - x_2)}_{h_4(\mathbf{x})}$$

$$\bullet \nabla_{\mathbf{x}} L(\mathbf{x}^*, \boldsymbol{\mu}) = \begin{bmatrix} 2(x_1 - \frac{3}{2}) + \mu_1 + \mu_2 - \mu_3 - \mu_4 \\ 4(x_2 - s)^3 + \mu_1 - \mu_2 + \mu_3 - \mu_4 \end{bmatrix} = \mathbf{0}$$

$$\bullet \text{ Am Punkt } \mathbf{x} = [1, 0]^T \text{ gilt } h_1(\mathbf{x}^*) = 0, h_2(\mathbf{x}^*) = 0, h_3(\mathbf{x}^*) < 0, h_4(\mathbf{x}^*) < 0$$

$$\Rightarrow \mu_3 = \mu_4 = 0$$

$$\bullet 2(x_1 - \frac{3}{2}) + \mu_1 + \mu_2 = 0 \text{ und } 4(x_2 - s)^3 + \mu_1 - \mu_2 = 0 \text{ liefern}$$

$$\mu_1 = \frac{1 + 4s^3}{2}, \quad \mu_2 = \frac{1 - 4s^3}{2}$$

$$\bullet \text{ Die KKT-Vorzeichenbedingung } \mu_1, \mu_2 \geq 0, \text{ siehe (4.29d), ist also für}$$

$$-\sqrt[3]{1/4} \leq s \leq \sqrt[3]{1/4}$$

erfüllt.

Lösung von Aufgabe 4.4 (Seite 54)

$$\bullet \text{ Satz 4.3: Hessematrix } \nabla_{\mathbf{x}\mathbf{x}}^2 L(\mathbf{x}^*, \boldsymbol{\mu}^*) = \begin{bmatrix} 2 & 0 \\ 0 & \frac{2}{9} \end{bmatrix} \text{ ist positiv definit}$$

$$\Rightarrow \mathbf{s}^T \nabla_{\mathbf{x}\mathbf{x}}^2 L(\mathbf{x}^*, \boldsymbol{\mu}^*) \mathbf{s} > 0 \quad \forall \mathbf{s} \neq \mathbf{0} \text{ erfüllt}$$

$$\bullet \text{ Satz 4.4: Beschränkungsfktn. } h_1(\mathbf{x}) \text{ und } h_2(\mathbf{x}) \text{ sind linear, also konvex}$$

$$\Rightarrow \mathbf{x}^* \text{ ist ein globales Minimum}$$

Lösung von Aufgabe 4.5 (Seite 56)

$$\bullet \mu_1^* = 22/27 \approx 0.815 \text{ und } \mu_2^* = 10/27 \approx 0.370 \text{ für } \mathbf{x}^* = [1, 0]^T$$

$$\Rightarrow \text{Änderung von } h_1(\mathbf{x}) \leq \varepsilon_1 \text{ hat stärkeren Einfluss auf } f \text{ als Änderung in } h_2(\mathbf{x}) \leq \varepsilon_2 \text{ (siehe Abb. 4.6).}$$

Lösung von Aufgabe 4.6 (Seite 57)

- Transformation von (4.49c) in Standardform (4.44):

$$-x_1 + x_2 + x_3 + z = 1 \quad \text{mit} \quad z \geq 0$$

- x_1 nicht in (4.49d) enthalten (x_1 kann < 0 sein) $\Rightarrow$ nach (4.47)

$$x_1 = x_1^+ - x_1^- \quad \text{mit} \quad x_1^-, x_1^+ \geq 0$$

- Darstellung in Standardform mit $\bar{x} = [x_1^+, x_2, x_3, x_1^-, z]^T \in \mathbb{R}^5$

$$\begin{aligned} \min_{\bar{x} \in \mathbb{R}^5} \quad & -x_1^+ + x_1^- + 2x_2 + x_3 \\ \text{u.B.v.} \quad & x_1^+ - x_1^- + x_3 = 1 \\ & -x_1^+ + x_1^- + x_2 + x_3 + z = 1 \\ & \bar{x}_i \geq 0, \quad i = 1, \dots, 5 \end{aligned}$$

Lösung von Aufgabe 4.7 (Seite 67)

- Lineares Optimierungsproblem

$$\begin{aligned} \min_{x \in \mathbb{R}^4} \quad & \sum_{i=1}^4 K_i x_i \\ \text{u.B.v.} \quad & V_1 = x_1 + x_3 \quad (\text{Verbrauchszentrum } Z_1) \\ & V_2 = x_2 - x_3 + x_4 \quad (\text{Verbrauchszentrum } Z_2) \\ & P_1 \geq x_1 + x_2 \quad (\text{Prod.kap. von } R_1) \\ & 0 \leq x_i \leq C_i, \quad i = 1, \dots, 4 \quad (\text{Kap. der Leitung } l_i) \end{aligned}$$

- Elimination von x_3 und x_4 durch Gleichungsbeschr.

$$x_3 = V_1 - x_1, \quad x_4 = V_2 - x_2 + x_3 = V_1 + V_2 - x_1 - x_2$$

- Ungleichungsbeschr. für x_3, x_4 :

$$\begin{aligned} 0 \leq x_3 = V_1 - x_1 \leq C_3 : \quad & \boxed{x_1 \geq V_1 - C_3} \\ & x_1 \leq V_1 \quad \text{mit } x_1 \leq C_1 < V_1 \text{ bereits erfüllt} \\ 0 \leq x_4 = V_1 + V_2 - x_1 - x_2 \leq C_4 : \quad & \boxed{x_1 + x_2 \geq V_1 + V_2 - C_4} \\ & x_1 + x_2 \leq V_1 + V_2 \text{ mit } x_{1,2} \leq C_{1,2} < V_{1,2} \text{ erfüllt} \end{aligned}$$

$\Rightarrow$ reduziertes Problem

$$\begin{aligned} \min_{x \in \mathbb{R}^2} \quad & (K_1 - K_3 - K_4)x_1 + (K_2 - K_4)x_2 + \underbrace{K_3 V_1 + K_4 (V_1 + V_2)}_{\text{vernachlässigbar}} \\ \text{u.B.v.} \quad & x_1 + x_2 \leq P_1 \\ & x_1 \geq V_1 - C_3 \quad (\text{aus Gl.beschr.}) \\ & x_1 + x_2 \geq V_1 + V_2 - C_4 \quad (\text{aus Gl.beschr.}) \\ & 0 \leq x_i \leq C_i, \quad i = 1, 2 \end{aligned}$$

Lösung von Aufgabe 4.8 (Seite 67)

- Mit gegebenen Parametern

$$\min_{\mathbf{x} \in \mathbb{R}^2} \quad 20x_1 + 10x_2 \quad (\text{A.40a})$$

$$\text{u.B.v.} \quad x_1 + x_2 \leq 45 \quad (\text{A.40b})$$

$$x_1 \geq 20, \quad x_1 + x_2 \geq 30 \quad (\text{A.40c})$$

$$0 \leq x_1 \leq 30, \quad 0 \leq x_2 \leq 20 \quad (\text{A.40d})$$

- Zulässige Menge $\mathcal{X}_{\text{ad}}$, siehe Abb. A.13

- Höhenlinien von Kostenfkt.: $20x_1 + 10x_2 = m = \text{konst.} \Rightarrow \boxed{x_2 = m - 2x_1}$

$\Rightarrow$ Optimaler Punkt $\mathbf{x}^* = [20, 10]^\top$

Überprüfung der Vorzeichenbedingung (4.29d) für μ_i bei $\mathbf{x}^*$:

- Kostenfkt. und Ungl.beschr. (lineares OP nicht in Standardform)

$$f(\mathbf{x}) = 20x_1 + 10x_2, \quad h_1(\mathbf{x}) = 30 - x_1 - x_2 \leq 0, \quad h_2(\mathbf{x}) = 20 - x_1 \leq 0 \quad (\text{A.41})$$

- Gradienten (siehe Abb. A.13)

$$\nabla f(\mathbf{x}) = \begin{bmatrix} 20 \\ 10 \end{bmatrix}, \quad \nabla h_1(\mathbf{x}) = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \quad \nabla h_2(\mathbf{x}) = \begin{bmatrix} -1 \\ 0 \end{bmatrix} \quad (\text{A.42})$$

$\Rightarrow -\nabla f(\mathbf{x}^*)$ ist positive Linearkombination von $\nabla h_1(\mathbf{x}^*)$ und $\nabla h_2(\mathbf{x}^*)$, d.h.

$$\nabla_{\mathbf{x}} L(\mathbf{x}^*, \boldsymbol{\mu}^*) = \nabla f(\mathbf{x}^*) + \mu_1^* \nabla h_1(\mathbf{x}^*) + \mu_2^* \nabla h_2(\mathbf{x}^*) = \mathbf{0}, \quad \boldsymbol{\mu}^* = [\mu_1^*, \mu_2^*]^\top > \mathbf{0}$$

$\Rightarrow$ KKT-Bedingungen sind erfüllt

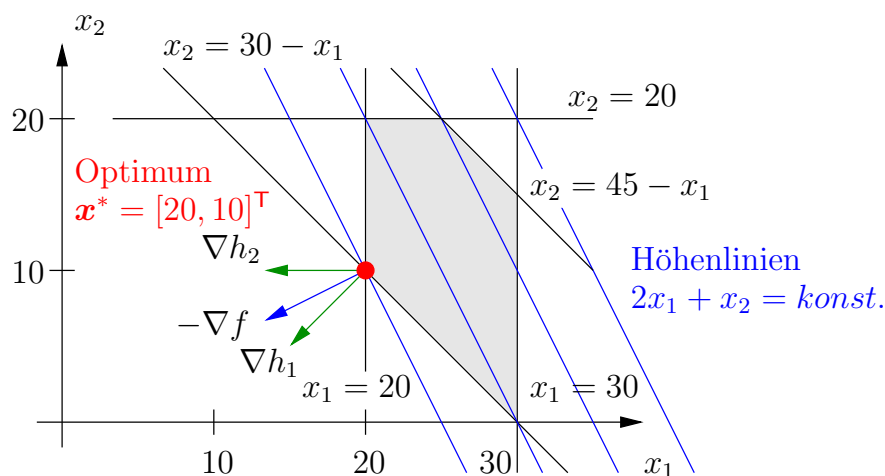


Abbildung A.13: Grafische Lösung von Aufgabe 4.7 (Gradienten nicht maßstabsgetreu!).

Lösung von Aufgabe 4.9 (Seite 67)

Alternative zu Matlab-Code in Abbildung 4.11: Implementierung der Beschränkungen unter linprog über $A_{eq}x = b_{eq}$ und $Ax \leq b$ (ohne $LB \leq x \leq UB$):

$$A_{eq} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & -1 & 1 \end{bmatrix}, \quad b_{eq} = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}$$

$$A = \begin{bmatrix} -I_4 \\ +I_4 \\ 1 & 1 & 0 & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ C \\ P_1 \end{bmatrix}, \quad I_4 = \text{diag}[1, 1, 1, 1]^T, \quad C = [C_1, C_2, C_3, C_4]^T$$

```
>> transportnetz_alternative
```

Simplex-Verfahren

Phase 1: Compute initial basic feasible point.

Iter	Infeasibility
0	90
1	60
2	45
3	20
4	-0

Phase 2: Minimize using simplex.

Iter	Objective f'*x	Dual Infeasibility A'*y+z-w-f
0	2150	14.1421
1	2100	20
2	2000	10
3	1900	0

Optimization terminated.

Berechnete Lösung: [20.00, 10.00, 30.00, 60.00] mit f(x) = 1900.00

Abstand zum optimalen Punkt: dx = [0, 0, 0, 0]

Interior-Point-Verfahren

Residuals:	Primal Infeas A*x-b	Dual Infeas A'*y+z-f	Duality Gap x'*z	Total Rel Error
Iter 0:	2.55e+002	1.48e+002	5.71e+004	2.33e+000
Iter 1:	7.10e+000	2.47e+000	3.13e+003	1.10e+000
Iter 2:	4.68e-010	4.18e-002	1.39e+002	6.74e-002
Iter 3:	5.96e-010	1.03e-003	3.53e+000	1.72e-003
Iter 4:	4.82e-009	6.11e-008	2.06e-004	1.00e-007
Iter 5:	6.09e-013	6.32e-014	2.06e-010	1.07e-013

Optimization terminated.

Berechnete Lösung: [20.00000, 10.00000, 30.00, 60.00] mit f(x) = 1900.00

Abstand zum optimalen Punkt: dx = [2.27e-013, 1.61e-012, -1.56e-013, -1.31e-012]

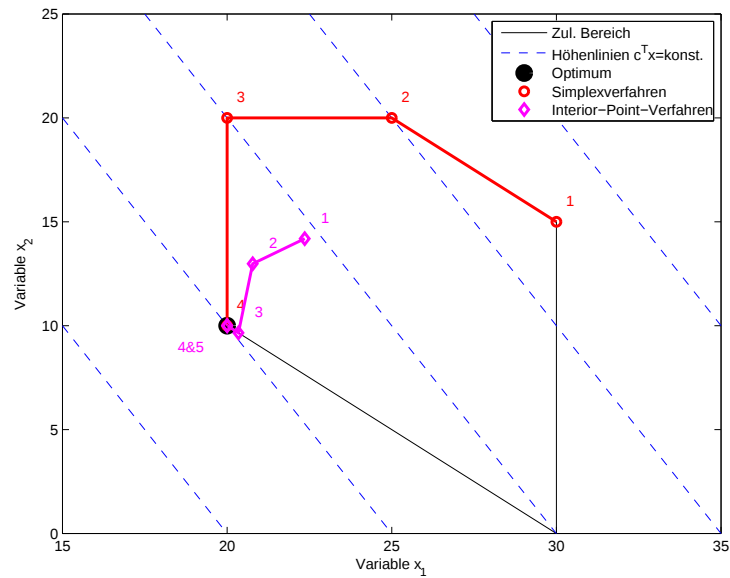


Abbildung A.14: Vergleich von Simplex- und Innere-Punkt-Verfahren in Aufgabe 4.9.

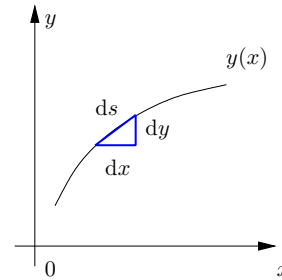
Der Vergleich und Abbildung A.14 zeigen, dass das Simplex-Verfahren entlang der Ecken des zulässigen Polyeders wandert und damit die optimale Lösung $\mathbf{x}^*$ *exakt* trifft. Statt dessen nähert das Innere-Punkt-Verfahren die Lösung von innen her an und erreicht $\mathbf{x}^*$ nur für $\tau \rightarrow 0$ exakt. Da in der Praxis mit Hilfe eines Toleranzkriteriums abgebrochen wird, ergibt sich also eine *approximative* Lösung.

A.4 Aufgaben in Kapitel 5

Lösung von Aufgabe 5.1 (Seite 87)

- Veranschaulichung von (5.21):

$$\begin{aligned} ds &= \sqrt{dx^2 + dy^2} \quad \text{mit} \quad y'(x) = \frac{dy}{dx} \\ &= \sqrt{1 + y'(x)^2} dx \end{aligned}$$



- Ziel: $\min \int_0^{s_B} ds$ (siehe Abb. A.15). Mit (5.21) folgt

$$J(y) = \int_{x_A}^{x_B} \underbrace{\sqrt{1 + y'(x)^2}}_{= \phi(y')} dx, \quad y(x_A) = y_A \quad y(x_B) = y_B \quad (\text{A.43})$$

- $\phi(y')$ ist unabhängig von $y \Rightarrow$ EL-Gl. (5.16) vereinfacht sich zu (5.19)

$$\frac{\partial \phi}{\partial y'} = \frac{y'(x)}{\sqrt{1 + y'(x)^2}} = \text{konst.} \quad (\text{A.44})$$

- Somit ist $y'(x) = \text{konst.} \Rightarrow$ Integration liefert $y(x) = c_1 x + c_2$
- Bestimmung von c_1 und c_2 aus RBen $y(x_A) = y_A$, $y(x_B) = y_B$:

$$y(x) = \frac{y_B - y_A}{x_B - x_A} x + \frac{y_A x_B - y_B x_A}{x_B - x_A} \quad (\text{A.45})$$

$\Rightarrow$ minimaler Abstand zwischen A und B ist eine Gerade (wie zu erwarten war).

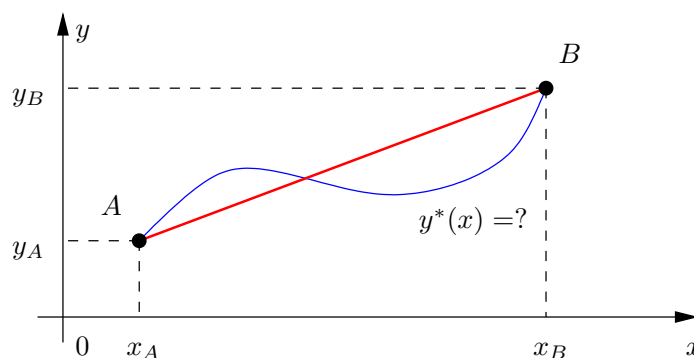


Abbildung A.15: Minimaler Abstand zwischen zwei Punkten A und B .

Lösung von Aufgabe 5.2 (Seite 93)

- Hamilton-Funktion $H(x, \lambda, u) = \frac{1}{2}u^2 + \frac{a}{2}x^2 + \lambda u$
- Optimalitätsbedingungen (5.38), (5.39)

$$\begin{aligned}\dot{\lambda} &= -\frac{\partial H}{\partial x} = -ax, & \lambda(1) & \text{frei} \\ 0 &= \frac{\partial H}{\partial u} = u + \lambda & \Rightarrow & u = -\lambda\end{aligned}$$

- Differenzieren von $\dot{x} = u = -\lambda$ führt zu

$$\ddot{x} = -\dot{\lambda} = ax \quad \Rightarrow \quad x(t) = c_1 e^{\sqrt{a}t} + c_2 e^{-\sqrt{a}t}.$$

- Mit RBen $x(0) = 1$ und $x(1) = 0$: $c_1 = \frac{1}{1 - e^{2\sqrt{a}}}$, $c_2 = \frac{-e^{2\sqrt{a}}}{1 - e^{2\sqrt{a}}}$

$\Rightarrow$ optimale Lösung:

$$x^*(t) = \frac{1}{1 - e^{2\sqrt{a}}} \left(e^{\sqrt{a}t} - e^{\sqrt{a}(2-t)} \right), \quad u^*(t) = -\lambda^*(t) = \frac{\sqrt{a}}{1 - e^{2\sqrt{a}}} \left(e^{\sqrt{a}t} + e^{\sqrt{a}(2-t)} \right)$$

- Abbildung 5.5: größerer Wert von a entspricht stärkerer Gewichtung von x^2 in der Kostenfunktion $\Rightarrow$ schnelleres Abklingen von $x^*(t)$

Lösung von Aufgabe 5.3 (Seite 94)

- Hamilton-Funktion $H(x, \lambda, u) = \frac{1}{2}u^2 + \lambda_1 x_2 + \lambda_2 u$
- Optimalitätsbedingungen (5.38), (5.39)

$$\begin{aligned}\dot{\lambda}_1 &= -\frac{\partial H}{\partial x_1} = 0, & \lambda_1(1) & \text{frei} \\ \dot{\lambda}_2 &= -\frac{\partial H}{\partial x_2} = -\lambda_1, & \lambda_2(1) &= ax_2(1) \\ 0 &= \frac{\partial H}{\partial u} = u + \lambda_2 & \Rightarrow & u = -\lambda_2\end{aligned}$$

- $\ddot{\lambda}_2(t) = 0$: Integration liefert $\lambda_2(t) = c_1 t + c_2$
- $\ddot{x}_1(t) = u(t) = -\lambda_2(t) \Rightarrow x_1(t) = -\frac{c_1}{6}t^3 - \frac{c_2}{2}t^2 - c_3 t - c_4$ (1)
- Bestimmung von c_i aus RBen:

$$\left. \begin{aligned}x_1(0) &= 1 : & c_4 &= -1 \\ x_2(0) &= 0 : & c_3 &= 0 \\ x_1(1) &= 0 : & c_1 + 3c_2 &= 6 \\ \lambda_2(1) &= ax_2(1) : & c_1 + c_2 &= ax_2(1)\end{aligned} \right\} \begin{aligned}c_1 &= -3 + \frac{3}{2}a x_2(1) \\ c_2 &= 3 - \frac{1}{2}a x_2(1)\end{aligned} \quad (2)$$

- Berechnung von Endzustand $x_2(1)$ mit $x_2(t) = -\frac{c_1}{2}t^2 - c_2 t$:

$$x_2(1) = -\frac{3}{4}a x_2(1) + \frac{3}{2} - 3 + \frac{1}{2}a x_2(1) \quad \Rightarrow \quad x_2(1) = -\frac{6}{4+a} \quad (3)$$

- (1)–(3) $\Rightarrow$ optimale Lösung

$$x_1^*(t) = \frac{2(1+a)}{4+a} t^3 - \frac{3(2+a)}{a+4} t^2 + 1, \quad u^*(t) = \frac{12(1+a)}{4+a} t - \frac{6(2+a)}{a+4}$$

- Abbildung 5.6: Für größere Werte von a überwiegt der Endterm $\frac{a}{2}x_2(1)^2$ in der Kostenfunktion $\Rightarrow$ glatterer Übergang ($x_2(1) \rightarrow 0$), was mehr Stellenergie $\int_0^1 \frac{1}{2}u^2 dt$ erfordert.

Lösung von Aufgabe 5.4 (Seite 117)

- Integration von $\dot{x}_1 = x_2$ und $\dot{x}_2 = u = \text{konst.}$:

$$\begin{aligned} x_2(t) &= x_{2,0} + ut \\ x_1(t) &= x_{1,0} + x_{2,0}t + \frac{1}{2}ut^2 \quad (x_{1,0}, x_{2,0} \text{ beliebig}) \end{aligned}$$

- Elimination der Zeit, $t = \frac{x_2 - x_{2,0}}{u}$ ($u = \pm 1 \neq 0$), liefert

$$\begin{aligned} x_1 &= x_{1,0} + x_{2,0} \frac{x_2 - x_{2,0}}{u} + \frac{1}{2}u \left(\frac{x_2 - x_{2,0}}{u} \right)^2 \\ &= x_{1,0} + \cancel{\frac{x_{2,0}x_2}{u}} - \frac{x_{2,0}^2}{u} + \frac{x_2^2 - \cancel{2x_{2,0}x_2} + x_{2,0}^2}{2u} \\ &= \frac{1}{2u}x_2^2 + \underbrace{x_{1,0} - \frac{x_{2,0}^2}{2u}}_{=c} \end{aligned}$$

Lösung von Aufgabe 5.5 (Seite 118)

- Integration von (5.103b) mit $u = \text{konst.}$

$$\begin{aligned} x_2(t) &= x_{2,0} + ut \\ x_1(t) &= x_{1,0} + x_{2,0}t + \frac{1}{2}ut^2 \end{aligned}$$

- Betrachtung von Fall $x_{1,0} > S(x_{2,0}) \Rightarrow u = -1$ (siehe Abb. 5.18)
- Berechnung von Schaltzeitpunkt t_s aus

$$\begin{aligned} x_1(t_s) &\stackrel{!}{=} S(x_2(t_s)) = \frac{1}{2}x_2(t_s)^2 \quad \text{mit } x_2(t_s) < 0 \\ x_{1,0} + x_{2,0}t_s - \frac{1}{2}t_s^2 &= \frac{1}{2}(x_{2,0} - t_s)^2 = \frac{1}{2}(x_{2,0}^2 - 2x_{2,0}t_s + t_s^2) \\ \Rightarrow 0 &= t_s^2 - 2x_{2,0}t_s + \frac{1}{2}x_{2,0}^2 - x_{1,0} \\ \Rightarrow t_s &= x_{2,0} (\pm) \sqrt{\frac{1}{2}x_{2,0}^2 + x_{1,0}} \quad (\text{"-": erster Schnittpkt.}) \\ \text{zusätzlich: } x_2(t_s) &= x_{2,0} - t_s = -\sqrt{\frac{1}{2}x_{2,0}^2 + x_{1,0}} \end{aligned}$$

- Restzeit t_{rest} auf Schaltkurve mit $u = 1$ (siehe Abb. 5.18)

$$0 \stackrel{!}{=} x_2(t_s) + t_{\text{rest}} \quad \Rightarrow \quad t_{\text{rest}} = -x_2(t_s) = \sqrt{\frac{1}{2}x_{2,0}^2 + x_{1,0}}$$

- Gesamtzeit $t_f = t_s + t_{\text{rest}} = x_{2,0} + \sqrt{2x_{2,0}^2 + 4x_{1,0}}$
- Fall $x_{1,0} < S(x_{2,0})$ mit $u = 1$ analog ...

Lösung von Aufgabe 5.6 (Seite 119)

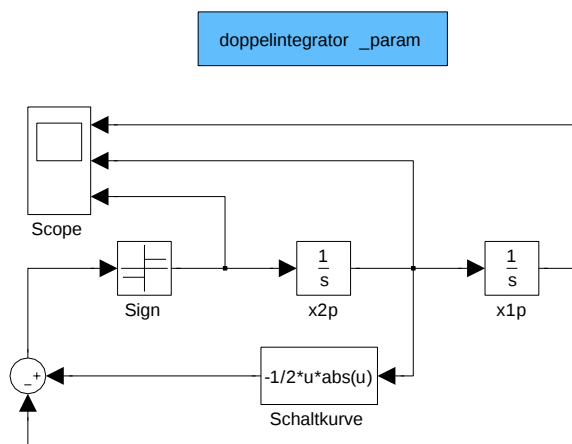
- doppelintegrator_param.m:

```
clear all

x10 = 1;
x20 = 3;

if x10 > -1/2*x20*abs(x20),
    Tsim = x20 + sqrt(4*x10+2*x20^2) - 1e-3;
else
    Tsim = -x20 + sqrt(-4*x10+2*x20^2) - 1e-3;
end
```

- SIMULINK-Schaltplan



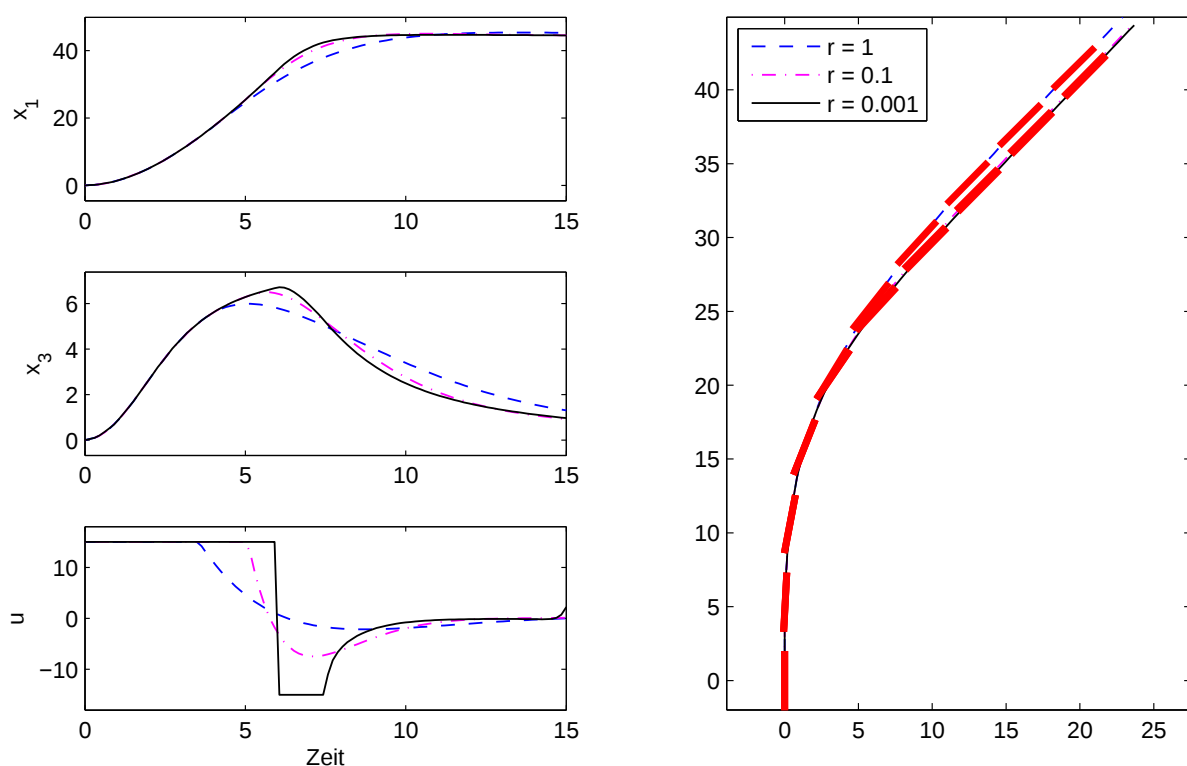
A.5 Aufgaben in Kapitel 6

Lösung von Aufgabe 6.1 (Seite 133)

Aufruf (Einstellung der Gewichtung r über `schiff_param.m`):

```
[vec,sol,p] = schiff_rwa(-15,15)
schiff_plot(vec,p,'b');
```

Die folgende Abbildung zeigt den Einfluß der Gewichtung r auf die Aggressivität der Stellgröße u für den Richtungswechsel des Schiffes.



Lösung von Aufgabe 6.2 (Seite 134)

- Einfluss der Integrationsgenauigkeit

```
optode = odeset('RelTol',1e-9,'AbsTol',1e-9);
-----
>> [adj0opt,vec,p] = schiff_schiess(-5,5,[0;0;0]);
```

Iteration	Func-count	f(x)	Norm of step	First-order optimality	Trust-region radius
0	4	1.18177e+007		9.64e+007	1
1	5	1.18177e+007	1	9.64e+007	1
2	6	1.18177e+007	0.25	9.64e+007	0.25
3	10	677265	0.0625	5.37e+007	0.0625
4	14	151141	0.15625	8.38e+006	0.156
5	15	151141	0.390625	8.38e+006	0.391
6	16	151141	0.0976562	8.38e+006	0.0977

```

...
15      43      111929      0.00238419      1.98e+005      0.00238
16      47      111323      0.00596046      4.88e+004      0.00596
17      51      109799      0.0149012      8.49e+004      0.0149
18      52      109799      0.0372529      8.49e+004      0.0373
19      53      109799      0.00931323      8.49e+004      0.00931
...
57      145      106719      8.07794e-011      6.51e+005      8.08e-011
58      146      106719      2.01948e-011      6.51e+005      2.02e-011
59      147      106719      5.04871e-012      6.51e+005      5.05e-012
60      148      106719      1.26218e-012      6.51e+005      1.26e-012
Optimizer appears to be converging to a point which is not a root.
Norm of relative change in X is less than max(options.TolX^2,eps) but
sum-of-squares of function values is greater than or equal to sqrt(options.TolFun)
Try again with a new starting guess.

```

Die hohe Genauigkeit von $\text{RelTol}=1\text{e-}13$ und $\text{AbsTol}=1\text{e-}14$ ist für die Schranken $u \in [-15^\circ, 15^\circ]$ notwendig, da in diesem Fall die Gradienten in den kanonischen Gleichungen größere Werte annehmen.

- Iterative Lösung mittels Fortsetzungsverfahren

```

>> adj0 = [0;0;0];
>> adj0 = schiff_schiess(-5,5,adj0)

```

Iteration	Func-count	f(x)	Norm of step	First-order optimality	Trust-region radius
0	4	1.18177e+007		8.68e+007	1
1	8	5.84694e+006	1	1.02e+008	1
2	9	5.84694e+006	1	1.02e+008	1
3	13	2.21786e+006	0.25	4.2e+007	0.25
4	17	116101	0.25	1.99e+006	0.25
...					
23	84	0.00250478	0.136622	3.35e+003	14.9
24	88	2.03942e-006	0.0152405	85.3	14.9
25	92	1.59182e-010	0.0019405	0.607	14.9
26	96	5.71313e-015	1.77205e-005	0.00363	14.9
27	100	2.03563e-019	1.05904e-007	2.17e-005	14.9

Optimization terminated: relative function value changing by less than $\max(\text{options.TolFun}^2, \text{eps})$ and sum-of-squares of function values is less than $\sqrt{\text{options.TolFun}}$.

```
adj0 =
```

```

-7.1995
-17.6124
0.0764

```

```
>> adj0 = schiff_schiess(-10,10,adj0)
```

Iteration	Func-count	f(x)	Norm of step	First-order optimality	Trust-region radius
0	4	2506.52		9.52e+006	1
1	8	454.706	1	5.59e+005	1
2	12	187.741	2.5	2.1e+005	2.5
3	13	187.741	5.01414	2.1e+005	6.25
4	17	105.833	1.25353	5.23e+004	1.25
5	21	16.0579	3.13384	1.52e+006	3.13
...					
20	63	0.00012013	0.0139365	3.38e+003	0.0529
21	67	5.27707e-008	0.00094326	37.3	0.0529
22	71	1.25447e-011	2.48772e-005	0.848	0.0529
23	75	2.7598e-015	3.51176e-007	0.0128	0.0529
24	79	6.07584e-019	5.18203e-009	0.000191	0.0529

Optimization terminated: relative function value changing by less than $\max(\text{options.TolFun}^2, \text{eps})$ and sum-of-squares of function values is less than $\sqrt{\text{options.TolFun}}$.

```
adj0 =
```

```
-4.6095
-8.8737
0.0764
```

```
>> adj0 = schiff_schiess(-12,12,adj0)
```

Iteration	Func-count	f(x)	Norm of step	First-order optimality	Trust-region radius
0	4	89.3997		1.11e+006	1
1	5	89.3997	1	1.11e+006	1
2	9	54.9325	0.25	2.77e+005	0.25
3	13	38.1542	0.25	2.41e+005	0.25
...					
15	55	0.011242	0.00782959	4.64e+003	0.00783
16	59	0.000583449	0.00986856	7.62e+003	0.0196
17	63	1.31202e-006	0.00124875	19.8	0.0247
18	67	4.53489e-010	7.42867e-005	3.34	0.0247
19	71	1.30166e-013	1.23264e-006	0.0656	0.0247
20	75	3.67626e-017	2.03688e-008	0.00111	0.0247

Optimization terminated: relative function value changing by less than max(options.TolFun^2,eps) and sum-of-squares of function values is less than sqrt(options.TolFun).

```
adj0 =
```

```
-4.1358
-7.4116
0.0764
```

```
>> adj0 = schiff_schiess(-15,15,adj0)
```

Iteration	Func-count	f(x)	Norm of step	First-order optimality	Trust-region radius
0	4	3918.42		1.13e+008	1
1	5	3918.42	1	1.13e+008	1
2	9	139.133	0.25	7.82e+006	0.25
3	10	139.133	0.625	7.82e+006	0.625
4	14	107.823	0.15625	1.19e+006	0.156
5	15	107.823	0.390625	1.19e+006	0.391
...					
21	73	1.25593	0.010625	1.02e+006	0.0434
22	77	0.174229	0.00265626	7.76e+004	0.00266
23	81	0.0119964	0.00664064	4.32e+004	0.00664
24	85	1.61201e-005	0.00150479	1.39e+003	0.0166
25	89	5.87806e-009	4.42919e-005	6.96	0.0166
26	93	1.67019e-012	2.11456e-006	0.0447	0.0166
27	97	5.58091e-016	3.92032e-008	0.000777	0.0166

Optimization terminated: relative function value changing by less than max(options.TolFun^2,eps) and sum-of-squares of function values is less than sqrt(options.TolFun).

```
adj0 =
```

```
-3.6410
-5.9534
0.0764
```