

Quasi – Newton Verfahren und  
Kurvenverfolgungsalgorithmen für die  
Lösung nichtlinearer Gleichungssysteme in  
der Strukturmechanik

Karl Schweizerhof  
Institut für Baustatik  
Universität Karlsruhe

Juni 1989  
Revidierte Version November 1989



# Vorwort

Die vorliegende Arbeit stellt die Zusammenfassung von Studien des Autors über Kurvenverfolgungsalgorithmen und Quasi-Newton Verfahren in den Jahren 1984-89 dar.

Angeregt durch die Arbeiten von Prof.Dr.-Ing.E.Ramm wurden zuerst Skalierungsfragen bei Kurvenverfolgungsalgorithmen erörtert. Von Herbst 1984 bis Februar 1986 wurde dann am Institut für Baustatik der Universität Stuttgart im Rahmen des DFG Schwerpunkts 'Nichtlineare Berechnungen im Konstruktiven Ingenieurbau' der Abschnitt über Quasi-Newton Verfahren und Kurvenverfolgungsalgorithmen mit einfachen Line Search Algorithmen erarbeitet. Die Weiterentwicklung der vektorisierten Algorithmen sowie der Abschnitt über konsistente Strahlminimierung folgte anschließend (1987/88) mit dem vorläufigen Abschluß am Institut für Baustatik der Universität Karlsruhe.

In erster Linie danke ich Professor Dr.-Ing. E.Ramm für seine wertvollen Anregungen und die großzügige Unterstützung, sowie der Deutschen Forschungsgemeinschaft für die freundliche Förderung. Dr.John Hallquist, Livermore, gab wichtige Hinweise zu den vektorisierten Algorithmen und Dr.-Ing. A.Matzenmiller setzte sich kritisch mit dem Problem 'Quasi-Newton Verfahren und Plastizität' auseinander. Beiden möchte ich für ihr Interesse und ihren Anteil danken. Außerdem danke ich Frau A.Elicker sowie Herrn L.Segi für die große Hilfe bei der Erstellung und Gestaltung des Manuskripts.

Karlsruhe, den 27.Juni 1989

Karl Schweizerhof



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>6</b>
<b>2</b>	<b>Übersicht über Lösungsverfahren für nichtlineare Gleichungssysteme in der Strukturmechanik – inkrementell – iteratives Vorgehen</b>	<b>9</b>
2.1	Inkrementelle Verfahren . . . . .	10
2.2	Inkrementell-iterative Verfahren . . . . .	11
<b>3</b>	<b>Das Newton/Raphson Verfahren</b>	<b>14</b>
3.1	Nullstellensuche und Minimierungsproblem – echtes Newton Verfahren . . . . .	14
3.2	Modifizierte Newton Verfahren . . . . .	17
<b>4</b>	<b>Sekantenverfahren</b>	<b>20</b>
4.1	Allgemeines . . . . .	20
4.2	Quasi – Newton Verfahren . . . . .	22
4.2.1	Rang – Eins Änderungen . . . . .	24
4.2.2	Rang – Zwei Änderungen . . . . .	28
4.2.3	Konvergenz . . . . .	32
4.2.4	Verbesserung der Aufdatierung durch Skalierung . . . . .	33
4.3	Vektorisierte Aufdatierungsformeln und Algorithmen . . . . .	36
4.3.1	Allgemeines . . . . .	36
4.3.2	Rang-Eins Änderungen . . . . .	37
4.3.3	Rang-Zwei Änderungen . . . . .	39
4.4	Quasi – Newton Verfahren mit reduzierter Aufdatierung . . . . .	46
<b>5</b>	<b>Strahlminimierung - Line Search</b>	<b>48</b>
5.1	Allgemeines . . . . .	48
5.2	Einfache Strahlminimierungsalgorithmen . . . . .	52
5.3	Exakte - Inexakte Strahlminimierung . . . . .	54

<b>6</b>	<b>Kurvenverfolgungsalgorithmen</b>	<b>55</b>
6.1	Allgemeines . . . . .	55
6.1.1	Einfache Verfahren . . . . .	55
6.1.2	Bogenlängenverfahren . . . . .	56
6.2	Erweiterte Gleichungssysteme – Newton Verfahren – Direkte Lösung . . . . .	58
6.2.1	Aufbau eines erweiterten Gleichungssystems . . . . .	58
6.2.2	Lösung des erweiterten Systems mit Gleichungslösern für unsymmetrischen Matrizen . . . . .	61
6.2.3	Reduktion auf ein Gleichungssystem der Ausgangsgröße	62
6.2.4	Direkte Erfüllung der Zusatzgleichung . . . . .	66
6.3	Algorithmen . . . . .	68
6.3.1	Allgemeines . . . . .	68
6.3.2	Prädiktorschritt . . . . .	70
6.3.3	Korrektorschritte . . . . .	71
6.3.4	Lineare Bedingungsgleichungen . . . . .	72
6.3.4.1	Einfache Verschiebungssteuerung . . . . .	72
6.3.4.2	Verschiebungssteuerung mit mehreren Verschiebungs- termen . . . . .	73
6.3.4.3	Last-Verschiebungssteuerung mit mehreren Verschie- bungen . . . . .	74
6.3.4.4	Iteration auf Normalenebenen . . . . .	74
6.3.4.5	Iteration mit konstanter äußerer Energie . . . . .	77
6.3.5	Nichtlineare Bedingungsgleichungen . . . . .	79
6.3.5.1	Quadratische Zusatzgleichung . . . . .	79
6.3.5.1.1	Newton-Verfahren für das Gesamtsystem . . . . .	80
6.3.5.1.2	Direkte Erfüllung der Zusatzgleichung . . . . .	80
6.3.5.1.3	Iterative Lösung der Zusatzgleichung . . . . .	82
6.3.5.1.4	Geometrische Überlegungen . . . . .	84
6.3.5.2	Allgemeine nichtlineare Bedingungen . . . . .	86
6.3.6	Wertung der Verfahren . . . . .	86
6.4	Skalierung - Größenangleich . . . . .	87

6.4.1	Skalierung bei einzelnen Algorithmen . . . . .	88
6.4.2	Kombination / Selektion von Algorithmen mittels Skalierung . . . . .	92
6.5	Schrittweitensteuerung . . . . .	92
6.6	Begleitende Maßnahmen zur Steuerung der Algorithmen . . .	95
<b>7</b>	<b>Strahlminimierung (Line Search) bei Kurvenverfolgungsalgorithmen</b>	<b>99</b>
7.1	Klassisches Vorgehen - Konsistente Minimierung . . . . .	99
7.2	Numerische Relaxation . . . . .	105
<b>8</b>	<b>Kombination der Quasi-Newton Verfahren mit Kurvenverfolgungsalgorithmen</b>	<b>107</b>
8.1	Algorithmen . . . . .	107
8.1.1	Rang – Eins Aufdatierungen . . . . .	108
8.1.2	Rang – Zwei Aufdatierungen . . . . .	111
8.2	Diskussion / Schrifttum . . . . .	115
8.3	Kombination mit konsistenter Strahlminimierung . . . . .	116
<b>9</b>	<b>Erweiterung auf viele Zusatzgleichungen</b>	<b>118</b>
<b>10</b>	<b>Numerische Beispiele</b>	<b>120</b>
10.1	Allgemeines . . . . .	120
10.2	Kragbalken unter gleichförmiger Belastung . . . . .	122
10.3	Quadratplatte unter gleichmäßiger Belastung . . . . .	124
10.4	Scheibenstreifen mit Loch unter Zugbelastung . . . . .	127
10.5	Durchschlagproblem einer Schale mit Dickenvariation . . . . .	130
10.5.1	Vergleich der Kurvenverfolgungsalgorithmen mit und ohne Strahlminimierung . . . . .	131
10.5.2	Kombination mit Quasi-Newton Verfahren . . . . .	136
10.6	Rahmen unter Einzellast - Große Rotationen . . . . .	139
10.7	Bogen unter Einzellast - Skalierungsfragen . . . . .	141
10.8	Kegelschale unter Innendruckbelastung . . . . .	143

<b>11 Zusammenfassung</b>	<b>146</b>
<b>12 Schrifttum</b>	<b>149</b>
<b>13 Anhang</b>	<b>157</b>
A. Sherman/Morrison Formel . . . . .	157
B. Rang-Eins Änderung nach Kleinmichel . . . . .	158
C. Quasi viskose Dämpfung nach Kröplin . . . . .	160

# 1 Einleitung

In der Strukturmechanik sind vielfach nichtlineare Gleichungssysteme zu lösen, die z.B. bei der direkten Aufstellung der Differentialgleichungen und ihrer Diskretisierung mit finiten Elementen oder finiten Differenzen an einem System, bei der Minimierung von Funktionen oder bei der Optimierung entstehen.

Die Gleichungssysteme werden allgemein geschrieben als

$$\mathbf{F}(\mathbf{x}) = 0 \quad (1.1)$$

mit  $\mathbf{x}$  als dem Vektor der Unbekannten des Systems (Verschiebungen und/oder Spannungen). In der linearen Mechanik ergeben sich lineare Gleichungssysteme

$$\mathbf{F}(\mathbf{x}) = \mathbf{K} \cdot \mathbf{x} - \mathbf{b} \quad (1.2)$$

wobei  $\mathbf{K}$  z.B. als Steifigkeits- oder Flexibilitätsmatrix,  $\mathbf{b}$  als Vektor äußerer Lasten und  $\mathbf{x}$  als Verschiebungsvektor gedeutet werden können.

Die gängigen Verfahren zur Lösung von Standard-Problemen werden im Schrifttum zur numerischen Mathematik ausführlich erörtert /Ortega, Rheinboldt 1970/, /Dahlquist, Björk 1970/ und sollen im Rahmen der vorliegenden Studie nur soweit im Kontext erforderlich ausschnittsweise wiedergegeben werden.

Bei allgemeinen Funktionen in der nichtlinearen Mechanik besteht nun die Schwierigkeit, ein geeignetes Verfahren zu finden, mit dem die Nullstellensuche für das nichtlineare System effizient vorgenommen werden kann.

Die einfachen Methoden zur Lösung von (1.1) versagen meist, wenn das Gleichungssystem  $\mathbf{F}(\mathbf{x})$  mehrere Nullstellen d.h. mehrere Lösungsvektoren besitzt oder mehrere Minima von  $\|\mathbf{F}(\mathbf{x})\|$  (Euklidische Norm) existieren. Eine einfache Abhilfe kann hierbei durch die Einbettung des gegebenen Problems in eine Problemfamilie

$$\mathbf{H}(\mathbf{x}, t) = 0$$

erreicht werden /Schwetlick 1979/.

Der Grundgedanke der Einbettung ist, die Lösung für  $\mathbf{H}(\mathbf{x}, 1)$  über die Lösung einfacherer Probleme mit  $t < 1$  z.B.  $\mathbf{H}(\mathbf{x}, 0) = 0$  schrittweise zu erzielen ( $t \leq 1$ ).

Eine übliche Einbettung stellt die vorliegende Form dar:

$$\mathbf{H}(\mathbf{x}, t) = t \cdot \mathbf{F}(\mathbf{x}) + (1 - t)\mathbf{G}(\mathbf{x})$$

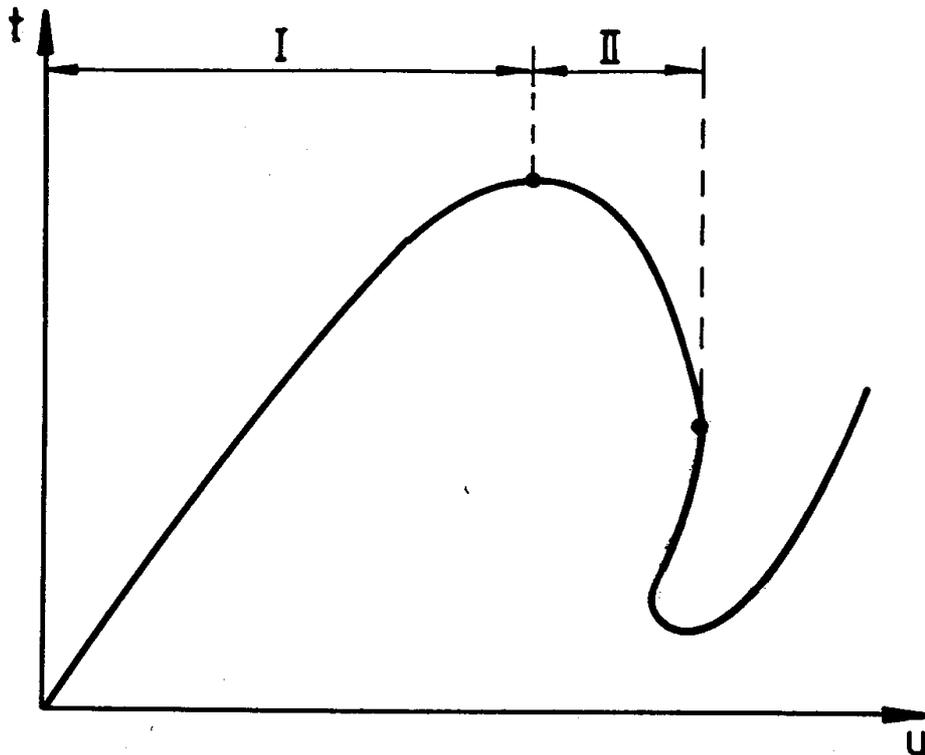
Hierbei ist  $\mathbf{G}(\mathbf{x})$  eine beliebige Funktion, deren Nullstellen bekannt sind. Man spricht in dem vorliegenden Fall von einer homotopen Abbildung, wenn die Abbildung  $\mathbf{H}$  im Einheitsintervall  $[0,1]$  stetig ist.  $\mathbf{G}(\mathbf{x})$  heißt dann homotop zu  $\mathbf{F}(\mathbf{x})$ .

Wird das eingebettete Problem nicht nur für ausgewählte Werte von  $t$  sondern für eine Sequenz von Werten für  $t$  aufsteigend von  $0 \rightarrow 1$  gelöst, erhält man Informationen, wie das System  $\mathbf{F}(\mathbf{x})$  entstanden ist. In der Strukturmechanik ist dieses Verfahren als inkrementelles Vorgehen bekannt;  $t = 0$  entspricht z.B. einer unbelasteten Struktur, wohingegen  $t = 1$  den aktuellen Lastzustand beschreibt, für den die Lösung gesucht wird. Im englischen Schrifttum ist für dieses Vorgehen auch der Begriff "Continuation Method" gebräuchlich.

In der Strukturmechanik entspricht  $t$  meist dem Lastparameter, mit dem ein vorgegebener Lastvektor multipliziert wird. In ähnlicher Form können Parameter für Verschiebungen mit Verschiebungsvektoren verwendet werden. Man spricht dann von last- bzw. verschiebungsgesteuerten Verfahren. Lastgesteuerte Verfahren sind sinnvoll, solange die Verschiebungswerte mit den Lastwerten kontinuierlich anwachsen (Bereich I in Bild 1.1). Mit verschiebungsgesteuerten Verfahren lassen sich auch Last- Verformungskurven mit abfallenden Ästen berechnen (Bereich II), sofern die Verschiebung kontinuierlich zunimmt. Beide Parameter versagen als Steuerung, wenn entweder für den vorgegebenen Wert keine Lösung existiert oder mehrere Lösungen existieren, d.h. der Steuerparameter in einem Bereich des Lösungspfads sowohl zu- als auch abnimmt.

Für allgemeine Funktionen bzw. Last-Verformungskurven, bei denen die kontinuierliche Zunahme weder der Verschiebungen noch der Belastung gesichert ist, empfiehlt es sich, auf eine verallgemeinerte Verfahrensklasse überzugehen, die Kurvenverfolgungsalgorithmen /Allgower 1981/. Hierbei kann aus einer Vielzahl von Bedingungen, die Last- und/oder Verschiebungswerte enthalten, eine sinnvolle Steuerung gewählt werden. Diese Verfahren sind in der Strukturmechanik unter dem Begriff Bogenlängenverfahren (arc-length-methods) bekannt, da im einfachsten Fall eines Ein-Freiheitsgrad-Systems, die Bedingung  $\|\lambda, u\| = c = \text{konst.}$  dazu führt, daß der Lösungspunkt auf einem Kreisbogen mit Radius  $c$  um den Punkt mit  $\lambda, u = 0$  liegt.

Der Einsatz von allgemeinen Kurvenverfolgungsalgorithmen ist besonders bei der Grenzlastuntersuchung von Tragwerken von Bedeutung. Durch die Berechnung von allgemeinen Last- Verformungskurven, auch mit physikalisch kaum zu realisierenden abfallenden Lastparametern, lassen sich wichtige Ein-



**Bild 1.1 Bereiche einer allgemeinen Last – Verschiebungskurve**

blicke in das wirkliche Tragverhalten gewinnen.

Die vorliegende Arbeit ist so aufgebaut, daß ausgehend von den Standardlösungsverfahren - Newton/Raphson und Modifikationen - auf die für Vektorrechner und Untersuchung von Systemen mit vielen Unbekannten besonders geeigneten Quasi - Newton Verfahren ausführlich eingegangen wird. Dabei wird auch das wichtige numerische Verfahren der Strahlminimierung (Line search) bei Ingenieurproblemen erörtert. Den zweiten Hauptpunkt bilden die Darstellung und Diskussion der Bogenlängenverfahren, für die einige Varianten angegeben werden. Den methodischen Teil der Arbeit schließt die Kombination der Quasi - Newton Verfahren und Strahlminimierung mit den Bogenlängenverfahren ab. Hierfür werden effiziente Algorithmen vorgestellt.

Die Verifizierung und Diskussion der Methoden erfolgt an einigen numerischen Beispielen mit dem Finite Element Programm NISA /Brendel et al 1981/, die den Anwendungsbereich und die Grenzen der dargestellten und entwickelten Verfahren aufzeigen.

Ein Ausblick und die kritische Wertung der Verfahren bilden den Abschluß der Arbeit.

## 2 Übersicht über Lösungsverfahren für nichtlineare Gleichungssysteme in der Strukturmechanik – inkrementell – iteratives Vorgehen

Die nichtlinearen Probleme in der Strukturmechanik führen bei Verwendung numerischer Verfahren auf nichtlineare Gleichungssysteme der Form

$$\mathbf{H}(\mathbf{x}, t) = \mathbf{G} \quad \text{bzw.} \quad \mathbf{F}(\mathbf{x}) = 0 \quad (2.1)$$

wobei der Einbettungsparameter  $t$  üblicherweise als Lastparameter  $\lambda$  gegeben ist. Für die zweite Form wird die Bedingung  $t = \lambda = \text{konst.}$  angenommen. Auf den Aspekt, wenn  $t$  über eine zusätzliche Gleichung gegeben ist, in der auch Terme  $f(\mathbf{x})$  auftreten, wird ausführlich in Abschnitt 6 eingegangen.

Die sinnvolle Vorgabe des Parameters  $t$  oder  $\lambda$  orientiert sich am (erwarteten) Lösungspfad in der Strukturmechanik, üblicherweise einer Last-Ver-schiebungskurve (Bild 2.1).

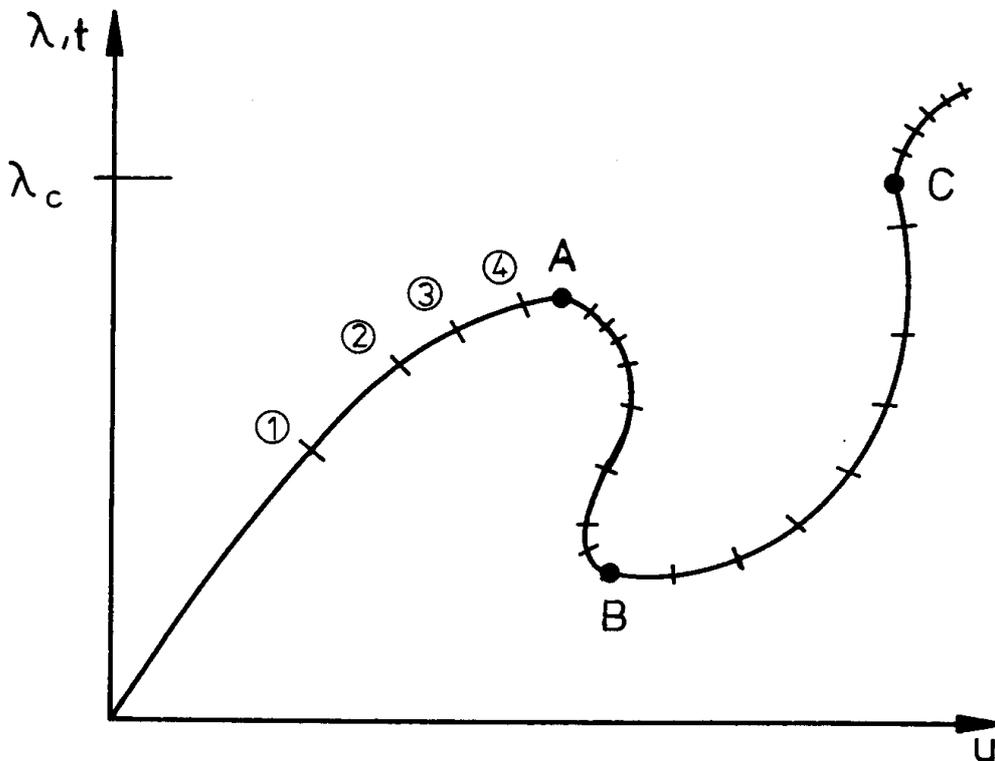


Bild 2.1 Allgemeiner, nichtlinearer Lösungspfad, Inkrementierung

Da der Lastparameter  $\lambda$  meist Schwankungen unterworfen ist, oder z.B. eine Belastung allmählich aufgebracht wird, genügt es meist nicht, nur die Lösung für einen Punkt A, B oder C der Kurve zu ermitteln, sondern es muß der Lösungspfad in wesentlichen Punkten bekannt sein. D.h. es muß schrittweise bzw. inkrementell vorgegangen werden.

Ein zweiter Grund schrittweise vorzugehen, auch wenn nur ein Lösungspunkt gesucht ist, liegt in der Schwierigkeit, ein Lösungsverfahren zu finden, das z.B. als Lösung für den Lastwert  $\lambda_c$  den Lösungspunkt C in einem Schritt liefert.

Ein dritter Grund ist bei wegabhängigem Verhalten (Plastizität, Kontakt) vorhanden, bei dem zu einer richtigen Beurteilung eines Zustandes der vollständige Lösungspfad bis zur aktuellen Belastung bekannt sein muß.

## 2.1 Inkrementelle Verfahren

Die Unterteilung in einzelne Schritte kann soweit geführt werden, daß der Lösungspfad durch viele lineare Teilstücke gut angenähert werden kann. Die Linearisierung um den einzelnen Lösungspunkt  $\mathbf{x}^{(i)}$  liefert die Steigung der Teilstücke  $\mathbf{F}'^{(m)\mathbf{x}} \dots$ . Auf die Gewinnung und genauere Bedeutung von  $\mathbf{F}'^{(m)\mathbf{x}}$  wird später eingegangen, hier sei nur festgehalten, daß dieses Vorgehen zu dem bekannten Euler-Verfahren führt.

$$\mathbf{F}'^{(m)\mathbf{x}} \cdot \Delta^{m+1}\mathbf{x} = \mathbf{P}^{(m+1)\lambda} - \mathbf{P}^{(m)\lambda} = \Delta^{m+1}\mathbf{P} \quad (2.2)$$

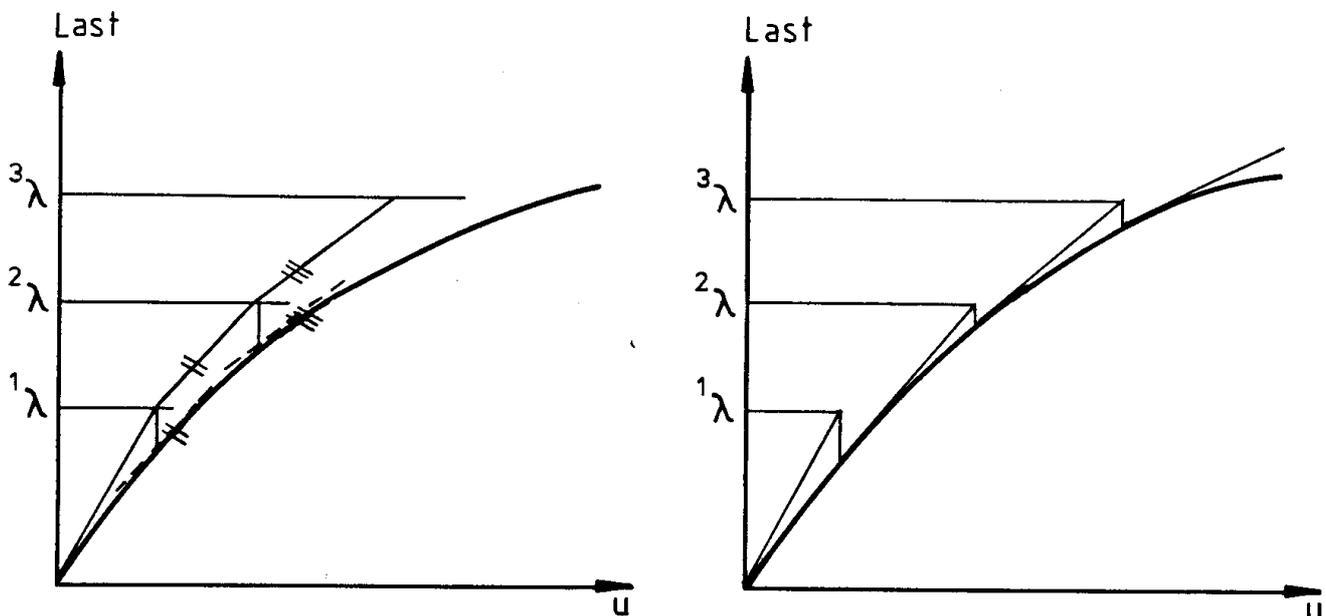
mit  $\Delta^{m+1}\mathbf{x}$  Zuwachs im Schritt  $m + 1$   
 $\mathbf{P}^{(m)\lambda}$  Gesamtlast im Schritt  $m$   
 $\Delta^{m+1}\mathbf{P}$  Lastzuwachs im Schritt  $m + 1$

Für einen Zuwachs der äußeren Last wird ein entsprechender Zuwachs der Unbekannten  $\mathbf{x}$  ermittelt. Deutlich ist, daß damit eine gewisse Abweichung vom eigentlichen Lösungspfad vorhanden ist, die während der Berechnung nicht korrigiert wird (Bild 2.2a).

Eine Korrektur kann auf einfache Weise erfolgen, indem das Residuum in jedem Schritt berücksichtigt wird.

$$\mathbf{F}'^{(m)\mathbf{x}} \cdot \Delta^{m+1}\mathbf{x} = -\mathbf{F}^{(m)\mathbf{x}, (m+1)\lambda} = \Delta^{m+1}\mathbf{P} - {}^m\mathbf{F} \quad (2.3)$$

Dieses Euler-Verfahren mit Rückwärtseinsetzen kann ein Anwachsen der Abweichung vom exakten Lösungspfad verhindern (Bild 2.2b). Die gewonnene Lösung stellt aber nur eine Umschließung des exakten Lösungspfades dar.



a) Euler Verfahren

b) Euler Verfahren mit Rückwärtseinsetzen

**Bild 2.2** Rein inkrementelle Verfahren (Ein-Dimensionales Problem)

## 2.2 Inkrementell-iterative Verfahren

Die Lösung im Lastschritt (= Inkrement) kann weiter verbessert werden, indem das Residuum als Fehler der Lösung iterativ beseitigt wird. Dies erfolgt in jedem Lastschritt vor einer weiteren Veränderung des Parameters  $\lambda$  (oder  $t$ ).

Zur Iteration steht eine Vielzahl von Lösungsverfahren zur Verfügung.

Die möglichen wesentlichen Unterschiede zur Lösung von

$$\mathbf{A}(\mathbf{x}^{(i)}) \cdot \Delta \mathbf{x} = \mathbf{Y}(\mathbf{x}^{(i)}) \quad (2.4)$$

sind:

1. Modifikation der Iterationsmatrix  $\mathbf{A}(\mathbf{x}^{(i)})$  und des Residuums  $\mathbf{Y}(\mathbf{x}^{(i)}) \hat{=} -\mathbf{F}(\mathbf{x}^{(i)}, \lambda^{(i)})$
2. Nur Modifikation des Residuums  $\mathbf{Y}(\mathbf{x}^{(i)})$  bei gleichbleibender Iterationsmatrix z.B.  $\mathbf{A} = \mathbf{A}(\mathbf{x}^{(0)})$
3. Nur Modifikation der Iterationsmatrix  $\mathbf{A}(\mathbf{x}^{(i)})$  bei gleichbleibender rechter Seite  $\mathbf{Y} = -\mathbf{F}(\mathbf{x}^{(0)}, \lambda^{(i)})$

Grafisch ist der Ablauf der Iteration in Bild 2.3 für alle drei Möglichkeiten exemplarisch an einem eindimensionalen Problem dargestellt.

Die wesentlichen Verfahren zu den Möglichkeiten 1) und 2) werden ausführlich in den folgenden Abschnitten behandelt.

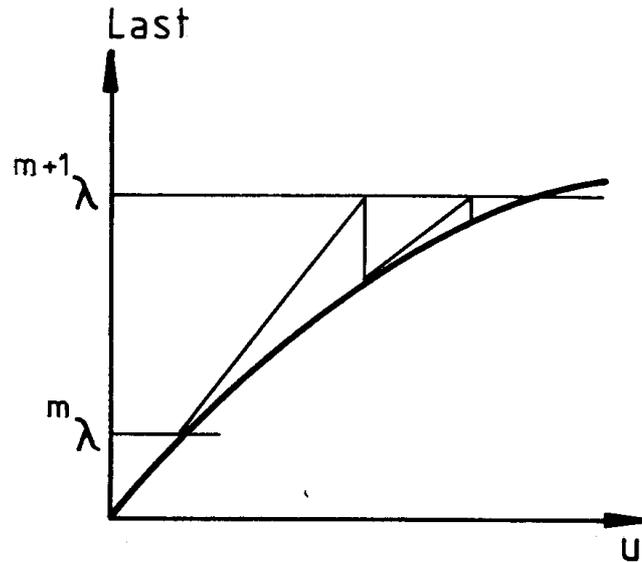
Die Möglichkeit 3) ist eines der klassischen Verfahren zur Lösung von nicht-linearen Gleichungssystemen, das auch unter dem Namen sukzessives Einsetzen oder gewöhnliches Iterationsverfahren bekannt ist. In dieser Form wird dies z.B. bei der Lösung von Problemen mit dem Übertragungsverfahren (Gesamtmatrixverfahren) verwendet, wenn die Koeffizienten der Matrix durch Integration gewonnen werden.

Die Iterationsmatrix  $\mathbf{A}(\mathbf{x}^{(i)})$  wird bei Möglichkeit 3) nicht als Tangenten- oder Jacobi Matrix des Systems bestimmt. Die rechte Seite hingegen stellt in jeder Iteration den aktuellen Gesamtlastzustand dar.

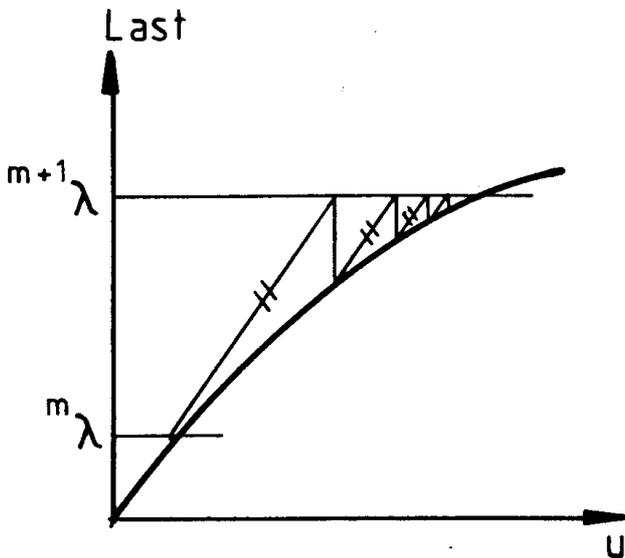
Diese Art von Sekantenverfahren ist üblicherweise sehr robust, besitzt aber keine besonders guten Konvergenzeigenschaften und wird daher in dieser Arbeit nicht weiter behandelt.

*Bemerkung 2.1:* Liegen zustandsabhängige Lastfälle vor, z.B. verformungsabhängige Belastungen oder ein Verschiebungslastfall, so ändert sich auch bei Möglichkeit 3 der Vektor der rechten Seite während der Iteration.

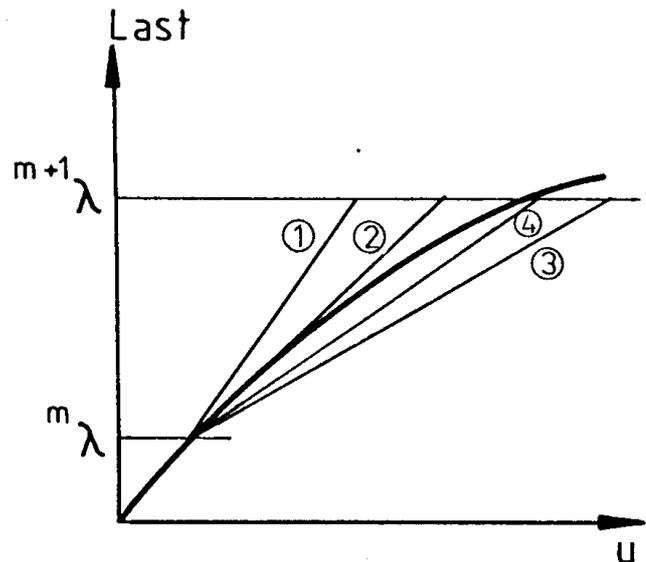
*Bemerkung 2.2:* Prinzipiell sind Kombinationen aller Verfahren miteinander möglich, z.B. Änderung der Iterationsmatrix nur in jeder dritten Iteration o.ä. Da diese Vorgaben aber starker Heuristik unterliegen, soll dies den Anwendern von Algorithmen nur als Wahlmöglichkeit gegeben werden. Eine Regel zur Anwendung läßt sich allgemein nicht angeben.



1) Iterationsmatrix und Rechte Seite



2) Nur Rechte Seite



3) Nur Iterationsmatrix

Änderung von:

- 1) Iterationsmatrix und Rechte Seite
- 2) Nur Rechte Seite
- 3) Nur Iterationsmatrix

**Bild 2.3 Inkrementell-iterative Verfahren - wesentliche Möglichkeiten**

### 3 Das Newton/Raphson Verfahren

Übersichten über Newton Verfahren sind in der klassischen Literatur häufig zu finden. Die im folgenden durchgeführte Kurzdarstellung lehnt sich in weiten Teilen an /Schwetlick 1979/ an (siehe auch /Mahnken 1984/).

#### 3.1 Nullstellensuche und Minimierungsproblem – echtes Newton Verfahren

Das Newton/Raphson Verfahren kann im Zusammenhang mit der Lösung nichtlinearer Probleme der Strukturmechanik an zwei Stellen eingesetzt werden. Diese sind

- i) die Nullstellensuche für ein nichtlineares Gleichungssystem
- ii) die Minimierungsaufgabe, die bei der Variationsformulierung der Mechanik anfällt.

In matrizieller Formulierung gilt für ein System mit  $n$  Unbekannten und Koeffizienten, die wiederum von Unbekannten abhängen :

$$\mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}), \dots, f_2(\mathbf{x}) \dots f_n(\mathbf{x})]^t = \mathbf{0} \quad (3.1)$$

$\mathbf{F}(\mathbf{x})$  kann als Vektor gedeutet werden, für den die Nullstelle(n) gesucht sind.  $f_i(\mathbf{x})$  wird als  $i$  te Komponente dieses Vektors bezeichnet.  $\mathbf{x}$  ist im allgemeinen Fall ebenfalls ein Vektor.

Wird hingegen von einem Potential ausgegangen, d.h. eine Minimierungsaufgabe gestellt, so nimmt das Gleichungssystem eine spezielle Form an.

$$\text{Funktion} = \text{Potential} \quad \Pi(\mathbf{x}), \quad \text{gesucht: Minimum} \quad (3.2)$$

Zur Suche dieses Minimums wird eine Näherung mittels einer Taylorreihe vorgenommen. Hierbei wird vorausgesetzt, daß die Funktion  $\Pi(\mathbf{x})$  zweimal differenzierbar ist. Außerdem wird hier zur Vereinfachung angenommen, daß ein Minimum existiert, die Funktion also konvex ist.

Bei einer quadratischen Näherung in einer Umgebung von  $\mathbf{x}^{(i)}$  gilt:

$$\tilde{\Pi}(\mathbf{x}) = \Pi(\mathbf{x}^{(i)}) + (\mathbf{x} - \mathbf{x}^{(i)})^t \Pi'(\mathbf{x}^{(i)}) + \frac{1}{2} (\mathbf{x} - \mathbf{x}^{(i)})^t \cdot \Pi''(\mathbf{x}^{(i)}) (\mathbf{x} - \mathbf{x}^{(i)}) \quad (3.3)$$

$\Pi'(\mathbf{x}^{(i)})$  ist der Gradient an der Stelle  $\mathbf{x}^{(i)}$ . In den Gleichungen der Strukturmechanik wird er auch als Fehlkraftvektor  $\mathbf{R}$  (out-of-balance vektor) bezeichnet.

$$\Pi'(\mathbf{x}^{(i)}) = \mathbf{R}(\mathbf{x}^{(i)}) = \left[ \frac{\partial \Pi(\mathbf{x}^{(i)})}{\partial x_1}, \frac{\partial \Pi(\mathbf{x}^{(i)})}{\partial x_2}, \dots, \frac{\partial \Pi(\mathbf{x}^{(i)})}{\partial x_n} \right]^t \quad (3.4)$$

$\Pi''(\mathbf{x}^{(i)})$  ist die Hesse-Matrix  $\mathbf{H}$  der Näherungsfunktion  $\tilde{\Pi}(\mathbf{x})$  an der Stelle  $\mathbf{x}^{(i)}$ . Diese entspricht bei einer Verschiebungsformulierung der Steifigkeitsmatrix der Mechanik.

$$\Pi''(\mathbf{x}^{(i)}) = \mathbf{H} = \begin{bmatrix} \frac{\partial^2 \Pi(\mathbf{x}^{(i)})}{\partial x_1^2} & \frac{\partial^2 \Pi(\mathbf{x}^{(i)})}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 \Pi(\mathbf{x}^{(i)})}{\partial x_1 \partial x_n} \\ \frac{\partial^2 \Pi(\mathbf{x}^{(i)})}{\partial x_n \partial x_1} & \frac{\partial^2 \Pi(\mathbf{x}^{(i)})}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 \Pi(\mathbf{x}^{(i)})}{\partial x_n \partial x_n} \end{bmatrix} \quad (3.5)$$

Da  $\tilde{\Pi}(\mathbf{x})$  minimiert werden soll, muß die erste Ableitung  $\tilde{\Pi}'(\mathbf{x})$  verschwinden

$$\begin{aligned} \tilde{\Pi}'(\mathbf{x}) &= \mathbf{0} = \mathbf{0} + \Pi'(\mathbf{x}^{(i)}) + \Pi''(\mathbf{x}^{(i)})(\mathbf{x} - \mathbf{x}^{(i)}) \\ &= \mathbf{R}(\mathbf{x}^{(i)}) + \mathbf{H}(\mathbf{x}^{(i)})(\mathbf{x} - \mathbf{x}^{(i)}) \end{aligned} \quad (3.6)$$

Dies nun ist ein lineares Gleichungssystem mit nichtlinearen Koeffizienten, wobei zu bemerken ist, daß  $\mathbf{H}(\mathbf{x}^{(i)})$  auf Grund der Herleitung symmetrisch ist.

$$\frac{\partial^2 \Pi}{\partial x_k \partial x_c} = \frac{\partial^2 \Pi}{\partial x_c \partial x_k} \quad (3.7)$$

Der Lösungsalgorithmus bietet sich direkt an. Iterativ kann die Nullstelle für das Gleichungssystem gesucht werden; vorausgesetzt wird, daß  $\mathbf{H}$  nicht singulär ist.

$$\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} - \mathbf{H}(\mathbf{x}^{(i)})^{-1} \cdot \mathbf{R}(\mathbf{x}^{(i)}) \quad (3.8)$$

In der mathematischen Literatur wird dies umgeformt mit

$$\begin{aligned} \mathbf{s}^{(i)} &= -\mathbf{H}(\mathbf{x}^{(i)})^{-1} \cdot \mathbf{R}(\mathbf{x}^{(i)}) \\ \mathbf{x}^{(i+1)} &= \mathbf{x}^{(i)} + \mathbf{s}^{(i)} \end{aligned} \quad (3.9)$$

Ist nach  $(i + k)$  Iterationen die Nullstelle für  $\tilde{\Pi}'(\mathbf{x}^{(i+k)})$  gefunden, so ist  $\mathbf{x}^{(i+k)}$  ein Lösungsvektor für das Minimalproblem, wenn die zweite Ableitung  $\Pi''(\mathbf{x}^{(i+k)})$  positiv ist. Dies entspricht der Fragestellung:

$$\det \mathbf{H}(\mathbf{x}^{(i+k)}) > 0 \quad ? \quad (3.10)$$

Das heißt, daß zur Minimierung der als Näherung angenommenen quadratischen Funktion die zugehörige Hesse-Matrix auch positiv definit sein muß.

Wird analog zur quadratischen Näherung für das Funktional das nichtlineare Gleichungssystem in eine Taylorreihe an der Stelle  $\mathbf{x}^{(i)}$  entwickelt, die nach dem linearen Glied abgebrochen wird, so gilt:

$$\tilde{\mathbf{F}}^{(i)}(\mathbf{x}^{(i)}) = 0 = \mathbf{F}(\mathbf{x}^{(i)}) + \mathbf{F}'(\mathbf{x}^{(i)}) \cdot (\mathbf{x} - \mathbf{x}^{(i)}) \quad (3.11)$$

Voraussetzung hierbei ist, daß  $\mathbf{F}(\mathbf{x})$  einmal stetig differenzierbar ist.

$$\mathbf{F}'(\mathbf{x}^{(i)}) = \mathbf{J}(\mathbf{x}^{(i)}) = \begin{bmatrix} \frac{\partial f_1(\mathbf{x}^{(i)})}{\partial x_1} & \frac{\partial f_2(\mathbf{x}^{(i)})}{\partial x_2} & \dots & \frac{\partial f_n(\mathbf{x}^{(i)})}{\partial x_1} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_1(\mathbf{x}^{(i)})}{\partial x_n} & \frac{\partial f_2(\mathbf{x}^{(i)})}{\partial x_n} & \dots & \frac{\partial f_n(\mathbf{x}^{(i)})}{\partial x_n} \end{bmatrix} \quad (3.12)$$

$\mathbf{J}(\mathbf{x}^{(i)})$  wird üblicherweise als Jacobimatrix des nichtlinearen Gleichungssystems  $\mathbf{F}(\mathbf{x})$  bezeichnet. Die Jacobimatrix ist nicht notwendigerweise symmetrisch oder positiv definit.

*Bemerkung 3.1:* Werden nichtlineare Gleichungssysteme über Gleichgewichtsüberlegungen, das Prinzip der virtuellen Arbeiten oder z.B. eine schwache Form entwickelt, erhebt sich häufig die Frage, ob für das betrachtete Problem ein Funktional bzw. Potential vorliegt und ob ein Minimierungsproblem gelöst wird. Diese Frage kann mittels der Untersuchung der Jacobimatrix beantwortet werden. Ist die Jacobimatrix symmetrisch, wird ein Extremalproblem bearbeitet. Ist sie zusätzlich positiv definit, so wird eine Minimierungsaufgabe gelöst. D.h. es muß ein zugehöriges Funktional vorliegen, siehe auch /Marsden, Hughes 1983/.

Der Nullstellensuchalgorithmus für das nichtlineare Gleichungssystem ist üblicherweise identisch mit dem Lösungsalgorithmus des Gleichungssystems für die Minimierungsaufgabe.

### Konvergenzverhalten:

Für ein quadratisches Funktional besitzt das Newton Verfahren quadratische Konvergenz /Schwetlick 1979/, d.h. die Fehlernorm nimmt quadratisch ab.

$$\alpha \cdot \|\mathbf{x}^{(i)} - \mathbf{x}^{(i+2)}\|^2 \geq \|\mathbf{x}^{(i+1)} - \mathbf{x}^{(i)}\| \quad i = 0, 1 \dots \quad (3.13)$$

Für nichtquadratische Funktionale ist quadratische Konvergenz nur in der näheren Umgebung der Nullstelle gesichert.

### Numerische Aspekte

Fehler aus vorherigen Schritten führen bei der Nullstellensuche und bei Minimierungsproblemen nicht zur Divergenz, sondern verschlechtern nur das Gesamtkonvergenzverhalten. Das Verhalten vom aktuellen Schritt  $\mathbf{x}^{(i)}$  bis zur Lösung  $\mathbf{x}^{(n)}$  ist jedoch nur von den Schritten  $> i$  abhängig.

Für nichtquadratische Funktionale ist die Schrittweite und damit der Startvektor  $\mathbf{x}^{(0)}$  von Bedeutung. Liegt der Startvektor nicht im "quadratischen" Einzugsbereich der Lösung, kann Divergenz auftreten. Bei Ingenieurproblemen heißt dies, daß für die Berechnung einer Last/Verschiebungskurve die Schrittgröße klein genug gewählt werden muß.

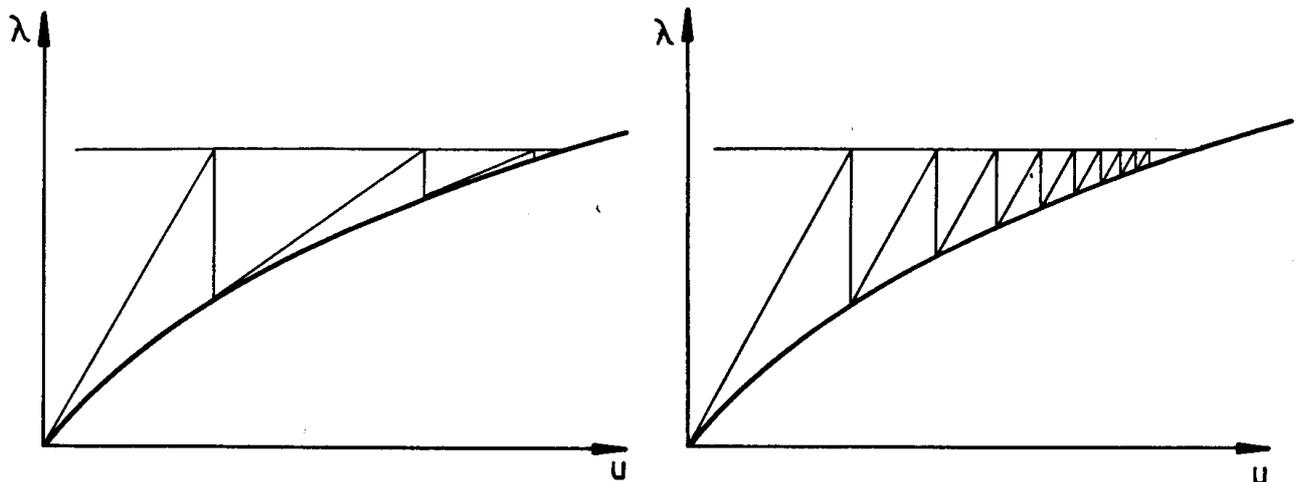
Die Aufstellung der Jacobi/Hesse- bzw. Steifigkeitsmatrizen gehört zu den numerisch aufwendigsten Prozessen in der F.E. Methode. Auch bei quadratischer Konvergenz ist die Effizienz des Verfahrens dann oft fraglich.

## 3.2 Modifizierte Newton Verfahren

Da die Aufstellung der Jacobi/Hesse - Matrix sehr aufwendig ist, wird versucht den Rechenaufwand durch Umgehung einer Neuaufstellung zu verringern.

Beim einfachsten Verfahren wird die Jacobi - Matrix nur einmal erstellt und bleibt für alle weiteren Iterationsschritte unverändert. Dieses Verfahren wird als das modifizierte Newton Verfahren bezeichnet; im Ingenieurschrifttum findet man hierfür die Begriffe Anfangssteifigkeitsiteration bzw. "initial load matrix iteration". Diese Vereinfachung macht sich beim Konvergenzverhalten bemerkbar. Für ein quadratisches Funktional ist die Konvergenz nur noch linear; der Hauptaufwand der Berechnung liegt dann in der häufigen Aufstellung der Gradienten. Bei nicht quadratischen Funktionalen mit starken Veränderungen der Gradienten kann die Konvergenzrate wesentlich verringert werden. In vielen Fällen wird dann ohne wesentliche Schrittweitenverringern keine Konvergenz mehr erreicht.

In einer häufig verwendeten Variante des modifizierten Newton Verfahrens wird die Jacobimatrix nur für einige Schritte beibehalten und danach wieder neu aufgestellt. Damit kann der Einzugsbereich des modifizierten Verfahrens vergrößert werden.



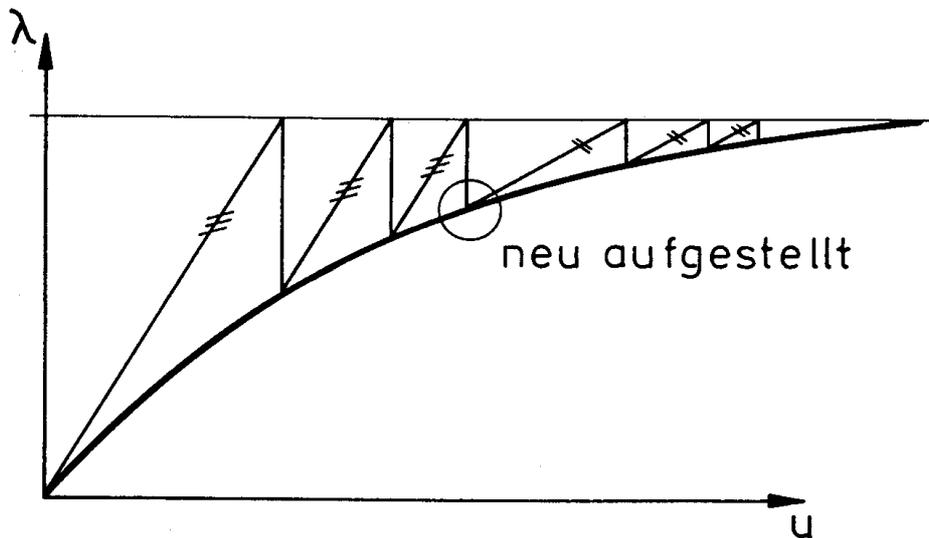
a) Standard Newton Verfahren

b) Modifiziertes Newton Verfahren

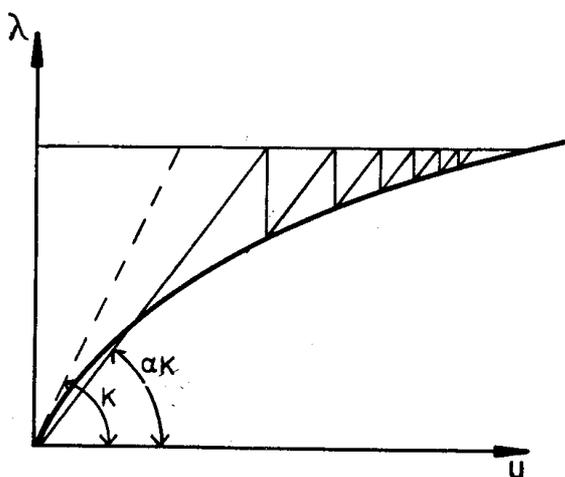
**Bild 3.1 Standard Newton und modifiziertes Newton Verfahren (einfachste Form)**

In anderen Varianten wird die Jacobimatrix mit einem Faktor multipliziert ( $\hat{=}$  Relaxation). In den folgenden Schritten wird dann mit dieser Matrix iteriert.

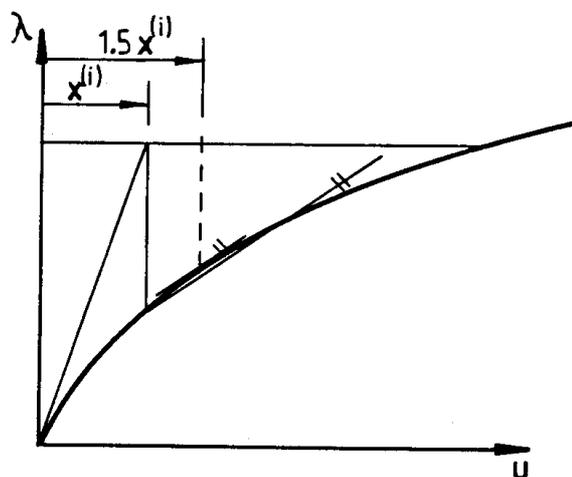
Eine weitere Variante stellt die Mittelpunktsteifigkeitsmethode dar, bei der die Jacobimatrix bei einer Neuaufstellung nicht für die Stelle  $\mathbf{x}^{(i)}$  sondern für  $(\mathbf{x}^{(i)} + 0.5 \cdot \mathbf{x}^{(i)})$  aufgestellt wird. Damit wird mit einer "Sekantenmatrix" iteriert, die zumindest im Bereich  $\mathbf{x}^{(i)} < (\mathbf{x}^{(i)} + 0.5 \cdot \mathbf{x}^{(i)})$  eine Sekantenmatrix darstellt.



a) Neuaufstellung der Jakobimatrix im Inkrement



b) Multiplikation der Jakobimatrix  
mit einem Faktor  $\alpha$



c) Mittelpunktsteifigkeit

### Bild 3.2 Varianten des modifizierten Newton Verfahrens

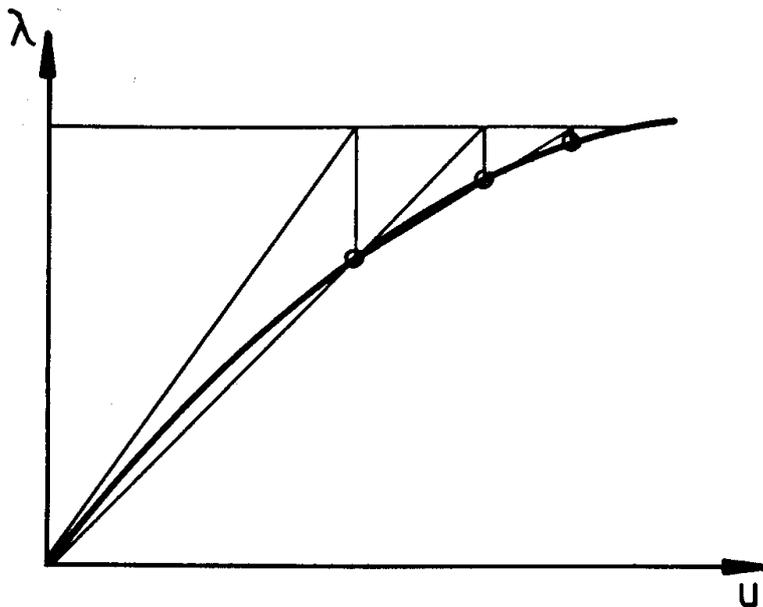
Im Prinzip sind sehr viele Modifikationen des Newton Verfahrens möglich, die für einzelne Probleme gute d.h. mehr als lineare Konvergenz aufweisen. Dies sind aber häufig Einzellösungen, deren Beurteilung sich nicht verallgemeinern läßt. Bessere Konvergenz ist im allgemeinen Fall nur mit Zusatzaufwand, z.B. der Suche von guten Relaxationsfaktoren oder Kombination mit Iterationsverfahren zu erreichen.

## 4 Sekantenverfahren

### 4.1 Allgemeines

Unter Sekantenverfahren werden Verfahren verstanden, bei denen die Jacobi/Hesse – Matrizen in den einzelnen Schritten über Sekantenbeziehungen aufgestellt bzw. angepaßt werden.

Im allgemeinen Fall können Sekantenverfahren als diskretisierte Newton-Verfahren interpretiert werden /Ortega, Rheinboldt 1970 S. 189 ff/



**Bild 4.1** Sekantenverfahren

Eindimensionaler Fall:

Diskretes Newton Verfahren

$$x^{(k+1)} = x^{(k)} - \left[ \frac{f(x^{(k)} + h^{(k)}) - f(x^{(k)})}{h^{(k)}} \right]^{-1} f(x^{(k)}) \quad (4.1)$$

Als Spezialfall die Regula-Falsi-Iteration

$$x^{(k+1)} = x^{(k)} - \left[ \frac{f(\bar{x}) - f(x^{(k)})}{\bar{x} - x^{(k)}} \right]^{-1} f(x^{(k)}) \quad (4.2)$$

mit  $h^{(k)} = \bar{x} - x^{(k)}$ , wobei  $\bar{x}$  fest vorgegeben wird.

Sekanten-Iteration

$$x^{(k+1)} = x^{(k)} - \left[ \frac{f(x^{k-1}) - f(x^{(k)})}{x^{k-1} - x^{(k)}} \right] f(x^{(k)}) \quad (4.3)$$

mit  $h^{(k)} = x^{k-1} - x^{(k)}$

Für viele Unbekannte lassen sich mit Hilfe von  $n + 1$  Schätzwerten oder vorliegenden Lösungen Sekantenmatrizen aufbauen. Der numerische Aufwand entspricht wegen der vielen Funktionsauswertungen mindestens dem eines Newton-Verfahrens. /Ortega, Rheinboldt 1970/

Bei der Sekantenbildung entstehen vollbesetzte Matrizen, die den Gesamtaufwand wegen der erforderlichen Gleichungsauflösung pro Schritt wesentlich erhöhen. Da außerdem immer  $n + 1$  Lösungsvektoren gespeichert werden müssen, sind diese Verfahren nach Gl. 4.1 - 4.3 nur für kleine Probleme zu empfehlen.

Einen weit größeren Anwendungsbereich finden Sekantenverfahren, die auf der Modifikation vorliegender Tangenten- oder Sekantenmatrizen beruhen, für die als Oberbegriff auch die Bezeichnung Modifikationsverfahren zu finden ist. Im allgemeinsten Fall wird die Einheitsmatrix als Ausgangsmatrix verwendet. Müssen die Modifikationen Sekantenbeziehungen genügen, so werden diese Verfahren als Quasi-Newton Verfahren bezeichnet.

**Modifikation von Matrizen und Inversen**

Ihre häufige Verwendung verdanken diese Verfahren der einfachen Modifikationsform, bei der die Ausgangsmatrix durch die Addition von Rang-Eins Matrizen verändert wird.

$$\mathbf{A}^{(k+1)} = \mathbf{A}^{(k)} + \mathbf{u}^{(k)} \cdot \mathbf{v}^{(k)t} \quad (4.4)$$

Im allgemeinen Fall erfolgt eine Änderung mit vielen ( $m$ ) Vektoren, wobei, wenn  $\mathbf{A}$  eine Matrix mit  $(n \cdot n)$  Zeilen/Spalten ist, die Zahl der Modifikationen  $m \leq n$  sein soll.  $\mathbf{U}, \mathbf{V}$  sind dann Matrizen, die die Modifikationsvektoren enthalten.

$$\mathbf{A}^{(k+1)} = \mathbf{A}^{(k)} + \mathbf{U} \cdot \mathbf{V}^t \quad (4.5)$$

Die einfache Modifikationsform führt dazu, daß die Inverse der modifizierten Matrix ebenfalls einfach darstellbar ist.

Im allgemeinen Fall gilt:

*Sherman-Morrison-Woodbury Formel*

$$(\mathbf{A} + \mathbf{UV}^t)^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1} \cdot \mathbf{U}(\mathbf{I} + \mathbf{V}^t \mathbf{A}^{-1} \mathbf{U})^{-1} \mathbf{V}^t \mathbf{A}^{-1} \quad (4.6)$$

Die Inversion ist durchführbar, wenn  $\mathbf{A}$  und  $(\mathbf{I} + \mathbf{V}^t \mathbf{A}^{-1} \mathbf{U})$  invertierbar sind.

Im Spezialfall  $m = 1$ , d.h. Änderung um eine Rang Eins - Matrix gilt:

*Sherman-Morrison-Formel:*

$$(\mathbf{A} + \mathbf{uv}^t)^{-1} = \mathbf{A}^{-1} - \frac{1}{1 + \mathbf{v}^t \mathbf{A}^{-1} \mathbf{u}} \mathbf{A}^{-1} \mathbf{uv}^t \mathbf{A}^{-1} \quad (4.7)$$

Dies ist durchführbar, wenn  $\mathbf{A}$  invertierbar und  $(1 + \mathbf{v}^t \mathbf{A}^{-1} \mathbf{u}) \neq 0$  ist.

Die Entdeckung der speziellen Inversen wird /Shermann, Morrison 1947/ zugeschrieben. Eine schöne Übersicht über Inverse modifizierter Matrizen ist in /Zielke 1970/ zu finden, siehe auch /Zielke 1968/.

In der Strukturmechanik entspricht  $\mathbf{A}$  üblicherweise der Steifigkeitsmatrix und ist invertierbar. Bei sinnvoller Wahl der Modifikationsvektoren  $\mathbf{v}$  und  $\mathbf{u}$  ist außerdem gesichert, daß  $\varphi = 1 + \mathbf{v}^t \mathbf{A}^{-1} \mathbf{u} \neq 0$  ist.

## 4.2 Quasi - Newton Verfahren

Die Entwicklung der Quasi-Newton Verfahren baut auf zwei Strängen auf. Der erste ausgehend von /Davidon 1959/ wurde bei der Lösung von Minimierungsaufgaben entwickelt. Der zweite diente nach einem Vorschlag von /Broyden 1965/ zur Lösung von nichtlinearen Gleichungssystemen. Beide Arbeiten haben den Grundstein zu einer neuen Klasse von Algorithmen gelegt, die unter den Namen Quasi Newton, Variable Metrik, Sekanten, Update oder Modifikationsverfahren bekannt sind. Einführungen in diese Verfahren sind in vielen Büchern zu finden /Ortega, Rheinboldt 1970/, /Murray 1972/, /Wolfe 1975/, /Luenberger 1984/, /Schwetlick 1979/. Eine grundlegende Übersicht geben /Dennis und More 1977/. In der vorliegenden Arbeit wird nur auf die vier gebräuchlichsten Verfahren, das Davidon-, das Broyden-, das Davidon- Fletcher- Powell- (DFP) und das Broyden- Fletcher- Goldfarb-

Shanno (BFGS)- Verfahren eingegangen. Ihre Vor- und Nachteile bezüglich Konvergenz, numerischer Empfindlichkeit und Effizienz werden diskutiert.

### Quasi-Newton Approximation der Jacobi/Hessematrix

Die Anforderung an die Approximation der Jacobi – Matrix lautet:

$$\mathbf{F}(\mathbf{x}^{(k)}) - \mathbf{F}(\mathbf{x}^{(k-1)}) = \mathbf{A}^{(k)}(\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)})$$

In der Strukturmechanik entspricht  $\mathbf{A}$  der Steifigkeitsmatrix  $\mathbf{K}$ ,  $\mathbf{F}$  den Ungleichgewichtskräften  $\mathbf{r}$  und  $\mathbf{x}$  den Lagevektoren.

$$\mathbf{r}^{(k)} - \mathbf{r}^{(k-1)} = \mathbf{K}^{(k)} \cdot \Delta \mathbf{u}^{(k)} \tag{4.8}$$

$$\text{mit } \Delta \mathbf{u}^{(k)} = \mathbf{x}^{(k)} - \mathbf{x}^{(k-1)} \tag{4.9}$$

Diese Beziehung wird als Quasi-Newton Gleichung bezeichnet. Im eindimensionalen Fall ist diese Sekantenbeziehung leicht darstellbar (Bild 4.2).

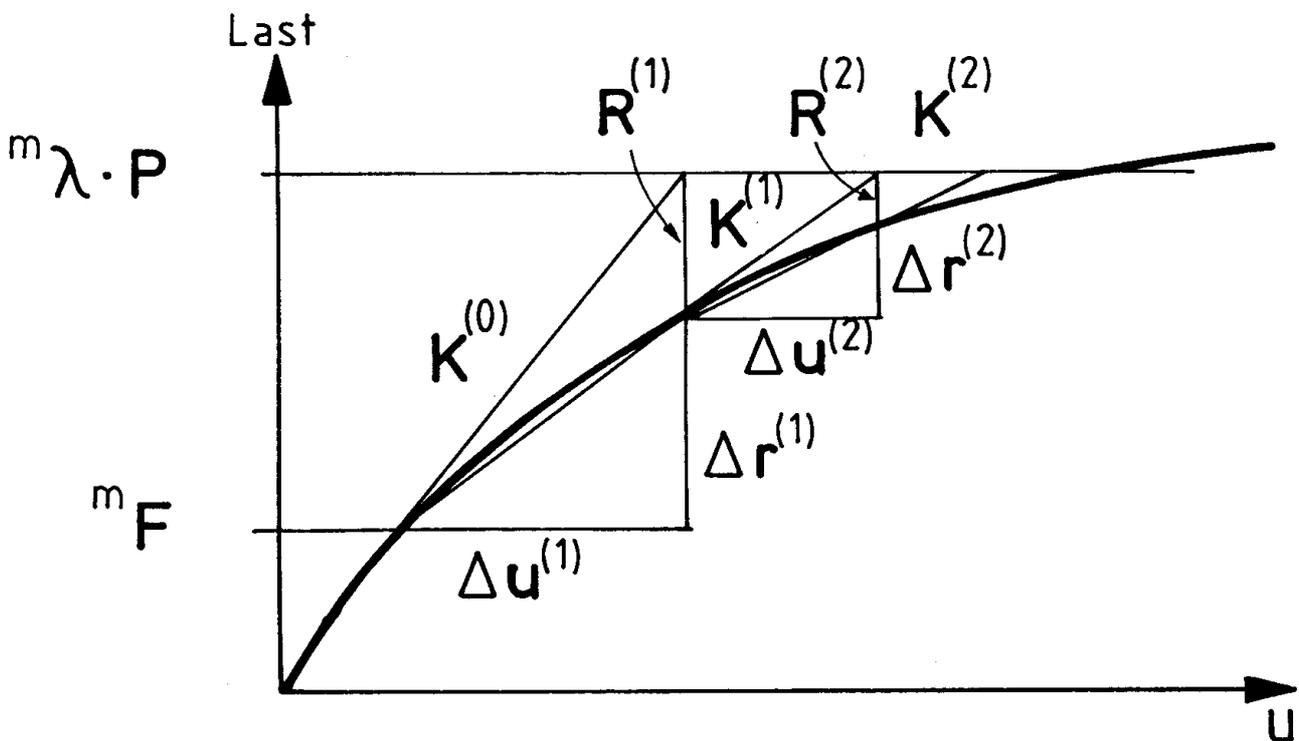


Bild 4.2 Quasi Newton Schema

Die neuen Matrizen  $\mathbf{A}^{(k+1)}$  bzw.  $\mathbf{K}^{(k+1)}$  haben die Sekantenbeziehung zu erfüllen  $\Delta \mathbf{u}^{(k)}$  ist der Verschiebungsvektor in der  $k$  - ten Iteration.

$\Delta \mathbf{r}^{(k)}$  ist die Differenz der Ungleichgewichtskräfte zwischen der  $k$  - ten und  $(k - 1)$ ten Iteration

$$\begin{aligned}\Delta \mathbf{r}^{(k)} &= \mathbf{r}^{(k-1)} - \mathbf{r}^{(k)} \\ &= (\mathbf{R}^{(k-1)} - \mathbf{F}^{(k-1)}) - (\mathbf{R}^{(k)} - \mathbf{F}^{(k)})\end{aligned}\quad (4.10)$$

mit:

$\mathbf{F}^{(k)}$  ... innere Kräfte im  $i$  - ten Inkrement und der  $k$  - ten Iteration

$\mathbf{R}^{(k)}$  ... äußere Kräfte im  $i$  - ten Inkrement und der  $k$  - ten Iteration

*Bemerkung 4.1:* Vorausgesetzt wird ein sogenanntes inkrementell iteratives Vorgehen. Hierbei wird bei dem Einbettungsproblem  $\mathbf{H}(\mathbf{x}, t) = \mathbf{0}$  das Nullstellenproblem für eine Familie von Parametern  $t = t_1, t_2, \dots, t_m$  dh.  $\mathbf{C}_i(\mathbf{x}) = \mathbf{H}(\mathbf{x}, t_i) = \mathbf{0}$  gelöst. In der Strukturmechanik entspricht  $t$  meist einem Lastfaktor (siehe Abschnitt 6).

### Modifikationsmöglichkeiten

Viele Matrizenformulierungen erfüllen die Quasi - Newton Gleichung. Für technische Anwendungen sind zwei Arten der Modifikationsverfahren (matrix update) von Bedeutung:

Rang - Eins Änderungen (rank one update)

$$\mathbf{K}^{(k)} = \mathbf{K}^{(k-1)} + \alpha \cdot \mathbf{z} \cdot \mathbf{x}^t \quad (4.11)$$

Rang - Zwei Änderungen (rank two updates)

$$\mathbf{K}^{(k)} = \mathbf{K}^{(k-1)} + \alpha \cdot \mathbf{z} \cdot \mathbf{x}^t + \beta \cdot \mathbf{v} \cdot \mathbf{y}^t \quad (4.12)$$

#### 4.2.1 Rang - Eins Änderungen

Wird Gleichung (4.11) in Gleichung (4.8) eingesetzt, so ergeben sich zwei Bedingungen:

$$\alpha = \frac{1}{\mathbf{x}^t \cdot \Delta \mathbf{u}^{(k)}} \quad (4.13)$$

$$\mathbf{z} = \Delta \mathbf{r}^{(k)} - \mathbf{K}^{(k-1)} \cdot \Delta \mathbf{u}^{(k)} \quad (4.14)$$

$\mathbf{x}$  ist hierbei ein beliebiger Vektor, der so zu wählen ist, daß  $\mathbf{x}^t \Delta \mathbf{u}^{(k)} \neq 0$  ist. D.h.  $\mathbf{x}$  darf nicht orthogonal zu  $\Delta \mathbf{u}^{(k)}$  sein. Nur dann ist diese Rang Eins Änderung möglich.

Aus der Vielzahl der möglichen Rang – Eins Änderungen wird hier auf die Vorschläge von /Broyden 1965/ und /Davidon 1959/ ausführlich eingegangen. Zur Diskussion anderer Vorschläge wie Powell Symmetric Broyden (PSB) /Powell 1970/ sowie /Kleinmichel 1981/ sei auf die Bemerkungen und das weiterführende Schrifttum verweisen.

### Broyden Rang-Eins Änderung

Broydens Vorschlag /1965/ mit

$$\mathbf{x} = \Delta \mathbf{u}^{(k)} = s \mathbf{K}^{(k-1)^{-1}} \mathbf{r}^{(k-1)} \quad (4.15)$$

$s \dots$  Line Search Parameter (siehe Abschnitt 5)

und

$$\mathbf{K}^{(k)} = \mathbf{K}^{(k-1)} + \frac{\Delta \mathbf{r}^{(k)} - \mathbf{K}^{(k-1)} \cdot \Delta \mathbf{u}^{(k)}}{(\Delta \mathbf{u}^{(k)})^t \Delta \mathbf{u}^{(k)}} (\Delta \mathbf{u}^{(k)})^t \quad (4.16)$$

basiert auf der Idee, daß die Matrizen  $\mathbf{K}^{(k-1)}$  und  $\mathbf{K}^{(k)}$  für den Fall, daß  $\mathbf{x}$  orthogonal zu  $\Delta \mathbf{u}^{(k)}$  ist, identisch sind bzw. sich auf einer Senkrechten zu  $\Delta \mathbf{u}^{(k)}$  gleich verhalten sollen.

Broydens Vorschlag ergibt eine nichtsymmetrische Sekanten-Matrix, die damit nur als Approximation zur Jacobi – Matrix und nicht zur Hesse-Matrix angesehen werden kann. Dies war auch das Ziel Broydens bei der Entwicklung eines effizienten Verfahrens zur Gleichungslösung.

Eine weitere Eigenschaft des Broyden'schen Vorschlags ist, daß aus der Menge der möglichen Rang – Eins Änderungen, die die Quasi – Newton Gleichung erfüllen, sich die Broyden Änderung  $\mathbf{K}^{(k)}$  als die der Matrix  $\mathbf{K}^{(k-1)}$  nächste Matrix erweist /Dennis, More 1977/. Als Maß dient hierbei die Frobenius Norm. /Schwarz et al 1968/

$$\|\mathbf{A}\| = (\sum_{i,k}^n |a_{ik}|^2)^{1/2}$$

Zur Verwendung des Verfahrens bei der Gleichungslösung muß die neue Matrix  $\mathbf{K}^{(k)}$  invertierbar (regulär) sein. Die Inversion erfolgt nach der Sherman-Morrison Formel: (siehe Anhang) /Zielke 1970/

$$\mathbf{K}^{(k)^{-1}} = \mathbf{K}^{(k-1)^{-1}} + \frac{(\Delta \mathbf{u}^{(k)} - \mathbf{K}^{(k-1)^{-1}} \Delta \mathbf{r}^{(k)}) \Delta \mathbf{u}^{(k)^t} \cdot \mathbf{K}^{(k-1)^{-1}}}{\Delta \mathbf{u}^{(k)^t} \cdot \mathbf{K}^{(k-1)^{-1}} \Delta \mathbf{r}^{(k)}} \quad (4.17)$$

Voraussetzung hierfür ist, daß der Nenner  $\neq 0$  ist.

$$\varphi = \Delta \mathbf{u}^{(k)t} \mathbf{K}^{(k-1)-1} \Delta \mathbf{r}^{(k)} \neq 0 \quad (4.18)$$

d.h. der Verschiebungszuwachs  $\Delta \mathbf{u}^{(k)}$  darf nicht orthogonal zu der Verschiebungsänderung  $\Delta \bar{\mathbf{u}} = \mathbf{K}^{(k-1)-1} \Delta \mathbf{r}^{(k)}$  infolge der Änderung der Ungleichgewichtskräfte sein.

*Bemerkung 4.2:* Die Determinante von Matrizen mit Rang Eins Änderungen läßt sich einfach bestimmen. Es ist siehe /Pearson 1969/:

$$\det(\mathbf{I} + \mathbf{v}\mathbf{w}^t) = 1 + \mathbf{v}^t\mathbf{w} \quad (4.19)$$

Damit gilt im Fall der Broyden Änderung:

$$\begin{aligned} \det(\mathbf{K}^{(k)}) &= \det\left[\mathbf{K}^{(k-1)} + \frac{(\Delta \mathbf{r}^{(k)} - \mathbf{K}^{(k-1)}\Delta \mathbf{u}^{(k)})\Delta \mathbf{u}^{(k)t}}{\Delta \mathbf{u}^{(k)t}\Delta \mathbf{u}^{(k)}}\right] \\ &= \det\mathbf{K}^{(k-1)} \cdot \det\left[\mathbf{I} + \frac{(\mathbf{K}^{(k-1)-1}\Delta \mathbf{r}^{(k)} - \Delta \mathbf{u}^{(k)})\Delta \mathbf{u}^{(k)t}}{\Delta \mathbf{u}^{(k)t}\Delta \mathbf{u}^{(k)}}\right] \\ &= \left(\frac{\Delta \mathbf{r}^{(k)t}\mathbf{K}^{(k-1)-1}\Delta \mathbf{u}^{(k)}}{\Delta \mathbf{u}^{(k)t}\Delta \mathbf{u}^{(k)}}\right) \cdot \det\mathbf{K}^{(k-1)} \end{aligned} \quad (4.20)$$

Da  $\Delta \mathbf{r}^{(k)t} \cdot \mathbf{K}^{(k-1)-1} \Delta \mathbf{u}^{(k)}$  auch  $\leq 0$  werden kann, wird in /Powell 1970/ empfohlen, die Rang - Eins Änderung mit einem Faktor  $\Theta$  zu multiplizieren. Damit kann gesichert werden, daß die neue Matrix nichtsingulär wird und positiv definit bleibt.

$$\mathbf{K}^{(k)} = \mathbf{K}^{(k-1)} + \Theta \frac{(\Delta \mathbf{r}^{(k)} - \mathbf{K}^{(k-1)}\Delta \mathbf{u}^{(k)})\Delta \mathbf{u}^{(k)t}}{\Delta \mathbf{u}^{(k)t}\Delta \mathbf{u}^{(k)}} \quad (4.21)$$

Die Determinante kann dabei mittels  $\Theta$  in einer festgelegten Relation zur alten Determinante  $\det \mathbf{K}^{(k-1)}$  gehalten werden.

$$\det\mathbf{K}^{(k)} = \det\mathbf{K}^{(k-1)}\left(1 - \Theta + \Theta \frac{\Delta \mathbf{r}^{(k)t}\mathbf{K}^{(k-1)-1}\Delta \mathbf{u}^{(k)}}{\Delta \mathbf{u}^{(k)t}\Delta \mathbf{u}^{(k)}}\right) \quad (4.22)$$

Da die Quasi Newton Verfahren im Rahmen dieser Arbeit auch zur Berechnung von Last-Verschiebungskurven mit negativen Determinanten eingesetzt werden sollten, wurde die geschilderte Variation mit  $\Theta$  nicht verwendet.

Die angegebene Formel kann aber gut für die Berechnung der augenblicklichen Determinante eingesetzt werden.

### Davidon Rang Eins Änderung

Davidons Vorschlag /1959/ mit

$$\mathbf{x} = \Delta \mathbf{r}^{(k)} - \mathbf{K}^{(k-1)} \cdot \Delta \mathbf{u}^{(k)} \quad (4.23)$$

und

$$\mathbf{K}^{(k)} = \mathbf{K}^{(k-1)} + \frac{\Delta \mathbf{r}^{(k)} - \mathbf{K}^{(k-1)} \cdot \Delta \mathbf{u}^{(k)}}{(\Delta \mathbf{r}^{(k)} - \mathbf{K}^{(k-1)} \Delta \mathbf{u}^{(k)})^t \Delta \mathbf{u}^{(k)}} (\Delta \mathbf{r}^{(k)} - \mathbf{K}^{(k-1)} \Delta \mathbf{u}^{(k)})^t \quad (4.24)$$

basiert auf dem Ziel, eine einfache Änderungsformel für Matrizen bei Minimierungsproblemen zu entwickeln. Diese Änderungsmatrix muß als Näherung zur Hesse Matrix symmetrisch sein.

Ein wesentlicher Nachteil der Änderungsmatrix ist, daß sie, wenn  $\mathbf{x}$  und  $\Delta \mathbf{u}^{(k)}$  orthogonal zu einander sind, d.h. bei  $\mathbf{x}^t \Delta \mathbf{u}^{(k)} = 0$  nicht definiert ist. Es sei weiter bemerkt, daß wegen der Unbestimmtheit des Nenners die positive Definitheit der Änderung nicht garantiert werden kann.

Die Inversion erfolgt wieder nach Sherman/Morrison:

$$\mathbf{K}^{(k)-1} = \mathbf{K}^{(k-1)-1} + \frac{(\Delta \mathbf{u}^{(k)} - \mathbf{K}^{(k-1)-1} \Delta \mathbf{r}^{(k)})(\Delta \mathbf{u}^{(k)} - \mathbf{K}^{(k-1)-1} \Delta \mathbf{r}^{(k)})^t}{(\Delta \mathbf{u}^{(k)} - \mathbf{K}^{(k-1)-1} \Delta \mathbf{r}^{(k)})^t \Delta \mathbf{r}^{(k)}} \quad (4.25)$$

Voraussetzung für die Invertierbarkeit ist, daß der Nenner  $\neq 0$  wird:

$$\varphi = (\Delta \mathbf{u}^{(k)} - \mathbf{K}^{(k-1)-1} \Delta \mathbf{r}^{(k)})^t \Delta \mathbf{r}^{(k)} \neq 0. \quad (4.26)$$

Die Determinante wird nach (4.19) zu

$$\det(\mathbf{K}^{(k)}) = \det(\mathbf{K}^{(k-1)}) \cdot \left( \frac{\Delta \mathbf{r}^{(k)t} \mathbf{K}^{(k-1)-1} \Delta \mathbf{r}^{(k)} - \Delta \mathbf{r}^{(k)t} \Delta \mathbf{u}^{(k)}}{\Delta \mathbf{r}^{(k)t} \Delta \mathbf{u}^{(k)} - \Delta \mathbf{u}^{(k)t} \mathbf{K}^{(k-1)} \Delta \mathbf{u}^{(k)}} \right) \quad (4.27)$$

mit

$$\Delta \mathbf{u}^{(k)} = s \cdot \mathbf{K}^{(k-1)-1} \cdot \mathbf{r}^{(k-1)}$$

und  $s \dots$  Line search Parameter (siehe Abschnitt 5).

wird der Ausdruck (4.27) zu

$$\det(\mathbf{K}^{(k)}) = \det(\mathbf{K}^{(k-1)}) \cdot \frac{\Delta \mathbf{r}^{(k)t} \mathbf{K}^{(k-1)^{-1}} [(1-s)\mathbf{r}^{(k-1)} - \mathbf{r}^{(k)}]}{\Delta \mathbf{u}^{(k)t} [(1-s)\mathbf{r}^{(k-1)} - \mathbf{r}^{(k)}]} \quad (4.28)$$

Für  $s = 1$ , d.h. ohne Strahlminimierung (Line Search) vereinfacht sich dies weiter zu

$$\det(\mathbf{K}^{(k)}) = \det(\mathbf{K}^{(k-1)}) \cdot \frac{\Delta \mathbf{r}^{(k)t} \mathbf{K}^{(k-1)^{-1}} \cdot \mathbf{r}^{(k)}}{\Delta \mathbf{u}^{(k)t} \cdot \mathbf{r}^{(k)}} \quad (4.29)$$

Da innerhalb der benutzten Algorithmen alle Größen in 4.28/4.29 vorliegen, ist die Determinante leicht bestimmbar.

*Bemerkung 4.3:* zu Rang-Eins Änderungen

Rang - Eins Änderungsmatrizen erscheinen sehr effizient. Die beiden diskutierten Formen können jedoch wesentliche Bedingungen nicht erfüllen.

1. Die in der Strukturmechanik vorliegenden Minimierungsprobleme führen zu symmetrischen Hesse Matrizen. Die Broyden Formel mit un-symmetrischen Rang-Eins Änderungen ist hingegen nur für die Gleichungslösung mit allgemeinen Jacobi Matrizen geeignet.
2. Das Minimierungsproblem erfordert außerdem, daß die Hessematrix positiv definit ist. Dies kann von der Davidon Änderung nicht garantiert werden.

In /Kleinmichel 1981/ wird hingegen eine Rang-Eins Änderung vorgeschlagen, die beide obengenannten Bedingungen erfüllt. Im Rahmen der vorliegenden Arbeit wurde dieses Verfahren numerisch nicht weiter untersucht. (siehe hierzu Anhang Teil B).

#### 4.2.2 Rang - Zwei Änderungen

Aus den vorab genannten Forderungen an die Aufdatierungsformeln - Symmetrie, positive Definitheit - für die Anwendung auf Minimierungsprobleme entwickelte /Powell 1970/ eine Rang - Zwei Formel für Broydens Änderung zur Erzeugung einer symmetrischen, positiv definiten Matrix.

## Entwicklung:

Quasi - Newton Bedingung

$$C_1 = A + \frac{(y - A \cdot s) \cdot c^t}{c^t s} \quad (4.30)$$

Symmetrie Bedingung

$$C_2 = \frac{1}{2}(C_1 + C_1^t) \quad (4.31)$$

Quasi-Newton Bedingung

$$C_2^{(k+1)} = C_2^{(k)} + \frac{(y - C_2^{(k)} \cdot s) \cdot c^t}{c^t s} \quad (4.32)$$

Symmetrie Bedingung

$$C_2^{(k+1)} = \frac{1}{2}(C_2^{(k+1)} + C_2^{(k+1)t}) \quad (4.33)$$

Diese Sequenz konvergiert zu

$$A^{(k)} = A^{(k-1)} + \frac{(y - A^{(k-1)} \cdot s)c^t + c(y - A^{(k-1)} \cdot s)^t}{c^t s} - \frac{(y - A^{(k-1)} \cdot s)^t s}{(c^t s)^2} \cdot c c^t \quad (4.34)$$

Mit  $c = s$  ergibt sich die Powell-Symmetrische-Broyden (PSB) Aufdatierung. Letztere hat wieder die Eigenschaft, daß die Frobeniusnorm der geänderten Matrix von allen Quasi-Newton Rang - Zwei Änderungen unter obengenannten Bedingungen der Ausgangsmatrix am nächsten kommt.

Dennoch hat sich die PSB Aufdatierung nicht als sehr erfolgreich erwiesen.

Die Erfüllung der **positiven Definitheit** erfolgt im nächsten Schritt. Für eine ausführliche Herleitung sei auf /Dennis & More 1977/ verwiesen. Die Bedingung lautet, daß alle Eigenwerte der aufdatierten Matrix größer als Null sind.

Wird die Rang - Zwei Aufdatierung umgeschrieben zu:

$$A^{(k)} = A^{(k-1)} + u_1 u_2^t + u_2 u_1^t \quad (4.35)$$

dann lautet die Determinante nach /Pearson 1969/

$$\det(\mathbf{I} + \mathbf{u}_1 \mathbf{u}_2^t + \mathbf{u}_2 \mathbf{u}_1^t) = (1 + \mathbf{u}_1^t \mathbf{u}_2)(1 + \mathbf{u}_1^t \mathbf{u}_2) - \mathbf{u}_1^t \mathbf{u}_1 \cdot \mathbf{u}_2^t \mathbf{u}_2$$

Über diese Bedingung läßt sich eine Familie von Rang - Zwei Aufdatierungsformeln entwickeln:

$$\mathbf{A}^{(k)} = \mathbf{A}^{(k-1)} - \frac{\mathbf{A}^{(k-1)} \mathbf{s} \mathbf{s}^t \mathbf{A}^{(k-1)}}{\mathbf{s}^t \mathbf{A}^{(k-1)} \mathbf{s}} + \frac{\mathbf{y} \mathbf{y}^t}{\mathbf{y}^t \mathbf{s}} + \tau \mathbf{w} \mathbf{w}^t \quad (4.36)$$

mit

$$\mathbf{w} = \sqrt{\mathbf{s}^t \mathbf{A}^{(k-1)} \mathbf{s}} \left[ \frac{\mathbf{y}}{\mathbf{y}^t \mathbf{s}} - \frac{\mathbf{A}^{(k-1)} \mathbf{s}}{\mathbf{s}^t \mathbf{A}^{(k-1)} \mathbf{s}} \right] \quad (4.37)$$

In ihrer ursprünglichen Form wurde diese allgemeine Darstellung von /Broyden 1967/ angegeben und später von /Fletcher 1970/ umgeformt. Beide Namen werden für dieselbe Formel verwendet: Broyden- $\beta$  Klasse bzw. Fletcher Familie.

Abhängig von  $\tau$  lassen sich verschiedene Aufdatierungsmatrizen erzeugen. Die bekanntesten sind für  $\tau = 1$  die Davidon- Fletcher- Powell (DFP) Aufdatierung und für  $\tau = 0$  die Broyden- Fletcher- Goldfarb- Shanno (BFGS) Aufdatierung.

Die allgemeine Form der Broyden Klasse wurde erst nach der unabhängigen Entwicklung der DFP und BFGS-Formeln entdeckt.

Die Ausgangsformel für die DFP Aufdatierung stammt von /Davidon 1959/.

$$\mathbf{A}^{(k)} = \left( \mathbf{I} - \frac{\mathbf{y} \cdot \mathbf{s}^t}{\mathbf{y}^t \mathbf{s}} \right) \mathbf{A}^{(k-1)} \left( \mathbf{I} - \frac{\mathbf{s} \mathbf{y}^t}{\mathbf{y}^t \mathbf{s}} \right) + \frac{\mathbf{y} \mathbf{y}^t}{\mathbf{y}^t \mathbf{s}} \quad (4.38)$$

Die Inverse lautet:

$$\mathbf{A}^{(k)-1} = \mathbf{A}^{(k-1)-1} + \frac{\mathbf{s} \mathbf{s}^t}{\mathbf{y}^t \mathbf{s}} - \frac{\mathbf{A}^{(k-1)-1} \mathbf{y} \mathbf{y}^t \cdot \mathbf{A}^{(k-1)-1}}{\mathbf{y}^t \mathbf{A}^{(k-1)-1} \mathbf{y}} \quad (4.39)$$

d.h. eine Inverse existiert nur, wenn  $\mathbf{A}^{(k-1)}$  regulär und  $\mathbf{y}^t \mathbf{s} \neq 0$  ist. Außerdem: Ist  $\mathbf{A}^{(k-1)}$  positiv definit und  $\mathbf{y}^t \mathbf{s} > 0$ , so ist auch  $\mathbf{A}^{(k)}$  positiv definit, denn

$$\det \mathbf{A}^{(k)} = \det \mathbf{A}^{(k-1)} \left[ \frac{\mathbf{y}^t \mathbf{A}^{(k-1)-1} \mathbf{y}}{\mathbf{y}^t \mathbf{s}} \right] \quad (4.40)$$

$\mathbf{A}^{(k)}$  ist dann nicht singulär, wenn  $\mathbf{y}^t \mathbf{A}^{(k-1)^{-1}} \neq 0$  ist.

Die Aufdatierungsformel des BFGS Verfahrens entspricht genau der inversen Form der DFP Aufdatierung, wenn außerdem  $\mathbf{s}$  und  $\mathbf{y}$  vertauscht werden.

$$\mathbf{A}^{(k)} = \mathbf{A}^{(k-1)} + \frac{\mathbf{y}\mathbf{y}^t}{\mathbf{y}^t\mathbf{s}} - \frac{\mathbf{A}^{(k-1)}\mathbf{s}\mathbf{s}^t\mathbf{A}^{(k-1)}}{\mathbf{s}^t\mathbf{A}^{(k-1)}\mathbf{s}} \quad (4.41)$$

Inverse der BFGS-Aufdatierung

$$\mathbf{A}^{(k)^{-1}} = \left(\mathbf{I} - \frac{\mathbf{s}\mathbf{y}^t}{\mathbf{y}^t\mathbf{s}}\right)\mathbf{A}^{(k-1)^{-1}}\left(\mathbf{I} - \frac{\mathbf{y}\mathbf{s}^t}{\mathbf{y}^t\mathbf{s}}\right) + \frac{\mathbf{s}\mathbf{s}^t}{\mathbf{y}^t\mathbf{s}} \quad (4.42)$$

Für die Inverse gelten die selben Bedingungen wie für die Inverse der DFP-Aufdatierung.

$\mathbf{A}^{(k)}$  ist für das BFGS Verfahren nicht singulär, wenn die Inverse  $\mathbf{A}^{(k)^{-1}}$  nicht singulär ist. Dies ist einfach zu überprüfen, da

$$\left(\mathbf{A}^{(k)BFGS}\right)^{-1} = \left(\mathbf{A}^{(k)DFP}\right)^{-1} + \mathbf{v}\mathbf{v}^t \quad (4.43)$$

mit

$$\mathbf{v} = \left(\mathbf{y}^t\mathbf{A}^{(k-1)^{-1}}\mathbf{y}\right)^{1/2}\left[\frac{\mathbf{s}}{\mathbf{y}^t\mathbf{s}} - \frac{\mathbf{A}^{(k-1)^{-1}}\mathbf{y}}{\mathbf{y}^t\mathbf{A}^{(k-1)^{-1}}\mathbf{y}}\right] \quad (4.44)$$

d.h. ist die Inverse der DFP Aufdatierung nicht singulär, so ist auch die Inverse der BFGS Aufdatierung nicht singulär, da die symmetrische Rang-Eins Änderung  $\mathbf{v}\mathbf{v}^t$  die Regularität nicht negativ beeinflusst.

Die Determinante der BFGS Aufdatierung lautet:

$$\begin{aligned} \det\mathbf{K}^{(k)} &= \det(\mathbf{K}^{(k-1)}) \cdot \det\left[\mathbf{I} + \frac{\mathbf{K}^{(k-1)^{-1}}\Delta\mathbf{r}^{(k)} \cdot \Delta\mathbf{r}^{(k)^t}}{\Delta\mathbf{u}^{(k)^t} \cdot \Delta\mathbf{r}^{(k)}} - \frac{\Delta\mathbf{u}^{(k)} \cdot \Delta\mathbf{u}^{(k)^t} \cdot \mathbf{K}^{(k-1)}}{\Delta\mathbf{u}^{(k)^t} \cdot \mathbf{K}^{(k-1)}\Delta\mathbf{u}^{(k)}}\right] \\ &= \det(\mathbf{K}^{(k-1)}) \cdot \frac{\Delta\mathbf{u}^{(k)^t}\Delta\mathbf{r}^{(k)}}{\Delta\mathbf{u}^{(k)^t}\mathbf{K}^{(k-1)}\Delta\mathbf{u}^{(k)}} \end{aligned} \quad (4.45)$$

### 4.2.3 Konvergenz

Die Eignung der geschilderten Aufdatierungsformeln für Probleme der Strukturmechanik ist davon abhängig, ob die Konvergenz zu den richtigen Lösungen gesichert ist. Die Konvergenz der betrachteten Verfahren wird wesentlich von der optimalen (perfekten) Schrittweite bestimmt. Da hierauf in einem gesonderten Abschnitt ausführlich eingegangen wird, wird hier nur eine Kurzform wiedergegeben.

Für jeden Lösungsvektor  $\Delta \mathbf{u}^{(k)}$  läßt sich die Funktion minimieren

$$f(\alpha) = f(\mathbf{x}^{(k)} + \alpha \cdot \Delta \mathbf{u}^{(k)}) \rightarrow \text{Minimum}$$

Das Minimum wird bestimmt mittels:

$$\frac{\partial f}{\partial \alpha} = f'(\mathbf{x}^{(k)} + \alpha_i \Delta \mathbf{u}^{(k)}) \cdot \Delta \mathbf{u}^{(k)} = \mathbf{F}^{(k)t}(\alpha_i) \cdot \Delta \mathbf{u}^{(k)} = 0 \quad (4.46)$$

d.h. dann sind der Gradient für die perfekte Schrittweite  $\alpha_i$  und der Lösungsvektor (Vektor der Suchrichtung)  $\Delta \mathbf{u}^{(k)}$  orthogonal zueinander.

Exakte (perfekte) Schrittweiten  $\alpha$  sind nur sehr schwer und mit großem Aufwand erreichbar. Daher ist es für praktische Probleme wichtig, die Konvergenz auch für nicht exakte Schrittweiten zu sichern.

Wird angenommen, daß die optimale Schrittweite  $\alpha_i$  mittels Strahlminimierung exakt oder auch asymptotisch exakt bestimmt werden kann, dann konvergieren alle Verfahren der Broydenklasse für  $\tau \in [0, 1]$  zur richtigen Lösung. Voraussetzung ist, daß das Funktional zweifach differenzierbar und stetig ist /Stoer 1984/. Die Konvergenz ist im Falle quadratischer Funktionen bei perfekter Schrittweite quadratisch und bei nicht-quadratischen Funktionen bei asymptotisch exakter Schrittweite überlinear.

Im Weiteren ist zu untersuchen, ob auch für nicht asymptotisch exakte Strahlminimierung ausreichende Konvergenz zu erzielen ist.

Für das DFP-Verfahren wird dies ausführlich in /Luenberger 1984/ dargestellt. Aus der Herleitung der DFP-Aufdatierung als ein Verfahren der konjugierten Richtungen wird gezeigt, daß das DFP-Verfahren wie das Verfahren der konjugierten Richtungen sehr sensibel gegenüber der Genauigkeit der Strahlminimierung ist. Wird die Strahlminimierung ungenau, geht bei ungünstiger Eigenwertstruktur der Hessematrix die Eigenschaft der konjugierten Richtungen schnell verloren und die Konvergenzrate sinkt. Konvergenz kann im ungünstigen Fall nicht mehr garantiert werden.

Das BFGS Verfahren konvergiert hingegen auch mit nicht exakter Strahlminimierung. /Powell 1976/ gibt zwei Ungleichungen an, deren Erfüllung die Konvergenz auch bei nicht exakter Strahlminimierung sichert.

Die erste Bedingung fordert, daß der Funktionswert kleiner wird, als die Summe aus altem Funktionswert plus der mit einem Faktor multiplizierten Änderung:

$$f(\mathbf{x}^{(k)} + \alpha_i \cdot \Delta \mathbf{u}^{(k)}) \leq f(\mathbf{x}^{(k)}) + c_1 \cdot \alpha_i \mathbf{F}^{(k)t} \cdot \Delta \mathbf{u}^{(k)} \quad (4.47)$$

Die zweite Bedingung sichert, daß die Strahlminimierung nicht zwischen zu kleinen Werten erfolgt

$$[\mathbf{F}(\mathbf{x}^{(k)} + \alpha_i \Delta \mathbf{u}^{(k)})]^t \cdot \Delta \mathbf{u}^{(k)} \geq c_2 \mathbf{F}^{(k)t} \Delta \mathbf{u}^{(k)} \quad (4.48)$$

wobei:  $0 < c_1 < \frac{1}{2}$

und  $c_1 < c_2 < 1$  zu erfüllen ist.

d.h. die Strahlminimierung hat nur innerhalb bestimmter Schranken zu erfolgen und sichert, daß die neuen Vektoren sich in der Umgebung befinden, in der  $f(\mathbf{x})$  konvex ist. Dies verhindert die Divergenz des Verfahrens und führt zu größerer numerischer Stabilität.

Die Reduktion des Aufwands für Verfahren ohne exakte Strahlminimierung ist beträchtlich, da die Ermittlung der Gradienten (Residuen) für mehrere Werte sehr viele numerische Operationen erfordern kann. (siehe hierzu Abschnitt 10).

#### 4.2.4 Verbesserung der Aufdatierung durch Skalierung

Neben der Empfindlichkeit des DFP Verfahrens auf die Genauigkeit der Strahlminimierung wurde auch der Einfluß einer Skalierung der Funktion oder der Variablen auf das Konvergenzverhalten beobachtet /Bard 1968/.

Den Einfluß der Skalierung untersuchten daraufhin /Oren, Luenberger 1974/ an Hand einer Familie von Rang-Zwei Änderungsformeln nach /Huang 1970/, die eine weitere Verallgemeinerung der Broyden- Klasse darstellen:

$$\mathbf{A}^{(k)} = \mathbf{A}^{(k-1)} - \frac{\mathbf{A}^{(k-1)} \cdot \mathbf{s}^{(k)} \mathbf{s}^{(k)t} \mathbf{A}^{(k-1)}}{\mathbf{s}^{(k)t} \mathbf{A}^{(k-1)} \mathbf{s}^{(k)}} + \gamma^{(k)} \frac{\mathbf{y}^{(k)} \mathbf{y}^{(k)t}}{\mathbf{y}^{(k)t} \mathbf{s}^{(k)}} + \Theta \cdot \mathbf{v}^{(k)} \cdot \mathbf{v}^{(k)t} \quad (4.49)$$

mit

$$\mathbf{v}^{(k)} = \mathbf{s}^{(k)\text{t}} \mathbf{A}^{(k-1)} \mathbf{s}^{(k)} \cdot \mathbf{y}^{(k)} - \mathbf{y}^{(k)\text{t}} \mathbf{s}^{(k)} \mathbf{A}^{(k-1)} \cdot \mathbf{s}^{(k)} \quad (4.50)$$

Das DFP-Verfahren entsteht hieraus mit  $\gamma^{(k)} = 1$ ;  $\Theta^{(k)} = 0$ .

Bei Forderung der Quasi-Newton Bedingung  $\mathbf{A}^{(k)} \cdot \mathbf{s}^{(k)} = \mathbf{y}^{(k)}$  wird dies für beliebige  $\theta^{(k)}$  und  $\gamma^{(k)}$  zu

$$\mathbf{A}^{(k)} = (\mathbf{A}^{(k-1)} - \frac{\mathbf{A}^{(k-1)} \cdot \mathbf{s}^{(k)} \mathbf{s}^{(k)\text{t}} \mathbf{A}^{(k-1)}}{\mathbf{s}^{(k)\text{t}} \mathbf{A}^{(k-1)} \mathbf{s}^{(k)}} + \Theta^{(k)} \cdot \mathbf{w}^{(k)} \mathbf{w}^{(k)\text{t}}) \gamma^{(k)} + \frac{\mathbf{y}^{(k)} \mathbf{y}^{(k)\text{t}}}{\mathbf{y}^{(k)\text{t}} \cdot \mathbf{s}^{(k)}} \quad (4.51)$$

mit

$$\mathbf{w}^{(k)} = (\mathbf{s}^{(k)\text{t}} \mathbf{A}^{(k-1)} \mathbf{s}^{(k)}) \left[ \frac{\mathbf{y}^{(k)}}{\mathbf{y}^{(k)\text{t}} \mathbf{s}^{(k)}} - \frac{\mathbf{A}^{(k-1)} \cdot \mathbf{s}^{(k)}}{\mathbf{s}^{(k)\text{t}} \mathbf{A}^{(k-1)} \mathbf{s}^{(k)}} \right] \quad (4.52)$$

Der einzige Unterschied zwischen dieser Untermenge der Huang- Aufdatierungen zur Broyden-Familie besteht im Skalierungsfaktor , der bei der Broyden-Familie  $\gamma = 1$  beträgt. Für positiv definite Matrizen sind  $\Theta^{(k)}$  und  $\gamma^{(k)}$  als positiv anzunehmen.

Zur sinnvollen Wahl der Größe von  $\Theta^{(k)}$  und  $\gamma^{(k)}$  muß die Eigenwertstruktur der Aufdatierungsformel untersucht werden.  $\Theta^{(k)}$  wird auf den Bereich zwischen 0 und 1 eingeschränkt. Als Maß der Annäherung der aufdatierten Matrix an die Hessematrix wird hier an Stelle des euklidischen Maßes  $\|\mathbf{M}^{(k)} - \mathbf{I}\|$  die Konditionszahl  $c(\mathbf{M}^{(k)})$  verwendet.

Wobei

$$\mathbf{M}^{(k)} = \mathbf{H}^{1/2} \mathbf{A}^{(k)} \mathbf{H}^{1/2} \quad (4.53)$$

mit

$\mathbf{H}$  .... Hessematrix

$\mathbf{A}^{(k)}$  ... Annäherung an die Inverse der Hessematrix

/Oren, Luenberger 1974/ zeigen, daß die unskalierte Aufdatierungsformel zu Matrizen führen kann, deren Konditionszahl schlechter ist als die der Hessematrix. Diese schlechte Konditionszahl - d.h. weit auseinander liegende Eigenwerte - verhindert häufig eine gute Konvergenz der Rang - Zwei Aufdatierungen. Wird der Skalierungsfaktor  $\gamma^{(k)}$  aber so gewählt, daß  $1/\gamma^{(k)}$  im Intervall der Eigenwerte von  $\mathbf{M}^{(k-1)}$  liegt, dann ist die Konditionszahl

$c(\mathbf{M}^k) \leq c(\mathbf{M}^{k-1})$ . Einige sinnvolle Vorschläge zur Ermittlung von  $\gamma^{(k)}$  werden in /Oren, Luenberger 1974/ angegeben.

Eine Möglichkeit besteht darin,  $\gamma^{(k)} = \alpha^{(i)}$  d.h. gleich dem Strahlminimierungswert zu setzen. Ist der Wert  $\alpha^{(i)}$  aber ungenau, z.B. bei einem Energieplateau (flachen Funktionen), so kann der Fehler in  $\gamma^{(k)}$  erheblich sein.

Eine bessere Möglichkeit ist

$$\gamma^{(k)} = (1 - \varphi) \frac{\mathbf{s}^{(k)t} \mathbf{y}^{(k)}}{\mathbf{y}^{(k)t} \mathbf{A}^{(k-1)} \mathbf{y}^{(k)}} + \varphi \frac{\mathbf{r}^{(k)t} \mathbf{s}^{(k)}}{\mathbf{r}^{(k)t} \mathbf{A}^{(k-1)} \mathbf{y}^{(k)}} \quad (4.54)$$

wobei

$$0 \leq \varphi \leq 1$$

Innerhalb eines Algorithmus kann die Skalierung in ausgewählten Schritten oder in jedem Schritt angewendet werden. Wird immer skaliert, so geht die Eigenschaft  $\mathbf{A}^{(k-1)} = \mathbf{H}^{-1}$  bei quadratischen Problemen verloren. Andererseits ist gesichert, daß die Konditionszahl  $c(\mathbf{M}^{(k)})$  der Vergleichsmatrix  $\mathbf{M}^{(k)}$  monoton abnimmt.

/Oren 1974/ weist nach, daß die Skalierung zu Algorithmen führt, die wesentlich weniger empfindlich gegen nicht quadratische Terme der Funktion sind als die DFP Aufdatierung. Außerdem sind die skalierten Aufdatierungen (self-scaling variable metric = SSVM algorithms) auch bei nicht exakter Strahlminimierung stabiler als die DFP Algorithmen. Sie konvergieren mindestens immer so gut wie die Methode des steilsten Abstiegs.

Eine optimale Konditionierung erfordert einen Abgleich der beiden Skalierungsparameter  $\gamma^{(k)}$  und  $\Theta^{(k)}$ . Diese ist dann gegeben /Oren, Spedicato 1976/, wenn

$$\Theta^{(k)} = \frac{\sigma(\epsilon - \gamma^{(k)}\sigma)}{\gamma^{(k)}(\epsilon\tau - \sigma^2)} \quad (4.55)$$

mit

$$\sigma = \mathbf{s}^t \mathbf{y}$$

$$\tau = \mathbf{y}^t \mathbf{A}^{(k-1)} \mathbf{y}$$

$$\epsilon = \mathbf{s}^t \mathbf{A}^{(k-1)^{-1}} \mathbf{s}$$

$$\mathbf{A}^{(k-1)^{-1}} \dots \text{Approximation an die Inverse der Hesse-Matrix}$$

und

$$\gamma^{(k)} \in [\sigma/\tau, \epsilon/\sigma] \quad (4.56)$$

/Oren, Spedicato 1976/ untersuchen verschiedene Vorschläge für  $\gamma^{(k)}$ . Aus den angegebenen numerischen Beispielen läßt sich aber kein bester Wert für  $\gamma^{(k)}$  ermitteln.

Im Rahmen der vorliegenden Arbeit wurden SSVM Algorithmen nicht implementiert. Eine Erweiterung der bestehenden Algorithmen ist aber mit geringem Aufwand möglich.

## 4.3 Vektorisierte Aufdatierungsformeln und Algorithmen

### 4.3.1 Allgemeines

Die Berechnung der Inversen und ihre Aufdatierung wird bei der Lösung der nichtlinearen Gleichungen nicht ausgeführt. An ihrer Stelle werden die Aufdatierungsvektoren direkt zur Lösung verwendet. Damit reduziert sich der Aufwand für die Lösung in jeder Iteration innerhalb eines Lösungsschritts auf einmal Vorwärts- /Rückwärtseinsetzen sowie mehrere Vektoroperationen für jede Aufdatierung. Innerhalb dieses Lösungskonzeptes müssen die Aufdatierungsvektoren und einige Faktoren für jede Iteration gespeichert werden. Außerdem wird die Ausgangsmatrix  $\mathbf{K}^{(0)}$  in ihrer zerlegten Form aufbewahrt. Das Vorgehen wird im Folgenden gezeigt. Die Lösung des nichtlinearen Gleichungssystems in der  $k$  - ten Iterationsschleife lautet:

$$\mathbf{d}^{(k+1)} = \mathbf{K}^{(k)-1} \cdot \mathbf{r}^{(k)} \quad (4.57)$$

Bei den meisten Quasi - Newton Algorithmen ist die Strahlminimierung (siehe Abschnitt 5 ) zur Sicherung der Konvergenz obligatorisch. Wird die Energie in Richtung von  $\mathbf{d}^{(k+1)}$  minimiert, gilt für den Lösungsvektor  $\Delta \mathbf{u}^{(k+1)}$  ( $\hat{=}$  Verschiebungsvektor) in der  $(k+1)$  - ten Iteration:

$$\Delta \mathbf{u}^{(k+1)} = s \cdot \mathbf{d}^{(k+1)} \quad (4.58)$$

mit  $s$  als Strahlminimierungsfaktor.

Die vektorisierten Formen für die Inversen, bei denen die Aufdatierungsvektoren und -faktoren für jede Iteration entweder im Kernspeicher oder auf Sekundärspeicher abgelegt werden, werden in den folgenden Abschnitten erläutert.

### 4.3.2 Rang-Eins Änderungen

#### BROYDEN

Inverse der aufdatierten Matrix:

$$\mathbf{K}^{(k)-1} = (\mathbf{I} + \alpha^{(k)} \cdot \mathbf{w}^{(k)} \cdot \Delta \mathbf{u}^{(k)t}) \left( \mathbf{I} + \alpha^{(k-1)} \mathbf{w}^{(k-1)} \Delta \mathbf{u}^{(k-1)t} \right) \dots \\ \dots \left( \mathbf{I} + \alpha^{(1)} \mathbf{w}^{(1)} \Delta \mathbf{u}^{(1)t} \right) \mathbf{K}^{(0)-1} \quad (4.59)$$

Aufdatierungsvektoren:  $\Delta \mathbf{u}^{(i)}, \mathbf{w}^{(i)}$

$$\mathbf{w}^{(i)} = \Delta \mathbf{u}^{(i)} - \mathbf{K}^{(i-1)-1} \mathbf{r}^{(i-1)} + \mathbf{K}^{(i-1)-1} \cdot \mathbf{r}^{(i)} \quad (4.60)$$

Aufdatierungsfaktor:

$$\alpha^{(i)} = \frac{1}{\Delta \mathbf{u}^{(i)t} (\Delta \mathbf{u}^{(i)} - \mathbf{w}^{(i)})} \quad (4.61)$$

#### Lösungsalgorithmus für nichtlineare Gleichungen

In der  $k$  ten Iteration

Schritt

1. Löse  $\mathbf{d} = \mathbf{K}^{(0)-1} \cdot \mathbf{r}^{(k)}$
2. Für  $i = 1, k - 1$   
hole  $\Delta \mathbf{u}^{(i)}, \mathbf{w}^{(i)}, \alpha^{(i)}$  aus dem Speicher  
 $\beta = \Delta \mathbf{u}^{(i)t} \cdot \mathbf{d}$   
 $\mathbf{d} = \mathbf{d} + \beta \cdot \alpha^{(i)} \cdot \mathbf{w}^{(i)}$
3.  $\mathbf{w}^{(k)} = \Delta \mathbf{u}^{(k)} - \Delta \mathbf{u}^{(k)}/s + \mathbf{d}$
4.  $\alpha^{(k)} = \frac{1}{\Delta \mathbf{u}^{(k)t} (\Delta \mathbf{u}^{(k)} - \mathbf{w}^{(k)})}$   
 $\beta = \Delta \mathbf{u}^{(k)t} \cdot \mathbf{d}$
5.  $\mathbf{d}^{(k+1)} = \mathbf{d} + \beta \cdot \alpha^{(k)} \cdot \mathbf{w}^{(k)}$
6. Speichern von  $\Delta \mathbf{u}^{(k)}, \mathbf{w}^{(k)}, \alpha^{(k)}$

Aus Gleichung (4.59) wird das Vorgehen offensichtlich. An das Vorwärts/-Rückwärtseinsetzen schließen sich pro Iteration bei  $n$  Freiheitsgraden

$(k - 1) * [(2n + 1) \text{Multiplikation}, n \text{Addition}]$   
 $+ (4n + 1) \text{Multiplikation}, 3n \text{Additionen}$

an.

Im Kernspeicher wird aktuell nur Platz für 4 Vektoren benötigt!

*Bemerkung 4.4:* Im Algorithmus muß gesichert werden, daß  $\alpha^{(k)}$  in Schritt 6 nicht sehr groß wird, d.h.  $\Delta \mathbf{u}^{(k)t} (\Delta \mathbf{u}^{(k)} - \mathbf{w}^{(k)})$  darf nicht zu null werden. In diesem Fall ist von einer Aufdatierung in der jeweiligen Iteration abzusehen.

## DAVIDON

Inverse der aufdatierten Matrix

$$\mathbf{K}^{(k)-1} = \mathbf{K}^{(0)-1} + \sum_{i=1}^k \alpha^{(i)} \cdot \mathbf{w}^{(i)} \cdot \mathbf{w}^{(i)t} \quad (4.62)$$

Aufdatierungsvektor:  $\mathbf{w}^{(i)}$

$$\mathbf{w}^{(i)} = \Delta \mathbf{u}^{(i)} - \mathbf{K}^{(i-1)-1} \cdot \mathbf{r}^{(i-1)} + \mathbf{K}^{(i-1)-1} \cdot \mathbf{r}^{(i)} = \Delta \mathbf{u}^{(i)} - \Delta \mathbf{u}^{(i)}/s + \mathbf{K}^{(i-1)-1} \cdot \mathbf{r}^{(i)} \quad (4.63)$$

Aufdatierungsfaktor

$$\alpha^{(i)} = \frac{1}{\mathbf{w}^{(i)t} (\mathbf{r}^{(i-1)} - \mathbf{r}^{(i)})} \quad (4.64)$$

## Lösungsalgorithmus für nichtlineare Gleichungen:

In der k-ten Iteration

Schritte

1. Löse  $\mathbf{d} = \mathbf{K}^{(0)-1} \cdot \mathbf{r}^{(k)}$
2. Für  $i = 1, k - 1$   
 hole  $\mathbf{w}^{(i)}, \alpha^{(i)}$  aus dem Speicher  
 $\beta = \mathbf{w}^{(i)t} \cdot \mathbf{r}^{(k)}$   
 $\mathbf{d} = \mathbf{d} + \beta \cdot \alpha^{(i)} \cdot \mathbf{w}^{(i)}$
3.  $\mathbf{w}^{(k)} = \Delta \mathbf{u}^{(k)} - \Delta \mathbf{u}^{(k)}/s + \mathbf{d}$
4.  $\alpha^{(k)} = \frac{1}{\mathbf{w}^{(k)t} (\mathbf{r}^{(k-1)} - \mathbf{r}^{(k)})}$
5.  $\beta = \mathbf{w}^{(k)t} \cdot \mathbf{r}^{(k)}$

$$\mathbf{d}^{(k+1)} = \mathbf{d} + \beta \cdot \alpha^{(k)} \cdot \mathbf{w}^{(k)}$$

6. Speichern von  $\mathbf{w}^{(k)}, \alpha^{(k)}$ .

Das einfache Vorgehen wird aus Gleichung (4.62) offensichtlich. An das Vorwärts-/Rückwärtseinsetzen schließen sich bei  $n$  Freiheitsgraden pro Iteration

$$(k-1) * [(2n+2) \text{Multiplikation}; 1n \text{Additionen}] \\ + (4n+2) \text{Multiplikation}, 3n \text{Additionen}$$

an.

Im Kernspeicher wird aktuell Platz für 5 Vektoren benötigt.

*Bemerkung 4.5:* Im Algorithmus muß gesichert werden, daß der Nenner in Schritt 4 nicht zu null wird. Ansonsten muß von der Aufdatierung in dieser Iterationsschleife abgesehen werden.

### 4.3.3 Rang-Zwei Änderungen

#### DFP - Davidon/Fletcher/Powell

Inverse der aufdatierten Matrix

$$\mathbf{K}^{(k)-1} = \mathbf{K}^{(0)-1} + \sum_{i=1}^k \alpha^{(i)} \Delta \mathbf{u}^{(i)} \Delta \mathbf{u}^{(i)t} - \sum_{i=1}^k \beta^{(i)} \mathbf{w}^{(i)} \mathbf{w}^{(i)t} \quad (4.65)$$

Aufdatierungsvektoren:  $\Delta \mathbf{u}^{(i)}, \mathbf{w}^{(i)}$

$$\mathbf{w}^{(i)} = \mathbf{K}^{(i-1)-1} \mathbf{r}^{(i-1)} - \mathbf{K}^{(i-1)-1} \mathbf{r}^{(i)} = \Delta \mathbf{u}^{(i)} / s - \mathbf{K}^{(i-1)-1} \cdot \mathbf{r}^{(i)} \quad (4.66)$$

Aufdatierungsfaktoren:

$$\alpha^{(i)} = \frac{1}{\Delta \mathbf{u}^{(i)t} (\mathbf{r}^{(i-1)} - \mathbf{r}^{(i)})} \quad \beta^{(i)} = \frac{1}{\mathbf{w}^{(i)t} (\mathbf{r}^{(i-1)} - \mathbf{r}^{(i)})} \quad (4.67)$$

### Lösungsalgorithmus für nichtlineare Gleichungen:

In der  $k$ -ten Iteration

Schritt

1. Löse  $\mathbf{d} = \mathbf{K}^{(0)-1} \cdot \mathbf{r}^{(k)}$

2. Für  $i = 1, k-1$

hole  $\Delta \mathbf{u}^{(i)}, \mathbf{w}^{(i)}, \alpha^{(i)}, \beta^{(i)}$  aus dem Speicher

$$\gamma = \Delta \mathbf{u}^{(i)t} \cdot \mathbf{r}^{(k)}$$

$$\delta = \mathbf{w}^{(i)t} \cdot \mathbf{r}^{(k)}$$

$$\mathbf{d} = \mathbf{d} + \gamma \cdot \alpha^{(i)} \Delta \mathbf{u}^{(i)} - \delta \cdot \beta^{(i)} \cdot \mathbf{w}^{(i)}$$

3.  $\mathbf{w}^{(k)} = \Delta \mathbf{u}^{(k)} / s - \mathbf{d}$

4.  $\alpha^{(k)} = \frac{1}{\Delta \mathbf{u}^{(k)t} (\mathbf{r}^{(k-1)} - \mathbf{r}^{(k)})}$ , ( besser mit  $\Delta \mathbf{r}^{(k)} = \mathbf{r}^{(k-1)} - \mathbf{r}^{(k)}$  )

$$\beta^{(k)} = \frac{1}{\mathbf{w}^{(k)t} (\mathbf{r}^{(k-1)} - \mathbf{r}^{(k)})}$$

5.  $\gamma = \Delta \mathbf{u}^{(k)t} \cdot \mathbf{r}^{(k)}$

$$\delta = \mathbf{w}^{(k)t} \cdot \mathbf{r}^{(k)}$$

$$\mathbf{d}^{(k+1)} = \mathbf{d} + \gamma \cdot \alpha^{(k)} \cdot \Delta \mathbf{u}^{(k)} - \delta \cdot \beta^{(k)} \cdot \mathbf{w}^{(k)}$$

6. Speichern von  $\Delta \mathbf{u}^{(k)}, \mathbf{w}^{(k)}, \alpha^{(k)}, \beta^{(k)}$

Das generelle Vorgehen wird aus Gleichung (4.65) offensichtlich. An das Vorwärts-/Rückwärtseinsetzen schließen sich bei  $n$  Freiheitsgraden pro Iteration

$(k-1) * [(4n+2) \text{Multiplikation}; 2n \text{Additionen}]$

$+(7n+2) \text{Multiplikation}, 4n \text{Additionen an.}$

Im Kernspeicher wird aktuell Platz für 5 Vektoren benötigt.

*Bemerkung 4.6:* Im Algorithmus muß gesichert werden, daß die Nenner im Schritt 4 nicht zu null werden. Ansonsten muß von der Aufdatierung in dieser Iterationsschleife abgesehen werden.

### **BFGS - Broyden-Fletcher-Goldfarb-Shanno**

Zwei vektorisierte Algorithmen für das BFGS-Verfahren werden in /Mathies, Strang 1979/ beschrieben. Aufbauend auf dem Aufsatz von /Brodliet al 1973/ wurde dort besonders auf die Vorteile der sogenannten Produktform eingegangen und das zugehörige Program in ALGOL beigegeben.

In /Brodliet al 1973/ wird gezeigt, daß sich im Fall positiv definiten Matrizen  $\mathbf{K}^{(0)}$  eine (scheinbar) besonders günstige Form der Aufdatierung für die gesamte Familie der Broyden -Verfahren erzielen läßt. Die aufdatierte Matrix ist als Produkt mehrerer Matrizen darstellbar:  $\rightarrow$  Produktform:

$$\mathbf{K}^{(k)-1} = (\mathbf{I} + \mathbf{xy}^t) \mathbf{K}^{(k-1)-1} (\mathbf{I} + \mathbf{xy}^t)^t = \mathbf{AK}^{(k-1)-1} \mathbf{A}^t \quad (4.68)$$

Für das BFGS Verfahren sind

$$\mathbf{x} = \frac{\Delta \mathbf{u}^{(k)}}{\Delta \mathbf{u}^{(k)t} \cdot \Delta \mathbf{r}^{(k)}} \quad (4.69)$$

$$\mathbf{y} = \pm \left( \frac{-\Delta \mathbf{u}^{(k)t} \Delta \mathbf{r}^{(k)}}{\Delta \mathbf{u}^{(k)t} \mathbf{K}^{(k-1)} \Delta \mathbf{u}^{(k)}} \right)^{1/2} \mathbf{K}^{(k-1)} \Delta \mathbf{u}^{(k)} + \Delta \mathbf{r}^{(k)} \quad (4.70)$$

Der wesentliche Vorteil der Formulierung in Produktform gegenüber den Formulierungen in Standardform liegt in der einfachen Ermittlung der Änderung der Konditionszahl zwischen der ursprünglichen Matrix  $\mathbf{K}^{(k)}$  und der aufdatierten Matrix  $\mathbf{K}^{(k+1)}$ .

Die Konditionszahl wird wie folgt definiert

$$C(\mathbf{K}^{(k)}) = \|\mathbf{K}^{(k)}\| \|\mathbf{K}^{(k)-1}\| \quad (4.71)$$

mit  $\|\cdot\|$  als Spektralnorm.

Die Spektralnorm von  $\mathbf{K}^{(k)}$  entspricht:

$$\|\mathbf{K}^{(k)}\| = \sqrt{\lambda_1} \quad (4.72)$$

mit  $\lambda_1$  als größtem Eigenwert der Matrix. Die Wurzel aus dem Verhältnis zwischen größtem und kleinstem Eigenwert der Matrix  $\mathbf{K}^{(k)t} \cdot \mathbf{K}^{(k)}$  bestimmt die Konditionszahl. Im Falle symmetrischer Matrizen  $\mathbf{K}^{(k)}$  entspricht  $\sqrt{\lambda_1}$  dem größten Eigenwert von  $\mathbf{K}^{(k)}$ . Es ist:

$$C(\mathbf{K}^{(k)}) = C(\mathbf{AK}^{(k-1)}\mathbf{A}^t) \leq C(\mathbf{K}^{(k-1)})[C(\mathbf{A})]^2$$

bzw.

$$C(\mathbf{K}^{(k-1)}) = C(\mathbf{A}^{-1}\mathbf{K}^{(k)}[\mathbf{A}^{-1}]^t) \leq C(\mathbf{K}^{(k)})[C(\mathbf{A}^{-1})]^2 \quad (4.73)$$

Für die spezielle Form von  $\mathbf{A} = \mathbf{I} - \mathbf{xy}^t$  gilt

$$C(\mathbf{A}^{-1}) = C(\mathbf{A}) \quad (4.74)$$

Damit kann das Verhältnis der Konditionszahl zweier aufeinander folgender Aufdatierungen zwischen zwei Schranken geschätzt werden

$$\frac{1}{[C(\mathbf{A})]^2} \leq \frac{C(\mathbf{K}^{(k)})}{C(\mathbf{K}^{(k-1)})} \leq [C(\mathbf{A})]^2 \quad (4.75)$$

Die Eigenwerte der Matrix  $\mathbf{H} = \mathbf{A}^t \mathbf{A}$ , mit

$$\mathbf{H} = \mathbf{A}^t \mathbf{A} = \mathbf{I} + \mathbf{x}\mathbf{y}^t + \mathbf{y}\mathbf{x}^t + (\mathbf{x}^t \mathbf{x})\mathbf{y}\mathbf{y}^t \quad (4.76)$$

liegen zwischen 0 und  $\lambda_1$  /Brodli et al 1973/. (n-2) Eigenwerte sind identisch 1. Die beiden restlichen Eigenwerte ergeben sich aus der folgenden quadratischen Gleichung:

$$\lambda^2 - \lambda[2(1 + \mathbf{x}^t \mathbf{y}) + (\mathbf{y}^t \mathbf{y}) \cdot (\mathbf{x}^t \mathbf{x})] + (1 + \mathbf{x}^t \mathbf{y})^2 = 0 \quad (4.77)$$

Es kann gezeigt werden, daß für die beiden Eigenwerte folgende Schranken gelten:

$$|1 < \lambda_1| \text{ und } 0 < \lambda_2 < 1$$

Damit ist:

$$C(\mathbf{A}) = \left(\frac{\lambda_1}{\lambda_2}\right)^{1/2} \quad (4.78)$$

mit

$$\frac{\lambda_1}{\lambda_2} = \frac{[(\mathbf{x}^t \mathbf{x})^{1/2}(\mathbf{y}^t \mathbf{y})^{1/2} + \{(\mathbf{x}^t \mathbf{x})(\mathbf{y}^t \mathbf{y}) + 4(1 + \mathbf{y}^t \mathbf{y})\}^{1/2}]^2}{[4(1 + \mathbf{x}^t \mathbf{y})]^2} \quad (4.79)$$

Mit den beiden Schranken für die Konditionszahl kann die Qualität der Aufdatierung beurteilt werden. Im Falle ungünstiger Verhältnisse - großer Konditionszahlen - ist dann von der Verwendung der Aufdatierung abzusehen.

*Bemerkung 4.7:* Ein Nachteil der Produktform ist, daß die Bildung einer Wurzel vorgenommen werden muß. Nur für positiv definite Matrizen ist gesichert, daß der Term  $\left(\frac{-\Delta \mathbf{u}^{(k)t} \Delta \mathbf{r}^{(k)}}{\Delta \mathbf{u}^{(k)t} \mathbf{K}^{(k-1)} \Delta \mathbf{u}_k}\right) \geq 0$  ist. Damit fällt die Produktform für die Verwendung bei nicht positiv definiten Matrizen aus.

Die Zahl der erforderlichen numerischen Operationen bei Anwendung der Produktform liegt außerdem nicht unterhalb anderer äquivalenter Formen.

## BFGS - Originalform

Die Inverse der aufdatierten Matrix lautet

$$\mathbf{K}^{(k)-1} = \left( \mathbf{I} - \frac{\Delta \mathbf{u}^{(k)} \cdot \Delta \mathbf{r}^{(k)t}}{\Delta \mathbf{u}^{(k)t} \cdot \Delta \mathbf{r}^{(k)}} \right) \mathbf{K}^{(k-1)-1} \left( \mathbf{I} - \frac{\Delta \mathbf{r}^{(k)} \cdot \Delta \mathbf{u}^{(k)t}}{\Delta \mathbf{u}^{(k)t} \Delta \mathbf{r}^{(k)}} \right) + \frac{\Delta \mathbf{u}^{(k)} \Delta \mathbf{u}^{(k)t}}{\Delta \mathbf{u}^{(k)t} \cdot \Delta \mathbf{r}^{(k)}} \quad (4.80)$$

Für diese ursprüngliche Form, wie sie in den Standardwerken /Luenberger 1984/ zu finden ist, geben /Matthies, Strang 1979/ den zugehörigen Algorithmus zur Gleichungslösung an.  $\mathbf{K}^{(k-1)}$  muß nicht positiv definit sein.

*Bemerkung 4.8:* Setzt man eine Zerlegung von  $\mathbf{K}^{(i)} = \mathbf{L}^{(i)} \mathbf{D}^{(i)} \mathbf{L}^{(i)t}$  voraus, so gilt, daß wie bei der Produktform auch für die Originalform bei einem sich nicht verändernden Lastvektor  $\mathbf{P}$  in jeder Iteration das Vorwärts/Rückwärtseinsetzen durchgeführt werden muß.

Die folgende modifizierte Form (siehe auch /Chrisfield 1979, 1984/) führt zu einer wesentlichen Reduktion des Aufwandes.

**Modifizierte BFGS-Aufdatierung = Additive Form:**

Inverse der aufdatierten Matrix

$$\begin{aligned} \mathbf{K}^{(k)-1} &= \mathbf{K}^{(k-1)-1} - \frac{\mathbf{K}^{(k-1)-1} \Delta \mathbf{r}^{(k)} \Delta \mathbf{u}^{(k)t}}{\Delta \mathbf{u}^{(k)t} \Delta \mathbf{r}^{(k)}} - \frac{\Delta \mathbf{u}^{(k)} \Delta \mathbf{r}^{(k)t} \mathbf{K}^{(k-1)-1}}{\Delta \mathbf{u}^{(k)t} \Delta \mathbf{r}^{(k)}} \\ &+ \left[ \frac{\Delta \mathbf{r}^{(k)t} \mathbf{K}^{(k-1)-1} \Delta \mathbf{r}^{(k)}}{\Delta \mathbf{u}^{(k)t} \Delta \mathbf{r}^{(k)}} + 1 \right] \frac{\Delta \mathbf{u}^{(k)} \Delta \mathbf{u}^{(k)t}}{\Delta \mathbf{u}^{(k)t} \Delta \mathbf{r}^{(k)}} \quad (a) \\ &= \mathbf{K}^{(0)-1} - \sum_{i=1}^k \alpha^{(i)} (\Delta \mathbf{u}^{(i)} \cdot \mathbf{w}^{(i)t} + \mathbf{w}^{(i)} \Delta \mathbf{u}^{(i)t}) + \sum_{i=1}^k \beta^{(i)} \Delta \mathbf{u}^{(i)} \Delta \mathbf{u}^{(i)t} \quad (b) \quad (4.81) \end{aligned}$$

Aufdatierungsvektoren:  $\Delta \mathbf{u}^{(i)}, \mathbf{w}^{(i)}$

$$\mathbf{w}^{(i)} = \mathbf{K}^{(i-1)-1} \cdot \mathbf{r}^{(i-1)} - \mathbf{K}^{(i-1)-1} \cdot \mathbf{r}^{(i)} = \Delta \mathbf{u}^{(i)} / s - \mathbf{K}^{(i-1)-1} \cdot \mathbf{r}^{(i)} \quad (4.82)$$

Aufdatierungsfaktoren:

$$\begin{aligned} \alpha^{(i)} &= \frac{1}{\Delta \mathbf{u}^{(i)t} (\mathbf{r}^{(i-1)} - \mathbf{r}^{(i)})} \\ \beta^{(i)} &= \alpha^{(i)} [\alpha^{(i)} \cdot \mathbf{w}^{(i)t} (\mathbf{r}^{(i-1)} - \mathbf{r}^{(i)}) + 1] \quad (4.83) \end{aligned}$$

## Lösungsalgorithmus für nichtlineare Gleichungen

In der  $k$ -ten Iteration

Schritt

1. Löse  $\mathbf{d} = \mathbf{K}^{(0)^{-1}} \cdot \mathbf{r}^{(k)}$
2. Für  $i = 1, k - 1$   
hole  $\Delta \mathbf{u}^{(i)}, \mathbf{w}^{(i)}, \alpha^{(i)}, \beta^{(i)}$  aus dem Speicher  
 $\gamma = \Delta \mathbf{u}^{(i)t} \cdot \mathbf{r}^{(k)}$   
 $\delta = \mathbf{w}^{(i)t} \cdot \mathbf{r}^{(k)}$   
 $\mathbf{d} = \mathbf{d} - \gamma \cdot \alpha^{(i)} \cdot \mathbf{w}^{(i)} + (-\delta \cdot \alpha^{(i)} + \gamma \cdot \beta^{(i)}) \Delta \mathbf{u}^{(i)}$
3.  $\mathbf{w}^{(k)} = \Delta \mathbf{u}^{(k)} / s - \mathbf{d}$
4.  $\alpha^{(k)} = \frac{1}{\Delta \mathbf{u}^{(k)t} (\mathbf{r}^{(k-1)} - \mathbf{r}^{(k)})}$   
 $\beta^{(k)} = \alpha^{(k)} [\alpha^{(k)} \cdot \mathbf{w}^{(k)t} (\mathbf{r}^{(k-1)} - \mathbf{r}^{(k)}) + 1]$
5.  $\gamma = \Delta \mathbf{u}^{(k)t} \cdot \mathbf{r}^{(k)}$   
 $\delta = \mathbf{w}^{(k)t} \cdot \mathbf{r}^{(k)}$   
 $\mathbf{d}^{(k+1)} = \mathbf{d} - \gamma \cdot \alpha^{(k)} \cdot \mathbf{w}^{(k)} + (-\delta \alpha^{(k)} + \gamma \cdot \beta^{(k)}) \Delta \mathbf{u}^{(k)}$
6. Speichern von  $\Delta \mathbf{u}^{(k)}, \mathbf{w}^{(k)}, \alpha^{(k)}, \beta^{(k)}$

Das generelle Vorgehen wird aus Gleichung (4.81) offensichtlich. An das Vorwärts-/Rückwärtseinsetzen schließen sich bei  $n$  Freiheitsgraden pro Iteration

$$(k - 1) * [(4n + 3) \text{Multiplikation}; 4n \text{Additionen}] \\ + (7n + 5) \text{Multiplikation}, 3n \text{Additionen}$$

an.

Im Kernspeicher wird aktuell Platz für 5 Vektoren benötigt.

*Bemerkung 4.9:* Da sich die Konditionszahl der modifizierten BFGS Form nicht so einfach bestimmen läßt wie für die Produktform, empfiehlt es sich, die Änderung der Determinante zu beobachten.

$$\det(\mathbf{K}^{(k)}) = \det(\mathbf{K}^{(k-1)}) \cdot \frac{\Delta \mathbf{u}^{(k)t} \Delta \mathbf{r}^{(k)}}{\Delta \mathbf{u}^{(k)t} \mathbf{K}^{(k-1)} \Delta \mathbf{u}^{(k)}} \quad (4.84)$$

Mit Strahlminimierung gilt:

$$\mathbf{K}^{(k-1)} \cdot \Delta \mathbf{u}^{(k)} = s \cdot \mathbf{r}^{(k-1)}$$

$$\frac{\det(\mathbf{K}^{(k)})}{\det(\mathbf{K}^{(k-1)})} = \Delta^{(k)} \det$$

$$\Delta^{(k)} \det = \frac{\Delta \mathbf{u}^{(k)t} (\mathbf{r}^{(k-1)} - \mathbf{r}^{(k)})}{s \Delta \mathbf{u}^{(k)t} \cdot \mathbf{r}^{(k-1)}} \quad (4.85)$$

Dieses kann in obigem Algorithmus in Schritt 4 eingefügt werden.

Schritt 4a:

$$\Delta \det = \frac{1}{\alpha^{(k)} \cdot s \cdot \Delta \mathbf{u}^{(k)t} \cdot \mathbf{r}^{(k-1)}}$$

*Bemerkung 4.10:* Ist in jedem Schritt die Berechnung

$$\mathbf{x}^{(k)} = \mathbf{K}^{(k)-1} \cdot \mathbf{P} \quad (4.86)$$

durchzuführen, so ergibt sich der Lösungsvektor  $\mathbf{x}^{(k)}$  mittels einfacher Vektoroperationen.

$$\mathbf{K}^{(k)-1} = \mathbf{K}^{(k-1)-1} + \alpha^{(k)} (\Delta \mathbf{u}^{(k)} \mathbf{w}^{(k)t} + \mathbf{w}^{(k)} \Delta \mathbf{u}^{(k)t}) + \beta^{(k)} \Delta \mathbf{u}^{(k)} \Delta \mathbf{u}^{(k)t} \quad (4.87)$$

$$\begin{aligned} \mathbf{x}^{(k)} &= \mathbf{K}^{(k)-1} \cdot \mathbf{P} \\ &= \underbrace{\mathbf{K}^{(k-1)-1} \cdot \mathbf{P}}_{-\mathbf{x}^{(k-1)}} + \alpha^{(k)} \cdot (\Delta \mathbf{u}^{(k)t} \cdot \mathbf{P}) \cdot \mathbf{w}^{(k)} \\ &\quad + [\alpha^{(k)} \cdot \mathbf{w}^{(k)t} \cdot \mathbf{P} + \beta^{(k)} \Delta \mathbf{u}^{(k)t} \cdot \mathbf{P}] \Delta \mathbf{u}^{(k)} \end{aligned} \quad (4.88)$$

D.h. es sind nur  $(4n+2)$  Multiplikationen;  $2n$  Additionen erforderlich.

*Bemerkung 4.11:* In /Matthies, Strang 1979/ wird empfohlen, im Fall schlechter Konditionszahlen für die Aufdatierung im betreffenden Iterationsschritt  $k$  mit der alten Matrix  $\mathbf{K}^{(k)} = \mathbf{K}^{(k-1)}$  weiter zu arbeiten. Die numerische Lösung des Beispiels Kragarm (Abschnitt 10) zeigte aber, daß die Konvergenz mit dieser Einschränkung wegen der geringeren Anzahl der Vektoren

deutlich langsamer wurde, d.h. die Anforderungen an die Konditionszahl können sicher noch herabgesetzt werden.

*Bemerkung 4.12:* Die dargestellten Algorithmen zeigen, daß auf die ursprüngliche Bandstruktur der Matrix bei den Aufdatierungen keine Rücksicht genommen wird. Es erscheint daher sinnvoll zu sein, auch die meist dünne Besetzung der Matrizen in den Quasi – Newton Algorithmus einzubeziehen. In /Kamat et al 1981 / werden solche Vorschläge an einigen Beispielen ausgetestet. Allerdings werden mit diesen Strategien die Matrizen selbst aufgebaut und dann mittels z.B. Gauß Elimination gelöst. Einige Vorschläge zielen auch auf die direkte Veränderung der faktorisierten Matrix  $\mathbf{K}^{(i)} = \mathbf{L}^{(i)}\mathbf{D}^{(i)}\mathbf{L}^{(i)t}$  d.h.  $\mathbf{L}^{(i)}$  bzw.  $\mathbf{D}^{(i)}$  werden aufdatiert.

Die Effizienz dieser Verfahren ist für Strukturprobleme, verglichen mit den vektorisierten Algorithmen, gering. Die Berücksichtigung der dünnen Belegung in Kombination mit vektorisierten Algorithmen wird im Schrifttum nicht diskutiert. Sie scheint aber angesichts der Rechengeschwindigkeiten, mit denen Skalarprodukte durchgeführt werden, ohne Bedeutung zu sein.

#### 4.4 Quasi – Newton Verfahren mit reduzierter Aufdatierung

Quasi-Newton Verfahren, die im Gegensatz zu den in den vorigen Abschnitten vorgestellten Algorithmen nur die Information der (k- 1)ten Iteration zur Bestimmung der Aufdatierung verwenden, werden im englischen Schrifttum /Nazareth 1981b/Nocedal 1980/ treffend als "memoryless methods" gekennzeichnet. In dieser Arbeit wird hierfür der Begriff reduzierte Aufdatierung benutzt.

/Chrisfield 1979,1984/ verwendet Modifikationen des "memoryless" BFGS Verfahrens zur Entwicklung "beschleunigter" modifizierter Newton Algorithmen (Accelerated modified Newton schemes). Diese Verfahren sind rechen-technisch sehr effizient und benötigen nur wenig mehr Speicherplatz als das normale modifizierte Newton Verfahren, da nur ein Aufdatierungsvektor benutzt wird.

Am Beispiel des BFGS Algorithmus soll das Vorgehen mit reduzierter Aufdatierung gezeigt werden.

##### BFGS-Verfahren mit reduzierter Aufdatierung

Inverse der aufdatierten Matrix

$$\mathbf{K}^{(k)-1} = \mathbf{K}^{(0)-1} - \alpha^{(k)}(\Delta \mathbf{u}^{(k)} \mathbf{w}^{(k)t} + \mathbf{w}^{(k)} \Delta \mathbf{u}^{(k)t}) + \beta^{(k)} \cdot \Delta \mathbf{u}^{(k)} \cdot \Delta \mathbf{u}^{(k)t} \quad (4.89)$$

Aufdatierungsvektoren:  $\Delta \mathbf{u}^{(k)}, \mathbf{w}^{(k)}$

$$\mathbf{w}^{(k)} = \mathbf{K}^{(0)^{-1}} \cdot \mathbf{r}^{(k-1)} - \mathbf{K}^{(0)^{-1}} \cdot \mathbf{r}^{(k)} = \Delta \mathbf{u}^{(k)} / s - \mathbf{K}_0^{-1} \cdot \mathbf{r}^{(k)} \quad (4.90)$$

Aufdatierungsfaktoren:

$$\alpha^{(k)} = \frac{1}{\Delta \mathbf{u}^{(k)t} (\mathbf{r}^{(k-1)} - \mathbf{r}^{(k)})} \quad (4.91)$$

$$\beta^{(k)} = \alpha^{(k)} [\alpha^{(k)} \cdot \mathbf{w}^{(k)t} (\mathbf{r}^{(k-1)} - \mathbf{r}^{(k)}) + 1] \quad (4.92)$$

Die einzige Änderung, die hierbei erfolgt, ist das Ersetzen von  $\mathbf{K}^{(i)^{-1}}$  durch  $\mathbf{K}^{(0)^{-1}}$ . Dann entfallen in allen beschriebenen Algorithmen die Schritte 2 sowie die Speicheroperationen in Schritt 6.

Der Mehraufwand der Quasi-Newton Verfahren mit reduzierter Aufdatierung verglichen mit dem modifizierten Newton Verfahren beläuft sich auf zwei Skalarprodukte sowie 4 - 5 Vektoradditionen für Rang - Eins Aufdatierungen sowie zwei weitere Skalarprodukte für Rang - Zwei Aufdatierungen.

*Bemerkung 4.13:* Chrisfields beschleunigte Newton Verfahren /Crisfield 1979, 1982, 1984/ wurden direkt aus Gleichung (4.81a) entwickelt. Es sei aber festgestellt, daß die erzeugten Ein- bzw. Zwei-Parameter beschleunigten Verfahren weder Symmetrie noch positive Definitheit der modifizierten Matrix bewahren. Ein wesentlicher Vorteil gegenüber den unveränderten Verfahren mit reduzierter Aufdatierung ist daher nicht feststellbar.

# 5 Strahlminimierung - Line Search

## 5.1 Allgemeines

Strahlminimierung ("Line Search") beschreibt die Suche nach einem Minimum längs einer Linie bzw eines Strahles. Hierbei wird eine eindimensionale Minimumsuche durchgeführt, d.h. es wird eine Variable variiert. Siehe /Lunenberger 1984/ für eine Einführung.

Interessant im Zusammenhang mit den in dieser Arbeit behandelten Verfahren ist der Einsatz der Strahlminimierung bei mehrdimensionalen Problemen.

Hierbei wird davon ausgegangen, daß ein erster Schätzwert für eine Lösung bereits vorliegt, z.B. aus einem Newtonschritt. Dargestellt an einem quadratischen Problem gilt:

$$\Pi = \frac{1}{2} \mathbf{x}^t \mathbf{Q} \mathbf{x} + \mathbf{x}^t \mathbf{P} \quad (5.1)$$

Gesucht: Minimum

gegeben: Startwert  $\mathbf{x}^{(k)}$  und Schätzwert  $\mathbf{d}^{(k)}$

Neuer Wert:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha \cdot \mathbf{d}^{(k)} \quad (5.2)$$

Minimierung:

$$\frac{\partial \Pi}{\partial \alpha} = 0 \rightarrow \frac{\partial \Pi}{\partial \mathbf{x}} \cdot \frac{\partial \mathbf{x}}{\partial \alpha} \Big|^{(k)} = 0 \quad (5.3)$$

$$\frac{\partial \mathbf{x}}{\partial \alpha} = \mathbf{d}^{(k)} \quad (5.4)$$

$$\frac{\partial \Pi}{\partial \mathbf{x}} = \mathbf{Q}(\mathbf{x}^{(k)} + \alpha \mathbf{d}^{(k)}) + \mathbf{P} \quad (5.5)$$

$$(\mathbf{Q}\mathbf{x}^{(k)} + \alpha \mathbf{Q} \cdot \mathbf{d}^{(k)} + \mathbf{P})^t \cdot \mathbf{d}^{(k)} = 0 \quad (5.6)$$

$$\alpha = \frac{\mathbf{x}^{(k)t} \mathbf{Q} \mathbf{d}^{(k)} + \mathbf{P}^t \mathbf{d}^{(k)}}{\mathbf{d}^{(k)t} \mathbf{Q} \mathbf{d}^{(k)}} \hat{=} - \frac{\mathbf{r}^t \mathbf{d}^{(k)}}{\mathbf{d}^{(k)t} \mathbf{Q} \mathbf{d}^{(k)}} \quad (5.7)$$

mit  $\mathbf{r} = \mathbf{Q}\mathbf{x}^{(k)} + \mathbf{P}$

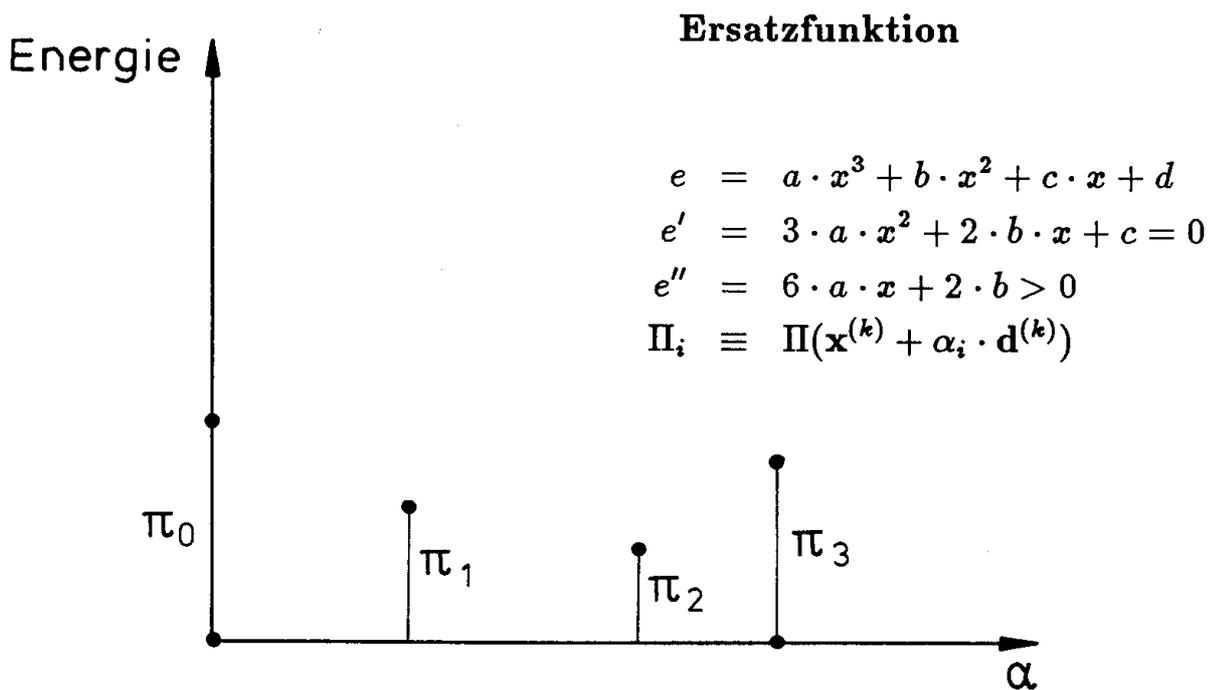
$\alpha$  beschreibt das Minimum von  $\Pi$  in Richtung  $\mathbf{d}^{(k)}$ .

Für quadratische Probleme ergibt sich eine lineare Gleichung in  $\alpha$ , die wie oben dargestellt exakt gelöst werden kann. Im allgemeinen Fall ist  $\Pi$  eine nichtlineare (nicht quadratische) Funktion von  $\mathbf{x}$  und die Minimierung hat iterativ zu erfolgen.

In der Strukturmechanik wird üblicherweise das Minimum der Gesamtenergie gesucht. Die Suche kann entweder über eine direkte Auswertung der Energie erfolgen, oder mit Hilfe der Residuen vorgenommen werden.

Werden die Energie und somit der Funktionswert der zu minimierenden Funktion direkt bestimmt, so kann ein klassisches Verfahren wie die Fibonacci-Regel oder ähnliche Methoden /Luenberger 1984/ zur Suche verwendet werden. Häufig wird die nichtlineare Energiefunktion an diskreten Stellen d.h. für verschiedene  $\alpha^{(i)}$  ausgewertet und unter Verwendung dieser Stützstellen durch ein Polynom quadratischer oder höherer Ordnung /Irons, Elswaf 1979/ angenähert. Dann wird das Minimum dieses Ersatzproblems bestimmt.

(Bild 5.1)

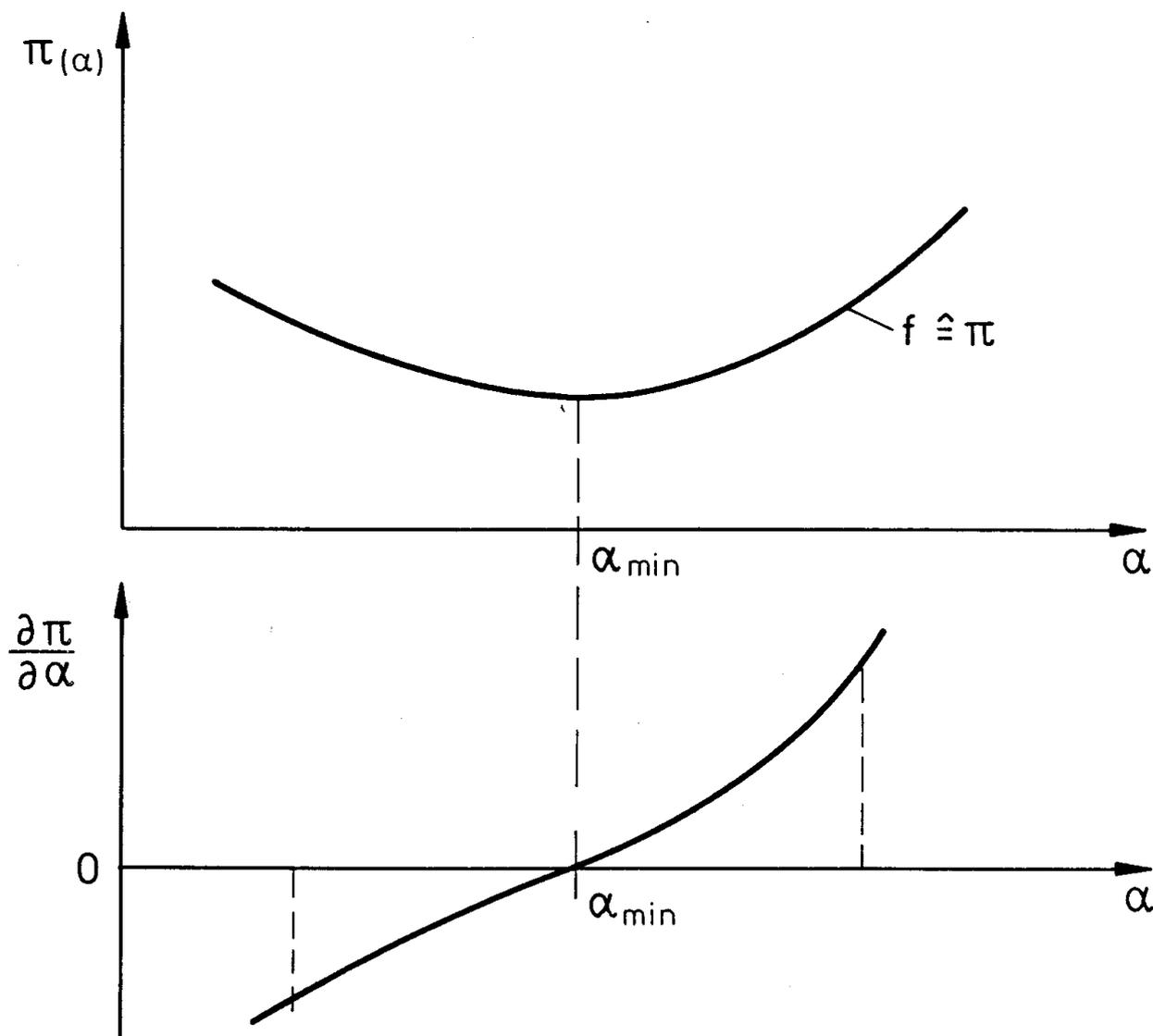


**Bild 5.1 Näherung der Energiefunktion durch Ersatzpolynome**

Da Energien in den Strukturberechnungen üblicherweise nicht ermittelt werden, bedeutet jede Funktionsauswertung bei Systemen mit vielen Freiheitsgraden einen erheblichen numerischen Aufwand.

Wird hingegen der Gradient  $\frac{\partial \Pi}{\partial \alpha}$  zur Minimumsuche herangezogen, bringt dies

einen wesentlichen numerischen Vorteil, da die hierbei erforderliche Berechnung des Residuums im Rahmen einer Berechnung ohnehin häufig erfolgt.



**Bild 5.2 Ermittlung des Energieminimums durch Nullstellensuche für den Gradienten**

Die erste Ableitung der Energie liegt bei den Aufgaben der Strukturmechanik immer als schwache Form der Differentialgleichung bzw. als Gateaux - Ableitung infolge eines Variationsprinzips vor. Hier gilt es diese Ableitung - das Residuum - für mehrere  $\alpha$ -Werte auszuwerten und die Nullstelle von  $\frac{\partial \Pi}{\partial \alpha} = \frac{\partial \Pi}{\partial \mathbf{x}} \cdot \frac{\partial \mathbf{x}}{\partial \alpha}$  zu bestimmen (Bild 5.2).

Damit ist zuerst das Residuum zu ermitteln.

$$\frac{\partial \Pi^{(i)}}{\partial \mathbf{x}} = \mathbf{r}^{(i)} \Big|_{\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(i)} \mathbf{d}^{(k)}} \quad (5.8)$$

Im zweite Schritt ist der Gradient  $\frac{\partial \Pi}{\partial \alpha}$  zu berechnen.

$$\frac{\partial \Pi^{(i)}}{\partial \alpha} = E^{(i)} = \mathbf{r}^{(i)t} \cdot \mathbf{d}^{(k)} \quad (5.9)$$

An den Nullstellen von  $E(\alpha)$  befindet sich das Minimum für  $\Pi(\alpha)$ .

Die Nullstellensuche muß effektiv gestaltet werden, da für jeden Suchschritt eine Auswertung des Residuums erforderlich ist.

Ein Vorschlag von /Irons, Elsayaf 1977/ beschränkt die Auswertung des Residuums auf die Werte für  $\alpha^{(i)} = [0, 0.25, 0.85, 2.8, 4.0]$ .  $E(\alpha)$  wird mittels einer Lagrangeinterpolation angenähert und für diese Näherung wird die Nullstelle bestimmt.

$$E(\alpha) = \sum_{i=1}^n E(\alpha^{(i)}) \cdot L^{(i)} \quad (5.10)$$

mit z.B. 5 Stützstellen

$$L^{(2)} = \frac{(\alpha - \alpha_1)(\alpha - \alpha_3)(\alpha - \alpha_4)(\alpha - \alpha_5)}{(\alpha_2 - \alpha_1)(\alpha_2 - \alpha_3)(\alpha_2 - \alpha_4)(\alpha_2 - \alpha_5)} \quad (5.11)$$

und

$$\frac{\partial E}{\partial \alpha} = 0 = \sum_{i=1}^n E(\alpha^{(i)}) \frac{\partial L^{(i)}}{\partial \alpha} \quad (5.12)$$

Zum Beispiel:

$$\frac{\partial L^{(2)}}{\partial \alpha} = L^{(2)} \cdot \left[ \frac{1}{\alpha - \alpha_1} + \frac{1}{\alpha - \alpha_3} + \frac{1}{\alpha - \alpha_4} + \frac{1}{\alpha - \alpha_5} \right] \quad (5.13)$$

Dieses Vorgehen reduziert die Minimumsuche auf 4 Auswertungen des Residuums und die Nullstellensuche in einem Ersatzproblem. Der Erfolg dieses Konzepts ist aber stark von der Güte der Näherung an den wirklichen Verlauf von  $E(\alpha)$  im betrachteten Bereich abhängig. Dies gilt ebenfalls für die Annäherung durch andere Polynome.

Die effizienteste Methode, wie sie auch in /Matthies, Strang 1979/ /Crisfield 1983/ und /Hallquist 1986/ enthalten ist, beruht auf einer einfachen linearen

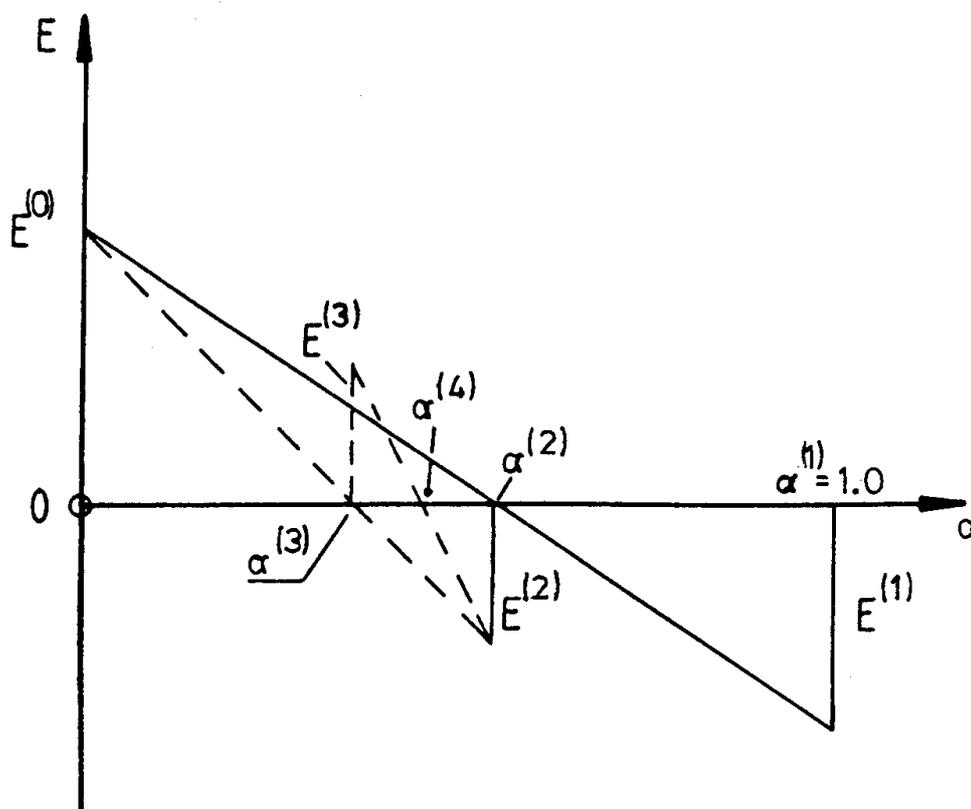
Inter- bzw. Extrapolation zwischen zwei Werten für  $E(\alpha)$ . /Crisfield 1989/ läßt auch Extrapolationen zu, die nach den Erfahrungen des Autors dieses Berichts häufig zu ineffektiven Lösungen führen. Extrapolationen destabilisieren den Lösungsalgorithmus und sind im Fall der Nichtkonvergenz nur schwer einzuschränken.

Die Verwendung der Strahlminimierung zur Eingrenzung des Minimums innerhalb eines gewählten Intervalls hingegen führt zu einem stabilen Algorithmus. Voraussetzung hierfür ist eine sinnvolle - nicht zu kleine - Ausgangsschrittweite.

## 5.2 Einfache Strahlminimierungsalgorithmen

Der Vorteil dieser Verfahren besteht darin, daß das Residuum nur für eine Mindestzahl von Punkten ausgewertet wird.

Der Algorithmus selbst basiert auf einem Regula Falsi Schema. (Bild 5.3). Hierbei wird zwischen zwei Gradientenwerten  $E^{(i-1)}$  und  $E^{(i)}$  interpoliert.



**Bild 5.3 Regula Falsi Schema zur Strahlminimierung**

An der Nullstelle  $\alpha^{(i+1)}$  wird ein neues Residuum  $r^{(i+1)}(\mathbf{x}^{(k)} + \alpha^{(i+1)} \cdot \mathbf{d}^{(k)})$  bestimmt und ein neuer Gradientenwert  $E^{(i+1)}$  berechnet. Die nächste Interpolation erfolgt zwischen  $E^{(i+1)}$  und dem der beiden Werte  $E^{(i-1)}$  bzw.  $E^{(i)}$ ,

der ein zu  $E^{(i+1)}$  entgegengesetztes Vorzeichen besitzt. Diese Interpolationsfolge wird solange fortgesetzt, bis der Absolutwert des Gradienten kleiner ist als  $\eta \cdot E^{(0)}$ .

Schranke:

$$|E^{(j)}| \leq \eta |E^{(0)}| \quad (5.14)$$

$\eta$  ... Line Search Toleranz

Um ein Suchen ohne wesentliche Änderung der Schrittweite zu verhindern, wird häufig eine weitere Schranke hinzugezogen.

$$|\alpha^{(j)} - \alpha^{(j-1)}| < 0.5\eta(\alpha^{(j)} + \alpha^{(j-1)}) \quad (5.15)$$

Bei Unterschreitung dieser Schranke wird die Suche ebenfalls abgebrochen.

Line Search Algorithmus:

1.  $E^{(0)} = \mathbf{r}^{(0)t}(\alpha^{(0)} = 0) \cdot \mathbf{d}^{(k)}$   
 $E^{(1)} = \mathbf{r}^{(1)t}(\alpha^{(1)} = 1) \cdot \mathbf{d}^{(k)}$   
 $a = E^{(0)} \quad , \quad \alpha^{(a)} = 0$   
 $b = E^{(1)} \quad , \quad \alpha^{(b)} = 1$
2. Abprüfung der Schranke:  
 $a \cdot b > 0 ? \quad \text{Ja} \rightarrow \text{Ende (keine Extrapolation!)}$   
 $|b| \leq \eta \cdot |E^{(0)}| ? \quad \text{Ja} \rightarrow \text{Ende konvergiert}$   
 $|\alpha^{(b)} - \alpha^{(a)}| < 0.5\eta(\alpha^{(b)} + \alpha^{(a)}) \quad \text{Ja} \rightarrow \text{Ende konvergiert}$
3. Interpolation (Regula Falsi)  
 $\alpha^{(i)} = [(\alpha^{(b)} - \alpha^{(a)}) / (a - b)] \cdot a + \alpha^{(a)}$
4. Neues Residuum:  
 $\mathbf{r}(\alpha^{(i)})$
5. Neuer Gradient  $E^{(i)} = [\mathbf{r}^t(\alpha^{(i)})]^t \cdot \mathbf{d}^{(k)}$

6.  $E^{(i)} \cdot b > 0?$  → weiter mit Schritt 7  
 Nein:  $a = b$   
 $\alpha^{(a)} = \alpha^{(b)}$
7.  $b = E^{(i)}$   
 $\alpha^{(b)} = \alpha^{(i)}$
8. Zugelassene Maximalanzahl der Iterationen erreicht?  
 Ja — Stop  
 Nein — Weiter mit Schritt 2.

### 5.3 Exakte - Inexakte Strahlminimierung

Wird die Strahlminimierung nur zur Konvergenzsicherung und -beschleunigung eingesetzt, so genügt meist eine grobe Toleranz von  $\eta = 0.8 \div 0.9$ . Dies bezeichnet man als inexakte Strahlminimierung, da das Minimum nicht exakt gesucht wird.

Bei den Quasi-Newton Verfahren genügt die inexakte Strahlminimierung nur beim BFGS-Verfahren zur Konvergenz. Alle anderen in den vorigen Abschnitten besprochenen Quasi-Newton Verfahren sollten theoretisch mit exakter Strahlminimierung angewendet werden. Bei der Durchführung der Berechnung auf einem Computer würde dies bedeuten, daß eine Toleranz im Rahmen der Maschinengenauigkeit gewählt werden müßte. Numerische Beispiele haben jedoch gezeigt, daß dies für praktische Anwendungen nicht unbedingt erforderlich ist. Meist genügt eine Toleranz von  $\eta = 0.3 - 0.5$ .

Im Rahmen eines effizienten Verfahrens sollte außerdem die Zahl der Strahlminimierungen nicht mehr als 3 - 4 pro Schritt betragen, sonst sind a priori kleinere Schrittweiten sinnvoller.

# 6 Kurvenverfolgungsalgorithmen

## 6.1 Allgemeines

### 6.1.1 Einfache Verfahren

Kurvenverfolgungsalgorithmen werden in der Strukturmechanik zur Berechnung des nichtlinearen Verhaltens von Strukturen unter bestimmten Belastungen benötigt. Ihre wesentliche Eigenschaft besteht darin, sich dem Kurvenverlauf anzupassen, um möglichst gute Konvergenz bei der Lösung der nichtlinearen Gleichungen zu ermöglichen.

Das klassische Vorgehen innerhalb eines Newton/Raphson Lösungsschemas ( Gl. 6.1 ) besteht darin, daß z.B mit wachsenden Verschiebungsinkrementen  $\Delta \mathbf{u}$  aus

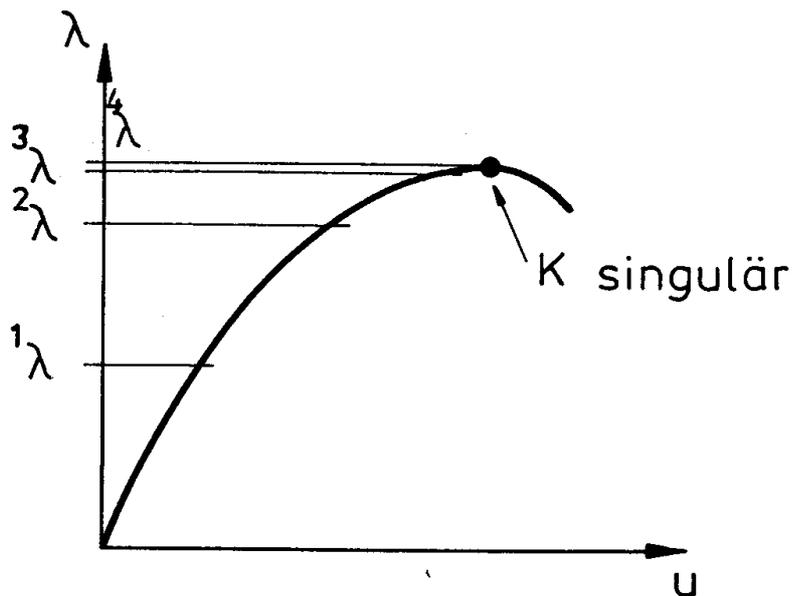
$$\mathbf{K} \cdot \Delta \mathbf{u} = \lambda \cdot \mathbf{P} - \mathbf{R} \quad (6.1)$$

die Vorgabe  $\Delta \lambda$  für die folgenden Schritte des Lastfaktors  $\lambda$  angepaßt, d.h. verkleinert oder vergrößert wird.

Die mathematische Seite dieses Vorgehens wird ausführlich in /Schwetlick 1971/ Kap. 11 diskutiert. Bekanntermaßen führt dieses Vorgehen der sogenannten Lastkontrolle nur dann zum Erfolg, wenn die Tangentenmatrix  $\mathbf{K}$  nicht singulär wird.

Eine einfache Abhilfe wird hierzu von /Bergan et al 1978/ vorgeschlagen. Zur Steuerung des Algorithmus in der Nähe der Singularität von  $\mathbf{K}$  werden der aktuelle Steifigkeitsparameter  $S_p = \frac{\Delta \mathbf{u}^{(0)t} \cdot \mathbf{P}}{\Delta \mathbf{u}^{(i)t} \cdot \mathbf{P}}$  und die Vorzeichen der Diagonalterme von  $\mathbf{D}$  bei einer  $\mathbf{LDL}^t$  Zerlegung verwendet.

Der aktuelle Steifigkeitsparameter wird in der Nähe eines Durchschlagpunktes, wie in Bild 6.1 dargestellt, sehr klein. Unterschreitet der  $S_p$  Parameter eine vorgegebene Schranke, so wird das Newton Verfahren mit Iterationen durch ein einfaches Euler Verfahren mit Rückwärtseinsetzen ersetzt. Der Lastfaktor und die zugehörigen Verschiebungen werden zurückskaliert, sofern die euklidische Norm des Verschiebungsinkrements eine vorgegebene Größe überschreitet. Zeigt sich bei einer Zerlegung von  $\mathbf{K}$  in  $\mathbf{LDL}^t$  Form ein negativer Diagonalterm in  $\mathbf{D}$ , so wird das Vorzeichen des Lastfaktors geändert. Überschreitet der aktuelle Steifigkeitsparameter  $S_p$  in einem späteren Schritt wieder die vorgegebene Schranke, wird wieder auf das Newton Verfahren mit Iteration umgeschaltet (siehe Bild 6.2).



**Bild 6.1 Kurvenverfolgungsalgorithmen**  
**Klassisches Vorgehen - Lastkontrolle**

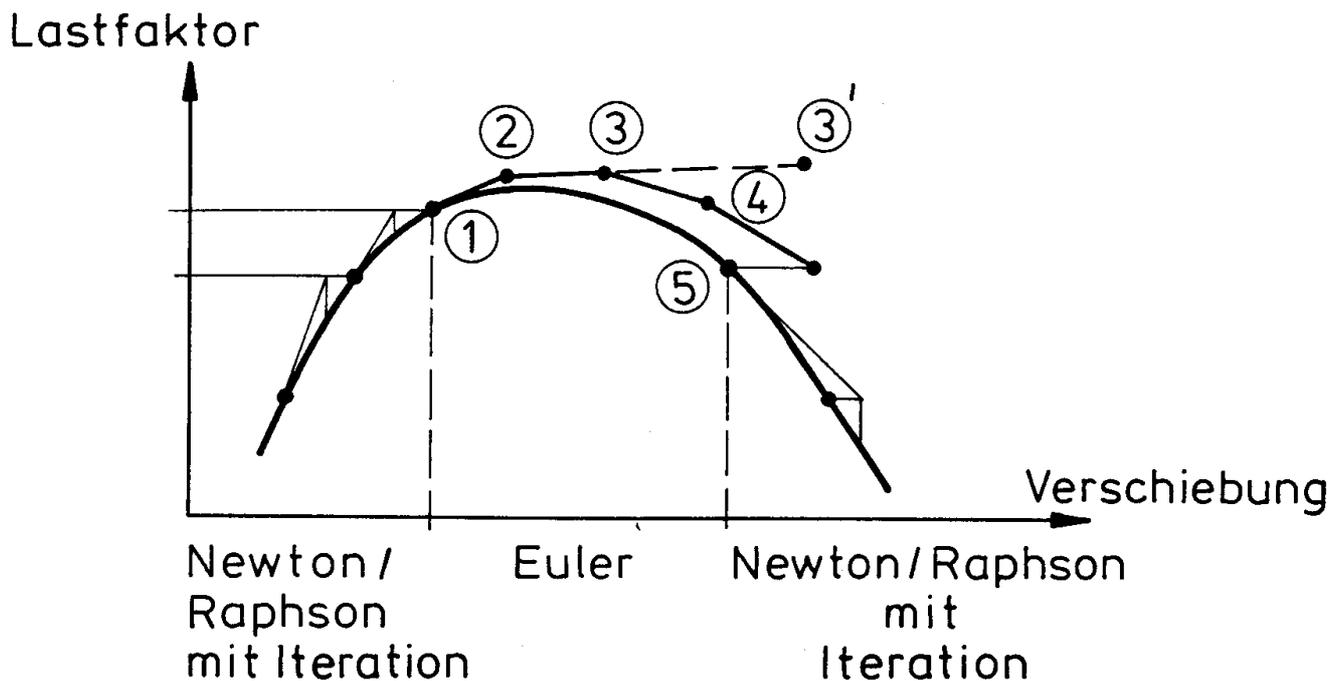
Dieses Vorgehen erfordert eine sehr feine Abstimmung der Lastinkremente, um Divergenz zu vermeiden.

Diesem einfachen Verfahren entsprechen die Vorschläge von Branin und Smale (siehe hierzu /Keller 1978/) bei Einbettungsverfahren, die ebenfalls die Vorzeichen der Determinante zur Steuerung des Vorzeichens des Einbettungsparameters nutzen.

Während die Steuerung des Vorzeichens der Last sich über die Determinante am Verhalten der Struktur orientiert, ist die Wahl der Schrittweite und ihre Änderung beim klassischen Verfahren weitgehend von der Erfahrung des Anwenders abhängig. Es liegen zwar Vorschläge vor /W. Schmidt 1978/, diese haben aber nur wenig Eingang in Ingenieur Anwendungen gefunden.

### 6.1.2 Bogenlängenverfahren

Einen wesentlichen Fortschritt sowohl in Bezug auf die Überwindung singulärer Punkte als auch in Bezug auf die Schrittweitensteuerung stellen die sogenannten Bogenlängenverfahren dar. Hierbei wird zu dem nichtlinearen Gleichungssystem  $\mathbf{K} \cdot \Delta \mathbf{u} = \lambda \cdot \mathbf{P} - \mathbf{F}$  eine weitere Gleichung hinzugefügt, die mindestens eine der Variablen aus dem Vektor  $\Delta \mathbf{u}$  bzw. den Lastfaktor  $\lambda$  enthält. Die Grundgedanken des Verfahrens stammen von Davidenko 1953 siehe /Allgöwer 1981/ und wurden schon von /Anselone, Moore 1966/ für Probleme der Strukturmechanik eingesetzt. Bis Anfang der 70 er Jahre waren



**Bild 6.2 Einfacher Kurvenverfolgungsalgorithmus für nichtsinguläre Matrizen nach /Bergan et al 1978/**

die Arbeiten auf dem Gebiet der Bogenlängenverfahren sowohl bei den Ingenieuren als auch den Mathematikern auf Einzellösungen beschränkt. Erst mit den Veröffentlichungen von /Riks 1972,1979/ und /Keener, Keller 1973/ wurde eine Systematisierung erreicht. Hier ist aus der Ingenieurliteratur ebenfalls /Wempner 1971/ zu erwähnen. In der Mathematik erlebten die Verfahren seit 1975 – 76 eine stürmische Entwicklung siehe hierzu /Schwetlick 1979/ /Menzel, Schwetlick 1978/ /Pönisch, Schwetlick 1982/ und die globale Übersicht von /Allgöwer/.

Bis 1980/81 sind hingegen nur wenige Anwendungen in der Ingenieurpraxis bekannt /Wessels 1977/. Der Durchbruch in der Anwendung wurde erst durch die Arbeiten von /Crisfield 1981/ und /Ramm 1981/ erreicht. Seither ist eine Vielzahl von Veröffentlichungen auf dem Gebiet der Bogenlängenverfahren in der Strukturmechanik und Ingenieurpraxis erschienen. Auf vereinzelte wesentliche Arbeiten wird in den folgenden Abschnitten eingegangen.

Zu bemerken ist, daß sowohl /Crisfield 1981,1982/ als auch /Ramm 1981/ das wesentliche algorithmische Vorgehen nach den Arbeiten von /Haisler, Stricklin, Key 1977/ sowie /Batoz, Dhett 1979/ über verschiebungsgesteuerte Verfahren konzipieren und von /Riks 1972/ nur die Grundidee der Iteration

im Last-/Verschiebungsraum übernehmen.

In den folgenden Abschnitten wird auf die verschiedenen Kurvenverfolgungsalgorithmen, die auf der Erweiterung des ursprünglichen Gleichungssystems um eine Gleichung beruhen eingegangen.

## 6.2 Erweiterte Gleichungssysteme – Newton Verfahren – Direkte Lösung

### 6.2.1 Aufbau eines erweiterten Gleichungssystems

Die wesentliche Idee in der Entwicklung leistungsfähiger Algorithmen zur Berechnung hoch nichtlinearer, parameterabhängiger Gleichungssysteme ist die Erweiterung des Ausgangssystems um eine oder mehrere Gleichungen. In der vorliegenden Arbeit wird die Diskussion auf die Erweiterung um eine Gleichung bei Problemen der Strukturmechanik beschränkt. Für die Möglichkeiten, die mit Erweiterungen um mehrere Gleichungen bestehen, sei neben Abschnitt 9 auf die Arbeit von /Keener, Keller 1973/ sowie die Übersicht in /Fink, Rheinboldt 1984/ und die ingenieurmäßige Anwendung in /Wriggers, Wagner, Miehe 1988/ sowie /Kröplin, Dinkler 1985/ verwiesen. (Siehe auch Abschnitt 9.)

In der Strukturmechanik läßt sich die schwache Form eines geometrisch und/oder materiell nichtlinearen Problem übersichtlich in Matrixform wie folgt darstellen:

$$G(\mathbf{x}, \lambda; \eta) = \eta^t(\mathbf{R}(\mathbf{x}) - \lambda \cdot \mathbf{P}) = 0 \quad (6.2)$$

mit  $\eta$  ... virtuelle Verschiebungen  
 $\mathbf{x}$  ... Positionsvektor der aktuellen Lage  
 $\mathbf{R}(\mathbf{x})$  ... Vektor der inneren Kräfte  
 $\mathbf{P}$  ... Vektor der äußeren Kräfte  
 $\lambda$ ... proportionaler Lastfaktor

Die Anwendung der Variationsrechnung auf Gl.(6.2) führt auf die nichtlinearen Gleichgewichtsbeziehungen

$$\mathbf{F}(\mathbf{x}, \lambda, \mathbf{P}) = \mathbf{R}(\mathbf{x}) - \lambda \cdot \mathbf{P} = 0 \quad (6.3)$$

Die Darstellung sei hier auf richtungstreue Belastungen d.h. einen konstanten Lastvektor beschränkt. Für die Behandlung verformungsabhängiger

Belastungen in Kombination mit Bogenlängenverfahren sei auf /Schweizerhof, Ramm 1986/ verwiesen.

Die klassische Lösungsmethode für Gl. (6.3) ist das Newton/Raphson Verfahren, das zu dem bereits in den vorigen Abschnitten beschriebenen inkrementell/iterativen Vorgehen führt:

$$\mathbf{K}_T^{(i)} \cdot \Delta \mathbf{u}^{(i)} = -[\mathbf{R}(\mathbf{x}^{(i)}) - \lambda \cdot \mathbf{P}] \quad (6.4)$$

mit

$$\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} + \Delta \mathbf{u}^{(i)} \quad (6.5)$$

$\mathbf{K}_T$  ist die Tangentensteifigkeitsmatrix  $\frac{\partial \mathbf{F}(\mathbf{x})}{\partial \mathbf{x}}$ , die durch konsistente Linearisierung gewonnen wird.

Die übliche Nebenbedingung bei diesen inkrementellen Verfahren besteht darin, den Lastparameter  $\lambda$  (- das Lastniveau -) in jedem Inkrement konstant zu halten.

Klassische Bedingung:

$$\lambda = 0 \quad (6.6)$$

Diese Bedingung wird allgemeiner gefaßt und als eine beliebige Funktion der augenblicklichen Lage, des Lastfaktors und des Lastvektors angenommen (Gl. 6.7).

$$f(\mathbf{x}, \lambda, \mathbf{P}) = 0 \quad (6.7)$$

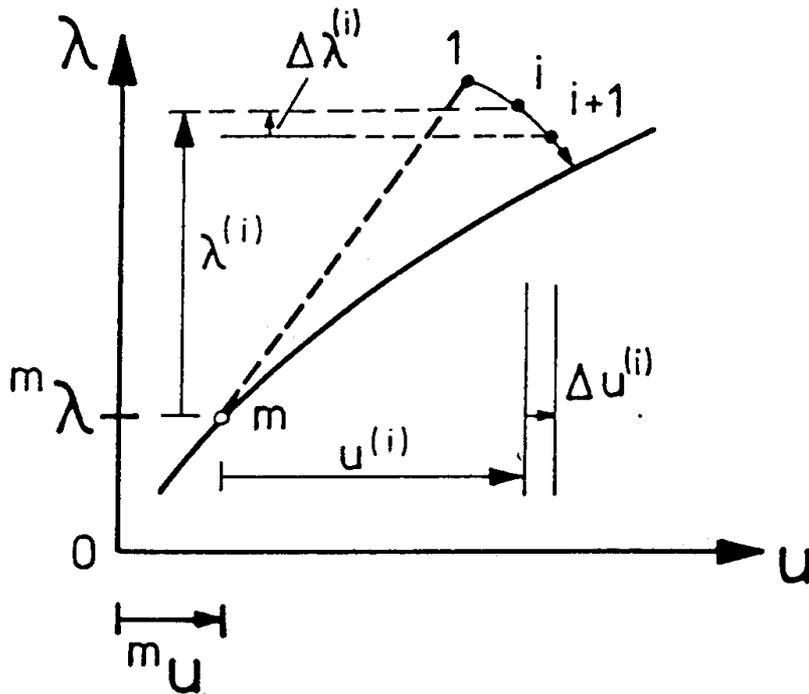
Damit wird  $\lambda$  auch als Variable innerhalb des Newton Schemas betrachtet und Gl. (6.4) lautet in konsistent linearisierter Form:

$$\mathbf{K}_T^{(i)} \cdot \Delta \mathbf{u}^{(i)} - \mathbf{P} \cdot \Delta \lambda^{(i)} = ({}^m \lambda + \lambda^{(i)}) \cdot \mathbf{P} - \mathbf{R}(\mathbf{x}^{(i)}) \quad (6.8)$$

mit  ${}^m \lambda, {}^m \mathbf{u}$  Lastfaktor, Verschiebungen im Ausgangszustand  $m$   
 $\lambda^{(i)}, \mathbf{u}^{(i)}$  inkrementeller Lastfaktor, Verschiebungsvektor  
 $\Delta \lambda^{(i)}, \Delta \mathbf{u}^{(i)}$  iterativer Lastfaktor, Verschiebungsvektor

Der Fußzeiger T steht für die Tangentensteifigkeitsmatrix.

Während der Iteration werden dann sowohl Last- als auch Verschiebungswerte verändert (Bild 6.3).



**Bild 6.3 Iteration im Last-/Verschiebungsraum**

Innerhalb des Newton-Schemas muß auch die zusätzliche Gleichung linearisiert werden:

$$\mathbf{v}^{(i)t} = \frac{d}{d\epsilon} f(\mathbf{x}^{(i)} + \epsilon \Delta \mathbf{u}^{(i)}, \lambda^{(i)}, \mathbf{P}) \big|_{\epsilon=0} \quad (6.9)$$

$$\alpha^{(i)} = \frac{d}{d\epsilon} f(\mathbf{x}^{(i)}, \lambda^{(i)} + \epsilon \Delta \lambda^{(i)}, \mathbf{P}) \big|_{\epsilon=0} \quad (6.10)$$

Damit ergibt sich das vollständige Gleichungssystem zu:

$$\underbrace{\begin{bmatrix} \mathbf{K}_T^{(k)} & -\mathbf{P} \\ \mathbf{v}^{(i)t} & \alpha^{(i)} \end{bmatrix}}_{\text{Systemmatrix A}} \cdot \begin{bmatrix} \Delta \mathbf{u}^{(i)} \\ \Delta \lambda^{(i)} \end{bmatrix} = - \begin{bmatrix} \mathbf{R}(\mathbf{x}^{(i)}) - (\lambda^m + \lambda^{(i)})\mathbf{P} \\ f(\mathbf{x}^{(i)}, \lambda^{(i)}, \mathbf{P}) \end{bmatrix} = - \begin{bmatrix} \mathbf{F}^{(i)} \\ f^{(i)} \end{bmatrix}$$

Systemmatrix A

$$(6.11)$$

Der Zusatzkopfzeiger ( $k$ ) wird für die Steifigkeitsmatrix eingeführt. Ist  $k = i$ , so handelt es sich um ein echtes Newton-Verfahren; ist  $k = 0$ , so wird ein modifiziertes Newton Verfahren eingesetzt.

Die neue Systemmatrix (Jacobimatrix)  $\mathbf{A}$  ist für allgemeine Zusatzgleichungen unsymmetrisch. Es lassen sich jedoch leicht Bedingungen konstruieren, für die sich symmetrische Systemmatrizen ergeben /Bathe, Dvorkin 1983/.

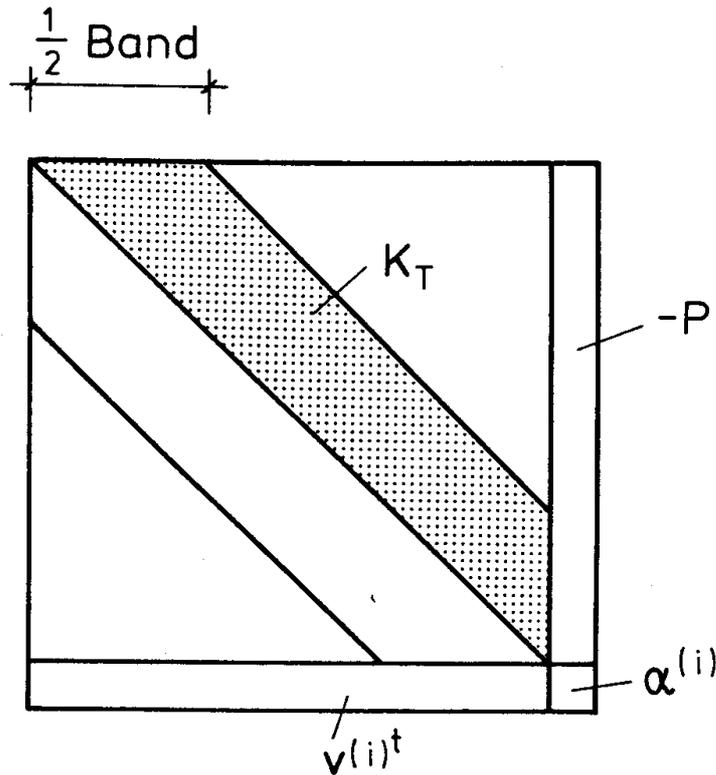
Die Jacobimatrix  $\mathbf{A}$  ist durch die Nebenbedingung für die Singularitätspunkte der Hessematrix  $\mathbf{K}_T^{(i)}$  a priori nicht singular, vorausgesetzt  $\mathbf{v}^{(i)} \neq \mathbf{0}$  /Riks 1979/Keller1982/.  $\mathbf{v}^{(i)}$  und  $\alpha^{(i)}$  sind im Fall nichtlinearer Nebenbedingungen im Inkrement nicht konstant.

### 6.2.2 Lösung des erweiterten Systems mit Gleichungslösern für unsymmetrischen Matrizen

Beim direkten Weg der Lösung von (Gl. 6.11) werden allgemeine Gleichungslöser für unsymmetrische Matrizen benötigt. Damit werden zwar alle Schwierigkeiten mit Singularitäten umgangen, aber wegen der durch das Hinzufügen einer Spalte und einer Zeile erheblich größerer Bandbreite von  $\mathbf{A}$  gegenüber der Bandbreite von  $\mathbf{K}_T$  erscheint dieses Vorgehen für große Gleichungssysteme aus Effizienzgründen nicht sinnvoll. Einige speziell entwickelte Gleichungslöser /Endo, Oden, Miller 1984 / Waszcyszyn 1983/ nutzen die besondere Unsymmetrie von  $\mathbf{A}$ , die sich auf eine Zeile und eine Spalte beschränkt. Wird die Bedingungsgleichung als letzte Gleichung angefügt, kann die Symmetrie der Matrix  $\mathbf{K}_T$  benutzt werden und die Operationen des Gleichungslösers beschränken sich auf die halbe Bandbreite von  $\mathbf{K}_T$  sowie die zusätzliche Zeile und Spalte (Bild 6.4).

Wird ein echtes Newton/Raphson Verfahren verwendet, so sind diese speziellen Gleichungslöser unabhängig von der verwendeten Zusatzgleichung numerisch ebenso effizient wie die im folgenden besprochenen Verfahren mit reduzierten Gleichungssystemen. Dies gilt auch bei Verwendung eines modifizierten Newton/Raphson Verfahrens und einer linearen Zusatzgleichung. Ist die Zusatzgleichung jedoch nichtlinear, so bleibt beim modifizierten Newton Verfahren mit der Gesamtmatrix auch der Anteil der Zusatzgleichung in  $\mathbf{A}$  konstant, wohingegen er bei den reduzierten Gleichungssystemen in jeder Iteration angepaßt wird. Alternativ kann zwar auf eine Neuaufstellung von  $\mathbf{K}_T$  verzichtet werden, aber bei nichtlinearen Zusatzgleichungen ist dann  $\mathbf{A}$  trotzdem auch beim modifizierten Newton-Verfahren neu zu faktorisieren.

Ein wesentlicher Nachteil der Lösung mit einer unsymmetrischen Systemmatrix  $\mathbf{A}$  ist, daß wesentliche Informationen über die Struktursteifigkeitsmatrix  $\mathbf{K}_T$  wie die Determinante oder Lage der Eigenwerte, die sich üblicherweise bei der Faktorisierung automatisch ergeben, nur mit zusätzlichem Aufwand zu erhalten sind.



**Bild 6.4 Verwendung spezieller Gleichungslöser für erweiterte Gleichungssysteme nach /Waszcyszyn 1983/**

### 6.2.3 Reduktion auf ein Gleichungssystem der Ausgangsgröße

Die Reduktion auf ein kleineres Gleichungssystem kann mittels Vorwegelimination der Lastgröße  $\Delta\lambda^{(i)}$  erfolgen. Dies ist nur möglich, sofern  $\alpha^{(i)} \neq 0$  ist. Liegen Zusatzgleichungen vor, die unabhängig vom Lastparameter sind, so muß zur Reduktion eine Aufteilung vorgenommen werden, wie sie zuerst von /Batoz, Dhatt 1979/ und danach von /Crisfield 1981/ und /Ramm 1981/ vorgeschlagen wurde.

Die in /Schweizerhof, Wriggers 1986/ analog zu /Keller 1978/ vorgestellte Reduktion erklärt den Einfluß einer Zusatzgleichung auf den Iterationsablauf.

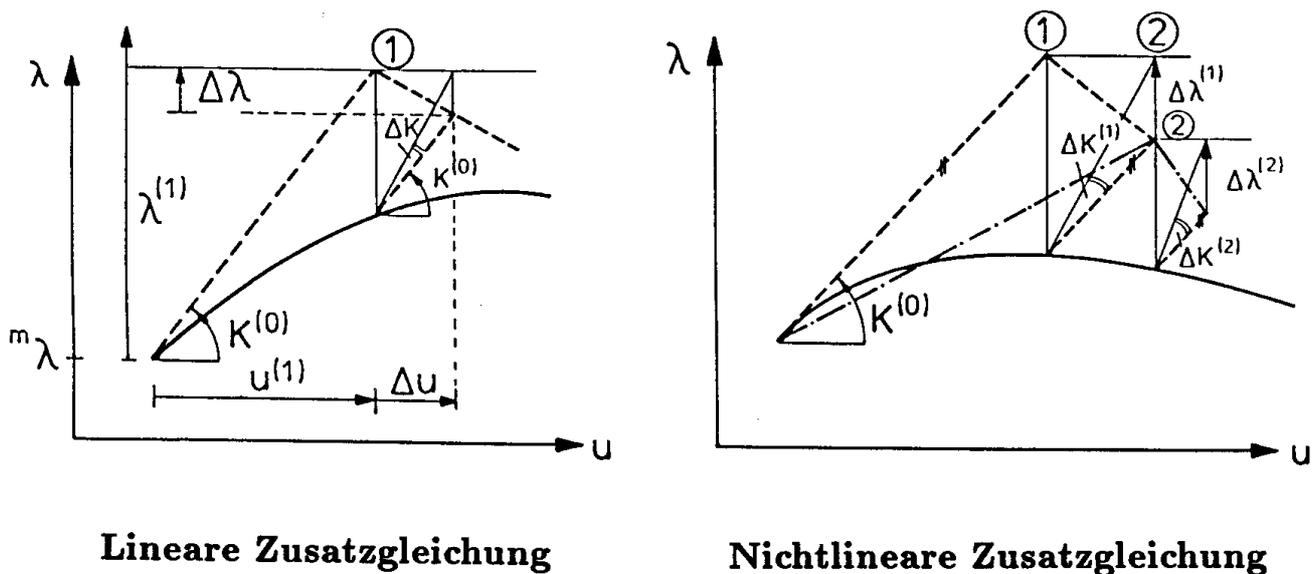
Die Umformung der linearisierten Zusatzgleichung ergibt:

$$\Delta\lambda^{(i)} = -\frac{1}{\alpha^{(i)}}(f^{(i)} + \mathbf{v}^{(i)t} \Delta\mathbf{u}^{(i)}) \quad (6.12)$$

Eingesetzt in Gleichung (6.11) gilt:

$$[\mathbf{K}_T^{(k)} + \underbrace{1/\alpha^{(i)} \cdot \mathbf{P} \cdot \mathbf{v}^{(i)t}}_{\Delta \mathbf{K}}] \Delta \mathbf{u}^{(i)} = -\mathbf{F}^{(i)} - \frac{f^{(i)}}{\alpha^{(i)}} \cdot \mathbf{P} \quad (6.13)$$

In Gl. (6.13) wird auf die ursprüngliche Matrix  $\mathbf{K}_T^{(k)}$  eine Matrix  $\Delta \mathbf{K}^{(i)}$  mit dem Rang Eins aufaddiert; außerdem wird die rechte Seite mit einem Residuum aus der Zusatzgleichung verändert. Diese Rang-Eins Änderung  $\Delta \mathbf{K}^{(i)}$  erzeugt für allgemeine Zusatzgleichungen eine unsymmetrische Koeffizientenmatrix. Ihre Auswirkung auf den Iterationsablauf ist für lineare und nichtlineare Zusatzgleichungen in Bild 6.5 für ein Lastinkrement dargestellt. Deutlich wird, daß sich  $\Delta \mathbf{K}^{(i)}$  für nichtlineare Zusatzgleichungen in jeder Iteration ändert.



**Bild 6.5** Iterationsablauf bei linearen und nichtlinearen Zusatzgleichungen

Die Inversion des reduzierten Gleichungssystems (6.13) wird mit Anwendung der Sherman/Morrison Formel /Sherman, Morrison 1947/ durchgeführt:

$$[\mathbf{K}_T^{(k)} + \frac{1}{\alpha^{(i)}} \cdot \mathbf{P} \cdot \mathbf{v}^{(i)t}]^{-1} = \mathbf{K}^{(k)-1} - \beta \cdot \mathbf{x} \cdot \mathbf{z}^t \quad (6.14)$$

$$\text{mit } \mathbf{z} = \mathbf{K}_T^{(k)(-1)} \cdot \mathbf{v}^{(i)} \quad (6.15)$$

$$\mathbf{x} = \mathbf{K}_T^{(k)-1} \cdot \mathbf{P}$$

$$\beta = (\alpha^{(i)} + \mathbf{P}^t \mathbf{K}_T^{(k)-1} \cdot \mathbf{v}^{(i)})^{-1} \quad (6.16)$$

Damit gilt:

$$[\mathbf{K}_T^{(k)} + \frac{1}{\alpha^{(i)}} \cdot \mathbf{P} \cdot \mathbf{v}^{(i)t}]^{-1} = \mathbf{K}_T^{(k)-1} - \frac{\mathbf{K}_T^{(k)-1} \cdot \mathbf{P} \cdot \mathbf{v}^{(i)t} \cdot \mathbf{K}_T^{(k)-1}}{\alpha^{(i)} + \mathbf{P}^t \mathbf{K}_T^{(k)-1} \cdot \mathbf{v}^{(i)}} \quad (6.17)$$

Die Lösung der Gleichung (6.13) erfolgt mit Hilfe von Gl. (6.17). Deutlich wird, daß die Matrix  $\mathbf{K}_T^{(k)}$  zu faktorisieren ist, sowie das Vorwärts/Rückwärtseinsetzen für zwei rechte Seiten vorzunehmen ist.

$$\Delta \mathbf{u}^I = \mathbf{K}_T^{(k)-1} \cdot \mathbf{P} \quad \Delta \mathbf{u}^{II} = -\mathbf{K}_T^{(k)-1} \cdot \mathbf{F}^{(i)} \quad (6.18)$$

Damit läßt sich die Lösung von Gl. (6.13) schreiben als

$$\Delta \mathbf{u}^{(i)} = -\frac{f^{(i)} + \mathbf{v}^{(i)t} \cdot \Delta \mathbf{u}^{II}}{\alpha^{(i)} + \mathbf{v}^{(i)t} \cdot \Delta \mathbf{u}^I} \cdot \Delta \mathbf{u}^I + \Delta \mathbf{u}^{II} \quad (6.19)$$

Wird Gl. (6.19) in Gl. (6.12) eingesetzt, ergibt sich die iterative Änderung des Lastfaktors  $\Delta \lambda^{(i)}$  zu:

$$\Delta \lambda^{(i)} = -\frac{f^{(i)} + \mathbf{v}^{(i)t} \cdot \Delta \mathbf{u}^{II}}{\alpha^{(i)} + \mathbf{v}^{(i)t} \cdot \Delta \mathbf{u}^I} \quad (6.20)$$

und

$$\Delta \mathbf{u}^{(i)} = \Delta \lambda^{(i)} \cdot \Delta \mathbf{u}^I + \Delta \mathbf{u}^{II} \quad (6.21)$$

Der iterative Verschiebungsvektor setzt sich somit - wie auch in /Batoz, Dhatt 1989/ und /Ramm 1981/ - aus dem Anteil  $\Delta \mathbf{u}^{II}$  infolge der Ungleichgewichtskräfte sowie einem Vielfachen des Lösungsvektors  $\Delta \mathbf{u}^I$  infolge des Lastvektors zusammen.

Der wesentliche Nachteil des Vorgehens mit der ursprünglichen Matrix  $\mathbf{K}_T^{(k)}$  besteht darin, daß im Gegensatz zum Vorgehen mit der unsymmetrischen

Matrix  $\mathbf{A}$  die Matrix  $\mathbf{K}_T$  singulär werden kann. Die theoretisch zu erwartenden Probleme bei der Gleichungslösung haben sich bei numerischen Beispielen aber als nicht wesentlich herausgestellt. Die Singularitätsstelle von  $\mathbf{K}_T$  wird bei numerischen Berechnungen mit Rechenanlagen mit hoher Wahrscheinlichkeit nicht genau getroffen. Nach /Keller 1978/ kann dieses Problem auch auf einfache Weise umgangen werden.

Für lineare Zusatzgleichungen entsprechen die Gleichungen (6.20) und (6.21) den in /Batoz,Dhatt 1979/ und /Ramm1981/ angegebenen Formeln.

Wesentlich ist außerdem, daß für Gleichung (6.20) die Einschränkung für  $\alpha^{(i)} = 0$  nicht mehr gilt, d.h. also auch Zusatzgleichungen ohne Anteile für den Lastfaktor verwendet werden können.

Die Vorteile des Vorgehens mit der ursprünglichen Matrix sind klar:

1.  $\mathbf{K}_T^{(k)}$  ist symmetrisch und besitzt die ursprüngliche Bandbreite. Damit haben die verschiedenen Bogenlängenverfahren keinen Einfluß auf das Vorgehen bei der Gleichungslösung.
2. Wird mit dem modifizierten Newton/Raphson Verfahren gearbeitet, so wird trotzdem die Zusatzgleichung in jedem Schritt in der Form des echten Newton Verfahrens erfüllt. D.h. es wird nicht nur mit dem Residuum und einer konstanten Matrix im Lastinkrement iteriert. Damit können auch beim modifizierten Newton Verfahren die Effekte nichtlinearer Zusatzgleichungen gut berücksichtigt werden.
3. Die Informationen über die Steifigkeitsmatrix  $\mathbf{K}_T$  liegen wie üblich vor: Determinante, Anzahl der Vorzeichenänderungen der Diagonalterme usw.
4. Die Änderungen der Determinante der modifizierten Steifigkeitsmatrix nach Gl. (6.13) läßt sich leicht mittels folgender Formel kontrollieren /Pearson 1969/.

$$\det\left[\mathbf{K}_T^{(k)} + \frac{1}{\alpha^{(i)}} \mathbf{P} \mathbf{v}^{(i)t}\right] = \det \mathbf{K}_T^{(k)} \cdot \left(1 + \frac{1}{\alpha^{(i)}} \cdot \Delta \mathbf{u}^{Tt} \cdot \mathbf{v}^{(i)}\right) \quad (6.22)$$

### 6.2.4 Direkte Erfüllung der Zusatzgleichung

Die für die Durchsetzung der Bogenlängenverfahren in der Anwendung wesentlichen Arbeiten von /Ramm 1981/ und /Crisfield 1981/ gehen nicht von einer Newton Formulierung /Riks 1972,1979/, /Keller 1978/ aus, sondern teilen das erweiterte Gleichungssystem mit einfachen Matrizenoperationen in zwei Teile auf, die sukzessive gelöst werden. Dieses Vorgehen wurde erstmals von /Batoz,Dhatt 1979/ für verschiebungsgesteuerte Verfahren vorgeschlagen.

Das nichtlineare, erweiterte Gleichungssystem lautet mit Gl. (6.3) und (6.7):

$$\begin{aligned} F(\mathbf{x}, \lambda, \mathbf{P}) &= \mathbf{R}(\mathbf{x}) - \lambda \cdot \mathbf{P} = \mathbf{0} & (a) \\ f(\mathbf{x}, \lambda, \mathbf{P}) &= 0 & (b) \end{aligned} \quad (6.23)$$

Die Anwendung des Newtonverfahrens auf Gl. (6.23a) mit den Verschiebungen als Variable führt auf:

$$\mathbf{K}_T^{(i)} \Delta \mathbf{u}^{(i)} = -(\mathbf{R}(\mathbf{x}^{(i)}) - \lambda \mathbf{P}) \quad (6.24)$$

Die Lösung von Gleichung (6.24) kann in zwei Terme aufgespalten werden

$$\begin{aligned} \Delta \mathbf{u}^{(i)} &= \underbrace{-\mathbf{K}_T^{(i)-1} \cdot \mathbf{R}(\mathbf{x}^{(i)})}_{\Delta \bar{\mathbf{u}}^{II}} + \lambda \underbrace{\mathbf{K}_T^{(i)-1} \cdot \mathbf{P}}_{\Delta \mathbf{u}^I} \\ &= \Delta \bar{\mathbf{u}}^{II} + \lambda \cdot \Delta \mathbf{u}^I \end{aligned} \quad (6.25)$$

Eine für strukturmechanische Aufgaben günstigere Aufteilung ergibt sich aus der Aufspaltung des Lastfaktors in

$$\lambda = {}^m \lambda + \lambda^{(i)} + \Delta \lambda^{(i)} \quad (6.26)$$

wie sie bereits in Gl. 6.8 zu Grunde gelegt wurde.

*Bemerkung 6.1:* Das hier geschilderte Vorgehen ist nicht auf lineare Proportionalitätsfaktoren beschränkt; es können in ähnlicher Weise z.B. globale Homotopien der Form  $\mathbf{H}(\mathbf{x}, \lambda) = \mathbf{F}(\mathbf{x}) - e^{-\lambda} \mathbf{F}(\mathbf{x}_0)$  /Allgower,Georg 1980/ untersucht werden. In diesem Fall muß dann von einer Aufteilung wie in Gl. (6.26) abgesehen werden. Wesentlich für das Vorgehen ist, daß der Vektor  $\mathbf{P}$  bzw.  $\mathbf{F}(\mathbf{x}^{(0)})$  unabhängig vom aktuellen Faktor  $\lambda$  ist.

Unter Berücksichtigung von Gl. (6.26) läßt sich die rechte Seite der Gleichung (6.24) darstellen als

$$\begin{aligned}
-(\mathbf{R}(\mathbf{x}^{(i)}) - \lambda \cdot \mathbf{P}) &= -(\mathbf{R}(\mathbf{x}^{(i)}) - ({}^m\lambda + \lambda^{(i)}) \cdot \mathbf{P}) + \Delta\lambda^{(i)} \cdot \mathbf{P} \\
&= -\mathbf{F}^{(i)} + \Delta\lambda^{(i)} \cdot \mathbf{P}
\end{aligned} \tag{6.27}$$

Da der Proportionalitätsfaktor  $\lambda$  in Gl. 6.23a nur linear auftritt, entspricht dies mit Gl. 6.24 dem entsprechenden Teil der Newton Gleichung des erweiterten Systems. Damit wird Gleichung (6.25) wieder in dieselbe Form überführt, wie sie sich als Ergebnis des Newton-Verfahrens mit Reduktion auf das ursprüngliche Gleichungssystem ergab (Gl. 6.20).

$$\Delta\mathbf{u}^{(i)} = \Delta\lambda^{(i)} \cdot \Delta\mathbf{u}^I + \Delta\mathbf{u}^{II} \tag{6.28}$$

siehe Gl. (6.18) und Gl. (6.21)

$\Delta\lambda^{(i)}$  ist nun mit Hilfe der zusätzlichen Gleichung (6.23b) zu bestimmen.

Im allgemeinen Fall ist  $f(\mathbf{x}, \lambda, \mathbf{P})$  nichtlinear für  $\lambda$  bzw.  $\Delta\lambda^{(i)}$ .

Für lineare bzw. quadratische Zusatzgleichungen lassen sich die Lösungen für  $\Delta\lambda^{(i)}$  direkt angeben, /Ramm 1981/ bzw. /Crisfield 1981/.

Wird das Newton Verfahren zur Lösung von Gl. 6.23b verwendet, so gilt:

$$f(\mathbf{x}, \lambda, \mathbf{P}) = f(\mathbf{x}(\lambda), \lambda, \mathbf{P}) = f_1(\mathbf{x}(\lambda)) + f_2(\lambda, \mathbf{P}) \tag{6.29}$$

Ableitung nach  $\lambda$  :

$$\begin{aligned}
\frac{d}{d\epsilon} f[\mathbf{x}(\lambda^{(i)} + \epsilon\Delta\lambda^{(i)}), \lambda^{(i)} + \epsilon\Delta\lambda^{(i)}, \mathbf{P}] |_{\epsilon=0} &= \frac{d}{d\epsilon} f_1[\mathbf{x}(\lambda^{(i)} + \epsilon\Delta\lambda^{(i)})] |_{\epsilon=0} \\
&+ \frac{d}{d\epsilon} f_2(\lambda^{(i)} + \epsilon\Delta\lambda^{(i)}, \mathbf{P}) |_{\epsilon=0}
\end{aligned} \tag{6.30}$$

Mit Gleichung 6.20 und

$$\begin{aligned}
\mathbf{x}^{(i+1)} &= \mathbf{x}^{(i)} + \Delta\mathbf{u}^{(i)} = \mathbf{x}^{(i)} + \Delta\lambda^{(i)} \cdot \Delta\mathbf{u}^I + \Delta\mathbf{u}^{II} \\
\frac{\partial \mathbf{x}^{(i+1)}}{\partial \lambda} &= \frac{d}{d\epsilon} [\mathbf{x}(\lambda + \epsilon\Delta\lambda^{(i)})] |_{\epsilon=0} = \Delta\mathbf{u}^I
\end{aligned} \tag{6.31}$$

gilt:

$$\frac{d}{d\epsilon} \cdot f_1[\mathbf{x}(\lambda^{(i)} + \epsilon\Delta\lambda^{(i)})] = \frac{\partial f}{\partial \mathbf{x}} \cdot \frac{\partial \mathbf{x}}{\partial \lambda} = \mathbf{v}^{(i)t} \Delta \mathbf{u}^I \quad (6.32)$$

$$\frac{d}{d\epsilon} f_2(\lambda^{(i)} + \epsilon\Delta\lambda^{(i)}, \mathbf{P}) = \alpha^{(i)} \quad (6.33)$$

$$\text{damit} \quad (\mathbf{v}^{(i)t} \cdot \Delta \mathbf{u}^I + \alpha^{(i)}) \cdot \Delta \lambda^{(i)} = -f^{(i)} \quad (6.34)$$

Die Änderung im Lastparameter wird dann in der i-ten Iteration zu

$$\Delta \lambda^{(i)} = \frac{-f^{(i)}}{\alpha^{(i)} + \mathbf{v}^{(i)t} \cdot \Delta \mathbf{u}^I} \quad (6.35)$$

Naturgemäß entspricht  $\Delta \lambda^{(i)}$  nach Gl. 6.35 nur für den Fall linearer Zusatzgleichungen dem Faktor nach Gl. 6.20.

## 6.3 Algorithmen

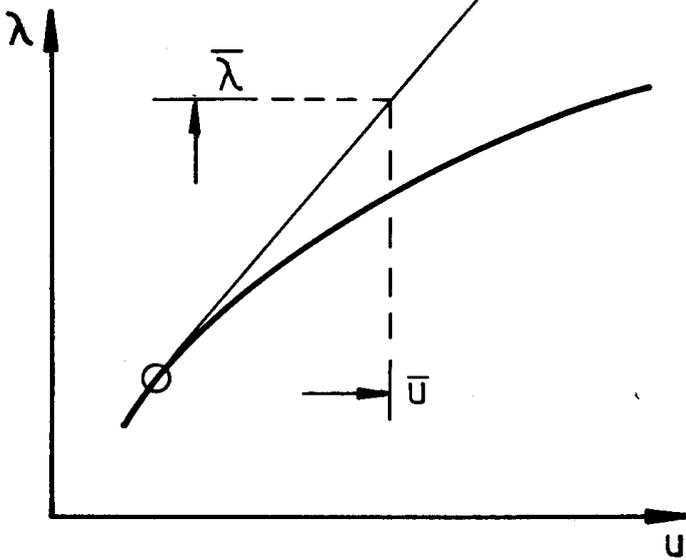
### 6.3.1 Allgemeines

Im Prinzip steht eine unendlich große Zahl von Zusatzgleichungen für mögliche Bogenlängenverfahren zur Verfügung. In den folgenden Abschnitten wird aber nur auf die wesentlichen Verfahren eingegangen.

Der Algorithmus der Bogenlängenverfahren beruht auf einer Zwei-Schritte Lösungsstrategie (Bild 6.6), die auch als Prädiktor-Korrektor Verfahren bezeichnet wird. Im ersten Schritt - dem Prädiktor-Schritt - wird der Faktor  $\lambda^{(*)}$  auf einen beliebigen Wert gesetzt und die Ausgangsgleichung (6.2) wird gelöst. Der Ergebnisvektor  $\Delta \mathbf{u}^{(*)}$  und  $\lambda^{(*)}$  werden bezüglich einer vorgegebenen Größe zurückskaliert. Als Größenvorgaben sind sowohl Lastwerte, einzelne Verschiebungswerte als auch Verschiebungsnormen oder generell die Länge des Last-Verschiebungsvektors  $\mathbf{v}^t = \alpha^{(*)} \cdot [\Delta \mathbf{u}^{(*)}, \lambda^{(*)}]$  möglich.

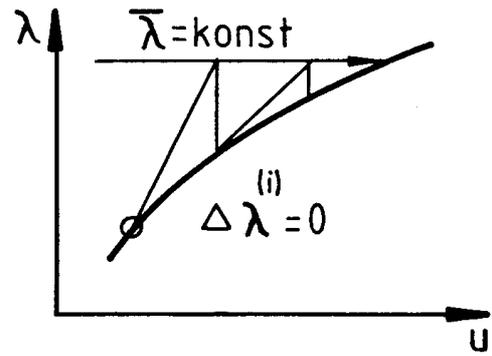
*Bemerkung 6.2:* Die Schrittweitenbegrenzung im Prädiktorschritt ist zwar in allen Algorithmen der Bogenlängenverfahren enthalten und für die Sicherung der Konvergenz wichtig. Der begrenzte Prädiktorschritt ist aber nicht das wesentliche Merkmal der Bogenlängenverfahren, auch wenn die Bezeichnung Bogenlängenverfahren (arc length procedures) weitgehend von dem Vorgehen in ersten Schritt geprägt ist.

Die wesentliche Unterscheidung zu den klassischen Verfahren mit Vorgabe eines konstanten Faktors  $\lambda$  bilden der zweite und alle weiteren Schritte - die Korrektorschritte -, die von der Wahl der Zusatzgleichung abhängig sind.

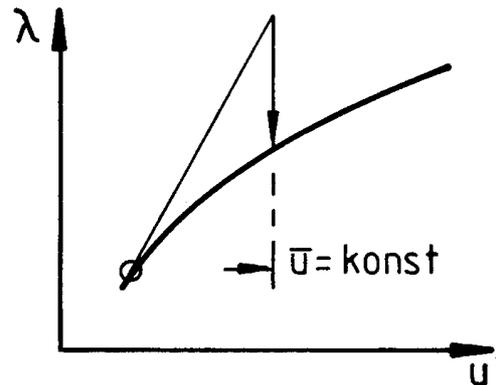


Vorgabe z.B. Lastwert  $\bar{\lambda}$   
oder Verschiebungswert  $\bar{u}$

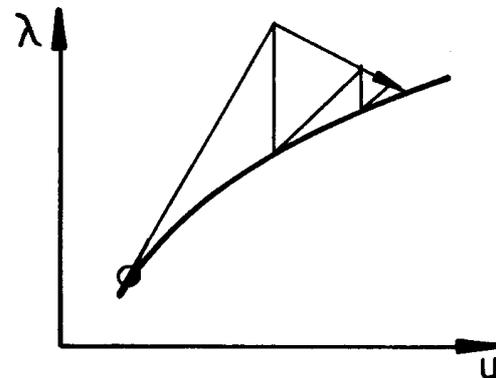
a) Prädiktorschritt



b1) Lastvorgabe



b2) Verschiebungsvorgabe



b3) Iteration auf konstanter Normalenebene

b) Korrektorschritt

**Bild 6.6 Prädiktor – Korrektor Verfahren – Möglichkeiten**

### 6.3.2 Prädiktorschritt

Vorgabe:

Verschiebungswert:  $u_j = \bar{u} = \text{const.}$

oder

Bogenlänge :  $s = \sqrt{\mathbf{x}^{(1)t} \mathbf{x}^{(1)} + \beta^2 \cdot \lambda^{(1)} \lambda^{(1)}} = \text{const.}$

mit  $s$ ... vorgegebener Wert

$\mathbf{x}^{(1)}$ ... Verschiebungswert im 1. Schritt

$\lambda^{(1)}$ ... Lastfaktor im 1. Schritt

Algorithmus:

Schritt 1.

$$\mathbf{K}_T^{(0)} \Delta \mathbf{u}^{(*)} = ({}^m \lambda + \lambda^{(*)}) \mathbf{P} - \mathbf{R}({}^m \mathbf{x}) = \lambda^{(*)} \cdot \mathbf{P}$$
$$\mathbf{x}^{(*)} = \Delta \mathbf{u}^{(*)}$$

Schritt 2. Skalierungsfaktor

Verschiebungsvorgabe

$$\alpha^{(*)} = \frac{u_j}{x_j^{(*)}}$$

oder

Bogenlängenvorgabe

$$\alpha^{(*)} = s / \sqrt{\mathbf{x}^{(*)t} \cdot \mathbf{x}^{(*)} + \beta^2 \lambda^{(*)} \cdot \lambda^{(*)}}$$

Schritt 3.

$$\mathbf{x}^{(1)} = \alpha^{(*)} \cdot \mathbf{x}^{(*)} = \Delta \mathbf{u}^I$$

$$\lambda^{(1)} = \alpha^{(*)} \cdot \lambda^{(*)}$$

${}^m \lambda$  und  ${}^m \mathbf{x}$  stehen für einen konvergierten Gleichgewichtszustand.  $\lambda^{(*)}$  ist ein beliebig vorgegebener Faktor;  $\beta$  ist ein Skalierungsfaktor, der in Abschnitt 6.4 diskutiert wird.  $s, u_j$  bzw.  $\mathbf{u}$  sind vorgegebene Werte bzw. Vektoren.

### 6.3.3 Korrektorschritte

Grundsätzlich können in der Strukturmechanik drei Typen von Zusatzgleichungen, häufig auch als Bedingungsgleichungen bezeichnet, aufgestellt werden.

1. Gleichungen, die nur Lastterme enthalten.

$$f = f(\lambda, \mathbf{P})$$

2. Gleichungen, die lineare Verschiebungsterme und evtl. Lastterme beinhalten.

$$f = f(x_i, x_j) \text{ oder } f = f(x_i, x_j, \lambda, \mathbf{P})$$

$x_i, x_j \dots$  Verschiebungskomponenten

3. Gleichungen, die auf Normen beruhen. Diese enthalten Verschiebungsterme und evtl. auch Lastterme in mehr als nur linearer Form.

Typ 1 entspricht dem klassischen Vorgehen der Lastkontrolle. Die übliche Bedingungsgleichung lautet:

$$f = \lambda^{(i)} - c = 0 \quad c \dots \text{Konstante}$$

Wird das Schema des Bogenlängenverfahrens angewendet, so ergibt Gl. 6.11 bzw. 6.20

$$\begin{aligned} \Delta \lambda^{(i)} &= 0 \\ \text{und} \quad \Delta \mathbf{u}^{(i)} &= \Delta \mathbf{u}^{II} \end{aligned}$$

D. h. die Bogenlängenverfahren enthalten als Untermenge auch die klassische Lastkontrolle.

In Typ 2 sind alle Zusatzgleichungen enthalten, die nur lineare Terme - Verschiebungen und evtl. Lastwerte - aufweisen. Zu diesem Typ gehören die Verschiebungskontrolle /Batoz, Dhatt 1979/ und die Iteration auf Normalenebenen /Riks 1972/, /Ramm 1981/.

Zu Typ 3 gehören alle Zusatzgleichungen, die auch Terme quadratischer und höherer Ordnung in den Verschiebungen und evtl. Lastwerten enthalten.

Diese Zusatzgleichungen beschreiben beispielsweise Normen, die Länge bzw. eine Teillänge des Last-Verschiebungsvektors /Crisfield 1981/.

### Algorithmus - Korrektorschritte

Iterationsschleife:

Schritt 4.

$$\Delta \mathbf{u}^{II} = -\mathbf{K}_T^{(k)-1} \cdot \mathbf{F}^{(i)}$$

Schritt 5.

$$\Delta \lambda^{(i)} = \dots \text{abhängig von der Zusatzgleichung} \\ \text{und dem gewählten Lösungsverfahren}$$

Schritt 6.

$$\Delta \mathbf{u}^{(i)} = \Delta \lambda^{(i)} \cdot \Delta \mathbf{u}^I + \Delta \mathbf{u}^{II}$$

Schritt 7.

$$\lambda^{(i+1)} = \lambda^{(i)} + \Delta \lambda^{(i)}; \\ \mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} + \Delta \mathbf{u}^{(i)}$$

Abfrage Konvergenz ?

Die neben der Standardlastkontrolle üblichen Bedingungsgleichungen, die zu unterschiedlichen Formeln für  $\Delta \lambda^{(i)}$  in Schritt 5 führen, werden in den folgenden Abschnitten diskutiert.

## 6.3.4 Lineare Bedingungsgleichungen

### 6.3.4.1 Einfache Verschiebungssteuerung

Die einfachste lineare Zusatzgleichung ist die Vorgabe eines Verschiebungswertes  $\bar{u}_j = konst.$  im Inkrement /Haisler,Stricklin,Key 1977/.

Damit gilt für die Zusatzgleichung:

$$f(\mathbf{x}) = u_j - \bar{u}_j = 0 \tag{6.36}$$

Da die Zusatzgleichung linear ist, führen alle in Abschnitt 6.2 geschilderten Verfahren zur selben Änderung im Lastwert  $\Delta \lambda^{(i)}$ .

$$\mathbf{v}^{(i)t} = [0 \ 0 \ 0 \ 0 \ \dots \ 1 \ \dots \ 0] \\ \quad \quad \quad 1 \ 2 \ \dots \dots \ j \ \dots$$

$$\alpha^{(i)} = 0$$

$$\mathbf{v}^{(i)t} \cdot \Delta \mathbf{u}^I = \Delta u_j^I$$

$$\mathbf{v}^{(i)t} \cdot \Delta \mathbf{u}^{II} = \Delta u_j^{II}$$

Damit:

$$\Delta \lambda^{(i)} = - \frac{\overbrace{(u_j^{(i)} - \bar{u}_j)}^{=0} + \Delta u_j^{II}}{\Delta u_j^I} = - \frac{\Delta u_j^{II}}{\Delta u_j^I} \quad (6.37)$$

Die Formel (6.37) entspricht exakt der in /Batoz,Dhatt 1979/ durch einfache Matrizenoperationen hergeleiteten Gleichung.

*Bemerkung 6.3:* Schwierigkeiten mit diesem Verfahren können bei sogenannten Verschiebungsumkehrpunkten (Turning points) auftreten, wenn in einer Last-Verschiebungskurve die Richtung der Verschiebung wechselt. Dann kann im ungünstigen Fall  $\Delta u_j^I = 0$  werden. Dies ist bei der Auswahl der steuernden Verschiebung zu beachten. Eventuell muß auf eine andere Verschiebung zur Steuerung übergegangen werden.

#### 6.3.4.2 Verschiebungssteuerung mit mehreren Verschiebungstermen

Um die in 6.3.4.1 geschilderten Schwierigkeiten zu umgehen, wurden von /Powell,Simons 1981/ mehrere Verschiebungswerte gleichzeitig zur Steuerung der Iteration herangezogen, siehe auch /Simons,Bergan,Nygard 1984/. Werden die Verschiebungen unterschiedlich gewichtet, so lautet die Zusatzgleichung

$$w_j \cdot u_j + w_k \cdot u_k + \dots w_l \cdot u_l = c = konst$$

damit

$$f = w_j \cdot u_j + w_k \cdot u_k + w_l \cdot u_l - c = 0 \quad (6.38)$$

Es ist:  $\mathbf{v}^{(i)t} = [0 \dots w_j \dots w_k \dots w_l \dots 0]$

$\mathbf{v}^{(i)}$   $\hat{=}$  Vektor der Wichtungskoeffizienten

$$\alpha^{(i)} = 0$$

$$\mathbf{v}^{(i)t} \cdot \Delta \mathbf{u}^I = w_j \cdot u_j^I + w_k \cdot u_k^I + w_l \cdot u_l^I$$

$$\mathbf{v}^{(i)t} \cdot \Delta \mathbf{u}^{II} = w_j \cdot u_j^{II} + w_k \cdot u_k^{II} + w_l \cdot u_l^{II}$$

Lastfaktor:

$$\Delta\lambda^{(i)} = -\frac{\overbrace{[\mathbf{v}^{(i)t} \cdot \mathbf{u}^{(i)} - c]}^{=0} + \mathbf{v}^{(i)t} \cdot \Delta\mathbf{u}^{II}}{\mathbf{v}^{(i)t} \cdot \Delta\mathbf{u}^I} = -\frac{\mathbf{v}^{(i)t} \cdot \Delta\mathbf{u}^{II}}{\mathbf{v}^{(i)t} \cdot \Delta\mathbf{u}^I} \quad (6.39)$$

*Bemerkung 6.4:* Wie bei der einfachen Verschiebungssteuerung muß vermieden werden, daß der Nenner in Gl. (6.39) zu Null wird. Hierzu ist der Vektor der Wichtungskoeffizienten anzupassen. Außerdem sollten Verschiebungen mit wesentlichen Komponenten im Verschiebungsvektor zur Kontrolle herangezogen werden. Diese Auswahl, die nur schlecht automatisiert werden kann, ist die Schwäche des Verfahrens.

#### 6.3.4.3 Last-Verschiebungssteuerung mit mehreren Verschiebungen

Wird die Zusatzgleichung (6.38) bzw (6.36) noch um einen Lastterm erweitert, so können die Probleme mit Verschiebungsumkehrpunkten vermieden werden.

$$f = w_j \cdot u_j + w_k \cdot u_k + w_l \cdot u_l - c + \beta \cdot \lambda = 0 \quad (6.40)$$

Im Unterschied zu 6.3.4.2 bzw. 6.3.4.1 ist nun  $\alpha^{(i)} = \beta$  und der Lastfaktor

$$\Delta\lambda^{(i)} = -\frac{\mathbf{v}^{(i)t} \Delta\mathbf{u}^{II}}{\beta + \mathbf{v}^{(i)t} \Delta\mathbf{u}^I} \quad (6.41)$$

Wie bei den Gleichungen (6.37) und (6.39) muß auch hier darauf geachtet werden, daß der Nenner nicht verschwindet, d.h. Iterationsebene und Kurvenverlauf parallel laufen. Dies ist auch bei dieser Zusatzgleichung nicht grundsätzlich gesichert.

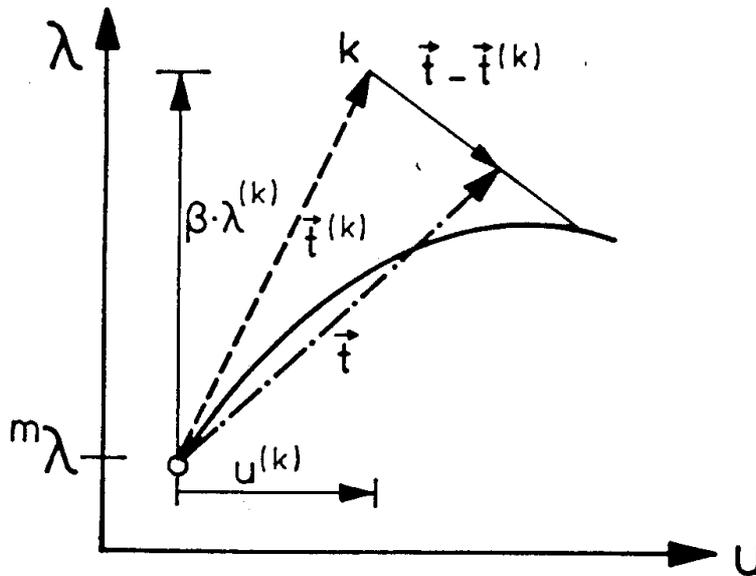
#### 6.3.4.4 Iteration auf Normalenebenen

Diese Form der Zusatzgleichung wurde ursprünglich eingeführt, um zu verhindern, daß die Determinante der Systemmatrix in Durchschlagspunkten zu Null wird. /Riks 1972/ wies nach, daß die Jacobideterminante des erweiterten Systems dann den größten Wert im Durchschlagspunkt annimmt, wenn

die Iterationsebene den Last/Verformungspfad möglichst orthogonal schneidet. Eine Bedingungsgleichung, die diese Forderung möglichst gut erfüllt, lautet

$$f = \mathbf{t}^{(k)} \cdot (\mathbf{t} - \mathbf{t}^{(k)}) = 0 \quad (6.42)$$

Für ein System mit einem Freiheitsgrad kann dies geometrisch interpretiert werden (Bild 6.7).



**Bild 6.7 Iteration auf Normalenebenen**

Deutlich wird, daß die oben zitierte optimale Forderung mit dieser Zusatzgleichung zwar nicht erreicht, aber für den Fall gleichmäßiger Krümmung der Lastverschiebungskurve gut angenähert wird.

Der Vektor  $\mathbf{t}$  beschreibt die aktuelle Lage ausgehend von einer Gleichgewichtslage  $m$ . Der Vektor  $\mathbf{t}^{(k)}$  stellt die Normale auf der Iterationsebene dar.

$$\mathbf{t}^{(k)} = \mathbf{t}(\mathbf{x}^{(k)}, \beta \cdot \mathbf{P}, \lambda^{(k)}) \quad (6.43)$$

$\beta$  ist ein Skalierungsfaktor, der in Abschnitt 6.4 diskutiert wird.

Bleibt der Normalenvektor  $\mathbf{t}^{(k)}$  im Lastschritt unverändert  $\mathbf{t}^{(k)} = \mathbf{t}^{(1)}$ , dann entspricht dies der Iteration auf einer konstanten Normalenebene. Wird der Normalenvektor jeweils der aktuellen Lage angepaßt  $\mathbf{t}^{(k)} = \mathbf{t}^{(i)}$ , so wird auf angepaßten Normalenebenen iteriert /Ramm 1981/ (siehe Bild 6.7).

Es ist klar, daß die optimale Iterationsebene, bei der die Normale  $\mathbf{t}^{(k)}$  der Tangente  $\mathbf{t}^*$  an die Last- Verschiebungskurve entspricht, während der Iteration nur verwendet werden kann, wenn mit einem echten Newton-Verfahren gearbeitet wird.  $m$  in Bild 6.7 fällt dann mit dem aktuellen Iterationsschritt  $i$  zusammen. Damit wird auf der aktuellen Normalenebene iteriert. Dann gilt:

$$f = (\mathbf{x}^{(k)} - {}^m \mathbf{x})^t (\mathbf{x} - \mathbf{x}^{(k)}) + \beta^2 \cdot \mathbf{P}^t \mathbf{P} (\lambda^{(k)} - {}^m \lambda) (\lambda - \lambda^{(k)}) \quad (6.44)$$

$\beta^2 \mathbf{P}^T \mathbf{P}$  läßt sich durch eine Konstante  $\bar{\beta} = \beta^2 \mathbf{P}^T \mathbf{P}$  ersetzen.

Damit lauten die Lastfaktoren für die verschiedenen Normalenebenen:

Iteration auf konstanten ( $k = 1$ ) und angepaßten ( $k = i$ ) Normalenebenen

$$\Delta \lambda^{(i)} = - \frac{(\mathbf{x}^{(k)} - {}^m \mathbf{x})^t \cdot \Delta \mathbf{u}^{II}}{\bar{\beta} \cdot \lambda^{(k)} + (\mathbf{x}^{(k)} - {}^m \mathbf{x})^t \cdot \Delta \mathbf{u}^I} \quad (6.45)$$

*Bemerkung 6.5:* Das Residuum  $f^{(i)}$  ist in jeder Iteration gleich Null, da die lineare Zusatzgleichung exakt erfüllt wird.

Iteration auf aktuellen Normalenebenen

Hier ist  $\mathbf{x}^{(k)} - {}^m \mathbf{x} = \lambda^{(k)} \cdot \Delta \mathbf{u}^I$ .  $\lambda^{(k)}$  entspricht dem Unterschied im Lastparameter zwischen dem Gleichgewichtspunkt  $i^*$  und  $i$  und ist während der Iteration unbekannt. D.h. in der hier dargestellten einfachen Form kann zwar die Tangente an die Last-Verschiebungskurve erzeugt werden, aber ihre Länge ist unbestimmt. Dies ist für die Forderung  $f = \mathbf{t}^{(k)} \cdot (\mathbf{t} - \mathbf{t}^{(k)}) = 0$  jedoch ohne Bedeutung, wie aus der Gleichung für den Lastparameter deutlich wird.

$$\Delta \lambda^{(i)} = - \frac{\lambda^{(k)} \Delta \mathbf{u}^{I^t} \Delta \mathbf{u}^{II}}{\bar{\beta} \cdot \lambda^{(k)} + \lambda^{(k)} \cdot \Delta \mathbf{u}^{I^t} \cdot \Delta \mathbf{u}^I} = - \frac{\Delta \mathbf{u}^{I^t} \cdot \Delta \mathbf{u}^{II}}{\bar{\beta} + \Delta \mathbf{u}^{I^t} \cdot \Delta \mathbf{u}^I} \quad (6.46)$$

**Diskussion:**

1. Die einfache Ermittlung der Tangente durch Gleichungslösung ist der Gewinnung über die Inversion der erweiterten Matrix /Riks 1972/ an Effizienz überlegen.

2. Die Iteration auf angepaßten Normalebenen ist der Iteration mit konstanten Normalenebenen vorzuziehen, da sich dann die Iterationsrichtung dem Verlauf der Last-Verschiebungskurve anpaßt.
3. Die Güte der Iteration auf aktuellen Normalenebenen hängt nach den Erfahrungen des Autors stark von der Schrittweite und der Korrektheit der Tangentensteifigkeitsmatrix ab. Liegt der iterative Tangentenvektor  $\Delta u^f$  weit neben dem Tangentenvektor einer Gleichgewichtslage, so kann die Iteration schnell in die falsche Richtung divergieren.
4. Die Iteration auf angepaßten Normalenebenen ist weniger empfindlich gegenüber der Änderung des Tangentenvektors und daher häufig stabiler als die Iteration auf aktuellen Normalenebenen. Außerdem kann auch bei Verwendung des modifizierten Newton Verfahrens auf angepaßten Normalenebenen iteriert werden, während aktuelle Normalenebenen dann nicht berechnet werden können.
5. Sowohl bei der Iteration auf konstanten als auch angepaßten Normalebenen kann der Nenner von Gleichung (6.46) zu Null werden, d.h. die Last- Verschiebungskurve parallel zur Iterationsebene verlaufen. Dies ist bei der Iteration auf aktuellen Normalenebenen nicht möglich.

*Bemerkung 6.6:* Die Iteration auf Normalenebenen kann als Sonderfall der Last-Verschiebungssteuerung mit dem inkrementellen Verschiebungsvektor + Lastfaktor bzw. dem aktuellen Tangentenvektor + Lastfaktor ( $\lambda^{(k)} = 1$ ) als Wichtungsvektor angesehen werden.

#### 6.3.4.5 Iteration mit konstanter äußerer Energie

Eine weitere Variation des Wichtungsvektors wird von /Bathe, Dvorkin 1983/ mit dem Kriterium der konstanten äußeren Energie im Inkrement vorgestellt. Der Algorithmus unterscheidet sich von den hier vorgestellten Methoden durch die Schrittweitenbegrenzung über einen Energiewert im Prädiktorschritt.

##### Prädiktorschritt:

Vorgabe eines Energiewertes:  $|W| = const.$

Skalierung des Lastwerts :

$$({}^m\lambda + \frac{1}{2}\alpha^{(*)} \cdot \lambda^{(*)}) \cdot \mathbf{P}^t \cdot \mathbf{x}^{(*)} = |W| \quad (6.47)$$

$$\alpha^{(*)} = \frac{2}{\lambda^{(*)}} \left( \frac{|W|}{\mathbf{P}^t \mathbf{x}^{(*)}} - {}^m\lambda \right) \quad (6.48)$$

Alle weiteren Schritte verlaufen analog wie in Abschnitt 6.3.1.

Im Korrektorschritt wird der Zuwachs der Energie zu Null festgelegt.  
/Bathe, Dvorkin 1983/ schreiben als Zusatzgleichung:

$$({}^m\lambda + \lambda^{(i)} + \frac{1}{2}\Delta\lambda^{(i)}) \mathbf{P}^t \cdot (\mathbf{x} - \mathbf{x}^{(i)}) = 0 \quad (6.49)$$

Diese Zusatzgleichung beinhaltet zwei Lösungen. Die erste Lösung ist trivial und würde das Lastinkrement völlig umkehren:

$$\Delta\lambda^{(i)} = -2({}^m\lambda + \lambda^{(i)}) \quad (6.50)$$

Innerhalb des Algorithmus ist diese Lösung ohne Bedeutung.

Der zweite Teil der Lösung ergibt den Lastfaktor:

mit

$$\mathbf{v}^{(i)} = \mathbf{P}$$

$$\alpha^{(i)} = 0$$

Lastfaktor:

$$\Delta\lambda^{(i)} = -\frac{\mathbf{P}^t \Delta \mathbf{u}^{II}}{\mathbf{P}^t \Delta \mathbf{u}^I} \quad (6.51)$$

*Bemerkung 6.7:* Bei Betrachtung von  $\mathbf{P}$  als speziellem Wichtungsvektor sind die Vor- und Nachteile dieses Vorgehens identisch mit denen der Verfahren mit Iteration auf Ebenen. Der Nenner kann ebenfalls zu Null werden, d.h. Iterationsebene und Lösungskurve können parallel verlaufen. Ein weiterer Nachteil des Verfahrens ist, daß nur die belasteten Freiheitsgrade die Iteration kontrollieren. Im Gegensatz zur Iteration auf Normalenebenen muß die Schrittweite  $W$  (= Energie) daher besonders stark skaliert werden. Bathe/Dvorkin empfehlen ihr Verfahren deswegen besonders für stark nichtlineare Bereiche z. B. in der Nähe von Durchschlagpunkten.

## 6.3.5 Nichtlineare Bedingungsgleichungen

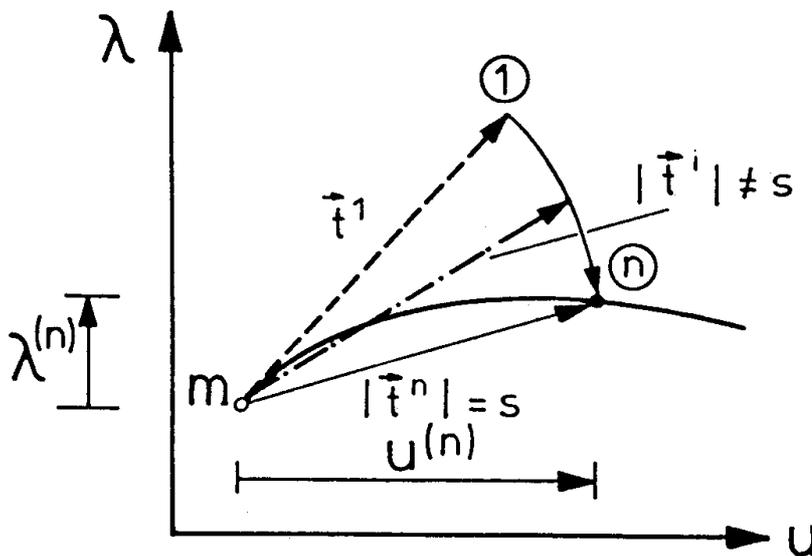
### 6.3.5.1 Quadratische Zusatzgleichung

Eine wesentliche Verbesserung bezüglich Schrittweitensteuerung und globaler Konvergenzsicherung /Watson, Holzer 1983/ stellt der Vorschlag von /Crisfield 1981/ und /Keller 1978/ mit einer quadratischen Zusatzgleichung dar. Damit wird vermieden, daß Iterationsebene und Last-Verschiebungskurve parallel zueinander bzw. auseinander laufen. Schnittpunkte der Iterationsfläche mit der Lastverschiebungskurve sind dann zumindest für sinnvolle Schrittweiten gesichert.

Die Zusatzgleichung lautet:

$$f = \sqrt{(\mathbf{x} - {}^m \mathbf{x})^t (\mathbf{x} - {}^m \mathbf{x}) + \beta^2 \cdot \mathbf{P}^t \cdot \mathbf{P} (\lambda - {}^m \lambda)^2} - s = 0 \quad (6.52)$$

Diese Form, bei der die Länge des Last-/ Verschiebungsincrementes während der Iterationen konstant gehalten wird, ist das Verfahren, welches der Beschreibung "Bogenlängenverfahren" am nächsten kommt.



**Bild 6.8** Iteration mit quadratischer Zusatzgleichung

Zur Bestimmung des Lastfaktors kann sowohl das Newton Verfahren für das Gesamtsystem nach Abschnitt 6.2.3 /Schweizerhof, Wriggers 1986/ benutzt werden, oder die Zusatzgleichung kann direkt erfüllt werden. Im Vorschlag, diese Zusatzgleichung direkt zu lösen, liegt der wesentliche Beitrag /Crisfields

1981/. Das Vorgehen nach dem Newton-Verfahren wird mit allen algorithmischen Einzelheiten bereits 1978 von /Keller/ beschrieben.

### 6.3.5.1.1 Newton-Verfahren für das Gesamtsystem

Es ist

$$\mathbf{v}^{(i)} = (\mathbf{x}^{(i)} - {}^m \mathbf{x})^t \cdot \frac{1}{\bar{f}}$$

$$\alpha^{(i)} = \beta^2 \cdot \mathbf{P}^t \mathbf{P} \lambda^{(i)} \cdot \frac{1}{\bar{f}}$$

mit

$$\bar{\beta} = \beta^2 \mathbf{P}^t \mathbf{P}$$

und

$$\bar{f} = \sqrt{(\mathbf{x}^{(i)} - {}^m \mathbf{x})^t (\mathbf{x}^{(i)} - {}^m \mathbf{x}) + \beta^2 \mathbf{P}^t \mathbf{P} \cdot \lambda^{(i)2}} \quad (6.53)$$

Lastfaktor:

$$\Delta \lambda^{(i)} = - \frac{f^{(i)} \cdot \bar{f} + (\mathbf{x}^{(i)} - {}^m \mathbf{x})^t \cdot \Delta \mathbf{u}^{II}}{\bar{\beta} \cdot \lambda^{(i)} + (\mathbf{x}^{(i)} - {}^m \mathbf{x})^t \cdot \Delta \mathbf{u}^I} \quad (6.54)$$

### 6.3.5.1.2 Direkte Erfüllung der Zusatzgleichung

/Crisfield 1981/ schlägt die direkte Erfüllung der quadratischen Zusatzgleichung vor.

Mit

$$\begin{aligned} \mathbf{x}^{(i+1)} &= \mathbf{x}^{(i)} + \Delta \mathbf{u}^{(i)} = \mathbf{x}^{(i)} + \Delta \lambda^{(i)} \Delta \mathbf{u}^I + \Delta \mathbf{u}^{II} \\ \mathbf{x}^{(i)} - {}^m \mathbf{x} &= \mathbf{u}^{(i)} \\ \mathbf{x} - {}^m \mathbf{x} &= \mathbf{u}^{(i)} + \Delta \mathbf{u}^{(i)} \\ \lambda &= {}^m \lambda + \lambda^{(i)} + \Delta \lambda^{(i)} \end{aligned}$$

gilt:

$$(\mathbf{u}^{(i)} + \Delta \mathbf{u}^{(i)})^t (\mathbf{u}^{(i)} + \Delta \mathbf{u}^{(i)}) + \bar{\beta} \cdot (\lambda^{(i)} + \Delta \lambda^{(i)})^2 = s^2 \quad (6.55)$$

Quadratische Gleichung für  $\Delta\lambda^{(i)}$ :

$$a \cdot \Delta\lambda^{(i)2} + 2b \cdot \Delta\lambda^{(i)} + c + d - s^2 = 0 \quad (6.56)$$

$$a = \Delta\mathbf{u}^{I^t} \cdot \Delta\mathbf{u}^I + \bar{\beta}$$

$$b = \mathbf{u}^{(i)t} \Delta\mathbf{u}^I + \Delta\mathbf{u}^{II^t} \Delta\mathbf{u}^I + \bar{\beta}\lambda^{(i)}$$

$$c = 2\mathbf{u}^{(i)t} \Delta\mathbf{u}^{II} + \Delta\mathbf{u}^{II^t} \Delta\mathbf{u}^{II}$$

$$d = \mathbf{u}^{(i)t} \mathbf{u}^{(i)} + \bar{\beta}\lambda^{(i)2}$$

$d \hat{=} s^2$ , wenn im vorigen Schritt die quadratische Zusatzgleichung angesetzt wurde.

Es ergeben sich zwei Lösungen für den Lastfaktor  $\Delta\lambda^{(i)}$ :

$$\Delta\lambda^{(i)} = \frac{-b \pm \sqrt{b^2 - ac}}{a} \quad (6.57)$$

Übliche Lösung: Reelle Wurzeln

Die Auswahl der richtigen Lösung kann nach verschiedenen Kriterien erfolgen. /Crisfield 1981/ schlägt vor, die Wurzel von (Gl.6.57) zu wählen, die zum größten Wert von  $g = \mathbf{u}^{(i+1)t} \mathbf{u}^{(i)}$  führt, d.h. der Lastfaktor wird gewählt, der den kleinsten Winkel zwischen den Verschiebungen  $\mathbf{u}^{(i)}$  und  $\mathbf{u}^{(i+1)}$  ergibt.

Vergleich der Werte:

$$\begin{aligned} g = \mathbf{u}^{(i+1)t} \mathbf{u}^{(i)} &= \mathbf{u}^{(i)t} (\mathbf{u}^{(i)} + \Delta\lambda^{(i)} \Delta\mathbf{u}^I + \Delta\mathbf{u}^{II}) \\ &= \mathbf{u}^{(i)t} (\mathbf{u}^{(i)} + \Delta\mathbf{u}^{II}) + \Delta\lambda^{(i)} \cdot \mathbf{u}^{(i)t} \cdot \Delta\mathbf{u}^I \end{aligned} \quad (6.58)$$

Als zweite Möglichkeit läßt sich die Newtonlösung verwenden:

Es wird das Vorzeichen gesucht und überprüft, welche Lösung der Newton Lösung am nächsten kommt.

Da die Zusatzgleichung in jedem Iterationsschritt erfüllt wird, gilt:

$$\text{sign}\Delta\lambda^{(i)} = \text{sign}\left[-\frac{\mathbf{u}^{(i)t} \Delta\mathbf{u}^{II}}{\bar{\beta} \cdot \lambda^{(i)} + \mathbf{u}^{(i)t} \Delta\mathbf{u}^I}\right] \quad (6.59)$$

### Sonderfall:

Wird  $b < ac$ , so ergeben sich keine reellen Lösungen für  $\Delta\lambda^{(i)}$ . Dann sollte mit der Lösung des Newton Verfahrens fortgefahren werden. In diesem - seltenen - Fall empfiehlt es sich auch, die Schrittweite zu reduzieren, da nach den Erfahrungen des Autors meist numerische Probleme bei der Ermittlung von  $\Delta\mathbf{u}^{II}$  bzw.  $\Delta\mathbf{u}^I$  hierfür verantwortlich sind.

#### 6.3.5.1.3 Iterative Lösung der Zusatzgleichung

Die Probleme mit der Auswahl der Wurzel bzw. mit komplexen Wurzeln können manchmal durch die iterative Lösung der Zusatzgleichung umgangen werden.

Nach den Gleichungen (6.28)-(6.35) gilt für diese separate Iteration:

Mit

$$f = (\mathbf{x} - {}^m\mathbf{x})^t(\mathbf{x} - {}^m\mathbf{x}) + \bar{\beta}(\lambda - {}^m\lambda)^2 - s^2 = 0 \quad (6.60)$$

Werden die einzelnen Iterationsschritte mit dem Laufindex  $k$  bezeichnet, so ist:

$\Delta^*\lambda^{(k)}$  ... Änderung im Iterationsschritt

Damit

$${}^*\lambda^{(k)} = \lambda^{(i)} + \sum_{j=1}^k \Delta^*\lambda^{(j-1)}$$

$${}^*\mathbf{x}^{(k)} = \mathbf{x}^{(i)} + \sum_{j=1}^k \Delta^*\lambda^{(j-1)} \Delta\mathbf{u}^I + \Delta\mathbf{u}^{II}$$

bzw.

$${}^*\mathbf{u}^{(k)} = \mathbf{u}^{(i)} + \sum_{j=1}^k \Delta\lambda^{(j-1)} \Delta\mathbf{u}^I + \Delta\mathbf{u}^{II} = {}^*\mathbf{x}^{(k)} - {}^m\mathbf{x}$$

Für das Newton-Verfahren sind nach Gl. (6.9), (6.10) bzw. (6.31 - 6.33)

$$\begin{aligned} {}^* \mathbf{v}^{(k)t} &= 2({}^* \mathbf{x}^{(k)} - {}^m \mathbf{x}) = 2{}^* \mathbf{u}^{(k)} \\ {}^* \alpha^{(k)} &= 2\bar{\beta} \quad {}^* \lambda^{(k)} \end{aligned} \quad (6.61)$$

Wird angenommen, daß im Prädiktorschritt bzw. im vorigen übergeordneten Iterationsschritt  $i$  die Zusatzgleichung exakt erfüllt wurde, gilt:

$$\Delta {}^* \lambda^{(0)} = 0 \quad \text{und} \quad {}^* f^{(0)} = 0$$

$$\begin{aligned} {}^* f^{(1)} &= (\mathbf{u}^{(i)} + \Delta \mathbf{u}^{II})^t (\mathbf{u}^{(i)} + \Delta \mathbf{u}^{II}) + \bar{\beta} (\lambda^{(i)} - \lambda)^2 - s^2 \\ &= (2\mathbf{u}^{(i)} + \Delta \mathbf{u}^{II})^t \Delta \mathbf{u}^{II} + \underbrace{[\mathbf{u}^{(i)t} \mathbf{u}^{(i)} + \bar{\beta} (\lambda^{(i)} - \lambda)^2 - s^2]}_{{}^* f^{(0)}=0} = 0 \end{aligned} \quad (6.62)$$

$$\Delta {}^* \lambda^{(1)} = - \frac{(2\mathbf{u}^{(i)} + \Delta \mathbf{u}^{II})^t \Delta \mathbf{u}^{II}}{2\bar{\beta} \lambda^{(i)} + 2(\mathbf{u}^{(i)} + \Delta \mathbf{u}^{II})^t \Delta \mathbf{u}^{II}} = - \frac{\mathbf{u}^{(i)t} \Delta \mathbf{u}^{II} + 1/2 \Delta \mathbf{u}^{IIt} \Delta \mathbf{u}^{II}}{\bar{\beta} \lambda^{(i)} + (\mathbf{u}^{(i)} + \Delta \mathbf{u}^{II})^t \Delta \mathbf{u}^{II}} \quad (6.63)$$

Für alle weiteren Schritte der separaten Iteration gilt:

$$\Delta {}^* \lambda^{(k)} = - \frac{{}^* f^{(k)}}{2\bar{\beta} \quad {}^* \lambda^{(k)} + 2{}^* \mathbf{u}^{(k)t} \Delta \mathbf{u}^{II}} \quad (6.64)$$

Konvergenz wird nach Unterschreiten einer vorgegebenen Schranke  $\epsilon$  für  $\Delta {}^* \lambda^{(k)}$  bzw.  ${}^* f^{(k)}$  angenommen.

Damit ergeben sich die Verschiebungen/Lastfaktor in der übergeordneten Iteration  $i$  zu:

$$\Delta \lambda^{(i)} = \sum_{j=1}^k \Delta {}^* \lambda^{(j)} \quad (6.65)$$

$$\Delta \mathbf{u}^{(i)} = \sum_{j=1}^k \Delta {}^* \lambda^{(j)} \Delta \mathbf{u}^{II} + \Delta \mathbf{u}^{II} \quad (6.66)$$

*Bemerkung 6.8:* Da im Fall reeller Wurzeln zwei Lösungen möglich sind, muß auch hier gesichert werden, daß die richtige Lösung gewählt wird, um ein Rückgehen auf dem Lösungspfad zu vermeiden. Daher müssen meist beide Wurzeln bestimmt werden.

### 6.3.5.1.4 Geometrische Überlegungen

/Forde, Stierner 1987/ schlagen aufgrund geometrischer Überlegungen ebenfalls eine iterative Lösung der quadratischen Zusatzgleichung vor.

Als erste Näherung wird die lineare Lösung nach Gleichung (6.20) angenommen mit

$$\begin{aligned}\Delta^* \lambda^{(1)} &= \frac{-\mathbf{u}^{(i)t} \Delta \mathbf{u}^{II}}{\bar{\beta} \cdot \lambda^{(i)} + \mathbf{u}^{(i)t} \Delta \mathbf{u}^I} \\ \Delta^* \mathbf{u}^{(1)} &= \Delta^* \lambda^{(1)} \cdot \Delta \mathbf{u}^I + \Delta \mathbf{u}^{II}\end{aligned}\quad (6.67)$$

Im zweiten Schritt wird ein Residuum nach Gleichung (6.71) ermittelt und in Gleichung (6.54) eingesetzt. Das Residuum soll die Längenänderung berücksichtigen, d.h. die Länge im 2. Schritt korrigieren.

$$*f^{(2)} = \mathbf{t}^{(i)t} \cdot \mathbf{r}^{(i+1)} \quad (6.68)$$

$\mathbf{r}^{(i+1)}$  ist der Längenzuwachs und läuft parallel zu  $\mathbf{t}^{(i+1)}$ .

$$\mathbf{r}^{(i+1)} = \frac{\mathbf{t}^{(i+1)}}{|\mathbf{t}^{(i+1)}|} (|\mathbf{t}^{(i+1)}| - |\mathbf{t}^{(i)}|) \quad (6.69)$$

$$\mathbf{t}^{(i+1)} = \mathbf{t}(\mathbf{u}^{(i)} + \Delta^* \mathbf{u}^{(i)}, \lambda^{(i)} + \Delta \lambda^{(i)}) = \mathbf{t}^{(1)} + \mathbf{n}(\Delta^* \mathbf{u}^{(1)}, \Delta^* \lambda^{(1)}) \quad (6.70)$$

Da  $\Delta^* \lambda^{(i)}$  mit den Normalenbedingungen ermittelt wurde, ist  $\mathbf{n}$  orthogonal zu  $\mathbf{t}^{(1)}$ . Damit gilt:

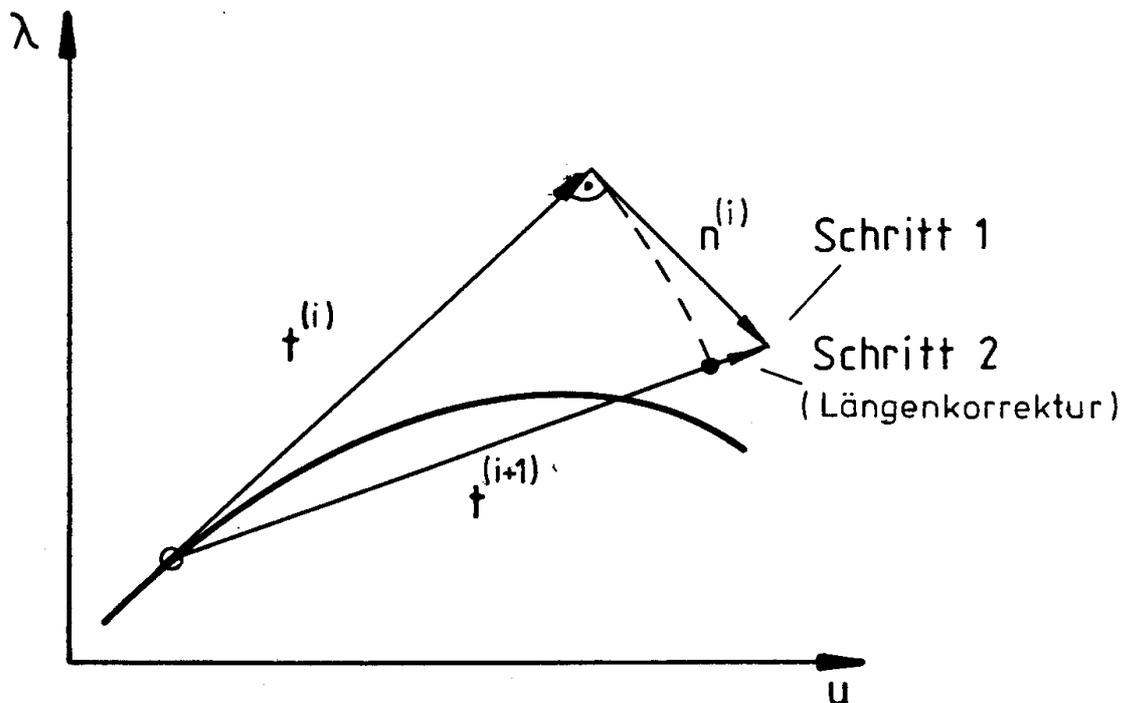
$$*f^{(2)} = \mathbf{t}^{(i)t} \cdot \mathbf{r}^{(i+1)} = \mathbf{t}^{(i)t} \cdot \mathbf{t}^{(i)} \left(1 - \frac{|\mathbf{t}^{(i)}|}{|\mathbf{t}^{(i+1)}|}\right) \quad (6.71)$$

$$|\mathbf{t}^{(i+1)}| = (\mathbf{t}^{(i)t} \mathbf{t}^{(i)} + \mathbf{n}^t \mathbf{n})^{0.5} = (\mathbf{t}^{(i)t} \mathbf{t}^{(i)} + \Delta^* \mathbf{u}^{(1)t} \Delta^* \mathbf{u}^{(1)} + \bar{\beta} \Delta^* \lambda^{(1)2})^{0.5} \quad (6.72)$$

Dies bedeutet, daß von Anbeginn an ein Residuum zur Korrektur der Verlängerung des Vektors im Schritt berücksichtigt wird.

Lastfaktor im 2. Schritt:

$$\Delta^* \lambda^{(2)} = -\frac{*f^{(2)} + \mathbf{u}^{(i)t} \cdot \Delta \mathbf{u}^{II}}{\bar{\beta} \lambda^{(i)} + \mathbf{u}^{(i)t} \Delta \mathbf{u}^I} \quad (6.73)$$



**Bild 6.9 Geometrisches Bogenlängenverfahren nach /Forde,Stiemer 1987/**

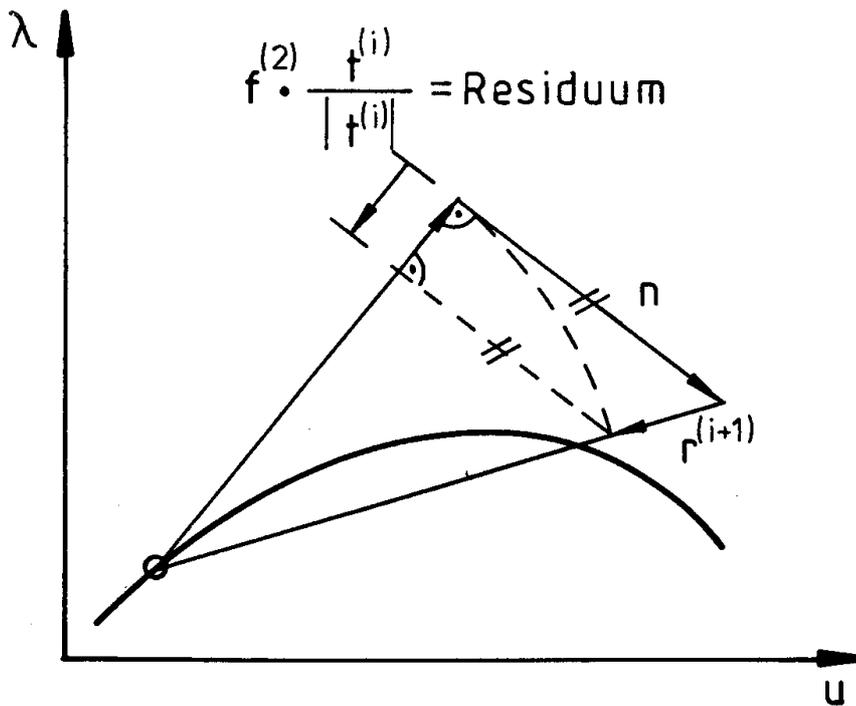
Der Lastfaktor nach beiden Iterationen lautet

$$\Delta\lambda^{(i)} = \Delta^*\lambda^{(2)} \quad (6.74)$$

Weitere Iterationen zur Erfüllung der quadratischen Gleichung finden nicht statt.

Im nächsten übergeordneten Iterationsschritt wird mit Schritt 1 fortgefahren.

*Bemerkung 6.9:* Das Vorgehen nach /Forde,Stiemer 1987/ ist nur für den eindimensionalen Fall exakt. Für alle mehrdimensionalen Fälle stellt das Vorgehen eine Näherung an die exakte Wurzel dar. Ein Übergang auf die exakte Lösung, d.h. die Erfüllung der quadratischen Gleichung, ist dann nicht mehr möglich. Es sei aber festgehalten, daß der erreichten Lastwerte üblicherweise sehr nahe an den exakten Werten liegen.



**Bild 6.10** Interpretation des Verfahrens nach /Forde, Stiemer 1987/

### 6.3.5.2 Allgemeine nichtlineare Bedingungen

Im Prinzip können Zusatzgleichungen beliebiger Ordnung verwendet werden. Es ist allerdings fraglich, inwieweit dies für die nichtlinearen Gleichungen in der Strukturmechanik von Bedeutung ist.

In /Runesson et al 1986/ wird eine allgemeine nichtlineare Zusatzgleichung mit verschiedenen Anteilen aus Verschiebungen und Lasten angegeben, die zu einer quadratischen Gleichung in  $\Delta\lambda$  führen soll. Der eigentliche Hintergrund für diese komplexe Gleichung wird nicht klar. In den Beispielen in /Runesson et al 1986/ zeigt sich auch kein besonderer Vorteil dieses Vorgehens, im Gegenteil, es werden extrem viele Iterationen benötigt.

### 6.3.6 Wertung der Verfahren

Ein wesentlicher Nachteil aller Iterationen auf Ebenen d.h. Lastkontrolle, Verschiebungskontrolle, Energiekontrolle, wie Iteration auf Normalenebenen besteht darin, daß die Kontrolle der Schrittweite nur auf einzelne Terme des Gesamtlösungsvektors  $(\mathbf{x}, \lambda)$  wirkt. Damit können diese Algorithmen nur

unvollständig adaptiv gestaltet werden. Außerdem bleibt der Lösungsraum offen und globale Konvergenz kann nicht gesichert werden, da nicht notwendigerweise Schnittpunkte der Last-Verschiebungskurven mit den Iterationsebenen existieren. In einer Erweiterung der von /Keller 1982/ für die Iteration auf Normalenebenen geprägten Bezeichnung auf alle Iterationen auf Ebenen können diese unter dem Titel "Pseudo Bogenlängen Verfahren" (pseudo arc length methods) zusammengefaßt werden.

Das eigentliche Bogenlängenverfahren nach /Crisfield 1981/ sichert, daß Schnittpunkte der Last-Verschiebungskurven mit der "kugelförmigen" Hyperfläche existieren. Durch die Fixierung der Bogenlänge im Inkrement wird bereits eine gute Schrittweitenanpassung des Lösungsalgorithmus an das nicht-lineare Verhalten des Problems erreicht. Da die quadratische Zusatzgleichung ohne großen Mehraufwand verglichen mit linearen Bedingungen lösbar ist, ist dieses Vorgehen allen anderen Verfahren überlegen.

Ob und wie die Verfahren miteinander kombiniert werden können, ist sehr stark problemabhängig. Der Übergang vom einen auf das andere Verfahren kann nur nach heuristischen Gesichtspunkten erfolgen /Bathe, Dvorkin 1983/ und wird daher in der Regel dem Anwender der dargestellten Algorithmen überlassen.

## 6.4 Skalierung - Größenangleich

Bei der einfachen Steuerung der Algorithmen über einzelne Werte, wie Lastkontrolle oder einfache Verschiebungssteuerung (Gl. 6.36) wird in allen Gleichungen mit Werten derselben Art gearbeitet, z.B. nur Lastwerten, nur Rotationen oder nur Verschiebungen. Wird hingegen eine Kontrolle nach Abschnitt 6.3.4.2 bzw. 6.3.4.3 mit mehreren Verschiebungswerten (Translationen und Rotationen) und evtl. Lastwerten vorgenommen (Gl. 6.75),

$$f = \mathbf{v}^t \cdot \mathbf{u} + \beta \cdot \lambda = 0 \quad (6.75)$$

so muß über die Faktoren im Wichtungsvektor  $\mathbf{v}$  bzw. dem Faktor  $\beta$  eine größenkonsistente Dimensionierung erreicht werden. Sonst wird der gewählte Iterationsalgorithmus dimensionsabhängig. Für diesen Größenangleich wird im Weiteren die Bezeichnung Skalierung verwendet.

Dieselbe Problematik tritt bei den Iterationen auf Normalenebenen und auf Kugelflächen auf, sofern nicht ausschließlich mit Translationen oder Rotationen gearbeitet wird. (Gl. 6.78/6.79, siehe auch Abschnitte 6.3.3.4/6.3.4.1).

Iteration auf Normalebenen.

$$f = (\mathbf{x}^{(k)} - {}^m \mathbf{x})^t (\mathbf{x} - \mathbf{x}^{(k)}) + \beta^2 \mathbf{P}^T \mathbf{P} (\lambda^{(k)} - {}^m \lambda) (\lambda - \lambda^{(k)}) = 0 \quad (6.76)$$

Iteration auf Kugelflächen

$$f = \sqrt{(\mathbf{x} - {}^m \mathbf{x})^t (\mathbf{x} - {}^m \mathbf{x}) + \beta^2 \mathbf{P}^T \mathbf{P} (\lambda - {}^m \lambda)^2} - s = 0 \quad (6.77)$$

Die Formulierung der Zusatzgleichung als Energiebedingung ( Abschnitt 6.3.4.5) führt auf konsistente Größen. Ein Größenangleich wird nur für die Kombination mit anderen Algorithmen erforderlich.

#### 6.4.1 Skalierung bei einzelnen Algorithmen

Das Problem des Größenangleichs über Wichtungsfaktoren wurde schon von /Crisfield 1981/ und /Ramm 1981/ erkannt, die in ihren Arbeiten jeweils die Anteile aus den Lastfaktoren vernachlässigen, jedoch Rotations- und Translationsanteile miteinander vermischen. /Padovan et al 1982,1984/ skalieren nur den Verschiebungsanteil (Rotationen + Translationen mit einem Faktor) und versuchen damit eine größere Adaptivität des Verfahrens zu erzeugen. /Park 1982/ schlägt eine einfache Skalierung mit zwei Faktoren  $\alpha$  und  $\beta$  vor (Gl.6.79).

$$f = \sqrt{\alpha^2 (\mathbf{x} - {}^m \mathbf{x})^t (\mathbf{x} - {}^m \mathbf{x}) + \beta^2 \mathbf{P}^T \mathbf{P} (\lambda - {}^m \lambda)^2} - s = 0 \quad (6.78)$$

Eine konsistente Skalierung wird erstmals in /Felippa 1984/ und /Simo, Wriggers, Schweizerhof, Taylor 1984/ vorgeschlagen:

Nach /Felippa 1984/ gilt:

$$f = \frac{\alpha^2}{\gamma^2} (\mathbf{x} - {}^m \mathbf{x})^t \mathbf{S} (\mathbf{x} - {}^m \mathbf{x}) + \beta^2 \mathbf{P}^T \mathbf{P} (\lambda - {}^m \lambda)^2 - s^2 = 0 \quad (6.79)$$

$\mathbf{S}$  ist eine beliebige symmetrische, nicht negative Matrix, üblicherweise eine Diagonalmatrix. Als Vorschläge werden angegeben:

$\mathbf{S} = \mathbf{I}$	keine Skalierung
$\mathbf{S} = \text{diag.}\mathbf{K}^{(k)}$	Diagonalmatrix
$\mathbf{S} = \mathbf{K}^{(i)}$	Tangentenmatrix

$\gamma$  ist ein Referenzwert mit

$$\gamma = \sqrt{(\mathbf{x}^{(j)} - {}^m \mathbf{x})^t \mathbf{S} (\mathbf{x}^{(j)} - {}^m \mathbf{x})} \quad (6.80)$$

$\mathbf{x}^{(j)}$  - Referenzzustand

In /Simo et al 1984/ werden die Elemente des Verschiebungsvektors einzeln skaliert; dies entspricht einer diagonalen Skalierungsmatrix  $\mathbf{S}$ . Ein sinnvoller Vorschlag für die Diagonalelemente von  $\mathbf{S}$  ist:

$$s_{ii} = (1/\mathbf{u}_{ii}^{(k)})^2; \quad |\mathbf{u}_{ii}^{(k)}| \neq 0 \quad (6.81)$$

D.h. die Elemente des Verschiebungsvektors in einem bekannten Zustand werden zur Skalierung verwendet. Wird noch jeder Skalierungswert mit einem Faktor multipliziert, so sind die Formulierungen in /Felippa 1984/ und /Simo et al 1984/ identisch.

*Bemerkung 6.10:* Die Skalierung mit einer Matrix  $\mathbf{S}$  und dem Faktor  $\beta$  dient in erster Linie zum Größenangleich der Last-/Translations- und Rotationswerte. In zweiter Linie kann hiermit eine Selektion der steuernden Größen getroffen werden. Werden z.B. alle Translationen bzw. Rotationen mit jeweils nur einen Faktor skaliert, so steuern die Größen mit den maximalen Absolutwerten die Algorithmen.

### Steuerung durch Größtwerte:

Skalierung der Translationen / Rotationen

z.B. durch Durchschnittswerte eines Referenzzustandes

$$|u_s| = \frac{1}{n} \sum_{k=1}^n |u_k^{(j)}| \quad n \text{ Translationen} \quad (6.82)$$

$$|\varphi_s| = \frac{1}{m} \sum_{k=1}^m |\varphi_s^{(j)}| \quad m \text{ Rotationen} \quad (6.83)$$

Werden zur Skalierung hingegen alle Werte eines Referenzzustandes gewählt, so steuern alle Größen gleichwertig.

Beide Skalierungen sind numerisch aufwendig. Bei der Skalierung mit Durchschnittswerten muß zusätzlich noch nach Translationen und Rotationen sortiert werden.

Wird mit einer fixen Skalierung d.h. ohne weitere Änderung innerhalb einer Berechnung gearbeitet, so paßt sich die Bedingungsgleichung automatisch an den Kurvenverlauf an. Wachsen die Verschiebungswerte (Translationen und Rotationen) an, so ergibt sich ein verschiebungsgesteuertes Verfahren; werden die Verschiebungswerte kleiner, so tendiert das Schema zur Lastkontrolle.

Beide Effekte können über Veränderungen des Parameters  $\beta$  auch im Inkrement verstärkt oder gemindert werden. Außerdem kann auch die Bogenlänge von Iterationsschritt zu Iterationsschritt geändert werden. Dieses Problem wird ausführlichst von /Padovan et al 1982,1984,1985/ diskutiert. Die Bogenlänge  $s$  und  $\beta$  der Parameter können zum Beispiel wiederum an die Verschiebungen oder Lastwerte oder auch an andere Größen wie Energie oder Verzerrungen gekoppelt werden.

In /Runesson et al 1986/ werden für eine nichtlineare Zusatzgleichung separate Skalierungen für verschiedene Verschiebungsanteile sowie den Lastterm benutzt. Der Vorschlag, den Elastizitätsmodul  $E$  zur Skalierung des Lastterms zu verwenden, führt auf eine den Verschiebungen adäquaten Dimension.

/Padovan,Arechaga 1982/ geben ein wesentliches Kriterium für die Korrektorschritte an, bei deren Einhaltung Konvergenz gesichert sein soll. Mit dem Skalierungsfaktor

$$\beta^2 = \frac{1}{\mu} \cdot \frac{\Delta \mathbf{u}^{I^t} \Delta \mathbf{u}^I}{\Delta \mathbf{P}^T \Delta \mathbf{P}} \quad (6.84)$$

$$\Delta \mathbf{P} = \lambda^{(1)} \cdot \mathbf{P}$$

und

$$s^2 = \frac{\Delta \mathbf{u}^{I^t} \Delta \mathbf{u}^I}{\mu} \quad (6.85)$$

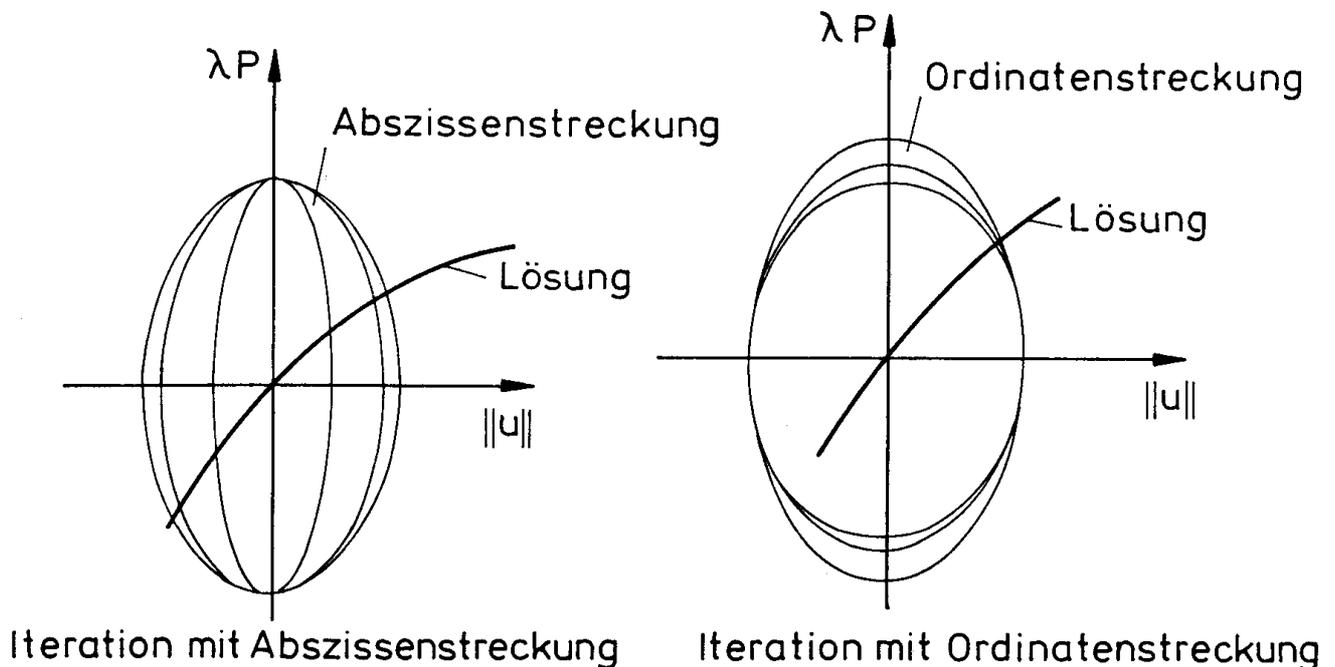
gilt:

Kriterium für den Skalierungsfaktor

$$\sqrt{\mu} \cdot \|\mathbf{u}^{(i)} - \lambda^{(i)} \cdot \Delta \mathbf{u}^I + \Delta \mathbf{u}^{II}\| \leq \|\Delta \mathbf{u}^I\| \quad (6.86)$$

$\| \quad \|$  euklidische Norm.

Dieses Kriterium sagt aus, daß die  $\sqrt{\mu}$ -fachen Verschiebungen infolge der Ungleichgewichtskräfte im Inkrement kleiner sein müssen als die linearen Verschiebungen infolge des Lastvektors  $\Delta \mathbf{P} = \lambda^{(1)} \cdot \mathbf{P}$  im Inkrement. Für große Werte  $\mu$  erhält man eine verschiebungsgesteuertes Verfahren, für kleine  $\mu$  ein lastkontrolliertes Schema. Wird  $\mu$  konstant gehalten, so wird das Verfahren über die Bogenlänge und damit über den Wert  $\lambda^{(1)}$  kontrolliert. Nimmt man an, daß die Werte eingangs so skaliert sind, daß auf einem Kreis bzw. einer Kugelfläche iteriert wird, so lassen sich Änderungen von  $\mu$  als Abszissenskalierungen deuten, während Änderungen von  $\lambda^{(1)}$  Ordinatenkalierungen darstellen.



**Bild 6.11** Skalierung nach /Padovan et al 1982, 1984/

/Padovan et al 1982,1984/ geben gute Erfahrungen bei der Anwendung dieses Kriteriums an. Der Einbau in alle bestehenden Algorithmen ist denkbar einfach.

### 6.4.2 Kombination / Selektion von Algorithmen mittels Skalierung

Mit Hilfe der Skalierungsparameter bzw. -matrizen kann neben der Skalierung auch eine Auswahl oder eine Kombination von Algorithmen erfolgen. Werden alle linearen und nichtlinearen Zusatzgleichungen analog zu /Gierlinski, Knowles 1985/ zusammengefaßt; so gilt:

Lineare Zusatzgleichungen

$$f_L = \mathbf{w}_1^t \cdot (\mathbf{x} - {}^m \mathbf{x}) + \mathbf{w}_2^t \cdot \mathbf{P} \cdot (\lambda - {}^m \lambda) + (\mathbf{x}^{(k)} - {}^m \mathbf{x})^t \mathbf{S}_1 (\mathbf{x} - \mathbf{x}^{(k)}) + \mathbf{P}^t \cdot \mathbf{S}_2 \mathbf{P} (\lambda^{(k)} - {}^m \lambda) (\lambda - \lambda^{(k)}) + c = 0 \quad (6.87)$$

Quadratische Zusatzgleichungen

$$f_Q = ({}^m \lambda + \lambda^{(i)} + \frac{1}{2} \Delta \lambda^{(i)}) \mathbf{P}^t \cdot \mathbf{S}_3 \cdot (\mathbf{x} - \mathbf{x}^{(i)}) + (\mathbf{x} - {}^m \mathbf{x})^t \cdot \mathbf{S}_4 (\mathbf{x} - {}^m \mathbf{x}) + \mathbf{P}^t \mathbf{S}_5 \mathbf{P} \cdot (\lambda - {}^m \lambda)^2 + c = 0 \quad (6.88)$$

Damit läßt sich durch geeignete Wahl der einzelnen Parameter in den Skalierungsvektoren  $\mathbf{w}_j$  und den Skalierungsmatrizen  $\mathbf{S}_j$  eine fast unbeschränkte Vielzahl von Algorithmen erzeugen.

Es ist offen, inwieweit diese Kombinationen sinnvoll sind.

## 6.5 Schrittweitensteuerung

Eine Schrittweitensteuerung erfolgt bei den einzelnen Kurvenverfolgungsalgorithmen auf unterschiedlichen Stufen. Die erste und wichtigere Steuerung geschieht durch die Vorgabe einer verschiebungs- und lastabhängigen Größe im Prädiktorschritt.

Bogenlänge

$$s_o = \sqrt{(\mathbf{x}^{(1)} - {}^m \mathbf{x})^t (\mathbf{x}^{(1)} - {}^m \mathbf{x}) + \bar{\beta} (\lambda^{(1)} - {}^m \lambda)^2} \quad (6.89)$$

## Äußere Energie

$$W_o = ({}^m\lambda + \frac{1}{2}\lambda^{(1)})\mathbf{P}^t \cdot (\mathbf{x}^{(1)} - {}^m\mathbf{x}) \quad (6.90)$$

Die beiden sinnvollsten Vorgaben sind Bogenlängenwerte oder Energiewerte. Eine Anpassung dieser Werte an das nichtlineare Verhalten kann z.B. über die Konvergenzcharakteristik erfolgen. Ein einfach verfügbares Maß ist die Anzahl der Iterationen  $I$  im aktuellen Schritt und die Anzahl der Iteration  $I_0$  in einem Referenzschritt.

$$\delta = \sqrt{\frac{I_o}{I}} \quad (6.91)$$

Wird eine Schranke  $I_{max}$  überschritten, bzw.  $I_{min}$  unterschritten so wird der Vorgabewert

$$s \text{ bzw. } W = \delta \cdot (s_0 \text{ bzw. } W_0).$$

Die numerische Erfahrung hat gezeigt, daß mit dieser einfachen Anpassung viele Probleme automatisch berechnet werden können. Es sei allerdings kritisch bemerkt, daß dies ein heuristisches Maß darstellt und mit nur sehr globalen Informationen skaliert wird.

Eine weitere einfache Möglichkeit ist die Einschränkung des Winkels zwischen aufeinanderfolgenden Lastschritten /Allgower, Georg 1980/.

$$({}^m\mathbf{x} - {}^{m-1}\mathbf{x})^t ({}^{m-1}\mathbf{x} - {}^{m-2}\mathbf{x}) + ({}^m\lambda - {}^{m-1}\lambda)({}^{m-1}\lambda - {}^{m-2}\lambda) \leq \epsilon \quad (6.92)$$

$\epsilon$  wird vom Anwender vorgegeben.

/Keller 1978/ gibt eine Formel zur Abschätzung einer maximalen Schrittweite an. Da hierbei neben einigen Funktionsauswertungen auch die Konditionszahl der Jacobimatrix (oder Tangentematrix = Hessematrix) zu berechnen ist, erscheint die Auswertung sehr aufwendig.

Im sonstigen mathematischen Schrifttum /Schwetlick 1984/ werden keine expliziten Formeln für Schrittgrößen angegeben. Es wird nur vorgeschlagen, im Sinne eines effizienten Verfahrens mit maximaler Schrittweite vorzugehen. Für eine kurze Diskussion und weitere Hinweise siehe /Schwetlick 1984/.

Die zweite Steuerung der Schrittweite kann in den Korrektorschritten erfolgen. Hierbei kann entweder der Lastparameter und/oder die Bogenlänge

geändert werden. Ein sinnvolles Maß wird hierzu von /Padovan 1985/ für quadratische Zusatzgleichungen angegeben. Siehe hierzu auch Abschnitt 6.4:

$$\begin{aligned} \sqrt{\mu} \|\mathbf{u}^{(i)} - \lambda^{(i)} \Delta \mathbf{u}^I + \Delta \mathbf{u}^{II}\| &\leq \|\Delta \mathbf{u}^I\| \\ \beta^2 &= \frac{1}{\mu} \frac{\Delta \mathbf{u}^{I^t} \Delta \mathbf{u}^I}{\Delta \mathbf{P}^T \Delta \mathbf{P}} & \Delta \mathbf{P} &= \lambda^{(1)} \cdot \mathbf{P} \\ s^2 &= \frac{\Delta \mathbf{u}^{I^t} \Delta \mathbf{u}^I}{\mu} & \Delta \mathbf{u}^I &= K^{-1} \cdot \Delta \mathbf{P} \end{aligned}$$

Bei konstantem  $\mu$  wird  $\lambda^{(1)}$  über die Bogenlänge angeglichen.

*Bemerkung 6.11:*

1. Eine Schrittweitenänderung ist im Fall linearer Zusatzgleichungen nur bei Bedingungen mit Wertvorgabe wie Verschiebungssteuerung oder Laststeuerung möglich. Als Kriterium ist z.B. die maximale Bogenlänge im Inkrement vorgegeben. Damit entstehen im Korrektorschritt Residuen:

Bogenlänge

$$s^{(i)} = \sqrt{\mathbf{u}^{(i)^t} \mathbf{u}^{(i)} + \beta^2 \lambda^{(i)^2}$$

wenn  $s^{(i)} > s_{max} \rightarrow$  Korrektur

Lastkontrolle:

$$\lambda_{soll} = \lambda^{(i)} \frac{s_{max}}{s^{(i)}} \quad (6.93)$$

$$\begin{aligned} \Delta^* \lambda^{(i)} &= \lambda^{(i)} - \lambda_{soll} \\ &= \left(1 - \frac{s_{max}}{s^{(i)}}\right) \lambda^{(i)} \end{aligned} \quad (6.94)$$

Verschiebungskontrolle:

$$(\mathbf{w}^t \cdot \mathbf{u})_{soll} = \mathbf{w}^t \mathbf{u}^{(i)} \cdot \frac{s_{max}}{s^{(i)}} \quad (6.95)$$

$$\Delta^* \lambda^{(i)} = - \frac{(\mathbf{w}^t \cdot \mathbf{u}^{(i)}) \cdot s_{max} / s^{(i)} - c}{\mathbf{w}^t \cdot \Delta \mathbf{u}^I} \quad (6.96)$$

2. Bei der Iteration auf Normalenebenen kann im Korrektorschritt direkt kein Residuum hinzugefügt werden. Dies muß über den Lastwert gesehen.

$$\Delta^* \lambda^{(i)} = \left(1 - \frac{s_{max}}{s^{(i)}}\right) \lambda^{(i)} \quad (6.97)$$

Damit

$$\Delta^* \mathbf{u}^{(i)} = \Delta^* \lambda^{(i)} \cdot \Delta \mathbf{u}^I \quad (6.98)$$

Die Konvergenz der Verfahren wird durch diese nachträgliche Bedingung verschlechtert.

### **Abschlußbemerkung:**

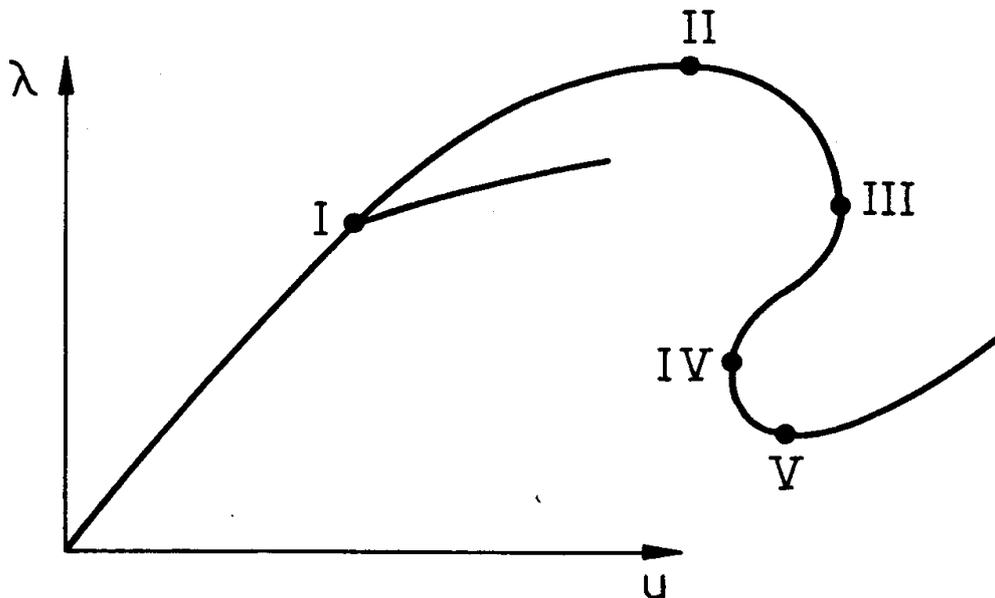
Eine automatische Schrittweitensteuerung ist der Wunsch vieler Anwender von Algorithmen. Dies kann auf die hier dargestellte Weise oder über ständige Versuche mit mehreren Schrittweiten pro Inkrement erreicht werden. Der Aufwand steigt allerdings erheblich.

Da Berechnungsingenieure in der Regel am Verhalten der Struktur unter bestimmten Belastungen interessiert sind, empfiehlt es sich, den schon vorhandenen Anpassungsmechanismus des Bogenlängenverfahrens noch innerhalb bestimmter Toleranzen automatisch weiter zu verstärken. Werden diese - problemabhängigen Toleranzen über - bzw. unterschritten, so sollte die Analyse abgebrochen und die bisher erzielten Ergebnisse genau analysiert werden. Meist weisen Konvergenzprobleme auch auf ein kritisches Systemverhalten hin. Ein Weiterarbeiten ist dann mit von Hand sinnvoll geänderten Steuerparametern sicher effektiver als ein vollautomatisches Vorgehen.

## **6.6 Begleitende Maßnahmen zur Steuerung der Algorithmen**

Zu Beginn jedes Inkrements muß entschieden werden, in welche Richtung der Prädiktorschritt erfolgt.

Ziel der Vorgabe ist, einen Rückschritt auf der Lastverschiebungskurve zu vermeiden. Das nächstliegende Maß zur Vorgabe der Richtung ist die Gradienten des Problems, über die Information in der Gestalt der Tangentenmatrix vorliegt.



**Bild 6.12 Kritische Punkte für Prädiktorschrittverfahren**

### Vorschlag 1

Wird der Parameter  $\lambda$  erhöht, so tritt bei der für eine Gleichungslösung erforderlichen  $LDL^t$ -Zerlegung /Bathe, Wilson 1976/ bei Überschreiten von Verzweigungspunkten und Durchschlagspunkten je Punkt ein negativer Diagonalterm auf.

Liegt ein Problem ohne Verzweigungspunkte vor, so kann damit folgendes Kriterium zur Veränderung des Parameters  $\lambda$  verwendet werden:

Prädiktorschritt:

$$\text{sign}(\Delta\lambda^{(*)}) = \text{sign}(\det\mathbf{K}) \quad (6.99)$$

mit  $\det\mathbf{K} = d_{11} \cdot d_{22} \cdots d_{nn}$

Treten Verzweigungspunkte auf - siehe Pkt. I in Bild 6.12, dann kann diese Steuerung versagen, sofern Verzweigungspunkte nicht als solche erkannt und berücksichtigt werden.

### Vorschlag 2

Eine alternative Steuerung ist über den Steifigkeitsparameter nach /Bergan et al 1978/ möglich.

$$S_P = \frac{\Delta\mathbf{u}^{(1)t} \cdot \mathbf{P}}{\Delta\mathbf{u}^{(n)t} \cdot \mathbf{P}} \quad (6.100)$$

mit  $\mathbf{P}$  ..... Einheitslastvektor

$\Delta \mathbf{u}^{(1)}$  Bezugsverschiebungszustand  $\Delta \mathbf{u}^{(1)} = (\mathbf{K}^{(1)})^{-1} \cdot \mathbf{P}$

$\Delta \mathbf{u}^{(n)}$  aktuelle Verschiebungen  $\Delta \mathbf{u}^{(n)} = (\mathbf{K}^{(n)})^{-1} \cdot \mathbf{P}$

Der Steifigkeitsparameter ändert dem Kurvenverlauf entsprechend sein Vorzeichen. Dies kann zur Steuerung des Prädiktorschritts verwendet werden, sofern keine Verschiebungsumkehrpunkte (Turning points) auftreten ( II und IV in Bild 6.12 ).

Prädiktorschritt:

$$\text{sign}(\Delta \lambda^{(*)}) = \text{sign}(S_p^{(*)}) \quad (6.101)$$

Verzweigungspunkte können hiermit problemlos überwunden werden. Bei Verschiebungsumkehrpunkten ändert der Steifigkeitsparameter das Vorzeichen, aber die Richtung des Prädiktorschritts muß nicht geändert werden.

### Vorschlag 3

Mit der Kombination der Beobachtung beider Parameter aus Vorschlag 1 und 2 kann ein Weg beschrieben werden, der eine meist eindeutige Festlegung des Prädiktorschritts erlaubt.

Ändern sich

a) Vorzeichen des Steifigkeitsparameters  $\text{sign}(S_p^{(i)}) \neq \text{sign}(S_p^{(i+1)})$

b) Anzahl der negativen Diagonalelemente von  $K_T$  verglichen mit einem vorliegenden Schritt

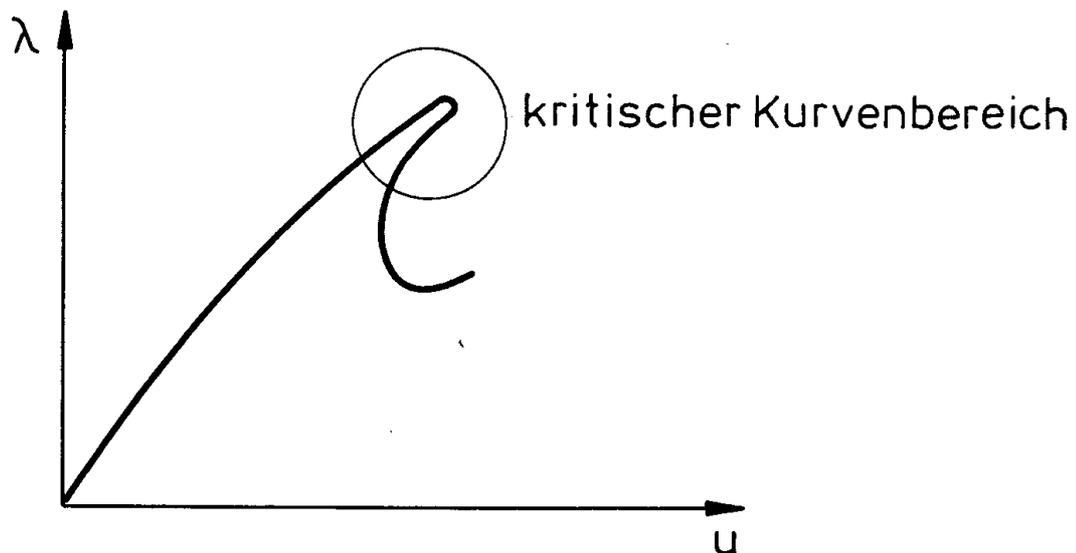
Anzahl neg.  $d_{ii}^{(i)} \triangleq$  Anzahl neg.  $d_{ii}^{(i+1)}$

so liegt ein Lastumkehrpunkt vor, d.h. das Vorzeichen des Prädiktorschritts muß geändert werden.

$$\text{sign}(\Delta \lambda^{(*)})^{(i+1)} = -\text{sign}(\Delta \lambda^{(*)})^{(i)} \quad (6.102)$$

Dieses Vorgehen versagt im seltenen Fall, wenn gleichzeitig Verschiebungsumkehrpunkt und Verzweigungspunkt zusammenfallen. Dann sollte auf eine Vorgabe "von Hand" übergegangen werden.

Werden zu große Schrittweiten bei starken Kurvenveränderungen gewählt, so kann bei Durchschlagspunkten der Vorzeichenwechsel unbeobachtet bleiben,



**Bild 6.13 Lastverschiebungskurve mit großen Kurvenänderungen in der Nähe des Durchschlagpunktes (Bsp. Zylinderschalen unter Längsbelastung)**

da dann in der Nähe des Durchschlagpunktes auch Verschiebungsumkehrpunkte auftreten können (Bild 6.13).

Mit kleinen Schrittweiten kann das Problem umgangen werden. Alternativ ist der Übergang auf eine Steuerung nach Vorschlag 1 möglich.

# 7 Strahlminimierung (Line Search) bei Kurvenverfolgungsalgorithmen

## 7.1 Klassisches Vorgehen - Konsistente Minimierung

Das prinzipielle Vorgehen folgt den Erläuterungen in Abschnitt 5.

Der wesentliche Unterschied zu den dort beschriebenen Verfahren liegt in der Änderung des Homotopie Parameters  $\lambda$  (siehe auch /Crisfield 1983/).

Die im Iterations-Schritt bestimmte Fortschrittsrichtung setzt sich aus zwei Anteilen zusammen.

$$\Delta \mathbf{u}^{(i)} = \Delta \lambda^{(i)} \Delta \mathbf{u}^I + \Delta \mathbf{u}^{II} \quad (7.1)$$

Wird die Fortschrittsrichtung des Iterationsschritts auch als Suchrichtung für die Strahlminimierung benutzt, so wird gleichzeitig angenommen, daß sich der Parameter  $\lambda$  ebenfalls linear ändert:

$$\Delta \mathbf{u}^{(i,j)} = s^{(j)} \cdot \Delta \mathbf{u}^{(i)} \quad (7.2)$$

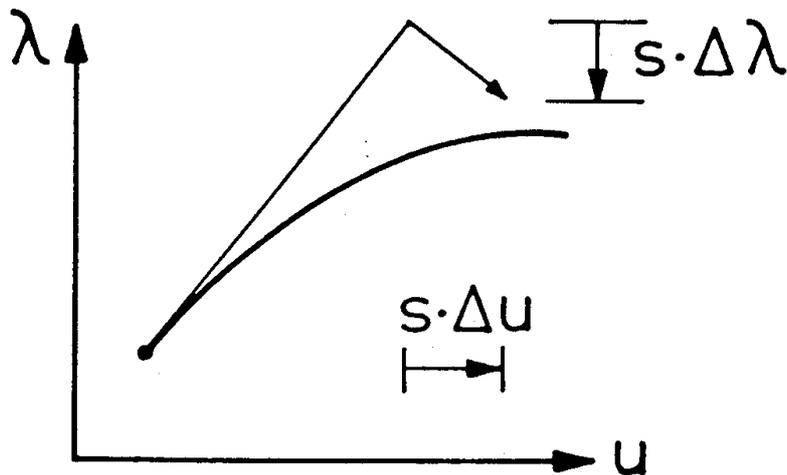
$$\Delta \lambda^{(i,j)} = s^{(j)} \cdot \Delta \lambda^{(i)} \quad (7.3)$$

Dies ist dann die beste Suchrichtung, wenn auf Ebenen bzw. Hyperebenen iteriert wird, d.h. ein linearer Zusammenhang zwischen  $\Delta \mathbf{u}^{(i,j)}$  und  $\Delta \lambda^{(i,j)}$  auch gegeben ist (Bild 7.1).

Liegt z.B. eine nichtlineare Zusatzgleichung vor, wird durch die Annahme (Gl.7.2–7.3) die Bedingung der konstanten Bogenlänge verletzt, d.h. es wird ein Residuum erzeugt. Im Sinne eines Newton-Verfahrens für das erweiterte System ist dies korrekt. Wird hingegen das Newton Verfahren für das ursprüngliche System verwendet und die Zusatzgleichung exakt erfüllt /Crisfield 1981/, siehe auch Abschnitt 6.3.5.1 Gln. (6.55 - 6.58), so muß die Suchrichtung auf andere Weise genauer bestimmt werden.

Es wird das Minimum der Energie  $\Pi(\mathbf{x}^{(i)}, \Delta \lambda^{(i,j)}, \Delta \mathbf{u}^I, \Delta \mathbf{u}^{II}, s^{(j)})$  gesucht. Hierzu ist die Nullstelle des Residuums  $\frac{\partial \pi}{\partial s}$  zu ermitteln.

$$\frac{\partial \pi}{\partial s} = \frac{\partial \pi}{\partial \mathbf{x}} \cdot \frac{\partial \mathbf{x}}{\partial s} \quad (7.4)$$



**Bild 7.1 Strahlminimierung bei linearen Zusatzgleichungen**

$$\frac{\partial \pi}{\partial \mathbf{x}} = \mathbf{r} \quad \text{Residuum des ursprünglichen Systems.}$$

$$\mathbf{x} = \mathbf{x}^{(i)} + \Delta \lambda^{(i)}(s) \Delta \mathbf{u}^I + s \Delta \mathbf{u}^{II} \quad (7.5)$$

$$\frac{\partial \mathbf{x}}{\partial s} = \frac{\partial \Delta \lambda^{(i)}}{\partial s} \cdot \Delta \mathbf{u}^I + \Delta \mathbf{u}^{II} \quad (7.6)$$

Bei linearen Zusatzgleichungen ist wegen Gl. 7.2 und 7.3

$$\Delta \lambda^{(i)}(s) = s \cdot \Delta \lambda^{(i)} \quad (7.7)$$

$$\frac{\partial \Delta \lambda^{(i)}(s)}{\partial s} = \Delta \lambda^{(i)} = - \frac{\mathbf{v}^t \cdot \Delta \mathbf{u}^{II}}{\beta \cdot \lambda^{(k)} + \mathbf{v}^{(k)t} \Delta \mathbf{u}^I} \quad (7.8)$$

Werden die Gleichungen 7.7 und 7.8 genauer betrachtet, so wird offensichtlich, daß  $\Delta \mathbf{u}^{II}$  die eigentliche Suchrichtung ist. Dies führt auf das allgemeine Konzept der korrekten Strahlminimierung bei beliebiger Form der Zusatzgleichung.

### Korrekte Strahlminimierung:

Als Suchrichtung wird  $\mathbf{d} = \Delta \mathbf{u}^{II}$  angenommen. Der Lastparameter ist im Zuge der Minimierung so anzupassen, daß die Zusatzgleichung erfüllt wird. Für die verschiedenen Zusatzgleichungen ist die korrekte Linearisierung des Lastfaktors  $\Delta \lambda$  bezüglich des Parameters  $s$  vorzunehmen.

Lineare Zusatzgleichungen:

$$\Delta\lambda^{(i)}(s) = -\frac{\mathbf{v}^{(k)t} \cdot (s\Delta\mathbf{u}^{II})}{\bar{\beta} \cdot \lambda^{(k)} + \mathbf{v}^{(k)t} \cdot \Delta\mathbf{u}^I} \quad (7.9)$$

$$\frac{\partial\Delta\lambda^{(i)}(s)}{\partial s} = -\frac{\mathbf{v}^{(k)t} \cdot \Delta\mathbf{u}^{II}}{\bar{\beta} \cdot \lambda^{(k)} + \mathbf{v}^{(k)t} \cdot \Delta\mathbf{u}^I} \quad (7.10)$$

Damit gilt für alle Iterationen der Strahlminimierung:

$$\frac{\partial\mathbf{x}}{\partial s} = -\frac{\mathbf{v}^{(i)t} \cdot \Delta\mathbf{u}^{II}}{\bar{\beta} \cdot \lambda^{(i)} + \mathbf{v}^{(i)t} \cdot \Delta\mathbf{u}^I} \cdot \Delta\mathbf{u}^I + \Delta\mathbf{u}^{II} \quad (7.11)$$

Quadratische Zusatzgleichungen:

*Konsistente Linearisierung:* /Schweizerhof, Wriggers 1986/

$$\Delta\lambda^{(i)}(s) = -\frac{f^{(i)} + s \cdot \mathbf{u}^{(i)t} \Delta\mathbf{u}^{II}}{\bar{\beta}\lambda^{(i)} + \mathbf{u}^{(i)t} \Delta\mathbf{u}^I} \quad (7.12)$$

$$\frac{\partial\Delta\lambda^{(i)}(s)}{\partial s} = -\frac{\mathbf{u}^{(i)t} \Delta\mathbf{u}^{II}}{\bar{\beta}\lambda^{(i)} + \mathbf{u}^{(i)t} \Delta\mathbf{u}^I} \quad (7.13)$$

Damit gilt für alle Iterationen der Strahlminimierung:

$$\frac{\partial\mathbf{x}}{\partial s} = -\frac{\mathbf{u}^{(i)t} \Delta\mathbf{u}^{II}}{\bar{\beta}\lambda^{(i)} + \mathbf{u}^{(i)t} \Delta\mathbf{u}^I} \Delta\mathbf{u}^I + \Delta\mathbf{u}^{II} \quad (7.14)$$

*Direkte Erfüllung der Zusatzgleichung* /Crisfield 1981/:

$$\Delta\lambda^{(i)}(s) = -\frac{b}{a} \pm \sqrt{(b/a)^2 - c/a} \quad (7.15)$$

$$a = \bar{\beta} + \Delta\mathbf{u}^{I^t} \Delta\mathbf{u}^I \quad (7.16)$$

$$b = \underbrace{\mathbf{u}^{(i)t} \Delta\mathbf{u}^I + \bar{\beta} \cdot \lambda^{(i)}}_{b_1} + s \cdot \underbrace{\Delta\mathbf{u}^{II^t} \Delta\mathbf{u}^I}_{b_2} = b_1 + s \cdot b_2 \quad (7.17)$$

$$c = s \cdot \underbrace{2 \cdot \mathbf{u}^{(i)t} \Delta\mathbf{u}^{II}}_{c_1} + s^2 \underbrace{\Delta\mathbf{u}^{II^t} \Delta\mathbf{u}^{II}}_{c_2} = s \cdot c_1 + s^2 \cdot c_2 \quad (7.18)$$

$$\frac{\partial \Delta \lambda^{(i)}(s)}{\partial s} = - \underbrace{\frac{b_2}{a}}_{e_1} \pm \underbrace{\frac{2(b_1 + s \cdot b_2) \cdot b_2 - a(c_1 + 2s \cdot c_2)}{2a \cdot \sqrt{(b_1 + s \cdot b_2)^2 - a(sc_1 + s^2 \cdot c_2)}}}_{e_2} = -e_1 \pm e_2(s) \quad (7.19)$$

Damit gilt für die Iteration  $j$  der Strahlminimierung:

$$\frac{\partial \mathbf{x}^{(i,j)}}{\partial s} = [-e_1 \pm e_2(s^{(j-1)})] \Delta \mathbf{u}^I + \Delta \mathbf{u}^{II} \quad (7.20)$$

Für den Startwert  $s^{(0)} = 0$  wird:

$$\frac{\partial \mathbf{x}^{(i,0)}}{\partial s} = - \frac{\mathbf{u}^{(i)t} \Delta \mathbf{u}^{II}}{\beta \cdot \lambda^{(i)} + \mathbf{u}^{(i)t} \Delta \mathbf{u}^I} \cdot \Delta \mathbf{u}^I + \Delta \mathbf{u}^{II} \quad (7.21)$$

Dies entspricht der Suchrichtung der konsistenten Linearisierung. Das Vorzeichen von  $e_2$  ist analog zu dem entsprechenden Vorzeichen des "Wurzel" terms in  $\Delta \lambda^{(i)}$  festzulegen. Während der Strahlminimierung sollte es nicht verändert werden.

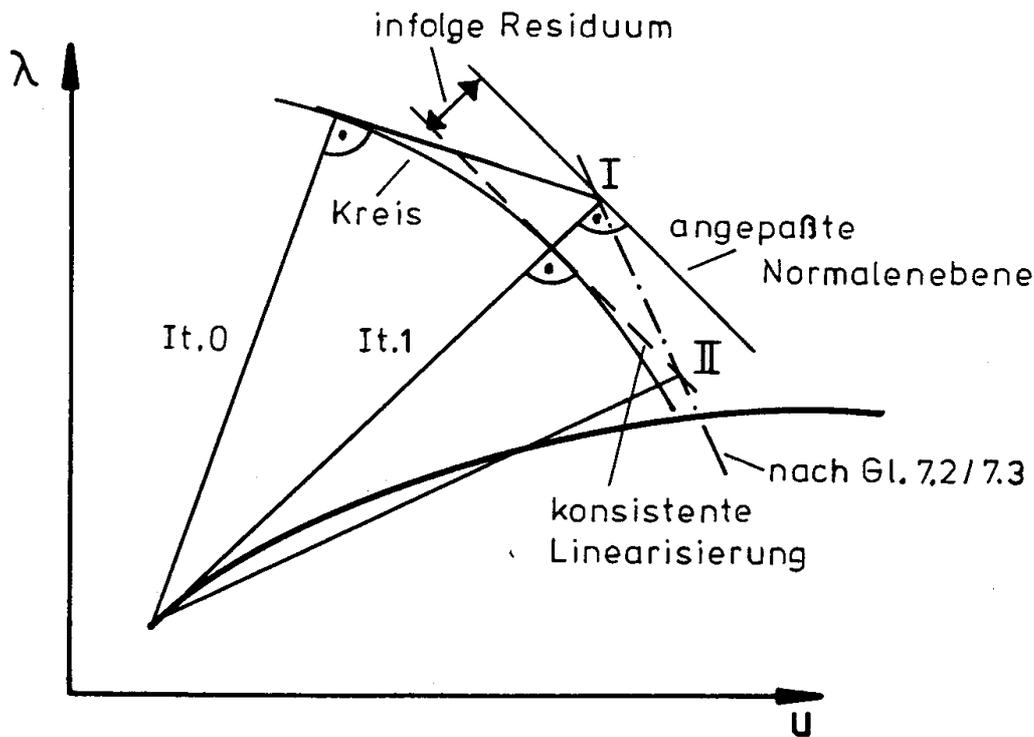
Wie oben angeführt, führen bei linearen Zusatzgleichungen die Suchrichtungen  $\Delta \mathbf{u}^{II}$  bzw.  $\Delta \mathbf{u}^{(i)}$  zum gleichen Ergebnis.

Bei quadratischen Zusatzgleichungen wird im Fall konsistenter Linearisierung wie bei linearen Zusatzgleichungen (angepaßte Normalenebenen) parallel zur Hyperebene mit der Normalen  $\mathbf{u}^{(i)}$  gesucht. Der Korrekturterm infolge des Residuums bedeutet die Versetzung der Hyperebene an den Kreis gegenüber der linearen Zusatzgleichung.

Wird die Strahlminimierung nach Gl. 7.2 bzw. 7.3 durchgeführt, so wird der Korrekturterm nicht berücksichtigt, und es wird auf einer Ebene durch die Punkte I ( $s = 0$ ) und II ( $s = 1$ ) in Bild 7.2 iteriert.

Bei der Suche mit direkter Erfüllung der quadratischen Zusatzgleichung wird in jedem Suchschritt auf der Kugelhyperebene iteriert. Mit Gl. 7.19 wird auch das zugehörige Residuum korrekt bestimmt.

Der Aufwand gegenüber einer inkonsistenten Suche nach Gl. 7.2/7.3 ist minimal größer, da die Werte  $a, b, c$  bzw.  $b_1, b_2, c_1$  und  $c_2$  bereits aus der ersten Berechnung von  $\Delta \lambda^{(i)}$  vorliegen. Der einzige Unterschied liegt im erforderlichen Speicherplatz, da für die Berechnung von  $\Delta \mathbf{u}^{(i,j)}$  jeweils noch  $\Delta \mathbf{u}^I$  und  $\Delta \mathbf{u}^{II}$  bereitgestellt werden müssen.



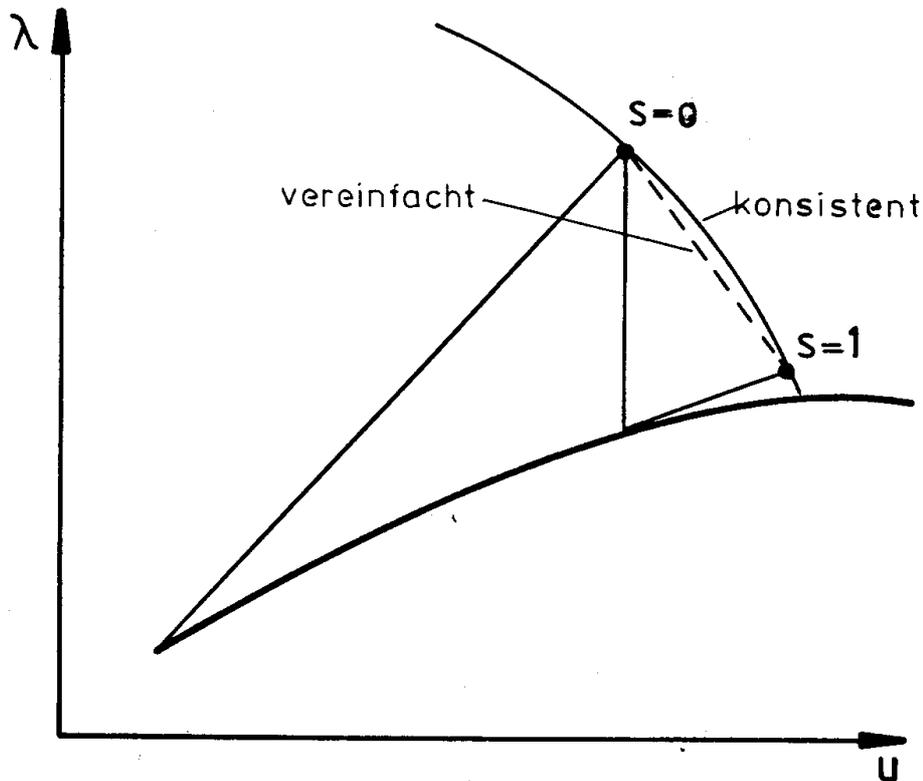
**Bild 7.2 Strahlminimierung bei angepassten Normalenebenen und bei konsistenter Linearisierung der quadratischen Zusatzgleichung (eindimensionaler Fall)**

Der Unterschied zwischen einer konsistenten und einer vereinfachten Strahlminimierung bei quadratischer Zusatzgleichung kann in Bild 7.3 am eindimensionalen Fall leicht dargestellt werden.

Bei der vereinfachten Vorgehensweise wird auf der Sekantenebene iteriert, wobei nach der Strahlminimierung die Zusatzgleichung nicht mehr exakt erfüllt wird.

Im Zuge einer nichtlinearen Berechnung ist dies aber kein grundsätzlicher Nachteil, da auch damit das eigentliche Ziel - die Berechnung einer Gleichgewichtslage - erreicht wird. Vorausgesetzt wird als Suchraum  $0 \leq s \leq 1$ .

*Bemerkung 7.1:* /Crisfield 1983/ schlägt eine Strahlminimierung für quadratische Zusatzgleichungen vor, die auf einer iterativen Erfüllung der Zusatzgleichung beruht. Hierbei wird auf Sekantenebenen gesucht und dann die Bogenlänge korrigiert. Das Verfahren kann nicht mehr einfach geometrisch gedeutet werden. Die hier vorgestellte konsistente Vorgehensweise erscheint wesentlich eindeutiger.



**Bild 7.3 Konsistente und vereinfachte Strahlminimierung bei nicht-linearen Zusatzgleichungen**

**Algorithmus:** Strahlminimierung bei nicht ausschließlich lastkontrollierten Verfahren:

Ungleichgewichtskräfte

$$\mathbf{r}^{(i,j)} = \mathbf{R}^{(i,j)}(\lambda^{(i)} \mathbf{x}^{(i)}, \Delta \mathbf{u}^{(i)}(s^{(j)})) - \mathbf{F}^{(i,j)}(\mathbf{x}^{(i)}, \Delta \mathbf{u}^{(i)}(s^{(j)})) \quad (7.22)$$

Für nicht verformungsabhängige Belastung wird dies zu:

$$\mathbf{r}^{(i,j)} = \lambda^{(i,j)}(s^{(j)}) \cdot \mathbf{P} - \mathbf{F}^{(i,j)}(\mathbf{x}^{(i)}, \Delta \mathbf{u}^{(i)}(s^{(j)})) \quad (7.23)$$

Suchrichtung:

$$\mathbf{d}^{(i)} = \frac{\partial \Delta \lambda^{(i,j)}(s^{(j)})}{\partial s} \cdot \Delta \mathbf{u}^I + \Delta \mathbf{u}^{II} \quad (7.24)$$

Es ist  $\mathbf{d}^{(i)} = \mathbf{d}^{(0)}$  für lineare Zusatzgleichungen und konsistent linearisierte Zusatzgleichungen.

Im Folgenden werden nur die vom Standard Line – Search Algorithmus abweichenden Schritte aufgeführt. Selbstverständlich kann jede Art von Skalierung nach Abschnitt 6.4 mit dem dargestellten Algorithmus verbunden werden.

### Ablauf

Schritt 1.

$$E_0 = \mathbf{r}^{(i,0)^t} \cdot \mathbf{d}^{(o)}$$

$$E_1 = \mathbf{r}^{(i,1)^t} (s^{(1)} = 1) \cdot \mathbf{d}^{(1)} (s^{(1)} = 1)$$

$$a = E_0$$

$$b = E_1$$

Schritte 2. - 3. identisch wie in Abschnitt 5

Schritt 4. Neues Residuum

$$\lambda^{(i,j)} = \lambda^{(i)}(s^j) \quad \text{Neuer Lastwert}$$

$$\Delta \mathbf{u}^{(i,j)} = \Delta \lambda^{(i,j)} \cdot \Delta \mathbf{u}^I + \Delta \mathbf{u}^{II}$$

$$\mathbf{r}^{(i,j)} = \lambda^{(i,j)} \cdot P - \mathbf{F}^{(i,j)}(\mathbf{x}^{(i)}, \Delta \mathbf{u}^{(i,j)})$$

Neue Suchrichtung

$$\mathbf{d}^{(j)} = \frac{\partial \Delta \lambda^{(i,j)}(s^j)}{\partial s} \cdot \Delta \mathbf{u}^I + \Delta \mathbf{u}^{II}$$

Schritte 5. - 8. wie in Abschnitt 5.

## 7.2 Numerische Relaxation

/Ramm 1981/ schlägt im Fall oszillierender Konvergenz für den Lastfaktor  $\Delta \lambda^{(i)}$  eine einfache numerische Dämpfung vor:

Kriterien:

$$\text{a) } \Delta \lambda^{(i)} \cdot \Delta \lambda^{(i-1)} < 0$$

$$\text{b) } |\Delta \lambda^{(i)}| \leq |\Delta \lambda^{(i-1)}|$$

Werden beide Kriterien erfüllt, dann werden der Lastfaktor und der Verschiebungszuwachs mit einem Faktor  $\alpha$  skaliert.

$$\Delta \lambda^{(i,j)} = \alpha \cdot \Delta \lambda^{(i,j)} \tag{7.25}$$

$$\Delta \mathbf{u}^{(i,j)} = \alpha \cdot \Delta \mathbf{u}^{(i,1)} = \alpha (\Delta \lambda^{(i,j)} \cdot \Delta \mathbf{u}^I + \Delta \mathbf{u}^{II}) \tag{7.26}$$

Üblicherweise wird  $\alpha = 0.5$  gesetzt.

Dies kann als Strahlminimierung mit einem festen Wert von  $s = \alpha$  gedeutet werden. Das dargestellte Verfahren könnte durch lineare Interpolation zwischen den beiden Werten  $\Delta\lambda^{(i)}$  und  $\Delta\lambda^{(i-1)}$  noch verfeinert werden.

z.B.

$$\alpha = \frac{\Delta\lambda^{(i-1)}}{\Delta\lambda^{(i-1)} - \Delta\lambda^{(i)}} \quad (7.27)$$

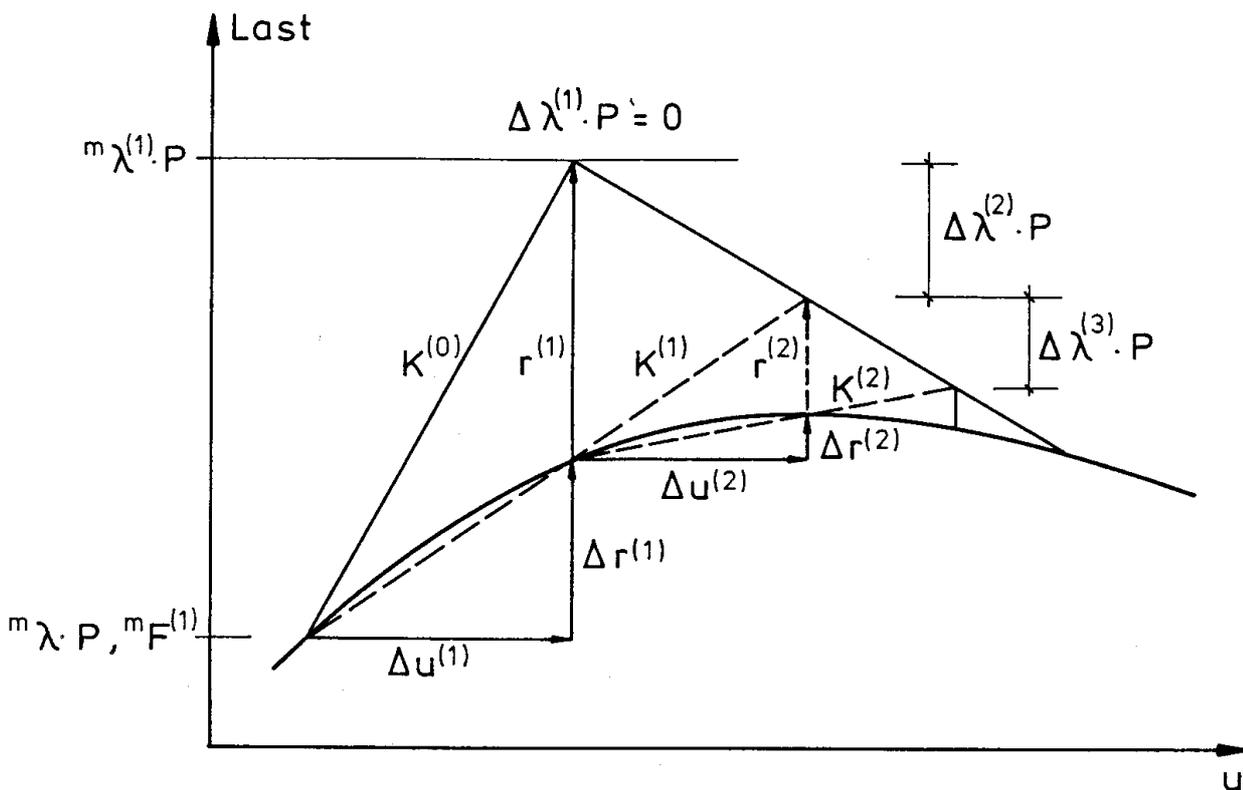
Dies wurde im Rahmen dieser Arbeit jedoch nicht in ein Rechenprogramm implementiert.

*Bemerkung 7.2:* Dieses einfache und offensichtlich wenig aufwendige Verfahren verbessert die Konvergenzrate in vielen Fällen beträchtlich. Allerdings fehlen dem Vorgehen die globalen Konvergenzeigenschaften der klassischen Strahlminimierungstechniken. D.h. es kann zur Konvergenzbeschleunigung eingesetzt werden, aber nicht zur Konvergenzsicherung.

# 8 Kombination der Quasi-Newton Verfahren mit Kurvenverfolgungsalgorithmen

## 8.1 Algorithmen

Zur Kombination mit Kurvenverfolgungsalgorithmen müssen die Quasi-Newton Verfahren modifiziert werden, um die Änderungen im Parameter  $\lambda$  - dem Lastfaktor - zu berücksichtigen (Bild 8.1)



**Bild 8.1** Kombination Quasi - Newton Verfahren - Bogenlängenverfahren (Iteration auf Normalenebenen)

Die modifizierte Quasi - Newton Gleichung lautet:

$$\begin{aligned} \mathbf{K}^{(k)} \cdot \Delta \mathbf{u}^{(k)} &= \lambda^{(k-1)} \cdot \mathbf{P} - \mathbf{F}^{(k-1)} - [\lambda^{(k)} \cdot \mathbf{P} - \mathbf{F}^{(k)}] + \Delta \lambda^{(k)} \cdot \mathbf{P} \\ &= \mathbf{r}^{(k-1)} - \mathbf{r}^{(k)} + \Delta \lambda^{(k)} \cdot \mathbf{P} \end{aligned} \quad (8.1)$$

$$\Delta \mathbf{r}^{(k)} = \mathbf{r}^{(k-1)} - (\mathbf{r}^{(k)} - \Delta \lambda^{(k)} \cdot \mathbf{P}) \quad (8.2)$$

*Bemerkung 8.1:* Das Vorzeichen von  $\Delta \lambda^k$  ist per Definition negativ, d.h. in Gleichung (8.1) erscheint der Term  $\Delta \lambda^{(k)} \cdot \mathbf{P}$  mit positivem Vorzeichen.

Als Folge dieser Modifikation wird es zur Bestimmung von  $\Delta\lambda^{(i)}$  außerdem notwendig, in jedem Schritt die Lösung des Gleichungssystems für zwei rechte Seiten vorzunehmen.

Wird auch noch eine einfache Strahlminimierung nach den Gln. (7.2/7.3) mit der angesprochenen Modifikation verknüpft, so gilt für die verschiedenen Algorithmen:

### 8.1.1 Rang – Eins Aufdatierungen

#### BROYDEN mit Kurvenverfolgungsalgorithmen

$$\begin{aligned} \mathbf{K}^{(k-1)^{-1}} \Delta \mathbf{r}^{(k)} &= \underbrace{\mathbf{K}^{(k-1)^{-1}} \cdot \mathbf{r}^{(k-1)} + \Delta\lambda^{(k)} \cdot \mathbf{K}^{(k-1)^{-1}} \cdot \mathbf{P}} - \mathbf{K}^{(k-1)^{-1}} \cdot \mathbf{r}^{(k)} \\ &= \Delta \mathbf{u}^{(k)} / s + \frac{(s-1)}{s} \Delta\lambda^{(k)} \mathbf{K}^{(k-1)^{-1}} \mathbf{P} - \mathbf{K}^{(k-1)^{-1}} \mathbf{r}^{(k)} \quad (8.3) \end{aligned}$$

Die Inverse der aufdatierten Matrix lautet (siehe Abschnitt 4.2)

$$\begin{aligned} \mathbf{K}^{(k)^{-1}} &= (\mathbf{I} + \alpha^{(k)} \cdot \mathbf{w}^{(k)} \cdot \Delta \mathbf{u}^{(k)t}) \left( \mathbf{I} + \alpha^{(k-1)} \cdot \mathbf{w}^{(k-1)} \cdot \Delta \mathbf{u}^{(k-1)t} \right) \dots \\ &\dots \left( \mathbf{I} + \alpha^{(1)} \cdot \mathbf{w}^{(1)} \cdot \Delta \mathbf{u}^{(1)t} \right) \mathbf{K}^{(0)^{-1}} \quad (8.4) \end{aligned}$$

Mit den Aufdatierungsvektoren  $\Delta \mathbf{u}^{(i)}$ ,  $\mathbf{w}^{(i)}$ :

$$\begin{aligned} \mathbf{w}^{(i)} &= \Delta \mathbf{u}^{(i)} - [\mathbf{K}^{(i-1)^{-1}} \cdot \mathbf{r}^{(i-1)} + \Delta\lambda^{(i)} \cdot \mathbf{K}^{(i-1)^{-1}} \mathbf{P}] + \mathbf{K}^{(i-1)^{-1}} \cdot \mathbf{r}^{(i)} \\ &= \Delta \mathbf{u}^{(i)} - \Delta \mathbf{u}^{(i)} / s - \frac{(s-1)}{s} \Delta\lambda^{(i)} \mathbf{K}^{(i-1)^{-1}} \mathbf{P} + \mathbf{K}^{(i-1)^{-1}} \mathbf{r}^{(i)} \quad (8.5) \end{aligned}$$

Der neue Term mit  $\Delta\lambda^{(i)}$  ist schon im Anteil  $1/s \cdot \Delta \mathbf{u}^{(i)}$  enthalten. Damit bleibt der eigentliche Lösungsalgorithmus unverändert.

Für die zweifache Lösung des Gleichungssystems bietet sich mit der vektorisierten Form eine besonders effiziente Vorgehensweise an:

*Algorithmus in der k-ten Iteration*

1. Lösung für die aktuelle Ungleichgewichtskraft  $\mathbf{r}^{(k)}$

$$\begin{aligned} \Delta \mathbf{u}^{(k+1)II} &= \underbrace{(\mathbf{I} + \alpha^{(k)} \mathbf{w}^{(k)} \Delta \mathbf{u}^{(k)t}) \dots (\mathbf{I} + \alpha^{(2)} \mathbf{w}^{(2)} \Delta \mathbf{u}^{(2)t}) (\mathbf{I} + \alpha^{(1)} \mathbf{w}^{(1)} \Delta \mathbf{u}^{(1)t})}_{\mathbf{K}^{(k)^{-1}}} \cdot \mathbf{K}^{(0)^{-1}} \mathbf{r}^{(k)} \quad (8.6) \end{aligned}$$

Die berechneten Vektoren  $\mathbf{w}^{(k)}$  und die Werte  $\alpha^{(k)}$  werden zusammen mit  $\Delta \mathbf{u}^{(k)}$  gesondert gespeichert.

## 2. Lösung für den Lastvektor $\mathbf{P}$

$$\Delta \mathbf{u}^{(k+1)I} = (\mathbf{I} + \alpha^{(k)} \cdot \mathbf{w}^{(k)} \cdot \Delta \mathbf{u}^{(k)t}) \dots \quad (8.7)$$

$$\underbrace{\dots (\mathbf{I} + \alpha^{(2)} \mathbf{w}^{(2)} \Delta \mathbf{u}^{(2)t}) (\mathbf{I} + \alpha^{(1)} \mathbf{w}^{(1)} \Delta \mathbf{u}^{(1)t})}_{\Delta \mathbf{u}^{(k)I}} \overbrace{\mathbf{K}^{(0)-1} \cdot \mathbf{P}}^{\Delta \mathbf{u}^{(1)}}$$

Alle Vektoren inklusive  $\Delta \mathbf{u}^{(1)}$  sind gesondert gespeichert. Damit kann  $\Delta \mathbf{u}^{(k+1)I}$  ( $\equiv \Delta \mathbf{u}^I$  in der  $k$ -ten Iteration) durch  $k$  skalare Produkte mit  $k$  Additionen von Vektoren berechnet werden. Wird hingegen  $\Delta \mathbf{u}^{(k)I}$  gespeichert, so ist nur ein skalares Produkt und eine Addition von Vektoren erforderlich.

Das schrittweise Vorgehen in der  $k$ -ten Iteration ist:

Schritte 1-3 wie beim Standardvorgehen

Schritt 4 wie beim Standardvorgehen

Schritt 5-6 wie beim Standardvorgehen

Ergebnis:  $\Delta \mathbf{u}^{(k+1)II}$

Schritt 7 Hole  $\Delta \mathbf{u}^{(k)I}$  aus dem Speicher

Schritt 8  $\beta = \Delta \mathbf{u}^{(k)t} \cdot \Delta \mathbf{u}^{(k)I}$   
 $\Delta \mathbf{u}^{(k+1)I} = \Delta \mathbf{u}^{(k)I} + \beta \cdot \alpha^{(k)} \cdot \mathbf{w}^{(k)}$

Schritt 9 Speichern von  $\Delta \mathbf{u}^{(k+1)I}$

*Bemerkung 8.2:* Der dargestellte Algorithmus gilt unverändert für alle bekannten Bogenlängenverfahren.

Die entsprechenden Modifikationen für die anderen Quasi-Newton Verfahren sind:

### DAVIDON mit Kurvenverfolgungsalgorithmen

Inverse der aufdatierten Matrix

$$\mathbf{K}^{(k)-1} = \mathbf{K}^{(0)-1} + \sum_{i=1}^k \alpha^{(i)} \cdot \mathbf{w}^{(i)} \cdot \mathbf{w}^{(i)t} \quad (8.8)$$

Der Aufdatierungsvektor wird verändert:

$$\begin{aligned}\mathbf{w}^{(i)} &= \Delta \mathbf{u}^{(i)} - \mathbf{K}^{(i-1)^{-1}}(\mathbf{r}^{(i-1)} + \Delta \lambda^{(i)} \cdot \mathbf{P}) + \mathbf{K}^{(i-1)^{-1}} \cdot \mathbf{r}^{(i)} \\ &= \Delta \mathbf{u}^{(i)} - \Delta \mathbf{u}^{(i)}/s - \frac{(s-1)}{s} \Delta \lambda^{(i)} \mathbf{K}^{(i-1)^{-1}} \mathbf{P} + \mathbf{K}^{(i-1)^{-1}} \mathbf{r}^{(i)}\end{aligned}\quad (8.9)$$

Neuer Aufdatierungsfaktor:

$$\alpha^{(i)} = 1/[\mathbf{w}^{(i)t}(\mathbf{r}^{(i-1)} - \mathbf{r}^{(i)} + \Delta \lambda^{(i)} \cdot \mathbf{P})] \quad (8.10)$$

Im Algorithmus erfordert die Änderung des Aufdatierungsvektors bei einfacher Strahlminimierung keine zusätzlichen Operationen. Die Änderung des Aufdatierungsfaktors hingegen ist zu beachten.

Die doppelte Lösung des Gleichungssystems im k-ten Iterationsschritt kann wieder vereinfacht werden:

1. Lösung für die aktuelle Ungleichgewichtskraft  $\mathbf{r}^{(k)}$

$$\Delta \mathbf{u}^{(k+1)I} = \mathbf{K}^{(0)^{-1}} \cdot \mathbf{r}^{(k)} + \sum_{i=1}^k \alpha^{(i)} \mathbf{w}^{(i)} (\mathbf{w}^{(i)t} \cdot \mathbf{r}^{(k)}) \quad (8.11)$$

Der berechnete Vektor  $\mathbf{w}^{(k)}$  und der Faktor  $\alpha^{(k)}$  werden gesondert gespeichert.

2. Lösung für den Lastvektor  $\mathbf{P}$

$$\Delta \mathbf{u}^{(k+1)I} = \underbrace{\overbrace{\mathbf{K}^{(0)^{-1}} \cdot \mathbf{P}}^{\Delta \mathbf{u}^{(1)}} + \underbrace{\sum_{i=1}^{k-1} [\alpha^{(i)} \cdot \mathbf{w}^{(i)} \cdot \mathbf{w}^{(i)t} \cdot \mathbf{P}]}_{\Delta \mathbf{u}^{(k)I}} + \alpha^{(k)} \cdot \mathbf{w}^{(k)} \cdot \mathbf{w}^{(k)t} \cdot \mathbf{P}}_{\Delta \mathbf{u}^{(k+1)I}} \quad (8.12)$$

Da nur die Vektoren  $\mathbf{w}^{(i)}$  und die Faktoren  $\alpha^{(i)}$  gespeichert werden, muß, um eine weitere Gleichungslösung (Vorwärts - Rückwärtseinsetzen) zu vermeiden, entweder  $\Delta \mathbf{u}^{(1)}$  oder  $\Delta \mathbf{u}^{(k)I}$  gespeichert werden. Wird  $\Delta \mathbf{u}^{(1)}$  gespeichert, so sind zur Berechnung von  $\Delta \mathbf{u}^{(k+1)I}$  k-Skalarprodukte und k-Vektoradditionen durchzuführen. Wird hingegen  $\Delta \mathbf{u}^{(k)I}$  gespeichert, reduziert sich dies auf ein Skalarprodukt und eine Vektoraddition.

Das schrittweise Vorgehen in der k-ten Iteration ist:

Schritte 1-3 wie beim Standardvorgehen

$$\text{Schritt 4} \quad \alpha^{(k)} = \frac{1}{\mathbf{w}^{(k)t}(\mathbf{r}^{(k-1)} - \mathbf{r}^{(k)} + \Delta\lambda^{(k)} \cdot \mathbf{P})}$$

Schritte 5-6 Wie beim Standardvorgehen

Ergebnis  $\Delta\mathbf{u}^{(k+1)II}$

Schritt 7 Hole  $\Delta\mathbf{u}^{(k)I}$  aus dem Speicher

$$\text{Schritt 8} \quad \beta = \mathbf{w}^{(k)t} \cdot \mathbf{P}$$

$$\Delta\mathbf{u}^{(k+1)I} = \Delta\mathbf{u}^{(k)I} + \alpha^{(k)} \cdot \beta \cdot \mathbf{w}^{(k)}$$

Schritt 9 Speichern von  $\Delta\mathbf{u}^{(k+1)I}$

### 8.1.2 Rang – Zwei Aufdatierungen

#### DFP - mit Kurvenverfolgungsalgorithmen

Inverse der aufdatierten Matrix

$$\mathbf{K}^{(k)-1} = \mathbf{K}^{(0)-1} + \sum_{i=1}^k \alpha^{(i)} \cdot \Delta\mathbf{u}^{(i)} \Delta\mathbf{u}^{(i)t} - \sum_{i=1}^k \beta^{(i)} \mathbf{w}^{(i)} \mathbf{w}^{(i)t} \quad (8.13)$$

Die Aufdatierungsvektoren  $\mathbf{w}^{(i)}$  werden verändert:

$$\begin{aligned} \mathbf{w}^{(i)} &= \mathbf{K}^{(i-1)-1} (\mathbf{r}^{(i-1)} + \Delta\lambda^{(i)} \cdot \mathbf{P}) - \mathbf{K}^{(i-1)-1} \cdot \mathbf{r}^{(i)} \\ &= \Delta\mathbf{u}^{(i)}/s + \frac{(s-1)}{s} \cdot \Delta\lambda^{(i)} \cdot \mathbf{K}^{(i-1)-1} \cdot \mathbf{P} - \mathbf{K}^{(i-1)-1} \cdot \mathbf{r}^{(i)} \end{aligned} \quad (8.14)$$

Neue Aufdatierungsfaktoren:

$$\begin{aligned} \alpha^{(i)} &= \frac{1}{\Delta\mathbf{u}^{(i)t}(\mathbf{r}^{(i-1)} - \mathbf{r}^{(i)} + \Delta\lambda^{(i)} \cdot \mathbf{P})} \\ \beta^{(i)} &= \frac{1}{\mathbf{w}^{(i)t}(\mathbf{r}^{(i-1)} - \mathbf{r}^{(i)} + \Delta\lambda^{(i)} \cdot \mathbf{P})} \end{aligned} \quad (8.15)$$

Wie bei den beiden anderen Quasi-Newton Algorithmen erfordert die Änderung des Aufdatierungsvektors  $\mathbf{w}^{(i)}$  bei einfacher Strahlminimierung keine zusätzlichen Operationen. Nur die Änderung der Aufdatierungsfaktoren ist zu beachten.

Die doppelte Lösung des Gleichungssystems im k-ten Iterationsschritt lautet:

1. Lösung für die aktuelle Ungleichgewichtskraft  $\mathbf{r}^{(k)}$

$$\Delta \mathbf{u}^{(k+1)II} = \mathbf{K}^{(0)-1} \cdot \mathbf{r}^{(k)} + \sum_{i=1}^k \alpha^{(i)} \Delta \mathbf{u}^{(i)} \Delta \mathbf{u}^{(i)t} \cdot \mathbf{r}^{(k)} - \sum_{i=1}^k \beta^{(i)} \mathbf{w}^{(i)} \mathbf{w}^{(i)t} \cdot \mathbf{r}^{(k)} \quad (8.16)$$

Der Vektor  $\mathbf{w}^{(k)}$  und die Faktoren  $\alpha^{(k)}$ ,  $\beta^{(k)}$  werden im Zuge dieses Schrittes berechnet und zusammen mit  $\Delta \mathbf{u}^{(k)}$  gesondert gespeichert.

2. Lösung für den Lastvektor  $\mathbf{P}$

$$\Delta \mathbf{u}^{(k+1)I} = \underbrace{\mathbf{K}^{(0)-1} \cdot \mathbf{P} + \sum_{i=1}^{k-1} \alpha^{(i)} \cdot \Delta \mathbf{u}^{(i)} \cdot \Delta \mathbf{u}^{(i)t} \cdot \mathbf{P} - \sum_{i=1}^{k-1} \beta^{(i)} \mathbf{w}^{(i)} \mathbf{w}^{(i)t} \cdot \mathbf{P}}_{\Delta \mathbf{u}^{(k)I}} + \alpha^{(k)} \cdot \Delta \mathbf{u}^{(k)} \cdot \Delta \mathbf{u}^{(k)t} \cdot \mathbf{P} - \beta^{(k)} \mathbf{w}^{(k)} \mathbf{w}^{(k)t} \cdot \mathbf{P} \quad (8.17)$$

Auch hier empfiehlt es sich den Vektor  $\Delta \mathbf{u}^{(k)I}$  zu speichern, dann sind für die zweite Lösung nur zwei Skalarprodukte und zwei Vektoradditionen erforderlich.

Das schrittweise Vorgehen in der k - ten Iteration ist:

Schritt 1-3. wie beim Standardvorgehen

Schritt 4. Aufdatierungsfaktoren

$$\alpha^{(k)} = \frac{1}{\Delta \mathbf{u}^{(k)t} (\mathbf{r}^{(k-1)} - \mathbf{r}^{(k)} + \Delta \lambda^{(k)} \cdot \mathbf{P})}$$

$$\beta^{(k)} = \frac{1}{\mathbf{w}^{(k-1)t} (\mathbf{r}^{(k-1)} - \mathbf{r}^{(k)} + \Delta \lambda^{(k)} \cdot \mathbf{P})}$$

Schritt 5-6 wie beim Standardvorgehen  
Ergebnis  $\Delta \mathbf{u}^{(k+1)II}$

Schritt 7. Hole  $\Delta \mathbf{u}^{(k)I}$  aus dem Speicher

Schritt 8.  $\gamma = \Delta \mathbf{u}^{(k)t} \cdot \mathbf{P}$        $\delta = \mathbf{w}^{(k)t} \cdot \mathbf{P}$

$$\Delta \mathbf{u}^{(k+1)T} = \Delta \mathbf{u}^{(k)T} + \alpha^{(k)} \cdot \gamma \cdot \Delta \mathbf{u}^{(k)} - \beta^{(k)} \cdot \delta \cdot \mathbf{w}^{(k)}$$

Schritt 9. Speichern von  $\Delta \mathbf{u}^{(k+1)T}$

### BFGS - mit Kurvenverfolgungsalgorithmen

Die effizienteste Form des BFGS-Verfahrens in Kombination mit Kurvenverfolgungsalgorithmen ist die modifizierte BFGS Aufdatierung nach Gl. ( 4.26 ).

Inverse der aufdatierten Matrix

$$\mathbf{K}^{(k)-1} = \mathbf{K}^{(0)-1} - \sum_{i=1}^k \alpha^{(i)} (\Delta \mathbf{u}^{(i)} \cdot \mathbf{w}^{(i)T} + \mathbf{w}^{(i)} \cdot \Delta \mathbf{u}^{(i)T}) + \sum_{i=1}^k \beta^{(i)} \cdot \Delta \mathbf{u}^{(i)} \cdot \Delta \mathbf{u}^{(i)T} \quad (8.18)$$

Der berechnete Aufdatierungsvektor  $\mathbf{w}^{(k)}$  sowie die Faktoren  $\alpha^{(k)}$  und  $\beta^{(k)}$  werden zusammen mit  $\Delta \mathbf{u}^{(k)}$  gesondert gespeichert.

Die Aufdatierungsvektoren  $\mathbf{w}^{(i)}$  werden verändert:

$$\begin{aligned} \mathbf{w}^{(i)} &= \mathbf{K}^{(i-1)-1} \cdot (\mathbf{r}^{(i-1)} + \Delta \lambda^{(i)} \cdot \mathbf{P}) - \mathbf{K}^{(i-1)-1} \cdot \mathbf{r}^{(i)} \\ &= \Delta \mathbf{u}^{(i)} / s + \frac{(s-1)}{s} \cdot \Delta \lambda^{(i)} \cdot \mathbf{K}^{(i-1)-1} \cdot \mathbf{P} - \mathbf{K}^{(i-1)-1} \cdot \mathbf{r}^{(i)} \end{aligned} \quad (8.19)$$

Neue Aufdatierungsfaktoren:

$$\begin{aligned} \alpha^{(i)} &= \frac{1}{\Delta \mathbf{u}^{(i)T} (\mathbf{r}^{(i-1)} - \mathbf{r}^{(i)} + \Delta \lambda^{(i)} \cdot \mathbf{P})} \\ \beta^{(i)} &= \alpha^{(i)} [\alpha^{(i)} \cdot \mathbf{w}^{(i)T} (\mathbf{r}^{(i-1)} - \mathbf{r}^{(i)} + \Delta \lambda^{(i)} \cdot \mathbf{P}) + 1] \end{aligned} \quad (8.20)$$

Wie bei allen anderen Quasi-Newton Algorithmen erfordert die Änderung des Aufdatierungsvektors bei einfacher Strahlminimierung keine zusätzlichen Operationen. Nur die Änderung der Aufdatierungsfaktoren ist zu beachten.

*Bemerkung 8.3:* Alle Quasi-Newton Verfahren können auch in der einfachen Fassung der Single Cycle oder Memoryless Versionen mit den Kurvenverfolgungsalgorithmen verknüpft werden. Die algorithmischen Änderungen der Standardverfahren zu den "Memoryless" Verfahren sind identisch wie in Abschnitt (4.4) geschildert.

Die doppelte Lösung des Gleichungssystems im k-ten Iterationsschritt lautet:

1. Lösung für die aktuelle Ungleichgewichtskraft  $\mathbf{r}^{(k)}$

$$\begin{aligned} \Delta \mathbf{u}^{(k+1)II} = & \mathbf{K}^{(0)-1} \cdot \mathbf{r}^{(k)} - \sum_{i=1}^k \alpha^{(i)} \cdot \Delta \mathbf{u}^{(i)} \cdot \mathbf{w}^{(i)t} \cdot \mathbf{r}^{(k)} \\ & + \sum_{i=1}^k (\alpha^{(i)} \cdot \mathbf{w}^{(i)} + \beta^{(i)} \cdot \Delta \mathbf{u}^{(i)}) \Delta \mathbf{u}^{(i)t} \cdot \mathbf{r}^{(k)} \quad (8.21) \end{aligned}$$

Der Vektor  $\mathbf{w}^{(k)}$  und die Faktoren  $\alpha^{(k)}$ ,  $\beta^{(k)}$  werden im Zuge dieses Schrittes berechnet und zusammen mit  $\Delta \mathbf{u}^{(k)}$  gesondert gespeichert.

2. Lösung für den Lastvektor  $\mathbf{P}$

$$\begin{aligned} \Delta \mathbf{u}^{(k+1)I} = & \overbrace{\mathbf{K}^{(0)-1} \cdot \mathbf{P}}^{\Delta \mathbf{u}^{(1)}} - \sum_{i=1}^{k-1} \alpha^{(i)} \Delta \mathbf{u}^{(i)} \mathbf{w}^{(i)t} \cdot \mathbf{P} \\ & + \sum_{i=1}^{k-1} (\alpha^{(i)} \cdot \mathbf{w}^{(i)} + \beta^{(i)} \cdot \Delta \mathbf{u}^{(i)}) \Delta \mathbf{u}^{(i)t} \cdot \mathbf{P} \quad \left. \vphantom{\sum_{i=1}^{k-1}} \right\} \Delta \mathbf{u}^{(k)I} \\ & - \alpha^{(k)} \Delta \mathbf{u}^{(k)} \mathbf{w}^{(k)t} \cdot \mathbf{P} + (\alpha^{(k)} \cdot \mathbf{w}^{(k)} + \beta^{(k)} \cdot \Delta \mathbf{u}^{(k)}) \Delta \mathbf{u}^{(k)t} \cdot \mathbf{P} \quad (8.22) \end{aligned}$$

Wird auch hier der Vektor  $\Delta \mathbf{u}^{(k)I}$  gespeichert, so sind für die zweite Lösung nur 2 Skalarprodukte und zwei Vektoradditionen erforderlich.

Das schrittweise Vorgehen in den k-ten Iterationen ist:

Schritte 1-3 Wie beim Standardverfahren

Schritt 4 Neue Aufdatierungsfaktoren

$$\alpha^{(k)} = \frac{1}{\Delta \mathbf{u}^{(k)t} (\mathbf{r}^{(k-1)} - \mathbf{r}^{(k)} + \Delta \lambda^{(k)} \cdot \mathbf{P})}$$

$$\beta^{(k)} = \alpha^{(k)} [\alpha^{(k)} \cdot \mathbf{w}^{(k)t} (\mathbf{r}^{(k-1)} - \mathbf{r}^{(k)} + \Delta \lambda^{(k)} \cdot \mathbf{P}) + 1]$$

Schritte 5-6 Wie beim Standardverfahren

Ergebnis  $\Delta \mathbf{u}^{(k+1)II}$

Schritt 7 Hole  $\Delta \mathbf{u}^{(k)I}$  aus dem Speicher

Schritt 8  $\gamma = \mathbf{w}^{(k)t} \cdot \mathbf{P} \quad \delta = \Delta \mathbf{u}^{(k)t} \cdot \mathbf{P}$

$$\Delta \mathbf{u}^{(k+1)t} = \Delta \mathbf{u}^{(k)t} - (\alpha^{(k)} \cdot \gamma - \beta^{(k)} \cdot \delta) \Delta \mathbf{u}^{(k)} + \alpha^{(k)} \cdot \delta \cdot \mathbf{w}^{(k)}$$

Schritt 9

Speichern von  $\Delta \mathbf{u}^{(k+1)t}$

## 8.2 Diskussion / Schrifttum

Die Verbindung der Quasi-Newton Verfahren mit allgemeinen Kurvenverformungsalgorithmen wird im Schrifttum vereinzelt im Zusammenhang mit dem BFGS-Algorithmus von /Padovan 1985/, /Crisfield 1982,1983/ /Runesson et al 1986/ und im Zusammenhang mit dem Broyden- und dem Davidon Verfahren von /Zhu, Parisch 1982/ angesprochen. Im letzteren werden die beiden Rang-Eins Verfahren mit dem BFGS bzw. DFP-Verfahren verwechselt.

In allen Arbeiten wird der eigentliche Vorteil der vektoriiellen Algorithmen in Bezug auf die doppelte Lösung des Gleichungssystems nicht erwähnt. /Crisfield 1983/ entwickelt aus der vektoriiellen BFGS-Form nach Gl. 4.81 nur einfache Beschleunigungsformeln. Ansonsten wird der von /Matthies, Strang 1979/ vorgeschlagene BFGS – Algorithmus mit der Produktform verwendet. Diese Form ist aber auf positiv definite Matrizen beschränkt, d.h. für negativ definite Matrizen, wie sie nach dem Überschreiten von Durchschlagpunkten vorliegen, gibt es diese spezielle Aufteilung nicht. Dann muß entweder mit der Originalform (siehe Abschnitt 4.3.3 Gl. 4.80 ) oder mit der in Abschnitt 8.1 benutzten vektorisierten Form gearbeitet werden. Dies wird in fast allen vorliegenden Arbeiten nicht beachtet.

/Padovan 1985/ schlägt zwar vor, dann im Gegensatz zu positiv definiten Matrizen den Vektorupdate nach /Brodie et al 1973/ mit den negativen Vorzeichen zu verwenden. Hierzu fehlt jedoch jegliche analytische Begründung.

### Effizienz

Im Vergleich zum modifizierten Newton Verfahren bieten die Quasi-Newton Verfahren auch in der Kombination mit den Kurvenverfolgungsalgorithmen eine sehr effiziente Alternative. Der wesentliche Mehraufwand beschränkt sich auf einige skalare Multiplikationen von Vektoren und die Speicherung der Aufdatierungsvektoren.

### 8.3 Kombination mit konsistenter Strahlminimierung

Wird bei der Kombination der Quasi-Newton Verfahren mit Kurvenverfolgungsalgorithmen eine konsistente Strahlminimierung durchgeführt, so ist dies in allen vier dargestellten Verfahren nur im Aufdatierungsvektor gesondert zu berücksichtigen.

Bei einfacher Strahlminimierung gilt:

$$\begin{aligned}\Delta \mathbf{u}^{(i)} &= s \cdot \mathbf{d}^{(i)} \\ \Delta \lambda^{(i)} &= s \cdot \Delta \lambda^{(i,1)}(s=1)\end{aligned}\quad (8.23)$$

mit

s... Konvergierter Wert der Strahlminimierung ( $0 \leq s \leq 1$ )

$$\mathbf{d}^{(i)} = \Delta \lambda^{(i,1)}(s=1) \Delta \mathbf{u}^I + \Delta \mathbf{u}^{II} \quad (8.24)$$

Dann kann vereinfachend gesetzt werden

$$\begin{aligned}\mathbf{K}^{(i-1)^{-1}} \cdot \mathbf{r}^{(i-1)} + \Delta \lambda^{(i)} \cdot \mathbf{K}^{(i-1)^{-1}} \cdot \mathbf{P} &= \Delta \mathbf{u}^{II} + \Delta \lambda^{(i)} \cdot \Delta \mathbf{u}^I \\ &= \Delta \mathbf{u}^{(i)}/s + \frac{(s-1)}{s} \Delta \lambda^{(i)} \Delta \mathbf{u}^I \\ &= \mathbf{d}^{(i)} + \frac{(s-1)}{s} \Delta \lambda^{(i)} \Delta \mathbf{u}^I\end{aligned}\quad (8.25)$$

Bei nichtlinearen Zusatzgleichungen und konsistenter Strahlminimierung kann  $\Delta \mathbf{u}^{(i)}$  nicht durch eine skalare Multiplikation unmittelbar aus  $\mathbf{d}^{(i)}$  gewonnen werden, da  $\Delta \lambda^{(i)}$  nicht linear von s abhngig ist.

Dann ist

$$\Delta \mathbf{u}^{(i)} = \Delta \lambda^{(i,s)}(s) \Delta \mathbf{u}^I + s \cdot \Delta \mathbf{u}^{II} \quad (8.26)$$

Da der Vektor  $\Delta \mathbf{u}^I$  bei den in Abschnitt 8.1 vorgeschlagenen vektorisierten Algorithmen bereits gespeichert wird, ist dann nur noch der Vektor  $\Delta \mathbf{u}^{II}$  zusätzlich zu speichern. Dieser Vektor muß in allen Algorithmen nur bis jeweils Schritt 3 bereit gehalten werden und wird später nicht mehr benötigt. Der Mehraufwand an Speicherplatz innerhalb der gesamten Analyse beträgt eventuell eine Vektorlänge. Dies ist eine vernachlässigbare Größe.

Eine elegantere Möglichkeit beruht auf der Feststellung, daß der Unterschied in  $\alpha^{(i)}$  und  $\Delta \mathbf{u}^{(i)}/s$  nur auf den Lastfaktor beschränkt ist.

Es gilt:

$$\Delta \mathbf{u}^{(i)}/s = \frac{\Delta \lambda^{(i)}}{s} \cdot \Delta \mathbf{u}^I + \Delta \mathbf{u}^{II} \quad (8.27)$$

und

$$\begin{aligned} \mathbf{K}^{(i-1)^{-1}} \cdot \mathbf{r}^{(i-1)} + \Delta \lambda^{(i)} \cdot \mathbf{K}^{(i-1)^{-1}} \cdot \mathbf{P} &= \Delta \mathbf{u}^{(i)}/s + (\Delta \lambda^{(i)} - \Delta \lambda^{(i)}/s) \Delta \mathbf{u}^I \\ &= \mathbf{d}^{(i)} + \frac{(s-1)}{s} \Delta \lambda^{(i)} \Delta \mathbf{u}^I \quad (8.28) \end{aligned}$$

Damit kann die Speicherung von  $\Delta \mathbf{u}^{II}$  entfallen.

## 9 Erweiterung auf viele Zusatzgleichungen

Die Erweiterung des Ausgangssystems um mehrere Zusatzgleichungen ist prinzipiell möglich. Ingenieurmäßige Anwendungen werden von /Kröplin, Dinkler 1985/ und /Wriggers et al 1988/ angegeben. Im mathematischen Schrifttum existiert eine Vielzahl von Arbeiten über solche Systeme siehe /Keener, Keller 1973/ und /Fink, Rheinboldt 1984/.

Das prinzipielle Vorgehen bei der Lösung des erweiterten Systems kann analog zu den Lösungsverfahren für einfach erweiterte Systeme erfolgen. Für die Reduktion auf das ursprüngliche System soll hier kurz der Lösungsweg gezeigt werden. Die formelmäßige Herleitung entstammt den Arbeiten von /Zielke 1970/ und /Keller 1978/.

Liegt ein beliebiger Satz nichtlinearer Zusatzgleichungen vor, so kann im Sinne des Newton-Verfahrens eine Linearisierung vorgenommen werden. Hierbei ist aber zu beachten, daß auch das Ausgangssystem bezüglich dieser Variablen linearisiert wird.

Ausgangssystem

$$G(\mathbf{x}, \underline{\lambda}, \eta) = \eta^t \mathbf{F}(\mathbf{x}, \underline{\lambda}, \mathbf{P}) \quad (9.1)$$

Zusätzlicher Variablenvektor  $\underline{\lambda}$  (Steuerungsvariable)

Die Anwendung der Variationsrechnung führt auf die nichtlinearen Gleichungen:

$$\mathbf{F}(\mathbf{x}, \underline{\lambda}, \mathbf{P}) = \mathbf{0} \quad (9.2)$$

Zusatzgleichungen

$$\mathbf{F}_z(\mathbf{x}, \underline{\lambda}) = \mathbf{0} \quad (9.3)$$

Die Anwendung der konsistenten Linearisierung führt auf:

$$\mathbf{K}^{(i)} = \frac{d}{d\epsilon} \mathbf{F}(\mathbf{x}^{(i)} + \epsilon \Delta \mathbf{x}^{(i)}, \underline{\lambda}^{(i)}) |_{\epsilon=0} \quad (9.4)$$

$$\mathbf{C}^{(i)} = \frac{d}{d\epsilon} \mathbf{F}(\mathbf{x}^{(i)}, \underline{\lambda}^{(i)} + \epsilon \Delta \underline{\lambda}^{(i)}) |_{\epsilon=0} \quad (9.5)$$

$$\mathbf{V}^{(i)} = \frac{d}{d\epsilon} \mathbf{F}_z(\mathbf{x}^{(i)} + \epsilon \Delta \mathbf{x}^{(i)}, \underline{\lambda}^{(i)}) |_{\epsilon=0} \quad (9.6)$$

$$\mathbf{H}^{(i)} = \frac{d}{d\epsilon} \mathbf{F}_z(\mathbf{x}^{(i)}, \underline{\lambda}^{(i)} + \epsilon \Delta \underline{\lambda}^{(i)}) \Big|_{\epsilon=0} \quad (9.7)$$

Damit ergibt sich das vollständige System zu:

$$\begin{bmatrix} \mathbf{K}^{(i)} & \mathbf{C}^{(i)} \\ \mathbf{V}^{(i)} & \mathbf{H}^{(i)} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x} \\ \Delta \underline{\lambda} \end{bmatrix} = - \begin{bmatrix} \mathbf{F}(\mathbf{x}^i, \underline{\lambda}^i) \\ \mathbf{F}_z(\mathbf{x}^i, \underline{\lambda}^i) \end{bmatrix} \quad (9.8)$$

Die Lösung basiert wieder auf der Aufteilung in zwei Teile:

$$\Delta \mathbf{X} = \mathbf{U}^I \cdot \Delta \underline{\lambda} + \mathbf{U}^{II} \quad (9.9)$$

mit

$$\mathbf{X} = -\mathbf{K}^{(i)-1} \cdot \mathbf{C}^{(i)} \cdot \Delta \underline{\lambda} - \mathbf{K}^{(i)-1} \cdot \mathbf{F}^{(i)} \quad (9.10)$$

$$\mathbf{U}^I = -\mathbf{K}^{(i)-1} \cdot \mathbf{C} \quad (9.11)$$

Damit:

$$\mathbf{V}^{(i)} \cdot \mathbf{U}^I \Delta \underline{\lambda} + \mathbf{V}^{(i)} \cdot \mathbf{U}^{II} + \mathbf{H} \cdot \underline{\lambda} = -\mathbf{F}_z^{(i)} \quad (9.12)$$

$$\Delta \underline{\lambda} = [\mathbf{V}^{(i)-1} \cdot \mathbf{K}^{(i)-1} \cdot \mathbf{C}^{(i)} - \mathbf{H}^{(i)}]^{-1} \cdot [\mathbf{F}_z^{(i)} + \mathbf{V}^{(i)} \cdot \mathbf{U}^{II}] \quad (9.13)$$

Voraussetzung hierfür ist, daß die Matrix  $[\mathbf{V}^{(i)} \mathbf{K}^{(i)-1} \mathbf{C}^{(i)} - \mathbf{H}]$  invertierbar ist.

Dasselbe Ergebnis läßt sich auch über die sukzessive Anwendung der Sherman/Morrison Formel erreichen. Formal muß die Matrix  $\mathbf{H}$  für Zwischenschritte invertierbar sein. Dies entfällt bei der sich ergebenden Formel.

Das hier dargestellte Vorgehen – von /Keller 1982/ als "bordering algorithm" bezeichnet – stellt in keinem Schritt irgendwelche Anforderungen an  $\mathbf{H}$ . D.h. die Zusatzbedingungen können alle frei von den Parametern  $\underline{\lambda}_i$  sein, z.B.  $\mathbf{H} = \mathbf{0}$ .

# 10 Numerische Beispiele

## 10.1 Allgemeines

Die meisten der vorgestellten Algorithmen wurden vom Verfasser in das Finite Element Programm NISA /Brendel et al 1977,1981/ des Instituts für Baustatik der Universität Stuttgart und in das Programm FEAP /Taylor 1985/ eingebaut, bzw. es wurden bestehende Algorithmen /Ramm 1981/ modifiziert.

Aus der Vielzahl der Möglichkeiten wurden die wesentlichen Algorithmen an den folgenden Beispielen getestet. Sie sollen einen Überblick über die Leistungsfähigkeit und die Grenzen der einzelnen Algorithmen geben. Es wird versucht, soweit möglich, eine rationale Erklärung für das Verhalten einzelner Algorithmen zu geben. Angesichts der Komplexität der behandelten Strukturprobleme kann es aber manchmal nur bei einer Beschreibung der Beobachtungen bleiben. An dieser Stelle sei bemerkt, daß ein etwaiges Versagen eines Algorithmus trotz sorgfältiger Prüfung auch auf etwaige Fehler in der Implementierung durch den Verfasser zurückgeführt werden kann.

Die numerischen Beispiele sind so gegliedert, daß zuerst die Quasi-Newton Verfahren an geometrisch und geometrisch/ physikalisch nichtlinearen Beispielen getestet werden. Danach wird der Einfluß der verschiedenen Kurvenverfolgungsalgorithmen auf den Iterationsablauf gezeigt, und die Wirkung der Strahlminimierung auf die Konvergenzverbesserung untersucht. Hierzu werden Beispiele mit vorwiegend geometrisch nichtlinearem Verhalten gewählt, um die Eignung der verschiedenen Verfahren im Nachbeulbereich stabilitätsgefährdeter Strukturen zu zeigen. An denselben Schalenbeispielen wird die Kombination der Quasi-Newton Verfahren mit Kurvenverfolgungsalgorithmen mit und ohne Strahlminimierung getestet. Den Abschluß bildet ein größeres geometrisch und physikalisch nichtlineares Problem aus der Schalenberechnung.

Für weitere Vergleiche sei auf /Ramm, Matzenmiller 1987/ und /Matzenmiller 1988/ verwiesen, wo die implementierten Algorithmen den konsistent hergeleiteten Newton-Algorithmen gegenübergestellt werden.

### Toleranzen

Zur Bestimmung der Konvergenz werden numerische Schranken vorgegeben. Üblich sind Schranken für Verschiebungs- und Kräfte normen /Bergan, Clough 1972/ evtl. auch Energienormen. In allen Beispielen wird gegen die Euklidische Norm der Verschiebungen

$$\| \Delta \mathbf{u}^{(i)} \| < u_{tol} \cdot \| \mathbf{u}^{(i)} \|$$

Verschiebungszuwachs gegen Verschiebungen im Lastschritt und gegen die Euklidische Norm der Ungleichgewichtskräfte

$$\| \mathbf{r}^{(i)} \| < r_{tol} \cdot \| \mathbf{r}^{(1)} - \mathbf{r}^{(0)} \|$$

abgeprüft. Die Energienorm  $|\Delta \mathbf{u}^{(i)} \cdot \mathbf{r}^{(i)}| \leq e_{tol} |\Delta \mathbf{u}^{(1)} \cdot \mathbf{r}^{(1)}|$  wurde stets beobachtet, ist aber in der Dokumentation der Beispiele nicht enthalten.

Die gewählten Schrankenwerte waren:

$$u_{tol} = 10^{-3} \quad \text{und} \quad r_{tol} = 10^{-1} \div 10^{-2}$$

*Bemerkung 10.1:* Von der Vorgabe bestimmter Größen für die Schrankenwerte ist auch die Beurteilung der einzelnen Algorithmen betroffen. Werden großzügige Schrankenwerte angegeben, so werden diese von den Sekantenverfahren in wenigen Schritten schnell erreicht, die echten Newton – Verfahren besitzen dann noch keinen Vorteil. Werden hingegen scharfe, kleine Schrankenwerte angegeben, so benötigen die Sekantenverfahren trotz eventuell superlinearer Konvergenz relativ viele Schritte zur Bewältigung. Liegt hingegen die Schrittweite noch im Anziehungsbereich des Newton-Verfahrens, wird die quadratische Konvergenzeigenschaft der Newton-Verfahren deutlich und auch sehr kleine Schranken werden mit wenigen (6-7) Iterationen unterschritten.

### Nichtlineare Formulierung

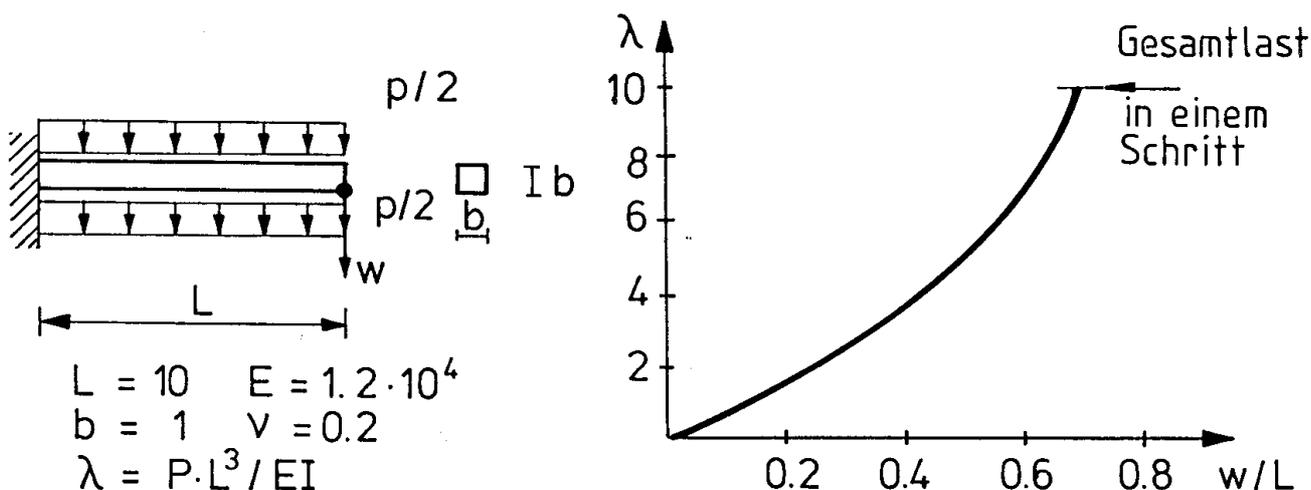
Bei allen Beispielen wird von einer materiellen Beschreibung ausgegangen /Green,Naghdi 1965/. Die speziellen Aspekte für geometrisch nichtlineare Problemstellungen sind in /Bathe,Ramm,Wilson 1975/ sowie in den weiterführenden Arbeiten /Ramm,Matzenmiller 1985/, /Matzenmiller 1988/ und für materiell nichtlineare Probleme /Sättele 1980/ /Ramm,Matzenmiller 1987/ ausführlich beschrieben (siehe dort auch für weiteres Schrifttum).

Für die Effizienz der Newton-Verfahren war die Entwicklung der konsistenten Linearisierung für geometrisch und materiell nichtlineare Probleme wesentlich. /Marsden,Hughes 1983/, /Simo,Taylor 1985/. Erst damit liegt eine richtige Tangentensteifigkeitsmatrix im Sinne des Newton-Verfahrens vor. Vorher wurde meist mit genäherten Tangentenmatrizen gearbeitet, bei denen z.B. im Falle von Schalenproblemen Terme bezüglich der Rotation fehlten und im Falle materiell nichtlinearer Probleme die Plastizität nur sehr unvollständig erfaßt wurde.

Dieser Aspekt muß für einen sinnvollen Vergleich mit Newton-Verfahren beachtet werden.

## 10.2 Kragbalken unter gleichförmiger Belastung

Ein einfaches Beispiel für den Test unterschiedlicher Verfahren stellt der in /Bathe,Cimento 1980/ für Vergleichsuntersuchungen verwendete Kragbalken dar. Es handelt sich um ein rein geometrisch nichtlineares Problem, für das eine analytische Lösung vorliegt /Holden 1972/.



**Bild 10.1 Kragbalken unter gleichförmiger Belastung  
Geometrie und Last- Verformungskurve**

Die Geometrie und Materialinformationen sind in Bild 10.1 gegeben. Die Finite Element Modellierung erfolgte mit fünf Serendipity Elementen mit je 8 Knoten als ebenes Problem. Aus Vergleichsgründen wurde die gesamte Last in einem Schritt aufgebracht.

In Tabelle 10.I sind die Ergebnisse in Bezug auf die Zahl der benötigten Iterationen, Strahlminimierung, Line Search und verbrauchter Rechenzeit angegeben. Die Line Search Toleranz betrug  $\eta = 0.9$ .

Das echte Newton Verfahren konvergierte in 7 Schritten, während mit dem modifizierten Newton Verfahren auch mit Strahlminimierung keine Konvergenz zu erzielen war.

Die Quasi - Newton Verfahren zeigten stark unterschiedliches Verhalten:

Das Broyden- und das DFP-Verfahren konvergierten auch nach 500 Iterationen noch nicht. Die Verschiebung des Broyden-Verfahrens lag dann noch um 16 % unterhalb der endgültigen Verschiebung und es traten große Ungleichgewichtskräfte auf. Beim DFP-Verfahren "verharrte" die Verschiebung rund 40 % unterhalb des erwarteten Ergebnisses ohne Konvergenzfortschritt oder Divergenz auch bei höheren Iterationszahlen. Das Davidon- und das BFGS-Verfahren in allen Variationen konvergierten ohne Zusatzmaßnahmen. Der Zeitvorteil der in Abschnitt 4.3.3 vorgestellten vektorisierten Algorithmen ist bei diesem kleinen Beispiel gering.

Toleranzen:  $\epsilon_F = \frac{\|R_i\|}{\|R_i\|} = 10^{-2}$  Ungleichgew. kräfte  
 $\epsilon_d = \frac{\|\Delta u_i\|}{\|\Delta u_1\|} = 10^{-3}$  Verschiebung  
 line search  $\eta = 0.9$

	Anzahl		CRAY CPU [sec]	Bemerkungen
	It.	LS		
Broyden	500	-	-	keine Konvergenz
	454	243	7.32	
Davidon	153		1.92	
	78	26	1.42	
DFP	500	-	-	keine Konvergenz  Line Search nur am Anfang
	469	10	8.52	
BFGS	74		1.31	
BFGS(P)	74	-	1.34	
BFGS (M/S)	74	-	1.34	
	67	21	1.37	
BFGS(P)	117	62	2.00	
Newton	7	-	.43	"schlechte" Updates nicht benutzt

**Tabelle 10.I Vergleich der Iterationsverfahren - Iterationszahlen/Rechenzeit**

Die Bedeutung der Strahlminimierung bei Quasi-Newton-Verfahren ist of-

fensichtlich. Das Broyden- und das DFP-Verfahren konvergieren mit dieser Zusatzmaßnahme zu richtigen Ergebnissen. Allerdings ist die Zahl der Iterationen in beiden Fällen sehr hoch. Beim Davidon-Verfahren wird der größte Gewinn an ersparter Rechenzeit erzielt. Bei den BFGS- Algorithmen wird zwar die Zahl der Iterationen verringert, aber der Aufwand für die Strahlminimierung führt zu einer insgesamt etwas höheren Rechenzeit.

Wird die Produktform des BFGS- Verfahrens mit der Beschränkung der Konditionszahl für verschiedene Update Vektoren verwendet /Matthies,Strang 1979/, so ergibt sich insgesamt ein größerer Rechenaufwand. Dieser Aspekt wurde nicht weiter verfolgt, da die Konditionszahl der Matrizen bei Durchschlagproblemen ohnehin sehr groß wird und dann eine Limitierung nicht mehr möglich ist. Allerdings läßt dieses Beispiel den Schluß zu, daß die zulässige Konditionszahl sehr groß gewählt werden kann, ohne auf numerische Probleme zu führen. Dies hat sich in weiteren numerischen Beispielen bestätigt.

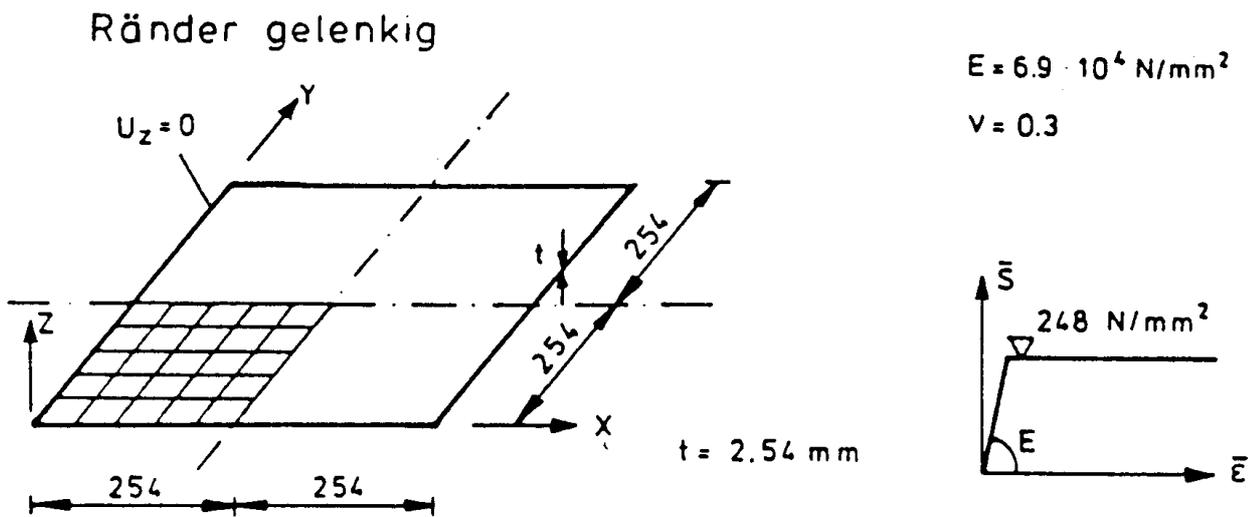
Die Line-Search Toleranz wurde nicht variiert, da damit zwar die Zahl der Iterationen eventuell zu verringern ist, aber der Gesamtaufwand wegen der häufigen Auswertungen des Residuums wieder steigt.

Die wesentlichen Ergebnisse dieses Beispiels sind:

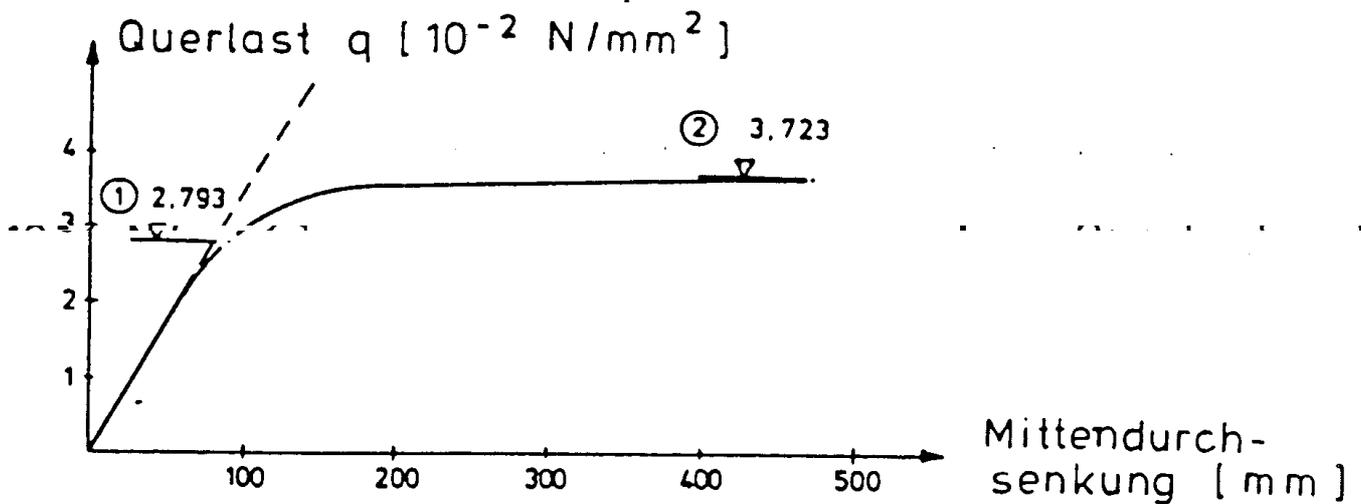
- Das Newton-Verfahren war am effizientesten.
- Das Davidon- und das BFGS-Verfahren liefern das beste Ergebnis der Quasi-Newton Verfahren.
- Die Strahlminimierung stabilisiert das Konvergenzverhalten und reduziert die Iterationszahl.

### 10.3 Quadratplatte unter gleichmäßiger Belastung

Das Biegeverhalten einer Quadratplatte unter gleichförmiger Belastung (Bild 10.2) sollte für elastisch, ideal plastisches Material untersucht werden siehe auch /Ramm, Matzenmiller 1987/. Als Rechenmodell wurde ein Viertel der Platte unter Einbeziehung der Symmetrie mit 5 biquadratischen Schalenelementen idealisiert. Da nichtlineare Effekte wegen des ausgeprägten Biegeverhaltens nur eine untergeordnete Rolle spielen, wurde geometrisch nur linear, aber materiell nichtlinear gerechnet. Das Lastverformungsverhalten ist für die Mittelpunktverschiebung in Bild 10.3 angegeben. Zum Vergleich der Verfahren wurde die Gesamtlast in zwei Schritten über Lastkontrolle aufgebracht (siehe Bild 10.3).



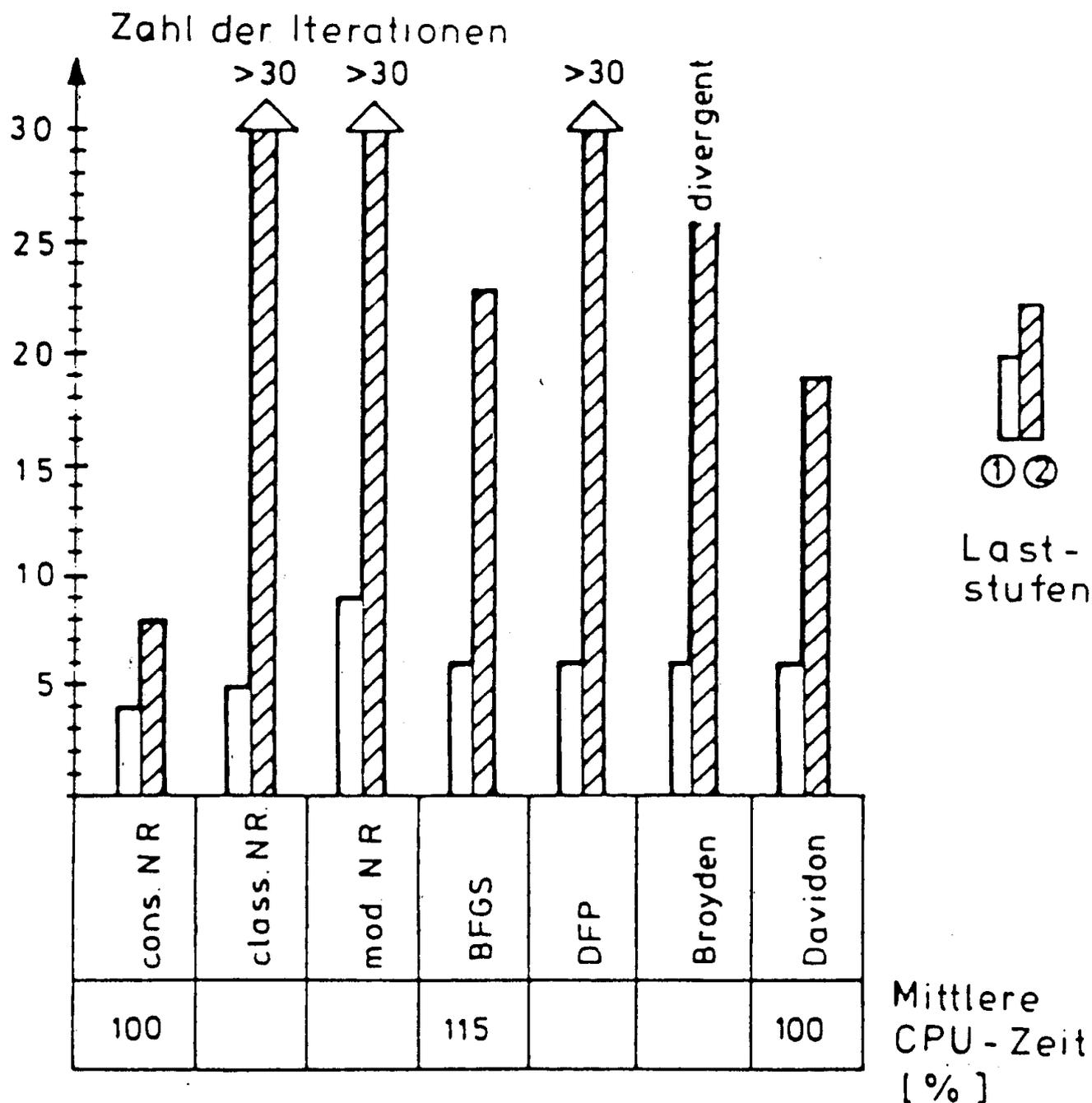
**Bild 10.2** Quadratplatte unter gleichförmiger Belastung  
Geometrie – Material – FE Modell



**Bild 10.3** Last – Verschiebungsdiagramm Quadratplatte

In Tabelle 10.II sind die benötigten Iterationszahlen der verschiedenen Verfahren einander graphisch gegenübergestellt. Die Beobachtungen aus dem ersten Beispiel 10.1 werden bestätigt. Das Newton-Verfahren mit konsisten-

ter Linearisierung ergibt die geringste Zahl an Iterationen, während das



**Tabelle 10.II Vergleich der Iterationsverfahren – Iterationszahlen/Rechenzeit**

„Newton“ Verfahren mit inkonsistenter Tangente (siehe Abschnitt 10.1) divergiert. Broyden- und DFP-Verfahren divergieren, bzw. konvergieren nicht, während sowohl die BFGS- als auch das Davidon-Verfahren konvergieren.

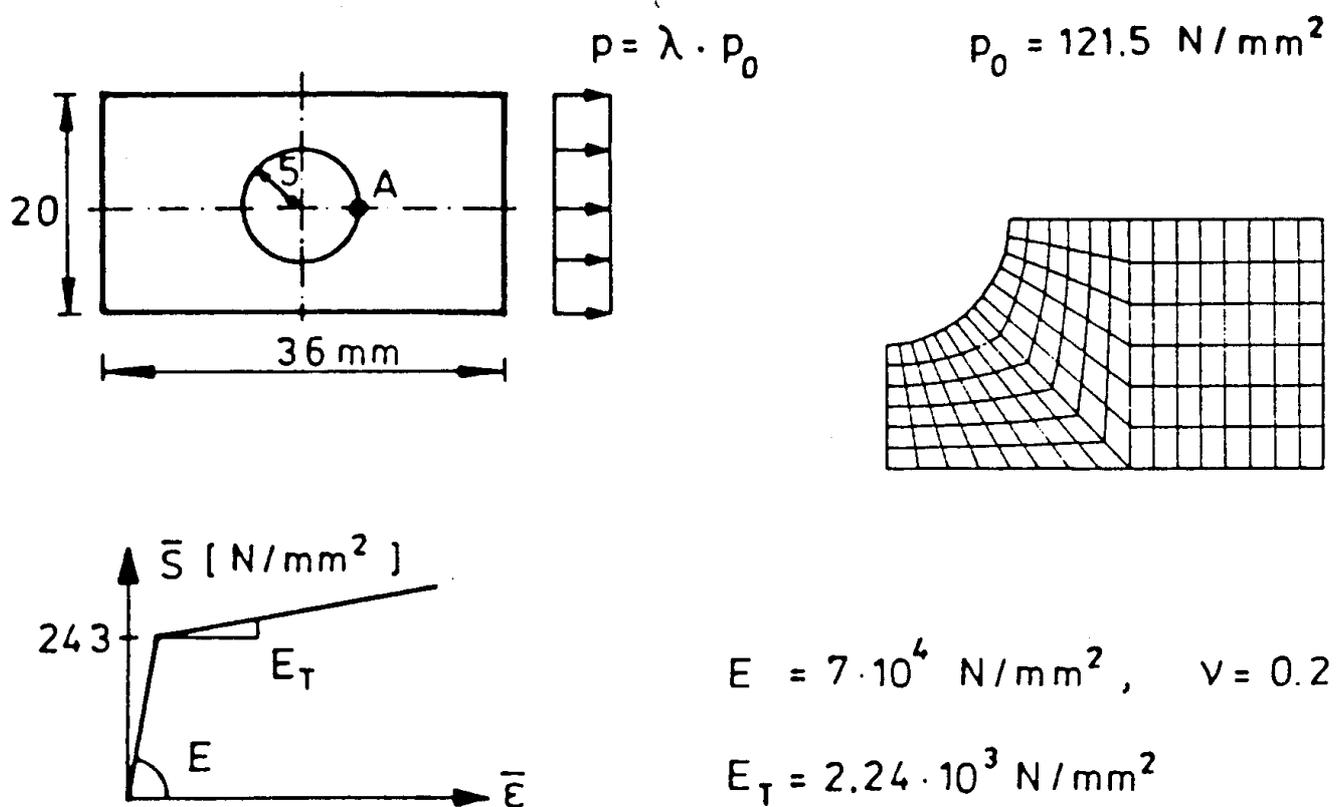
Eine wesentliche Eigenschaft der Quasi-Newton Verfahren wird beim Ver-

gleich der Rechenzeit deutlich. Trotz wesentlich größerer Iterationszahlen ist der Gesamtaufwand beim Davidon-Verfahren ebenso hoch wie beim echten Newton-Verfahren; beim BFGS-Verfahren ist er nur geringfügig höher.

Die Strahlminimierung wurde an diesem Beispiel nicht getestet.

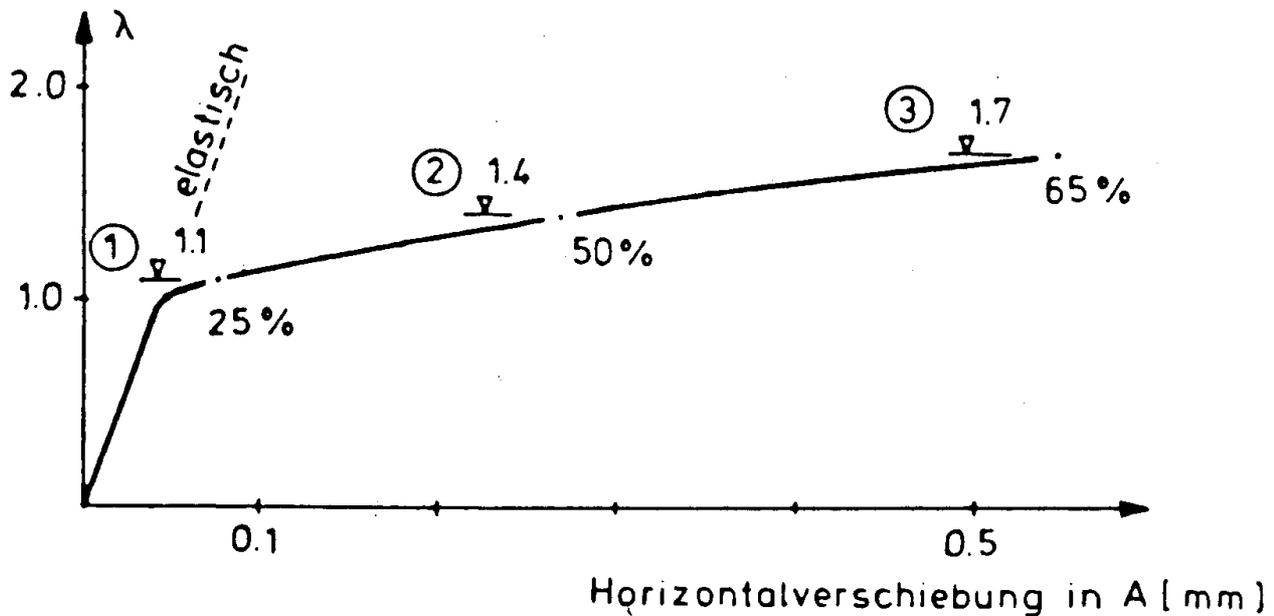
## 10.4 Scheibenstreifen mit Loch unter Zugbelastung

Dieser Scheibenstreifen (Bild 10.4) ist ein Standardbeispiel für den Vergleich für Rechenalgorithmen bei der Verwendung elastoplastischer Werkstoffe /Ramm, Matzenmiller 1987/.



**Bild 10.4** Scheibenstreifen mit Loch – Geometrie – Material – FE Modell

Als Berechnungsmodell wird ein Viertel der Struktur unter Symmetriebedingungen - diskretisiert durch 182 bilineare Scheibenelemente - gewählt. Das allgemeine Last-Verformungsverhalten ist in Bild 10.5 gegeben.

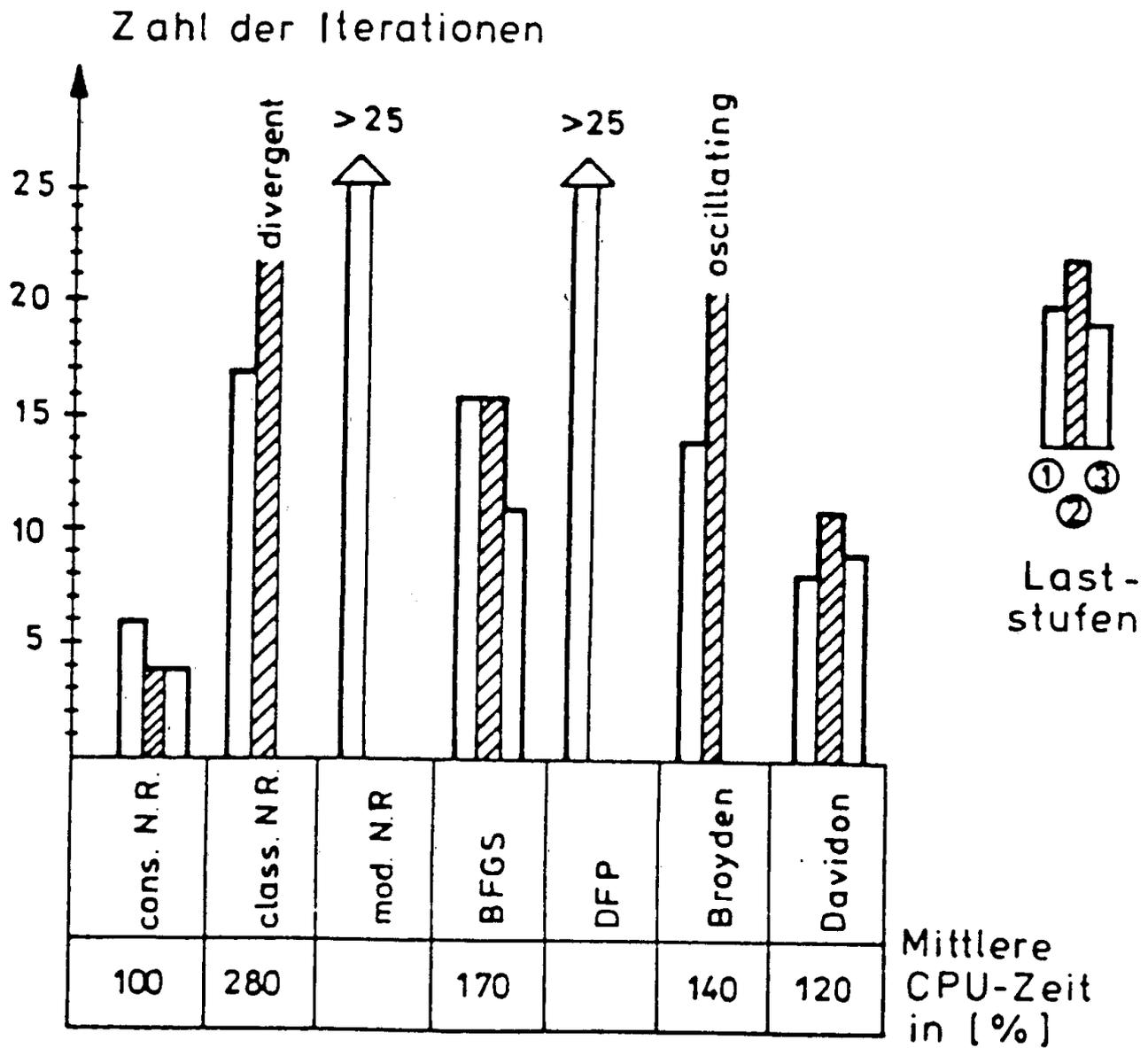


**Bild 10.5 Last- Verformungsdiagramm für Punkt A**

Für den Vergleich der Verfahren wurde die Last in drei Schritten über Lastkontrolle aufgebracht (siehe Bild 10.5).

Die in Tabelle 10.III dargestellten Ergebnisse bestätigen die Ergebnisse der vorigen Beispiele. Das Newton Verfahren mit konsistenter Linearisierung sowie die BFGS- und das Davidon- Verfahren konvergieren in allen Lastschritten. Alle anderen Verfahren divergieren bzw. konvergieren nicht, das DFP-Verfahren sogar schon im ersten Schritt.

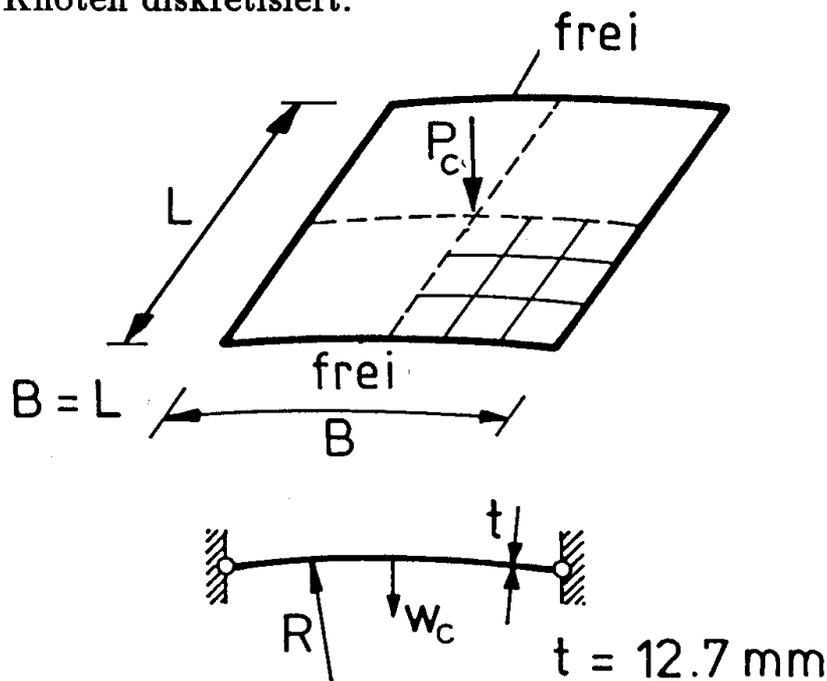
Der Gesamtrechenaufwand für die Quasi-Newton Verfahren ist erheblich höher als für das konvergente Newton Verfahren. Dies ist bei diesem kleinen Beispiel eine Folge des geringen Aufwands, der für die Erstellung und Faktorisierung der Tangentenmatrix benötigt wird.



**Tabelle 10.III Vergleich der Iterationsverfahren – Iterationszahlen/  
Rechenzeit**

## 10.5 Durchschlagproblem einer Schale mit Dickenvariation

Die in Bild 10.6 dargestellte flache Schale unter mittiger Einzellast zeigt bei Modellierung eines Viertels der Struktur mit Symmetriebedingungen ein ausgeprägtes Durchschlagverhalten (Bild 10.7). Wird die Schalendicke gleichförmig um die Hälfte verringert, so ist in der Last- Verschiebungskurve des Mittelpunktes noch eine Verschiebungsumkehr zu beobachten (Bild 10.8). Beide Schalen wurden durch ein Netz von 3 x 3 Serendipity-Schalenelementen mit je 8 Knoten diskretisiert.



$$R/L = 5.04 \quad R/t = 200, \gamma = 0.3$$

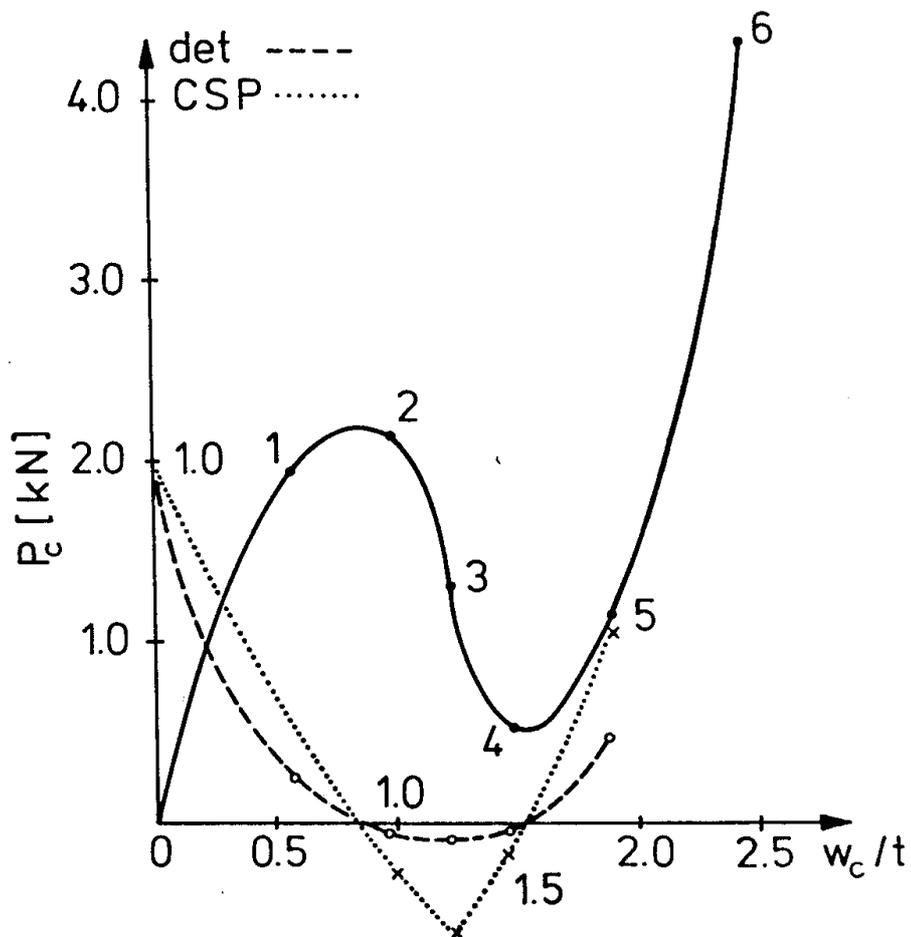
$$E = 3105 \text{ N/mm}^2$$

FE-Netz: 3x3 8Knoten Schalenelemente

**Bild 10.6** Flache Schale unter mittiger Einzellast,  
Geometrie : Dünn  $R/t = 400$ , Dick  $R/t = 200$ ; FE Modell

An diesen Beispielen sollen die verschiedenen Kurvenverfolgungsalgorithmen und ihre Kombination mit den Quasi-Newton-Verfahren und Strahlminimierung untersucht werden. Die Steuerung der Prädiktorschritte erfolgte mit der Kombination der Beobachtung des Steifigkeitsparameters und der Vorzeichen der Diagonale der  $LDL^t$  zerlegten Steifigkeitsmatrix.

Deutlich wird an Hand des Verlaufs des Steifigkeitsparameters, daß dessen Beobachtung alleine keine eindeutige Steuerung zuläßt.



**Bild 10.7 Last- Verformungsverhalten, dicke Schale  $R/t = 200$**

### 10.5.1 Vergleich der Kurvenverfolgungsalgorithmen mit und ohne Strahlminimierung

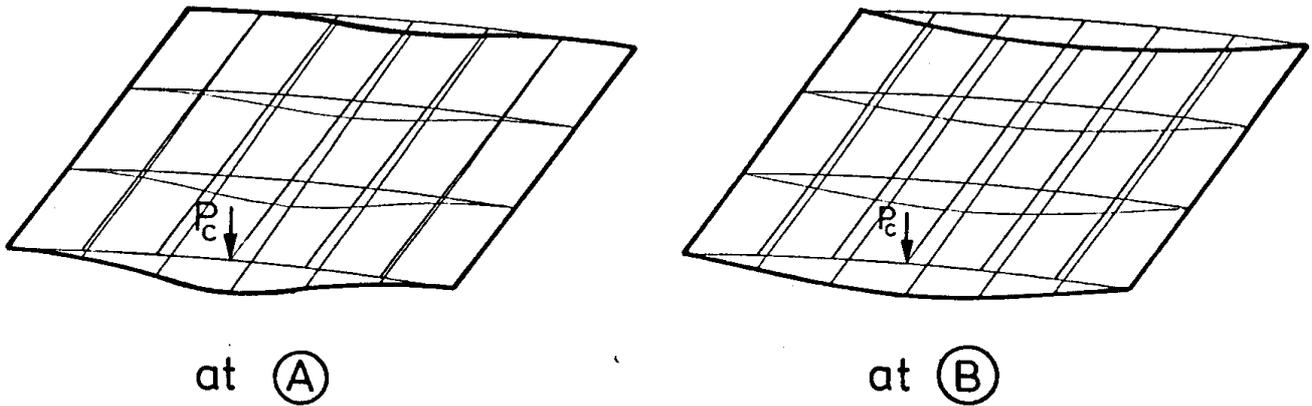
Der Vergleich beschränkt sich auf folgende Algorithmen

- Iteration auf angepaßten Normalenebenen (lineare Zusatzgleichung)
- Iteration auf Kugelflächen (quadratische Zusatzgleichung)
  - konsistente Linearisierung
  - explizite Erfüllung der Zusatzgleichung

Diese werden mit dem modifizierten Newton Verfahren und zwei Strahlminimierungsformen kombiniert.

- Einfache numerische Relaxation
- Strahlminimierung mit linearer Interpolation.



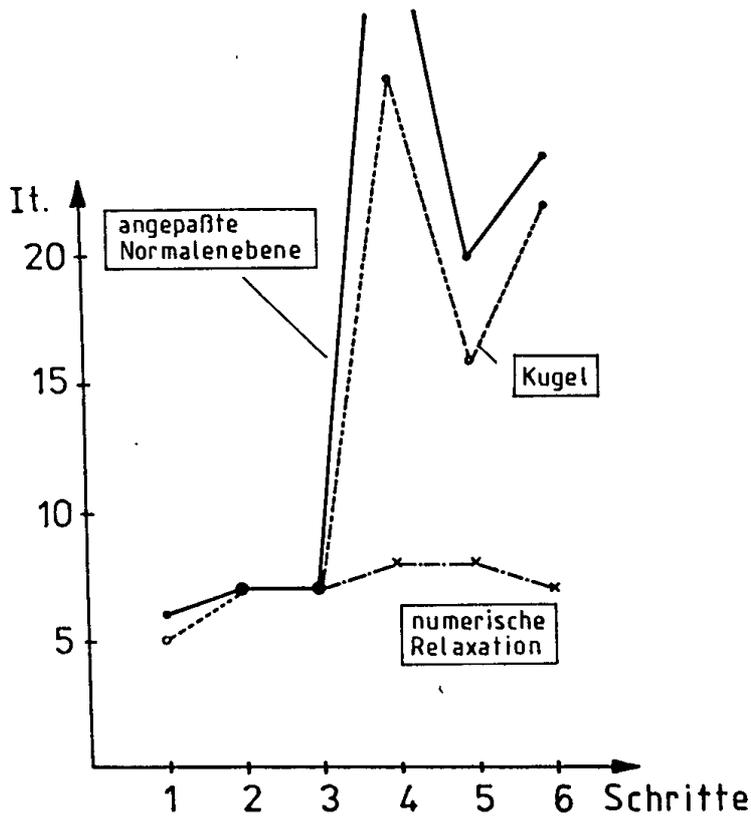


**Bild 10.8b Verschiebungsbild (halbe Schale)  $R/L = 5.0$ ,  $R/t = 400$ ,  $t = 6.35$  mm**

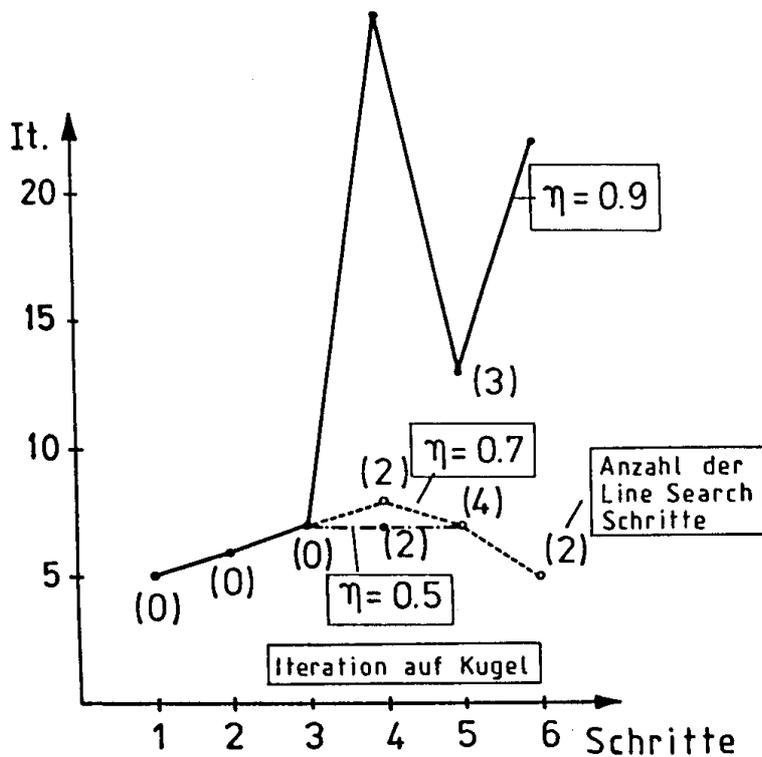
Bei Anwendung der Strahlminimierung ergibt eine grobe Toleranzschranke nur eine kaum merkbare Verringerung der Zahl der Iterationen bei gleichbleibendem bzw. vergrößertem Gesamtaufwand. Die positive Auswirkung der Strahlminimierung wird erst bei schärferen Toleranzschranken auch beim Gesamtaufwand deutlich. Eine weitere Herabsetzung der Schranke bringt keine wesentliche Verringerung des Gesamtaufwandes.

Beim Beispiel der dünnen Schale ( $t = 6.35$  mm) führt keiner der beiden Kurvenverfolgungsalgorithmen auch in der Kombination mit numerischer Relaxation zur Konvergenz (Bild 10.10a). In Schritt 4 ist für die explizite Erfüllung der quadratischen Zusatzgleichung zwar etwas Konvergenz erkennbar, die Berechnung wurde aber nach 200 Iterationen abgebrochen. Die numerische Relaxation verhindert die Divergenz in den ersten Schritten, kann aber das Versagen aller Verfahren nicht verhindern.

Die Kombination mit der Strahlminimierung führt für alle drei Algorithmen zur Konvergenz (Bild 10.10b), wobei die explizite Erfüllung der Zusatzgleichung keinen Vorteil gegenüber der konsistent linearisierten Form erbrachte. Bei groben Line-Search Toleranzen ist die Gesamtzahl der Iterationen und der Gesamtaufwand für die lineare Zusatzgleichung der Iteration auf Normalebenen etwas größer. Bei schärferen Schranken ist der Aufwand für beide

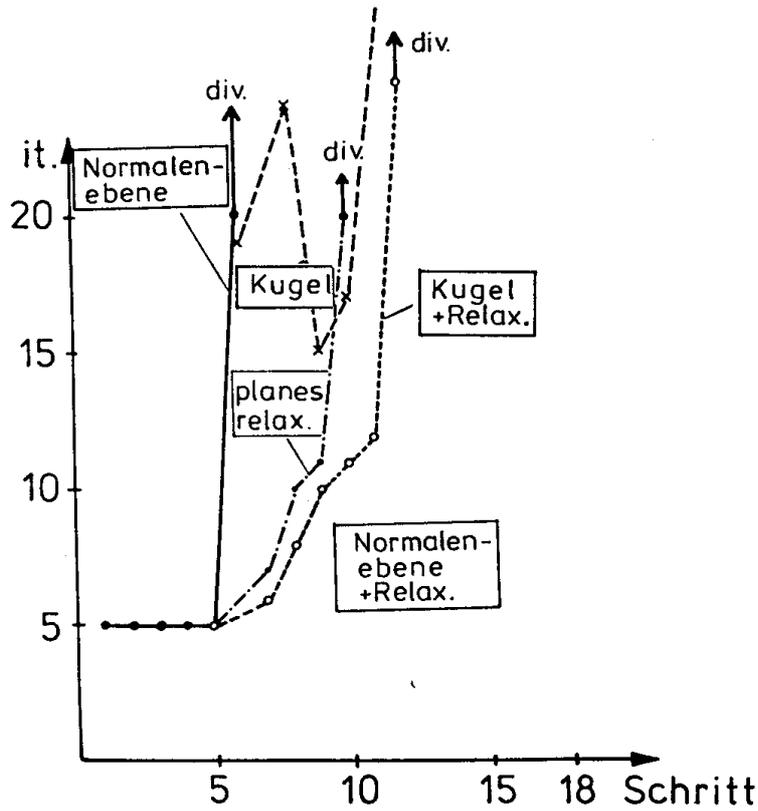


a) Vergleich der verschiedenen Zusatzgleichungen, Kombination mit numerischer Relaxation

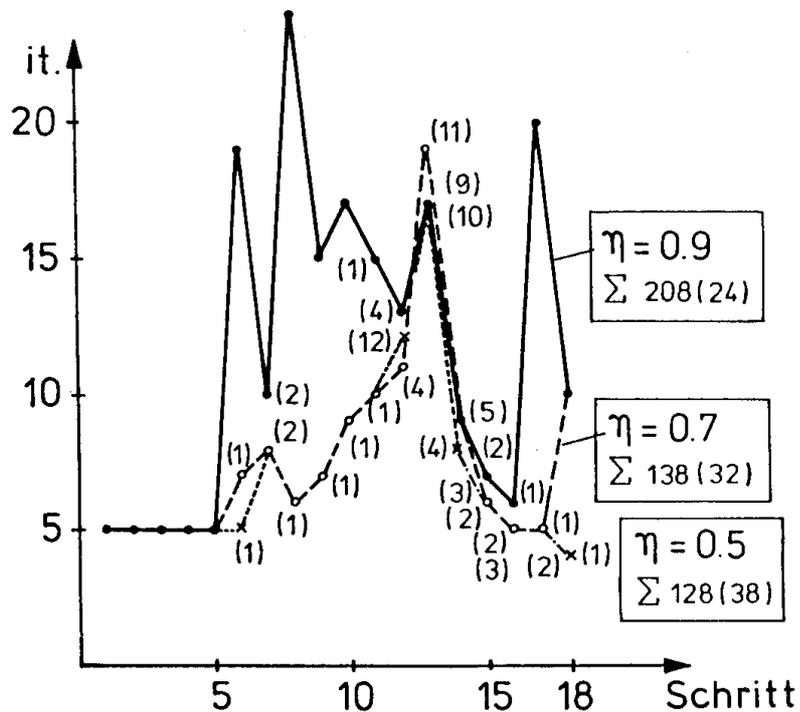


b) Kombination mit Strahlminimierung

Bild 10.9 Vergleich der Kurvenverfolgungsalgorithmen, Dicke Schale  $R/t = 100$



a) Vergleich der verschiedenen Zusatzgleichungen, Kombination mit numerischer Relaxation



b) Kombination mit Strahlminimierung (quadratische Gleichung konsistent linearisiert)

Bild 10.10 Vergleich der Kurvenverfolgungsalgorithmen,  
Dünne Schalen  $R/t = 400$

Gleichungstypen fast identisch.

### 10.5.2 Kombination mit Quasi-Newton Verfahren

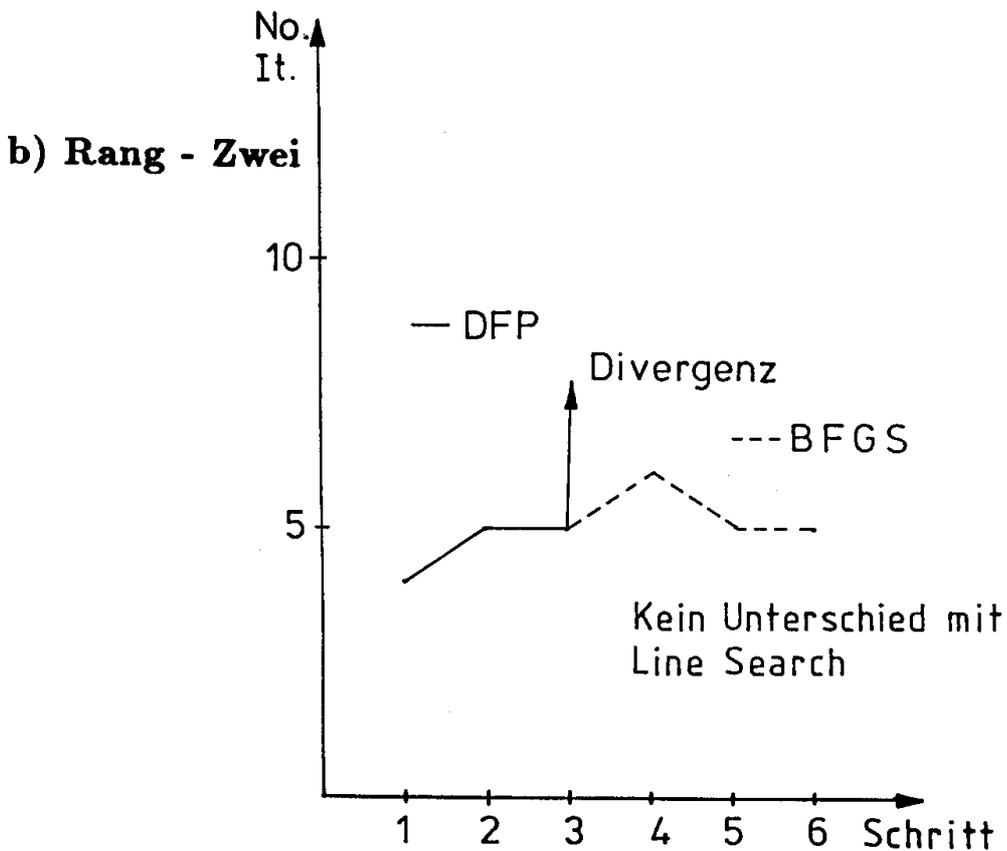
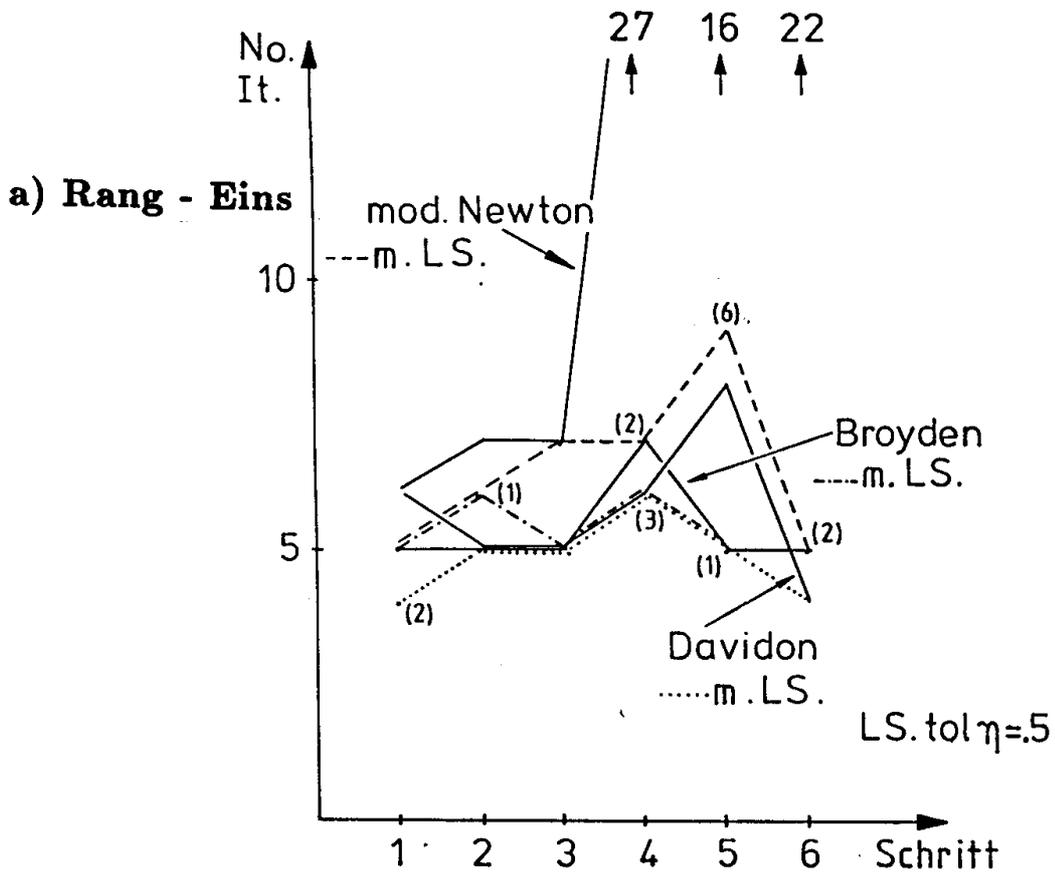
Da die quadratische Zusatzgleichung bisher zur besten Konvergenz geführt hat, wurde am Beispiel des Schalendurchschlagproblems nur die Kombination der Quasi-Newton Verfahren mit dieser Zusatzgleichung und Strahlminimierung untersucht. Als Line Search Toleranz wird  $\eta = 0.5$  gewählt. Die Diagramme in Bild 10.11 (a-b) zeigen, daß bei der Schale mit Dicke  $t = 12.7$  mm alle Quasi-Newton Verfahren bis auf das DFP-Verfahren auch ohne Strahlminimierung konvergieren. Die Strahlminimierung verbessert das Konvergenzverhalten der Rang-Eins Verfahren nur unwesentlich; bei den Rang-Zwei Verfahren wird keine Strahlminimierung durchgeführt, da die Bedingungen und Toleranzen hierfür im Zuge der Berechnung nicht erfüllt bzw. unterschritten werden. Das BFGS-Verfahren zeigt monotone Konvergenz, während die beiden Rang-Eins-Verfahren nichtmonotone Konvergenz aufweisen.

Bei der dünnen Schale ( $t = 6.35$  mm) (Bild 10.12a,b) zeigt sich für die Rang-Eins Verfahren ein ähnliches Verhalten wie für die dicke Schale. Die Rang-Zwei Verfahren bieten hingegen ein schwer deutbares Bild. Das BFGS-Verfahren konvergiert mit deutlichem Mehraufwand gegenüber den Rang-Eins Verfahren, während das DFP-Verfahren divergiert.

Wird zusätzlich Strahlminimierung angewandt, divergiert das BFGS-Verfahren, während das DFP-Verfahren konvergiert. Eine genauere Untersuchung des Verhaltens des BFGS-Verfahrens wies auf große Konvergenzschwierigkeiten in der Nähe des Verschiebungsumkehrpunktes der Kurve in Bild 10.8 hin. Die Diskriminante der expliziten Lösung ist in diesem Bereich negativ, und es mußte auf die konsistente Formulierung übergegangen werden. Auch kleinere Schrittweiten führten zu keiner Behebung der Schwierigkeiten. Diese Probleme erfordern noch eine gesonderte Untersuchung.

Zusammenfassend kann festgestellt werden:

- Das Davidon- und BFGS-Verfahren zeigten für die dicke Schale das beste Verhalten.
- Das Davidon- und das Broyden-Verfahren waren bei der dünnen Schale überlegen.



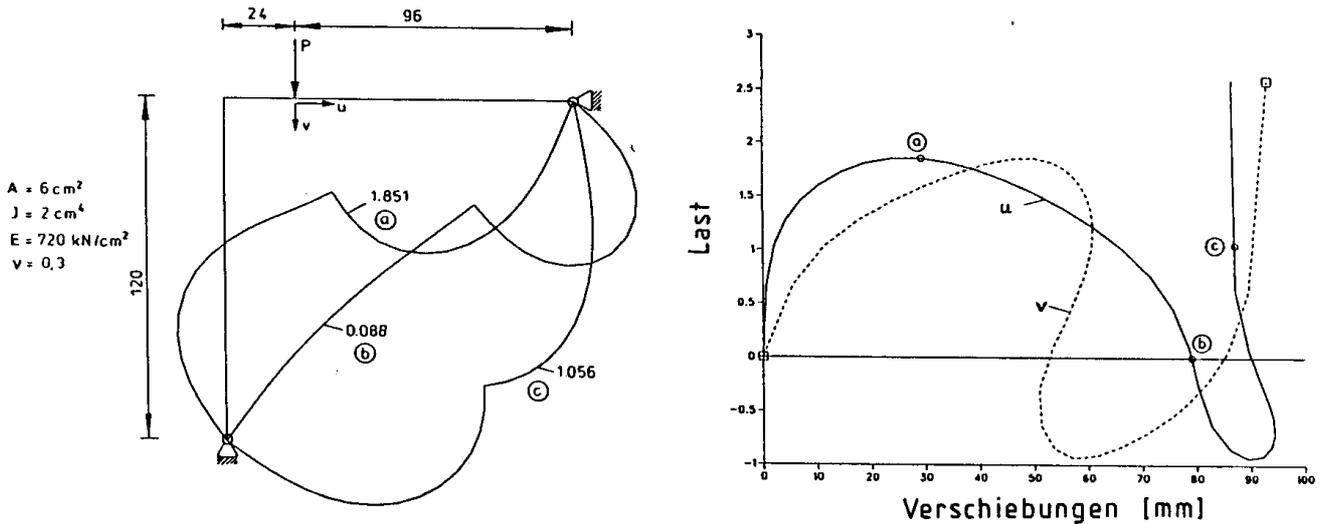
**Bild 10.11 Dicke Schalen, Kombination mit Quasi Newton Verfahren**

- a) Rang - Eins Änderungen
- b) Rang - Zwei Änderungen



## 10.6 Rahmen unter Einzellast - Große Rotationen

Der in Bild 10.13a dargestellte, ebene Rahmen ist ein Beispiel /Lee 1968/, bei dem das Stabilitätsversagen erst bei großen Verschiebungen auftritt (Bild 10.13b). Außerdem ist hier eine Verschiebungsumkehr für den Lastaufpunkt zu beobachten.



**Bild 10.13. Rahmen unter Einzellast**

a) Geometrie – Verformte Figur

b) Last – Verschiebungskurven, Lastaufpunkt

Versuche, mit dem modifizierten Newton Verfahren oder den Quasi-Newton Verfahren zu arbeiten, scheiterten, da die Schrittweiten für eine sinnvolle Berechnung zu klein gewählt werden mußten. Daher wurde das echte Newton Verfahren mit drei Variationen der Kurvenverfolgungsalgorithmen zur Lösung herangezogen.

- Iteration auf angepaßten Normalenebenen (lineare Zusatzgleichung)
- Iteration auf Kugelflächen (quadratische Zusatzgleichung)
  - konsistente Linearisierung
  - explizite Erfüllung der Zusatzgleichung

Den Vergleich der Iterationszahlen der drei Algorithmen für unterschiedliche Schrittweiten zeigt Tabelle 10.IV.

Schrittweite	Iterationsverfahren	Schritt 1	Schritt 2	Schritt 3	Schritt 4	Schritt 5	Schritt 6	Schritt 7	Schritt 8	Schritt 9	Schritt 10	Schritt 11	Schritt 12	Schritt 13	Schritt 14	Schritt 15	Schritt 16	Schritt 17	Schritt 18	Schritt 19	Schritt 20	Schritt 21	Schritt 22	Schritt 23	Schritt 24	Schritt 25	Schritt 26	Schritt 27	
25.0	Angepaßte Normalenebenen	5	5	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	
		5	5	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
		LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP									
35.0	Konsistent Linearisiert	5	5	5	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	
		5	5	5	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
		LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP									
40.0	Kugel (explizit)	6	5	5	5	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	
		6	5	5	5	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
		LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP									
40.0	Kugel (explizit)	6	5	5	5	6	6	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	
		6	5	5	5	6	6	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
		LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP									
40.0	Kugel (explizit)	6	5	5	5	6	6	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	
		6	5	5	5	6	6	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
		LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP									
40.0	Kugel (explizit)	6	5	5	5	6	6	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	
		6	5	5	5	6	6	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
		LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP									
40.0	Kugel (explizit)	6	5	5	5	6	6	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	
		6	5	5	5	6	6	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
		LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP									
40.0	Kugel (explizit)	6	5	5	5	6	6	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	
		6	5	5	5	6	6	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
		LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP									
40.0	Kugel (explizit)	6	5	5	5	6	6	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	
		6	5	5	5	6	6	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
		LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP									
40.0	Kugel (explizit)	6	5	5	5	6	6	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	
		6	5	5	5	6	6	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
		LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP	LP									

**Tabelle 10.IV Vergleich der Bogenlängenverfahren in Kombination mit dem Standard Newton Verfahren  $ds = \|u^{(i)}\|$ ;**

Toleranzen: Verschiebungen  $rtol = 10^{-1}$ ; Ungleichgewichtskräfte  $ftol = 10^{-1}$ ;

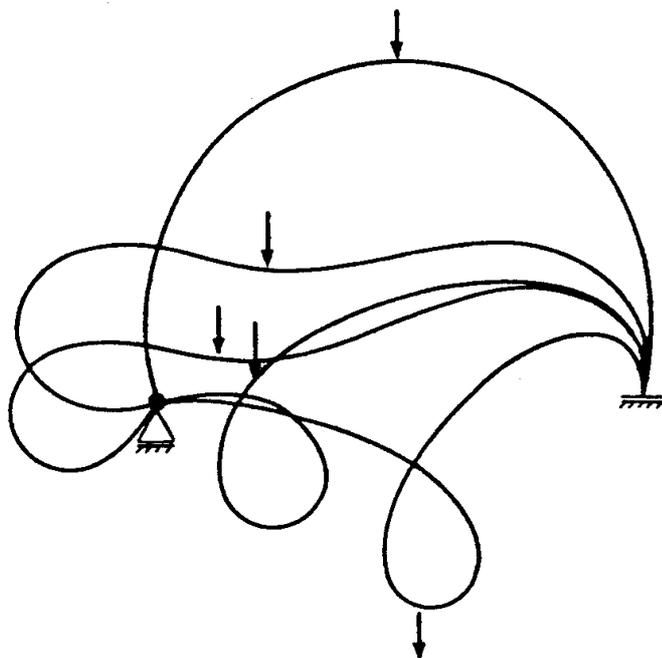
LP Durchschlagspunkt; BP unterer Umkehrpunkt; b siehe Bild 10.13

Für Schrittweiten mittlerer Größe ( $ds = 25$ ) ist fast kein Unterschied zwischen den Verfahren feststellbar. Wird die Schrittweite vergrößert ( $ds = 35$ ), so konvergiert nur noch Crisfields Verfahren mit der expliziten Erfüllung der Zusatzgleichung. Bei weiterer Vergrößerung der Schrittweite divergiert auch dieses Verfahren.

Strahlminimierung führte zu keiner Konvergenzverbesserung und konnte die Divergenz in keinem Fall verhindern.

## 10.7 Bogen unter Einzellast - Skalierungsfragen

Die Frage der Anpassung der Kurvenverfolgungsalgorithmen wurde am Beispiel des Bogens unter Einzellast untersucht (Bild 10.14) Für eine ausführliche Beschreibung des verwendeten Balkenelements sei auf /Simo et al 1984/ verwiesen. Das Verformungsverhalten bei variiertem Lastniveau ist in Bild 10.15 dargestellt.

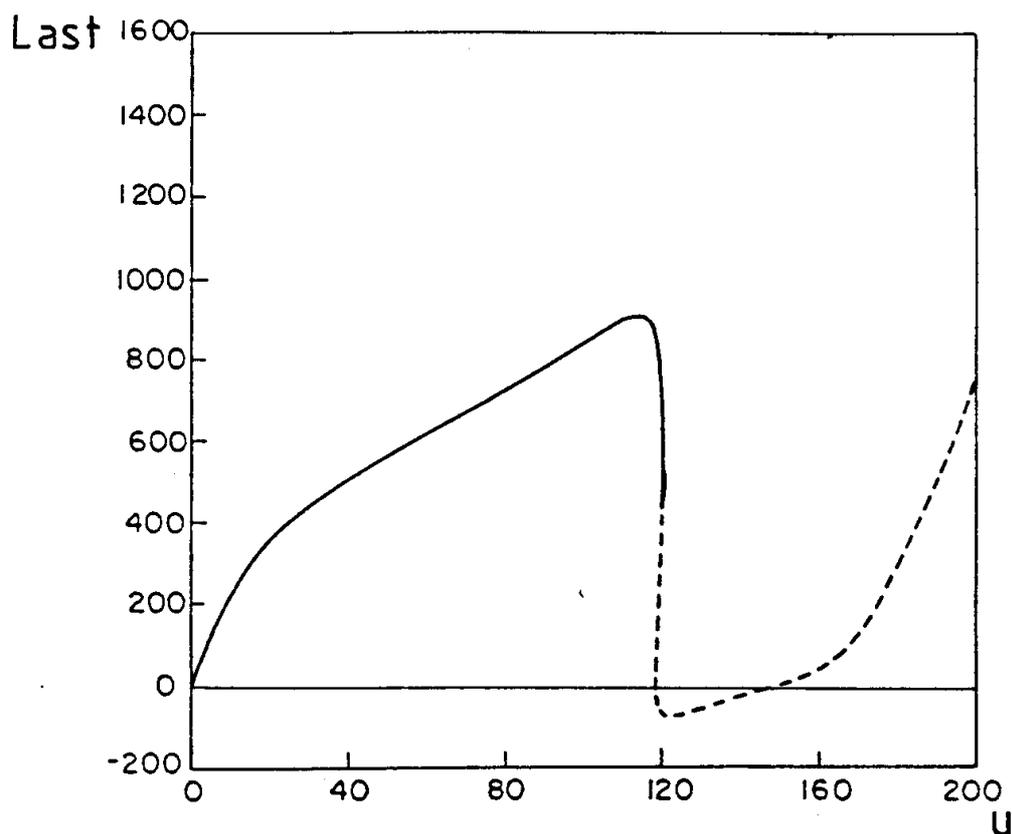


**Bild 10.14 Bogen unter Einzellast – Verformungsfiguren**

Die Anpassung des Algorithmus an das Verhalten des Systems wurde mit der Skalierung der linearen Zusatzgleichung (Iteration auf angepaßten Normalebenen) durchgeführt.

$$f = (\mathbf{x}^{(k)} - \mathbf{x})^t \cdot \mathbf{S} \cdot (\mathbf{x} - \mathbf{x}^{(k)}) + \beta^2 \mathbf{P}^t \mathbf{P} (\lambda^{(k)} - \lambda) (\lambda - \lambda^{(k)})$$

$\beta \dots$  Lastskalierungsfaktor



**Bild 10.15 Lastverschiebungsdiagramm Vertikalverschiebung**

Skaliert wurde mit einer Diagonalmatrix  $S$  mit

a)  $s_{ii} = \frac{1}{u_{ii}^{(k)^2}}$  Skalierung mit Referenzvektor  $u_{ii}^{(k)} \neq 0$

b)  $s_{ii,disp} = \frac{a}{\bar{u}^2}$   $s_{ii,rot} = \frac{b}{\bar{\varphi}^2}$   $a, b \dots$  Konstante  
 Skalierung mit Durchschnittswerten  $\bar{u}, \bar{\varphi}$

Für den Skalierungsfall a) wurde als Referenzvektor der Verschiebungsvektor des ersten linearen Schritts gewählt. Entsprechend wurde auch der Faktor  $\beta = \frac{1}{(\lambda^{(1)})^2 \cdot \mathbf{P}^T \mathbf{P}}$  bestimmt.

Als globales Resultat kann folgendes angegeben werden:

Während der Berechnung wurde eine zu starke Verkleinerung der Schrittgrößen beobachtet, die für einen effizienten Algorithmus um vieles zu klein waren. Dies war besonders ausgeprägt im Bereich des Durchschlagpunktes. Da diese Skalierung schwer zu steuern ist, aber große Adaptivität zeigt, kann sie nur für kritische Bereiche empfohlen werden. Für ein effizientes Verfahren sollte der Skalierungsvektor wieder mit einem Skalar angepaßt werden, damit in unterschiedlichen Bereichen optimale Schrittweiten möglich sind. Dem Anwender des Algorithmus bietet diese Skalierung viele Möglichkeiten; die sinnvollste Wahl ist jedoch problemabhängig.

Im Skalierungsfall b), bei dem mit den Mittelwerten des ersten linearen Schnitts skaliert wurde, bestimmen im wesentlichen die Größtwerte der Verschiebungen und der Rotationen die Schrittweite.

Variiert wurden noch die Faktoren  $a$ ,  $b$  und  $\beta$  mit den Werten 0 bzw. 1, so daß alle Formen der reinen oder gemischten Translations-, Rotations- und Lastkontrolle möglich waren.

Auch hier sei wegen der schwierigen Darstellbarkeit als Ergebnis nur folgendes gesagt:

Die Steuerung mit Rotationstermen alleine und auch in Kombination mit Lasttermen erbrachte die Lösung mit dem geringsten Aufwand, d.h. die optimale Kombination Schrittweite /Anzahl der Iterationen. Dies ist der Dominanz der Rotationen bei diesem Beispiel zuzuschreiben, die im Verhältnis zu den Verschiebungen größer sind.

Als Ergebnis dieser Skalierung kann festgehalten werden:

1. Zur Steuerung der Verfahren sind die Größen mit den prozentual größten Änderungen optimal; dann ist auch die Adaptivität der Verfahren sehr gut.
2. Im Hinblick auf einen effizienten Algorithmus bieten Mittelwerte einen Ausgleich zwischen großer Adaptivität und sinnvollen Schrittweiten.

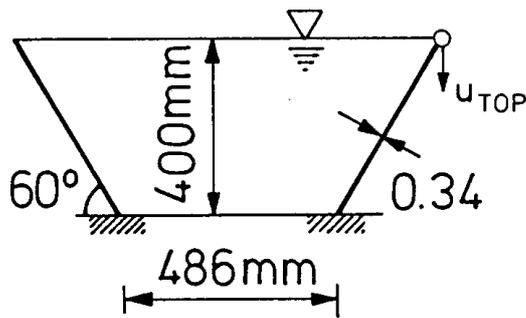
## 10.8 Kegelschale unter Innendruckbelastung

Zum Test der Quasi-Newton-Verfahren in Kombination mit den Bogenlängenverfahren an einem Beispiel mit größerer Unbekanntenzahl wurde das Durchschlagproblem der wassergefüllten Kegelschale (Bild 10.16a) gewählt. Das grundsätzliche Verhalten dieser Struktur wird ausführlich in /Stegmüller 1985/ beschrieben; das berechnete Lastverformungsverhalten ist in Bild 10.16b gegeben.

Das Finite Elementmodell stellt einen 15 Grad Ausschnitt aus der Schale dar. Die Geometrie wurde zur Erzeugung eines Durchschlagproblems mit einer kleinen Imperfektion in Form einer höheren, unsymmetrischen Eigenform beaufschlagt.

Als Kurvenverfolgungsalgorithmus wurde die lineare Zusatzgleichung mit Iteration auf angepaßten Normalenebenen gewählt.

Der Vergleich der Iterationsverfahren ist in Diagrammform (Bild 10.17) gegeben. Bei allen Verfahren wurde die Berechnung nur bis zum Überschreiten des Durchschlagpunktes durchgeführt.

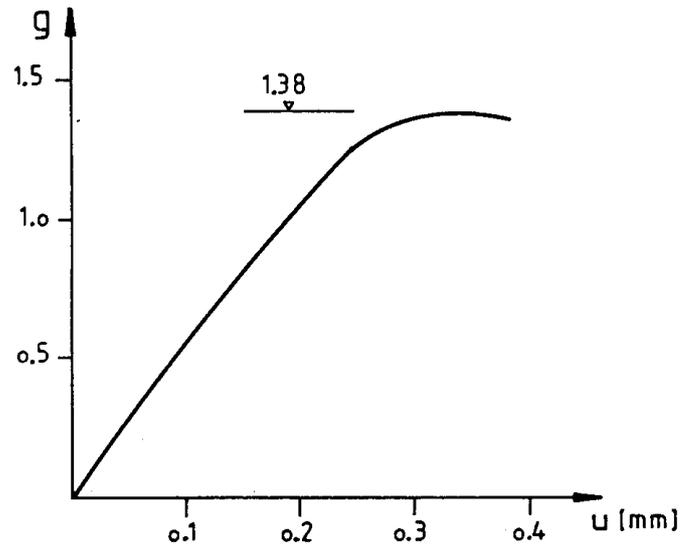
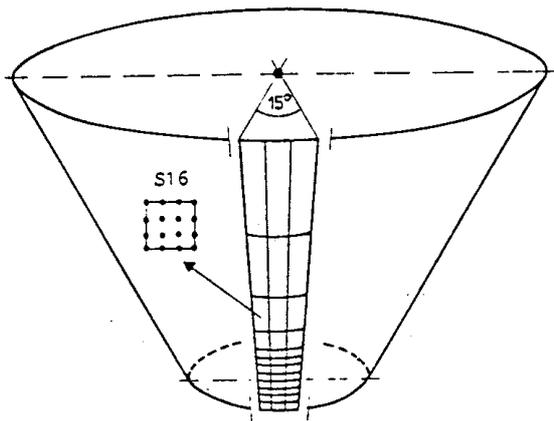


Kegelförmiger Behälter  
 $E = 4300 \text{ N/mm}$   $\mu = 0.3$

FE - Modell

Sektor mit unsymmetrischen  
 Imperfektionen

36 bikubische Schalenelemente  
 1656 Unbekannte



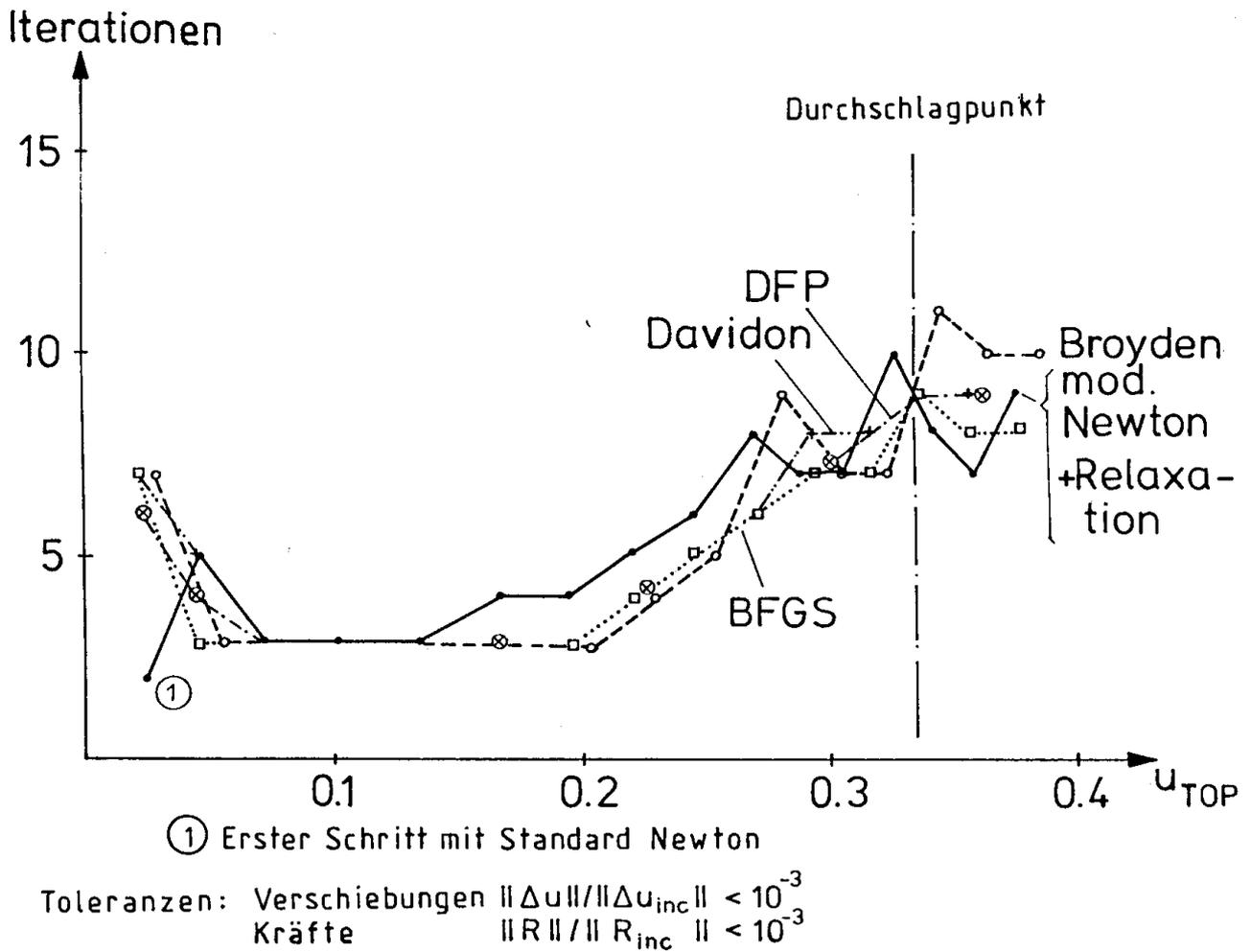
**Bild 10.16 Wassergefüllte Kegelschale**  
 a) Geometrie FE - Modell  
 b) Last - Verformungsdiagramm

Während das modifizierte Newton - Verfahren im ersten Schritt zu keiner Konvergenz führte und hierzu das echte Newton - Verfahren verwendet werden mußte, war die gesamte Berechnung der Last-Verformungskurve mit den Quasi-Newton-Verfahren problemlos möglich.

Beim modifizierten Newton-Verfahren war die Konvergenzbeschleunigung mit numerischer Relaxation ein wesentlicher Faktor. Die Strahlminimierung erbrachte bei allen Quasi-Newton-Verfahren innerhalb des berechneten Bereichs keine Verringerung der Iterationszahlen.

Der Vergleich der Rechenzeiten, der noch mit einer nicht optimalen Version der Kombination Quasi-Newton/Bogenlängenverfahren erfolgte, zeigt den Vorteil der Quasi-Newton Verfahren bezogen auf den Gesamtaufwand.

Weitere Untersuchungen wie Vergleiche bei größeren Schrittweiten und mit unterschiedlichen Line-Search Toleranzen unterblieben aus Kostengründen.



**Bild 10.17 Vergleich der Iterationsverfahren, Quasi Newton Verfahren und Kurvenverfolgungsalgorithmen**

# 11 Zusammenfassung

Die vorliegende Arbeit über Iterationsverfahren bei nichtlinearen Berechnungen in der Strukturmechanik weist vier Schwerpunkte auf:

1. Quasi-Newton Verfahren
2. Kurvenverfolgungsalgorithmen
3. Kombination von Quasi-Newton Verfahren und Kurvenverfolgungsalgorithmen
4. Strahlminimierung bei Kurvenverfolgungsalgorithmen

Neben einer umfassenden Schrifttumsdurchsicht wurden die Verfahren klassifiziert und geordnet. Wesentliche Neuentwicklungen waren:

1. Die Bereitstellung weiterer bekannter Quasi-Newton Algorithmen in Vektorform (Davidon, Broyden, DFP und BFGS modifiziert).
2. die Kombination der Quasi-Newton Algorithmen in Vektorform mit Kurvenverfolgungsalgorithmen.
3. die Bereitstellung von Strahlminimierungsalgorithmen für beliebige lineare und nichtlineare Nebenbedingungen.

Die entwickelten bzw. dargestellten Verfahren wurden an einer beschränkten Anzahl von Beispielen getestet. Für den beobachteten Problembereich können folgende Aussagen getroffen werden:

- i) Das Davidon- und das BFGS-Verfahren erwiesen sich als die am besten verwendbaren Quasi-Newton Verfahren. Dies betrifft sowohl die Konvergenzgeschwindigkeit als auch die Zahl der erforderlichen Rechenoperationen pro Schritt.
- ii) Das BFGS-Verfahren in der modifizierten Form ist das effizienteste der drei dargestellten Variationen der BFGS-Änderung der Systemmatrix auch für nicht positiv-definite Ausgangsmatrizen.
- iii) Crisfields Vorschlag der expliziten Iteration auf kugelförmigen Hyperflächen erwies sich als robustester Kurvenverformungsalgorithmus.

- iv) Die Kombination von Kurvenverfolgungsalgorithmen mit Quasi-Newton Verfahren erfordert nur wenig mehr Aufwand als die Kombination mit dem modifizierten Newton-Verfahren.
- v) Die Strahlminimierung führt zu einer Konvergenzbeschleunigung bei den hier betrachteten Verfahren und zu einer Konvergenzsicherung. Es ist von Vorteil, wenn Strahlminimierungsrichtung und Iterationsrichtung übereinstimmen.

Die Quasi-Newton Verfahren mit reduzierter Aufdatierung bedürfen noch einer Untersuchung hinsichtlich der Effizienz und Konvergenzgeschwindigkeit. Einige numerische Testprobleme zeigen ein beschleunigtes Konvergenzverhalten gegenüber den modifizierten Newton Verfahren und bestätigen die Ergebnisse in /Crisfield 1984/. Weitere sicher belegbare Aussagen für allgemeine Problemklassen können kaum getroffen werden.

### Offene Fragen und Ausblick

Die Quasi-Newton Verfahren lassen dann die beste Konvergenz erwarten, wenn die Änderungen der korrekten Matrix den vorgeschlagenen Rang-Eins bzw. Rang-Zwei Änderungen ähnlich sind. Z.B. könnte das Broyden-Verfahren für Probleme mit unsymmetrischen Matrizen (nicht assoziierte Fließgesetze, große Starrkörperrotationen) bessere Ergebnisse zeigen als bei den beobachteten Beispielen. Ähnliches kann für das DFP-Verfahren bei z.B. Kontaktproblemen, die zu Rang-Zwei Änderungen führen, erwartet werden.

Auch die Frage der optimalen Skalierung bei Kurvenverfolgungsalgorithmen kann nicht abschließend beantwortet werden. Dies liegt zum einen an der Vielzahl der Möglichkeiten und zum anderen an der Bindung an das zu lösende Problem. Für praktische Anwendungen können aus Gründen der Überschaubarkeit nur zwei-, maximal drei-parametrische Skalierungen empfohlen werden.

Eine weitere offene Frage ist die Kombination der Quasi-Newton Verfahren mit dem Newton-Verfahren z.B. in der Form, daß einige Iterationsschritte mit dem einen und die folgenden mit dem anderen Verfahren ablaufen oder daß z.B. ständig zwischen den Verfahren abgewechselt wird. Erste Untersuchungen /Gelin, Picart 1988/ führen zwar zu einigen ad-hoc Vorschlägen, die für praktische Berechnungen als erste Faustregel dienen können. Eine rigorose mathematische Untersuchung fehlt noch und ist auch angesichts der Vielfalt der möglichen Aufgaben kaum zu erwarten.

Ein sinnvoller Weg scheint daher zu sein, daß die Programme zur Berechnung nichtlinearer Probleme die wesentlichen Algorithmen beinhalten, und es dem Benutzer überlassen wird, wann und in welcher Kombination diese Algorithmen eingesetzt werden. Damit steht dem Benutzer offen, seine Erfahrungen für seine eigene Problemklasse optimal nutzen. Eine Benutzerunterstützung könnte z.B. durch wissensbasierte Systeme erfolgen.

## 12 Schrifttum

*Allgower, E.L.* "A survey of homotopy methods for smooth mappings", In Allgower, E.L.; Glashoff, K.; Peitgen, H.O. eds. "Numerical Solutions of Nonlinear Equations". Lecture Notes in Mathematics, Vol 878, 1-29, Springer, Berlin, 1981

*Allgower, E.L. Georg, K.* "Simplicial and continuation methods of approximating fixed points and solutions to systems of equations", SIAM Rev. 22, 28-85, (1980)

*Anselone, P., Moore, R.* "An extension of the Newton-Kantorowitch method for solving nonlinear equations with an application to elasticity", J. Math. Anal. Appl. 13 (1966), 476-501

*Bard, Y.* "On a numerical instability of Davidon-like methods", Math. of Comp., 22, 665-666, (1968)

*Bathe, K.J., Cimento A.P.* "Procedures for the solution of nonlinear finite element equations", Computer Meth. Applied Mech. Eng. 22 (1980) 59-85

*Bathe, K.J., Dvorkin, E.N.* "On the automatic solution of nonlinear finite element equations", Comp. Struct. 17 (1983), p. 871-879

*Bathe, K.J., Ramm, E. Wilson, E.L.* "Finite Element Formulation for Large Deformation Dynamic Analysis", Int. J. Num. Meth. Eng. 9 (1975)

*Bathe, K. J., Wilson, E. L.:* "Numerical methods in finite element analysis", Prentice Hall (1976)

*Batoz, J.L., Dhatt, G.* "Incremental displacement algorithms for non-linear problems", Internat. J. Numer. Meth. Eng. 14 (1979) 1262-1267

*Bergan, P.* "Solution algorithms for nonlinear structural problems", Comp & Structures 12, (1980) 497-509

*Bergan, P., Clough, R.W.* "Convergence Criteria for Iterative Processes", AIAA Journal 10 (1972) 1107/1108

*Bergan, P.G., Horrigmoe, G., Krakeland, B., Soreide, T.H.* "Solution techniques for non-linear finite element problems", Internat. J. Numer. Meth. Eng. 12 (1978) 1677-1696

*Branin, F.H.:* "Widely convergent method for finding multiple solutions of simultaneous nonlinear equations", IBM J. Research Develop 16. (1972) 504-522

*Brendel, B., Häfner, L., Sättele, J.M., Stegmüller, H., Ramm, E.* "Programmdokumentation und Eingabebeschreibung zum Programmsystem NISA",

Interne Dokumentation des Instituts für Baustatik, Universität Stuttgart, 1977, 1980, 1983

*Brodie, K., Gourlay, A. and J. Greenstadt* "Rank-one and rank-two corrections to positive definite matrices expressed in product form", *J. Inst. Math. Appl.* 11 (1973) 73-82

*Broyden, C.G.* "A class of methods for solving nonlinear simultaneous equations", *Math. Comp.* 19 (1965) 577-593

*Broyden, C.G.* "Quasi-Newton methods and their application to function minimization", *Math. Comp.* 21 (1967) 368-381

*Broyden, C.G.* "The convergence of a class of double-rank minimization algorithms", *J. Inst. Math. Appl.* 6 (1970) 76-90 and 222-231

*Crisfield, M.A.* "An arc-length method including line searches and accelerations", *Internat. J. Numer. Meth. Eng.* 19 (1983) 1269-1289

*Crisfield, M.A.* "A faster modified Newton-Raphson iteration", *Comp. Meth. Appl. Mech. Engng.* 20 (1979) 267-278

*Crisfield, M.A.* "A fast incremental/iterative solution procedure that handles snap through", *Computers & Structures*, 13 (1981) 55-62

*Crisfield, M.A.* "Solution procedures for non-linear structural analysis", in 'Recent advances in non-linear computational mechanics', Ed. C. Taylor, E. Hinton and D.R.J. Owen, 1-39, Pineridge Press, Swansea, 1982

*Crisfield, M.A.* "Accelerated solution techniques and concrete cracking", *Comp. Meth. in Appl. Mech. Engng.* 30 (1982) 585-607

*Crisfield, M.A.* "Accelerating and damping the modified Newton-Raphson method", *Computers & Structures* 18 (1984) 395-407

*Crisfield, M.A.* "Variable step-lengths for nonlinear structural analysis", TRRL Report 1049, Transport and Road Research Laboratory, Crowthorne, Berkshire (1982)

*Dahlquist, G. Björck, A.* "Numerical Methods", Prentice Hall, New Jersey, 1970

*Davidon, W.C.* "Variable metric method for minimization", Research and development report ANL-5990, U.S. Atomic Energy Commission, Argonne National Laboratories, 1959

*Davidon, W.C.* "Variance algorithm for minimization", *Computer J.* 10 (1968) 406-410

*Davidon, W.C.* "Optimally conditioned optimization algorithms without line searches", *Mathematical Programming* 9 (1975), 1-30

- Dennis, J.E., J.J. More* "Quasi-Newton methods, motivation and theory", SIAM Review 19 (1977) 46-89
- Endo, T.A., Oden, J.T., Becker, E.B., Miller, T.* "Numerical analysis of contact and limit-point behavior in a class of problems of finite deformations", Computers & Structures 18 (1984) 899-910
- Felippa, C.A.* "Dynamic relaxation under general increment control", in Innovative Methods for nonlinear problems", ed. Liu, Belytschko, Park, Pineridge Press 1984
- Fink, J.P., Rheinboldt, W.C.* "The role of tangent mapping in analyzing bifurcation behaviour", ZAMM 64 (1984) 407-412
- Fletcher, R., Powell M.J.D.* "A rapidly convergent descent method for minimization", Computer J. 6 (1963) 163-168
- Fletcher, R.* "A new approach to variable metric algorithms", Computer J. 13 (1970) 317-322
- Forde, B.W.R., Stiemer, S.F.* "Improved arc length orthogonality methods for nonlinear finite element analysis", Comp. Struct. 27 (1987), 625-630
- Geradin, M., Idelsohn, S. and M. Hogge* "Computational strategies for the solution of large nonlinear problems via Quasi-Newton methods", Computers & Structures 13 (1981) 73-81
- Gelin, J.C., Picart, P.* "Use of Quasi-Newton methods for large strain elastic-plastic finite element computations", Communications Appl. Num. Meth. 4 (1988) 457-469
- Gierlinski, J.T., Graves Smith, T.R.* "A variable load iteration procedure for thinwalled structures", Computers & Structures 21 (1985) 1085-1094
- Gierlinski, B.T., Knowles, N.C.* "Evaluation of incremental- iterative procedures for nonlinear structural problems", in "Quality assurance in FEM technology" ed. J. Robinson, 1985
- Goldfarb, D.* "A family of variable metric methods derived by variational means", Math. of Computation 24 (1970) 23-26
- Green, A.E., Naghdi, P.M.* "A general theory of an elasto-plastic continuum" Arch. Rational Mech. Anal. 18 (1965) 251-281
- Haisler, W.E., Stricklin, J.A., Key, J.E.* "Displacement incrementation in nonlinear structural analysis by the self-correcting method", Int.J.Num. Meth. Eng. 11 (1977) 3-10
- Hallquist, J. H.:* "Nike2D - A vectorized implicit, finite deformation finite element code for analyzing the static and dynamic response of 2-D solids

with interactive rezoning and graphics", Users manual, Lawrence Livermore Nat. Lab, California, Rep. UCID – 19677, 1986

*Holden, J.T.* "On finite deflections of thin beams", *Int. J. Solids Struct.* 8 (1972), 1051-1055

*Huang, H.Y.* "Unified approach to quadratically convergent algorithms for function minimization", *Journal of Optimization, Theory and Applications*, 5 (1970), 405-423

*Irons, B.M., Elsawaf, A.-F.* "The conjugate Newton algorithm for solving finite element equations", in "Formulations and computational algorithms in finite element analysis", eds. Bathe, Oden, Wunderlich, MIT press, 665-672 (1977)

*Kamat, M.P., Watson, L.T., Van den Brink, D.J.* "An assessment of Quasi Newton sparse update techniques for nonlinear structural analysis", *Comp. Meth. Appl. Mech. Eng.* 26 (1981), 363-375

*Keener, J.P., Keller, H.B.* "Perturbed bifurcation theory", *Arch. Rat. Mech. Anal.* 50 (1973) 159-175

*Keller, H.B.* "Global homotopies and Newton methods", in "Recent Advances in Numerical Analysis", ed. C. De Boer, G.H. Golub, Academic Press New York, 1978

*Keller, H.B.* "Practical procedures in path following near limit points", in "Computing Methods in Applied Sciences and Engineering", R. Glowinski and J.-L. Lions, eds., North-Holland, Amsterdam, 1982

*Kleinmichel, H.* "Quasi-Newton Verfahren vom Rang-Eins Typ zur Lösung unrestringierter Minimierungsprobleme", *Numer. Math.* 38 (1981), 219-228

*Kröplin, B.H., Dinkler, D.* "Quasi viscous analysis and arc-length- method. A comparison". NUMETA Conf., Swansea 1985 pp. 743-747

*Lee, S.L., Manuel, F.S., Rossow, E.C.* "Large deflection and stability of elastic frames", *ASCE J. Eng. Mech. Div.* 94 (1968) 521-533

*Luenberger, D.G.* "Linear and nonlinear programming", Addison-Wesley, Reading, MA, 1984

*Mahnken, R.* "Konvergenzuntersuchungen und Möglichkeiten der Konvergenzverbesserung bei einem Finite- Element-Programm zur geometrisch nicht-linearen Scheibenberechnung", Diplomarbeit, Institut für Baumechanik und Numerische Methoden, Hannover 1984

*Marsden, J.E., Hughes, T.R.J.* "Mathematical Foundations of Elasticity", Prentice Hall, New Jersey, 1982

- Matthies, H., Strang G.* "The solution of nonlinear finite element equations", *Int.J.Num.Meth.Eng.* 14 (1979) 1613-1626
- Matzenmiller, A.* "Ein rationales Lösungskonzept für geometrisch und physikalisch nichtlineare Strukturberechnungen" Diss. Institut für Baustatik, Universität Stuttgart, 1988
- Menzel, R., Schwetlick, H.:* "Zur Lösung parameterabhängiger Gleichungen mit singulären Jacobi Matrizen", *Numer. Math.* 80 (1978) 65-79
- Murray, W.* "Numerical methods for unconstrained optimization", Academic Press, London 1972
- Nazareth, L.* "A relationship between the BFGS and conjugate gradient algorithms and its implications for new algorithms", *SIAM J.Numer.Anal.* 16 (1979) 794-800
- Nazareth, L.* "On the BFGS method part I: Properties of the standard algorithm", Dept. Industrial Eng. Operations Research, University of California, Berkeley, California, 1981
- Nazareth, L.* "On the BFGS method part II: Extended and limited memory versions", Dept. Industrial Eng. Operations Research, University of California, Berkeley, California, 1981
- Nocedal, J.* "Updating Quasi-Newton matrices with limited storage", *Math. of. Computation*, 35 (1980) 773-7782
- Oren, S.S., Luenberger, D.G.* "Self-scaling variable metric (SSVM) algorithms. Part I: Criteria and sufficient conditions for scaling a class of algorithms", *Management Sci.* 20 (1974) 845-862
- Oren, S.S.* "Self-scaling variable metric (SSVM) algorithms. Part II: Implementation and experiments", *Management Sci.* 20 (1974) 863-874
- Oren, S.S., Spedicato, E.* "Optimal conditioning of self-scaling variable metric algorithms", *Math. Programming* 10 (1976) 70-90
- Ortega, I.M., Rheinboldt, W.C.* "Iterative Solutions of Nonlinear Equations in Several Variables", Academic Press, New York and London, 1970
- Padovan, J.* "Solving postbuckling collapse of structures", *Finite Element Analysis Design* 1 (1985) 363-385
- Padovan, J., Arechaga, T.* "Formal convergence characteristics of elliptically constrained incremental Newton-Raphson algorithms", *Int. J. Eng. Sci.* 20 (1982) 1077-1097
- Padovan, J., Tovichakchaikul, S.* "Self-adaptive predictor-corrector algorithm for static nonlinear structural analysis", *Comp. Struct.* 15 (1982) 365-377

- Padovan, J., Tovichakchaikul, S.* "Algorithms for elasto-plastic creep post-buckling", *J. Eng. Mech.* 110 (1984) 911-929
- Park, K.C.* "A family of solution algorithms for nonlinear structural analysis based on relaxation equations", *Int. J. Num. Meth. Eng.* 18 (1982), 1337-1347
- Pearson, J.D.* "Variable metric methods of minimization", *Computer J.* 12 (1969) 171-178
- Pönisch, G., Schwetlick, H.* "Ein lokal überlinear konvergentes Verfahren zur Bestimmung von Rückkehrpunkten implizit definierter Raumkurven", *Numer. Math* 38 (1982) 455-466
- Powell, G., Simons, I.* "Improved iteration strategies for nonlinear structures", *Internat. J. Numer. Meth. Eng.* 17 (1981) 1455-1467
- Powell, M.J.D.* "Rank-one methods for unconstrained optimization", In 'Integer and nonlinear programming', Ed. J. Abadie, 139-156, North-Holland, Amsterdam 1970
- Powell, M.J.D.* "A hybrid method for nonlinear equations." In: 'Numerical Methods for Non-linear Algebraic equations', P. Rabinowitz ed., Gordon and Breach, London, 1970
- Powell, M.J.D.* "A new algorithm for unconstrained optimization". In "Non-linear Programming", J.B. Rosen, O.L. Mungasarian, K. Ritter eds. Academic Press, New York, 1970
- Powell, M.J.D.* "Some global convergence properties of a variable metric algorithm for minimization without exact line search", *SIAM-AMS Proceedings*, Vol 9, 53-71 (1976)
- Ramm, E.* "Strategies for tracing the nonlinear response near limit points", in: "Europe-US-Workshop on Nonlinear Finite Element Analysis in Structural Mechanics", K.J. Bathe, E. Stein and W. Wunderlich, eds., Ruhr Universität Bochum, Germany, (Springer, Berlin, 1980, 63-89)
- Ramm, E., Matzenmiller, A.* "Large deformation shell analysis based on the degeneration concept". In "Finite Element Methods for Plate and Shell Structures" T.J.R. Hughes, E. Hinton eds., Pineridge Press, 1986, 365-393
- Ramm, E., Matzenmiller, A.* "Computational aspects of elasto - plasticity in shell analysis", *Conf. COMPLASI*, Barcelona, 1987
- Rheinboldt, W.C.* "Numerical analysis of continuation methods for nonlinear structural problems", *Comput. & Structures* 13 (1981) 103-113
- Riks, E.* "The application of Newton's method to the problem of elastic stability", *Journal Appl. Mech. Trans. ASME*, 39 (1972) 1060-1066

- Riks, E.* "An incremental approach to the solution of snapping and buckling problems", *Internat. J. Solids and Structures* 15 (1979) 529-551
- Riks, E.* "Progress in collapse analysis", Rept. NLRMP 84031U, National Aerospace Laboratory, Amsterdam, 1984
- Riks, E.* "Some computational aspects of the stability analysis of nonlinear structures", *Comput. Meths. Appl. Mech. Eng.* 47 (1984) 219-259
- Runesson, K., Samuelsson, A., Bernspang, L.* "Numerical techniques in plasticity including solution advancement control", *Int. J. Num. Meth. Eng.* 22 (1986) 769-788
- Runxiang, Z., Parisch, H.* "A study of the different solution methods for nonlinear problems", Institut für Statik und Dynamik der Luft- und Raumfahrt-konstruktionen, University of Stuttgart, ISD - Report No.304, 1982
- Sättele, J.M.* "Ein finites Elementkonzept zur Berechnung von Platten und Schalen bei stofflicher und geometrischer Nichtlinearität", Diss. 1980, Universität Stuttgart
- Schmidt, W.F.* "Adaptive step size selection for use with the continuation method", *Int. J. Num. Meth. Eng.* 12 (1978) 677-694
- Schwarz, H.R., Rutishauser, H., Stiefel, E.* "Numerik symmetrischer Matrizen", Teubner, Stuttgart 1968
- Schweizerhof, K., Ramm, E.* "Displacement dependent pressure loads in nonlinear finite element analysis", *Comput. & Structures* 18 (1984) 1099-1114
- Schweizerhof, K., Wriggers, P.* "Consistent linearization of path following methods in nonlinear FE analysis", *Comp. Meth. Appl. Mech. Eng.* 59 (1986) 261-279
- Schwetlick, H.* "Numerische Lösung nichtlinearer Gleichungen", Deutscher Verlag der Wissenschaften, Berlin 1979
- Schwetlick, H.* "On the choice of step length in path following methods", *ZAMM* 64 (1984) 391-396
- Schwetlick, H.* "Algorithms for finite-dimensional turning point problems from viewpoint to relationships with constraint optimization methods", *Internat. Ser. Numer. Math.* 70 (1984) 459-479
- Shanno, D.F.* "Conditioning of Quasi-Newton methods for function minimization", *Math. Computation* 24 (1970)
- Sherman, I., Morrison, W.I.* "Adjustment of an inverse matrix corresponding to changes in the elements of a given column or a given row of the original matrix", *Ann. Math. Statist.* 20 (1947) 621 ff.

- Simo, J.C., Taylor, R.L.* "Consistent tangent operators for rate independent elastoplasticity", *Comp. Meth. Appl. Mech. Eng.* 48 (1985) 101-118
- Simo, J.C., Wriggers, P., Schweizerhof, K., Taylor, R.L.* "Finite deformation postbuckling analysis involving inelasticity and contact constraints", in "Innovative Methods for Nonlinear Problems" ed. Liu, Belytschko, Park, 345-363, Pineridge Press, 1984
- Simons, I., Bergan, P.G., Nygard, M.K.* "Hyperplane displacement control methods in nonlinear analysis", in *Proc. Int. Conf. "Innovative Methods for Nonlinear Problems"* ed. Liu, Belytschko, Park, Pineridge Press, Swansea, UK, 1984
- Stegmüller, H.* "Grenzlastberechnungen flüssigkeitsgefüllter Schalen mit degenerierten Schalenelementen", *Diss., Universität Stuttgart* 1985
- Stoer, J.* "The convergence of matrices generated by rank-2 methods from the restricted  $\beta$ -class of Broyden", *Numer. Math.* 44, 37-52 (1984)
- Taylor, R. L.* "FEAP Finite Element Analysis Program", Vers. 1985, also in. *Zienkiewicz, O.C. "The Finite Element Method"*, Mc Graw Hill, 1977
- Waszczyszyn, Z.* "Numerical problems of nonlinear stability analysis of elastic structures", *Comp. Structures* 17 (1983) 13-24
- Watson, L.T., Holzer, S. M.* "Quadratic convergence of Crisfield's method", *Comp. & Structures* 17 (1983) 69-72
- Wempner, G.A.* "Discrete approximations related to nonlinear theories of solids", *Int. J. Solids. Struct.* 7 (1971) 1581-1599
- Wessels, M.* "Das statische und dynamische Durchschlagsproblem der imperfekten flachen Kugelschale bei elastischer rotationssymmetrischer Verformung", *Diss., Universität Hannover* 1977
- Wolfe, M.A.* "Numerical methods for unconstrained optimization. An introduction", *Van Nostrand /Rheinholdt Co, London* 1975
- Wriggers, P., Wagner, W., Miehe, C.* "A quadratically convergent procedure for the calculation of stability points in finite element analysis", *IBNM Bericht 88/1, Institut für Baumechanik und Numerische Mechanik, Universität Hannover*, 1988
- Zielke, G.* "Inversion of modified symmetric matrices", *J. Assoc. Comput. Mach.* 15 (1968) 402-408
- Zielke, G.* "Numerische Berechnung von benachbarten inversen Matrizen und linearen Gleichungssystemen", *Vieweg und Sohn, Braunschweig*, 1970

# 13 Anhang

## A. Sherman/Morrison Formel

Die Beobachtung von /Sherman, Morrison 1947/, daß sich die Inverse einer mit einer um den Rang-Eins veränderten Matrix in einfacher Form schreiben läßt, bildet die eigentliche Grundlage zum Erfolg des Quasi-Newton Verfahrens in den Ingenieurwissenschaften.

$$(\mathbf{A} + \alpha \mathbf{u} \mathbf{v}^t)^{-1} = \mathbf{A}^{-1} - \beta \mathbf{x} \mathbf{z}^t \quad (\text{A.1})$$

$$\mathbf{x} = \mathbf{A}^{-1} \mathbf{u} \quad (\text{A.2})$$

$$\mathbf{z} = \mathbf{A}^{-1} \mathbf{v} \quad (\text{A.3})$$

$$\beta = \frac{\alpha}{1 + \alpha \cdot \mathbf{v}^t \mathbf{A}^{-1} \mathbf{u}} \quad (\text{A.4})$$

$\mathbf{A}$  ... Systemmatrix

$\mathbf{u}, \mathbf{v}$ ... Änderungsvektoren

$\alpha$  .. beliebiger Faktor

In /Zielke 1970/ wird neben einer interessanten historischen Betrachtung auch der breite Anwendungsbereich basierend auf der Grundidee von Sherman/Morrison gezeigt.

## B. Rang-Eins Änderung nach Kleinmichel

/Kleinmichel 1981/ schlägt eine Rang-Eins Änderung vor, die sowohl symmetrische als auch positiv-definitive Matrizen erzeugt. Entwickelt aus der Oren/Luenberger Klasse /Oren, Luenberger 1970/ ergibt sich:

$$\mathbf{K}^{(k)-1} = \frac{\alpha}{\gamma} \left[ \mathbf{K}^{(k-1)-1} - \frac{(\mathbf{K}^{(k-1)-1} \mathbf{y}^{(k)})(\mathbf{K}^{(k-1)-1} \mathbf{y}^{(k)})^t}{\mathbf{y}^{(k)t} \mathbf{K}^{(k-1)-1} \mathbf{y}^{(k)} + \gamma \mathbf{y}^{(k)t} \Delta \mathbf{u}^{(k)} / \alpha} \right] \quad (B.1)$$

$\alpha$  ist der Strahlminimierungsfaktor (line-search faktor).

$\gamma$  ist ein freier Parameter zur Skalierung.

Vorschlag:  $\gamma = 0.5$

$\mathbf{y}^{(k)}$  ist definiert als

$$\mathbf{y}^{(k)} = \Delta \mathbf{r}^{(k)} - \gamma \mathbf{r}^{(k-1)} = (1 - \gamma) \mathbf{r}^{(k-1)} - \mathbf{r}^{(k)} \quad (B.2)$$

$\mathbf{d}^{(k)}$  ist definiert als

$$\mathbf{d}^{(k)} = \frac{\Delta \mathbf{u}^{(k)}}{\alpha} \quad (B.3)$$

Bei genügend genauer Strahlminimierung ist die obige Matrix positiv definit und die Inversion ist möglich. Die ursprüngliche Matrix lautet dann:

$$\mathbf{K}^{(k)} = \frac{\gamma}{\alpha} \left[ \mathbf{K}^{(k-1)} + \frac{\mathbf{y}^{(k)} \mathbf{y}^{(k)t}}{\gamma \mathbf{y}^{(k)t} \Delta \mathbf{u}^{(k)}} \right] \quad (B.4)$$

Voraussetzung hierfür ist außerdem, daß  $\mathbf{r}^{(k-1)} \neq 0$  ist.

Das wesentliche Problem dieser Aufdatierungsformel ist die geforderte Genauigkeit der Strahlminimierung, die jeweils zu überprüfen ist. Hierbei muß die Strahlminimierung so genau erfolgen, daß

$$\mu^{(k)} = \frac{\tilde{\alpha} - \alpha^{(k)}}{\tilde{\alpha}} < tol \quad (B.5)$$

erfüllt ist.  $\tilde{\alpha}$  ist der Strahlminimierungswert bei exakter Minimierung.

Nach /Kleinmichel 1981/ muß der relative Fehler der folgenden Bedingung genügen:

$$0 \leq \mu^{(k)} \leq \mu < 1 - \gamma \quad (B.6)$$

Im vorgeschlagenen Fall

$$0 \leq \mu^{(k)} \leq \frac{1}{2} \quad (B.7)$$

Dies kann in den meisten Fällen mit Sicherheit garantiert werden.

## C. Quasi viskose Dämpfung nach Kröplin

### Standardform

In einer Vielzahl von Veröffentlichungen schlagen Kröplin et al, siehe auch /Kröplin, Dinkler 1985/ als alternatives Verfahren zur Bestimmung nichtlinearer Gleichgewichtszustände die sogenannte -" Quasi viskose Dämpfung" vor. Zur numerischen Stabilisierung der Lösung wird ähnlich zur Dynamischen Relaxation ein geschwindigkeitsabhängiger Term zum nichtlinearen Gleichungssystem addiert, der in der Nähe der Lösung des Ausgangsgleichungssystems verschwindet.

Wird von einem Newton-Verfahren ausgegangen, so lauten die iterativen Gleichungen in der Strukturmechanik:

Ausgangssystem:

$$\mathbf{K}(\mathbf{x}^{(i)})\Delta\mathbf{x}^{(i)} = -\mathbf{F}(\mathbf{x}^{(i)}) \quad (\text{C.1})$$

Im weiteren:

$$\mathbf{K}^{(i)} \equiv \mathbf{K}(\mathbf{x}^{(i)}) \quad \mathbf{F}^{(i)} \equiv \mathbf{F}(\mathbf{x}^{(i)})$$

Addition des geschwindigkeitsabhängigen Terms:

$$\mathbf{K}^{(i)}\Delta\mathbf{x}^{(i)} + \mathbf{D}\dot{\mathbf{x}}^{(i)} = -\mathbf{F}^{(i)} \quad (\text{C.2})$$

Wird für die Verschiebung ein linearer Verlauf über die Zeit angenommen, so gilt:

$$\mathbf{x}(t) = \left(1 - \frac{t}{\Delta t}\right)\mathbf{x}^{(i-1)} + \frac{t}{\Delta t}\mathbf{x}^{(i)} = \frac{t}{\Delta t}\mathbf{x} + c \quad (\text{C.3})$$

$$\dot{\mathbf{x}}(t) = \frac{1}{\Delta t}(\mathbf{x}^{(i)} - \mathbf{x}^{(i-1)}) = \frac{\Delta\mathbf{x}}{\Delta t} \quad (\text{C.4})$$

Gl. (C.1) eingesetzt in Gl. (C.2) ergibt:

$$[\mathbf{K}^{(i)} + \frac{1}{\Delta t}\mathbf{D}]\Delta\mathbf{x}^{(i)} = -\mathbf{F}^{(i)} \quad (\text{C.5})$$

Als Matrix für den Dämpfungsterm wird eine Diagonalmatrix  $\mathbf{D}^* = \frac{1}{\Delta t} \mathbf{D}$  vorgeschlagen.

$$D_{jj}^{*(i)} = \text{abs}\left[\frac{F_j^{(i)}}{\Delta x_j^{(i)}}\right] \quad (\text{C.6})$$

Um die Division durch Null zu vermindern, muß jede Komponente  $\Delta x_j^{(i)}$  des Vektors  $\Delta \mathbf{x}^{(i)}$  größer als eine vorgegebene Toleranz sein.

$$\Delta x_j^{(i)} > \epsilon_{\text{tol}}$$

Diese künstliche Änderung mit einer Diagonalmatrix entspricht dem physikalischen Einbau von ungekoppelten Einzeldämpfern an jedem Freiheitsgrad. Dabei muß in jedem Schritt die Hessematrix  $\mathbf{K}(\mathbf{x}^{(i)})$  des Systems um diese Diagonalterme verändert und für die Gleichungslösung neu faktorisiert werden.

### Vektorisierte Form

Die hier im weiteren vorgestellte Variation des Verfahrens vermeidet dieses Problem und erlaubt sogar die Verwendung der unveränderten Matrix  $\mathbf{K}(\mathbf{x}^{(k)})$  mit  $(k \neq i)$  im Iterationsschritt  $i$  im Sinne eines Modifizierten Newton Verfahrens.

Es wird als Matrix des Dämpfungsterms eine Matrix vom Rang-Eins gewählt:

$$\tilde{\mathbf{D}} = \mathbf{y}\mathbf{y}^t \quad (\text{C.7})$$

Die Diagonalterme von  $\tilde{\mathbf{D}}$  sollen identisch mit den Diagonaltermen von  $\mathbf{D}^*$  sein:

$$\tilde{D}_{jj}^{(i)} = D_{jj}^{*(i)} = \frac{F_j^{(i)}}{\Delta x_j^{(i)}} \quad \Delta x_j^{(i)} > \epsilon \quad (\text{C.8})$$

Damit wird

$$\mathbf{y}_j^{(i)} = \sqrt{\frac{|F_j^{(i)}|}{|\Delta x_j^{(i)}|}} \quad (\text{C.9})$$

Durch die einfache Form der Rang-Eins Änderung ergibt sich eine effiziente Lösungsmöglichkeit mit Anwendung der Sherman/Morrison Formel:

$$[\mathbf{K}^{(k)} + \mathbf{y}^{(i)}\mathbf{y}^{(i)t}]^{-1} = \mathbf{K}^{(k)-1} - \beta\mathbf{z}\mathbf{z}^t \quad (C.10)$$

mit

$$\begin{aligned} \mathbf{z} &= \mathbf{K}^{(k)-1}\mathbf{y}^{(i)} \\ \beta &= \frac{1}{(1 + \mathbf{y}^{(i)t}\mathbf{K}^{(k)-1}\mathbf{y}^{(i)})} \end{aligned}$$

Damit gilt:

$$\Delta\mathbf{x}^{(i)} = \mathbf{K}^{(k)-1}(-\mathbf{F}^{(i)}) + \beta\mathbf{z}\mathbf{z}^t\mathbf{F}^{(i)} \quad (C.11)$$

Bei dem geschilderten Vorgehen müssen innerhalb einer direkten Gleichungslösung zwei rechte Seiten berücksichtigt werden.

Gesamtlösung:

$$\begin{aligned} \Delta\mathbf{x}^{(i)} &= \Delta\mathbf{x}^{(i)I} + \Delta\mathbf{x}^{(i)II} \\ \Delta\mathbf{x}^{(i)I} &= -\mathbf{K}^{(k)-1}\mathbf{F}^{(i)} \\ \mathbf{z}^{(i)} &= \mathbf{K}^{(k)-1}\mathbf{y}^{(i)} \\ \Delta\mathbf{x}^{(i)II} &= \frac{1}{(1 + \mathbf{y}^{(i)t}\mathbf{z}^{(i)})}\mathbf{z}^{(i)t}\mathbf{F}^{(i)}\mathbf{z}^{(i)} \end{aligned} \quad (C.12)$$

Der Mehraufwand für das Vorwärts-/Rückwärtseinsetzen und zwei Skalarmultiplikationen von Vektoren und eine Vektoraddition ist verglichen mit einer Standardlösung gering. Außerdem wird zusätzlicher Speicherplatz für nur einen Vektor benötigt.

Der Vorteil gegenüber dem in /Kröplin, Dinkler 1985/ vorgeschlagenen Vorgehen ist offensichtlich:

1. Die Originalinformation über die Steifigkeitsmatrix wie Determinante, Eigenwerte  $< \text{null}$  etc. ist für jeden Zwischenzustand verfügbar.
2. Es kann das Modifizierte Newton Verfahren verwendet werden.
3. Der Dämpfungseinfluß kann genau beobachtet werden.
4. Innerhalb eines bestehenden Rechenprogramms müssen nur in geringstem Maße Änderungen vorgenommen werden.