

# Formoptimierung von Flächentragwerken

von

**Kai-Uwe Bletzinger**

Bericht Nr. 11 (1990)  
Institut für Baustatik der Universität Stuttgart  
Professor Dr.-Ing. E. Ramm  
Stuttgart 1990



2  
11.8.91

---

Berichte können über das Institut für Baustatik der Universität Stuttgart,  
Pfaffenwaldring 7, Postfach 1140, 7000 Stuttgart 80, bezogen werden.

# **Formoptimierung von Flächentragwerken**

von der Fakultät Bauingenieur- und Vermessungswesen der Universität Stuttgart  
zur Erlangung der Würde eines Doktors der Ingenieurwissenschaften (Dr.-Ing.)  
genehmigte Abhandlung

vorgelegt von

**Kai-Uwe Bletzinger**

aus Stuttgart

Hauptberichter: Prof. Dr.-Ing. E. Ramm

Mitberichter: Prof. Dr.-Ing. K. Linkwitz

Tag der mündlichen Prüfung: 29. Juni 1990

Institut für Baustatik der Universität Stuttgart

1990

## ZUSAMMENFASSUNG

Die vorliegende Arbeit befaßt sich mit der Formoptimierung von Flächentragwerken. Gesucht werden Tragwerke, deren Form sich nach bestimmten Zielvorstellungen (z.B. minimales Gewicht, minimale Formänderungsenergie) unter Berücksichtigung des zulässigen Spannungs- und Verformungsverhaltens linear elastischen Materials ergeben. Es werden dazu Methoden aus den Bereichen

- der finiten Elemente (zur Tragwerksberechnung)
- der mathematischen Programmierung (zur Optimierung)
- des Computer Aided Geometric Design, CAGD (zur Geometriebeschreibung)

zusammen eingesetzt. Dabei entscheiden die Interaktionen der verschiedenen Methoden und der mit ihnen beschreibbaren Teilmodelle über die Flexibilität und Anwendbarkeit des resultierenden Gesamtmodells der Strukturoptimierung. Um die vielfältigen und weitreichenden Möglichkeiten insbesondere der rein mathematisch formulierten Verfahren der mathematischen Programmierung und des CAGD für die Anwendung zur Formoptimierung von Tragwerken nutzbar zu machen, bedarf es detaillierter Kenntnisse der Grundlagen. Die Kapitel 2 bis 4 befassen sich deshalb eingehend mit diesen Verfahren unter den Aspekten der Anwendung in der Strukturoptimierung sowie der Effizienzsteigerung für die betrachteten Spezialfälle der Tragwerksoptimierung. In Kapitel 5 werden die genannten Teilmodelle zusammengefaßt und anwendungsgerecht strukturiert, so daß selbst aufwendige Optimierungsprobleme mit wenigen charakteristischen Optimierungsvariablen flexibel formuliert werden können. Dies wird in Kapitel 6 an verschiedenen Beispielen demonstriert: der Anwendung auf grundlegende Fragen der Formgestaltung von Minimalflächen und Schalen sowie der beanspruchungsgerechten, praxisorientierten Auslegung eines Trägers mit vielen Durchbrüchen.

## Abstract

The subject of this thesis is shape optimization of plates and shells. Such kind of structural shapes are to be developed which are determined by certain objectives (e.g. minimal weight, minimal strain energy) where the stress and displacement behaviour of linear elastic material is considered. For that purpose methods of

- finite elements (to analyze the structural behaviour)
- mathematical programming (to optimize)
- Computer Aided Geometric Design, CAGD (to describe the geometry)

are used together. The interactions of these different methods and of the sub-models which can be described by them decides about the flexibility and practicability of the overall model of structural optimization. Detailed knowledge about the underlying fundamental concepts is necessary to utilize the manifold capabilities of the purely mathematically formulated methods of mathematical programming and CAGD. Therefore, chapters two to four are dealing extensively with those methods with special regard to applications and increased efficiency for problems in the field of structural optimization. The sub-models are brought together in chapter five. They are structured to improve applicability in a way that even complex problems can be formulated in a flexible manner by a few characteristic optimization variables. This will be demonstrated by different examples in chapter six: applications to fundamental questions of form finding of minimal surfaces and shells well as the practical design of a Vierendeel girder.

## **Vorwort**

Die vorliegende Arbeit entstand als Dissertation während meiner Tätigkeit am Institut für Baustatik der Universität Stuttgart. Ich bedanke mich ganz besonders bei Professor Dr.-Ing. E. Ramm für die Anregungen, für die Unterstützung und die Übernahme des Hauptberichtes.

Herrn Professor Dr.-Ing. K. Linkwitz danke ich für das Interesse an dieser Arbeit, die Übernahme des Mitberichts und für die angenehme und lehrreiche Zusammenarbeit im Sonderforschungsbereich 230 "Natürliche Konstruktionen".

Mein Dank gilt auch den Kollegen des Instituts für Baustatik für die gute Zusammenarbeit und die angenehme Zeit. Ganz besonders danke ich Stefan Kimmich und Hans Stegmüller für die Zusammenarbeit bei den umfangreichen Programmentwicklungen, die im Laufe der Arbeit bewältigt werden mußten.

Die Arbeit wurde wesentlich inspiriert von der Mitarbeit im bereits erwähnten SFB 230. Die regelmäßigen Treffen mit den Arbeitskollegen aus den Bereichen der Architektur, Biologie, Physik und Philosophie und die gemeinsame Arbeit waren ein stimulierendes Erlebnis, das ich nicht missen möchte.

Kai-Uwe Bletzinger

<b>1. Einleitung</b>	<b>3</b>
<b>2 Mathematische Programmierung</b>	<b>11</b>
2.1 Allgemeine mathematische Definition eines Optimierungsproblems	11
2.2 Ein kleines Beispiel	12
2.3 Problemklassen der mathematischen Programmierung und Einordnung von Problemen der Strukturoptimierung	17
2.4 Zur Größe von Problemen der mathematischen Programmierung	20
2.5 Methoden der nichtlinearen Programmierung	21
2.6 Zur Auswahl einer Methode	28
2.7 Grundlagen der mathematischen Programmierung	30
2.7.1 Notwendige und hinreichende Bedingungen für lokale Minima	30
2.7.1.1 Unbeschränkte Probleme	30
2.7.1.2 Beschränkte Probleme	30
2.7.2 Eigenschaften der Lagrange-Funktion	35
2.7.3 Qualifikation und Redundanz von Nebenbedingungen	37
2.7.4 Konvergenzeigenschaften der Abstiegsmethoden	39
2.7.4.1 Abstiegsmethoden	39
2.7.4.2 Konvergenzordnung und Konvergenzrate	39
2.8 Algorithmen	41
2.8.1 Verfahren der eindimensionalen Optimierung (Line Search)	41
2.8.1.1 Quadratische Interpolation	41
2.8.1.2 Kubische Interpolation	43
2.8.1.3 Armijo-Test	43
2.8.2 Quasi-Newton-Methoden	45
2.8.2.1 Theorie der Quasi-Newton-Methoden	45
2.8.2.2 Konvergenzeigenschaften der Quasi-Newton-Verfahren	50
2.8.3 Ein Verfahren der konjugierten Gradienten für beschränkte Probleme mit einfachen Variablenrestriktionen	55
2.8.3.1 Allgemeines über Gradientenverfahren	55
2.8.3.2 Erweiterung für einfache Variablen-Restriktionen	57
2.8.4 Penalty- und Barriere-Methoden	60
2.8.4.1 Penalty-Verfahren	60
2.8.4.2 Barriere-Verfahren	64

2.8.4.3	Evolutionstrategie als zufallsgelenkte Barriere-Methode nullter Ordnung	66
2.8.5	Duale Methoden	69
2.8.5.1	Allgemeines	69
2.8.5.2	Lösung der dualen Aufgabe (Methode von Uzawa)	72
2.8.5.3	Die erweiterte Lagrange-Funktion; Multiplikatorenmethode	73
2.8.6	Lagrange-Methoden	78
2.8.6.1	Allgemeines	78
2.8.6.2	Abstiegssfunktionen	80
2.8.6.3	Das Verfahren von Han und Powell; SQP und BFGS	89
2.8.6.4	Bedeutung von Skalierung und Line Search in der Sequentiellen Quadratischen Programmierung	93
2.8.6.5	Sequentiell-Lineare Programmierung als ein Verfahren der Lagrange-Methoden	97
<b>3</b>	<b>Approximationstechniken</b>	<b>103</b>
3.1	Überblick	103
3.2	Explizite Approximation von Zielfunktion und Nebenbedingungen	107
3.2.1	Physikalische Begründung, konvexe Linearisierung	107
3.2.2	Separierbarkeit und Dualität	110
3.2.3	Gemeinsamkeiten mit den Verfahren der Optimalitätskriterien	112
3.2.4	Das Verfahren der beweglichen Asymptoten	114
3.2.4.1	Approximationsvorschrift	114
3.2.4.2	Lösung des approximierten Unterproblems	116
3.2.4.3	Adaption der Asymptoten	124
3.2.4.4	Inkonsistente Nebenbedingungen	127
3.2.4.5	Modifikationen	129
3.2.4.5.1	Unbeschränkte Probleme	129
3.2.4.5.2	Gleichheitsbedingungen	133
3.2.4.5.3	Verallgemeinerte Approximation	135
3.2.4.6	Formoptimierung einer Zuglasche	136
3.3	Direkte Approximation von Spannungsbedingungen	140
3.3.1	Fließspannungsbedingungen in Fachwerken	140
3.3.2	Knickspannungen in Fachwerken	144
3.3.3	Erweiterung für ebene Flächentragwerke	148

<b>4</b>	<b>Formbeschreibung, Formkontrolle</b>	<b>149</b>
4.1	Formoptimierung	149
4.2	Formgenerierung	152
4.2.1	Das Konzept der Design-Elemente	152
4.2.2	Formoptimierung mit Design-Elementen	154
4.2.3	FE-Netzgeneratoren	155
4.3	Formgenerierung und -optimierung mit CARAT	156
4.3.1	Zur Programmorganisation	156
4.3.2	Elementkatalog	158
4.3.2.1	Elementhierarchie	158
4.3.2.2	1-D-Elemente	160
4.3.2.2.1	Lagrange-Elemente	160
4.3.2.2.2	Bézier-Spline	161
4.3.2.2.3	B-Spline	163
4.3.2.2.4	Kreise und Spirallinien	168
4.3.2.3	2-D-Elemente	170
4.3.2.3.1	Lagrange-Elemente	170
4.3.2.3.2	Bézier-Element	172
4.3.2.3.3	Coons-Elemente	173
4.3.2.4	3-D-Elemente	176
4.3.3	Geometrische Kontinuität	176
4.3.3.1	1-D-Elemente, Tangentenelemente	176
4.3.3.2	2-D-Elemente, Kontinuitätsselemente	180
4.3.4	Koordinatensysteme	186
4.3.5	Design-Macros	193
4.3.6	Sensibilitätsanalyse	196
<b>5</b>	<b>Variablenverknüpfungen</b>	<b>203</b>
5.1	Optimierungsmodell und Variablen	203
5.2	Linking Levels	204
5.3	Sensibilitätsanalysis	207
5.4	Vorgeschriebene Bewegungsrichtungen	207
5.5	Symmetriebedingungen	209
5.6	Linearkombinationen	210

5.7	Restriktionen der unabhängigen Variablen	212
5.8	Programmtechnisches	212
5.9	Definition von Verknüpfungen	214
<b>6.</b>	<b>Beispiele</b>	<b>215</b>
6.1	Minimalflächen	215
6.2	Vierendeel-Träger	221
6.2.1	Situation	221
6.2.2	Modellierung	222
6.2.3	Ergebnisse	226
6.3	Formoptimierung rotationssymmetrischer Schalen	229
6.3.1	Problemstellung	229
6.3.2	Modellbeschreibung	230
6.3.3	Ergebnisse	233
	<b>Schlussfolgerungen</b>	<b>237</b>
	<b>Literatur</b>	<b>239</b>
	<b>Anhang</b>	<b>250</b>
A1.	Kantorowitsch-Ungleichung	250
A2.	Formbeiwerte für Stabquerschnitte	252
A3.	3-D Coons-Element, vollständige Formfunktion	253

## BEZEICHNUNGEN

Soweit wie möglich wurde darauf geachtet, daß Symbole und Bezeichnungen an den Stellen ihrer Erscheinung erklärt werden. Die Kapitel sind in sich abgeschlossen, so daß nur wenige übergeordnete Bezeichnungen verbleiben:

$x$	Optimierungsvariablen
$x_L$	untere Variablenschranken
$x_U$	obere Variablenschranken
$f(x)$	Zielfunktion
$h_j(x)$	Gleichheitsbedingungen
$g_j(x)$	Ungleichheitsbedingungen
$\lambda_j$	Lagrangemultiplikatoren der Ungleichheitsbedingungen
$\mu_j$	Lagrangemultiplikatoren der Gleichheitsbedingungen
$L(x, \lambda, \mu)$	Lagrangefunktion
$n$	Anzahl der Optimierungsvariablen
$m$	Anzahl der Nebenbedingungen
$m_e$	Anzahl der Gleichheitsbedingungen
$k$	Iterationsindex
$(\cdot)_{,y}$	partielle Ableitung nach $y$
$\nabla, \nabla_x$	Vektor der partiellen Ableitungen bzgl. der Optimierungsvariablen, Gradient
$\nabla^2$	Matrix der zweiten Ableitungen bzgl. der Optimierungsvariablen, Hessematrix
$\nabla_{\mu}, \nabla_{\lambda}$	partielle Ableitungen bzgl. der Lagrangemultiplikatoren
$I$	Einheitsmatrix
$r(u)$	1-dimensionale Gestaltsfunktion
$r(u, v)$	2-dimensionale Gestaltsfunktion
$r(u, v, w)$	3-dimensionale Gestaltsfunktion
$r_i$	Kontrollknoten der Gestaltsfunktionen
$u, v, w$	Formparameter

## 1. EINLEITUNG

*„... Ich bin der festen Überzeugung, daß selbst Philosophen, die sich mit Fragen der Ästhetik befassen, es als schwierig empfinden, die Ursachen unserer Gefühle für Formen zu erklären, die von den Gesetzen der Statik oder Dynamik bestimmt werden, weil diese Gesetze weder intuitiv verstanden noch von der Erfahrung unserer Vorfahren erklärt werden können. Aber es gibt keinen Zweifel, daß irgendein Produkt von hoher Effizienz immer ästhetisch befriedigend ist. In der Architektur, in welcher funktionale, statische und ökonomische Ansprüche ineinander verwoben sind, ist Wahrhaftigkeit eine unverzichtbare Bedingung für gute ästhetische Ergebnisse. Diese Maxime wird von den auffälligsten Bauwerken der letzten hundert Jahre offen gelegt. ...”*

Mit dieser Meinung steht Nervi [92] nicht alleine. Er befindet sich in Übereinstimmung mit vielen, nicht nur mit den anderen Großen seiner Art, die wie er das sichere und untrügliche Gespür für Form mit der Funktion eines Bauwerkes in Einklang bringen können. Dabei war er nicht nur Schöpfer und Konstrukteur, der sich auch mathematischer und mechanischer Hilfsmittel bediente und ausdrücklich dazu bekannte, sondern gleichzeitig auch der ausführende Unternehmer, der sich der Konkurrenz zu stellen hatte. Es bleibt sein Geheimnis und macht seine Einmaligkeit aus, wie er es verstand, Intuition mit wissenschaftlichen Verfahren zu vereinen, im freien Spiel der Ideen, technischen Methoden, ihrer Grenzen und Möglichkeiten.

Es ist anregend, das Zusammenspiel von Kraft und Form zu entdecken und die Gesetzmäßigkeiten zu studieren. Torroja [140] ist der Meinung, daß nur derjenige in der Lage sein kann, geeignete und damit gute Formen zu finden, der dieses Wissen verinnerlicht habe. Er schreibt:

*„... Vergeblich ist die Mühe dessen, der danach trachtete, die geeignete Form eines Tragwerkes aufzufinden, ohne sich vorher bis ins Mark die Spannungsgrundlagen angeeignet zu haben, die alle in diesem Tragwerk vorkommenden Festigkeitsprobleme regieren. ... Es genügt nicht, die Theorie der Festigkeitslehre und die Entwicklung der entsprechenden Berechnungen durchstudiert zu haben. Man muß darüber nachgedacht und experimentiert haben, bis man es als etwas Eigenes, Natürliches und Angeborenes empfindet, wie sich die Spannungen und Verformungen auswirken. ...”*

Das Experiment ist dabei ein wichtiges und für jeden leicht zu begreifendes, eher zu erführendes, Hilfsmittel. Frei Otto definiert in vielen seiner Schriften, nicht nur in [96], "natürliche" Formen als solche, die sich in Selbstbildungsprozessen im Spiel der Kräfte von selbst einstellen und die deshalb den Menschen seit jeher instinktiv vertraut sind und als richtig und optimal verstanden werden. Diese Formen sind von minimalem Ge-

wicht und stellen sich unter einem formgebenden Lastfall ein. Dazu gehören neben den Hängeformen und den aus ihrer Umkehrung entwickelten Bögen und Gewölben, Netzstrukturen [83] und pneumatisch vorgespannte Membrane. Isler [110] bewies, daß mit dem Prinzip der Hängeform Tragwerke gefunden werden können, die ästhetisch ansprechend, den Beanspruchungen optimal angepaßt sind, beliebige auch unregelmäßige Grundrisse überspannen können und dazuhin auch noch ökonomisch konkurrenzfähig sind.

Es ist von ganz besonderem Reiz, die den physikalischen Experimenten zugrunde liegenden Prinzipien mit mechanischen und mathematischen Verfahren zu erfassen und mit modernsten computer-orientierten Methoden abzubilden. Neben dem reinen, forschenden Erkenntnisdrang und der schieren Neugier, ob es denn gelänge, solche numerischen Experimente zu formulieren, modellieren und auszuführen, motiviert die Tatsache, daß es auf diesem Wege möglich ist, Parameter, deren Einflüsse und Auswirkungen zu trennen und quantitativ zu erfassen, die im physikalischen Experiment nur schwer oder gar nicht zugänglich sind. Zum Beispiel kann die Hängeform als Kompromißlösung zweier oder mehrerer formgebender Lastfälle besser auf analytischem bzw. numerischem Wege gefunden werden. Auch sind so schnell und einfach Parametervariationen durchführbar, ohne aufwendige Umbauten an der bestehenden Modelleinrichtung. Von zukünftigen, weiterführenden Möglichkeiten im Rahmen des Computer Aided Design oder gar Manufacturing soll hier gar nicht erst die Rede sein.

Dabei ist die Idee, anstelle des Festigkeitsnachweises eines Tragwerks bekannter Form gleichsam rückwärts mit vorgeschriebener Festigkeit sowie einer gewissen Zielvorstellung wie z.B. minimales Gewicht die zunächst unbekannte Form zu bestimmen, schon sehr alt und kann bis zu Galileo Galilei zurückverfolgt werden [151]. Wohlbekannt sind die charakteristischen Formen der hochaufgelösten, fachwerkartigen Michell-Strukturen, die sich trajektorienartig dem Kräfteverlauf anschmiegen. Obwohl sie gewöhnlich in ihrer reinen Form wegen ihrer hohen und deshalb kompromißlosen Spezialisierung technisch nicht verwirklicht sind, kehren die Prinzipien in jedem einigermaßen optimierten Fachwerk wieder, wenngleich sie sich vielfach erst auf den zweiten Blick zu erkennen geben. Viele haben versucht, auf analytischem Wege optimale Tragwerke gemäß den Gesetzen der Mechanik zu finden. Einer der bekanntesten war Prager [104], [116], dessen Ideen und Gedanken von seinen Mitarbeitern und Schülern bis heute bewahrt und weiterentwickelt worden sind. Allerdings ist die Anwendbarkeit dieses Vorgehens relativ beschränkt, denn es gelingt nicht, für alle denkbaren Tragwerke und Formen geschlossene Lösungen anzugeben. Deshalb ist es nicht verwunderlich, daß die Tragwerkoptimierung oder auch Strukturoptimierung erst mit der Entwicklung von Computern und den entsprechenden numerischen Verfahren der Mathematik und Informatik

aufzublühen begann. Einen ersten Höhepunkt erlebte sie in den sechziger Jahren, als erstmals die Verfahren der mathematischen Programmierung eingesetzt worden sind [122].

Es folgte eine große Zahl von Anwendungen, vor allem im Bereich der Luft- und Raumfahrttechnik, wo auch heute noch die Verfahren der Strukturoptimierung hauptsächlich verwendet werden. Inzwischen fanden sich auch breite Anwendungsgebiete im Maschinen- und Automobilbau. Neben dem Gewicht werden hier auch möglichst geringe Kerbspannungen oder homogene Spannungszustände als Optimierungsziele formuliert. Beabsichtigt werden dabei verbesserte Tragwerkseigenschaften bei Dauerbeanspruchungen oder die infolge des homogenisierten Spannungszustandes ermöglichte Fertigung mit einer konstanten Blechdicke. Das Bauwesen folgte eher zögerlich, obwohl viele der Pioniere der Strukturoptimierung Bauingenieure sind. Optimale Bemessung im Sinne der Kosten, häufig an den Normen orientiert, ist die wichtigste Zielvorstellung [139]. Balken- und Rahmentragwerke sind hier meistens Gegenstand der Strukturoptimierung; Platten, Scheiben oder gar Schalen sind schon seltener. Hingegen sind aus dem Maschinen- und Automobilbau einige, wenngleich wenige Anwendungen zur Formoptimierung von Schalen bekannt [21], [78]. Meist handelt es sich um Rotationsschalen oder andere, einfach zu beschreibende Geometrien; die Erweiterung auf freie Flächen im Raum steht noch weitgehend aus [14].

Seit ihren Anfängen wurden die Methoden der Strukturoptimierung ständig weiterentwickelt und erleben nun in den achtziger Jahren eine zweite Blüte, nachdem eine Phase der Stagnation während der siebziger Jahre erfolgreich überwunden werden konnte. Damals mußte festgestellt werden, daß die zur Verfügung stehenden Methoden speziell der mathematischen Programmierung und auch die Computer selbst nicht ausgereift genug waren, um einigermaßen realistische Probleme in vernünftiger Zeit zu lösen. In der Folge wurden spezielle Verfahren, die sogenannten Verfahren der Optimalitätskriterien, entwickelt, die sich an den spezifischen Strukturen bestimmter Probleme orientieren und in der Lage sind, diese äußerst effizient zu lösen. Diese Verfahren befassen sich allerdings meist mit Problemen der optimalen Querschnittsgestaltung bei fester Form.

Heute stehen ausgereifte numerische Verfahren der Mathematik und Mechanik zur Verfügung, mit deren Hilfe komplexe Aufgaben der Formoptimierung gelöst werden können. Dabei müssen die drei wichtigen Komponenten Geometriebeschreibung, mathematische Optimierung und Tragwerksberechnung zu der funktionalen Einheit "Strukturoptimierung" zusammengefaßt werden. Eschenauer und seine Mitarbeiter nennen das daraus abgeleitete Programmkonzept deshalb auch "Drei-Säulen-Konzept" [35]. Nach allgemeiner Überzeugung sind noch zwei weitere wichtige Bestandteile der

Strukturoptimierung zu nennen. Es sind dies die Sensibilitätsanalyse und die interaktive Unterstützung des Optimierungsfortschrittes. Die Sensibilitätsanalyse gibt Auskunft über das Tragwerksverhalten infolge der Änderung eines Entwurfsparameters und ist deshalb von großer Bedeutung bei der Bestimmung von Tendenzen, wie das zu optimierende Tragwerk zu verbessern ist. Sie ist der mit Abstand rechenintensivste Teil der Strukturoptimierung und hat sich auch deshalb zu einem eigenständigen Gebiet der Forschung entwickelt [67], [76]. Insbesondere bei Formoptimierungen ist die Bedeutung einer ständigen visuellen Kontrolle des Optimierungsfortschrittes mit Methoden der interaktiven Computergraphik von großer Bedeutung.

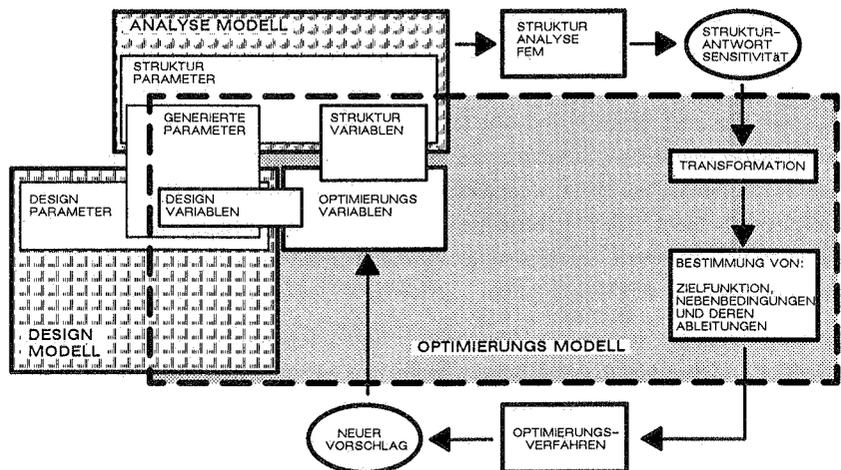


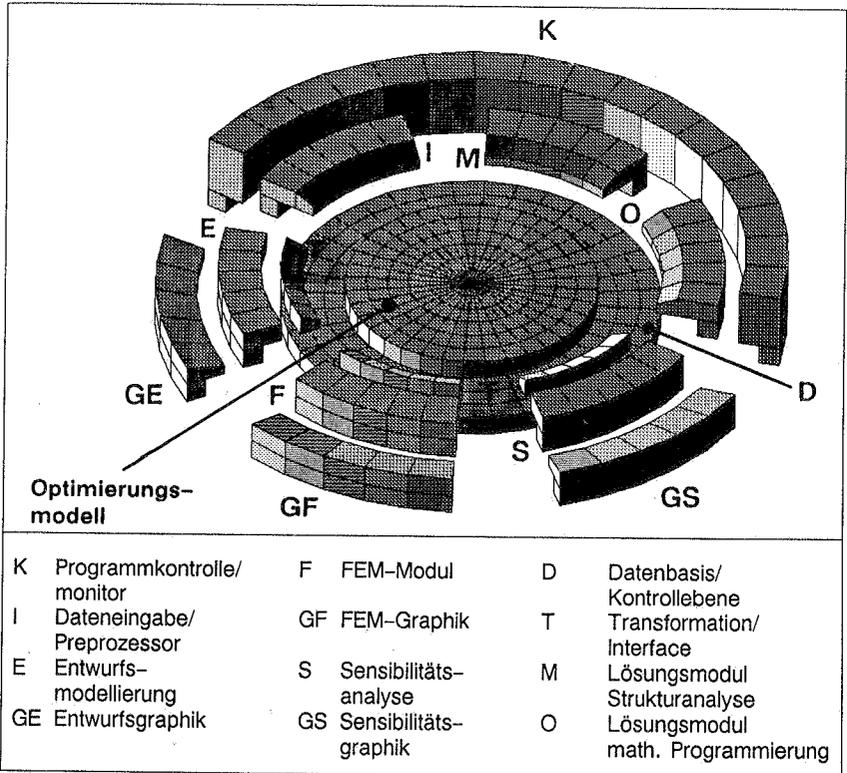
Bild 1.1: Interaktionen des Optimierungsmodells in CARAT

Die genannten fünf Punkte:

- Geometriebeschreibung (Entwurfsmodell, Methoden des CAGD)
- Analysemodell (Methode der finiten Elemente)
- Optimierungsverfahren (mathematische Programmierung)
- Sensibilitätsanalyse
- interaktive graphische Kontrolle

sind auch die Grundlagen des Optimierungsmodelles (Bild 1.1), welches im Programmsystems CARAT (Computer Aided Research Analysis Tool) [75], [132] verwirklicht ist. Dabei entscheidet die Flexibilität dieses Modells über die Anwendungsbreite und Differenzierung möglicher Aufgaben der Strukturoptimierung, deren besonderer Schwer-

punkt bei der Formoptimierung von Flächentragwerken liegt. Die Grundlage der Programmentwicklung ist eine übergreifende und durchdachte Datenstruktur, auf welcher alle vorhandenen Programm-Module aufgebaut sind. Damit ist eine enge datentechnische Verzahnung aller zur Optimierung notwendigen Programmteile möglich, bei gleichzeitig freien und voneinander unabhängigen Programmstrukturen. Bild 1.2 zeigt den schematischen Aufbau von CARAT mit der gemeinsamen Datenbasis, auf welche die übrigen Programmteile "aufgesteckt" sind.



**Bild 1.2:** schematische Darstellung der Programmstruktur von CARAT

Die vorliegende Arbeit befaßt sich mit den grundlegenden Aspekten der mathematischen Programmierung, der Methoden zur Geometriebeschreibung aus dem Bereich des Computer Aided Geometric Design (CAGD) sowie Methoden zur eleganten und treffenden Auswahl und Verknüpfung von Optimierungsvariablen sowie ihrer Anwendungen zur Formoptimierung von Flächentragwerken. Die Kenntnis all dieser Verfahren, ihrer Mög-

lichkeiten und Grenzen sowie ihrer komplexen Interaktionen (wie dies in Bild 1.1 angedeutet wird) ist die Voraussetzung dafür, daß die Anwendung auf Probleme der Formoptimierung gelingen kann. Angestrebt wird eine möglichst einfach handzuhabende, aber dennoch flexible und kontrollierbare Methode, mit der ohne weitere Programmierarbeiten Optimierungsaufgaben formuliert und gelöst werden können.

Kapitel 2 befaßt sich ausführlich mit den Methoden der mathematischen Programmierung. Es wird der gesamte Themenkreis aufgezeigt, dessen Kenntnis zum Verständnis der in CARAT zur Strukturoptimierung eingesetzten Verfahren, die Methode der sequentiellen quadratischen Programmierung in Verbindung mit Quasi-Newton-Verfahren sowie ein duales Verfahren, dem ein Verfahren der konjugierten Gradienten zugrunde liegt, notwendig ist.

In Kapitel 3 werden besondere Techniken vorgestellt, mit deren Hilfe das komplexe, implizit vorliegende eigentliche Optimierungsproblem approximiert und einer expliziten Formulierung zugeführt werden kann, um eine Effizienzsteigerung des gesamten Ablaufes zu bewirken. Diese Techniken sind zur Zeit als die Methoden der "konvexen Linearisierung" [44] bzw. der "beweglichen Asymptoten" [135] bekannt.

Kapitel 4 gibt einen Überblick über die Methoden des Computer Aided Geometric Design, wie sie in CARAT zur Modellierung der FE-Netze ein-, zwei- und dreidimensionaler Strukturen zum Einsatz kommen. Besonderes Gewicht wurde auf die Formulierung von Bedingungen gelegt, die garantieren, daß die generierten Formen auch während einer Formoptimierung die gewünschten Kontinuitätseigenschaften beibehalten. Sämtliche Ableitungen nach den Optimierungsvariablen, die zur Bestimmung der Sensibilität der Struktur benötigt werden, sind explizit aufgeführt.

In Kapitel 5 wird eine konsequente Strukturierung aller möglichen Optimierungsvariablen vorgenommen, die eine flexible und problemgerechte Formulierung des Optimierungsproblems ermöglichen. Es werden Verknüpfungsvorschriften angegeben, die mit einer einfach handzuhabenden, jedoch komplexen Sprache formuliert werden können. Zuletzt werden in Kapitel 6 drei typische Beispiele zur Formoptimierung von Flächentragwerken vorgestellt, an denen viele Einzelheiten der vorherigen Kapitel in der Anwendung gezeigt werden. In Beispiel 1 werden Minimalflächen über verschiedenen Grundrissen und mit verschiedenen vorgeschriebenen Rändern ermittelt. Beispiel 2 zeigt eine typische, praxisorientierte Anwendung zur Formoptimierung von Scheiben. Ermittelt wird das minimale Gewicht eines Vierendeel-Trägers mit 6 Durchbrüchen, die jeweils von ebenfalls 6 Variablen beschrieben werden. Nebenbedingungen sind hier maximal zulässige Höchstwerte der von Mises-Vergleichsspannungen sowie die Durchbiegung des Trägers. Im letzten Beispiel werden die optimalen Formen zunächst rotationssymmetrischer Schalen unter verschiedenen Lagerbedingungen und Lasten bestimmt wie Later-

nenlast, Eigengewicht, Schnee und Wind. Angestrebt werden hier Formen, die möglichst membrangerecht beansprucht werden. Dies wird erreicht über eine Gewichtsminimierung mit zulässigen von Mises-Spannungen als Nebenbedingungen oder der Minimierung der Formänderungsenergie bezüglich den Optimierungsvariablen. Alle Beispiele demonstrieren die Flexibilität des eingeschlagenen Vorgehens, die Ergebnisse spiegeln die engen Beziehungen zwischen Kraft und Form wider.

Der Gedanke ist verlockend, mit der Strukturoptimierung sei ein Computer-orientiertes Verfahren gefunden, mit dem auf einem Schlag die Formfindung und der Festigkeitsnachweis erledigt seien. Das mag für einzelne Fälle durchaus gelten, doch wie bei allen Methoden der Mathematik und Mechanik werden auch hier Voraussetzungen und Annahmen getroffen, beeinflussen die Werte vorgegebener Parameter das Ergebnis. Deshalb sollten diese Verfahren nur verantwortungsbewußt eingesetzt werden und die Ergebnisse stets einer strengen Interpretation unterworfen werden. Auch die Verfahren der Strukturoptimierung sollen den Entwerfenden nur unterstützen, nicht ersetzen.



## 2 MATHEMATISCHE PROGRAMMIERUNG

### 2.1 Allgemeine mathematische Definition eines Optimierungsproblems

Ganz allgemein ist ein mathematisches Programm ein Optimierungsproblem der Form:

$$\begin{aligned} & \text{minimiere} && f(\mathbf{x}) \\ & \text{unter Berücksichtigung von:} && \text{(NLP)} \\ & && h_j(\mathbf{x}) = 0 ; \quad j = 1, \dots, m_e \\ & && g_j(\mathbf{x}) \leq 0 ; \quad j = m_e + 1, \dots, m \\ & && \mathbf{x} \in S \{x_{Li} \leq x_i \leq x_{Ui} ; i = 1, \dots, n\} \subset \mathbb{R}^n \end{aligned}$$

Nach Minoux [88] läßt sich der Begriff "mathematische Programmierung" auf die ersten Entwicklungen in den Wirtschaftswissenschaften und den Anwendungen auf ökonomische Programme zurückführen. Ob für ökonomische Probleme oder wie in dieser Arbeit für Probleme der Strukturoptimierung, mit der Transformation der Optimierungsaufgabe in die Sprache der Mathematik führt der Wunsch, unter verschiedenen möglichen Lösungen bzw. Varianten die beste, d.h. die optimale, aufzufinden, zwangsläufig zu einer Quantifizierung. Dabei ist es gleichbedeutend, ob die optimale Lösung mit dem Minimum oder dem Maximum einer Maßfunktion, der sogenannten **Zielfunktion**  $f(\mathbf{x})$ , identifiziert wird. Überwiegend wird in der Literatur der Weg der Minimierung eingeschlagen, weshalb dies auch hier so eingeführt worden ist. Neben dem Maß der Zielfunktion wird eine Lösung oder ein Entwurf im Kontext der mathematischen Programmierung durch eine Menge von unabhängigen **Variablen** oder **Parametern** beschrieben, die für Aufgaben der Strukturoptimierung auch **Entwurfsvariable** genannt werden. In der hier gewählten Schreibweise werden die Variablen in einem Vektor  $\mathbf{x}$  zusammengefaßt. Die im **Entwurfsraum**  $\mathbb{R}^n$  definierten Variablen können i.a. jedoch nicht völlig frei gewählt werden, sondern müssen einem Satz von Nebenbedingungen genügen, die im allgemeinen nichtlineare Funktionen der Variablen sind. Diese Nebenbedingungen trennen den Entwurfsraum in einen **unzulässigen** und einen **zulässigen Bereich**, in welchem unter der Menge der zulässigen Lösungen die optimale zu suchen ist. Man unterscheidet dabei **Gleichheitsbedingungen**  $h(\mathbf{x})$ , **Ungleichheitsbedingungen**  $g(\mathbf{x})$  und **obere** sowie **untere Schranken** für die Wahl jeder einzelnen Variablen  $x_i$ . Zur Unterscheidung von den anderen Nebenbedingungen werden die Schranken im weiteren auch **Restriktionen** genannt. Die besondere Behandlung der Schranken, die im Grunde auch Ungleichheitsbedingungen sind, läßt sich durch ihre mathematisch äußerst einfache Formulierung als lineare Funktion einer einzigen Variablen begründen, weshalb sie auch in nahezu allen Lösungsverfahren besonders berücksichtigt werden. Bei der Entwicklung einer ein-

heitlichen Theorie und von Lösungsalgorithmen erweisen sich die Ungleichheitsbedingungen als besonders unangenehm. Denn gewöhnlich kann a priori nicht entschieden werden, ob irgendeine der Nebenbedingungen für die Wahl einer zulässigen Lösung maßgebend sein wird. In diesem Fall ist  $g_j(x) = 0$ , und die betreffende Nebenbedingung wird als **aktiv** bezeichnet. Dagegen ist eine Nebenbedingung **nicht aktiv**, wenn  $g_j(x) < 0$  ist. Natürlicherweise sind Gleichheitsbedingungen nach dieser Terminologie stets aktiv. Die Menge der aktiven Nebenbedingungen, auch der **aktive Satz** (englisch active set) genannt, ändert sich während einer Optimierung ständig, und es liegt auf der Hand, daß viel Aufwand gespart werden kann, wenn bei den Berechnungen tatsächlich nur die aktiven Nebenbedingungen berücksichtigt werden. Ist dagegen der aktive Satz der Optimallösung bekannt, können diese Nebenbedingungen als Gleichheitsbedingungen behandelt werden. Gleichheitsbedingungen haben den Vorzug, daß mit ihnen eine ihrer Anzahl entsprechende Zahl von Variablen eliminiert werden kann. Damit läßt sich die Dimension des Problems im Einzelfall enorm reduzieren. Eine bedeutende, wenngleich meist nicht erwähnte Klasse von Gleichheitsbedingungen stellen in der Strukturoptimierung die Gleichgewichtsbedingungen dar. Die Erfüllung des Gleichgewichts ist jedoch eine solche Selbstverständlichkeit, daß man gewöhnlich nicht gewahr wird, daß dabei die unbekanntes statischen Größen aus dem Optimierungsproblem herausgelöst werden. Nichtsdestoweniger gibt es auch Ansätze, die bewußt die Elimination nicht vornehmen und die Menge der Optimierungsvariablen um die statischen Unbekannten erweitern (s. Haftka [61]). Man verspricht sich davon gewisse Vorteile bei der Lösung der ohnehin schon großen Gleichungssysteme oder will auf diesem Wege die Inversion bzw. Dreieckszerlegung der Steifigkeitsmatrix umgehen.

Im folgenden sollen die eingeführten Begriffe an einem kleinen Beispiel dargestellt werden.

## 2.2 Ein kleines Beispiel

Form und Querschnitte des in Bild 2.1 dargestellten Zweibocks sollen für die angegebene Laststellung bestimmt werden, so daß das Gewicht des Tragwerkes minimal werde, ohne daß vorgeschriebene maximal zulässige Werte für die Durchbiegung unter der Last und für die Druckspannungen in den Tragwerkselementen überschritten werden.

Das Problem hat zwei freie Variablen, die Stichhöhe  $x$  und die Querschnittsfläche  $A$  sowie die unbekanntes Zustandsgröße  $u$ , die die Verschiebung des Tragwerkes unter dem Lastangriffspunkt beschreibt. Insgesamt sind vier Nebenbedingungen zu formulieren, die Gleichheitsbedingung für das Gleichgewicht sowie je eine Ungleichheitsbedingung für die maximal zulässige Verschiebung, die zulässige Fließ- und Knickspannung im Tragwerk. Aufgrund der Symmetrie von Tragwerk und Last genügt die Kontrolle der



2.) Fließspannungen:

$$-\frac{P}{2A} \cdot \frac{\sqrt{a^2 + x^2}}{x} \geq -\beta_s$$

-  $\beta_s$  ... zul. Druckspannung

3.) Euler-Knickspannung:

$$-\frac{P}{2A} \cdot \frac{\sqrt{a^2 + x^2}}{x} \geq -\frac{\pi^2 EcA}{(a^2 + x^2)}$$

c ... Formbeiwert (siehe Anhang A2)

Das Beispiel wird im folgenden wiederholt zur Erläuterung der Begriffe und Methoden der mathematischen Programmierung herangezogen. Dabei werden folgende Werte für die festen Parameter eingesetzt:

$$\begin{aligned} a &= 2 & ; & \beta_s = 1 & ; & E = 10 & ; & \gamma = 1,0 ; \\ A_L &= 0,1 & ; & x_L = 0,1 & ; & c = 1/(4\pi), & \text{Vollkreisquerschnitt (siehe Anhang A2)} \end{aligned}$$

Es werden vier Fälle unterschieden:

Fall a) Verschiebungs- und Fließbedingung (Bild 2.2a):

$$P = 1 ; u_{\max} = 0,8$$

$$\text{optimale Lösung: } x^* = 2,000 \quad A^* = 0,707 \quad f^* = 4,000$$

die Fließbedingung ist aktiv

Fall b) Fließ- und Knickbedingung (Bild 2.2b):

$$P = 1$$

$$\text{optimale Lösung: } x^* = 1,551 \quad A^* = 0,816 \quad f^* = 4,130$$

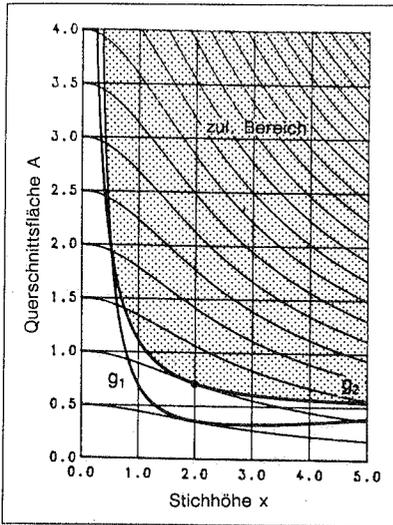
beide Nebenbedingungen sind aktiv

Fall c) Verschiebungs- und Fließbedingung (Bild 2.2c):

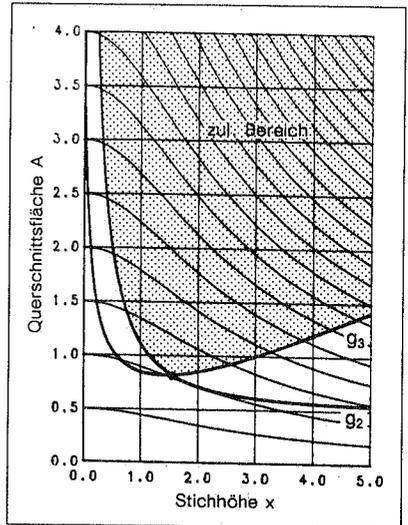
$$P = 1 ; u_{\max} = 0,4$$

$$\text{optimale Lösung: } x^* = 2,000 \quad A^* = 0,707 \quad f^* = 4,000$$

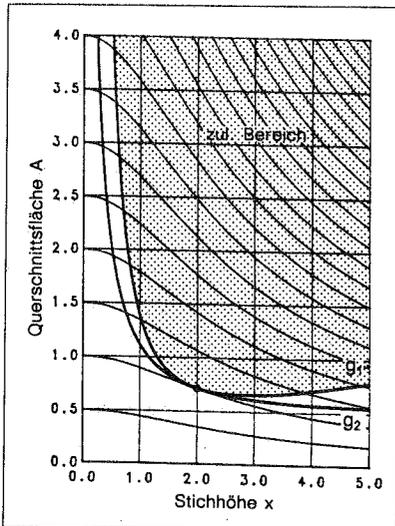
beide Nebenbedingungen sind aktiv; die Qualifikation der Nebenbedingungen ist allerdings nicht erfüllt (siehe Abschnitt 2.7.3)



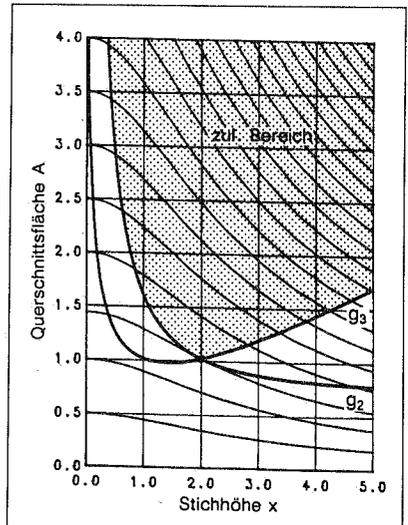
**Bild 2.2a:** Verschiebungsbedingung  $g_1$ ,  
Fließbedingung  $g_2$ , Fall a)



**Bild 2.2b:** Fließbedingung  $g_2$ ,  
Knickbedingung  $g_3$ , Fall b)



**Bild 2.2c:** Nichterfüllte Qualifikation,  
Fall c)



**Bild 2.2d:** redundante Nebenbedingung,  
Fall d)

**Bild 2.2:** Form- und Querschnittsoptimierung des zweistöbigen Fachwerks

Fall d) Fließ- und Knickbedingung (Bild 2.2d):

$$P = 1,441$$

$$\text{optimale Lösung: } x^* = 2,000 \quad A^* = 1,019 \quad f^* = 5,762$$

beide Nebenbedingungen sind aktiv; die Knickbedingung ist redundant  
(siehe Abschnitt 2.7.3)

Im Bild 2.2 sind alle Fälle des Beispiels graphisch verdeutlicht. Dargestellt sind Ausschnitte aus dem Entwurfsraum  $\mathbf{R}^2$  mit der Formvariablen  $x$  als Abszisse und der Querschnittsvariablen  $A$  als Ordinate. Die Probleme sind zweidimensional. In der dritten Dimension werden die Werte der Zielfunktion als Funktion der beiden Optimierungsvariablen dargestellt, die hier in der Projektion als Höhenlinien sichtbar werden. Im Gegensatz dazu interessiert von den Nebenbedingungen, die wie die Zielfunktion Flächen im dreidimensionalen Raum als Funktion der Variablen darstellen, nur die Höhenlinie mit der Kote 0, der charakteristische Wert, an dem sich gemäß der gewählten Definition der zulässige vom unzulässigen Bereich trennt. Der zulässige Bereich wird in den Fällen a) und c) im oberen Teil der Darstellungen von der Verschiebungsnebenbedingung  $g_1$  und im rechten Teil von der Spannungsnebenbedingung  $g_2$  begrenzt. In den Fällen b) und d) wird er im oberen Teil von der Fließbedingung  $g_2$  und im rechten Teil der Darstellung von der Knickbedingung  $g_3$  begrenzt. Er ist durch eine Schraffur hervorgehoben. Die Restriktionen werden erst für unrealistische Variablenkombinationen aktiv. Das Optimum findet sich an denjenigen Stelle im zulässigen Bereich, an dem die Zielfunktion am kleinsten ist.

Im Fall a) ist an der Lösung nur eine einzige Nebenbedingung aktiv, das Optimum befindet sich am Rand des zulässigen Bereich. Dagegen sind im Fall b) beide Nebenbedingungen aktiv, und das Optimum befindet sich an einer Ecke des zulässigen Bereichs. Ebenso, wenngleich nicht in diesem Beispiel, könnte sich die optimale Lösung mitten im zulässigen Bereich befinden, was bedeutet, daß am Optimum keine einzige Nebenbedingung aktiv wäre. In diesem Fall würde es sich um ein **unbeschränktes Optimum** handeln, im Gegensatz zu den **beschränkten** Optima der dargestellten Fälle. Ein guter Algorithmus der mathematischen Programmierung, der für nichtlineare Probleme geeignet sein soll, muß in der Lage sein, in allen genannten Fällen das Optimum zuverlässig und effizient aufzufinden.

Dieses Beispiel eignet sich aber auch dafür, um kurz auf einige spezielle Techniken und Eigenarten der Strukturoptimierung einzugehen, die jedoch nichts mit den allgemeingültigen Begriffen und Lösungsmethoden der mathematischen Programmierung, vielmehr mit einer eleganteren oder effizienteren Formulierung der Funktionen und Bedingungen im Sinne der definierten Standardform (NLP) zu tun haben, um so die Effizienz

der gesamten Lösungsprozedur zu erhöhen. Zum Beispiel wurde hier bereits mit der Gleichgewichtsbedingung die unbekannte Verschiebung eliminiert, wie dies in der Strukturoptimierung üblich ist, um die Größe des Problems zu reduzieren. Verzichtet man auf die Elimination, erhält man drei unabhängige Variablen  $x$ ,  $A$  und  $u$ , und die Verschiebungsnebenbedingung, die soeben noch eine hochgradig nichtlineare Funktion zweier Unbekannter war, wird zu einer simplen Restriktion für die Variable  $u$ .

Weiter zeigt dieses kleine Beispiel auch schon die charakteristischen Eigenschaften von Problemen der Strukturoptimierung. Zielfunktion und Nebenbedingungen sind nichtlineare Funktionen der Optimierungsvariablen, wobei für Querschnittsgrößen und Formgrößen unterschieden werden kann. Querschnittsvariablen, wie in diesem Beispiel die Querschnittsfläche  $A$ , finden sich in der Zielfunktion meist linear und in den Nebenbedingungen als Potenzen ihrer reziproken Werte wieder. Dies bedeutet, daß Spannungs- und Verschiebungsnebenbedingungen für Fachwerke charakterisiert sind durch  $1/A$ , während bei biegebeanspruchten Tragwerken das Widerstandsmoment im Nenner erscheint. Diese typischen Eigenschaften dieser beiden Klassen von Nebenbedingungen können z. B. durch eine einfache Variablentransformation wie  $y = 1/A$  besonders berücksichtigt werden, um das Problem möglichst linear zu gestalten. Dies wird im Kapitel 3 "Approximationstechniken" näher erläutert.

### 2.3 Problemklassen der mathematischen Programmierung und Einordnung von Problemen der Strukturoptimierung

Zum ersten Mal scheint der Begriff "mathematische Programmierung" 1959 aufzutreten [88]. Damit wurde eine damals neue Disziplin der angewandten Mathematik begründet, die sich daraus ergab, daß die bis dato selbständigen Teildisziplinen der mathematischen Programmierung theoretische Grundlagen erarbeitet hatten, die allen gemein sind. Den Anfang der Entwicklung stellte die **lineare Programmierung** dar, die auf Dantzig (1949) zurückgeht. Das von Dantzig entwickelte Simplex-Verfahren zur Lösung linearer Optimierungsprobleme ist bis heute in seiner Robustheit und Effizienz ungeschlagen und ist auch die Grundlage für die Entwicklung von vielen Verfahren zur Lösung nichtlinearer Aufgaben. Die theoretischen Grundlagen zur Lösung nichtlinearer Optimierungsprobleme wurden von Kuhn und Tucker 1951 erarbeitet. Die sogenannten "Kuhn-Tucker"-Bedingungen, die notwendigen Bedingungen für eine Optimalstelle, reichen jedoch über die Grenzen der **nichtlinearen Programmierung** hinaus und gehören zum fundamentalen theoretischen Rüstzeug der gesamten mathematischen Programmierung. Probleme mit Variablen, die nur diskrete oder ganzzahlige Werte einnehmen können, werden von Verfahren der **ganzzahligen Programmierung** erfaßt. Dieses Gebiet der mathematischen Programmierung geht auf Gomory (1958) zurück. Es ist aufgrund der theoretischen Schwierigkeiten und der praktischen Bedeutung immer noch ein sehr

aktives Feld der Forschung. Die **dynamische Programmierung** unterscheidet sich von den anderen drei Disziplinen dadurch, daß sie nicht wie diese Algorithmen zur Verfügung stellt, sondern vielmehr ein allgemeines Prinzip zur sukzessiven Lösung vieler linearer oder nichtlinearer Probleme darstellt. Dieser Vorgang der Dekomposition und der schrittweisen Abfolge von Optimierungen von Untersystemen kann selbstverständlich nur auf solche Probleme angewendet werden, die sich auch in Teilprobleme zerlegen lassen. Insbesondere trifft dies auf zeitabhängige Aufgaben zu, wovon diese Disziplin auch ihren Namen hat. Die grundlegenden Arbeiten zu einer Theorie der dynamischen Programmierung werden Bellman (1957) zugeschrieben. Wie bei der dynamischen Programmierung wird bei der **Mehrkriterienoptimierung** ein Prinzip zur Lösung von Problemen mit mehreren, sich auch widersprechenden Zielfunktionen angegeben. Erste Theorien gehen auf Pareto (1896) zurück. Weitere Gebiete der mathematischen Programmierung sind die Entwicklung spezieller **Verfahren für große Probleme**, die sich an der besonderen Problemstruktur orientieren müssen, damit der Lösungsprozeß noch effektiv genug ist, und die Optimierung von **Problemen der Funktionalanalysis**. In diese beiden Kategorien lassen sich mit Einschränkungen die in der Strukturoptimierung bekannten Verfahren der Optimalitätskriterien einstufen. Eine Zusammenfassung dieser Klassifizierung ist in Tafel 2.1 angegeben.

• Lineare Programmierung	(Dantzig, 1949)
• Nichtlineare Programmierung für beschränkte und unbeschränkte Probleme, stetig differenzierbar oder nicht	(Kuhn und Tucker, 1951)
• Diskrete und ganzzahlige Programmierung	(Gomory, 1958)
• Dynamische Programmierung	(Bellmann, 1957)
• Mehrkriterienoptimierung	(Pareto, 1896)
• Programmierung für große Probleme der beiden ersten Klassen	(z.B. Lasdon, 1970)
• Optimierung von Funktionalen	(z.B. Luenberger, 1969)

**Tafel 2.1:** *Problemklassen (nach Minoux [88])*

Die genannten Klassen der mathematischen Programmierung decken fast ausnahmslos alle in der Strukturoptimierung denkbaren Fälle ab. Die lineare Programmierung kann vorteilhaft bei der Gewichtsoptimierung von Tragwerken eingesetzt werden, wobei ein ideal plastisches Materialverhalten angenommen wird. Auf diesem Gebiet hat die Gruppe um W. Prager [104] viel Pionierarbeit geleistet. Ebenfalls kann die lineare Programmierung zur Bestimmung der Traglast mittels der Fließgelenk- oder Fließlinientheorie eingesetzt werden.

Der weitaus größte Teil der Aufgaben der Strukturoptimierung kann mit den Verfahren der nichtlinearen Programmierung bearbeitet werden. Die ersten Anwendungen gehen auf Schmit [122] zurück. Nach einer Durststrecke in den siebziger Jahren setzte Anfang der 80iger Jahre ein wahrer Boom ein, wie viele Fachtagungen sowie eine Fülle von Artikeln und theoretischen wie anwendungsbezogenen Lehrbüchern beweisen, z.B. [1], [2], [9], [34], [36], [52], [60], [66], [81], [77], [79], [85], [88], [90], [91], [94], [112], [118], [144]. Insbesondere sind die Verfahren der nichtlinearen Programmierung für Probleme der Querschnitts- und Formoptimierung in Verbindung mit der Methode der finiten Elemente geeignet ([31], [35], [62], [71], [111], [122], [139], [143], [154]). In diesem Gebiet ist auch die vorliegende Arbeit angesiedelt, weshalb im folgenden besonders auf diese speziellen Verfahren eingegangen wird.

Bei der Modellierung von Aufgaben der Strukturoptimierung wird gewöhnlich die Tatsache übergangen, daß viele der freien Optimierungsparameter, besonders die Querschnittswerte, nur diskrete Werte einnehmen können. In den meisten Fällen läßt sich dieses Problem unter praktischen Gesichtspunkten sehen, wie auch in dieser Arbeit, in der es vor allem um die Formoptimierung von Flächentragwerken geht, deren Form ohnehin kontinuierlich veränderbar sein soll. Wobei der Gestaltung von Betonschalen mit kontinuierlich veränderlicher Dicke auch in der praktischen Durchführung nicht viel im Wege steht, wie die Schalen von H. Isler beweisen [110]. Dagegen werden Verfahren der diskreten Optimierung zur Querschnittsoptimierung von Stahlrahmentragwerken herangezogen (z.B. Grierson [56]), die aufgrund der zur Auswahl stehenden Profile aus Standardstahllisten ein zwar alltägliches Problem der Bemessung, jedoch ein mathematisch äußerst anspruchsvolles Optimierungsproblem darstellen. Diese Verfahren haben sich, von Ausnahmen abgesehen, noch nicht durchsetzen können.

Die Idee der dynamischen Programmierung, das eigentliche Problem in Unterprobleme aufzuteilen und diese dann einzeln zu optimieren, fand in Form von Dekompositionsverfahren Eingang in die Strukturoptimierung. Ein profiliertes Vertreter dieser Richtung ist Sobieszczyński-Sobieski [128]. Eine aktuelle Arbeit zu diesem Thema wurde von Bremicker [25] vorgelegt.

Grundlegende Arbeiten zur Anwendung der Mehrkriterienmethoden im Ingenieurwesen stammen von Stadler [129], [130]. Anwendungen sind u.a. von Baier [3] und der Gruppe um Eschenauer (z.B. Eschenauer, Schäfer, Bernau in [36]) bekannt. Osyczka [94] veröffentlichte ein Lehrbuch zum Thema.

Gewöhnlich werden unter dem Begriff der mathematischen Programmierung Verfahren zur Lösung von Parameter-Problemen verstanden. Minoux [88] weicht jedoch davon ab und bringt in seinem Buch über Theorie und Algorithmen der mathematischen Programmierung auch einen Abschnitt über die Optimierung von Problemen, die in Form

von Funktionalen vorliegen. Den Kuhn-Tucker-Bedingungen entspricht hier das Maximumsprinzip von Pontryagin (siehe z.B. in [88]). Der übliche Weg, diesen Ansatz der Strukturoptimierung verfügbar zu machen, ist, analytisch ein sogenanntes Optimalitätskriterium herzuleiten und dies dann in einem numerischen Prozeß bei der optimalen Querschnitts- bzw. Formfindung einzusetzen. Dieses Vorgehen ist stets auf spezielle Probleme abgestimmt, womit aber auch große und größte Optimierungsaufgaben gelöst werden können. Da der numerischen Lösung zuerst eine Ableitung des Optimalitätskriteriums vorausgeht, wird dieses Vorgehen auch oft als indirekt im Vergleich zu dem der linearen und nichtlinearen Programmierung bezeichnet, die dagegen als direkte Verfahren gelten. Die Tatsache, daß Optimalitätskriterien oftmals intuitiv entwickelt werden, läßt sie als eine theoretisch schlecht untermauerte Alternative zu den direkten Verfahren erscheinen und lenkt von den Gemeinsamkeiten ab. Bekannte Vertreter der Optimalitätskriterienmethoden sind Berke, Venkayya und Khot [10], [147], einen guten Überblick geben u.a. Save und Prager [116]. Einen wichtigen Beitrag zu den Gemeinsamkeiten der direkten und indirekten Methoden lieferte Fleury [42]. In Kapitel 3 wird darauf näher eingegangen.

klein	mittel	groß
$n \leq 5$ $m \leq 5$ n ... Zahl der Variablen	$5 \leq n \leq 100$	$n \geq 100$ ./ 1000 $m \geq k \cdot 1000$ m ... Zahl der Nebenbedingungen
Handmethoden	Computerverfahren	große Computer
Lösung der notwendigen Bedingungen	"direktes" Vorgehen, allgemeine Verfahren der nichtlinearen Programmierung	problemorientierte Verfahren

Tafel 2.2: Problemgrößen (nach Luenberger [85])

#### 2.4 Zur Größe von Problemen der mathematischen Programmierung

Die Charakterisierung eines Problems der mathematischen Programmierung anhand seiner Größe kann nur sehr grob vorgenommen werden. Sie hängt vom Problem, von den aktuell zur Verfügung stehenden Methoden, der apparativen Ausstattung (Taschenrechner, PC, Mainframe, Supercomputer u.s.w.) und zuletzt vom subjektiven Standpunkt des einzelnen ab. Um dennoch eine grobe Abschätzung zu wagen, sei die Klassifizierung

von Luenberger [85] übernommen (Tafel 2.2), Demnach sind die üblichen Probleme der Strukturoptimierung in die mittlere Kategorie einzustufen. Dabei sind bei den bisherigen Anwendungen die Probleme der Querschnittsoptimierung gewöhnlich größer als die der Formoptimierung, manchmal reichen sie auch bis in die Kategorie "groß". Dann werden sie meistens mit Verfahren der Optimalitätskriterien gelöst. Daß jedoch Probleme der Formoptimierung oft nicht mehr als 20, höchstens 30 oder gar deutlich weniger als 10 Variable haben und damit schon zur Klasse der kleinen Probleme gehören, hat zwei Gründe. Zum einen sind diese Probleme meist stark nichtlinear, was den Lösungsaufwand stark anwachsen läßt, zum anderen – und das ist der wichtigere Grund – ist es mitunter sehr schwierig, die Auswirkungen der vielen Freiheitsgrade so abzuschätzen, daß während der Optimierung keine physikalisch unmöglichen Formen entstehen, oder solche, die vom verwendeten statischen Analyseverfahren (z.B. finite Element-Methode) nicht mehr bearbeitet werden können. Im Rahmen dieser Arbeit werden vorwiegend Probleme der Kategorien "klein" und "mittel" behandelt.

## 2.5 Methoden der nichtlinearen Programmierung

Wie im Abschnitt 2.3 festgestellt wurde, können die meisten Probleme der Strukturoptimierung mit den Verfahren der nichtlinearen Programmierung angegangen werden. Da auch im Rahmen dieser Arbeit auf Verfahren dieser Klasse zurückgegriffen wird, soll an dieser Stelle ein Überblick gegeben und eine weitere Differenzierung vorgenommen werden, ohne jedoch zu sehr auf die Details der verschiedenen Methoden einzugehen. Dies und die Darstellung der wichtigsten theoretischen Grundlagen sei den folgenden Abschnitten vorbehalten. Eine Zusammenfassung dieses Abschnitts findet man in den Tafeln 2.3 und 2.4.

Die offensichtlichste Unterscheidung ist die in Methoden für unbeschränkte und für beschränkte Probleme. Dabei ist festzustellen, daß fast alle Verfahren für Aufgaben mit Nebenbedingungen in irgendeiner Weise auf Methoden der unbeschränkten Art zurückgreifen oder ihre Eigenschaften teilweise übertragbar sind. Deshalb ist es nicht möglich, die Verfahren für beschränkte Aufgaben unabhängig von denjenigen für unbeschränkte Fälle zu betrachten.

Die Methoden für unbeschränkte Optimierungsprobleme teilen sich auf in Verfahren für ein- und mehrdimensionale Aufgaben. Die eindimensionalen Verfahren haben eine große Bedeutung, da sie ein wichtiger Bestandteil vieler mehrdimensionaler Verfahren sind. Vielfach ist es so, daß erst durch die eindimensionalen "line searches" die Konvergenz der mehrdimensionalen Verfahren gesichert werden kann. Man unterscheidet hier Intervallschachtelung, Interpolationsverfahren, Regula-Falsi- und Newton-Verfahren, die oft auch in Kombinationen angewendet werden, je nachdem, wie groß die Anforder-

rungen an die Genauigkeit der eindimensionalen Suche sind. (Im weiteren wird der Begriff "line search" verwendet, der sich auch im einschlägigen deutschen Sprachgebrauch durchgesetzt hat.)

Eine Klassifikation der mehrdimensionalen Verfahren für unbeschränkte Verfahren orientiert sich in sinnvoller Weise an der Art der zur Verfügung stehenden Informationen über das Wesen der zu lösenden Aufgabe. Hier unterscheidet man in Verfahren nullter, erster und zweiter Ordnung, auch direkte Suchmethoden, Gradienten- und Newton-Verfahren genannt. Das Gemeinsame aller Verfahren ist, daß sie iterativ vorgehen. In jedem Iterationsschritt sind zwei Fragen zu beantworten: In welcher Richtung soll weitergesucht werden, so daß die Zielfunktion geringer wird, und wie weit soll in dieser Richtung fortgeschritten werden?

Die Verfahren zweiter Ordnung bedienen sich Informationen über die Zielfunktion selbst, deren erster und zweiter Ableitungen. Neben dem bekannten Newton-Raphson-Verfahren wäre noch die Methode der konjugierten Gradienten in der Formulierung von Hestenes und Stiefel zu nennen, die gegenüber der Newton-Methode Vorteile bei der Lösung großer Probleme bietet, da bei diesem Verfahren die Invertierung bzw. Faktorisierung der Hesse-Matrix (die Matrix der zweiten Ableitungen) entfällt.

Dagegen verzichten Verfahren erster Ordnung ganz (Gradientenverfahren) oder wenigstens auf die explizite Bestimmung der zweiten Ableitungen (z.B. konjugierte Gradientenverfahren, Quasi-Newton-Verfahren). Die konsequente Umsetzung in eine Methode, die sich streng und ausschließlich an den gerade vorhandenen Funktionswerten und ersten Ableitungen orientiert, ist die Methode des steilsten Abstiegs. Dieses Verfahren sucht immer in der Fallrichtung der Zielfunktion und garantiert, daß in jedem Iterationsschritt die Lösung wenigstens lokal verbessert wird. Es wird deshalb oft mit anderen Verfahren kombiniert, die diese Eigenschaft nicht aufweisen. Als eigenständiges Verfahren jedoch neigt die Methode des steilsten Abstiegs fern der Lösung infolge zu kurzer Schrittweiten steckenzubleiben, da sie sich nicht an den zweiten Ableitungen des Problems orientieren kann. Die Methoden der konjugierten Gradienten nach Fletcher/Reeves, Polak/Ribière oder die stark an Bedeutung gewonnenen Quasi-Newton-Verfahren (z.B. DFP-, BFGS- oder Rang 1 -Korrektur) überwinden dieses Defizit und bauen sich im Laufe der Iterationen Informationen über die zweiten Ableitungen auf. Wie auch bei den Verfahren zweiter Ordnung sollten bei großen Problemen konjugierte Gradientenverfahren den Quasi-Newton-Verfahren vorgezogen werden. Ein ganz wichtiger Bestandteil der Verfahren erster Ordnung, bei den Gradientenverfahren noch mehr als bei den Quasi-Newton-Verfahren, ist der line search. Nur mit seiner Hilfe kann die Konvergenz garantiert werden.

- **1-dimensionale Probleme**

- Intervallschachtelung
 

Fibonacci
Goldener Schnitt
Bisektion
- Polynominterpolation
- Regula-Falsi
- Newton

- **Mehrdimensionale Probleme**

- direkte Suchmethoden:

- Rastersuche
- Koordinatenstrategie
 

Gauss/Seidel
--------------
- rotierende Koordinaten
 

Rosenbrock	1960
------------	------
- Simplex-Verfahren
 

Nelder/Mead	1965
-------------	------
- ableitungsfreies Verfahren der konj. Richtungen von Powell
 

	1964
--	------

- Verfahren erster Ordnung:

- steilster Abstieg
- Methoden der konjugierten Gradienten:
 

Fletcher/Reeves	1964
Polak/Ribière	1964
- Methode der parallelen Tangenten:
 

Shah, Buehler,	
Kemphthorne	1964
- Quasi-Newton:
 

Methode der variablen Metrik:	Davidon	1959
	Fletcher, Powell	1963
Rang 1-update:	Davidon	1968
	Broyden	1967
BFGS update:	Broyden	1967
	Fletcher	1970
	Goldfarb	1970
	Shanno	1970

- Verfahren zweiter Ordnung:

- Newton-Verfahren
- Methode der konjugierten Gradienten
 

Hestenes, Stiefel	1952
-------------------	------

**Tafel 2.3:** *Methoden für unbeschränkte stetige Funktionen*  
 (nach Luenberger [85], Gill et al. [52], Minoux [88], Schwefel [123])

Die Verfahren nullter Ordnung kommen völlig ohne Gradienten und Krümmungsbestimmungen aus, weshalb sie auch direkte Suchmethoden genannt werden. Man kann sie unter den Veteranen der Methoden der nichtlinearen Programmierung einstufen, denn heute haben sie ihre Berechtigung nur noch für die Lösung von hochgradig diskontinuierlichen Problemen bewahrt, an denen die höherwertigen Verfahren scheitern müssen. Das Verfahren von Powell ist insofern erwähnenswert, da es allein aus Funktionsauswertungen konjugierte Richtungen bestimmt. Dies entspricht dem Verfahren der konjugierten Gradienten, wobei die Gradienten aus Differenzenquotienten ermittelt werden. Ebenso bedient sich dieses Verfahren eines line searches. Es kommt deshalb auch nur für genügend glatte Funktionen in Frage.

Literatur zu den einzelnen Verfahren findet man in vielen Lehrbüchern, z.B. die von Luenberger [85], Gill, Murray, Wright [52], Minoux [88], Künzi, Krelle, v. Randow [81] und viele andere mehr, die gerade innerhalb des letzten Jahrzehnts erschienen sind. Eine gute Übersicht über direkte Suchmethoden gibt Schwefel [123].

Die Methoden für beschränkte nichtlineare Optimierungsaufgaben lassen sich in vier Klassen einteilen, die jede für sich einen jeweils anderen Aspekt der zugrunde liegenden gemeinsamen Theorie der nichtlinearen Programmierung verfolgt.

Die primären Methoden arbeiten im zulässigen Bereich des Problems, d.h. bei einem Problem mit  $n$  Unbekannten und  $m$  Gleichheitsbedingungen in einem Raum der Dimension  $(n - m)$ . Der Vorteil der Verfahren, gerade auch für Anwendungen in der Strukturoptimierung, ist, daß sie den zulässigen Bereich nie verlassen. Das heißt, auch bei vorzeitigem Abbruch der Iteration ist die Lösung brauchbar und praktisch verwertbar. Eine Schwierigkeit ist dabei jedoch, daß auch der Startpunkt der Iteration, das entspricht dem Anfangsentwurf der Strukturoptimierung, innerhalb des zulässigen Bereiches liegen muß. Eine Forderung, die nicht immer leicht zu erfüllen ist. Die primären Methoden sind Gradientenmethoden, wobei die Erfüllung der Nebenbedingungen durch den line search garantiert wird, der die Grenze des zulässigen Bereichs in der jeweils eingeschlagenen Suchrichtung auszuloten hat. Bekannte Verfahren sind die Methode der zulässigen Richtungen, die Verfahren der projizierten und der reduzierten Gradienten. Eine erfolgreiche, kommerzielle Version der Methode der zulässigen Richtungen namens CONMIN (constrained minimization) stammt von Vanderplaats [142]. Dieses Verfahren zählt zu den am meisten zitierten Verfahren und wurde schon in einem weiten Bereich der Strukturoptimierung eingesetzt (z.B. Bushnell [26] oder Botkin et al. in [9]), wenngleich auch weniger für Aufgaben der Formoptimierung.

In der Anwendung sehr einfache Verfahren sind die Penalty- und Barriere-Methoden. Hier wird das beschränkte Problem durch ein unbeschränktes approximiert, das dann mit einem dafür geeigneten Verfahren gelöst werden kann. Diese Verfahren arbeiten

im  $n$ -dimensionalen Raum der Unbekannten. Der Nachteil der Verfahren ist, daß die exakte Lösung nicht erreicht werden kann, wobei sich die Penalty-Verfahren von unzulässigen und die Barriere-Methoden aus dem zulässigen Bereich her nähern. Vom Standpunkt der Strukturoptimierung aus gesehen, kommen also nur Barriere-Methoden in Betracht. Deren spezielle Einschränkungen sind, daß auch hier der Startpunkt im zulässigen Bereich liegen muß und daß keine Gleichheitsbedingungen erfaßt werden können. Diese Einschränkungen sind aber für viele praktische Anwendungen ohne Bedeutung. Eine spezielle Form der Barriere-Methoden, das SUMT-Verfahren von Fiacco/McCormick (Sequential Unconstrained Minimization Technique, siehe z.B. in [1]) ist in der Strukturoptimierung weit verbreitet [59], [27] und findet auch in kommerziellen Programmpaketen wie z.B. ANSYS [30] seine Anwendungen. Eine interessante Barriere-Methode ist die Evolutionsstrategie von Schwefel [123], die zu den direkten Suchverfahren zählt und sich Zufallsentscheidungen zunutze macht. Ihre Vorteile liegen bei diskontinuierlichen Problemstellungen, die von den anderen Verfahren nicht mehr gelöst werden können. Die Nachteile sind, daß das Verfahren nur für wenige Variablen einsetzbar ist und daß es sehr langsam konvergiert. Einige Anwendungen im Bereich der Strukturoptimierung stammen von Hartmann [65], Lawo/Thierauf [82] sowie Höfler [70] und [11].

Die dualen Methoden gehen das Problem indirekt an, indem sie das sogenannte duale Problem lösen, dessen Unbekannte die Lagrange-Multiplikatoren sind. Deren Anzahl entspricht derjenigen der Nebenbedingungen (näheres dazu siehe Abschnitt 2.7.1.2 und 2.8.5). Damit arbeiten diese Methoden in einem  $m$ -dimensionalen Raum. Die eigentlich gesuchten Werte der Optimierungsvariablen ergeben sich aus einer Rückrechnung. Rechentechisch gesehen, zerfällt bei den dualen Methoden das eigentliche Problem in zwei Teilprobleme, wovon das eine unbeschränkt ist und das andere im Falle von Ungleichheitsbedingungen nur von einfachen Variablenbeschränkungen begrenzt wird. Im Falle von ausschließlich Gleichheitsbedingungen ist es ebenfalls unbeschränkt. Diese Teilprobleme sind mit den üblichen Methoden für unbeschränkte Aufgaben direkt oder mit den für die Berücksichtigung der Schranken einfachen Erweiterungen lösbar. Besondere Beachtung in der Strukturoptimierung fanden in letzter Zeit die Verfahren von Fleury [44] und Svanberg [135], die die Ideen der dualen Methoden mit Approximationsmethoden und den Vorteilen separabler Probleme in überzeugender Weise zu vereinen wissen. Auf diese Weise konnten effiziente und auch für große Probleme geeignete Verfahren der Strukturoptimierung geschaffen werden. Interessant sind die Parallelen zwischen diesem Vorgehen und den Optimalitätskriterienmethoden [42]. In den Abschnitten 2.8.5.2 und 3.2 wird näher auf die zugrunde liegenden Theorien eingegangen.

Die letzte Gruppe der Verfahren für beschränkte Probleme ist die Gruppe der Lagrange-Methoden. Sie streben eine direkte Lösung der notwendigen Bedingungen für Optimalität an (Kuhn-Tucker-Bedingungen, Abschnitt 2.7.1.2) und kombinieren die primären und dualen Eigenschaften der vorhin genannten Methoden. Demnach arbeiten sie im  $(n + m)$ -dimensionalen Raum der Optimierungsvariablen und Lagrange-Multiplikatoren, welche auch duale Variablen genannt werden. Die sich ergebenden Gleichungssysteme können iterativ mit Newton- oder Quasi-Newton-Verfahren gelöst werden. Wichtiger Bestandteil dieser Verfahren ist die Lösung eines Unterproblems mit quadratischer Zielfunktion und linearen Nebenbedingungen, ein Problem der sogenannten quadratischen Programmierung. Deshalb werden diese Verfahren auch SQP (sequentiell quadratische Programmierung) oder RQP (rekursive quadratische Programmierung) genannt. Eine bekannte SQP-Methode, die zu den Quasi-Newton-Verfahren zu zählen ist, ist die Methoden von Han [64] und Powell [100]. Obwohl die SQP-Verfahren von vielen Autoren als die leistungsfähigsten bezeichnet werden [8], [137], haben sie bis jetzt noch wenig Verbreitung in der Strukturoptimierung gefunden [35], was auch auf die zugrunde liegende komplexe Theorie zurückzuführen ist.

Als eine besondere Form von Lagrange-Methoden kann die sogenannte sequentielle lineare Programmierung (SLP) verstanden werden. Sie arbeitet ebenfalls im  $(n + m)$ -dimensionalen Raum der primären und dualen Variablen. Im Gegensatz zu den SQP-Methoden wird allerdings ein lineares Unterproblem mit dem dazu geeigneten und bekannten Simplex-Verfahren der linearen Programmierung gelöst. Damit ist die SLP ein äußerst robustes und einfach zu handhabende Verfahren, hat aber den Nachteil, daß die linearen Unterprobleme unbeschränkt sein können. Dies muß durch geeignete Maßnahmen berücksichtigt werden, was sich aber nachteilig auf die Konvergenzeigenschaften auswirkt. Dennoch ist die sequentielle lineare Programmierung ein Verfahren, das in der Vergangenheit viele Anwendungen in der Strukturoptimierung gefunden hat (z.B. [3], [78], [97], [115], [157]), besonders bei Querschnittsoptimierungen.

Verschiedentlich wird versucht, die Lagrange-Methoden mit anderen (insbesondere primären Methoden) zu kombinieren, um die Vorteile beider Verfahren auszunützen [35], [138]. Die so entstehenden Verfahren werden oft als hybride Methoden bezeichnet.

Die hier gewählte Einteilung in primäre Methoden, Penalty- und Barriere-Methoden, duale Methoden und Lagrange-Methoden orientiert sich an Luenberger [85]. Auch andere Autoren [52], [81] nehmen diese Klassifikation vor, wengleich mit anderer Terminologie. Vor allem sei auf Minoux [88] hingewiesen, der die dualen Methoden "Lagrange-Methoden" nennt und die Lagrange-Methoden "primär dual".

● Lineare Programmierung:	Simplex	(Dantzig, 1949)
● Primäre Methoden:	Methode der zulässigen Richtungen	(Zoutendijk, 1960)
	projizierte Gradienten	(Rosen, 1960)
	reduzierte Gradienten	(Wolfe, 1967)
	Complex-Verfahren	(Abadie, Carpentier, 1969)
		(Box, 1965)
● Penalty- und Barriere-Methoden:	Penalty-Methode	(Courant, 1943)
	SUMT	(Fiacco, McCormick, 1968)
	Barriere-Methode	(Carroll, 1961)
	Evolutionstrategie	(Schwefel, 1977)
● Duale Methoden:	Methode von Uzawa	(1958)
	Methode von Arrow-Hurwicz	(1958)
	Multiplikatorenmethode	(Hestenes, 1969)
		(Powell, 1969)
	Erweiterung für Ungleichheitsbedingungen	(Rockafellar, 1973)
	Separable Probleme	(Everett, 1963)
	Quadratische Programmierung	(Dantzig, 1963)
● Lagrange-Methoden:	SLP	
	Rekursive quadratische Programmierung	(Wilson, 1963)
		(Garcia-Palomares, Mangasarian, 1976)
	SQP (mit Quasi-Newton)	(Han, 1977)
		(Powell, 1978)

Tafel 2.4: Methoden für beschränkte stetige Funktionen

## 2.6 Zur Auswahl einer Methode

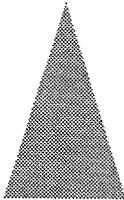
Bei der Auswahl einer geeigneten Methode zur Lösung eines vorliegenden Problems der Strukturoptimierung lohnt es sich, Aufwand und Nutzen gegeneinander zu stellen. Dabei sollte unter Aufwand neben der reinen Rechenzeit auch der Aufwand verstanden werden, der getrieben werden muß, um die gewählte Methode richtig zu verstehen und anwenden zu können. Auch ist es angezeigt, ein Verfahren zu wählen, das dem zu lösenden Problem gerecht werden kann. In Tafel 2.5 sind einige charakteristische Merkmale von Zielfunktionen und Nebenbedingungen aufgelistet, die in verschiedenen Kombinationen zu unterschiedlichen Entscheidungen führen können. Auf jeden Fall ist es jedoch unsinnig, zugunsten einer Verringerung der Zahl von Variablen oder Nebenbedingungen eine vorhandene günstige Problemstruktur aufzugeben. Das geht meist zu Lasten der Konvergenzgeschwindigkeit und -sicherheit. Sind die Funktionen der Aufgabe stetig ableitbar oder nur wenig diskontinuierlich, sollte die Wahl stets für Gradienten- oder gar Newton-Verfahren ausfallen. Wie aus Tafel 2.6 ersichtlich, ist für diese Verfahren die Konvergenzgeschwindigkeit am größten, d.h. die Konvergenzrate am kleinsten. Dagegen steigt dabei der Aufwand unter Umständen sogar gewaltig an. Besonders für die Strukturoptimierung ist gerade die Gradientenbestimmung über die Sensitivitätsanalyse der absolut aufwendigste Teil der gesamten Prozedur; das heißt, daß daneben die übrige Rechenzeit für die Methode selbst, Datenmanipulationen usw. vernachlässigbar klein ausfällt. Verschiedentlich werden deshalb direkte Suchmethoden besser beurteilt als die höherwertigen, siehe z.B. die Untersuchung von Eason und Fenton [32], die allerdings inzwischen als überholt gelten kann. Dem gegenüber haben sich auch in der Strukturoptimierung Gradientenverfahren und Quasi-Newton-Methoden durchgesetzt, die gerade in den siebziger Jahren eine gewaltige Entwicklung erfahren haben. Methoden zweiter Ordnung haben bisher noch keinen Eingang in die Strukturoptimierung gefunden, da die Bestimmung der zweiten Ableitungen nach allgemeiner Ansicht den bereits vorhandenen Aufwand potenzieren würde. Jedoch wird auch in dieser Richtung weitergeforscht [45], [63], so daß vielleicht schon bald auch diese Verfahren ihre Leistungskraft für die Strukturoptimierung einsetzen können.

Für Aufgaben der Querschnittsoptimierung wird in der Literatur über gute Erfahrungen mit allen gängigen Verfahren berichtet [8], [27], [137], während für Formoptimierungen der Trend zu theoretisch aufwendigeren Verfahren (z.B. duale Verfahren oder Lagrange-Methoden) gehen dürfte. Allerdings beschränken sich die meisten Anwendungen auf die Optimierung von Detailproblemen (siehe z.B. den Übersichtsartikel von Ding [31]), während Anwendungen auf die Formoptimierung von Freiformflächen und Schalenstrukturen noch weitgehend ausstehen. Deshalb fiel im Rahmen dieser Arbeit die Entscheidung für ein SQP-Verfahren in der Formulierung von Schittkowski [117] sowie

für ein selbstentwickeltes duales Verfahren, das sich an den Arbeiten von Fleury [44] und Svanberg [135] orientiert. Daneben wurde ein SLP-Verfahren entwickelt, das vor allem für die Querschnittsoptimierung eingesetzt wird. Darüber hinaus wurde die Evolutionsstrategie nach Schwefel [123] zur Form- und Querschnittsoptimierung von Fachwerkstrukturen eingesetzt.

Eigenschaften der Zielfunktion	der Nebenbedingungen
Funktion einer einzigen Variablen lineare Funktion Quadratsumme linearer Funktionen quadratische Funktion Quadratsumme nichtlinearer Funktionen kontinuierliche nichtlineare Funktion schwach gekoppelte nichtlineare Funktion nicht-kontinuierliche nichtlineare Funktion	keine Nebenbedingungen einfache Variablenschranken linear schwach gekoppelt nichtlinear kontinuierlich nichtlinear  schwach gekoppelt nichtlinear  nicht-kontinuierlich nichtlinear

Tafel 2.5: Problemstrukturen (nach Gill, Murray, Wright [52])

Aufwand	Methode	Konvergenzrate
	Newton-Verfahren mit analytischen 2. Abl. Newton-Verfahren mit numerischen 2. Abl. Quasi-Newton-Verfahren mit analytischen 1. Abl. Quasi-Newton-Verfahren mit numerischen 1. Abl. konjugierte Gradienten mit analytischen 1. Abl. konjugierte Gradienten mit numerischen 1. Abl. Suchmethoden	

Tafel 2.6: Aufwand gegen Konvergenzrate (nach Gill, Murray, Wright [52])

## 2.7 Grundlagen der mathematischen Programmierung

Die folgenden Ausführungen stellen eine kurze Einführung in bekannte Aussagen der Theorie der mathematischen Programmierung dar. Es sollen auf diese Weise die wichtigsten Begriffe genannt werden, vor deren Hintergrund die anschließende Beschreibung der Algorithmen, deren Eigenschaften und der damit gewonnenen Erfahrungen zu sehen sind. Die Ausführungen lehnen sich an die Standardwerke von Luenberger [85], Minoux [88], Gill, Murray, Wright [52] sowie Künzi, Krelle und v. Randow [81] an, wo auch weitere, hier nicht genannte Einzelheiten gefunden werden können.

### 2.7.1 Notwendige und hinreichende Bedingungen für lokale Minima

#### 2.7.1.1 Unbeschränkte Probleme

Für das Problem

$$f(x) \rightarrow \min ; x \in \mathbb{R}^n$$

lauten die hinreichenden Bedingungen für ein lokales Minimum an  $x^*$ :

$$\nabla f(x^*) = 0$$

$$\nabla^2 f(x^*) \text{ positiv definit} \quad (2.1)$$

Dabei soll  $f(x)$  zweifach stetig differenzierbar sein.  $\nabla^2 f(x)$  ist die Hesse-Matrix der Funktion  $f(x)$ , die Matrix der zweiten Ableitungen. Ist  $f(x)$  konvex, dann ist  $x^*$  das globale Minimum.

#### 2.7.1.2 Beschränkte Probleme

Zunächst soll zur Vereinfachung der Darstellung ein Problem mit linearen Nebenbedingungen und Variablenrestriktionen betrachtet werden. Auch hier wird angenommen, daß alle Funktionen zweifach stetig differenzierbar sind. Also:

$$f(x) \rightarrow \min$$

$$A x - b \leq 0 \quad (2.2)$$

$$x \in S \subset \mathbb{R}^n$$

Weiter sei der Satz J der am Optimum  $x^*$  aktiven Nebenbedingungen bekannt. Diese können dann als Gleichheitsbedingungen behandelt werden, während die übrigen als strikte Ungleichungen keinen Einfluß auf das Ergebnis haben:

$$\hat{A}_j x - \hat{b}_j = 0 ; j \in J \quad (2.3)$$

$$A_j x - b_j < 0 ; j \in K$$

$$J \cup K = \{1, \dots, m\}$$

J ... aktiver Satz

$\hat{x}$  sei ein zulässiger Punkt, der die aktiven Nebenbedingungen erfüllt. Von  $\hat{x}$  aus ist  $\mathbf{p}$  eine zulässige Richtung, wenn gilt:

$$\hat{\mathbf{A}} (\hat{\mathbf{x}} + a \mathbf{p}) \leq \hat{\mathbf{b}}$$

woraus mit (2.3) folgt:

$$\hat{\mathbf{A}} \mathbf{p} \leq \mathbf{0} \quad (2.4)$$

Richtungen  $\mathbf{p}$ , die die strikte Bedingung  $\hat{\mathbf{A}} \mathbf{p} = \mathbf{0}$  erfüllen, liegen im sogenannten **Tangentialraum** der Nebenbedingungen. (Zu einem eigentlichen Tangentialraum wird dieser Raum natürlich erst nach einer Erweiterung dieser Darstellung auf nichtlineare Nebenbedingungen.) Er zeichnet sich dadurch aus, daß die Gradientenvektoren der aktiven Nebenbedingungen, die in  $\hat{\mathbf{A}}$  zusammengefaßt sind, auf diesem senkrecht stehen. Andererseits existiert eine Basis  $\mathbf{Z}$ , die den Tangentialraum aufspannt. Hat das Problem  $n$  Unbekannte und  $m_a$  aktive Nebenbedingungen, dann ist die Dimension des Tangentialraumes  $(n - m_a)$ , d.h.  $\mathbf{Z}$  besteht aus  $(n - m_a)$  linear unabhängigen Basisvektoren. Gemäß der gewählten Definitionen sind  $\hat{\mathbf{A}}$  und  $\mathbf{Z}$  orthogonal, und es gilt:

$$\hat{\mathbf{A}} \mathbf{Z} = \mathbf{0} \quad (2.5)$$

Damit kann die zulässige Richtung  $\mathbf{p}$  als eine Linearkombination der Basisvektoren von  $\mathbf{Z}$  und  $\hat{\mathbf{A}}$  ausgedrückt werden:

$$\mathbf{p} = \mathbf{Z} \cdot \mathbf{p}_Z + \hat{\mathbf{A}}^T \cdot \mathbf{p}_A \quad (2.6)$$

Diese Gleichung in (2.4) eingesetzt, ergibt die Vorschrift für zulässige Richtungen  $\mathbf{p}_A$ , die zurück in den zulässigen Bereich führen:

$$\hat{\mathbf{A}} \mathbf{p} = \hat{\mathbf{A}} \mathbf{Z} \mathbf{p}_Z + \hat{\mathbf{A}} \hat{\mathbf{A}}^T \mathbf{p}_A = \hat{\mathbf{A}} \hat{\mathbf{A}}^T \mathbf{p}_A \leq \mathbf{0} \quad (2.7)$$

Da der am Optimum  $\mathbf{x}^*$  aktive Satz der Nebenbedingungen bereits identifiziert ist, muß an  $\mathbf{x}^*$  gelten:

$$f(\mathbf{x}^* + a \mathbf{Z} \mathbf{p}_Z + \beta \hat{\mathbf{A}}^T \mathbf{p}_A) = f^* ; \quad a = \beta = 0 \quad (2.8)$$

$$\text{und } f^* = \min_{\mathbf{x}} f(\mathbf{x})$$

für beliebige Richtungen  $\mathbf{p}_Z$  und  $\hat{\mathbf{A}} \hat{\mathbf{A}}^T \cdot \mathbf{p}_A \leq \mathbf{0}$ .

Die aktiven Nebenbedingungen sind identisch erfüllt und damit eliminiert, das Problem ist deshalb unbeschränkt für  $\mathbf{p}_Z$ . Die notwendigen Bedingungen für ein Minimum in der Tangentialrichtung  $\mathbf{p}_Z$  sind deshalb:

$$\nabla_{\mathbf{p}_Z} f = \mathbf{Z}^T \nabla_{\mathbf{x}} f = \mathbf{0} \quad (2.9)$$

$$\nabla_{\mathbf{p}_Z}^2 f = \mathbf{Z}^T \nabla_{\mathbf{x}\mathbf{x}}^2 f \mathbf{Z} \quad \text{positiv definit}$$

Der Vergleich mit Gleichung (2.5) führt zur Erkenntnis, daß am Optimum  $\nabla f$  eine lineare Kombination der Nebenbedingungsgradienten sein muß.

$$\begin{aligned} \mathbf{Z}^T \nabla f + \mathbf{Z}^T \hat{\mathbf{A}}^T &= \mathbf{0} \\ \mathbf{Z}^T (\nabla f + \hat{\mathbf{A}}^T \boldsymbol{\lambda}^*) &= \mathbf{0} \\ \nabla f + \hat{\mathbf{A}}^T \boldsymbol{\lambda}^* &= \mathbf{0} \end{aligned} \quad (2.10)$$

$$\boldsymbol{\lambda}^* \geq \mathbf{0} \quad (2.11)$$

Der  $m$ -Vektor  $\boldsymbol{\lambda}$  ist der Vektor der Lagrange-Multiplikatoren. Die zusätzliche Bedingung, daß die Lagrange-Multiplikatoren positiv sein müssen, läßt sich leicht verifizieren. Jede Richtung  $\mathbf{p}$  weg vom Minimum  $\mathbf{x}^*$  muß aufwärts führen, das heißt:

$$\nabla f^T(\mathbf{x}^*) \mathbf{p} \geq 0$$

mit (2.10): 
$$-\boldsymbol{\lambda}^{*T} \hat{\mathbf{A}} \mathbf{p} \geq 0$$

mit (2.6): 
$$-\boldsymbol{\lambda}^{*T} \hat{\mathbf{A}} (\mathbf{Z} \mathbf{p}_Z + \hat{\mathbf{A}}^T \mathbf{p}_A) \geq 0$$

mit (2.5): 
$$-\boldsymbol{\lambda}^{*T} \hat{\mathbf{A}} \hat{\mathbf{A}}^T \mathbf{p}_A \geq 0$$

Was mit den Aussagen der Gleichungen (2.7) und (2.11) auch bestätigt werden kann.

Dieses letzte Ergebnis entspricht auch der Forderung, daß Richtungen  $\mathbf{p}_A$ , die in den zulässigen Bereich zurückführen, aufwärts zeigen müssen:

$$\nabla_{\mathbf{p}_A}^T f(\mathbf{x}^*) \mathbf{p}_A \geq 0$$

Damit lassen sich die notwendigen Bedingungen zusammenfassen:

$$\begin{aligned} \nabla f(\mathbf{x}^*) + \hat{\mathbf{A}}^T \boldsymbol{\lambda}^* &= \mathbf{0} \\ \hat{\mathbf{A}}_j \mathbf{x}^* - \hat{\mathbf{b}}_j &= \mathbf{0} \quad ; \quad j \in J \\ \boldsymbol{\lambda}^*_{j} &\geq \mathbf{0} \end{aligned}$$

Mit der zusätzlichen Bedingung, daß Lagrange-Multiplikatoren für nicht aktive Nebenbedingungen Null sein sollen, lassen sich diese Bedingungen für alle Nebenbedingungen gemeinsam anschreiben:

$$\begin{aligned}\nabla f(\mathbf{x}^*) + \mathbf{A}^T \boldsymbol{\lambda}^* &= \mathbf{0} \\ (\mathbf{A} \mathbf{x}^* - \mathbf{b})^T \boldsymbol{\lambda}^* &= 0 \\ \boldsymbol{\lambda}^* &\geq \mathbf{0}\end{aligned}$$

Diese Bedingungen sind die sogenannten **Kuhn-Tucker-Bedingungen**. Sie formulieren die notwendigen Bedingungen für eine Extremalstelle des linear beschränkten Problems (2.2). Sie bilden die Grundlage für die Theorie der nichtlinearen Programmierung. Die Erweiterung für Probleme mit nichtlinearen Nebenbedingungen entspricht den dargestellten Ableitungen. Anstelle einer geradlinigen zulässigen Richtung  $\mathbf{p}$  wird entlang eines Bogenstücks durch das Minimum vorgegangen. In Erweiterung auf nichtlineare Problemstellungen mit Gleichheits- und Ungleichheitsbedingungen lauten die Kuhn-Tucker-Bedingungen:

$$\begin{aligned}\nabla f(\mathbf{x}^*) + \nabla^T \mathbf{g}(\mathbf{x}^*) \boldsymbol{\lambda}^* + \nabla^T \mathbf{h}(\mathbf{x}^*) \boldsymbol{\mu}^* &= \mathbf{0} \\ \mathbf{g}^T(\mathbf{x}^*) \boldsymbol{\lambda}^* &= 0 \\ \boldsymbol{\lambda}^* &\geq \mathbf{0}\end{aligned}\tag{2.12}$$

mit der zusätzlichen Bedingung, die ein lokales Minimum garantiert:

$$\mathbf{Z}^T(\mathbf{x}^*) \nabla^2 L(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \mathbf{Z}(\mathbf{x}^*) \quad \text{positiv semi-definit}\tag{2.13}$$

Dabei ist die Qualifikation der Nebenbedingungen zu beachten, die fordert, daß deren Gradienten am Optimum linear unabhängig sein müssen. Sonst ist die Gleichung (2.10) nicht mehr eindeutig erfüllbar.

Die in Gleichung (2.13) eingeführte Funktion  $L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu})$  ist die sogenannte **Lagrange-Funktion** für das Problem (NLP). Sie lautet:

$$L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\mathbf{x}) + \boldsymbol{\lambda}^T \mathbf{g}(\mathbf{x}) + \boldsymbol{\mu}^T \mathbf{h}(\mathbf{x})\tag{2.14}$$

Sie entspricht der klassischen Lagrange-Funktion im ausschließlichen Fall von Gleichheitsnebenbedingungen.

Mit (2.14) lassen sich die Kuhn-Tucker-Bedingungen auch schreiben als:

$$\begin{aligned}\nabla_{\mathbf{x}} L(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) &= \mathbf{0} \\ \boldsymbol{\lambda}^{*T} \nabla_{\boldsymbol{\lambda}} L(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) &= 0 \\ \nabla_{\boldsymbol{\mu}} L(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) &= \mathbf{0} \\ \boldsymbol{\lambda}^* &\geq \mathbf{0}\end{aligned}\tag{2.15}$$

Damit (2.15) ein lokales Minimum liefert, muß die in den Tangentialraum projizierte Hesse-Matrix der Lagrange-Funktion (2.13) positiv semi-definit sein. Sind Zielfunktion und Nebenbedingungen konvex, dann ist (2.15) die Bedingung für ein globales Minimum. Dieser Fall kann in der Strukturoptimierung jedoch nicht garantiert werden, es ist sogar so, daß unterschiedliche, aber physikalisch gleichwertige Formulierungen von Nebenbedingungen im einen Fall konvexe und im anderen nicht konvexe Funktionen liefern. Im allgemeinen – vor allem bei der Formoptimierung in Verbindung mit der finiten Element-Methode zur Tragwerksberechnung – sind die Eigenschaften von Zielfunktion und Nebenbedingungen nicht näher zu bestimmen. Es muß daher davon ausgegangen werden, daß nicht konvexe Funktionen vorliegen und daß die Ergebnisse lokale Minima sind. Vom praktischen Standpunkt ist diese Einschränkung des absoluten Werts der Ergebnisse gewöhnlich nicht so bedeutend, da meist schon eine ausreichende Verbesserung der Entwürfe ausreicht. Hierbei erhöht die wiederholte Berechnung mit verschiedenen Startwerten die Sicherheit, das globale Optimum zu erreichen, wesentlich.

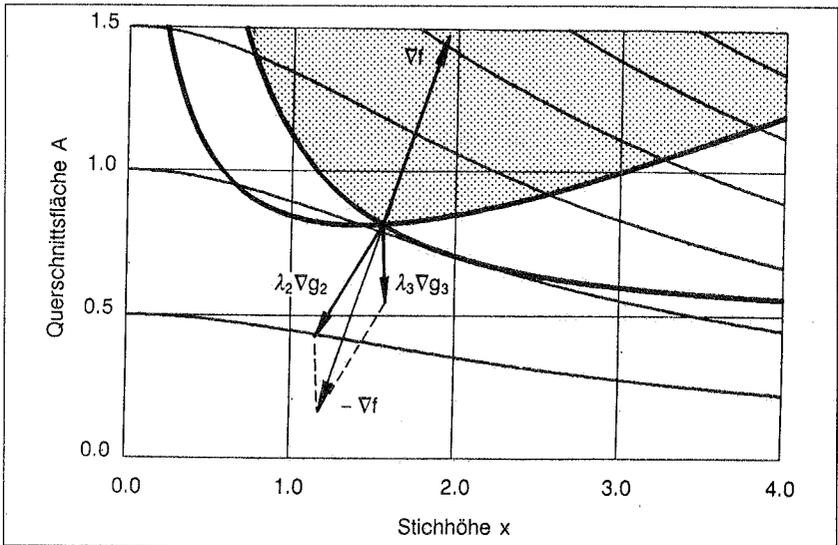


Bild 2.3: Kuhn-Tucker-Bedingung

Die Kuhn-Tucker-Bedingungen lassen sich geometrisch deuten. Wie bereits festgestellt, heben sich am Optimum der Gradient der Zielfunktion und die lineare Kombination der Gradienten der Nebenbedingungen gegenseitig auf. Da für Ungleichheitsbedingungen die Lagrange-Multiplikatoren nicht negativ sein dürfen, bedeutet dies geometrisch, daß der negative Gradientenvektor der Zielfunktion sich innerhalb des Ke-

gels befinden muß, der von den Gradientenvektoren der Nebenbedingungen aufgespannt wird. Dies ist in Bild 2.3 für den Fall b) des zweistöbigen Fachwerks aus Abschnitt 2.2 dargestellt.

### 2.7.2 Eigenschaften der Lagrange-Funktion

Die Lagrange-Funktion (2.14) im gemeinsamen  $(n + m)$ -dimensionierten Raum der primären Variablen  $x$  und der dualen Variablen  $\lambda$  und  $\mu$  ist eine Sattelfläche. Der Minimalpunkt  $x^*$  des primären Problems entspricht einem Sattelpunkt der Lagrange-Funktion. Das bedeutet: für die dualen Variablen ist die Lösung eine Maximalstelle (min-max-Eigenschaft).

$$L(x^*, \lambda, \mu) \leq L(x^*, \lambda^*, \mu^*) \leq L(x, \lambda^*, \mu^*) \quad (2.16)$$

Im Bild 2.4 ist die Lagrange-Funktion des zweistöbigen Fachwerks aus Abschnitt 2.2 für den Fall a) dargestellt. Der optimale Wert der Querschnittsfläche ist festgehalten, damit die Funktion in Abhängigkeit der beiden verbleibenden Variablen  $x$  und  $\lambda$  dargestellt werden kann. Man erkennt in der Aufsicht den Sattelpunkt der Fläche und die dazugehörige Lösung  $x^* = 2$  und  $\lambda^* = 4$ .

Die Sattelpunkteigenschaft gilt allgemein und trifft auch für Fälle zu, die bislang ausgeschlossen waren. Sie gilt außer für zweifach stetig ableitbare Funktionen für konvexe, nicht konvexe, nicht ableitbare oder sogar diskontinuierliche Probleme [88]. Schmit und Fleury [121] nutzen dies beim Einsatz eines dualen Algorithmus zur Lösung von Problemen der Strukturoptimierung mit diskreten Variablen. In der Tat lassen sich die Kuhn-Tucker-Bedingungen auch ohne die im vorherigen Abschnitt getroffenen Annahmen herleiten (siehe z.B. bei Künzi et al. [81]). Dann gelten selbstverständlich auch andere Qualifikationen für die Nebenbedingungen [98]. Des weiteren gibt es außer den Kuhn-Tucker-Bedingungen noch andere Formulierungen für die notwendigen Bedingungen wie z.B. das Fritz John-Kriterium. Bazaraa [4] gibt in seinem Buch einen umfassenden Überblick über bestehende Möglichkeiten.

Die "min-max-Eigenschaften" der Lagrange-Funktion werden bei der Entwicklung von Algorithmen aus den Klassen der dualen und der Lagrange-Methoden eingebracht. Im Gegensatz zu den primären Verfahren und den Penalty-Methoden liefern diese deshalb sowohl die optimalen Werte der primären wie der dualen Variablen.

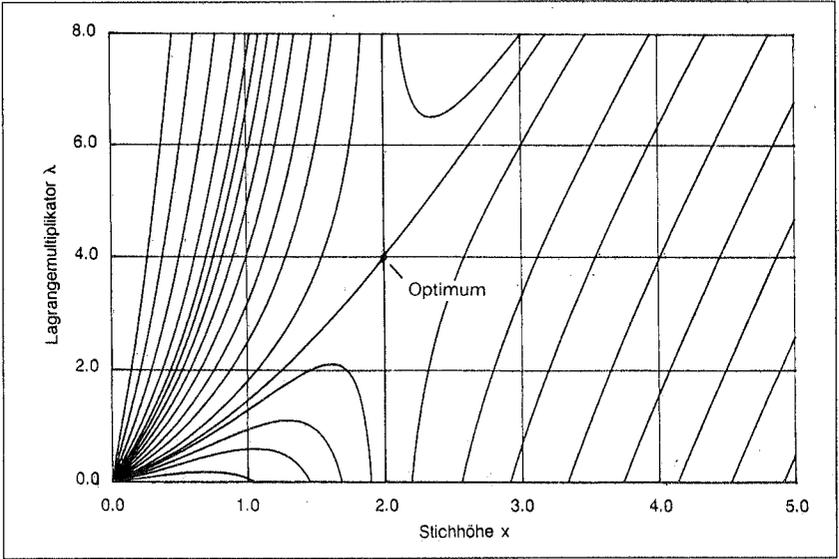


Bild 2.4a: Lagrangefunktion des Fachwerkproblems (Fall a,  $A = A^*$ ), Aufsicht

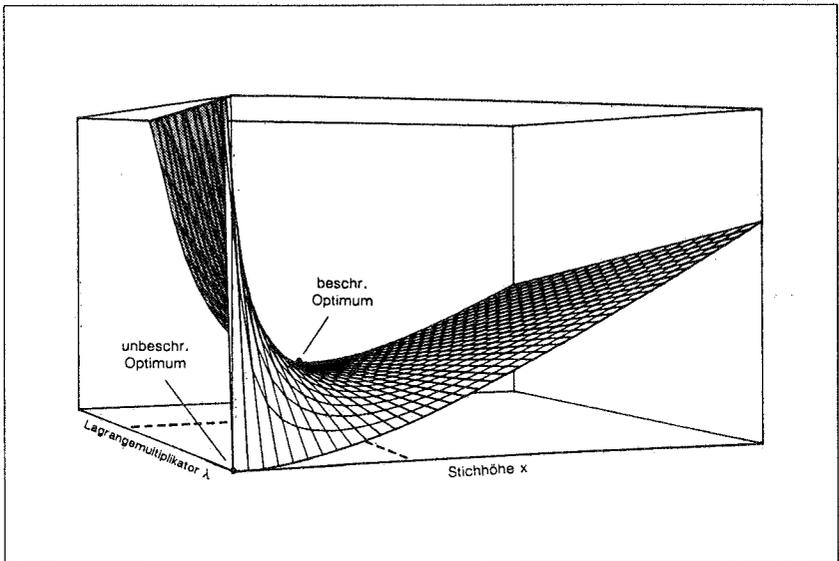


Bild 2.4b: Lagrangefunktion des Fachwerkproblems (Fall a,  $A = A^*$ ), Ansicht

### 2.7.3 Qualifikation und Redundanz von Nebenbedingungen

Wichtige Voraussetzung bei der Herleitung der Kuhn–Tucker–Bedingungen (2.15) war die Forderung einer Qualifikation der Nebenbedingungen. In der üblichen und für Algorithmen bedeutenden Form heißt das, daß die Gradienten der Nebenbedingungen am Optimum linear unabhängig sein müssen. Gewöhnlich geht man davon aus, daß dies zutrifft; jedoch schon das kleine Beispiel von Abschnitt 2.2 zeigt für den Fall c) (Verschiebungs- und Fließbedingungen sind gleichzeitig aktiv), daß Vorsicht geboten ist. Im Bild 2.5 ist der Sachverhalt noch einmal verdeutlicht: an der Lösung sind die Gradienten der Zielfunktion und beider Nebenbedingungen kollinear, die Qualifikationsbedingung ist verletzt, die Kuhn–Tucker–Bedingungen sind nicht lösbar. Ein guter Algorithmus ist dennoch in der Lage, solche Situationen zu bewältigen, indem er nur die gerade notwendige Zahl an Nebenbedingungen als aktiv erkennt und die übrigen nicht beachtet. Deren Lagrange–Multiplikatoren werden dann zu null gesetzt. Allerdings kann auch der beste Algorithmus scheitern, und deshalb ist es zu empfehlen, im Falle von Konvergenzschwierigkeiten oder von aktiven Nebenbedingungen mit Lagrange–Multiplikatoren vom Wert null, die Qualifikation der Nebenbedingungen zu überprüfen und neu zu starten. Leider ist es in der Regel unmöglich, a priori Aussagen über die Verhältnisse am (unbekannten) Optimum zu machen.

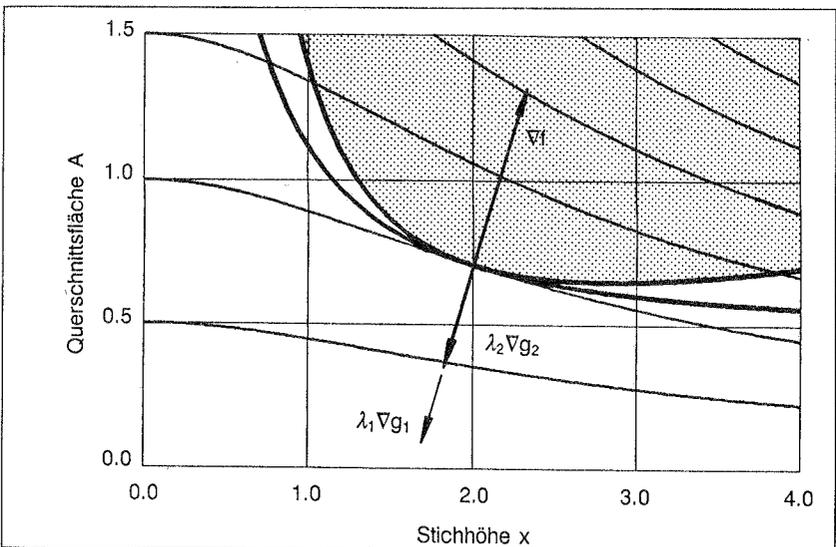


Bild 2.5: nichterfüllte Qualifikation von Nebenbedingungen mit linear abhängigen Gradienten

Ein anderer Fall, in dem Lagrange-Multiplikatoren null sein können, die dazugehörigen Nebenbedingungen jedoch aktiv sind, ist der Fall von redundanten Nebenbedingungen. Auch dies kann an dem zweiständigen Fachwerk von Abschnitt 2.2 demonstriert werden. Im Fall d) sind Fließ- und Knickbedingung gleichzeitig aktiv, und die Gradienten sind linear unabhängig (siehe Bild 2.6). Dennoch ist der Lagrange-Multiplikator der Knickbedingung null, da die Gradienten der Fließbedingung und der Zielfunktion bereits in der gleichen Richtung liegen. Die Knickbedingung ist redundant, da sie zur Erfüllung der Kuhn-Tucker-Bedingungen nicht herangezogen werden muß. Auch diesen Fall können gute Algorithmen lösen, jedoch sollten auch hier die Ergebnisse einer näheren Untersuchung unterzogen werden, um falsche Resultate zu vermeiden. Insbesondere wenn Lagrange-Multiplikatoren sich nur sehr wenig von null unterscheiden und die Auswahl des aktiven Satzes von ihnen abhängig ist, können numerische Schwierigkeiten die Konvergenz verhindern. In diesen Fällen empfiehlt sich eine Skalierung der Nebenbedingungen [53] und eine wiederholte Rechnung.

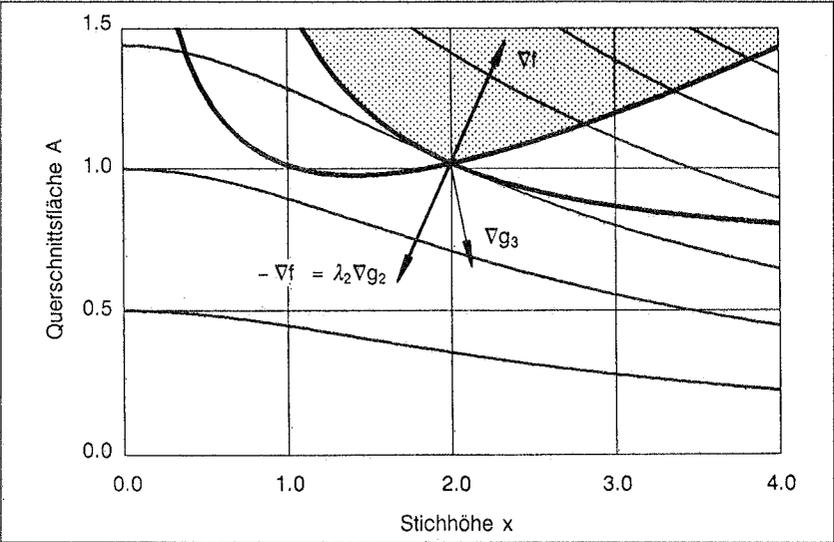


Bild 2.6: redundante Nebenbedingung

## 2.7.4 Konvergenzeigenschaften der Abstiegsmethoden

### 2.7.4.1 Abstiegsmethoden

Den meisten Algorithmen liegt ein gemeinsames Konzept zugrunde: Ausgehend von einem Startpunkt wird eine Folge von Punkten generiert, die so beschaffen ist, daß jeweils aufeinander folgenden Punkten stetig abnehmende Werte einer sogenannten Abstiegsfunktion (auch merit function) entsprechen. Als Abstiegsfunktion kommt bei unbeschränkten Problemen die Zielfunktion selbst in Frage, während bei beschränkten Problemen Funktionen benötigt werden, die auch den Einfluß der Nebenbedingungen erfassen (siehe dazu den Abschnitt "Lagrange-Methoden"). Zwei Fragen sind zu Beginn einer jeden Iteration zu beantworten: In welcher abwärts führenden Richtung soll der nächste Punkt gesucht werden und wie weit soll in dieser Richtung fortgeschritten werden? Während das Bestimmen der Suchrichtung das Geheimnis jedes Verfahrens ist, wird die anschließende Schrittweitenbestimmung durch eine Linienminimierung der Abstiegsfunktion in dieser Richtung vorgenommen (line search). Mit Ausnahme des klassischen Newton-Verfahrens, das auf einen line search verzichtet, sind alle gängigen Verfahren in dieser Weise aufgebaut. Deshalb kommt auch den eindimensionalen line search Verfahren bei der Beurteilung der globalen Konvergenz eines Abstiegsverfahrens eine wichtige Bedeutung zu.

Ein iteratives Abstiegsverfahren heißt dann global konvergent, wenn garantiert ist, daß die Folge der generierten Punkte zu einer Lösung konvergiert. Es ist zu beachten, daß global konvergent nicht heißt, der Algorithmus konvergiere zum globalen Minimum. Globale Konvergenz ist eine wichtige Voraussetzung für einen guten Algorithmus, der sich auch in der Anwendung bewähren soll. Dennoch sind viele der wichtigsten Algorithmen in ihrer reinen Form nicht global konvergent, was sich aber durch Kombination mit einem Verfahren, das garantiert global konvergiert, beheben läßt. Gewöhnlich wird dazu das Verfahren des steilsten Abstiegs eingesetzt. Eine einheitliche Theorie zur Konvergenz der Algorithmen wurde von Zangwill ausgearbeitet und von Luenberger [85] weiterentwickelt. Von ihm stammen auch wichtige Aussagen zur lokalen Konvergenz, das sind Konvergenzordnung und  $-rate$ .

#### 2.7.4.2 Konvergenzordnung und Konvergenzrate

Eine Folge von reellen Werten  $x_k$  konvergiere für  $k = 0$  bis  $\infty$  zu ihrem Grenzwert  $x^*$ . Dann ist die Ordnung der Konvergenz von  $x_k$  als der größtmögliche, positive Wert der Zahl  $p$  definiert, der folgende Beziehung erfüllt:

$$0 \leq \lim_{k \rightarrow \infty} \frac{|x_{k+1} - x^*|}{|x_k - x^*|^p} < \infty \quad (2.17)$$

Die meisten Verfahren haben eine Konvergenz der Ordnung eins, man spricht dann auch von linearer Konvergenz. Das klassische Newton-Verfahren ist eines der wenigen Verfahren mit quadratischer Konvergenz, d.h. Konvergenz der Ordnung zwei.

Neben der Ordnung interessiert auch die Konvergenzrate, d.h. eine Aussage, wie schnell ein Algorithmus innerhalb einer gegebenen Konvergenzordnung konvergiert. Die Konvergenzrate  $\beta$  ist dabei der Grenzwert der Beziehung (2.17):

$$\beta = \lim_{k \rightarrow \infty} \frac{|x_{k+1} - x^*|}{|x_k - x^*|} < 1 \quad (2.18)$$

Im dargestellten Fall (2.18) konvergiert die Folge  $x_k$  linear mit der Konvergenzrate  $\beta$ . Ist  $\beta = 0$ , dann konvergiert sie superlinear, wie auch alle Folgen mit einer Ordnung der Konvergenz größer als eins. Konvergenzordnung und -rate werden auch als lokale Konvergenz bezeichnet.

Die Eigenart aller Konvergenzbetrachtungen ist, daß sie das Konvergenzverhalten im Limes betrachten, also das Verhalten kurz vor der Lösung. Andererseits sind in der praktischen Anwendung jedoch meist die Verbesserungen in der Startphase eines Algorithmus von besonderem Interesse, da oft nur eine ausreichende Modifikation der Startlösung anstelle eines lokalen oder gar globalen Minimums gesucht ist. Deshalb können Konvergenzaussagen nur zu einer Klassifikation verschiedener Verfahren dienen, haben sich aber in der Praxis als ein Mittel zur Vorauswahl geeigneter Verfahren bewährt. Auch kann es sich für praktische Anwendungen durchaus lohnen, die besonderen Konvergenzeigenschaften eines Verfahrens näher zu betrachten, um daraus Schlüsse für eine verbesserte Problemformulierung zu ziehen. Am Beispiel des BFGS-Verfahrens (Abschnitt 2.8.2.2) wird dies verdeutlicht werden.

## 2.8 Algorithmen

### 2.8.1 Verfahren der eindimensionalen Optimierung (Line Search)

Verfahren der eindimensionalen Optimierung (line search – Methoden) sind ein wichtiger Bestandteil vieler Abstiegsmethoden. Oft sind exakt ausgeführte line searches auch Voraussetzung für den Nachweis globaler Konvergenz. Die Anforderungen an die line search–Methoden sind dabei recht unterschiedlich. Während Gradientenverfahren (steilster Abstieg, konjugierte Gradienten–Verfahren, Verfahren der zulässigen Richtungen, etc.) auf eine hohe Genauigkeit der eindimensionalen Suche angewiesen sind, genügt modifizierten oder Quasi–Newton–Verfahren meist eine grobe Abschätzung, da die vorgegebene Schrittweite mit zunehmender Konvergenz immer besser an das "eindimensionale" Minimum rückt. Dementsprechend gibt es zwei verschiedene Strategien, denen beiden die Annahme zugrunde liegt, daß die zu minimierende eindimensionale Funktion unimodal ist.

Eine Strategie für einen "exakten" line search umfaßt die Schritte:

1. Finde ein Intervall  $[0,1]$ , das das Linienminimum einschließt.
2. Finde das Linienminimum  $\alpha$  innerhalb dieses Intervalls,  $\alpha \in [0,1]$ 
  - mit Intervallschachtelung (Fibonacci, Goldener Schnitt),
  - mit Interpolationsverfahren,
  - einer Kombination verschiedener Verfahren  
(safe guarded line search, Gill et al. [52])

Die Strategie für einen groben line search (insbesondere in Verbindung mit Quasi–Newton–Methoden) ist:

1. Überprüfe, ob die geforderte Genauigkeit schon mit der vorgegebenen Schrittweite erbracht ist (Goldstein/Armijo–Test).
2. Verringere die Schrittweite, falls 1.) nicht zutrifft und wiederhole den Vorgang. Eine Extrapolation ist nicht erlaubt, denn durch eine unvorsichtige Extrapolation wäre ein Abstieg nicht garantiert.

Line search – Verfahren werden in der Literatur sehr ausführlich beschrieben [52], [81], [85] und [88], so daß an dieser Stelle nur die quadratische und kubische Interpolation sowie der Goldstein/Armijo–Test näher erläutert werden sollen, da im Abschnitt 2.8.6.4 über SQP–Verfahren auf gewisse Eigenschaften dieser Verfahren verwiesen wird.

#### 2.8.1.1 Quadratische Interpolation

Interpolationsverfahren nähern die Kurve durch Polynome an, indem an verschiedenen Taststellen die Funktionswerte und eventuell auch Ableitungen und Krümmungen be-

stimmt werden. Anschließend wird das Minimum des angenäherten Polynoms ermittelt. An dieser Stelle wird die Originalfunktion neu ausgewertet, approximiert, minimiert usw., bis das Linienminimum gefunden ist. Voraussetzungen sind dabei, daß die zweiten Ableitungen in Suchrichtung existieren und kontinuierlich sind. Für eine quadratische Interpolation mit drei unbekanntenen Parametern gibt es zwei Möglichkeiten zur Parameterbestimmung:

- a) Funktionswert und Ableitung am Startpunkt plus Funktionswert am Ende des Tastschrittes.
- b) Funktionswerte am Startpunkt, Ende des Tastschrittes und dazwischen (möglichst in der Mitte).

Für Verfahren a) spricht, daß im Idealfall der Schrittweite eins keine zusätzliche Funktionsauswertung anfällt, jedoch kann die so vorgenommene Approximation deutlich falsch sein (siehe Bild 2.7). Möglichkeit b) nähert meist den Funktionsverlauf zwar besser an, benötigt dafür aber mehr Funktionsaufrufe. Für beide Verfahren spricht, daß sie ohne zusätzliche Gradientenbestimmungen auskommen, was sehr viel Aufwand spart. Die Interpolationsvorschriften sind in Tafel 2.7 angegeben.

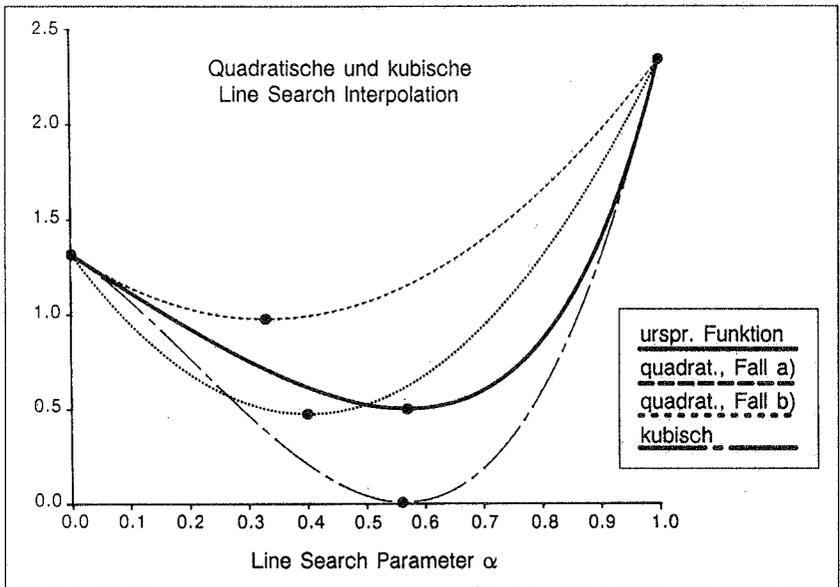


Bild 2.7: Quadratische und kubische line search-Interpolation

### 2.8.1.2 Kubische Interpolation

Bei der kubischen Interpolation sind vier Freiwerte zu bestimmen. Dies sind die Funktionswerte und Ableitungen am Intervallanfang und -ende. Damit gelingt eine gute Approximation, allerdings mit dem hohen Preis, daß zusätzliche Gradientenauswertungen anfallen, wenn die Schrittweite korrigiert werden muß. Es hängt vom Einzelfall ab, ob dieser Mehraufwand gegenüber der quadratischen Approximation gerechtfertigt ist. Die Interpolationsvorschrift ist in Tafel 2.7 angegeben, eine graphische Verdeutlichung findet man in Bild 2.7. Auch hier wird vorausgesetzt, daß die zweiten Ableitungen in Suchrichtung existieren und kontinuierlich sind.

quadratische Interpolation	kubische Interpolation
Fall a) $a_m = \frac{1}{2} \frac{a_1^2 \phi'(0)}{a_1 \phi'(0) - (\phi(a_1) - \phi(0))}$	$a_m = \left[ 1 - \frac{\phi'(a_1) + b - a}{\phi'(a_1) - \phi'(0) + 2b} \right] a_1$
Fall b) $a_m = \frac{3\phi(0) - 4\phi\left(\frac{a_1}{2}\right) + \phi(a_1)}{2(\phi(0) - 2\phi\left(\frac{a_1}{2}\right) + \phi(a_1))} a_1$	$a = \phi'(0) + \phi'(a_1) + 3 \frac{\phi(0) - \phi(a_1)}{a_1}$ $b = (a^2 - \phi'(0) \cdot \phi'(a_1))^{\frac{1}{2}}$

Tafel 2.7: quadratische und kubische Interpolationsvorschriften

### 2.8.1.3 Armijo-Test

Der Armijo-Test ist ein besonders in Verbindung mit Quasi-Newton-Verfahren oft gebrauchtes Verfahren für einen groben line search. Die Idee ist, daß die eigentliche Funktion mit einer abwärts zeigenden Sekante durch die Funktion am Intervallanfang verglichen wird (siehe Bild 2.8). Solange die Funktion unterhalb der Sekante bleibt, ist garantiert, daß der Gesamtalgorithmus abwärts führt. Der Test ist erfüllt, wenn folgende Gleichung gilt:

$$\phi(a) \leq \phi(0) + \epsilon \phi'(0) \cdot a \quad ; \quad 0 < \epsilon < 1 \quad (2.19)$$

Ist  $\phi$  eine quadratische Funktion, dann trifft die Sekante mit  $\epsilon = 0,5$  genau das Liniennminimum. Deshalb sollte man  $\epsilon$  nie größer als 0,5 wählen. Luenberger [85] empfiehlt einen Wert  $\epsilon = 0,2$ .

Wenn der Test nicht erfüllt wurde, wird mit einem der oben genannten Interpolationsverfahren ein neuer Testpunkt gesucht. Andererseits darf die Schrittweite nicht zu klein werden, damit der Gesamtalgorithmus noch genügend "Vortrieb" hat. Dafür wird folgende Vorschrift angegeben:

$$a_m \geq \eta \cdot a_1 \tag{2.20}$$

Schittkowski [117] schlägt für seinen Algorithmus  $\eta = 0,1$  vor. Eine Alternative für die untere Schranke wird von Goldstein angegeben, die einer weiteren Sekante entspricht, die nicht unterschritten werden darf. Die Regel lautet:

$$\phi(a) > \phi(0) + (1 - \epsilon) \phi'(0) a$$

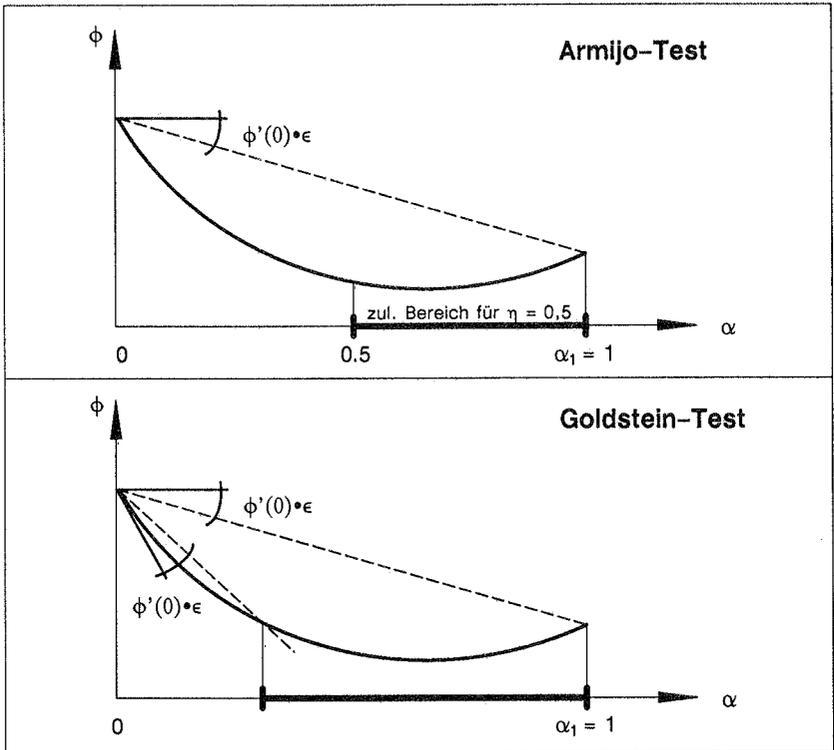


Bild 2.8: Armijo- und Goldstein-Test

## 2.8.2 Quasi-Newton-Methoden

Die zur Zeit mit am besten beurteilten Verfahren zur Lösung nichtlinearer Gleichungssysteme und unbeschränkter, nichtlinearer Optimierungsaufgaben sind die Quasi-Newton-Methoden. Sie stellen einen Kompromiß zwischen der klassischen Newton-Methode und den Gradientenverfahren dar. Einen guten Überblick über Theorie und Motivation dieser Verfahren findet man in dem als Standardwerk zu bezeichnenden Beitrag von Dennis und Moré [29]. Die Entwicklung der Quasi-Newton-Verfahren reicht zurück bis ins Jahr 1959, als Davidon das Verfahren der variablen Metrik fand, das später, 1963, von Fletcher und Powell entwickelt wurde. Den vorläufigen Höhepunkt fand die Entwicklung 1970 mit der BFGS-Korrekturformel, die unabhängig voneinander von Broyden, Fletcher, Goldfarb und Shanno vorgeschlagen wurde. In Verbindung mit beschränkten, nichtlinearen Optimierungsaufgaben kam insbesondere dem BFGS-Verfahren 1977 weitere Bedeutung zu, als Han [64] und Powell [100] ein Verfahren der sequentiellen quadratischen Programmierung (SQP) entwickelten.

### 2.8.2.1 Theorie der Quasi-Newton-Methoden

Die klassische Newton-Methode vereint die zwei wichtigen Teilaufgaben eines iterativen Algorithmus: durch die Wahl einer quadratischen Approximation der Zielfunktion können Abstiegsrichtung und Schrittweite gleichzeitig bestimmt werden. Ein line search kann damit entfallen. Das Verfahren zeigt zumindest in der Nähe der Lösung ein Konvergenzverhalten zweiter Ordnung. Das zugrunde liegende quadratische Modell wird durch eine nach dem zweiten Glied abgebrochene Taylor-Reihen-Entwicklung der Zielfunktion gewonnen, wobei vorausgesetzt ist, daß die dazu notwendigen ersten und zweiten Ableitungen existieren und ermittelt werden können:

$$f(\mathbf{x}_k + \mathbf{p}_k) = f(\mathbf{x}_k) + \mathbf{g}_k^T \mathbf{p}_k + \frac{1}{2} \mathbf{p}_k^T \mathbf{F}_k \mathbf{p}_k \quad (2.21)$$

$$\begin{aligned} \text{mit:} \quad \mathbf{p}_k &= \mathbf{x}_{k+1} - \mathbf{x}_k \\ \mathbf{g}_k &= \nabla f(\mathbf{x}_k) \\ \mathbf{F}_k &= \nabla^2 f(\mathbf{x}_k) \end{aligned}$$

Der nächste Schritt  $\mathbf{p}_k$  bestimmt sich aus der Minimierung der quadratischen Ersatzfunktion zu:

$$\mathbf{p}_k = -\mathbf{F}_k^{-1} \mathbf{g}_k \quad (2.22)$$

Vorausgesetzt wurde hier weiter, daß die Hesse-Matrix  $\mathbf{F}_k$  positiv definit sei. Andernfalls gibt es verschiedene Strategien, diesen Defekt auszugleichen und dennoch eine Abstiegsrichtung zu finden (siehe z.B. Gill et al. [52]).

Der Nachteil der klassischen Newton-Methode ist die Notwendigkeit, daß die zweiten Ableitungen laufend, in jedem Schritt bestimmt werden müssen. Um diesen Aufwand zu verringern, aber dennoch den Vorzug der Gleichung (2.22) bei der Bestimmung von Suchrichtung und Schrittweite weiterhin genießen zu können, entstanden modifizierte Newton-Verfahren. Anstelle der Hesse-Matrix wird nun in Gleichung (2.22) eine Approximation vorgenommen. Damit wird der neue Schritt  $\mathbf{p}_k$  berechnet zu:

$$\mathbf{p}_k = -\alpha_k \mathbf{S}_k \mathbf{g}_k \quad (2.23)$$

Der Schrittweitenparameter  $\alpha_k$  wird dabei so gewählt, daß  $f(\mathbf{x}_k + \alpha_k \mathbf{p}_k)$  minimal wird (line search). Gleichung (2.23) ist die Basisgleichung einer Reihe bekannter Algorithmen. Sind  $\mathbf{S}_k = \mathbf{F}_k^{-1}$  und  $\alpha_k = 1$ , dann entspricht (2.23) dem Newton-Verfahren. Wird für  $\mathbf{S}_k$  die Inverse der Hesse-Matrix  $\mathbf{F}_0$  am Startpunkt während der gesamten Iteration gesetzt, handelt es sich um das übliche modifizierte Newton-Verfahren. Dabei werden nicht immer line searches durchgeführt. Ist  $\mathbf{S}_k$  dagegen ständig gleich der Einheitsmatrix, wird (2.23) zum Verfahren des steilsten Abstiegs mit der notwendigen Bestimmung von  $\alpha_k$  in einem line search.

Die Klasse der Quasi-Newton-Verfahren verwendet ebenfalls Gleichung (2.23) zur Bestimmung der Abstiegsrichtung, die Matrix  $\mathbf{S}_k$  wird aber während der Iteration der inversen Hesse-Matrix angeglichen. Die Grundidee dabei ist, daß in jedem Schritt die Krümmung der Zielfunktion in Suchrichtung durch einen Differenzenausdruck der ersten Ableitungen approximiert wird. Aus:

$$\mathbf{p}_k^T \mathbf{B}_{k+1} \mathbf{p}_k = (\mathbf{g}_{k+1} - \mathbf{g}_k)^T \mathbf{p}_k \quad (2.24)$$

$$\text{mit:} \quad \mathbf{g}_{k+1} = \nabla f(\mathbf{x}_k + \mathbf{p}_k)$$

$$\text{und} \quad \mathbf{S}_{k+1} = \mathbf{B}_{k+1}^{-1}$$

folgt die Quasi-Newton-Bedingung, die die neue Approximation  $\mathbf{B}_{k+1}$  der Hesse-Matrix in Beziehung mit den bekannten ersten Ableitungen bringt:

$$\mathbf{B}_{k+1} \mathbf{p}_k = \mathbf{g}_{k+1} - \mathbf{g}_k = \mathbf{q}_k \quad (2.25)$$

Geht man weiter davon aus, daß sich die Matrix  $\mathbf{B}_{k+1}$  nur wenig von  $\mathbf{B}_k$  unterscheidet, bietet sich eine Korrektur mit einer Matrix vom Rang eins an, die allgemein folgendermaßen dargestellt werden kann:

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \mathbf{u} \cdot \mathbf{v}^T \quad (2.26)$$

Diese Beziehung in (2.25) eingesetzt, führt zur Korrekturformel:

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \frac{1}{\mathbf{v}^T \mathbf{p}_k} (\mathbf{q}_k - \mathbf{B}_k \mathbf{p}_k) \mathbf{v}^T \quad (2.27)$$

wobei sich  $\mathbf{u}$  zu

$$\frac{1}{\mathbf{v}^T \mathbf{p}_k} (\mathbf{q}_k - \mathbf{B}_k \mathbf{p}_k)$$

ergibt.

Aus (2.27) leiten sich die bekannten Rang-eins-Korrekturen von Broyden und Davidon ab:

Broyden (2.28)	Davidon (2.29)
$\mathbf{v} = \mathbf{p}_k$ $\mathbf{B}_{k+1} = \mathbf{B}_k + \frac{1}{\mathbf{p}_k^T \mathbf{p}_k} (\mathbf{q}_k - \mathbf{B}_k \mathbf{p}_k) \mathbf{p}_k^T$	$\mathbf{v} = \mathbf{q}_k - \mathbf{B}_k \mathbf{p}_k$ $\mathbf{B}_{k+1} = \mathbf{B}_k + \frac{1}{(\mathbf{q}_k - \mathbf{B}_k \mathbf{p}_k)^T \mathbf{p}_k} (\mathbf{q}_k - \mathbf{B}_k \mathbf{p}_k) (\mathbf{q}_k - \mathbf{B}_k \mathbf{p}_k)^T$

Die Broyden-Korrektur ist unsymmetrisch, was sie im Grunde für eine Approximation der symmetrischen Hesse-Matrix nicht empfiehlt. Dennoch wurden auch mit dieser Formel gute Erfahrungen bei der Lösung nichtlinearer Gleichungssysteme gemacht [124]. Dagegen ist die Formel von Davidon symmetrisch, weshalb sie auch symmetrische Rang-eins-Korrektur genannt wird (symmetric rank-one update).

Die erfolgreichsten Quasi-Newton-Methoden sind Verfahren mit symmetrischen Korrekturen vom Rang zwei:

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \mathbf{u}\mathbf{u}^T + \mathbf{v}\mathbf{v}^T \quad (2.30)$$

Im Gegensatz zu (2.28) und (2.29) können mit diesem Korrekturschema konjugierte Richtungen erzeugt werden. Die zwei bekanntesten Verfahren sind das Verfahren der variablen Metrik von Davidon, Fletcher und Powell, auch DFP-Verfahren genannt, und das BFGS-Verfahren.

Die DFP-Formel lautet:

$$\mathbf{B}_{k+1} = \mathbf{B}_k - \frac{1}{\mathbf{p}_k^T \mathbf{B}_k \mathbf{p}_k} \mathbf{B}_k \mathbf{p}_k \mathbf{p}_k^T \mathbf{B}_k + \frac{1}{\mathbf{q}_k^T \mathbf{p}_k} \mathbf{q}_k \mathbf{q}_k^T + (\mathbf{p}_k^T \mathbf{B}_k \mathbf{p}_k) \mathbf{w}_k \mathbf{w}_k^T \quad (2.31)$$

mit:

$$\mathbf{w}_k = \frac{1}{\mathbf{q}_k^T \mathbf{p}_k} \mathbf{q}_k - \frac{1}{\mathbf{p}_k^T \mathbf{B}_k \mathbf{p}_k} \mathbf{B}_k \mathbf{p}_k \quad (2.31)$$

und die BFGS-Formel:

$$\mathbf{B}_{k+1} = \mathbf{B}_k - \frac{1}{\mathbf{p}_k^T \mathbf{B}_k \mathbf{p}_k} \mathbf{B}_k \mathbf{p}_k \mathbf{p}_k^T \mathbf{B}_k + \frac{1}{\mathbf{q}_k^T \mathbf{p}_k} \mathbf{q}_k \mathbf{q}_k^T \quad (2.32)$$

Der Unterschied zwischen (2.31) und (2.32) beruht im letzten Term der DFP-Formel. Versieht man diesen Term mit einem Faktor  $\phi$ , der zwischen 0 und 1 variieren kann, gewinnt man eine unbegrenzte Zahl möglicher Korrekturvorschriften zwischen der BFGS-Formel für  $\phi = 0$  und der DFP-Formel für  $\phi = 1$ . Diese Menge von verschiedenen Möglichkeiten wird auch "Broyden-Familie" genannt [85].

Die Verfahren der Broyden-Familie verhalten sich identisch, sofern der line search zur Bestimmung der Schrittweite  $\mathbf{p}_k$  (siehe Gleichung (2.23)) exakt ausgeführt wird. Dagegen führt ein ungenauer line search zu völlig unterschiedlichen Iterationsverläufen und, was vom praktischen Standpunkt aus gesehen am wichtigsten ist, zu einer unterschiedlichen Anzahl von Funktionsauswertungen. Dabei zeigt die BFGS-Formel die besten Eigenschaften, wie sich in vielen numerischen Anwendungen herausgestellt hat.

Werden die Quasi-Newton-Methoden zur Lösung eines  $n$ -dimensionalen quadratischen Problems herangezogen, ist die Hesse-Matrix des Problems nach höchstens  $n$  Schritten exakt approximiert. Man spricht dann von einem vollständigen Zyklus. Dementsprechend benötigen die Verfahren auch  $n$  Schritte zur Lösung des quadratischen Problems, was ihre Verwandtschaft mit den konjugierten Gradientenverfahren aufzeigt. (Näheres dazu in [85] oder [52]). Zur Beurteilung der Lösungsgeschwindigkeit beliebig nichtlinearer Probleme orientiert man sich am besten an der Zahl der Korrekturzyklen. Die globale Konvergenzeigenschaft der Verfahren kann für beliebige Probleme garantiert werden, indem die Korrekturzyklen jedesmal mit der Einheitsmatrix für  $\mathbf{B}_k$  begonnen werden. Dies entspricht jeweils einem Schritt des steilsten Abstiegs. Die Konvergenzordnung ist eins, die Rate im idealen Fall superlinear.

Eine Eigenschaft der Quasi-Newton-Methoden, die sich gerade bei großen strukturierteren Problemen mit einer dünnen Belegung der Hesse-Matrix bemerkbar macht, ist, daß die Korrekturmatrizen diese Struktur zerstören. In diesen Fällen ist es vorteilhaft, die Korrekturvektoren der einzelnen Schritte aufzubewahren und beim nächsten Schritt über die rechte Seite des Gleichungssystems zu berücksichtigen [87]. Dieses Vorgehen kann allerdings nur für weniger als einen vollen Korrekturzyklus effizient sein. Eine andere Möglichkeit ist es, nur den aktuellen Korrekturvektor zu berücksichtigen und als Approximationsmatrix des vorherigen Schrittes die Einheitsmatrix anzunehmen [52]. Liegt die Hesse-Matrix in faktorisierter Form vor, wird von Gill et al. [52] ein Verfahren beschrieben, das es erlaubt, direkt die faktorisierten Matrizen zu verbessern. Dies bietet neben verringertem Rechenaufwand auch die Möglichkeit, die positive Definitheit der

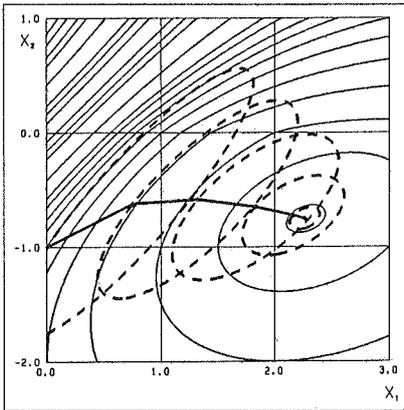
Matrix zu garantieren. Auf diese Eigenschaft weist Arora [1] im Zusammenhang mit der Strukturoptimierung besonders hin.

Der line search hat bei den Methoden der Broyden-Familie die wichtige Funktion, die positive Definitheit der Matrix  $\mathbf{B}$  während der ganzen Iteration zu garantieren. Dies ist der Fall, sofern

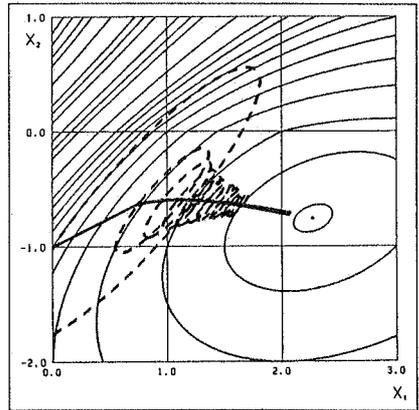
$$\mathbf{q}_k^T \mathbf{p}_k > 0 \quad \text{ist.} \quad (2.33)$$

Diese Bedingung kann schon durch einen ausreichend genauen line search erfüllt werden. Die exakte Bestimmung des Linienminimums ist nicht nötig, z.B. genügt ein line search nach der Armijo/Goldstein-Regel. Gewöhnlich geht man so vor, daß der line search - Parameter  $\alpha_k$  nie größer als eins (der idealen Newton-Schrittweite), aber auch nicht negativ sein darf. Andererseits darf die Bedeutung des line search auch bei Quasi-Newton-Verfahren nicht unterschätzt werden, wie die in Abschnitt 2.8.6.4 im Zusammenhang mit dem SQP-Verfahren geschilderten Erfahrungen zeigen. Powell hat diesem Problem eine Veröffentlichung gewidmet mit dem provokativen Titel: "Wie schlecht sind die DFP- und BFGS-Methoden?" [103]. Ein Ergebnis seiner Untersuchungen zeigt auch die Überlegenheit der BFGS-Formel über die DFP-Formel.

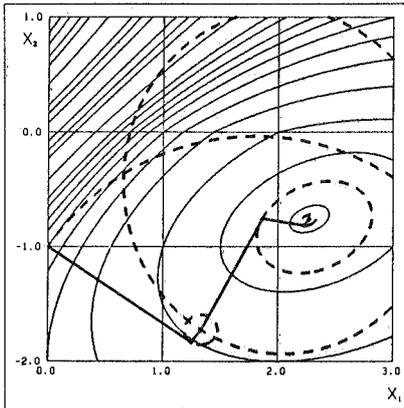
In Bild 2.9 sind für ein in [11] beschriebenes unbeschränktes Problem der Strukturoptimierung die Iterationsverläufe für verschiedene Newton-Verfahren angegeben. Es handelt sich dabei um die Formoptimierung eines 5-ständigen Fachwerkes, dessen Gewicht bei vorgegebenen zulässigen Spannungen zu minimieren ist. Die Unbekannten  $x_1$  und  $x_2$  sind vertikale Koordinaten von Ober- bzw. Untergurt. Gestrichelt sind je eine charakteristische Höhenlinie (Ellipsen) der quadratischen Unterprobleme eingetragen, wie sie sich während der Iteration ergeben. Trotz eines exakten line searches fällt das BFGS-Verfahren deutlich gegen die reine Newton-Raphson Methode ab, was allerdings von den Startwerten der am Anfang diagonalen Hessematrix abhängt. Für geeignete Startwerte fällt die mehrfach erwähnte Verwandtschaft mit den konjugierten Gradientenverfahren auf, wie ein Vergleich der Bilder 2.9c und 2.11b zeigt. Das modifizierte Newton-Verfahren ist für dieses Beispiel und diesen Startpunkt ungeeignet, eine Modifikation der Hessematrix ist offensichtlich für eine rasche Konvergenz unerlässlich. Das modifizierte Verfahren benötigt bei gleicher Lösungsgenauigkeit 139 Iterationen gegenüber 9 des reinen Newton-Verfahrens.



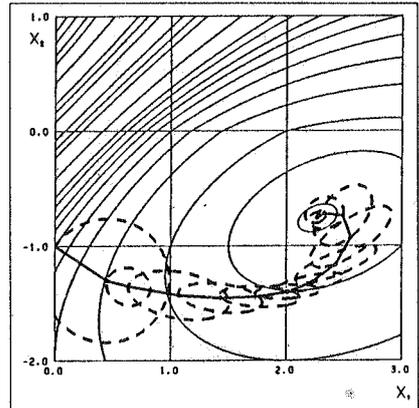
**Bild 2.9a:** *Iterationsverlauf  
reiner Newton-Raphson*



**Bild 2.9b:** *Iterationsverlauf  
mod. Newton-Verfahren*



**Bild 2.9c:** *Iterationsverlauf BFGS  
Anfangswert = 50.0  
(für die Diagonalwerte der Hessematrix)*



**Bild 2.9c:** *Iterationsverlauf BFGS  
Anfangswert = 200.0  
(für die Diagonalwerte der Hessematrix)*

### 2.8.2.2 Konvergenzeigenschaften der Quasi-Newton-Verfahren

Quasi-Newton-Verfahren sind inzwischen zu Standardverfahren zur Lösung nichtlinearer Gleichungssysteme gereift wie zum Beispiel bei der Berechnung des Tragverhaltens geometrisch und materiell nichtlinearer Strukturen [87], [124]. Dabei werden in der Regel sehr viel weniger Iterationsschritte ausgeführt als die Zahl der Unbekannten beträgt, bis der nächste Zyklus begonnen wird und neue Startwerte der Hesse-Matrix (sie

entspricht der Tangentensteifigkeitsmatrix) bestimmt werden. Das heißt, einerseits liegt eine gute, sogar exakte Startmatrix vor, die andererseits keinem vollständigen Korrekturzyklus unterzogen wird. Dagegen ist die Situation in der Strukturoptimierung völlig anders. Erstens wird zu Beginn der Berechnung auf die aufwendige Bestimmung der Hesse-Matrix verzichtet, weshalb meist mit der Einheitsmatrix begonnen werden muß. Zweitens ist die Zahl der Unbekannten relativ gering, und es können auch "unterwegs" keine exakten Hesse-Matrizen bestimmt werden, so daß während einer Optimierung mehrere vollständige Korrekturzyklen durchlaufen werden. Dabei stellt es sich heraus, daß das Konvergenzverhalten der Quasi-Newton-Verfahren sehr sensibel auf Skalierungen der Variablen und der Hesse-Matrix bzw. der Zielfunktion reagiert. Ein Verhalten, das sich auch bei einer guten Startmatrix zeigen kann, jedoch erst nach einer gewissen Zahl von Iterationen auffällt. Im folgenden sollen deshalb die wichtigsten Konvergenzeigenschaften der Quasi-Newton-Verfahren an einem quadratischen Problem dargestellt werden, um daraus die wichtigsten Schlüsse zur Effizienzsteigerung ziehen zu können.

Es sei folgendes quadratisches Problem gegeben:

$$f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{b}^T \mathbf{x} \quad \rightarrow \min. \quad (2.34)$$

Nach Gleichung (2.23) lautet die iterative Formel für  $\mathbf{x}_{k+1}$ :

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{B}_k^{-1} \mathbf{g}_k \quad (2.35)$$

Da  $f(\mathbf{x})$  eine quadratische Funktion in  $\mathbf{x}$  ist, läßt sich das exakte Linienminimum explizit bestimmen, und es ergibt sich für  $\alpha_k$ :

$$\alpha_k = \frac{\mathbf{g}_k^T \mathbf{B}_k^{-1} \mathbf{g}_k}{\mathbf{g}_k^T \mathbf{B}_k^{-1} \mathbf{Q} \mathbf{B}_k^{-1} \mathbf{g}_k} \quad (2.36)$$

Mit der Fehlerfunktion ( $\mathbf{x}^* = -\mathbf{Q}^{-1} \mathbf{b}$  ist die Lösung):

$$E(\mathbf{x}) = \frac{1}{2} (\mathbf{x} - \mathbf{x}^*)^T \mathbf{Q} (\mathbf{x} - \mathbf{x}^*) \quad (2.37)$$

kann die relative Verbesserung in einem Schritt nach einigen Zwischenrechnungen ermittelt werden:

$$\Delta = \frac{E(\mathbf{x}_k) - E(\mathbf{x}_{k+1})}{E(\mathbf{x}_k)} = \frac{(\mathbf{g}_k^T \mathbf{B}_k^{-1} \mathbf{g}_k)^2}{(\mathbf{g}_k^T \mathbf{B}_k^{-1} \mathbf{Q} \mathbf{B}_k^{-1} \mathbf{g}_k) (\mathbf{g}_k^T \mathbf{Q}^{-1} \mathbf{g}_k)} \quad (2.38)$$

Mit den Transformationen  $\mathbf{T}_k = \mathbf{B}_k^{-\frac{1}{2}} \mathbf{Q} \mathbf{B}_k^{-\frac{1}{2}}$  und  $\mathbf{s}_k = \mathbf{B}_k^{-\frac{1}{2}} \mathbf{g}_k$  ergibt sich:

$$\Delta = \frac{(\mathbf{s}_k^T \mathbf{s}_k)^2}{(\mathbf{s}_k^T \mathbf{T}_k \mathbf{s}_k) (\mathbf{s}_k^T \mathbf{T}_k^{-1} \mathbf{s}_k)} \quad (2.39)$$

Gleichung (2.39) wiederum läßt sich mittels der Kantorowitsch-Ungleichung (siehe Anhang A1) nach unten abschätzen, wobei  $a_k$  und  $A_k$  der kleinste bzw. der größte Eigenwert der Matrix  $\mathbf{T}_k$  sind.

$$\Delta = \frac{(\mathbf{s}_k^T \mathbf{s}_k)^2}{(\mathbf{s}_k^T \mathbf{T}_k \mathbf{s}_k) (\mathbf{s}_k^T \mathbf{T}_k^{-1} \mathbf{s}_k)} \geq \frac{4a_k A_k}{(a_k + A_k)^2} \quad (2.40)$$

Durch die folgende Transformation der Matrix  $\mathbf{T}_k$  in die Matrix  $\mathbf{R}_k$  ändern sich die Eigenwerte nicht:

$$\mathbf{R}_k = \mathbf{B}_k^{-\frac{1}{2}} \mathbf{T}_k \mathbf{B}_k^{\frac{1}{2}} = \mathbf{B}_k^{-1} \mathbf{Q} \quad (2.41)$$

Daraus folgt, daß die Matrix  $\mathbf{B}_k^{-1} \mathbf{Q}$  dieselben Eigenwerte wie  $\mathbf{T}_k$  besitzt.

Damit errechnet sich der Wert der Fehlerfunktion aus  $x_{k+1}$ :

$$E_{k+1} \leq \left( \frac{A_k - a_k}{A_k + a_k} \right)^2 E_k \quad (2.42)$$

Das heißt, die Konvergenz ist dann am schnellsten, wenn der kleinste und der größte Eigenwert zusammenfallen. Dieses Ergebnis ist vom Konvergenzverhalten des Verfahrens des steilsten Abstiegs wohl bekannt. Für Quasi-Newton-Verfahren sind aber auch die Absolutwerte der Eigenwerte entscheidend, wie sich nachfolgend zeigen wird.

Die Matrix  $\mathbf{B}_k$  wird iterativ der Hesse-Matrix  $\mathbf{Q}$  angenähert. Es folgt daraus, daß die Matrix  $\mathbf{B}_k^{-1} \mathbf{Q}$  für  $k > 0$  auch immer den  $k$ -fachen Eigenwert eins besitzt. Mittels der Quasi-Newton-Bedingung (2.25) ist dies leicht gezeigt:

$$\begin{aligned} \mathbf{B}_k \mathbf{p}_{k-1} &= \mathbf{g}_k - \mathbf{g}_{k-1} = \mathbf{Q} \mathbf{p}_{k-1} \\ \mathbf{p}_{k-1} &= \mathbf{B}_k^{-1} \mathbf{Q} \mathbf{p}_{k-1} \end{aligned} \quad (2.43)$$

Offensichtlich ist  $\mathbf{p}_{k-1}$  ein Eigenvektor der Matrix  $\mathbf{B}_k^{-1} \mathbf{Q}$  mit dem Eigenwert eins.

Während der Iteration verändern sich die Eigenwerte der Matrix  $\mathbf{B}_k^{-1} \mathbf{Q}$  ständig. Waren alle Eigenwerte der Startmatrix  $\mathbf{B}_0^{-1} \mathbf{Q}$  (gewöhnlich ist  $\mathbf{B}_0^{-1} = \mathbf{I}$ ) ursprünglich stark unterschiedlich von eins, wird gemäß Gleichung (2.42) während der Iteration die Konver-

genz durch die Quasi-Newton-Korrektur in dem Maße verschlechtert, wie sich der ursprünglich größte bzw. kleinste Eigenwert von eins unterschied. Dabei spielt es keine Rolle, ob  $Q$  gut konditioniert ist oder nicht, da dieser Effekt auch dann auftritt, wenn alle Eigenwerte von  $Q$  gleich groß, aber ungleich eins sind. Abhilfe kann hier nur eine ausreichende Vorkonditionierung von  $Q$  schaffen, die zwei Dinge leisten sollte:

1. eine Konditionszahl  $A/a$ , die möglichst wenig größer als eins sein sollte,
2. eine gleichmäßige Verteilung der Eigenwerte um den Wert eins herum, d.h.  $1 \in [a, A]$ .

Die zweite Forderung läßt sich erfüllen, indem die Quasi-Newton-Bedingung (2.25) für  $B_k$  in dem Sinne angewendet wird, daß die durch einen Differenzschritt ermittelte tatsächliche Krümmung in Richtung  $p_k$  mit der vorhandenen Matrix  $B_k$  verglichen wird. Dafür wird der Skalierungsfaktor  $\gamma$  eingeführt:

$$\gamma \cdot p_k^T B_k p_k = q_k^T p_k \tag{2.44}$$

$$\gamma = \frac{q_k^T p_k}{p_k^T B_k p_k}$$

Mit diesem Faktor  $\gamma$  muß die Matrix  $B_k$  multipliziert werden, bevor sie in die Korrekturformel (DFP oder BFGS) zur Bestimmung von  $B_{k+1}$  eingeht. Luenberger [85] gibt dazu einen vollständigen, selbstskalierenden Algorithmus an. In Bild 2.10 sind die Verbesserungen zu erkennen, die sich bei Anwendung der Selbstskalierung für den bereits in Bild 2.9d gezeigten Iterationsverlauf ergeben.

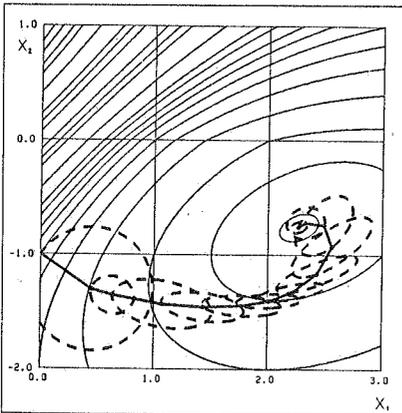


Bild 2.10a: Iterationsverlauf BFGS

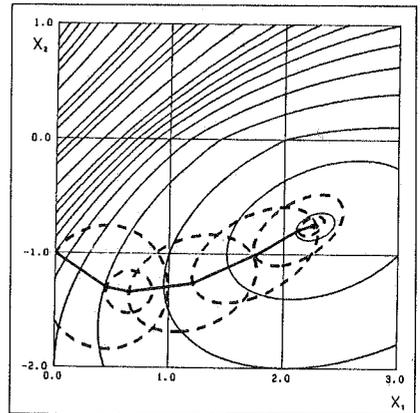


Bild 2.10b: BFGS, selbstskalierend

Eine andere Möglichkeit, die notwendige Verschiebung der Eigenwerte vorzunehmen, ist, anstelle der approximierten Hesse-Matrix  $\mathbf{B}_k$ , die Zielfunktion vor Beginn der Rechnung zu skalieren. Dies kann ohne Eingriffe in den Algorithmus geschehen, so daß dazu jeder in der Lage sein kann, wenn kein selbstskalierender Algorithmus zur Verfügung steht. Dabei wird angenommen, daß sich während der Iteration die Eigenwerte der Matrix  $\mathbf{B}_k^{-1} \mathbf{Q}$  nicht mehr so stark ändern, daß alle größer bzw. kleiner als eins werden können. Da die Iteration mit  $\mathbf{B}_0 = \mathbf{I}$  begonnen wird, liegt als erste Suchrichtung die negative Gradienten der Zielfunktion vor ( $\mathbf{p}_k = -\mathbf{g}_k$ ). Mit einem kleinen Tastschritt der Länge  $\epsilon$  wird die Krümmung der Zielfunktion in Suchrichtung abgeschätzt, und es ergibt sich der Faktor  $\delta$ , mit dem die Zielfunktion skaliert werden muß:

$$\delta = \frac{(\mathbf{g}(\mathbf{x}_k + \epsilon \mathbf{p}_k) - \mathbf{g}_k)^T \mathbf{g}_k}{\epsilon \mathbf{g}_k^T \mathbf{g}_k} \quad (2.45)$$

Die erste Forderung läßt sich durch eine weitere geeignete Skalierung der Zielfunktion erreichen, denn nach Skalierung mit (2.40) bzw. (2.45) bestimmt jetzt noch die Differenz ( $A - a$ ) zwischen dem größten und dem kleinsten Eigenwert, d.h. die Kondition der tatsächlichen Hesse-Matrix, die Konvergenzgeschwindigkeit. Diese Skalierung erfolgt indirekt durch eine Variablentransformation:

$$\mathbf{x} = \mathbf{W} \cdot \mathbf{y} \quad (2.46)$$

und damit:

$$\begin{aligned} f &= \frac{1}{2} \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{b}^T \mathbf{x} \\ &= \frac{1}{2} \mathbf{y}^T \mathbf{W}^T \mathbf{Q} \mathbf{W} \mathbf{y} + \mathbf{b}^T \mathbf{W} \mathbf{y} \end{aligned} \quad (2.47)$$

oder: 
$$f = \frac{1}{2} \mathbf{y}^T \hat{\mathbf{Q}} \mathbf{y} + \hat{\mathbf{b}}^T \mathbf{y}$$

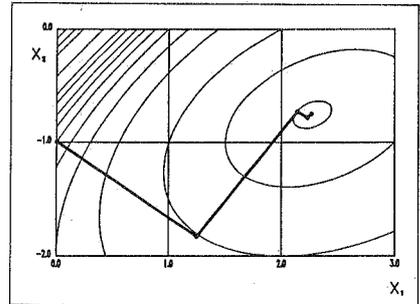
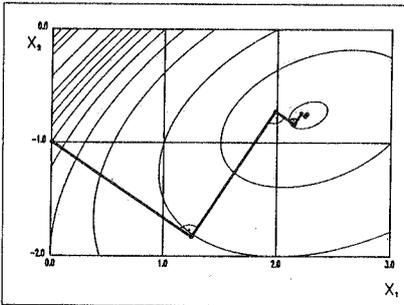
mit 
$$\hat{\mathbf{Q}} = \mathbf{W}^T \mathbf{Q} \mathbf{W} \quad \text{und} \quad \hat{\mathbf{b}} = \mathbf{W}^T \mathbf{b}$$

Zweckmäßigerweise ist  $\mathbf{W}$  eine Diagonalmatrix, deren Koeffizienten sich am vorliegenden Problem orientieren. Als eine grobe Regel, mit der aber gute Erfahrungen gewonnen wurden, können die Kehrwerte der Variablen am Startpunkt genommen werden. Über Erfahrungen mit den verschiedenen Skalierungen im Zusammenhang mit dem SQP-Verfahren wird in [13] sowie in den Abschnitten 2.8.6.4 und 6.1 berichtet. Die Verfahren werden dort zur Optimierung von Stabtragwerken bzw. zur Formfindung von Minimalflächen eingesetzt.

## 2.8.3 Ein Verfahren der konjugierten Gradienten für beschränkte Probleme mit einfachen Variablenrestriktionen

### 2.8.3.1 Allgemeines über Gradientenverfahren

Ursprünglich wurde das Verfahren der konjugierten Gradienten zur iterativen Lösung von großen linearen Gleichungssystemen eingeführt. Dem entspricht die Lösung eines unbeschränkten quadratischen Minimierungsproblems. Gegenüber der Methode des steilsten Abstiegs als des einfachsten Gradientenverfahrens berücksichtigt die Methode der konjugierten Gradienten bei quadratischen Problemen Krümmungsinformationen und konvergiert deshalb auch bei nicht ideal konditionierten,  $n$ -dimensionalen, quadratischen Problemen innerhalb von  $n$  Schritten. Bild 2.11 zeigt den unterschiedlichen Verlauf der Iterationen der beiden Verfahren für ein nicht quadratisches Problem (vergleiche mit Bild 2.9c). Dabei ist der Aufwand, in jedem Schritt eine Suchrichtung zu bestimmen, die konjugiert zur letzten sein soll, vernachlässigbar gering. Ein weiterer Vorteil des Verfahrens ist, daß innerhalb eines Zyklus von  $n$  Schritten die generierten Suchrichtungen linear unabhängig sind. Erweiterungen für nichtlineare Probleme wurden von Fletcher/Reeves und Polak und Ribière vorgeschlagen. Der Methode der konjugierten Gradienten wird große Effektivität bescheinigt, und sie gehört zweifellos zu den robusten Verfahren. Insbesondere eignet sie sich auch zur Lösung großer Probleme. Die Konvergenzeigenschaften sind weniger komplex als die der Quasi-Newton-Verfahren (insbesondere nicht so skalierungsanfällig), jedoch von einer größeren Rate.



**Bild 2.11a:** Methode des steilsten Abstiegs **Bild 2.11b:** konj. Gradientenverfahren

Ausgehend von folgender quadratischer Zielfunktion:

$$f(x) = \frac{1}{2}x^T Q x + b^T x \quad (2.48)$$

mit positiv definiten Matrix  $\mathbf{Q}$ , wird bei gegebener Suchrichtung  $\mathbf{p}_k$  und gegebenem Startwert  $\mathbf{x}_k$  das Linienminimum ermittelt. Dies ist für quadratische Probleme explizit bestimmbar.

$$f(\mathbf{x}_k + a \mathbf{p}_k) = f(\mathbf{x}_k) + a \mathbf{g}_k^T \mathbf{p}_k + \frac{1}{2} a^2 \mathbf{p}_k^T \mathbf{Q} \mathbf{p}_k \quad (2.49)$$

mit:  $\mathbf{g}_k = \nabla_x f(\mathbf{x}_k) = \mathbf{Q} \mathbf{x}_k + \mathbf{b}$

$$\nabla_a f(\mathbf{x}_k + a \mathbf{p}_k) = \mathbf{g}_k^T \mathbf{p}_k + a \mathbf{p}_k^T \mathbf{Q} \mathbf{p}_k$$

$$\rightarrow a_m = -\frac{\mathbf{g}_k^T \mathbf{p}_k}{\mathbf{p}_k^T \mathbf{Q} \mathbf{p}_k} \quad (2.50)$$

und:  $\mathbf{x}_{k+1} = \mathbf{x}_k + a_m \mathbf{p}_k \quad (2.51)$

Die neue Suchrichtung  $\mathbf{p}_{k+1}$  ergebe sich als eine lineare Kombination aus der alten und der Gradienten an der Stelle  $\mathbf{x}_{k+1}$ . Weiter seien die beiden Suchrichtungen  $\mathbf{p}_k$  und  $\mathbf{p}_{k+1}$  konjugiert zueinander. Damit kann  $\mathbf{p}_{k+1}$  bestimmt werden.

$$\mathbf{p}_{k+1} = -\mathbf{g}_{k+1} + \beta_k \mathbf{p}_k \quad ; \quad \mathbf{g}_{k+1} = \nabla_x f(\mathbf{x}_{k+1})$$

$$\mathbf{p}_{k+1}^T \mathbf{Q} \mathbf{p}_k = 0$$

$$\beta_k = \frac{\mathbf{g}_{k+1}^T \mathbf{Q} \mathbf{p}_k}{\mathbf{p}_k^T \mathbf{Q} \mathbf{p}_k} \quad (2.52)$$

Für  $k = 0$  ist  $\mathbf{p}_k = -\mathbf{g}_k$ , d.h. der erste Schritt ist ein Schritt des steilsten Abstiegs.

Eine wichtige Erweiterung des Verfahrens geht auf Fletcher und Reeves zurück. Anstelle der Gleichung (2.52) schreiben sie für  $\beta_k$ :

$$\beta_k = \frac{\mathbf{g}_{k+1}^T \mathbf{g}_{k+1}}{\mathbf{g}_k^T \mathbf{g}_k} \quad (2.53)$$

Es kann gezeigt werden, daß mit exaktem Line Search die Gleichungen (2.52) und (2.53) für quadratische Probleme identisch sind. Jedoch ist (2.53) leicht für beliebige nichtlineare Probleme erweiterbar, da kein Gebrauch von der Hesse-Matrix mehr gemacht wird. Die richtige Schrittweite muß dann allerdings in einem line search ermittelt werden. Da keine gute Schätzung der Schrittweite vorliegt, wie das bei den Quasi-Newton-Verfahren der Fall ist, kommt hier nur ein "genauer" line search in Frage, der sowohl interpoliert als auch extrapoliert. Dieses Vorgehen benötigt nur  $3n$  Speicherplätze ( $\mathbf{x}_k$ ,  $\mathbf{g}_k$ ,  $\mathbf{p}_k$ ) und ist deshalb besonders für große Probleme geeignet.

Von Polak und Ribière stammt folgende Formel, die für quadratische Probleme ebenfalls mit Gleichung (2.52) identisch ist, da  $\mathbf{g}_k^T \mathbf{g}_{k+1} = 0$  [85].

$$\beta_k = \frac{(\mathbf{g}_{k+1} - \mathbf{g}_k)^T \mathbf{g}_{k+1}}{\mathbf{g}_k^T \mathbf{g}_k} \quad (2.54)$$

Luenberger [85] zeigt eine interessante Gemeinsamkeit dieses Verfahrens mit der sogenannten "BFGS-Methode ohne Gedächtnis".

### 2.8.3.2 Erweiterung für einfache Variablen-Restriktionen

Optimierungsaufgaben mit einfachen Variablenrestriktionen können in der nichtlinearen Programmierung als Teile eines übergeordneten Verfahrens entstehen. Solche übergeordnete Verfahren sind z.B. die duale Methode von Uzawa oder die Multiplikatoren-methode.

Das Verfahren der konjugierten Gradienten kann für die Lösung dieser Teilaufgabe leicht erweitert werden. Damit zählt das Verfahren zu den primären Methoden, die den zulässigen Bereich des Problems nie verlassen. Dabei werden die Suchrichtungen so transformiert, daß sie diese Bedingung erfüllen. Dem line search kommt dabei die zusätzliche Aufgabe zu, das Verlassen des zulässigen Bereiches zu verhindern. Damit ist das Problem mit den Verfahren der konjugierten Gradienten lösbar.

Schranken sind lineare Nebenbedingungen. Sie haben eine besondere Form:

$$\begin{aligned} \text{untere Schranken:} & \quad -\mathbf{I} \mathbf{x} + \mathbf{x}_L \leq \mathbf{0} \\ \text{obere Schranken:} & \quad \mathbf{I} \mathbf{x} - \mathbf{x}_U \leq \mathbf{0} \end{aligned} \quad (2.55)$$

Die Matrix  $\hat{\mathbf{A}}$  der aktiven Nebenbedingungsgradienten (siehe Gl. (2.3)) ist somit eine Diagonalmatrix mit den Werten + 1 oder - 1, je nachdem die untere oder obere Schranke aktiv ist. Beide können für ein nicht entartetes Problem nie gleichzeitig aktiv sein. Gemäß der Orthogonalitätsbedingung:

$$\hat{\mathbf{A}} \mathbf{Z} = \mathbf{0} \quad (2.5)$$

ergibt sich  $\mathbf{Z}$  (die Matrix der Basisvektoren des Tangentialraumes) aus der Einheitsmatrix  $\mathbf{I}$ , aus der diejenigen Spalten herausgestrichen werden, deren zugehörige Variablen auf Schranken sitzen und für die keine Tendenz erkennbar ist, daß sie sich von dort

wegbewegen wollen. Diese Variablen sind durch folgende Beziehung charakterisiert (siehe Abschnitt 2.7.1.2):

$$-\nabla_{x_i} f(\mathbf{x}) \cdot \mathbf{a}_i \geq 0 \quad (2.56)$$

mit:

$$\begin{aligned} \mathbf{a}_i &= -1 && \text{für untere Schranken} \\ \mathbf{a}_i &= 1 && \text{für obere Schranken} \end{aligned}$$

Mit den Matrizen  $\hat{\mathbf{A}}$  und  $\mathbf{Z}$  läßt sich die Transformationsmatrix  $\mathbf{P}$  bestimmen, die das Problem in den von  $\mathbf{Z}$  aufgespannten Tangentialraum abbildet (Gln. 2.6/2.7):

$$\mathbf{Z} \cdot \mathbf{p}_Z = \left( \mathbf{I} - \hat{\mathbf{A}}^T (\hat{\mathbf{A}} \hat{\mathbf{A}}^T)^{-1} \hat{\mathbf{A}} \right) \mathbf{p} \quad (2.57)$$

$$\mathbf{P} = \left( \mathbf{I} - \hat{\mathbf{A}}^T (\hat{\mathbf{A}} \hat{\mathbf{A}}^T)^{-1} \hat{\mathbf{A}} \right) \quad (2.58)$$

$\mathbf{p}$  ... Suchrichtung im  $\mathbb{R}^n$   
 $\mathbf{p}_Z$  ... Suchrichtung im Tangentialraum

Das Problem kann nun im erweiterten Tangentialraum gelöst werden, wobei beachtet werden muß, daß die Schranken der freien Variablen eingehalten werden. Damit kann der folgende erweiterte Algorithmus der Methode der konjugierten Gradienten angegeben werden:

1. Setze die Startwerte:  
 $\mathbf{k} = 0$ ,  $\mathbf{x}_0$ ,  $f_0 = f(\mathbf{x}_0)$ ,  $\mathbf{g}_0 = \nabla f(\mathbf{x}_0)$ ,  $\mathbf{p}_0 = -\mathbf{g}_0$
2. Gehe nach 10.) und bestimme die Transformationsmatrix  $\mathbf{P}_0$ , den aktiven Satz der Nebenbedingungen  $J_k$  und die maximal zulässige Schrittweite  $\alpha_{\max}$ .  
 Transformiere die Suchrichtung:  $\mathbf{p}_0 := \mathbf{P}_0 \cdot \mathbf{p}_0$ .
3. Prüfe die Konvergenz.
4. Führe einen line search durch, Startpunkt  $\mathbf{x}_k$ , Suchrichtung  $\mathbf{p}_k$ ; bestimme die Schrittweite zum Linienminimum  $\alpha \leq \alpha_{\max}$ .
5. Korrigiere Variablen, Funktionswerte und Ableitungen:  
 $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha \cdot \mathbf{p}_k$   
 $f_{k+1} = f(\mathbf{x}_{k+1})$   
 $\mathbf{g}_{k+1} = \nabla f(\mathbf{x}_{k+1})$
6. Gehe nach 10.) und bestimme die Transformationsmatrix  $\mathbf{P}_{k+1}$ , den aktiven Satz der Nebenbedingungen  $J_{k+1}$  und die maximale zulässige Schrittweite  $\alpha_{\max}$ .

7. Ist  $J_k \neq J_{k+1}$  oder  $k = n$ , setze  $\mathbf{p}_{k+1} = -\mathbf{P}_{k+1} \mathbf{g}_{k+1}$ .  
Das entspricht einem Schritt des Verfahrens des steilsten Abstiegs (spacer step).  
Sonst, bestimme die neue Suchrichtung nach der Formel von Fletcher/Reeves (oder Polak/Ribière):

$$\mathbf{p}_{k+1} = -\mathbf{P}_{k+1} \mathbf{g}_{k+1} + \frac{\mathbf{g}_{k+1}^T \mathbf{P}_{k+1}^T \mathbf{P}_{k+1} \mathbf{g}_{k+1}}{\mathbf{g}_k^T \mathbf{P}_k^T \mathbf{P}_k \mathbf{g}_k} \mathbf{p}_k$$

8. Im Falle der "konjugierten" Suchrichtung:  
überprüfe die Zulässigkeit der neuen Richtung.

$$\tilde{J}_{k+1} = \left\{ i \mid (x_{Li} = x_{(k+1)i} \wedge p_{(k+1)i} < 0) \vee (x_{Ui} = x_{(k+1)i} \wedge p_{(k+1)i} > 0) ; i = 1, \dots, n \right\}$$

Ist  $\tilde{J}_{k+1} \neq J_{k+1}$ , setze  $\mathbf{p}_{k+1} = -\mathbf{P}_{k+1} \mathbf{g}_{k+1}$  (steilster Abstieg).

9.  $k = k+1$ , gehe nach 3.).

10. Bestimme den aktiven Satz:

$$J_k = \left\{ i \mid (x_{Li} = x_{ki} \wedge g_{ki} \geq 0) \vee (x_{Ui} = x_{ki} \wedge g_{ki} \leq 0) ; i = 1, \dots, n \right\}$$

Bestimme die Transformationsmatrix  $\mathbf{P}_k$ :

$$P_{ki} = \begin{cases} 0 & | i \in J_k \\ 1 & | i \notin J_k \end{cases}$$

Bestimme max. mögliche Schrittweite  $\alpha_{\max}$  bis zur nächsten Schranke:

$$\gamma_i = \begin{cases} (x_{Li} - x_{ki}) / p_{ki} & | i \notin J_k \wedge p_{ki} < 0 \\ (x_{Ui} - x_{ki}) / p_{ki} & | i \notin J_k \wedge p_{ki} > 0 \\ \infty & | \text{sonst} \end{cases}$$

$$\alpha_{\max} = \left\{ \min (\gamma_i) \mid i = 1, \dots, n \right\}$$

## 2.8.4 Penalty- und Barriere-Methoden

Diese Verfahren approximieren beschränkte Probleme durch unbeschränkte. Beim Penalty-Verfahren wird dabei zur Zielfunktion ein Term hinzuaddiert, der die Verletzung von Nebenbedingungen bestraft. Dagegen verhindert ein solcher Strafterm beim Barriere-Verfahren, daß der zulässige Bereich des Problems verlassen werden kann. Dabei werden Punkte im Inneren des zulässigen Bereichs solchen an den Grenzen vorgezogen. Wegen dieser grundsätzlich unterschiedlichen Charakteristika beider Ansätze, wird die Penalty-Methode auch Verfahren der äußeren und die Barriere-Methode Verfahren der inneren Straffunktion genannt. In beiden Fällen ist der Betrag der Straffunktion mit dem Wert eines Penalty-Faktors verbunden, der, wenn er gegen unendlich strebt, dafür sorgt, daß das eigentliche Problem zunehmend genauer approximiert wird. Andererseits nimmt mit zunehmendem Penalty-Faktor die Kondition des approximierten, unbeschränkten Problems ab, weshalb bei der Lösung des Ersatzproblems numerische Schwierigkeiten auftreten können. Bei den Barriere-Methoden kommt hinzu, daß die Ersatzfunktion außerhalb des zulässigen Bereichs nicht definiert ist und deshalb für den line search noch besondere Vorkehrungen getroffen werden müssen, damit garantiert jede Zwischenlösung im Inneren des zulässigen Bereichs bleibt. Auch können Gleichheitsbedingungen mit Barriere-Verfahren nicht erfaßt werden. Andererseits nähern sich Penalty-Verfahren vom unzulässigen Bereich her der Lösung. Deshalb können Zwischenlösungen erst nach einer Skalierung an den Rand des zulässigen Bereichs verwendet werden, die die Zulässigkeit garantiert. Insgesamt sind die beiden Verfahren einfach zu handhaben und machen wenig Gebrauch von der Theorie der nichtlinearen Programmierung. (Zum Beispiel gibt es keine Anforderungen an besondere Qualifikationen der Nebenbedingungen, wie dies bei den Kuhn-Tucker-Bedingungen der Fall ist.) Deshalb haben sie sich auch in der praktischen Anwendung bewährt. Ein bekanntes Verfahren ist das SUMT-Verfahren (Sequential Unconstrained Minimization Technique) von Fiacco und McCormick, bei dem eine Folge von approximierten Unterproblemen mit anwachsendem Penalty-Faktor gelöst wird. Auf diese Weise können eventuelle numerische Schwierigkeiten beherrscht werden, während sich die Lösung ständig dem exakten Ergebnis nähert.

### 2.8.4.1 Penalty-Verfahren

Das ursprüngliche Problem:

$$\begin{aligned} f(\mathbf{x}) &\rightarrow \min \\ h_j(\mathbf{x}) &= 0 ; \quad j = 1, \dots, m_e \\ g_j(\mathbf{x}) &\leq 0 ; \quad j = m_e + 1, \dots, m \end{aligned} \quad (\text{NLP})$$

wird ersetzt durch ein unbeschränktes Problem der Form:

$$f_p(\mathbf{x}, c) = f(\mathbf{x}) + c \cdot P(\mathbf{x}) \rightarrow \min \quad (2.59)$$

$P(\mathbf{x})$  ... Penalty-Funktion

$c$  ... Penalty-Faktor

Die Penalty-Funktion  $P(\mathbf{x})$  wird in den aktiven Nebenbedingungen dargestellt:

$$P(\mathbf{x}) = \gamma(\mathbf{g}^+(\mathbf{x})) \quad (2.60)$$

mit:  $\mathbf{g}_j^+(\mathbf{x}) = h_j(\mathbf{x}) ; j = 1, \dots, m_e$

$$\mathbf{g}_j^+(\mathbf{x}) = \max[0, g_j(\mathbf{x})] ; j = m_e + 1, \dots, m \quad (2.61)$$

Die Funktion  $\gamma$  soll nur außerhalb des zulässigen Bereichs für eine Bestrafung sorgen, deshalb muß für sie gelten (für bel. Werte  $\mathbf{y}$ ):

$$\begin{aligned} \gamma(\mathbf{y}) &= 0 ; \text{ falls } \mathbf{y} = \mathbf{0} \\ \gamma(\mathbf{y}) &> 0 ; \text{ falls } \mathbf{y} \geq \mathbf{0}, \text{ aber } \mathbf{y} \neq \mathbf{0} \end{aligned} \quad (2.62)$$

Für die Penalty-Funktion sind folgende Ausdrücke gebräuchlich:

a) Nicht differenzierbare  $L_1$ -Penalty-Funktion:

$$P(\mathbf{x}) = \sum_{j=1}^m |g_j^+(\mathbf{x})| \quad (2.63)$$

b) Quadratische Penalty-Funktion:

$$P(\mathbf{x}) = \frac{1}{2} \sum_{j=1}^m (g_j^+(\mathbf{x}))^2 \quad (2.64)$$

c)

$$P(\mathbf{x}) = \mathbf{g}^+(\mathbf{x})^T \mathbf{G} \mathbf{g}^+(\mathbf{x}) \quad \mathbf{G} \text{ positiv definit, symmetrisch} \quad (2.65)$$

d) Eine Verallgemeinerung von b):

$$\text{Gleichheitsbedingungen: } P(\mathbf{x}) = \left[ \sum_{j=1}^{m_e} (h_j(\mathbf{x}))^2 \right]^{\frac{\epsilon}{2}} \quad (2.66)$$

$$\text{Ungleichheitsbedingungen: } P(\mathbf{x}) = \sum_{j=m_e+1}^m (g_j^+(\mathbf{x}))^\epsilon$$

mit  $\epsilon > 0$

Eine alternative Darstellung ist auch mit jeweils einem Penalty-Faktor für jede Nebenbedingung möglich. Wie ein beschränktes Problem umgeformt wird, ist im Bild 2.12 für das Beispiel des zweiständigen Fachwerks Fall b) (Bild 2.2b) gezeigt. Es wird deutlich, wie sich die numerische Struktur des Problems mit zunehmendem Penalty-Faktor verschlechtert.

Die nicht differenzierbare  $L_1$ -Penalty-Funktion (2.63) wird auch "exakte Penalty-Funktion" genannt, da sie die einzige ist, die für einen Penalty-Faktor  $c < \infty$  die richtige Lösung liefern kann; selbstverständlich für den Preis, daß das Ersatzproblem aufgrund der diskontinuierlichen Ableitungen für viele der üblichen Verfahren nicht mehr zugänglich ist. Diese Funktion hat aber als Abstiegsfunktion (merit function) eine wichtige Bedeutung im Zusammenhang mit den Lagrange-Methoden, wie auch die anderen Penalty-Funktionen. Die quadratische Penalty-Funktion hat auch Bedeutung für die Multiplikatorenmethoden bei der Definition der erweiterten Lagrange-Funktion (siehe Absatz 2.8.5.3).

Für die quadratische Penalty-Funktion lassen sich Approximationen der Lagrange-Multiplikatoren durch einen Vergleich der Ersatzfunktion mit der Lagrange-Funktion bestimmen. Die Stationärbedingungen der beiden Funktionen lauten:

$$\begin{aligned}\nabla f_p(\mathbf{x}, c) &= \nabla f + c \nabla \mathbf{g}^{+T} \mathbf{g}^+ \\ \nabla L(\mathbf{x}, \lambda, \mu) &= \nabla f + \nabla \mathbf{g}^T \lambda + \nabla \mathbf{h}^T \mu\end{aligned}\tag{2.67}$$

Daraus folgen die Lagrange-Multiplikatoren:

$$\begin{aligned}\mu_j &= c \cdot g_j^+ \quad ; \quad j = 1, \dots, m_e \\ \lambda_j &= c \cdot g_j^+ \quad ; \quad j = m_e + 1, \dots, m\end{aligned}\tag{2.68}$$

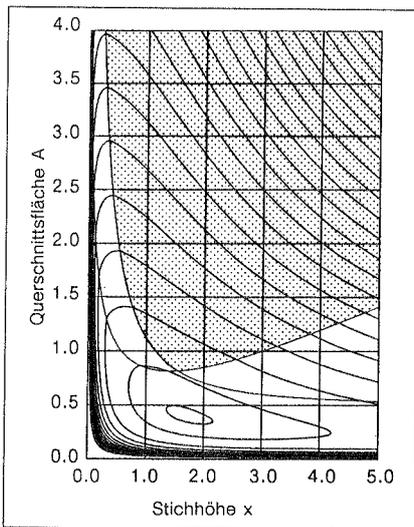


Bild 2.12a: Penaltyfaktor  $c = 0,5$

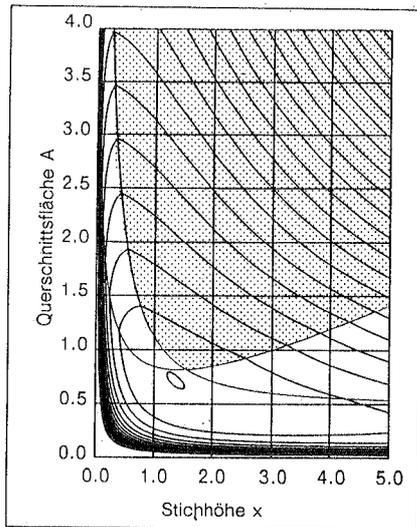


Bild 2.12b: Penaltyfaktor  $c = 1,0$

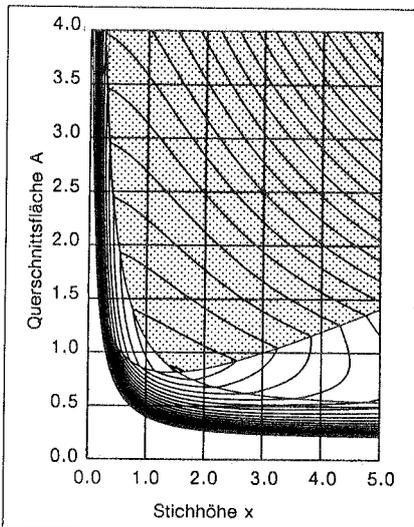


Bild 2.12c: Penaltyfaktor  $c = 10,0$

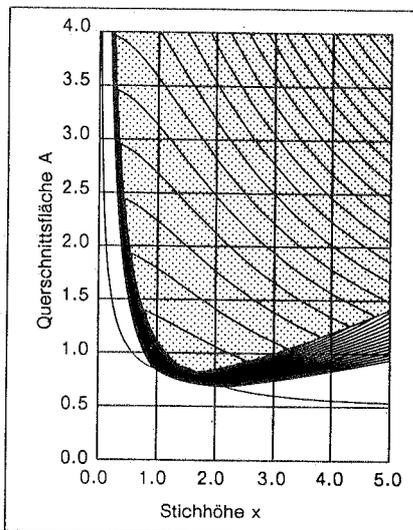


Bild 2.12d: Penaltyfaktor  $c = 100,0$

Bild 2.12:  $L_1$ -Penaltyfunktion angewendet für das zweistöbige Fachwerk, Fall b)

### 2.8.4.2 Barriere-Verfahren

Hier wird das ursprüngliche Problem (das keine Gleichheitsbedingungen haben darf!) durch folgendes ersetzt:

$$f_p(\mathbf{x}, c) = f(\mathbf{x}) + \frac{1}{c} B(\mathbf{x}) \rightarrow \min \quad (2.69)$$

$B(\mathbf{x})$  ... Barriere - Funktion

$c$  ... Penalty - Faktor

Dabei sind für  $B(\mathbf{x})$  folgende Ansätze üblich:

$$\text{a) } B(\mathbf{x}) = \sum_{j=1}^m \left( -\frac{1}{g_j(\mathbf{x})} \right) \quad (2.70)$$

$$\text{b) } B(\mathbf{x}) = \sum_{j=1}^m \frac{1}{(-g_j(\mathbf{x}))^\epsilon} \quad ; \quad \epsilon \in \mathbb{R} > 0 \quad (2.71)$$

Meist werden für  $\epsilon$  die Werte 1 (dann ergibt sich (a)) oder 2 gewählt.

$$\text{c) } B(\mathbf{x}) = - \sum_{j=1}^m \ln(g_j(\mathbf{x})) \quad (2.72)$$

Für den Fall a) lassen sich analog zum Vorgehen bei den Penalty-Verfahren Approximationen der Lagrange-Multiplikatoren bestimmen:

$$\lambda_j = \frac{1}{c g_j^2} \quad : \quad j = 1, \dots, m \quad (2.73)$$

Das approximierte Beispiel des zweiständigen Fachwerks Fall b) mit der Barriere-Funktion vom Typ a), Gleichung (2.70), ist im Bild 2.13 dargestellt. Auch hier sei auf die mit zunehmendem Penalty-Faktor abnehmende Qualität des Ersatzproblems hingewiesen.

Wie bei den primären Verfahren muß auch bei der Barriere-Methode der Startpunkt der Iteration innerhalb des zulässigen Bereichs liegen. Erfüllt der vorliegende Startpunkt diese Bedingung nicht, geben Künzi et al. [81] ein interessantes Verfahren an, wie mit Hilfe einer Barriere-Funktion ein zulässiger Punkt gefunden werden kann.

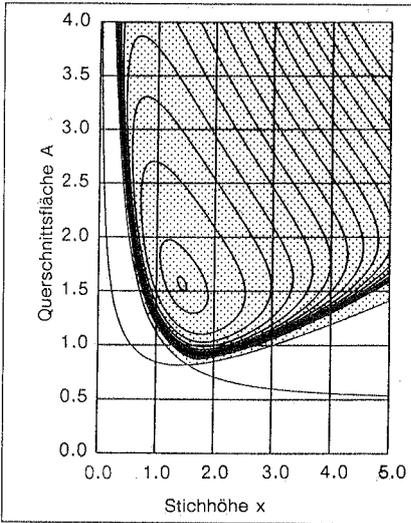


Bild 2.13a: Penaltyfaktor  $c = 0,5$

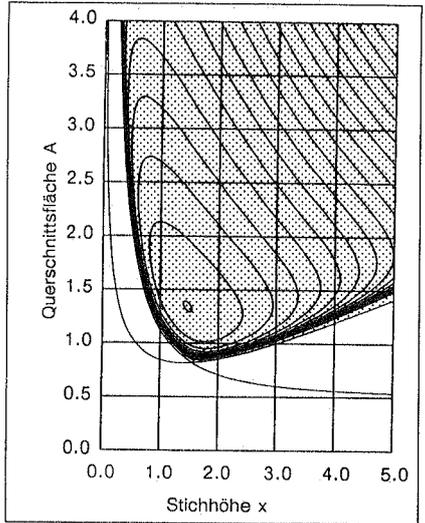


Bild 2.13b: Penaltyfaktor  $c = 1,0$

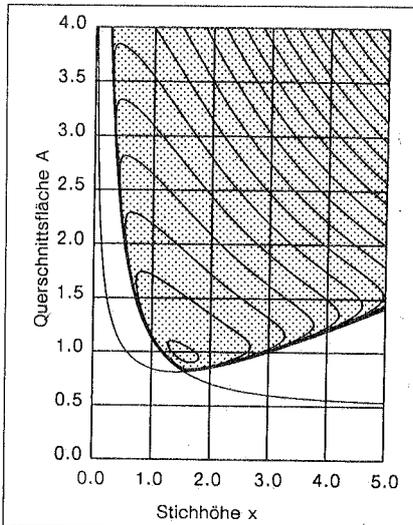


Bild 2.13c: Penaltyfaktor  $c = 10,0$

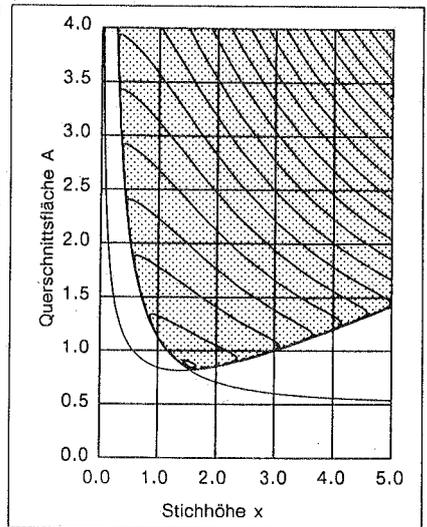


Bild 2.13d: Penaltyfaktor  $c = 100,0$

Bild 2.13: Barriere-Funktion (2.70) angewendet für das zwei-ständige Fachwerk, Fall b)

### 2.8.4.3 Evolutionsstrategie als zufallsgelenkte Barriere-Methode nullter Ordnung

Ein Verfahren aus der Klasse der direkten Suchmethoden, das von seinen Verfassern sehr ungewöhnlich begründet wird und deshalb aus dem bisher beschriebenen mathematischen Rahmen fällt, ist die sogenannte Evolutionsstrategie. Der Grundgedanke geht auf Rechenberg [113] zurück und wurde von Schwefel [123] als mathematischer Algorithmus formuliert. Die Idee dieses Verfahrens ist, mit den aus der Darwinschen Evolutionstheorie bekannten wichtigsten Evolutionsfaktoren wie Mutation, Selektion und Populationsstärke ein Konzept für die numerische Optimierung zu entwickeln. Mutation steht für Parametervariation, Selektion für Aussonderung der Zwischenlösungen mit schlechteren Werten der Zielfunktion als der bis dahin besten Lösung und Population für eine Menge an Punkten im Entwurfsraum, an denen die Zielfunktion auszuwerten ist. Eine Grundpopulation konstanter Stärke  $\mu$  erzeugt durch Mutation  $\lambda$  Nachkommen. Durch Selektion, d.h. Überleben der  $\mu$  Besten, entsteht eine neue Grundpopulation, und das Spiel beginnt von neuem. Werden aus der Gesamtmenge aus  $\mu$  Eltern und  $\lambda$  Nachkommen die neuen  $\mu$  Eltern bestimmt, spricht man von  $(\mu + \lambda)$  - Strategie, sterben jedoch die Eltern aus, und die neue Grundpopulation wird nur aus den  $\mu$  besten der  $\lambda$  Nachkommen gebildet, so spricht man von  $(\mu, \lambda)$  - Strategie. Die einfachste Form der Evolutionsstrategie ist demnach die  $(1 + 1)$  - Strategie, von Rechenberg auch zweigliedrige Evolutionsstrategie genannt.

Die Mutation interpretiert Rechenberg als eine Zufallsentscheidung mit geringer Streuung der Parameteränderung. In den Worten der mathematischen Programmierung entspricht dies der Wahl eines Suchschrittes mit normal verteilter Richtung und Schrittweite. Im Vergleich zum Monte-Carlo-Verfahren, das als reines Zufallsverfahren in jedem Schritt den gesamten Entwurfsraum mit jeweils gleicher Wahrscheinlichkeit auswertet, bezieht sich die Evolutionsstrategie in ihren Entscheidungen immer auf die zuletzt berechnete Zwischenlösung. Deshalb zählt sie zu den sequentiellen Zufallsverfahren. Die Wahl der Streubreite ist jedoch ein Problem, denn die Forderung nach schneller Konvergenz fordert eine kleine, die Garantie der globalen Konvergenz (auch für multimodale Probleme!) jedoch eine große Streuung. Das Mittel Rechenbergs zur Erhöhung der globalen Konvergenzsicherheit bei gleichzeitig akzeptabler Konvergenzgeschwindigkeit ist die Erhöhung der Grundpopulation, d.h. der Anwendung sogenannter mehrgliedriger Strategien. Das entspricht der gleichzeitigen Ausführung mehrerer sequentieller Suchen in verschiedenen Richtungen, was nach Schwefel besonders für die  $(\mu, \lambda)$  - Strategie erfolgreich ist. Damit wird das Verfahren für die kommende Generation der Transputer interessant.

Andererseits ist die Evolutionsstrategie eine Barriere-Methode, d.h. das Verfahren bewegt sich innerhalb des zulässigen Bereichs und läßt keine Gleichheitsbedingungen zu. Die Barriere-Funktion entspricht im Grunde der nicht differenzierbaren Penalty-Funktion (2.63) mit unendlich großem Penalty-Faktor, weshalb die Evolutionsstrategie einerseits die exakte Lösung bestimmen und andererseits auch als Penalty-Verfahren gesehen werden kann. Die Barriere-Funktion lautet:

$$B(\mathbf{x}) = \begin{cases} \infty & \text{für ein } g_j(\mathbf{x}) > 0 ; j = 1, \dots, m \\ 0 & \text{für alle } g_j(\mathbf{x}) \leq 0 ; j = 1, \dots, m \end{cases} \quad (2.74)$$

Die (1 + 1) - Evolutionsstrategie ist wohl die populärste Version. Das Schema dieses Algorithmus ist folgendermaßen:

1. Setze die Startwerte (an einem zulässigen Startpunkt):

Variablen  $\mathbf{x}_0 \in \mathbb{R}^n$

Standardabweichung der Suchschritte  $\sigma$

Startwert der Ersatzfunktion  $\hat{f}_B = f_B(\mathbf{x}_0) = f(\mathbf{x}_0) + B(\mathbf{x}_0)$

Iterationszähler  $k = 0$

Erfolgszähler  $n_e = 0$

Zyklenzähler  $n_z = 0$

2.  $n_z = n_z + 1$
3. Konvergenzabfrage.
4. Bestimme Suchrichtung und Schrittweite:

$$z_k = \sigma_i \cdot \xi_i ; \quad \xi_i \dots \text{normalverteilte Zufallszahlen} \\ i = 1, \dots, n$$

5. Bestimme den nächsten zulässigen Testpunkt und werte die Ersatzfunktion  $f_B$  aus:

$$\mathbf{x}'_k = \mathbf{x}_k + z_k$$

$$\text{falls } f_B(\mathbf{x}'_k) \leq \hat{f}_B : \quad \begin{aligned} n_e &= n_e + 1 \\ \hat{f}_B &= f_B(\mathbf{x}'_k) \\ \mathbf{x}_{k+1} &= \mathbf{x}'_k \end{aligned}$$

6.  $\mathbf{x}_{k+1} = \mathbf{x}_k, k = k + 1$
7. Falls  $n_z < n$ , gehe nach 2.).
8. Schrittweitenadaption ("1/5-Erfolgsregel")

Ist während der letzten zehn Zyklen das Verhältnis der Erfolge zur Gesamtzahl der Versuche:

$$< \frac{1}{5} \quad \text{dann setze:} \quad \text{fac} = 0,85$$

$$= \frac{1}{5} \quad \text{fac} = 1,0$$

$$> \frac{1}{5} \quad \text{fac} = \frac{1}{0,85}$$

korrigiere die Standardabweichungen  $\sigma_i = \text{fac} \cdot \sigma_i; \quad i = 1, \dots, n$

9. Gehe nach 2.).

Wie man dem Schema entnehmen kann, leidet der Algorithmus unter folgenden Schwächen, weshalb er zur Lösung von Problemen mit kontinuierlich ableitbaren Funktionen nicht in Frage kommt:

- Das Verfahren benötigt sehr viele Testauswertungen, um die Schrittweiten zu adaptieren und kommt deshalb auch nur für wenige Variablen in Frage.
- Es gehen dabei keine höherwertigen Informationen ein.
- Die Standardabweichungen  $\sigma$  können sich nicht relativ zueinander verändern. Das heißt, das Verfahren hat eine fest eingestellte Metrikapproximation, weshalb das Problem gut konditioniert sein muß.

Diese Eigenschaften der Evolutionsstrategie sind auch am Iterationsverlauf bei der Lösung eines gut konditionierten Problems zu erkennen (Bild 2.14).

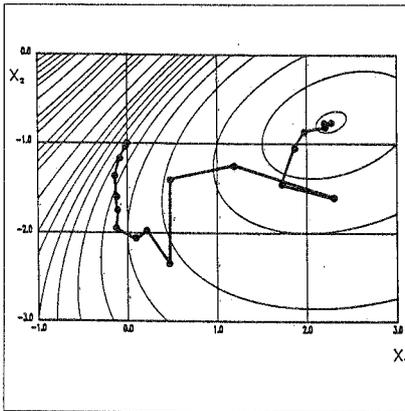


Bild 2.14a: Evolutionsstrategie

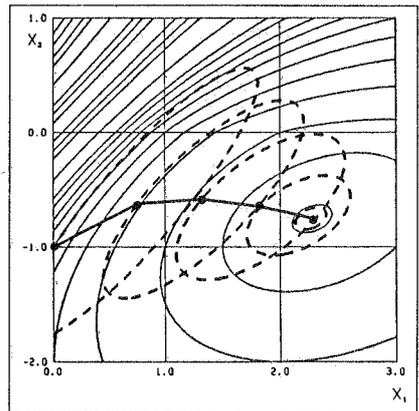


Bild 2.14b: Newton-Verfahren

Die Vorteile der Evolutionsstrategie sind die überaus einfache Anwendung, die Fähigkeit auch hochgradig diskontinuierliche Fragestellungen lösen zu können und die zwar geringe, jedoch vorhandene Wahrscheinlichkeit, daß das globale Optimum bestimmt wird. Unter diesen Voraussetzungen ist die Evolutionsstrategie auch für die Strukturoptimierung interessant [11], [70].

## 2.8.5 Duale Methoden

### 2.8.5.1 Allgemeines

Aufgrund der Sattelpunkteigenschaften (2.16) der Lagrange-Funktion kann die primäre Lösung  $\mathbf{x}^*$  eines beschränkten Problems durch eine einfach unbeschränkte Minimierung gefunden werden, sofern die optimalen Werte  $\lambda^*$  und  $\mu^*$  der Lagrange-Multiplikatoren bekannt sind. Diese Tatsache machen sich die dualen Methoden zunutze. Das Problem wird indirekt gelöst, indem hier die Lagrange-Multiplikatoren, die bei anderen Verfahren immer die zweite Rolle spielen, die unabhängigen Variablen darstellen und die primäre Lösung sich erst mit deren Vorgabe bestimmen läßt. Die Voraussetzungen sind aber, daß das Problem in der Umgebung der Lösung konvex ist (lokale Dualität). Wie andere Methoden auch liefern die dualen Methoden die globale Lösung, wenn alle Funktionen im  $\mathbb{R}^n$  konvex sind. Ein bekanntes duales Lösungsverfahren wurde von Uzawa angegeben (siehe Minoux [88]). Duale Verfahren haben in letzter Zeit eine große Bedeutung in der Strukturoptimierung gefunden. Es sind sehr effektive Optimierungsstrategien entwickelt worden, die die Ideen der dualen Methoden mit Approximationsverfahren und den besonderen Eigenschaften separabler Probleme verbinden [44], [135].

Gegeben sei das Standardproblem (NLP) der beschränkten Optimierung:

$$\begin{aligned}
 f(\mathbf{x}) &\rightarrow \min \\
 \mathbf{h}(\mathbf{x}) &= \mathbf{0} \\
 \mathbf{g}(\mathbf{x}) &\leq \mathbf{0} \\
 \mathbf{x} &\in S \subset \mathbb{R}^n
 \end{aligned}
 \tag{NLP}$$

Mit der Annahme, daß das Problem in der Umgebung der Lösung lokal konvex sei, daß also die Hesse-Matrix der Lagrange-Funktion im  $\mathbb{R}^n$  und nicht nur im Tangentialraum der aktiven Nebenbedingungen positiv definit sei, ist  $\mathbf{x}^*$  nicht nur eine Lösung von (NLP), sondern auch eine lokale Lösung des leicht beschränkten Problems:

$$\min_{\mathbf{x}} \left\{ f(\mathbf{x}) + \lambda^{*T} \mathbf{g}(\mathbf{x}) + \mu^{*T} \mathbf{h}(\mathbf{x}) \right\} \quad (2.75)$$

$$\lambda^* \geq \mathbf{0}$$

Da die Funktionen in der Umgebung der Lösung  $C_2$ -kontinuierlich sein sollen, ist für irgendwelche  $\lambda$  und  $\mu$ , die nahe an der Lösung  $\lambda^*$  und  $\mu^*$  sind, auch die primäre Lösung  $\mathbf{x}$  nahe an  $\mathbf{x}^*$ . Damit ist die Beziehung zwischen  $\mathbf{x}$  und  $\lambda, \mu$  kontinuierlich ableitbar, und es kann die duale Funktion definiert werden:

$$\begin{aligned} \phi(\lambda, \mu) &= \min_{\mathbf{x}} \left( f(\mathbf{x}) + \lambda^T \mathbf{g}(\mathbf{x}) + \mu^T \mathbf{h}(\mathbf{x}) \right) \\ &= f(\mathbf{x}(\lambda, \mu)) + \lambda^T \mathbf{g}(\mathbf{x}(\lambda, \mu)) + \mu^T \mathbf{h}(\mathbf{x}(\lambda, \mu)) \end{aligned} \quad (2.76)$$

mit den ersten und zweiten Ableitungen nach  $\lambda$  bzw.  $\mu$

$$\nabla_{\lambda} \phi = \mathbf{g}(\mathbf{x}(\lambda, \mu)) \quad ; \quad \nabla_{\mu} \phi = \mathbf{h}(\mathbf{x}(\lambda, \mu)) \quad (2.77)$$

und der Hesse-Matrix der Lagrange-Funktion  $F = F(\mathbf{x}(\lambda, \mu), \lambda, \mu)$  :

$$\nabla^2 \phi = \begin{bmatrix} -\nabla_{\mathbf{x}} \mathbf{g} F^{-1} \nabla_{\mathbf{x}} \mathbf{g}^T & -\nabla_{\mathbf{x}} \mathbf{g} F^{-1} \nabla_{\mathbf{x}} \mathbf{h}^T \\ -\nabla_{\mathbf{x}} \mathbf{h} F^{-1} \nabla_{\mathbf{x}} \mathbf{g}^T & -\nabla_{\mathbf{x}} \mathbf{h} F^{-1} \nabla_{\mathbf{x}} \mathbf{h}^T \end{bmatrix} \quad (2.78)$$

Das Originalproblem (NLP) ist äquivalent mit der Maximierung der dualen Funktion  $\phi(\lambda, \mu)$  bezüglich  $\lambda$  und  $\mu$ . Dies ist leicht verifiziert, da die Hesse-Matrix (2.78) der dualen Funktion offensichtlich negativ definit ist. Sind  $f$  und  $\mathbf{g}$  konvexe und  $\mathbf{h}$  affine Funktionen, dann ist die Lösung dieser Maximierungsaufgabe auch die globale Lösung.

In Bild 2.15 sind die Höhenlinien der Lagrange-Funktion des zweistöbigen Fachwerks (Bilder 2.1 und 2.2) für den Schnitt  $A = \text{konst} = 1/2 \sqrt{2}$  dargestellt. Als Nebenbedingung ist Staßfließen berücksichtigt (Fall a). Im Bild 2.15 ist die Spur der dualen Funktion  $\phi(\lambda)$  dargestellt. Mit alpinem Vokabular ausgedrückt, erklimmt sie den Sattelpunkt im Tobel der Lagrange-Funktion. Man könnte auch von einem "dualen Tal" sprechen. Diese Spur gibt die Orte aller Minimalstellen der Lagrange-Funktion bezüglich  $\mathbf{x}$  und festem  $\lambda$  an.

Eine alternative Darstellung macht die konkave Eigenschaft der dualen Funktion  $\phi(\lambda)$  deutlich. In Bild 2.16 sind mehrere Schnitte durch die Lagrange-Funktion parallel zur  $\mathbf{x}$ -Achse mit jedoch verschiedenen Werten für den Lagrange-Multiplikator dargestellt. Die Minima dieser Kurven sind Punkte der dualen Funktion, die ebenfalls eingezeichnet ist und an der Lösung ihr Maximum erreicht.

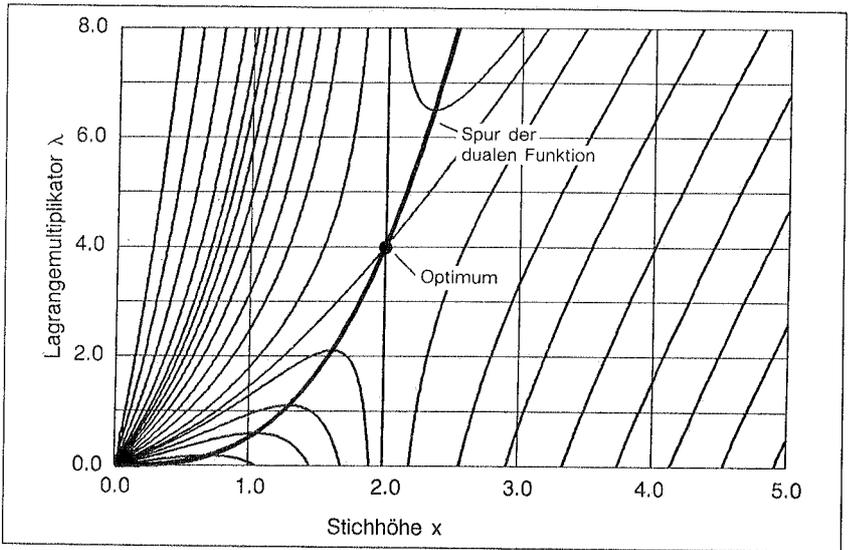


Bild 2.15: Spur der dualen Funktion

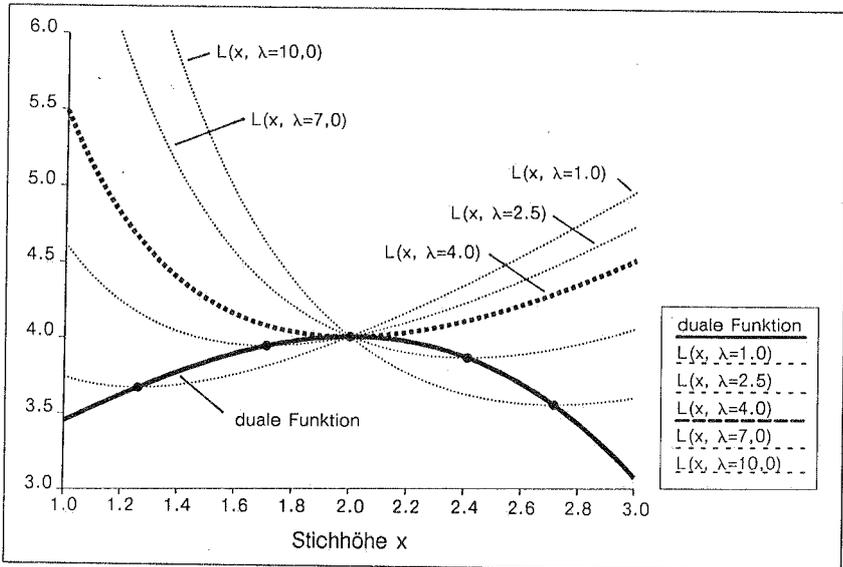


Bild 2.16: Schnitte im Primärraum der Lagrangefunktion

### 2.8.5.2 Lösung der dualen Aufgabe (Methode von Uzawa)

Die Lösung des dualen Problems zerfällt in zwei Teiloptimierungen:

1. Bestimme mit festen Lagrange-Multiplikatoren das Minimum der Lagrange-Funktion bezüglich  $x$ ; bestimme den Wert der dualen Funktion:

$$\phi(\lambda, \mu) = \min_x \left( f(x(\lambda, \mu)) + \lambda^T g(x(\lambda, \mu)) + \mu^T h(x(\lambda, \mu)) \right) \quad (2.79)$$

2. Maximiere die duale Funktion:

$$\max_{\lambda, \mu} \phi(\lambda, \mu) \quad \text{mit: } \lambda \geq \mathbf{0} \quad (2.80)$$

Aufgabe 1) ist dabei ein Unterproblem, das sehr oft gelöst werden muß, da es die Funktionsbestimmung der dualen Funktion  $\phi$  ist. Deshalb ist dieses Vorgehen nur ratsam, wenn Aufgabe 1) eine besondere Struktur hat, die es erlaubt, daß das Problem schnell und effizient gelöst werden kann. Das ist z.B. bei sogenannten separablen Problemen der Fall. Dann hat  $\phi$  die Form:

$$\phi(\lambda, \mu) = \sum_{i=1}^n \min_{x_i} \left[ f_i(x_i(\lambda, \mu)) + \sum_{j=1}^{m_e} \mu_j h_{ij}(x_i(\lambda, \mu)) + \sum_{j=m_e+1}^m \lambda_j g_{ij}(x_i(\lambda, \mu)) \right] \quad (2.81)$$

Die Minimierung nach  $x$  zerfällt bei separablen Problemen in  $n$  eindimensionale Unterprobleme für jede Variable  $x_i$ . Diese Aufgaben lassen sich oft sehr effizient lösen (siehe dazu Abschnitt 3.2.2). Beispiele separabler Probleme sind alle linearen Probleme bzw. Linearisierungen nichtlinearer Aufgaben.

Teilaufgabe 2) läßt sich z.B. mit dem in Abschnitt 2.8.3 beschriebenen modifizierten konjugierten Gradientenalgorithmus lösen. Über die Erfahrung damit wird im Kapitel über Approximationsmethoden berichtet. Die duale Funktion ist unempfindlich gegenüber Variablentransformationen, dagegen kann die Kondition der Funktion durch eine Skalierung der Nebenbedingungen beeinflusst werden; eine Eigenschaft, die sich in keinem anderen Verfahren zeigt und den absoluten Gegensatz zu den hierzu gemachten Erfahrungen mit den Quasi-Newton-Verfahren darstellt.

Abschließend sei noch auf den Umstand hingewiesen, daß die Nebenbedingungen des Problems (NLP) die Ableitungen der dualen Funktion darstellen. Dies ist ein weiterer positiver Aspekt des Verfahrens, der einigen Rechenaufwand sparen hilft. Oft ist es nicht möglich, die Restriktionen der Variablen in der dualen Funktion zu erfassen. Dann ist auch das Teilproblem 1) mit Restriktionen versehen. Das ist im Grunde keine wesentliche Komplizierung, jedoch hat die direkte Beachtung der Restriktionen im Teilproblem 1) zur Folge, daß Nebenbedingungen in weiteren Bereichen für  $\lambda$  und  $\mu$  konstant sein können. Das bedeutet für die duale Funktion, daß die zweiten Ableitungen nur ab-

schnittsweise kontinuierlich sein können, mit den ganzen Konsequenzen für die verwendbaren Lösungsalgorithmen (siehe Abschnitt 3.2.4.2).

### 2.8.5.3 Die erweiterte Lagrange-Funktion; Multiplikatorenmethode

Eine wichtige Voraussetzung für die Anwendbarkeit der dualen Methoden ist, daß das Problem an der Lösung konvex sein muß (lokale Dualität). Die Erweiterung der Lagrange-Funktion um einen quadratischen Penalty-Term (2.64) hebt diesen Nachteil auf, denn durch die richtige Wahl des Penalty-Faktors gelingt es, das Problem zu einem konvexen zu machen. Vom Standpunkt der Penalty-Verfahren her gesehen, wird auf diese Weise die numerische Stabilität erhöht, wobei es bei der Vorgabe der richtigen Lagrange-Multiplikatoren für beliebige Penalty-Faktoren möglich ist, die exakte Lösung zu bestimmen. Für ein Problem mit ausschließlich Gleichheitsbedingungen lautet dann die erweiterte Lagrange-Funktion  $L_c$ :

$$L_c(x, \mu) = f(x) + \mathbf{h}(x)^T \mu + \frac{1}{2} c \mathbf{h}(x)^T \mathbf{h}(x) \quad (2.82)$$

Dabei ist  $L_c$  aus der Sicht der dualen Methode die Lagrange-Funktion und aus der der Penalty-Verfahren die Ersatzfunktion für folgende Aufgaben, zusammengefaßt in einer Tabelle:

dualer Aspekt (2.83)	Penalty-Aspekt (2.84)
$f(x) + \frac{1}{2} c \mathbf{h}(x)^T \mathbf{h}(x) \rightarrow \min$ mit: $\mathbf{h}(x) = 0$ konvex, wenn $c$ genügend groß	$f(x) + \mu^T \mathbf{h}(x) \rightarrow \min$ mit: $\mathbf{h}(x) = 0$ exakt lösbar für $c \neq \infty$ , wenn $\mu$ gegen $\mu^*$ konvergiert

Die erweiterte Lagrange-Funktion läßt sich auch für Ungleichheitsbedingungen herleiten, indem Schlupfvariablen  $s$  eingeführt werden.

$$\left. \begin{array}{l} f(x) \rightarrow \min \\ g(x) \leq 0 \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} f(x) \rightarrow \min \\ g_j(x) + s_j = 0 \end{array} \right. \quad (2.85)$$

$$s_j \geq 0 ; \quad j = 1; \dots, m$$

Damit ergibt sich die duale Funktion zu:

$$\phi(\lambda) = \min_{s, \mathbf{x}} \left[ f(\mathbf{x}) + \sum_{j=1}^m (\lambda_j (g_j(\mathbf{x}) + s_j) + \frac{1}{2} c (g_j(\mathbf{x}) + s_j)^2) \right] \quad (2.86)$$

Die Minimierung für  $s$  kann explizit ausgeführt werden:

$$s_j = \begin{cases} -g_j(\mathbf{x}) - \frac{\lambda_j}{c} & \text{für } -g_j(\mathbf{x}) - \frac{\lambda_j}{c} \geq 0 \\ 0 & \text{sonst} \end{cases} \quad (2.87)$$

die erweiterte Lagrange-Funktion kann geschrieben werden als

$$L_c(\mathbf{x}, \lambda) = f(\mathbf{x}) + \sum_{j=1}^m P_j (g_j(\mathbf{x}), \lambda_j) \quad (2.88)$$

Mit der Penalty-Funktion  $P_j$ , wobei (2.87) in einer alternativen Schreibweise bereits eingearbeitet worden ist:

$$P_j (g_j, \lambda_j) = \frac{1}{2c} \left[ \left\{ \max \left( 0, \lambda_j + c g_j(\mathbf{x}) \right) \right\}^2 - \lambda_j^2 \right] \quad (2.89)$$

In Bild 2.17 ist der Verlauf dieser Penalty-Funktion über der Nebenbedingung  $g_j(\mathbf{x})$  aufgetragen, und man kann erkennen, daß  $P_j$  für  $g_j(\mathbf{x}) < 0$  konstant negativ ist, jedoch nicht null, wie das eigentlich von einer Penalty-Funktion gefordert wird. Mit  $c \rightarrow \infty$  geht dieser Betrag jedoch zusehends gegen null. Andererseits erkennt man auch den  $C_1$ -kontinuierlichen Übergang der Horizontalen in die Parabel, der die stetige Ableitbarkeit der erweiterten Lagrange-Funktion garantiert. Allerdings sind die zweiten Ableitungen diskontinuierlich, jedoch nicht am Minimum des Originalproblems bei  $g_j(\mathbf{x}) = 0$ .

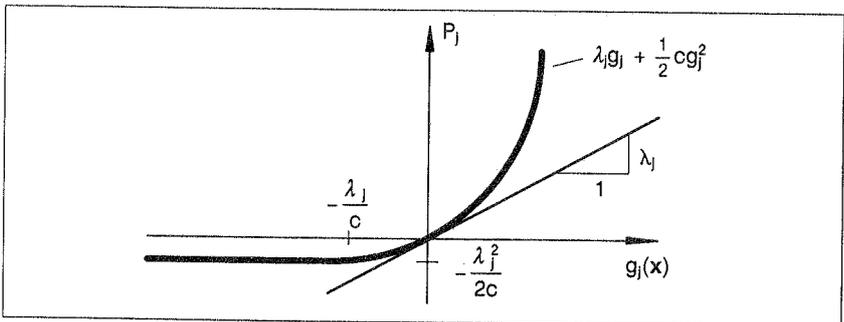


Bild 2.17: Penaltyfunktion einer Ungleichbedingung

Ein üblicher Weg zur Lösung eines Optimierungsproblems mit Hilfe der erweiterten Lagrangefunktion ist, analog zum Vorgehen bei den dualen Methoden, der folgende:

1. Minimiere bei gegebenen Lagrange-Multiplikatoren  $\lambda_k$  und  $\mu_k$  die duale Funktion  $\phi(\lambda_k, \mu_k)$  und bestimme  $\mathbf{x}(\lambda_k, \mu_k)$ :

$$\min_{\mathbf{x}} \phi(\lambda_k, \mu_k) = \min_{\mathbf{x}} \left[ f(\mathbf{x}) + \mathbf{h}^T(\mathbf{x}) \mu_k + \sum_{j=m_e+1}^m P_j(g_j(\mathbf{x}), \lambda_{j_k}) \right] \quad (2.90)$$

2. Ermittle neue Schätzungen für die Lagrange-Multiplikatoren:

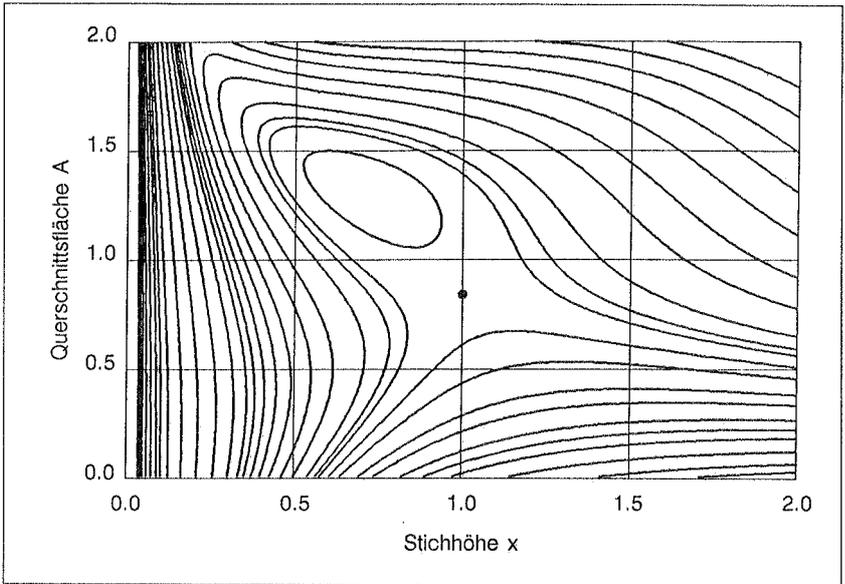
$$\begin{aligned} \lambda_{k+1} &= \max(0, \lambda_k + c_k \mathbf{g}(\mathbf{x}(\lambda_k, \mu_k))) \\ \mu_{k+1} &= c_k \mathbf{h}(\mathbf{x}(\lambda_k, \mu_k)) \end{aligned} \quad (2.91)$$

Dieses Lösungsschema wird Multiplikatorenmethode genannt.

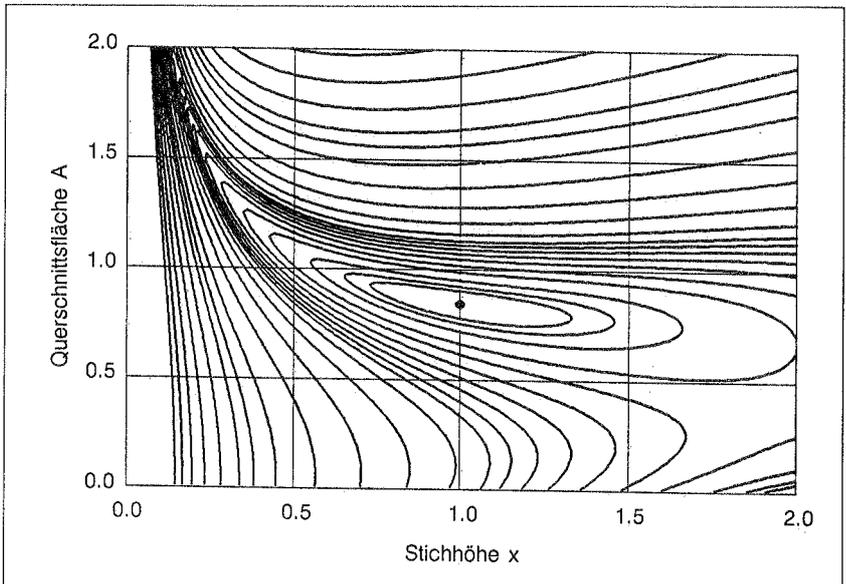
Schritt 1) ist derselbe wie bei der Methode von Uzawa, während Schritt 2) der Approximationsformel für Lagrange-Multiplikatoren bei den Penalty-Verfahren entspricht. Auch hier macht sich die Mischung der Verfahren bemerkbar.

Die Wahl des Penalty-Faktors sollte so groß vorgenommen werden, daß einerseits die Hesse-Matrix der erweiterten Lagrange-Funktion an der Lösung positiv definit wird, andererseits jedoch so, daß die Kondition der Hesse-Matrix und die Struktur des Problems nicht zu sehr leiden. Bild 2.18 zeigt zwei Fälle mit unterschiedlichen Penalty-Faktoren. Dargestellt ist der primäre Teil der Lagrange-Funktion des zweistöbigen Fachwerks für Euler-Knicken alleine. Allerdings wurde hier die Nebenbedingung in Kräften dargestellt, wodurch die Nebenbedingung konkav für die Querschnittsfläche wird.

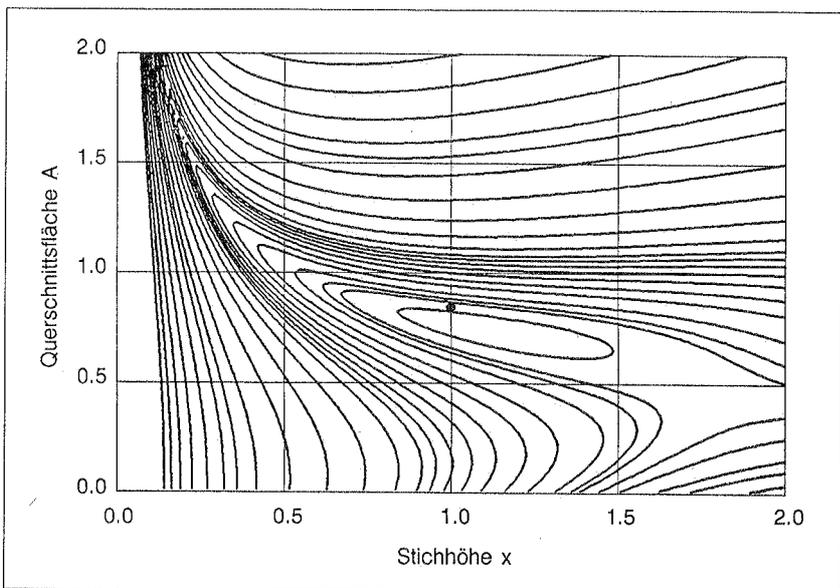
Die Bedeutung der Lagrange-Multiplikatoren ist nicht zu unterschätzen, denn die richtige Lösung  $\mathbf{x}^*$  kann nur erreicht werden, wenn die Lagrange-Multiplikatoren zu ihren richtigen Lösungen  $\lambda^*$  und  $\mu^*$  konvergieren. Dabei bestimmen sie die Konvergenzrate des gesamten Problems, wobei die dargestellte Korrekturformel (2.91) eine lineare Konvergenzrate aufweist. Gill et al. [52] schlagen deshalb auch eine quadratische Korrekturformel vor. Umgekehrt ist für  $\lambda \neq \lambda^*$  und  $\mu \neq \mu^*$  das Minimum der dualen Funktion nicht bei  $\mathbf{x}^*$ . Wie sich der Ort des Minimums mit einer Veränderung des Lagrange-Multiplikators ändert, ist in Bild 2.19 gezeigt, ebenfalls für das zweistöbige Fachwerk mit der Nebenbedingung "Euler-Knicken".



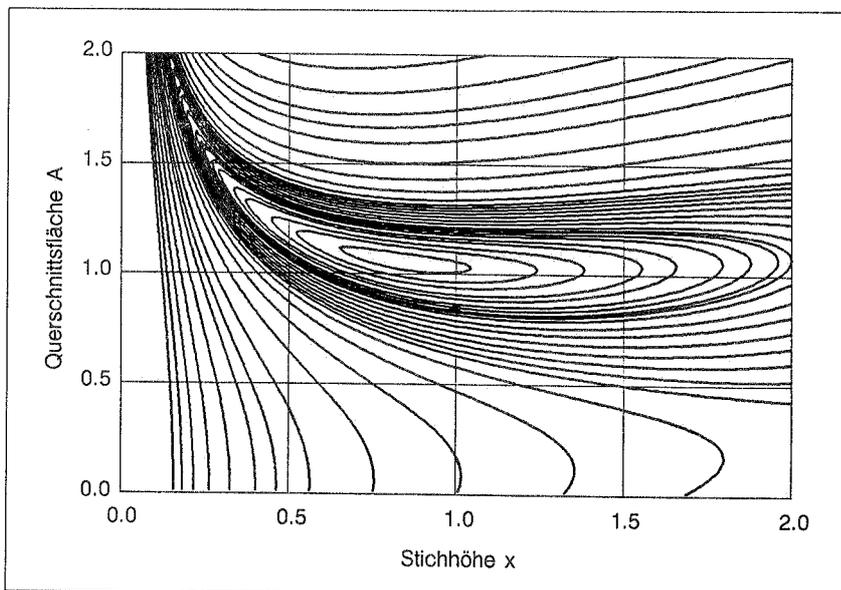
**Bild 2.18a:** erweiterte Lagrangefunktion, Variation des Penaltyfaktors ( $r=0,5$ ;  $\lambda=\lambda^*=1,687$ )



**Bild 2.18b:** erweiterte Lagrangefunktion, Variation des Penaltyfaktors ( $r=10,0$ ;  $\lambda=\lambda^*=1,687$ )



**Bild 2.19a:** erweiterte Lagrangefunktion, Variation des Lagrangemultiplikators ( $r=10$ ;  $\lambda=0$ )



**Bild 2.19b:** erweiterte Lagrangefunktion, Variation des Lagrangemultiplikators ( $r=10$ ;  $\lambda=7$ )

### 2.8.6 Lagrange-Methoden

Lagrange-Methoden beruhen auf der direkten Lösung der notwendigen Bedingungen (2.15). Man erhält so ein Gleichungssystem mit  $(n + m) -$  Unbekannten, den primären Optimierungsvariablen und den dualen Lagrange-Multiplikatoren. Diese beiden Gruppen von Variablen gehen gleichberechtigt in die Lagrange-Verfahren ein, weshalb hier konsequent von allem Gebrauch gemacht wird, was die Grundlage der anderen Verfahren, die in den vorherigen Abschnitten beschrieben worden sind, darstellt. Aus diesem Grund nennt Minoux [88] die Verfahren auch "primär-dual". Bekannte Vertreter der Lagrange-Methoden sind die Verfahren der "sequentiellen quadratischen Programmierung" (SQP), die sich den Umstand zunutze machen, daß die wiederholte Lösung eines quadratischen Unterproblems sich hervorragend zur Lösung der eigentlichen Aufgabe eignet. Diese Verfahren zeigen in der Nähe der Lösung superlineare Konvergenzeigenschaften, sind für Gleichheits- und Ungleichheitsbedingungen und für einen weiten Bereich von konvexen sowie nicht konvexen Aufgabenstellungen geeignet. Arora und seine Mitarbeiter haben diese Verfahren ausgiebigen Tests und vergleichenden Untersuchungen unterzogen [138] und empfehlen sie zur Anwendung in der Strukturoptimierung. Ein anderes bekanntes Verfahren, das Verfahren der "sequentiellen linearen Programmierung" (SLP), kann ebenfalls als eine Lagrange-Methode betrachtet werden [52]. Der Kern dieses Vorgehens ist die wiederholte Berechnung eines linearen Programms.

#### 2.8.6.1 Allgemeines

Für ein Problem mit  $n$  Unbekannten und  $m$  Gleichheitsbedingungen:

$$\begin{aligned} f(\mathbf{x}) &\rightarrow \min \\ \text{mit: } \mathbf{h}(\mathbf{x}) &= \mathbf{0} \end{aligned} \tag{2.92}$$

lauten die notwendigen Bedingungen (Kuhn-Tucker-Bedingungen):

$$\begin{aligned} \nabla f(\mathbf{x}) + \nabla \mathbf{h}(\mathbf{x})^T \boldsymbol{\mu} &= \mathbf{0} \\ \mathbf{h}(\mathbf{x}) &= \mathbf{0} \end{aligned} \tag{2.93}$$

Dieses nichtlineare Gleichungssystem kann unter anderem mit dem Newtonschen Verfahren iterativ gelöst werden. Dazu wird es an  $\mathbf{x}_k$  und  $\boldsymbol{\mu}_k$  linearisiert, und es ergeben sich folgende Ausdrücke:

$$\begin{aligned}
\nabla f(\mathbf{x}_k) + \nabla \mathbf{h}(\mathbf{x}_k)^T \mu_k + \nabla^2 f(\mathbf{x}_k) (\mathbf{x}_{k+1} - \mathbf{x}_k) + \\
\mu_k^T \nabla^2 \mathbf{h}(\mathbf{x}_k) (\mathbf{x}_{k+1} - \mathbf{x}_k) + \nabla \mathbf{h}(\mathbf{x}_k)^T (\mu_{k+1} - \mu_k) &= \mathbf{0} \\
\mathbf{h}(\mathbf{x}_k) + \nabla \mathbf{h}(\mathbf{x}_k) (\mathbf{x}_{k+1} - \mathbf{x}_k) &= \mathbf{0}
\end{aligned} \tag{2.94}$$

Mit:  $\mathbf{p}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$  und  $\nabla^2 L_k = \nabla^2 f(\mathbf{x}_k) + \mu_k^T \nabla^2 \mathbf{h}(\mathbf{x}_k)$

kann das Gleichungssystem in Matrixform dargestellt werden:

$$\begin{bmatrix} \nabla^2 L_k & \nabla \mathbf{h}(\mathbf{x})^T \\ \nabla \mathbf{h}(\mathbf{x}) & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{p}_k \\ \mu_{k+1} \end{Bmatrix} = - \begin{Bmatrix} \nabla f(\mathbf{x}_k) \\ \mathbf{h}(\mathbf{x}_k) \end{Bmatrix} \tag{2.95}$$

Für die Konvergenz einer iterativen Lösung von Gleichung (2.95) gelten die üblichen Voraussetzungen für das Newton-Verfahren:

- Das linearisierte System darf an der Lösung nicht singulär sein.
- Der Lösungsprozeß darf nicht zu weit entfernt von der Lösung gestartet werden.
- Für entfernter liegende Punkte sollten die bestimmten Suchrichtungen  $\mathbf{p}_k$  für einen anschließenden line search in einer Abstiegsfunktion verwendet werden.
- Die Projektion der Hesse-Matrix  $\nabla^2 L$  der Lagrange-Funktion muß im Tangentialraum der Nebenbedingungen positiv definit sein; d.h.  $\nabla^2 L$  selbst braucht nicht positiv definit sein, solange die Systemmatrix von Gleichung (2.95) invertierbar ist.

Die Gleichungen (2.95) können aber auch als die notwendigen Bedingungsgleichungen eines quadratischen Programms identifiziert werden. Zum Beispiel:

$$\nabla f(\mathbf{x}_k)^T \mathbf{p}_k + \frac{1}{2} \mathbf{p}_k^T \nabla^2 L_k(\mathbf{x}_k, \mu_k) \mathbf{p}_k \rightarrow \min \tag{2.96}$$

$$\text{mit: } \nabla \mathbf{h}(\mathbf{x}_k) \mathbf{p}_k + \mathbf{h}(\mathbf{x}_k) = \mathbf{0}$$

Zur Lösung dieses Problems können verschiedene Standardverfahren der quadratischen Programmierung herangezogen werden. Von Künzi et al. [81] werden eine ganze Reihe verschiedener Algorithmen besprochen, siehe aber auch bei Gill et al. [52], die besonders auf die effektive Lösung quadratischer Programme in Verbindung mit Matrixfaktorisierungen eingehen. Mit der Verwendung eines Verfahrens der quadratischen Programmierung zur Lösung eines Iterationsschrittes der Newton-Approximation kann das

Verfahren auch leicht für Probleme mit Ungleichheitsbedingungen erweitert werden. Ein linearisierter Schritt bei der Lösung des allgemeinen Problems (NLP) der nichtlinearen Programmierung lautet dann:

$$\nabla f(\mathbf{x}_k) \mathbf{p}_k + \frac{1}{2} \mathbf{p}_k^T \nabla^2 L_k \mathbf{p}_k \rightarrow \min$$

mit:  $\nabla h(\mathbf{x}_k) \mathbf{p}_k + \mathbf{h}_k = \mathbf{0}$  (2.97)

$$\nabla g(\mathbf{x}_k) \mathbf{p}_k + \mathbf{g}_k \leq \mathbf{0}$$

Die wiederholte Lösung eines quadratischen Programms hat die Namen "sequentielle" und "rekursive quadratische Programmierung" geprägt (auch SQP- oder RQP-Verfahren). Man kann das Vorgehen auch als die Minimierung einer Approximation der Hesse-Matrix im Tangentialraum der linearisierten, aktiven Nebenbedingungen interpretieren [52], weshalb das Verfahren auch "projizierte Lagrange-Methode" genannt wird.

Anstelle der Hesse-Matrix  $\nabla^2 L$  können auch Approximationsmatrizen im Sinne der Quasi-Newton-Verfahren der unbeschränkten Optimierung eingesetzt werden. Dann jedoch ist ein line search unerlässlich. Und hier liegt das inhärente Problem aller Lagrange-Verfahren. Da die notwendigen Bedingungen (2.93) die Stationärbedingungen eines Sattelpunktes der Lagrange-Funktion darstellen, kommt die Lagrange-Funktion selbst als Abstiegsfunktion, die ein Maß zur Beurteilung des Lösungsfortschrittes darstellt, nicht in Betracht.

### 2.8.6.2 Abstiegsfunktionen

Eine geeignete Abstiegsfunktion (merit function) muß folgende Voraussetzungen erfüllen:

- (1) Sie soll ihr Minimum im Raum der primären und dualen Variablen am Lösungspunkt der beschränkten Optimierungsaufgabe haben, d.h. am Sattelpunkt der Lagrange-Funktion.
- (2) Die im Unterproblem bestimmte Richtung  $\mathbf{p}_k$  muß in der Abstiegsfunktion abwärts führen.
- (3) Die Abstiegsfunktion darf die Konvergenz des Gesamtverfahrens nicht negativ beeinflussen.
- (4) Die Auswertung der Abstiegsfunktion sollte nicht zu aufwendig sein.

Interessanterweise scheint keine Abstiegsfunktion bekannt zu sein, die alle vier genannten Anforderungen gleichzeitig erfüllt. An der Abstiegsfunktion unterscheiden sich auch die verschiedenen Algorithmen; ein Punkt, der immer noch diskutiert wird, wie der erst kürzlich erschienene Beitrag von Boggs und Tolle [17] belegt. Wegen der ersten Bedin-

gung sind alle bekannten Abstiegsfunktionen mit Penalty-Termen versehen, um zumindest in der Nähe der Lösung die Konvexität der Abstiegsfunktion garantieren zu können. Deshalb ist die richtige Adaption der Penaltyfaktoren während des Lösungsfortschrittes ein zentraler Punkt der Algorithmen. Es ergeben sich interessante Beziehungen zu den Penalty-Verfahren. Nachfolgend sind einige gebräuchliche Abstiegsfunktionen aufgelistet.

### **$L_1$ - Penalty-Funktion**

Die Funktion, auch exakte Penalty-Funktion genannt, erfüllt die Anforderungen (1) und (4), jedoch kann sie die Konvergenz eines SQP-Verfahrens in der Nähe der Lösung drastisch verlangsamen, da sie die ideale Newton-Schrittweite nicht zuläßt. Dieser Effekt wird auch "Maratos-Effekt" genannt [85], [117]. Ein weiterer Nachteil in Verbindung mit dem line search ist, daß diese Funktion nicht differenzierbar ist und deshalb Interpolationsverfahren nicht in Frage kommen. Zur Abschätzung der Penaltyfaktoren werden die Lagrangemultiplikatoren herangezogen. Ursprünglich schlug Han [64] die  $L_1$ -Penalty-Funktion als Abstiegsfunktion vor. Im Unterschied zu (2.63) führt er für jede Nebenbedingung einen Penalty-Faktor ein:

$$m(\mathbf{x}, \mathbf{r}) = f(\mathbf{x}) + \sum_{j=1}^{m_e} r_j |h_j(\mathbf{x})| + \sum_{j=m_e+1}^m r_j (\max(0, g_j(\mathbf{x}))) \quad (2.98)$$

mit:

$$r_j = \begin{cases} \max(|\mu_j|, 0,5 (r_j + |\mu_j|)) ; & j = 1, \dots, m_e \\ \max(\lambda_j, 0,5 (r_j + \lambda_j)) ; & j = m_e+1, \dots, m \end{cases}$$

### **Erweiterte Lagrange-Funktion**

die erweiterte Lagrangefunktion ist die Grundlage einer Reihe von differenzierbaren Abstiegsfunktionen, die sich im wesentlichen in der Abschätzung der Lagrangemultiplikatoren und im Penaltyterm unterscheiden. Im Gegensatz zur Anwendung in der Multiplikatorenmethode, wo es darauf ankommt, den Primärteil der Lagrangefunktion bei gegebenen Schätzungen der Lagrangemultiplikatoren konvex zu halten, sollen die Abstiegsfunktionen auch unter Berücksichtigung veränderlicher Lagrangemultiplikatoren konvex sein. Dabei werden die Lagrangemultiplikatoren als Funktionen der primären Variablen eingeführt. Da die Lagrangefunktion immer eine mehrdimensionale Sattelfläche darstellt, hängt die Wahl geeigneter (d.h. genügend großer) Penaltyfaktoren auch von den Beziehungen zwischen Lagrangemultiplikatoren und primären Variablen ab, was die numerisch ohnehin schon sensible Aufgabe weiter erschweren kann. Die allgemeine Form der erweiterten Lagrangefunktion in der Verwendung als Abstiegsfunktion ist:

$$m(\mathbf{x}, \bar{\lambda}(\mathbf{x}), \bar{\mu}(\mathbf{x}), \mathbf{r}) = L(\mathbf{x}, \bar{\lambda}(\mathbf{x}), \bar{\mu}(\mathbf{x})) + P(\mathbf{r}, \mathbf{x}, \bar{\lambda}(\mathbf{x}), \bar{\mu}(\mathbf{x})) \quad (2.99)$$

### Der Vorschlag von Schittkowski

Schittkowski [117] verwendet als Penaltyterm die für Ungleichheitsbedingungen erweiterte quadratische Penaltyfunktion (2.89). Die Beziehung zwischen Lagrangemultiplikatoren und primären Variablen macht er vom Iterationsverlauf abhängig. Die Abstiegsfunktion ist entlang eines linearen Schnittes durch die erweiterte Lagrangefunktion definiert:

$$m(\bar{\mathbf{x}}, \bar{\lambda}(\bar{\mathbf{x}}), \bar{\mu}(\bar{\mathbf{x}}), \mathbf{r}) = f(\bar{\mathbf{x}}) + \sum_{j=1}^{m_e} (\bar{\mu}_j h_j(\bar{\mathbf{x}}) + \frac{1}{2} r_j h_j(\bar{\mathbf{x}})^2) \quad (2.100)$$

$$+ \sum_{j=m_e+1}^m \begin{cases} \bar{\lambda}_j g_j(\bar{\mathbf{x}}) + \frac{1}{2} r_j g_j(\bar{\mathbf{x}})^2; & \text{falls } g_j(\bar{\mathbf{x}}) \geq -\frac{\lambda_j}{r_j} \\ -\frac{1}{2r_j} \lambda_j^2 & ; \text{sonst} \end{cases}$$

mit:

$$\bar{\mathbf{x}} = \mathbf{x}_k + \alpha(\mathbf{x}_{k+1} - \mathbf{x}_k)$$

$$\bar{\lambda} = \lambda_k + \alpha(\lambda_{k+1} - \lambda_k) = \lambda_k + (\lambda_{k+1} - \lambda_k) \frac{\bar{\mathbf{x}} - \mathbf{x}_k}{\mathbf{x}_{k+1} - \mathbf{x}_k}$$

$$\bar{\mu} = \mu_k + \alpha(\mu_{k+1} - \mu_k) = \mu_k + (\mu_{k+1} - \mu_k) \frac{\bar{\mathbf{x}} - \mathbf{x}_k}{\mathbf{x}_{k+1} - \mathbf{x}_k}$$

$\alpha$  ... line search Parameter

$\mathbf{x}_k, \lambda_k, \mu_k$  ... letzte Näherung

$\mathbf{x}_{k+1}, \lambda_{k+1}, \mu_{k+1}$  ... neue Näherung der Lösung, ermittelt aus dem quadratischen Unterproblem.

$r_j$  ... Penaltyfaktoren (siehe Absatz 2.8.6.4)

Die Funktion (2.100) ist stetig ableitbar und erfüllt für geeignete Penaltyfaktoren die Bedingungen (2) bis (4). In [117] leitet Schittkowski aus der Konvergenzanalyse eine Abschätzung für die Wahl der Penaltyfaktoren ab. Der Einfluß dieser Abschätzung auf das line search Verfahren wird in Abschnitt 2.8.6.4 diskutiert. Bedingung (1) ist nur in Sonderfällen erfüllbar, wenn die lineare Spur von (2.100) die Lösung beinhaltet, insbesondere dann, wenn die Lagrangemultiplikatoren bereits den optimalen Wert eingenommen haben. Auf diese theoretische Schwäche weisen Boggs und Tolle [17] sowie Shanno [126] hin. Demnach ist die Tauglichkeit dieses Vorgehens sehr stark vom Konvergenzverhalten der Lagrangemultiplikatoren abhängig. Dies äußert sich in einem starken Anwachsen der Penaltyfaktoren, unter Umständen bis an die numerisch zulässigen

Grenzen und tritt in den Fällen auf, wenn die primären Variablen die Lösung bereits erreicht haben, die dualen jedoch noch nicht. Nach [53] ist dies jedoch nicht die Regel. Powell [101] berichtet von ernststen Schwierigkeiten des Verfahrens bei der Anwendung auf Probleme mit linear abhängigen Gleichheitsbedingungen. Hier sind wegen der fehlenden Qualifikation der Nebenbedingungen die Lagrangemultiplikatoren nicht eindeutig bestimmbar, und ihr Konvergenzverhalten unterliegt der Willkür des gewählten Verfahrens zur Lösung des quadratischen Unterproblems. Andererseits zeigt ein von ihm angestellter numerischer Vergleich die Effektivität der von Schittkowski vorgeschlagenen Abstiegsfunktion bei stark nichtlinearen Problemen, die nicht an den genannten Schwächen leiden. Dies kann aus der eigenen Erfahrung bestätigt werden.

### Der Vorschlag von Boggs und Tolle

In [16] führen Boggs und Tolle eine Abstiegsfunktion folgender Form ein:

$$m(\mathbf{x}, r) = L(\mathbf{x}, \bar{\boldsymbol{\mu}}(\mathbf{x})) + \frac{1}{2}r \mathbf{h}(\mathbf{x})^T [\nabla \mathbf{h}(\mathbf{x}) \nabla \mathbf{h}(\mathbf{x})^T]^{-1} \mathbf{h}(\mathbf{x}) \quad (2.101)$$

$$\bar{\boldsymbol{\mu}}(\mathbf{x}) = -[\nabla \mathbf{h}(\mathbf{x}) \nabla \mathbf{h}(\mathbf{x})^T]^{-1} \nabla \mathbf{h}(\mathbf{x}) \nabla f(\mathbf{x}) \quad (2.102)$$

Die Funktion wird nur für Gleichheitsbedingungen formuliert, müßte allerdings über eine Strategie des aktiven Satzes auch für Ungleichheitsbedingungen erweitert werden können. Der Penalty-Term ist von der Art (2.65) und die Lagrange-Multiplikatoren werden näherungsweise als das Fehlerquadratminimum von  $\|\nabla f(\mathbf{x}) + \nabla \mathbf{h}(\mathbf{x})^T \bar{\boldsymbol{\mu}}\|$  bestimmt. Mit dieser Abschätzung für die Multiplikatoren entspricht  $L(\mathbf{x}, \bar{\boldsymbol{\mu}}(\mathbf{x}))$  näherungsweise der konkaven dualen Funktion, diesmal jedoch in Abhängigkeit der primären Variablen. Anschaulich gesprochen, nähert sich  $L(\mathbf{x}, \bar{\boldsymbol{\mu}}(\mathbf{x}))$  im "dualen Tal" der Lösung, stetig nach oben strebend. Der dualen Funktion wird der lokal konvexe Penalty-Term überlagert, so daß die Abstiegsfunktion für genügend großes  $r$  am Sattelpunkt der Lagrange-Funktion ein Minimum hat und konvex ist. Sie ist kontinuierlich ableitbar und erfüllt die Punkte (1) bis (3) des Anforderungskataloges. Jedoch ist die Auswertung der Funktion und insbesondere der Ableitung mit einem sehr hohen Aufwand verbunden, da jedesmal die Gradienten bzw. zweiten Ableitungen der Zielfunktion und der Nebenbedingungen zu bestimmen sind. Damit kommt diese Funktion für Aufgaben der Strukturoptimierung nicht in Frage. In einem neueren Aufsatz tragen Boggs und Tolle diesem Problem Rechnung und führen eine lineare Approximation für (2.101) ein [17], die sich wie die Funktion Schittkowskis am aktuellen Iterationspunkt orientiert. Dieses Vorgehen erweist sich in einem numerischen Test [126] anderen Verfahren für Probleme mit Gleichheitsbedingungen überlegen.

### Der Vorschlag von Powell und Yuan

Eine Abstiegsfunktion, die sich von (2.100) in der Abschätzung der Lagrangemultiplikatoren und von (2.101) im Penalty-Term unterscheidet, wird von Powell und Yuan vorgeschlagen [102]:

$$m(\mathbf{x}, r) = L(\mathbf{x}, \bar{\mu}(\mathbf{x})) + \frac{1}{2} r \mathbf{h}^T(\mathbf{x}) \mathbf{h}(\mathbf{x}) \quad (2.103)$$

Diese Funktion, ebenfalls nur für Probleme mit Gleichheitsbedingungen vorgeschlagen, hat dieselben Vorzüge und Einschränkungen wie (2.101). Powell und Yuan versuchen ebenfalls durch eine Linearisierung der Abstiegsfunktion am aktuellen Iterationspunkt  $\mathbf{x}_k$  die numerische Effizienz zu erhöhen.

### Die Abstiegsfunktion von Pshenichny

Arora und seine Mitarbeiter verwenden in ihren Algorithmen [141] die von Pshenichny formulierte Abstiegsfunktion:

$$m(\mathbf{x}, r) = f(\mathbf{x}) + r v(\mathbf{x}) \quad (2.104)$$

mit: 
$$r \geq \sum_{j=1}^{m_e} |\mu_j| + \sum_{j=m_e+1}^m \lambda_j \quad (2.105)$$

und: 
$$v(\mathbf{x}) = \max \{ 0, |h_1(\mathbf{x})|, \dots, |h_{m_e}(\mathbf{x})|, g_{m_e+1}(\mathbf{x}), \dots, g_m(\mathbf{x}) \} \quad (2.106)$$

Eigene Erfahrungen mit dieser Funktion liegen nicht vor, aber es ist anzunehmen, daß sich diese Abstiegsfunktion im wesentlichen wie die  $L_1$ -Penalty-Funktion verhalten wird.

In Bild (2.20) sind die vier ersten Abstiegsfunktionen am Beispiel des 2-ständigen Fachwerks dargestellt. Es wurde ein Schnitt durch den Entwurfsraum mit der optimalen Stichtiefe  $x^* = 2$  gewählt und somit das Problem auf eine Dimension verringert, um die Übersichtlichkeit der Darstellung zu erleichtern. Die verbleibende freie Variable ist die Querschnittsfläche  $A$ . Für  $x^* = 2$  ergeben sich folgende Beziehungen:

Gewicht:  $f = 4\sqrt{2} A$

Stabfließen:  $h = \frac{1}{\sqrt{2} A} - 1$  (als Gleichheitsbedingung)

$$\begin{aligned}
L_1\text{-Penalty-Funktion: } m &= 4\sqrt{2} A + r \left( \frac{1}{\sqrt{2} A} - 1 \right) && ; 0 < A \leq A^* = \frac{1}{2} \sqrt{2} \\
m &= 4\sqrt{2} A && ; A > \frac{1}{2} \sqrt{2} \\
r &> 8 A^2 && ; 0 < A \leq \frac{1}{2} \sqrt{2} \\
r &= 0 && ; A > \frac{1}{2} \sqrt{2} \\
r_{\max} &= 4
\end{aligned}$$

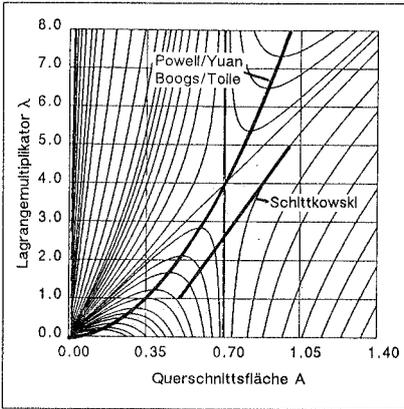
$$\begin{aligned}
\text{Schittkowski: } m_k &= 4\sqrt{2} A_k + \lambda_k \left( \frac{1}{\sqrt{2} A_k} - 1 \right) + \frac{1}{2} r \left( \frac{1}{\sqrt{2} A_k} - 1 \right)^2 \\
r_k &> \sqrt{2} A_k \frac{8 A_k^2 - \lambda_k}{1 - \sqrt{2} A_k} \\
r_{k\max} &= r_{k\max}(\lambda_k) \rightarrow \infty
\end{aligned}$$

$$\begin{aligned}
\text{Boggs und Tolle: } m &= 8 A (\sqrt{2} - A) + \frac{1}{2} r A^2 (1 - \sqrt{2} A)^2 \\
r &> \left[ \frac{1}{8} A \left( 2 A - \frac{1}{\sqrt{2}} \right) \right]^{-1} \\
r_{\max} &\rightarrow \infty
\end{aligned}$$

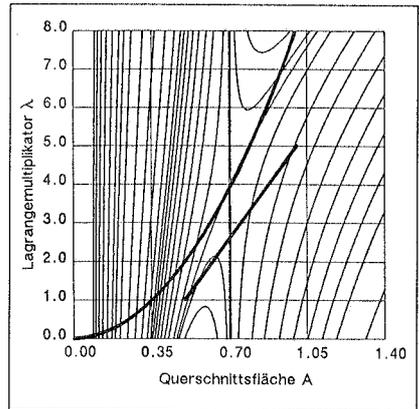
$$\begin{aligned}
\text{Powell und Yuan: } m &= 8 A (\sqrt{2} - A) + \frac{1}{2} r \left( \frac{1}{\sqrt{2} A} - 1 \right)^2 \\
r &> 16\sqrt{2} A^3 \\
r_{\max} &\rightarrow \infty
\end{aligned}$$

Davon abgesehen, daß sie nicht differenzierbar ist, stellt sich die  $L_1$ -Penalty-Funktion am problemlosesten dar. Die exakte Lösung ist mit einem endlichen Penalty-Faktor erreichbar, auch ist der zur Konvexifizierung nötige Mindestbetrag sowohl nach unten wie nach oben beschränkt.

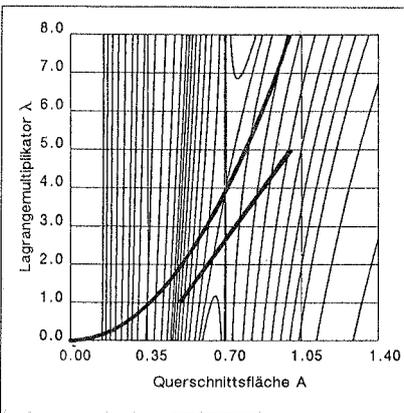
Die aus der erweiterten Lagrangefunktion abgeleiteten Abstiegsfunktionen von Boggs/Tolle und Powell/Yuan haben ebenfalls für einen endlichen Penalty-Faktor ein lokales Minimum an der Optimalstelle. Jedoch weist die Funktion von Boggs/Tolle in nächster Nähe der Lösung ein weiteres Minimum auf, das selbst für  $r \rightarrow \infty$  nicht verschwinden kann. Für dieses Beispiel erweist sich die Funktion als besonders unpraktisch, da sich dieses Verhalten im unzulässigen Bereich zeigt, wo ein wohlgefälliger Verlauf wünschenswert ist.



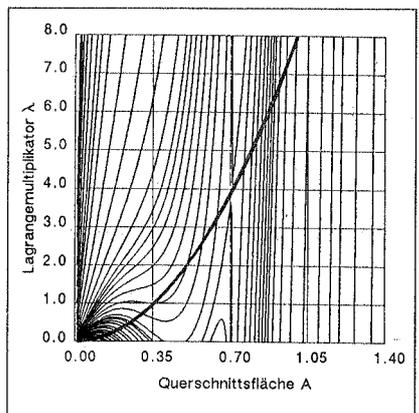
**Bild 2.20a:** erweiterte Lagrangefunktion  
Penaltyfaktor  $r = 0,0$



**Bild 2.20b:** Powell/Yuan, Schittkowski  
Penaltyfaktor  $r = 10,0$



**Bild 2.20c:** Powell/Yuan, Schittkowski  
Penaltyfaktor  $r = 32,0$



**Bild 2.20d:** Boggs/Tolle  
Penaltyfaktor  $r = 64,0$

**Bild 2.20 :** Abstiegsfunktionen

In den Höhenlinienbildern von Bild 2.20 ist die Spur der dualen Funktion eingetragen, entlang welcher die Abstiegsfunktionen von Powell/Yuan bzw. Boggs/Tolle definiert sind. Man erkennt, wie eine Vergrößerung des Penaltyfaktors die erweiterte Lagrangefunktion entlang dieser Spur in der Umgebung der Lösung konvexifiziert. Dies trifft auch auf die lineare Spur des Verfahrens von Schittkowski zu, die in den Bildern 2.20a bis 2.20c zusätzlich eingetragen ist. Gewählt wurde eine beliebige, mögliche Kombinati-

on von primären und dualen Variablen. Man kann sich leicht vorstellen, daß bei einer relativ zur eingetragenen links herum gedrehten Spur (was einem schlechteren Konvergenzverhalten des Lagrangemultiplikators entspräche) der Wert des Penaltyfaktors vergrößert werden muß. Dagegen kann sein Wert verringert werden, wenn die Spur rechts herum gedreht wird. Dies kann durch eine Skalierung der Nebenbedingung beeinflusst werden (Abschnitt 2.8.6.4).

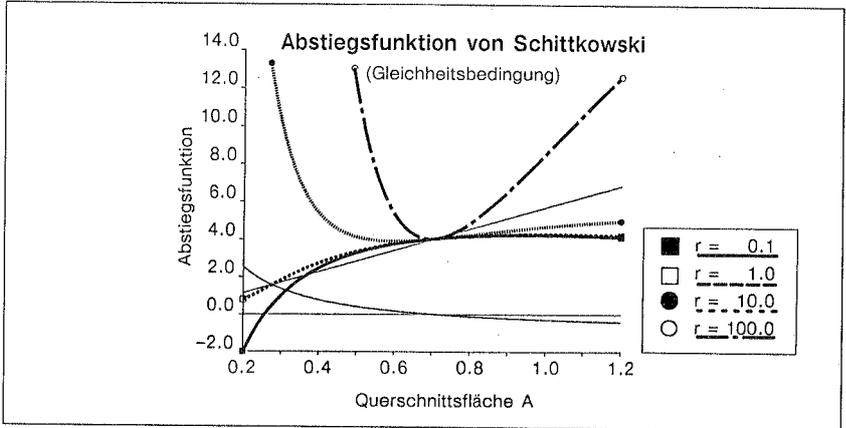


Bild 2.20e: Abstiegssfunktion von Schittkowski (entlang der Spur)

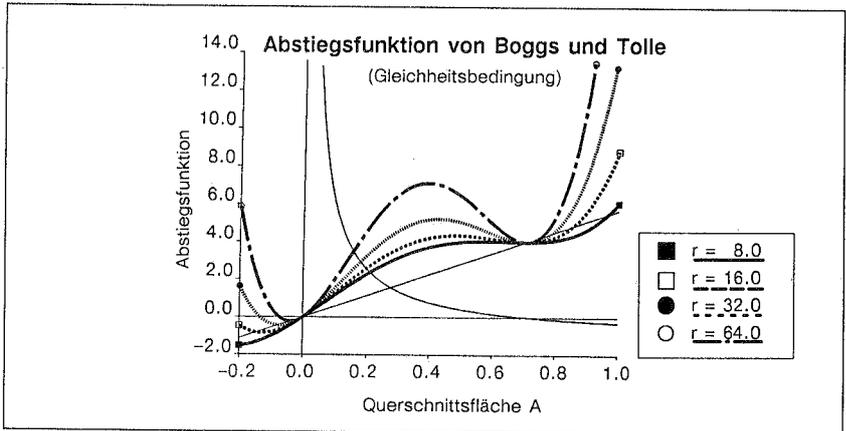


Bild 2.20f: Abstiegssfunktion von Boggs/Tolle (entlang der Spur)

Bild 2.20: Abstiegssfunktionen

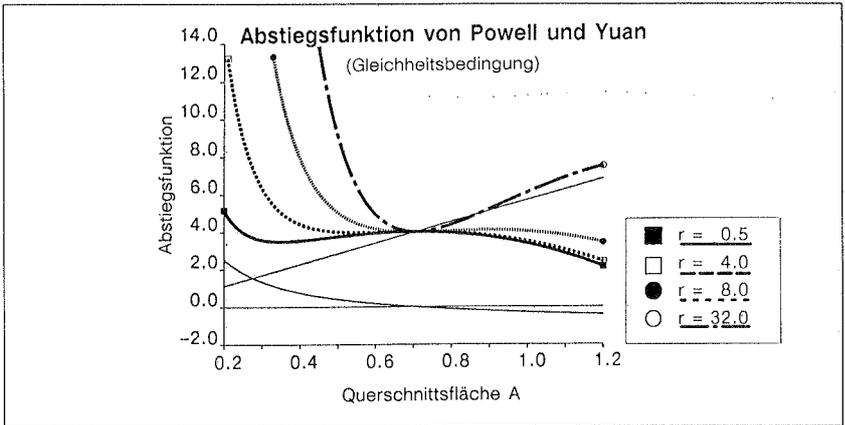


Bild 2.20g: *Abstiegswfunktion von Powell/Yuan (entlang der Spur)*

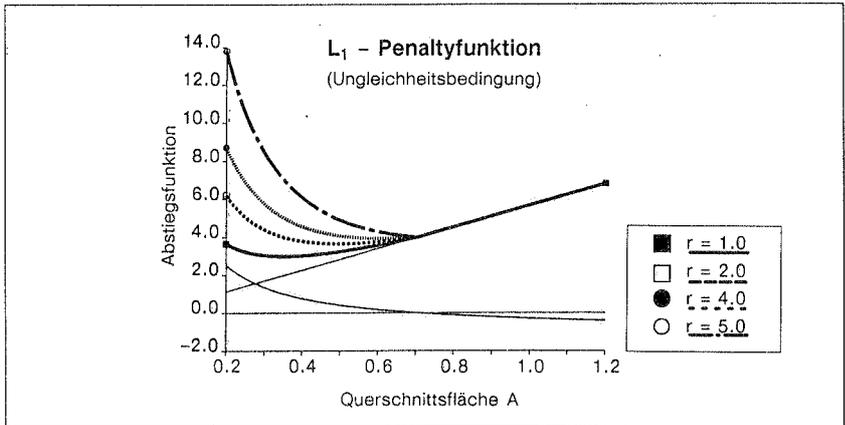


Bild 2.20h: *L<sub>1</sub>-Penaltyfunktion (entlang der Spur)*

Bild 2.20: *Abstiegswfunktionen*

### 2.8.6.3 Das Verfahren von Han und Powell; SQP und BFGS

Eine natürliche Erweiterung im Prozeß der sequentiellen quadratischen Programmierung stammt von Han [64] und Powell [100], die anstelle der expliziten Bestimmung der Hesse-Matrix, wie es ursprünglich vorgeschlagen worden war, die Korrekturformeln der Quasi-Newton-Verfahren heranzogen. Hier kam der Algorithmus in der Version von Schittkowski [117] zur Anwendung. Er lautet:

1. Wähle Startwerte für  
 die Variablen  $x_0$  und die Lagrange-Multiplikatoren  $\mu_0, \lambda_0$ ,  
 die Matrix  $B_0$  zur Approximation der Hessematrix der Lagrangefunktion,  
 setze den Iterationszähler  $k = 0$ .
2. Iterationszähler  $k := k+1$   
 Löse das quadratische Unterproblem (Gl. 2.96), bestimme  $p_k, \mu_{k+1}, \lambda_{k+1}$ :

$$\min \frac{1}{2} p_k^T B_k p_k + \nabla f(x_k)^T p_k$$

mit:

$$\nabla h(x_k) p_k + h(x_k) = 0$$

$$\nabla g(x_k) p_k + g(x_k) \leq 0$$

$$p_k \leq x_U - x_k$$

$$-p_k \leq x_k - x_L$$

3. Konvergenzabfrage:

$$p_k^T B_k p_k \leq \epsilon^2$$

$$\sum_{j=1}^{m_e} |\mu_{jk} h_j(x_k)| + \sum_{j=m_e+1}^m |\lambda_{jk} g_j(x_k)| \leq \epsilon$$

$$\| \nabla_x L(x_k, \lambda_k, \mu_k) \|^2 \leq \epsilon$$

$$\sum_{j=1}^{m_e} |h_j(x_k)| + \sum_{j=m_e+1}^m \max(0, g_j(x_k)) \leq \sqrt{\epsilon}$$

4. Bestimme neue Werte für die Penalty-Faktoren der Abstiegswfunktion.
5. Führe einen line search (gewöhnlich nach der Armijo/Goldstein-Regel) mit einer der genannten Abstiegswfunktionen durch. Für die Funktion von Schittkowski ist die Suchrichtung dabei  $(p_k, \mu_{k+1} - \mu_k, \lambda_{k+1} - \lambda_k)^T$ . Bestimme Schrittweite  $\alpha$ .

6. Bestimme:
- $$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha \mathbf{p}_k$$
- $$\lambda_{k+1} = \lambda_k + \alpha (\lambda_{k+1} - \lambda_k)$$
- $$\mu_{k+1} = \mu_k + \alpha (\mu_{k+1} - \mu_k)$$

Werte die Funktion für  $\mathbf{x}_{k+1}$  aus und bestimme die Gradienten.

7. Korrigiere die Approximation der Hesse-Matrix mit der BFGS-Formel (2.32):

$$\mathbf{B}_{k+1} = \mathbf{B}_k - \frac{\mathbf{B}_k \mathbf{d}_k \mathbf{d}_k^T \mathbf{B}_k}{\mathbf{d}_k^T \mathbf{B}_k \mathbf{d}_k} + \frac{\mathbf{r}_k \mathbf{r}_k^T}{\mathbf{d}_k^T \mathbf{r}_k}$$

mit:

$$\mathbf{d}_k = \alpha \mathbf{p}_k$$

$$\mathbf{q}_k = \nabla L(\mathbf{x}_{k+1}, \mu_{k+1}, \lambda_{k+1}) - \nabla L(\mathbf{x}_k, \mu_k, \lambda_k)$$

$$\mathbf{r}_k = \Theta \mathbf{q}_k + (1 - \Theta) \mathbf{B}_k \mathbf{d}_k$$

dabei ist:

$$\Theta = \begin{cases} 1 & \text{falls } \mathbf{d}_k^T \mathbf{q}_k \geq 0, 2 \mathbf{d}_k^T \mathbf{B}_k \mathbf{d}_k \\ \frac{0,8 \mathbf{d}_k^T \mathbf{B}_k \mathbf{d}_k}{\mathbf{d}_k^T \mathbf{B}_k \mathbf{d}_k - \mathbf{d}_k^T \mathbf{q}_k} & \text{falls } \mathbf{d}_k^T \mathbf{q}_k < 0, 2 \mathbf{d}_k^T \mathbf{B}_k \mathbf{d}_k \end{cases}$$

Diese Modifikation der BFGS-Korrektur stammt von Powell und garantiert, daß der Faktor  $\mathbf{d}_k^T \mathbf{r}_k$  größer als null ist. Somit wird erreicht, daß die Approximation der Hesse-Matrix stets positiv definit ist. Verschiedentlich wird jedoch berichtet [17], [126], [137], daß diese Modifikation die Konvergenz verlangsamt oder gar destabilisiert. Alternativ wird deshalb vorgeschlagen, die Korrektur zu unterlassen oder bei der Verbesserung einer LDL<sup>T</sup>-zerlegten Approximationsmatrix direkt die Werte der Diagonalelemente zu kontrollieren. Ein Algorithmus zur Verbesserung faktorisierter Matrizen wird von Gill et al. angegeben [52].

8. Gehe nach 2.).



Im Laufe der Iteration kann es vorkommen, daß das in 2.) zu lösende Unterproblem inkonsistent ist, d.h. daß sich die linearisierten Nebenbedingungen gegenseitig ausschließen, obwohl ursprünglich ein intaktes Problem formuliert wurde. In diesem Fall wird eine zusätzliche Relaxationsvariable  $\delta$  eingeführt, mit der das Unterproblem erneut gelöst wird. Das erweiterte quadratische Programm lautet dann:

$$\min \frac{1}{2} \mathbf{p}_k^T \mathbf{B}_k \mathbf{p}_k + \nabla f(\mathbf{x}_k)^T \mathbf{p}_k + \frac{1}{2} \varrho \delta^2$$

$$\begin{aligned} \text{mit:} \quad & \nabla \mathbf{h}(\mathbf{x}_k) \mathbf{p}_k + (1 - \delta) \mathbf{h}(\mathbf{x}_k) = \mathbf{0} \\ & \nabla g_j(\mathbf{x}_k) \mathbf{p}_k + (1 - \delta) g_j(\mathbf{x}_k) \leq 0 \quad ; \quad j \in J_k \\ & \nabla g_j(\mathbf{x}_k) \mathbf{p}_k + (1 - \delta) g_j(\mathbf{x}_k) \leq 0 \quad ; \quad j \notin J_k \\ & 0 \leq \delta \leq 1 \end{aligned}$$

$\varrho$  ... zusätzlicher Penaltyfaktor

Dabei bestimmt  $J_k$  den Satz der aktiven Nebenbedingungen. Ist ein Unterproblem mehrfach hintereinander inkonsistent, wird angenommen, daß auch die ursprüngliche Aufgabe unlösbar ist, und der Algorithmus bricht ab. In der Version von Schittkowski [117] wird der aktive Satz folgendermaßen bestimmt:

$$J_k = \{ 1, \dots, m_e \} \cup \left\{ j \mid m_e + 1 \leq j \leq m, \quad g_j(\mathbf{x}_k) \geq -\epsilon \text{ oder } \lambda_{jk} > 0 \right\}$$

Nebenbedingungen werden auch dann als aktiv erkannt, wenn sie zulässig, aber die zugehörigen Lagrange-Multiplikatoren größer null sind. Dieser Fall tritt vor allem bei nicht konvexen Nebenbedingungen regelmäßig auf.

Weiter werden bei Schittkowski nur die Gradienten der aktiven Nebenbedingungen bestimmt, was erheblichen Aufwand spart. Für die nicht aktiven Nebenbedingungen werden die Gradienten des vorherigen Schrittes übernommen. Im Unterproblem (Schritt 2) werden alle Nebenbedingungen berücksichtigt. So wird rechtzeitig registriert, wenn eine bislang nicht aktive Nebenbedingung aktiv wird. Verschiedentlich kann es allerdings vorkommen, daß auf diese Weise das Unterproblem zu stark restringiert wird, weshalb von Arora und seinen Mitarbeitern [137], [141] dieses Vorgehen zugunsten einer "potential constraint strategy" abgelehnt wird.

Zusammengefaßt sind die Vorteile der Han-Powell-Methode:

- Sie ist global konvergent.
- Lokal ist die Konvergenz superlinear, wenn der ideale Newton-Schritt mit der gewählten Abstiegsfunktion harmoniert ("Maratos-Effekt").
- Das Verfahren berücksichtigt Informationen zweiter Ordnung.
- Es können hochgradig nichtlineare Probleme mit Gleichheits- und Ungleichheitsbedingungen gelöst werden.

Nachteile sind:

- Das Verfahren ist speicher- und rechenintensiv. Dabei ist für die Verhältnisse der Strukturoptimierung nur der erste Punkt von Belang, denn der Aufwand, der bei der Gradientenbestimmung getrieben werden muß, übertrifft den Aufwand des Verfahrens bei weitem.
- Die gewählte Abstiegsfunktion kann das Konvergenzverhalten stark beeinträchtigen. Neben dem Maratos-Effekt können auch schlechte Werte für die Penalty-Faktoren der Grund sein.
- Das Verfahren ist sehr skalierungsempfindlich. Variablen, Zielfunktion und Nebenbedingungen müssen aufeinander abgestimmt sein. Luenberger [85] erwartet, daß für dieses Verfahren eine Selbstskalierung noch wichtiger ist als für unbeschränkte Quasi-Newton-Verfahren. Insgesamt ist viel Erfahrung notwendig, um in den Genuß aller Vorzüge dieser Methode zu gelangen.
- Die Methode nähert die Lösung meist vom unzulässigen Bereich her an.

In Bild 2.21 sind typische Konvergenzverläufe des Verfahrens für die Fälle a) und b) des zweiständigen Fachwerks (Abschnitt 2.2) dargestellt. Gestrichelt sind Höhenlinien der quadratischen Unterprobleme eingetragen. Es ist zu erkennen, wie ausgehend von Diagonalmatrix (Kreis) sich die Approximation der Hessematrix im Laufe der Iteration infolge der Einflüsse der aktiven Nebenbedingungen ändert. Der relativ unruhige Iterationsverlauf in Bild 2.21b belegt, daß dazu "Testschritte" notwendig sind.

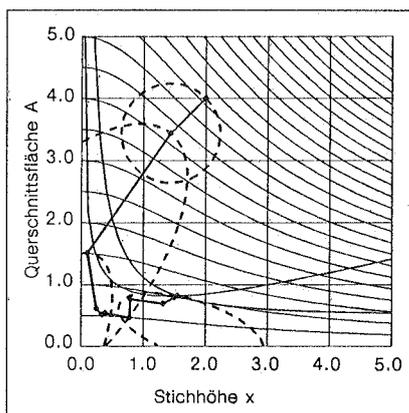
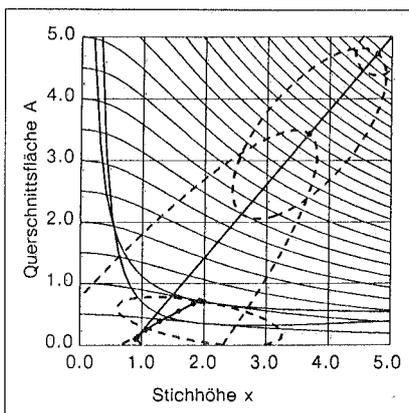


Bild 2.21a: Fachwerkproblem, Fall a)

Bild 2.21b: Fachwerkproblem, Fall b)

Bild 2.21: Iterationsverläufe des Han-Powell-Verfahrens nach Schittkowski

#### **2.8.6.4 Bedeutung von Skalierung und Line Search in der Sequentiellen Quadratischen Programmierung**

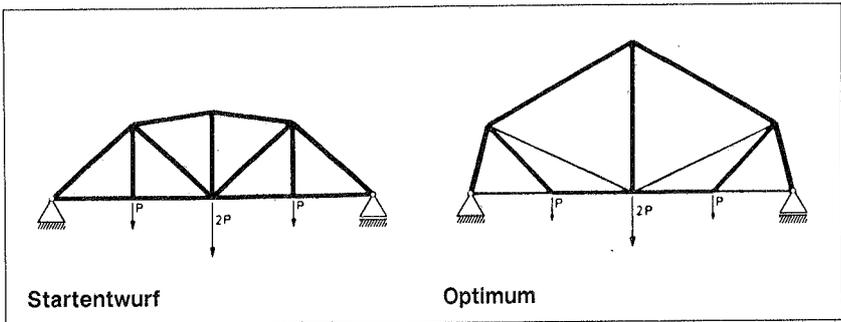
Die Anwendung von Quasi-Newton-Verfahren im Rahmen der SQP hat dieselben Konsequenzen wie bei unbeschränkten Problemen: die Konvergenzrate ist stark von den Eigenwertverhältnissen der approximierten Hessematrix abhängig. Zusätzlich zu den in Abschnitt 2.8.2.2 ausführlich geschilderten Einflüssen von Skalierungen der Zielfunktion, der Variablen oder direkt der Hessematrix kommen hier diejenigen der Nebenbedingungen und Lagrangemultiplikatoren hinzu. Anders als bei den unbeschränkten Problemen konnte für beschränkte Probleme im Schrifttum keine der Abschnitt 2.8.2.2 entsprechende Theorie zur automatischen Abschätzung und Berücksichtigung dieser Effekte gefunden werden. Gill et al. geben in [53] einige Anhaltspunkte zur richtigen Modellierung und Skalierung. Aus der eigenen Erfahrung mit der Anwendung der Methode in der Version von Schittkowski [117] können folgende Empfehlungen ausgesprochen werden.

##### **Skalierung der Variablen**

Besonders bei gemischten Form- und Querschnittsoptimierungen empfiehlt es sich, die Variablen mit einem ihrer charakteristischen Dimension entsprechendem Wert zu skalieren. Als eine einfache Regel, die sich in der Anwendung bewährt hat, können die Startwerte der Variablen verwendet werden.

##### **Skalierung der Zielfunktion**

Bei wenig oder im Idealfall nicht beschränkten Problemen können die im Abschnitt 2.8.2.2 beschriebenen Techniken zur Skalierung der Hessematrix bzw. zur Bestimmung eines Wertes für die Skalierung der Zielfunktion verwendet werden. Dieser Wert sollte allerdings in der Regel nur als eine grobe Abschätzung der Krümmungsverhältnisse verstanden werden, da die Hessematrix der Lagrangefunktion starken Metrikänderungen während der Iteration ausgesetzt ist (Bild 2.21). Für lineare Zielfunktionen kann kein Faktor auf diese Weise gefunden werden. Grundsätzlich sollte das Iterationsverhalten während der ersten Schritte beobachtet werden: ist die Verbesserung gering bei gleichzeitig idealer Schrittweite eins, ist oft die angenommene Krümmung zu groß und der Skalierungsfaktor kann vergrößert werden. Werden dagegen schon am Anfang viele linearesearches mit eventuell drastischen Verringerungen der Schrittlänge durchgeführt, könnte der Skalierungsfaktor zu groß sein und sollte deshalb reduziert werden. Tafel 2.8 verdeutlicht diese Regel. Sie gibt die Anzahl der Funktions- und Gradientenauswertungen sowie der Iterationen in Abhängigkeit der Zielfunktionskalierung bei der Optimierung des in Bild 2.22 dargestellten 13-stäbigen Fachwerkes wieder.



**Bild 2.22:** 13-stübiges Fachwerk

Das Problem hat drei Form- und sieben Querschnittsvariablen. Die Zielfunktion ist Gewicht, und es sind sieben Spannungsnebenbedingungen formuliert. Obwohl dieses Beispiel beschränkt ist, erweisen sich hier die automatische Skalierung der Hessematrix sowie die Technik zur Bestimmung eines Skalierungsfaktors für die Zielfunktion als erfolgreich, wie dies Tafel 2.9 entnommen werden kann.

SCL	NFUNC	NSEN	ITER	
10,000	16	90	12	SCL ... Zielfunktionsskalierung
1,000	15	83	11	NFUN ... Anzahl der Funktionsauswertungen
0,100	15	83	11	NSEN ... Anzahl der Gradientenbestimmungen (jede aktive Nebenbed. einzeln gezählt)
0,010	15	70	9	
0,001	23	147	22	ITER ... Anzahl der Iterationsschritte

**Tafel 2.8:** 13-stübiges Fachwerk; Auswirkung der Zielfunktionsskalierung

V	Z	H	SCL	NFUNC	NSEN	ITER
			1,0	30	210	29
	x		683,1	12	77	10
		x	1,0	14	91	11
	x	x	683,1	13	77	10
x			1,0	15	83	11
x	x		0,0136	11	70	9
x		x	1,0	12	76	10
x	x	x	0,0136	11	70	9

V ... Skalierung der Variablen Z ... der Zielfunktion H ... der Hessematrix

**Tafel 2.9:** 13-stübiges Fachwerk; Auswirkung verschiedener Skalierungen

### Skalierung der Nebenbedingungen

Wie direkt aus den Kuhn-Tucker-Bedingungen folgt, wirkt sich eine Skalierung der Nebenbedingungen auf die Lagrangemultiplikatoren aus:

$$\hat{g}_j = s_{gj} g_j \quad \rightarrow \quad \hat{\lambda}_j = \frac{1}{s_{gj}} \lambda_j \quad (2.107)$$

$s_{gj}$  ... Skalierungsfaktor der  $j$ -ten Nebenbedingung

Dies hat keine Auswirkungen auf den primären Teil der Lösung eines quadratischen Unterproblems und somit auf den Iterationsverlauf, von numerischen Einflüssen abgesehen. Andererseits wirkt sich die Skalierung der Nebenbedingungen auch auf die Penaltyfaktoren der Abstiegsfunktion aus. Dabei ändert sich am von Schittkowski bevorzugten quadratischen Penaltyterm nichts, wenn der Skalierungsfaktor quadratisch reziprok bei den Penaltyfaktoren berücksichtigt wird. Wegen:

$$P_j = \frac{1}{2} r_j g_j^2 = \frac{1}{2} \frac{r_j}{s_{gj}^2} (s_{gj} g_j)^2 \quad (2.108)$$

Darauf nimmt auch die von Schittkowski vorgeschlagene Adaptionvorschrift für die Penaltyfaktoren Rücksicht. Sie lautet:

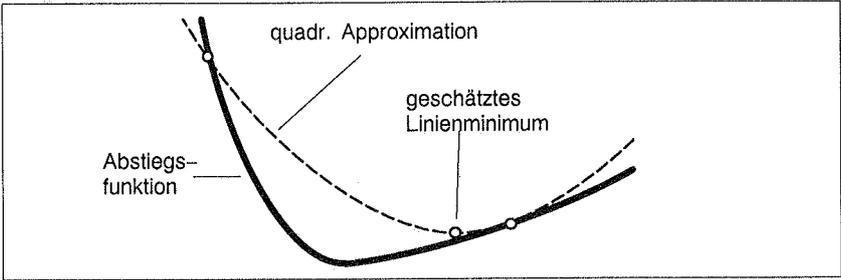
$$r_{jk+1} = \max \left[ \sigma_{jk} r_{jk}, \frac{2m(\lambda_{jk+1} - \lambda_{jk})^2}{\mathbf{p}_k^T \mathbf{B}_k \mathbf{p}_k} \right]; \quad j = 1, \dots, m \quad (2.109)$$

$$\text{mit } \sigma_{jk} = \min \left( 1, \frac{k}{\sqrt{r_{jk}}} \right)$$

Im Regelfall, wenn der zweite Term im Klammerausdruck von (2.109) greift, wird wegen (2.107) die Skalierung der Nebenbedingungen konsistent an die Penaltyfaktoren weitergegeben. Dann bleiben auch die Abstiegsfunktion und somit der Iterationsverlauf unverändert. Trifft allerdings der erste Term zu oder werden die Penaltyfaktoren aus numerischen Gründen auf die im Algorithmus vorgesehenen unteren bzw. oberen zulässigen Schranken gesetzt, wird die Abstiegsfunktion meist nachteilig beeinflusst. Entweder gelingt es nicht mehr, die erweiterte Lagrangefunktion zu konvexifizieren, weil die Penaltyfaktoren nicht groß genug sind, oder es tritt der nachfolgend beschriebene Effekt im Zusammenhang mit dem line search auf, wenn die Penaltyfaktoren nicht klein genug werden können.

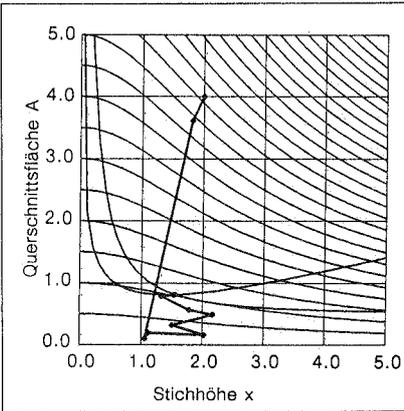
### Der Line Search

Wenn sich die Penaltyfaktoren nicht frei einstellen können, ergeben sich in der Regel sehr unausgeglichene Abstiegsfunktionen. Dies kann auch bei einer unzutreffenden Skalierung der Zielfunktion eintreten. Ein solcher Fall ist in Bild 2.23 skizziert.

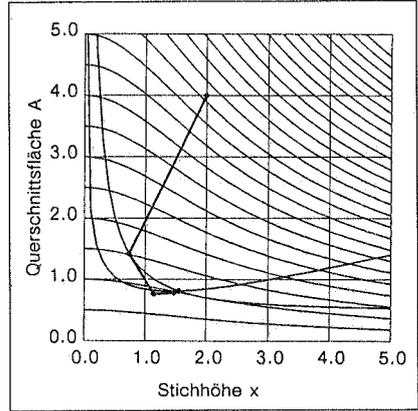


**Bild 2.23:** *mangelhafte Approximation einer unausgegliehenen Abstiegsfunktion*

Nähert sich der line search-Algorithmus vom flachen Teil der Kurve (im Bild von rechts) und wird, wie von Schittkowski vorgeschlagen, die Armijo-Regel in Verbindung mit einer "unsymmetrischen", quadratischen Interpolation (Fall a) in Abschnitt 2.8.1.1) verwendet, wird die Schrittweite zum Linienminimum wesentlich zu kurz geschätzt. Die Konvergenzgeschwindigkeit nimmt drastisch ab. Abhilfe kann hier neben einer geänderten Skalierung (Zielfunktion bzw. Nebenbedingungen) eine kubische Interpolation schaffen, die auch die Gradienten am steilen Ende der Kurve berücksichtigt. Allerdings gilt es hier bei der praktischen Anwendung, die Vorteile des verbesserten line searches gegenüber dem steigenden Rechenaufwand abzuschätzen. Bild 2.24 zeigt die unter-



**Bild 2.24a:** *quadr. Interpolation*



**Bild 2.24b:** *kubische Interpolation*

**Bild 2.24:** *Auswirkung verschiedener line search Interpolationen*

schiedlichen Iterationsverläufe infolge einer zu starken Skalierung der Nebenbedingungen für eine quadratische und eine kubische Interpolation. In Bild 2.24a fällt der erste Iterationsschritt wegen der unvollkommenen quadratischen Interpolation zu kurz aus mit auch weiterhin dramatischen Auswirkungen. Dagegen ist die kubische Interpolation wesentlich flexibler und führt zu dem in Bild 2.24b gezeigten gleichmäßigen und schnellen Iterationsverlauf.

### 2.8.6.5 Sequentiell-Lineare Programmierung als ein Verfahren der Lagrange-Methoden

Anstelle der rekursiven Lösung von quadratischen Unterproblemen, können die notwendigen Kuhn-Tucker-Bedingungen auch durch eine Reihe von linearen Programmen iterativ gelöst werden. Dabei läßt sich das lineare Teilproblem aus einem besonderen Fall der quadratischen Unterprobleme des Newton-Zyklus herleiten. Setzt man in Gleichung (2.97) anstelle der Hesse-Matrix die Einheitsmatrix, so erhält man folgende Aufgabe:

$$\begin{aligned} \min \quad & \frac{1}{2} \mathbf{p}_k^T \mathbf{p}_k + \nabla f(\mathbf{x}_k) \mathbf{p}_k \\ \text{mit:} \quad & \nabla \mathbf{h}(\mathbf{x}_k) \mathbf{p}_k + \mathbf{h}(\mathbf{x}_k) = \mathbf{0} \\ & \nabla \mathbf{g}(\mathbf{x}_k) \mathbf{p}_k + \mathbf{g}(\mathbf{x}_k) \leq \mathbf{0} \end{aligned} \quad (2.110)$$

Dieses Problem ist äquivalent mit dem folgenden, wenn für  $\epsilon$  der richtige Wert gewählt wird:

$$\begin{aligned} \min \quad & \nabla f(\mathbf{x}_k) \mathbf{p}_k \\ \text{mit:} \quad & \nabla \mathbf{h}(\mathbf{x}_k) \mathbf{p}_k + \mathbf{h}(\mathbf{x}_k) = \mathbf{0} \\ & \nabla \mathbf{g}(\mathbf{x}_k) \mathbf{p}_k + \mathbf{g}(\mathbf{x}_k) \leq \mathbf{0} \\ \text{und:} \quad & \frac{1}{2} \mathbf{p}_k^T \mathbf{p}_k \leq \epsilon \end{aligned} \quad (2.111)$$

Das quadratische Problem wurde durch ein vorwiegend lineares Problem ersetzt, wobei sich die Verbesserung  $\mathbf{p}_k$  nur innerhalb einer Hyperkugel mit Radius  $\epsilon$  einstellen darf. Ersetzt man diese Hyperkugel durch einen Hyperkubus, z.B.:

$$\sqrt{2} \epsilon \leq p_{ki} \leq \sqrt{2} \epsilon \quad ; \quad i = 1, \dots, n \quad (2.112)$$

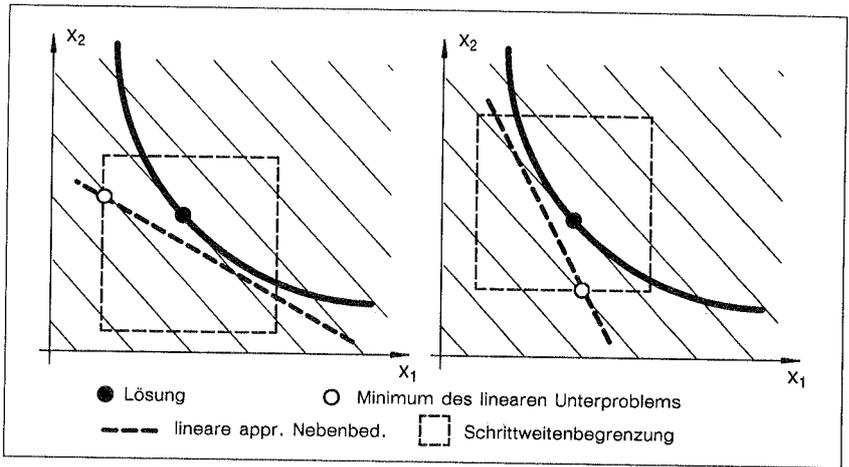
so ist das Unterproblem vollständig linearisiert und kann mit dem Simplexverfahren von Dantzig (siehe z.B. in [85]) gelöst werden. Damit ergibt sich  $\mathbf{x}_{k+1}$ , an dem eine

erneute Linearisierung vorgenommen wird. Auf diese Weise werden die Kuhn–Tucker-Bedingungen des ursprünglichen, nichtlinearen Problems sequentiell linear gelöst.

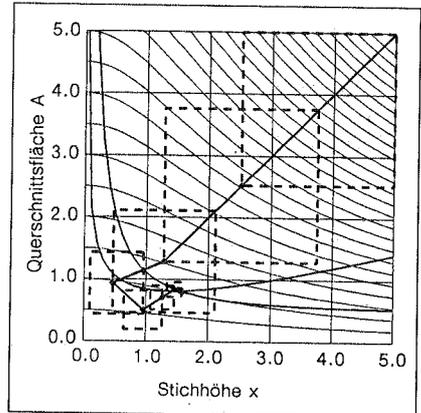
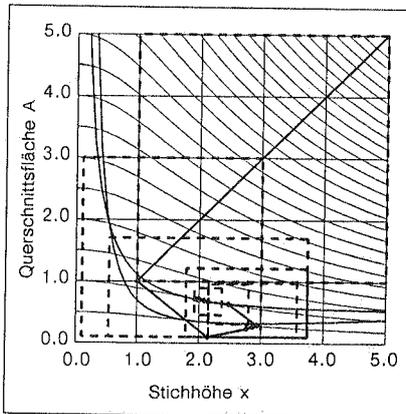
Es wurden schon sehr früh nichtlineare Probleme auf diese Weise angegangen, siehe z.B. bei Knepe [78], der Arbeiten von Cheney und Goldstein sowie Griffith und Stewart zitiert. Allerdings ist das oben skizzierte Vorgehen nicht mit dem Schnittebenenverfahren von Kelley (siehe z.B. in [81]) zu verwechseln, das auch auf einer Linearisierungsstrategie beruht. Kelleys Verfahren schränkt den Fortschritt  $p_k$  nicht durch eine Schrittweitenbegrenzung wie in Gl. (2.112) ein, sondern beläßt die linearisierten Nebenbedingungen des vorherigen Schrittes als zusätzliche Bedingungen in den nachfolgenden linearen Unterproblemen. So wird um den zulässigen Bereich eine dichte Schar von linearen Nebenbedingungen gelegt, die dem nichtlinearen Verlauf der Begrenzung folgen. Damit können ausschließlich konvexe Probleme gelöst werden, zu dem Preis, daß die Größe der Unterprobleme mit jedem Iterationsschritt anwächst.

In der Strukturoptimierung wird das Verfahren der sequentiellen linearen Programmierung gerne und häufig angewendet, siehe beispielsweise Reinschmidt [115], Zienkiewicz und Campbell [157], Baier [3] oder Knepe [78]. Ein glühender Verfechter der SLP ist Pauli Petersen, der keine Gelegenheit ausläßt, die Vorteile dieses Verfahrens zu propagieren [97]. Zweifellos hat das Verfahren einige wichtige Vorzüge: Es ist leicht anwendbar, unempfindlich gegenüber Skalierungen, das Simplexverfahren ist überall verfügbar. Dagegen verweist Arora [1] auf den wunden Punkt der SLP, die Adaption der Schrittweitenbegrenzung, und empfiehlt es aus diesem Grunde nicht. Tatsächlich ist es so, daß die Wahl der Schrittweitenbegrenzung nur dann ein Problem darstellt, wenn die Lösung des nichtlinearen Problems nicht in einer Ecke des zulässigen Bereiches liegt (siehe Bild 2.25). Dann oszilliert das Verfahren aufgrund der fehlenden Krümmungsinformation und des fehlenden line searches um die richtige Lösung.

Konvergenz kann dann nur durch die Kontraktion des Hyperkubus erzielt werden. Die Regeln für die Adaption der Schrittweitenbegrenzungen sind allesamt heuristisch, so daß sich das Verfahren einer genaueren theoretischen Beurteilung der Konvergenzrate entzieht. Liegt jedoch die Lösung an einer Ecke des zulässigen Bereiches, kann es dieselbe Konvergenzgeschwindigkeit wie Newton–Verfahren erreichen. Aus den genannten Gründen empfiehlt sich der Einsatz eines SLP–Verfahrens nur für Probleme mit vielen Nebenbedingungen wie zum Beispiel Probleme der Querschnittsoptimierung.



**Bild 2.25:** Oszillieren des SLP-Verfahrens in der Nähe der Lösung



**Bild 2.26a:** Fachwerkproblem, Fall a)

**Bild 2.26b:** Fachwerkproblem, Fall b)

**Bild 2.26:** Iterationsverläufe mit dem Verfahren der sequentiellen linearen Programmierung

Im folgenden wird ein SLP-Algorithmus beschrieben, der in dieser Arbeit Verwendung fand [80]. Die Entwicklung des beschriebenen Verfahrens lehnte sich an die Arbeiten von Baier [3] und Knepe [78] an. Im Gegensatz zu Knepe wird in Schritt (7) der line search nicht an der Zielfunktion vorgenommen, sondern die  $L_1$ -Penalty-Funktion verwendet. In Bild 2.26 sind mit diesem SLP-Algorithmus ermittelte Iterationsverläufe für die Fälle a) und b) des zweiständigen Fachwerkes angegeben. Im Fall a) mit nur einer am Optimum aktiven Nebenbedingung ist die Kontraktion des Hyperkubus für die Ga-

rantie der Konvergenz unerlässlich. Fall b) mit zwei aktiven Nebenbedingungen ist für die SLP ideal geeignet.

Der Algorithmus lautet:

1. Setze die Startwerte;  $k = 0$   
 wähle  $\beta, \rho, \Theta, \theta$ .  
 Variablen  $\mathbf{x}_0$ ; Adaptionfaktor  $\gamma_0 = 1$   
 Zielfunktion  $f(\mathbf{x}_0), \nabla f(\mathbf{x}_0)$   
 Nebenbedingungen  $\mathbf{h}(\mathbf{x}_0), \nabla \mathbf{h}(\mathbf{x}_0), \mathbf{g}(\mathbf{x}_0), \nabla \mathbf{g}(\mathbf{x}_0)$   
 Schrittweitenbegrenzungen  $\mathbf{p}_L = -\theta (\mathbf{x}_U - \mathbf{x}_L)$   
 $\mathbf{p}_U = \theta (\mathbf{x}_U - \mathbf{x}_L)$
2. Konvergenzabfrage (wie bei der Han-Powell-Methode).
3. Stelle das Simplextableau auf. Für  $k > 0$  adaptiere die Begrenzung der Schrittweite nach:

$$\left. \begin{array}{l} \text{a) Baier} \\ \text{b) Kneppe} \end{array} \right\} \begin{array}{l} \gamma_k = \frac{\gamma_{k-1}}{1 + \gamma_{k-1}^2} \\ \gamma_k = \frac{\gamma_{k-1}}{1 + \gamma_{k-1}} \end{array} \quad \left. \begin{array}{l} \mathbf{p}_L = -\frac{1}{2} \gamma_k (\mathbf{x}_U - \mathbf{x}_L) \\ \mathbf{p}_U = \frac{1}{2} \gamma_k (\mathbf{x}_U - \mathbf{x}_L) \end{array} \right\}$$

4. Löse das lineare Programm:

$$\begin{array}{ll} \min & \nabla f(\mathbf{x}_k)^T \mathbf{p}_k \\ \text{mit:} & \nabla \mathbf{h}(\mathbf{x}_k)^T \mathbf{p}_k + \mathbf{h}(\mathbf{x}_k) = \mathbf{0} \\ & \nabla \mathbf{g}(\mathbf{x}_k)^T \mathbf{p}_k + \mathbf{g}(\mathbf{x}_k) \leq \mathbf{0} \\ & \mathbf{p}_L \leq \mathbf{p}_k \leq \mathbf{p}_U \end{array}$$

5. Im Falle inkonsistenter Nebenbedingungen, vergrößere den Hyperkubus der Schrittweitenbegrenzungen und löse erneut. Ist das Problem immer noch inkonsistent, relaxiere die Nebenbedingungen.

$$\begin{array}{ll} \min & \nabla f(\mathbf{x}_k)^T \mathbf{p}_k + \varrho \cdot \delta \\ \text{mit:} & \nabla \mathbf{h}(\mathbf{x}_k)^T \mathbf{p}_k + (1 - \delta) \mathbf{h}(\mathbf{x}_k) = \mathbf{0} \\ & \nabla \mathbf{g}(\mathbf{x}_k)^T \mathbf{p}_k + (1 - \delta) \mathbf{g}(\mathbf{x}_k) \leq \mathbf{0} \\ & 0 \leq \delta \leq 1 \\ & \mathbf{p}_L \leq \mathbf{p}_k \leq \mathbf{p}_U \end{array}$$

Ist das Problem wiederholt inkonsistent, breche den Algorithmus ab.

6. Bestimme die neuen Variablenwerte:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{p}_k$$

7. Untersuche die Variablen auf Oszillieren:

$$(x_{ik+1} - x_{ik}) \cdot (x_{ik} - x_{ik-1}) < 0 \quad ?$$

Ozilliert eine Variable, dann:

setze  $p_{Li} = \Theta_i p_{Li} \quad ; \quad p_{Ui} = \Theta_i p_{Ui} \quad ; \quad 0 < \Theta \leq 1$

und a) setze  $x_{ik+1} = x_{ik} + \beta p_{ik} \quad ; \quad 0 < \beta \leq 1$

oder b) führe einen line search an der  $L_1$ -Penalty-Funktion durch,  
setze  $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha \mathbf{p}_k$

Verwendet wird dabei ein Armijo-line search mit einem Steigungsfaktor von 0,2 für die Sekante und wahlweise quadratischer oder kubischer Interpolation.

8. Bestimme neue Funktionswerte und Ableitungen, setze  $k = k + 1$  und gehe nach 2.).



**Bemerkungen zu 3.) – Schrittweitenadaption:** Die Schrittweitenbegrenzung (auch "move limits") definiert den Hyperkubus, der dafür sorgt, daß die linearen Unterprobleme beschränkt sind. Bei Problemen, deren Lösung nicht in der Ecke des zulässigen Bereichs liegt, kann das Verfahren nur konvergieren, wenn sich der Hyperkubus im Verlauf der Iteration einschnürt. Ursprünglich schlug Baier [3] vor, die Schrittweitenbegrenzung einer Variablen dann mit dem angegebenen Faktor  $\gamma_k$  zu skalieren, wenn die betreffende Variable oszilliert. Er empfahl, mit  $\gamma_0 = 0,6 \div 0,8$  zu beginnen. Knepe [78] modifizierte diese Regel und verringert in jeder Iteration den Hyperkubus mit  $\gamma_k$ . Er beginnt dabei mit  $\gamma_0 = 1$ .

**Bemerkungen zu Schritt 7) – Oszillieren:** Zusätzlich wird die entsprechende Seitenlänge des Hyperkubus verringert, wenn eine Variable oszilliert. Auf diese Weise richtet sich der Hyperkubus aus, was der Anpassung der Hesse-Matrix bei den SQP-Verfahren entspricht. Darüber hinaus schlägt Knepe vor, im Falle oszillierender Variablen einen line search durchzuführen, um so den Konvergenzverlauf zu stabilisieren. Als Abstiegsfunktion wählt er die Zielfunktion. Hier wurde dieses Vorgehen modifiziert und die  $L_1$ -Penalty-Funktion eingesetzt. Dabei wird mit der Armijo-Regel ein einziger Suchschritt mit

einem relativ großen Sekantensteigungsfaktor ( $0,1 \div 0,3$ ) und wahlweise quadratischer oder kubischer Interpolation durchgeführt.

Bemerkungen zu 5.) – inkonsistente Nebenbedingungen: Wenn der Entwicklungspunkt unzulässig ist und die Schrittweitenbegrenzungen zu klein gewählt wurden, kann es deshalb vorkommen, daß das linearisierte Problem inkonsistent ist. In diesem Falle müssen die Grenzwerte des Hyperkubus vergrößert werden. Dagegen kann ein inkonsistentes Unterproblem auch durch mangelhafte Linearisierungen entstehen, obwohl das eigentliche Problem intakt ist. Dann hat eine Vergrößerung des Hyperkubus keinen Erfolg. In diesem Falle müssen die Nebenbedingungen relaxiert werden.

## 3 APPROXIMATIONSTECHNIKEN

### 3.1 Überblick

Nach einer stürmischen Entwicklung der Strukturoptimierung mit Mitteln der mathematischen Programmierung in den 60iger Jahren [122] mußte mit Ernüchterung festgestellt werden, daß die damals zur Verfügung stehenden Verfahren, Algorithmen und Computer nicht in der Lage waren, realitätsnahe Probleme mit vielen Variablen und Nebenbedingungen zu bearbeiten. In der Folge gewannen die Verfahren der Optimalitätskriterien verstärkt an Bedeutung [10], [51], [146], [147]. Ihr wesentlichstes Charakteristikum vom Standpunkt der Lösungsgeschwindigkeit ist, daß die Zahl der Iterationen nicht von der Anzahl der Variablen, sondern von Tragwerksverhalten sowie Art und Anzahl der Nebenbedingungen abhängig ist [10]. Damit konnten vor allem für Fälle mit nur einer Nebenbedingung Probleme mit hunderten bis tausenden von Variablen, wie sie in der Luft- und Raumfahrtindustrie auftreten können, effektiv in wenigen Iterationsschritten (oft weniger als 10) gelöst werden. Allerdings zu dem Preis, daß oft das mathematisch richtige Optimum (Erfüllung der Kuhn-Tucker-Bedingungen) nicht erreicht werden kann und mehrere Nebenbedingungen nur mit Mühe gleichzeitig erfaßt werden können.

Die Effektivität dieser Verfahren stellte eine große Herausforderung für die Vertreter der mathematischen Programmierung dar. Es wurden deshalb Techniken entwickelt, die auf verschiedene Weise das Ziel verfolgen, die Effizienz der Methoden zu steigern. Die Grundidee ist, das komplexe Originalproblem durch ein möglichst exakt approximiertes, jedoch leichter und effizienter zu lösendes Unterproblem zu ersetzen.

Dabei haben sich folgende Approximationstechniken bewährt:

- Auswahl der problemspezifischen Variablen  
(Verringerung der Variablenzahl)
- das Konzept der reduzierten Basis  
(Verringerung der Variablenzahl)
- das Streichen nichtkritischer Nebenbedingungen  
(Verringerung der Problemkomplexität)
- Ersetzen der Zielfunktion und Nebenbedingungen durch explizite Approximationsfunktionen  
(Verringerung der Problemkomplexität, schnellere Funktionsauswertung)
- genäherte Bestimmung der zweiten Ableitungen  
(verbesserte Problembeschreibung, Einsatz höherwertigerer Lösungsverfahren)

Die Entwicklung dieser Techniken und deren Einsatz geht zurück bis in die späten 60iger und frühen 70iger Jahre. Bei Reinschmidt et al. [115] kann bereits eine Approximation

der Nebenbedingungen gefunden werden. Einen Überblick über die damaligen Techniken geben Schmit und Farshi [119] 1974. Neuere Arbeiten von Prasad [107], Lust und Schmit [86], Haftka [60], [63] sowie Fleury [45] geben den aktuellen Entwicklungsstand wieder. Die neuesten Tendenzen weisen in Richtung einer zunehmenden Berücksichtigung der zweiten Ableitungen.

Es ist eine Frage des persönlichen Standpunktes, ob die beiden ersten Punkte der voranstehenden Liste den Approximationstechniken zugeschlagen werden oder ob sie im Lichte der Problemformulierung gesehen werden. Werden zur Auswahl der spezifischen Variablen die Ergebnisse einer Sensitivitätsanalyse herangezogen bzw. die Basisvektoren aufgrund deterministischer Regeln gefunden, handelt es sich zweifellos um Approximationen. Besonders bei der Formoptimierung ist die Problemformulierung jedoch aus vielerlei Gründen derart stark restringiert, wobei geometrische und technische Beschränkungen, nicht näher bestimmbare Formvorstellungen sowie die Grenzen der Anwendbarkeit der Strukturanalyseverfahren eine Rolle spielen, so daß sie nur mit viel Erfahrung vorgenommen werden kann. Hier zeigt sich eine besondere Spielart der reduzierten Basentechnik als besonders hilfreich, wobei die Basisvektoren bei der Problemmodellierung als Problemparameter festgelegt werden und im weiteren unverändert bleiben. In diesem Falle sind diese Techniken Bestandteil der im weitesten Sinne willkürlichen und subjektiven Problemformulierung. In diesem Sinne wird in Kapitel 5 "Variablenverknüpfungen" näher darauf eingegangen.

Mit dem Streichen nichtkritischer Nebenbedingungen soll die Problemgröße verringert werden. Hierbei handelt es sich um eine Vorauswahl des Satzes der aktiven und möglicherweise aktiv werdenden Nebenbedingungen, der in den nachfolgenden Iterationen jedoch nicht mehr überprüft und gegebenenfalls korrigiert wird. Deshalb werden zur Sicherheit zusätzliche Schrittweitenbegrenzungen eingeführt, die ein Ausbrechen des Lösungsfortschrittes verhindern sollen, falls eine wichtige Nebenbedingung nicht erfaßt worden ist. Zur Auswahl der Nebenbedingungen müssen sie verglichen und beurteilt werden. Dazu werden sie mit einer charakteristischen Größe skaliert. Bei Spannungsnebenbedingungen können dies die zulässigen Spannungen sein. Damit gewinnen die Nebenbedingungen folgende Form:

$$g_j = R_j - 1 \leq 0 ; \quad j = 1, \dots, m \quad (3.1)$$

mit:  $R_j$  ... "response ratio"  
 z.B.  $R_j = \frac{\sigma_j}{\sigma_{zul}}$

Gestrichen werden Nebenbedingungen dann, wenn gilt:

$$R_j < 0,33 R_{\max} \quad ; \text{ nach [119]} \quad (3.2)$$

$$R_j < \min \{ \max \{ R_{\max}, 0.3 \}, 0.7 \} \quad ; \text{ nach [86]} \quad (3.3)$$

oder:

$$g_j < \min \{ 2 g_{\max}, 0, g_{\max} - 0.2 \} \quad ; \text{ nach [60]} \quad (3.4)$$

mit:

$$R_{\max} = \max \{ R_1, \dots, R_m \} \quad ; \quad g_{\max} = \max \{ g_1, \dots, g_m \}$$

Die Regeln sind heuristischer Natur und sind aus der Erfahrung ihrer Verfasser gewonnen.

Auf die Approximation von Zielfunktion und Nebenbedingungen durch explizit auswertbare Funktionen wird im nächsten Abschnitt ausführlich eingegangen. Auch hier ergibt sich die zwingende Notwendigkeit zusätzlicher Schrittweitenbegrenzungen. Denn im Regelfall gelten die getroffenen Approximationen nur in der Nähe des Entwicklungspunktes. Allerdings muß die Adaption der Schrittweitenbegrenzungen sehr sorgfältig vorgenommen werden, da sie über Konvergenzgeschwindigkeit und -sicherheit entscheidet, wie dies auch beim Verfahren der sequentiellen linearen Programmierung der Fall ist (Abschnitt 2.8.6.4).

Ein Verfahren der Strukturoptimierung, das sich der Approximationstechniken bedient, besteht im wesentlichen aus zwei ineinander geschachtelten Iterationsschleifen. In der äußeren werden die Approximationen vorgenommen und die Schrittweitenbegrenzungen festgelegt. In der inneren löst ein Verfahren der mathematischen Programmierung das approximierte Problem (siehe Bild 3.1). Dieses Vorgehen ist insbesondere dann der direkten Anwendung der mathematischen Programmierung überlegen, wenn Line Searches notwendig werden, die bei der direkten Anwendung zu wiederholten, aufwendigen Strukturberechnungen führen, bei der Verwendung der Approximationstechniken jedoch nur die Auswertung eines explizit formulierten Optimierungsproblems bedeuten. Auf dieser Grundlage sind viele bekannte Programme konzipiert. Eines der ältesten ist ACCESS [120], [43] das auch in seinem Namen auf diesen Umstand hinweist: Approximation Concepts Code for Efficient Structural Synthesis. In Verbindung mit dualen Lösungsmethoden zeigen Approximationstechniken Eigenschaften wie die Optimalitätskriterienmethoden, ohne jedoch deren Nachteile zu besitzen. Dazu mehr in Abschnitt 3.2.3.

Die Erfassung der zweiten Ableitungen ist noch Gegenstand der Forschung, obwohl auch hier schon Erfahrungen seit längerem vorliegen [89]. In dieser Arbeit wurde dies bei der Definition expliziter Approximationsfunktionen weiter verfolgt. Darüber hinaus wird auf die oben angeführte Literatur verwiesen.

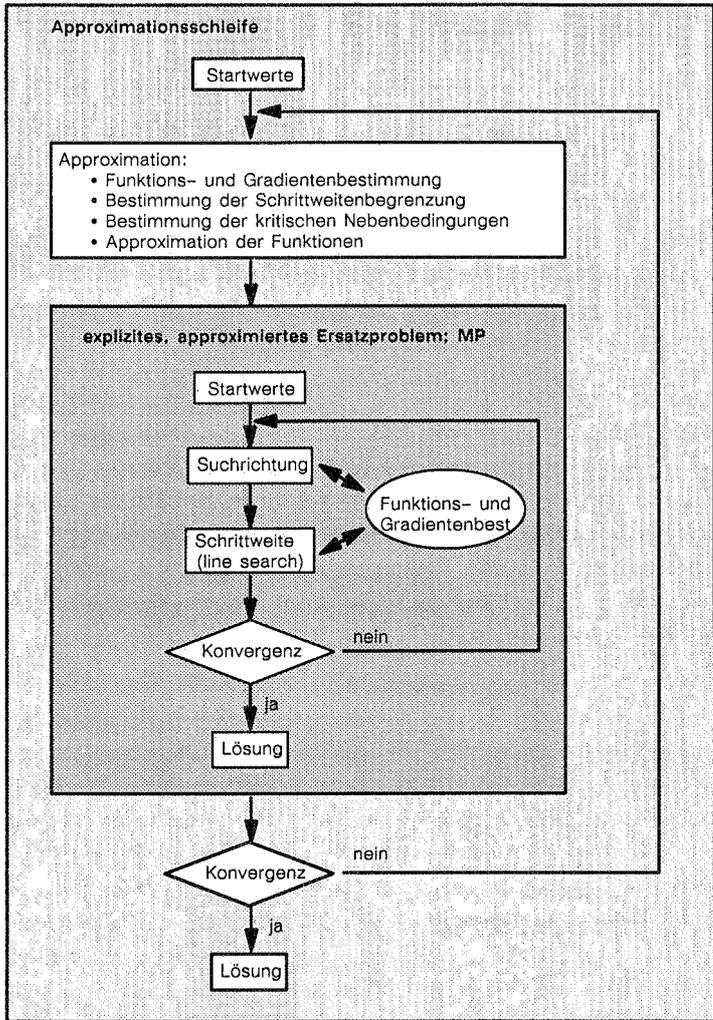


Bild 3.1: Ablauf einer Strukturoptimierung mit Approximations-Techniken

## 3.2 Explizite Approximation von Zielfunktion und Nebenbedingungen

### 3.2.1 Physikalische Begründung, konvexe Linearisierung

Für eine Reihe von Spezialfällen lassen sich einfache Approximationsformeln für Zielfunktion und Nebenbedingungen herleiten. Offensichtlich gelingt dies für den einfachsten Fall, der Querschnittsoptimierung statisch bestimmter Fachwerke, die eine bedeutende Rolle in der Entwicklungsgeschichte der Strukturoptimierung spielt. Die Zielfunktion "Gewicht" ist linear bezüglich der unbekanntenen Querschnittsflächen:

$$f(\mathbf{x}) = \gamma \sum_{i=1}^n (l_i x_i) \quad (3.5)$$

$\gamma$  ... spezifisches Gewicht

$l_i$  ... Stablänge

$x_i$  ... unbekannte Querschnittsfläche

$n$  ... Anzahl der Stäbe und Unbekannten

Für die Spannungsnebenbedingungen gilt:

$$g_i(x_i) = \begin{cases} \frac{F_i}{x_i} - \beta_s \leq 0 \text{ (Zug)} \\ -\frac{F_i}{x_i} - \beta_s \leq 0 \text{ (Druck)} \end{cases} \quad (3.6)$$

$F_i$  ... Stabkraft in Stab  $i$

$\beta_s$  ... Fließspannung

Die Spannungsnebenbedingungen sind linear bezüglich der reziproken Variablen  $y_i = 1/x_i$  und entkoppelt, da die Stabkräfte unabhängig von der Dimensionierung sind. Die Einführung der reziproken Variablen ist eine alte Technik und kann u.a. bei Reinschmidt [115] gefunden werden. Chern und Prager [28] weisen auf die Konvexifizierung des Problems hin, die mit der Einführung von  $y_i$  anstelle von  $x_i$  erreicht wird. Neben der Genauigkeit der Approximation ist dies eine weitere gewünschte Eigenschaft des Ersatzproblems, die eine unproblematische und eindeutige Lösung mit den Mitteln der mathematischen Programmierung garantieren soll. Mit den ersten beiden Gliedern einer Taylor-Reihe lassen sich folgende Approximationen angeben:

Gewicht (Entwicklung um  $\mathbf{x}_0$ ):

$$\tilde{f}(\mathbf{x}) = f(\mathbf{x}_0) + \sum_{i=1}^n \left( \frac{\partial f(\mathbf{x}_0)}{\partial x_i} (x_i - x_{i0}) \right) \quad (3.7)$$

Spannungsnebenbedingung (Entwicklung um  $y_0$ ):

$$\bar{g}_j(\mathbf{y}) = g_j(\mathbf{y}_0) + \sum_{i=1}^n \left( \frac{\partial g_j(\mathbf{y}_0)}{\partial y_i} (y_i - y_{i0}) \right) \quad (3.8)$$

oder:

$$\bar{g}_j(\mathbf{x}) = g_j(\mathbf{x}_0) + \sum_{i=1}^n \left( \frac{\partial g_j(\mathbf{x}_0)}{\partial x_i} \frac{dx_i(\mathbf{x}_0)}{dy_i} \left( \frac{1}{x_i} - \frac{1}{x_{i0}} \right) \right) \quad (3.9)$$

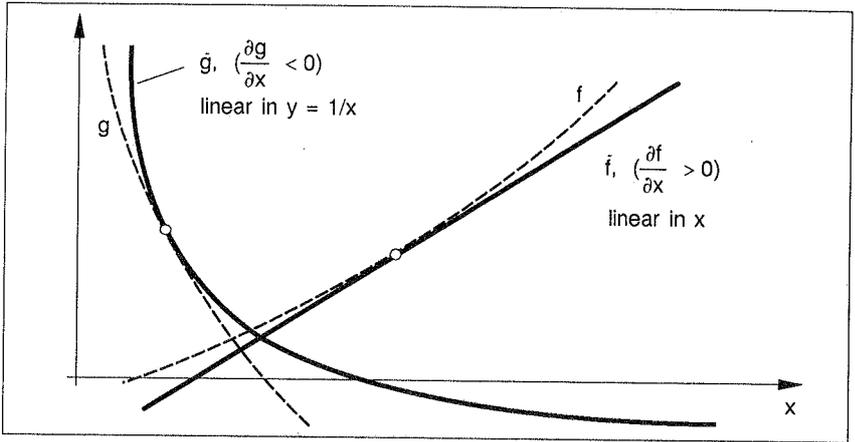
$$\text{mit: } y_i = \frac{1}{x_i} \quad ; \quad \frac{dx_i}{dy_i} = -x_i^2$$

Die angegebenen Approximationsvorschriften (3.7) bis (3.9) sind exakt, auch weit entfernt vom Entwicklungspunkt  $\mathbf{x}_0$  bzw.  $\mathbf{y}_0$ . Zu ihrer Erstellung sind nur die Funktionswerte und ersten Ableitungen am Entwicklungspunkt notwendig. Die Auswertung der hier explizit vorliegenden Approximationsfunktionen ist wesentlich effektiver als die der Originalfunktionen, da diese über den Umweg einer vollständigen Strukturanalyse nur implizit bestimmt werden können, die bei komplizierten Tragwerken nur mit der Methode der finiten Elemente durchgeführt werden kann.

Die Taylor-Entwicklung in den Original- bzw. inversen Variablen zur Bestimmung von Approximationsfunktionen wird nicht nur bei der Querschnittsoptimierung statisch bestimmter Fachwerke angewendet. Läßt sich die Auswertung auf statisch unbestimmte Fachwerke noch damit begründen, daß die optimale Lösung bei nur einem Lastfall in der Regel statisch bestimmt ist, ist die Anwendung für Formoptimierungen – nicht nur von Fachwerken – physikalisch nicht mehr begründbar. Gleichwohl zeigt die Erfahrung, daß auch in diesen Fällen die vorgeschlagenen Approximationstechniken erfolgreich eingesetzt werden können [22]. Dafür wird eine weitere Formalisierung durchgeführt, die von Fleury "konvexe Linearisierung" [44], von Starnes und Haftka [131] "hybride Approximation" genannt wird. Die Motivation ist hierbei, ein konvexes Ersatzproblem zu bestimmen, das möglichst noch konvexer als das Originalproblem ist. Dies wird auch "konservative Approximation" [131] genannt. Dabei entscheidet das Vorzeichen der partiellen Ableitungen, ob bezüglich  $x_i$  oder  $y_i$  linearisiert wird (Bild 3.2). Es gilt für die Zielfunktion (entsprechend für die Nebenbedingungen  $g_j$ ):

$$\bar{f}(\mathbf{x}) = f(\mathbf{x}_0) + \sum_{i=1}^n B_i \frac{\partial f(\mathbf{x}_0)}{\partial x_i} \quad (3.10)$$

$$\text{mit: } B_i = \begin{cases} (x_i - x_{i0}) & \text{für } \frac{\partial f(\mathbf{x}_0)}{\partial x_i} \geq 0 \\ -x_{i0}^2 \left( \frac{1}{x_i} - \frac{1}{x_{i0}} \right) & \text{für } \frac{\partial f(\mathbf{x}_0)}{\partial x_i} < 0 \end{cases}$$



**Bild 3.2:** hybride Approximation

Weitere Approximationsvorschriften werden von Prasad [106] angegeben, worunter die von ihm "generalized inverse-power approximation", kurz GPA, genannte Approximation besondere Beachtung verdient. Dort wird die Hilfsvariable  $y_i$  folgendermaßen definiert.

$$y_i = \begin{cases} \frac{1}{b-1} x_i^{1-b} & ; \text{für } b \neq 1 \\ -\ln(x_i) & ; \text{für } b = 1 \end{cases} \quad (3.11)$$

Damit ergibt sich für die approximierten Funktionen:

$$\tilde{f}(x) = f(x_0) + \sum_{i=1}^n \begin{cases} -\frac{1}{b-1} x_{i0}^b (x_i^{1-b} - x_{i0}^{1-b}) \frac{\partial f(x_0)}{\partial x_i} & ; \text{für } b \neq 1 \\ x_{i0} \ln\left(\frac{x_i}{x_{i0}}\right) \frac{\partial f(x_0)}{\partial x_i} & ; \text{für } b = 1 \end{cases} \quad (3.12)$$

Für  $b = 0$  erhält man die übliche Linearisierung in den ursprünglichen Variablen  $x$  und für  $b = 2$  diejenige in den reziproken Variablen, entsprechend den Gleichungen (3.8) und (3.9). Mit dem gewählten Ansatz (3.11) können auch Biegespannungen als Funktion der Balkenhöhe exakt approximiert werden. Bei konstanter Balkenbreite ist dann  $b = 3$  zu setzen.

Wie bereits erwähnt, führen die genannten Approximationstechniken nur in bestimmten Fällen zu exakten Beschreibungen. Meistens treffen die Voraussetzungen nicht zu, wie bei statisch unbestimmten Tragwerken, kombinierten Biege- und Normalkraftbeanspruchungen oder Problemen der Formoptimierung. Für diese Fälle schlägt Prasad vor, die Wahl des Exponenten  $b$  in (3.11) so zu wählen, daß ein ausgewogenes Gleichgewicht

zwischen Genauigkeit und Konservativität der Approximation erzielt wird. Für einige Spezialfälle gibt er untere und obere Grenzen für  $b$  an. Fleury [45] beschreibt einen Algorithmus, der sich an den Krümmungen von Zielfunktion und Nebenbedingungen orientiert und die Wahl des Exponenten während der Iteration automatisch justiert. Damit ist die Anwendbarkeit der geschilderten Approximationstechnik wesentlich erweitert. Im Rahmen dieser Arbeit wurden speziell für die Methode der beweglichen Asymptoten ähnliche Überlegungen durchgeführt, die zu einer Erweiterung des Verfahrens führten (siehe dazu Abschnitt 3.2.4.5). Schmit und Fleury beschreiben in [121] eine interessante Erweiterung des Verfahrens für Probleme mit kontinuierlichen sowie diskontinuierlichen Variablen.

### 3.2.2 Separierbarkeit und Dualität

Eine Anforderung an die zuvor geschilderten Approximationstechniken ist, daß das explizite Ersatzproblem konvex sein soll. Andererseits basieren die Approximationen auf Linearisierungen bezüglich der verallgemeinerten Variablen  $y$ . Damit ist das Ersatzproblem entkoppelt und erfüllt die Bedingungen der Separabilität. Da es konvex ist, können duale Lösungsalgorithmen effizient eingesetzt werden (siehe Abschnitt 2.8.5). Fleury [44], [42] schildert ein duales Lösungsschema in Verbindung mit der hybriden Approximation (3.10). Unabhängig davon gibt Svanberg [134] ein ähnliches Verfahren an. Sein Ersatzproblem lautet:

$$\begin{aligned} \tilde{f}(\mathbf{x}) &= r_0 + \sum_{i=1}^n \left( p_{0i} x_i + \frac{q_{0i}}{x_i} \right) \rightarrow \min \\ \text{und } \tilde{g}_j(\mathbf{x}) &= r_j + \sum_{i=1}^n \left( p_{ji} x_i + \frac{q_{ji}}{x_i} \right) \leq 0 \quad ; j = 1, \dots, m \\ & \quad \quad \quad x_{Li} \leq x_i \leq x_{ri} \quad ; i = 1, \dots, n \end{aligned} \tag{3.13}$$

mit den Parametern, für die Zielfunktion:

$$\begin{aligned} p_{0i} &= \begin{cases} \frac{\partial f(\mathbf{x}_0)}{\partial x_i} & ; \text{für } \frac{\partial f(\mathbf{x}_0)}{\partial x_i} > 0 \\ 0 & ; \text{für } \frac{\partial f(\mathbf{x}_0)}{\partial x_i} \leq 0 \end{cases} \\ q_{0i} &= \begin{cases} 0 & ; \text{für } \frac{\partial f(\mathbf{x}_0)}{\partial x_i} \geq 0 \\ -x_{i0}^2 \frac{\partial f(\mathbf{x}_0)}{\partial x_i} & ; \text{für } \frac{\partial f(\mathbf{x}_0)}{\partial x_i} < 0 \end{cases} \\ r_0 &= f(\mathbf{x}_0) - \sum_{i=1}^n \left( p_{0i} x_{i0} + \frac{q_{0i}}{x_{i0}} \right) \end{aligned} \tag{3.14}$$

und für die Nebenbedingungen:

$$\begin{aligned}
 p_{ji} &= \begin{cases} \frac{\partial g_j(\mathbf{x}_0)}{\partial x_i} & ; \text{für } \frac{\partial g_j(\mathbf{x}_0)}{\partial x_i} > 0 \\ 0 & ; \text{für } \frac{\partial g_j(\mathbf{x}_0)}{\partial x_i} \leq 0 \end{cases} \\
 q_{ji} &= \begin{cases} 0 & ; \text{für } \frac{\partial g_j(\mathbf{x}_0)}{\partial x_i} \geq 0 \\ -x_{i0}^2 \frac{\partial g_j(\mathbf{x}_0)}{\partial x_i} & ; \text{für } \frac{\partial g_j(\mathbf{x}_0)}{\partial x_i} < 0 \end{cases}
 \end{aligned} \tag{3.15}$$

$$r_j = g_j(\mathbf{x}_0) - \sum_{i=1}^n \left( p_{ji} x_{i0} + \frac{q_{ji}}{x_{i0}} \right)$$

$\mathbf{x}_0$  ... Entwicklungspunkt  
 $x_{i0}$  ... i-te Komponente von  $\mathbf{x}_0$

Die Lagrange-Funktion des approximierten Problems (3.13) ist dann:

$$\begin{aligned}
 \tilde{L}(\mathbf{x}, \lambda) &= r_0 + \sum_{i=1}^n \left( p_{0i} x_i + \frac{q_{0i}}{x_i} \right) \\
 &\quad + \sum_{j=1}^m \lambda_j \left\{ r_j + \sum_{i=1}^n \left( p_{ji} x_i + \frac{q_{ji}}{x_i} \right) \right\}
 \end{aligned} \tag{3.16}$$

Die Lösung von (3.13) ist ein Sattelpunkt der Lagrangefunktion (3.16). Da die Lagrangefunktion separierbar ist, kann die Bestimmung der dualen Funktion  $W(\lambda)$  als Teilaufgabe des zu (3.13) äquivalenten dualen Problems

$$\begin{aligned}
 \max_{\lambda} \left( \min_{\mathbf{x}} \tilde{L}(\mathbf{x}, \lambda) \right) &= \max_{\lambda} W(\lambda) ; \quad \lambda \geq 0 \\
 W(\lambda) &\dots \text{ duale Funktion}
 \end{aligned} \tag{3.17}$$

in n eindimensionale Minimierungsaufgaben zerlegt werden. Sie lauten:

$$W(\lambda) = \min_{\mathbf{x}} \tilde{L}(\mathbf{x}, \lambda) = \sum_{i=1}^n \left( \min_{x_i} \tilde{L}_i(x_i, \lambda) \right) + r_0 + \sum_{j=1}^m \lambda_j r_j \tag{3.18}$$

$$\text{mit: } \tilde{L}_i = p_{0i} x_i + \frac{q_{0i}}{x_i} + \sum_{j=1}^m \lambda_j \left( p_{ji} x_i + \frac{q_{ji}}{x_i} \right) \tag{3.19}$$

für gegebene Lagrange-Multiplikatoren  $\lambda_j$ .

Die Minimierungsaufgabe (3.18) kann explizit gelöst werden. Aus der Stationärbedingung:

$$\frac{\partial \bar{L}_i}{\partial x_i} = \bar{L}'_i = p_{0i} - \frac{q_{0i}}{x_i^2} + \sum_{j=1}^m \lambda_j \left( p_{ji} - \frac{q_{ji}}{x_i^2} \right) = 0 \quad (3.20)$$

ergibt sich für  $x_i$ :

$$\begin{aligned} x_i &= \sqrt{\frac{q_{0i} + \sum \lambda_j q_{ji}}{p_{0i} + \sum \lambda_j p_{ji}}} && \text{für } \bar{L}'_i(x_{Li}) < 0 \text{ und } \bar{L}'_i(x_{Ui}) > 0 \\ x_i &= x_{Li} && \text{für } \bar{L}'_i(x_{Li}) \geq 0 \\ x_i &= x_{Ui} && \text{für } \bar{L}'_i(x_{Ui}) \leq 0 \end{aligned} \quad (3.21)$$

Damit sind die primären Variablen als Funktion der Lagrange-Multiplikatoren bekannt, und es kann das Maximum der dualen Funktion  $W(\lambda)$ , das dem Minimum des primären Problems entspricht, mit einem Verfahren bestimmt werden, wie es in Abschnitt 2.8.5.2 beschrieben wird.

Zur Lösung können Gradienten- und Newton-Verfahren eingesetzt werden, wobei beachtet werden sollte, daß die Hesse-Matrix der dualen Funktion an denjenigen Stellen diskontinuierlich ist, an denen die primären Variablen die oberen bzw. unteren Restriktionen erreichen. Gleichheitsbedingungen können mit dem Verfahren nicht direkt erfaßt werden, da im Falle negativer Lagrange-Multiplikatoren die Konvexität des Ersatzproblems nicht mehr garantiert werden kann. Fleury schlägt deshalb vor, Gleichheitsbedingungen durch zwei Ungleichheitsbedingungen zu ersetzen [45]. Dieses Vorgehen erweist sich jedoch als kaum umsetzbar, da aufgrund der angestrebten konservativen Approximationen die Unterprobleme häufig inkonsistent werden und eine formale Unterscheidung von tatsächlich inkonsistenten Problemen schwer fällt. Zur Behandlung von Unterproblemen, die infolge der konservativen Approximation inkonsistent werden, wird in [44] eine Relaxationstechnik vorgeschlagen.

### 3.2.3 Gemeinsamkeiten mit den Verfahren der Optimalitätskriterien

Die kombinierte Anwendung der hybriden Approximationsvorschrift (3.10) auf Zielfunktion und alle Nebenbedingungen sowie dualer Verfahren der mathematischen Programmierung führte zu einem separierbaren Ersatzproblem, das in  $n$  eindimensionale Minimierungsprobleme zerfällt. Aufgrund der besonderen Struktur der Approximationen können die Lösungen der eindimensionalen Probleme explizit angegeben werden und resultieren in Rekursionsformeln (3.21) für die primären Variablen  $x$  als Funktionen der Lagrange-Multiplikatoren. Die Lösungseigenschaften des Gesamtproblems werden von der Konvergenz der dualen Variablen (Lagrange-Multiplikatoren) geprägt und sind unabhängig von der Zahl der primären Variablen  $x$ . Gerade dies aber war lange Zeit

der Vorteil der Optimalitätskriterienmethoden gegenüber den Verfahren der mathematischen Programmierung. Auf diese Gemeinsamkeit der verschiedenen Lösungsansätze, die sich aus der dualen Betrachtung des Problems ergibt, wies Templeman [136] hin, Fleury [42] und Svanberg [134] ergänzten den Aspekt der Separabilität. Es ist bemerkenswert, daß sich für den Spezialfall der Gewichtsminimierung mit einer Verschiebungsnebenbedingung aus (3.21) genau dieselbe Rekursionsformel ergibt, wie sie von Berke [10] bereits früher für die Optimalitätskriterienverfahren entwickelt worden ist. Dabei gehen die Verfahren grundsätzlich unterschiedlich vor (Tafel 3.1). Der wesentliche Unterschied ist, daß die Verfahren der mathematischen Programmierung das Problem direkt angehen, indem sie die lokalen Eigenschaften des Problems auswerten, während bei den Optimalitätskriterienmethoden die Bedingungen für Optimalität a priori bekannt sind, woraus Rekursionsformeln für die primären Variablen gewonnen werden. Dieses Verfahren wird auch indirekt genannt [147] und führt zu effizienten Lösungsalgorithmen, die allerdings nur für die Fälle der betrachteten Optimalitätsbedingungen anwendbar sind. Besondere Schwierigkeiten ergeben sich bei mehreren Nebenbedingungen, wenn die am Optimum aktiven unbekannt sind. Dazu wurden verschiedene Methoden entwickelt, die die Lagrange-Multiplikatoren der aktiven Nebenbedingungen abschätzen sollen [74]. Die Eleganz und theoretische Fundiertheit des in Abschnitt 3.2.2 geschilderten Vorgehens konnte damit allerdings nicht erreicht werden, auch wenn so Algorithmen entstanden, die sich in der praktischen Anwendung bewährten. Gerade bei mehreren aktiven Nebenbedingungen sind die Unterschiede zwischen Verfahren der Optimalitätskriterien und der dualen mathematischen Programmierung inzwischen fast verschwunden [10].

Mathematische Programmierung (MP)	Methoden der Optimalitätskriterien (OKM)
<ul style="list-style-type: none"> <li>● Konservative (hybride) Approximation</li> <li>● Betrachtung der lokalen Eigenschaften des Problems</li> <li>● Suchrichtung, Schrittweite (Line Search)</li> <li>● Abbruch: wenn lokale Kuhn-Tucker-Bedingungen erfüllt sind</li> </ul>	<ul style="list-style-type: none"> <li>● Annahme unveränderten Tragwerksverhaltens bei Variablenänderungen</li> <li>● Aufstellen der Bedingungen für das zu erreichende Optimum (intuitiv oder formal, K.T.-Bed.)</li> <li>● Rekursionsformeln für primäre und duale Variablen</li> <li>● Abbruch: wenn Rekursion konvergiert ist</li> </ul>

**Tafel 3.1:** Gegenüberstellung der Verfahren der mathematischen Programmierung (MP) und der Optimalitätskriterienmethoden (OKM)

### 3.2.4 Das Verfahren der beweglichen Asymptoten

#### 3.2.4.1 Approximationsvorschrift

Svanberg [135] gibt ein Verfahren an, das er die Methode der beweglichen Asymptoten nennt, kurz MMA (Method of Moving Asymptotes). Es stellt eine Erweiterung der konvexen Linearisierung dar, wobei die Eigenschaften der Konvexität und Separabilität erhalten bleiben. Anstelle von (3.13) bis (3.15) lautet das approximierte Unterproblem nun:

$$\tilde{f}(\mathbf{x}) = r_0 + \sum_{i=1}^n \left( \frac{p_{0i}}{u_i - x_i} + \frac{q_{0i}}{x_i - l_i} \right) \rightarrow \min \quad (3.22)$$

mit:

$$\tilde{g}_j(\mathbf{x}) = r_j + \sum_{i=1}^n \left( \frac{p_{ji}}{u_i - x_i} + \frac{q_{ji}}{x_i - l_i} \right) \leq 0 \quad ; j = 1, \dots, m$$

$$l_i < x_{Li} \leq a_i \leq x_i \leq \beta_i \leq x_{Ui} < u_i \quad ; i = 1, \dots, n$$

Die Parameter sind folgendermaßen definiert:

$$p_{0i} = \begin{cases} (u_i - x_{i0})^2 \frac{\partial f(\mathbf{x}_0)}{\partial x_i} & ; \text{für } \frac{\partial f(\mathbf{x}_0)}{\partial x_i} > 0 \\ 0 & ; \text{für } \frac{\partial f(\mathbf{x}_0)}{\partial x_i} \leq 0 \end{cases} \quad (3.23)$$

$$q_{0i} = \begin{cases} 0 & ; \text{für } \frac{\partial f(\mathbf{x}_0)}{\partial x_i} \geq 0 \\ -(x_{i0} - l_i)^2 \frac{\partial f(\mathbf{x}_0)}{\partial x_i} & ; \text{für } \frac{\partial f(\mathbf{x}_0)}{\partial x_i} < 0 \end{cases}$$

$$r_0 = f(\mathbf{x}_0) - \sum_{i=1}^n \left( \frac{p_{0i}}{u_i - x_{i0}} + \frac{q_{0i}}{x_{i0} - l_i} \right)$$

und:

$$p_{ji} = \begin{cases} (u_i - x_i)^2 \frac{\partial g_j(\mathbf{x}_0)}{\partial x_i} & ; \text{für } \frac{\partial g_j(\mathbf{x}_0)}{\partial x_i} > 0 \\ 0 & ; \text{für } \frac{\partial g_j(\mathbf{x}_0)}{\partial x_i} \leq 0 \end{cases} \quad (3.24)$$

$$q_{ji} = \begin{cases} 0 & ; \text{für } \frac{\partial g_j(\mathbf{x}_0)}{\partial x_i} \geq 0 \\ -(x_i - l_i)^2 \frac{\partial g_j(\mathbf{x}_0)}{\partial x_i} & ; \text{für } \frac{\partial g_j(\mathbf{x}_0)}{\partial x_i} < 0 \end{cases}$$

$$r_j = g_j(\mathbf{x}_0) - \sum_{i=1}^n \left( \frac{p_{ji}}{u_i - x_{i0}} + \frac{q_{ji}}{x_{i0} - l_i} \right)$$

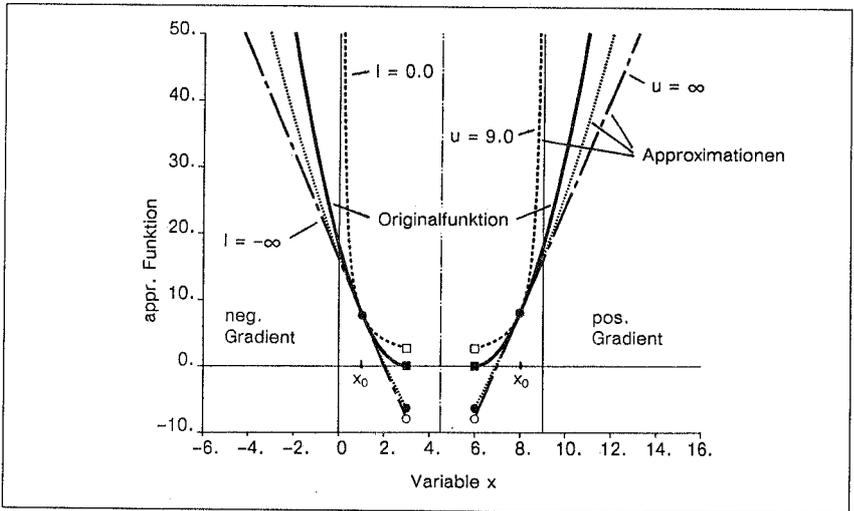
- $x_0$  .... Entwicklungspunkt
- $x_{j0}$  ... Komponenten des Entwicklungspunktes
- $l_j, u_j$  ... untere bzw. obere Asymptoten
- $a_i, \beta_i$  ... untere bzw. obere Schrittweitenbegrenzungen

Die dargestellten Approximationen ergeben sich aus der Linearisierung von Zielfunktion und Nebenbedingungen mit den verallgemeinerten Variablen:

$$y_i = \frac{1}{u_j - x_i} \quad \text{für positive partielle Ableitungen} \tag{3.25}$$

und:  $y_i = \frac{1}{x_j - l_j}$  für negative partielle Ableitungen

Mit der Einführung der Asymptoten  $l_j$  und  $u_j$  wird das Verfahren automatisch auch für negative Variablen  $x_j$  anwendbar. Im Gegensatz dazu ist beim weiter oben geschilderten Verfahren von Fleury [44] eine Variablentransformation notwendig, um bei den reziproken Variablen eine Nulldivision zu vermeiden. Weiter werden die Asymptoten während des Rechenvorganges laufend dem Problem angepaßt, so daß das Verfahren die Konservativität der Approximation in jedem Iterationsschritt selbst bestimmt, was in der Regel stabilisierend wirkt. Mit der Wahl der Asymptoten können die Krümmungen der approximierten Funktionen gesteuert werden (Bild 3.3). Der Grenzfall  $l_j = -\infty$  und  $u_j = \infty$  führt



**Bild 3.3:** *Einfluß der Asymptoten auf die Krümmungen der approximierten Funktionen (1-D Beispiel)*

zum Verfahren der sequentiellen linearen Programmierung (Abschnitt 2.6.8.4). Werden hingegen  $l_i = 0$  und  $u_i = \infty$  gesetzt, ergibt sich die hybride Approximation (3.10) und daraus das Verfahren von Fleury. Neben der prinzipiellen Bedeutung, die Schrittweitenbeschränkungen in Verbindung mit Approximationstechniken haben, müssen sie hier auch verhindern, daß die Nenner in (3.22) bis (3.24) null werden können. Svanberg gibt folgende Bestimmungsregel an, die sich in der Anwendung bewährt hat:

$$\begin{aligned} \alpha_i &= \max(x_{Li}; 0,9 l_i + 0,1 x_i) \\ \beta_i &= \min(x_{Uj}; 0,9 u_i + 0,1 x_i) \end{aligned} \quad (3.26)$$

Auf die Wahl der Asymptoten wird noch später eingegangen.

### 3.2.4.2 Lösung des approximierten Unterproblems

Das Problem (3.22) ist wie (3.13) konvex und separierbar. Deshalb bietet sich auch hier ein dualer Algorithmus zur Lösung an. Analog zu (3.16) lautet die Lagrangefunktion:

$$\begin{aligned} \tilde{L}(x, \lambda) &= r_0 + \sum_{i=1}^n \left( \frac{P_{0i}}{(u_i - x_i)} + \frac{Q_{li}}{(x_i - l_i)} \right) \\ &\quad + \sum_{j=1}^m \left\{ \lambda_j r_j + \lambda_j \sum_{i=1}^n \left( \frac{P_{ji}}{(u_i - x_i)} + \frac{Q_{ji}}{(x_i - l_i)} \right) \right\} \end{aligned} \quad (3.27)$$

Das zu (3.22) duale Problem lautet:

$$\max_{\lambda} W(\lambda) \quad \text{mit} \quad \lambda \geq 0 \quad (3.28)$$

wobei die duale Funktion  $W(\lambda)$  definiert ist als:

$$W(\lambda) = \min_x \tilde{L}(x, \lambda) \quad (3.29)$$

Aufgrund der Separabilität von (3.22) vereinfacht sich Gleichung (3.29) zu  $n$  eindimensionalen Minimierungsaufgaben:

$$W(\lambda) = \sum_{i=1}^n \left( \min_{x_i} \tilde{L}_i(x_i, \lambda) \right) + r_0 + \sum_{j=1}^m \lambda_j r_j \quad (3.30)$$

Dabei ist:

$$\bar{L}_i(x_i, \lambda) = \frac{p_{0i} + \sum_j \lambda_j p_{ji}}{u_i - x_i} + \frac{q_{0i} + \sum_j \lambda_j q_{ji}}{x_i - l_i} \quad (3.31)$$

Aus den Stationärbedingungen  $d\bar{L}_i/dx_i = \bar{L}'_i = 0$  können die  $x_i$  explizit als Funktionen der Lagrange-Multiplikatoren bestimmt werden:

$$x_i(\lambda) = \frac{(p_{0i} + \sum_j \lambda_j p_{ji})^{\frac{1}{2}} l_i + (q_{0i} + \sum_j \lambda_j q_{ji})^{\frac{1}{2}} u_i}{(p_{0i} + \sum_j \lambda_j p_{ji})^{\frac{1}{2}} + (q_{0i} + \sum_j \lambda_j q_{ji})^{\frac{1}{2}}} \quad (3.32)$$

$$\text{für: } \bar{L}'_i(\alpha_i, \lambda) < 0 \quad \text{und} \quad \bar{L}'_i(\beta_i, \lambda) > 0$$

$$x_i(\lambda) = \alpha_i \quad \text{für} \quad \bar{L}'_i(\alpha_i, \lambda) \geq 0$$

$$x_i(\lambda) = \beta_i \quad \text{für} \quad \bar{L}'_i(\beta_i, \lambda) \leq 0$$

Da die zweiten Ableitungen (Hauptdiagonalterme der Hessematrix der approximierten Lagrangefunktion):

$$\frac{d^2 \bar{L}_i}{dx_i^2} = \bar{L}''_i = 2 \left\{ \frac{p_{0i} + \sum_j \lambda_j p_{ji}}{(u_i - x_i)^3} + \frac{q_{0i} + \sum_j \lambda_j q_{ji}}{(x_i - l_i)^3} \right\} \quad (3.33)$$

stets positiv sind, gibt (3.32) tatsächlich eine Minimalstelle an. Damit sind die duale Funktion  $W(\lambda)$  und ihre Ableitungen definiert:

$$W(\lambda) = r_0 + \sum_{j=1}^m \lambda_j r_j + \sum_{i=1}^n \left\{ \frac{p_{0i} + \sum_j \lambda_j p_{ji}}{(u_i - x_i(\lambda))} + \frac{q_{0i} + \sum_j \lambda_j q_{ji}}{(x_i(\lambda) - l_i)} \right\} \quad (3.34)$$

$$\frac{\partial W}{\partial \lambda_j} = r_j + \sum_{i=1}^n \left\{ \frac{p_{ji}}{(u_i - x_i(\lambda))} + \frac{q_{ji}}{(x_i(\lambda) - l_i)} \right\} = \bar{g}_j(x(\lambda)) \quad (3.35)$$

Zur Lösung der dualen Aufgabe (3.28) schlägt Svanberg einen modifizierten Gradientenalgorithmus vor, der die Restriktionen für die Lagrange-Multiplikatoren berücksichtigt. Dagegen verwendet Fleury [44], [42] ein Newton-Verfahren, das jedoch mit Vorsicht angewendet werden muß. Wie man bei einer weiteren Ableitung von (3.35)

$$\frac{\partial^2 W}{\partial \lambda_j \partial \lambda_k} = \sum_{i=1}^n \left\{ \frac{p_{ji}}{(u_i - x_i(\lambda))^2} - \frac{q_{ji}}{(x_i(\lambda) - l_i)^2} \right\} \frac{\partial x_i(\lambda)}{\partial \lambda_k} \quad (3.36)$$

erkennt, ergeben sich an denjenigen Stellen Unstetigkeiten in der Hesse-Matrix der dua-

len Funktion, an denen die primären Variablen auf die Schrittweitenbegrenzungen stoßen. Dort ist nämlich:

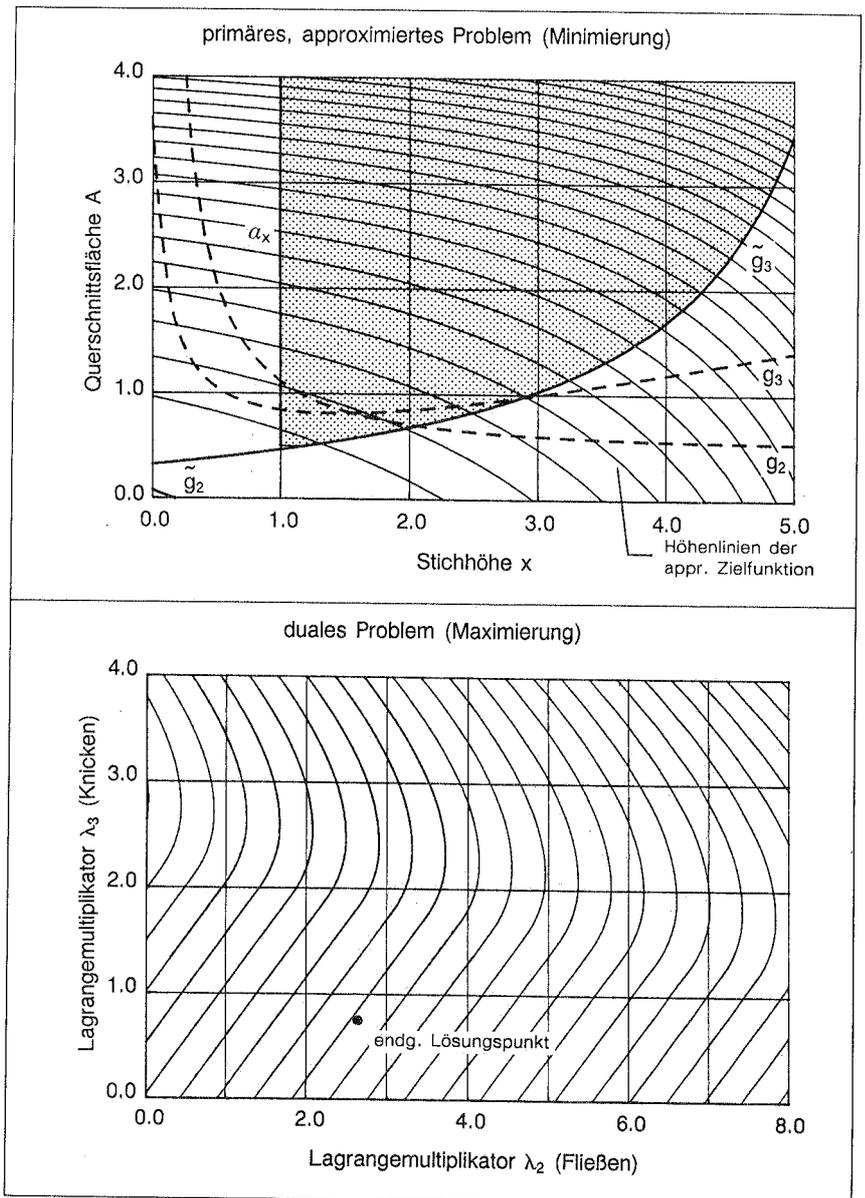
$$\frac{\partial x_i(\lambda)}{\partial \lambda_k} = 0$$

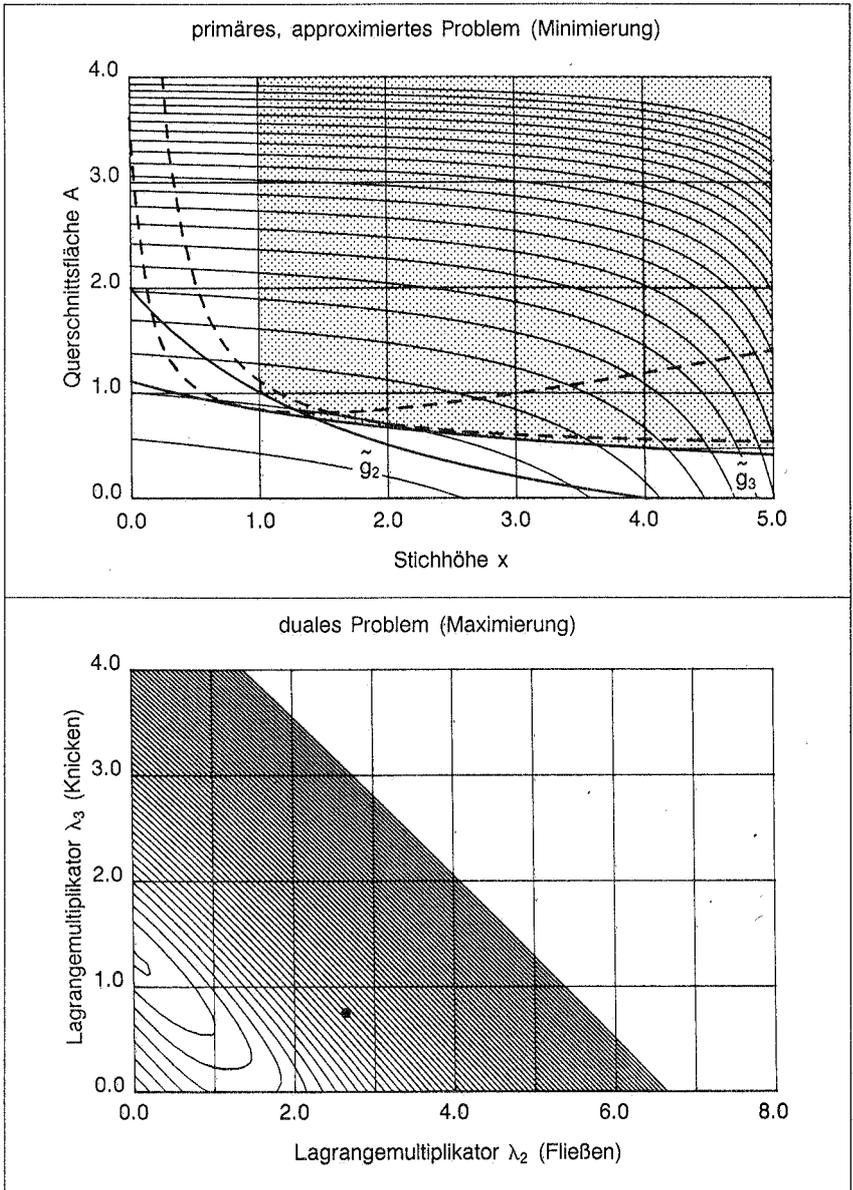
Im Rahmen dieser Arbeit werden der Vorschlag Svanbergs aufgegriffen und der in Abschnitt 2.8.3 geschilderte Algorithmus, der eine Modifikation des Verfahrens von Fletcher und Reeves zur Berücksichtigung einfacher Variablenrestriktionen darstellt, mit Erfolg eingesetzt. In der Regel ist die duale Funktion relativ gutmütig und, wenn die Nebenbedingungen normiert werden, auch gut konditioniert, weshalb der verwendete Gradientenalgorithmus schnell konvergiert. Die Bilder 3.4a–d zeigen eine Folge von Iterationsschritten, wie sie bei der Optimierung des Zweibocks von Abschnitt 2.2 mit der Methode der beweglichen Asymptoten auftraten. Berücksichtigt wurden die Fließ- und die Knickbedingungen (Fall b).

Im jeweils oberen Teil der Bilder sind die approximierten Funktionen (Zielfunktion und Nebenbedingungen) sowie gestrichelt die ursprünglichen Nebenbedingungen dargestellt (vergleiche mit Bild 2.2b). Besonders an den Höhenlinien der approximierten Zielfunktion ist der Einfluß der Asymptoten sehr gut zu erkennen. Bei diesem Beispiel der Formoptimierung ist nicht zu erwarten, daß die Approximationen der Nebenbedingungen exakt sein könnten. Selbst bei Querschnittsoptimierungen kann die Knickbedingung mit den gewählten Vorschriften (3.25) nicht exakt approximiert werden. Dennoch konvergiert der Algorithmus bei den gewählten Startwerten innerhalb weniger Iterationen (Tafel 3.2).

k	x	A	$\lambda_1$	$\lambda_2$	f(x)	zulässig
0	4,0	3,0	0,0	0,0	26,83	ja
1	1,0	0,75	0,0	2,83	3,36	nein
2	1,40	0,77	0,01	1,27	3,78	nein
3	1,54	0,81	1,85	0,87	4,09	nein
4	1,55	0,82	2,56	0,76	4,13	ja
5	1,55	0,82	2,64	0,75	4,13	ja

**Tafel 3.2:** Iterationsverlauf, 2-stabiges Fachwerk mit MMA





**Bild 3.4b:** Optimierung des zweiständigen Fachwerks mit MMA, Iterationsschritt 2

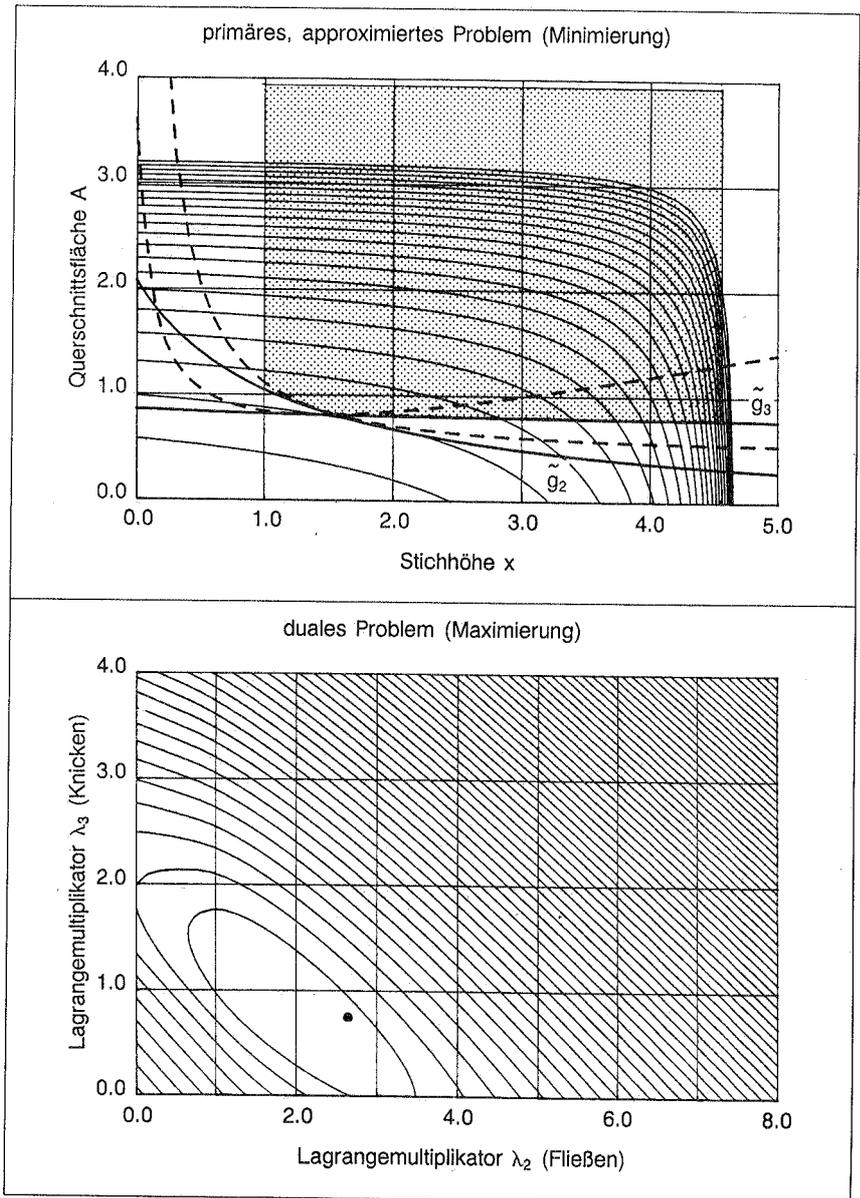
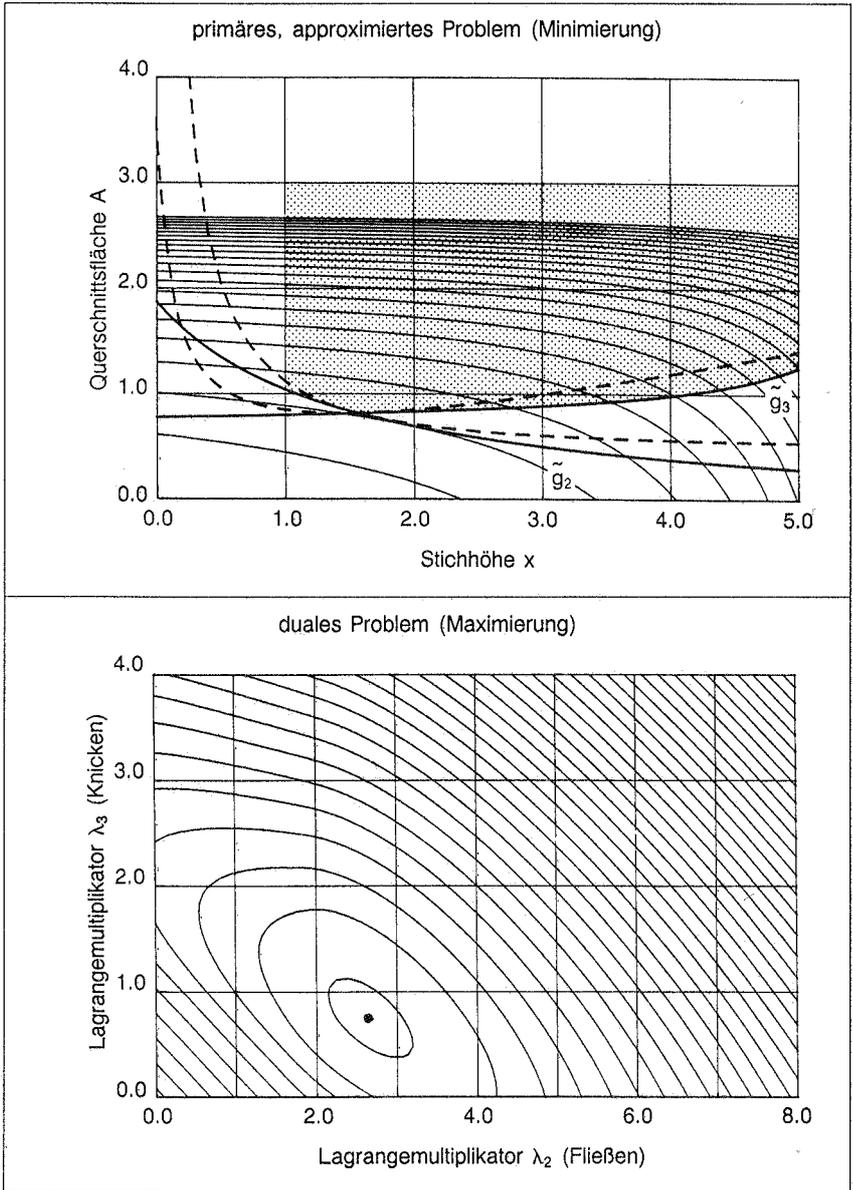
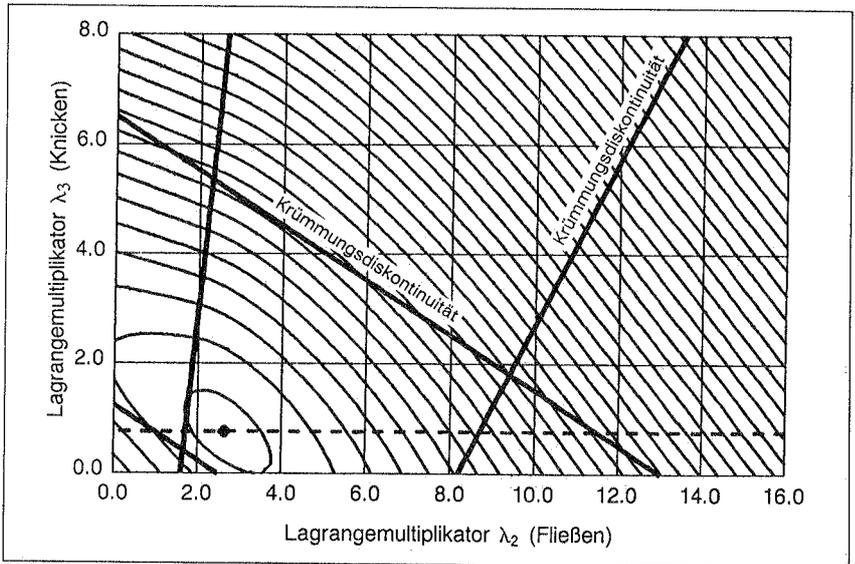


Bild 3.4c: Optimierung des zweistöbigen Fachwerks mit MMA, Iterationsschritt 3



**Bild 3.4d:** Optimierung des zweistäbigen Fachwerks mit MMA, Iterationsschritt 6  
Konvergenz

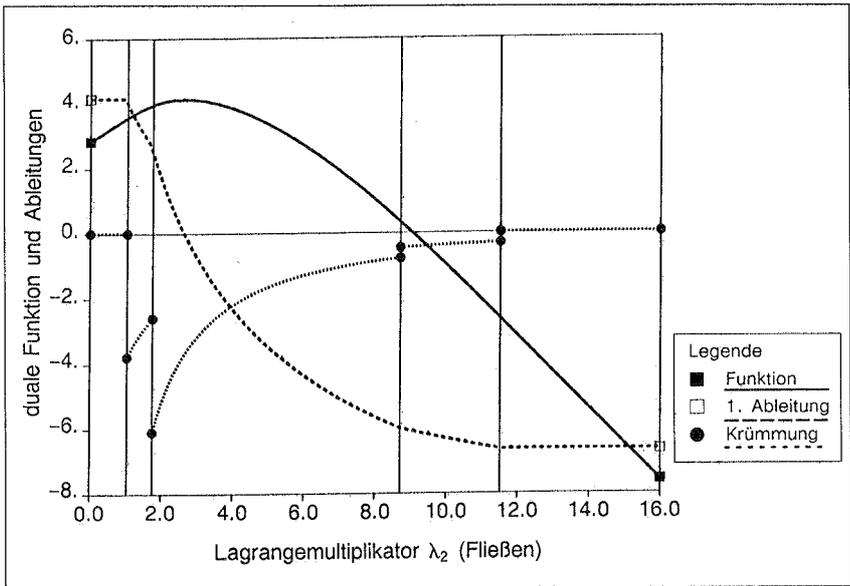
Im unteren Teil der Bilder 3.4a–d sind die Höhenlinien der dualen Funktionen dargestellt, wie sie sich für die jeweils getroffenen Approximationen ergeben. Der eingetragene Punkt markiert die optimale Lösung. Schon im zweiten Iterationsschritt zeigt die duale Funktion deutlich quadratische Eigenschaften, weshalb der verwendete Gradientenalgorithmus keinerlei Konvergenzschwierigkeiten hat. Im ersten Iterationsschritt werden die angesprochenen Diskontinuitäten der zweiten Ableitungen an den geraden Abschnitten der Höhenlinien deutlich. Für eine von den Bildern 3.4 geringfügig abweichende Approximation mit anderen Asymptoten und Schrittweitenbegrenzungen sind in Bild 3.5 in das Höhenlinienbild der dualen Funktion die geometrischen Orte eingezeichnet, an denen die zweiten Ableitungen der dualen Funktion diskontinuierlich sind.



**Bild 3.5:** Diskontinuitätsstellen der dualen Funktion (Iterationsschritt 6)

Ein Schnitt durch die duale Funktion (gestrichelt in Bild 3.5) sowie die ersten und zweiten Ableitungen sind in Bild 3.6 dargestellt und verdeutlichen noch einmal die besonderen Eigenschaften der dualen Funktion. In diesen Fällen werden an das verwendete Line Search-Verfahren, das ein wichtiger Grundbestandteil des Algorithmus ist, besondere Ansprüche gestellt. Einfache Kurveninterpolationen genügen hier nicht, da die zweiten Ableitungen diskontinuierlich sind (siehe Abschnitt 2.8.1). Deshalb wurde ein Verfahren eingesetzt, das Kurveninterpolationen mit Intervallschachtelungen (z.B. goldener Schnitt bzw. Intervallhalbierung) kombiniert. Gill et al. [52] nennen das "safe guarded line search". Fleury [42] schlägt vor, die Diskontinuitätsstellen vorab zu bestimmen, um

innerhalb der so bestimmten Intervalle die Newton–Raphson–Methode zur Maximumbestimmung zu verwenden.



**Bild 3.6:** Schnitt durch die duale Funktion,  $\lambda_3 = \text{const.} = 0,746$

### 3.2.4.3 Adaption der Asymptoten

Die Wahl der Asymptoten hat entscheidenden Einfluß auf das Konvergenzverhalten des Verfahrens. Dafür gilt folgende Faustregel:

- oszilliert der Iterationsprozess, wird er durch ein Zusammenrücken der Asymptoten stabilisiert;
- ist der Iterationsprozess monoton und langsam, wird er durch ein Auseinanderücken der Asymptoten beschleunigt.

Gemäß dieser Regel macht Svanberg folgenden Vorschlag für eine heuristische Adaptationsvorschrift:

- während der ersten beiden Iterationsschritte,  $k=0$ ,  $k=1$ :

$$\begin{aligned} l_{ik} &= x_{ik} - s_a (x_{U_i} - x_{L_i}) \\ u_{ik} &= x_{ik} + s_a (x_{U_i} - x_{L_i}) \end{aligned} \quad (3.37)$$

- für alle weiteren Iterationsschritte ( $k \geq 2$ ):

ist  $(x_{ik} - x_{i(k-1)}) (x_{i(k-1)} - x_{i(k-2)}) < 0$  (oszillieren), dann

$$l_{ik} = x_{ik} - s_b (x_{i(k-1)} - l_{i(k-1)})$$

$$u_{ik} = x_{ik} + s_b (u_{i(k-1)} - x_{i(k-1)})$$

(3.38)

ist  $(x_{ik} - x_{i(k-1)}) (x_{i(k-1)} - x_{i(k-2)}) > 0$  (monotoner Fortschritt), dann

$$l_{ik} = x_{ik} - \frac{1}{s_b} (x_{i(k-1)} - l_{i(k-1)})$$

$$u_{ik} = x_{ik} + \frac{1}{s_b} (u_{i(k-1)} - x_{i(k-1)})$$

(3.39)

Svanberg schlägt für die Faktoren  $s_a=1$  bzw.  $s_b=0.7$  vor. Dabei entscheidet vor allem der Faktor  $s_a$  über das Verhalten des Verfahrens. Die richtige Wahl hängt von den Abständen der entsprechenden unteren und oberen Restriktionen ab (Gleichung 3.37) und ist deshalb problemabhängig. Als Richtlinie mag gelten, daß der Faktor  $s_a$  eher kleiner als zu groß gewählt werden sollte, um in den ersten Iterationsschritten zu große Variablenänderungen zu vermeiden. Das in den Bildern 3.4 dargestellte Beispiel wurde mit  $s_a = 0.5$  und  $s_b = 0.7$  berechnet. Die Abstände der Restriktionen betragen für beide Variablen 10. Tafel 3.3 gibt einen Eindruck vom Einfluß des Parameters  $s_a$  auf die Iterationszahlen. In [135] gibt Svanberg eine weitere Adaptationsregel an, die besonders für Querschnittsoptimierungen in Frage kommen kann.

Eine Adaptationsregel, die sich an den Krümmungen der Zielfunktion und den Nebenbedingungen orientiert, geben Fleury und Smaoui in [46] an. Damit gehen zwar weitere Informationen über das Problem mit in die Formulierung des Unterproblems ein, jedoch läuft dieser Vorschlag auf die Bestimmung von eigenen Asymptoten für jede Funktion (Zielfunktion und Nebenbedingungen) hinaus. Der ursprüngliche Vorteil des Verfahrens, daß die eindimensionale Minimierung von (3.31) explizit möglich ist, ist somit verloren.

k	$s_a=0,1$	$s_a=0,2$	$s_a=0,3$	$s_a=0,4$	$s_a=0,5$	$s_a=0,6$	$s_a=0,7$	$s_a=0,8$	$s_a=0,9$	$s_a=1,0$
0	26,83	26,83	26,83	26,83	26,83	26,83	26,83	26,83	26,83	26,83
1	16,77	10,60	6,48	3,97	2,63	1,61	0,72	0,40	0,40	0,40
2	9,61	3,51	2,76	0,53	0,41	0,40	0,40	0,40	0,40	0,40
3	4,17	1,09	0,41	1,07	0,42	0,42	0,41	0,41	0,41	0,41
4	3,91	2,13	0,42	2,10	0,50	0,48	0,46	0,44	0,44	0,44
5	4,12	3,58	0,50	3,25	1,02	0,90	0,68	0,56	0,56	0,56
6	4,13	4,08	1,06	3,95	1,73	1,21	0,99	1,22	1,22	1,22
7	4,13	4,13	1,99	4,12	3,18	0,73	0,64	2,31	2,05	1,82
8		4,13	3,36	4,13	3,81	1,20	1,22	3,50	3,75	0,82
9		4,13	4,01	4,13	4,06	2,43	2,67	3,96	3,48	0,96
10			4,12		4,13	3,76	4,14	4,12	1,77	0,46
11			4,13		4,13	4,11	4,12	4,13	2,91	0,78
12			4,13		4,13	4,13	4,13	4,13	3,99	1,72
13						4,13	4,13		4,09	4,16
14						4,13	4,13		4,13	3,67
15									4,13	0,41
16										0,43
17										0,55
18										1,34
19										4,95
20										17,55
										☆

**Tafel 3.3:** Iterationsverläufe infolge verschiedener  $s_a$ -Parameter (2-stäbiges Fachwerk) für Zielfunktion  $f$  (Gewicht)

Ein eigener Vorschlag, der weiterhin die explizite Minimierung von (3.31) erlaubt, aber dennoch Krümmungsinformationen mit für die Bestimmung der Asymptoten heranzieht, wird im folgenden geschildert. Die Krümmung der Lagrangefunktion des approximierten Problems (3.22) soll am Entwicklungspunkt  $x_0, \lambda_0$  weitestgehend mit der Krümmung der Lagrangefunktion des Originalproblems übereinstimmen. Es werden dabei die Hauptdiagonalglieder der Hessematrix verwendet, und aus (3.33) ergibt sich:

$$\frac{\partial^2 \tilde{L}(x_0, \lambda_0)}{\partial x_i^2} = 2 \left\{ \frac{p_{0i} + \sum_j \lambda_{0j} p_{ji}}{(u_i - x_{i0})^3} + \frac{q_{0i} + \sum_j \lambda_{0j} q_{ji}}{(x_{i0} - l_i)^3} \right\} \quad (3.40)$$

Gleichung (3.23) und (3.24) werden in (3.40) eingesetzt:

$$\delta_i = 2 \frac{\left| \frac{\partial f(x_0)}{\partial x_i} \right| + \sum_j \lambda_{0j} \left| \frac{\partial g_j(x_0)}{\partial x_i} \right|}{\frac{\partial^2 L(x_0, \lambda_0)}{\partial x_i^2}} \quad (3.41)$$

wobei  $\delta_i = (u_i - x_{i0}) = (x_{i0} - l_i)$  gesetzt wurde.

Damit bestimmen sich die Asymptoten zu:

$$\begin{aligned} l_i &= x_{i0} - \delta_i \\ u_i &= x_{i0} + \delta_i \end{aligned} \quad (3.42)$$

Dieser Vorschlag lehnt sich an das Vorgehen der Lagrangemethoden an (Abschnitt 2.8.6), bei der Formulierung der Unterprobleme die Hessematrix der Lagrangefunktion sowie die Gradienten von Zielfunktion und Nebenbedingungen zu verwenden. Er hat sich jedoch nicht bewährt, was vor allem auf die unvollständige und sehr ungenaue Bestimmung der zweiten Ableitungen mit der Differenzenregel der Quasi-Newton-Korrektur (hier BFGS) sowie die notwendige Beschränkung auf die Hauptdiagonalglieder der Hessematrix zurückgeführt wird. Letzteres bedingt, daß nur Metrikänderungen in Richtung der Achsen des Entwurfsraumes erfaßt werden können. Ebenfalls ungünstig auf die Krümmungen der Lagrangefunktion wirkt sich das Konvergenzverhalten der Lagrangemultiplikatoren aus, das vor allem zu Beginn der Iteration und bei inkonsistenten Unterproblemen sehr unregelmäßig ist.

Insgesamt hat sich die Adaptationsregel (3.37) bis (3.39) als sehr robust und einfach zu handhaben erwiesen und hat sich bei vielen Beispielen der Form- und Querschnittsoptimierung bewährt [47], [114].

#### 3.2.4.4 Inkonsistente Nebenbedingungen

Eine Folge der konservativen Approximation kann sein, daß das Ersatzproblem (3.22) inkonsistent ist, obwohl in Wirklichkeit eine Lösung existiert. In solchen Fällen muß der Entwurfsraum des approximierten Ersatzproblems künstlich aufgeweitet werden. Damit ist die Hoffnung verbunden, daß so eine Zwischenlösung gefunden wird, die als Stützstelle für ein nächstes, dann konsistentes Ersatzproblem dient. Der Algorithmus bricht ab, wenn zwei direkt aufeinanderfolgende Ersatzprobleme inkonsistent sind, da vermutet werden kann, daß das Originalproblem selbst keine Lösung hat.

In (3.22) wird für jede Nebenbedingung je eine Relaxationsvariable  $z_j$  eingeführt:

$$\tilde{f}(\mathbf{x}, \mathbf{z}) = r_0 + \sum_{i=1}^n \left( \frac{p_{0i}}{(u_i - x_i)} + \frac{q_{0i}}{(x_i - l_i)} \right) + \sum_{j=1}^m (d_j z_j + d_j z_j^2) \rightarrow \min$$

mit (3.43)

$$\tilde{g}(\mathbf{x}, \mathbf{z}) = r_j + \sum_{i=1}^n \left( \frac{p_{ji}}{(u_i - x_i)} + \frac{q_{ji}}{(x_i - l_i)} \right) - z_j \leq 0$$

$$a_i \leq x_i \leq \beta_i \quad ; i = 1, \dots, n$$

$$0 \leq z_j \quad ; i = 1, \dots, m$$

$d_j$  ... Penaltyparameter ( $d_j > 0$ )

$z_j$  ... Relaxationsvariablen

Sind die Relaxationsvariablen null, entspricht (3.43) dem ursprünglichen Problem (3.22). Für genügend große Penaltyparameter ergibt sich dies von selbst, sofern eine Lösung möglich ist. Eine Abschätzung für eine geeignete Wahl folgt aus der Betrachtung der dualen Funktion. Die Erweiterung von (3.30) ergibt, da (3.43) auch bezüglich der Relaxationsvariablen separierbar ist:

$$W(\lambda) = \sum_{i=1}^n \left[ \min_{x_i} \bar{L}_i(x_i, \lambda) \right] + \sum_{j=1}^m \left[ \min_{z_j} \bar{L}_j(z_j, \lambda) \right] \quad (3.44)$$

mit

$$\bar{L}_j(z_j, \lambda) = d_j z_j + d_j z_j^2 - \lambda_j z_j \quad (3.45)$$

Die  $z_j$  können explizit bestimmt werden:

$$\text{aus} \quad \frac{\partial \bar{L}_j}{\partial z_j} = d_j + 2d_j z_j - \lambda_j = 0$$

$$\text{folgt} \quad z_j = \frac{\lambda_j - d_j}{2d_j} \quad (3.46)$$

Zulässige Werte für  $z_j$  ergeben sich, wenn  $\lambda_j \geq d_j$  ist. Damit möglichst selten und nur für inkonsistente Probleme Gebrauch von (3.46) gemacht wird, werden bei der Bestimmung der approximierten Ersatzprobleme die Penaltyparameter folgendermaßen gesetzt:

$$\begin{aligned}
 d_j &= \varrho \lambda_j && \text{für } \lambda_j > 0 \\
 d_j &= \varrho \frac{|\nabla f|}{|\nabla g_j|} && \text{für } \lambda_j = 0
 \end{aligned}
 \tag{3.47}$$

mit  $\varrho = 10$  oder  $\varrho = 100$

Der Faktor  $\varrho$  darf nicht zu klein gewählt werden, allerdings auch nicht übermäßig groß, da bei tatsächlich inkonsistenten Problemen die Lagrangemultiplikatoren solange ansteigen werden, bis Gleichung (3.46) eine Relaxation erlaubt. Mit dieser Relaxationstechnik, die in [135] nur angedeutet, aber nicht im Detail beschrieben wird, wurden gute Erfahrungen gemacht. Für Fälle, bei denen die Approximationen so konservativ sind, daß zwei aufeinanderfolgende Ersatzprobleme inkonsistent werden und der Algorithmus abbricht, muß die Wahl der Asymptoten verbessert werden (z.B. durch einen größeren  $s_a$ -Faktor).

### 3.2.4.5 Modifikationen

#### 3.2.4.5.1 Unbeschränkte Probleme

Die Approximationsvorschriften (3.23) und (3.24) erfassen nur die ersten Ableitungen, wobei auch die Zielfunktion nur linear bezüglich der verallgemeinerten Variablen  $y_i$  approximiert wird und somit nach unten und oben unbeschränkt ist. Deshalb ergeben sich ernste Konvergenzschwierigkeiten, wenn das Optimum innerhalb des zulässigen Bereiches liegt oder das Problem gänzlich unbeschränkt ist. In diesen Fällen bestimmt die Adaptationsregel der Asymptoten die Konvergenzrate. Da die geschilderte Regel (3.27) bis (3.39) sehr flexibel ausgelegt ist, ist sie dann meist nicht in der Lage, die Asymptoten und damit die Schrittweitenbegrenzungen an das Optimum heranzuführen. Der Algorithmus beginnt zu oszillieren (vergleiche auch mit Tafel 3.3 für  $s_a = 1,0$ ). Es wird deshalb folgende Modifikation vorgeschlagen: die Approximation der Zielfunktion soll für jede Variable gleichzeitig sowohl die unteren als auch die oberen Asymptoten erfassen. Damit hat die approximierte Zielfunktion ein Minimum. Als die dafür notwendige zusätzliche Information wird die Krümmung der Zielfunktion am Entwicklungspunkt eingesetzt. Die Werte werden dabei aus der Differenz der Gradienten des aktuellen und letzten Iterationsschrittes gewonnen. Berücksichtigt werden nur die Hauptdiagonalelemente der Hessematrix, um die Eigenschaft der Separabilität nicht zu verlieren.

Die Approximation der Zielfunktion lautet nun:

$$\tilde{f}(x) = r_0 + \sum_{i=1}^n \left( \frac{p_{0i}}{(u_i - x_i)} + \frac{q_{0i}}{(x_i - l_i)} \right) \quad (3.48)$$

mit den Parametern:

$$p_{0i} = k_i (u_i - x_{i0})^2 \left[ \frac{f'_{i0}}{x_{i0} - l_i} + \frac{1}{2} f''_{i0} \right]$$

$$q_{0i} = k_i (x_{i0} - l_i)^2 \left[ -\frac{f'_{i0}}{u_i - x_{i0}} + \frac{1}{2} f''_{i0} \right]$$

$$r_0 = f(x_0) - \sum_{i=1}^n \left( \frac{p_{0i}}{(u_i - x_{i0})} + \frac{q_{0i}}{(x_{i0} - l_i)} \right)$$

$$k_i = \frac{(x_{i0} - l_i)(u_i - x_{i0})}{(x_{i0} - l_i) + (u_i - x_{i0})}$$

$$f'_{i0} = \frac{\partial f(x_0)}{\partial x_i} \quad ; \quad f''_{i0} = \frac{\partial^2 f(x_0)}{\partial x_i^2}$$

$x_0$  ... Entwicklungspunkt

$x_{i0}$  ... Komponenten des Entwicklungspunktes

$u_i, l_i, \dots$  obere bzw. untere Asymptoten für  
die  $i$ -te Optimierungsvariable

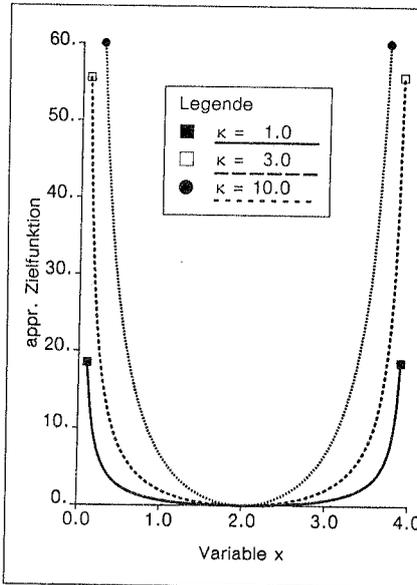
Damit (3.48) konvex ist, müssen zusätzlich folgende Bedingungen gelten:

$$\begin{aligned} p_{0i} > 0 & \rightarrow f''_{i0} \geq -\frac{2}{x_{i0} - l_i} f'_{i0} (1 + c_K) \\ q_{0i} > 0 & \rightarrow f''_{i0} \geq \frac{2}{u_i - x_{i0}} f'_{i0} (1 + c_K) \end{aligned} \quad (3.49)$$

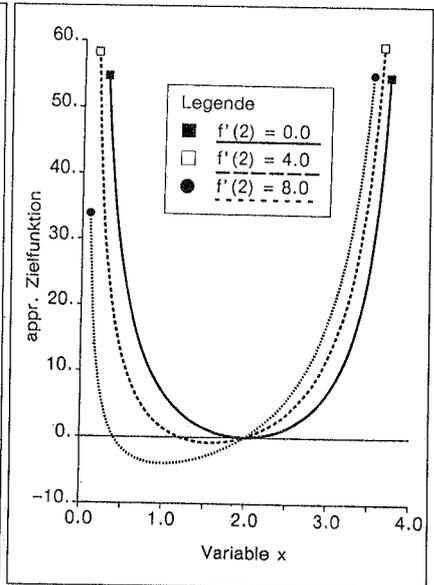
$c_K$  ... Krümmungskorrekturfaktor,  $c_K > 0$

Der Krümmungskorrekturfaktor garantiert auch in Extremfällen (z. B. kleine Krümmung, große Steigung und geringer Abstand der Asymptoten) eine konvexe Approximation. Dabei rückt das Minimum der approximierten Funktion mit abnehmendem Korrekturfaktor

turfaktor näher an die entsprechende Asymptote. Ist der Krümmungskorrekturfaktor null, liefert (3.49) im Grenzfall die alte Approximation (3.29). Bild 3.7 zeigt die Einflüsse von Krümmung, Steigung und Korrekturfaktor auf eine approximierte, eindimensionale Funktion. In Bild 3.8a ist das mit der modifizierten Regel approximierte 2-stäbige Fachwerkproblem dargestellt. Zum Vergleich ist in Bild 3.8b das ursprüngliche Ersatzproblem mit der Approximation nach Svanberg im gleichen Bildausschnitt wiederholt. Mit der modifizierten Approximationsregel besitzt die Ersatzzielfunktion ein Minimum, die Lösung des Ersatzproblems kann nun auch unbeschränkt sein.



**Bild 3.7a:** *Einfluß der Krümmung auf die Approximation*



**Bild 3.7b:** *Einfluß der Steigung auf die Approximation*

Abschließend sei darauf hingewiesen, daß die genannte Modifikation für unbeschränkte Probleme nicht ausreicht, um das Verfahren konkurrenzfähig mit den ausgefeilten Verfahren der mathematischen Programmierung werden zu lassen. Dagegen steht vor allem die mangelhafte Erfassung der Metrik, insbesondere bei nicht separierbaren Problemen. Die Erweiterung paßt sich jedoch nahtlos in das bestehende Instrumentarium ein und wirkt in der Regel oszillationshemmend (auch bei beschränkten Problemen).

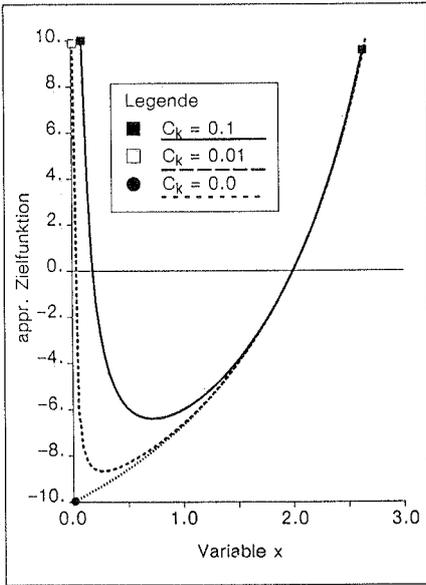


Bild 3.7c: Einfluß des Korrekturfaktors

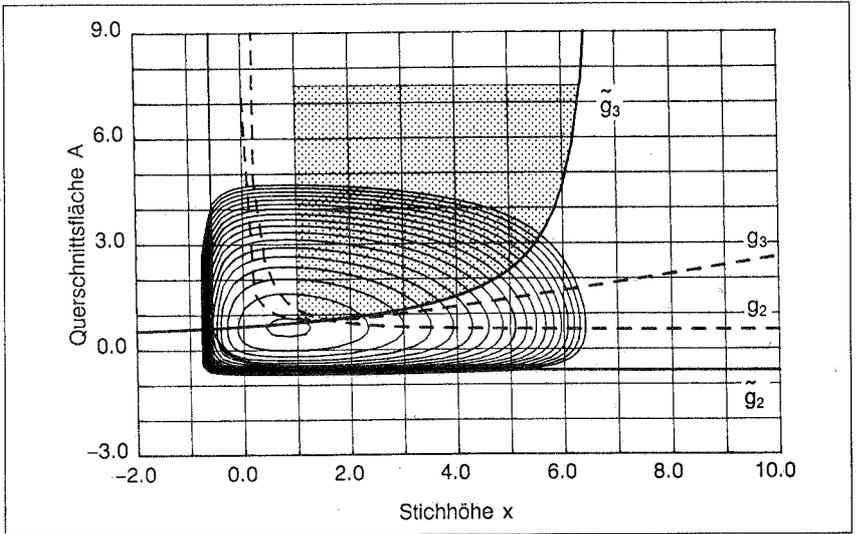
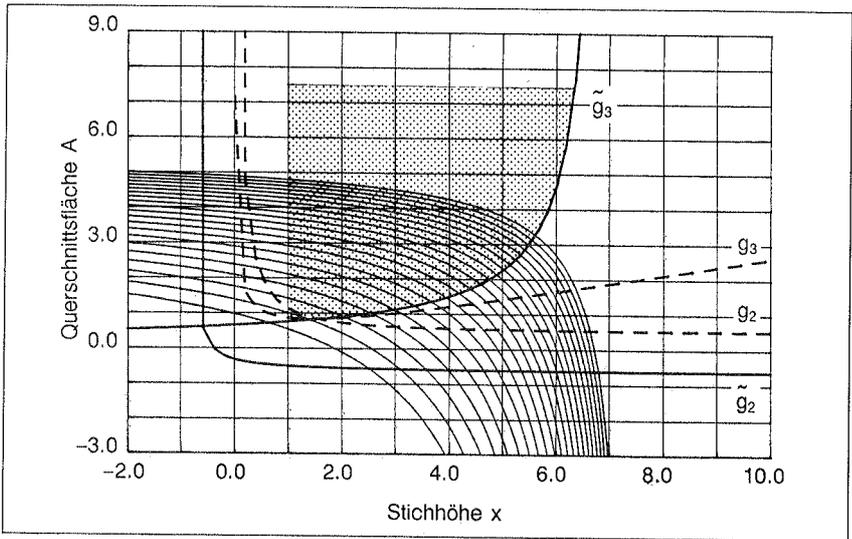


Bild 3.8a: Optimierung des zweistöbigen Fachwerks mit MMA, Iterationsschritt 1 modifizierte Approximation



**Bild 3.8b:** Optimierung des zweistöbigen Fachwerks mit MMA, Iterationsschritt 1  
Approximation nach Svanberg (vergleiche mit Bild 3.4a)

### 3.2.4.5.2 Gleichheitsbedingungen

In der ursprünglichen Formulierung der Methode der beweglichen Asymptoten können keine Gleichheitsbedingungen erfaßt werden, da dann aufgrund der konservativen Approximation die Konvexität des Ersatzproblems nicht garantiert werden kann und infolge davon (3.32) keine reellen Lösungen aufweist. Die einzige mögliche Approximation von Gleichheitsbedingungen wäre demnach eine Linearisierung bezüglich den ursprünglichen Variablen  $x$ . Damit jedoch wäre die explizite Minimierung von (3.31) nicht mehr möglich. Unter Berücksichtigung dieser verschiedenen, sich widersprechenden Umstände wird folgende Approximation von Gleichheitsbedingungen eingeführt, die zwar die Konvexität des Ersatzproblems nicht in letzter Konsequenz garantieren kann, wenigstens jedoch eine Näherung vornimmt, die "im Mittel" die Krümmung null aufweist und die Inkonsistenz von Gleichheitsbedingungen im Ersatzproblem ausschließt. Wie zuvor bei der Zielfunktion werden auch hier beide Asymptoten einer Variablen bei der Approximation berücksichtigt. Zusätzlich soll die Krümmung der approximierten Nebenbedin-

gung in der Mitte des Intervalls  $(l_i, u_i)$  null sein. Damit ergibt sich für die Gleichheitsbedingungen:

$$\bar{h}_j(\mathbf{x}) = r_j + \sum_{i=1}^n \left( \frac{p_{ji}}{(u_i - x_i)} + \frac{q_{ji}}{(x_i - l_i)} \right) \quad (3.50)$$

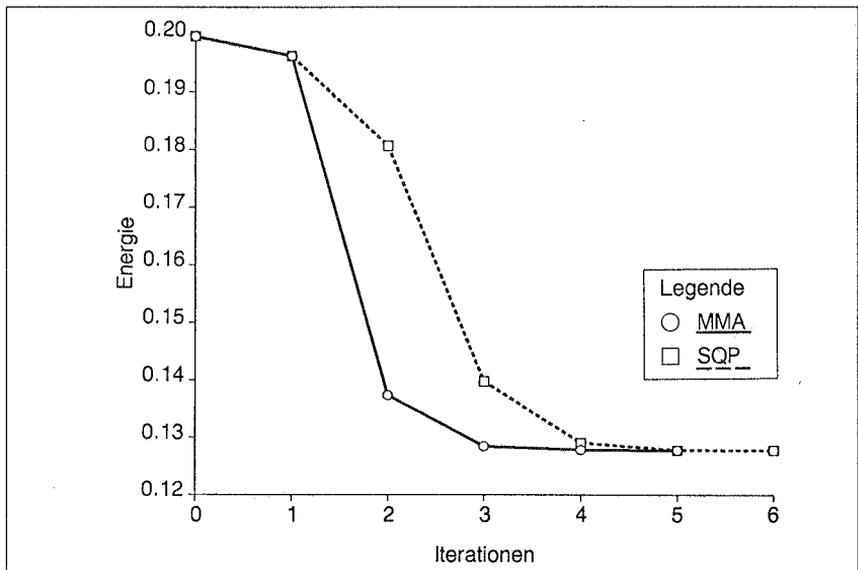
mit:

$$p_{ji} = -q_{ji} = \frac{1}{k_i} \frac{\partial h_j(\mathbf{x}_0)}{\partial x_i}$$

$$r_j = h_j(\mathbf{x}_0)$$

$$k_i = \frac{1}{(u_i - x_{i0})^2} + \frac{1}{(x_{i0} - l_i)^2}$$

Gleichung (3.50) weist eine Nullstelle im Intervall  $(l_i, u_i)$  auf und ist damit konsistent. So ist sichergestellt, daß der Absolutwert des zugehörigen Lagrangemultiplikators nicht so stark ansteigt, daß eine reellwertige Lösung von (3.22) verhindert wird.



**Bild 3.9:** Optimierung mit Gleichheitsbedingung (modifizierte MMA gegen SQP)

Bild 3.9 zeigt den Iterationsverlauf von MMA, der sich mit den geschilderten Modifikationen für Zielfunktion und Gleichheitsnebenbedingungen ergibt, wenn die Formänderungsenergie des zweistöbigen Fachwerks (Fall a) minimiert wird. Dabei soll das Ge-

wicht, das sich für die Startwerte  $x = 4$  und  $A = 0.7$  ergibt, konstant gehalten werden. Zum Vergleich ist der Iterationsverlauf des SQP-Verfahrens (Abschnitt 2.8.6.3) eingetragen. Da in beiden Verfahren mit der Einheitsmatrix als erste Näherung für die Hessematrix begonnen wurde, und die Asymptoten zufälligerweise gerade so liegen, daß die konservative Approximation der quadratischen sehr nahe kommt, sind die Ergebnisse des ersten Iterationsschrittes ungefähr dieselben.

### 3.2.4.5.3 Verallgemeinerte Approximation

Die von Prasad vorgeschlagene "Generalized Inverse-Power Approximation" (3.11), wurde für MMA erweitert. Die verallgemeinerten Variablen  $y$  werden nun in Abhängigkeit eines Exponenten folgendermaßen definiert:

$$y_i = \begin{cases} \frac{1}{b-1} (u_i - x_i)^{1-b} & \text{für } \frac{\partial f(x_0)}{\partial x_i} > 0 \\ \frac{1}{b-1} (x_i - l_i)^{1-b} & \text{für } \frac{\partial f(x_0)}{\partial x_i} < 0 \end{cases} \quad (3.51)$$

Für  $b = 2$  ergibt sich die ursprüngliche Vorschrift (3.25). Die Zielfunktion wird damit folgendermaßen approximiert (entsprechendes gilt für die Nebenbedingungen):

$$\tilde{f}(x) = r_0 + \sum_{i=1}^n \left( \frac{p_{0i}}{(u_i - x_i)^{b-1}} + \frac{q_{0i}}{(x_i - l_i)^{b-1}} \right) \quad (3.52)$$

mit

$$p_{0i} = \begin{cases} \frac{1}{b-1} \frac{\partial f(x_0)}{\partial x_i} (u_i - x_{0i})^b ; & \frac{\partial f(x_0)}{\partial x_i} > 0 \\ 0 & ; \frac{\partial f(x_0)}{\partial x_i} \leq 0 \end{cases}$$

$$q_{0i} = \begin{cases} 0 & ; \frac{\partial f(x_0)}{\partial x_i} \geq 0 \\ -\frac{1}{b-1} \frac{\partial f(x_0)}{\partial x_i} (x_{i0} - l_i)^b & ; \frac{\partial f(x_0)}{\partial x_i} < 0 \end{cases}$$

$$r_0 = f(x_0) - \sum_{i=1}^n \left( \frac{p_{0i}}{(u_i - x_{i0})^{b-1}} + \frac{q_{0i}}{(x_{i0} - l_i)^{b-1}} \right)$$

Wird in (3.51) derselbe Exponent für alle Variablen gebraucht, können die eindimensionalen Minimierungen bei der Bestimmung der dualen Funktion weiterhin explizit ausge-

führt werden. Diese Eigenschaft geht verloren, wenn ein eigener Exponent  $b_i$  für jede Variablensubstitution eingeführt wird. Dagegen können die Nebenbedingungen flexibler und dem Problem angepaßter approximiert werden [114]. Die Exponenten  $b_i$  lassen sich aber auch mit den Krümmungen bestimmen, wenn die Wahl nicht eindeutig ist, z.B. bei statisch unbestimmten Tragwerken oder gemischten Biege- und Normalkraftbeanspruchungen. Für die Exponenten gilt dann:

$b_i > 0$	$b_i < 0$	
$b_i = \frac{f''_{i0}}{f'_{i0}} (u_i - x_{i0})$	$b_i = \frac{f''_{i0}}{f'_{i0}} (l_i - x_{i0})$	; $f'_{i0} \geq 0$
$b_i = \frac{f''_{i0}}{f'_{i0}} (l_i - x_{i0})$	$b_i = \frac{f''_{i0}}{f'_{i0}} (u_i - x_{i0})$	; $f'_{i0} < 0$

(3.53)

mit:  $f'_{i0} = \frac{\partial f(\mathbf{x}_0)}{\partial x_i}$  ;  $f''_{i0} = \frac{\partial^2 f(\mathbf{x}_0)}{\partial x_i^2}$

Die Krümmungen werden auch hier aus der Differenz der Ableitungen am aktuellen und letzten Entwicklungspunkt bestimmt.

Zur Lösung des approximierten konvexen Ersatzproblems wird ein SQP-Algorithmus [117] eingesetzt, der auch bei vielen Variablen und Nebenbedingungen schnell konvergiert. Es ist somit möglich, verschiedene Approximationsvorschriften bei der Formulierung des Ersatzproblems zu mischen, um die Nebenbedingungen ganz individuell und ihren Charakteristiken entsprechend möglichst exakt zu erfassen. Zum Beispiel können dann bei der Querschnittsoptimierung von Fachwerken die in Kräften formulierten Spannungs- und Euler-Knickbedingungen für die ursprünglichen bzw. quadratischen Variablen und Verschiebungen für die reziproken Variablen approximiert und gemeinsam in dasselbe Ersatzproblem eingebracht werden [114].

### 3.2.4.6 Formoptimierung einer Zuglasche

An diesem Beispiel können sehr eindrucksvoll die Leistungsfähigkeit des Verfahrens der beweglichen Asymptoten, aber auch die Grenzen, demonstriert werden. Es soll die Form einer Zuglasche bestimmt werden, so daß das Gewicht minimal werde und die von Mises-Vergleichsspannungen im Bauteil einen vorgeschriebenen, zulässigen Wert nicht überschreiten. Die Ausgangsform sowie die Optimierungsvariablen können Bild 3.10 entnommen werden. Die Aufgabe weist keine augenfälligen Eigenschaften auf, die die Approximation mit den verallgemeinerten Variablen nach Gleichung (3.25) empfiehlt, aber ein Blick auf den Iterationsverlauf (Bild 3.11) beweist den Erfolg des Verfahrens, das schon nach drei Iterationsschritten konvergiert. Im Vergleich dazu ist der Itera-

tionsverlauf mit dem SQP-Verfahren nach Schittkowski [117] aufgetragen. Da das SQP-Verfahren sehr skalierungsanfällig ist, wurde zuvor die Skalierung optimal eingestellt. Die Konvergenzkriterien sind dieselben [117]. Auch hier ist die Konvergenzgeschwindigkeit von MMA hauptsächlich vom Wert des  $s_a$ -Parameters abhängig (Bild 3.12). Ein zweiter Blick auf den Konvergenzverlauf des SQP-Verfahrens (Bild 3.11) zeigt, daß dieses Verfahren gegen Ende zwar langsam konvergiert, aber offensichtlich zu einem geringeren Gewicht als MMA, bei allerdings kaum sichtbaren Unterschieden der "optimalen" Formen. Dieser Unterschied rührt daher, daß das Problem gekoppelt ist und demnach die Annahme der Separierbarkeit nicht zutrifft. Dies kann man den Ergebnissen einer Sensibilitätsanalyse der optimalen Form entnehmen (Bild 3.13). Man sieht, daß die Änderung einiger Variablen Spannungsänderungen von gleicher Größenordnung in entfernten Tragwerksteilen hervorrufen. Das SQP-Verfahren kann diese Kopplung berücksichtigen, bei allerdings geringerer Konvergenzgeschwindigkeit in den ersten Iterationen. Es wird deshalb verschiedentlich vorgeschlagen [27], für die ersten Schritte ein Verfahren der Optimalitätskriterien oder ein MMA entsprechendes und für die weiteren ein Verfahren der mathematischen Programmierung zu verwenden.

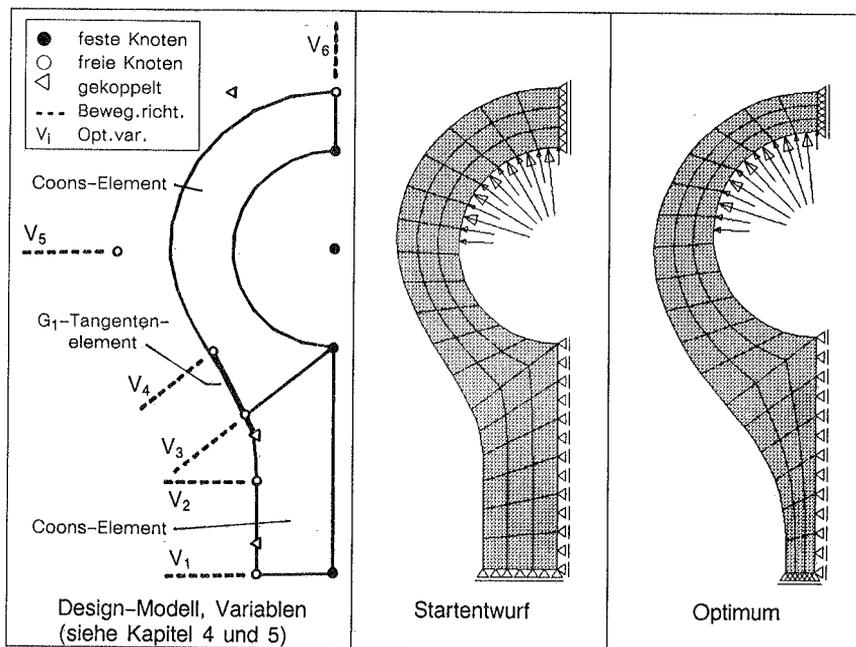


Bild 3.10: Zuglasche, Design-Modell, Startentwurf und Optimum

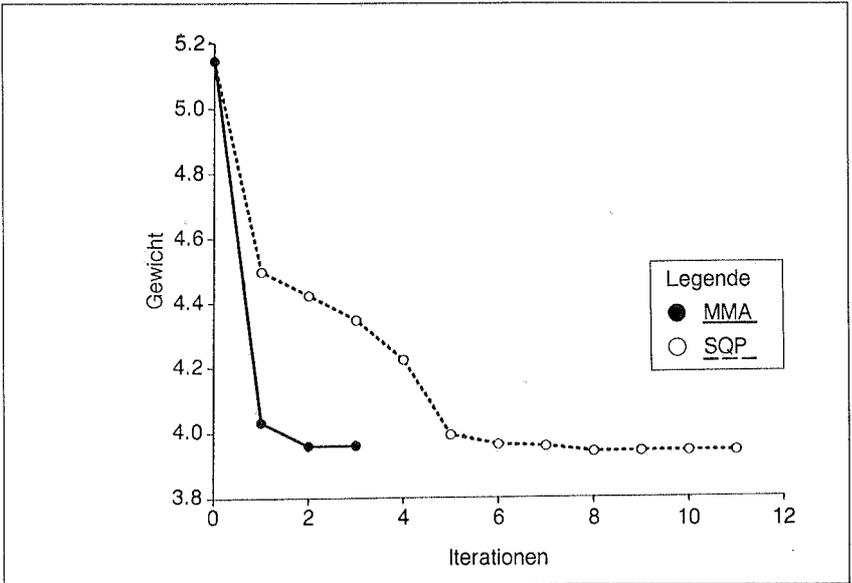


Bild 3.11: Zuglasche, Iterationsverläufe von MMA und SQP

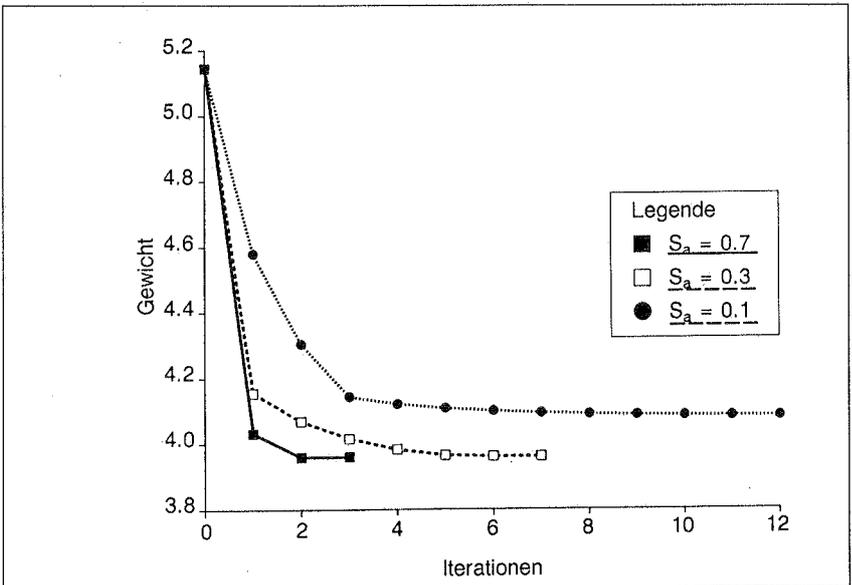
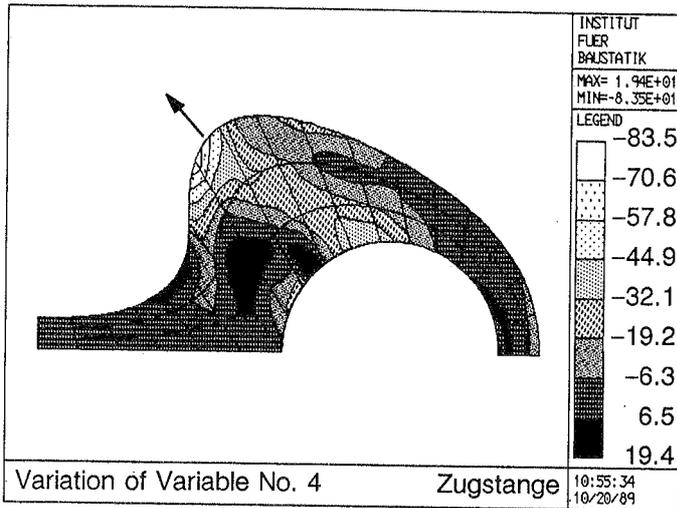


Bild 3.12: Zuglasche, Iterationsverläufe von MMA bei variiertem  $s_a$ -Faktor



**Bild 3.13:** *Zuglasche, Sensibilitätsanalyse für Variable Nr.  $V_4$ ; Änderung der v. Mises Vergleichsspannung in  $[N/mm^2 /m]$  infolge einer Einheitsverschiebung in Richtung von  $V_4$*

### 3.3 Direkte Approximation von Spannungsbedingungen

#### 3.3.1 Fließspannungsbedingungen in Fachwerken

Die Entwicklung expliziter Approximationen für Nebenbedingungen wurde maßgeblich vom Sonderfall der Querschnittsoptimierung statisch bestimmter Fachwerke beeinflusst (Abschnitt 3.2.1). Dies führte zur Linearisierung unter anderem der Fließspannungsbedingungen nach den reziproken variablen Querschnittsflächen. Da die lineare Struktur der Zielfunktion Gewicht bezüglich der ursprünglichen Variablen beibehalten werden sollte, wurde die Technik der hybriden Approximation eingeführt. Deshalb wurde auch eine Modifikation der Programmstruktur notwendig (Bild 3.1).

Eine weitaus einfachere Möglichkeit, die Spannungsbedingungen für den genannten Sonderfall linear in den originalen Variablen auszudrücken, ist, die Nebenbedingungen anstelle von Spannungen in Kräften zu formulieren. Formal resultiert dies in einer Multiplikation der ursprünglichen Spannungsbedingung mit der Querschnittsfläche des entsprechenden Fachwerkstabes. War die Spannungsbedingung (hier für Zug) im  $j$ -ten Stab:

$$g_j = \sigma_j - \beta_s = \frac{F_j}{x_j} - \beta_s \leq 0 \quad (3.54)$$

$\sigma_j, F_j$  ... Spannung bzw. Kraft im  $j$ -ten Stab

$x_j$  ... unbekannte Querschnittsfläche

$\beta_s$  ... Fließspannung

so ergibt sich nun, da  $x_j \geq 0$ :

$$\hat{g}_j = (\sigma_j - \beta_s) x_j = F_j - \beta_s x_j \leq 0 \quad (3.55)$$

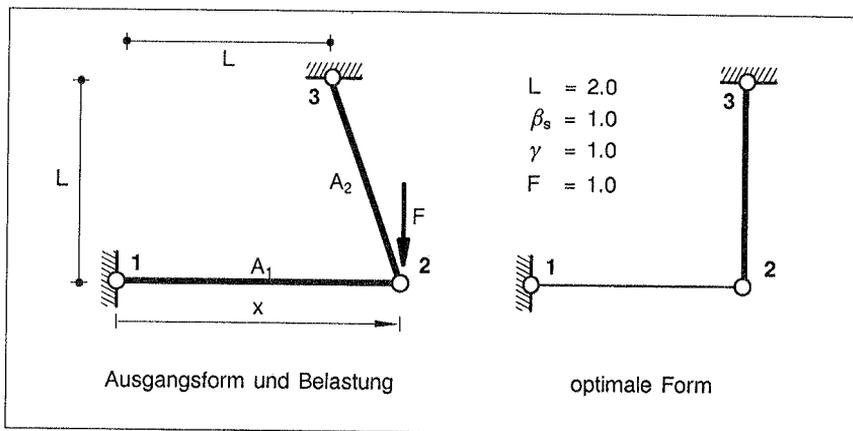
Für statisch bestimmte Fachwerke sind die Stabkräfte unabhängig von der Dimensionierung und Gleichung (3.55) entspricht der "stress-ratio"-Rekursionsformel des "Fully-Stressed-Design" [50]. Querschnittsoptimierungen können mit (3.55) auch bei direkter Verwendung einer Methode der mathematischen Programmierung in einem Schritt ausgeführt werden.

Die modifizierte Nebenbedingung (3.55) verliert auch bei statisch unbestimmten Fachwerken nichts an ihrer Bedeutung, wenn bei der Ableitung die Abhängigkeit der Kraft  $F_j$  von der Variablen  $x_i$  berücksichtigt wird. Es gilt:

$$\frac{\partial \hat{g}_j}{\partial x_i} = (\sigma_j - \beta_s) \delta_{ij} + \frac{\partial \sigma_j}{\partial x_i} x_j \quad (3.56)$$

$\delta_{ij}$ ...Kronnecker-Delta

Die Formulierung der Spannungsbedingung in Kräften hat aber auch auf die Formoptimierung günstige Auswirkungen, wie an folgendem charakteristischen Beispiel gezeigt werden kann. Die optimale Form und die optimalen Querschnittsflächen des in Bild 3.14 gezeigten zweiständigen Fachwerkes sind gesucht, variabel ist dabei die horizontale Koordinate des Lastangriffpunktes.



**Bild 3.14:** Formoptimierung eines zweiständigen Fachwerkes

Die optimale Lösung ergibt sich bei einer vertikal wirkenden Last offensichtlich für  $x = L$ ,  $A_1 = 0$ . Die Spannungsbedingungen in Stab 1 sind:

$$g_1 = -\frac{(x-L)}{LA_1} F - \beta_s \leq 0 \quad (\text{Zug})$$

$$g_2 = \frac{(x-L)}{LA_1} F - \beta_s \leq 0 \quad (\text{Druck})$$

Die Ableitungen der Nebenbedingungen ergeben sich zu:

$$\frac{\partial g_1}{\partial A_1} = \frac{(x-L)}{LA_1^2} F \quad \frac{\partial g_1}{\partial A_2} = 0 \quad \frac{\partial g_1}{\partial x} = -\frac{F}{LA_1}$$

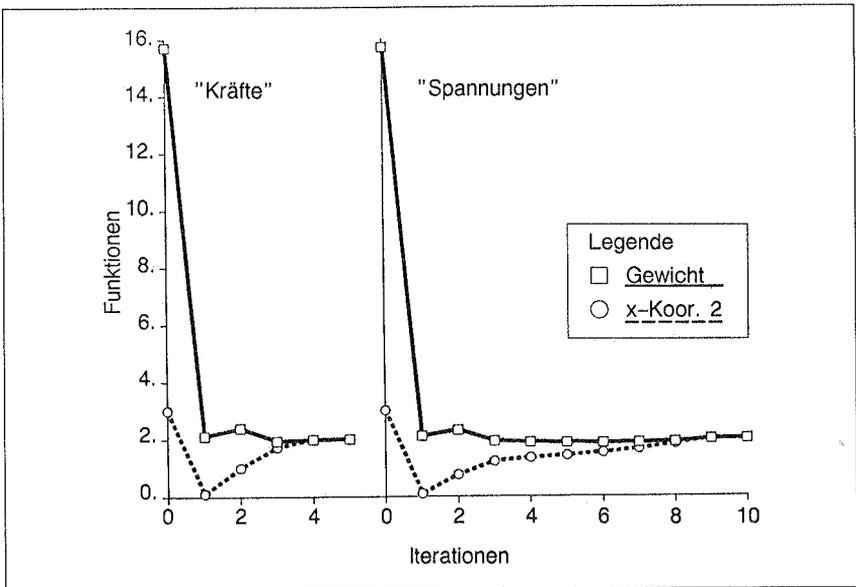
$$\frac{\partial g_2}{\partial A_1} = -\frac{(x-L)}{LA_1^2} F \quad \frac{\partial g_2}{\partial A_2} = 0 \quad \frac{\partial g_2}{\partial x} = \frac{F}{LA_1}$$

Am Optimum  $x = L$ ,  $A_1 = 0$ , haben die Nebenbedingungen jeweils einen Pol mit großen Gradientenänderungen in seiner Umgebung. Da bei einer Lösung der Aufgabe mit Mitteln der mathematischen Programmierung die Querschnittsfläche  $A_1$  aus numerischen Gründen nicht null sein darf, entscheidet die untere Schranke, ob die nicht vorhandene

Qualifikation der Nebenbedingungen an der analytischen Lösung sich in Konvergenzschwierigkeiten des verwendeten Algorithmus (SQP [117]) äußert (Bild 3.15). Die Verwendung einer "active-set"-Strategie kann diesen Effekt mildern, aber nicht verhindern. Die alternative Formulierung in Kräften behebt diesen Schaden:

$$\hat{g}_1 = -\frac{(x-L)}{L} F - \beta_s A_1 \leq 0 \quad (\text{entsprechend für } g_2)$$

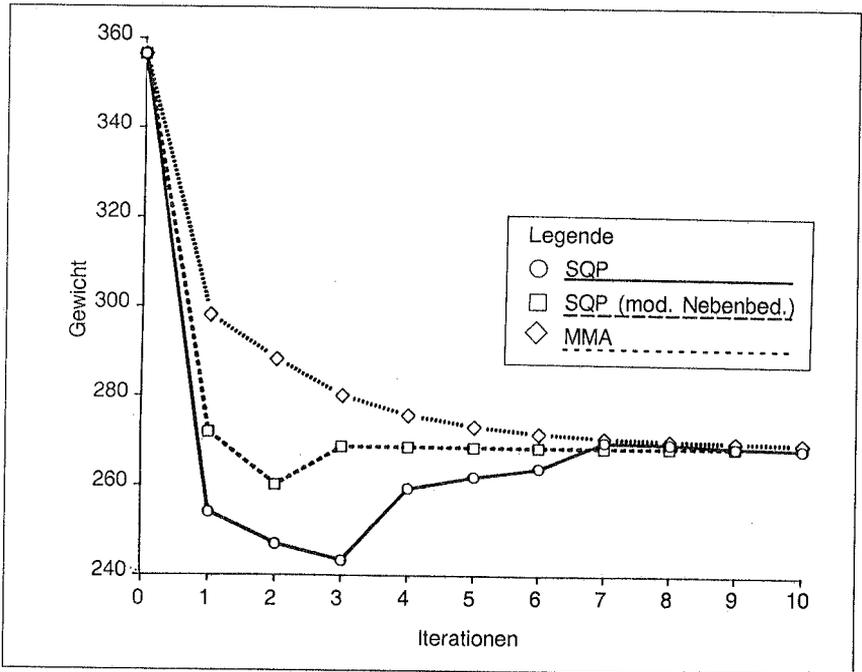
$$\frac{\partial \hat{g}_1}{\partial A_1} = -\beta_s \quad \frac{\partial \hat{g}_1}{\partial A_2} = 0 \quad \frac{\partial \hat{g}_1}{\partial x} = -\frac{F}{L}$$



**Bild 3.15:** Iterationsverläufe, Formulierung der Nebenbedingungen in Kräften / Spannungen

Der Gradient ist numerisch ausgewogen (hier sogar konstant), die Konvergenz wird stabil und schnell (Bild 3.15). Das dargestellte Phänomen stellt sich auch bei komplizierten Tragwerken immer dann ein, wenn ein Stab "wegoptimiert" wird. Die Erfahrung zeigt, daß die Formulierung der Spannungsbedingungen in Kräften die Konvergenz beschleunigt und stabilisiert. Die alternativen Nebenbedingungen können direkt mit einem Verfahren der mathematischen Programmierung angegangen werden und zeigen bei Formoptimierungen Vorteile vor den Approximationstechniken. Bild 3.16 zeigt die unterschiedlichen Iterationsverläufe bei der Formoptimierung eines 13-stäbigen Fach-

werkes (siehe Bild 2.22) mit SQP und a) Bedingungen in Spannungen, b) in Kräften formuliert und c) MMA und konvexe Linearisierung nach Fleury (siehe auch [13]).



**Bild 3.16:** 13-stäbiges Fachwerk, Iterationsverläufe

Seltsamerweise taucht die geschilderte, direkte Approximationstechnik, die einfacher zu verwirklichen ist als die herkömmlichen Techniken, aber von denselben Voraussetzungen ausgeht, nur einmal in der gesichteten Literatur bei Svanberg [134] 1982 auf. Er verfolgt den Aspekt jedoch nicht weiter. Vanderplaats und Salajegheh [145] beschreiben 1989 dieselbe Technik und integrieren diese aber in den Approximationsablauf, wie es in Bild 3.1 dargestellt und in Abschnitt 3.2.4.5.3 kurz beschrieben ist.

### 3.3.2 Knickspannungen in Fachwerken

Soll das Knicken einzelner Stäbe in einem Fachwerk verhindert werden, läßt sich mit der Knickspannung nach Euler folgende Nebenbedingung für die Spannungen formulieren:

$$g_j = -\frac{\pi^2 c E x_j}{l_j^2} - \frac{F_j}{x_j} = \sigma_{kj} - \sigma_j \leq 0 \quad (3.57)$$

mit

$\sigma_{kj}$	... Eulerknickspannung im j-ten Stab
$\sigma_j, F_j$	... Spannung bzw. Kraft im j-ten Stab
$x_j$	... unbekannte Querschnittsfläche (nur Querschnittsoptimierung)
$l_j$	... Stablänge
$c$	... Formfaktor (siehe Anhang A2)
$E$	... E-Modul

Auch diese Bedingung läßt sich in Kräften formulieren:

$$\hat{g}_j = -\frac{\pi^2 c E x_j^2}{l_j^2} - F_j = (\sigma_{kj} - \sigma_j) x_j \leq 0 \quad (3.58)$$

Für den Spezialfall der Querschnittsoptimierung statisch bestimmter Fachwerke kann (3.58) exakt approximiert werden, indem für  $x_j^2$  linearisiert wird. Damit paßt diese Formulierung weitaus besser in ein Approximationskonzept als (3.57), worin sowohl die Originalvariable als auch ihre Inverse vorkommen. Die Erfahrung zeigt, daß bei Formoptimierungen zuerst die Formvariablen und dann die Querschnittsvariablen konvergieren, womit die vorgeschlagene Formulierung in Kräften auch in diesen Fällen wenigstens gegen Ende der Iteration Vorteile zeigt. Allerdings ist (3.58) zumindest im Falle der Querschnittsoptimierung statisch bestimmter Fachwerke konkav, wie die zweimalige Ableitung zeigt:

$$\frac{\partial \hat{g}_j}{\partial x_j^2} = -2 \frac{\pi^2 c E}{l_j^2} \leq 0 \quad (3.59)$$

D.h. bei Anwendung von (3.58) liegt ein nichtkonvexes Problem mit mehreren lokalen und globalen Minima vor. Eine numerisch ermittelte Lösung muß also mit Vorsicht betrachtet werden. Andererseits jedoch ist es gerade bei statisch unbestimmten Fachwerken so, daß die globale Lösung nur mit (3.58) gefunden werden kann, bei Verwendung von (3.57) ist sie im Problem nicht enthalten. Das liegt daran, daß die aus den Dehnungen ermittelte Spannung in einem Fachwerkstab theoretisch auch dann von null verschieden sein kann, wenn der zugehörige Stabquerschnitt gleich null ist, der zugehörige Stab

also aus dem Tragwerk entfernt worden ist. Dies kann an folgendem kleinen Beispiel illustriert werden (Bild 3.17):

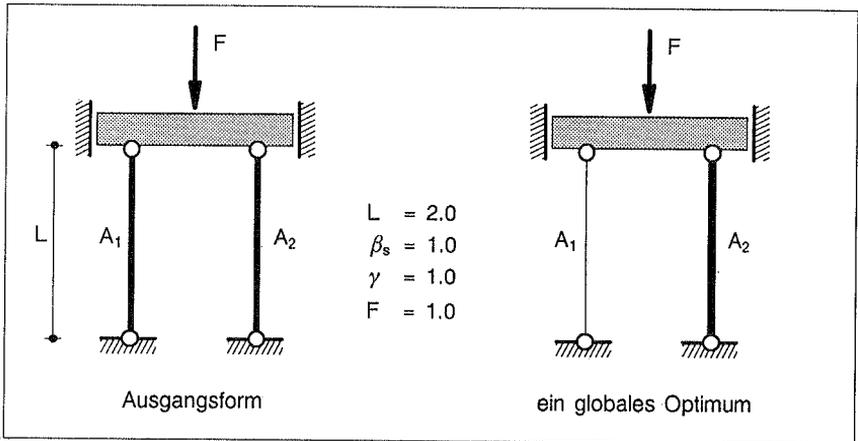


Bild 3.17: Querschnittsoptimierung mit Berücksichtigung des "Euler-Knickens"

Eine Kraft wird über eine starre und nur vertikal verschiebbliche Traverse auf zwei gleich lange Stäbe übertragen. Das Gewicht der Stäbe soll minimiert werden, wobei sie nicht knicken dürfen. Das Materialfließen sei nicht maßgebend, anderenfalls müsste es durch zusätzliche Nebenbedingungen berücksichtigt werden. Die Spannung ist in beiden Stäben gleich groß, selbst beim Grenzübergang, wenn eine der beiden Querschnittsflächen null ist:

$$\sigma_1 = \sigma_2 = \frac{-F}{A_1 + A_2}$$

Damit ergeben sich die Knickbedingungen (in Spannungen):

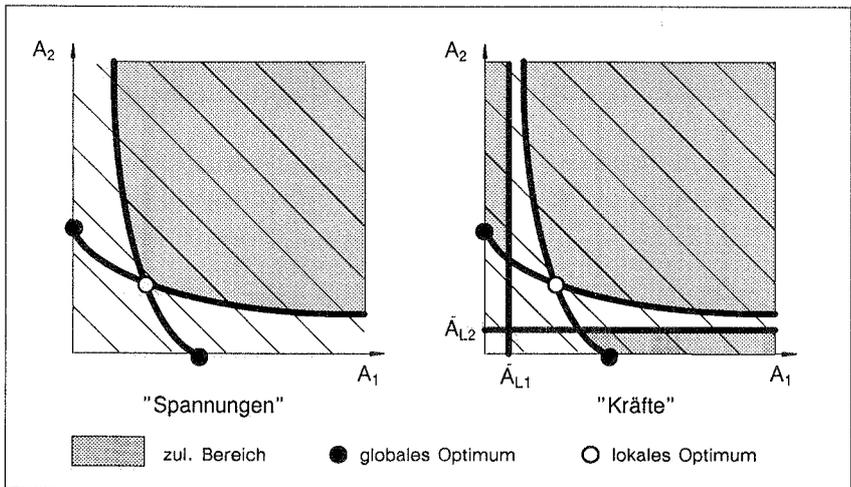
$$g_j = \frac{F}{A_1 + A_2} - \frac{c \pi^2 E A_j}{l^2} \leq 0 \quad ; \quad j = 1, 2$$

Das Gewicht ist:

$$f = \gamma l (A_1 + A_2)$$

Die Lösung der Optimierungsaufgabe ergibt sich am Schnittpunkt der beiden Nebenbedingungen  $g_1 = g_2 = 0$  (Bild 3.18). Offensichtlich jedoch existieren zwei weitere Lösungen, die ein niedrigeres Gewicht aufweisen und die sich dann ergeben müssten, wenn einer der beiden Stäbe entfernt worden ist. Bei der vorliegenden Problemformulierung

liegen sie aber im unzulässigen Bereich. Werden dagegen die Knickbedingungen in Kräften formuliert, sind die Fälle  $A_1 = 0$  bzw.  $A_2 = 0$  möglich, und die globalen Lösungen können ermittelt werden.



**Bild 3.18:** zul. Bereich und alternative Formulierung der Nebenbedingungen

Lösungstechnisch gesehen ergibt sich aber ein Problem. Die neu hinzugekommenen Teile des zulässigen Bereiches sind Abschnitte auf den Variablenachsen ( $A_1 = 0$  oder  $A_2 = 0$ ). Damit ist der zulässige Bereich nichtkonvex und mehrfach zusammenhängend, wobei Teile von niedrigerer Dimension sind. Allein der letztgenannte Punkt verhindert eine Lösung mit einem numerischen Verfahren. Deshalb empfiehlt es sich, eine Transformation vorzunehmen und die Knickbedingungen folgendermaßen zu formulieren, um den zulässigen Bereich etwas aufzuweiten (Bild 3.18):

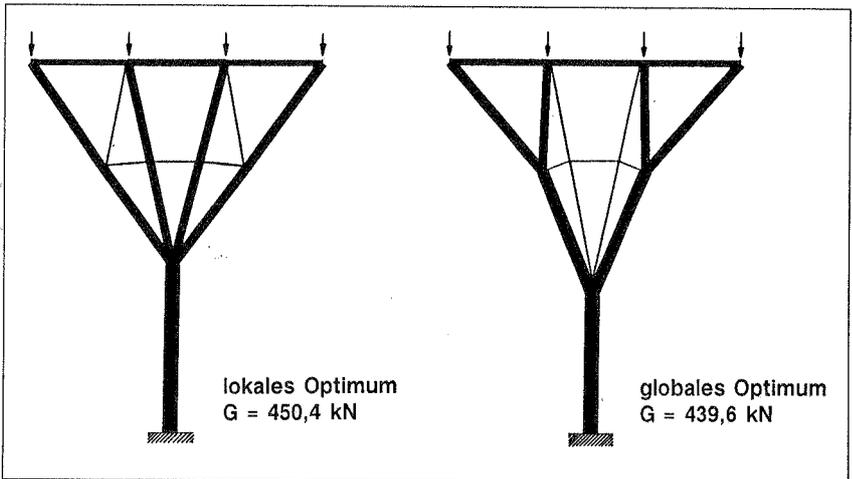
$$\hat{g}_j = \left( \frac{F}{A_1 + A_2} - \frac{c \pi^2 E A_j}{l^2} \right) (A_j - \tilde{A}_{Lj}) \leq 0$$

oder allgemeiner:

$$\hat{g}_j = \left( -\frac{c \pi^2 E x_j}{l^2} - \sigma_j \right) (x_j - \tilde{x}_{Lj}) \leq 0 \quad (3.60)$$

mit  $\tilde{x}_{Lj}$  ... zusätzliche untere Schranke  $\tilde{x}_{Lj} > x_{Lj} \geq 0$

Die globale Lösung ist somit zwar nicht leicht, aber immerhin numerisch bestimmbar. Das lokale Minimum wird etwas in den zulässigen Bereich hineinverschoben, liegt jedoch auf der sicheren Seite. Die zusätzlichen unteren Schranken könnten als Relaxationsvariablen verstanden und in den Optimierungsablauf eingebracht werden, um sie so klein wie möglich zu halten. Es wird dabei vorausgesetzt, daß die Stabquerschnittsfläche den kleinsten zulässigen Wert erreicht hat. Eine Anwendung dieses Vorgehens auf eine komplexere Struktur zeigt Bild 3.19, wo das minimale Gewicht einer Baumstütze gesucht war. Ausgehend von einem hochgradig statisch unbestimmten Tragwerk ergaben sich die beiden dargestellten Formen, die einem lokalen und dem globalen Minimum entsprechen.



**Bild 3.19:** Formoptimierung einer Baumstütze

### 3.3.3 Erweiterung für ebene Flächentragwerke

Die alternative Formulierung der Spannungsbedingungen in Fachwerken lässt sich verallgemeinern und auf ebene Flächentragwerke übertragen, wenn eine charakteristische Dimension des Tragwerks variabel ist. Zum Beispiel soll die Form minimalen Gewichts der in Bild 3.20 dargestellten Konsole bestimmt werden. Variabel sind die Formen der Ober- bzw. Unterkante, die von Mises-Vergleichsspannung der soll einen vorgeschriebenen Wert nicht überschreiten. Die Modifikation der Nebenbedingungen gemäß (3.55), indem sie mit der Höhe der entsprechenden finiten Elemente multipliziert werden, beschleunigt die Konvergenz. Das direkte Vorgehen ist so mit den Approximationsverfahren wie z.B. MMA vergleichbar (Bild 3.20).

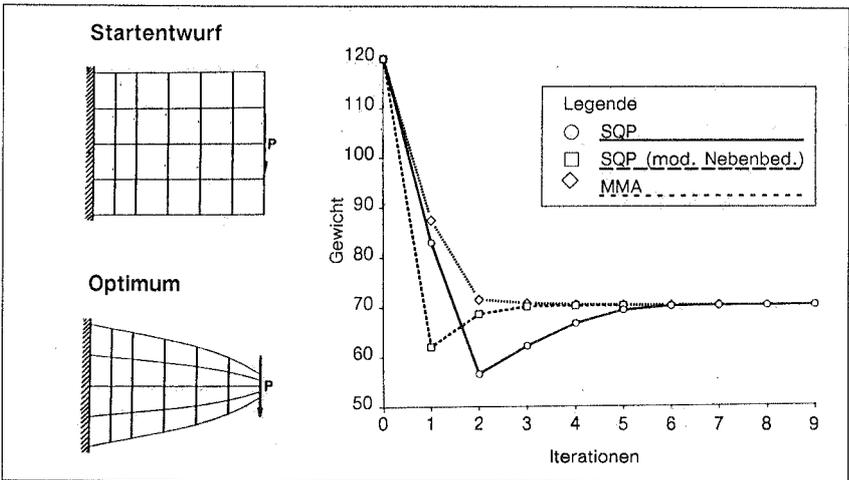


Bild 3.20: Formoptimierung einer Konsole

## 4 FORMBESCHREIBUNG, FORMKONTROLLE

### 4.1 Formoptimierung

In ihrer allgemeinsten Formulierung sind Probleme der Formoptimierung Aufgaben der Variationsrechnung. Gesucht sind Funktionsverläufe, die die Form der zu optimierenden Körper beschreiben und gewisse Zielfunktionale optimal erfüllen. Dies steht im Gegensatz zu den in Kapitel 2 beschriebenen Problemen, wo die Werte von freien Parametern gesucht waren, die Zielfunktionen optimal zu erfüllen hatten. Entsprechend dazu sind hier Nebenbedingungen durch Nebenbedingungsoperatoren anstelle von Funktionen definiert.

Knepe [78] nennt die formbeschreibenden Funktionen "Gestaltsfunktionen", was hier übernommen werden soll, um eine Verwechslung mit dem Begriff "Formfunktion" zu vermeiden, mit dem oft die Ansatzfunktionen von Interpolationsvorschriften bezeichnet werden.

Die Gestaltsfunktion  $\vec{r}$  eines dreidimensionalen Körpers (Bild 4.1) ist i.a. eine Funktion krummliniger körpereigener Koordinaten  $u, v, w$ :

$$\vec{r} = \vec{r}(u, v, w) \quad (4.1)$$

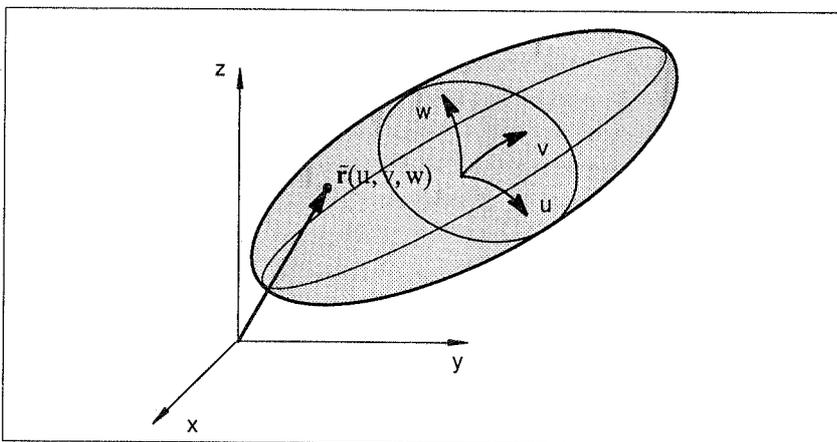


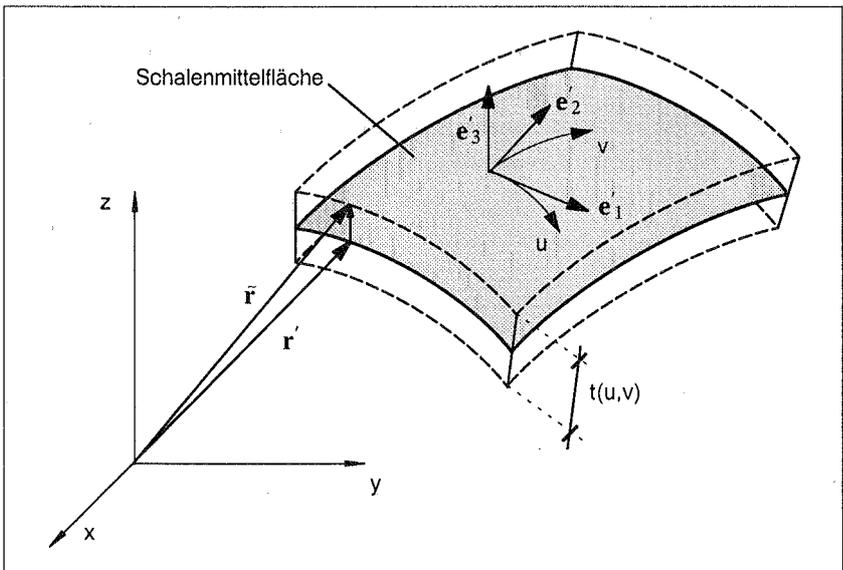
Bild 4.1: Oberfläche eines Körpers

Für Flächentragwerke nimmt die Gestaltsfunktion die bekannte Form an (Bild 4.2):

$$\bar{\mathbf{r}} = \mathbf{r}'(u, v) + \frac{1}{2} t(u, v) \mathbf{e}'_3(u, v) \quad (4.2)$$

- wobei:
- $\mathbf{r}'(u, v)$  ... Gestaltsfunktion der Schalenmittelfläche
  - $t(u, v)$  ... Dickenverteilung der Schale
  - $u, v$  ... krummlinige Flächenkoordinaten
  - $\mathbf{e}'_{1,2}$  ... kovariante Basisvektoren in  $u$ - bzw.  $v$ -Richtung
  - $\mathbf{e}'_3$  ... Schalennormale, senkrecht auf  $\mathbf{e}'_1$  und  $\mathbf{e}'_2$
  - $w$  ... Koordinate in Richtung  $\mathbf{e}'_3$

Die Formoptimierung eines Schalentragwerkes besteht nach (4.2) aus der gleichzeitigen Optimierung der Mittelfläche und der Querschnittsverteilung.



**Bild 4.2:** Gestaltsbeschreibung eines dünnen Flächentragwerkes

Eine typische Aufgabe der Formoptimierung ist die Minimierung des Volumens eines Körpers. Für Flächen ergibt die Lösung dieser Aufgabe die sogenannten Minimalflächen. In diesem Fall ist das Zielfunktional:

$$F = \int_v dv = \iiint |J| du dv dw \quad (4.3)$$

mit:

$J =$  Jakobi-Matrix

$$J = \begin{bmatrix} r_{1,u} & r_{2,u} & r_{3,u} \\ r_{1,v} & r_{2,v} & r_{3,v} \\ r_{1,w} & r_{2,w} & r_{3,w} \end{bmatrix} \quad ; \quad \mathbf{r} = (r_1, r_2, r_3)^T$$

Die Lösung dieser und ähnlicher Aufgaben kann analytisch mittels der Verfahren der Variationsrechnung gefunden werden. Allerdings sind nur wenige geschlossene Lösungen bekannt (z.B. die "Katenoiden", die in jedem Lehrbuch der Variationsrechnung angeführt werden [33]), so daß zur Optimierung von beliebigen Formen Näherungsverfahren wie z.B. das Ritz-Verfahren herangezogen werden. Die Gestaltsfunktion  $\bar{r}$  wird dann durch Ansatzfunktionen  $\mathbf{r}$  angenähert. Die freien Parameter dieser Ansatzfunktionen sind die verbleibenden Unbekannten eines Formoptimierungsproblems. Analog dem Verfahren der finiten Elemente haben auch bei der Formbeschreibung und Formkontrolle im Zusammenhang mit der Strukturoptimierung Verfahren die größte Bedeutung erlangt, die die Form des Tragwerkes bereichsweise durch geeignete Ansätze annähern. Damit können das Variationsproblem in ein Parameteroptimierungsproblem der bekannten Form (NLP) überführt und die im Kapitel 2 beschriebenen Verfahren zur Formoptimierung auch komplexer Tragwerke herangezogen werden. Gewöhnlich bedeutet dieses Vorgehen keine wesentliche Einschränkung der Lösungsvielfalt, da meist von vornherein gewisse ästhetische Vorstellungen oder Anforderungen der Fertigung für die optimale Form bestehen. Weiter kann auf diese Weise die Anzahl der Optimierungsvariablen bei ausreichend großer Formfreiheit wesentlich reduziert werden. Dies wirkt sich günstig auf die Komplexität des Optimierungsmodells sowie die Konvergenzstabilität und -geschwindigkeit der Lösungsverfahren aus. Verschiedentlich wurden die Knoten des finiten Elementmodells als Formparameter eingeführt. Dies erwies sich jedoch schnell als ungeeignet für die Formoptimierung, denn die Zahl der Optimierungsvariablen ist dann von der Diskretisierung abhängig und die Netze neigen zu entarten und damit unbrauchbar zu werden.

Grundsätzlich können zur Formkontrolle und -veränderung während der Optimierung bestehende Programme verwendet werden, die ursprünglich zur Erstellung von FE-Daten konzipiert worden sind (sogenannte Preprozessoren). In Verbindung mit der Strukturoptimierung werden an die Preprozessoren jedoch erhöhte Anforderungen gestellt: sie werden wiederholt (teilweise sehr oft) aufgerufen und werden auch zur Generierung der Gradienteninformationen verwendet. Viel Rechenaufwand und Datenmanipulationen können deshalb gespart werden, wenn die Datenstrukturen von Optimierungs-, Analyse- und Formkontrollprogrammen aufeinander abgestimmt sind, wie dies bei CARAT verwirklicht ist. Dies wird sich bei wiederholten Generierungen auswirken.

Andererseits sollten die Preprozessoren auch in der Lage sein, die Gradienteninformationen direkt zu generieren, d.h. nicht über einen Differenzenschritt.

In den nächsten Abschnitten werden Formgenerierungsverfahren und die Besonderheiten im Zusammenhang mit der Formoptimierung näher beschrieben.

## 4.2 Formgenerierung

### 4.2.1 Das Konzept der Design-Elemente

Wie bereits angedeutet, haben sich Verfahren zur Formgenerierung durchgesetzt, die die Form abschnittsweise approximieren. In Abgrenzung zum Verfahren der finiten Elemente werden hier die Formabschnitte "Design-Elemente" [72], "Super-Elemente" [24] oder "Makroelemente" [125] genannt. Dies auch deshalb, da die Design-Elemente den finiten Elementen übergeordnet sind und zu deren Generierung herangezogen werden. Analog den Verfahren der finiten Elemente ist ein Design-Element durch "Design-Knoten" und "Interpolationsfunktionen", "Ansatzfunktionen" oder auch "Formfunktionen" definiert. Auch gibt es ein-, zwei- und dreidimensionale Design-Elemente (Bild 4.3). In jedem Fall lautet die Interpolationsvorschrift für die Gestalt eines Tragwerks innerhalb eines Design-Elementes:

$$\mathbf{r}(\mathbf{u}, \mathbf{v}, \mathbf{w}, \mathbf{r}_i) = \sum_{i=1}^n \phi_i(\mathbf{u}, \mathbf{v}, \mathbf{w}) \mathbf{r}_i \quad (4.4)$$

mit:

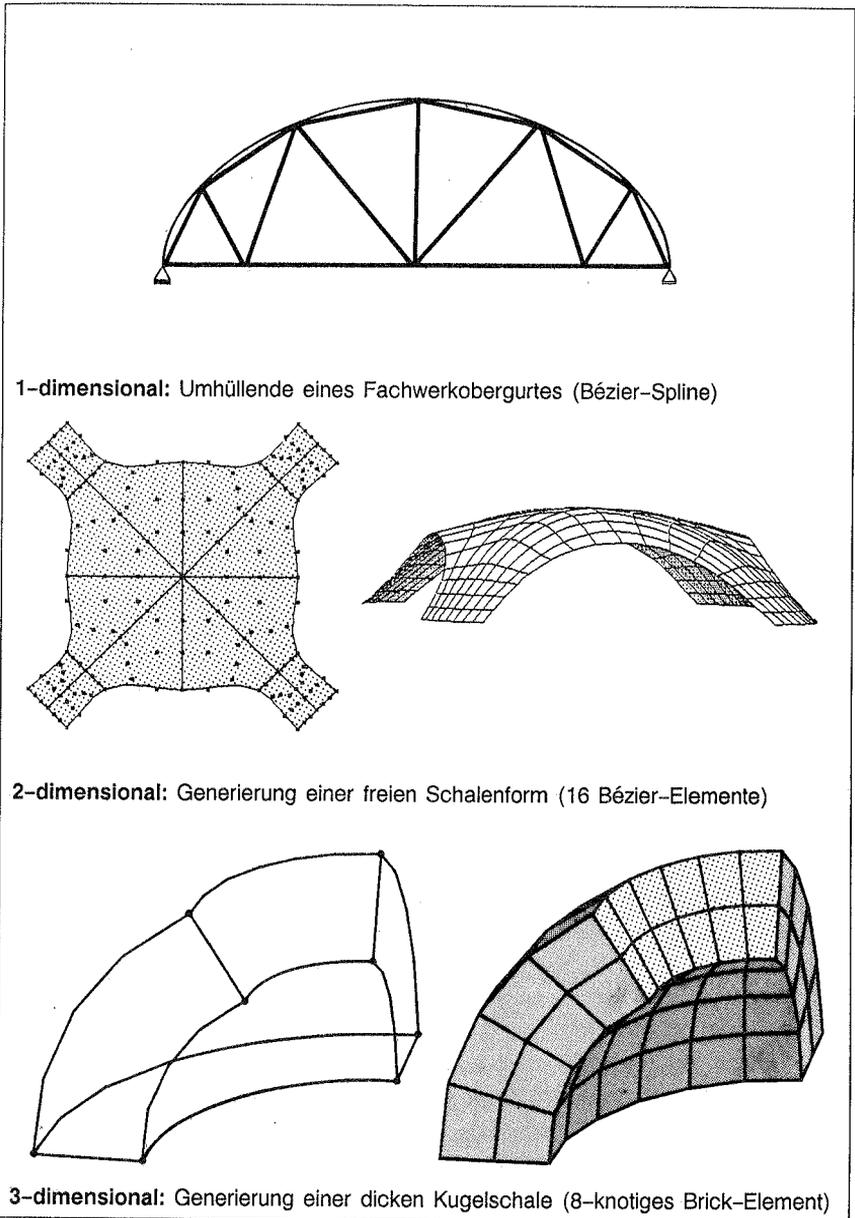
$n$	...	Anzahl der Knoten eines Design-Elementes
$\phi_i$	...	Formfunktionen
$\mathbf{u}, \mathbf{v}, \mathbf{w}$	...	elementeigene Koordinaten
$\mathbf{r}_i$	...	Koordinaten des Design-Knoten $i$ ; $i = 1, \dots, n$

Für die Generierung der Dickenverteilung in Flächentragwerken kann Gleichung (4.4) ebenfalls herangezogen werden. Es gilt dann entsprechend:

$$t(\mathbf{u}, \mathbf{v}, \mathbf{w}, t_i) = \sum_{i=1}^n \phi_i(\mathbf{u}, \mathbf{v}, \mathbf{w}) t_i \quad (4.5)$$

mit:

$t$	...	Dickenverteilung
$t_i$	...	Dicken des Tragwerks an Design-Knoten $i$ ; $i = 1, \dots, n$



**Bild 4.3:** Übersicht über die Einsatzgebiete von Design-Elementen zur Formgenerierung

Mit der Entwicklung und Bereitstellung verschiedenster Formfunktionen zur Beschreibung auch komplexester Formen befaßt sich der Forschungszweig des Computer Aided Geometric Design (CAGD) [6], [15], [39], [40], [55], [99]. Das Konzept der Design-Elemente hat eine weite Verbreitung gefunden und ist die Grundlage für viele Netzgeneratoren und CAD-Verfahren. Dies ist im wesentlichen durch folgende Vorteile begründet, was dieses Konzept auch für die Formoptimierung besonders attraktiv macht:

- Wenige, aber charakteristische Formfreiheitsgrade.
- Beliebig genaue Approximation der angestrebten Form durch
  - Verdichtung des Design-Elementnetzes,
  - andere Ansätze.
- Mögliche Kombination verschiedener Formfunktionen und Design-Elemente zur Beschreibung derselben Form.
- Geringer Eingabeaufwand, dadurch
- bequeme und weitgehend fehlerfreie Generierung von FE-Netzen.
- Interaktive Formkontrolle durch Koordinatenänderungen der Design-Knoten.

Im Sinne der Strukturoptimierung können in einem Design-Element dessen Koordinaten und, wenn es sich um ein ein- oder zweidimensionales Element handelt, dessen Knotendicken sein. Weitere Variablen sind Freiwerte der Übergangsbedingungen zwischen verschiedenen Design-Elementen.

#### 4.2.2 Formoptimierung mit Design-Elementen

Das Konzept der Geometriebeschreibung eines Gegenstandes mittels Design-Elementen eignet sich hervorragend zum Einsatz bei der Formoptimierung. Hier vereinen sich in "optimaler" Weise zwei fundamentale Anforderungen an die Formulierung eines Optimierungsproblems, die sich im Grunde widersprechen:

- die Beschreibung der Geometrie mit möglichst wenig Freiheitsgraden
- eine dennoch möglichst große Formvielfalt.

Der Verdienst, das Design-Element-Konzept in die Strukturoptimierung eingeführt zu haben, wird allgemein Imam [72] '1982 zugesprochen. Er war einer der ersten, die die FE-Netzgenerierung und Formkontrolle als eigenständige Programmodule in den Gesamtprozeß einführten. Imam wendete das Verfahren zur Formoptimierung dreidimensionaler Strukturen an. Eine erste Anwendung auf flächige Strukturen ist von Botkin und Bennett [20] ebenfalls 1982 veröffentlicht worden. In Verbindung mit den Design-Elementen beschreiben Wang et al. [150] 1985 ein Verfahren zur expliziten Bestimmung

der Sensibilität von dreidimensionalen finiten Elementen. Die Begriffe und die Methoden des CAD und CAGD wurden 1985 von Esping [37] und 1986 von Braibant und Fleury [23] verwendet. In diesen Arbeiten wurden die Verfahren zur Optimierung ebener Strukturen eingesetzt. Braibant und Fleury beschreiben hier auch Verfahren zur Verknüpfung der Optimierungsvariablen. Der Stand der Technik ist in [9] umfassend dokumentiert, eine Literaturübersicht kann bei Ding [31] gefunden werden. Eine neuere Arbeit stammt von Yang [154], der ein Programmsystem zur Formoptimierung dreidimensionaler Strukturen mittels Design-Elementen beschreibt.

#### 4.2.3 FE-Netzgeneratoren

In den vergangenen Jahren wurden unzählige Verfahren zur Generierung von FE-Netzen veröffentlicht. Zunächst hauptsächlich auf ebene Flächenstrukturen beschränkt, haben sich die Methoden zuletzt wesentlich verbessert. Die Generierung komplizierter dreidimensionaler Flächen- und Volumenmodelle ist heute selbstverständlich. Dies war auch eine Folge der raschen Entwicklung der Computertechnologie und der Analyseprogramme. Einen der ersten Generatoren zur Definition ebener und räumlich gekrümmter finiter Elementnetze beschreiben Zienkiewicz und Phillips [156] 1971. Ein Programmcode zur Generierung ebener Netze wurde 1979 von Hinton und Owen [69] publiziert. In diesen beiden Arbeiten wurden Design-Elemente mit den bekannten Formfunktionen der isoparametrischen finiten Elemente verwendet. Ein ähnliches Verfahren wurde von Bremer 1986 [24] angegeben. Dort kann auch ein weiterer Überblick und eine umfassende Literaturliste gefunden werden. Lo [84] beschreibt eine Methode zur Generierung von räumlich gekrümmten Flächen. Ein interaktiv unterstütztes Verfahren, das transfinit Transformationen (siehe Coons-Elemente, Abschnitt 4.3.2.3.3) zur Abbildung der Geometrie verwendet, wird von Haber et al. [58] angegeben. Die Autoren heben in ihrer Arbeit die Notwendigkeit einer graphisch-interaktiven Unterstützung des Generierungsvorganges besonders hervor. Neuere Übersichtsarbeiten wie z.B. von Shepard et al. [127] oder Schwenzer [125] zeigen die aktuellen Entwicklungstendenzen auf: Mehr und mehr werden Netzgeneratoren Bestandteil einer umfassenden geometrischen Beschreibung der zu berechnenden Bauteile mittels der Verfahren des CAD und des CAGD. Spezielle Probleme aus der Sicht des CAGD in Verbindung mit Netzgenerierungen werden in [41] behandelt. Dabei gibt es grundsätzlich zwei Möglichkeiten, die CAD-Programme mit den Netzgeneratoren zu verbinden: einerseits über spezielle Interface-Routinen oder zum anderen durch eine direkte Integration der Programm-Module. Der letztere Weg empfiehlt sich insbesondere für wiederholte Anwendungen wie z. B. in der Strukturoptimierung. Eine Integration von CAD- und FE-Programmteilen ist insbesondere in CARAT [12], [75], [132], aber auch in vielen kommerziellen Programmpaketen wie ANSYS [30] oder NASTRAN verwirklicht. Eigenständige Programmpakete zur

Computer- unterstützten Erstellung von FE-Netzen sind PATRAN [93], SuperTab oder IDEAS. Ein heutzutage nicht mehr wegzudenkendes Hilfsmittel ist die interaktive graphische Unterstützung, die gerade von "workstations" optimal gewährt wird. Verbreitete Anwendung dieser hochentwickelten Verfahren zur Generierung von FE-Daten finden sich in der Automobil-, der Luft- und Raumfahrtindustrie sowie des Maschinenbaus. Das Bauwesen ist gerade dabei, der Entwicklung zu folgen.

## 4.3 Formgenerierung und -optimierung mit CARAT

### 4.3.1 Zur Programmorganisation

Das Programmsystem CARAT verfügt über den Preprozessor CARAT-Design. Dieser Preprozessor ist direkt in das System integriert und teilt dieselbe Incore-Datenbasis wie alle anderen Programmodule des Systems. Damit ist der in der Strukturoptimierung beträchtliche Aufwand für den Datentransfer auf ein Minimum reduziert, der durch die Interaktion von Optimierungs-, Entwurfs- und Analysemodellen entsteht (siehe Kapitel 1, Bild 1.1). CARAT-Design hat im wesentlichen drei Aufgaben zu erfüllen, die im Eingabe- und Durchführungsteil von CARAT anfallen, weshalb CARAT-Design eher einen "Perprozessor" als einen Preprozessor darstellt:

- Generierung und interaktive Modifikation der FE-Daten.
- Modifikation der Geometrie und Dickenverteilung während der Optimierung.
- Generierung der abgeleiteten FE-Daten nach Vorgabe der abgeleiteten Design-Parameter für die Sensibilitätsanalyse.

Im Bild 4.4 sind schraffiert die Stellen im Gesamtablauf von CARAT dargestellt, an denen CARAT-Design eingreift. Für den Gesamtaufwand der Berechnungen ist aus der Sicht von CARAT-Design vor allem die effiziente Programmierung der beiden Schlüsselstellen im Optimierungszyklus von Bedeutung. Durch die gemeinsame Incore-Datenbasis stehen zu jeder Zeit alle Design- und FE-Daten zur Verfügung, so daß an dieser Stelle ausschließlich die sich ändernden Werte generiert werden müssen. In der vorliegenden Implementierung sind dies die Koordinaten des FE-Netzes und die Dickenverteilung bei fester Topologie. Die Ableitung des Design-Modells nach den Optimierungsvariablen ist explizit programmiert. Besonderer Wert wurde dabei auf die Ableitung der Kontinuitätsbedingungen zwischen Design-Elementen gelegt, so daß insgesamt ein äußerst effektiver Algorithmus vorliegt, für beides: die Formmodifikation und die analytische Sensitivitätsanalyse während einer Optimierung.

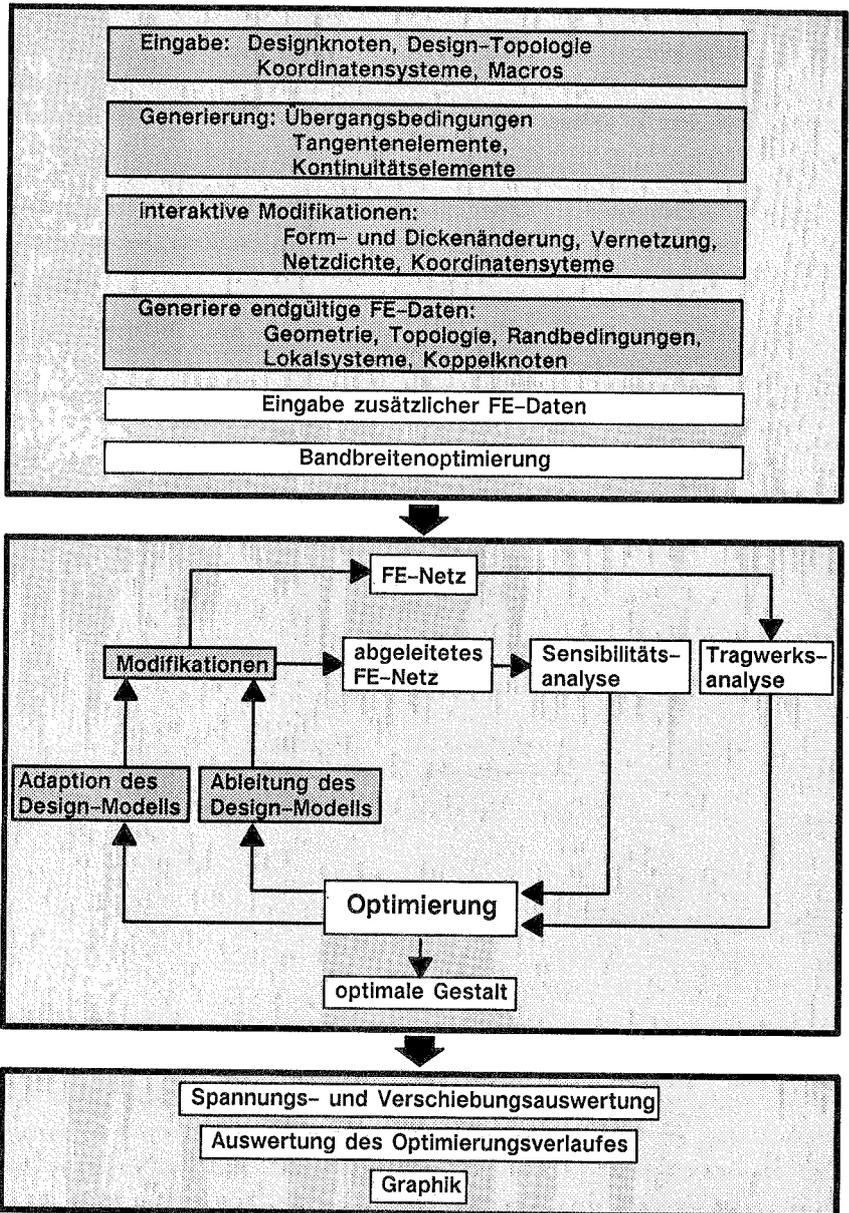


Bild 4.4: Integration von CARAT-Design in den Gesamttablauf

### 4.3.2 Elementkatalog

#### 4.3.2.1 Elementhierarchie

Die Design-Elemente sind hierarchisch geordnet. Die jeweils höhere Stufe wird von der jeweils vorhergehenden definiert (Bild 4.5):

- 1-D-Elemente, sogenannte "Design-Kanten", durch Design-Knoten.
- 2-D-Elemente durch Kanten (da 2-D-Elemente am meisten eingesetzt werden, werden sie im folgenden auch als "Design-Elemente" bezeichnet).
- 3-D-Elemente, sogenannte "Design Bricks", durch Design-Elemente.

Zusätzlich sind innere Knoten zu beachten, wie sie bei 9- bzw. 16-knotigen 2-D-Elementen oder 27-knotigen Bricks vorkommen können.

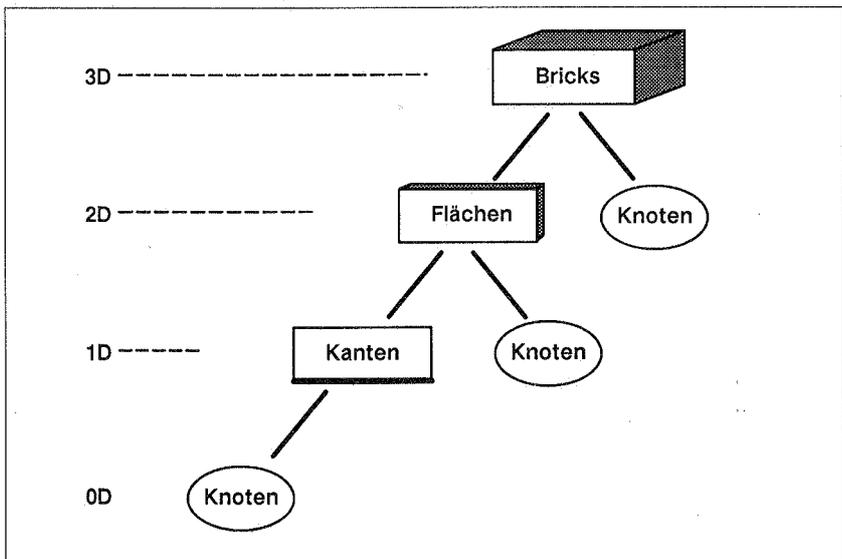
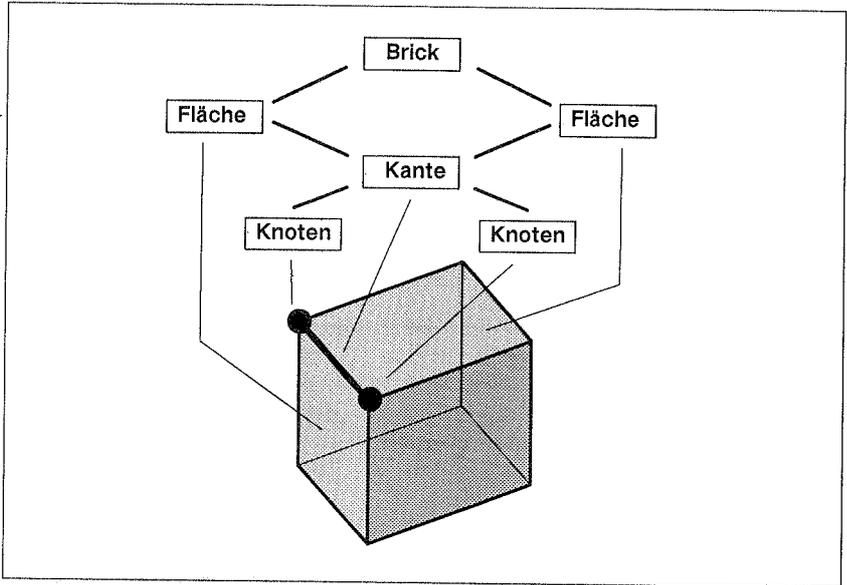


Bild 4.5: Hierarchie der Design-Elemente

Diese Ordnung orientiert sich am Vorgehen vieler CAD-Verfahren und erlaubt mannigfache topologische Verknüpfungen zwischen den einzelnen Hierarchiestufen (siehe Bild 4.6, wo eine Kante gleichzeitig Teil zweier Flächen und eines Brick ist). Neben der natürlichen, am Vorgang des Entwerfens orientierten kontinuierlichen Dimensionserweiterung mit zunehmender Hierarchieebene hat dieses Vorgehen den Vorteil, daß benachbarte Design-Elemente automatisch kompatibel sind. Zum Beispiel haben die in Bild 4.6 dargestellten Flächen-Elemente gemeinsame Kanten. Diese Kanten sind Träger zusätzlicher Informationen über die Anzahl der zu generierenden finiten Elemente, ihre

Verdichtung, die Randbedingungen und Kontinuitätsbedingungen. Automatisch haben beide Design-Elemente entlang dieser Kante dieselben Eigenschaften, zusätzliche aufwendige Datenabgleiche sind somit überflüssig. Entsprechendes gilt für gemeinsame Oberflächen von Design Bricks.



**Bild 4.6:** *Topologische Zusammenhänge zwischen Design-Elementen verschiedener Hierarchieebenen.*

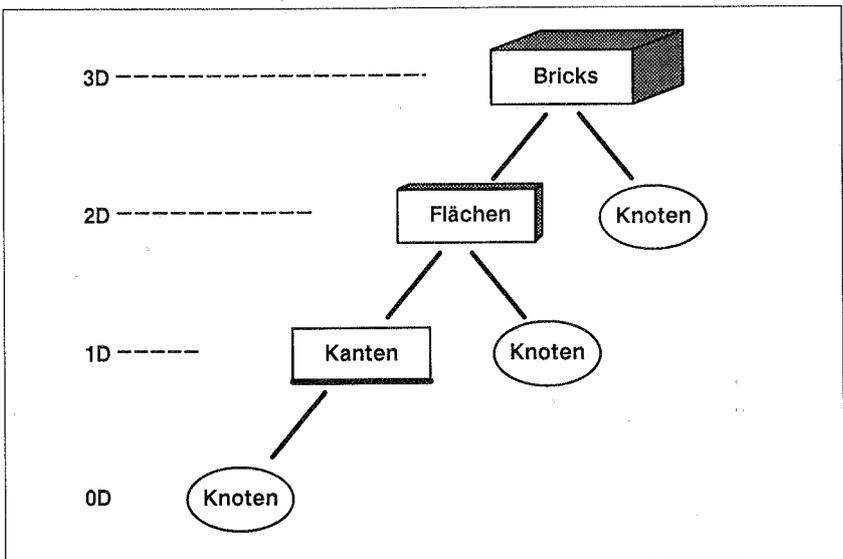
## 4.3.2 Elementkatalog

### 4.3.2.1 Elementhierarchie

Die Design-Elemente sind hierarchisch geordnet. Die jeweils höhere Stufe wird von der jeweils vorhergehenden definiert (Bild 4.5):

- 1-D-Elemente, sogenannte "Design-Kanten", durch Design-Knoten.
- 2-D-Elemente durch Kanten (da 2-D-Elemente am meisten eingesetzt werden, werden sie im folgenden auch als "Design-Elemente" bezeichnet).
- 3-D-Elemente, sogenannte "Design Bricks", durch Design-Elemente.

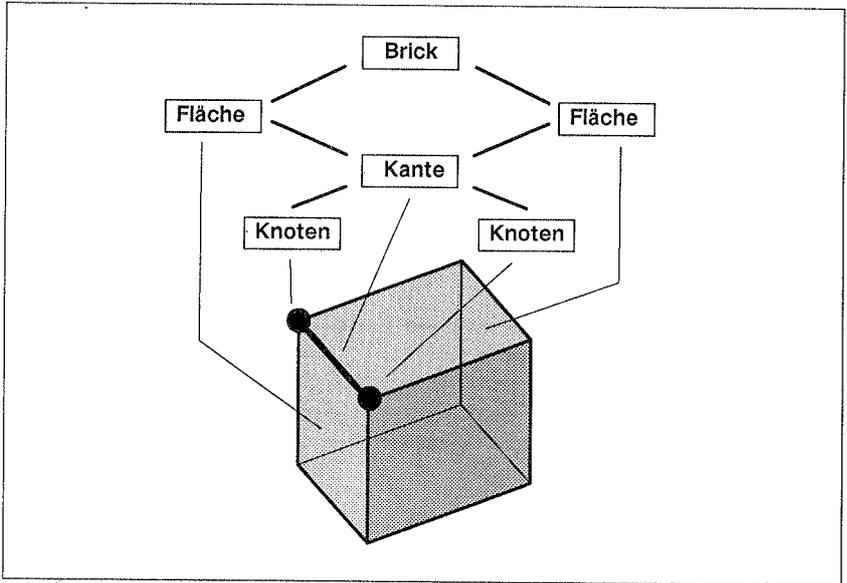
Zusätzlich sind innere Knoten zu beachten, wie sie bei 9- bzw. 16-knotigen 2-D-Elementen oder 27-knotigen Bricks vorkommen können.



**Bild 4.5:** Hierarchie der Design-Elemente

Diese Ordnung orientiert sich am Vorgehen vieler CAD-Verfahren und erlaubt mannigfache topologische Verknüpfungen zwischen den einzelnen Hierarchiestufen (siehe Bild 4.6, wo eine Kante gleichzeitig Teil zweier Flächen und eines Brick ist). Neben der natürlichen, am Vorgang des Entwerfens orientierten kontinuierlichen Dimensionaufweitung mit zunehmender Hierarchieebene hat dieses Vorgehen den Vorteil, daß benachbarte Design-Elemente automatisch kompatibel sind. Zum Beispiel haben die in Bild 4.6 dargestellten Flächen-Elemente gemeinsame Kanten. Diese Kanten sind Träger zusätzlicher Informationen über die Anzahl der zu generierenden finiten Elemente, ihre

Verdichtung, die Randbedingungen und Kontinuitätsbedingungen. Automatisch haben beide Design-Elemente entlang dieser Kante dieselben Eigenschaften, zusätzliche aufwendige Datenabgleiche sind somit überflüssig. Entsprechendes gilt für gemeinsame Oberflächen von Design Bricks.



**Bild 4.6:** *Topologische Zusammenhänge zwischen Design-Elementen verschiedener Hierarchieebenen.*

## 4.3.2.2 1-D-Elemente

### 4.3.2.2.1 Lagrange-Elemente

Vor allem in der Methode der finiten Elemente viel gebrauchte Formfunktionen zur Interpolation von Geometrie und Verschiebungen sind Polynomansätze verschiedener Ordnung. Hier werden zur Beschreibung der Gestalt eines Tragwerks (siehe Gl. (4.4)) Polynome 1., 2. und 3. Ordnung verwendet. Dem entsprechen 1-D-Design-Elemente oder Design-Kanten mit 2, 3 oder 4 Knoten (siehe Bild 4.7). Die Ansatzfunktionen sind in der lokalen Koordinate  $u$  definiert, die von 0 bis 1 läuft.

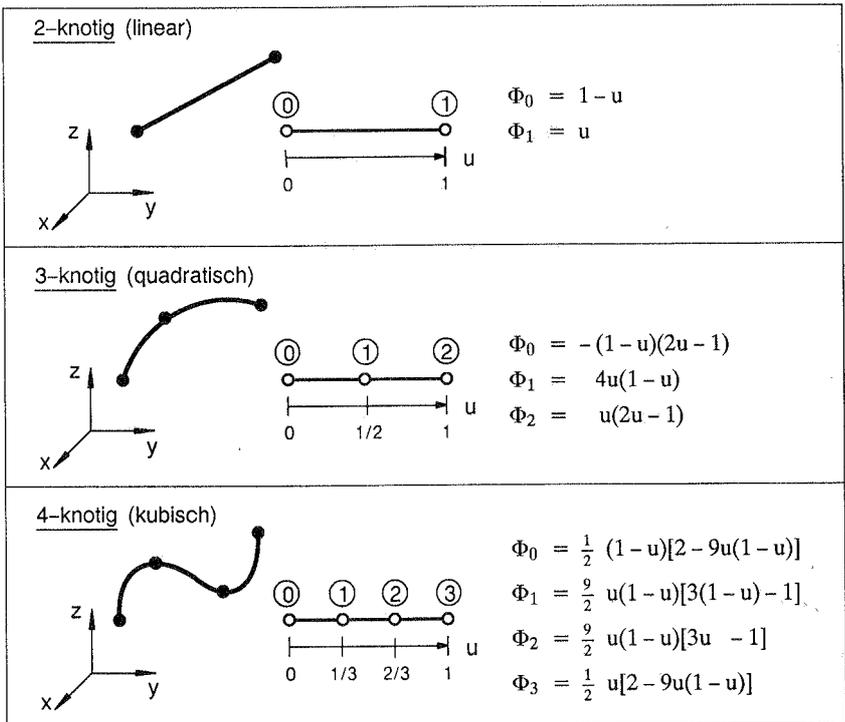


Bild 4.7: 1-D Lagrange Elemente

#### 4.3.2.2.2 Bézier-Spline

Alternativ zu der oben angegebenen kubischen Lagrange-Interpolation, die als Funktion von vier Kontrollknoten formuliert ist, kann dieselbe Kurve auch durch jeweils zwei Kontrollknoten und Tangentenvektoren am Anfang ( $u = 0$ ) bzw. Ende ( $u = 1$ ) des Design-Elementes bestimmt werden. Dies wird auch Hermite-Interpolation genannt. Ferguson führte diese Darstellung als erster zur Beschreibung von Kurven und Flächen beim Entwurf von Flugzeugen ein [40]. Ein Punkt  $r$  wird dann durch folgende Interpolationsvorschrift gewonnen (siehe auch Bild 4.8).

$$\begin{aligned} r(u) = & r_0 (1 - 3 u^2 + 2 u^3) + r_1 (3 u^2 - 2 u^3) \\ & + \dot{r}_0 (u - 2 u^2 + u^3) + \dot{r}_1 (-u^2 + u^3) \end{aligned} \quad (4.6)$$

mit:  $r_0, r_1 \dots$  Ortsvektoren der Anfangs- bzw. Endknoten  
 $\dot{r}_0, \dot{r}_1 \dots$  Tangentenvektoren am Anfang und Ende der Kurve

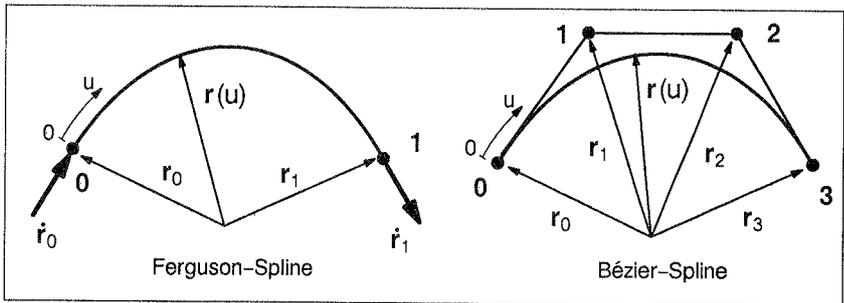


Bild 4.8: kubische Splines

Die Gleichung (4.6) eignet sich wenig für eine direkte Manipulation, wie sie beim interaktiven Entwerfen notwendig ist, da die Tangentenvektoren schlecht graphisch dargestellt werden können und für den Entwerfer recht abstrakte Größen darstellen. Eine alternative Formulierung von (4.6), die diese Nachteile überwindet, stammt von Bézier. Hier werden die Tangentenvektoren ersetzt durch die Differenzvektoren von Anfangs- bzw. Endknoten eines kubischen Design-Elementes zu zwei zusätzlich eingeführten Kontrollknoten (Bild 4.8). Somit ergibt sich ein Punkt  $r$  innerhalb des Elementes nach der Vorschrift:

$$r(u) = (1 - u)^3 r_0 + 3 u (1 - u)^2 r_1 + 3 u^2 (1 - u) r_2 + u^3 r_3 \quad (4.7)$$

Die Formfunktionen sind dann:

$$\begin{aligned}\Phi_0 &= (1-u)^3 \\ \Phi_1 &= 3u(1-u)^2 \\ \Phi_2 &= 3u^2(1-u) \\ \Phi_3 &= u^3\end{aligned}$$

Wie das 4-knotige Lagrange-Element ist auch das kubische Bézier-Element durch vier Kontrollknoten bestimmt, jedoch ist zu beachten, daß die beiden inneren Knoten Nr. 1 und 2 nicht auf der generierten Kurve liegen. Deshalb wird die Transformationsvorschrift des Bézier-Spline auch als Approximation bezeichnet, im Gegensatz zu den Lagrange-Interpolationen. Eine Verallgemeinerung für höhere Polynome wird durch die Bernstein-Polynome angegeben:

$$\mathbf{r}(u) = \sum_{i=0}^n \frac{n!}{(n-i)! i!} u^i (1-u)^{n-i} \mathbf{r}_i \quad (4.8)$$

mit:

- u ... lokale Koordinate  $0 \leq u \leq 1$
- $\mathbf{r}_i$  ... Kontrollknoten
- n ... Grad des Polynoms

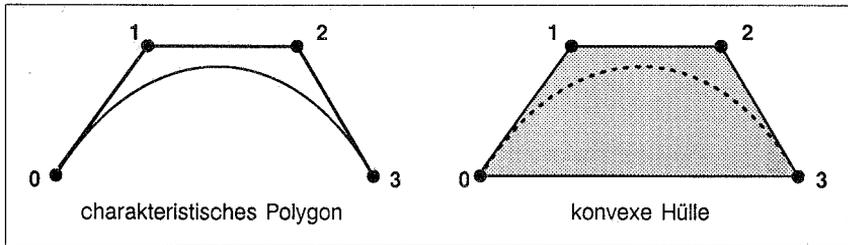


Bild 4.9: Eigenschaften des Bézier-Splines

Der kubische Bézier-Spline ergibt sich für  $n = 3$ . Grundsätzlich liegen alle inneren Kontrollknoten nicht auf der approximierten Kurve. Eine besondere Eigenschaft des Bézier-Spline ist das charakteristische Polygon (Bild 4.9), das sich aus der geradlinigen Verbindung der Kontrollknoten ergibt. Innerhalb der "konvexen Hülle" dieses charakteristischen Polygons wird die generierte Kurve stets zu liegen kommen. Dies ist besonders in Verbindung mit der Strukturoptimierung eine wichtige Eigenschaft, da allein durch die Kenntnis der geometrischen Orte der Kontrollpunkte die resultierende Form abgeschätzt und damit schon vor der Generierung durch geeignete Restriktionen für die Formfreiheiten der Kontrollpunkte entartete Formen vermieden werden können. Mit zunehmendem Grad der Polynome nimmt allerdings die Aussagekraft des charakteristischen Polygons ab.

#### 4.3.2.2.3 B-Spline

Wie der Bézier-Spline ist auch der B-Spline eine Approximationsvorschrift, d.h. die generierte Kurve geht in der Regel nicht durch die Kontrollknoten. Eine umfassende Darstellung kann bei de Boor [18] und Bartels et al. [6] gefunden werden. Einen einführenden Überblick geben [40] und [15]. Die B-Splines, auch Fundamental-Splines oder Basisfunktionen genannt, sind Funktionen eines Parameters  $u$ , die abschnittsweise aus Polynomen des Grades  $n$  zusammengesetzt werden. Sie sind nur innerhalb eines Intervalls  $(u_i, u_{i+n+1})$  ungleich null und  $C_{n-1}$ -kontinuierlich. Zum Beispiel ergibt sich ein kubischer B-Spline mit Grad  $n = 3$  aus 4 Abschnitten, die von 5 Stützstellen (Parameterknoten) definiert werden (Bild 4.10). Dieser B-Spline ist  $C_2$ -kontinuierlich. Den 16 unbekanntenen Koeffizienten der vier kubischen Abschnittspolynome stehen 15 Übergangsbedingungen gegenüber. Die noch fehlende zusätzliche Bedingung ist, daß die Fläche unter der Kurve gleich eins sein soll. Analog Gleichung (4.4) lautet die Approximationsvorschrift:

$$\mathbf{r}(u) = \sum_{i=0}^m B_i^n(u) \mathbf{r}_i \quad (4.9)$$

$m + 1$  ... Anzahl der Kontrollknoten  $\mathbf{r}_i$

$n$  ... Grad des B-Spline

$B_i^n(u)$  ... B-Spline, der dem  $i$ -ten Kontrollknoten zugeordnet ist

$u$  ... Parameter

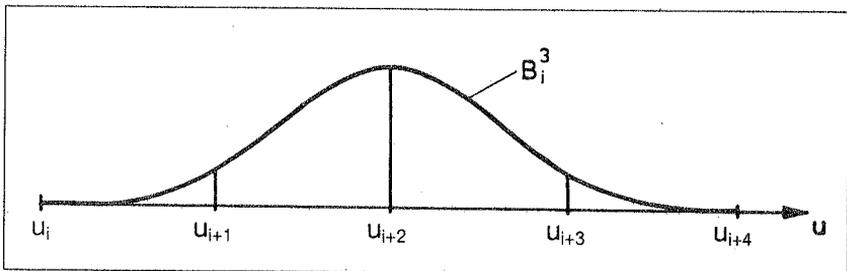


Bild 4.10: Kubischer B-Spline

Parameter- und Kontrollknoten dürfen nicht verwechselt werden. Eine Kurve, die nach (4.9) mit 4 Kontrollknoten und kubischen B-Splines approximiert werden soll, hat insgesamt 8 Parameterknoten, die zur Definition der Basisfunktionen notwendig sind. Sie sind in Bild 4.11 dargestellt. Es wird auch deutlich, daß kein Kontrollknoten auf der approximierten Kurve liegt und sie nur zwischen dem 4. und 5. Parameterknoten bestimmt ist.

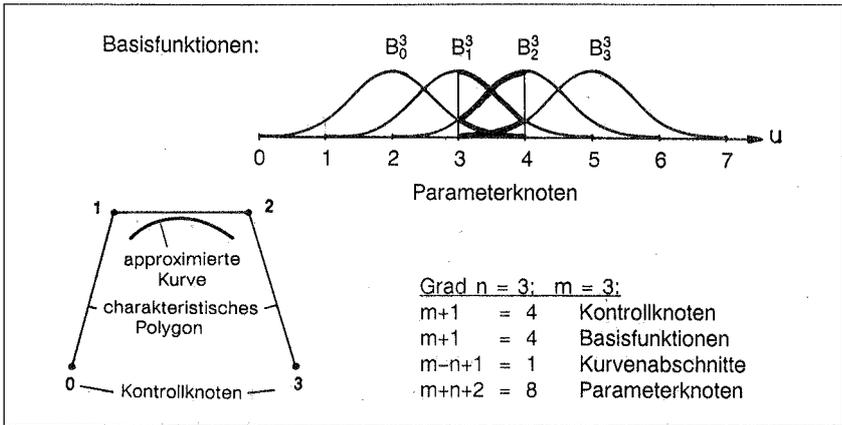


Bild 4.11: Kubische B-Spline-Kurve mit 4 Kontrollknoten

Die Abschnittspolynome des  $i$ -ten kubischen B-Spline sind folgendermaßen definiert (Bild 4.10):

$$B_i^3 = \left\{ \begin{array}{l} \frac{u-u_j}{u_{i+3}-u_j} \cdot \frac{u-u_j}{u_{i+2}-u_j} \cdot \frac{u-u_j}{u_{i+1}-u_j} \quad ; \quad u_i \leq u \leq u_{j+1} \\ \frac{u-u_j}{u_{i+3}-u_j} \cdot \left( \frac{u-u_j}{u_{i+2}-u_j} \cdot \frac{u_{j+2}-u}{u_{j+2}-u_{j+1}} + \frac{u_{j+3}-u}{u_{i+3}-u_{j+1}} \cdot \frac{u-u_{j+1}}{u_{j+2}-u_{j+1}} \right) \\ + \frac{u_{j+4}-u}{u_{j+4}-u_{j+1}} \cdot \frac{u-u_{j+1}}{u_{i+3}-u_{j+1}} \cdot \frac{u-u_{j+1}}{u_{j+2}-u_{j+1}} \quad ; \quad u_{j+1} \leq u \leq u_{j+2} \\ \frac{u-u_j}{u_{i+3}-u_j} \cdot \frac{u_{j+3}-u}{u_{i+3}-u_{j+1}} \cdot \frac{u_{j+3}-u}{u_{i+3}-u_{j+2}} \\ + \frac{u_{j+4}-u}{u_{j+4}-u_{j+1}} \cdot \left( \frac{u-u_{j+1}}{u_{i+3}-u_{j+1}} \cdot \frac{u_{j+3}-u}{u_{i+3}-u_{j+2}} + \frac{u_{j+4}-u}{u_{j+4}-u_{j+2}} \cdot \frac{u-u_{j+2}}{u_{i+3}-u_{j+2}} \right) \\ \quad ; \quad u_{j+2} \leq u \leq u_{j+3} \\ \frac{u_{j+4}-u}{u_{j+4}-u_{j+1}} \cdot \frac{u_{j+4}-u}{u_{j+4}-u_{j+2}} \cdot \frac{u_{j+4}-u}{u_{j+4}-u_{j+3}} \quad ; \quad u_{j+3} \leq u \leq u_{j+4} \end{array} \right. \quad (4.10)$$

Für konstante Abstände  $\Delta$  zwischen den Parameterknoten läßt sich (4.10) vereinfachen [6], wobei  $0 \leq \bar{u} \leq 1$  ist:

$$B_i^3 = \begin{cases} \frac{1}{6} \bar{u}^3 & ; \bar{u} = \frac{1}{\Delta} (u - u_i) , u_i \leq u \leq u_{i+1} \\ \frac{1}{6} (1 + 3 \bar{u} + 3 \bar{u}^2 - 3 \bar{u}^3) & ; \bar{u} = \frac{1}{\Delta} (u - u_{i+1}) , u_{i+1} \leq u \leq u_{i+2} \\ \frac{1}{6} (4 - 6 \bar{u}^2 + 3 \bar{u}^3) & ; \bar{u} = \frac{1}{\Delta} (u - u_{i+2}) , u_{i+2} \leq u \leq u_{i+3} \\ \frac{1}{6} (1 - \bar{u})^3 & ; \bar{u} = \frac{1}{\Delta} (u - u_{i+3}) , u_{i+3} \leq u \leq u_{i+4} \end{cases} \quad (4.11)$$

Einen rekursiven Algorithmus zur Bestimmung von  $B_i^n$  gibt de Boor in [18] an, der als effizient und stabil bekannt ist. Der B-Spline von Grad  $n$  wird additiv durch Basisfunktionen niedrigeren Grades bestimmt:

$$B_i^r(u) = (u - u_i) \frac{B_i^{r-1}(u)}{u_{i+r} - u_i} + (u_{i+r+1} - u) \frac{B_{i+1}^{r-1}(u)}{u_{r+i+1} - u_{i+1}}$$

mit: 
$$B_i^0 = \begin{cases} 1 ; u \in [u_i, u_{i+1}) \\ 0 ; u \notin [u_i, u_{i+1}) \end{cases} \quad (4.12)$$

Die Ableitungen eines B-Spline nach  $u$  lassen sich aus (4.10) bzw. (4.11) leicht bestimmen, hingegen sind für rekursiv definierte Funktionen besondere Algorithmen notwendig, die die Kettenregel der Differentiation berücksichtigen. De Boor [18] und Bartels et al. [6] geben dazu Auskunft. Böhm et al. [15] geben einen nach eigener Aussage besonders effektiven Algorithmus an. Die Kenntnis der ersten Ableitung ist wichtig zur Bestimmung der Tangentenrichtungen (z.B. für tangentielle Lagerbedingungen) oder zur Bestimmung der Normalenrichtung auf Coons-Elementen.

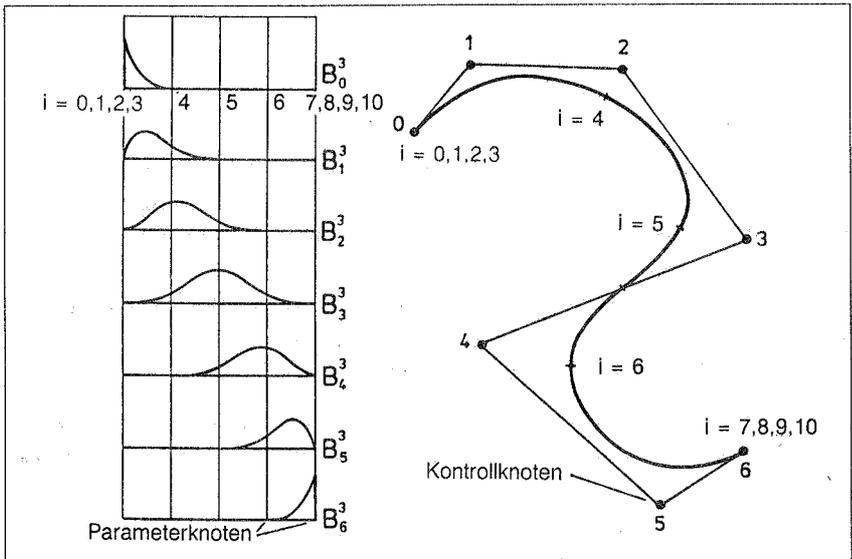
Fallen mehrere Parameterknoten aufeinander, verringern sich die Kontinuitätseigenschaften der generierten Kurve an der entsprechenden Stelle. Für kubische B-Splines gilt:

einfacher Parameterknoten:	$C_2$ -Kontinuität
doppelter Parameterknoten:	$C_1$
dreifacher Parameterknoten:	$C_0$
vierfacher Parameterknoten:	diskontinuierlicher Funktionsverlauf

Fallen die ersten bzw. letzten vier Parameterknoten eines B-Splines aufeinander, dann interpoliert dieser Spline den ersten und letzten Kontrollknoten, wie das auch bei den Bézier-Splines der Fall ist. Bild 4.12 zeigt die Verläufe der so entstehenden Basisfunktionen. Der zweite bzw. vorletzte Kontrollknoten bestimmt dann, äquivalent zum Bézier-Spline, die Tangentenrichtung der approximierten Kurve am Anfang bzw. Ende.

Bézier-Splines lassen sich aus kubischen B-Splines konstruieren. Es ist zu bemerken, daß die Kontrollknoten der Bézier- und B-Splines gewöhnlich nicht aufeinanderfallen. Nach Böhm et al. [15] werden sie respektive Bézier- bzw. de Boor-Knoten genannt. Diese Unterscheidung ist insbesondere bei der Betrachtung der Kontinuitätsbedingungen aneinanderstoßender B-Splines von Bedeutung. Im erwähnten Fall von vierfachen Parameterknoten am Beginn bzw. Ende eines B-Splines fallen die ersten bzw. die letzten Bézier- und de Boor-Knoten zusammen.

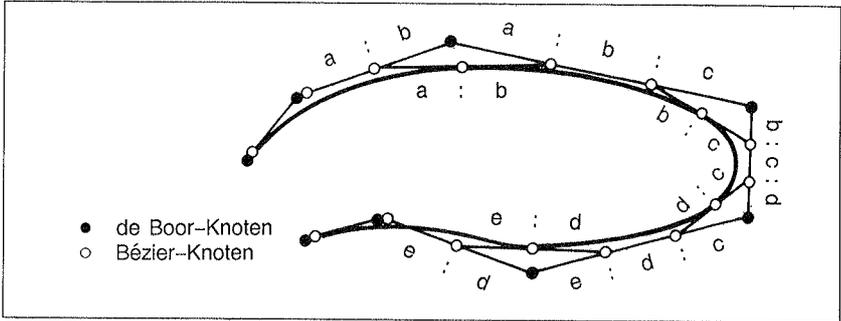
Im Bild 4.13 ist eine solche Konstruktion dargestellt. Da eine kubische B-Spline-Kurve per definitionem  $C_2$ -kontinuierlich ist, liefert diese Konstruktion auch die Bedingungen, die die Kontrollpunkte der Bézier-Kurven für einen kontinuierlichen Übergang der Krümmungen erfüllen müssen. Umgekehrt lassen sich so aus Bézier-Kurven und Übergangsbedingungen B-Spline-Kurven erstellen. Eine klare Darstellung dieses Sachverhalts findet man in [15].



**Bild 4.12:** Mehrere Endknoten

Eine besondere Eigenschaft des B-Splines, die auch für die Formoptimierung von Bedeutung ist, ist die lokale Kontrolleigenschaft der Kontrollknoten. Das heißt, wird der Ort eines Kontrollknotens geändert (von Hand oder vom Optimierungsalgorithmus), sind davon nur vier Abschnitte der generierten Kurve betroffen. Der Rest bleibt unverändert. Dagegen werden den Bézier-Splines globale Eigenschaften zugesprochen, denn die Veränderung eines Kontrollknotens wirkt sich auf die gesamte Kurve aus. Tatsächlich jedoch

wird hier mit zweierlei Maß gemessen, denn wie aus Bild 4.13 ersichtlich, entsprechen Bézier-Splines den Abschnittspolynomen der B-Spline-Kurven. Das bedeutet: erfüllen die Kontrollknoten der Bézier-Splines die Übergangsbedingungen, betreffen auch dann Kurvenveränderungen nur vier Abschnitte der generierten Kurve.



**Bild 4.13:** 5 Bézier-Splines aus einem B-Spline (nach [15])

Weitere Formfreiheiten ergeben sich, wenn die in Bild 4.13 dargestellten Abschnittsverhältnisse des charakteristischen Polynoms intern als zusätzliche Parameter berücksichtigt werden. Solche Splines werden dann als  $\beta$ -Splines bezeichnet [5]. Das übliche Vorgehen ist jedoch, die Abschnittsverhältnisse konstant gleich groß zu belassen, wie das auch in CARAT-Design der Fall ist. Eine andere Weiterentwicklung stellen die "Non-Uniform-Rational-B-Splines" (kurz NURBS) dar. Damit können auch Kegelschnitte (Kreise, Ellipsen, Parabeln, Hyperbeln) exakt generiert werden. Eine kurze Einführung dazu findet man in [149].

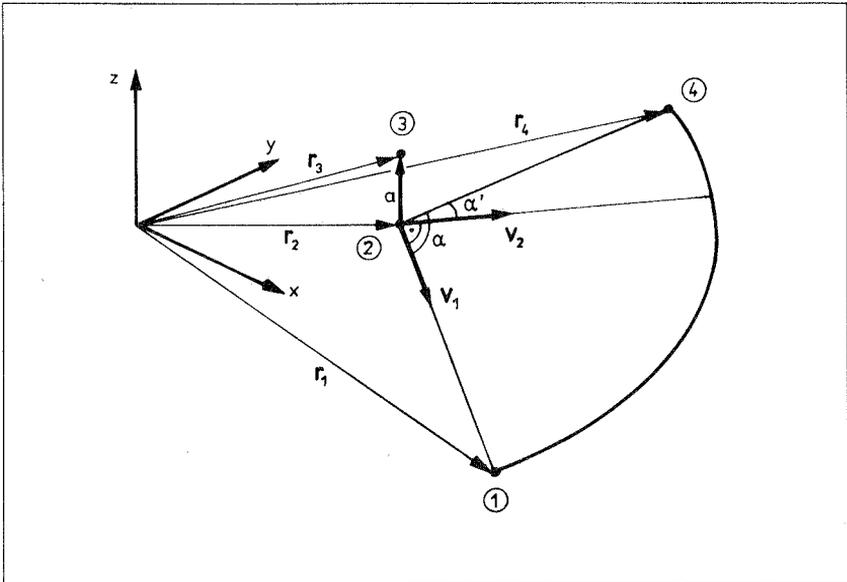
Zusammengefaßt haben B-Splines folgende Eigenschaften, die sie für die Anwendung in der Formoptimierung besonders vorteilhaft erscheinen lassen:

- beliebig viele Kontrollknoten bei konstantem Grad des Polynoms,
- $C_2$ -Kontinuität (kubische B-Splines),
- lokale Kontrolle,
- Kontrollknoten bilden eine konvexe Hülle (siehe Bézier-Spline),
- die Basisfunktionen sind stets größer oder gleich null, d.h. die approximierten Kurven oszillieren nicht.

#### 4.3.2.2.4 Kreise und Spirallinien

In einem kartesischen Koordinatensystem werden ebene Kreis- und Spirallinien durch vier Kontrollknoten beschrieben (Bild 4.14):

- Knoten 1: Anfang des Linienabschnittes ( $u = 0$ )
- Knoten 2: Mittelpunkt
- Knoten 3: Richtung der Drehachse
- Knoten 4: Ende des Linienabschnittes ( $u = 1$ )



**Bild 4.14:** Definition einer Spirallinie im kartesischen Raum

Aus den Kontrollknoten werden diejenigen Größen berechnet, mit denen die Kreis- bzw. Spirallinie eindeutig bestimmt werden kann (Bilder 4.14, 4.15):

Radien am Anfang und Ende:

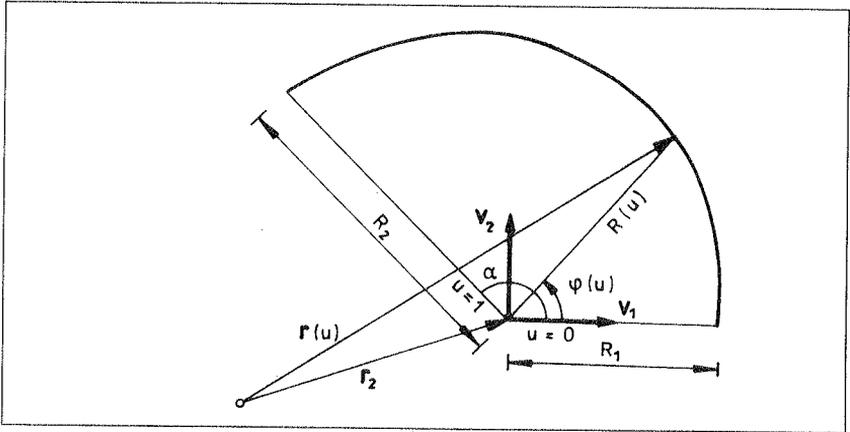
$$R_1 = |\mathbf{r}_1 - \mathbf{r}_2| \quad R_2 = |\mathbf{r}_4 - \mathbf{r}_2|$$

Drehachse:  $\mathbf{a} = \frac{\mathbf{r}_3 - \mathbf{r}_2}{|\mathbf{r}_3 - \mathbf{r}_2|}$

Basisvektoren:  $\mathbf{v}_1 = \frac{\mathbf{r}_1 - \mathbf{r}_2}{R_1} \quad \mathbf{v}_2 = \mathbf{a} \times \mathbf{v}_1$

Öffnungswinkel:  $\cos \alpha = \frac{1}{R_2} v_1 (r_4 - r_2) \quad \sin \alpha' = \frac{1}{R_2} v_2 (r_4 - r_2)$

$$\alpha = \begin{cases} \arccos(\cos \alpha) ; & \sin \alpha' > 0 \\ 2\pi - \arccos(\cos \alpha) ; & \sin \alpha' \leq 0 \end{cases}$$



**Bild 4.15:** *Interpolation einer Spirallinie*

Damit gilt in Abweichung von Gleichung (4.4) folgende Interpolationsvorschrift:

$$\mathbf{r}(u) = R(u) (\cos \phi(u) \mathbf{v}_1 + \sin \phi(u) \mathbf{v}_2) + \mathbf{r}_2 \quad (4.13)$$

$$R(u) = R_1 + (R_2 - R_1) u$$

$$\phi(u) = \alpha u$$

Formvariable im Sinne der Strukturoptimierung können die Koordinaten der Kontrollknoten sein, wobei beachtet werden muß, daß die Kreise und Spiralen in einer Ebene liegen müssen.

### 4.3.2.3 2-D-Elemente

#### 4.3.2.3.1 Lagrange-Elemente

Zur Formbeschreibung ebener und gekrümmter Flächen im Raum stehen verschiedene Interpolationsfunktionen, wie sie auch bei den isoparametrischen Elementen der finiten Element-Methode verwendet werden, zur Verfügung (siehe Bild 4.16). Die allgemeine Interpolationsvorschrift lautet hier:

$$\mathbf{r}(u, v, \mathbf{r}_i) = \sum_{i=1}^n \Phi_i(u, v) \mathbf{r}_i \quad (4.14)$$

mit:  $u, v \dots$  lokale Koordinaten  $0 \leq v \leq 1$   
 $n \dots$  Anzahl der Kontrollknoten eines Design-Elements  
 $\Phi_i \dots$  Formfunktionen  
 $\mathbf{r}_i \dots$  Koordinaten des Design-Knoten  $i; i = 1, \dots, n$

Die Formfunktionen dieser Elemente sind oft beschrieben, so daß an dieser Stelle auf eine nähere Angabe verzichtet wird. Es sei statt dessen auf die entsprechende Spezialliteratur verwiesen (z.B. [158], [7], [108], [109]). Netzgeneratoren, die auf Lagrange-Elementen beruhen, sind unter anderen in [24], [69] und [156] angegeben.

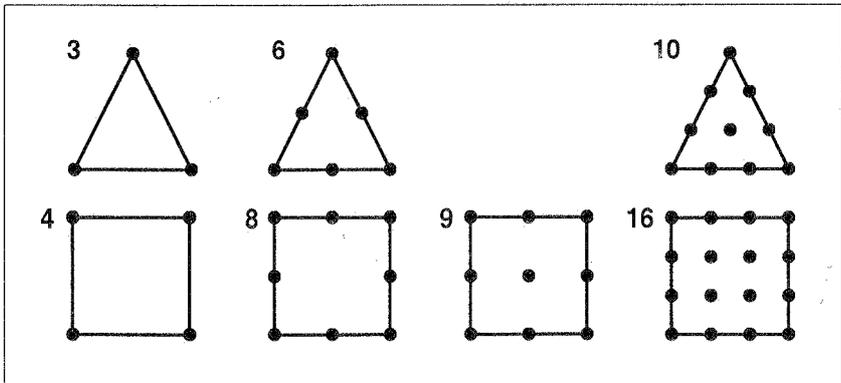
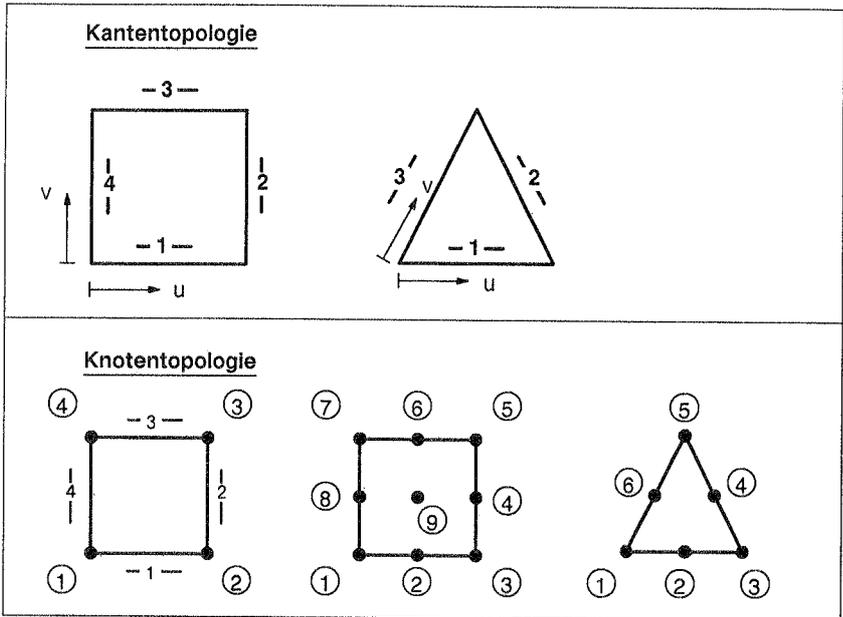


Bild 4.16: 2-D Lagrange Elemente

Gemäß der hier gewählten hierarchischen Struktur, sind die 2D-Lagrange-Elemente durch Kanten beschrieben. Mit der Vorgabe dieser Kantentopologie ist das lokale Koordinatensystem  $u, v$  festgelegt, und es kann durch besondere Auswertalgorithmen auch die Knotentopologie bestimmt werden, die zur Auswertung von (4.14) bekannt sein muß (Bild 4.17). Die Kanten müssen zu den Elementen passen. Ein neunknotiges Lagrange-

Element kann demnach nur von vier 3-knotigen Lagrange-Kanten beschrieben werden. Der zusätzliche Innenknoten wird von einem Fangalgorithmus automatisch zugeordnet. Dies gilt auch für die 10- und 16-knotigen kubischen Elemente.



**Bild 4.17:** Kanten- und Knotentopologie der 2-D Design-Elemente

Im allgemeinen eignen sich die 2D-Lagrange-Elemente besonders zur Darstellung von ebenen Strukturen. Übergangsbedingungen, die eine  $C_1$ -kontinuierliche Form mehrerer zusammengesetzter Elemente im Raum garantieren, sind selbst für das bikubische 16-knotige Element schwer einzuhalten. Deshalb werden für diesen Zweck, vor allem auch in der Formoptimierung, Bézier- oder B-Spline-Elemente vorgezogen. Andererseits sind Coons-Elemente, die eine beliebige Kombination verschiedener Kanten in einem Element erlauben, wesentlich flexibler bei der Formoptimierung ebener Strukturen, weshalb die Bedeutung der Lagrange-Elemente für die Formoptimierung insgesamt sehr gering ist.

#### 4.3.2.3.2 Bézier-Element

Der kubische Bézier-Spline (4.7) kann sehr einfach für zwei Dimensionen erweitert werden:

$$\mathbf{r}(u, v) = \sum_{i=0}^3 \sum_{j=0}^3 \mathbf{r}_{ij} \frac{3!}{(3-i)!i!} \frac{3!}{(3-j)!j!} u^i (1-u)^{3-i} v^j (1-v)^{3-j} \quad (4.15)$$

Dies ist die Formfunktion einer bikubischen Fläche. Deren 16 Kontrollknoten  $\mathbf{r}_{ij}$  formen das charakteristische Polyeder (Bild 4.18), dabei liegen nur die vier Eckknoten auf der approximierten Fläche. Das charakteristische Polyeder hat die angenehme Eigenschaft der konvexen Hülle, womit, wie bei den Bézier- und B-Splines, die Lage der generierten Form abgeschätzt werden kann. Dies ist für die Anwendung in der Formoptimierung sehr vorteilhaft. Die eigentlichen Vorzüge des Bézier-Elements gegenüber der alternativen Formulierung von Ferguson sind die anschaulichen Beschreibungen der Tangentialbedingungen an den Kanten durch die Lage der Kontrollknoten, was erst eine anwendergerechte Kontrolle der Form möglich macht. Neben den Tangentenvektoren in  $u$ - und  $v$ -Richtung sind noch die Drillvektoren an den Ecken des Elementes von Bedeutung.

Sie beschreiben die gemischten Ableitungen  $\frac{\partial^2 \mathbf{r}}{\partial u \partial v}$  der Fläche und sorgen für die "Völligkeit" der generierten Form ("Bubble-Term"). Mit den 16 Kontrollknoten können an jeder Ecke des Bézier-Elementes folgende Vektorgrößen angegeben werden (hier am Beispiel der Ecke  $u = v = 0$ ):

$$\begin{aligned} \text{Ortsvektor:} \quad & \mathbf{r}(0, 0) = \mathbf{r}_{00} \\ \text{Tangentenvektoren:} \quad & \mathbf{r}_{,u}(0, 0) = 3 (\mathbf{r}_{10} - \mathbf{r}_{00}) \\ & \mathbf{r}_{,v}(0, 0) = 3 (\mathbf{r}_{01} - \mathbf{r}_{00}) \\ \text{Drillvektor:} \quad & \mathbf{r}_{,uv}(0, 0) = 9 (\mathbf{r}_{00} - \mathbf{r}_{01} - \mathbf{r}_{10} + \mathbf{r}_{11}) \end{aligned} \quad (4.16)$$

Die Definition Béziers, die er 1972 für Renault entwickelte (nach [40]), hat sich für den Entwurf und die interaktive Kontrolle von Flächen gegenüber der Beschreibung von Ferguson durchgesetzt. Eine häufige Vereinfachung bei der Verwendung von Ferguson-Elementen ist, daß der Drillvektor null gesetzt wird, was jedoch zu abgeflachten Formen führt. Gegenüber Lagrange-Elementen hat das Bézier-Element den Vorteil, daß mit der Kenntnis der Tangenten- und Drillvektoren sehr einfach Übergangsbedingungen für eine  $C_1$ -kontinuierliche Form aus mehreren Elementen formuliert und kontrolliert werden können. (Siehe dazu Abschnitt 4.3.3.2). Für eine Erweiterung von Gleichung (4.15) für beliebige Polynomgrade siehe [15] oder [6]. Eine ausführliche Darstellung dreieckiger Bézier-Elemente gibt Farin in [38]. Zur Definition von B-Spline-Elementen siehe ebenfalls [15] und [6] sowie [5].

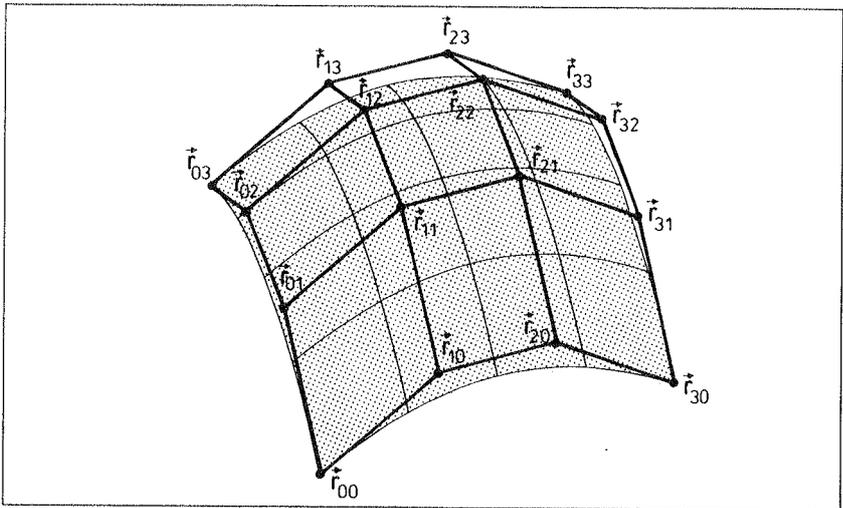


Bild 4.18: Bézier-Element mit charakteristischem Polyeder

#### 4.3.2.3.3 Coons-Elemente

Die Elemente dieser Klasse sind durch die Angabe der Randkurven, das sind die Kanten, eindeutig definiert. Damit können verschiedenartige Kanten zu einem Flächenelement kombiniert werden, im Gegensatz zu Lagrange- und Bézier-Elementen. Diese Methode wurde von Coons entwickelt (siehe in [15] und [40]) und wird auch "transfinite Interpolation" genannt; dies deshalb, weil eine unbeschränkte Zahl von Punkten, d.h. die Kanten, von dieser Methode interpoliert wird. Die Formfunktion des bilinearen überblendeten Coons-Elements lautet (siehe auch Bild 4.19):

$$\begin{aligned}
 \mathbf{r}(u, v) = & \left[ L_0^1(u) \ L_1^1(u) \right] \begin{Bmatrix} \mathbf{r}_4(0, v) \\ \mathbf{r}_2(1, v) \end{Bmatrix} + \left[ \mathbf{r}_1(u, 0) \ \mathbf{r}_3(u, 1) \right] \begin{Bmatrix} L_0^1(v) \\ L_1^1(v) \end{Bmatrix} \\
 & - \left[ L_0^1(u) \ L_1^1(u) \right] \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_4 \\ \mathbf{r}_2 & \mathbf{r}_3 \end{bmatrix} \begin{Bmatrix} L_0^1(v) \\ L_1^1(v) \end{Bmatrix}
 \end{aligned} \tag{4.17}$$

mit den linearen Blendfunktionen:

$$\begin{aligned}
 L_0^1(u) &= 1 - u & L_1^1(u) &= u \\
 L_0^1(v) &= 1 - v & L_1^1(v) &= v
 \end{aligned}$$

und den Randfunktionen  $\mathbf{r}_1(u, 0)$ ,  $\mathbf{r}_2(1, v)$ ,  $\mathbf{r}_3(u, 1)$ ,  $\mathbf{r}_4(0, v)$  der vier Kanten.

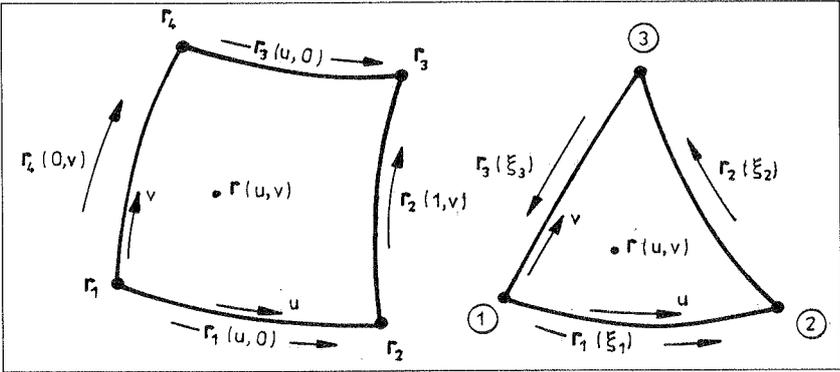


Bild 4.19: Definitionen im vier- und dreiseitigen Coons-Element

Nach Gordon kann (4.17) auch als eine Boolesche Summe von lineären Operatoren gesehen werden (siehe dazu [15]). Mit der Wahl linearer Blendfunktionen sind Coons-Elemente  $C_0$ -kontinuierlich. Sie eignen sich deshalb vor allem für ebene Probleme. Da alle verfügbaren Kantenlelemente bei diesem Ansatz kombiniert werden können, eröffnen sich viele Möglichkeiten für die Formoptimierung ebener Probleme [23], [37]. Werden kubische Polynome als Blendfunktionen gewählt, können auch  $C_1$ -kontinuierliche Flächen dargestellt werden. Sind gleichzeitig alle vier Kanten ebenfalls kubische Polynome, läßt sich zeigen, daß die Interpolationsvorschrift derjenigen des bikubischen Ferguson-Elementes entspricht (Faux und Pratt [40]). Sind alle Kanten quadratische Polynome und die Blendfunktionen linear, entspricht (4.17) den Formfunktionen des 8-knotigen Serendipity-Elements, wie es in der Methode der finiten Elemente bekannt ist.

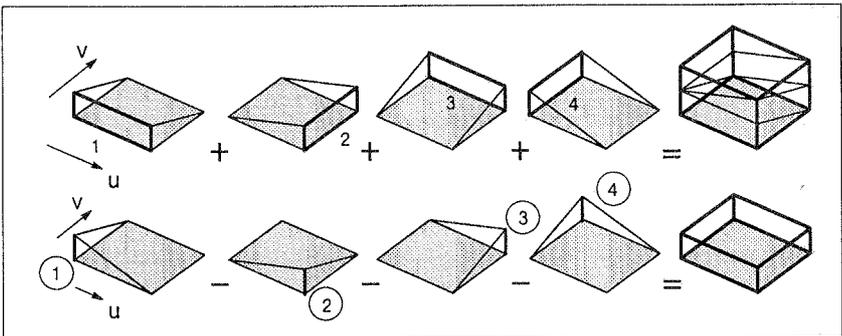
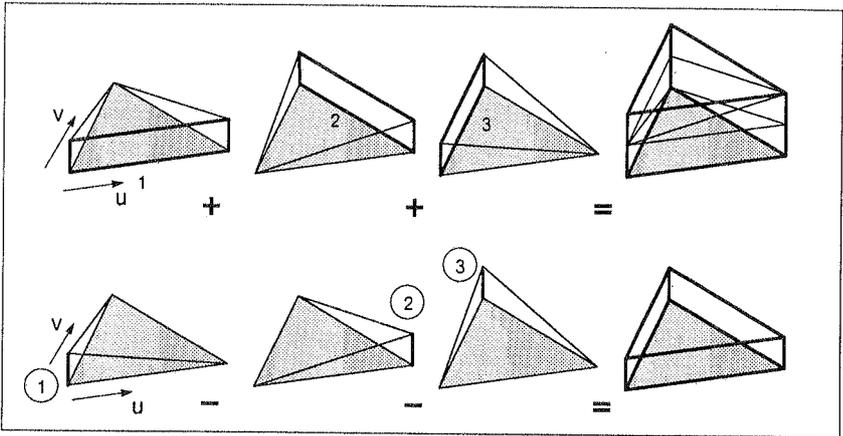


Bild 4.20a: Coons-Interpolation von Rechtecken



**Bild 4.20b:** Coons-Interpolation von Dreiecken

Das Vorgehen der Interpolationsvorschrift (4.17) läßt sich graphisch veranschaulichen (Bild 4.20a). Zunächst werden alle Kanten auf die jeweils gegenüberliegenden abgeblendet. Diese Teilformfunktionen werden aufaddiert. Man erkennt, daß die resultierende Formfunktion insgesamt zu groß ausfällt, und es werden die abgeblendeten Funktionswerte an den Ecken des Elementes wieder abgezogen. Voraussetzung ist dabei, daß die Kanten kompatibel sind, das heißt, daß sie sich an den Ecken des Coons-Elementes berühren. Entsprechend läßt sich auch ein lineares Blendschema für dreieckige Coons-Elemente entwerfen (Bild 4.20b). Die zugehörige Formfunktion lautet:

$$\begin{aligned} r(u, v) = & (1 - v) r_1(\xi_1) + (u + v) r_2(\xi_2) + (1 - u) r_3(\xi_3) \\ & - (1 - u - v) r_1(0) - u r_2(0) - v r_3(0) \end{aligned} \quad (4.18)$$

mit den zusätzlich zu definierenden Laufparametern  $\xi_i$  entlang den Kanten:

$$\xi_1 = \frac{u}{1 - v} \quad \xi_2 = \frac{v}{u + v} \quad \xi_3 = \frac{1 - u - v}{1 - u} \quad (4.19)$$

An den Polstellen  $v = 1$ ,  $u = 1$ ,  $u + v = 0$  sind die  $\xi_i$  zwar nicht definiert, jedoch sind an diesen Stellen die zu interpolierenden Kanten bereits auf Null abgeblendet (Gl. (4.18)), so daß definiert werden kann:

$$\xi_1(v = 1) = \xi_2(u + v = 0) = \xi_3(u = 1) = 0 \quad (4.20)$$

Ein ähnliches Interpolationsschema geht auf Barnhill zurück und wird in [15] und [58] geschildert. Im letzteren Beitrag wird besonders auf die Anwendung des Verfahrens zur Generierung von FE-Netzen eingegangen.

#### 4.3.2.4 3-D-Elemente

Da der Schwerpunkt dieser Arbeit auf der Formbeschreibung und -optimierung von Flächentragwerken liegt, sollen die in CARAT vorhandenen 3-D-Design-Elemente nur kurz erwähnt, jedoch nicht im Detail erläutert werden. Es sind das 8- und 20-knotige Lagrange-Element sowie ein Coons-Element mit linearen Blendfunktionen implementiert (siehe Anhang A3). Das letztere gestattet die freie Kombination aller vorhandenen Kanten und Flächenbeschreibungen zu einem 6-seitigen Volumenmodell. Eine mögliche Anwendung ist in Bild 4.3 dargestellt.

#### 4.3.3 Geometrische Kontinuität

##### 4.3.3.1 1-D-Elemente, Tangentenelemente

Der Begriff "geometrische Kontinuität" oder auch "visuelle Kontinuität" [68], [152] ist in Abgrenzung zur "parametrischen Kontinuität" zu sehen. Eine generierte Form aus mehreren, zusammengesetzten Design-Elementen ist geometrisch kontinuierlich ( $G_1$ -kontinuierlich), wenn die generierte (sichtbare) Form an den Nahtstellen der Design-Elemente die entsprechenden Übergangsbedingungen erfüllt. Dagegen kann sich an diesen Stellen die Parametrisierung der einzelnen Design-Elemente ändern, was bedeutet, daß die Form nicht parametrisch kontinuierlich ( $C_1$ -kontinuierlich) ist. Die Bedingungen für geometrische Kontinuität orientieren sich also am sichtbaren Produkt, die für parametrische Kontinuität an den Verhältnissen im Parameterraum.

Für eindimensionale Design-Elemente lautet die Bedingung für  $G_1$ -Kontinuität zwischen zwei Kurven  $r_1(u)$  und  $r_2(u)$  (vergleiche Bild 4.21):

$$\beta_1 \dot{r}_1(1) = \dot{r}_2(0) \quad (4.21)$$

wobei  $(\dot{\phantom{x}})$  die Differentiation nach dem Parameter  $u$  darstellt.

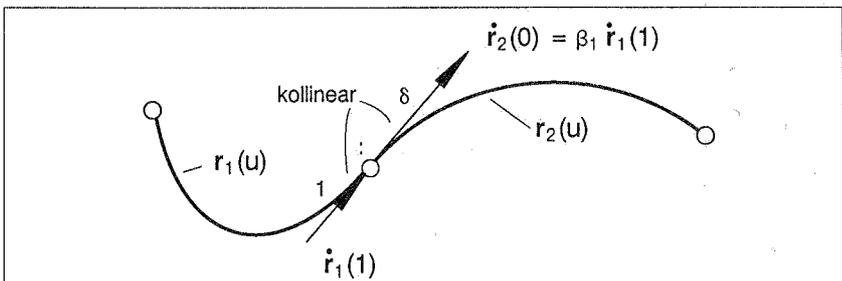


Bild 4.21: geometrische Kontinuität ( $G_1$ -Kontinuität)

Gleichung (4.21) bedeutet, daß die Tangentenvektoren der beiden Kurven zwar kollinear, jedoch unterschiedlich lang sein können. Für  $\beta_1 = 1$  ergibt sich parametrische  $C_1$ -Kontinuität. Mit:

$$\mathbf{k}_i(\mathbf{u}) = \frac{(\dot{\mathbf{r}}_i(\mathbf{u}) \times \ddot{\mathbf{r}}_i(\mathbf{u})) \times \dot{\mathbf{r}}_i(\mathbf{u})}{|\dot{\mathbf{r}}_i(\mathbf{u})|^4} \quad (4.22)$$

für den Krümmungsvektor entlang einer Kurve  $\mathbf{r}_i(\mathbf{u})$  und (4.21) gilt für die geometrische Krümmungskontinuität am Übergang von der Kurve 1 zur Kurve 2:

$$\frac{(\beta_1 \dot{\mathbf{r}}_1(1) \times \ddot{\mathbf{r}}_2(0)) \times \beta_1 \dot{\mathbf{r}}_1(1)}{|\beta_1 \dot{\mathbf{r}}_1(1)|^4} = \frac{(\dot{\mathbf{r}}_1(1) \times \ddot{\mathbf{r}}_1(1)) \times \dot{\mathbf{r}}_1(1)}{|\dot{\mathbf{r}}_1(1)|^4} \quad (4.23)$$

Aus einem Vergleich der beiden Seiten von (4.23) läßt sich die Übergangsbedingung für den Krümmungsvektor ableiten (da  $\dot{\mathbf{r}}_1 \times \dot{\mathbf{r}}_1 = 0$ ):

$$\ddot{\mathbf{r}}_2(0) = \beta_1^2 \ddot{\mathbf{r}}_1(1) + \beta_2 \dot{\mathbf{r}}_1(1) \quad (4.24)$$

Parametrische  $C_2$ -Kontinuität ergibt sich für  $\beta_1 = 1$  und  $\beta_2 = 0$ . Eine gute Darstellung der Kontinuitätsbedingungen findet man bei Barsky [5], aber auch bei Bartels et al. [6] oder Faux und Pratt [40]. Barsky verallgemeinert (4.24) für die Definition der B-Splines und leitet daraus die sogenannten "β-Splines" ab.

Für den Sonderfall der Bézier-Splines lauten die Gleichungen (4.21) bzw. (4.24) (vergleiche Bild 4.22):

$$\beta_1 3 (\mathbf{r}_0 - \mathbf{r}_{-1}) = 3 (\mathbf{r}_1 - \mathbf{r}_0) \quad (4.21)'$$

$$\begin{aligned} 6 (\mathbf{r}_0 - 2 \mathbf{r}_1 + \mathbf{r}_2) &= \beta_1^2 6 (\mathbf{r}_{-2} - 2 \mathbf{r}_{-1} + \mathbf{r}_0) \\ &+ \beta_2 3 (\mathbf{r}_0 - \mathbf{r}_{-1}) \end{aligned} \quad (4.24)'$$

Böhm et al. [15] geben für die  $G_2$ -Kontinuität gekoppelter Bézier-Splines eine einfache Beziehung an, die leicht geometrisch zu konstruieren ist. Sie bringen dabei die Kontrollknoten  $\mathbf{r}_i$  (Bézier-Knoten) der beiden gekoppelten Bézier-Splines in Beziehung zu den Kontrollknoten  $\mathbf{d}_i$  (de Boor-Knoten) des infolge der Kopplung resultierenden B-Splines (vergleiche dazu Bild 4.22):

$$\delta \mathbf{d}_0 = \delta ((1 + \gamma \delta) \mathbf{r}_{-1} - \gamma \delta \mathbf{r}_{-2}) = (\gamma + \delta) \mathbf{r}_1 - \gamma \mathbf{r}_2 \quad (4.25)$$

mit: 
$$\beta_1 = \delta ; \quad \beta_2 = 2 \frac{\delta}{\gamma} (1 - \gamma) (1 + \delta) \quad (4.26)$$

Die Übergangsbedingung (4.25) entspricht der in Bild 4.13 dargestellten Konstruktion von Bézier-Splines aus einem B-Spline.

Die Forderung nach geometrischer Kontinuität auch während der Formoptimierung oder interaktiver Formmodifikationen läßt sich erfüllen, wenn die topologischen Beziehungen

(4.21)' und (4.25) zwischen den Kontrollknoten  $r_i$  eingehalten werden. In CARAT-Design wird die Aufgabe so gelöst, daß diese Bedingungen in besonderen "Tangentenelementen" zusammengefaßt werden. Diese werden auf Anforderung automatisch generiert und den Design-Elementen überlagert. Während aller Formänderungen bleiben so die Kontinuitätsbedingungen explizit erfüllt.

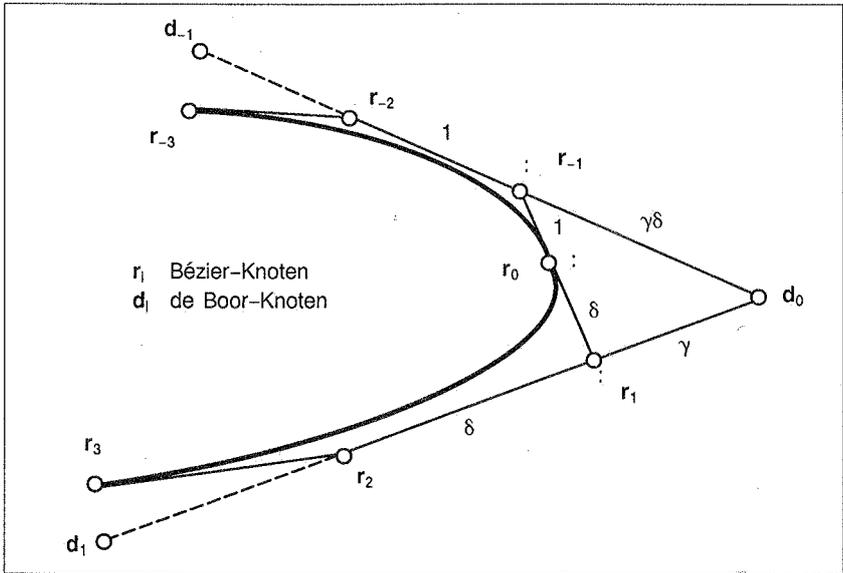


Bild 4.22:  $G_2$ -Kontinuität am Bézier-Spline

### $G_1$ -Tangentenelement:

Dieses Element faßt die Kontrollknoten  $r_{-1}, r_0$  und  $r_1$  (Bild 4.23) zusammen. Es kann in Verbindung mit Bézier- und B-Spline-Elementen eingesetzt werden. Gemäß (4.21) und (4.26) gilt folgender Zusammenhang:

$$r_1 = r_0 + \delta (r_0 - r_{-1}) \quad (4.27)$$

$\delta$  wird der "Tangentenfaktor" genannt. Gleichung (4.27) ist eine lineare Verknüpfungsregel. Bei der Formoptimierung wird mit jedem  $G_1$ -Tangentenelement die Zahl der Unbekannten um maximal 3 reduziert (Anzahl der Koordinaten von  $r_1$ ). Es verbleiben 7 mögliche Formvariablen: die Koordinaten von  $r_0$  und  $r_{-1}$  sowie der Tangentenfaktor  $\delta$ . Im Bild 4.24 sind die Formfreiheiten eines  $G_1$ -Tangentenelementes in der Formoptimierung dargestellt.

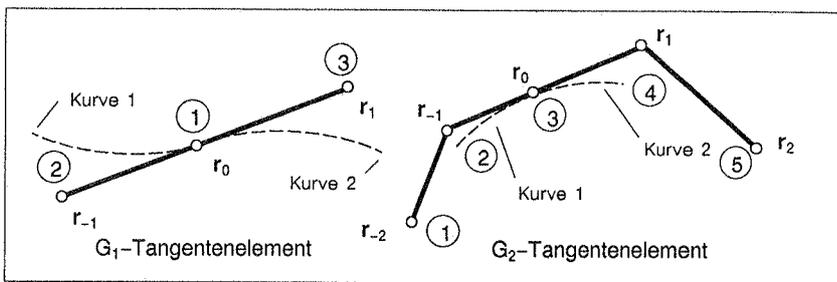


Bild 4.23: Tangentenelemente

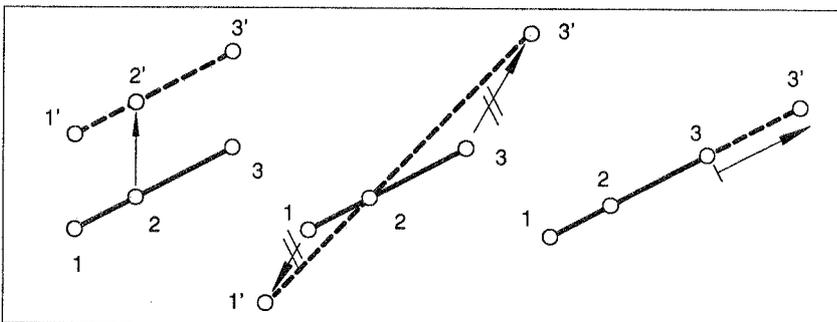


Bild 4.24: Formfreiheiten des  $G_1$ -Tangentenelementes

### $G_2$ -Tangentenelement:

Dieses Element faßt alle fünf an der  $G_2$ -Bedingung beteiligten Kontrollknoten zusammen (Bild 4.23). Vereinfachend wird in CARAT-Design  $\gamma = 1$  gesetzt (Gln. 4.25/4.26). Da von der  $G_2$ -Bedingung vier Bézier-Splines bzw. vier Abschnitte eines B-Splines betroffen sind, würden sich  $G_2$ -Bedingungen an beiden Enden eines Bézier-Elementes oder eines "kurzen" B-Spline-Elementes gegenseitig beeinflussen. Dies ist in CARAT-Design nicht vorgesehen, weshalb  $G_2$ -Tangentenelemente nur zusammen mit B-Spline-Elementen zugelassen sind, die mindestens 6 Kontrollknoten besitzen und die Anfangs- und Endknoten interpolieren (vierfache Parameterknoten). Damit ergeben sich an der Koppelstelle aus (4.21)' und (4.25) folgende Zusammenhänge für die Kontrollknoten  $r_1$  dieser speziellen B-Splines:

$$\begin{aligned} r_{-1} &= \frac{1}{(1+3\delta)(2+\delta)} [\delta(1+\delta)r_{-2} + 2(1+\delta)(2+\delta)r_0 - 2r_2] \\ r_1 &= \frac{1}{(1+3\delta)(2+\delta)} [-\delta^2(1+\delta)r_{-2} + (1+\delta)^2(2+\delta)r_0 + 2\delta r_2] \end{aligned} \quad (4.28)$$

Bei der Entwicklung dieser Gleichungen wurde bereits berücksichtigt, daß es sich bei den hier eingeführten Kontrollknoten  $r_1$  um die Boor-Knoten handelt; d.h.,  $r_{-2}$  und  $r_2$  der Gln. (4.28) entsprechen den Knoten  $d_{-1}$  und  $d_1$  des resultierenden B-Splines in Bild 4.22, das die Verhältnisse bei der  $G_2$ -kontinuierlichen Kopplung zweier Bézier-Splines illustriert.

In der Formoptimierung verbleiben insgesamt 10 unabhängige Formvariablen: die Koordinaten von  $r_{-2}$ ,  $r_0$  und  $r_2$  sowie der Tangentenfaktor  $\delta$ .

In Bild 4.25 ist eine mögliche Anwendung der Tangentenelemente für die Ausformung einer Bauteilkontur dargestellt.

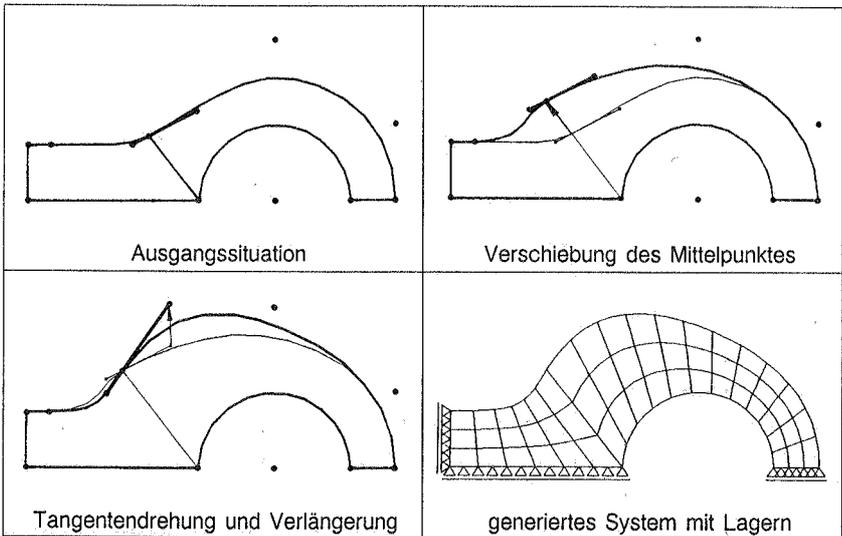


Bild 4.25:  $G_1$ -kontinuierliche Konturen mit Tangentenelementen

#### 4.3.3.2 2-D-Elemente, Kontinuitäts-elemente

Die Bedingung für  $G_1$ -Kontinuität entlang der gemeinsamen Kante zweier Design-Elemente (Bild 4.26) ist, daß die beiden Elemente an jedem Punkt der Kante eine gemeinsame Tangentenebene besitzen müssen. Das heißt, die Normalenvektoren auf beiden Elementen müssen entlang der gemeinsamen Kante kollinear sein. Für den im Bild 4.26 dargestellten Fall ergibt sich:

$$\mathbf{r}_{2,u}(0, v) \times \mathbf{r}_{2,v}(0, v) = \delta(v) \mathbf{r}_{1,u}(1, v) \times \mathbf{r}_{1,v}(1, v) \quad (4.29)$$

Weiter sollen beide Elemente vom gleichen Typ sein (z.B. Bézier-Elemente). Dann ist die gemeinsame Kante in beiden Elementen auch in der parametrischen Darstellung identisch. Und mit

$$\mathbf{r}_{2,v}(0, v) = \mathbf{r}_{1,v}(1, v) \quad (4.30)$$

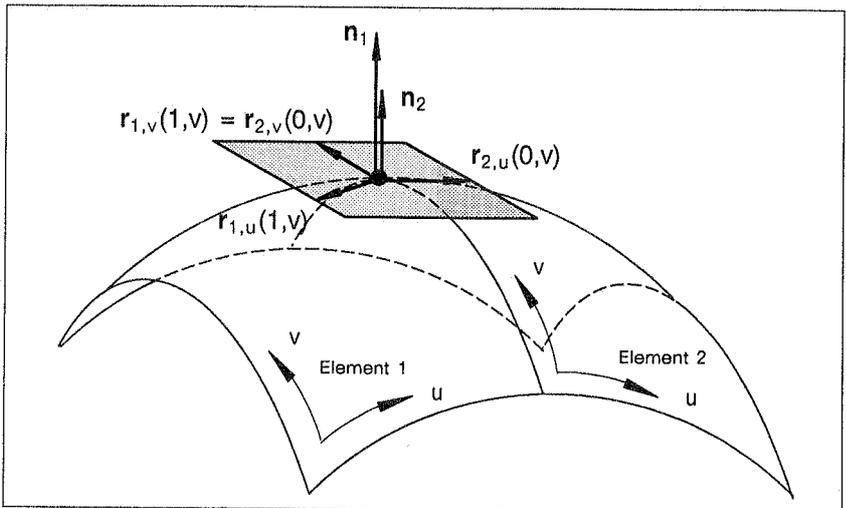
folgt aus (4.29) eine Bedingung für  $G_1$ -Kontinuität quer zur gemeinsamen Kante:

$$\mathbf{r}_{2,u}(0, v) = \delta(v) \mathbf{r}_{1,u}(1, v) \quad (4.31)$$

Da beide Elemente vom selben Typ sind, dürfen sich die Funktionen auf beiden Seiten von (4.31) höchstens durch einen konstanten Faktor unterscheiden. Es verbleibt damit:

$$\delta(v) = \delta = \text{konst.} \quad (4.32)$$

Für Kanten, die in  $u$  parametrisiert sind, folgt Entsprechendes.



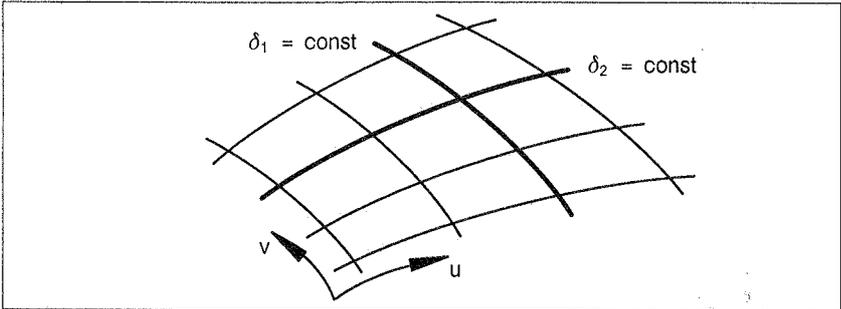
**Bild 4.26:**  $G_1$ -Kontinuität an Flächen

Folgende Definition wird damit eingeführt:

$$\delta(u) = \delta_1 = \text{konst.} \quad ; \quad \delta(v) = \delta_2 = \text{konst.} \quad (4.33)$$

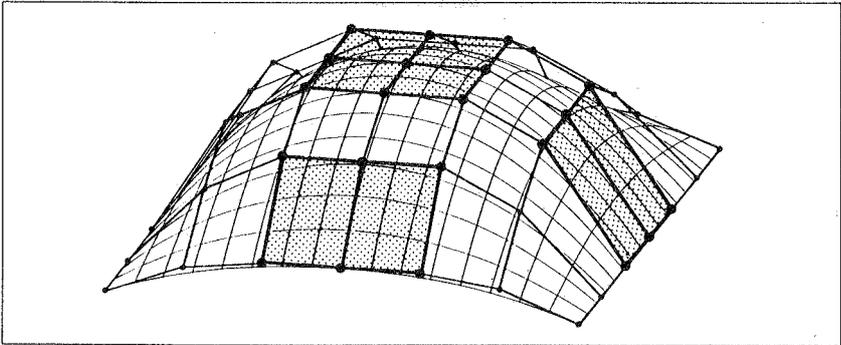
Die Gleichungen (4.33) haben weitere Konsequenzen. Schließen in  $v$ -Richtung an die beiden in Bild 4.26 dargestellten Design-Elemente weitere an, dann gilt auch für diese die Übergangsbedingung (4.31). Die Faktoren  $\delta_1$  und  $\delta_2$  sind demnach entlang einer Reihe aufeinander folgender Kanten in  $u$ - bzw.  $v$ -Richtung gültig und konstant (siehe Bild 4.27). Eine solche Reihe von Kanten wird im folgenden Linie genannt, die Faktoren

$\delta_1$  und  $\delta_2$  sind die Linienfaktoren. Sie wirken senkrecht zur Linie im Gegensatz zu den Tangentenfaktoren, die in Kantenrichtung wirken.



**Bild 4.27:** Design-Netz mit Linien konstanter  $\delta$ -Faktoren

Alternativ zu den dargestellten Übergangsbedingungen bringen Faux und Pratt [40] eine weitere Möglichkeit, die für jede Kante einen weiteren Faktor einführt und die Formfreiheit erhöht. Veron et al. [148] beschreiben Übergangsbedingungen für höhere Ableitungen. Jensen [73] beschreibt eine Technik zur kontinuierlichen Anbindung von rechteckigen und dreieckigen Design-Elementen.



**Bild 4.28:** Übergangsbedingungen an Bézier-Elementen

Für Bézier-Elemente lassen sich die Bedingungen (4.31) relativ einfach erfüllen und kontrollieren. Die entsprechenden Abschnitte der charakteristischen Polyeder aneinanderstoßender Bézier-Elemente müssen kollinear sein (Bild 4.28). In Erweiterung des Konzeptes der Tangentenelemente lassen sich somit Kontinuitätselemente definieren, die die notwendigen topologischen Beziehungen zwischen den Design-Knoten kontrollieren. Sie werden dem Netz der Design-Elemente überlagert. Im allgemeinsten Fall besteht ein Kontinuitätselement aus 9 Knoten, wenn vier Design-Elemente aneinander

grenzen. In Eckbereichen oder wenn entlang einer Kante keine Kontinuität herrschen soll, kann es auch aus vier, sechs oder acht Knoten bestehen (Bild 4.29).

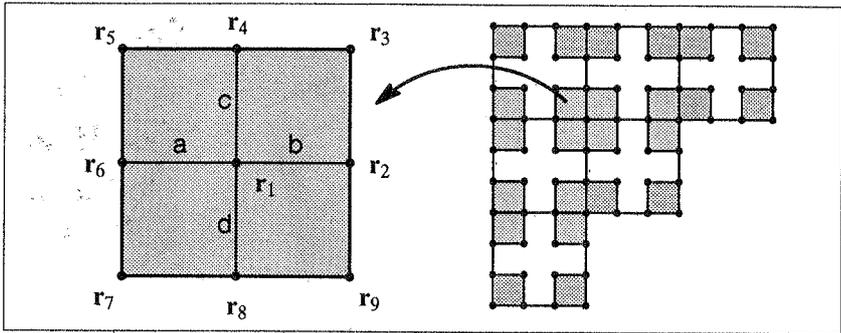


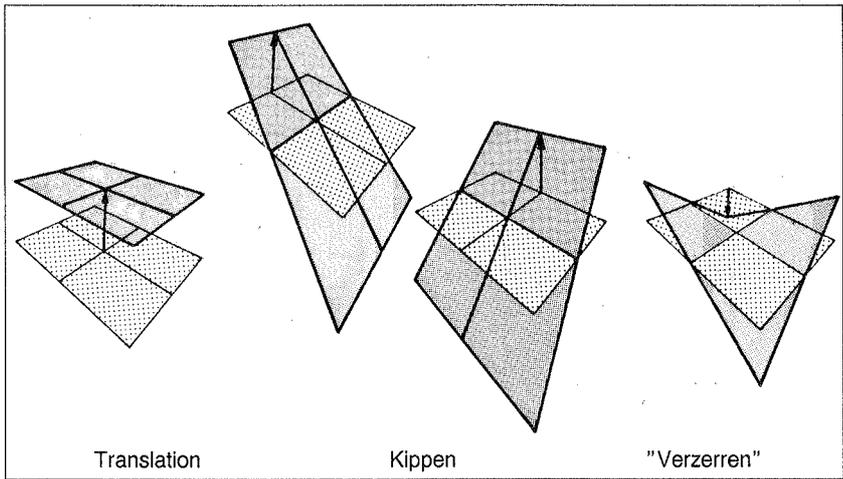
Bild 4.29: Kontinuitätselemente, Definitionen

Aus diesem Grund sind die Knoten eines Kontinuitätselementes spiralförmig durchnummeriert, denn somit kann in jedem Falle davon ausgegangen werden, daß die Knoten 1 bis 4 existieren. Die Seitenverhältnisse  $a : b$  und  $d : c$  werden von den Linienfaktoren bestimmt. Mit den ersten vier Knoten und den Linienfaktoren wird die Lage der übrigen Knoten festgelegt:

$$\begin{aligned}
 r_5 &= r_4 + \left(\frac{a}{b}\right) (r_4 - r_3) \\
 r_6 &= r_1 + \left(\frac{a}{b}\right) (r_1 - r_2) \\
 r_7 &= r_1 + \left(\frac{a}{b}\right) (r_1 - r_2) + \left(\frac{d}{c}\right) (r_1 - r_4) + \left(\frac{a}{b}\right) \left(\frac{d}{c}\right) (r_1 - r_2 + r_3 - r_4) \\
 r_8 &= r_1 + \left(\frac{d}{c}\right) (r_1 - r_4) \\
 r_9 &= r_2 + \left(\frac{d}{c}\right) (r_2 - r_3)
 \end{aligned} \tag{4.34}$$

mit:  $\frac{d}{c} = \delta_1$  und  $\frac{a}{b} = \delta_2$

Die Bedingungen (4.34) bleiben bei interaktiven Formänderungen oder während der Formoptimierung explizit erfüllt. Dies entspricht einer Verknüpfungsvorschrift im Sinne von Kapitel 5. In jedem Kontinuitätselement verbleiben 12 unbekannte Koordinaten von vier freien Knoten. Die Anzahl der Linienfaktoren entspricht der Zahl der Linien. Bild 4.30 macht die möglichen Formfreiheiten eines Kontinuitätselementes deutlich.



**Bild 4.30:** *Formfreiheiten am Kontinuitätselement*

Auch wird der hierarchische Aufbau der Formvariablen gezeigt: die Verschiebung des Mittelpunktes resultiert in einer Translation, die Verschiebungen der Seitenmittelknoten in Kippungen und die Verschiebung eines Eckknotens in einer "Verzerrung" des Elementes zu einem Hyperparaboloid. In Bild 4.31 ist eine mögliche Anwendung der Kontinuitätselemente in Verbindung mit Bézier-Elementen zur Generierung  $G_1$ -kontinuierlicher Flächen dargestellt.

Die Kontrolle der Linienfaktoren birgt programmtechnisch einige Tücken, da weitere Querverweise zur Darstellung der topologischen Verknüpfungen zwischen Kanten und Linien notwendig sind. In CARAT ist das Problem so gelöst, daß die Kontinuitätselemente das Bindeglied zwischen Kanten und Linien sind. Zu einer gegebenen Linie können im entsprechenden Attributfeld eines Kontinuitätselementes die Verweise zu den vorherigen sowie zu den nächsten Kanten und Kontinuitätselementen gefunden werden (Bild 4.32). Damit ist es möglich, die gesamte Linie abzugehen. Anfang und Ende einer Linie sind durch die dort befindlichen Kontinuitätselemente definiert und als Attribute der Linie zugewiesen. Diese komplexen topologischen Beziehungen, die, das sei ausdrücklich bemerkt, die gesamte Form netzartig überziehen, werden automatisch generiert. Es müssen nur diejenigen Kanten bezeichnet werden, an denen in Querrichtung die  $G_1$ -Kontinuitätsbedingung gelten soll.

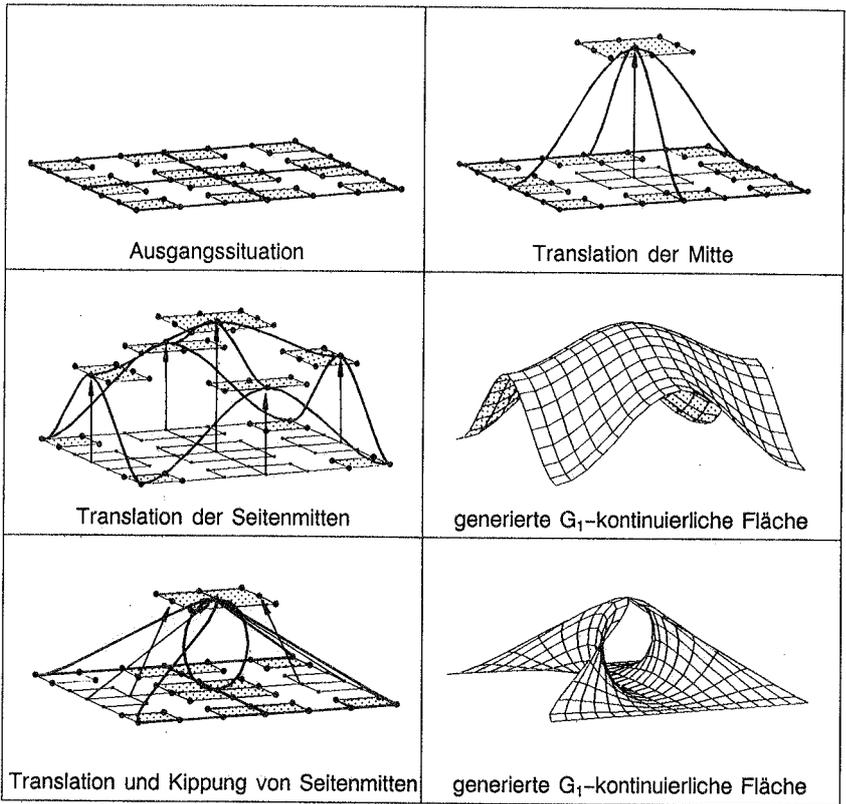


Bild 4.31: Generierung  $G_1$ -kontinuierlicher Flächen mit Kontinuitätselementen

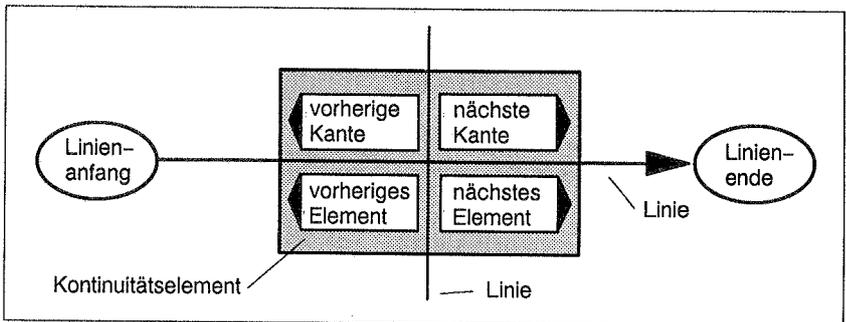


Bild 4.32: Topologische Verknüpfung im Kontinuitätselement

#### 4.3.4 Koordinatensysteme

Die Entwurfsfreiheiten können wesentlich erweitert werden, wenn ein Preprozessor zur Verfügung steht, der es erlaubt, verschiedene, zueinander verschobene und verdrehte kartesische Koordinatensysteme zu kombinieren. Damit können Strukturen in Unterstrukturen aufgeteilt werden, die zweckmäßigerweise in einem besonderen Koordinatensystem definiert werden, zum Beispiel im Raum gedrehte und verschobene Ebenen oder Kugeln um einen beliebigen Punkt im Raum.

Das allem zugrunde liegende kartesische Koordinatensystem ist das globale Koordinatensystem. Neue Koordinatensysteme können nur in einer bereits bekannten, "alten" Umgebung, d.h. innerhalb eines Referenzsystems, definiert werden. Damit ergibt sich ein Baum mit einer weit verzweigten Verschachtelung von Definitionen. Die "Wurzel" ist das globale System (siehe z.B. Bild 4.33). Für jedes System – mit Ausnahme des globalen Systems – ist das System der jeweils unteren Verzweigungsebene oder Generation das Referenzsystem. Im Bild 4.33 heißt dies: für die Systeme 1, 2 und 3 ist das globale System das Referenzsystem, für das System 4 ist 1 und für die Systeme 5 und 6 das System 3 das Referenzsystem.

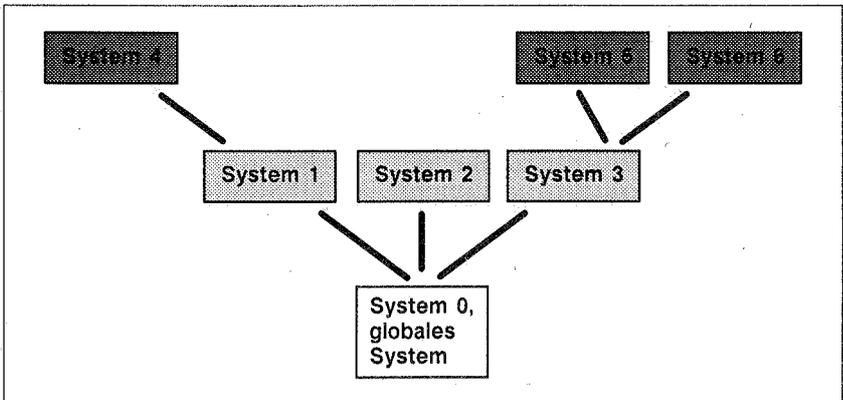
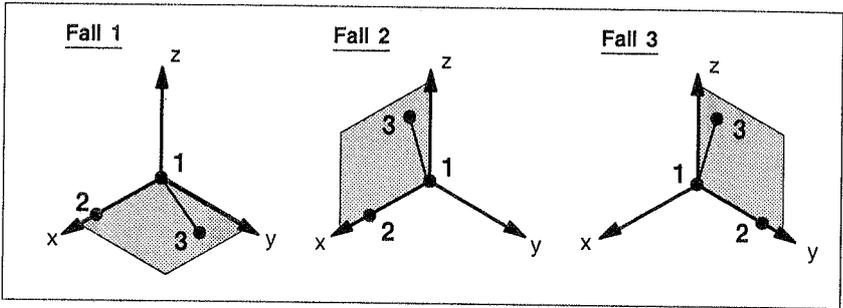


Bild 4.33: "Baumstruktur" der Koordinatensysteme

Die Koordinatensysteme werden durch drei Design-Knoten definiert (Bild 4.34):

- ein Ursprung (Knoten 1).
- Knoten 2, der die erste Richtung festlegt (z.B. x-Richtung).
- Knoten 3, der zusammen mit den beiden anderen die erste Ebene festlegt, auf die die dritte Richtung senkrecht steht. Daraus bestimmt sich dann die zweite Richtung.



**Bild 4.34:** Definitionen eines Rechtssystems mit Designknoten

Für die Definition von Rechtssystemen gibt es demnach drei Möglichkeiten:

- 1.) Angegeben werden: der Ursprung ( $\mathbf{r}_1$ ), die Knoten, die die  $x$ -Richtung ( $\mathbf{r}_2$ ) und die  $x$ - $y$ -Ebene ( $\mathbf{r}_3$ ) festlegen (Bild 4.35). Damit ist festgelegt:

$$x\text{-Richtung: } \mathbf{e}_1 = \frac{\mathbf{r}_2 - \mathbf{r}_1}{|\mathbf{r}_2 - \mathbf{r}_1|}$$

$$z\text{-Richtung: } \mathbf{e}_3 = \frac{(\mathbf{r}_2 - \mathbf{r}_1) \times (\mathbf{r}_3 - \mathbf{r}_1)}{|(\mathbf{r}_2 - \mathbf{r}_1) \times (\mathbf{r}_3 - \mathbf{r}_1)|} \quad (4.35)$$

$$y\text{-Richtung: } \mathbf{e}_2 = \mathbf{e}_3 \times \mathbf{e}_1$$

- 2.) Angegeben werden: der Ursprung ( $\mathbf{r}_1$ ), die Knoten, die die  $x$ -Richtung ( $\mathbf{r}_2$ ) und die  $x$ - $z$ -Ebene ( $\mathbf{r}_3$ ) festlegen:

$$x\text{-Richtung: } \mathbf{e}_1 = \frac{\mathbf{r}_2 - \mathbf{r}_1}{|\mathbf{r}_2 - \mathbf{r}_1|}$$

$$y\text{-Richtung: } \mathbf{e}_2 = \frac{(\mathbf{r}_3 - \mathbf{r}_1) \times (\mathbf{r}_2 - \mathbf{r}_1)}{|(\mathbf{r}_3 - \mathbf{r}_1) \times (\mathbf{r}_2 - \mathbf{r}_1)|} \quad (4.36)$$

$$z\text{-Richtung: } \mathbf{e}_3 = \mathbf{e}_1 \times \mathbf{e}_2$$

- 3.) Angegeben werden: der Ursprung ( $\mathbf{r}_1$ ), die Knoten, die die  $y$ -Richtung ( $\mathbf{r}_2$ ) und die  $y$ - $z$ -Ebene ( $\mathbf{r}_3$ ) festlegen:

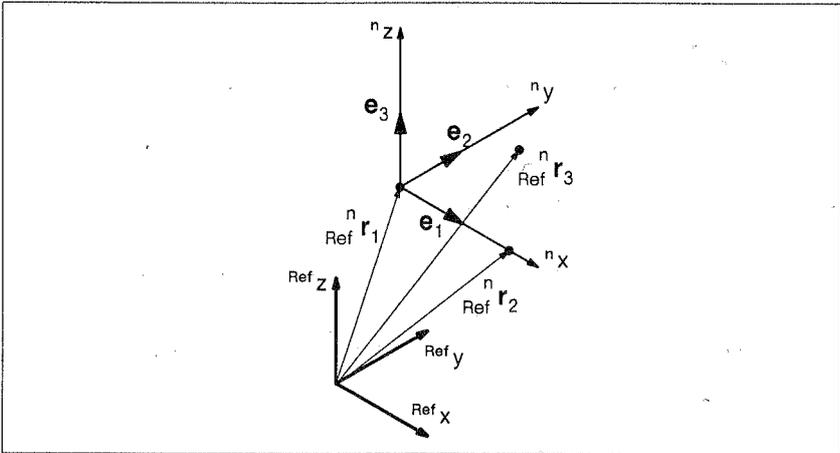
$$y\text{-Richtung: } \mathbf{e}_2 = \frac{\mathbf{r}_2 - \mathbf{r}_1}{|\mathbf{r}_2 - \mathbf{r}_1|}$$

$$x\text{-Richtung: } \mathbf{e}_1 = \frac{(\mathbf{r}_2 - \mathbf{r}_1) \times (\mathbf{r}_3 - \mathbf{r}_1)}{|(\mathbf{r}_2 - \mathbf{r}_1) \times (\mathbf{r}_3 - \mathbf{r}_1)|} \quad (4.37)$$

$$z\text{-Richtung: } \mathbf{e}_3 = \mathbf{e}_1 \times \mathbf{e}_2$$

Die Einheitsvektoren  $e_i$  werden in einer Transformationsmatrix  $T$  zusammengefaßt:

$$T = (e_1, e_2, e_3) \quad (4.38)$$



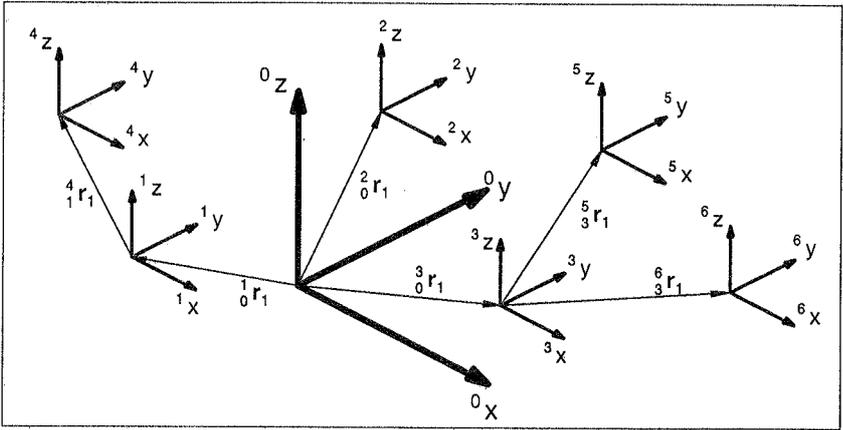
**Bild 4.35:** Rechtssystem, Fall 1

Äquivalent zum geschilderten Vorgehen können in CARAT-Design auch Linkssysteme definiert werden. Die indirekte Angabe der Koordinatenrichtungen durch Kontrollknoten hat die Vorteile, daß interaktiv die Lage von Koordinatensystemen sehr anschaulich durch ein Verschieben von Knoten verändert werden kann. Für die Strukturoptimierung brauchen keine neuen Variablen definiert werden, variable Design-Koordinaten reichen vollständig aus.

Die Ortsvektoren  $r_1$ ,  $r_2$  und  $r_3$ , die zur Definition eines Koordinatensystems dienen, erhalten zwei linke Indizes, deren oberer die Nummer des zu definierenden Koordinatensystems und deren unterer die Nummer des Referenzsystems angibt.

$$\begin{aligned} \text{KS}_{\text{Ref}} r_1 & \dots \text{ Ortsvektor zum Ursprung des neuen Koordinatensystems KS,} \\ & \text{gemessen im System Ref} \\ \text{KS}_{\text{Ref}} r_2 & \dots \text{ Ortsvektor zum zweiten Knoten} \\ \text{KS}_{\text{Ref}} r_3 & \dots \text{ Ortsvektor zum dritten Knoten} \end{aligned} \quad (4.39)$$

Die zum Bild 4.33 gehörende Anordnung von Koordinatensystemen ist in Bild 4.36 dargestellt.



**Bild 4.36:** Geometrische Beziehungen zwischen Koordinatensystemen

Ein Koordinatensystem KS ist bestimmt durch den Ortsvektor zum Ursprung  ${}^{\text{KS}}_{\text{Ref}}\mathbf{r}_1$  und die Matrix  ${}^{\text{KS}}_{\text{Ref}}\mathbf{T}$  der Basisvektoren  ${}^{\text{KS}}_{\text{Ref}}\mathbf{e}_i$ . Vor dem Abspeichern werden der Vektor zum Ursprung und die Matrix in das globale System transformiert. Damit liegt fest:

${}^{\text{KS}}_0\mathbf{e}_1$ ...	normierter Richtungsvektor in x-Richtung des Systems KS, dargestellt in Komponenten des globalen Systems
${}^{\text{KS}}_0\mathbf{e}_2$ ...	entsprechend für die y-Richtung des Systems
${}^{\text{KS}}_0\mathbf{e}_3$ ...	entsprechend für die z-Richtung des Systems
${}^{\text{KS}}_0\mathbf{r}_1$ ...	globaler Ortsvektor zum Ursprung des Systems KS

Bei der Transformation in das globale System müssen die Verkettungen im Baum der Systemdefinitionen (Bild 4.33) berücksichtigt werden, von einem Referenzsystem zum nächst tieferen bis hinab zum globalen System. Das System KS sei in der n-ten Generation definiert, dann lautet die Transformationsvorschrift:

- 1.) Setze:  $i = 0$
- 2.)  $i = n ?$ , falls ja: Ende der Transformation, gehe nach 6.)
- 3.)  $i = i + 1$
- 4.)  ${}^{\text{KS}}_{n-(i+1)}\mathbf{T} = {}^{\text{KS}}_{n-(i+1)}\mathbf{T} \begin{pmatrix} n \\ n-i \end{pmatrix} \mathbf{T}$ ;  ${}^{\text{KS}}_{n+(i+1)}\mathbf{r}_1 = {}^{\text{KS}}_{n-(i+1)}\mathbf{T} {}^{\text{KS}}_{n-i}\mathbf{r}_1 + {}^{\text{KS}}_{n-(i+1)}\mathbf{r}_1$  (4.40)
- 5.) Gehe nach 2.)
- 6.)  ${}^{\text{KS}}_0\mathbf{T} = {}^{\text{KS}}_0\mathbf{T}$ ;  ${}^{\text{KS}}_0\mathbf{r}_1 = {}^{\text{KS}}_0\mathbf{r}_1$

Mit der Definition der Vektor- und Matrixkomponenten im globalen System wird eine Vereinheitlichung vorgenommen, mit der es möglich ist, bequem von einem System in irgendein anderes zu transformieren; allerdings stets mit dem "Umweg" über das globale System, dagegen ohne das ständige Nachvollziehen von (4.40). Die Transformation von System n in das System m erfolgt demnach in zwei Schritten:

- 1.) Transformiere von n in das globale System 0:

$${}^0_0\mathbf{r} = {}^n_0\mathbf{T} \begin{pmatrix} n\mathbf{r} \\ n\mathbf{r}_1 \end{pmatrix} + {}^n_0\mathbf{r}_1 \quad (4.41a)$$

- 2.) Transformiere vom globalen System in das System m:

$${}^m_m\mathbf{r} = {}^m_0\mathbf{T}^T \begin{pmatrix} {}^0_0\mathbf{r} \\ {}^0_0\mathbf{r}_1 \end{pmatrix} \quad (4.41b)$$

Zusammengefaßt:

$${}^m_m\mathbf{r} = {}^m_0\mathbf{T}^T \left( {}^n_0\mathbf{T} \begin{pmatrix} n\mathbf{r} \\ n\mathbf{r}_1 \end{pmatrix} + {}^n_0\mathbf{r}_1 - {}^m_0\mathbf{r}_1 \right) \quad (4.41c)$$

Weitere Möglichkeiten ergeben sich mit der Einführung von Zylinder- und Kugelkoordinaten, z.B. zur Beschreibung rotationssymmetrischer Tragwerke, die in der Praxis sehr häufig vorkommen. Die geometrischen Beziehungen zwischen Zylinder- bzw. Kugelkoordinaten sind in Bild 4.37 angegeben.

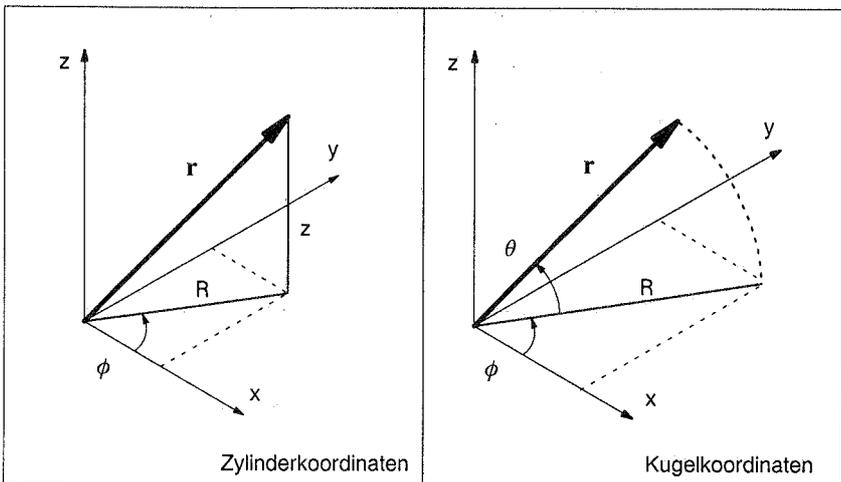


Bild 4.37: Zylinder- und Kugelkoordinaten

Es ergeben sich folgende Transformationsoperatoren  ${}^1B$ :

Zylinderkoordinaten,  ${}^1B = Z$ :

$${}^n\mathbf{r} = Z ({}^n\mathbf{z}) ; \quad {}^n\mathbf{z} = Z^{-1} ({}^n\mathbf{r}) = {}^n\mathbf{b}$$

$$\text{mit: } {}^n\mathbf{z} = (R, \phi, z)^T ; \quad {}^n\mathbf{r} = (x, y, z)^T$$

$$Z = \left\{ \begin{array}{c} R \cos \phi \\ R \sin \phi \\ z \end{array} \right\} \quad Z^{-1} = \left\{ \begin{array}{c} \sqrt{x^2 + y^2} \\ \text{sign} \left( \arccos \left( \frac{x}{\sqrt{x^2 + y^2}} \right), \frac{y}{\sqrt{x^2 + y^2}} \right) \\ z \end{array} \right\} \quad (4.42)$$

In der dargestellten Form ist  $Z^{-1}$  für  $-\pi < \phi \leq \pi$  und  $R > 0$  eindeutig.

Kugelkoordinaten,  ${}^1B = K$ :

$${}^n\mathbf{r} = K ({}^n\mathbf{k}) ; \quad {}^n\mathbf{k} = K^{-1} ({}^n\mathbf{r}) = {}^n\mathbf{b}$$

$$\text{mit: } {}^n\mathbf{k} = (R, \phi, \theta)^T ; \quad {}^n\mathbf{r} = (x, y, z)^T$$

$$K = \left\{ \begin{array}{c} R \cos \phi \cos \theta \\ R \sin \phi \cos \theta \\ R \sin \theta \end{array} \right\} \quad (4.43)$$

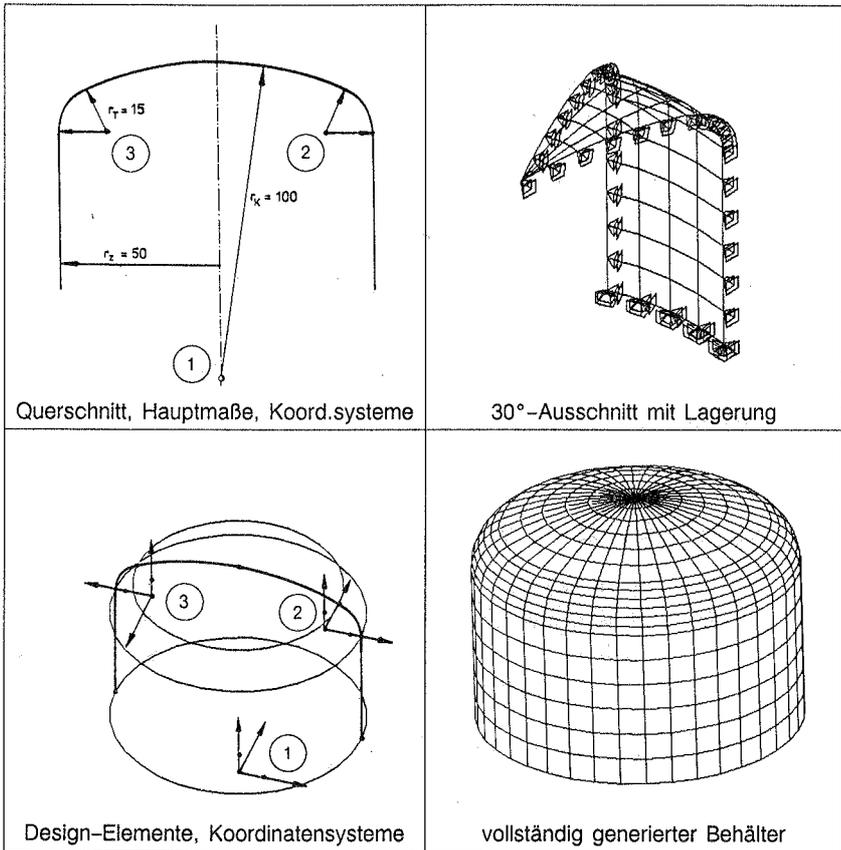
$$K^{-1} = \left\{ \begin{array}{c} \sqrt{x^2 + y^2 + z^2} \\ \text{sign} \left( \arccos \left( \frac{x}{\sqrt{x^2 + y^2}} \right), \frac{y}{\sqrt{x^2 + y^2}} \right) \\ \arcsin \frac{z}{\sqrt{x^2 + y^2 + z^2}} \end{array} \right\}$$

Auch hier gilt:  $-\pi < \phi \leq \pi$  und  $R > 0$  sowie  $-\pi/2 < \theta < \pi/2$ , damit  $K^{-1}$  eindeutig ist.

Mit den Transformationsoperatoren kann (4.41c) verallgemeinert werden:

$${}^m\mathbf{b} = {}^mB^{-1} ({}^m\mathbf{r}) = {}^mB^{-1} \left\{ \begin{array}{c} {}^m\mathbf{T}^T ({}^n\mathbf{T} \quad {}^nB ({}^n\mathbf{b}) + {}^n\mathbf{r}_1 - {}^m\mathbf{r}_1) \end{array} \right\} \quad (4.44)$$

Unter Berücksichtigung der Definitionsbereiche der Zylinder- und Kugelkoordinaten ist (4.44) die Grundlage für eine freizügige Formbeschreibung gleichzeitig in verschiedenen Koordinatensystemen mit kartesischen Koordinaten, Zylinder- und Kugelkoordinaten. Es kann beliebig in diesen Koordinaten generiert werden, wobei alle beschriebenen Design-Elemente zur Verfügung stehen. Bild 4.38 zeigt ein Beispiel.



**Bild 4.38:** Formbeschreibung eines Tankbehälters (Zylinder, Torus, Kugel)

Für die Strukturoptimierung haben die geschilderten Erweiterungen des Design-Modells weitreichende Konsequenzen. Es können gleichzeitig in allen definierten Koordinatensystemen variable Design-Koordinaten existieren, gleichgültig in welcher Form (kartesisch, Zylinder, Kugel). Sind die Design-Koordinaten von Knoten variabel, die die Lage von Koordinatensystemen beschreiben, können ganze Teile der Struktur auf einmal in ihrer Lage verändert und optimiert werden. Damit sind komplexe Kombinationen von Optimierungsvariablen möglich, womit gleichzeitig die Formfreiheiten vergrößert und die Zahl der Unbekannten verringert werden können. Dies wirkt sich günstig auf die Aussagekraft des Modells und die Stabilität sowie Effektivität des Lösungsalgorithmus aus.

#### 4.3.5 Design-Macros

Design-Macros sind Prozeduren, mit denen ganze Teiläste des "Koordinatensystembaumes" mitsamt den in diesen Systemen definierten Elementen, Kanten usw. in andere Systeme transformiert werden können. Mit diesen Prozeduren können bequem symmetrische Ergänzungen und Vervielfachungen eines Strukturteiles erzielt werden. Die generierte Gesamtstruktur liegt am Ende vollständig vor. Im Gegensatz zu einer nachträglichen symmetrischen Ergänzung einer Berechnung am Teilsystem können hier auch symmetrische Systeme mit unsymmetrischer Belastung berechnet werden. Bei Design-Macros werden FE-Knoten und -Elemente fortlaufend generiert. Randbedingungen können an der Gesamtstruktur definiert werden.

Wenn das System n in das System m transformiert werden soll, wird bei der Abarbeitung der Generierungsvorschriften geprüft, ob diese Vorschriften (das sind die Interpolations- bzw. Approximationsvorschriften der Design-Elemente) im System n oder einem Untersystem k von n definiert sind. Ist dies der Fall, wird die Generierung im System k vorgenommen und anschließend werden die notwendigen Transformationen durchgeführt. Andernfalls wird das betreffende Design-Element übergangen. Die Transformationsvorschrift der Macro-Prozedur lautet für die Überführung von System n in das System m und anschließender Transformation in das globale System:

$${}^0_0\mathbf{r} = {}^m_0\mathbf{T} \quad {}^n_0\mathbf{r} + {}^m_0\mathbf{r}_1 \quad (4.45)$$

Das System k ist ein Untersystem des Systems n. Mit Gleichung (4.41 c) ergibt sich folgende allgemeine Transformationsvorschrift für die Abbildung von k nach m:

$${}^0_0\mathbf{r} = {}^m_0\mathbf{T} \left\{ {}^n_0\mathbf{T}^T \left( {}^k_0\mathbf{T} \quad {}^k_0\mathbf{r} + {}^k_0\mathbf{r}_1 - {}^n_0\mathbf{r}_1 \right) \right\} + {}^m_0\mathbf{r}_1 \quad (4.46)$$

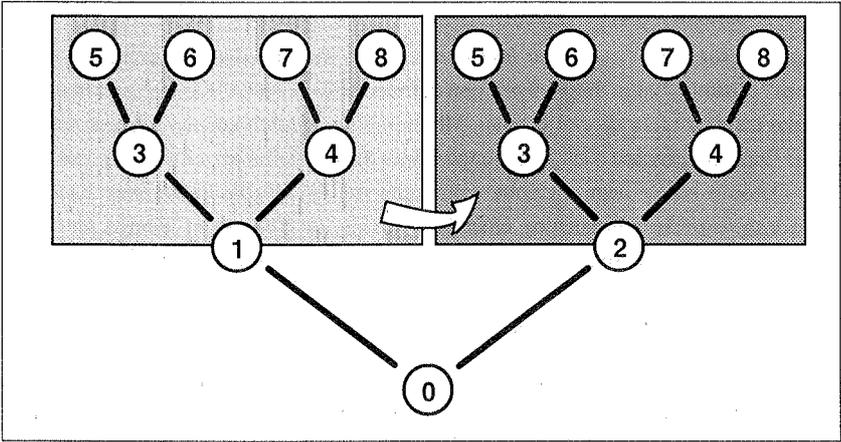
Hier kann wieder vorteilhaft Gebrauch von der Tatsache gemacht werden, daß die Transformationsmatrizen T in globalen Komponenten vorgehalten werden. Damit vereinfacht sich die Transformation von k nach n wesentlich.

Im Bild 4.39 soll das Vorgehen schematisch verdeutlicht werden.

Mit dem im Datensatz anzugebenden Macro-Befehl:

MACRO	1	SYS	1	TO	2	END
-------	---	-----	---	----	---	-----

werden das gesamte System 1 sowie alle Untersysteme 3 bis 8 in das System 2 gemäß (4.46) transformiert. Durch Verkettung von Macro-Prozeduren können mit wenig Aufwand große, regelmäßige Strukturen definiert werden.

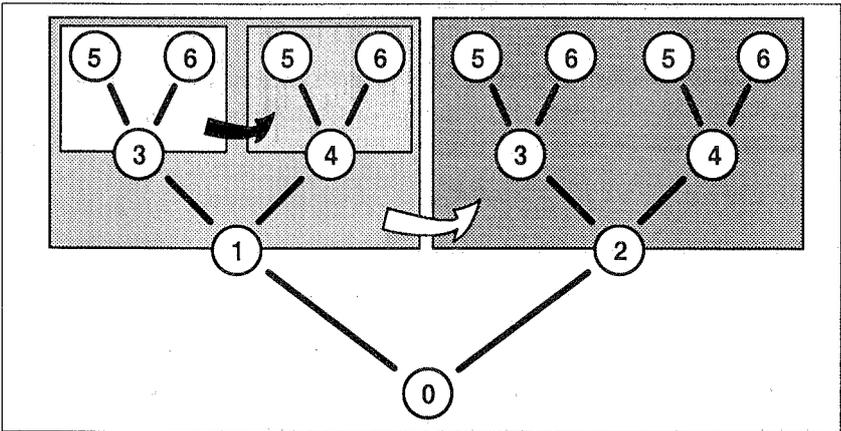


**Bild 4.39:** Transformation von System 1 nach System 2

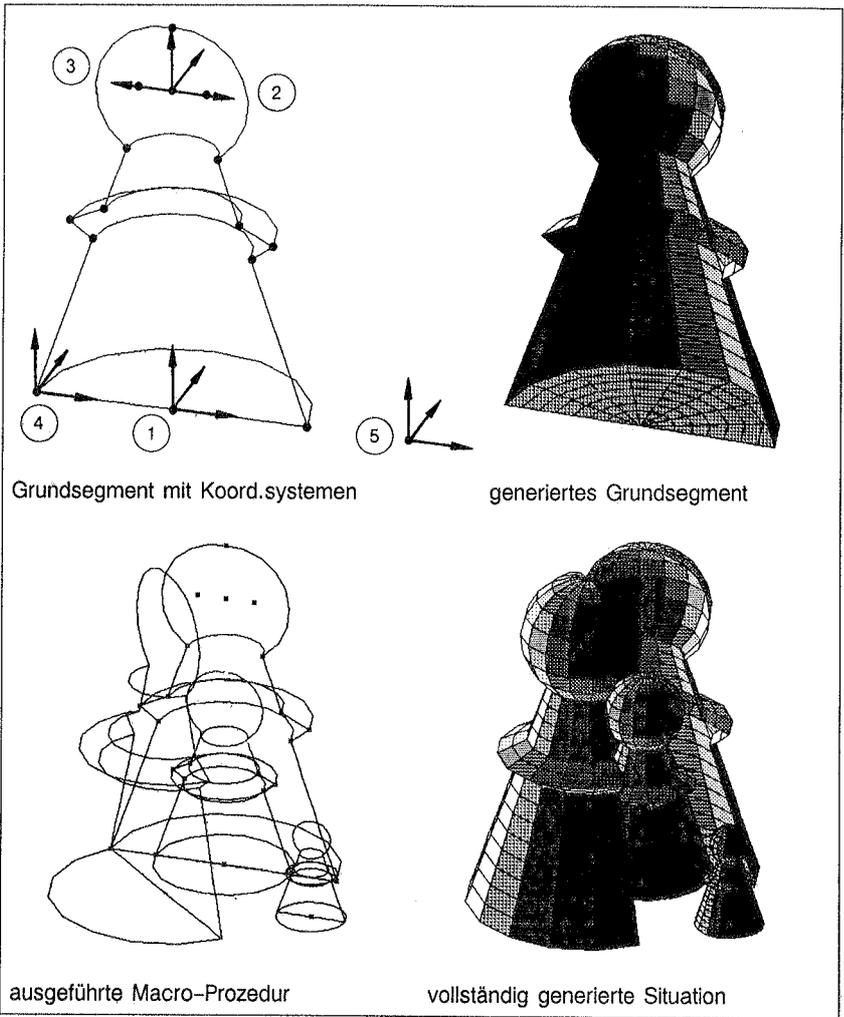
Mit den Befehlen:

MACRO	1	SYS	3	TO	4	
MACRO	2	SYS	1	TO	2	END

wird der in Bild 4.40 gezeigte Baum mit zunächst 6 Koordinatensystemen auf insgesamt 14 aufgefüllt. Das in Bild 4.41 gezeigte Figurenensemble wurde aus nur einer Figurenhälfte erzeugt. Die kompletten Figuren wurden durch wiederholte und verkettete Macro-Prozeduren ergänzt. Wie man sieht, sind dabei auch Skalierungen möglich.



**Bild 4.40:** Verkettung von Transformationen



**Bild 4.41:** Generieren komplexer Strukturen mit Design-Macros

#### 4.3.6 Sensibilitätsanalyse

Zur Bestimmung der Sensibilität der Tragwerksantworten (Spannungen, Verschiebungen etc.) bezüglich Änderungen der Optimierungsvariablen müssen die Ableitungen der generierten Form und Dickenverteilung nach den Variablen bekannt sein. Sie gehen über die Kettenregel in die Berechnung ein. In CARAT-Design werden die Formableitungen explizit bestimmt. Damit ist die angestrebte enge Verzahnung von Entwurfs-, Berechnungs- und Optimierungsmodell erst möglich. Diese Eigenschaften gehen über das hinaus, was übliche Preprozessoren leisten können. Dabei müssen alle vorne beschriebenen Verknüpfungen und Transformationen vom Kontinuitätselement über die Formfunktionen bis zur Macro-Prozedur berücksichtigt werden. Im folgenden werden alle im Entwurfsmodell auftretenden Ableitungen in der Reihenfolge ihre Auftretens dargestellt. Exemplarisch für die Koordinaten, für Tragwerksdicken gelten die Formulierungen entsprechend. Für die Ableitung der Lokalsysteme sind noch weiterführende Überlegungen notwendig, die hier aus Platzgründen allerdings nicht angegeben werden können. Das prinzipielle Vorgehen entspricht dem dargestellten.

Nachdem die Verknüpfungsvorschriften (Kapitel 5) berücksichtigt worden sind, liegen die Ableitungen der Design-Koordinaten nach den Optimierungsvariablen  $\mathbf{x}$  vor:

kartesische Koordinaten:	$\mathbf{r}_{1,x}$ ,
Zylinder- oder Kugelkoordinaten:	$\mathbf{b}_{1,x}$ ,
Linien- und Tangentenfaktoren:	$\delta_{,x}$

Im nächsten Schritt werden die Verknüpfungsvorschriften der Tangenten- und Kontinuitätselemente ausgewertet.

##### Ableitung der $G_1$ -Tangentenelemente:

Die Ableitung der Gleichung (4.27) ergibt:

$$\mathbf{r}_{1,x} = \mathbf{r}_{0,x} + \delta (\mathbf{r}_{0,x} - \mathbf{r}_{-1,x}) + \delta_{,x} (\mathbf{r}_0 - \mathbf{r}_{-1}) \quad (4.47)$$

##### Ableitung der $G_2$ -Tangentenelemente:

Die Ableitungen der Verknüpfungsvorschriften (4.28) lauten:

$$\begin{aligned} \mathbf{r}_{-1,x} = & \frac{1}{(1+3\delta)(2+\delta)} \left[ \delta (1+\delta) \mathbf{r}_{-2,x} + 2 (1+\delta)(2+\delta) \mathbf{r}_{0,x} - 2 \mathbf{r}_{2,x} \right] \\ & + \left( \frac{1}{(1+3\delta)(2+\delta)} \right)^2 \left[ 2 (1+2\delta(1+\delta)) \mathbf{r}_{-2} - 4 (\delta+2)^2 \mathbf{r}_0 + 2 (7+6\delta) \mathbf{r}_2 \right] \delta_{,x} \end{aligned} \quad (4.48a)$$

$$\begin{aligned}
\mathbf{r}_{1,x} = & \frac{1}{(1+3\delta)(2+\delta)} \left[ -\delta^2 (1+\delta) \mathbf{r}_{-2,x} + (1+\delta)^2(2+\delta) \mathbf{r}_{0,x} + 2 \delta \mathbf{r}_{2,x} \right] \\
& + \left( \frac{1}{(1+3\delta)(2+\delta)} \right)^2 \left[ -\delta (4+13\delta+14\delta^2+3\delta^3) \mathbf{r}_{-2} + \right. \\
& \left. + (-4+4\delta+19\delta^2+14\delta^3+3\delta^4) \mathbf{r}_0 + (4-6\delta^2) \mathbf{r}_2 \right] \delta_x
\end{aligned} \tag{4.48b}$$

Ableitung der Kontinuitätselemente:

Aus Gleichung (4.34) folgt:

$$\begin{aligned}
\mathbf{r}_{5,x} &= \mathbf{r}_{4,x} + \delta_{2,x} (\mathbf{r}_4 - \mathbf{r}_3) + \delta_2 (\mathbf{r}_{4,x} - \mathbf{r}_{3,x}) \\
\mathbf{r}_{6,x} &= \mathbf{r}_{1,x} + \delta_{2,x} (\mathbf{r}_1 - \mathbf{r}_2) + \delta_2 (\mathbf{r}_{1,x} - \mathbf{r}_{2,x}) \\
\mathbf{r}_{7,x} &= \mathbf{r}_{1,x} + \delta_{2,x} (\mathbf{r}_1 - \mathbf{r}_2) + \delta_2 (\mathbf{r}_{1,x} - \mathbf{r}_{2,x}) \\
&+ \delta_{1,x} (\mathbf{r}_1 - \mathbf{r}_4) + \delta_1 (\mathbf{r}_{1,x} - \mathbf{r}_{4,x}) \\
&+ (\delta_{1,x} \delta_2 + \delta_1 \delta_{2,x}) (\mathbf{r}_1 - \mathbf{r}_2 + \mathbf{r}_3 - \mathbf{r}_4) \\
&+ \delta_1 \delta_2 (\mathbf{r}_{1,x} - \mathbf{r}_{2,x} + \mathbf{r}_{3,x} - \mathbf{r}_{4,x}) \\
\mathbf{r}_{8,x} &= \mathbf{r}_{1,x} + \delta_{1,x} (\mathbf{r}_1 - \mathbf{r}_4) + \delta_1 (\mathbf{r}_{1,x} - \mathbf{r}_{4,x}) \\
\mathbf{r}_{9,x} &= \mathbf{r}_{2,x} + \delta_{1,x} (\mathbf{r}_2 - \mathbf{r}_3) + \delta_1 (\mathbf{r}_{2,x} - \mathbf{r}_{3,x})
\end{aligned} \tag{4.49}$$

Damit sind alle Design-Koordinaten abgeleitet und sämtliche Verknüpfungsmöglichkeiten berücksichtigt.

Als nächstes folgt die eigentliche Formgenerierung und deren Ableitung nach den Optimierungsvariablen. Dazu werden des öfteren die abgeleiteten Transformationsmatrizen  $T$  der Koordinatensysteme benötigt.

Ableitung der Basisvektoren der Koordinatensysteme:

Die Ableitungen werden exemplarisch für ein Rechtssystem des Falles 1 angegeben (siehe Bilder 4.34 und 4.35):

$$\begin{aligned}
\text{mit: } \quad \mathbf{d}_1 &= {}^n_m \mathbf{r}_2 - {}^n_m \mathbf{r}_1 \quad ; \quad \mathbf{d}_2 = {}^n_m \mathbf{r}_3 - {}^n_m \mathbf{r}_1 \\
l_1 &= |\mathbf{d}_1| \quad ; \quad l_2 = |\mathbf{d}_2|
\end{aligned}$$

$n$  ... Nummer des Koordinatensystems

$m$  ... Nummer des Referenzsystems

Es gelten für die Basisvektoren die Gleichungen (4.35), hier etwas anders geschrieben:

$$\begin{aligned} {}^n_m \mathbf{e}_1 &= \frac{1}{l_1} \mathbf{d}_1 \\ {}^n_m \mathbf{e}_3 &= \frac{1}{l_1 l_2} (\mathbf{d}_1 \times \mathbf{d}_2) \\ {}^n_m \mathbf{e}_2 &= {}^n_m \mathbf{e}_3 \times {}^n_m \mathbf{e}_1 \end{aligned} \quad (4.35')$$

Für die Ableitungen folgt:

$$\begin{aligned} {}^n_m \mathbf{e}_{1,x} &= \frac{1}{l_1} \mathbf{d}_{1,x} - \frac{1}{l_1^2} l_{1,x} \mathbf{d}_1 \\ {}^n_m \mathbf{e}_{3,x} &= \frac{1}{l_1 l_2} (\mathbf{d}_{1,x} \times \mathbf{d}_2 + \mathbf{d}_1 \times \mathbf{d}_{2,x}) \\ &\quad - \frac{1}{l_1 l_2} \left( \frac{1}{l_1} l_{1,x} + \frac{1}{l_2} l_{2,x} \right) (\mathbf{d}_1 \times \mathbf{d}_2) \\ {}^n_m \mathbf{e}_{2,x} &= {}^n_m \mathbf{e}_{3,x} \times {}^n_m \mathbf{e}_1 + {}^n_m \mathbf{e}_3 \times {}^n_m \mathbf{e}_{1,x} \end{aligned} \quad (4.50)$$

Dabei gilt:

$$\begin{aligned} \mathbf{d}_{1,x} &= {}^n_m \mathbf{r}_{2,x} - {}^n_m \mathbf{r}_{1,x} & ; & \quad \mathbf{d}_{2,x} = {}^n_m \mathbf{r}_{3,x} - {}^n_m \mathbf{r}_{1,x} \\ l_{1,x} &= \frac{1}{l_1} \left( \mathbf{d}_1^T \mathbf{d}_{1,x} \right) & ; & \quad l_{2,x} = \frac{1}{l_2} \left( \mathbf{d}_2^T \mathbf{d}_{2,x} \right) \end{aligned}$$

Die gesuchten Ableitungen  ${}^n_0 \mathbf{e}_{i,x}$  werden analog (4.40) aus den abfolgenden Transformationen von System n bis hinab zum globalen System gewonnen. Für die abgeleitete Transformationsmatrix gilt dann:

$${}^n_0 \mathbf{T}_{,x} = \left( {}^n_0 \mathbf{e}_{1,x}, {}^n_0 \mathbf{e}_{2,x}, {}^n_0 \mathbf{e}_{3,x} \right)$$

Transformation der Design-Knoten in das Koordinatensystem der Design-Elemente:

Die Generierung soll in einem Design-Element vorgenommen werden, das im System k definiert ist. Die Koordinaten der Kontrollknoten  $\mathbf{b}_i$  sind von einem anderen Typ, als verlangt (z.B. Zylinderkoordinaten statt kartesische) und im System p definiert. Gemäß Gleichung (4.44) lauten dann die Transformation

$${}^k_b \mathbf{b}_i = {}^k_B^{-1} \left\{ {}^k_0 \mathbf{T}^T \left( {}^p_0 \mathbf{T} {}^p_B \left( \frac{p}{p} \mathbf{b}_i \right) + {}^p_0 \mathbf{r}_1 - {}^k_0 \mathbf{r}_1 \right) \right\} \quad (4.51a)$$

und deren Ableitungen:

$$\begin{aligned}
 {}^k\mathbf{b}_{i,x} &= {}^k\mathbf{B}_b^{-1} \left\{ {}^k_0\mathbf{T}^T \left( {}^p_0\mathbf{T} \mathbf{PB} \left( {}^p_p\mathbf{b}_i \right) + {}^p_0\mathbf{r}_1 - {}^k_0\mathbf{r}_1 \right) \right\} \\
 &\quad \left[ {}^k_0\mathbf{T}_{,x}^T \left( {}^p_0\mathbf{T} \mathbf{PB} \left( {}^p_p\mathbf{b}_i \right) + {}^p_0\mathbf{r}_1 - {}^k_0\mathbf{r}_1 \right) \right. \\
 &\quad \left. + {}^k_0\mathbf{T}^T \left( {}^p_0\mathbf{T}_{,x} \mathbf{PB} \left( {}^p_p\mathbf{b}_i \right) + {}^p_0\mathbf{T} \mathbf{PB}_b \left( {}^p_p\mathbf{b}_i \right) \frac{{}^p_p\mathbf{b}_{i,x}}{p} \right. \right. \\
 &\quad \left. \left. + {}^p_0\mathbf{r}_{1,x} - {}^k_0\mathbf{r}_{1,x} \right) \right]
 \end{aligned} \tag{4.51b}$$

Generierung im Koordinatensystem der Design-Elemente:

Mit den Kontrollknoten  ${}^k\mathbf{b}_i$  und den Ableitungen kann generiert werden (vergleiche mit Gleichung (4.4)):

$${}^k\mathbf{b} = \sum_i \Phi_i {}^k\mathbf{b}_i \tag{4.52}$$

Außer für das Kreiselement (Abschnitt 4.3.2.2.4) sind alle Formfunktionen nur Funktionen der Parameter  $u$  und  $v$ , so daß gilt:

$${}^k\mathbf{b}_{,x} = \sum_i \Phi_i {}^k\mathbf{b}_{i,x} \tag{4.53}$$

und die anschließende Transformation in kartesische Koordinaten:

$${}^k\mathbf{r} = {}^k\mathbf{B} ({}^k\mathbf{b}) \quad \text{bzw.} \quad {}^k\mathbf{r}_{,x} = {}^k\mathbf{B}_b ({}^k\mathbf{b}) {}^k\mathbf{b}_{,x} \tag{4.54}$$

Ableitung des Transformationsoperators für Zylinderkoordinaten:

In den Gleichungen (4.51) und (4.54) tritt der abgeleitete Transformationsoperator auf. Im Falle der Zylinderkoordinaten ( $\mathbf{B} = \mathbf{Z}$ ) gilt:

$$\mathbf{Z} = \begin{Bmatrix} R \cos \phi \\ R \sin \phi \\ z \end{Bmatrix} \quad \mathbf{Z}_{,z} = \begin{bmatrix} \cos \phi & -R \sin \phi & 0 \\ \sin \phi & R \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.55}$$

$$\mathbf{Z}_{,z}^{-1} = \frac{1}{R} \begin{bmatrix} R \cos \phi & R \sin \phi & 0 \\ -\sin \phi & \cos \phi & 0 \\ 0 & 0 & R \end{bmatrix}$$

mit:  $\mathbf{r} = \mathbf{Z}(z) \quad \mathbf{r}_{,x} = \mathbf{Z}_{,z}(z) z_{,x}$   
 $\mathbf{z} = \mathbf{Z}^{-1}(\mathbf{r}) \quad \mathbf{z}_{,x} = \mathbf{Z}_{,z}^{-1}(\mathbf{z}) r_{,x}$

Ableitungen des Transformationsoperators für Kugelkoordinaten:

Entsprechend (4.55) gilt für Kugelkoordinaten ( $B = K$ ):

$$\mathbf{K} = \begin{Bmatrix} R \cos \phi \cos \theta \\ R \sin \phi \cos \theta \\ R \sin \theta \end{Bmatrix} \quad \mathbf{K}_{,k} = \begin{bmatrix} \cos \phi \cos \theta & -R \sin \phi \cos \theta & -R \cos \phi \sin \theta \\ \sin \phi \cos \theta & R \cos \phi \cos \theta & -R \sin \phi \sin \theta \\ \sin \theta & 0 & R \cos \theta \end{bmatrix}$$

$$\mathbf{K}_{,k}^{-1} = \begin{bmatrix} \cos \phi \cos \theta & \sin \phi \cos \theta & \sin \theta \\ -\frac{\sin \phi}{R \cos \theta} & -\frac{\cos \phi}{R \cos \theta} & 0 \\ -\frac{1}{R} \cos \phi \sin \theta & -\frac{1}{R} \sin \phi \sin \theta & \frac{1}{R} \cos \theta \end{bmatrix} \quad (4.56)$$

Transformation der generierten Koordinaten in das globale System:

Zuletzt verbleibt noch die Transformation der generierten Koordinaten vom System  $k$  in das globale System unter Berücksichtigung der Macro-Prozeduren. Dafür gilt die Gleichung (4.46):

$${}^0_0 \mathbf{r} = {}^m_0 \mathbf{T} \left\{ {}^n_0 \mathbf{T}^T \left( {}^k_0 \mathbf{T} \quad {}^k_0 \mathbf{r} + {}^k_0 \mathbf{r}_1 - {}^n_0 \mathbf{r}_1 \right) \right\} + {}^m_0 \mathbf{r}_1 \quad (4.46)$$

Die Ableitungen lauten:

$${}^0_0 \mathbf{r}_{,x} = {}^m_0 \mathbf{T}_{,x} \left\{ {}^n_0 \mathbf{T}^T \left( {}^k_0 \mathbf{T} \quad {}^k_0 \mathbf{r} + {}^k_0 \mathbf{r}_1 - {}^n_0 \mathbf{r}_1 \right) \right\} + {}^m_0 \mathbf{r}_{1,x}$$

$$+ {}^m_0 \mathbf{T} \left\{ {}^n_0 \mathbf{T}_{,x}^T \left( {}^k_0 \mathbf{T} \quad {}^k_0 \mathbf{r} + {}^k_0 \mathbf{r}_1 - {}^n_0 \mathbf{r}_1 \right) \right. \quad (4.47)$$

$$\left. + {}^n_0 \mathbf{T}^T \left( {}^k_0 \mathbf{T}_{,x} \quad {}^k_0 \mathbf{r} + {}^k_0 \mathbf{T} \quad {}^k_0 \mathbf{r}_{,x} + {}^k_0 \mathbf{r}_{1,x} - {}^n_0 \mathbf{r}_{1,x} \right) \right\}$$

Damit wäre die Form generiert und abgeleitet. Abschließend seien noch die Ableitungen der Formfunktion des Kreis- bzw. Spiralelements angeben.

Ableitung der Formfunktion des Kreis- und Spiralelementes:

Die Formfunktion lautet:

$$\mathbf{r}(u) = R(u) (\cos \phi(u) \mathbf{v}_1 + \sin \phi(u) \mathbf{v}_2) + \mathbf{r}_2 \quad (4.13)$$

$$\text{mit: } \mathbf{R}(u) = R_1 + (R_2 - R_1) u$$

$$\phi(u) = \alpha u$$

Für die Ableitung gilt:

$$\begin{aligned}
 \mathbf{r}_{,x}(u) &= \mathbf{R}(u) (\cos \phi(u) \mathbf{v}_{1,x} + \sin \phi(u) \mathbf{v}_{2,x}) + \mathbf{r}_{2,x} \\
 &+ \mathbf{R}_{,x}(u) (\cos \phi(u) \mathbf{v}_1 + \sin \phi(u) \mathbf{v}_2) \\
 &+ \mathbf{R}(u) (-\sin \phi(u) \mathbf{v}_1 + \cos \phi(u) \mathbf{v}_2) \phi_{,a}(u) a_{,x}
 \end{aligned} \tag{4.58}$$

$$\begin{aligned}
 \text{mit: } \mathbf{R}_{,x}(u) &= \mathbf{R}_{1,x} + (\mathbf{R}_{2,x} - \mathbf{R}_{1,x}) u \\
 \phi_{,a}(u) &= u
 \end{aligned}$$

und:

$$\begin{aligned}
 \mathbf{R}_{1,x} &= \frac{1}{R_1} (\mathbf{r}_1 - \mathbf{r}_2)^T (\mathbf{r}_{1,x} - \mathbf{r}_{2,x}) \\
 \mathbf{R}_{2,x} &= \frac{1}{R_2} (\mathbf{r}_4 - \mathbf{r}_2)^T (\mathbf{r}_{4,x} - \mathbf{r}_{2,x}) \\
 |\mathbf{r}_3 - \mathbf{r}_2|_{,x} &= \frac{1}{|\mathbf{r}_3 - \mathbf{r}_2|} (\mathbf{r}_3 - \mathbf{r}_2)^T (\mathbf{r}_{3,x} - \mathbf{r}_{2,x}) \\
 \mathbf{v}_{1,x} &= \frac{1}{R_1} (\mathbf{r}_{1,x} - \mathbf{r}_{2,x} - \mathbf{v}_1 R_{1,x}) \\
 \mathbf{a}_{,x} &= \frac{1}{|\mathbf{r}_3 - \mathbf{r}_2|} (\mathbf{r}_{3,x} - \mathbf{r}_{2,x} - \mathbf{a} |\mathbf{r}_3 - \mathbf{r}_2|_{,x}) \\
 \mathbf{v}_{2,x} &= \mathbf{a}_{,x} \times \mathbf{v}_1 + \mathbf{a} \times \mathbf{v}_{1,x}
 \end{aligned}$$

Zur Definition der angegebenen Größen vergleiche mit den Bildern 4.14 und 4.15 in Abschnitt 4.3.2.2.4.

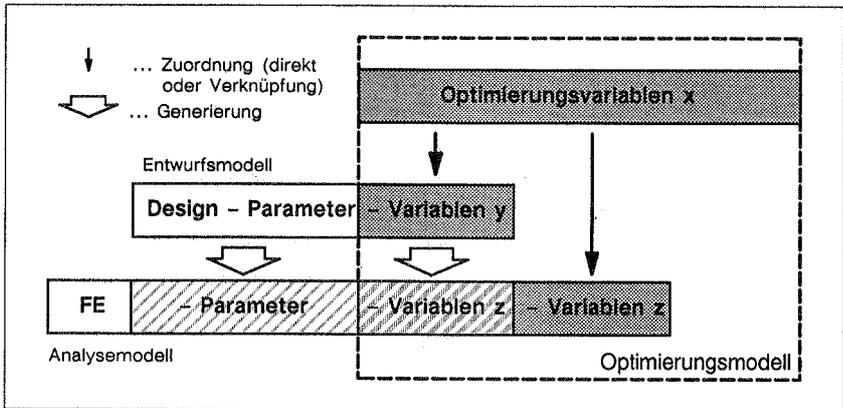


## 5 VARIABLENVERKNÜPFUNGEN

### 5.1 Optimierungsmodell und Variablen

Strukturoptimierung ist zu einem guten Teil die Fügung der Teilaufgaben mathematische Programmierung, Entwurfsbeschreibung, Strukturanalyse sowie Sensitivitätsanalyse zu einer funktionalen Einheit. Bild 1.1 vermittelt einen Eindruck von der Komplexität dieser Aufgabe. Neben der programmtechnischen Verwirklichung und der Datenbewältigung müssen die Variableninteraktionen der verschiedenen Teilmodelle ausformuliert werden. Dabei ist eine Hierarchie festzustellen (Bild 5.1):

- 1.) Optimierungsvariablen  $x$
- 2.) Entwurfparameter, Entwurfsvariablen  $y$  (z.B. Koor. der Designknoten)
- 3.) Strukturparameter, generierte Strukturparameter, Strukturvariablen  $z$  (z.B. Koor. der FE-Knoten)



**Bild 5.1:** Hierarchie der Variablen im Optimierungsmodell (vgl. mit Bild 1.1)

Variablen sind in dieser Definition Variablen im Sinne der Optimierung. Parameter sind zusätzliche, feste Werte, die zur endgültigen Beschreibung des jeweiligen Modells benötigt werden, und generierte Parameter sind Werte, die z.B. mit den in Kapitel 4 beschriebenen Methoden der Formgenerierung erzeugt worden sind. Als Optimierungsvariablen  $x$  werden die im Optimierungsproblem unabhängigen Variablen bezeichnet. Ihre Zahl ist in der Regel wesentlich kleiner als die Zahl der Variablen  $y$  und  $z$  in den Ebenen darunter. Dies kommt vor allem daher, daß neben direkten Variablenzuordnungen

gen (d.h.  $z_i = x_i$  oder  $y_i = x_i$ ) auch lineare Verknüpfungen definiert sein können. Dafür gibt es mehrere Gründe:

- vorgeschriebene Symmetriebedingungen
- vorgeschriebene Beziehungen zwischen Variablen (z.B. feste Richtungen der Variablenveränderungen, gerade Bauteilränder)
- Erhalt der physikalischen Bedeutung des Analysemodells (z.B. keine degenerierten Netze, Locking)
- verbesserte Aussagekraft des Optimierungsmodells

Mit der zusätzlichen Reduktion der Zahl an Optimierungsvariablen wirken sich Verknüpfungen (linking) günstig auf die Lösungsgeschwindigkeit und -stabilität der Optimierungsalgorithmen aus. Verschiedentlich wird auch dies als Grund für die Einführung von Verknüpfungsvorschriften angeführt [144]. In der praktischen Anwendung genügen zusätzliche lineare Verknüpfungsvorschriften, so daß folgende allgemeinen Zusammenhänge definiert werden können:

$$\begin{aligned} \mathbf{z} &= \mathbf{z}^0 + \mathbf{I}_{zy}(\mathbf{y}) \\ \mathbf{y} &= \mathbf{y}^0 + \mathbf{L}_{yx} \mathbf{x} \end{aligned} \quad (5.1)$$

mit

- $\mathbf{z}$  ... allgemeine Variablen (hier: Strukturvariablen)
- $\mathbf{z}^0$  ... konstanter Teil der allgemeinen Variablen
- $\mathbf{y}$  ... Kontrollvariablen (hier: Entwurfsvariablen)
- $\mathbf{y}^0$  ... konstanter Teil der Kontrollvariablen
- $\mathbf{I}_{zy}$  ... vektorwertige Funktion der Kontrollvariablen
- $\mathbf{x}$  ... Optimierungsvariablen
- $\mathbf{L}_{yx}$  ... konstante Verknüpfungsmatrix (linking matrix)

## 5.2 Linking Levels

### Linking Level 1:

allgemeine Variablen = Optimierungsvariablen

$$\mathbf{z} = \mathbf{x}$$

(5.2)

In diesem Fall liegt z.B. eine direkte Verknüpfung der Strukturvariablen mit den Optimierungsvariablen vor. Ein Entwurfsmodell ist nicht definiert. Dieser Fall entspricht einfachen Optimierungsproblemen, z.B. der Strukturoptimierung von Fachwerken.

Linking Level 2a:

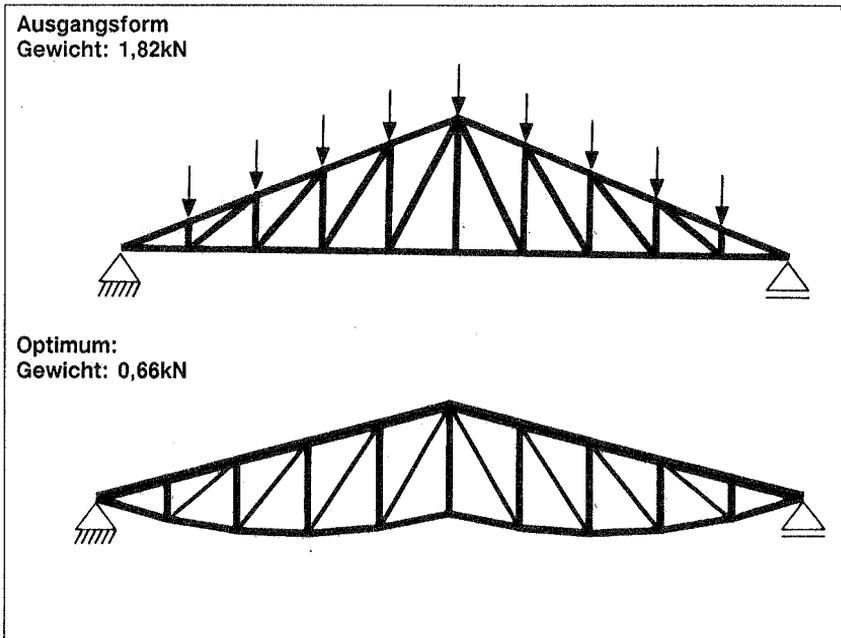
Kontrollvariablen = Optimierungsvariablen

$$y = x$$

$z$  ist eine Linearkombination von  $x$

$$z = z^0 + L_{zx} x \quad (5.3)$$

Auch hier ist kein Entwurfsmodell definiert, jedoch liegen lineare Verknüpfungsvorschriften der Optimierungsvariablen vor. Dies entspricht Fällen der Fachwerksoptimierung mit zusätzlichen Bedingungen, wie Symmetrie der Formänderung oder vorgeschriebenen Tragwerksformen (z.B. gerade Obergurte, Bild 5.2)



**Bild 5.2:** *Fachwerkträger mit geraden Obergurten*

Linking Level 2b:

Kontrollvariablen = Optimierungsvariablen

$$y = x$$

z ist eine nichtlineare Funktion von x

$$z = z^0 + l_{zy}(x) \quad (5.4)$$

Dieser Fall ist in CARAT nicht vorgesehen und wird nur der Vollständigkeit halber angegeben.

Linking Level 3a:

z ist eine Linearkombination von y

y ist eine Linearkombination von x

$$z = z^0 + L_{zy}(y^0 + L_{yx}x) \quad (5.5)$$

Die üblichsten Fälle der Formoptimierung entsprechen dieser Verknüpfungsvorschrift. Dann geht (5.5) in Gleichung (4.4) über.  $L_{zy}$  kann dann als die Formfunktionen der Designelemente identifiziert werden, die i. a. als Funktionen der Parameter u und v unabhängig von den Koordinaten der Designknoten sind.  $z^0$  verschwindet dann. Weiter sind Verknüpfungen zwischen den variablen Designknoten möglich, z.B. vorgeschriebene Richtungen oder Symmetriebedingungen. Diese Regeln werden in der Linking-Matrix  $L_{yx}$  vermerkt.

Linking Level 3b:

z ist eine Funktion von y

y ist eine Linearkombination von x

$$z = z^0 + l_{zy}(y^0 + L_{yx}x) \quad (5.6)$$

Dieser im Rahmen der getroffenen Definitionen (5.1) allgemeinste Fall tritt im Falle von Linking Level 3a auf, wenn die Formfunktion des Kreiselements verwendet wird, oder wenn bei der Formgenerierung eine Transformation mit Kugel- und Zylinderkoordinaten vorgenommen wird (Gleichung 4.44).

### 5.3 Sensibilitätsanalyse

Die Verknüpfungsvorschriften müssen auch bei der Sensibilitätsanalyse berücksichtigt werden, indem ein weiteres Glied in der Kettenregel hinzugefügt wird. Für Linking Level 3a folgt:

$$z_{,x} = L_{zy} L_{yx} \quad (5.7)$$

und für Linking Level 3b:

$$z_{,x} = L_{zy,y} (y) L_{yx} \quad (5.8)$$

### 5.4 Vorgeschriebene Bewegungsrichtungen

Besonders in der Formoptimierung ist es wichtig, die Variablenänderungen während der Prozedur sorgfältig zu kontrollieren. Insbesondere bei der Konturoptimierung von ebenen Tragwerken besteht die Gefahr, daß das Analysemodell durch eine unvorhergesehene Änderung variabler Designkoordinaten entartet oder sich unerwünschte Begleiterscheinungen über Gebühr bemerkbar machen. In Bild 5.3 ist dies am Beispiel der Formoptimierung einer einfachen Kragsscheibe gezeigt. Das Tragwerk ist durch ein Coons-Element mit je einer Bézier-Kante oben und unten beschrieben (Bild 5.3a,b). Gesucht ist die Form minimalen Gewichts. Vorgeschrieben sind die maximal zulässige Vergleichsspannung, die Länge der Scheibe und die Auflagerhöhe. Bilder 5.3c und d zeigen die vom Optimierungsalgorithmus ermittelte optimale jedoch entartete Form der Scheibe infolge unzulässig großer horizontaler Lageänderungen der Kantenmittelpunkte. Die finiten Elemente verhalten sich bei dieser Geometrie viel zu steif und verfälschen die Strukturantwort.

In diesem Fall wären diese Entartungen einfach zu vermeiden gewesen, indem nur die vertikalen Koordinaten als Variablen zugelassen worden wären. Im allgemeinen jedoch, entlang gekrümmter Kanten, sind vorgeschriebene Bewegungsrichtungen notwendig (Bild 5.4). Dabei ergibt sich folgende Verknüpfungsvorschrift:

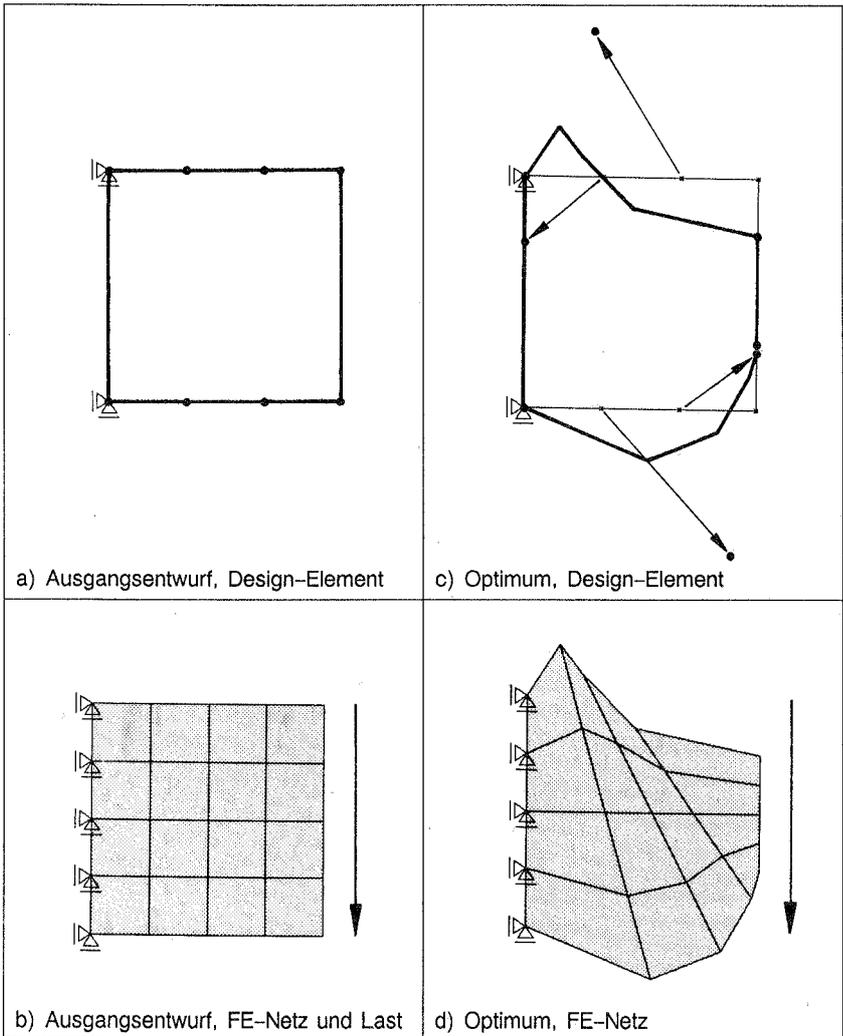
$$\mathbf{r}_i = \begin{Bmatrix} y_j \\ y_{j+1} \\ y_{j+2} \end{Bmatrix} = \begin{Bmatrix} y_j^0 \\ y_{j+1}^0 \\ y_{j+2}^0 \end{Bmatrix} + \begin{Bmatrix} s_1 \\ s_2 \\ s_3 \end{Bmatrix} x_k \quad (5.9)$$

mit  $y_j, y_{j+1}, y_{j+2}$  ...variable Koordinaten des Designknotens  $\mathbf{r}_i$

$y_j^0, y_{j+1}^0, y_{j+2}^0$ ...Koordinaten der Startposition  $\mathbf{r}_i^0$

$s_1, s_2, s_3$  ...Komponenten der vorgeschriebenen Richtung

$x_k$  ...k-te Optimierungsvariable



**Bild 5.3:** Entartetes FE-Netz

An Gleichung (5.9) wird die Bedeutung der konstanten Teile der Variablen deutlich. In Verbindung mit Verknüpfungsvorschriften beziehen sich die Optimierungsvariablen auf die Ausgangsform. Sind keine Verknüpfungsvorschriften vorhanden, ist dies nicht nötig.

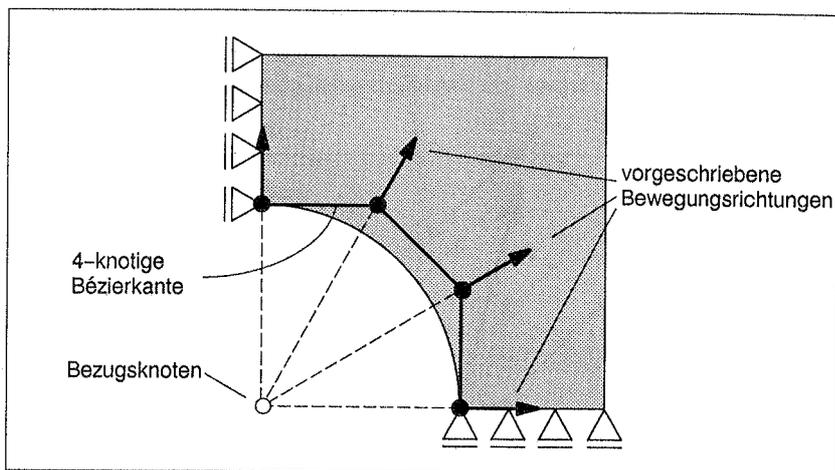


Bild 5.4: vorgeschriebene Bewegungsrichtungen

### 5.5 Symmetriebedingungen

Oft sind symmetrische Formänderungen gewünscht. Z.B. sollen sich die Ober- und Unterkanten der in Bild 5.3 dargestellten Scheibe symmetrisch bezüglich der Scheibenmittellinie ändern. Die Symmetriebedingung bezieht sich auf bereits definierte Bewegungsrichtungen und Verknüpfungsvorschriften.

Sie lautet: (siehe Bild 5.5):

$$\mathbf{s}' = \mathbf{s} - 2(\mathbf{n}^T \mathbf{s}) \mathbf{n} = (\mathbf{I} - 2\mathbf{n}\mathbf{n}^T) \mathbf{s} \quad (5.10)$$

mit den bereits vorliegenden Verknüpfungsvorschriften  $\mathbf{s} = \mathbf{L}_s \mathbf{x}$

folgt

$$\mathbf{s}' = (\mathbf{I} - 2\mathbf{n}\mathbf{n}^T) \mathbf{L}_s \mathbf{x}$$

womit sich eine Modifikation der Linking Matrix ergibt:

$$\mathbf{L}_{ij(\text{mod})} = \left[ 1 - \sum_k (2n_k n_k^T \mathbf{L}_{kj}) \right] \mathbf{L}_{ij}$$

dabei ist:

- $\mathbf{s}$  ... Bewegungsrichtung des Bezugsknotens
- $\mathbf{s}'$  ... gespiegelte Bewegungsrichtung für neue Knoten
- $\mathbf{n}$  ... Normalenvektor auf die Spiegelebene
- $\mathbf{L}_s$  ... Teil der Linking Matrix  $\mathbf{L}$
- $\mathbf{x}$  ... Optimierungsvariablen

Es ist dabei zu beachten, daß nur die Bewegungsrichtung, nicht jedoch der absolute Ort des Knotens, gespiegelt wird. D.h. an der Startlösung unsymmetrische Strukturen werden auch durch die Vorgabe von Symmetrieverknüpfungen während der Optimierung nicht symmetrisch.

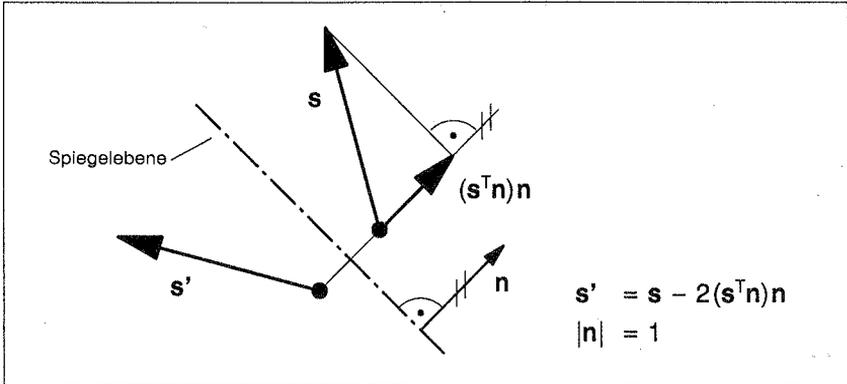


Bild 5.5: Symmetriebedingung

### 5.6 Linearkombinationen

Eine weitere, wichtige Verknüpfungsvorschrift von Optimierungsvariablen ergibt sich aus der Kombination bereits bestehender Vorschriften. Diese Verknüpfung läßt sich außer für Koordinaten auch für andere Variablentypen anwenden. Eine typische Anwendung ergibt sich in der Formoptimierung, wenn Designknoten im "Inneren" des beschriebenen Bereichs sich infolge einer Kantenveränderung mitbewegen müssen. Die Vorschrift lautet (siehe Bild 5.6):

$$s_i = \sum_k (\alpha_k s_k) \quad (5.11)$$

mit den bereits vorliegenden Verknüpfungsvorschriften  $s_k = L_k x$

$$s_i = \sum_k (\alpha_k L_k x)$$

womit sich eine Modifikation der Linking Matrix ergibt:

$$L_{ij(\text{mod})} = \sum_k (\alpha_k L_{kj})$$

dabei sind:  $\alpha_j$  ... Skalierungsfaktoren  
 $s_i$  ... kombinierte Richtungen  
 $s_j$  ... vorliegende Richtungen

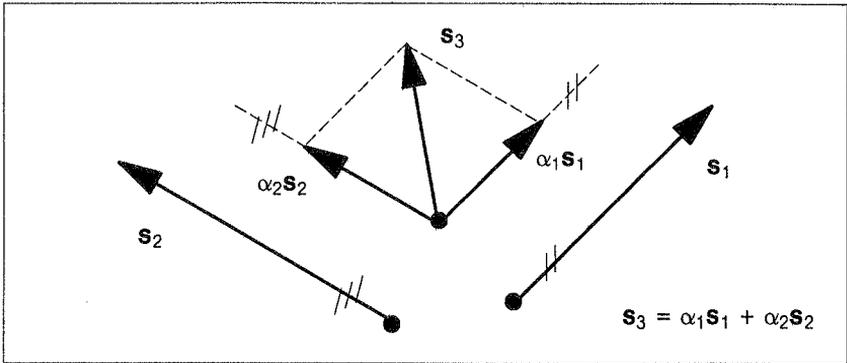


Bild 5.6: *Linearkombination*

Bild 5.7 zeigt eine Anwendung dieser Verknüpfungsvorschrift. Durch eine Verknüpfung der horizontalen Koordinaten der Designknoten im Inneren und am unteren Rand der dargestellten Struktur mit den vertikalen Koordinaten der freien Designknoten am oberen Rand, kann das generierte Netz der finiten Elemente während der Optimierung auch im Bereich der Ecke regelmäßig gehalten werden. Yao und Choi weisen in [155] darauf hin, daß dies mit den in Kapitel 4 beschriebenen Formgenerierungsmethoden nicht möglich ist. Mit der angegebenen Verknüpfungsvorschrift ist dieses Defizit überwunden.

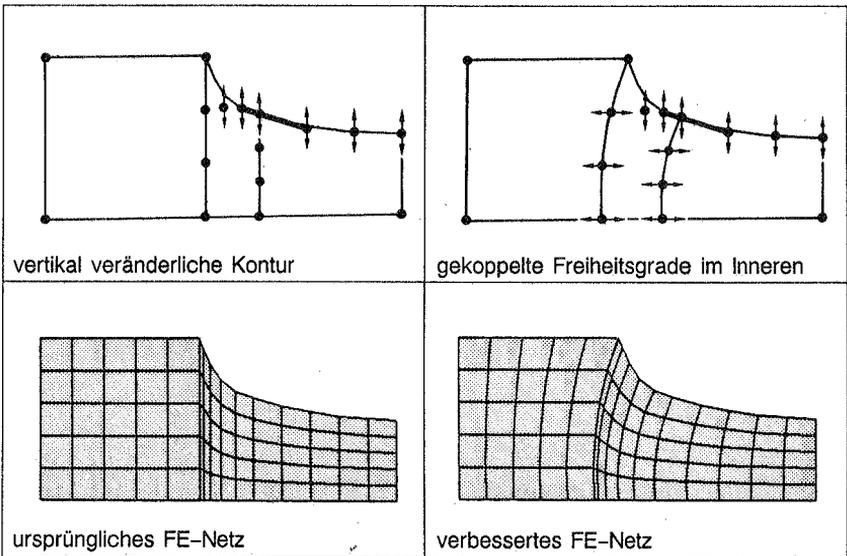


Bild 5.7: *Anwendung der Linearkombination zur Verbesserung von FE-Netzen*

## 5.7 Restriktionen der unabhängigen Variablen

Eine eindeutige Bestimmung der Restriktionen ist nur dann möglich, wenn eine direkte Zuordnung oder eine Verknüpfung einer abhängigen Variablen mit nur einer einzigen unabhängigen existiert. In diesen Fällen gilt:

$$\begin{aligned}
 x_{Lj} &= \max \left( x_{Lj}, \frac{y_{Li} - y_i^0}{L_{ij}} \right) \\
 x_{Uj} &= \min \left( x_{Uj}, \frac{y_{Ui} - y_i^0}{L_{ij}} \right)
 \end{aligned}
 \tag{5.12}$$

$y_i$  = alle von  $x_j$  abhängigen Kontrollvariablen

Liegt eine mehrfache Verknüpfung vor, werden die betreffenden Restriktionen der abhängigen Variablen als zusätzliche Nebenbedingungen eingeführt. Dies wird vom Programm automatisch erkannt und durchgeführt. Für jede mehrfach verknüpfte abhängige Variable wird zusätzlich definiert:

$$\begin{aligned}
 y_{Li} - y_i^0 - L_{ij} x_j &\leq 0 \\
 -y_{Ui} + y_i^0 + L_{ij} x_j &\leq 0
 \end{aligned}
 \tag{5.13}$$

## 5.8 Programmtechnisches

Im Regelfall liegen nur relativ wenige Verknüpfungen vor, im wesentlichen sind die Variablen einander direkt zugeordnet. D.h. die Komponenten der Linkingmatrix haben vorwiegend die Werte 0 und 1. Ein typisches Beispiel könnte so aussehen:

$$y = y^0 + L x$$

$$\begin{Bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{Bmatrix} = \begin{Bmatrix} y_1^0 \\ y_2^0 \\ y_3^0 \\ y_4^0 \\ y_5^0 \end{Bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0.5 & 1 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix}$$

Die Variablen  $y_1$ ,  $x_1$  sowie  $y_2$ ,  $x_2$  und  $y_4$ ,  $x_3$  sind direkt zugewiesen, der Wert 0 kommt überdurchschnittlich häufig vor. Von der Linkingmatrix werden ebenfalls nur die Elemente abgespeichert, die ungleich null sind und die nicht einer direkten Zuweisung entsprechen. Das betrifft in obigem Beispiel alle Elemente der Linkingmatrix mit den Werten 0 und 1, außer der 1 in der dritten Zeile. Das gewählte Speicherschema ist

in Bild 5.8 dargestellt. Dabei ist jeder abhängigen Variablen  $y_i$  ein Element NR des Zeigervektors IV zugeordnet, das zwei Informationen vermitteln kann. Ist es positiv, gibt es direkt die Nummer der zugeordneten unabhängigen Variablen  $x_j$  an. Ist es negativ, gibt es die Zeile der Linkingmatrix an, die die Koeffizienten der in diesem Falle vorliegenden Verknüpfung beinhaltet. Ist die Nummer der Zeile bekannt, kann aus dem Feld LINKROW bestimmt werden, wieviele Spalten in der betreffenden Zeile belegt sind und wo im Feld LINKCOL die Nummern der unabhängigen Variablen zu finden sind, bzw. im Feld LINKMAT die Koeffizienten der Linkingmatrix. Mit bekanntem NR läßt sich der Wert der abhängigen Variablen folgendermaßen bestimmen:

$$\begin{aligned} \text{NR} > 0: & \quad y_i = x(\text{NR}) \\ \text{NR} < 0: & \quad \text{NS} = \text{LINKROW}(-\text{NR}) + 1 \\ & \quad \text{NE} = \text{LINKROW}(-\text{NR} + 1) \end{aligned}$$

$$y_i = y_i^0 + \sum_{n=\text{NS}}^{\text{NE}} \text{LINKMAT}(n) x(\text{LINKCOL}(n))$$

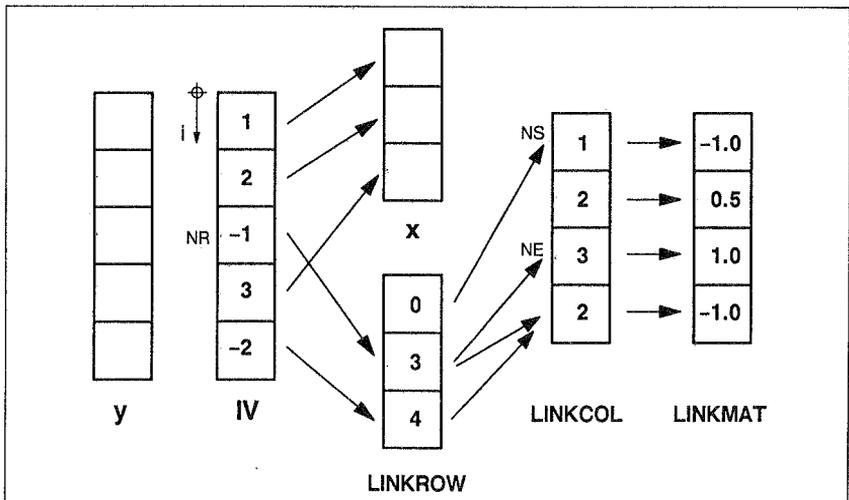


Bild 5.8: Speicherschema der Verknüpfungsmatrix und ihrer Zuordnungen

### 5.9 Definition von Verknüpfungen bei der Eingabe

Ein wesentliches Merkmal eines gut formulierten Optimierungsmodells ist, wie flexibel Verknüpfungsvorschriften definiert werden können. Da diese Vorschriften maßgeblich die Aussagekraft des Modells beeinflussen, dies auch ausdrücklich sollen, ist es wichtig, daß solche Vorschriften leicht und ohne Aufwand in das Problem eingebracht und vor allem modifiziert werden können. Es ist deshalb nicht ratsam, dabei für jedes Problem direkt den Programmcode zu beeinflussen, um so die Linkingvorschrift zu "verdrahten". Diese Option ist in CARAT auch vorhanden, jedoch wurde großen Wert darauf gelegt, daß die Standardfälle bequem im Datensatz eingegeben werden können. Für die variablen Koordinaten von Designknoten stehen z.B. folgende Befehle zur Verfügung:

```
NODE 1 shall be moved in DIRection FROM 2 TO 3
NODE 4 shall be moved LIKE 1 with FACTor 0.5
NODE 5 shall be moved LIKE 4 SYMMetry XY plane
NODE 6 COORDinate X is SUM OF  1 with FACTor 0.1 >
                                2 with FACTor 0.5 >
                                3 with FACTor 1.0  END
```

Die Eingabe erfolgt mit Kennwörtern und ist formatfrei. Verschiedene Verknüpfungsvorschriften können kombiniert werden. Die Elemente der Zeigerfelder und der Linkingmatrix werden dabei automatisch erstellt. Eine Generierung von Vorschriften ist selbstverständlich möglich. Für die anderen denkbaren Variablen (FE-Knoten, Elementvariablen, Tangenten- und Linienfaktoren, Strukturdicken) können Verknüpfungsvorschriften ebenfalls auf diese Weise eingegeben werden.

## 6. BEISPIELE

### 6.1 Minimalflächen

Eine wichtige Klasse von Bauformen, die sich hauptsächlich bei pneumatisch gestützten Membrantragwerken [95], Zelten und dünnen Schalen findet, ist die Klasse der Minimalflächen. Gemäß der strengen mathematischen Definition versteht man darunter Flächen minimalen Flächeninhalts bei vorgegebenen Rändern im Raum. In dieser Formulierung handelt es sich um eine Aufgabe der Variationsrechnung. Eine andere Definition läßt sich über die Eulersche Differentialgleichung aus der Variationsaufgabe entwickeln: Minimalflächen weisen an jeder Stelle die mittlere Krümmung null auf. Diese Formulierung geht auf Lagrange zurück. Die physikalische Analogie der Seifenhaut, deren Gleichgewichtsformen unter gleichmäßiger innerer Vorspannung ebenfalls Minimalflächen sind, ist eine weitere mögliche Art der Formulierung ([153], Haug in [95]), die gern und häufig zur numerischen Ermittlung von Minimalflächen herangezogen wird. Verfahren, mit denen Minimalflächen auf direktem Wege über die Minimierung des Flächeninhalts ermittelt werden können, werden u.a. von Gründig [57] zur Formfindung von Membran- und Netztragwerken sowie von Pramila und Virtanen [105] zur Formgestaltung von Schiffsrümpfen angegeben. Von praktischem Interesse bei der Formfindung von Lufthallen sind Formen, die sich pneumatisch ergeben. In Abweichung von der strengen mathematischen Definition werden auch diese Formen Minimalflächen genannt, da sie, bezogen auf das eingeschlossene Volumen, einen minimalen Flächeninhalt aufweisen [95].

Im Rahmen der gewählten Formfunktionen und Freiheitsgrade können mit den in Kapitel 4 beschriebenen Methoden des CAGD ebenfalls Minimalflächen ermittelt werden. Diese Formen können zur Formfindung von Schalen verwendet werden oder als Startvorgabe für eine anschließende Optimierung, die auch Belastungen, Spannungen und Verformungen mit einbezieht, herangezogen werden. Minimalflächen können außerdem bei entsprechenden Randvorgaben mögliche geometrische Grenzen der Formvielfalt aufzeigen, auf deren Basis das gewählte Optimierungsmodell beurteilt werden kann.

Die Formulierung der Aufgabe ist folgendermaßen. Vorgegeben werden:

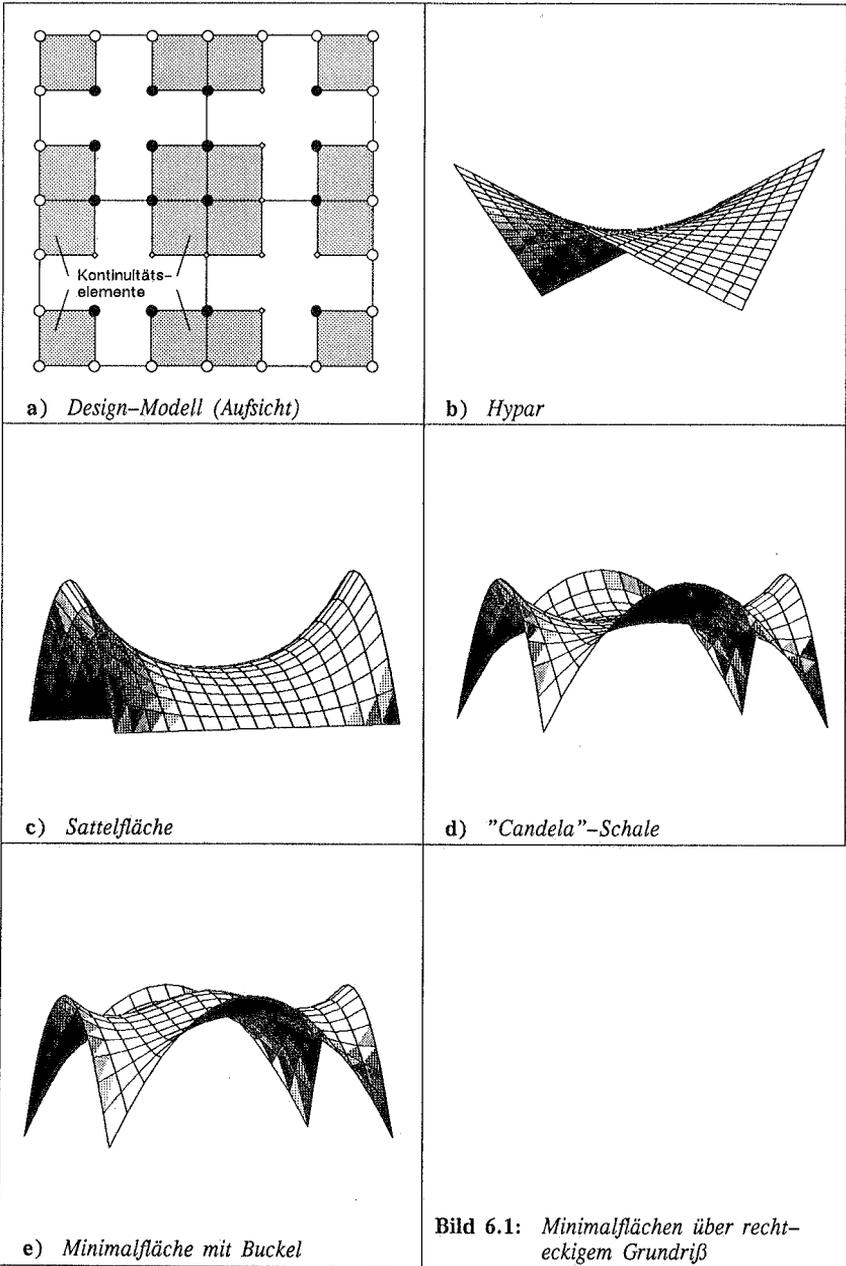
- ein vollständiges Design-Modell mit Knoten, Kanten und Elementen,
- die Art der Elemente (Coons, Bézier, Lagrange ....),
- die Kontinuitätsanforderungen (Tangenten- und Kontinuitätselemente)
- die Lage der Randkurven durch Angabe der sie definierenden Knoten.

Als Zielfunktion wird der Flächeninhalt der generierten Fläche gewählt, was dem Gewicht eines Schalentragswerks bei gegebener Dicke und Wichte entspricht. Als

Formfreiheiten können die Koordinaten der Kontrollknoten sowie Tangenten- und Linienfaktoren eingeführt werden. Mit der Vorgabe entsprechender Verknüpfungsvorschriften können die Formfreiheiten so beeinflusst werden, daß nur solche in der Formulierung berücksichtigt werden, die auch Einfluß auf die Zielfunktion besitzen. Zum Beispiel tragen Lageänderungen der Kontrollknoten, die nur zu einer Änderung der FB-Aufteilung, aber nicht zu einer Formänderung führen, nicht zu einer Reduzierung der Zielfunktion bei und sollten nicht erlaubt sein.

In Bild 6.1a wird der quadratische Grundriß einer Minimalfläche gezeigt. Ihre Form wird dargestellt von vier Bézier-Elementen, die durch Vorgabe der entsprechenden Kontinuitäts-elemente (schraffiert)  $G_1$ -kontinuierlich aneinander gekoppelt sind. Vorgegeben sind die vertikalen Koordinaten der Randknoten (Kreise), variabel sind die vertikalen Koordinaten der durch ausgefüllte schwarze Punkte gekennzeichneten Knoten. Es verbleiben Knoten, deren Lageänderungen aufgrund der gewählten Übergangsbedingungen mit denjenigen der freien Knoten verknüpft sind. Auf eventuelle Symmetriebedingungen wurde verzichtet, die eine Reduzierung der Zahl an Optimierungsvariablen ermöglicht hätten, so daß sich eine unbeschränkte Optimierungsaufgabe mit insgesamt 16 Variablen ergibt. Die Vorgabe linearer, geneigter Randkurven führt zu der bekannten Form des hyperbolischen Paraboloids (Bild 6.1b). Paarweise gerade und parabolische Randkurven ergeben die in Bild 6.1c dargestellte Sattelfläche, und vier parabelförmige Randkurven führen zu der in Bild 6.1d gezeigten Form. Wird zusätzlich die Stichhöhe vorgeschrieben, ergibt sich eine Form, die materialisiert als ein Schalentragwerk in Frage kommen könnte (Bild 6.1e), da sie an jeder Stelle genügend große Krümmungen aufweist, um lokale Beulerscheinungen zu vermeiden.

Da dieses Beispiel eine unbeschränkte Optimierungsaufgabe ist, eignet es sich besonders, um den Effekt der in Abschnitt 2.8.2.2 geschilderten Skalierungstechniken auf die Konvergenz von Quasi-Newton-Algorithmen mit BFGS-Korrektur zu demonstrieren. Für den Fall 6.1d sind in Tafel 6.1 die Anzahl an Iterationen und Funktionsauswertungen zu verschiedenen Skalierungen und ihre Kombinationen angegeben. In jedem Fall wurde die gleiche Lösungsgenauigkeit gefordert, weshalb die unterschiedlichen Optimalwerte auf ein sehr flaches Optimum hindeuten. Eingesetzt wurde das SQP-Verfahren nach Schittkowski [117] mit einem Armijo-line search (Sekantenfaktor  $10^{-4}$ ) und quadratischer Interpolation. Mit und ohne Skalierung der Variablen, als Skalierungswerte wurden die inversen Startwerte eingesetzt, erweist sich die automatische Skalierung der Hesse-Matrix nach dem Vorschlag von Luenberger als sehr erfolgreich. Die Skalierung der Zielfunktion mit dem vorgeschlagenen Faktor von Gleichung (2.45) liefert in Verbindung mit der Variablenskalierung einen hervorragenden Wert, wie die geringe Zahl an Iterationsschritten und Funktionsauswertungen belegt. Hingegen scheint der



gemäß Gleichung (2.45) ermittelte Faktor zu klein auszufallen, wenn die Variablen nicht skaliert werden. Deshalb sollte die Skalierung der Zielfunktion mit diesem Faktor nur als ein Vorschlag des Verfahrens verstanden werden, der mit der Erfahrung verglichen und dementsprechend angewendet werden sollte.

V	H	Z	ITER	NFUNC	F <sub>opt</sub>
x			19	31	173,61
x	x		15	19	172,35
x		x	14	15	174,67
			22	26	172,40
	x		19	24	172,11
		x	28	29	172,55

V ... Variablenskalerung                      ITER ... Zahl der Iterationen  
H ... Skalierung der Hesse-Matrix      NFUNC ... Zahl der Funktionsauswertungen  
Z ... Skalierung der Zielfunktion      F<sub>opt</sub> ... optimaler Zielfunktionswert

**Tafel 6.1:** die Auswirkungen verschiedener Skalierungen (Beispiel 1)

Auch bei der Formfindung von Schalen über unregelmäßigen Grundrissen kann die Ermittlung von Minimalflächen sehr hilfreich sein. Zu der in Bild 4.3 gezeigten Schale, deren Form bei vorgegebenem Grundriß, vorgeschriebenen Randkurven und Raumhöhe frei bestimmt wurde, soll eine Minimalfläche bestimmt werden. Die Fläche wird von insgesamt 16 bikubischen Bézier-Elementen beschrieben (Bild 6.2). Aufgrund mehrfacher Symmetrien und geforderter G<sub>1</sub>-kontinuierlicher Elementübergänge verbleiben 7 freie Kontrollknoten, deren vertikale Koordinaten als Unbekannte in das Formfindungsproblem eingehen. Diese Knoten sind in Bild 6.2 hervorgehoben. Als Minimalfläche ergibt sich eine Form mit aufgebörtelten Rändern (Bild 6.3), die wegen dieser Randversteifungen als ein Schalentragerwerk in Frage kommen könnte.

Rotationssymmetrische Minimalflächen können mit Coons-Elementen bestimmt werden, deren Knoten und Kanten in Zylinderkoordinaten beschrieben sind. Bild 6.4 zeigt ein solches Design-Element. Vorgegeben sind Höhe und Radius der Leitkreise. Die Mantellinie wird von 9-knotigen B-Splines beschrieben. Die r- und z-Koordinaten der hervorgehobenen Kontrollknoten sind variabel. Mit entsprechenden Verknüpfungsvorschriften (Bild 6.5) kann die mehrfache Symmetrie des Problems ausgenutzt werden, so daß 7 Variablen verbleiben. Es sind dies die r- und z-Koordinaten der Knoten 2, 3 und 4 sowie die r-Koordinate des Knotens 5. Die Minimalfläche soll sich als ein Kom-

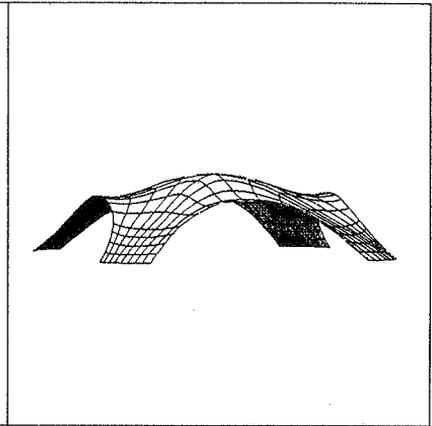
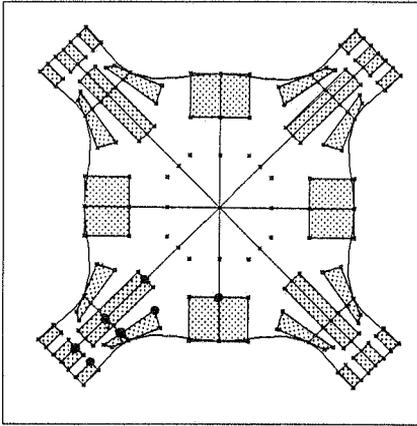
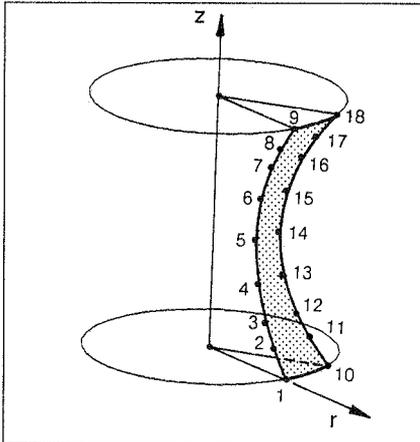


Bild 6.2: Grundriß des Design-Modells

Bild 6.3: Minimalfläche

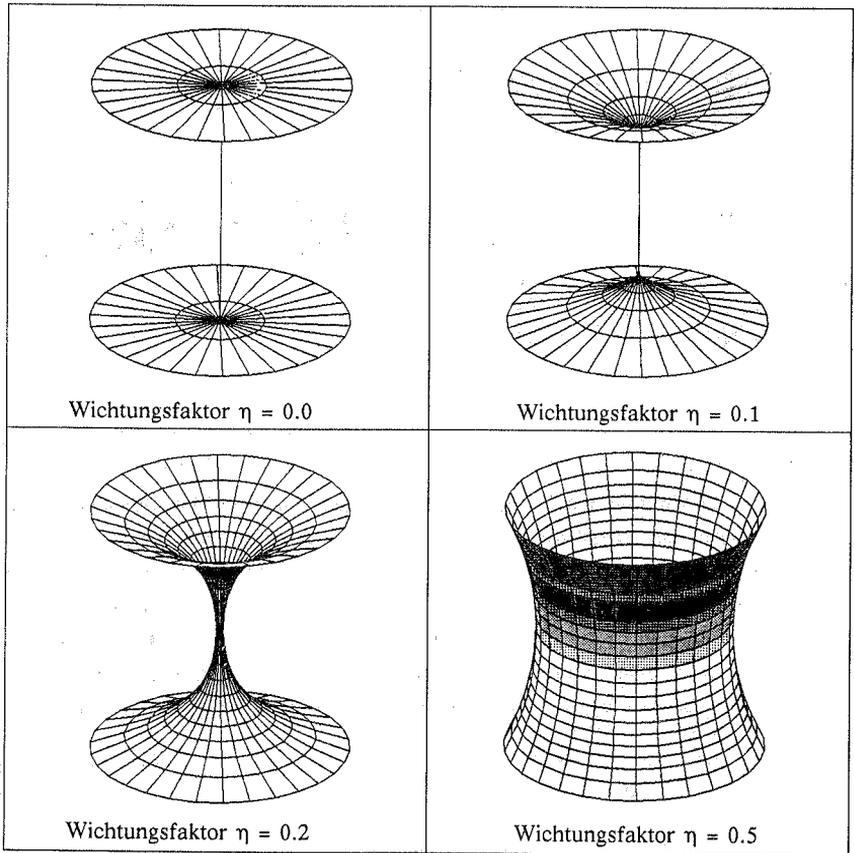
promiß aus der Minimierung der Oberfläche sowie der Länge der Mantellinie einstellen. Dies entspricht der physikalischen Analogie einer in Achsrichtung zusätzlich vorgespannten Membran und der sich so einstellenden Gleichgewichtsform. Wegen der Rotationssymmetrie genügt zur Berechnung ein kleiner Ausschnitt, z.B.  $\alpha = 30^\circ$ . Bild 6.6 zeigt verschiedene Minimalflächen, die sich in Abhängigkeit der Vorspannung, d.h. der Wichtung der beiden Beiträge zur Zielfunktion, ergeben. Angegeben sind die Wichtungsfaktoren für die Länge der Mantellinie bei konstanter Wichtung 1 der Manteloberfläche.



```

NODE 11 LIKE 2
NODE 14 LIKE 5 GEN 11,14
NODE 8 COOR R SUM OF 2 FAC 1
NODE 8 COOR Z SUM OF 2 FAC -1
NODE 7 COOR R SUM OF 3 FAC 1
NODE 7 COOR Z SUM OF 3 FAC -1
NODE 6 COOR R SUM OF 4 FAC 1
NODE 6 COOR Z SUM OF 4 FAC -1
NODE 17 COOR R SUM OF 2 FAC 1
NODE 17 COOR Z SUM OF 2 FAC -1
NODE 16 COOR R SUM OF 3 FAC 1
NODE 16 COOR Z SUM OF 3 FAC -1
NODE 15 COOR R SUM OF 4 FAC 1
NODE 15 COOR Z SUM OF 4 FAC -1 END
    
```

Bild 6.4: rotationssymmetrische Minimalfläche, Bild 6.5: Verknüpfungsvorschriften Design-Modell



**Bild 6.6:** *Minimalflächen mit variierter Vorspannung*

## 6.2 Vierendeel-Träger

### 6.2.1 Situation

Eine Halle soll von außen liegenden Querträgern, die die gesamte Dachlast abtragen, überspannt werden. Die Querträger haben eine Stützweite von 36,40 m und stehen in einem regelmäßigen Abstand von 10,80 m (Bild 6.7). Die Dachkonstruktion wird neben direkten Einleitungen indirekt über Seilverspannungen in die Obergurte der Träger eingeleitet, wobei diese eine seitliche Abstützung erfahren und ein Ausknicken der Obergurte verhindert werden soll. Es war eine Form des Trägers gesucht, die ästhetischen und technischen Ansprüchen genügt, um die bekannten Nachteile im Tragverhalten eines Vierendeel-Trägers zumindest teilweise auszugleichen. Aus zwei jeweils 22 mm dicken Stahlblechen aus St 52, die gegenseitig verschraubt werden, sollen deshalb den Kräfteverhältnissen angepasste Löcher geschnitten werden. Die Gestaltung der Löcher soll über eine Gewichtsminimierung des Trägers gefunden werden ("Ausmagern"). Als formgebender Lastfall werden Eigengewicht sowie aus dem Gewicht der Dachkonstruktion und der Schneebelastung herrührende Einzelkräfte an Ober- und Untergurt eines Trägers angenommen.

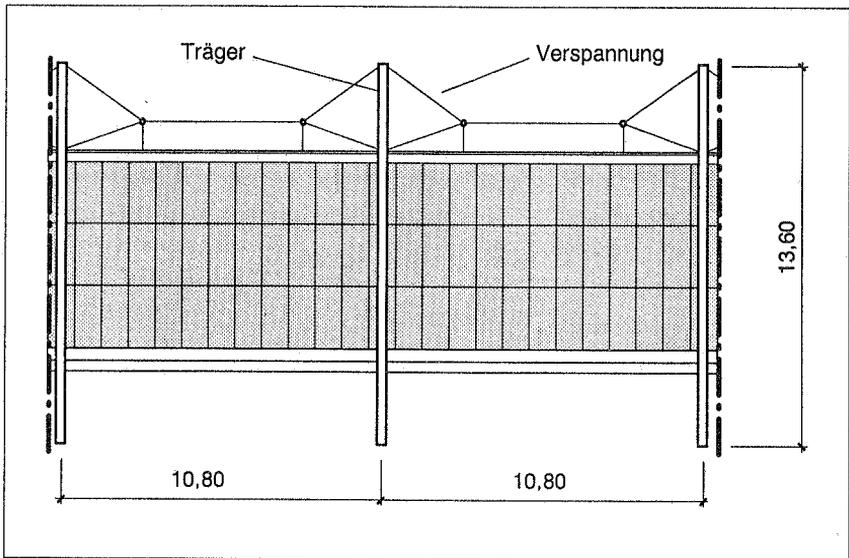


Bild 6.7: Ansicht der Halle (Ausschnitt)

## 6.2.2 Modellierung

### a) Design-Modell

Ein Träger soll 12 Durchbrüche haben. Aus konstruktionsbedingten Gründen sind die Achsabstände der Durchbrüche sowie die Höhe des Trägers vorgegeben. Es ergeben sich somit Felder von  $2,7 \times 2,7$  m Größe, innerhalb derer die Löcher eingeschnitten werden können (Bild 6.8). Aus Symmetriegründen genügt es, nur die Hälfte des Trägers zu modellieren. Jedes Feld wird von einem eigenen Koordinatensystem beschrieben. Felder mit einzuschneidenden Löchern bestehen aus vier Coons-Elementen. Die Lochränder werden jeweils von vier 6-knotigen B-Splines modelliert, die mittels Tangentenelementen untereinander  $G_2$ -kontinuierlich verbunden sind (Bild 6.9). Damit kann eine große Bandbreite verschiedenartiger Lochformen vom Kreis bis zu einem an den Ecken ausgerundeten Quadrat erfaßt werden (Bild 6.10). An den Auflagern sind Endfelder ohne Durchbrüche vorgesehen, die von einem einzigen Design-Element beschrieben werden.

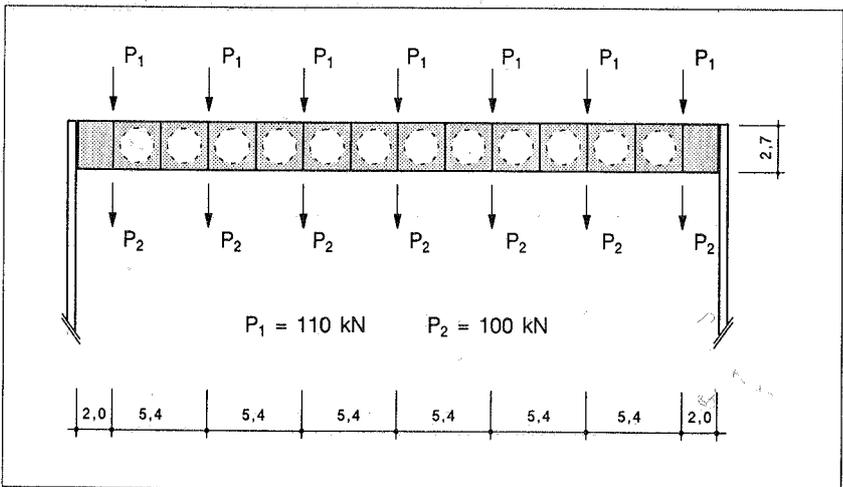
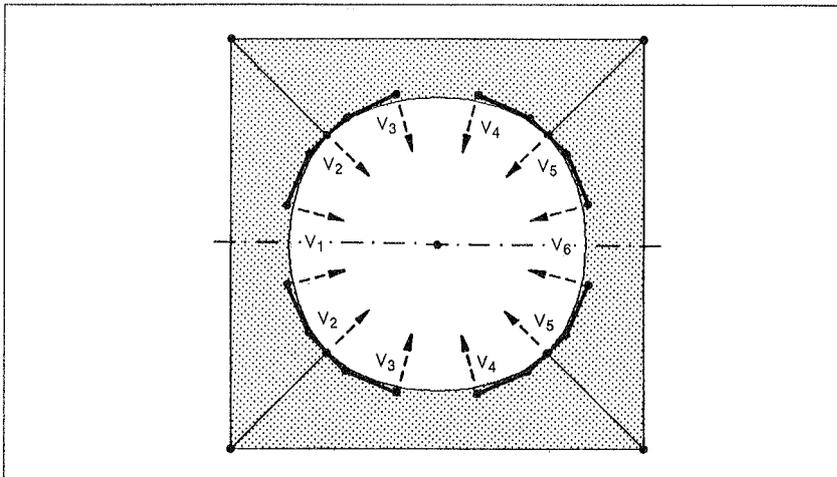


Bild 6.8: Träger, Systemskizze

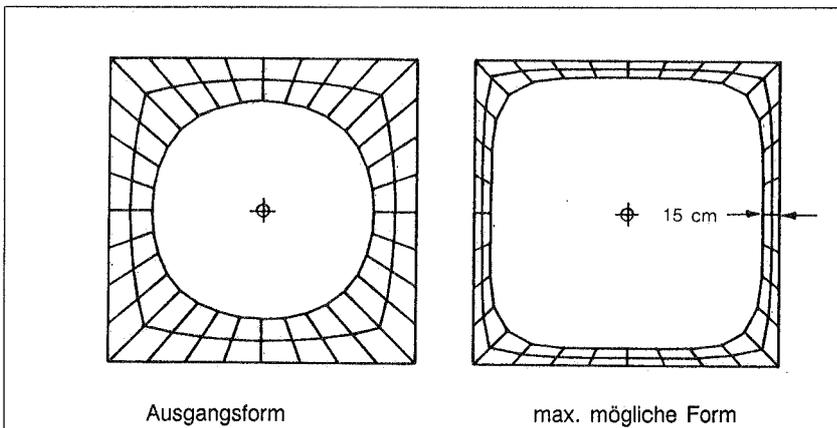
### b) FE-Modell

Da die Optimierungsaufgabe für Probleme der Formoptimierung viele Unbekannte hat und deshalb besonders die Sensitivitätsanalyse zur Bestimmung der Gradienten von Zielfunktion und Nebenbedingungen sehr rechenintensiv ist, muß das FE-Netz mit möglichst wenig Freiheitsgraden auskommen. Zusätzlich wirkt sich die mehrfach zusammenhängende Topologie trotz einer Bandbreitenoptimierung auf die Größe der Stei-

figkeitsmatrix und die Geschwindigkeit der Gleichungslösung ungünstig aus. Gewählt werden deshalb vierknotige isoparametrische Schalenelemente mit der in Bild 6.12 dargestellten Vernetzung. Das Auflager wird von drei zusätzlichen Elementspalten mit verringertem Elastizitätsmodul modelliert, um die vorab bestimmte, jedoch sehr geringe Einspannwirkung der Stützen (Bild 6.8) zu simulieren.



**Bild 6.9:** Design-Modell eines Lochfeldes (mit Angabe der Optimierungsvariablen)



**Bild 6.10:** mögliche Formvariationen eines Loches

### c) Optimierungsmodell

Gesucht ist eine optimal ausgemagerte Konstruktion. Als Zielfunktion kommt deshalb das Gewicht des Trägers in Frage. An keiner Stelle des Tragwerks sollen die zulässigen Spannungen und in der Trägermitte die zulässige Durchbiegung ( $l/300 = 12 \text{ cm}$ ) überschritten werden. Da meist zweiaxiale Spannungszustände vorherrschen, werden entlang aller Lochränder sowie an den Rändern von Ober- und Untergurt die von Mises-Vergleichsspannungen sowie in der Mitte der Lochstege die Schubspannungen kontrolliert. Da zu erwarten ist, daß die gewählten vierknotigen finiten Elemente sich zu steif verhalten und deshalb zu niedrige Spannungen ermittelt werden, und um Reserven für getrennt zu führende Stabilitätsnachweise zu bewahren, werden für die Vergleichsspannung der zulässige Wert  $\sigma_v = 200 \text{ N/mm}^2$  und für die Schubspannung  $\tau = 80 \text{ N/mm}^2$  eingesetzt.

Die Form der Löcher soll stets symmetrisch zur Längsachse des Trägers sein und sich radial bezüglich des Lochmittelpunktes aufweiten bzw. einschnüren können. Dies wird gefordert, damit die Anzahl an Optimierungsvariablen auf ein vertretbares Maß reduziert werden kann. Außerdem ist für den betrachteten Lastfall zu erwarten, daß die Beanspruchungen von Ober- und Untergurt infolge Normal- und Querkräften sowie Biegemomenten ungefähr dieselben sind, was bei der Vorgabe derselben zulässigen Spannungen für Druck- wie Zugbereiche ohnehin zu symmetrischen Löchern führen würde. Die Forderung der radialen Formveränderlichkeit ist keine nennenswerte Einschränkung der Formfreiheiten. Unter Berücksichtigung dieser Forderungen, die als Verknüpfungsvorschrift verwirklicht werden können, und der  $G_2$ -Kontinuität der Lochränder mittels der Tangentenelemente, verbleiben für jedes Loch 6 freie Kontrollknoten, die die Lage der B-Splines bestimmen (Bild 6.9). Insgesamt hat das Optimierungsproblem damit 36 Variablen.

Aus gestalterischen Gründen sollen die Durchbrüche konvex sein. Um dies während der Optimierung garantieren zu können, müssen zusätzliche geometrische Nebenbedingungen eingeführt werden. Dabei können die günstigen Eigenschaften des charakteristischen Polygons eines B-Splines ausgenützt und diese Nebenbedingungen für die Kontrollknoten anstelle der generierten Form formuliert werden. Die Konvexität des charakteristischen Polygons und damit die der generierten Kurve ist dann gesichert, wenn die Flächeninhalte eines jeden der in Bild 6.11 dargestellten Dreiecke gemäß der Gaußschen Definition positiv sind.

Zum Beispiel ergibt sich für ein solches Dreieck der Flächeninhalt zu (vgl. Bild 6.11):

$$A = \frac{1}{2} [(\mathbf{r}_2 - \mathbf{r}_1) \times (\mathbf{r}_3 - \mathbf{r}_1)]^T \mathbf{n}$$

$$A = \frac{1}{2} [x_1 y_2 - x_2 y_1 + x_2 y_3 - x_3 y_2 + x_3 y_1 - x_1 y_3]$$

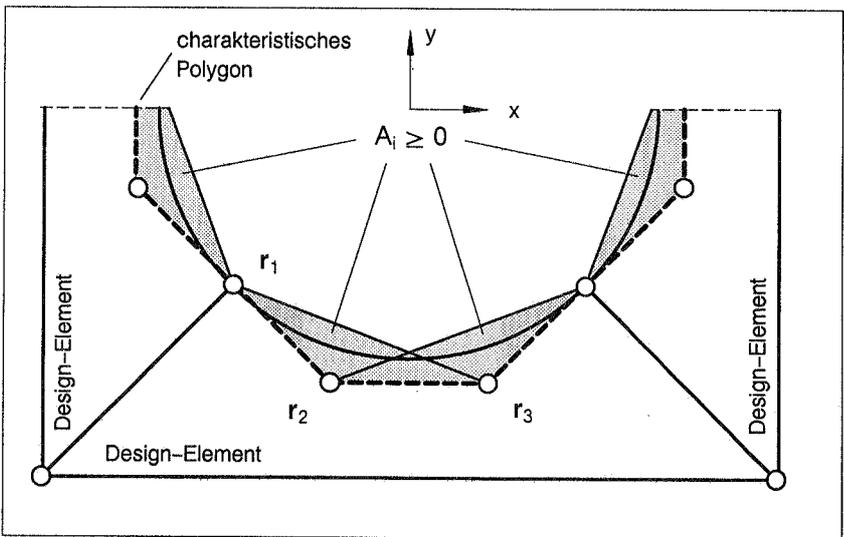
mit:  $\mathbf{n}^T = (0, 0, 1)^T$  für die dargestellte Situation in der x-y-Ebene

Für jedes "Lochfeld" sind zusätzlich vier solcher geometrischer Nebenbedingungen der Form:

$$\hat{g}_j = A_j \geq 0$$

bzw.  $\hat{g}_j = -A_j \leq 0$

zu formulieren und in das Optimierungsproblem einzubringen.



**Bild 6.11:** Nebenbedingungen für konvexe Lochränder,  $A_i \geq 0$

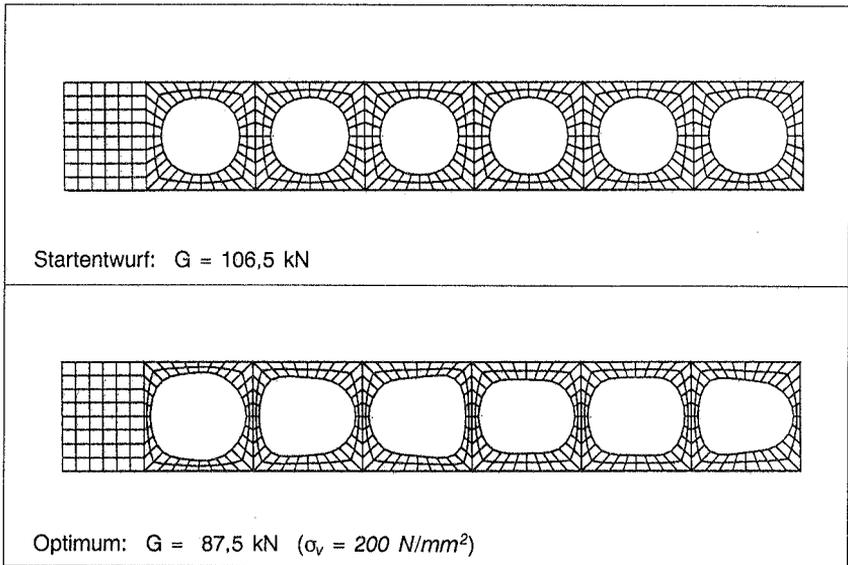
Gemäß der gewählten Vernetzung ergeben sich somit für jedes "Lochfeld" 48 Nebenbedingungen zur Kontrolle der Vergleichsspannungen entlang der Lochkontur und am Ober- und Untergurt, eine Nebenbedingung zur Kontrolle der Schubspannungen im Steg sowie vier geometrische Bedingungen zur Kontrolle der Lochform. Zusammen mit zusätzlichen Nebenbedingungen zur Kontrolle der Vergleichsspannungen im End-

feld sowie in Trägermitte und einer Bedingung zur Kontrolle der zulässigen Verschiebung ergeben sich somit insgesamt 347 Nebenbedingungen, die erfüllt werden müssen.

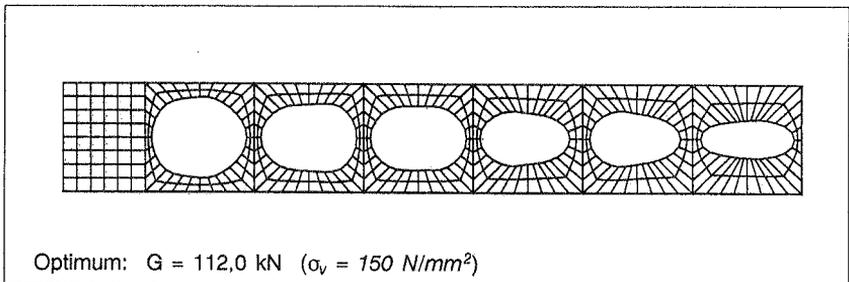
Es ist für Aufgaben der Formoptimierung typisch, daß die Zahl der Nebenbedingungen diejenige der Unbekannten weit übersteigt. Hier kommen auf jede Optimierungsvariable ungefähr 10 Nebenbedingungen. Am Optimum können jedoch nur höchstens soviel Nebenbedingungen erfüllt werden, wie es Unbekannte gibt. Wären also die an der Lösung aktiven Nebenbedingungen bekannt, könnte der Formulierungs- und Lösungsaufwand stark reduziert werden. Die Erfahrung zeigt jedoch, daß Optimierungsalgorithmen die Lücken unvollständig formulierter Aufgaben sehr schnell erkennen und somit Tragwerksformen ermitteln, die aus vielen Gründen nicht akzeptabel sind. Sei es, daß an anderen Stellen die Spannungen überschritten werden, die Form aus technischer Sicht unzulässig wird oder entartet und infolge davon ein stark verzerrtes oder gar bereits blockierendes FE-Netz falsche Strukturantworten liefert. Bei diesem Beispiel zeigt es sich, daß während des Optimierungsvorganges über die gesamte Struktur hinweg von Iterationsschritt zu Iterationsschritt immer wieder andere Nebenbedingungen aktiv werden. Erst gegen Ende der Iteration stabilisiert sich die Auswahl des aktiven Satzes der Nebenbedingungen, und es schälen sich die tatsächlich maßgeblichen Nebenbedingungen heraus. Eine vorweg genommene Auswahl ist nicht möglich.

### 6.2.3 Ergebnisse

Das Ergebnis der Formoptimierung (Bild 6.12) spiegelt im wesentlichen die typischen Schnittgrößenverläufe eines Vierendeel-Trägers wieder: In Tragwerksmitte erfordern die großen Normalkräfte in den Gurten große Querschnitte im ganzen Feld, gegen die Auflager hin nehmen die Normalkräfte ab und die Querkräfte und Biegemomente zu. Es entwickeln sich deutlich angevoutete Gurtquerschnitte. Die Stege magern an der Tragwerkslängsachse auf ihr zulässiges Minimum (30 cm Breite) ab; an dieser Stelle müssen sie im wesentlichen nur Querkräfte übertragen. Die geringen Unterschiede der eingeleiteten Kräfte an Ober- und Untergurt wirken sich nicht nennenswert in Form von Normalspannungen aus. Die zu erkennenden Gesetzmäßigkeiten prägen sich noch stärker aus, wenn die zulässigen Vergleichsspannungen weiter auf  $\sigma_v = 150 \text{ N/mm}^2$  verringert werden (Bild 6.13). Die zulässige maximale Verschiebung in Trägermitte ist in beiden Fällen eingehalten. Kritisch für die Entwürfe sind die zulässigen Vergleichsspannungen in Trägermitte sowie an den stärksten Umlenkungen infolge der eingeschnittenen Löcher. Ebenso erkennt man an einigen Lochformen, daß die Ränder wegen der eingeführten geometrischen Bedingungen abschnittsweise geradlinige Verläufe haben. Hier sind diese Nebenbedingungen kritisch.



**Bild 6.12:** Ausgangsentwurf und optimale Form für  $\sigma_v = 200 \text{ N/mm}^2$



**Bild 6.13:** optimale Form für  $\sigma_v = 150 \text{ N/mm}^2$

Die Optimierung wurde mit dem SQP-Verfahren nach Schittkowski durchgeführt (siehe Abschnitt 2.8.6.3). Das Problem erweist sich als hervorragend konditioniert, wie ein Blick auf den Iterationsverlauf zeigt (Bild 6.14). Bei der vorgeschriebenen Genauigkeit von  $\epsilon = 10^{-3}$  konvergiert das Verfahren innerhalb von nur 20 Schritten, das entspricht nicht einmal einem ganzen Zyklus, wie er zur Lösung eines allgemeinen quadratischen Problems notwendig gewesen wäre. Dies läßt auf geringe Kopplungen der Variablen schließen, weshalb auch das Verfahren der beweglichen Asymptoten wahrscheinlich erfolgreich hätte eingesetzt werden können. Allerdings verrät der Iterationsverlauf nichts über die bei nahezu konstantem Gewicht noch vor sich gehenden Formänderun-

gen. Das Gewicht des Trägers wird gegenüber der Startlösung ( $G = 106,5 \text{ kN}$ ) für die erste Lösung (Bild 6.12) auf  $G = 87,5 \text{ kN}$  reduziert. Dabei weist der Startentwurf trotz des höheren Gewichts an der schmalsten Stelle der Gurte in Trägermitte Vergleichsspannungen auf, die um 10 % den zulässigen Wert übertreffen. Die anschließende Optimierung mit dem verminderten zulässigen Wert der Vergleichsspannung vergrößert das Gewicht auf  $G = 112 \text{ kN}$  (zugehörige Lösung, Bild 6.13).

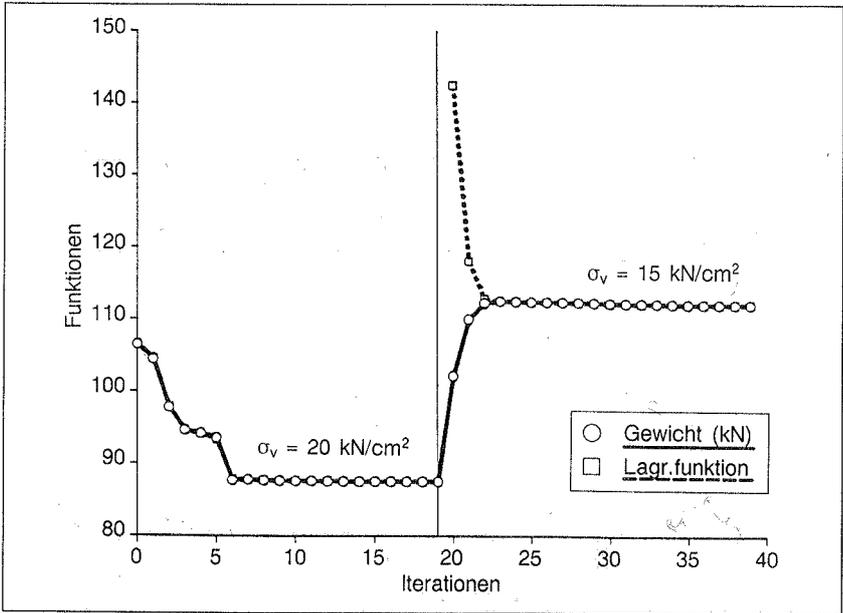


Bild 6.14: Vierendeel-Träger, Iterationsverläufe

## 6.3 Formoptimierung rotationssymmetrischer Schalen

### 6.3.1 Problemstellung

Wichtige, weil oft gebaute Schalentypen sind rotationssymmetrische Schalen. Die Form wird meist aus den bekannten geometrischen Grundformen Zylinder, Kugel, Kegel oder Torus zusammengesetzt. Seltener wird sie mittels einer Optimierung nach statischen Gesichtspunkten ermittelt. (Kneppel bestimmt in [78] optimale Formen von Reflektoren an Radioteleskopen mit einer Mehrkriterienoptimierung (Gewicht, Formtreue).) Dabei ist dies eine Fragestellung, die schon seit langer Zeit besteht. Nachweislich zum ersten Male wurde sie 1670 von Hooke beantwortet, der die umgekehrte Kettenlinie als die statisch günstigste Form eines nur durch sein Eigengewicht belasteten Bogens erkannte. Die Arbeit von Graefe [54] erwähnt neben diesem historischen Ereignis weitere historische Anwendungen, wie die Idee der Kettenlinie und deren Umkehrform nicht nur zur Formfindung von ebenen Bögen, sondern auch von räumlichen Schalentragwerken eingesetzt wurde. Die prominentesten sind die Zwischenkuppeln von St. Pauls in London (1708) und Saint Geneviève in Paris (1790), die die schwere Last der Laterne abzutragen haben. Dabei übersah der französische Baumeister Jean-Baptiste Rondelet die dominante Last der Laterne, die anstelle der für Eigengewicht richtigen Kettenlinie eine mehr kegelförmige Form der Schale erfordert hätte, wie sie auch das englische Bauwerk aufweist. Daß die so ermittelte "falsche" Form dennoch in der Lage ist, die konzentriert am Scheitel eingeleiteten Kräfte sicher abzutragen, liegt neben der günstigen zweiaxialen Tragwirkung von doppeltgekrümmten Schalen, die im unteren Teil allerdings von großen Durchbrüchen gestört werden, an der geschickt einbezogenen Mitwirkung von zusätzlichen Balastgewichten und Ringankern [133].

In der Tat erlaubt die räumliche Tragwirkung einer doppelt gekrümmten Schale eine unbegrenzte Vielfalt möglicher Formen, die die aufgebrachten Lasten ausschließlich durch Membrankräfte abzutragen in der Lage sind. Zwangspunkte sind dabei die Randbedingungen, die fordern, daß alle Kräfte, d.h. Belastung und Lagerkräfte, membrangerecht eingeleitet werden müssen. Solche Formen können mittels einer Formoptimierung gefunden werden. Das eigentliche Ziel, die Biegung in der Schale auf ein Minimum zu reduzieren bzw. ganz zu vermeiden, sofern dies möglich ist, kann dabei auf zwei verschiedenen Wegen erreicht werden. Sie führen allerdings in der Klasse der "Membranformen" zu verschiedenen Ergebnissen. Die Minimierung des Gewichts bei freier Form und freien Querschnittsdicken und mit vorgegebenen zulässigen Spannungen ergibt zwangsläufig Formen, die weitest möglich membranbeansprucht sind, da versucht wird, die Querschnitte möglichst voll auszunutzen. Dies ist nur für Membrankräfte möglich. Dagegen können mit der Minimierung der Formänderungsenergie bezüglich der freien Form bzw. Querschnitte möglichst verformungsarme, d.h. steife Formen gefun-

den werden. Dies sind ebenfalls membranbeanspruchte Schalenformen, wenn die Randbedingungen dies zulassen. Allerdings führt dieses Optimierungsziel zu übermäßigen Materialanlagerungen, womit die Steifigkeit, aber auch das Gewicht immer weiter ansteigen und unrealistische Entwürfe liefern würden. Hier taucht die Dualität von Dehnungsenergie und Gewicht der Michell-Tragwerke wieder auf, die sich in fast allen Problemen der Formoptimierung wiederfindet. Die Begrenzung der zur Verfügung stehenden Bauwerksmasse wirkt dieser Entwicklung entgegen und sollte als Nebenbedingung eingeführt werden. Eine andere Möglichkeit ist, die Querschnittsdicke nicht als Optimierungsvariable zu definieren.

Vor dem bis hier geschilderten Hintergrund soll die Fragestellung Rondelets wieder aufgegriffen werden und optimale Formen von Rotationsschalen für verschiedene Lasten u.a. Laternenlast, Eigengewicht, Schnee und Wind sowie verschiedene Lagerungsbedingungen gefunden werden [14], [19]. Für den Lastfall Wind soll die optimale Form von der Rotationssymmetrie abweichen dürfen.

Die Minimierungsziele Gewicht und Formänderungsenergie sollen nur als Mittel zum Zweck dienen, die Absolutwerte sind nur von zweitrangiger Bedeutung. Variabel soll neben der Form im Falle der Gewichtsminimierung die Dicke der Schale sein. Zur Reduzierung der Zahl an Unbekannten wird angenommen, daß die Dicke im ganzen Tragwerk überall gleich groß sein soll. Damit können Biegebeanspruchungen zwar nicht vermieden, aber weitestgehend reduziert werden. An die Gestaltsfunktion, die zur geometrischen Beschreibung der Schalenform verwendet werden soll, wird vor allem die Forderung gestellt, daß sie die membrangerechten Randbedingungen erfüllen können muß.

### 6.3.2 Modellbeschreibung

Ausgangsform ist eine Kugelschale mit einem Radius  $R_K = 10$  m sowie einer Scheitelöffnung mit Radius  $R_S = 2,5$  m. Das Tragwerk kann mit einem einzigen Coons-Element in Zylinderkoordinaten beschrieben werden. Dabei sind beliebig große Ausschnitte möglich (Bild 6.15). Die Fläche wird in Meridianrichtung von 6-knotigen B-Spline-Kanten beschrieben, im Umfangsrichtung genügen 2-knotige Lagrange-Kanten, die zur Interpolation in Zylinderkoordinaten ausreichen. Bild 6.16 zeigt die möglichen Formfreiheiten dieser geometrischen Beschreibung mit 4 freien Knoten. Dieses Modell kommt zur Optimierung der Tragwerke mit rotationssymmetrischer Last in Frage: Laternenlast, Eigengewicht und Schnee.

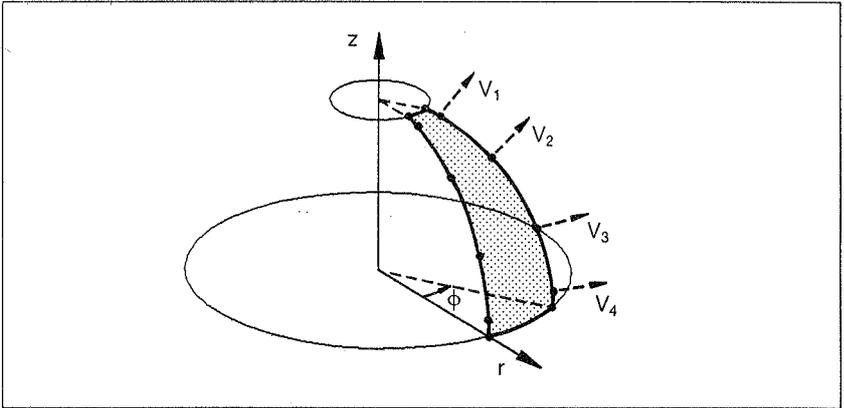


Bild 6.15: Coons-Element zur Beschreibung der Rotationsschalen

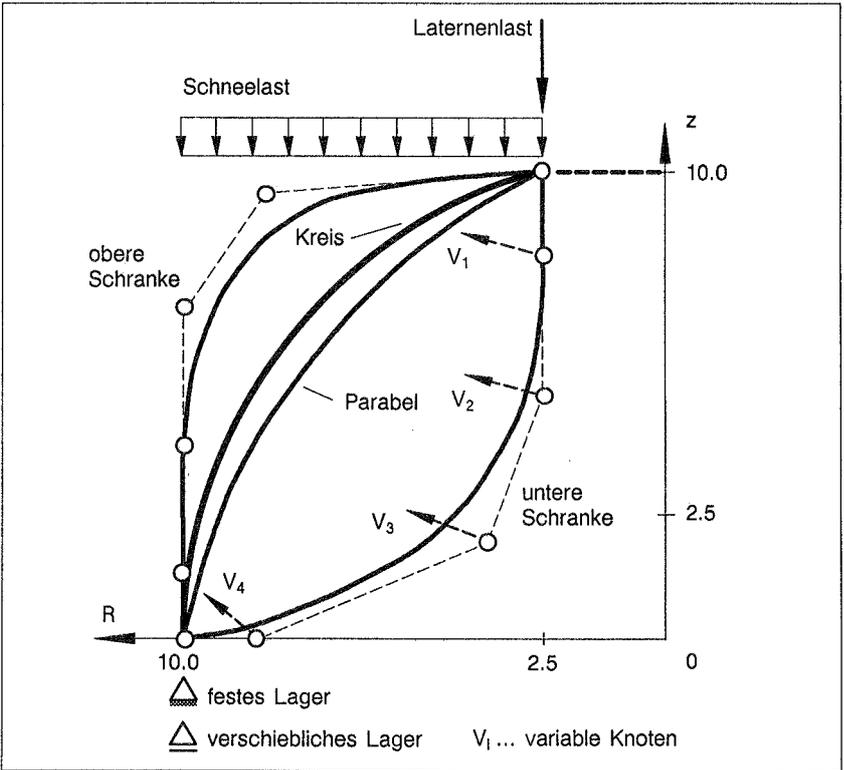
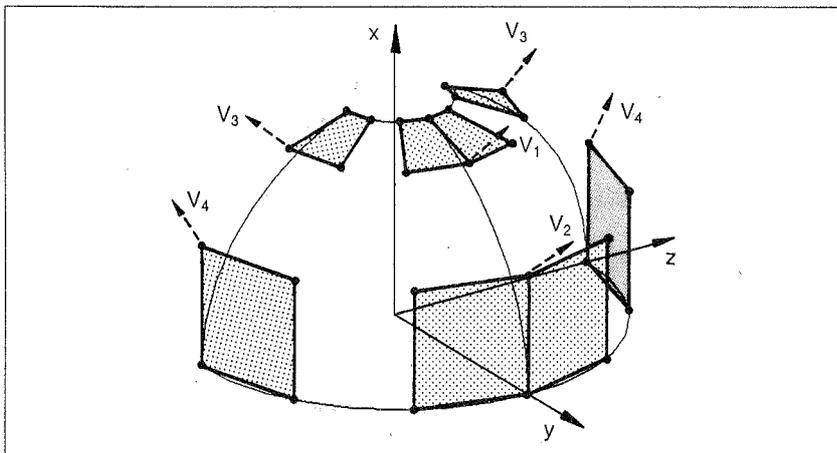


Bild 6.16: Formfreiheiten der Rotationsschale (Schnitt durch eine Meridianebene)

Zur Formoptimierung bezüglich Windbeanspruchung werden zwei Bézier-Elemente verwendet, die die halbe Struktur beschreiben. Einfachheit halber werden hier auch nur vier Formvariablen zugelassen (Bild 6.17). Eine Besonderheit der hier gewählten Darstellung ist, daß auch die Bézier-Elemente sowie die  $G_1$ -Kontinuitätsbedingungen in Zylinderkoordinaten formuliert sind. Das bedeutet, daß die Kontinuitätselemente in Umfangsrichtung mit einem Knick dargestellt werden. Die  $G_1$ -Kontinuität der resultierenden Form bleibt jedoch erhalten.



**Bild 6.17:** zwei Bézier-Elemente zur Beschreibung der halben Rotationschale

Als Material wird Stahl St 37 gewählt:

$$E = 210\,000 \text{ N/mm}^2$$

$$\nu = 0,3$$

$$\gamma = 78,5 \text{ kN/m}^3$$

$$\sigma_V = 160 \text{ N/m}^2$$

Die Lastintensitäten sind:

Laternenlast:  $p = 25,5 \text{ kN/m}$

Schnee:  $s = 0,75 \text{ kN/m}^2$

Wind  $w = 0,72 \text{ kN/m}^2$

(Druck und Sog)

### 6.3.3 Ergebnisse

Die Ergebnisse sind in Tafel 6.2 zusammengefaßt. In den Bildern 6.18a–e sind die optimalen Formen, wie sie sich für die verschiedene Belastungen und Lagerbedingungen ergeben, dargestellt. In allen Fällen passen sich die Formen den gegebenen Randbedingungen so gut es geht an. Unter der Laternenlast sowie an den verschieblichen Auflagern stellen sich die membrangerechten vertikalen Tangenten ein. Bei festen Auflagern stützen sich die Schalen direkt in den Untergrund ab. Außer im Falle der Windbelastung ergeben sich somit Formen, die im Rahmen der gewählten Geometriebeschreibung und Diskretisierung nur mit Hilfe von Membrankräften abtragen. Für den Lastfall a) "Laterne" mit verschieblichen Auflagern sind in Bild 6.19 die Schnittgrößen für einen 10°-Ausschnitt dargestellt. Im Vergleich zwischen Startentwurf (Kugel) und Optimum erkennt man eine deutliche Reduzierung der Momente sowie der Normalkraftbeanspruchung in Meridianrichtung, wie es auch erwünscht war.

Nr.	Lastfall	Lagerung	Zielfunktion	Verbesserung [%]	Variablen		Verfahren	Iterationen
					Koor.	Dicke		
a	Laterne	verschieblich	Gewicht	86,0	4	1	MMA	10
b	Laterne	fest	Gewicht	75,5	4	1	MMA	12
c	Eigen-gewicht	verschieblich	Form-änderungs-energie	3,5 *)	4	0	SQP	3 *)
d	Schnee	verschieblich	Gewicht	63,1	4	1	MMA	15
e	Wind	fest	Form-änderungs-energie	88,3	4	0	SQP	9

\*) voroptimierter Startentwurf

Tafel 6.2: Ergebnisse der Formoptimierung von Beispiel 3

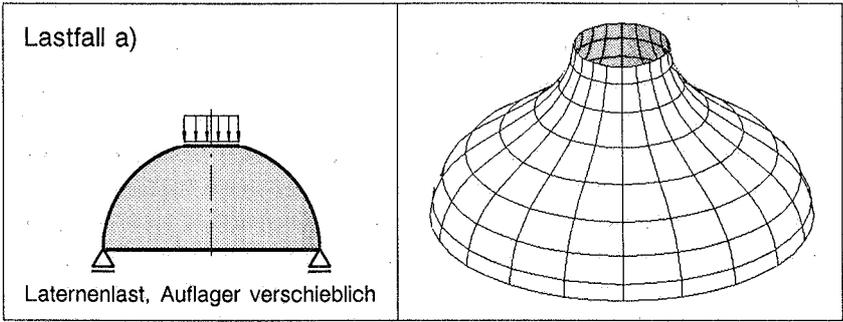


Bild 6.18a: System und optimale Form; Lastfall a), Laternenlast

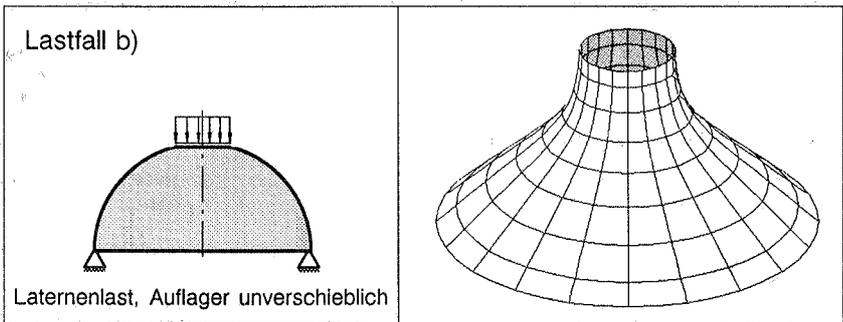


Bild 6.18b: System und optimale Form; Lastfall b), Laternenlast

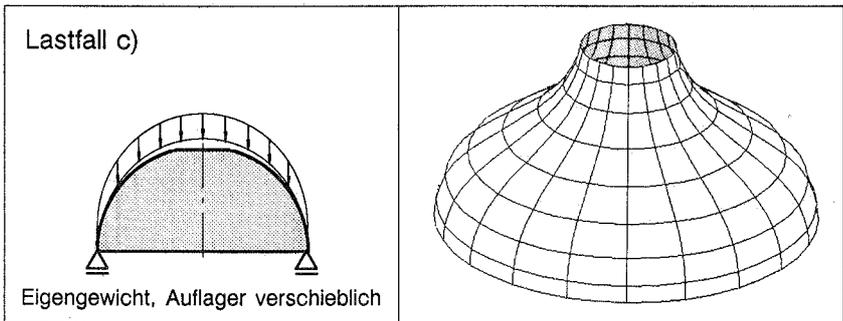
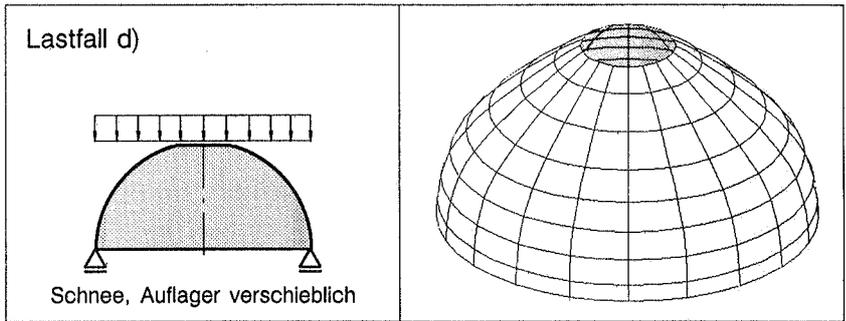
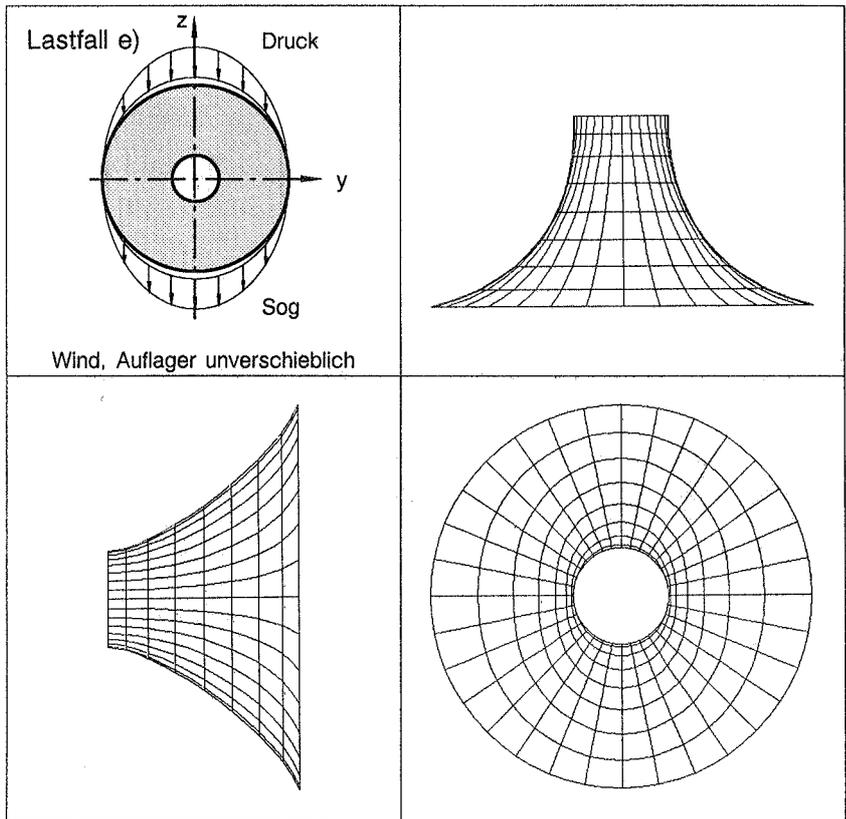


Bild 6.18c: System und optimale Form; Lastfall c), Eigengewicht



**Bild 6.18d:** System und optimale Form; Lastfall d), Schneelast



**Bild 6.18e:** System und optimale Form; Lastfall e), Wind

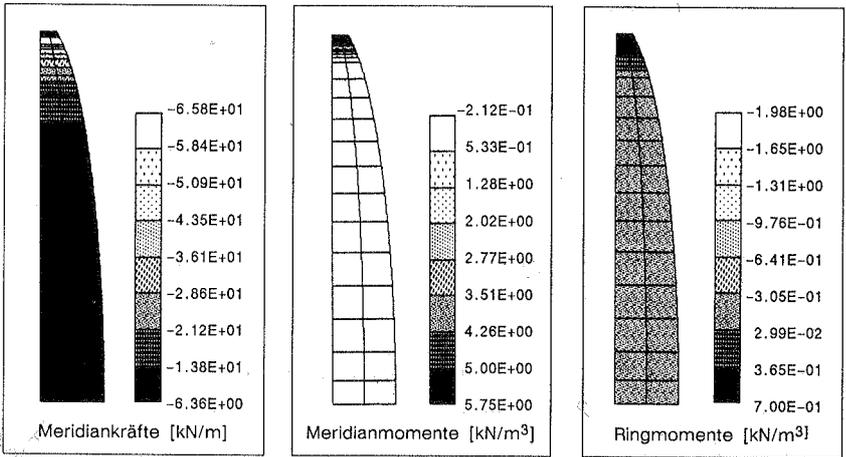


Bild 6.19a: Schnittkräfte der Kugelkappe, Lastfall a), Laternenlast

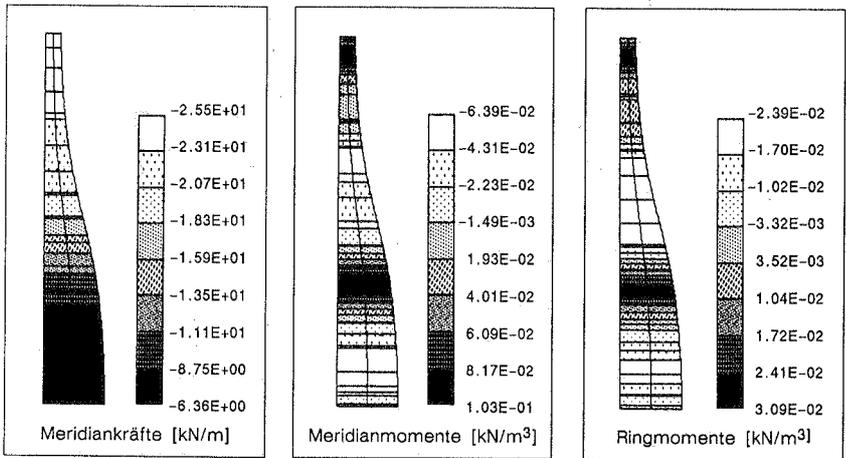


Bild 6.19b: Schnittkräfte der optimalen Form, Lastfall a), Laternenlast

## 7. SCHLUSSFOLGERUNGEN

Probleme der Formoptimierung von Tragwerken (besonders von Flächentragwerken) sind sehr komplex. Zur Lösung werden Verfahren aus den Bereichen

- der finiten Elemente (zur Tragwerksberechnung)
- der mathematischen Programmierung (zur Optimierung)
- des Computer Aided Geometric Design, CAGD (zur Geometriebeschreibung)

zusammen eingesetzt. Jeder Bereich stellt dabei ein eigenes, in sich abgeschlossenes Forschungsgebiet dar, weshalb besonders auf die Interaktionen geachtet werden muß. Neben der treffenden physikalischen Formulierung der Optimierungsaufgabe, müssen deshalb auch von Beginn an die technischen Grenzen der Methoden berücksichtigt werden. Besonders bei der Formoptimierung kommt es darauf an, Formulierungen zu finden, die garantieren, daß während des automatisch ablaufenden Optimierungsvorganges die finiten Element-Netze stets aussagekräftig bleiben und nicht infolge übermäßig stark verzerrter Geometrien falsche Strukturantworten liefern. Die Lücken unvollständig formulierter Optimierungsaufgaben werden von den Algorithmen sofort erkannt. Die Folge sind physikalisch oder technisch unsinnige Ergebnisse. Deshalb ist es notwendig, schon bei der Aufgabenstellung die Vorzüge der Verfahren aufeinander abzustimmen und richtig einzusetzen. Dennoch sollte das Optimierungsmodell einfach sein und mit wenigen Variablen auskommen, die trotzdem die Vielfalt der Formen nicht übermäßig einschränken. Dies kann mit den Methoden des CAGD sowie geeigneter Variablenverknüpfungen erreicht werden. (Kapitel 4 und 5)

Ein weiterer wichtiger, technischer Aspekt ist die Effizienz des gesamten Lösungsverfahrens. Während eines Optimierungsvorganges wird das Tragwerksverhalten wiederholt analysiert. Insbesondere ist dabei die Bestimmung der Tragwerkssensibilität sehr rechenaufwendig und dominiert den gesamten Prozeß. Soweit es möglich ist, sollte deshalb von vornherein darauf geachtet werden, das Problem so zu formulieren, daß der gewählte Optimierungsalgorithmus mit möglichst wenigen Analysen auskommt. Dabei können besondere Eigenschaften der Probleme der Tragwerksoptimierung mit den Mitteln von Approximationstechniken berücksichtigt werden. In diesem Zusammenhang hat sich eine duale Formulierung des Optimierungsproblemens bewährt, die besonders bei beschränkten und weitgehend entkoppelten Problemen erfolgreich ist. Ansonsten erweisen sich die Methoden der sequentiellen quadratischen Programmierung als sehr geeignet. Allerdings erfordern sie genaue Kenntnisse der zugrunde liegenden Theorie, da sie sehr empfindlich auf Skalierungen der Variablen, der Zielfunktion und der Nebenbedingungen reagieren. Selbstskalierende Verfahren können zwar Anhaltspunkte

liefern, entbinden jedoch nicht von der Notwendigkeit einer problem- und theoriegerechten Formulierung. (Kapitel 2 und 3)

Werden die genannten Verfahren richtig eingesetzt, können verschiedenste Probleme der Tragwerksoptimierung gelöst werden. Dabei ermöglicht die Kombination der abstrakten, rein mathematisch formulierten Verfahren der mathematischen Programmierung und des CAGD die Formulierung beinahe aller denkbaren, physikalisch sinnvollen Zielfunktionen und Nebenbedingungen. Dies wurde demonstriert bei der Ermittlung von Minimalflächen und optimaler Schalenformen. Bei letzteren wurde die Zielfunktionen minimales Gewicht und minimale Formänderungsenergie zusammen mit maximal zulässigen von Mises-Vergleichsspannungen eingesetzt. Die optimale, ausgemagerte Form eines Vierendeel-Trägers wurde ebenfalls mit der Zielfunktion minimales Gewicht und maximal zulässige von Mises-Vergleichsspannungen sowie einer maximal zulässigen Verformung in Trägermitte ermittelt. In allen Fällen ergeben sich Formen, die sich den Beanspruchungen optimal anpassen und die Interaktionen von Kraft und Form sichtbar machen. (Kapitel 6)

Angewendet werden die Verfahren heute hauptsächlich in der Luft- und Raumfahrt sowie im Maschinen- und Automobilbau. Die Verfeinerung der Berechnungs- und Herstellungsmethoden berechtigt aber zur Hoffnung, daß die Methoden der Strukturoptimierung sich zukünftig auch im Bauwesen noch mehr bewähren werden, wobei im Einzelfall die technologischen und ökonomischen Einflußfaktoren zu berücksichtigen sind. Unabhängig davon, gewähren die Methoden tiefe, prinzipielle Einblicke in die Zusammenhänge von Kraft und Tragwerksformen, die ihre Existenz als Verfahren zur Durchführung numerischer Experimente begründen. Auf diese Weise können bessere, den Kraftverläufen auch mehrerer Lastfälle angepasste Tragwerksformen gefunden werden, die im Idealfall auch als ästhetisch befriedigend und "natürlich" empfunden werden. In diesem Sinne werden in Zukunft auch die Verfahren der Sensibilitätsanalyse vermehrt an eigenständiger Bedeutung gewinnen.

## LITERATUR

- [1] Arora, J.S.: Introduction to Optimum Design; McGraw-Hill; New York (1989).
- [2] Atrek, E.; Gallagher, R.H.; Ragsdell, K.M.; Zienkiewicz, O.C.: New Directions in Optimum Structural Design; J. Wiley & sons, Chichester, New York (1984).
- [3] Baier, H.: Mathematische Programmierung von Tragwerken insbesondere bei mehrfachen Zielen; Diss., Darmstadt (1978).
- [4] Bazaraa, M.S.; Shetty, C.M.: Nonlinear Programming - Theory and Algorithm; Wiley & sons, New York (1979).
- [5] Barsky, B.A.: Computer Graphics and Geometric Modeling Using Beta-Splines; Springer-Verlag, Berlin, Heidelberg (1988).
- [6] Bartels, R.H.; Beatly, J.C.; Barsky, B.A.: An Introduction to Splines for Use in Computer Graphics and Geometric Modeling; Morgan Kaufmann Publishers, Los Altos, California (1987).
- [7] Bathe, K.-J.: Finite Element Procedures in Engineering Analysis; Prentice-Hall, London (1982).
- [8] Belegundu, A.D., Arora, J.S.: A Study of Mathematical Programming Methods for Structural Optimization, Part I: Theory, Part II: Numerical Results; Int. J. Num. Meth. Engng., **21**, 1583-1623 (1985).
- [9] Bennett, J.A.; Botkin, M.E. (eds.): The Optimum Shape-Automated Structural Design; Plenum Press, New York, London (1986).
- [10] Berke, L.; Khot, N.S.: Structural Optimization using Optimality Criteria; in [91], 271-311 (1987).
- [11] Bletzinger, K.-U.; Kimmich, S.: Strukturoptimierung; Konzeptheft 7, SFB 230, Stuttgart (1985).
- [12] Bletzinger, K.-U.: Optimale Tragwerke und CAD; Mitteilungen des SFB 230, Heft 2, Stuttgart (1988).
- [13] Bletzinger, K.-U.: Finite Elements and Structural Optimization - some Aspects of Problem Formulation; in [36], 64-70 (1989).
- [14] Bletzinger, K.-U.; Kimmich, S.; Ramm, E.: Interactive Shape Optimization of Shells; in: Pande, G.N.; Middleton, J. (eds.): NUMETA 90, Numerical Methods in Engineering - Theory and Applications; Elsevier Applied Science, London, New York (1990).
- [15] Böhm, W.; Farin, G.; Kahmann, J.: A Survey of Curve and Surface Methods in CAGD; Comp. Aided Geom. Des., **1**, 1-60 (1984).

- [16] Boggs, P.T.; Tolle, J.W.: A Family of Descent Functions for Constrained Optimization; *SIAM, J. Numer. Anal.*, **21**, 6, 1146–1161 (1984).
- [17] Boggs, P.T.; Tolle, J.W.: A Strategy for Global Convergence in a Sequential Quadratic Programming Algorithm; *SIAM, J. Numer. Anal.*, **26**, 3, 600–623 (1989).
- [18] de Boor, C.: A Practical Guide to Splines; Springer-Verlag, New York, Heidelberg, Berlin (1978).
- [19] Bothner, K.-D.: Die Modellierung von Problemen der Form- und Querschnittsoptimierung im Ingenieurbau; Diplomarbeit, Inst. für Baustatik, Universität Stuttgart (1989).
- [20] Botkin, M.E.: Shape Optimization of Plate and Shell Structures; *AIAA Journal*, **20**, 2, 268–273 (1982).
- [21] Botkin, M.E.; Bennett, J.A.: Shape Optimization of Three-Dimensional Folded-Plate Structures; *AIAA Journal*, **23**, 11, 1804–1810 (1985).
- [22] Braibant, V.; Fleury, C.: An Approximation Concepts Approach to Shape Optimal Design; *Comp. Meth. Appl. Mech. Engng.*, **53**, 119–148 (1985).
- [23] Braibant, V.; Fleury, C.: Shape Optimal Design and CAD Oriented Formulation; *Engng. with Comp.*, **1**, 193–204 (1986).
- [24] Bremer, C.: Algorithmen zum effizienten Einsatz der Finite-Element-Methode; Diss., Universität Braunschweig (1986).
- [25] Bremicker, M.: Dekompensationsstrategie in Anwendung auf Probleme der Gestaltoptimierung; Diss., Universität Siegen, VDI-Verlag, Reihe 1, Nr. 173, Düsseldorf (1989).
- [26] Bushnell, D.: PANDA2 – Program for Minimum Weight Design of Stiffened, Composite, Locally Buckled Panels; *Comp. & Struct.*, **25**, 469–605 (1987).
- [27] Canfield, R.A.; Grandhi, R.V.; Venkayya, V.B.: Optimum Design of Structures with Multiple Constraints; *AIAA Journal*, **26**, 78–85 (1988).
- [28] Chern, J.-M.; Prager, W.: Minimum-Weight Design of Statically Determinate Trusses subject to Multiple Constraints; *Int. J. Solids and Structures*, **7**, 931–940 (1971).
- [29] Dennis, J.E.; Moré, J.J.: Quasi-Newton Methods, Motivation and Theory; *SIAM Review*, **19**, 46–89 (1977).

- [30] DeSalvo, G.; Gorman, R.W.: ANSYS – Engineering Analysis System User's Manual; Revision 4; Swanson Analysis Systems, Houston Pennsylvania (1987).
- [31] Ding, Y.: Shape Optimization of Structures: A Literature Survey; *Comp. & Struct.*, **24**, 985–1004 (1986).
- [32] Eason, E.D.; Fenton, R.G.: A Comparison of Numerical Optimization Methods for Engineering Design; *J. Engng. Industry, Trans ASME, Series B*, **96**, 196–200 (1974).
- [33] Elsgolc, L.E.: Variationsrechnung; Hochschultaschenbücher, Bibliographisches Institut, Mannheim, Wien, Zürich (1970).
- [34] Eschenauer, H.; Olhoff, N. (eds.): Optimization Methods in Structural Design; BI-Verlag, Mannheim, Wien, Zürich (1983).
- [35] Eschenauer, H.; Post, U.; Bremicker, M.: Einsatz der Optimierungsprozedur SAPOP zur Auslegung von Bauteilkomponenten; *Bauingenieur*, **63**, 515–526 (1988).
- [36] Eschenauer, H.A.; Thierauf, G. (eds.): Discretization Methods and Structural Optimization-Procedures and Applications; Springer-Verlag, Berlin, Heidelberg (1989).
- [37] Esping, B.J.D.: A CAD Approach to the Minimum Weight Design Problem, *Int. J. Num. Meth. Engng.*, **21**, 1049–1066 (1985).
- [38] Farin, G.: Triangular Bernstein-Bézier Patches; *Comp. Aided Geom. Des.*, **3**, 83–127 (1986).
- [39] Farin, G. (ed.): Geometric Modeling: Algorithms and New Trends; SIAM, Philadelphia, Pennsylvania (1987).
- [40] Faux, I.D.; Pratt, M.J.: Computational Geometry for Design and Manufacture; Ellis Horwood Publishers, Chichester, U.K. (1979).
- [41] Field, D.A.: Mathematical Problems in Solid Modeling: A Brief Survey; in [39], 91–107 (1987).
- [42] Fleury, C.: Structural Weight Optimization by Dual Methods of Convex Programming; *Int. J. Num. Meth. Engng.*, **14**, 1761–1783 (1979).
- [43] Fleury, C.; Ramanathan, R.K.; Salama, M.; Schmit, L.A.: ACCESS Computer Programm for the Synthesis of Large Structural Systems; in [2], 541–561 (1984).
- [44] Fleury, C.; Braibant, V.: Structural Optimization – a new Dual Method using Mixed Variables; *Int. J. Num. Meth. Engng.*, **23**, 409–428 (1986).

- [45] Fleury, C.: Efficient Approximation Concepts using second Order Informations; *Int. J. Num. Meth. Engng.*, **28**, 2041–2058 (1989).
- [46] Fleury, C.; Smaoui, H.: Convex Approximation Strategies in Structural Optimization; in [36], 118–126 (1989).
- [47] Franke, H.J.: *Duale Methoden der Tragwerksoptimierung*; Diplomarbeit, Institut für Baustatik, Universität Stuttgart (1988).
- [48] Gallagher, R.H.; Yamada, Y.; Oden, J.T. (eds.): *Recent Advances in Matrix Methods of Structural Analysis and Design*; University of Alabama Press, Huntsville (1971).
- [49] Gallagher, R.H.; Zienkiewicz, O.C. (eds.): *Optimum Structural Design—Theory and Applications*; J. Wiley & sons, London, New York (1973).
- [50] Gallagher, R.H.: Fully-Stressed Design; in [49], 19–32 (1973).
- [51] Gallatly, R.A.; Berke, L.: Optimality–Criterion–Based Algorithms; in [49], 33–49 (1973).
- [52] Gill, P.E.; Murray, W.; Wright, M.H.: *Practical Optimization*; Academic Press, London (1981).
- [53] Gill, P.E.; Murray, W.; Saunders, M.A.; Wright, M.H.: Model Building and Practical Aspects of Nonlinear Programming; in [118], 209–247 (1985).
- [54] Graefe, R.: Zur Formgebung von Bögen und Gewölben; *architectura, Zeitschrift für Geschichte der Baukunst*, 50–67 (1986).
- [55] Gregory, J.A. (ed.): *The Mathematics of Surfaces*; Clarendon Press, Oxford (1986).
- [56] Grierson, D.E.; Lee, W.H.: Optimal Synthesis of Steel Framework using Standard Sections; *J. Struct. Mech.*, **12**, 335–370 (1984).
- [57] Gründig, L.: Minimal Surfaces for Finding Forms of Structural Membranes; *Comp. & Struct.*, **30**, 679–683 (1988).
- [58] Haber, R.; Shephard, M.S.; Abel, J.F.; Gallagher, R.H.; Greenberg, D.P.: A General Two–Dimensional, Graphical Finite Element Preprocessor Utilizing Discrete Transfinite Mapping; *Int. J. Num. Meth. Engng.*, **17**, 1015–1044 (1981).
- [59] Haftka, R.T.; Starnes, J.H.: Applications of a Quadratic Extended Penalty Function for Structural Optimization; *AIAA Journal*, **14**, 718–724 (1976).
- [60] Haftka, R.T.; Kamat, M.P.: *Elements of Structural Optimization*; Martinus Nijhoff Publishers, The Hague, Boston (1985).

- [61] Haftka, R.T.: Simultaneous Analysis and Design; *AIAA Journal*, **23**, 1099–1103 (1985).
- [62] Haftka, R.T.; Grandhi, R.V.: Structural Shape Optimization – A Survey; *Comp. Meth. Appl. Mech. Engng.*, **57**, 91–106 (1986).
- [63] Haftka, R.T.: First and Second Order Constraint Approximations in Structural Optimization; *Compt. Mech.*, **3**, 89–104 (1988).
- [64] Han, S.-P.: A Globally Convergent Method for Nonlinear Programming; *J. Optimization Theory Appl.*, **22**, 297–309 (1977).
- [65] Hartmann, D.: Optimierung flacher hyperbolischer Paraboloidschalen; *Beton- und Stahlbetonbau*, **9**, 216–222 (1977).
- [66] Haug, E.J.; Cea, J. (eds.): Optimization of Distributed Parameter Structures; Sijthoff and Noordhoff (1981).
- [67] Haug, E.J.; Choi, K.K.; Komkov, V.: Design Sensitivity Analysis of Structural Systems; Academic Press, Orlando, New York, London (1986).
- [68] Herron, G.: Techniques for Visual Continuity; in [39], 163–174 (1987).
- [69] Hinton, E.; Owen, D.R.J.: An Introduction to Finite Element Computations; Pineridge Press Limited, Swansea, U.K. (1979).
- [70] Höfler, A.: Formoptimierung von Leichtbaufachwerken durch Einsatz einer Evolutionsstrategie; Diss., TU Berlin (1976).
- [71] Hörnlein, H.: Take-Off in Optimum Structural Design; in [91], 901–920 (1987).
- [72] Imam, M.H.: Three-Dimensional Shape Optimization; *Int. J. Num. Meth. Engng.*, **18**, 661–673 (1982).
- [73] Jensen, T.: Assembling Triangles and Rectangular Patches and Multivariate Splines; in [39], 203–220 (1987).
- [74] Khot, N.S.; Berke, L.: Structural Optimization using Optimality Criteria Methods; in [2], 47–74 (1984).
- [75] Kimmich, S.; Ramm, E.: Structural Optimization and Analysis with Program System CARAT, in [36] (1989).
- [76] Kimmich, S.: Strukturoptimierung und Sensibilitätsanalyse mit finiten Elementen; Dissertation, Institut für Baustatik, Universität Stuttgart (1990).
- [77] Kirsch, U.: Optimum Structural Design; McGraw-Hill (1981).
- [78] Knepe, G.: Direkte Lösungsstrategien zur Gestaltoptimierung von Flächentragwerken; Diss., Universität Siegen, VDI-Verlag, Düsseldorf (1986).

- [79] Krabs, W.: Einführung in die lineare und nichtlineare Optimierung für Ingenieure; Teubner Verlag, Stuttgart (1983).
- [80] Krieger, K.: Form- und Querschnittsoptimierung von Tragwerken mit der Methode der sequentiellen linearen Programmierung; Diplomarbeit, Inst. für Baustatik, Universität Stuttgart (1989).
- [81] Künzi, H.P.; Krelle, W.; von Randow, R.: Nichtlineare Programmierung; Springer-Verlag, Berlin, Heidelberg, 2. Auflage (1979).
- [82] Lawo, M.; Thierauf, G.: Optimal Design for Dynamic, Stochastic Loading – a Solution by Random Search; in [34], 346–352 (1983).
- [83] Linkwitz, K.; Gründig, L.; Hangleiter, U.; Bahndorf, J.: Mathematisch-Numerische Methoden der Netzberechnung; SFB 64 "Weitgespannte Flächentragwerke", Mitteilung Nr. 72, Stuttgart (1984).
- [84] Lo, S.H.: Finite Element Mesh Generation over Curved Surfaces; *Comp. & Struct.*, **29**, 731–742 (1988).
- [85] Luenberger, D.G.: Linear and Nonlinear Programming; Addison-Wesley (1984).
- [86] Lust, R.V.; Schmit, L.A.: Alternative Approximation Concepts for Space Frame Synthesis; *AIAA Journal*, **24**, 1676–1684 (1986).
- [87] Matthies, H.; Strang, G.: The Solution of Nonlinear Finite Element Equations; *Int. J. Num. Meth. Engng.*, **14**, 1613–1626 (1979).
- [88] Minoux, M.: Mathematical Programming – Theory and Algorithms; J. Wiley & sons, Chichester, New York (1986).
- [89] Miura, H.; Schmit, L.A.: Second Order Approximation of Natural Frequency Constraints in Structural Synthesis; *Int. J. Num. Meth. Engng.*, **13**, 337–351 (1978).
- [90] Morris, A.J. (ed.): Foundations of Structural Optimization: A Unified Approach; J. Wiley & sons, Chichester, New York (1982).
- [91] Mota Soares, C.A. (ed.): Computer Aided Optimal Design – Structural and Mechanical Systems; NATO ASI Series F: Computer and Systems Sciences, vol. 27, Springer-Verlag, Heidelberg, New York (1987).
- [92] Nervi, P.L.: Structures; McGraw-Hill; New York, Toronto, London (1956).
- [93] N.N.: PATRAN Plus User Manual; PDA-Engineering, Costa Mesa, Kalifornien (1987).
- [94] Osyczka, A.: Multicriterion Optimization in Engineering; Ellis Horwood (1984).

- [95] Otto, F.; Happold, E.; Bubner, E. (Hrsgb.): Lufthallenhandbuch; IL 15, Inst. für leichte Flächentragwerke, Universität Stuttgart (1982).
- [96] Otto, F.: Das hängende Dach; Bauwelt-Verlag, Berlin (1954).
- [97] Petersen, P.: A Unified Approach to Optimal Design; in [34], 182–187 (1983).
- [98] Peterson, D.W.: A Review of Constraint Qualifications in Finite-Dimensional Spaces; *SIAM Review*, **25**, 639–654 (1973).
- [99] Piegel, L.: Key Developments in Computer Aided Design; **21**, 5, 262–274 (1989).
- [100] Powell, M.J.D.: A Fast Algorithm for Nonlinearly Constrained Optimization Calculations; in: G.A. Watson (ed.): *Numerical Analysis. Proceedings of the Biennial Conference held at Dundee, June 1977. Lecture Notes in Mathematics*, vol. 630, Springer-Verlag, Berlin, Heidelberg, New York (1978).
- [101] Powell, M.J.D.: The Performance of Two Subroutines for Constrained Optimization on Some Difficult Test Problems; in: Boggs, P.T.; Byrd, R.H.; Schnabel, R.B. (eds.): *Numerical Optimization 1984*, SIAM, Philadelphia (1985).
- [102] Powell, M.J.D.; Yuan, Y.: A Recursive Quadratic Programming Algorithm that uses Differentiable Exact Penalty Functions; *Math. Programming* **35**, 265–278 (1986).
- [103] Powell, M.J.D.: How Bad are the BFGS and DFP Methods when the Objective Function is Quadratic ?; *Math. Programming* **34**, 34–47 (1986).
- [104] Prager, W.; Shield, R.T.: A General Theory of Optimal Plastic Design; *J. Appl. Mechanics*, **34**, 184–186 (1967).
- [105] Pramila, A.; Virtanen, S.: Surfaces of Minimum Area by FEM; *Int. J. Num. Meth. Engng.*, **23**, 1669–1677, (1986).
- [106] Prasad, B.: Explicit Constraint Approximation Forms in Structural Optimization, Part 1: Analyses and Projections; *Comp. Meth. Appl. Mech. Engng.*, **40**, 1–26 (1983).
- [107] Prasad, B.: Novel Concepts for Constraint Treatments and Approximations in Efficient Structural Synthesis; *AIAA Journal*, **22**, 957–966 (1984).
- [108] Ramm, E.: Geometrisch nichtlineare Elastostatik und finite Elemente; Universität Stuttgart, Bericht Nr. 76–2 des Instituts für Baustatik, Stuttgart (1976).

- [109] Ramm, E.; Kammler, G.: Finite Elemente für Tragwerksberechnungen; Seminarunterlagen, CAD-FEM, München (1985).
- [110] Ramm, E.; Schunk, E.: Heinz Isler - Schalen; Krämer Verlag, Stuttgart (1986).
- [111] Ramm, E.; Bletzinger, K.-U., Kimmich, S.: Strukturoptimierung; in: Mitteilungen des SFB 230, Heft 2, Stuttgart (1988).
- [112] Rao, S.S.: Optimization; Wiley Eastern Ltd., New Dehli, 2. Auflage (1984).
- [113] Rechenberg, I.: Evolutionsstrategie; Frommann-Holzboog, Stuttgart (1973).
- [114] Rehle, N.: Approximationsmethoden in der Strukturoptimierung; Diplomarbeit, Institut für Baustatik, Universität Stuttgart (1989).
- [115] Reinschmidt, K.F.; Cornell, C.A., Brothie, J.F.: Iterative Design and Structural Optimization; ASCE, J. Struct. Div., **92**, ST 6, 281-318 (1966).
- [116] Save, M.; Prager, W. (eds.): Structural Optimization, vol. 1: Optimality Criteria; Plenum Press, New York, London (1985).
- [117] Schittkowski, K.: On the Convergence of a Sequential Quadratic Programming Method with an Augmented Lagrangian Line Search Function; Math. Operationsforschung und Statistik, Ser. Optimization, **14**, 2, 197-216 (1983).
- [118] Schittkowski, K. (ed.): Computational Mathematical Programming; NATO ASI Series, vol. F 15, Springer-Verlag, Berlin, Heidelberg (1985).
- [119] Schmit, L.A.; Farshi, B.: Some Approximation Concepts for Structural Synthesis; AIAA Journal, **12**, 5, 692-699 (1974).
- [120] Schmit, L.A.; Miura, H.: A new Structural Analysis/Synthesis Capability-ACCESS 1; AIAA Journal, **14**, 661-671 (1976).
- [121] Schmit, L.A.; Fleury, C.: Discrete-Continuous Variable Structural Synthesis using Dual Methods; AIAA Journal, **18**, 1515-1524 (1980).
- [122] Schmit, L.A.: Structural Synthesis - its Genesis and Development; AIAA Journal, **19**, 10, 1249-1263 (1981).
- [123] Schwefel, H.P.: Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie; Birkhäuser Verlag, Basel, Stuttgart (1977).
- [124] Schweizerhof, K.; Ramm, E.: Combining quasi-Newton and Arclength Methods for the Analysis of Nonlinear Problems into the Postlimit Range; Paper presented at IACM, First World Congress on Computational Mechanics. The University of Texas, Austin (1986).
- [125] Schwenzer, C.: Neue Entwicklungen zur Simulation technischer Prozesse. Teil I; CAD-CAM Report 6, 54-58 (1989).

- [126] Shanno,D.F., Phua, K.H.: Numerical Experience with Sequential Quadratic Programming Algorithms for Equality Constrained Nonlinear Programming; ACM Transactions on Mathematical Software, **15**, 1, 49–63 (1989).
- [127] Shephard, M.S.; Grice, K.R.; Lo, J.A.; Schroeder,W.J.: Trends in Automatic Three-Dimensional Mesh Generation; Comp. & Struct., **30**, 1/2, 421–429 (1988).
- [128] Sobieszczanski-Sobieski, J.; James, B.B.; Dori, A.R.: Structural Optimization by Multilevel Decomposition; AIAA Journal, **23**, 1775–1782 (1985).
- [129] Stadler, W.: Multicriteria Optimization in Mechanics ( A Survey ); Appl. Mech. Review, **37**, 277–286 (1984).
- [130] Stadler, W. (ed.): Multicriteria Optimization in Engineering and in the Sciences; Plenum Press, New York, London (1988).
- [131] Starnes, H.; Haftka, R.T.: Preliminary Design of Composite Wings for Buckling, Strength and Displacement Constraints; Journal of Aircraft, **16**, 564–570 (1979).
- [132] Stegmüller, H.; Bletzinger, K.-U.; Kimmich, S.: Programmsystem CARAT, Eingabebeschreibung und Dokumentation; Inst. für Baustatik, Universität Stuttgart (1988).
- [133] Stocker, J.: Zum Tragverhalten der Kuppelkonstruktion von St. Geneviève in Paris; Diplomarbeit, Inst. für Baustatik, Universität Stuttgart (1987).
- [134] Svanberg, K.: An Algorithm for Optimum Structural Design Using Duality; Math. Progr. Study, **20**, 161–177 (1982).
- [135] Svanberg, K.: The Method of Moving Asymptotes – a new Method for Structural Optmization; Int. J. Num. Meth. Engng., **24**, 359–373 (1987).
- [136] Templeman, A.B.: A Dual Approach to Optimum Truss Design; J. Struct. Mech., **4**, 235–255 (1976).
- [137] Thanedar, P.B.; Arora, J.S.; Tseng, C.H. et al.: Performance of some SQP Algorithms on Structural Optimization Problems; J. Num. Meth. Engng., **23**, 2187–2203 (1986).
- [138] Thanedar, P.B.; Arora, J.S.; Tseng, C.H.: A Hybrid Optimization Method and its Role in Computer-Aided Design; Comp. & Struct., **23**, 3, 305–314 (1986).

- [139] Thierauf, G.: Optimale Bemessung und Optimierung im konstruktiven Ingenieurbau; Bauingenieur, **64**, 463–472 (1989).
- [140] Torroja, E.: Logik der Form; Verlag Georg D.W. Callwey, München (1961).
- [141] Tseng, C.H.; Arora, J.S.: On the Implementation of Computational Algorithms for Optimal Design, Part I: Preliminary Investigations, Part II: Extensive Numerical Investigations; Int. J. Num. Meth. Engng., **26**, 1365–1402 (1988).
- [142] Vanderplaats, G.N.; Moses, F.: Structural Optimization by Methods of Feasible Directions; Comp. & Struct., **3**, 739–755 (1973).
- [143] Vanderplaats, G.N.: Structural Optimization – Past, Present, Future; AIAA Journal, **20**, 992–1000 (1982).
- [144] Vanderplaats, G.N.: Numerical Optimization Techniques for Engineering Design: With Applications; McGraw–Hill, New York (1984).
- [145] Vanderplaats, G.N.; Salajegheh, E.: New Approximation Method for Stress Constraints in Structural Synthesis; AIAA Journal, **27**, 352–358 (1989).
- [146] Venkayya, V.B.: Design of Optimum Structures; Comp. & Struct., **1**, 265–309 (1971).
- [147] Venkayya, V.B.: Structural Optimization: A Review and some Recommendations; Int. J. Num. Meth. Engng., **13**, 203–228 (1978).
- [148] Veron, M.; Ris, G.; Musse, J.-P.: Continuity of Biparametric Surface Patches; Computer Aided Design, **4**, 267–273 (1976).
- [149] Walter, U.: Was sind NURBS ? – Eine kleine Einführung; CAD/CAM, **3**, 96–98 (1989).
- [150] Wang, S.-Y.; Sun, Y.; Gallagher, R.H.: Sensitivity Analysis in Shape Optimization of Continuum Structures; Comp. & Struct., **20**, 5, 855–867 (1985).
- [151] Wasiutynski, Z.; Brandt, A.: The Present State of Knowledge in the Field of Optimum Design of Structures; App. Mech. Reviews, **16**, 5, 341–350 (1963).
- [152] Watkins, M.A.: Problems in Geometric Continuity; Computer Aided Design, **20**, 8, 499–502 (1988).
- [153] Williams, C.J.K.: Use of Structural Analogy in Generation of Smooth Surfaces for Engineering Purposes; Computer Aided Design, **19**, 6, 310–322 (1987).

- [154] Yang, R.J.: A Three-Dimensional Shape Optimization System - SHOP3D; *Comp. & Struct.*, **31**, 881-890 (1989).
- [155] Yao, T.-M.; Choi, K.K.: 3-D Shape Optimal Design and Automatic Finite Element Regridding; *Int. J. Num. Meth. Engng.*, **28**, 369-384 (1989).
- [156] Zienkiewicz, O.D.; Phillips, D.V.: An Automatic Mesh Generation Scheme for Plane and Curved Surfaces by "Isoparametric" Co-Ordinates; *Int. J. Num. Meth. Engng.*, **3**, 519-528 (1971).
- [157] Zienkiewicz, O.C.; Campell, J.S.: Shape Optimization and Sequential Linear Programming; in [49], 109-126 (1973).
- [158] Zienkiewicz, O.D.; Taylor, R.: *The Finite Element Method*; McGraw-Hill, London, 4. Auflage (1989).

## ANHANG

### A1. Kantorowitsch–Ungleichung

Die Kantorowitsch–Ungleichung ist ein wichtiges Hilfsmittel für die Konvergenzanalyse von Algorithmen. Sie besagt:

$Q$  sei eine positiv definite, symmetrische  $n \times n$  Matrix. Für jeden Vektor  $x$  gelte dann:

$$\frac{(x^T x)^2}{(x^T Q x) (x^T Q^{-1} x)} \geq \frac{4aA}{(a + A)^2} \quad (\text{A1.1})$$

wobei  $a$  und  $A$  der kleinste bzw. größte Eigenwert der Matrix  $Q$  seien.

Beweis:

Die Eigenwerte  $\{\lambda_i \mid i = 1, \dots, n\}$  von  $Q$  seien der Größe nach geordnet:

$$0 < a = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n = A \quad (\text{A1.2})$$

Durch eine geeignete Koordinatentransformation kann  $Q$  auf Diagonalform gebracht werden, mit der Diagonalen  $(\lambda_1, \dots, \lambda_n)$ . In diesem Koordinatensystem gilt:

$$\begin{aligned} \frac{(x^T x)^2}{(x^T Q x) (x^T Q^{-1} x)} &= \frac{\left(\sum_{i=1}^n x_i^2\right)^2}{\left(\sum_{i=1}^n \lambda_i x_i^2\right) \left(\sum_{i=1}^n x_i^2 / \lambda_i\right)} \\ &= \frac{1 / \sum_{i=1}^n w_i \lambda_i}{\sum_{i=1}^n w_i / \lambda_i} = \frac{\Phi(w)}{\Psi(w)} \end{aligned}$$

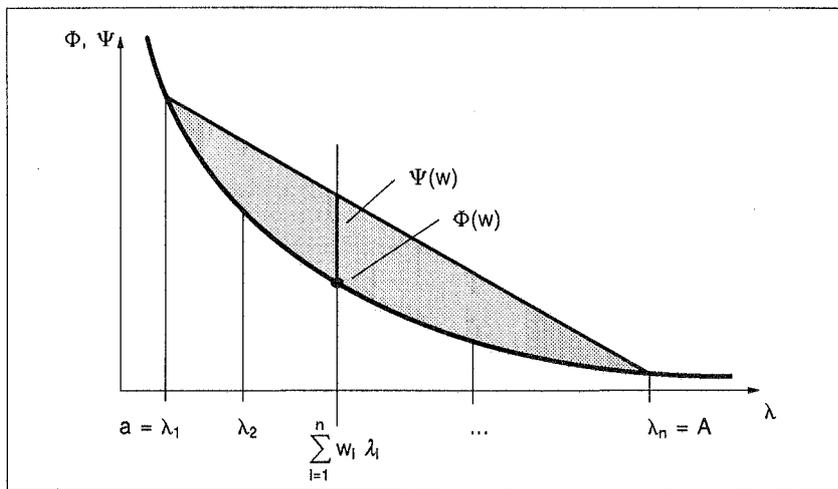
$$\text{mit } w_i = \frac{x_i^2}{\sum_{i=1}^n x_i^2} \quad ; \quad \text{d.h. } \sum_{i=1}^n w_i = 1$$

In Bild A1.1 ist die Sachlage verdeutlicht.  $\Phi(w)$  stellt Punkte auf der Kurve  $1/\lambda$  dar, während von  $\Psi(w)$  Punkte zwischen der Hyperbel und der Geraden beschrieben werden. Für dasselbe  $w$  werden  $\Phi$  und  $\Psi$  von Punkten dargestellt, die auf einer vertikalen Geraden liegen.

In einer Abschätzung ist der Quotient  $\Phi/\Psi$  dort am kleinsten, wo Gerade und Hyperbel am weitesten auseinanderklaffen. Somit kann geschrieben werden:

$$\frac{\Phi(\mathbf{w})}{\Psi(\mathbf{w})} \geq \min_{\lambda} \frac{1/\lambda}{(a + A - \lambda)/(a + A)} = \frac{4aA}{(a + A)^2}$$

womit Aussage (A1.1) bewiesen ist. ■



**Bild A1.1:** Kantorowitsch-Ungleichung

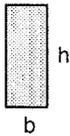
## A2. Formbeiwerte für Stabquerschnitte

Formbeiwert  $c = \frac{I}{A^2}$

Trägheitsmoment  $I$

Querschnittsfläche  $A$

### Rechteck



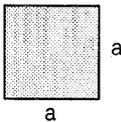
$$I = \frac{1}{12} b h^3$$

$$A = b h$$

$$c = \frac{1}{12} \frac{b h^3}{(b h)^2}$$

$$c = \frac{1}{12} \frac{h}{b}$$

### Quadrat

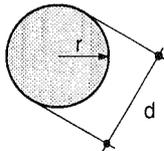


$$I = \frac{1}{12} a^4$$

$$A = a^2$$

$$c = \frac{1}{12} = 0,0833$$

### Vollkreis



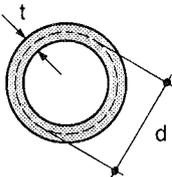
$$I = \frac{\pi d^4}{64}$$

$$A = \frac{\pi d^2}{4}$$

$$c = \frac{16\pi d^4}{64\pi^2 d^2}$$

$$c = \frac{1}{4} \pi = 0,0796$$

### dünnwandiges Rohr



$$I = \frac{\pi t d^3}{8}$$

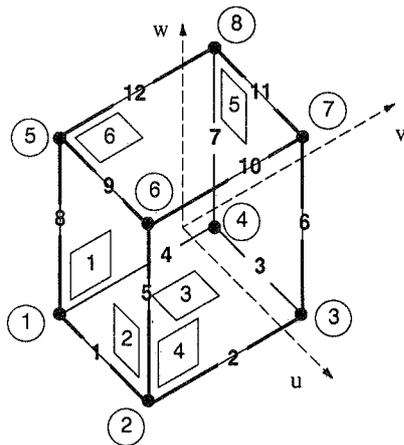
$$A = \pi d t$$

$$c = \frac{\pi t d^3}{8\pi^2 t^2 d^2}$$

$$c = \frac{d}{8\pi t}$$

### A3. 3-D Coons-Element, vollständige Formfunktion

$$\begin{aligned}
 \mathbf{r} = & \frac{1}{2}(1-u) \mathbf{r}_1(v, w) & + \frac{1}{2}(1+u) \mathbf{r}_4(v, w) \\
 & + \frac{1}{2}(1-v) \mathbf{r}_2(u, w) & + \frac{1}{2}(1+v) \mathbf{r}_5(u, w) \\
 & + \frac{1}{2}(1-w) \mathbf{r}_3(u, v) & + \frac{1}{2}(1+w) \mathbf{r}_6(u, v) \\
 \\ 
 & - \frac{1}{4}(1-v) (1-w) \mathbf{r}_1(u) & - \frac{1}{4}(1-u) (1+v) \mathbf{r}_7(w) \\
 & - \frac{1}{4}(1+u) (1-w) \mathbf{r}_2(v) & - \frac{1}{4}(1-u) (1-v) \mathbf{r}_8(w) \\
 & - \frac{1}{4}(1+v) (1-w) \mathbf{r}_3(u) & - \frac{1}{4}(1-v) (1+w) \mathbf{r}_9(u) \\
 & - \frac{1}{4}(1-u) (1-w) \mathbf{r}_4(v) & - \frac{1}{4}(1+u) (1+w) \mathbf{r}_{10}(v) \\
 & - \frac{1}{4}(1+u) (1-v) \mathbf{r}_5(w) & - \frac{1}{4}(1+v) (1+w) \mathbf{r}_{11}(u) \\
 & - \frac{1}{4}(1+u) (1+v) \mathbf{r}_6(w) & - \frac{1}{4}(1-u) (1+w) \mathbf{r}_{12}(w) \\
 \\ 
 & + \frac{1}{8}(1-u) (1-v) (1-w) \mathbf{r}_1 & + \frac{1}{8}(1-u) (1-v) (1+w) \mathbf{r}_5 \\
 & + \frac{1}{8}(1+u) (1-v) (1-w) \mathbf{r}_2 & + \frac{1}{8}(1+u) (1-v) (1+w) \mathbf{r}_6 \\
 & + \frac{1}{8}(1+u) (1+v) (1-w) \mathbf{r}_3 & + \frac{1}{8}(1+u) (1+v) (1+w) \mathbf{r}_7 \\
 & + \frac{1}{8}(1-u) (1+v) (1-w) \mathbf{r}_4 & + \frac{1}{8}(1-u) (1+v) (1+w) \mathbf{r}_8
 \end{aligned}$$



$-1 \leq u \leq 1$
$-1 \leq v \leq 1$
$-1 \leq w \leq 1$

## Lebenslauf von Kai-Uwe Bletzinger

14. Juni 1958 geboren in Stuttgart  
Sohn des Architekten Helmut Bletzinger  
und dessen Ehefrau Ursula, geb. Klemm
- 1965 – 1968 Grundschule in Stuttgart-Plieningen
- 1968 – 1977 Besuch des Paracelsus-Gymnasiums Hohenheim  
in Stuttgart-Plieningen
10. Mai 1977 Abitur
- 1977 – 1978 15-monatiger Grundwehrdienst bei der Bundeswehr
- 1978 – 1984 Studium des Bauingenieurwesens an der Universität  
Stuttgart
- 1981 – 1982 zwei Semester Auslandsstudium an der University  
of Calgary in Kanada
24. August 1984 Abschluß Diplom-Ingenieur
7. Februar 1985 Emil Mörsch-Studienpreis 1984 für besondere  
Studienleistungen
- seit Oktober 1984 wissenschaftlicher Mitarbeiter am Institut für  
Baustatik der Universität Stuttgart

Folgende Berichte sind bereits erschienen:

- 74-1 M. Becker, J. Bühler, G. Lang-Lendorff, K. Papailiou, J. M. Sättele:  
Kontaktkurs EDV im konstruktiven Ingenieurbau.
- 74-2 G. Werner:  
Experimentelle und theoretische Untersuchungen zur Ermittlung des Trag-  
verhaltens biege- und verdrehbeanspruchter Stäbe mit I-Querschnitt.
- 74-3 K. Tompert:  
Berechnung kreiszylindrischer Silos auf elastischer Unterlage.
- 74-4 W. Riehle:  
Studie über verallgemeinerte Variationsfunktionale und ihre Anwendung bei  
der Methode der finiten Plattenelemente.
- 75-1 G. Müller, R. W. Rembold, J. M. Sättele,  
K. H. Schweizerhof, W. Wissmann:  
Platten – Theorie, Berechnung, Bemessung. Teil I.
- 75-2 G. Müller:  
Numerische Behandlung der Kirchhoffschen und Reissnerschen Platten-  
theorie nach einer diskretisierten und erweiterten Trefftz-Methode.
- 75-3 E. A. Castrillón O.:  
Beitrag zur Berechnung langer dünnwandiger dreizelliger Träger unter Be-  
rücksichtigung der Profilverformung.
- 76-1 W. Block, G. Eisenbieglér, R. D. Kugler, H. Lieb, G. Müller,  
J. Müller, K.-H. Reineck, J. Schlaich, K. H. Schweizerhof, F. Seible:  
Platten – Theorie, Berechnung, Bemessung. Teil II.
- 76-2 E. Ramm:  
Geometrisch nichtlineare Elastostatik und finite Elemente.
- 77-1 B.-M. Sulke:  
Berechnung dünnwandiger prismatischer Falwerke mit verformbarem mehr-  
zelligen Querschnitt.
- 78-1 F. Fujii:  
Anwendung der Methode der finiten Elemente auf die Berechnung von Stahl-  
betonplatten.
- 79-1 B. Brendel:  
Geometrisch nichtlineare Elastostabilität.
- 79-2 H. G. Berg:  
Tragverhalten und Formfindung versteifter Kuppelschalen über quadrati-  
schem Grundriß auf Einzelstützen.
- 79-3 F. W. Bornscheuer, B. Brendel, L. Häfner, E. Ramm, J. M. Sättele:  
Fallstudien zu Schalentragwerken (in englischer Sprache).
- 80-1 R. I. Del Gaizo:  
Liegende zylindrische Behälter und Rohre auf Sattellagern endlicher Breite.
- 80-2 R. W. Rembold:  
Beitrag zum Tragverhalten ausgewählter Plattentragwerke unter Berücksich-  
tigung der Reissnerschen Theorie und der Methode der gemischten finiten  
Elemente.

- 80-3 J. M. Sättele:  
Ein finites Elementkonzept zur Berechnung von Platten und Schalen bei stofflicher und geometrischer Nichtlinearität.
- 82-1 L. Häfner:  
Einfluß einer Rundschweißnaht auf die Stabilität und Traglast des axialbelasteten Kreiszylinders.
- 82-2 K. Schweizerhof:  
Nichtlineare Berechnung von Tragwerken unter verformungsabhängiger Belastung mit finiten Elementen.
- 82-3 H.-P. Andrä:  
Zum Tragverhalten des Auflagerbereichs von Flachdecken.
- 1 (1983) P. Osterrieder:  
Traglastberechnung von räumlichen Stabwerken bei großen Verformungen mit finiten Elementen.
- 2 (1983) T. A. Kompfner:  
Ein finites Elementmodell für die geometrisch und physikalisch nichtlineare Berechnung von Stahlbetonschalen.
- 3 (1983) A. Diack:  
Beitrag zur Stabilität diskret längsversteifter Kreiszyllinderschalen unter Axialdruck.
- 4 (1984) A. Burmeister, F. W. Borscheuer, E. Ramm:  
Traglasten von Kugelbehältern mit Stützen und Formabweichungen unter Innendruck und Stützenlängskraft.
- 5 (1985) H. Stegmüller:  
Grenzlastberechnungen flüssigkeitsgefüllter Schalen mit "degenerierten" Schalenelementen.
- 6 (1987) A. Burmeister:  
Dynamische Stabilität nach der Methode der finiten Elemente mit Anwendungen auf Kugelschalen.
- 7 (1987) G. Kammler:  
Ein finites Elementmodell zur Berechnung von Trägern und Stützen mit offenem, dünnwandigem Querschnitt unter Berücksichtigung der Interaktion zwischen globalem und lokalem Versagen.
- 8 (1988) A. Matzenmiller:  
Ein rationales Lösungskonzept für geometrisch und physikalisch nichtlineare Strukturberechnungen.
- 9 (1989) D. Tao:  
Die Technik der reduzierten Basis bei nichtlinearen finiten Element-Berechnungen.
- 10 (1989) K. Weimar:  
Ein nichtlineares Balkenelement mit Anwendung als Längssteifen axialbelasteter Kreiszyllinder.