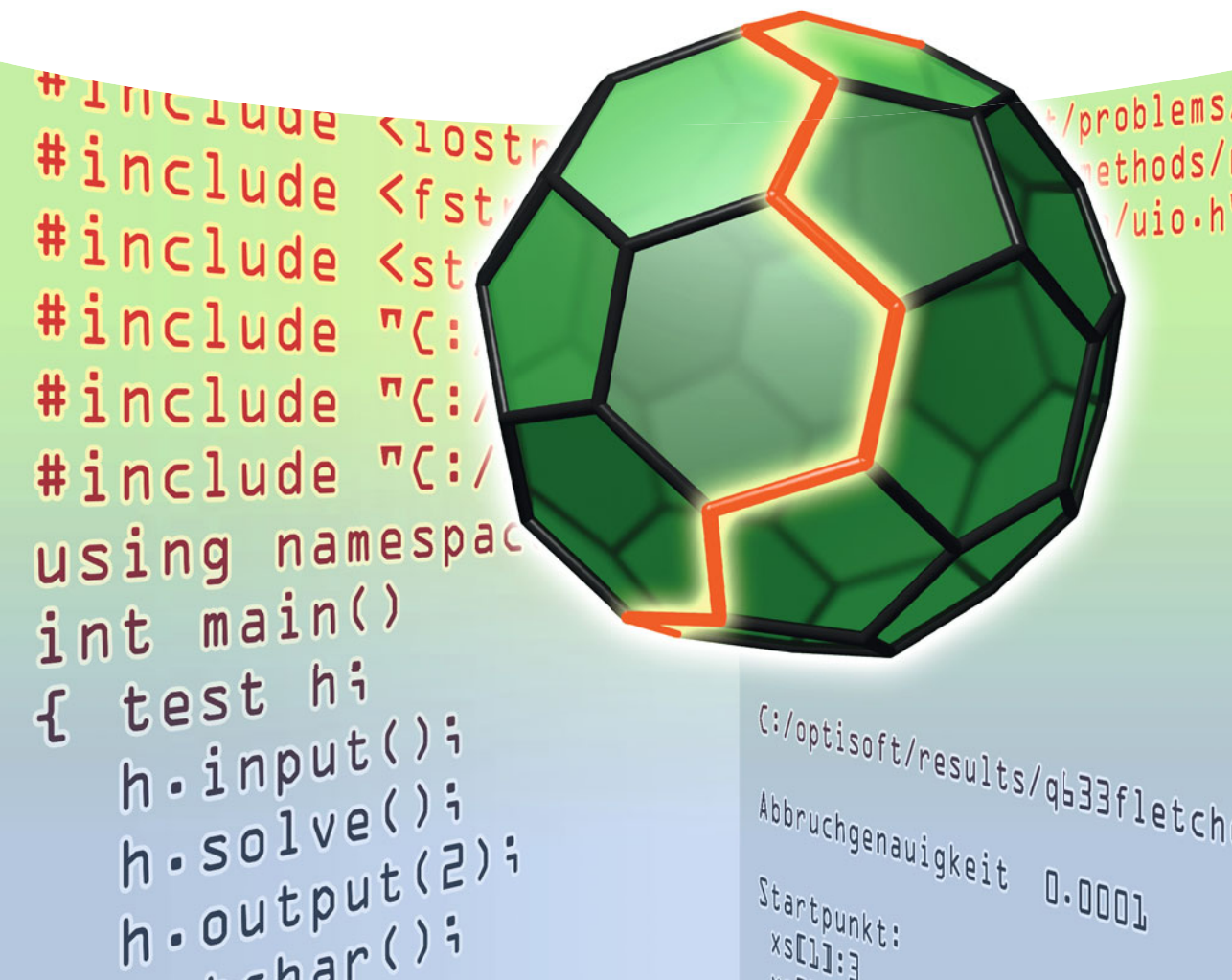


Claus Richter

Optimierung in C++

Grundlagen und Algorithmen



Claus Richter

Optimierung in C++

Claus Richter

Optimierung in C++

Grundlagen und Algorithmen

WILEY-VCH
Verlag GmbH & Co. KGaA

Autor**Claus Richter**

Gerhart-Hauptmann-Str. 1
01219 Dresden
Deutschland

■ Alle Bücher von Wiley-VCH werden sorgfältig erarbeitet. Dennoch übernehmen Autoren, Herausgeber und Verlag in keinem Fall, einschließlich des vorliegenden Werkes, für die Richtigkeit von Angaben, Hinweisen und Ratschlägen sowie für eventuelle Druckfehler irgendeine Haftung.

**Bibliografische Information der
Deutschen Nationalbibliothek**

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

© 2017 WILEY-VCH Verlag GmbH & Co. KGaA,
Boschstr. 12, 69469 Weinheim, Germany

Alle Rechte, insbesondere die der Übersetzung in andere Sprachen, vorbehalten. Kein Teil dieses Buches darf ohne schriftliche Genehmigung des Verlages in irgendeiner Form – durch Photokopie, Mikroverfilmung oder irgendein anderes Verfahren – reproduziert oder in eine von Maschinen, insbesondere von Datenverarbeitungsmaschinen, verwendbare Sprache übertragen oder übersetzt werden. Die Wiedergabe von Warenbezeichnungen, Handelsnamen oder sonstigen Kennzeichen in diesem Buch berechtigt nicht zu der Annahme, dass diese von jedermann frei benutzt werden dürfen. Vielmehr kann es sich auch dann um eingetragene Warenzeichen oder sonstige gesetzlich geschützte Kennzeichen handeln, wenn sie nicht eigens als solche markiert sind.

Umschlaggestaltung Formgeber, Mannheim, Deutschland

Satz le-tex publishing services GmbH, Leipzig, Deutschland

Print ISBN 978-3-527-34107-8

ePDF ISBN 978-3-527-80079-7

ePub ISBN 978-3-527-80080-3

Mobi ISBN 978-3-527-80081-0

Gedruckt auf säurefreiem Papier.

Für meine liebe Frau Hannelore

Inhaltsverzeichnis

Vorwort *XIII*

1	Einleitung	1
1.1	Das lineare und das nichtlineare Optimierungsproblem	1
1.2	Definitionen und Bezeichnungen	1
1.3	Spezialfälle linearer und nichtlinearer Optimierungsaufgaben	2
1.4	Anwendungen	4
1.4.1	Strukturoptimierung	4
1.4.2	Das Least-Squares-Problem	5
1.4.3	Optimale Steuerung	6
2	Grundlagen	9
2.1	Regularitätsbedingungen	9
2.1.1	Slater-Bedingung	9
2.1.2	Abadie-Bedingung	9
2.1.3	Bedingung der linearen Unabhängigkeit – LICQ	10
2.1.4	Constraint Qualification	10
2.1.5	Bemerkungen	10
2.2	Optimalitätsbedingungen	10
2.2.1	Optimalitätskriterium mittels zulässiger Richtungen	11
2.2.2	Karush-Kuhn-Tucker-Bedingung	11
2.2.3	Bezeichnungen	11
2.2.4	Notwendige Bedingungen 2. Ordnung	12
2.2.5	Hinreichende Bedingungen 2. Ordnung	12
2.2.6	Strenge hinreichende Bedingungen 2. Ordnung	13
2.3	Optimalitätskriterien für spezielle Optimierungsaufgaben	13
2.4	Wünschenswerte Eigenschaften von Optimierungsverfahren	14
2.4.1	Theoretische Richtung	15
2.4.2	Empirische Richtung	17
2.5	Vom C++-Programm zum nutzerfreundlichen Softwaresystem	18

3	Mathematische Hilfsmittel	21
3.1	Das Austauschverfahren	22
3.2	Lösung von Gleichungssystemen mit der QR-Zerlegung	26
3.2.1	Aufbau des Algorithmus	28
3.3	Cholesky-Zerlegung	29
3.3.1	Grundlagen des Verfahrens	29
3.3.2	Aufbau des Algorithmus	30
3.3.3	Weiterführende Bemerkungen	31
3.4	Fibonacci-Verfahren	31
3.4.1	Grundlagen des Verfahrens	31
3.4.2	Aufbau des Algorithmus	33
3.5	Das Verfahren des Goldenen Schnitts	34
3.5.1	Grundlagen des Verfahrens	34
3.5.2	Aufbau des Algorithmus	35
3.6	Newton-Verfahren	36
3.6.1	Grundlagen des Verfahrens	36
3.6.2	Aufbau des Algorithmus	36
3.7	Runge-Kutta-Verfahren zur Lösung von Differenzialgleichungen	37
3.7.1	Weiterführende Bemerkungen	40
4	Probleme und Algorithmen als C++-Klassen	43
4.1	Die Programmiersprache C++	43
4.2	Der Weg zur objektorientierten Programmierung	44
4.3	Begriffe der objektorientierten Programmierung	45
4.4	Lösungsverfahren und Probleme als Klassen	46
5	Lineare Optimierung	55
5.1	Das Simplexverfahren	56
5.1.1	Grundlagen des Verfahrens	56
5.1.2	Aufbau des Algorithmus	59
5.1.3	Konstruktion eines ersten Simplextableaus	61
5.2	Das revidierte Simplexverfahren	63
5.2.1	Grundlagen des Verfahrens	63
5.2.2	Aufbau des Algorithmus	66
5.3	Weiterführende Bemerkungen	67
5.4	Das Ellipsoidverfahren	67
5.4.1	Grundlagen des Verfahrens	67
5.4.2	Aufbau des Algorithmus	70
5.5	Weiterführende Bemerkungen	71
6	Quadratische Optimierung	73
6.1	Das Relaxationsverfahren	74
6.1.1	Grundlagen des Verfahrens	74
6.1.2	Aufbau des Algorithmus	74
6.1.3	Weiterführende Bemerkungen	76

6.2	Methode der aktiven Restriktionen von Fletcher	76
6.2.1	Grundlagen des Verfahrens	76
6.2.2	Der Algorithmus	77
6.2.3	Weiterführende Bemerkungen	78
7	Unbeschränkte nichtlineare Optimierung	79
7.1	Das Verfahren der stochastischen Suche	80
7.1.1	Grundlagen des Verfahrens	80
7.1.2	Aufbau des Algorithmus	80
7.1.3	Weiterführende Bemerkungen	81
7.2	Das Verfahren der koordinatenweisen Suche	82
7.2.1	Grundlagen des Verfahrens	82
7.2.2	Aufbau des Algorithmus	82
7.3	Das einfache Polytopverfahren	83
7.3.1	Grundlagen des Verfahrens	83
7.3.2	Aufbau des Algorithmus	85
7.3.3	Weiterführende Bemerkungen	86
7.4	Das Verfahren des steilsten Abstiegs	87
7.4.1	Grundlagen des Verfahrens	87
7.4.2	Aufbau des Algorithmus	88
7.4.3	Weiterführende Bemerkungen	89
7.5	Das Verfahren der konjugierten Gradienten	89
7.5.1	Grundlagen des Verfahrens	89
7.5.2	Aufbau des Algorithmus	91
7.5.3	Weiterführende Bemerkungen	91
7.6	Das Newton-Verfahren	92
7.6.1	Grundlagen des Verfahrens	92
7.6.2	Aufbau des Algorithmus	93
7.6.3	Weiterführende Bemerkungen	94
7.7	Das Newton-Verfahren mit konsistenter Approximation der Hesse-Matrix	95
7.7.1	Grundlagen des Verfahrens	95
7.7.2	Aufbau des Algorithmus	96
7.7.3	Weiterführende Bemerkungen	97
7.8	Das Verfahren der variablen Metrik (Quasi-Newton-Verfahren)	97
7.8.1	Grundlagen des Verfahrens	97
7.8.2	Aufbau des Algorithmus	99
7.8.3	Weiterführende Bemerkungen	100
8	Beschränkte nichtlineare Optimierung	101
8.1	Die adaptive Zufallssuche	102
8.1.1	Grundlagen des Verfahrens	102
8.1.2	Aufbau des Algorithmus	103
8.1.3	Weiterführende Bemerkungen	104
8.2	Das erweiterte Polytopverfahren	104

8.2.1	Grundlagen des Verfahrens	104
8.2.2	Algorithmus	106
8.2.3	Weiterführende Bemerkungen	108
8.3	Das Schnittebenenverfahren	109
8.3.1	Grundlagen des Verfahrens	109
8.3.2	Aufbau des Algorithmus	110
8.3.3	Weiterführende Bemerkungen	111
8.4	Das SQP-Verfahren	112
8.4.1	Grundlagen des Verfahrens	112
8.4.2	Aufbau des Algorithmus	113
8.4.3	Weiterführende Bemerkungen	114
8.5	Das erweiterte Newton-Verfahren	114
8.5.1	Grundlagen des Verfahrens	114
8.5.2	Aufbau des Algorithmus	116
8.5.3	Weiterführende Bemerkungen	117
8.6	Verfahren mit Straf- und Barrierefunktionen	117
8.6.1	Grundlagen des Verfahrens	117
8.6.2	Der Algorithmus	119
8.6.3	Weiterführende Bemerkungen	120
9	Globalisierung	123
9.1	Dämpfungs- und Regularisierungsmethoden	123
9.2	Hybride Methoden	127
9.3	Einbettungsmethoden	128
10	Innere-Punkte-Methoden	131
10.1	Das Projektionsverfahren	131
10.1.1	Grundlagen des Verfahrens	131
10.1.2	Aufbau des Algorithmus	133
10.1.3	Weiterführende Bemerkungen	136
10.2	Kurzschrittverfahren	136
10.2.1	Herleitung des Verfahrens	136
10.2.2	Beschreibung des Algorithmus	138
10.2.3	Weiterführende Bemerkungen	139
11	Parameteridentifikation	141
11.1	Parameterschätzung auf der Grundlage linearer Quadratmittelprobleme	142
11.2	Nichtlineare Parameterschätzung und nichtlineare Optimierungsverfahren	145
11.3	Das Gauß-Newton-Prinzip und ein darauf beruhendes Verfahren	146
11.3.1	Aufbau des Algorithmus	147
11.4	Parameterschätzung und SQP-Verfahren	149
11.5	Parameteridentifikation in Differenzialgleichungen	150
11.5.1	Grundlagen	150

11.5.2	Weiterführende Bemerkungen	152
12	Optimale Steuerung	155
12.1	Einführung	155
12.2	Umwandlung in eine nichtlineare Optimierungsaufgabe	155
12.3	Aufbau des Algorithmus	156
12.4	Implementierte numerische Methoden	157
13	Form- und Strukturoptimierung	161
13.1	Zusammenhang zwischen Bemessungsvariablen und Zustandsvariablen	161
13.2	Lösung von Strukturoptimierungsproblemen mit SQP-Verfahren	163
13.3	Ein weiteres Beispiel	166
14	Optisoft – Ein C++-Softwaresystem zur Optimierung	167
14.1	Einführung	167
14.2	Allgemeine Informationen über Optisoft	168
14.3	Handhabung von Optisoft	170
14.3.1	Formulierung eines Problems	171
14.3.2	Auswahl des Algorithmus	182
14.4	Übersicht über Softwarepakete	184
	Anhang A Referenzmanual	187
	Anhang B Liste der Beispiele	193
	Literatur	195
	Stichwortverzeichnis	199

Vorwort

Bücher zur Implementierung numerischer Verfahren der Optimierung sind seit vielen Jahren gefragt. Die Behandlung mathematisch-naturwissenschaftlicher, technischer und ökonomischer Fragestellungen erfordert in wachsendem Umfang die Lösung linearer oder nichtlinearer Optimierungsaufgaben. Gegenüber den ersten Bemühungen in den 40er- und 50er-Jahren haben sich hierfür die Voraussetzungen auf dem Gebiet der Informatik wesentlich verbessert. Nicht nur Rechenzeit und Speicherplatz haben eine andere Bewertung erfahren, auch Programmierparadigmen und die Nutzung von Dialogmöglichkeiten haben sich geändert. Dieser Entwicklung folgend, werden im vorliegenden Buch Probleme und Lösungsverfahren als Klassen der objektorientierten Programmierung aufgefasst. Die Formulierung der zu lösenden Optimierungsaufgabe und die Auswahl der Lösungsmethode erfolgt im Dialog, die Ergebnisse der Berechnung werden automatisch gespeichert. Im Unterschied zu komplexen Systemen, wie Matlab sind die einzelnen Routinen modifizierbar und separat nutzbar. Seit den Arbeiten von Kantorovich [1] und Dantzig [2] zum Simplexverfahren hat auch die Entwicklung effektiver numerischer Verfahren der Optimierung eine stürmische Entwicklung genommen. Ihre theoretische Begründung und sachgerechte Implementierung stellt inzwischen einen eigenständigen Problembereich dar, welcher als Numerik der Optimierung (in englischer Sprache als „Computational Mathematical Programming“, in russischer Sprache als „Vycislitelnye metody programirovaniya“) bezeichnet wird. Die Aneignung der auf diesem Gebiet vorhandenen Erkenntnisse, noch mehr aber das Erleben des Zusammenhangs von beschriebenem Algorithmus, umgesetztem Programm und bereitgestellter Nutzeroberfläche werden zum Bedürfnis des an der Optimierung interessierten Praktikers. Gegenstand des Buches sind deshalb nicht in erster Linie theoretische Grundlagen, sondern Fragen der praktischen Realisierung der Verfahren mit modernen Mitteln der Informatik. Es soll einen Einstieg in die Behandlung von Optimierungsaufgaben auf Computern ermöglichen.

Für praktische Hilfeleistungen beim Zustandekommen des Buches bin ich Klaus Schönefeld zu Dank verpflichtet. In gleicher Weise danke ich Thomas Cassebaum für die Möglichkeit, die von ihm bereitgestellte Entwicklungsumgebung „SmallCpp“ nutzen zu können und in C++-Fragen in ihm jederzeit einen guten Gesprächspartner gefunden zu haben. Ermutigende Worte und gute Ratschläge

vieler Kollegen, insbesondere von Diethard Pallaschke, Oleg Burdakov, Manfred Grauer und Gerd Langensiepen, haben den Entstehungsprozess befördert. Dem Wiley-Verlag danke ich für die Möglichkeit, die Ergebnisse meiner Überlegungen zu publizieren. Schließlich möchte ich meiner Frau Hannelore für das Verständnis danken, mit dem sie die Belastung mitgetragen hat, welche dem Autor aus dem Schreiben eines Buches erwächst. Die Publikation von Algorithmen und Programmen schließt zu erwartende Kritiken und Hinweise von vornherein ein. Sie werden von mir sorgfältig berücksichtigt und in die Aufbereitung weiterer Programmversionen eingearbeitet.

Claus Richter

1

Einleitung

1.1

Das lineare und das nichtlineare Optimierungsproblem

Im vorliegenden Buch werden Optimierungsaufgaben betrachtet, die dadurch charakterisiert sind, dass eine lineare oder nichtlineare Zielfunktion f unter linearen oder nichtlinearen Ungleichungsnebenbedingungen minimiert wird, d. h.

$$f(x) = \min! \quad \text{bei} \quad x \in G = \{x : g_i(x) \leq 0 \quad i \in I\}, \quad (1.1)$$

wobei I

$$I = \{i : i = 1, \dots, m\}$$

die Indexmenge der Ungleichungsrestriktionen bezeichnet. Gleichungsrestriktionen werden der Übersichtlichkeit halber zunächst weggelassen. An geeigneten Stellen werden sie zusätzlich berücksichtigt.

1.2

Definitionen und Bezeichnungen

Für die weiteren Überlegungen benötigen wir folgende Bezeichnungen:

- n -dimensionaler Euklidischer Raum: R^n ,
- Menge der reellen Zahlen: R ,
- nichtnegativer Orthant des n -dimensionalen Euklidischen Raumes: R_+^n ,
- Euklidische Norm: $\|x\|_2 = (x^T x)^{1/2}$,
- Betragssummennorm: $\|x\|_1 = \sum_{i=1}^m |x_i|$,
- (m, n) -Matrix A : rechteckiges Zahlenschema $A = (a_{i,j})$ von $m * n$ Zahlen, angeordnet in m Zeilen und n Spalten,
- quadratische Matrix: (m, n) -Matrix A mit $m = n$,
- Diagonalmatrix A : quadratische Matrix A mit $a_{ij} = 0$ für $i \neq j$ und $a_{ii} \neq 0$,
- Einheitsmatrix I : Diagonalmatrix A mit $a_{ii} = 1$,
- obere Dreiecksmatrix A : quadratische Matrix A mit $a_{ij} = 0, i > j$,
- untere Dreiecksmatrix A : quadratische Matrix A mit $a_{ij} = 0, i < j$,

- positiv definite Matrix A : quadratische Matrix A mit $x^T A x > 0$ für alle $x \neq 0$,
- symmetrische Matrix A : quadratische Matrix A mit $a_{ij} = a_{ji}$,
- transponierte Matrix A^T zu A : Matrix A^T mit $A^T = (a_{ji})$,
- inverse Matrix A^{-1} zur Matrix A : Matrix mit der Eigenschaft $A * A^{-1} = I$,
- nichtsinguläre Matrix A : die inverse Matrix A^{-1} zu A existiert,
- orthogonale Matrix A : Matrix mit der Eigenschaft $A^T = A^{-1}$,
- transponierter Vektor: $x^T = (x_1, \dots, x_n)$,
- Gradient einer Funktion $f : R^n \rightarrow R$

$$\nabla f(x) := \left(\frac{\partial}{\partial x_1} f(x), \dots, \frac{\partial}{\partial x_n} f(x) \right)^T,$$

- Hesse-Matrix einer Funktion $f : R^n \rightarrow R$

$$\nabla^2 f(x)_{ij} := \frac{\partial^2}{\partial x_i \partial x_j} f(x) \quad i, j = 1, \dots, n,$$

- Lagrange-Funktion für die Aufgabe (1.1)

$$L(x, u) = f(x) + \sum_{i=1}^m u_i g_i(x),$$

- Ableitung der Lagrange-Funktion nach den Komponenten des 1. Arguments

$$\nabla_1 L(x, u) = \nabla f(x) + \sum_{i=1}^m u_i \nabla g_i(x),$$

- zweite Ableitung der Lagrange-Funktion nach den Komponenten des 1. Arguments

$$\nabla_{11} L(x, u) = \nabla_{11} f(x) + \sum_{i=1}^m u_i \nabla_{11} g_i(x),$$

- Indexmenge $I(x)$ der in x aktiven Restriktionen

$$I(x) = \{i : g_i(x) = 0, i \in \{1, \dots, m\}\},$$

- Vektor, dessen Komponenten alle gleich 1 sind: $e = (1, \dots, 1)^T$,
- i -ter Einheitsvektor: $e_i = (0, \dots, 0, 1, 0, \dots, 0)^T$,
- die Menge $G^0 := \{x : g_i(x) < 0, i = 1, \dots, m\}$.

1.3

Spezialfälle linearer und nichtlinearer Optimierungsaufgaben

Besitzen Zielfunktion f und der zulässige Bereich G bzw. Nebenbedingungen g_i und g_j eine spezielle Gestalt, so können zur Lösung von (1.1) spezielle Verfahren herangezogen werden. Für die Zielfunktion f sind folgende Strukturen

interessant:

1. Allgemeine nichtlineare Zielfunktion $f(x)$.
2. Lineare Zielfunktion $f(x) = c^T x$.
3. Quadratische Zielfunktion $f(x) = \frac{1}{2} x^T C x + d^T x$.
4. Quadratsumme (Regression)

$$f(x) = \sum_{i=1}^m (y_i - f(x, t_i))^2, \quad y_i - \text{Messwerte zum Messpunkt } t_i.$$

5. Maximum von Funktionen $f(x) = \max f_j(x) \quad (j = 1, \dots, l)$.

In Bezug auf die Nebenbedingungen N sind folgende Situationen typisch:

1. Allgemeine nichtlineare Nebenbedingungen.
2. Lineare Nebenbedingungen $g_i(x) = a_i^T x + b_i \quad (i = 1, \dots, m)$.
3. Keine Nebenbedingungen $G = R^n$.

In den folgenden Kapiteln werden spezielle Kombinationen von Zielfunktion und Nebenbedingungen eine besondere Rolle spielen:

- lineare Optimierung (L): $f2 + N2$,
- quadratische Optimierung (Q): $f3 + N2$,
- allgemeine nichtlineare Optimierungsaufgabe (C): $f1 + N1$,
- unbeschränkte Minimierung (U): $f1 + N3$,
- Regressionsprobleme (P): $f4 + N3, f4 + N1$.

Die Spezifikationen L, Q, C, U und P werden in der Charakterisierung der implementierten Beispiele im Programmsystem „Optisoft“ verwendet. Über die dargestellten Kombinationen von Zielfunktion und Nebenbedingungen hinaus spielen Aufgaben der nichtglatten Optimierung eine besondere Rolle. Diese finden im vorliegenden Buch keine Beachtung. Gleiches gilt auch für Optimierungsaufgaben mit sehr vielen Variablen: $n > 100$, sofern sie nicht als Teilprobleme zur Lösung von (1.1) auftreten.

Obwohl die spezifische Gestalt von Zielfunktion und Nebenbedingungen interessant ist, wie etwa in der geometrischen Optimierung

$$f(x) = \sum_{k=1}^r c_{k(o)} \prod_{i=1}^n x_i a_{ik}^0$$

$$g_j(x) = \sum_{k=1}^r c_{k(j)} \prod_{i=1}^n x_i a_{ij} \quad (j = 1, \dots, m),$$

wird diese nicht explizit berücksichtigt.

In der Betrachtung von Optimierungsverfahren gehen wir von dem Grundmodell (1.1) aus. Für Least-Square-Probleme in Differenzialgleichungsmodellen und bei Strukturoptimierungsproblemen liegen spezielle Aufgaben zugrunde. Diese werden in den folgenden Kapiteln näher erläutert.

1.4

Anwendungen

Nichtlineare Optimierungsprobleme spielen in vielen Anwendungsbereichen eine wichtige Rolle, z. B. in der

- Luft- und Raumfahrt (Steuerung, Konstruktion),
- Mechanik (Optimierung mechanischer Strukturen, z. B. von Tragwerken),
- Elektrotechnik (Transformator konstruktion),
- Chemie (Gleichgewichtsprobleme),
- Medizin, Soziologie (Statistische Probleme),
- Betriebswirtschaft (Planungsmodelle),
- Physik (Kernforschung),
- Energiewesen (Energieverteilung).

Typische Anwendungsbeispiele finden sich in den Büchern von Bracken und McCormick [3] oder Beightler und Phillips [4]. Einige mathematische Fragestellungen, welche bei der Lösung praktischer Probleme auf Optimierungsverfahren zurückgreifen, werden im Buch näher betrachtet:

1.4.1

Strukturoptimierung

Die Strukturoptimierung wird schon seit einigen Jahren in der computergestützten Konstruktion eingesetzt. In der zugrundeliegenden Aufgabenstellung wird dabei zwischen Querschnitts-, Form-, und Topologieoptimierung (der eigentlichen Strukturoptimierung) unterschieden. Grundlegende Fragestellung ist dabei, die Struktur und die Abmessungen von Konstruktionen derart zu wählen, dass zum einen die mechanischen Randbedingungen erfüllt und zum anderen der Materialeinsatz und damit die Kosten möglichst gering sind.

Obwohl die Berücksichtigung der Nebenbedingungen oft die Koppelung mit komplizierten Berechnungsvorschriften – z. B. FEM-Solvern – erfordert, soll das Grundprinzip an folgendem Beispiel erläutert werden:

Beispiel 1.1 Ziel ist die Erstellung von Bemessungstabellen für geschweißte I-Träger mit Querschnitten minimalen Gewichts (Abb. 1.1).

Da das Gewicht eines Trägers mit vorgegebener Länge proportional zum Querschnitt ist, lautet die Zielfunktion

$$f(x) = (x_1 - 2x_4)x_2 + 2x_3x_4 .$$

Tragsicherheitsnachweise (g_1, g_2), Beulsicherheitsnachweise (g_3, g_4) und konstruktive Restriktionen ($g_5 - g_{10}$) führen zu den Nebenbedingungen $g_i \leq 0$:

$$g_1(x) = \begin{cases} M_{pl}(x) - M_v & \text{für } |N_v|/f(x) < 0.1\sigma_F \\ M_{pl}(x)(1.1 - |N_v|/(f(x)\sigma_F)) - M_v & \text{sonst} \end{cases}$$

$$g_2(x) = 0.8 f(x)\sigma_F - N_v$$

$$g_3(x) = 17 - \frac{x_3}{x_4}$$

$$g_4(x) = \begin{cases} 43 - x_1/x_2 & \text{für } |N_v|/f(x) < 0.27\sigma_F \\ 70(1.1 - 1.4|N_v|/(f(x)\sigma_F)) - x_1/x_2 & \text{sonst} \end{cases}$$

$$g_5(x) = 1000 - x_1$$

$$g_6(x) = 60 - x_2$$

$$g_7(x) = 60 - x_4$$

$$g_8(x) = x_2 - 5$$

$$g_9(x) = x_4 - 5$$

$$g_{10}(x) = \frac{x_1}{4} - x_3$$

wobei $M_{pl}(x) := \sigma_F((x_1 - x_4)x_3x_4 + (x_1 - 2x_4)^2x_2/4)$.

Die Größen M_v , N_v und σ_F sind konstante Parameter. Die Anzahl der Variablen ist 4, und es liegen 10 Ungleichheitsrestriktionen vor.

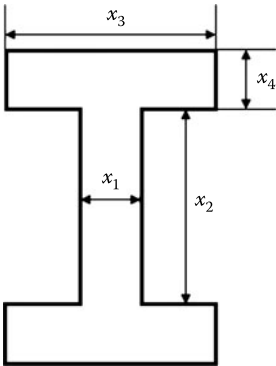


Abb. 1.1 Stahlträger.

1.4.2

Das Least-Squares-Problem

Spezielle nichtlineare Optimierungsaufgaben treten bei der Parameterbestimmung von Modellen auf, die einen in Natur- oder Technikwissenschaften vorliegenden Zusammenhang qualitativ beschreiben. Sind über diesen Zusammenhang Resultate von Experimenten bekannt, kann man die Methode der kleinsten

Quadrate anwenden, um die Koeffizienten näherungsweise zu bestimmen. Das zugehörige Optimierungsproblem lautet:

$$f(z) = \|z\| = \min!$$

bei

$$\begin{aligned} z_i &= h_i(x) = y_i - y(x, t_i) \quad (i = 1, \dots, l), \\ g_j(x) &= 0 \quad (j = 1, \dots, m_1) \\ g_j(x) &\leq 0 \quad (j = m_1 + 1, \dots, m) \end{aligned} \quad a \leq x \leq b. \quad (1.2)$$

Hierbei sind

- $y(x, t)$ – die gewählte Modellfunktion,
- x – der Parametervektor, dessen Komponentenwerte zu bestimmen sind
- t_i – der i -te Wert der (u. U. vektorwertigen) unabhängigen Veränderlichen,
- y_i – die i -te Beobachtung der (u. U. vektorwertigen) unabhängigen Veränderlichen,
- a, b – Schrankenvektoren für den Vektor x .

Entsprechend der Wahl der Norm haben wir es mit einer linearen oder quadratischen Zielfunktion zu tun. Die vorliegende Formulierung gestattet die Berücksichtigung zusätzlicher Nebenbedingungen. Beim Vorliegen von Differenzialgleichungen wird die Aufgabe wie folgt modifiziert:

$$f(z) = \|z\| = \min!$$

bei

$$\begin{aligned} z_i &= y_{ij} - y(x, t_j) \\ g_j(x) &= 0 \\ g_j(x) &\leq 0 \end{aligned} \quad a \leq x \leq b. \quad (1.3)$$

$$\dot{y} = \phi(x, y, t)$$

Eventuell treten zusätzlich Anfangsbedingungen der Form

$$y(t_o) = y^o \quad (1.4)$$

auf.

Diese können gegebenenfalls in die Least-Square-Formulierung einbezogen werden.

1.4.3

Optimale Steuerung

Das Problem der optimalen Steuerung besteht darin, eine Funktion unter Differenzialgleichungsnebenbedingungen sowie Anfangs- und Endbedingungen zu