

Dr.-Ing. F. Buchhardt
Dr.-Ing. K. Brandes

Methodische Zusammenführung von Strukturanalyse und Messung

mit Anhang A: Spezifische Beispiele
mit Anhang B: Matrizen-Übersicht und Lösungsweg

Impressum

Forschungsbericht 286:
**Methodische Zusammenführung von
Strukturanalyse und Messung**

2009

Herausgeber:
BAM Bundesanstalt für Materialforschung und -prüfung
Unter den Eichen 87
12205 Berlin
Telefon: +49 30 8104-0
Telefax: +49 30 8112029
E-Mail: info@bam.de
Internet: www.bam.de

Copyright © 2009 by
BAM Bundesanstalt für Materialforschung und -prüfung

ISSN 0938-5533
ISBN 978-3-9812910-0-1

Vorbemerkung

Mit Carl Friedrich Gauß' „Theorie der Bewegung der Himmelskörper, welche in Kegelschnitten die Sonne umlaufen“, 1809 (in lateinischer Sprache), beginnt die systematische Beschäftigung mit der Auswertung von Daten aus Beobachtungsreihen. Gauß selbst benutzte schon die Bezeichnung „Methode der kleinsten Quadrate“. Dass die Methode von Gauß dann auch als Variationsmethode begriffen wurde, die mit etwas Vorlauf von Lagrange entwickelt worden war, eröffnete schon kurze Zeit später die Möglichkeit, über Nebenbedingungen zusätzliche Forderungen in mathematischer Form einzuführen. Gleichungen, mit Lagrange'schen Multiplikatoren versehen, erweiterten die Anwendbarkeit ungemein. Auch auf dem Feld der Ausgleichsrechnung ist dieses Verfahren immer wieder einmal eingesetzt worden /10/.

In großem Umfange sind dann die erwähnten Methoden in der Statistik eingeführt worden, vor allem seitdem die programmgesteuerten elektronischen Rechenanlagen die Behandlung sehr großer Datenmengen möglich machten.

Die Verbindung der Methode der kleinsten Quadrate mit Nebenbedingungen ist zunächst erfolgreich vor allem in der Ökonometrie und in der Biometrie eingesetzt worden /1/. Auf diesen Gebieten ist auch die besondere Bedeutung und Interpretierbarkeit der Lagrange'schen Multiplikatoren als Indikator für die Güte der Übereinstimmung von Modell und Messergebnissen herausgearbeitet worden, womit ein weiteres Element der Analyse verfügbar wurde. Allerdings erst in den 1980er Jahren sind diese Verfahren auch für die experimentelle Untersuchung von Konstruktionen hin und wieder eingesetzt worden, wobei erwähnenswert ist, dass die Nebenbedingungen, die aus Struktursystemen folgen, vergleichsweise komplex sein können.

In diesem Zusammenhang kommt den Materialprüfungsanstalten eine besondere Rolle zu; denn dort ist sowohl die Sensorik (Entwicklung von Messwert-Aufnehmern) als auch die experimentelle Untersuchung des mechanisch-technologischen Verhaltens von Konstruktionen beheimatet, und diese Institutionen unterliegen in besonderem Maße der Neutralität. Auf diese Besonderheit und Aufgabe hat u. a. Adolf Martens 1904 in einer Denkschrift zur Eröffnung des Königlichen Materialprüfungsamtes auf dem Gelände in Berlin-Dahlem hingewiesen („woraus später die BAM entstand). Vergleichbare Gedanken hat schon 1896 G. C. Mehrrens in der Zeitschrift Stahl und Eisen geäußert, der Professor an der TH Dresden war und dort das außerordentlich renommierte Otto-Mohr-Laboratorium leitete.

In der BAM ist der so gestalteten Kopplung von Experiment und analytisch-numerischer Behandlung von Problemen der hier angesprochenen Art erst relativ spät Aufmerksamkeit gewidmet worden, vor allem auch im Zusammenhang mit der Strukturanalyse des kerntechnischen Ingenieurbaus, die mit dem Eintritt von Thomas A. Jaeger in die BAM 1968 begann.

Einen neuen Impuls empfangen diese Arbeiten jüngst mit der Aufgabe, das langzeitige Verhalten der Spannbetonbrücken des neuen Hauptbahnhofs in Berlin (zunächst noch als "Lehrter Bahnhof" bezeichnet) zu beobachten (Langzeit-Monitoring). Es stellte sich heraus, dass über das zuvor bereits Geleistete hinaus noch erhebliche Anstrengungen erforderlich waren, die mathematischen und mechanischen Grundlagen des Verfahrens in umfassender Weise zusammenzuführen. Dass zeitgleich mit der Aufgabe der Langzeitbeobachtungen der Brücken am Hauptbahnhof Berlin Studien ähnlicher Zielrichtung für die zerstörungsfreie Prüfung in Angriff genommen worden sind - einige davon in der BAM - sei zur Vervollständigung des Bildes hier erwähnt /11/,/27/.

Die zum Ende des 20. Jahrhunderts entwickelten und von Instituten und Firmen angebotenen Verfahren zur Langzeitüberwachung der Sicherheit von Konstruktionen waren zuweilen nicht immer geeignet, der Aufgabe gerecht zu werden, und erst langsam beginnt ein Konsolidierungs-Prozess in dieser Frage.

Der vorliegende Bericht soll einige Grundlagen für die hier angesprochenen Verfahren auf eine rationale Basis stellen und die notwendige Diskussion anregen. In einem weiteren Bericht wird dann die spezifische Anwendung auf die Untersuchungen an den Brücken des Hauptbahnhofs Berlin vorgelegt werden.

Inhalt

Kurzfassung	6
1 Einführung	7
2 Allgemeine Ableitung	8
2.1 Grundsätzliche Beziehungen	8
2.2 Fehlerausgleich mit diskreten Messwerten allein	9
2.3 Fehlerausgleich mit Zusatzbedingungen	10
3 Typ der Ansatzfunktion	11
3.1 Hermite-Polynome	11
3.2 Bernoulli-Ansatz zur Balkenbiegung	12
4 Formulierung der Gesamtmatrizen	15
4.1 System- und Lastmatrizen aus Durchbiegungsmessungen	15
4.2 System- und Lastmatrizen aus Winkelneigungsmessungen	16
4.3 System- und Lastmatrizen aus Dehnungsmessungen	17
5 Berücksichtigung des mechanischen Systems mittels Lagrange'scher Multiplikatoren	19
5.1 Randbedingungen	19
5.2 Innere Kopplung	20
5.3 Äußere Kopplung	21
6 Lösungswege zur Verbindung von Messung und Strukturanalyse	22
6.1 Integrale Formulierung	22
6.2 Transformation in virtuelle Durchbiegungen	23
6.2.1 Virtuelle Durchbiegungen aus Winkelneigungsmessungen	24
6.2.2 Virtuelle Durchbiegungen aus Dehnungsmessungen	25
6.2.3 Residuale Lösung	26
7 Grundsätze zur Auswahl der Messpunkte	27
7.1 Prinzip von St. Vénant	27
7.2 Konstruktionsart	27
7.2.1 Zwängungsfreie Biegung	27
7.2.2 Biegung mit innerem Zwang	28
7.3 Belastungsgeschichte	30
8 Grundlegende Beispiele	31
8.1 Beispiel für zwängungsfreie Biegung	31
8.2 Beispiel für Biegung mit innerem Zwang	37
8.2.1 Fall 1: Volle Vorspannung	38
8.2.2 Fall 2: Teilweise Vorspannung	39
8.2.3 Fall 3: Überhöhte Vorspannung	39
9 Fazit und Ausblick	42
10 Literaturhinweis	43
11 Anhang A: Spezifische Beispiele	45
12 Anhang B: Matrizen-Übersicht und Lösungsweg	77

Kurzfassung

Ziel der Untersuchung ist, eine methodische Verbindung zwischen statischer Tragwerksanalyse und Messung am Bauwerk herzustellen.

Die grundsätzlichen Beziehungen werden in Verbindung mit der zusätzlichen Einführung der Lagrange'schen Multiplikatoren hergeleitet; ebenso wird auf einen Sonderfall hingewiesen. Am Beispiel des Linien-tragwerkes - der Balkenbiegung - wird der Ansatz nach Bernoulli als Regressionsfunktion gewählt. Ein Vergleich zu Hermite-Polynomen wird gezogen, die hier keine optimale Lösung erbringen können.

Für die 3 Messgrößen - Durchbiegung w sowie Winkelneigung w' und Krümmung w'' bzw. Dehnung ε - werden die grundsätzlichen Matrizen hergeleitet. Für die Forderung, dass die 3 Messgrößen grundsätzlich in jeder Messreihe auftreten, wird deren Kopplung ausgewiesen, ebenso die Berücksichtigung vorhandener statischer Randbedingungen. Die Kopplung geschieht mittels der Lagrange'schen Multiplikatoren.

Für die Lösung des Gesamtsystems wird eine Gesamtschritt-Lösung und eine Zwischenschritt-Lösung definiert und deren Vor- und Nachteile kurz beleuchtet.

Zusätzlich dürfen die Grundsätze zum Anordnen der Messpunkte und deren Messgröße nicht unberücksichtigt bleiben, da ansonsten keine sinnvolle Aussage vorgenommen werden kann. Hier darf das Prinzip von St. Vénant keineswegs unberücksichtigt bleiben. Bei Konstruktionen, die einer Zwängung unterliegen (können), sind mindestens 2 Dehnungsmessungen jeweils über eine Querschnittshöhe unumgänglich, damit eine Umrechnung der Messwerte in reine Biegemesswerte vorgenommen werden kann.

An grundsätzlichen Beispielen wird der Lösungsweg und das erzielte Resultat angegeben. Hier wird noch einmal auf die eminent wichtige Rolle der Randbedingungen hingewiesen, die immer zur Stabilisierung des Ergebnisses beitragen. Des Weiteren wird die Bandbreite der Lagrange'schen Multiplikatoren untersucht [s. Anhang A]. Bei diesen Beispielen sollte die Größe der Lagrange'schen Multiplikatoren jeweils sehr klein sein, da diese immer die Größe der Korrektur anzeigt.---Ferner wird eine Übersicht zum Formelapparat und der wesentlichste Weg der Abstandsminimierung in toto angegeben [s. Anhang B].

Der grundsätzliche Ansatz beinhaltet zunächst keine Aussage über die zu verwendenden Funktionen; d. h. es liegt per se keine Beschränkung auf ein linienhaftes Tragwerk vor. So kann das Procedere ohne weiteres auf andere Tragwerke – beispielsweise Flächentragwerke – ausgeweitet werden, für die ebenfalls vorzügliche Lösungs- und damit Ansatzfunktionen bekannt sind. Dass der messtechnische Aufwand entsprechend steigt, dürfte selbstverständlich sein.

Eine Ausweitung auf die permanente Strukturüberwachung bietet sich von selbst an, so dass nach Auswertung der Messungen sofort eine Aussage zum Tragverhalten – im Vergleich zur vorliegenden statischen Berechnung – vorgenommen werden kann.

1 Einführung

Die Grundlagen der numerischen Verfahren und deren herausragende Vorteile im Vergleich zu den geschlossenen klassischen Lösungen sind bereits seit längerer Zeit bekannt; ihre umfassende Anwendung erfordert jedoch den Einsatz leistungsfähiger Rechner in Verbindung mit großer Speicherkapazität sowie schnellem Zugriff. Diese Aufrüstung auf der Hardware Seite erfolgte mit der Einführung des PC's erst in einem späteren Zeitabschnitt. Nunmehr stand der stürmischen Entwicklung und Anwendung analytisch-numerischer Verfahren nichts mehr entgegen.

Im Bauwesen ist hier vor allem die Anwendung der finiten Element-Methode zu nennen, die prinzipiell mit einem lokalen Ritz-Ansatz identisch ist. Ihre immensen Vorteile haben zur Entwicklung umfangreicher (kommerzieller) Programmsysteme geführt, deren Einsatz selbstverständlich auch das Entwurfsverhalten des Ingenieurs nachhaltig beeinflusst hat. Zum einen werden nun FE-Programme für nahezu jede – auch einfach überschaubare – Konstruktion verwendet; zum anderen ermöglicht der Einsatz der FE-Methode jetzt das Entwerfen hochgradig komplexer Strukturen mit einem wesentlich verbesserten Verständnis des Gesamttragverhaltens. So ist es jetzt erheblich einfacher, beispielsweise Aussagen zum Querabtrag von Lasten oder zur Umlagerung auf Grund veränderter Steifigkeiten zu geben, Fragen, die bei einer Ingenieurgeneration zuvor noch zu umfangreichen Diskussionen führten.

Das „verbesserte“ Durchdringen des strukturellen Tragvermögens erleichtert es dem Ingenieur, jetzt auch modifizierte Konstruktionen zu entwerfen bzw. auch neue Wege zu beschreiten, wogegen frühere Erfahrungswerte standen; hier wäre die Verwendung unterschiedlicher Baumaterialien bzw. Konstruktionstypen mit erheblichem Umlagerungspotential zu nennen /6/,/7/.

Die Anwendung der umfangreichen numerischen Analytik darf jedoch keineswegs die experimentelle Bestätigung der Berechnungen ausschließen; dies gilt vor allem im Falle neuartiger bzw. komplexer Strukturen. So wird im Bauwesen der Einsatz des Monitoring kontinuierlich verstärkt. Als herausragendes Beispiel wäre hier der Lehrter Bahnhof in Berlin zu nennen, für den an einigen Positionen eine kontinuierliche Überwachung installiert wurde /23/,/25/,/26/.

In einem weiteren Schritt wäre jetzt die Kopplung zwischen den Resultaten der Messergebnisse und der numerischen Analyse vorzunehmen, und zwar derart, dass - beispielsweise interaktiv - eine Interpretation des tatsächlichen strukturellen Verhaltens vorgenommen werden kann. Einer methodischen Kopplung zwischen Experiment und theoretisch-numerischer Analyse ist im Bauwesen bisher ein eher geringes Interesse entgegengebracht worden /22/.

In dieser Untersuchung werden die hiermit verbundenen Fragestellungen aus der Sicht des Bauingenieurs dargelegt. Ziel ist letztlich, mit relativ geringem Aufwand eine Interpretation des tatsächlichen, aktuellen Tragverhaltens einer berechneten und experimentell überwachten Konstruktion zu erhalten.

Eine derartige kontinuierliche Überwachung leistet auch einen sinnvollen Beitrag zur künftigen Substanzerhaltung des Bauwerks.

2 Allgemeine Ableitung

2.1 Grundsätzliche Beziehungen

Die Messwerte einer beliebigen Konstruktion liegen zunächst in allgemeiner Form vor, zu deren Qualität anfangs keine weitere Aussage getroffen wird. Mit diesen bekannten Messwerten soll eine möglichst optimale Beurteilung zum Tragverhalten der Struktur vorgenommen werden, d.h. es ist eine Regressionsanalyse durchzuführen.

Es wird die übersichtliche Matrizen-Formulierung gewählt. Mit

$\mathbf{F} = \{F_{ij}\}$ Matrix mit den Werten der beliebigen Ansatzfunktionen an den Messpunkten i

$\mathbf{c} = \{c_j\}$ Vektor der Regressionsparameter j

$\mathbf{G} = \{G_i\}$ Vektor der Messwerte an den Messpunkten i

$\mathbf{d} = \{d_i\}$ Vektor der Differenzen zwischen Messwert und Regressionskurve an den Punkten i

und

$$Q = \sum d_i^2 = \mathbf{d}^T \times \mathbf{d} \quad \text{mit} \quad \mathbf{d}^T \quad \text{als Transponierte von} \quad \mathbf{d}$$

erhält man die Ausgangsformulierung

$$\mathbf{F} * \mathbf{c} - \mathbf{G} = \mathbf{d} \quad , \quad (2.1)$$

deren Summe der Fehlerquadrate

$$(\mathbf{F} * \mathbf{c} - \mathbf{G})^T * (\mathbf{F} * \mathbf{c} - \mathbf{G}) = Q \quad \text{bzw. nach Umformen}$$

$$\mathbf{c}^T \mathbf{F}^T \mathbf{F} \mathbf{c} - 2 \mathbf{c}^T \mathbf{F}^T \mathbf{G} + \mathbf{G}^T \mathbf{G} = Q \quad (2.2)$$

mit $\frac{\partial Q}{\partial \mathbf{c}}$ zu minimieren ist. Nach der Ableitung erhält man die grundsätzliche Ausgangsgleichung zur Regressionsanalyse

$$2 \mathbf{F}^T \mathbf{F} \mathbf{c} - 2 \mathbf{F}^T \mathbf{G} = 0 \quad . \quad (2.3)$$

Die Beziehung (2.3) fokussiert sich allein auf die Aussage der Messergebnisse, hat aber ansonsten keinerlei Bezug zum mechanisch-physikalischen System. Um diesen Bezug stringent herzustellen, nimmt man in der Ausgangsbeziehung (2.2) eine Korrektur derart vor, dass die Summe Q um einen Korrekturterm zu Q_{ges} erweitert wird, der die physikalischen Eigenschaften des gemessenen Systems zusätzlich berücksichtigt; also gilt mit

$\mathbf{R} = \{R_{ij}\}$ Matrix der Rand- bzw. Nebenbedingungen für beliebige Punkte i bzw. für beliebige Messpunkte i

$\mathbf{b} = \{b_j\}$ Vektor der am System bekannten Rand- bzw. Nebenbedingungen

$\boldsymbol{\lambda} = \{\lambda_j\}$ Vektor der Lagrange'schen Multiplikatoren für die Anpassung an das gegebene mechanische System

$$Q_{ges} = Q + \boldsymbol{\lambda}^T (\mathbf{R} \mathbf{c} - \mathbf{b}) . \quad (2.4)$$

Die modifizierte Ausgangsgleichung erhält man durch Bildung des vollständigen Differentials

$$\frac{\partial Q_{ges}}{\partial \mathbf{c}} = 0 \quad \text{und} \quad \frac{\partial Q_{ges}}{\partial \boldsymbol{\lambda}} = 0 .$$

Die Auflösung von (2.4) führt zum Gleichungssystem (2.5); in dieser nun erweiterten Beziehung sind die noch unbekannt Parameter \mathbf{c} [aus dem Funktionsansatz] und $\boldsymbol{\lambda}$ [aus der physikalischen Justierung des Systems] zu ermitteln, so dass sich folgendes Matrix-System ergibt:

$$\begin{bmatrix} 2 * \mathbf{F}^T \mathbf{F} & \mathbf{R}^T \\ \mathbf{R} & \mathbf{0} \end{bmatrix} * \begin{bmatrix} \mathbf{c} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} 2 * \mathbf{F}^T \mathbf{G} \\ \mathbf{b} \end{bmatrix} \quad \text{bzw.} \\ \begin{bmatrix} \mathbf{B} & \mathbf{R}^T \\ \mathbf{R} & \mathbf{0} \end{bmatrix} * \begin{bmatrix} \mathbf{c} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{A} \\ \mathbf{b} \end{bmatrix} \quad (2.5)$$

Bisher sind auch für die Art des Fehlerausgleichs keine weiteren Voraussetzungen getroffen worden.

2.2 Fehlerausgleich mit diskreten Messwerten allein

Üblicherweise wird man den Gauß'schen Ansatz wählen, in dem allein die diskreten Messwerte direkt zur Minimierung verwendet werden; schreibt man die Beziehung (2.2) als Summenausdruck

$$\left[\sum_i^{1,N} \left\{ \sum_j^{1,M} f_{ij} c_j - g_i \right\} \right]^2 = Q \implies \mathbf{Min} , \quad (2.2aa)$$

ist das Minimum aus der Differenz des Fehlerquadrats der gegebenen Messwerte g_i (mit $i=1,N$ und N = Zahl der Messwerte) und dem Ansatz beliebiger Funktionen f_{ij} (mit $j=1,M$ und M = Zahl der gewählten Funktionen) zu bilden, wobei die Funktionsparameter c_j zu ermitteln sind. Unter Berücksichtigung der physikalischen System-Eigenschaften wird der Ausdruck erweitert zu

$$\left[\sum_i^{1,N} \left\{ \sum_j^{1,M} f_{ij} c_j - g_i \right\} \right]^2 + \left[\sum_n^{1,K} \lambda_n \left\{ \sum_j^{1,M} r_{nj} c_j - b_n \right\} \right] = Q \implies \mathbf{Min} . \quad (2.2ab)$$

Die systembekannten Rand- bzw. Zwischenwerte b_n (mit $n=1,K$ und K = Zahl der definierten Systemwerte) korrigieren den Fehlerausgleich über die Lagrange'schen Multiplikatoren λ_n . Weitere Bedingungen werden für diesen Lösungsansatz nicht getroffen.

2.3 Fehlerausgleich mit Zusatzbedingungen

In einigen Sonderfällen kann es wünschenswert sein, die Lösung durch Einführung weiterer Zusatzbedingungen zu erstellen; dies gilt z. B. zur Herstellung einer stabilen Lösung, die u. U. mit der Minimalforderung in Abschnitt 2.2 nicht zu leisten ist.

In (2.2aa) bzw. (2.2ab) werden ausschließlich die diskreten Messwerte herangezogen; für den Verlauf der Regressionskurve zwischen diesen Messwerten ist jedoch keine Voraussetzung getroffen, d. h. in wenigen Sonderfällen können sich nicht gewünschte Durchschlageffekte der Lösungskurve einstellen.

In diesem Fall kann die Formulierung des Fehlerausgleichs modifiziert werden, indem beispielsweise – im Sinne eines variierten Fourier-Ansatzes - das Minimum der Fläche zwischen Regressionsfunktion und der Funktion, gebildet aus den vorhandenen Messwerten, postuliert wird. Dann gilt sinngemäß nach Abschnitt 2.2:

$$\int_{x_1}^{x_2} \left[\sum_i^{1,N} \left\{ \sum_j^{1,M} f(x)_{ij} c_j - g(x)_i \right\} \right]^2 dx = Q \implies \mathbf{Min} \quad (2.2ba)$$

bzw.

$$\int_{x_1}^{x_2} \left[\sum_i^{1,N} \left\{ \sum_j^{1,M} f(x)_{ij} c_j - g(x)_i \right\} \right]^2 dx + \left[\sum_n^{1,K} \lambda_n \left\{ \sum_j^{1,M} r_{nj} c_j - b_n \right\} \right] \implies \mathbf{Min} \quad (2.2bb)$$

Hierbei ist zu beachten, dass $\mathbf{g(x)}_i$ die Funktion darstellt, die noch aus den diskreten Messpunkten i zu definieren ist; d. h. die Funktion $\mathbf{g(x)}$ ist bisher nicht bekannt.

Die Generierung der Funktion $\mathbf{g(x)}$ erfordert die grundsätzliche Voraussetzung, dass die Zahl der gleichartigen Messwerte in der betrachteten Längeneinheit mindestens ≥ 2 ist, damit beispielsweise eine Approximation mittels eines Polygonzuges erfolgen kann. Bei vorhandenen Messpunkten ≥ 3 bietet sich die Generierung der Funktion $\mathbf{g(x)}$ mit Parabelabschnitten an, die zu wesentlich genauerer Gesamtlösung führen wird. Hier wird bereits prinzipiell die Lösungsform vordefiniert; d. h. es wird - in Abweichung zu Abschnitt 2.2 - dem System noch zusätzliche Energie zugeführt.

Dieser Weg der Regression stellt jedoch immer einen Sonderfall dar, falls – z. B. auf Grund ungenügender Messdaten – anderweitig keine realistische Lösung zu erzielen ist.

3 Typ der Ansatzfunktion

3.1 Hermite-Polynome

In der bisherigen allgemeinen Schreibweise sind die Ausgleichsfunktionen f_{ij} entsprechend Abschnitt 2.2 bzw. 2.3 noch nicht definiert.

Als erste Ausgleichsfunktion bieten sich Hermite-Polynome für alle Messdaten - beispielsweise Durchbiegungs- oder Dehnungswerte - an; diese Polynome haben den Vorteil der übersichtlichen Kopplung an den Abschnittsgrenzen $/4/$ [, weshalb sie auch häufig in der finiten Element-Analyse verwendet werden].

Für Hermite-Polynome 4. Ordnung folgt dann $\mathbf{f}^T = \mathbf{H}^4 = \mathbf{a}_4 * \mathbf{x}_4$
bzw. explizit formuliert:

$$\begin{bmatrix} 1 & 0 & -3 & 2 \\ 0 & 0 & 3 & -2 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & -1 & 1 \end{bmatrix} * \begin{bmatrix} 1 \\ \xi \\ \xi^2 \\ \xi^3 \end{bmatrix} = \begin{bmatrix} H^{1-4} \\ H^{2-4} \\ H^{3-4} \\ H^{4-4} \end{bmatrix} \quad (3.1)$$

Hier ist

\mathbf{a}_4 die 4-parametrische Grundmatrix und
 \mathbf{x}_4 die zugehörige Vektorfunktion der Hermite-Polynome 4. Ordnung.

In gleicher Weise können beispielsweise Hermite-Polynome 6. Ordnung angewandt werden:

$$\mathbf{f}^T = \mathbf{H}^6 = \mathbf{a}_6 * \mathbf{x}_6 \quad \text{oder}$$

$$\begin{bmatrix} 1 & 0 & 0 & -10 & -15 & -6 \\ 0 & 0 & 0 & -10 & -15 & 6 \\ 0 & 1 & 0 & -6 & 8 & -3 \\ 0 & 0 & 0 & -4 & 7 & -3 \end{bmatrix} * \begin{bmatrix} 1 \\ \xi \\ \xi^2 \\ \xi^3 \\ \xi^4 \\ \xi^5 \end{bmatrix} = \begin{bmatrix} H^{1-6} \\ H^{2-6} \\ H^{3-6} \\ H^{4-6} \end{bmatrix} \quad (3.2)$$

Der Verlauf der Hermite-Polynome 4. und 6. Ordnung ist in Bild 3.1 dargestellt. Hierbei stellen die Funktionen H^1 und H^2 die Durchbiegungen und die Funktionen H^3 und H^4 die Winkeldrehungen an den Abschnittsgrenzen dar.

Bei näherer Betrachtung der „auf den ersten Blick als praktisch angenommenen“ Hermite-Polynome zeigen sich jedoch einige Nachteile. Die Polynome sind eine aufsteigende Linearkombination geometrischer Funktionen mit den definierten Bedingungen der Funktionen H^1 und H^2 bzw. H^3 und H^4 . Untersucht man jedoch des weiteren die Krümmungsbeziehungen, d.h. die 2. Ableitung, so liegt für jedes einzelne Polynom 4. Ordnung eine lineare Krümmung über den gesamten Abschnitt vor; dieses Verhalten kann bei der Formulierung der Systembeziehungen zu linearen Abhängigkeiten, also singulären Formulierungen führen.

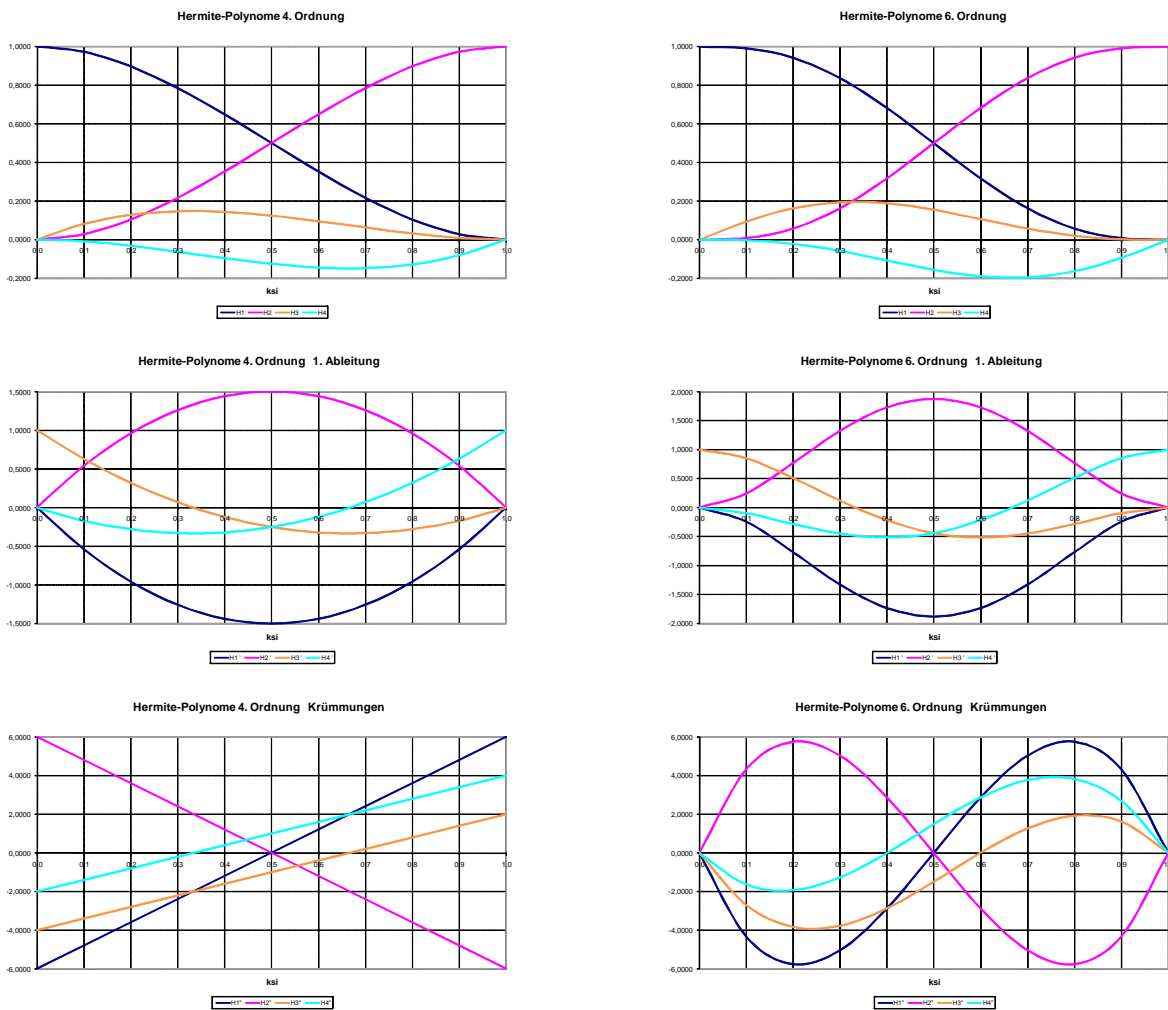


Bild 3.1: Verlauf der Hermite-Polynome

Hermite-Polynome 6. Ordnung haben die nachteilige Eigenschaft, an allen Abschnittsgrenzen die Krümmung Null aufzuweisen; soll das mechanische System aus mehreren Abschnitten gebildet werden, kann dann trotz höheren Ansatzes keine Krümmungsidentität mehr übertragen werden.

3.2 Bernoulli-Ansatz zur Balkenbiegung

Um eine höhere Flexibilität bei der Auswertung zu gewinnen, empfiehlt sich die Anwendung (aufsteigend) geometrischer Funktionen ohne vorgegebene Bedingungen, wobei der Grad des Ansatzes aus der Strukturanalyse zur Balkenbiegung für Einzellasten bzw. kontinuierliche Belastung nach Bernoulli ideal für die Aufgabenstellung gewählt ist; also gilt für

gleichmäßige Belastung

$$\mathbf{f}_p^T = \mathbf{b}_5 * \mathbf{x}_5$$

$$\mathbf{f}'_p^T = \mathbf{b}'_5 * \mathbf{x}'_5$$

$$\mathbf{f}''_p^T = \mathbf{b}''_5 * \mathbf{x}''_5 \quad \text{bzw. für}$$

Einzellasten

$$\mathbf{f}_E^T = \mathbf{b}_4 * \mathbf{x}_4$$

$$\mathbf{f}'_E{}^T = \mathbf{b}'_4 * \mathbf{x}'_4$$

$$\mathbf{f}''_E{}^T = \mathbf{b}''_4 * \mathbf{x}''_4 .$$

Die untere Indexzahl gibt die Parameterzahl des definierten Ansatzes an. Für die Gleichlast erhält man aus der Theorie der reinen Balkenbiegung nach Bernoulli die vollständigen Beziehungen der Ansatzfunktion:

$$\mathbf{f}^T = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} * \begin{bmatrix} 1 \\ \xi \\ \xi^2 \\ \xi^3 \\ \xi^4 \end{bmatrix}$$

$$\mathbf{f}'^T = \begin{bmatrix} 0 & & & \\ & 1/s_k & & \\ & & 2/s_k & \\ & & & 3/s_k \\ & & & & 4/s_k \end{bmatrix} * \begin{bmatrix} 0 \\ 1 \\ \xi \\ \xi^2 \\ \xi^3 \end{bmatrix}$$

$$\mathbf{f}''^T = \begin{bmatrix} 0 & & & \\ & 0 & & \\ & & 2/s_k^2 & \\ & & & 6/s_k^2 \\ & & & & 12/s_k^2 \end{bmatrix} * \begin{bmatrix} 0 \\ 0 \\ 1 \\ \xi \\ \xi^2 \end{bmatrix}$$

(3.3)

In (3.3) bedeutet $\xi = x/s_k$ die Abschnittskordinate mit s_k als Länge des jeweiligen Abschnitts k .

Im Falle der reinen Anwendung für Einzellasten muss nur die Löschung der letzten Zeile und Spalte in den \mathbf{b}_5 -Matrizen verbunden mit der Löschung der letzten Zeile der Vektoren \mathbf{x}_5 erfolgen.

Für diese allgemeineren Funktionen \mathbf{f}_p bzw. \mathbf{f}_E sind - im Gegensatz zu den Hermite-Polynomen - keine Vorfestlegungen für die Abschnittsgrenzen definiert worden.

Damit können nun entsprechend (2.2) die Gesamtmatrizen \mathbf{F} bzw. \mathbf{F}^T gebildet werden; hier gilt die Zahl der Messergebnisse N , wobei jeder Messung i die spezifische Koordinate ξ_i zuzuordnen ist. \mathbf{F} bzw. \mathbf{F}^T sind Rechteckmatrizen mit $\mathbf{F} = [N \times 5]$ bzw. $\mathbf{F}^T = [5 \times N]$, deren Größe sich aus der Zahl der Parameter der Regressionsfunktion sowie der Zahl der Messwerte definiert.

Die Funktionsmatrizen \mathbf{F} für den vollständigen Bernoulli-Ansatz werden in (3.4) für N Messergebnisse explizit ausgewiesen:

$$\mathbf{F} = \begin{bmatrix} 1 & \xi_1 & \xi_1^2 & \xi_1^3 & \xi_1^4 \\ 1 & \xi_2 & \xi_2^2 & \xi_2^3 & \xi_2^4 \\ 1 & \xi_3 & \xi_3^2 & \xi_3^3 & \xi_3^4 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & \xi_i & \xi_i^2 & \xi_i^3 & \xi_i^4 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & \xi_N & \xi_N^2 & \xi_N^3 & \xi_N^4 \end{bmatrix} \quad \text{bzw.} \quad \left. \begin{array}{l} \\ \\ \\ \\ \\ \\ \\ \end{array} \right\} \quad (3.4)$$

$$\mathbf{F}^T = \begin{bmatrix} 1 & 1 & 1 & \cdot & \cdot & 1 & \cdot & 1 \\ \xi_1 & \xi_2 & \xi_3 & \cdot & \cdot & \xi_i & \cdot & \xi_N \\ \xi_1^2 & \xi_2^2 & \xi_3^2 & \cdot & \cdot & \xi_i^2 & \cdot & \xi_N^2 \\ \xi_1^3 & \xi_2^3 & \xi_3^3 & \cdot & \cdot & \xi_i^3 & \cdot & \xi_N^3 \\ \xi_1^4 & \xi_2^4 & \xi_3^4 & \cdot & \cdot & \xi_i^4 & \cdot & \xi_N^4 \end{bmatrix}$$

4 Formulierung der Gesamtmatrizen

In den vorangehenden Schritten ist die Grundlage zur Bildung des Gesamtsystems gelegt worden; es werden jetzt die Einzelmatrizen des Gesamtsystems für die weitere Berechnung spezifiziert. Die allgemeine Beziehung (2.5) ist für die Regression der Biegelinie einer gemessenen Struktur abgeleitet worden, und zwar für einen beliebigen Abschnitt k ; anschließend kann dann das komplette System „in toto“ für K als Zahl der gewählten Abschnitte gebildet und berechnet werden.

Die gestellte Aufgabe wird dadurch komplexer, dass in praxi unterschiedliche Messungen durchgeführt werden, und zwar für die

Durchbiegungen	w	,
Winkelneigungen	w'	und
Krümmungen	w''	,

wobei letztere aus den Dehnungen ε in Verbindung zum Abstand z zur neutralen Achse gebildet werden.

Unter Berücksichtigung dieser geometrischen Beziehungen ist (2.5) dann zu modifizieren:

$$\begin{bmatrix} \mathbf{B}_{[w]} & \mathbf{0} & \mathbf{0} & \mathbf{R}^T_{[w]} \\ \mathbf{0} & \mathbf{B}_{[w']} & \mathbf{0} & \mathbf{R}^T_{[w']} \\ \mathbf{0} & \mathbf{0} & \mathbf{B}_{[w'']} & \mathbf{R}^T_{[w'']} \\ \mathbf{R}_{[w]} & \mathbf{R}_{[w']} & \mathbf{R}_{[w'']} & \mathbf{0} \end{bmatrix} * \begin{bmatrix} \mathbf{c}_{[w]} \\ \mathbf{c}_{[w']} \\ \mathbf{c}_{[w'']} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{[w]} \\ \mathbf{A}_{[w']} \\ \mathbf{A}_{[w'']} \\ \mathbf{b} \end{bmatrix} \quad (4.1)$$

4.1 System- und Lastmatrizen aus Durchbiegungsmessungen

Für die reinen Durchbiegungsmessungen sind die entsprechenden Matrizen bereits in Abschnitt 2 aufgestellt worden; es gilt (2.5), also ist die Systemmatrix

$$\mathbf{B}_{[w]} = 2 * \mathbf{F}^T \mathbf{F}$$

und die Lastmatrix

$$\mathbf{A}_{[w]} = 2 * \mathbf{F}^T \mathbf{G}_{[w]}$$

mit N_1 als Anzahl der Durchbiegungsmesswerte w_i , $i=1, N_1$.

Die vollständige Auswertung führt zu

$$\mathbf{B}_{[w]} = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} & b_{15} \\ b_{21} & b_{22} & b_{23} & b_{24} & b_{25} \\ b_{31} & b_{32} & b_{33} & b_{34} & b_{35} \\ b_{41} & b_{42} & b_{43} & b_{44} & b_{45} \\ b_{51} & b_{52} & b_{53} & b_{54} & b_{55} \end{bmatrix} \quad \mathbf{A}_{[w]} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} \quad (4.2)$$

bzw.

$$\mathbf{B}[\mathbf{w}] = 2 * \begin{bmatrix} \sum_i^{1,N_1} 1 * 1 & \sum_i^{1,N_1} 1 * \xi_i & \sum_i^{1,N_1} 1 * \xi_i^2 & \sum_i^{1,N_1} 1 * \xi_i^3 & \sum_i^{1,N_1} 1 * \xi_i^4 \\ \sum_i^{1,N_1} \xi_i * 1 & \sum_i^{1,N_1} \xi_i * \xi_i & \sum_i^{1,N_1} \xi_i * \xi_i^2 & \sum_i^{1,N_1} \xi_i * \xi_i^3 & \sum_i^{1,N_1} \xi_i * \xi_i^4 \\ \sum_i^{1,N_1} \xi_i^2 * 1 & \sum_i^{1,N_1} \xi_i^2 * \xi_i & \sum_i^{1,N_1} \xi_i^2 * \xi_i^2 & \sum_i^{1,N_1} \xi_i^2 * \xi_i^3 & \sum_i^{1,N_1} \xi_i^2 * \xi_i^4 \\ \sum_i^{1,N_1} \xi_i^3 * 1 & \sum_i^{1,N_1} \xi_i^3 * \xi_i & \sum_i^{1,N_1} \xi_i^3 * \xi_i^2 & \sum_i^{1,N_1} \xi_i^3 * \xi_i^3 & \sum_i^{1,N_1} \xi_i^3 * \xi_i^4 \\ \sum_i^{1,N_1} \xi_i^4 * 1 & \sum_i^{1,N_1} \xi_i^4 * \xi_i & \sum_i^{1,N_1} \xi_i^4 * \xi_i^2 & \sum_i^{1,N_1} \xi_i^4 * \xi_i^3 & \sum_i^{1,N_1} \xi_i^4 * \xi_i^4 \end{bmatrix}$$

$$\mathbf{A}[\mathbf{w}] = 2 * \begin{bmatrix} \sum_i^{1,N_1} 1 * w_i \\ \sum_i^{1,N_1} \xi_i * w_i \\ \sum_i^{1,N_1} \xi_i^2 * w_i \\ \sum_i^{1,N_1} \xi_i^3 * w_i \\ \sum_i^{1,N_1} \xi_i^4 * w_i \end{bmatrix}$$

ξ_i sind die auf die Länge s_k des Abschnitts k bezogenen Längskoordinaten der zugehörigen Durchbiegungen w_i . Matrix $\mathbf{B}[\mathbf{w}]$ ist voll-symmetrisch.

4.2 System- und Lastmatrizen aus Winkelneigungsmessungen

Entsprechend Beziehung (4.1) gilt für die Systemmatrix

$$\mathbf{B}[\mathbf{w}'] = 2 * \mathbf{F}'^T \mathbf{F}'$$

und die Lastmatrix

$$\mathbf{A}[\mathbf{w}'] = 2 * \mathbf{F}'^T \mathbf{G}[\mathbf{w}']$$

mit N_2 als Anzahl der Winkelneigungsmesswerte

$w'_i, i=1, N_2$.

Die vollständige Auswertung führt zu

$$\mathbf{B}[\mathbf{w}'] = \begin{bmatrix} b'_{11} & b'_{12} & b'_{13} & b'_{14} & b'_{15} \\ b'_{21} & b'_{22} & b'_{23} & b'_{24} & b'_{25} \\ b'_{31} & b'_{32} & b'_{33} & b'_{34} & b'_{35} \\ b'_{41} & b'_{42} & b'_{43} & b'_{44} & b'_{45} \\ b'_{51} & b'_{52} & b'_{53} & b'_{54} & b'_{55} \end{bmatrix} \quad \mathbf{A}[\mathbf{w}'] = \begin{bmatrix} a'_1 \\ a'_2 \\ a'_3 \\ a'_4 \\ a'_5 \end{bmatrix} \quad (4.3)$$

bzw.

$$\mathbf{B}[\mathbf{w}'] = 2 * \frac{1}{s_k^2} \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 1 * 1 \sum_{i=1, N_2} 1 * 1 & 1 * 2 \sum_{i=1, N_2} 1 * \xi_i & 1 * 3 \sum_{i=1, N_2} 1 * \xi_i^2 & 1 * 4 \sum_{i=1, N_2} 1 * \xi_i^3 \\ \mathbf{0} & 2 * 1 \sum_{i=1, N_2} \xi_i * 1 & 2 * 2 \sum_{i=1, N_2} \xi_i * \xi_i & 2 * 3 \sum_{i=1, N_2} \xi_i * \xi_i^2 & 2 * 4 \sum_{i=1, N_2} \xi_i * \xi_i^3 \\ \mathbf{0} & 3 * 1 \sum_{i=1, N_2} \xi_i^2 * 1 & 3 * 2 \sum_{i=1, N_2} \xi_i^2 * \xi_i & 3 * 3 \sum_{i=1, N_2} \xi_i^2 * \xi_i^2 & 3 * 4 \sum_{i=1, N_2} \xi_i^2 * \xi_i^3 \\ \mathbf{0} & 4 * 1 \sum_{i=1, N_2} \xi_i^3 * 1 & 4 * 2 \sum_{i=1, N_2} \xi_i^3 * \xi_i & 4 * 3 \sum_{i=1, N_2} \xi_i^3 * \xi_i^2 & 4 * 4 \sum_{i=1, N_2} \xi_i^3 * \xi_i^3 \end{bmatrix}$$

$$\mathbf{A}[\mathbf{w}'] = 2 * \frac{1}{s_k} \begin{bmatrix} \mathbf{0} \\ 1 \sum_{i=1, N_2} 1 * w'_i \\ 2 \sum_{i=1, N_2} \xi_i * w'_i \\ 3 \sum_{i=1, N_2} \xi_i^2 * w'_i \\ 4 \sum_{i=1, N_2} \xi_i^3 * w'_i \end{bmatrix}$$

ξ_i sind die auf die Länge s_k des Abschnitts k bezogenen Längskoordinaten der zugehörigen Durchbiegungen w_i . Matrix $\mathbf{B}[\mathbf{w}']$ ist nur in den Zeilen/Spalten 2-5 voll-symmetrisch. Da die Winkelneigung die 1. Ableitung der Durchbiegung darstellt, bleibt die Konstante c_1 mit der zugehörigen 1. Zeile und Spalte frei. Zur Erfüllung der Matrix bedarf es 1er zusätzlichen Durchbiegungsbedingung w in dem jeweiligen Abschnitt k , da anderweitig keine eindeutige Lösung zu erzielen ist.

4.3 System- und Lastmatrizen aus Dehnungsmessungen

Entsprechend Beziehung (4.1) gilt für die Systemmatrix

$$\mathbf{B}[\mathbf{w}''] = 2 * \mathbf{F}''^T \mathbf{F}''$$

und die Lastmatrix

$$\mathbf{A}[\mathbf{w}''] = 2 * \mathbf{F}''^T \mathbf{G}[\mathbf{w}'']$$

mit N_3 als Anzahl der Dehnungsmesswerte

$$w''_i, \quad i=1, N_3.$$

Die vollständige Auswertung führt zu

$$\mathbf{B}[\mathbf{w}''] = \begin{bmatrix} b''_{11} & b''_{12} & b''_{13} & b''_{14} & b''_{15} \\ b''_{21} & b''_{22} & b''_{23} & b''_{24} & b''_{25} \\ b''_{31} & b''_{32} & b''_{33} & b''_{34} & b''_{35} \\ b''_{41} & b''_{42} & b''_{43} & b''_{44} & b''_{45} \\ b''_{51} & b''_{52} & b''_{53} & b''_{54} & b''_{55} \end{bmatrix} \quad \mathbf{A}[\mathbf{w}''] = \begin{bmatrix} a''_1 \\ a''_2 \\ a''_3 \\ a''_4 \\ a''_5 \end{bmatrix} \quad (4.4)$$

bzw. mit $w''_i = \varepsilon_i / z_i$

$$\mathbf{B}[\mathbf{w}''] = 2 * \frac{1}{s_k^4} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 * 2 \sum_i^{1, N_3} 1 * 1 & 2 * 6 \sum_i^{1, N_3} 1 * \xi_i & 2 * 12 \sum_i^{1, N_3} 1 * \xi_i^2 \\ 0 & 0 & 6 * 2 \sum_i^{1, N_3} \xi_i * 1 & 6 * 6 \sum_i^{1, N_3} \xi_i * \xi_i & 6 * 12 \sum_i^{1, N_3} \xi_i * \xi_i^2 \\ 0 & 0 & 12 * 2 \sum_i^{1, N_3} \xi_i^2 * 1 & 12 * 6 \sum_i^{1, N_3} \xi_i^2 * \xi_i & 12 * 12 \sum_i^{1, N_3} \xi_i^2 * \xi_i^2 \end{bmatrix}$$

$$\mathbf{A}[\mathbf{w}''] = 2 * \frac{1}{s_k^2} \begin{bmatrix} 0 \\ 0 \\ 2 * 1 \sum_i^{1, N_3} 1 * \varepsilon_i / z_i \\ 3 * 2 \sum_i^{1, N_3} \xi_i * \varepsilon_i / z_i \\ 4 * 3 \sum_i^{1, N_3} \xi_i^2 * \varepsilon_i / z_i \end{bmatrix}$$

ξ_i sind die auf die Länge s_k des Abschnitts k bezogenen Längskoordinaten der zugehörigen Durchbiegungen w_i . Matrix $\mathbf{B}[\mathbf{w}'']$ ist nur in den Zeilen/Spalten 3-5 voll-symmetrisch. Da die Dehnung bzw. die Krümmung die 2. Ableitung der Durchbiegung darstellt, bleiben die Konstanten c_1 und c_2 der zugehörigen 1. und 2. Zeile und Spalte frei. Zur Erfüllung der Matrix bedarf es 2er zusätzlichen Bedingungen in dem jeweiligen Abschnitt k , da anderweitig keine eindeutige Lösung zu erzielen ist; diese können beispielsweise 2 Durchbiegungen w oder 1 Durchbiegung w zusammen mit 1 Winkelneigung w' sein.

Die Matrix $\mathbf{B}[\mathbf{w}']$ lässt sich auch direkt aus der originären Matrix $\mathbf{B}[\mathbf{w}] = 2 * \mathbf{F}^T \mathbf{F}$ herleiten, indem dort die jeweiligen Funktionsanteile in den Summen $1x$ abgeleitet werden, wobei der Faktor $1/s_k$ wegen der Dimensionslosigkeit der Ausdrücke nicht übersehen werden darf. In gleicher Weise wird dann (noch einmal) die Matrix $\mathbf{B}[\mathbf{w}'']$ generiert. Bei den lastseitigen Matrizen ist entsprechend zu verfahren.

5 Berücksichtigung des mechanischen Systems mittels Lagrange'scher Multiplikatoren

Lagrange'sche Multiplikatoren sind überaus geeignet, ein definiertes System ohne größeren Aufwand zu ergänzen oder weiter zu konditionieren /21/. Diese Methodik wird hier eingesetzt, um die mechanischen Eigenschaften des betrachteten Systems zusätzlich in der Regressionsanalyse zu berücksichtigen; d. h. es werden alle weiteren physikalisch vorhandenen Informationen - jenseitig des Messvorganges - verarbeitet. Die Lagrange'schen Multiplikatoren können nicht nur in beliebiger Form hinzugefügt werden, sondern sind sogar in der Lage, das Gesamtsystem u. U. auch zu stabilisieren. Im Grundsatz sollten die erhaltenen Beiträge der Lagrange'schen Multiplikatoren relativ klein im Verhältnis zu den Parameterlösungen sein; dies ist ein deutliches Überprüfungs-kriterium, dass der Gesamtfehler der Regressionsanalyse im kleinen Bereich verbleibt. Im anderen Fall könnte es sein, dass die Messwerte u. U. das strukturelle Verhalten noch nicht genügend darstellen konnten oder diese auch sehr von der struktur-analytischen Berechnung abweichen /22/.

Jede zusätzliche Systeminformation trägt in den meisten Fällen zur Verbesserung der gesuchten Lösung bei, solange es sich nicht um eine (abhängige) Linearkombination handelt.

5.1 Randbedingungen

Auflager bzw. Auflagerbereiche stellen häufig wegen ihrer erheblich größeren Steifigkeiten im Vergleich zur übrigen Konstruktion sog. natürliche Randbedingungen dar. Die Größe des die Lastseite definierenden Vektors \mathbf{b} in (2.5) wird daher meistens „0“ sein. Die zugeordnete Zeile in der Systemmatrix \mathbf{R} richtet sich dann nach der Wahl der Regressionsfunktion; am Beispiel des Bernoulli-Ansatzes nach Abschnitt 3.2 ist hier die entsprechende Koordinate ξ_i im jeweiligen Abschnitt k zu wählen, wobei überwiegend entweder der linke Abschnittsrand $\xi_i = 0$ oder der rechte Abschnittsrand $\xi_i = 1$ eingestellt wird.

Für die entsprechende Zeile der Systemmatrix \mathbf{R} mit dem zugehörigen **Abschnitt k** erhält man dann:

$$\begin{array}{llll} \text{links} & \xi_i = 0 & f(0) & [1 \quad 0 \quad 0 \quad 0 \quad 0] \\ & & f'(0) & [0 \quad 1/s_k \quad 0 \quad 0 \quad 0] \\ (5.1a) & & f''(0) & [0 \quad 0 \quad 2/s_k^2 \quad 0 \quad 0] \end{array}$$

$$\begin{array}{llll} \text{rechts} & \xi_i = 1 & f(1) & [1 \quad 1 \quad 1 \quad 1 \quad 1] \\ & & f'(1) & [0 \quad 1/s_k \quad 2/s_k \quad 3/s_k \quad 4/s_k] \\ (5.1b) & & f''(1) & [0 \quad 0 \quad 2/s_k^2 \quad 6/s_k^2 \quad 12/s_k^2] \end{array}$$

Liegen die Randbedingungen jedoch innerhalb des betrachteten Abschnitts k, ist die Beziehung (3.3) vollständig mit der vorhandenen ξ_i -Koordinate auszuwerten.

allgemein	ξ_i	$f(\xi_i)$	$[$	1	ξ_i	ξ_i^2	ξ_i^3	ξ_i^4	$]$	
		$f'(\xi_i)$	$[$	0	$1/s_k$	ξ_i/s_k	ξ_i^2/s_k	ξ_i^3/s_k	$]$	(5.1c)
		$f''(\xi_i)$	$[$	0	0	$2/s_k^2$	ξ_i/s_k^2	ξ_i^2/s_k^2	$]$	

Hier sollte der Hinweis nicht fehlen, dass - auch bei Vorhandensein aller der 3 Messgrößen - die entsprechende Bedingung nur 1x definiert werden darf, da das Gesamtsystem im gegenteiligen Fall singular werden kann.

5.2 Innere Kopplung

Zum einen kann die betrachtete Konstruktion mehrere Felder aufweisen, zum anderen ist auch eine Unterteilung eines Feldes in mehrere Abschnitte k denkbar. In diesem Fall sind also - entsprechend der inneren Tragweise des Systems - alle Abschnitte an den Übergängen miteinander zu koppeln. Die Justierung von Abschnitt k-1 zu Abschnitt k wird dann folgend auch als innere Kopplung bezeichnet.

Abschnitt k-1 liege links von Abschnitt k; somit gilt folgende Kopplung für die Funktion f bzw. deren 1. und 2. Ableitung [s. (3.3)] an der definierten Abschnittsgrenze k-1,k für die Systemmatrix R :

Abschnitt k-1	$\xi_i = 1$	=	Abschnitt k	$\xi_i = 0$								
f	$[$	1	1	1	1	1	-1	0	0	0	0	$]$
f'	$[$	0	$1/s_k$	$2/s_k$	$3/s_k$	$4/s_k$	0	$-1/s_k$	0	0	0	$]$
f''	$[$	0	0	$2/s_k^2$	$6/s_k^2$	$12/s_k^2$	0	0	$-2/s_k^2$	0	0	$]$

(5.2)

Zur Erfüllung von (5.2) ist zu beachten, dass bei gegebenen unterschiedlichen Messgrößen die innere Kopplung nur einmal vorgenommen wird, damit Linearkombinationen mit der Konsequenz der Singularität ausgeschlossen werden.

6 Lösungswege zur Verbindung von Messung und Strukturanalyse

In den vorangehenden Abschnitten sind die Grundlagen erläutert worden; es gilt nun, eine übersichtliche Verbindung zwischen Messergebnissen einerseits und der Interpretation des strukturellen Tragverhaltens andererseits herzustellen. Aus der Vielzahl der Optionen werden hier zwei Wege erörtert.

6.1 Integrale Formulierung

Grundvoraussetzung sei eine flexible Anwendungsmöglichkeit, d.h. alle Messgrößen - Durchbiegungen, Winkelneigungen und Dehnungen - sollen erfasst werden können, und zwar zusätzlich in beliebiger Reihenfolge. Das System wird aus mehreren Abschnitten bestehen, so dass damit in jedem Abschnitt k folgende Mess- bzw. „Last“-Kombinationen postuliert sind:

- | | | | |
|----|-----|------|-------|
| 1. | w | w' | w'' |
| 2. | w | w' | |
| 3. | w | | w'' |
| 4. | | w' | w'' |

Des Weiteren wird eine Lösung in toto gefordert. Die Grundlage ist hier in (4.1) gegeben. Die weitere Vorgabe an die Matrizenarchitektur sei, dass die Anordnung der Submatrizen $\mathbf{B}[w]$, $\mathbf{B}[w']$ bzw. $\mathbf{B}[w'']$ in (4.1) auch für das Gesamtsystem weiter gültig ist, also hintereinander zunächst die w -, dann w' - und w'' -Anteile. Mit Einführung und Besetzung von $\mathbf{B}[w']$ bzw. $\mathbf{B}[w'']$ ist das Gesamtsystem nicht mehr vollsymmetrisch, da hier zusätzliche Bedingungen eingeführt werden müssen, um die Parameter c_1 bzw. c_1 und c_2 zu ermitteln [s. Abschnitt 4 mit (4.3) und (4.4)].

Die unabdingbare Forderung nach größter Flexibilität führt u. U. zu dem Nachteil, dass diese Submatrizen nicht zwangsläufig identische Regelmäßigkeit aufweisen müssen. Liegen z. B. in allen Abschnitten k nicht ausschließlich Messwerte nach Kombination 1. vor, sind die Submatrizen von unterschiedlicher Größe.

Anschließend wird die innere Kopplung nach Abschnitt 5.2 vollzogen. Hierbei wird - je nach Vorgabe für die vorliegende Struktur - an den Nachbarabschnitten $k-1, k$ Kontinuität für Durchbiegung, Neigung und Krümmung hergestellt. Zur Vermeidung von unzulässigen Linearkombinationen erfolgt diese Kontinuitätsbildung aufsteigend nur für den Bereich $w \div w$ oder $w \div w'$ oder $w \div w''$ oder $w' \div w''$. Hier führt die Forderung der maximalen Flexibilität u. U. zu einer recht unregelmäßig, schwach besetzten und damit recht unübersichtlichen Submatrix.

Die Lösung soll integral erreicht werden, so dass die Lagrange'schen Multiplikatoren nach Abschnitt 5.3 zur Kopplung der (unabhängigen) Messbereiche aus Durchbiegung, Winkelneigung und Dehnung hinzuzufügen sind. Auch hier muss auf die Vermeidung der Matrizensingularität geachtet werden, d.h. je nach vorliegender Messwertkombination des Abschnitts k ist die Parameteridentität nur für den Bereich $w \div w'$ und $w \div w''$ zu fordern; für den Bereich $w' \div w''$ ist sie dann noch einzufügen, wenn beispielsweise im Abschnitt k Messwerte w fehlen. Auch hier kann der schwach besetzte Submatrixbereich bei entsprechender Messwertkombination recht unregelmäßig werden.

Im letzten Schritt sind die Randbedingungen entsprechend Abschnitt 5.1 einzusetzen. Auch hier gilt es zur Vermeidung der Systemsingularität, dass diese Bedingungen aufsteigend entweder im w - oder w' oder w'' -Bereich (nur 1x) formuliert werden dürfen.

Die entsprechende Reihenfolge wird im lastseitigen Vektor - gebildet aus den Messwerten - vorgenommen.

Damit liegt die Matrizenarchitektur für eine integrale Lösung vor; dieses Procedere wird folgend auch mit **Gesamtschritt**-Lösung bezeichnet.

Mit den Kürzeln

- „inKo“ für die innere Kopplung ,
- „äuKo“ für die äußere Kopplung und
- „Rand“ für die Randbedingungen

folgt die Gesamtschritt-Lösung (4.1) zu:

$$\begin{bmatrix}
 \mathbf{B}_{[w]} & \mathbf{0} & \mathbf{0} & \mathbf{R}^T_{[w]}^{\text{inKo}} & \mathbf{R}^T_{[w]}^{\text{äuKo}} & \mathbf{R}^T_{[w]}^{\text{Rand}} \\
 \mathbf{0} & \mathbf{B}_{[w']} & \mathbf{0} & \mathbf{R}^T_{[w']}^{\text{inKo}} & \mathbf{R}^T_{[w']}^{\text{äuKo}} & \mathbf{R}^T_{[w']}^{\text{Rand}} \\
 \mathbf{0} & \mathbf{0} & \mathbf{B}_{[w'']} & \mathbf{R}^T_{[w'']}^{\text{inKo}} & \mathbf{R}^T_{[w'']}^{\text{äuKo}} & \mathbf{R}^T_{[w'']}^{\text{Rand}} \\
 \mathbf{R}_{[w]}^{\text{inKo}} & \mathbf{R}_{[w']}^{\text{inKo}} & \mathbf{R}_{[w'']}^{\text{inKo}} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
 \mathbf{R}_{[w]}^{\text{äuKo}} & \mathbf{R}_{[w']}^{\text{äuKo}} & \mathbf{R}_{[w'']}^{\text{äuKo}} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
 \mathbf{R}_{[w]}^{\text{Rand}} & \mathbf{R}_{[w']}^{\text{Rand}} & \mathbf{R}_{[w'']}^{\text{Rand}} & \mathbf{0} & \mathbf{0} & \mathbf{0}
 \end{bmatrix}
 *
 \begin{bmatrix}
 \mathbf{c}_{[w]} \\
 \mathbf{c}_{[w']} \\
 \mathbf{c}_{[w'']} \\
 \boldsymbol{\lambda}^{\text{inKo}} \\
 \boldsymbol{\lambda}^{\text{äuKo}} \\
 \boldsymbol{\lambda}^{\text{Rand}}
 \end{bmatrix}
 =
 \begin{bmatrix}
 \mathbf{A}_{[w]} \\
 \mathbf{A}_{[w']} \\
 \mathbf{A}_{[w'']} \\
 \mathbf{0} \\
 \mathbf{0} \\
 \mathbf{b}^{\text{Rand}}
 \end{bmatrix}
 \tag{6.1}$$

Die gesuchten Parameter \mathbf{c} stellen die Lösungen dar; hier ergibt die Überprüfung, dass die Parameter $\mathbf{c}_{[w]} = \mathbf{c}_{[w']} = \mathbf{c}_{[w'']}$ identisch sein müssen.

Die Größen der Lagrange'schen Multiplikatoren $\boldsymbol{\lambda}$ sind der Ausweis zur Güte der Lösung bzw. der Messwerte. $\boldsymbol{\lambda}$ sollte im Vergleich zu \mathbf{c} relativ klein sein. Dies zeigen die Beispiele der Regressionsanalysen.

6.2 Transformation in virtuelle Durchbiegungen

Es sollen die in Abschnitt 6.1 postulierten Anforderungen beibehalten, aber die Nachteile recht unübersichtlicher Submatrizenbereiche - so weit wie möglich - vermieden werden. In diesem Fall bietet es sich an, die unterschiedlichen Messwertarten auf ein gleichrangiges Niveau zu transformieren, beispielsweise mit Bezugnahme auf die Durchbiegungen \mathbf{w} , also:

- Umwandlung der Winkelneigungen in Durchbiegungen ($\mathbf{w}' \rightarrow \mathbf{w}$)
- Umwandlung der Dehnungen in Durchbiegungen ($\boldsymbol{\varepsilon} \rightarrow \mathbf{w}$)

Die Fokussierung auf die Durchbiegung muss nicht zwingend vorgenommen werden, empfiehlt sich jedoch aus Praktikabilitätsgründen.

Unter dieser Prämisse lässt sich (6.1) entsprechend umschreiben:

1. Da alle Messwerte dann letztlich als Durchbiegungen \mathbf{w} vorliegen, entfallen die Anteile der inneren Kopplung aus Winkelneigung und Dehnung; die Submatrizen verschwinden.
2. Die Transformation der „höheren“ Messwerte zu Durchbiegungen \mathbf{w} stellt zunächst eine Entkopplung dieser Anteile dar, so dass dann die Submatrizen aus äußerer Kopplung näherungsweise zu Null gesetzt werden dürfen.
3. Bei der Generierung der Submatrix der Randbedingungen sind bereits alle Durchbiegungsbereiche der Übersichtlichkeit wegen favorisiert worden; jetzt sind alle Randbedingungen in der Submatrix

$\mathbf{R}_{[w]}^{\text{Rand}}$ konzentriert, so dass die anderen Matrizen $\mathbf{R}_{[w']^{\text{Rand}}}$ und $\mathbf{R}_{[w'']^{\text{Rand}}}$ ebenfalls identisch verschwinden.

So erhält man aus (6.1) ein vereinfachtes System

$$\begin{bmatrix} \mathbf{B}_{[w]} & \mathbf{0} & \mathbf{0} & \mathbf{R}^T_{[w]}^{\text{inKo}} & \approx \mathbf{0} & \mathbf{R}^T_{[w]}^{\text{Rand}} \\ \mathbf{0} & \mathbf{B}_{[w']} & \mathbf{0} & \mathbf{0} & \approx \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{B}_{[w'']} & \mathbf{0} & \approx \mathbf{0} & \mathbf{0} \\ \mathbf{R}_{[w]}^{\text{inKo}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \approx \mathbf{0} & \approx \mathbf{0} & \approx \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{R}_{[w]}^{\text{Rand}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} * \begin{bmatrix} \mathbf{c}_{[w]} \\ \mathbf{c}_{[w']} \\ \mathbf{c}_{[w'']} \\ \boldsymbol{\lambda}^{\text{inKo}} \\ \approx \mathbf{0} \\ \boldsymbol{\lambda}^{\text{Rand}} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{[w]} \\ \mathbf{A}_{[w']} \\ \mathbf{A}_{[w'']} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{b}^{\text{Rand}} \end{bmatrix},$$

das nach Streichung der nur mit Null besetzten Spalten bzw. Zeilen sowie nach Zeilen- und Spaltentausch eine entkoppelte und übersichtliche Architektur ergibt:

$$\begin{bmatrix} \mathbf{B}_{[w']} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_{[w'']} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{B}_{[w]} & \mathbf{R}^T_{[w]}^{\text{inKo}} & \mathbf{R}^T_{[w]}^{\text{Rand}} \\ \mathbf{0} & \mathbf{0} & \mathbf{R}_{[w]}^{\text{inKo}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{R}_{[w]}^{\text{Rand}} & \mathbf{0} & \mathbf{0} \end{bmatrix} * \begin{bmatrix} \mathbf{c}_{[w']} \\ \mathbf{c}_{[w'']} \\ \mathbf{c}_{[w]} \\ \boldsymbol{\lambda}^{\text{inKo}} \\ \boldsymbol{\lambda}^{\text{Rand}} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{[w']} \\ \mathbf{A}_{[w'']} \\ \mathbf{A}_{[w]} \\ \mathbf{0} \\ \mathbf{b}^{\text{Rand}} \end{bmatrix}$$

(6.2)

6.2.1 Virtuelle Durchbiegungen aus Winkelneigungsmessungen

Unter der Prämisse vorliegender Winkelneigungsmessungen ist im ersten Schritt am entkoppelten System (6.2) zu lösen:

$$\mathbf{B}_{[w']} * \mathbf{c}_{[w']} = \mathbf{A}_{[w']} \tag{6.2-1}$$

Da hier die einzelnen Abschnitte der Struktur nicht mehr gekoppelt sind, bedeutet dies die sukzessive Lösung an jedem Abschnitt k. System- und lastseitige Matrix sind in Abschnitt 4 (4.3) definiert worden; hier erfolgt der Hinweis auf die nicht besetzte 1. Zeile und Spalte, so dass unbedingt 1 Zusatzbedingung für die Durchbiegung vorliegen muss. Zur Verbesserung der Lösungen aus (6.2-1) können die Vorteile der Lagrange'schen Multiplikatoren eingesetzt werden, indem im Abschnitt k bereits bekannte Bedingungen aus Randlagerung, aber auch der Messung zusätzlich eingeführt werden.

Damit wird aus (6.2-1):

$$\begin{bmatrix} \bar{\mathbf{B}}_k[\mathbf{w}'] & \bar{\mathbf{R}}_k^T[\mathbf{w}']^{\text{Rand}} \\ \bar{\mathbf{R}}_k[\mathbf{w}']^{\text{Rand}} & \mathbf{0} \end{bmatrix} * \begin{bmatrix} \bar{\mathbf{c}}_k[\mathbf{w}'] \\ \bar{\lambda}_k[\mathbf{w}'] \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{A}}_k[\mathbf{w}'] \\ \bar{\mathbf{b}}_k[\mathbf{w}'] \end{bmatrix} \quad (6.2-1a)$$

Die 1. Zeile der Matrix $\bar{\mathbf{B}}_k[\mathbf{w}']$ muss mit einer Durchbiegungsbedingung belegt werden, den Nebendiagonalen $\bar{\mathbf{R}}_k[\mathbf{w}']^{\text{Rand}}$ bzw. $\bar{\mathbf{R}}_k^T[\mathbf{w}']^{\text{Rand}}$ werden restliche bekannte Werte im Abschnitt k zugewiesen. Nach Lösen von (6.2-1a), das nicht mehr voll-symmetrisch sein muss, sind die jeweiligen Abschnittsparameter $\mathbf{c}_1 \div \mathbf{c}_5$ und damit auch (näherungsweise) die Durchbiegungen entsprechend der Ansatzfunktion nach Bernoulli bekannt:

$$w(\xi) = c_1 + c_2 \xi + c_3 \xi^2 + c_4 \xi^3 + c_5 \xi^4 \quad (6.3)$$

Mit (6.3) sind also die gemessenen Winkelneigungen zu *virtuellen Durchbiegungen* je Abschnitt k transformiert worden und können nun lastseitig für den weiteren Lösungsweg zusätzlich verwendet werden.

6.2.2 Virtuelle Durchbiegungen aus Dehnungsmessungen

Unter der Prämisse vorliegender Dehnungsmessungen ist im weiteren Schritt am entkoppelten System (6.2) zu lösen:

$$\mathbf{B}[\mathbf{w}''] * \mathbf{c}[\mathbf{w}''] = \mathbf{A}[\mathbf{w}''] \quad (6.2-2)$$

Da hier die einzelnen Abschnitte der Struktur nicht mehr gekoppelt sind, bedeutet dies die sukzessive Lösung an jedem Abschnitt k. System- und lastseitige Matrix sind in Abschnitt 4 (4.4) definiert worden; hier erfolgt der Hinweis auf die nicht besetzte 1. und 2. Zeile und Spalte, so dass unbedingt 2 Zusatzbedingungen für 2 Durchbiegungen bzw. 1 Durchbiegung und 1 Winkelneigung vorliegen müssen. Zur Verbesserung der Lösungen aus (6.2-2) können die Vorteile der Lagrange'schen Multiplikatoren eingesetzt werden, indem im Abschnitt k bereits bekannte Bedingungen aus Randlagerung, aber auch der Messung zusätzlich eingeführt werden.

Damit wird aus (6.2-2):

$$\begin{bmatrix} \bar{\mathbf{B}}_k[\mathbf{w}''] & \bar{\mathbf{R}}_k^T[\mathbf{w}'']^{\text{Rand}} \\ \bar{\mathbf{R}}_k[\mathbf{w}'']^{\text{Rand}} & \mathbf{0} \end{bmatrix} * \begin{bmatrix} \bar{\mathbf{c}}_k[\mathbf{w}''] \\ \bar{\lambda}_k[\mathbf{w}''] \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{A}}_k[\mathbf{w}''] \\ \bar{\mathbf{b}}_k[\mathbf{w}''] \end{bmatrix} \quad (6.2-2a)$$

Nachdem die 1. und 2. Zeile der Matrix $\bar{\mathbf{B}}_k[\mathbf{w}'']$ - wie beschrieben - belegt sind, werden den Nebendiagonalen $\bar{\mathbf{R}}_k[\mathbf{w}'']^{\text{Rand}}$ bzw. $\bar{\mathbf{R}}_k^T[\mathbf{w}'']^{\text{Rand}}$ restliche bekannte Werte im Abschnitt k zugewiesen. Nach Lösen von (6.2-2a), das nicht mehr voll-symmetrisch sein muss, sind die jeweiligen Abschnittsparameter $\mathbf{c}_1 \div \mathbf{c}_5$ und damit auch (näherungsweise) die Durchbiegungen entsprechend der Ansatzfunktion nach Bernoulli bekannt.

Mit (6.3) sind also die gemessenen Dehnungen zu *virtuellen Durchbiegungen* je Abschnitt k transformiert worden und können nun lastseitig für den weiteren Lösungsweg zusätzlich verwendet werden.

6.2.3 Residuale Lösung

Nach der Transformation der „höheren“ Messwerte in Durchbiegungen \mathbf{w} ist in (6.2) das residuale System zu lösen, also bleibt:

$$\begin{bmatrix} \mathbf{B}[\mathbf{w}] & \mathbf{R}^T[\mathbf{w}]^{\text{inKo}} & \mathbf{R}^T[\mathbf{w}]^{\text{Rand}} \\ \mathbf{R}[\mathbf{w}]^{\text{inKo}} & \mathbf{0} & \mathbf{0} \\ \mathbf{R}[\mathbf{w}]^{\text{Rand}} & \mathbf{0} & \mathbf{0} \end{bmatrix} * \begin{bmatrix} \mathbf{c}[\mathbf{w}] \\ \boldsymbol{\lambda}^{\text{inKo}} \\ \boldsymbol{\lambda}^{\text{Rand}} \end{bmatrix} = \begin{bmatrix} \mathbf{A}[\mathbf{w}] \\ \mathbf{0} \\ \mathbf{b}^{\text{Rand}} \end{bmatrix} \quad (6.2-3)$$

Hier wird die system- und lastseitige Matrix gebildet aus den unterschiedlichen Messanteilen:

$$\mathbf{B}[\mathbf{w}] = \mathbf{B}_{\text{Durchbiegung } \mathbf{w}}[\mathbf{w}] + \mathbf{B}_{\text{Winkelneigung } \mathbf{w}'}[\mathbf{w}] + \mathbf{B}_{\text{Dehnung } \boldsymbol{\varepsilon}}[\mathbf{w}]$$

oder

$$\mathbf{B}[\mathbf{w}] = \mathbf{B}_{\mathbf{w}} + \mathbf{B}_{\mathbf{w}'} + \mathbf{B}_{\boldsymbol{\varepsilon}}$$

und ebenso

$$\mathbf{A}[\mathbf{w}] = \mathbf{A}_{\text{Durchbiegung } \mathbf{w}} + \mathbf{A}_{\text{Winkelneigung } \mathbf{w}'} + \mathbf{A}_{\text{Dehnung } \boldsymbol{\varepsilon}}$$

oder

$$\mathbf{A}[\mathbf{w}] = \mathbf{A}_{\mathbf{w}} + \mathbf{A}_{\mathbf{w}'} + \mathbf{A}_{\boldsymbol{\varepsilon}}$$

In den Matrizen ist dann entsprechend der Zahl der Messwerte die Summation zu führen

- für Durchbiegungsmesswerte \mathbf{w} in $\mathbf{B}_{\mathbf{w}}$ und $\mathbf{A}_{\mathbf{w}}$ $i=1, N_1$
- für Winkelneigungsmesswerte \mathbf{w}' in $\mathbf{B}_{\mathbf{w}'}$ und $\mathbf{A}_{\mathbf{w}'}$ $i=1, N_2$
- für Dehnungsmesswerte $\boldsymbol{\varepsilon}$ in $\mathbf{B}_{\boldsymbol{\varepsilon}}$ und $\mathbf{A}_{\boldsymbol{\varepsilon}}$ $i=1, N_3$

(6.2-3) geht also über zu:

$$\begin{bmatrix} \mathbf{B}_{\mathbf{w}} + \mathbf{B}_{\mathbf{w}'} + \mathbf{B}_{\boldsymbol{\varepsilon}} & \mathbf{R}^T[\mathbf{w}]^{\text{inKo}} & \mathbf{R}^T[\mathbf{w}]^{\text{Rand}} \\ \mathbf{R}[\mathbf{w}]^{\text{inKo}} & \mathbf{0} & \mathbf{0} \\ \mathbf{R}[\mathbf{w}]^{\text{Rand}} & \mathbf{0} & \mathbf{0} \end{bmatrix} * \begin{bmatrix} \mathbf{c}[\mathbf{w}] \\ \boldsymbol{\lambda}^{\text{inKo}} \\ \boldsymbol{\lambda}^{\text{Rand}} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{\mathbf{w}} + \mathbf{A}_{\mathbf{w}'} + \mathbf{A}_{\boldsymbol{\varepsilon}} \\ \mathbf{0} \\ \mathbf{b}^{\text{Rand}} \end{bmatrix} \quad (6.2-3a)$$

Die Systemmatrix ist voll-symmetrisch. Nach dieser Aufbereitung können die gesuchten Parameter \mathbf{c} mit (6.2-3a) in toto berechnet werden. Durch die zusätzliche Berücksichtigung der Lagrange'schen Multiplikatoren kommt es nicht zu einer vollständigen Entkopplung des Systems. Das residuale System ist übersichtlich, leicht kontrollierbar und die Zahl der Freiheitsgrade erheblich geringer. Der Lösungsweg wird künftig auch als **Zwischenschritt**-Lösung bezeichnet und stellt - im Vergleich zur Gesamtschritt-Lösung - eine Näherung dar, die nicht immer zum Erfolg führt.

7 Grundsätze zur Auswahl der Messpunkte

Ziel der Untersuchungen ist, eine Tragwerksanalyse aus der Messung durchzuführen; hier gilt zu beachten, dass bisher auf die korrekte Auswahl der Messpunkte noch nicht eingegangen ist.

7.1 Prinzip von St. Vénant

Als Regressionsfunktion ist i. w. der Verformungsansatz nach Bernoulli gewählt worden, der den Krümmungsansatz beinhaltet. Die erhaltene Lösung stellt dann die Verformung der neutralen Linie der (Balken-) Struktur dar, d. h. es liegt hier ein Ebenbleiben des Querschnitts mit der zugehörigen Spannungsverteilung vor.

Bei der Auswahl der Messpunkte ist auf diese Grundvoraussetzungen der Biegetheorie nach Bernoulli unbedingt zu achten; d. h. im Bereich von Lastkonzentrationen oder punktuellen Verformungsbehinderungen gilt hier, das Prinzip von St. Vénant stringent anzuwenden. Ein aussagefähiger Messpunkt muss dann genügend weit entfernt von dieser Einleitungszone liegen. Wird das Prinzip verletzt, erhält man zwar aussagefähige Messwerte hinsichtlich lokaler Spannungs- bzw. Dehnungskonzentrationen, für die folgende Tragwerksanalyse sind diese dann sinnvoller Weise nicht verwertbar.

7.2 Konstruktionsart

An der einfachsten Konstruktion - dem Balken auf 2 Stützen unter Gleichlast - sollen die wesentlichen Merkmale für die Wahl der Messpunkte erläutert werden.

7.2.1 Zwängungsfreie Biegung

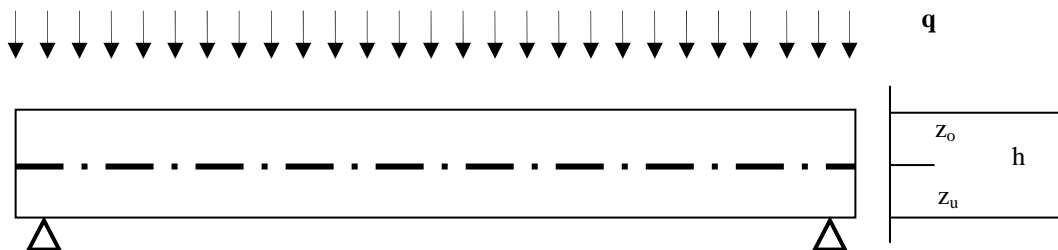


Bild 7.1: Reine Biegung

Bei der reinen Biegung - das ist ein Tragwerk ohne inneren Zwang - lässt sich das Tragwerksverhalten durch wenige Messwerte der

1. Durchbiegungen w und / oder
2. Winkelneigungen w' und / oder
3. Dehnungen ε mit dem bekannten Abstand z von der neutralen Achse

ermitteln; bei Winkelneigung und Dehnung sind die jeweiligen Randbedingungen hinzuzufügen. Die Dehnungswerte [an beliebiger Stelle unter Beachtung von Abschnitt 7.1] können direkt in die Rechnung einfließen. Für alle Messgrößen liegen keine besonderen Zusatzbedingungen vor.

Nach der Ausgleichsrechnung ist dann das Tragwerksverhalten global bekannt.

7.2.2 Biegung mit innerem Zwang

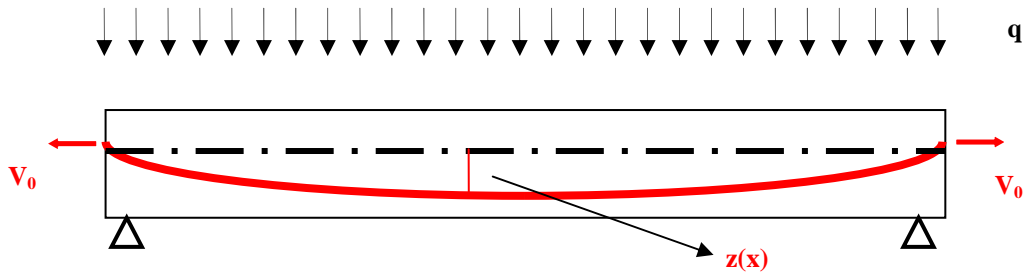


Bild 7.2: Biegung mit beliebiger Vorspannung

Handelt es sich um ein Tragwerk mit innerem Zwang - hier im Beispiel mit beliebiger Vorspannung -, so unterliegen die verschiedenen Messgrößen bestimmten Bedingungen.

Bei den gemessenen Durchbiegungen w und den Winkelneigungen w' liegen dann die gemeinsamen Werte aus der Beanspruchung vor, die in die weitere Rechnung ohne Zusatz eingeführt werden können.

Die Dehnungen ϵ selbst bestehen aus dem Längszwang und der Biegebeanspruchung. Um jedoch den Längszwang zweifelsfrei identifizieren zu können, **müssen über jede Querschnittshöhe mindestens 2 Dehnungswerte** [bei Voraussetzung des Ebenbleibens des Querschnittes] gemessen werden; die Verbindungslinie beider Messpunkte **muss** hierbei die Längsachse vertikal schneiden, da ansonsten noch mehr Messpunkte pro Querschnitt erforderlich wären. Diese Gesamt-Dehnungswerte müssen bei Kenntnis der Querschnittswerte nachträglich zu Biege-Dehnungen umgerechnet werden, mit denen anschließend die Regressionsrechnung durchzuführen ist.

Es werden alle grundsätzlichen Dehnungsvariationen am Querschnitt betrachtet:

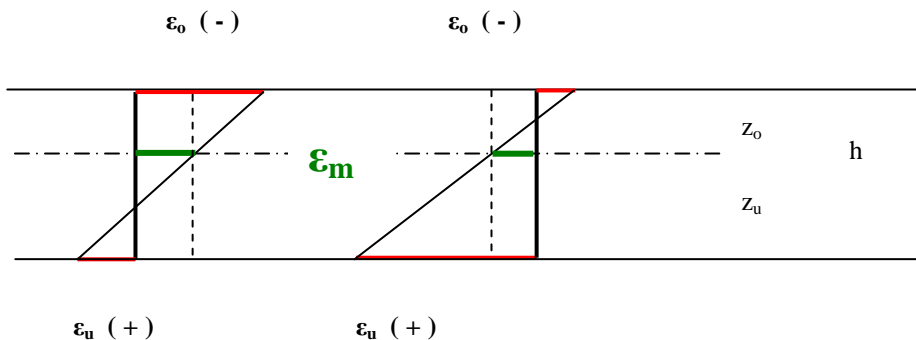


Bild 7.3a: obere Dehnung negativ, untere Dehnung positiv

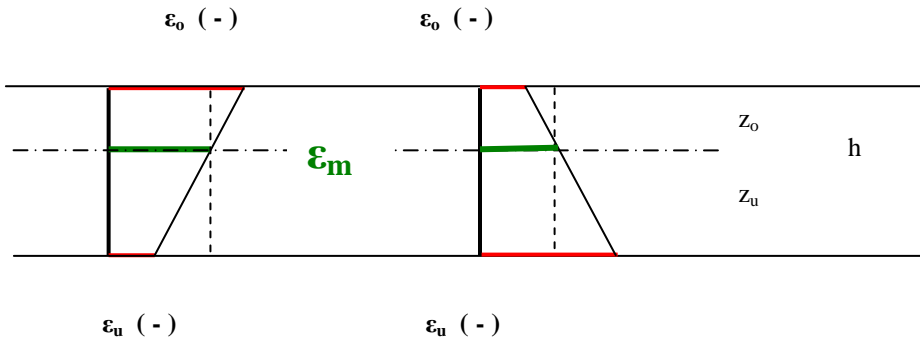


Bild 7.3b: obere und untere Dehnung negativ

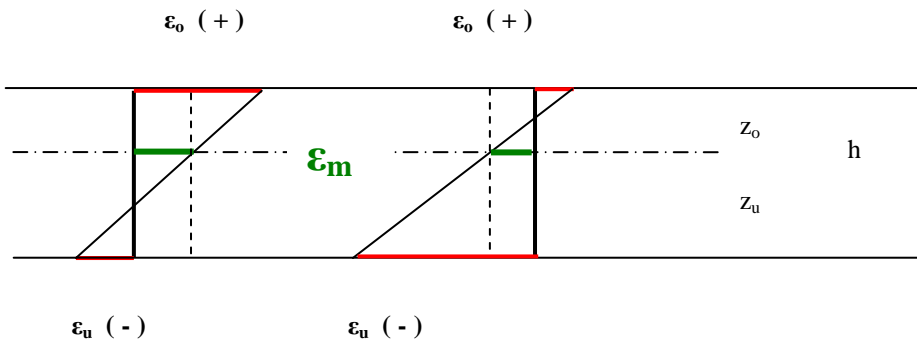


Bild 7.4a: obere Dehnung positiv, untere Dehnung negativ

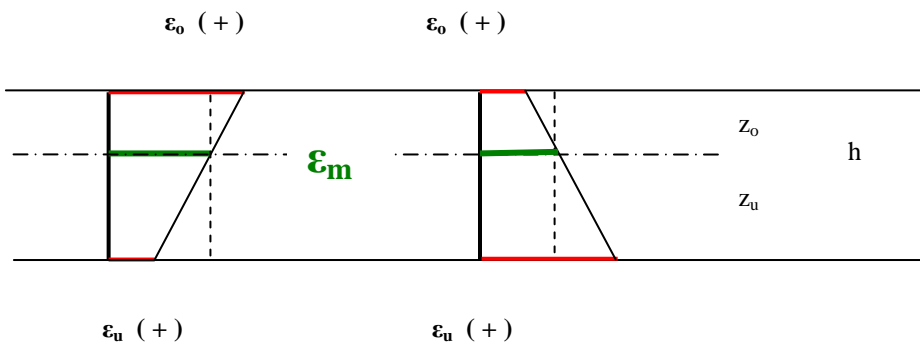


Bild 7.4b: obere und untere Dehnung positiv

Aus den gemessenen Werten ϵ_o und ϵ_u ist mit Kenntnis des Systemquerschnitts die mittlere Längsdehnung ϵ_m aus Zwang zu ermitteln, so dass danach die „reinen“ Gesamt-Biegedehnungen ϵ_o^b und ϵ_u^b bestimmt werden können [s. Tab. 7.1]; hierbei bedeuten „ \langle “ bzw. „ \rangle “ jeweils die Absolutwerte der gemessenen Dehnungen.

gemessene Dehnung oben mit Vorzeichen ϵ_o	gemessene Dehnung unten mit Vorzeichen ϵ_u	Dehnung in neutraler Achse mit Vorzeichen ϵ_m	„reine“ Biegedehnung oben ϵ_o^b	„reine“ Biegedehnung unten ϵ_u^b
negativ $\frac{\langle \epsilon_o \rangle}{\langle \epsilon_u \rangle} \geq \frac{z_o}{z_u}$	positiv	$-\left[\frac{z_u}{h} \langle \epsilon_o \rangle - \frac{z_o}{h} \langle \epsilon_u \rangle \right]$	$-\left[\langle \epsilon_o \rangle - \langle \epsilon_m \rangle \right]$	$+\left[\langle \epsilon_m \rangle + \langle \epsilon_u \rangle \right]$
negativ $\frac{\langle \epsilon_o \rangle}{\langle \epsilon_u \rangle} \leq \frac{z_o}{z_u}$	positiv	$+\left[-\frac{z_u}{h} \langle \epsilon_o \rangle + \frac{z_o}{h} \langle \epsilon_u \rangle \right]$	$-\left[\langle \epsilon_o \rangle + \langle \epsilon_m \rangle \right]$	$+\left[\langle \epsilon_u \rangle - \langle \epsilon_m \rangle \right]$
negativ $\langle \epsilon_o \rangle \geq \langle \epsilon_u \rangle$	negativ	$-\left[\frac{z_u}{h} \langle \epsilon_o \rangle + \frac{z_o}{h} \langle \epsilon_u \rangle \right]$	$-\left[\langle \epsilon_o \rangle - \langle \epsilon_m \rangle \right]$	$+\left[\langle \epsilon_m \rangle - \langle \epsilon_u \rangle \right]$
negativ $\langle \epsilon_o \rangle \leq \langle \epsilon_u \rangle$	negativ	$-\left[\frac{z_u}{h} \langle \epsilon_o \rangle + \frac{z_o}{h} \langle \epsilon_u \rangle \right]$	$+\left[\langle \epsilon_m \rangle - \langle \epsilon_o \rangle \right]$	$-\left[\langle \epsilon_u \rangle - \langle \epsilon_m \rangle \right]$
positiv $\frac{\langle \epsilon_o \rangle}{\langle \epsilon_u \rangle} \geq \frac{z_o}{z_u}$	negativ	$+\left[\frac{z_u}{h} \langle \epsilon_o \rangle - \frac{z_o}{h} \langle \epsilon_u \rangle \right]$	$+\left[\langle \epsilon_o \rangle - \langle \epsilon_m \rangle \right]$	$-\left[\langle \epsilon_m \rangle + \langle \epsilon_u \rangle \right]$
positiv $\frac{\langle \epsilon_o \rangle}{\langle \epsilon_u \rangle} \leq \frac{z_o}{z_u}$	negativ	$-\left[-\frac{z_u}{h} \langle \epsilon_o \rangle + \frac{z_o}{h} \langle \epsilon_u \rangle \right]$	$+\left[\langle \epsilon_o \rangle + \langle \epsilon_m \rangle \right]$	$-\left[\langle \epsilon_u \rangle - \langle \epsilon_m \rangle \right]$
positiv $\langle \epsilon_o \rangle \geq \langle \epsilon_u \rangle$	positiv	$+\left[\frac{z_u}{h} \langle \epsilon_o \rangle + \frac{z_o}{h} \langle \epsilon_u \rangle \right]$	$+\left[\langle \epsilon_o \rangle - \langle \epsilon_m \rangle \right]$	$-\left[\langle \epsilon_m \rangle - \langle \epsilon_u \rangle \right]$
positiv $\langle \epsilon_o \rangle \leq \langle \epsilon_u \rangle$	positiv	$+\left[\frac{z_u}{h} \langle \epsilon_o \rangle + \frac{z_o}{h} \langle \epsilon_u \rangle \right]$	$-\left[\langle \epsilon_m \rangle - \langle \epsilon_o \rangle \right]$	$+\left[\langle \epsilon_u \rangle - \langle \epsilon_m \rangle \right]$

Tab. 7.1: Umrechnung in „reine“ Gesamt-Biegedehnungen

7.3 Belastungsgeschichte

Grundsätzlich sei noch einmal an die (abgelaufene) Last-Zeit-Geschichte erinnert; denn alle gemessenen Werte können sich nur auf den Zeitpunkt beziehen, an dem die Belastungsmessung initiiert wird. Über davor liegende Belastungen - also beispielsweise auch Einleiten der Vorspannung - können dann keine Aussagen mehr gemacht werden.

8 Grundlegende Beispiele

Im Folgenden werden die Lösungsansätze an 2 grundsätzlichen Beispielen demonstriert:

8.1 Beispiel für zwängungsfreie Biegung

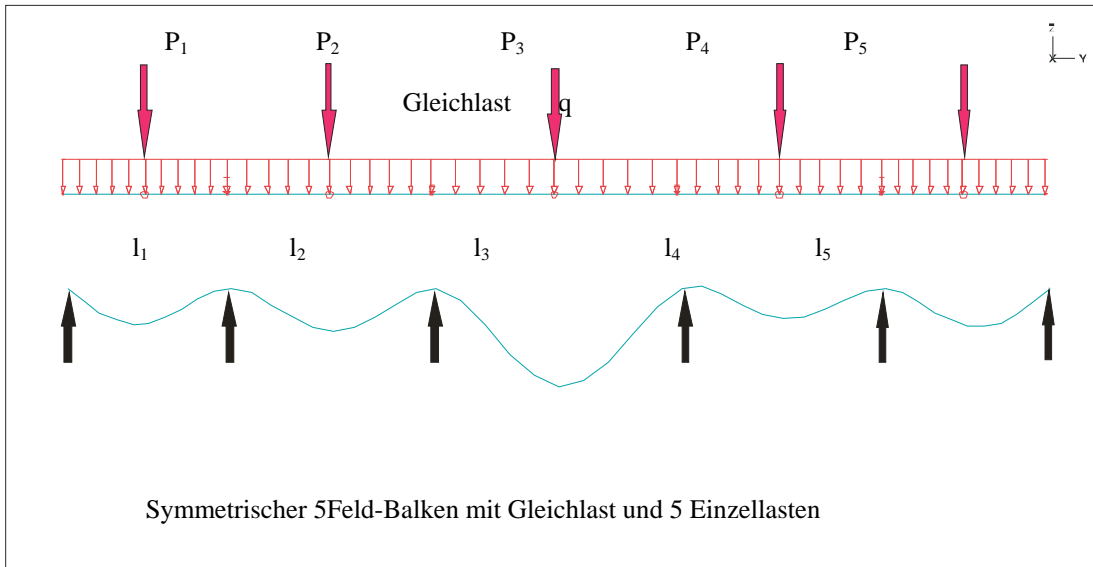


Bild 8.1a: Beispiel für zwängungsfreie Biegung

Ein symmetrischer 5Feld-Balken mit konstantem Querschnitt ist durch symmetrische Einzellasten und eine Gleichlast beansprucht - wie in Tab. 8.1 angegeben.

Feld	Stützweite [mm]	Einzellast [N]	Gleichlast [N/mm]
1	8000	10000	10
2	10000	10000	10
3	12000	10000	10
4	10000	10000	10
5	8000	10000	10

Tab. 8.1: Systemwerte des Beispiels für zwängungsfreie Biegung

Das größere Mittelfeld bewirkt zusammen mit der Belastung, dass jeweils in den Nachbarfeldern ein Anheben der Biegelinie entsteht.

Zur Ermittlung der erforderlichen „Messdaten“ wird die statische Berechnung mit dem Programmcode ADINA durchgeführt, und zwar mit einer Breite $b = 50$ mm und Höhe $h = 2000$ mm sowie einem Elastizitätsmodul von 30000 N/mm². Es werden alle Daten wie Durchbiegungen, Winkelneigungen und Krümmungen bzw. Dehnungen ermittelt.

Die anschließende Regressionsrechnung wird mit der Gesamtschritt-Lösung nach Abschnitt 6.1 durchgeführt; entsprechend Tab. 8.2 werden folgende Varianten untersucht:

Varianten	Abschnitte	Polynomgrad	f f' f''	Durchbiegung w	Winkelneigung w'	Krümmung w''	Randbedin- gungen
1	2	3	4	5	6	7	8
ADINA							
1	5	4	3	11-11-11-11-11	-	-	8-1-2
2	5	4	3	-	11-11-11-11-11	-	8-1-2
3	5	4	3	-	-	11-11-11-11-11	8-1-2
4	5	4	3	-	5-5-5-5-5	6-6-6-6-6	8-1-2
5	5	4	3	3-3-1-3-3	3-3-3-3-3	6-6-6-6-6	8-1-2
6	5	4	3	3-3-1-3-3	3-3-3-3-3	6-6-6-6-6	-
7	5	4	3	11-11-11-11-11	-	-	-

Tab. 8.2: Untersuchte Varianten am 5Feld-Balken

In den 5 Abschnitten besteht die Last sowohl aus Einzel- als auch aus Gleichlast, so dass der Polynomansatz 4. Grades (Spalte 3) gewählt wird. Alle 3 Übergangsbedingungen der Einzelfunktionen sollen Kontinuität ausweisen (Spalte 4).

Zahl und Art der jeweiligen „Messpunkte“ sind in Spalte 5 - 7 verzeichnet; hier ist feldweise die jeweilige Zahl der verwendeten Messpunkte aufgeführt. Die Randbedingungen sind in der Qualität in Spalte 8 angegeben, und zwar der Reihe nach für Durchbiegung, Winkelneigung und Krümmung.

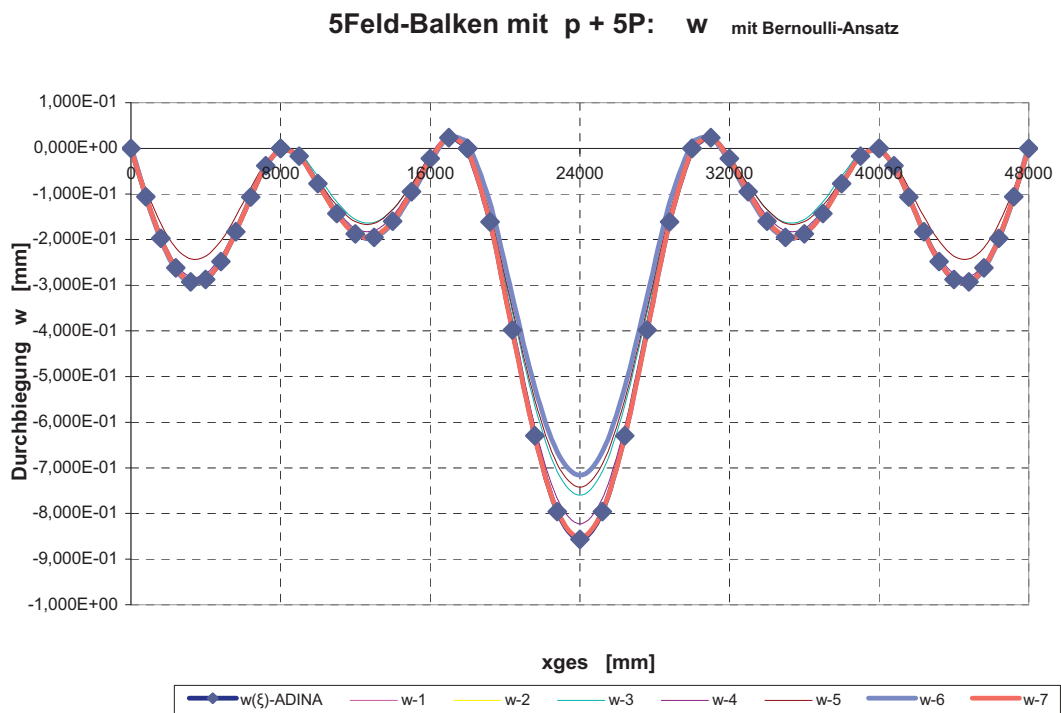


Bild 8.1b: Beispiel für zwängungsfreie Biegung

5Feld-Balken mit $p + 5P$: w' mit Bernoulli-Ansatz

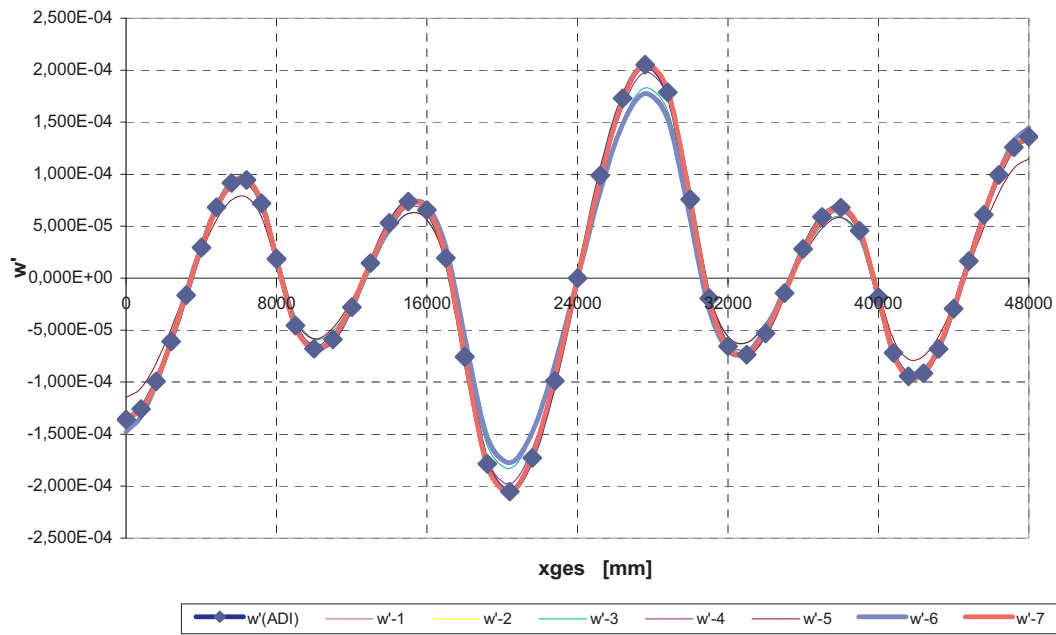


Bild 8.1c: Beispiel für zwangungsfreie Biegung

5Feld-Balken mit $p + 5P$: w'' mit Bernoulli-Ansatz

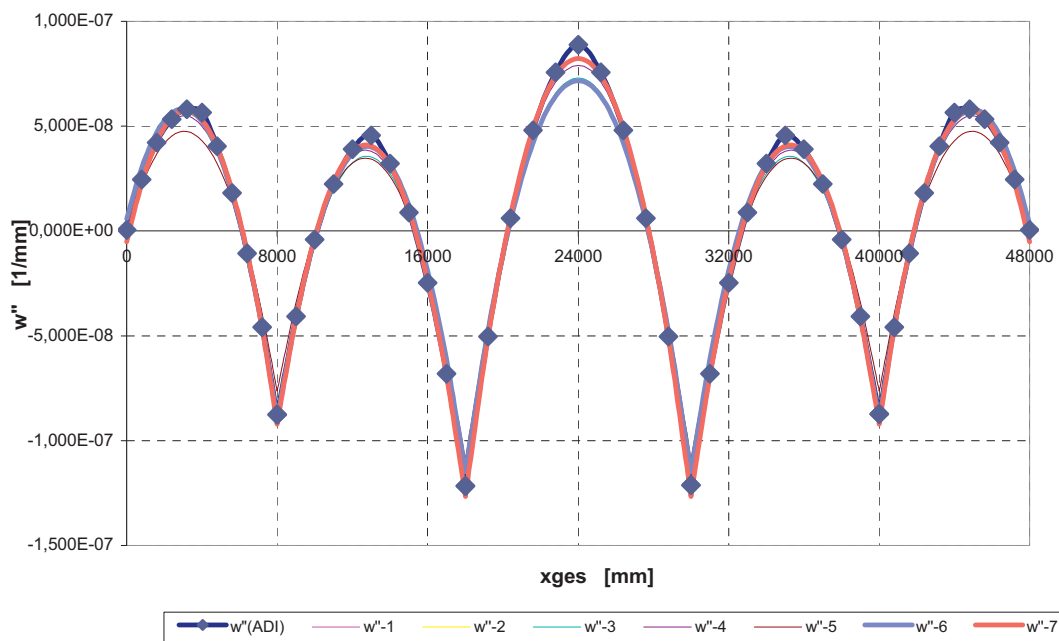


Bild 8.1d: Beispiel für zwangungsfreie Biegung

Grundsätzlich wird gezeigt, dass das Strukturverhalten des globalen Systems mit genügender Zahl der Messwerte und/oder zusätzlicher Randbedingungen dargestellt werden kann; den Unterschied macht die Güte der Übereinstimmung mit der ADINA-Rechnung:

- Variante 7 zeigt gute Korrelation zur ADINA-Ausgangslösung; hier liegt eine genügende Zahl von Durchbiegungswerten vor, so dass Randbedingungswerte nicht erforderlich sind.
- Variante 1 erhält zusätzliche Randbedingungen und führt konsequenterweise zur vorzüglichen Übereinstimmung mit der ADINA-Rechnung.
- Variante 2, deren Messwerte ausschließlich aus Winkelneigungen bestehen, liefert ebenfalls eine gleichartige Übereinstimmung wie zuvor.
- Variante 3 mit dem Unterschied zu Variante 2 besteht ausschließlich aus Krümmungs- bzw. Dehnungswerten (der reinen Biegeanteile). Hier zeigt sich schon eine deutlich schlechtere Übereinstimmung; die zweite Ableitung (der Krümmung aus der Durchbiegung) macht sich klar bemerkbar.
- Variante 4 mit einer Mischung von Winkelneigung und Krümmung liefert bessere Übereinstimmung trotz kleinerer Zahl für jeden Typ der Messwerte.
- Reduziert man die Messwertezahl noch stärker, so gibt es weit schlechtere Übereinstimmung für das globale System.

Um den Einfluss der erforderlichen Zahl der Messwerte noch etwas genauer zu analysieren, wird das Mittelfeld des 5-Feld-Balkens noch einmal gesondert betrachtet:

Varianten	Abschnitte	Polynom-grad	$\frac{f}{f'}$ $\frac{f'}{f''}$	Durchbiegung w	Winkelneigung w'	Krümmung w''	Randbedin- gungen
1	2	3	4	5	6	7	8
ADINA							
1	1	4	-	11			2-0-0
2	1	4	-		11		2-0-0
3	1	4	-			11	2-0-0
4	1	4	-			6	2-0-0
5	1	4	-			5	2-0-0
6	1	4	-			3	3-2-0
7	1	4	-	9			3-2-0
8	1	4	-			2	3-2-0
9	1	4	-			2	2-2-0
10	1	4	-	3	3	2	-
11	1	4	-				3-2-0*
12	1	4	-	3			-
13	1	4	-	3			0-2-0

Tab. 8.3: Untersuchte Varianten am Mittelfeld des 5Feld-Balkens

Mittelfeld 5Feld-Balken mit p + 5P: w mit Bernoulli-Ansatz

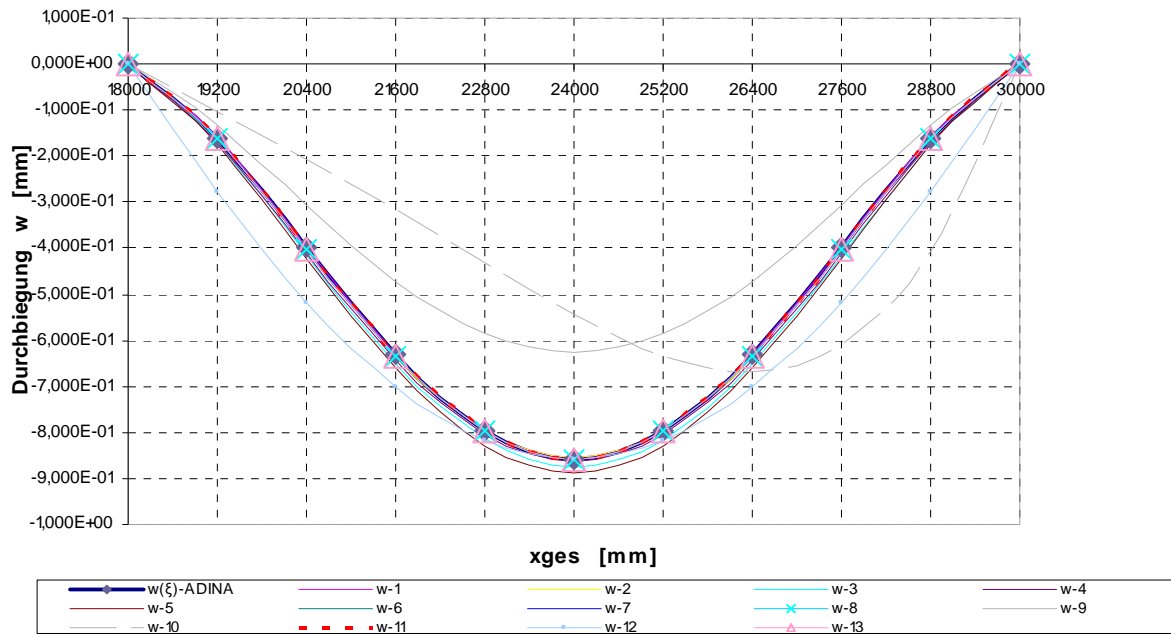


Bild 8.2a: Beispiel für zwangungsfreie Biegung

Mittelfeld 5Feld-Balken mit p + 5P: w' mit Bernoulli-Ansatz

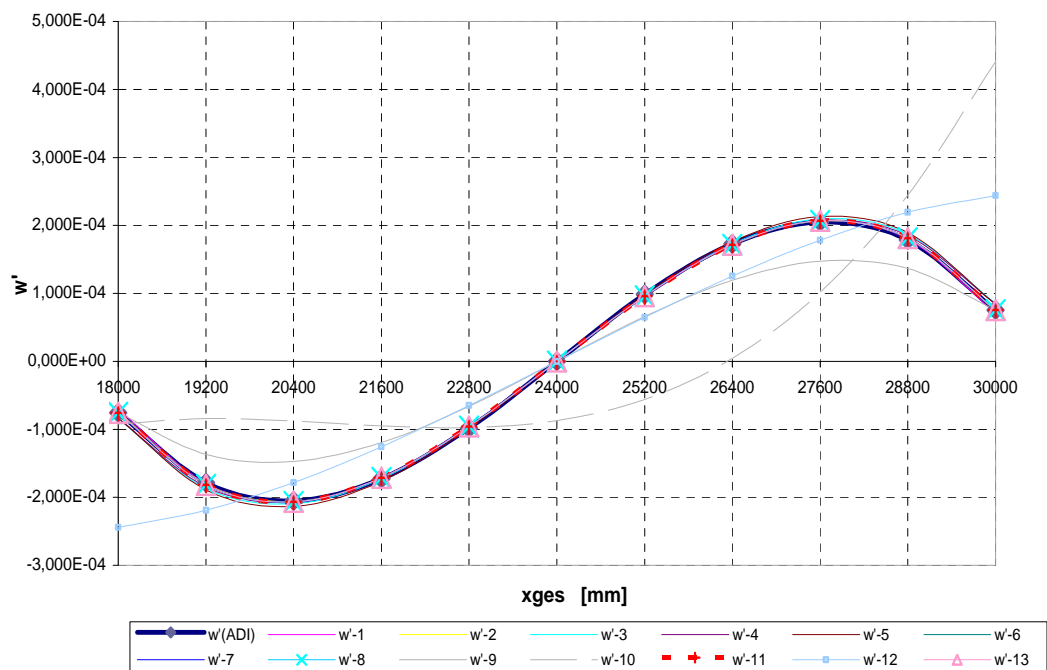


Bild 8.2b: Beispiel für zwangungsfreie Biegung

Mittelfeld 5Feld-Balken mit $p + 5P$: w'' mit Bernoulli-Ansatz

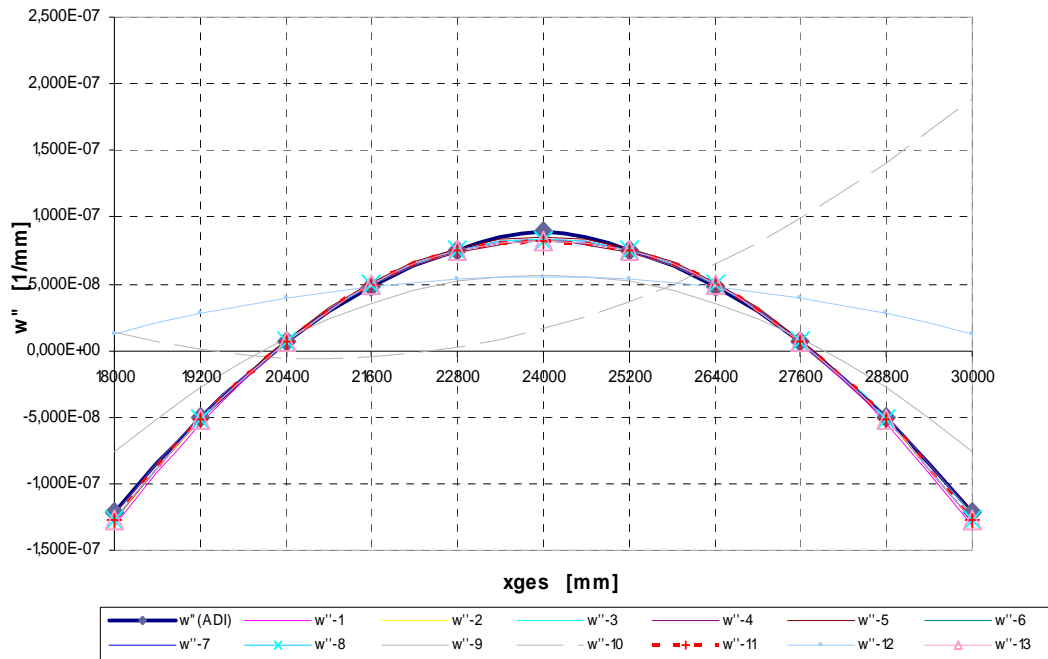


Bild 8.2c: Beispiel für zwangungsfreie Biegung

- Varianten 1,2 und 3 weisen jeweils eine genügende Zahl von Messpunkten aus, wobei für Winkelneigungen und Krümmungen die Randbedingungen notwendig zur stabilen Lösung sind.
- Bei den Varianten 4 und 6 wird die Zahl der gemessenen Krümmungen systematisch reduziert; es wird für alle Varianten eine gute Übereinstimmung erzielt; selbst Variante 5 mit minimaler Datenzahl führt noch zu einer genügenden Korrelation.
- Liegen die Randbedingungen „komplett“ vor, so erhält man ebenfalls eine sehr gute Übereinstimmung (Varianten 6, 7, 8 und der Sonderfall 11).
- Varianten 9 und 10 ergeben ein falsches Resultat; es liegen zu wenige Daten bzw. Randbedingungen vor.
- Variante 12 kann wegen fehlender Randbedingungen bei zu wenigen Durchbiegungswerten die Krümmung an den Auflagern nur ungenügend darstellen.

8.2 Beispiel für Biegung mit innerem Zwang

Es wird entsprechend Bild 8.3 ein 1Feld-Balken unter Gleichlast gewählt, der mit der konstanten Vorspannkraft V_0 bei variabler Funktion $z(x)$ für den Verlauf des Spanngliedes zusätzlich vorgespannt ist. Der wesentliche Unterschied zur zwängungsfreien Biegung ist hier, dass in genügender Zahl jeweils 2 Dehnungsmessungen über die Querschnittshöhe - also doppelt so viele Messungen wie bei der reinen Biegung - vorliegen müssen, da andernfalls keine zielsichere Aussage über den Gesamtzustand des Tragwerkes gemacht werden kann. Diese Dehnungen bzw. Krümmungen müssen dann vor der weiteren (Regressions-)Analyse entsprechend Tab. 7.1 in *Gesamt-Biegekrümmungen umgerechnet* werden.

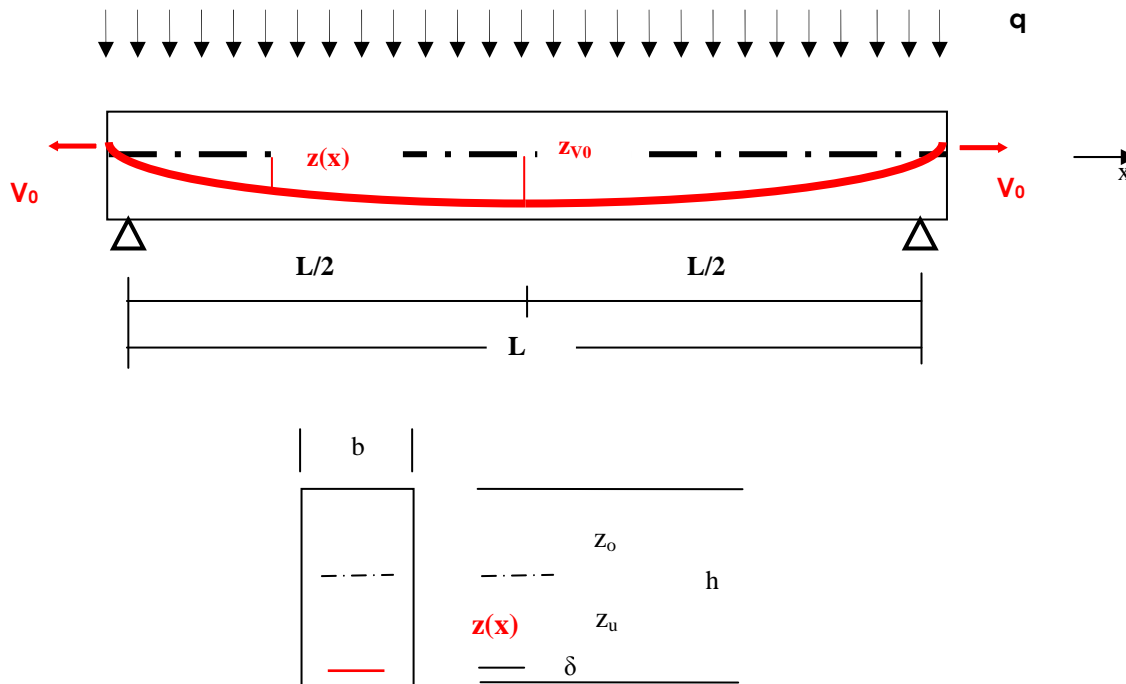


Bild 8.3: Beispiel für Biegung mit innerem Zwang

Die maximale Exzentrizität des Spanngliedes sei in Feldmitte $z_{V0} = z_u - \delta$. Mit der Forderung, dass die Vorspannung entsprechend der Biegebeanspruchung verlaufe, wird die Spanngliedkraft

$$V_0 = \alpha (1/z_{V0}) q L^2 / 8$$

und deren Exzentrizität gegenüber der Nullachse zu

$$z(x) = (4 z_{V0} / L^2) x (L - x) \quad .$$

Zur Ermittlung der „Messdaten“ werden folgende Vorgaben gemacht:

- Stützweite $L = 12,00$ m
- Breite $b = 0,30$ m
- Höhe $h = 1,00$ m
- max. Spanngliedexzentrizität $z_{V0} = 0,40$ m

Der Elastizitätsmodul sei 30000 MN/m^2 .--- Die Auflast wird gesetzt zu $q = 0,1 \text{ MN/m}$.

Der Faktor α symbolisiert den Grad der Vorspannung.

8.2.1 Fall 1: Volle Vorspannung

Im Fall 1 der vollen Vorspannung ist der Faktor $\alpha = 1$.

Damit ergibt sich die Vorspannkraft $V_0 = 1/0,4 * 0,1 * 12^2 / 8$ zu 4,5 MN, woraus eine mittlere Spannung $\sigma_m = -15 \text{ MN/m}^2$ und eine mittlere Dehnung $\epsilon_m = -0,5 * 10^{-5}$ folgen. Die Biegelinie $w(x)$ und deren Winkelneigung $w'(x)$ sind in diesem Fall identisch Null.

Die Dehnungsmessstreifen werden nach Erhärten des Betons vor Aufbringen der Vorspannung angebracht, und zwar werden sie über die Balkenlänge - neben Messstellen der Biegelinie und Winkelneigungen - in genügend vielen Querschnitten jeweils doppelt installiert, z. B. ϵ_o und ϵ_u entsprechend Bilder 7.3 und 7.4.

Aus Tab. 7.1 kann dann mit der 3. bzw. 4. Zeile die mittlere Dehnung ϵ_m berechnet werden. Da die „gemessenen“ oberen und unteren Dehnungen ϵ_o und ϵ_u hier dem Fall 1 identisch sind, werden die reinen Biegedehnungen ϵ_o^b und ϵ_u^b zu Null; konsequenterweise sind auch die Durchbiegungen bzw. Winkelneigungen Null, jedoch **ohne** Kenntnis der Längsdehnung ϵ_m könnte keine Aussage gemacht werden, dass es sich im Fall 1 um ein Tragwerk mit innerem Zwang handelt.

$\alpha=1,0$			
	"gemessen"	"gemessen"	umgerechnet
xges	w(1,0)	w'(1,0)	$\epsilon-o-b(1,0)$
0	0,000000E+00	0,000000E+00	0,000000E+00
1200	0,000000E+00	0,000000E+00	0,000000E+00
2400	0,000000E+00	0,000000E+00	0,000000E+00
3600	0,000000E+00	0,000000E+00	0,000000E+00
4800	0,000000E+00	0,000000E+00	0,000000E+00
6000	0,000000E+00	0,000000E+00	0,000000E+00
7200	0,000000E+00	0,000000E+00	0,000000E+00
8400	0,000000E+00	0,000000E+00	0,000000E+00
9600	0,000000E+00	0,000000E+00	0,000000E+00
10800	0,000000E+00	0,000000E+00	0,000000E+00
12000	0,000000E+00	0,000000E+00	0,000000E+00

Tab. 8.4a: Volle Vorspannung---anwendbare „Messdaten“

Die in Tab. 8.4a dargestellten Werte können dann als „Messdaten“ in die weitere Regressionsanalyse eingeführt werden, die dann formal wie in Abschnitt 8.1 zu vollziehen ist.

8.2.2 Fall 2: Teilweise Vorspannung

Wird der Faktor $\alpha < 1$ gesetzt, handelt es sich um eine Teilvorspannung. In Fortsetzung des Beispiels bedeutet dies, dass die Biegebeanspruchung aus der Auflast nur mit dem Faktor α kompensiert wird. Wie in Punkt 8.2.1 sind die über die Querschnittshöhe gemessenen Dehnungen für die weitere Analyse in reine Biegedehnungen umzurechnen.

$\alpha=0,8$			
	"gemessen"	"gemessen"	umgerechnet
xges	w(0,8)	w'(0,8)	$\epsilon-o-b(0,8)$
0	0,000000E+00	1,920000E-03	0,000000E+00
1200	2,260224E+00	1,812480E-03	-8,640000E-05
2400	4,276224E+00	1,520640E-03	-1,536000E-04
3600	5,854464E+00	1,090560E-03	-2,016000E-04
4800	6,856704E+00	5,683200E-04	-2,304000E-04
6000	7,200000E+00	0,000000E+00	-2,400000E-04
7200	6,856704E+00	-5,683200E-04	-2,304000E-04
8400	5,854464E+00	-1,090560E-03	-2,016000E-04
9600	4,276224E+00	-1,520640E-03	-1,536000E-04
10800	2,260224E+00	-1,812480E-03	-8,640000E-05
12000	0,000000E+00	-1,920000E-03	0,000000E+00

Tab. 8.4b: Teilweise Vorspannung---anwendbare „Messdaten“

Die in Tab. 8.4b dargestellten Werte können dann als „Messdaten“ in die weitere Regressionsanalyse eingeführt werden, die dann formal wie in Abschnitt 8.1 zu vollziehen ist.

8.2.3 Fall 3: Überhöhte Vorspannung

$\alpha=1,2$			
	"gemessen"	"gemessen"	umgerechnet
xges	w(1,2)	w'(1,2)	$\epsilon-o-b(1,2)$
0	0,000000E+00	-1,920000E-03	0,000000E+00
1200	-2,260224E+00	-1,812480E-03	8,640000E-05
2400	-4,276224E+00	-1,520640E-03	1,536000E-04
3600	-5,854464E+00	-1,090560E-03	2,016000E-04
4800	-6,856704E+00	-5,683200E-04	2,304000E-04
6000	-7,200000E+00	0,000000E+00	2,400000E-04
7200	-6,856704E+00	5,683200E-04	2,304000E-04
8400	-5,854464E+00	1,090560E-03	2,016000E-04
9600	-4,276224E+00	1,520640E-03	1,536000E-04
10800	-2,260224E+00	1,812480E-03	8,640000E-05
12000	0,000000E+00	1,920000E-03	0,000000E+00

Tab. 8.4c: Überhöhte Vorspannung---anwendbare „Messdaten“

Wird der Faktor $\alpha > 1$ gesetzt, handelt es sich um eine Vorspannung, die die Biegemomente überwiegt; d. h. das Tragwerk erhält eine Überhöhung. Wie in Punkt 8.2.1 sind die über die Querschnittshöhe gemessenen Dehnungen für die weitere Analyse in reine Biegedehnungen umzurechnen. Die in Tab. 8.4c dargestellten Werte können dann als „Messdaten“ in die weitere Regressionsanalyse eingeführt werden, die dann formal wie in Abschnitt 8.1 zu vollziehen ist.

Alle in Tab. 8.4a - c dargestellten „Messdaten“ sind noch einmal dargestellt und werden mit der reinen Biegebeanspruchung, also mit demselben Lastfall ohne Vorspannung, verglichen.

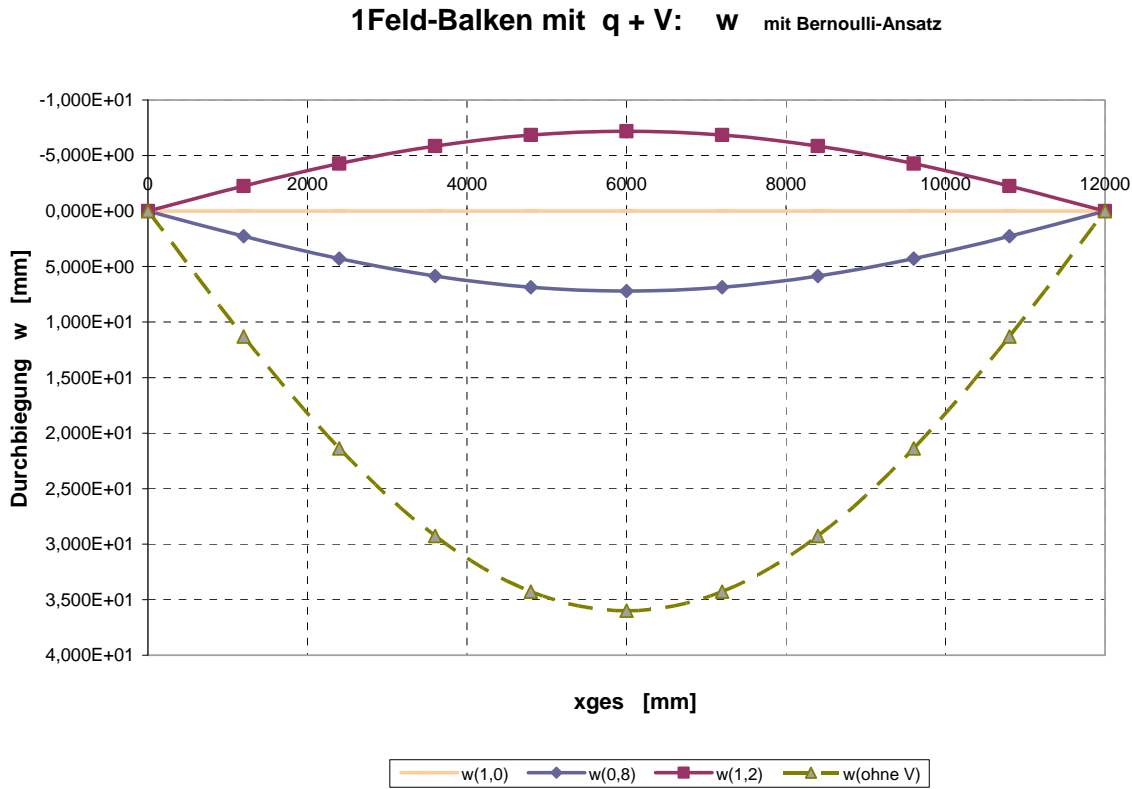


Bild 8.4a: Beispiel für Biegung mit innerem Zwang

1Feld-Balken mit $q + V$: w' mit Bernoulli-Ansatz

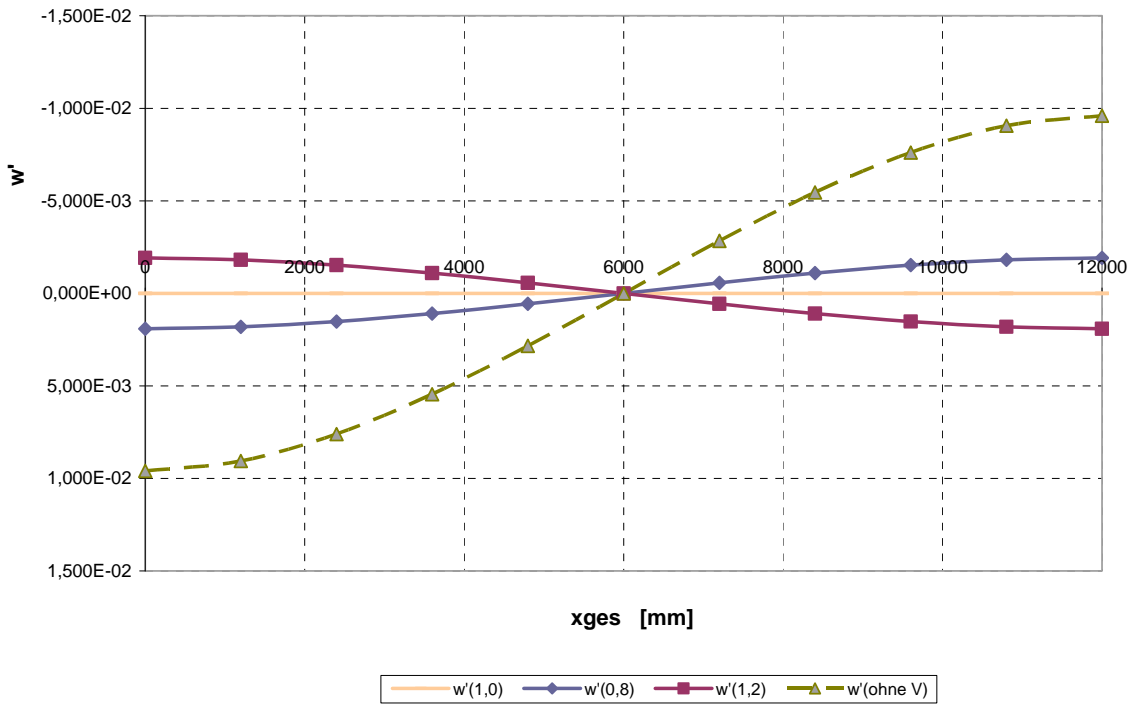


Bild 8.4b: Beispiel für Biegung mit innerem Zwang

1Feld-Balken mit $q + V$: $\epsilon-o-b$ mit Bernoulli-Ansatz

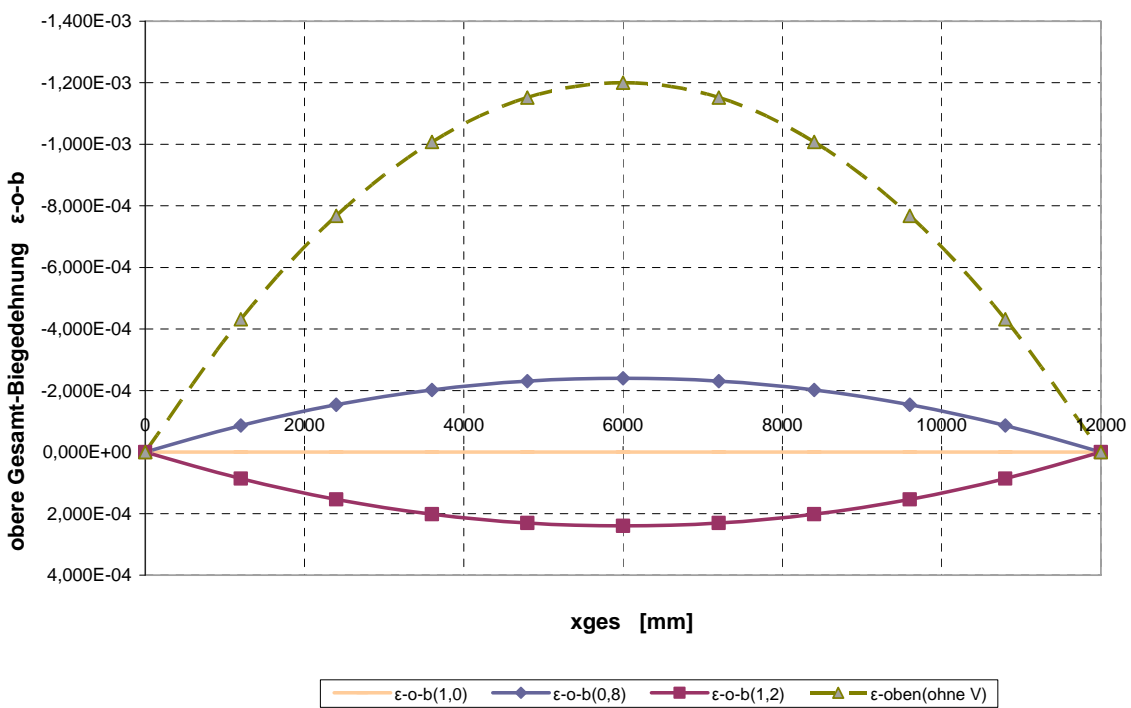


Bild 8.4c: Beispiel für Biegung mit innerem Zwang

9 Fazit und Ausblick

Die Beurteilung des strukturellen Verhaltens eines Tragwerkes wird am Beispiel der Balkenbiegung gezeigt. Der grundsätzliche Ansatz und einige mögliche Variationen werden diskutiert. Es werden zwei Lösungswege aufgezeigt.

Zusätzlich dürfen die Grundsätze zum Anordnen der Messpunkte und deren Messgröße nicht unberücksichtigt bleiben, da ansonsten keine sinnvolle Aussage vorgenommen werden kann.

An grundsätzlichen Beispielen wird der Lösungsweg und das erzielte Resultat angegeben. Hier wird auf die eminent wichtige Rolle der Randbedingungen hingewiesen, die immer zur Stabilisierung des Ergebnisses beitragen. Des Weiteren wird noch einmal die Bandbreite der Lagrange'schen Multiplikatoren untersucht [s. Anhang A] und eine Übersicht zum Formelapparat und Lösungsweg gegeben [s. Anhang B].

Der grundsätzliche Ansatz beinhaltet zunächst keine Aussage über die zu verwendenden Funktionen; d. h. es liegt keine Beschränkung auf ein linienhaftes Tragwerk vor. So kann das Procedere ohne weiteres auf andere Tragwerke - beispielsweise Flächentragwerke - ausgeweitet werden, für die ebenfalls vorzügliche Lösungs- und damit Ansatzfunktionen bekannt sind. Dass der messtechnische Aufwand entsprechend steigt, dürfte selbstverständlich sein.

Eine Ausweitung auf die permanente Strukturüberwachung bietet sich von selbst an, so dass nach Auswertung der Messungen sofort eine Aussage zum Tragverhalten - im Vergleich zur vorliegenden statischen Berechnung - vorgenommen werden kann.

10 Literaturhinweis

- /1/ Encyclopedia of Statistical Sciences
John Wiley & Sons, New York, Vol 4, 1983
- /2/ I. Szabó
Einführung in die Technische Mechanik
Springer Verlag
- /3/ I. Szabó
Höhere Technische Mechanik
Springer Verlag
- /4/ R. Zurmühl
Praktische Mathematik für Ingenieure und Physiker
Springer Verlag
- /5/ R. Zurmühl
Matrizen und ihre technischen Anwendungen
Springer Verlag
- /6/ O. C. Zienkiewicz
The Finite Element Method in Engineering Science
McGraw-Hill 1971
- /7/ K.-J. Bathe, E. L. Wilson
Numerical Methods in Finite Element Analysis
Prentice Hall, Inc., New Jersey
- /8/ S. P. Timoshenko, J. N. Goodier
Theory of Elasticity
McGraw-Hill 1970
- /9/ R. H. Gallagher
Finite-Element Analysis
Springer Verlag, 1976
- /10/ W. Großmann
Grundzüge der Ausgleichsrechnung
Springer Verlag, 1961
- /11/ K.-H. Laermann
Inverse Problems in Experimental Structural Analysis
Shaker Verlag, Aachen, 2008

- /21/ S. D. Silvey
The Lagrangian Multiplier Test
The Annals of Mathematical Statistics , Vol. 30, No. 2. (Jun., 1959), pp. 389-407
- /22/ K. Brandes, K. Berner
Gesamtmatrix-Formulierung für hybride experimentell-analytische Verfahren
Bautechnik 68 (1991), H. 6
- /23/ I. Milev
Integrierte Modelle zur physikalischen Interpretation Geodätischer Deformationsuntersuchungen
Diss. am FB 9 - Bauingenieurwesen und angewandte Geowissenschaften - der TUB, Berlin, 2001
- /24/ H. B. Nielsen
Introduction to Vector and Matrix Differentiation
Econometrics 2, Sept. 21, 2005
- /25/ K. Brandes, W. Daum, F. Buchhardt
Methodische Symbiose von Messung und mechanischer Analyse beim
Langzeit-Monitoring von Spannbetonbrücken
GESA-Symposium 2005 Strukturanalyse
VDI- Berichte Nr. 1899, Düsseldorf, pp. 137 - 151
- /26/ K. Brandes, W. Daum, F. Buchhardt
Monitoring of Bridges by Advanced Methods of Experimental-Mechanical Symbiosis
IABSE Symposium, Weimar, Sept. 19 – 21, 2007
- /27/ Chr. Maierhofer et al.
Complementary application of radar, impact echo and ultrasonics for testing concrete structures and
metallic tendon ducts; Transportation Research Record
Journal of the Transportation Research Board, No. 1892, TRB, National Research Council, Wash-
ington D. C., 2004, pp. 170-177

11 Anhang A: Spezifische Beispiele

1	Erläuterungen zu den Lagrange'schen Multiplikatoren	46
1.1	Beispiele aus der Mechanik	46
1.1.1	Beispiel 1.1 Zugstab	46
1.1.2	Beispiel 1.2 Zugstab mit 2 Federn	47
1.1.3	Beispiel 1.3 Zugstab mit 4 Federn	48
1.2	Beispiele aus der Geometrie	50
1.2.1	Beispiel 2.1 Gerade mit 1 Zwangspunkt	50
1.2.2	Beispiel 2.2 Gerade mit 1 Zwangssteigung	53
1.2.3	Beispiel 2.3 Gerade mit 2 Zwangspunkten	55
1.3	Beispiele aus der Ausgleichsrechnung	58
1.3.1	Beispiel 3.1 Symmetrisches System mit 11 Messpunkten	58
1.3.2	Beispiel 3.2 Schief-symmetrisches System mit 11 Messpunkten	61
2	Beispiele der statischen Anwendung	63
2.1	Matrix-System bei Gesamtschritt- und Zwischenschritt-Lösung	63
2.1.1	Beispiel 2-1.1 3 Felder	64
2.1.2	Beispiel 2-1.2 4 Felder	64
2.1.3	Beispiel 2-1.3 8 Felder	65
2.2	Spezifische Anwendungen	66
2.2.1	Beispiel 2-2.1 3Feld-Balken mit Gleichlast q	66
2.2.2	Beispiel 2-2.2 5Feld-Balken mit Gleichlast q und 5 Einzellasten	69
2.2.3	Beispiel 2-2.3 Sonderfall	74

1 Erläuterungen zu den Lagrange'schen Multiplikatoren

Die Lagrange'schen Multiplikatoren sind immer dimensionsbehaftet, und zwar mit derjenigen Dimension, deren Zusatzbedingung sie darstellen.

Zur Erläuterung hinsichtlich der Einschätzung der Größenordnung der Lagrange'schen Multiplikatoren λ_i werden einige Beispiele gegeben:

1.1 Beispiele aus der Mechanik

1.1.1 Beispiel 1.1 Zugstab

Ein einfacher Zugstab wird untersucht.



Bild A-1-1: Zugstab

Das Matrix-System sei definiert: $\mathbf{G} * \mathbf{u} = \mathbf{P}$ bzw.

u_1	u_2	λ	=	Absolutwert
AE/L	-AE/L	1	=	0
-AE/L	AE/L	0	=	F_2
1	0	0	=	0

Der Zugstab mit der Querschnittsfläche A, dem Elastizitätsmodul E und der Stablänge L wird am Knoten 2 mit der Zugkraft F_2 belastet. Der Knoten 1 sei starr gehalten. Die System-Matrix wird **ohne Berücksichtigung der Randlagerung** definiert; anschließend erfolgt durch Hinzufügen der 3. Zeile die Berücksichtigung der Randbedingung $u_1 = 0$, was die Einführung eines Korrekturfaktors λ - im symmetrischen System - bedeutet. Damit folgen die Lösungswerte zu:

$$\begin{aligned} \lambda &= F_2 && [\text{N}] \\ u_2 &= F_2 * L/AE && [\text{mm}] \\ u_1 &= 0. && [\text{mm}] \end{aligned}$$

Am noch freien System führt die zusätzliche Bedingung $u_1 = 0$ zu dem Korrekturfaktor $\lambda = F_2$, also der Auflagerkraft aus der eingeleiteten Kraft F_2 .

Der Korrekturfaktor λ ist in diesem Fall direkt interpretierbar. Mit $k = AE/L$ als Federkonstante erhält man zur Überprüfung der positiven Definitheit

$$Q = \mathbf{u}^T * \mathbf{G} * \mathbf{u} = \frac{(F_2)^2}{k} \geq 0 \quad \text{und die Determinante } \text{Det} = -k \quad .$$

1.1.2 Beispiel 1.2 Zugstab mit 2 Federn

Ein Zugstab wird mit 2 Federn untersucht.

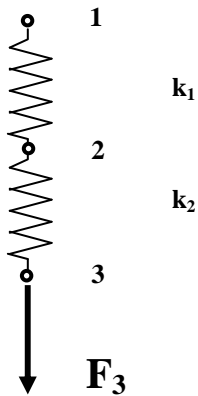


Bild A-1-2: Zugstab mit 2 Federn

Das Matrix-System sei definiert: $\mathbf{G} * \mathbf{u} = \mathbf{P}$ bzw.

u_1	u_2	u_3	λ	=	Absolutwert
k_1	$-k_1$	0	1	=	0
$-k_1$	k_1+k_2	$-k_2$	0	=	0
0	$-k_2$	k_2	0	=	F_3
1	0	0	0	=	0

Der Zugstab als 2 nacheinander geschaltete Federn wird am Knoten 3 mit der Zugkraft F_3 belastet. Der Knoten 1 sei starr gehalten. Die System-Matrix wird **ohne Berücksichtigung der Randlagerung** definiert; anschließend erfolgt durch Hinzufügen der 4. Zeile die Berücksichtigung der Randbedingung $u_1 = 0$, was die Einführung eines Korrekturfaktors λ - im symmetrischen System - bedeutet. Damit folgen die Lösungswerte zu:

$$\begin{aligned}
 \lambda &= F_3 && \text{[N]} \\
 u_3 &= (1 + k_2/k_1) * F_3/k_2 && \text{[mm]} \\
 u_2 &= F_3/k_1 && \text{[mm]} \\
 u_1 &= 0. && \text{[mm]}
 \end{aligned}$$

Am noch freien System führt die zusätzliche Bedingung $u_1 = 0$ zu dem Korrekturfaktor $\lambda = F_3$, also der Auflagerkraft aus der eingeleiteten Kraft F_3 .

Der Korrekturfaktor λ ist in diesem Fall direkt interpretierbar. Zur Überprüfung der positiven Definitheit erhält man mit

$$\mathbf{u}^T = F_3 \begin{bmatrix} 0 & 1/k_1 & 1/k_1 (1 + k_2/k_1) & 1 \end{bmatrix} \quad \text{und} \quad \mathbf{u} = F_3 \begin{bmatrix} 0 \\ 1/k_1 \\ 1/k_1 (1 + k_2/k_1) \\ 1 \end{bmatrix}$$

und bei konstanter Federsteifigkeit $k = k_1 = k_2$

$$Q = \mathbf{u}^T * \mathbf{G} * \mathbf{u} = \frac{2(F_3)^2}{k} \geq 0 \quad \text{und die Determinante} \quad \text{Det} = -k^2$$

1.1.3 Beispiel 1.3 Zugstab mit 4 Federn

Ein Zugstab wird mit 4 Federn untersucht.

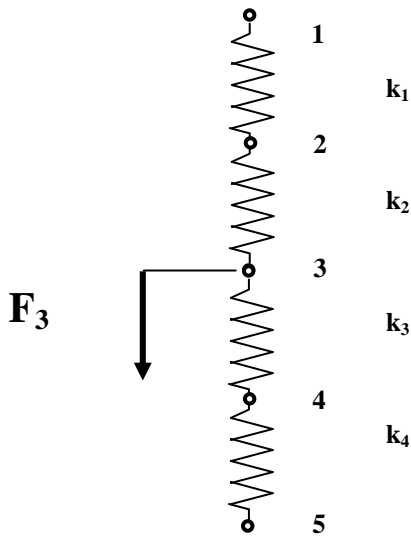


Bild A-1-3: Zugstab mit 4 Federn

Das Matrix-System sei definiert: $\mathbf{G}^* \mathbf{u} = \mathbf{P}$ bzw.

u_1	u_2	u_3	u_4	u_5	λ_1	λ_2	=	Absolutwert
k_1	$-k_1$	0	0	0	1	0	=	0
$-k_1$	k_1+k_2	$-k_2$	0	0	0	0	=	0
0	$-k_2$	k_2+k_3	$-k_3$	0	0	0	=	F_3
0	0	$-k_3$	k_3+k_4	$-k_4$	0	0	=	0
0	0	0	$-k_4$	$-k_4$	0	1	=	0
1	0	0	0	0	0	0	=	0
0	0	0	0	1	0	0	=	0

Der Zugstab als 4 nacheinander geschaltete Federn wird am Knoten 3 mit der Zugkraft F_3 belastet. Die Knoten 1 und 5 seien starr gehalten. Die System-Matrix wird **ohne Berücksichtigung der Randlagerung** definiert; anschließend erfolgt durch Hinzufügen der 6. und 7. Zeile die Berücksichtigung der Randbedingungen $u_1 = 0$ und $u_5 = 0$, was die Einführung zweier Korrekturfaktoren λ_1 und λ_2 - im symmetrischen System - bedeutet. Mit den Abkürzungen

$$a = \frac{\frac{1}{k_3} + \frac{1}{k_4}}{\frac{1}{k_1} + \frac{1}{k_2} + \frac{1}{k_3} + \frac{1}{k_4}} \quad \text{und} \quad 1 - a = \frac{\frac{1}{k_1} + \frac{1}{k_2}}{\frac{1}{k_1} + \frac{1}{k_2} + \frac{1}{k_3} + \frac{1}{k_4}}$$

und

$$b_1 = 0 \quad b_2 = \frac{\frac{k_2}{k_3} (1 + \frac{k_2}{k_3}) (1 - a)}{k_2} \quad b_3 = \frac{(1 + \frac{k_3}{k_4}) (1 - a)}{k_3} \quad b_4 = \frac{(1 - a)}{k_4} \quad b_5 = 0$$

folgen die Lösungswerte zu:

$$\begin{array}{rcl}
 \lambda_1 & = & a * F_3 \quad [\text{N}] \\
 \lambda_2 & = & (1-a) * F_3 \quad [\text{N}] \\
 u_5 & = & 0. \quad [\text{mm}] \\
 u_4 & = & b_4 * F_3 \quad [\text{mm}] \\
 u_3 & = & b_3 * F_3 \quad [\text{mm}] \\
 u_2 & = & b_2 * F_3 \quad [\text{mm}] \\
 u_1 & = & 0. \quad [\text{mm}]
 \end{array}$$

Am noch freien System führen die zusätzlichen Bedingungen $u_1 = 0$ sowie $u_5 = 0$ zu den Korrekturfaktoren $\lambda_1 = a * F_3$ bzw. $\lambda_2 = (1-a) * F_3$, also den Auflagerkräften aus der eingeleiteten Kraft F_3 .

Die Korrekturfaktoren λ_1 und λ_2 sind in diesem Fall direkt interpretierbar.

Zur Überprüfung der positiven Definitheit erhält man mit

$$\mathbf{u}^T = F_3 [0 \quad b_2 \quad b_3 \quad b_4 \quad 0 \quad a \quad (1-a)] \quad \text{und} \quad \mathbf{u} = F_3 \begin{bmatrix} 0 \\ b_2 \\ b_3 \\ b_4 \\ 0 \\ a \\ (1-a) \end{bmatrix}$$

und bei konstanter Federsteifigkeit $k = k_1 = k_2 = k_3 = k_4$

$$Q = \mathbf{u}^T * \mathbf{G} * \mathbf{u} = \frac{F_3^2}{k} \geq 0 \quad \text{und die Determinante} \quad \text{Det} = 4 k^3 .$$

1.2 Beispiele aus der Geometrie

1.2.1 Beispiel 2.1 Gerade mit 1 Zwangspunkt

Eine Gerade mit 1 Zwangspunkt wird untersucht.

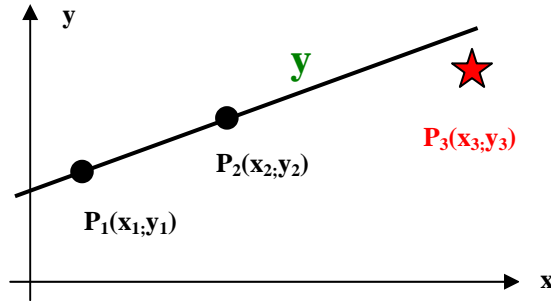


Bild A-1-4: Gerade mit 1 Zwangspunkt

Das Matrix-System sei definiert: $\mathbf{A} * \mathbf{x} = \mathbf{y}$ bzw.

a_0	a_1	λ	=	Absolutwert
1	x_1	1	=	y_1
1	x_2	x_3	=	y_2
1	x_3	0	=	y_3

Die Gerade $y = a_0 + a_1 x$ wird durch die Punkte $P_1(x_1; y_1)$ und $P_2(x_2; y_2)$ gebildet; anschließend erfolgt durch Hinzufügen der 3. Zeile die Definition eines **Zwangspunktes $P_3(x_3; y_3)$ für die Gerade $y(x)$** , was die Einführung eines Korrekturfaktors λ – im symmetrischen System – bedeutet. Damit folgen die Lösungswerte mit dem geometrischen Faktor

$$g(x_3) = \frac{1}{\frac{1-x_3}{x_1-x_2} - \frac{1}{x_1-x_3}} \quad \text{zu:}$$

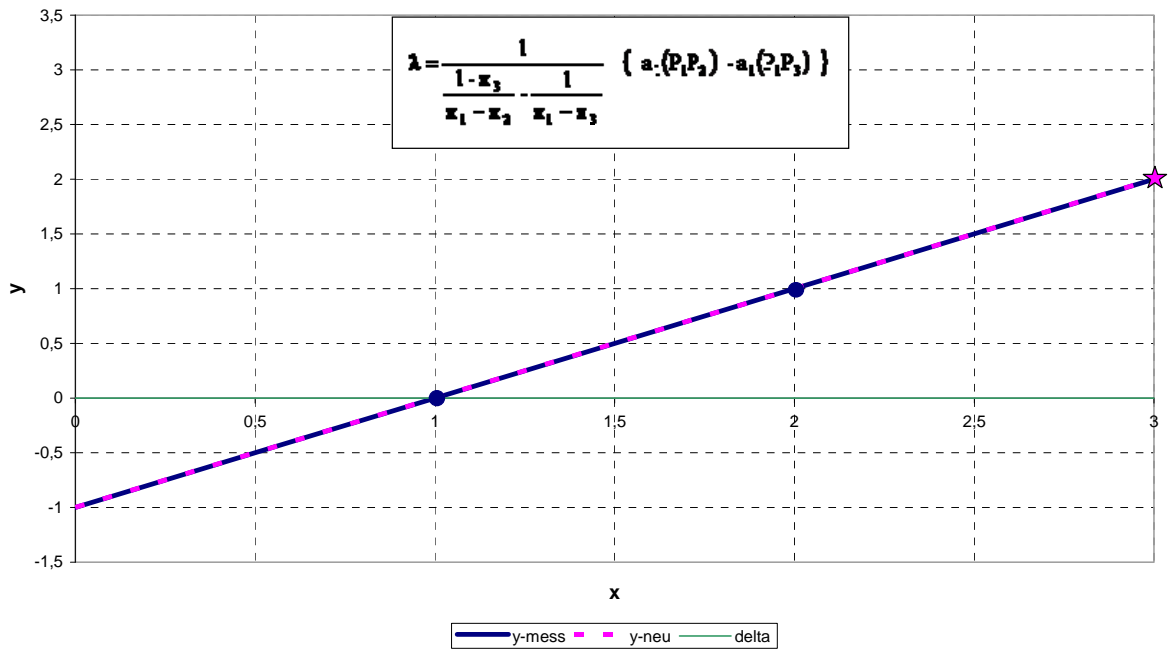
$$\lambda = g(x_3) \{ a_1(P_1P_2) - a_1(P_1P_3) \}$$

$$a_1 = \frac{y_1 - y_2}{x_1 - x_2} - \frac{1 - x_3}{x_1 - x_2} \lambda$$

$$a_0 = y_1 - \lambda - x_1 a_1$$

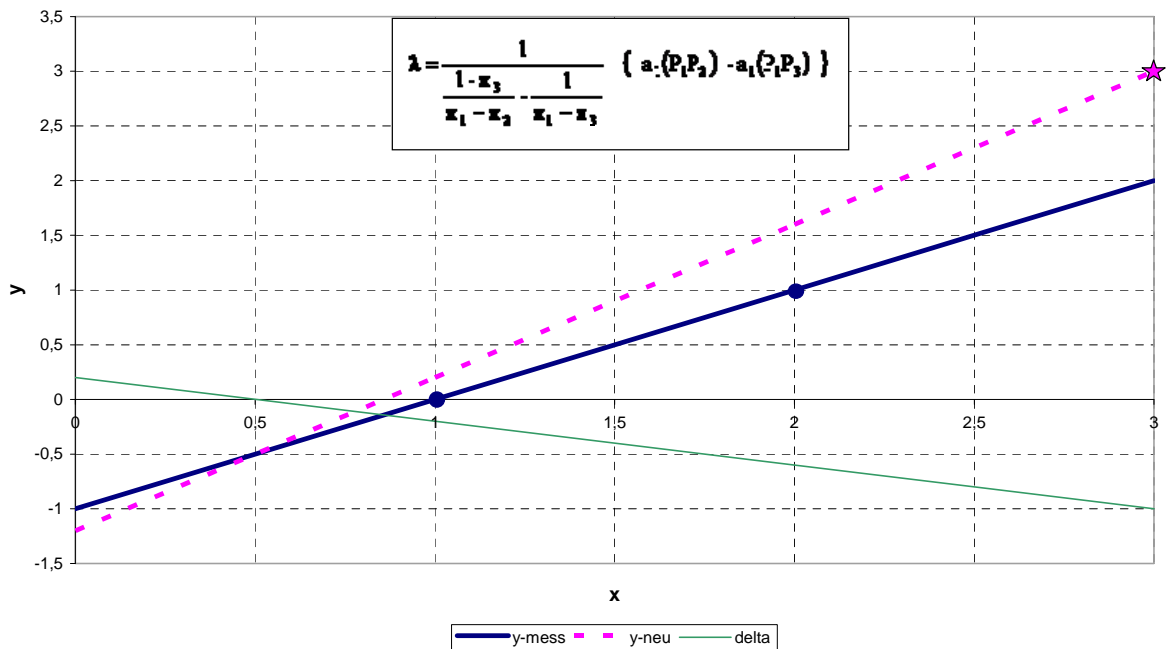
Die zusätzliche Bedingung des Zwangspunktes P_3 für die vordefinierte Gerade führt zu dem Korrekturfaktor λ , der hier zunächst einen allgemeinen Ausdruck ergibt, jedoch nach Umformung die Differenz der Steigungen zwischen den Punkten P_1 und P_2 sowie dem Punkt P_1 mit dem Zwangspunkt P_3 in Korrelation mit einem geometrischen Faktor bildet. Liegt der Zwangspunkt P_3 auf der vordefinierten Geraden, so ist keine Korrektur erforderlich, also ist $\lambda = 0$, wogegen die Korrektur $\lambda \neq 0$ wird, falls der Zwangspunkt P_3 nicht auf der vordefinierten Geraden liegt.

Beispiel 2.1:



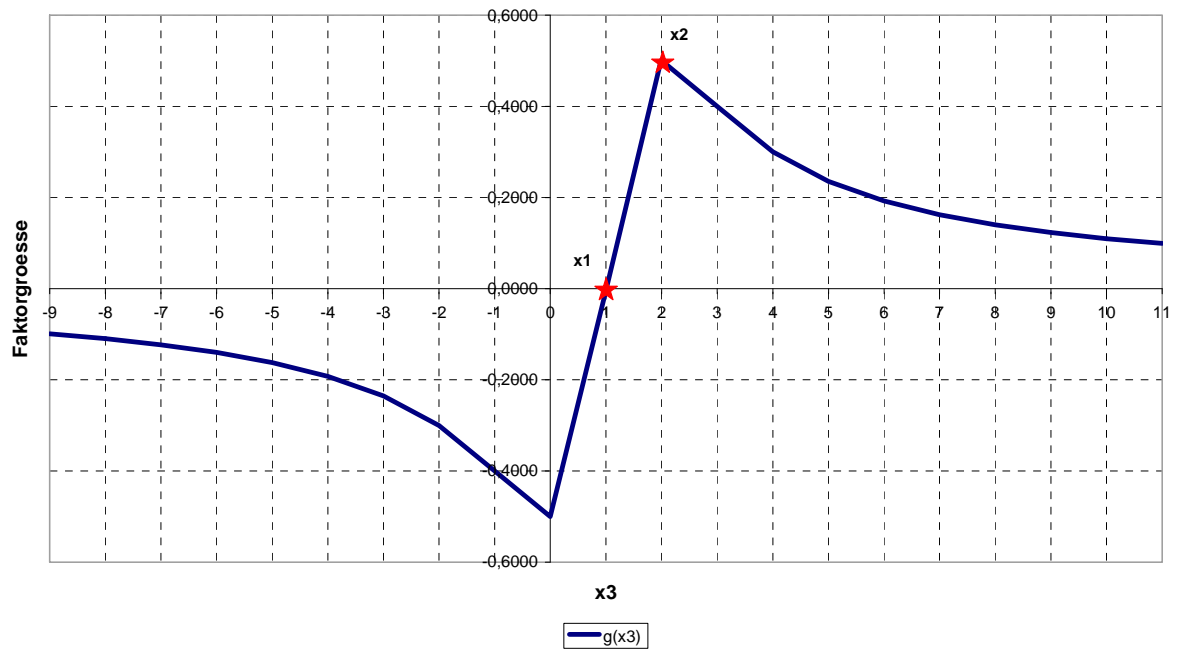
Zwangspunkt ★ auf der vordefinierten Geraden ● - ●
 → λ = 0

Beispiel 2.1:



Zwangspunkt ★ außerhalb der vordefinierten Geraden ● - ●
 → λ = -0.2

geometrischer Faktor $g(x_3)$ bei festen x_1 und x_2



1.2.2 Beispiel 2.2 Gerade mit 1 Zwangssteigung

Eine Gerade mit 1 Zwangssteigung wird untersucht.

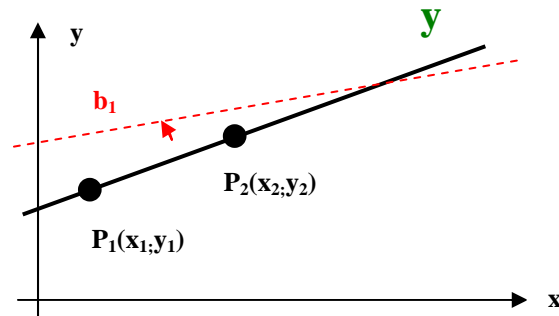


Bild A-1-5: Gerade mit 1 Zwangssteigung

Das Matrix-System sei definiert: $\mathbf{A} * \mathbf{x} = \mathbf{y}$ bzw.

a_0	a_1	λ	=	Absolutwert
1	x_1	0	=	y_1
1	x_2	1	=	y_2
0	1	0	=	b_1

Die Gerade $y = a_0 + a_1 x$ wird durch die Punkte $P_1(x_1; y_1)$ und $P_2(x_2; y_2)$ gebildet; anschließend erfolgt durch Hinzufügen der 3. Zeile die Definition einer **Zwangssteigung b_1 für die Gerade $y(x)$** , was die Einführung eines Korrekturfaktors λ - im symmetrischen System - bedeutet. Damit folgen die Lösungswerte zu:

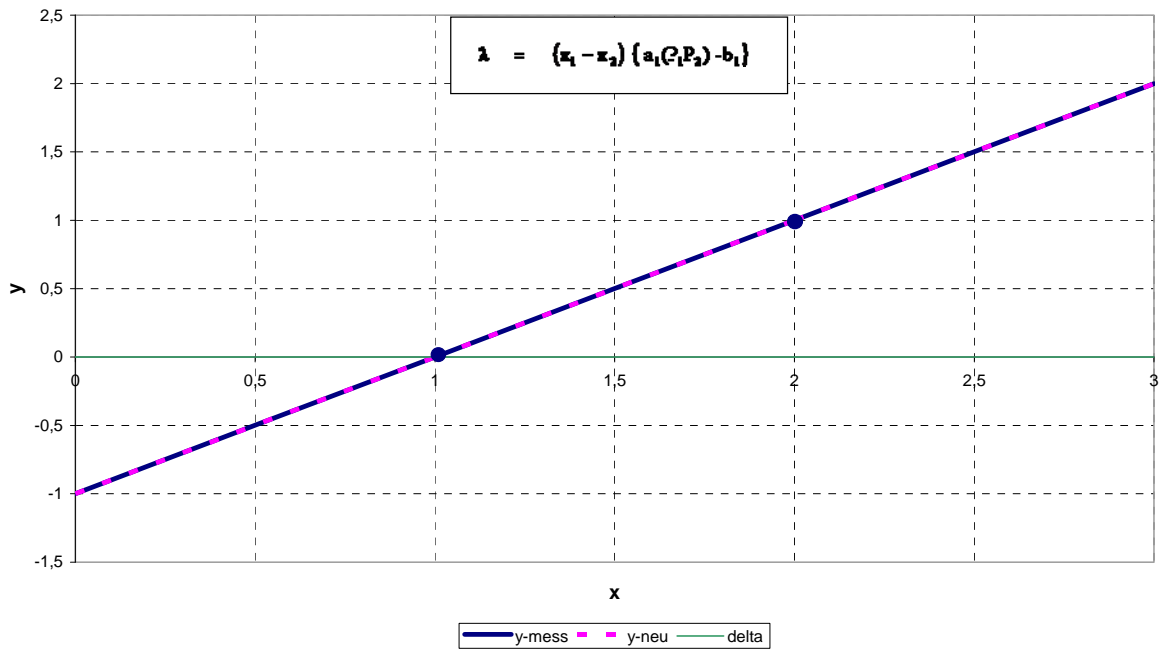
$$\lambda = (x_1 - x_2) \{ a_1(P_1 P_2) - b_1 \} \quad \text{mit ursprünglich} \quad a_1(P_1 P_2) = \frac{y_1 - y_2}{x_1 - x_2}$$

$$a_1 = \frac{y_1 - y_2}{x_1 - x_2} - \frac{1}{x_1 - x_2} \lambda$$

$$a_0 = y_1 - a_1 x_1$$

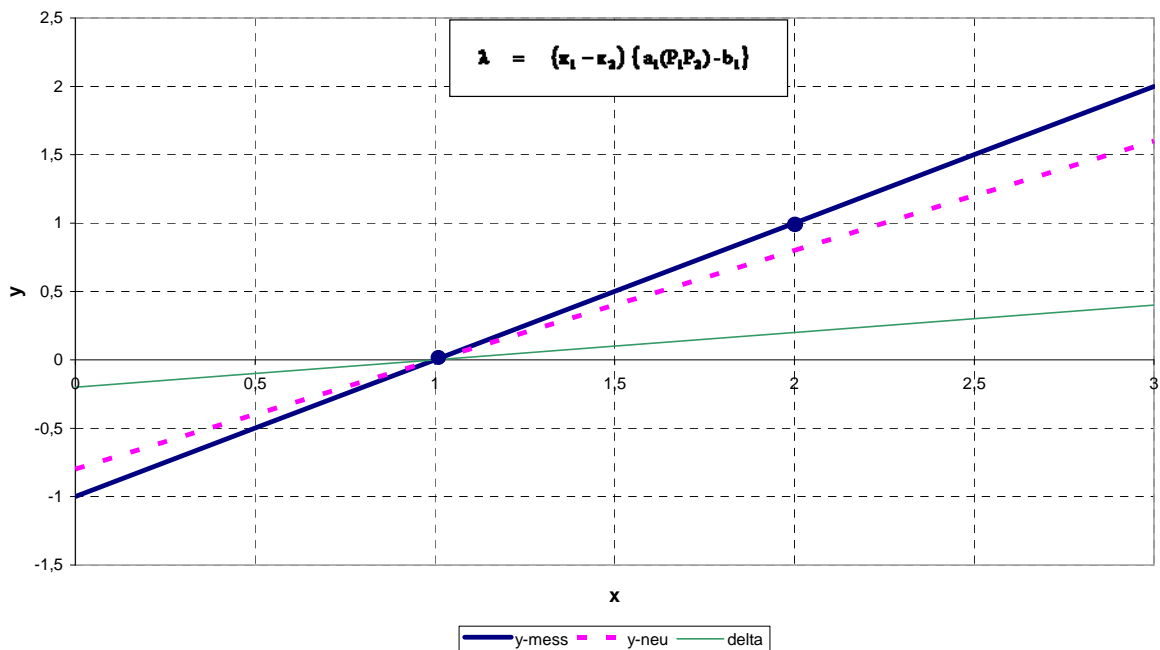
Die zusätzliche Bedingung der Zwangssteigung b_1 für die vordefinierte Gerade führt zu dem Korrekturfaktor λ , der hier als Differenz der alten zur vorgegebenen Steigung in Bezug auf das Längenintervall interpretiert werden kann. Sind beide Steigungen identisch, so ist keine Korrektur erforderlich, also ist $\lambda = 0$, wogegen die Korrektur $\lambda \neq 0$ wird, falls die Steigungen verschieden sind.

Beispiel 2.2:



Zwangssteigung b_1 identisch zur Ausgangssteigung $\frac{y_1 - y_2}{x_1 - x_2}$
 $\rightarrow \lambda = 0$

Beispiel 2.2:



Zwangssteigung b_1 verschieden zur Ausgangssteigung $\frac{y_1 - y_2}{x_1 - x_2}$
 $\rightarrow \lambda = 0.2$

1.2.3 Beispiel 2.3 Gerade mit 2 Zwangspunkten

Eine Gerade mit 2 Zwangspunkten wird untersucht.

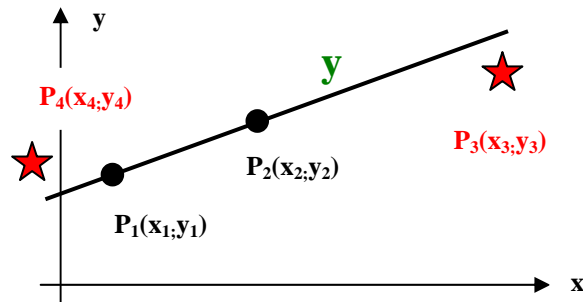


Bild A-1-5: Gerade mit 2 Zwangspunkten

Das Matrix-System sei definiert: $\mathbf{A} * \mathbf{x} = \mathbf{y}$ bzw.

\mathbf{a}_0	\mathbf{a}_1	λ_0	λ_1	=	Absolutwert
1	x_1	1	1	=	y_1
1	x_2	x_3	x_4	=	y_2
1	x_3	0	0	=	y_3
1	x_4	0	0	=	y_4

Die Gerade $y = a_0 + a_1 x$ wird durch die Punkte $P_1(x_1;y_1)$ und $P_2(x_2;y_2)$ gebildet; anschließend erfolgt durch Hinzufügen der 3. und 4. Zeile die Definition **2er Zwangspunkte $P_3(x_3;y_3)$ und $P_4(x_4;y_4)$ für die Gerade $y(x)$** , was die Einführung 2er Korrekturfaktoren λ_0 und λ_1 - im symmetrischen System - bedeutet.

Mit den Abkürzungen

$$\alpha_{12} = \frac{1-x_3}{x_1-x_2} \qquad \alpha_{13} = \frac{1-x_4}{x_1-x_2} \qquad \beta_1 = \frac{y_1-y_2}{x_1-x_2}$$

$$\alpha_{22} = \frac{1-x_3}{x_1-x_2} - \frac{1}{x_1-x_3} \qquad \alpha_{23} = \frac{1-x_4}{x_1-x_2} - \frac{1}{x_1-x_3} \qquad \beta_2 = \frac{y_1-y_2}{x_1-x_2} - \frac{y_1-y_3}{x_1-x_3}$$

$$\alpha_{32} = \frac{1-x_3}{x_1-x_2} - \frac{1}{x_1-x_4} \qquad \alpha_{33} = \frac{1-x_4}{x_1-x_2} - \frac{1}{x_1-x_4} \qquad \beta_3 = \frac{y_1-y_2}{x_1-x_2} - \frac{y_1-y_4}{x_1-x_4}$$

folgen die Lösungswerte zu:

$$\lambda_1 = \frac{\beta_2 / \alpha_{22} - \beta_3 / \alpha_{32}}{\alpha_{23} / \alpha_{22} - \alpha_{33} / \alpha_{32}}$$

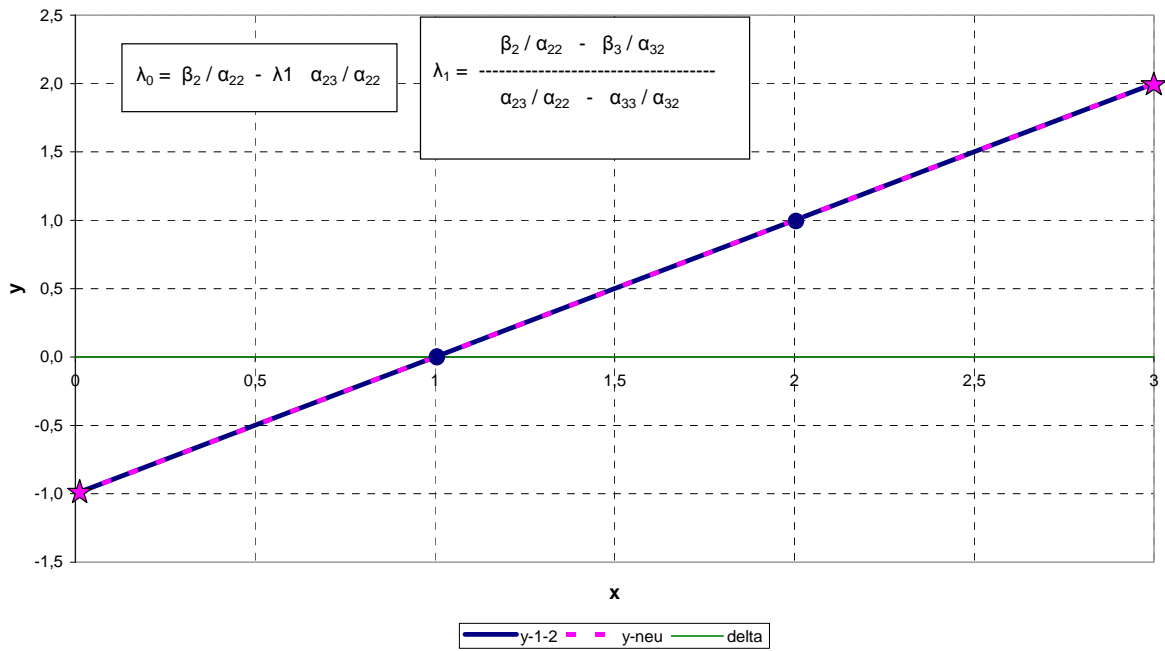
$$\lambda_0 = \beta_2 / \alpha_{22} - \lambda_1 \alpha_{23} / \alpha_{22}$$

$$a_1 = \beta_1 - \lambda_1 \alpha_{13} - \lambda_0 \alpha_{12}$$

$$a_0 = y_1 - \lambda_1 - \lambda_0 - a_1 x_1$$

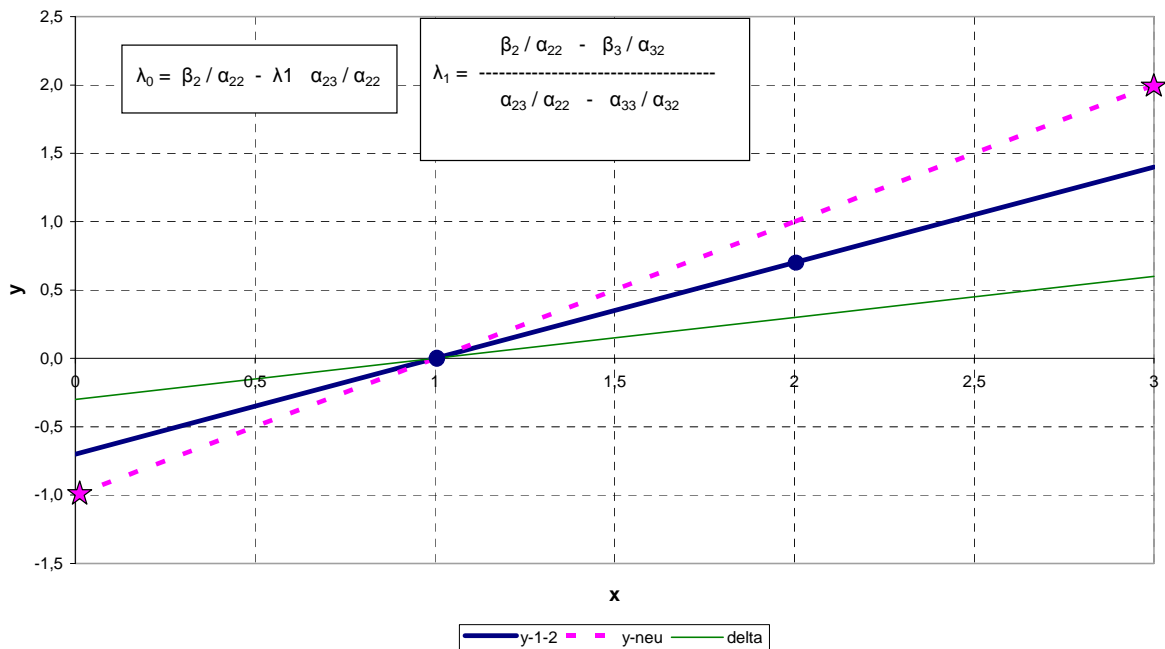
Die zusätzliche Bedingung der Zwangspunkte \mathbf{P}_3 und \mathbf{P}_4 für die vordefinierte Gerade führen zu den Korrekturfaktoren λ_0 und λ_1 , die hier allgemeine Beziehungen darstellen. Liegen die Zwangspunkte \mathbf{P}_3 und \mathbf{P}_4 auf der vordefinierten Geraden, so ist keine Korrektur erforderlich, also ist $\lambda = 0$, wogegen die Korrekturen λ_0 und $\lambda_1 \neq 0$ werden, falls die Zwangspunkte \mathbf{P}_3 und \mathbf{P}_4 nicht auf der vordefinierten Geraden liegen.

Beispiel 2.3:



2 Zwangspunkte ★ auf der vordefinierten Geraden ● - ●
 → $\lambda_1 = -0$ und $\lambda_0 = 0$

Beispiel 2.3:



2 Zwangspunkte ★ außerhalb der vordefinierten Geraden ● - ●
 → $\lambda_1 = -0.1$ und $\lambda_0 = 0.1$

1.3 Beispiele aus der Ausgleichsrechnung

1.3.1 Beispiel 3.1 Symmetrisches System mit 11 Messpunkten

Ein symmetrisches System mit $N = 11$ Messpunkten ist gegeben, für das die Ausgleichsrechnung mittels Bernoulli-Ansatz mit 4 Konstanten durchzuführen ist. Die hier gewählte Ansatzfunktion entspricht der Balkenbiegeline unter Einzellast.

Symmetrisches Matrix-System: $\mathbf{S} * \mathbf{c} = \mathbf{T}$ bzw.

\mathbf{c}_1	\mathbf{c}_2	\mathbf{c}_3	\mathbf{c}_4	λ_1	λ_2	=	Absolutwert
b_{11}	b_{12}	b_{13}	b_{14}	1	1	=	a_1
b_{21}	b_{22}	b_{23}	b_{24}	0	1	=	a_2
b_{31}	b_{32}	b_{33}	b_{34}	0	1	=	a_3
b_{41}	b_{42}	b_{43}	b_{44}	0	1	=	a_4
1	0	0	0	0	0	=	w^R_1
1	1	1	1	0	0	=	w^R_2

Es gelten die Abkürzungen mit $b_{ik} = b_{ki}$:

$$\begin{aligned} \frac{b_{11}}{2} &= \sum_i^{1,N} 1 * 1 & \frac{b_{12}}{2} &= \sum_i^{1,N} 1 * \xi_i & \frac{b_{13}}{2} &= \sum_i^{1,N} 1 * \xi_i^2 & \frac{b_{14}}{2} &= \sum_i^{1,N} 1 * \xi_i^3 \\ & & \frac{b_{22}}{2} &= \sum_i^{1,N} \xi_i * \xi_i & \frac{b_{23}}{2} &= \sum_i^{1,N} \xi_i * \xi_i^2 & \frac{b_{24}}{2} &= \sum_i^{1,N} \xi_i * \xi_i^3 \\ & & & & \frac{b_{33}}{2} &= \sum_i^{1,N} \xi_i^2 * \xi_i^2 & \frac{b_{34}}{2} &= \sum_i^{1,N} \xi_i^2 * \xi_i^3 \\ & & & & & & \frac{b_{44}}{2} &= \sum_i^{1,N} \xi_i^3 * \xi_i^3 \end{aligned}$$

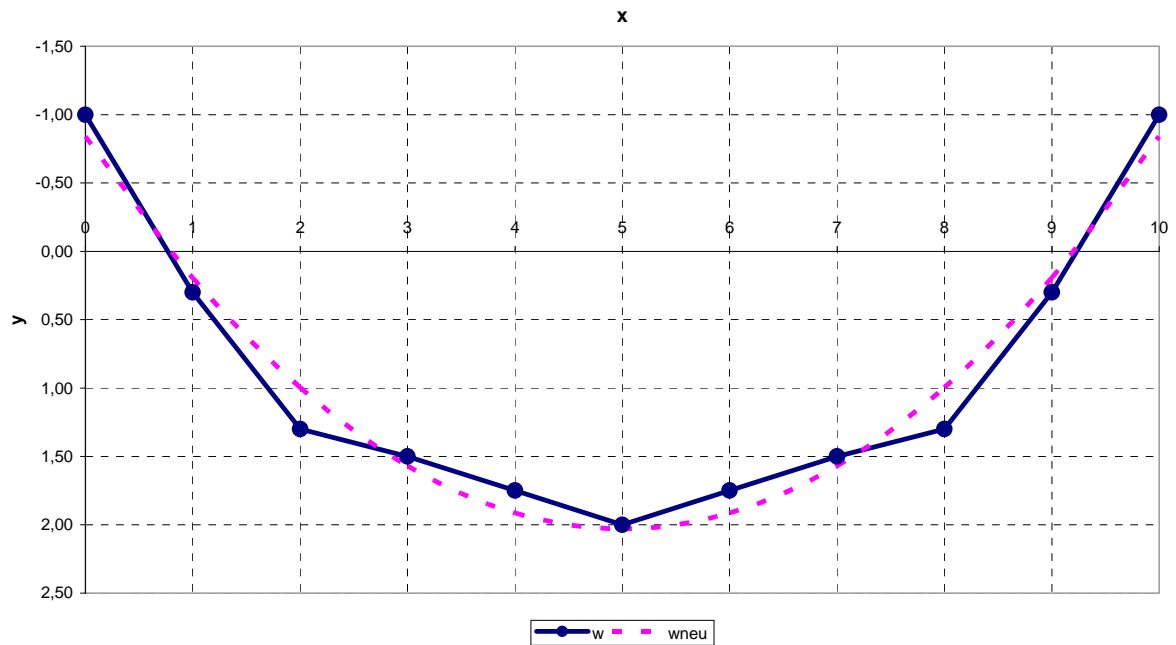
$$\frac{a_1}{2} = \sum_i^{1,N} 1 * w_i \quad \frac{a_2}{2} = \sum_i^{1,N} \xi_i * w_i \quad \frac{a_3}{2} = \sum_i^{1,N} \xi_i^2 * w_i \quad \frac{a_4}{2} = \sum_i^{1,N} \xi_i^3 * w_i$$

Die ξ_j -Werte sind die bezogenen Längen-Koordinaten der einzelnen Messwerte w_i .

Die Werte w^R_1 und w^R_2 sind definierte Zwangswerte an den Rändern.

Eine allgemeine Lösung für die λ -Werte ist mit großem Aufwand verbunden und auch nicht zu aussagefähig, da sie deutlich auch von der Zahl der Messpunkte N abhängt.

Beispiel 3.1:

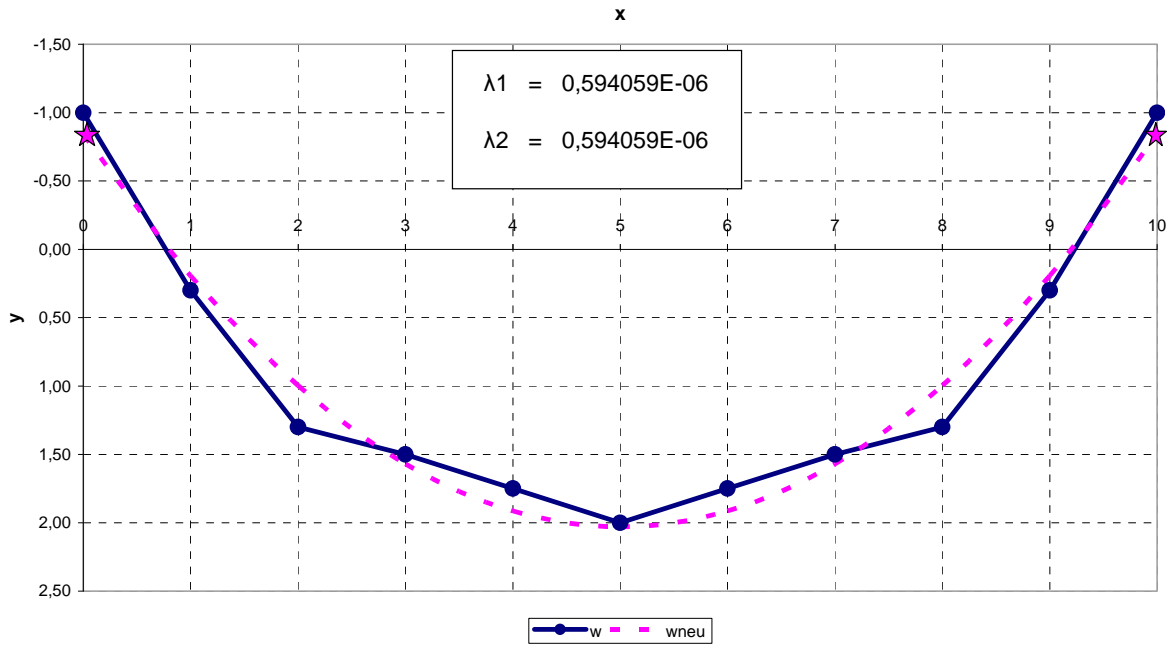


vordefinierte Punkte ● ohne Zwangspunkte
kein λ

Im ersten Schritt sind keine Zwangsbedingungen, sondern der reine Ausgleich durchgeführt worden; das Ergebnis zeigt an beiden Rändern dann die entsprechenden Randwerte.

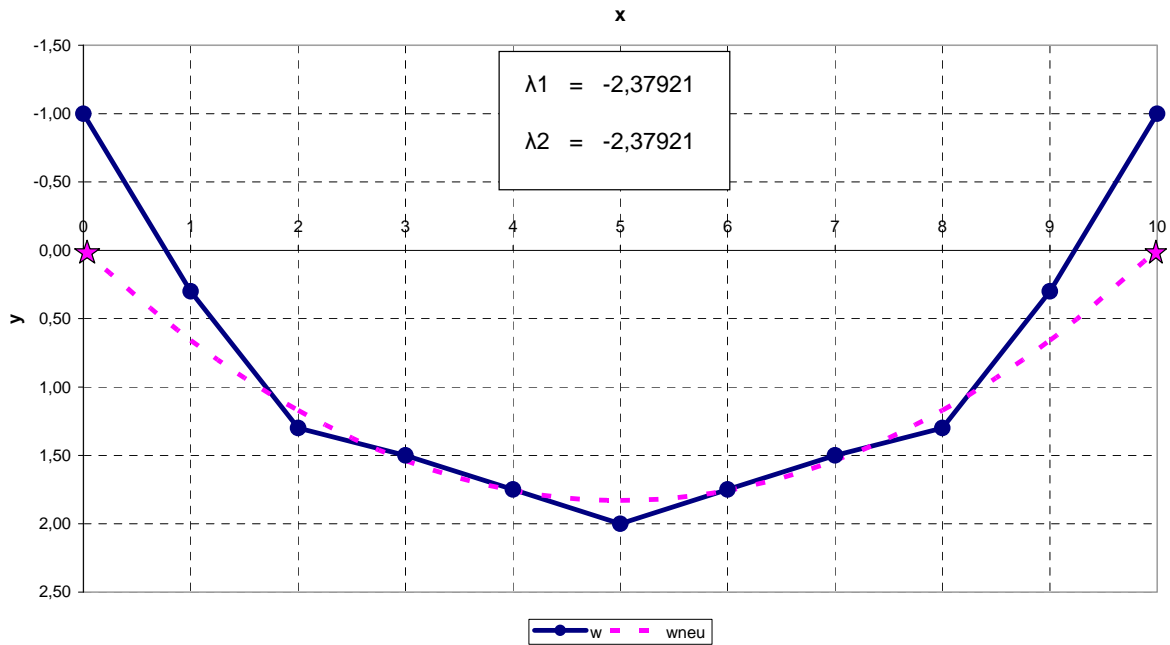
Im nächsten Schritt werden diese Randwerte als Zwangswerte eingeführt und ergeben sehr kleine λ -Werte, wogegen danach beliebige Zwangswerte gewählt werden, die zu entsprechend deutlichen λ -Werten führen.

Beispiel 3.1:



vordefinierte Punkte ● mit 2 „korrekten“ Zwangspunkten ★
 → $\lambda_1 = \lambda_2 = 0,594059E-06$

Beispiel 3.1:



vordefinierte Punkte ● mit 2 „willkürlichen“ Zwangspunkten ★
 → $\lambda_1 = \lambda_2 = -2,37921$

1.3.2 Beispiel 3.2 Schief-symmetrisches System mit 11 Messpunkten

Ein schief-symmetrisches System mit $N = 11$ Messpunkten ist gegeben, für das die Ausgleichsrechnung mittels Bernoulli-Ansatz mit 4 Konstanten durchzuführen ist. Die gewählte Ansatzfunktion entspricht der Balkenbiegeline unter Einzellast.

Schief-symmetrisches Matrix-System: $\mathbf{S} * \mathbf{c} = \mathbf{T}$ bzw.

\mathbf{c}_1	\mathbf{c}_2	\mathbf{c}_3	\mathbf{c}_4	λ_1	λ_2	=	Absolutwert
b_{11}	b_{12}	b_{13}	b_{14}	1	1	=	a_1
b_{21}	b_{22}	b_{23}	b_{24}	0	1	=	a_2
b_{31}	b_{32}	b_{33}	b_{34}	0	1	=	a_3
b_{41}	b_{42}	b_{43}	b_{44}	0	1	=	a_4
1	0	0	0	0	0	=	w^R_1
1	1	1	1	0	0	=	w^R_2

Es gelten die Abkürzungen mit $b_{ik} = b_{ki}$:

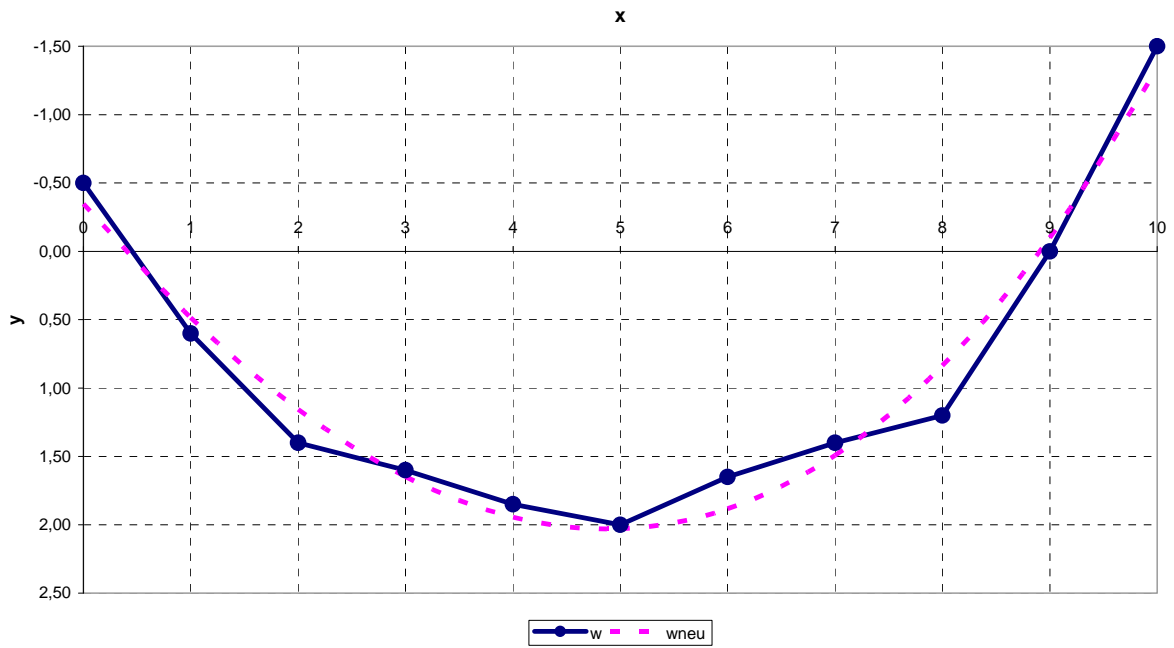
$$\begin{aligned} \frac{b_{11}}{2} &= \sum_i^{1,N} 1 * 1 & \frac{b_{12}}{2} &= \sum_i^{1,N} 1 * \xi_i & \frac{b_{13}}{2} &= \sum_i^{1,N} 1 * \xi_i^2 & \frac{b_{14}}{2} &= \sum_i^{1,N} 1 * \xi_i^3 \\ \frac{b_{22}}{2} &= \sum_i^{1,N} \xi_i * \xi_i & \frac{b_{23}}{2} &= \sum_i^{1,N} \xi_i * \xi_i^2 & \frac{b_{24}}{2} &= \sum_i^{1,N} \xi_i * \xi_i^3 \\ \frac{b_{33}}{2} &= \sum_i^{1,N} \xi_i^2 * \xi_i^2 & \frac{b_{34}}{2} &= \sum_i^{1,N} \xi_i^2 * \xi_i^3 \\ \frac{b_{44}}{2} &= \sum_i^{1,N} \xi_i^3 * \xi_i^3 \\ \frac{a_1}{2} &= \sum_i^{1,N} 1 * w_i & \frac{a_2}{2} &= \sum_i^{1,N} \xi_i * w_i & \frac{a_3}{2} &= \sum_i^{1,N} \xi_i^2 * w_i & \frac{a_4}{2} &= \sum_i^{1,N} \xi_i^3 * w_i \end{aligned}$$

Die ξ_i -Werte sind die bezogenen Längen-Koordinaten der einzelnen Messwerte w_i .

Die Werte w^R_1 und w^R_2 sind definierte Zwangswerte an den Rändern.

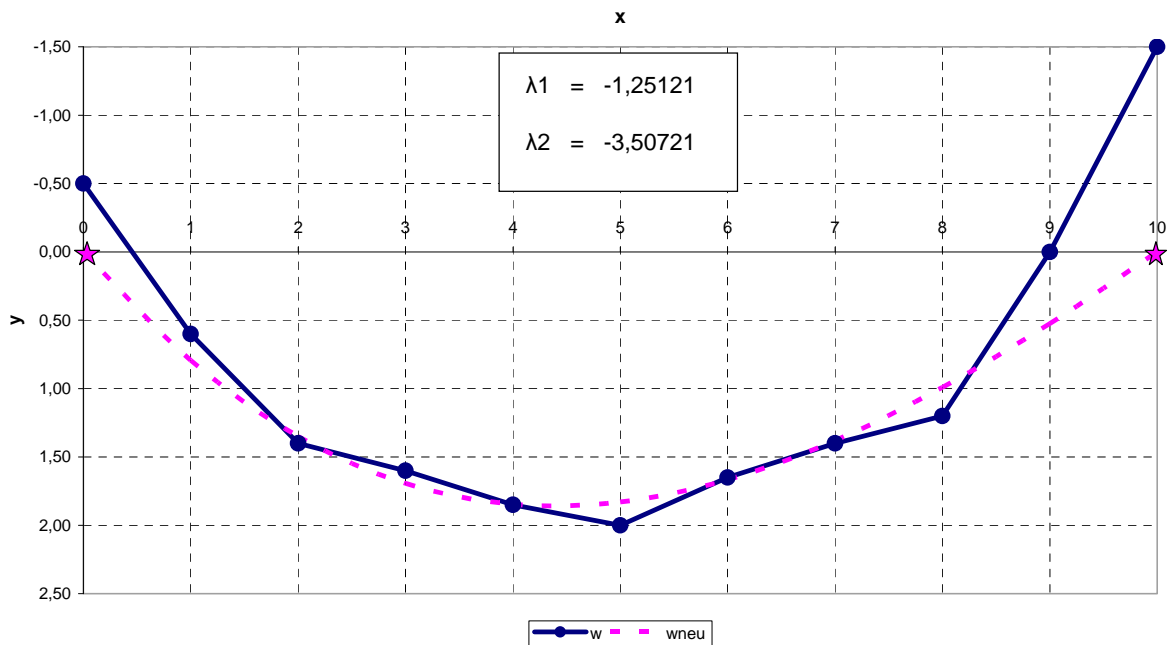
Eine allgemeine Lösung für die λ -Werte ist mit großem Aufwand verbunden und auch nicht zu aussagefähig, da sie deutlich auch von der Zahl der Messpunkte N abhängt.---Der Vergleich mit Beispiel 3.1 ergibt die gleiche Summe der Absolutbeträge für $|\lambda_1| + |\lambda_2|$.Die größere Korrektur ist hier auf der rechten Seite - also bei λ_2 - vorzunehmen, was auch die Rechnung ausweist.

Beispiel 3.2:



vordefinierte Punkte ● ohne Zwangspunkte
kein λ

Beispiel 3.2:



vordefinierte Punkte ● mit 2 „willkürlichen“ Zwangspunkten ★
 $\rightarrow \lambda_1 = -1,25121$ und $\lambda_2 = -3,50721$

2 Beispiele der statischen Anwendung

2.1 Matrix-System bei Gesamtschritt- und Zwischenschritt-Lösung

Zunächst erfolgt eine Gegenüberstellung beider Lösungswege, indem die Gesamtmatrizen verglichen werden; hierzu werden in Tab. A-2-1 die symbolischen Bezeichnungen zusammengefasst, so dass die einzelnen Abschnitte unterschieden werden können.

	Erläuterung
	Matrix- und Messwerte aus Durchbiegung w
	Matrix- und Messwerte aus Winkelneigung w'
	Matrix- und Messwerte aus Dehnung ϵ bzw. Krümmung w''
	nur Matrixwerte der inneren Kopplung der Nachbarfelder $i, i+1$ [w und/oder w' und/oder w'']
	nur Matrixwerte der äußeren Kopplung der Einzelfelderparameter [c_1 c_2 c_3 c_4 c_5]
	Matrix- und Systemwerte aus definierten Randbedingungen
	Die Zahlen innerhalb der farbigen Felder geben die Feld-Nummern an, die miteinander korreliert werden müssen.


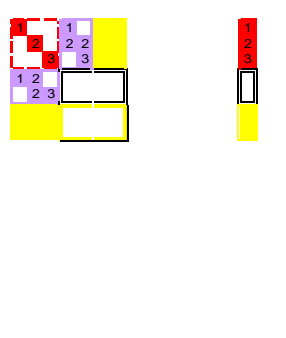
Tab. A-2-1: Erläuterung der Matrix-Bezeichnungen

In den einzelnen Beispielen werden die bekannten „Messwerte für die einzelnen Felder“ angegeben. Die unterschiedliche Qualität [Durchbiegungen w und/oder Winkelneigungen w' und/oder Krümmungen w'' bzw. Dehnungen ϵ] bedingen jeweils eine unterschiedliche Kopplung der Sub-Matrizen. Die Zwischenschritt-Lösung, die eine Näherung darstellt, führt immer zu kleinerer Zahl der DOF gegenüber der Gesamtschritt-Lösung.

Da die Zahl der Randbedingungen unterschiedlich sein kann, ist dieser Bereich in der Darstellung nur schematisch vorgesehen.

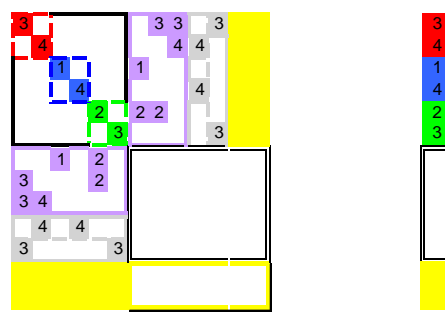
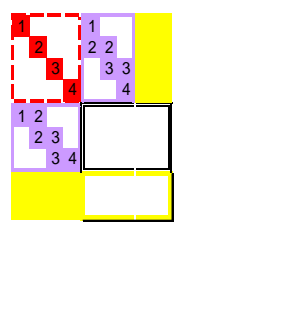
2.1.1 Beispiel 2-1.1 3 Felder

Messwerte für die Felder:			
Feld 1	W	W ⁱ	W ⁱⁱ
Feld 2	W	W ⁱ	W ⁱⁱ
Feld 3	W	W ⁱ	W ⁱⁱ

<p>Matrix-System bei Gesamtschritt-Lösung</p> <p style="text-align: center;">Lastseite</p> <p>Zahl der DOF = 88</p> 	<p>Matrix-System bei Zwischenschritt-Lösung</p> <p style="text-align: center;">Lastseite</p> <p>Zahl der DOF = 28</p> 
---	--

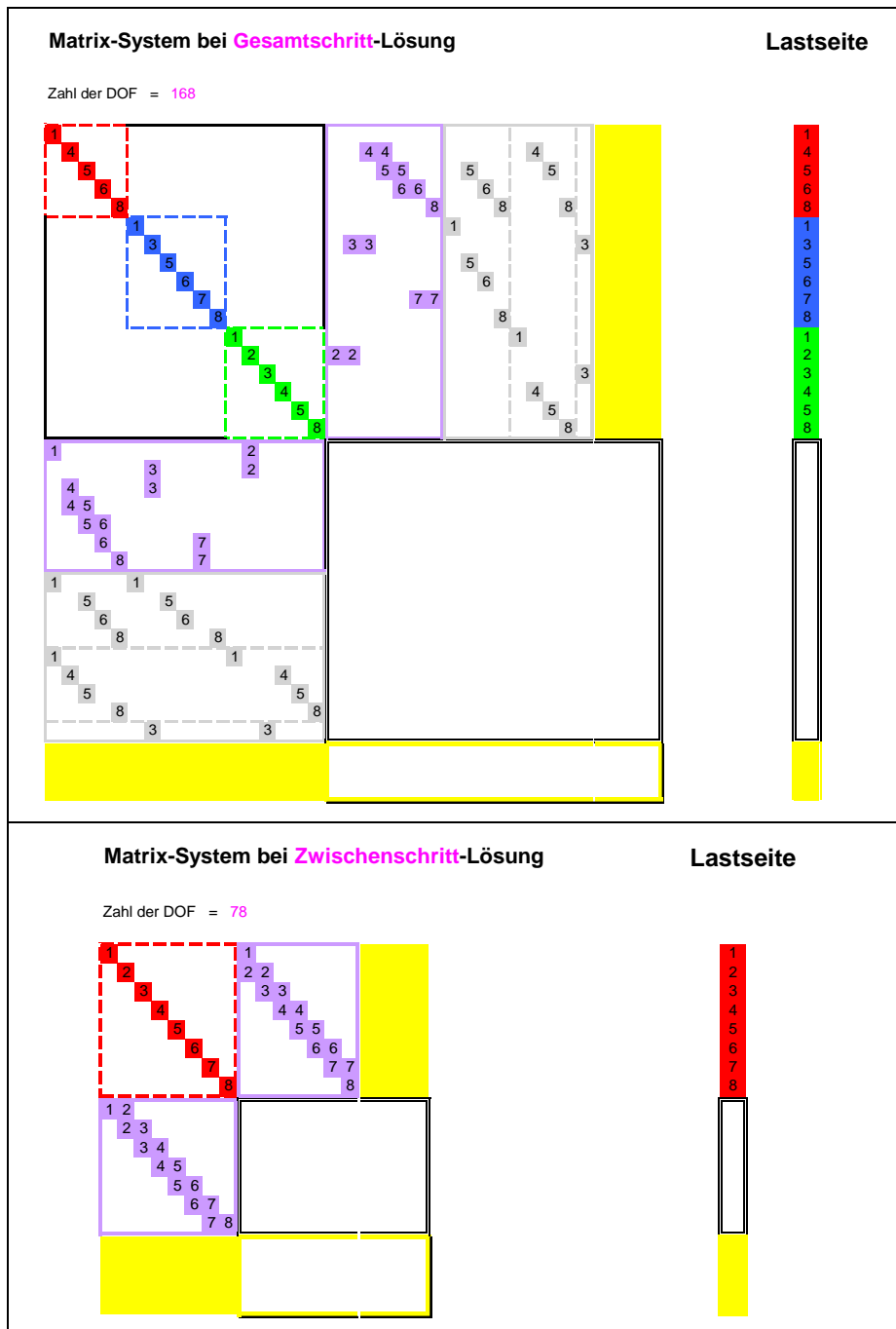
2.1.2 Beispiel 2-1.2 4 Felder

Messwerte für die Felder:			
Feld 1		W ⁱ	
Feld 2			W ⁱⁱ
Feld 3	W		W ⁱⁱ
Feld 4	W	W ⁱ	

<p>Matrix-System bei Gesamtschritt-Lösung</p> <p style="text-align: center;">Lastseite</p> <p>Zahl der DOF = 58</p> 	<p>Matrix-System bei Zwischenschritt-Lösung</p> <p style="text-align: center;">Lastseite</p> <p>Zahl der DOF = 38</p> 
---	--

2.1.3 Beispiel 2-1.3 8 Felder

Messwerte für die Felder:			
Feld 1	W	W'	W''
Feld 2			W''
Feld 3		W'	W''
Feld 4	W		W''
Feld 5	W	W'	W''
Feld 6	W	W'	
Feld 7		W'	
Feld 8	W	W'	W''



2.2 Spezifische Anwendungen

Im Folgenden werden ausführlich einige Beispiele diskutiert. Es wird jeweils eine zwangungsfreie Biegung unterstellt, da Biegung mit Zwang letztlich nur zur genaueren Messung über die Querschnittshöhe und -länge führt. Bei den Varianten werden nicht nur die unterschiedlichen Lösungsarten - Gesamtschritt- bzw. Zwischenschritt-Lösung -, sondern auch das entsprechende Verhalten in Abhängigkeit der Zahl und Qualität der Messpunkte und der hinreichenden Zahl der Randbedingungen untersucht. Zusätzlich wird im Regressionsprogramm auch die Stabilität der Gesamtmatrix als Matrizenprodukt und der Determinantenwert angegeben.

2.2.1 Beispiel 2-2.1 3Feld-Balken mit Gleichlast q

Das in Bild A-2-1 dargestellte System ist entsprechend Tab. A-2-2 mit dem ADINA-Code berechnet worden, und zwar für einen konstanten Rechteckquerschnitt mit einer Höhe $h = 1200$ mm und einer Breite $b = 200$ mm bei einem konstanten Elastizitätsmodul $E_b = 30000$ N/mm². Damit liegen aus der statischen Berechnung alle notwendigen „Messpunkte“ zur Verwendung in der Regressionsrechnung vor.

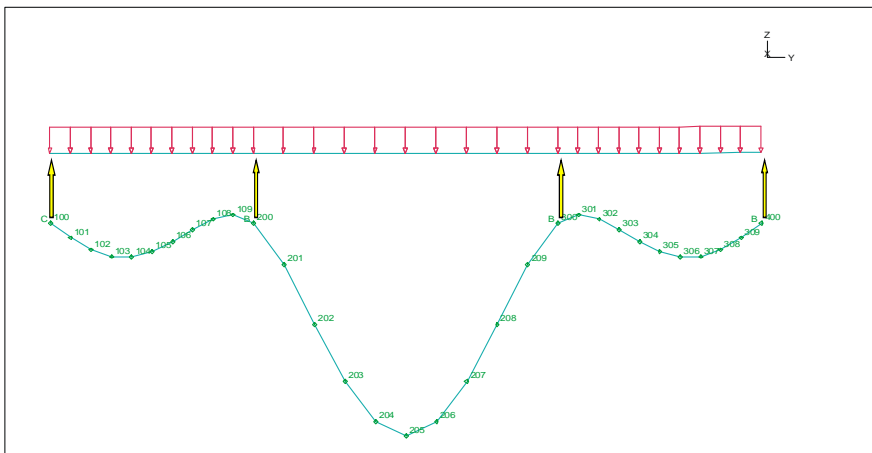


Bild A-2-1: 3Feld-Balken mit Gleichlast q

Feld	Stützweite [mm]	Gleichlast [N/mm]
1	8000	1
2	10000	1
3	12000	1

Tab. A-2-2: Systemwerte des Beispiels

Die Ergebnisse unterschiedlicher Vorgaben der „Messwerte“ sind in Tab. A-2-3 zusammengestellt:

Grundsätzlich werden beide Lösungsarten jeweils unter gleichen Voraussetzungen verglichen [s. Spalte 1 bzw. 2]. In Spalte 3 werden die Basis-Input-Werte zur Regression aufgelistet mit Zahl der Abschnitte [Ab], Grad des Polynoms [Po] und Funktion der Kopplung [Fu]. Danach folgen je Abschnitt Zahl der Durchbiegungen w , der Winkelneigungen w' und der Krümmungen w'' bzw. Dehnungen ϵ , gefolgt von den Randbedingungen. Die Längskoordinaten ξ werden in Spalte 4 aufgelistet, die Randbedingungen in Spalte 5.---Die Ergebnisse zur Matrixstabilität sind in Spalten 6 und 7 angegeben. Diese weichen je nach Art und Zahl der Messpunkte und/oder der Qualität der Randbedingungen stark voneinander ab; im Allgemeinen ist die Lösung stabil – solange keine ungenügenden Bedingungen zur Beschreibung des „Biegeverlaufes“ des Tragwerkes vorliegen. Mit 1 Ausnahme (der Näherungslösung) liegen die Ergebnisse eng beieinander.

Die Ergebnisse für alle Tragwerkszustände sind in den Bildern A-2-2a - c dargestellt.

GS Gesamtschritt-Lösung
 ZS Zwischenschritt-Lösung

1	2	3	4	5	6	7
Art	Comment	3F nur q Ab/Po/Fu /w/w'/e/ r=	Abschnittskordinaten ξ	Randbedingungen	Q = Bt * A * B	Determinante
ADI	ADINA-Rechnung					
1-V1	GS / Var. 1	3/4/3 /11-11-11/11-11-11/11-11-11/0-0-0	0.0-0.1-0.2-0.3-0.4-0.5-0.6-0.7-0.8-0.9-1.0	keine	7,599430E-02	7,639920E+00
2-V1	ZS / Var. 1	3/4/3 /11-11-11/11-11-11/11-11-11/0-0-0	0.0-0.1-0.2-0.3-0.4-0.5-0.6-0.7-0.8-0.9-1.0	keine	2,135320E-01	4,596510E+03
1-V2	GS / Var. 2	3/4/3 /5-5-5/5-5-5/5-5-5/0-0-0	0.0-0.5-0.8-0.9-1.0/0.0-0.2-0.5-0.8-1.0/0.0-0.1-0.2-0.5-1.0	keine	3,382630E-02	9,536570E-03
2-V2	ZS / Var. 2	3/4/3 /5-5-5/5-5-5/5-5-5/0-0-0	0.0-0.5-0.8-0.9-1.0/0.0-0.2-0.5-0.8-1.0/0.0-0.1-0.2-0.5-1.0	keine	6,964950E-02	1,403350E-01
1-V3	GS / Var. 3	3/4/3 /3-3-3/3-3-3/3-3-3/6-0-0	0.5-0.8-0.9/0.2-0.5-0.8/0.1-0.2-0.5	Nulldurchgang: Feld 1 $\xi=0.7625$ Feld 3 $\xi=0.2375$ alle Auflager	4,108590E-02	0,000000E+00
2-V3	ZS / Var. 3	3/4/3 /3-3-3/3-3-3/3-3-3/6-0-0	0.5-0.8-0.9/0.2-0.5-0.8/0.1-0.2-0.5	Nulldurchgang: Feld 1 $\xi=0.7625$ Feld 3 $\xi=0.2375$ alle Auflager	6,964820E-02	1,349820E-07
1-V4	GS / Var. 4	3/4/3 /3-3-3/3-3-3/3-3-3/4-0-0	0.5-0.8-0.9/0.2-0.5-0.8/0.1-0.2-0.5	alle Auflager	4,108630E-02	0,000000E+00
2-V4	ZS / Var. 4	3/4/3 /3-3-3/3-3-3/3-3-3/4-0-0	0.5-0.8-0.9/0.2-0.5-0.8/0.1-0.2-0.5	alle Auflager	6,964950E-02	9,233860E-06
1-V5	GS / Var. 5	3/4/3 /3-0-0/3-0-0/3-0-0/4-0-0	0.5-0.8-0.9/0.2-0.5-0.8/0.1-0.2-0.5	alle Auflager	2,321650E-02	4,074180E-08
2-V5	ZS / Var. 5	3/4/3 /3-0-0/3-0-0/3-0-0/4-0-0	0.5-0.8-0.9/0.2-0.5-0.8/0.1-0.2-0.5	alle Auflager	2,321650E-02	4,074180E-08
1-V6	GS / Var. 6	3/4/3 /3-0-3/3-0-3/0-0-3/4-0-0	0.5-0.8-0.9/0.2-0.5-0.8/0.1-0.2-0.5	alle Auflager	4,108630E-02	1,283790E-05
2-V6	ZS / Var. 6	3/4/3 /3-0-3/3-0-3/0-0-3/4-0-0	0.5-0.8-0.9/0.2-0.5-0.8/0.1-0.2-0.5	alle Auflager	4,643300E-02	1,335500E-06
1-V7	GS / Var. 7	3/4/3 /3-0-3/3-0-3/0-0-3/2-0-0	0.5-0.8-0.9/0.2-0.5-0.8/0.1-0.2-0.5	mittlere Auflager	4,130910E-02	4,070060E-14
2-V7	ZS / Var. 7	3/4/3 /3-0-3/3-0-3/0-0-3/2-0-0	0.5-0.8-0.9/0.2-0.5-0.8/0.1-0.2-0.5	mittlere Auflager	4,643300E-02	7,693350E-16
1-V8	GS / Var. 8	3/4/3 /3-0-3/3-0-3/0-0-3/0-0-0	0.5-0.8-0.9/0.2-0.5-0.8/0.1-0.2-0.5	keine	4,130910E-02	3,547370E-13
2-V8	ZS / Var. 8	3/4/3 /3-0-3/3-0-3/0-0-3/0-0-0	0.5-0.8-0.9/0.2-0.5-0.8/0.1-0.2-0.5	keine	4,643300E-02	1,165930E-14
1-V9	GS / Var. 9	3/4/3 /0-0-3/0-0-3/0-0-3/6-0-0	0.0-0.7625/0.0-0.5/0.2375-1.0	alle Auflager, 2 Nullst., Mitte	1,779410E-02	3,935170E-48
2-V9	ZS / Var. 9	3/4/3 /0-0-3/0-0-3/0-0-3/6-0-0	0.0-0.7625/0.0-0.5/0.2375-1.0	alle Auflager, 2 Nullst., Mitte	2,322000E-02	6,133740E-09
1-V10	GS / Var. 10	3/4/3 /0-0-2/0-0-2/0-0-2/6-0-0	0.0-0.7625/0.0-0.5/0.2375-1.0	alle Auflager, 2 Nullst., Mitte	2,648000E-15	3,533950E-28
2-V10	ZS / Var. 10	3/4/3 /0-0-3/0-0-3/0-0-3/6-0-0	0.0-0.7625/0.0-0.5/0.2375-1.0	alle Auflager, 2 Nullst., Mitte	3,035700E-02	3,035700E-02
1-V11	GS / Var. 11	3/4/3 /0-0-2/0-0-2/0-0-2/7-0-2	0.0-0.7625/0.0-0.5/0.0-0.2375-1.0 li+re w''=0	alle Auflager, 2 Nullst., Mitte	1,793310E-02	-7,802170E-02
2-V11	ZS / Var. 11	3/4/3 /0-0-2/0-0-2/0-0-2/7-0-2	0.0-0.7625/0.0-0.5/0.0-0.2375-1.0 li+re w''=0	alle Auflager, 2 Nullst., Mitte	1,747520E-02	-2,349200E-01

Tab. A-2-3: Zusammenstellung der Ergebnisse

3Feld-Balken mit Gleichlast q: Durchbiegung w beide Lösungswege

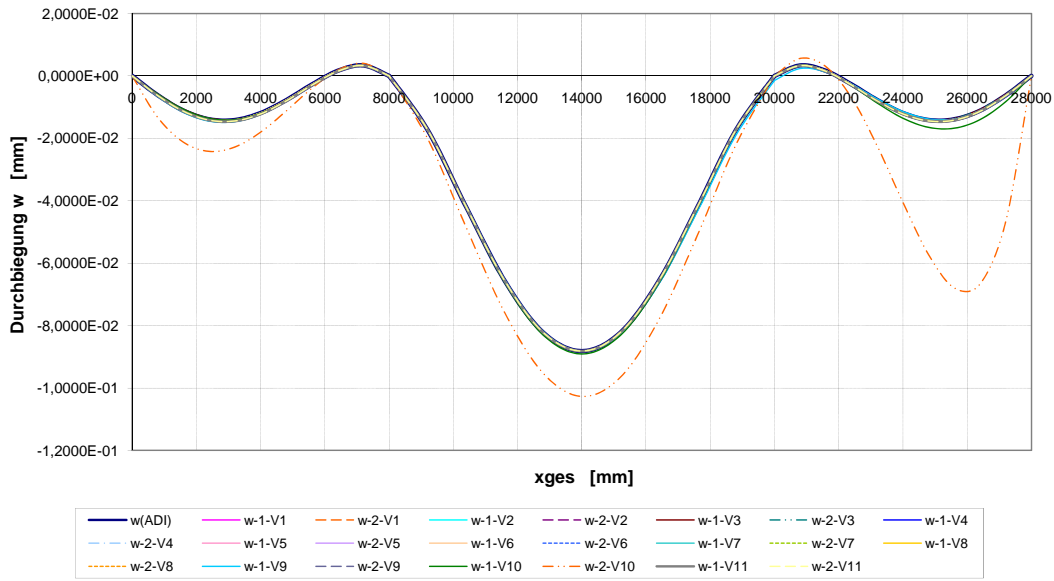


Bild A-2-2a

3Feld-Balken mit Gleichlast q: Winkelneigung w' beide Lösungswege

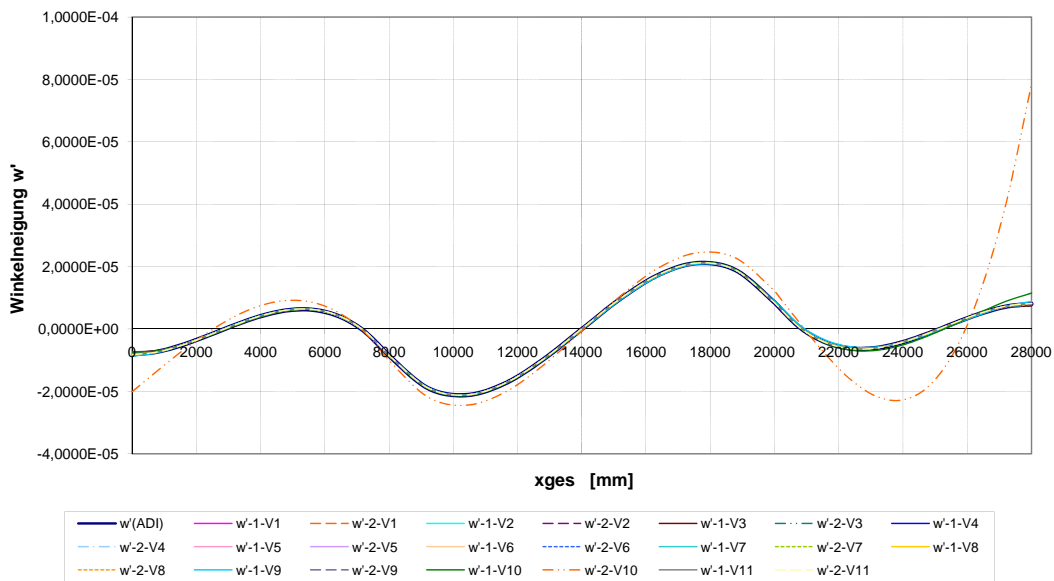


Bild A-2-2b

3Feld-Balken mit Gleichlast q: Krümmung w'' beide Lösungswege

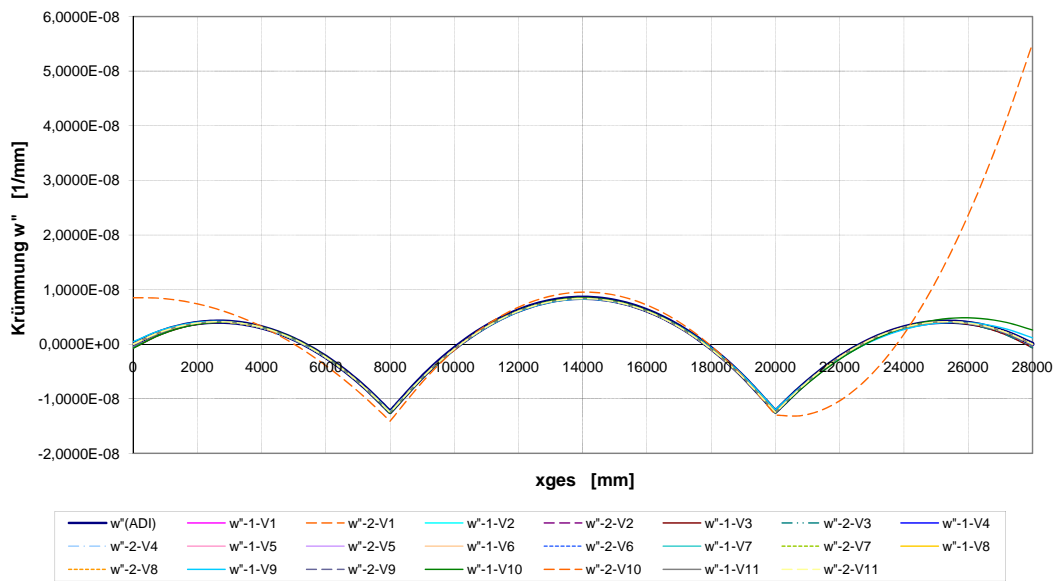


Bild A-2-2c

2.2.2 Beispiel 2-2.2 5Feld-Balken mit Gleichlast q und 5 Einzellasten

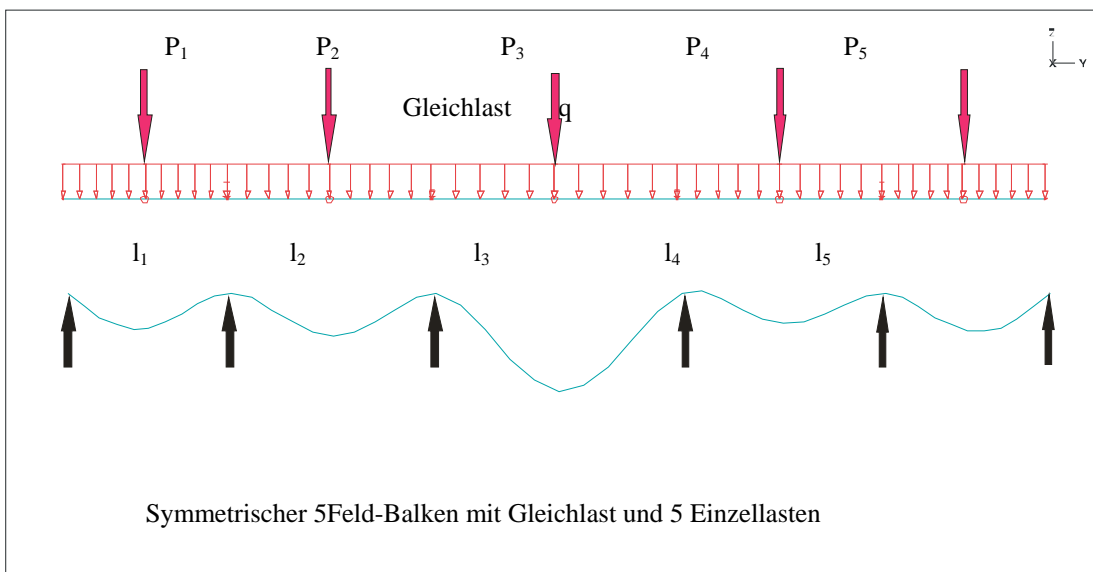


Bild 8.1a: Beispiel für zwängungsfreie Biegung

Für das in Abschnitt 8.1 dargestellte Beispiel des 5Feld-Balkens wird noch einmal zum einen der Einfluss der Randbedingungen besonders hervorgehoben, und zum anderen werden auch die Lösungen der unterschiedlichen Ausgleichsansätze gegenübergestellt.

Es liegt eine limitierte Zahl der Messpunkte vor, und zwar nur als Dehnungen; die Lösung erfordert dann feldweise mindestens je 2 Zusatzbedingungen. Es werden für das symmetrische System auch symmetrische Randbedingungen zur Stabilisierung der Lösung eingeführt.

	Ab/Polw/Funw/w/w'/ε/r=	Q = Bt * A * B	Determinante
ADINA			
Gesamt-Abstand	5/4/3 /0/0/3-3-3-3-3/ 11-0-0	2,139610E+00	-4,207240E-33
Gesamt-Flaeche-Parabel	5/4/3 /0/0/3-3-3-3-3/ 11-0-0	2,138520E+00	-2,817720E-27
Zwischen-Abstand	5/4/3 /0/0/3-3-3-3-3/ 11-0-0	4,541270E+00	-2,013360E-09
mod. Zwischen-Abstand	5/4/3 /0/0/3-3-3-3-3/ 11-0-0	3,846500E+00	-1,549320E-08
mod. Zwischen-Flaeche-Parabel	5/4/3 /0/0/3-3-3-3-3/ 11-0-0	-6,002530E+03	-7,628870E-02

Tab. A-2-4a: Alle Rechnungen mit 11 Randbedingungen

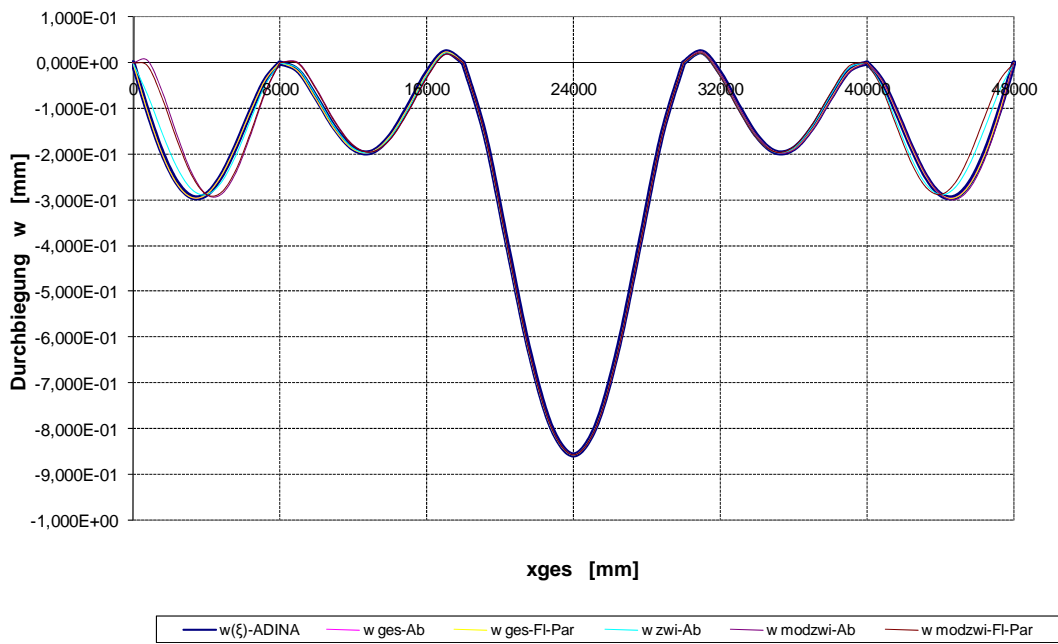
	Ab/Polw/Funw/w/w'/ε/r=	Q = Bt * A * B	Determinante
ADINA			
Gesamt-Abstand	5/4/3 /0/0/3-3-3-3-3/ 13-0-0	2,648090E+00	-9,709870E-15
Gesamt-Flaeche-Parabel	5/4/3 /0/0/3-3-3-3-3/ 13-0-0	2,648090E+00	-9,709870E-15
Zwischen-Abstand	5/4/3 /0/0/3-3-3-3-3/ 13-0-0	4,741680E+00	-2,400270E-14
mod. Zwischen-Abstand	5/4/3 /0/0/3-3-3-3-3/ 13-0-0	6,788230E+00	-9,709870E-15
mod. Zwischen-Flaeche-Parabel	5/4/3 /0/0/3-3-3-3-3/ 13-0-0	-3,423510E+03	-9,709870E-15

Tab. A-2-4b: Alle Rechnungen mit 13 Randbedingungen

Die Lösungen sind in Bild A-2-3a-c zusammengestellt; hier wird noch einmal der eminente Einfluss zusätzlicher Zwangsbedingungen deutlich.

Die Gesamtschritt-Lösung ergibt in fast allen Fällen die beste Übereinstimmung mit dem „genauen“ Verlauf [z. B. aus der statischen Berechnung].

5Feld-Balken mit p + 5P: W mit Bernoulli-Ansatz + 11 Randbedingungen



5Feld-Balken mit p + 5P: W mit Bernoulli-Ansatz + 13 Randbedingungen

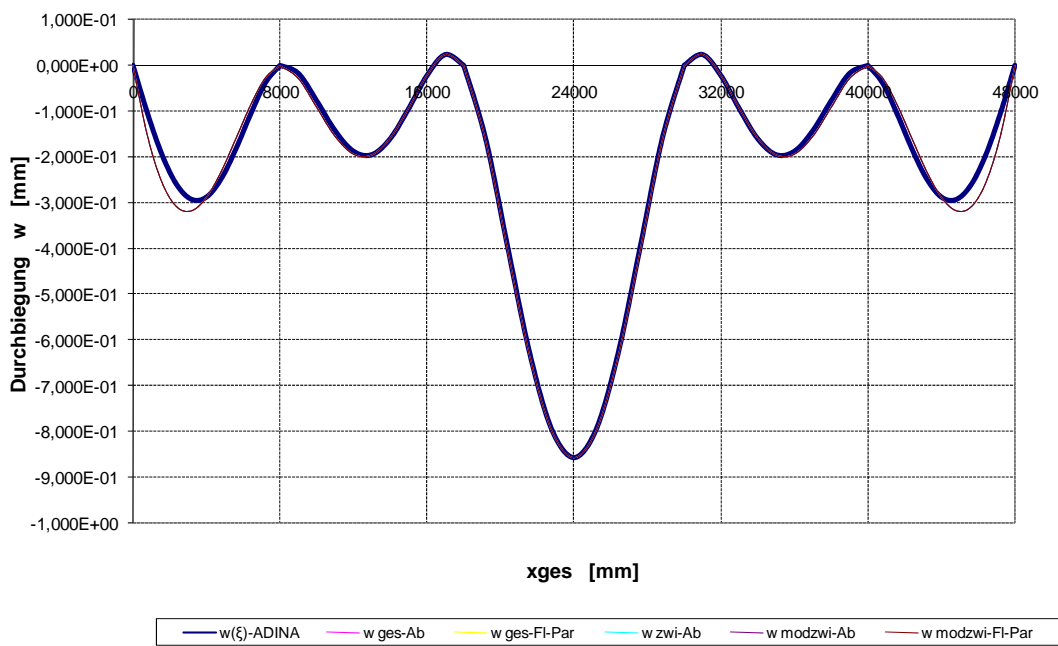
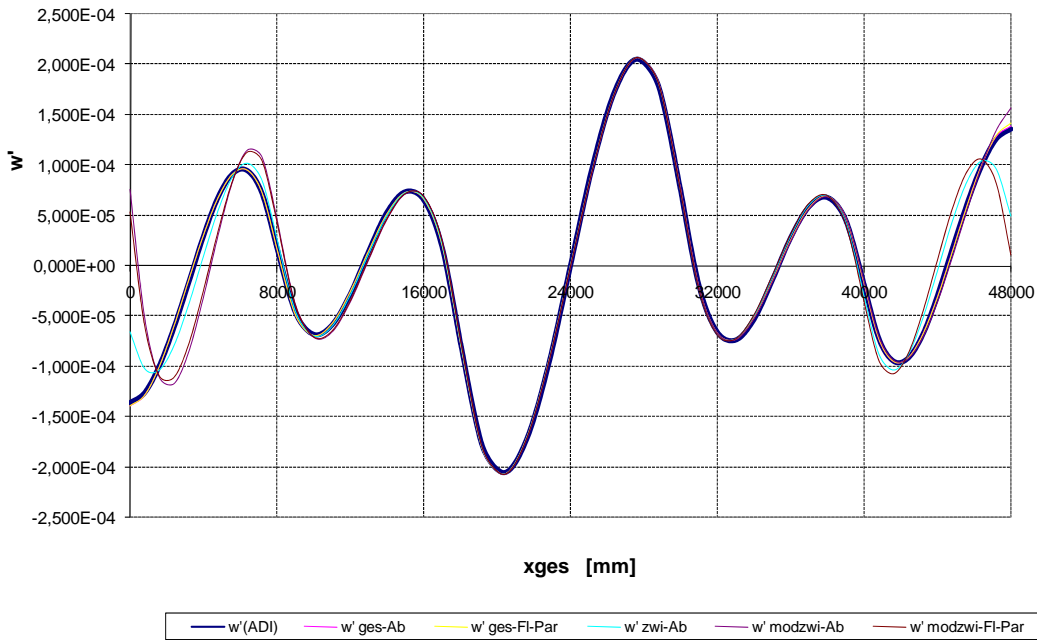


Bild A-2-3a

5Feld-Balken mit p + 5P: w' mit Bernoulli-Ansatz 11 Randbedingungen



5Feld-Balken mit p + 5P: w' mit Bernoulli-Ansatz + 13 Randbedingungen

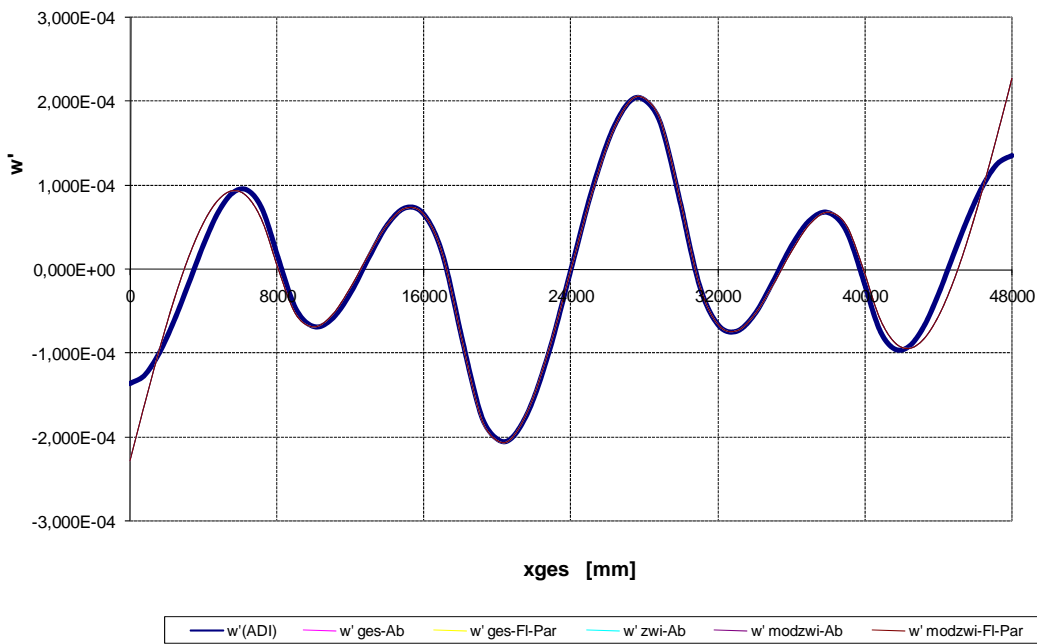
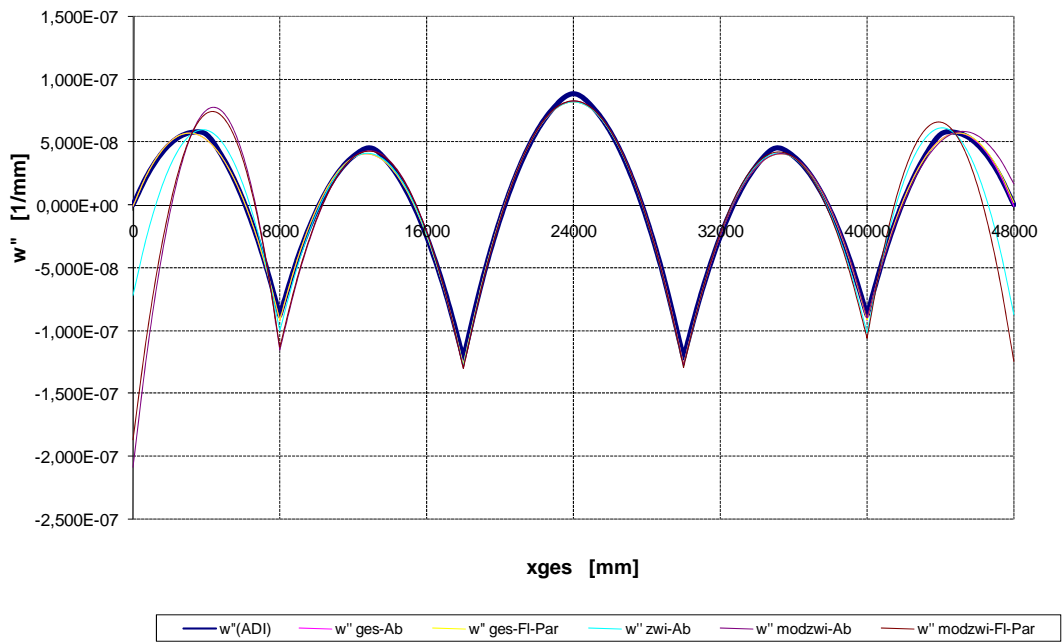


Bild A-2-3b

5Feld-Balken mit $p + 5P$: w'' mit Bernoulli-Ansatz + 11 Randbedingungen



5Feld-Balken mit $p + 5P$: w'' mit Bernoulli-Ansatz + 13 Randbedingungen

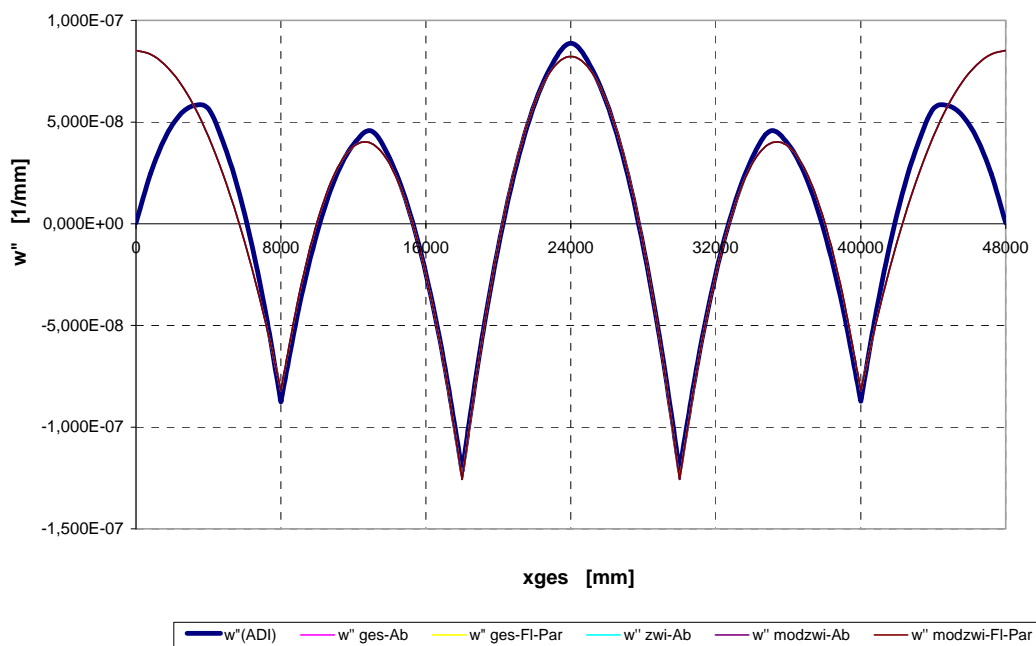


Bild A-2-3c

2.2.3 Beispiel 2-2.3 Sonderfall

Liegen unter Umständen zu wenige Messpunkte vor, kann es im Sonderfall zum Durchschlageeffekt bei der sonst besten Lösung mit der Optimierung zum Abstand zur Längsachse kommen. Dies soll am Beispiel der Funktion $\sin\left(\frac{\pi}{180} x\right)$ gezeigt werden.

Die \star -Symbole bezeichnen die gegebenen (symmetrischen) Messpunkte, die hier jeweils Durchbiegungen w bedeuten, und zwar

- in Bild A-2-4a bei 0 90 180
- in Bild A-2-4b bei 10 90 170
- in Bild A-2-4c bei 30 90 150

Randbedingungen sind nicht definiert. Der Ausgleich mit der üblichen Abstandsoptimierung führt in diesem Sonderfall bei Bild A-2-4a zu einem Durchschlagen der Funktion; die Zusatzbedingung der Flächenoptimierung ergibt bei linearer „Verbindung“ der Messpunkte eine relative Verbesserung, dagegen liefert der parabolische Flächenausgleich eine gute Lösung. Hierbei gilt zusätzlich, dass die sinus-Funktion durch einen begrenzten Ansatz approximiert wird.

Gleiches gilt im Grundsatz auch für Bilder A-2-4b und c, jedoch wird der Flächenausgleich nur innerhalb der definierten Punkte vollzogen. Beim Abstandsausgleich ist eine merkbare Korrektur des Durchschlageeffektes zu beobachten, und zwar allein dadurch, dass die Randpunkte mehr nach innen gerückt werden.

Prinzipiell ist zu konstatieren, dass wenige Punkte zur Erstellung einer brauchbaren Lösung notwendigerweise gute Zusatzbedingungen erfordern.

Vergleich sin: 3 w 0-90-180 keine Rdbd.

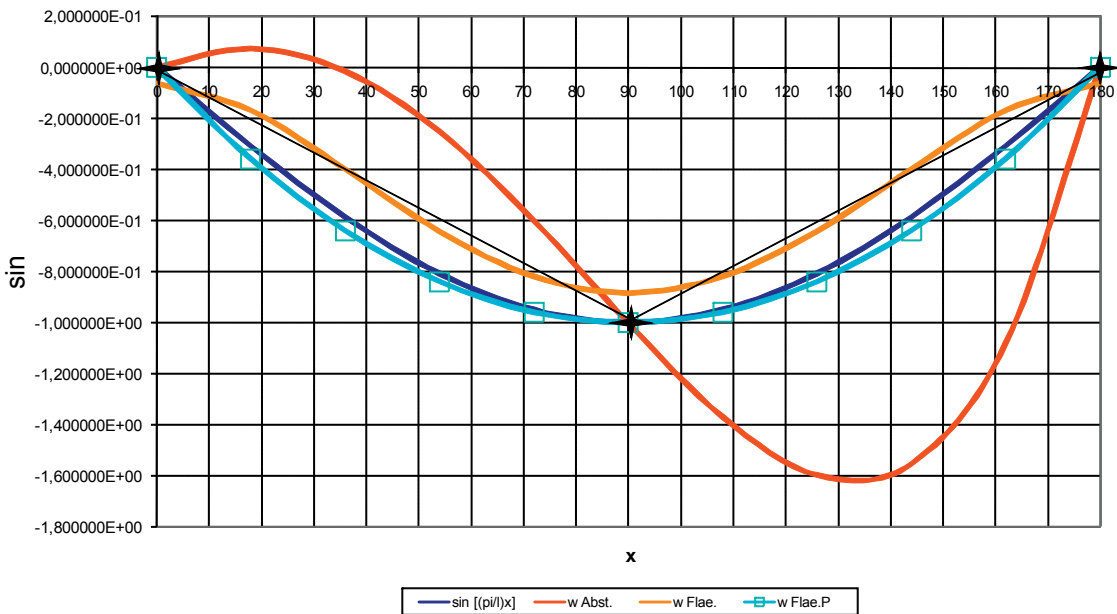


Bild A-2-4a

Vergleich sin: 3 w 10-90-170 keine Rdbd.

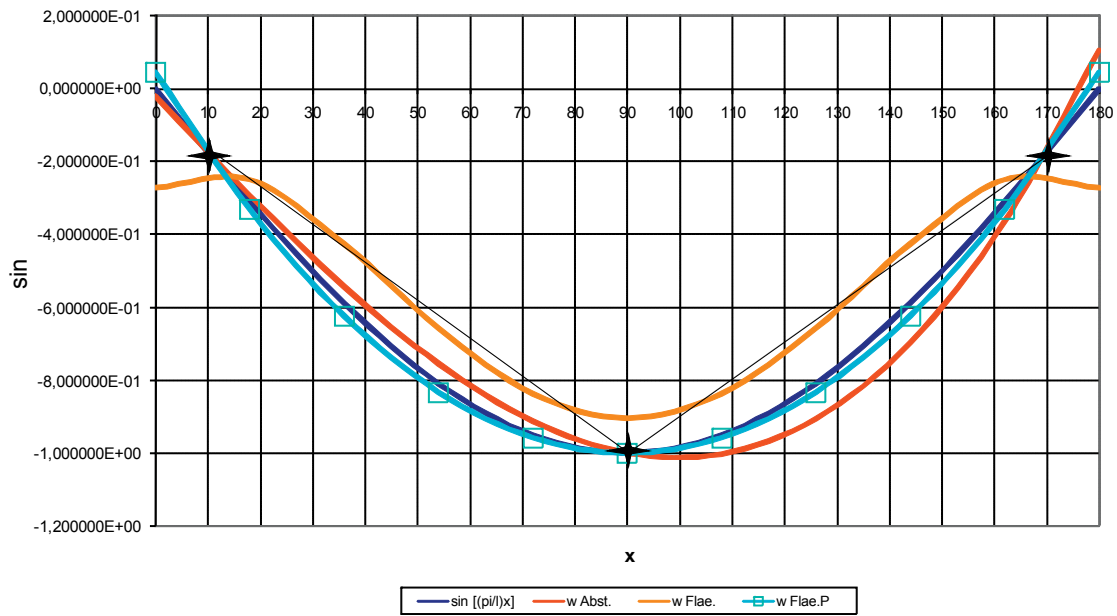


Bild A-2-4b

Vergleich sin: 3 w 30-90-150 keine Rdbd.

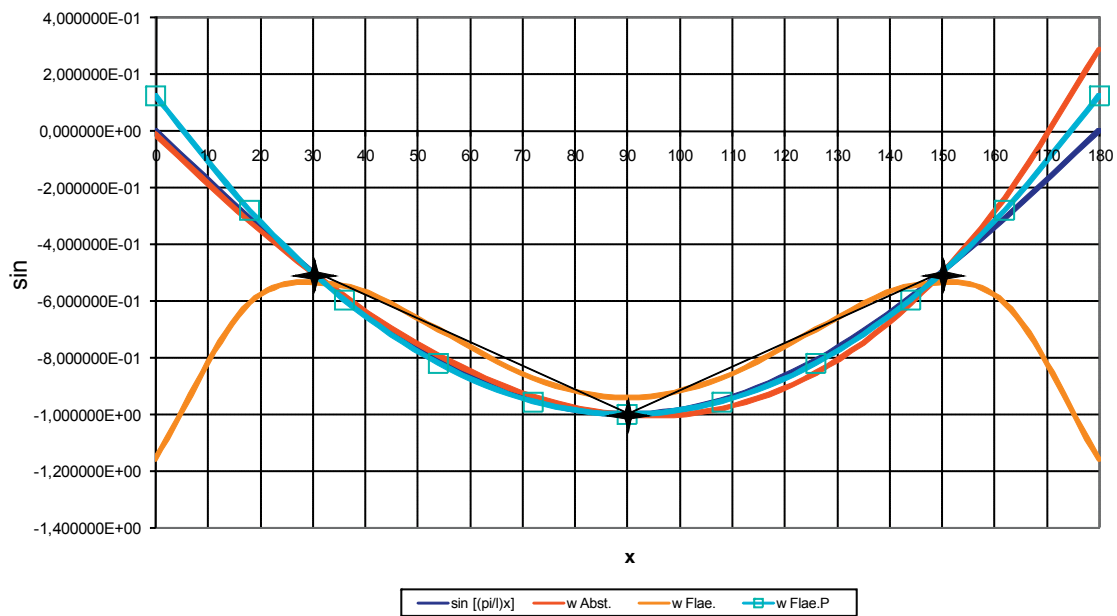


Bild A-2-4c

12 Anhang B: Matrizen-Übersicht und Lösungsweg

1	Zusammenstellung der grundsätzlichen Beziehungen	78
2	Zusammenstellung der Ausgangsmatrizen nach Bernoulli-Ansatz	79
2.1	Fehlerausgleich mit diskreten Messwerten allein	79
2.2	Fehlerausgleich mit Zusatzbedingungen	82
3	Zusammenstellung der Ausgangsmatrizen mit Hermite-Polynomen	84
3.1	Hermite-Polynome 4. Ordnung	84
3.2	Hermite-Polynome 6. Ordnung	86
4	Lösungsweg	88
4.1	Programm-Input	89
4.2	Quellcode zur Gesamtschritt- Lösung	91

1 Zusammenstellung der grundsätzlichen Beziehungen

Einleitend werden noch einmal die grundsätzlichen Beziehungen zusammengestellt:

$\mathbf{F} = \{F_{ij}\}$ Matrix mit den Werten der beliebigen Ansatzfunktionen an den Messpunkten i

$\mathbf{c} = \{c_j\}$ Vektor der Regressionsparameter j

$\mathbf{G} = \{G_i\}$ Vektor der Messwerte an den Messpunkten i

$\mathbf{d} = \{d_i\}$ Vektor der Differenzen zwischen Messwert und Regressionskurve an den Punkten i

und

$Q = \sum d_i^2 = \mathbf{d}^T \times \mathbf{d}$ mit \mathbf{d}^T als Transponierte von \mathbf{d}

$\mathbf{R} = \{R_{ij}\}$ Matrix der Rand- bzw. Nebenbedingungen für beliebige Punkte i bzw. für beliebige Messpunkte j

$\mathbf{b} = \{b_j\}$ Vektor der am System bekannten Rand- bzw. Nebenbedingungen

$\boldsymbol{\lambda} = \{\lambda_j\}$ Vektor der Lagrange'schen Multiplikatoren für die Anpassung an das gegebene mechanische System

Das Ausgangs-System ist dann in Matrizen-Schreibweise:

$$\begin{bmatrix} 2*\mathbf{F}^T \mathbf{F} & \mathbf{R}^T \\ \mathbf{R} & \mathbf{0} \end{bmatrix} * \begin{bmatrix} \mathbf{c} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} 2*\mathbf{F}^T \mathbf{G} \\ \mathbf{b} \end{bmatrix} \quad \text{bzw.}$$

$$\begin{bmatrix} \mathbf{B} & \mathbf{R}^T \\ \mathbf{R} & \mathbf{0} \end{bmatrix} * \begin{bmatrix} \mathbf{c} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{A} \\ \mathbf{b} \end{bmatrix} \quad (2.5)$$

Die Differenzierung nach der Art des Fehlerausgleichs kann übersichtlicher in Summenschreibweise erfolgen; der Fehlerausgleich mit diskreten Messwerten allein ist dann:

$$\left[\sum_i^{1,N} \left\{ \sum_j^{1,M} f_{ij} c_j - g_i \right\} \right]^2 + \left[\sum_n^{1,K} \lambda_n \left\{ \sum_j^{1,M} r_{nj} c_j - b_n \right\} \right] = Q \implies \mathbf{Min} \quad (2.2ab)$$

Wird der Fehlerausgleich mit Zusatzbedingungen vollzogen, folgt in allgemeiner Schreibweise:

$$\int_{x_1}^{x_2} \left[\sum_i^{1,N} \left\{ \sum_j^{1,M} f(x)_{ij} c_j - g(x)_i \right\} \right]^2 dx + \left[\sum_n^{1,K} \lambda_n \left\{ \sum_j^{1,M} r_{nj} c_j - b_n \right\} \right] \implies \mathbf{Min} \quad (2.2bb)$$

Letzterer Weg der Regression stellt jedoch immer einen Sonderfall dar, falls - z. B. auf Grund ungenügender Messdaten - anderweitig keine realistische Lösung zu erzielen ist.

2 Zusammenstellung der Ausgangsmatrizen nach Bernoulli-Ansatz

Entsprechend (2.5) werden die Ausgangsmatrizen für die Balkenbiegung nach Bernoulli zusammengestellt:

2.1 Fehlerausgleich mit diskreten Messwerten allein

Die gängigste und allgemein optimale Art des Ausgleiches wird hier angegeben:

$$\mathbf{F} = \begin{bmatrix} 1 & \xi_1 & \xi_1^2 & \xi_1^3 & \xi_1^4 \\ 1 & \xi_2 & \xi_2^2 & \xi_2^3 & \xi_2^4 \\ 1 & \xi_3 & \xi_3^2 & \xi_3^3 & \xi_3^4 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & \xi_i & \xi_i^2 & \xi_i^3 & \xi_i^4 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & \xi_N & \xi_N^2 & \xi_N^3 & \xi_N^4 \end{bmatrix} \quad \text{bzw.} \quad \left. \begin{array}{l} \\ \\ \\ \\ \\ \\ \\ \end{array} \right\} \quad (3.4)$$

$$\mathbf{F}^T = \begin{bmatrix} 1 & 1 & 1 & \cdot & \cdot & 1 & \cdot & 1 \\ \xi_1 & \xi_2 & \xi_3 & \cdot & \cdot & \xi_i & \cdot & \xi_N \\ \xi_1^2 & \xi_2^2 & \xi_3^2 & \cdot & \cdot & \xi_i^2 & \cdot & \xi_N^2 \\ \xi_1^3 & \xi_2^3 & \xi_3^3 & \cdot & \cdot & \xi_i^3 & \cdot & \xi_N^3 \\ \xi_1^4 & \xi_2^4 & \xi_3^4 & \cdot & \cdot & \xi_i^4 & \cdot & \xi_N^4 \end{bmatrix}$$

Die Grundmatrix

$$\mathbf{B}[\mathbf{w}] = 2 * \mathbf{F}^T \mathbf{F}$$

und die Lastmatrix

$$\mathbf{A}[\mathbf{w}] = 2 * \mathbf{F}^T \mathbf{G}[\mathbf{w}]$$

werden gebildet mit

- N_1 als Anzahl der Durchbiegungsmesswerte w_i , $i=1,N_1$ bzw.
- N_2 als Anzahl der Winkelneigungsmesswerte w_i' , $i=1,N_2$ und
- N_3 als Anzahl der Krümmungsmesswerte w_i'' , $i=1,N_3$.

Die vollständige Auswertung führt zu:

$$\mathbf{B}[\mathbf{w}] = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} & b_{15} \\ b_{21} & b_{22} & b_{23} & b_{24} & b_{25} \\ b_{31} & b_{32} & b_{33} & b_{34} & b_{35} \\ b_{41} & b_{42} & b_{43} & b_{44} & b_{45} \\ b_{51} & b_{52} & b_{53} & b_{54} & b_{55} \end{bmatrix} \quad \mathbf{A}[\mathbf{w}] = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} \quad (4.2)$$

$$\mathbf{B}[\mathbf{w}] = 2 * \begin{bmatrix} \sum_i^{1,N_1} 1 * 1 & \sum_i^{1,N_1} 1 * \xi_i & \sum_i^{1,N_1} 1 * \xi_i^2 & \sum_i^{1,N_1} 1 * \xi_i^3 & \sum_i^{1,N_1} 1 * \xi_i^4 \\ \sum_i^{1,N_1} \xi_i * 1 & \sum_i^{1,N_1} \xi_i * \xi_i & \sum_i^{1,N_1} \xi_i * \xi_i^2 & \sum_i^{1,N_1} \xi_i * \xi_i^3 & \sum_i^{1,N_1} \xi_i * \xi_i^4 \\ \sum_i^{1,N_1} \xi_i^2 * 1 & \sum_i^{1,N_1} \xi_i^2 * \xi_i & \sum_i^{1,N_1} \xi_i^2 * \xi_i^2 & \sum_i^{1,N_1} \xi_i^2 * \xi_i^3 & \sum_i^{1,N_1} \xi_i^2 * \xi_i^4 \\ \sum_i^{1,N_1} \xi_i^3 * 1 & \sum_i^{1,N_1} \xi_i^3 * \xi_i & \sum_i^{1,N_1} \xi_i^3 * \xi_i^2 & \sum_i^{1,N_1} \xi_i^3 * \xi_i^3 & \sum_i^{1,N_1} \xi_i^3 * \xi_i^4 \\ \sum_i^{1,N_1} \xi_i^4 * 1 & \sum_i^{1,N_1} \xi_i^4 * \xi_i & \sum_i^{1,N_1} \xi_i^4 * \xi_i^2 & \sum_i^{1,N_1} \xi_i^4 * \xi_i^3 & \sum_i^{1,N_1} \xi_i^4 * \xi_i^4 \end{bmatrix}$$

$$\mathbf{A}[\mathbf{w}] = 2 * \begin{bmatrix} \sum_i^{1,N_1} 1 * w_i \\ \sum_i^{1,N_1} \xi_i * w_i \\ \sum_i^{1,N_1} \xi_i^2 * w_i \\ \sum_i^{1,N_1} \xi_i^3 * w_i \\ \sum_i^{1,N_1} \xi_i^4 * w_i \end{bmatrix}$$

Für die Winkelneigungen erhält man:

$$\mathbf{B}[\mathbf{w}'] = 2 * \frac{1}{s_k^2} \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 1 * 1 \sum_i^{1, N_2} 1 * 1 & 1 * 2 \sum_i^{1, N_2} 1 * \xi_i & 1 * 3 \sum_i^{1, N_2} 1 * \xi_i^2 & 1 * 4 \sum_i^{1, N_2} 1 * \xi_i^3 \\ \mathbf{0} & 2 * 1 \sum_i^{1, N_2} \xi_i * 1 & 2 * 2 \sum_i^{1, N_2} \xi_i * \xi_i & 2 * 3 \sum_i^{1, N_2} \xi_i * \xi_i^2 & 2 * 4 \sum_i^{1, N_2} \xi_i * \xi_i^3 \\ \mathbf{0} & 3 * 1 \sum_i^{1, N_2} \xi_i^2 * 1 & 3 * 2 \sum_i^{1, N_2} \xi_i^2 * \xi_i & 3 * 3 \sum_i^{1, N_2} \xi_i^2 * \xi_i^2 & 3 * 4 \sum_i^{1, N_2} \xi_i^2 * \xi_i^3 \\ \mathbf{0} & 4 * 1 \sum_i^{1, N_2} \xi_i^3 * 1 & 4 * 2 \sum_i^{1, N_2} \xi_i^3 * \xi_i & 4 * 3 \sum_i^{1, N_2} \xi_i^3 * \xi_i^2 & 4 * 4 \sum_i^{1, N_2} \xi_i^3 * \xi_i^3 \end{bmatrix}$$

$$\mathbf{A}[\mathbf{w}'] = 2 * \frac{1}{s_k} \begin{bmatrix} \mathbf{0} \\ 1 \sum_i^{1, N_2} 1 * w'_i \\ 2 \sum_i^{1, N_2} \xi_i * w'_i \\ 3 \sum_i^{1, N_2} \xi_i^2 * w'_i \\ 4 \sum_i^{1, N_2} \xi_i^3 * w'_i \end{bmatrix}$$

Für die Krümmungen folgt:

$$\mathbf{B}[\mathbf{w}''] = 2 * \frac{1}{s_k^4} \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 2 * 2 \sum_i^{1, N_3} 1 * 1 & 2 * 6 \sum_i^{1, N_3} 1 * \xi_i & 2 * 12 \sum_i^{1, N_3} 1 * \xi_i^2 \\ \mathbf{0} & \mathbf{0} & 6 * 2 \sum_i^{1, N_3} \xi_i * 1 & 6 * 6 \sum_i^{1, N_3} \xi_i * \xi_i & 6 * 12 \sum_i^{1, N_3} \xi_i * \xi_i^2 \\ \mathbf{0} & \mathbf{0} & 12 * 2 \sum_i^{1, N_3} \xi_i^2 * 1 & 12 * 6 \sum_i^{1, N_3} \xi_i^2 * \xi_i & 12 * 12 \sum_i^{1, N_3} \xi_i^2 * \xi_i^2 \end{bmatrix}$$

$$\mathbf{A}[\mathbf{w}''] = 2 * \frac{1}{s_k^2} \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ 2 * 1 \sum_i^{1, N_3} 1 * \varepsilon_i / z_i \\ 3 * 2 \sum_i^{1, N_3} \xi_i * \varepsilon_i / z_i \\ 4 * 3 \sum_i^{1, N_3} \xi_i^2 * \varepsilon_i / z_i \end{bmatrix}$$

2.2 Fehlerausgleich mit Zusatzbedingungen

Im Hauptteil sind die Matrizen nur für den Abstandsausgleich angegeben worden; sinngemäß erfolgt dies hier für das Kriterium des Flächenausgleichs unter der Annahme von jeweils versetzten Parabeln, die die Kurve der Messwerte approximieren. Die allgemeinen Beziehungen behalten ihre Gültigkeit. Die Summation ist über die Zahl der stückweisen Parabeln zu führen.

$$\mathbf{B}[\mathbf{w}]^{\text{PF}} = 2s_k * \begin{bmatrix} \sum_i^{1, N_1-1} (\xi_2 - \xi_1) & \sum_i^{1, N_1-1} \frac{1}{2} (\xi_2^2 - \xi_1^2) & \sum_i^{1, N_1-1} \frac{1}{3} (\xi_2^3 - \xi_1^3) & \sum_i^{1, N_1-1} \frac{1}{4} (\xi_2^4 - \xi_1^4) & \sum_i^{1, N_1-1} \frac{1}{5} (\xi_2^5 - \xi_1^5) \\ \sum_i^{1, N_1-1} \frac{1}{2} (\xi_2^2 - \xi_1^2) & \sum_i^{1, N_1-1} \frac{1}{3} (\xi_2^3 - \xi_1^3) & \sum_i^{1, N_1-1} \frac{1}{4} (\xi_2^4 - \xi_1^4) & \sum_i^{1, N_1-1} \frac{1}{5} (\xi_2^5 - \xi_1^5) & \sum_i^{1, N_1-1} \frac{1}{6} (\xi_2^6 - \xi_1^6) \\ \sum_i^{1, N_1-1} \frac{1}{3} (\xi_2^3 - \xi_1^3) & \sum_i^{1, N_1-1} \frac{1}{4} (\xi_2^4 - \xi_1^4) & \sum_i^{1, N_1-1} \frac{1}{5} (\xi_2^5 - \xi_1^5) & \sum_i^{1, N_1-1} \frac{1}{6} (\xi_2^6 - \xi_1^6) & \sum_i^{1, N_1-1} \frac{1}{7} (\xi_2^7 - \xi_1^7) \\ \sum_i^{1, N_1-1} \frac{1}{4} (\xi_2^4 - \xi_1^4) & \sum_i^{1, N_1-1} \frac{1}{5} (\xi_2^5 - \xi_1^5) & \sum_i^{1, N_1-1} \frac{1}{6} (\xi_2^6 - \xi_1^6) & \sum_i^{1, N_1-1} \frac{1}{7} (\xi_2^7 - \xi_1^7) & \sum_i^{1, N_1-1} \frac{1}{8} (\xi_2^8 - \xi_1^8) \\ \sum_i^{1, N_1-1} \frac{1}{5} (\xi_2^5 - \xi_1^5) & \sum_i^{1, N_1-1} \frac{1}{6} (\xi_2^6 - \xi_1^6) & \sum_i^{1, N_1-1} \frac{1}{7} (\xi_2^7 - \xi_1^7) & \sum_i^{1, N_1-1} \frac{1}{8} (\xi_2^8 - \xi_1^8) & \sum_i^{1, N_1-1} \frac{1}{9} (\xi_2^9 - \xi_1^9) \end{bmatrix}$$

Mit dem stückweise parabolischen Ansatz für die Durchbiegung $w(\xi) = a_1 + a_2 \xi + a_3 \xi^2$, die aus den Messwerten w_i vorab zu ermitteln sind, erhält man lastseitig:

$$\mathbf{A}[\mathbf{w}]^{\text{PF}} = 2s_k * \begin{bmatrix} \sum_i^{1, N_1-1} \left[a_1 (\xi_2 - \xi_1) + a_2 \frac{1}{2} (\xi_2^2 - \xi_1^2) + a_3 \frac{1}{3} (\xi_2^3 - \xi_1^3) \right] \\ \sum_i^{1, N_1-1} \left[a_1 \frac{1}{2} (\xi_2^2 - \xi_1^2) + a_2 \frac{1}{3} (\xi_2^3 - \xi_1^3) + a_3 \frac{1}{4} (\xi_2^4 - \xi_1^4) \right] \\ \sum_i^{1, N_1-1} \left[a_1 \frac{1}{3} (\xi_2^3 - \xi_1^3) + a_2 \frac{1}{4} (\xi_2^4 - \xi_1^4) + a_3 \frac{1}{5} (\xi_2^5 - \xi_1^5) \right] \\ \sum_i^{1, N_1-1} \left[a_1 \frac{1}{4} (\xi_2^4 - \xi_1^4) + a_2 \frac{1}{5} (\xi_2^5 - \xi_1^5) + a_3 \frac{1}{6} (\xi_2^6 - \xi_1^6) \right] \\ \sum_i^{1, N_1-1} \left[a_1 \frac{1}{5} (\xi_2^5 - \xi_1^5) + a_2 \frac{1}{6} (\xi_2^6 - \xi_1^6) + a_3 \frac{1}{7} (\xi_2^7 - \xi_1^7) \right] \end{bmatrix}$$

Für die Winkelneigungen erhält man:

$$\mathbf{B}[\mathbf{w}']^{\text{PF}} = \frac{2}{s_k} * \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 * 1 \sum_i^{1, N_2-1} (\xi_2 - \xi_1) & 1 * 2 \sum_i^{1, N_2-1} \frac{1}{2} (\xi_2^2 - \xi_1^2) & 1 * 3 \sum_i^{1, N_2-1} \frac{1}{3} (\xi_2^3 - \xi_1^3) & 1 * 4 \sum_i^{1, N_2-1} \frac{1}{4} (\xi_2^4 - \xi_1^4) \\ 0 & 2 * 1 \sum_i^{1, N_2-1} \frac{1}{2} (\xi_2^2 - \xi_1^2) & 2 * 2 \sum_i^{1, N_2-1} \frac{1}{3} (\xi_2^3 - \xi_1^3) & 2 * 3 \sum_i^{1, N_2-1} \frac{1}{4} (\xi_2^4 - \xi_1^4) & 2 * 4 \sum_i^{1, N_2-1} \frac{1}{5} (\xi_2^5 - \xi_1^5) \\ 0 & 3 * 1 \sum_i^{1, N_2-1} \frac{1}{3} (\xi_2^3 - \xi_1^3) & 3 * 2 \sum_i^{1, N_2-1} \frac{1}{4} (\xi_2^4 - \xi_1^4) & 3 * 3 \sum_i^{1, N_2-1} \frac{1}{5} (\xi_2^5 - \xi_1^5) & 3 * 4 \sum_i^{1, N_2-1} \frac{1}{6} (\xi_2^6 - \xi_1^6) \\ 0 & 4 * 1 \sum_i^{1, N_2-1} \frac{1}{4} (\xi_2^4 - \xi_1^4) & 4 * 2 \sum_i^{1, N_2-1} \frac{1}{5} (\xi_2^5 - \xi_1^5) & 4 * 3 \sum_i^{1, N_2-1} \frac{1}{6} (\xi_2^6 - \xi_1^6) & 4 * 4 \sum_i^{1, N_2-1} \frac{1}{7} (\xi_2^7 - \xi_1^7) \end{bmatrix}$$

Mit dem stückweise parabolischen Ansatz für die Winkelneigung $w'(\xi) = b_1 + b_2 \xi + b_3 \xi^2$, die aus den Messwerten w'_i vorab zu ermitteln sind, erhält man lastseitig:

$$\mathbf{A}[\mathbf{w}']^{\text{PF}} = 2 * \begin{bmatrix} 0 \\ 1 \sum_i^{1, N_2-1} \left[b_1 (\xi_2 - \xi_1) + b_2 \frac{1}{2} (\xi_2^2 - \xi_1^2) + b_3 \frac{1}{3} (\xi_2^3 - \xi_1^3) \right] \\ 2 \sum_i^{1, N_2-1} \left[b_1 \frac{1}{2} (\xi_2^2 - \xi_1^2) + b_2 \frac{1}{3} (\xi_2^3 - \xi_1^3) + b_3 \frac{1}{4} (\xi_2^4 - \xi_1^4) \right] \\ 3 \sum_i^{1, N_2-1} \left[b_1 \frac{1}{3} (\xi_2^3 - \xi_1^3) + b_2 \frac{1}{4} (\xi_2^4 - \xi_1^4) + b_3 \frac{1}{5} (\xi_2^5 - \xi_1^5) \right] \\ 4 \sum_i^{1, N_2-1} \left[b_1 \frac{1}{4} (\xi_2^4 - \xi_1^4) + b_2 \frac{1}{5} (\xi_2^5 - \xi_1^5) + b_3 \frac{1}{6} (\xi_2^6 - \xi_1^6) \right] \end{bmatrix}$$

Für die Krümmungen folgt:

$$\mathbf{B}[\mathbf{w}'']^{\text{PF}} = \frac{2}{s_k^3} * \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 * 2 \sum_i^{1, N_3-1} (\xi_2 - \xi_1) & 2 * 6 \sum_i^{1, N_3-1} \frac{1}{2} (\xi_2^2 - \xi_1^2) & 2 * 12 \sum_i^{1, N_3-1} \frac{1}{3} (\xi_2^3 - \xi_1^3) \\ 0 & 0 & 6 * 2 \sum_i^{1, N_3-1} \frac{1}{2} (\xi_2^2 - \xi_1^2) & 6 * 6 \sum_i^{1, N_3-1} \frac{1}{3} (\xi_2^3 - \xi_1^3) & 6 * 12 \sum_i^{1, N_3-1} \frac{1}{4} (\xi_2^4 - \xi_1^4) \\ 0 & 0 & 12 * 2 \sum_i^{1, N_3-1} \frac{1}{3} (\xi_2^3 - \xi_1^3) & 12 * 6 \sum_i^{1, N_3-1} \frac{1}{4} (\xi_2^4 - \xi_1^4) & 12 * 12 \sum_i^{1, N_3-1} \frac{1}{5} (\xi_2^5 - \xi_1^5) \end{bmatrix}$$

Mit dem stückweise parabolischen Ansatz für die Krümmung $w''(\xi) = c_1 + c_2 \xi + c_3 \xi^2$, die aus den Messwerten w_i vorab zu ermitteln sind, erhält man lastseitig:

$$\mathbf{A}[\mathbf{w}'']^{\text{PF}} = \frac{2}{s_k} * \begin{bmatrix} 0 \\ 0 \\ 2 * 1 \sum_i^{1, N_3-1} \left[c_1 (\xi_2 - \xi_1) + c_2 \frac{1}{2} (\xi_2^2 - \xi_1^2) + c_3 \frac{1}{3} (\xi_2^3 - \xi_1^3) \right] \\ 3 * 2 \sum_i^{1, N_3-1} \left[c_1 \frac{1}{2} (\xi_2^2 - \xi_1^2) + c_2 \frac{1}{3} (\xi_2^3 - \xi_1^3) + c_3 \frac{1}{4} (\xi_2^4 - \xi_1^4) \right] \\ 4 * 3 \sum_i^{1, N_3-1} \left[c_1 \frac{1}{3} (\xi_2^3 - \xi_1^3) + c_2 \frac{1}{4} (\xi_2^4 - \xi_1^4) + c_3 \frac{1}{5} (\xi_2^5 - \xi_1^5) \right] \end{bmatrix}$$

3 Zusammenstellung der Ausgangsmatrizen mit Hermite-Polynomen

3.1 Hermite-Polynome 4. Ordnung

Hier wird nur die gängigste Art des Abstands-Ausgleiches angegeben; für Hermite-Polynome 4. Ordnung folgt dann $\mathbf{f}^T = \mathbf{H}^4 = \mathbf{a}_4 * \mathbf{x}_4$ bzw. explizit formuliert:

$$\begin{bmatrix} 1 & 0 & -3 & 2 \\ 0 & 0 & 3 & -2 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & -1 & 1 \end{bmatrix} * \begin{bmatrix} 1 \\ \xi \\ \xi^2 \\ \xi^3 \end{bmatrix} = \begin{bmatrix} H^{1-4} \\ H^{2-4} \\ H^{3-4} \\ H^{4-4} \end{bmatrix} \tag{3.1}$$

Hier ist

\mathbf{a}_4 die 4-parametrische Grundmatrix und \mathbf{x}_4 die zugehörige Vektorfunktion der Hermite-Polynome 4. Ordnung.

Die zugehörigen Ableitungen sind dann:

$$\begin{bmatrix} 1 & 0 & -3 & 2 \\ 0 & 0 & 3 & -2 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & -1 & 1 \end{bmatrix} * \frac{1}{s_k} \begin{bmatrix} 0 \\ 1 \\ 2\xi \\ 3\xi^2 \end{bmatrix} = \begin{bmatrix} H'^{1-4} \\ H'^{2-4} \\ H'^{3-4} \\ H'^{4-4} \end{bmatrix}$$

und

$$\begin{bmatrix} 1 & 0 & -3 & 2 \\ 0 & 0 & 3 & -2 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & -1 & 1 \end{bmatrix} * \frac{1}{s_k^2} \begin{bmatrix} 0 \\ 0 \\ 2 \\ 6\xi \end{bmatrix} = \begin{bmatrix} H''^{1-4} \\ H''^{2-4} \\ H''^{3-4} \\ H''^{4-4} \end{bmatrix}$$

$$\mathbf{F} = \begin{bmatrix} H_1^{1-4} & H_1^{2-4} & H_1^{3-4} & H_1^{4-4} \\ H_2^{1-4} & H_2^{2-4} & H_2^{3-4} & H_2^{4-4} \\ H_3^{1-4} & H_3^{2-4} & H_3^{3-4} & H_3^{4-4} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ H_i^{1-4} & H_i^{2-4} & H_i^{3-4} & H_i^{4-4} \\ \cdot & \cdot & \cdot & \cdot \\ H_N^{1-4} & H_N^{2-4} & H_N^{3-4} & H_N^{4-4} \end{bmatrix} \quad \text{bzw.} \quad (3.4)$$

$$\mathbf{F}^T = \begin{bmatrix} H_1^{1-4} & H_2^{1-4} & H_3^{1-4} & \cdot & \cdot & H_i^{1-4} & \cdot & H_N^{1-4} \\ H_1^{2-4} & H_2^{2-4} & H_3^{2-4} & \cdot & \cdot & H_i^{2-4} & \cdot & H_N^{2-4} \\ H_1^{3-4} & H_2^{3-4} & H_3^{3-4} & \cdot & \cdot & H_i^{3-4} & \cdot & H_N^{3-4} \\ H_1^{4-4} & H_2^{4-4} & H_3^{4-4} & \cdot & \cdot & H_i^{4-4} & \cdot & H_N^{4-4} \end{bmatrix}$$

Der untere Index gibt hier die Zahl des Messpunktes an, der erste obere Index die Reihe des Hermite-Polynoms, während der letzte obere Index die Ordnung des Hermite-Polynoms bezeichnet.

Mit

- N_1 als Anzahl der Durchbiegungsmesswerte w_i , $i=1, N_1$ bzw.
- N_2 als Anzahl der Winkelneigungsmesswerte w_i' , $i=1, N_2$ und
- N_3 als Anzahl der Krümmungsmesswerte w_i'' , $i=1, N_3$ folgt:

$$\mathbf{B}[\mathbf{w}] = 2 * \begin{bmatrix} \sum_i^{1, N_1} H_i^{1-4} H_i^{1-4} & \sum_i^{1, N_1} H_i^{1-4} H_i^{2-4} & \sum_i^{1, N_1} H_i^{1-4} H_i^{3-4} & \sum_i^{1, N_1} H_i^{1-4} H_i^{4-4} \\ \sum_i^{1, N_1} H_i^{2-4} H_i^{1-4} & \sum_i^{1, N_1} H_i^{2-4} H_i^{2-4} & \sum_i^{1, N_1} H_i^{2-4} H_i^{3-4} & \sum_i^{1, N_1} H_i^{2-4} H_i^{4-4} \\ \sum_i^{1, N_1} H_i^{3-4} H_i^{1-4} & \sum_i^{1, N_1} H_i^{3-4} H_i^{2-4} & \sum_i^{1, N_1} H_i^{3-4} H_i^{3-4} & \sum_i^{1, N_1} H_i^{3-4} H_i^{4-4} \\ \sum_i^{1, N_1} H_i^{4-4} H_i^{1-4} & \sum_i^{1, N_1} H_i^{4-4} H_i^{2-4} & \sum_i^{1, N_1} H_i^{4-4} H_i^{3-4} & \sum_i^{1, N_1} H_i^{4-4} H_i^{4-4} \end{bmatrix}$$

$$\mathbf{A}[\mathbf{w}] = 2 * \begin{bmatrix} \sum_i^{1, N_1} H_i^{1-4} * w_i \\ \sum_i^{1, N_1} H_i^{2-4} * w_i \\ \sum_i^{1, N_1} H_i^{3-4} * w_i \\ \sum_i^{1, N_1} H_i^{4-4} * w_i \end{bmatrix}$$

In entsprechender Weise sind dann $\mathbf{B}[\mathbf{w}']$ und $\mathbf{A}[\mathbf{w}']$ sowie $\mathbf{B}[\mathbf{w}'']$ und $\mathbf{A}[\mathbf{w}'']$ zu bilden.

3.2 Hermite-Polynome 6. Ordnung

In gleicher Weise können beispielsweise Hermite-Polynome 6. Ordnung angewandt werden:

$$\mathbf{f}^T = \mathbf{H}^6 = \mathbf{a}_6 * \mathbf{x}_6 \quad \text{oder}$$

$$\begin{bmatrix} 1 & 0 & 0 & -10 & -15 & -6 \\ 0 & 0 & 0 & -10 & -15 & 6 \\ 0 & 1 & 0 & -6 & 8 & -3 \\ 0 & 0 & 0 & -4 & 7 & -3 \end{bmatrix} * \begin{bmatrix} 1 \\ \xi \\ \xi^2 \\ \xi^3 \\ \xi^4 \\ \xi^5 \end{bmatrix} = \begin{bmatrix} \mathbf{H}^{1-6} \\ \mathbf{H}^{2-6} \\ \mathbf{H}^{3-6} \\ \mathbf{H}^{4-6} \end{bmatrix} \quad (3.2)$$

Hier ist

\mathbf{a}_6 die 6-parametrische Grundmatrix und
 \mathbf{x}_6 die zugehörige Vektorfunktion der Hermite-Polynome 6. Ordnung.

Die zugehörigen Ableitungen sind dann:

$$\begin{bmatrix} 1 & 0 & 0 & -10 & -15 & -6 \\ 0 & 0 & 0 & -10 & -15 & 6 \\ 0 & 1 & 0 & -6 & 8 & -3 \\ 0 & 0 & 0 & -4 & 7 & -3 \end{bmatrix} * \frac{1}{s_k} \begin{bmatrix} 0 \\ 1 \\ 2\xi \\ 3\xi^2 \\ 4\xi^3 \\ 5\xi^4 \end{bmatrix} = \begin{bmatrix} \mathbf{H}'^{1-6} \\ \mathbf{H}'^{2-6} \\ \mathbf{H}'^{3-6} \\ \mathbf{H}'^{4-6} \end{bmatrix}$$

und

$$\begin{bmatrix} 1 & 0 & 0 & -10 & -15 & -6 \\ 0 & 0 & 0 & -10 & -15 & 6 \\ 0 & 1 & 0 & -6 & 8 & -3 \\ 0 & 0 & 0 & -4 & 7 & -3 \end{bmatrix} * \frac{1}{s_k^2} \begin{bmatrix} 0 \\ 0 \\ 2 \\ 6\xi \\ 12\xi^2 \\ 20\xi^3 \end{bmatrix} = \begin{bmatrix} \mathbf{H}''^{1-6} \\ \mathbf{H}''^{2-6} \\ \mathbf{H}''^{3-6} \\ \mathbf{H}''^{4-6} \end{bmatrix}$$

$$\mathbf{F} = \begin{bmatrix} H_1^{1-6} & H_1^{2-6} & H_1^{3-6} & H_1^{4-6} \\ H_2^{1-6} & H_2^{2-6} & H_2^{3-6} & H_2^{4-6} \\ H_3^{1-6} & H_3^{2-6} & H_3^{3-6} & H_3^{4-6} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ H_i^{1-6} & H_i^{2-6} & H_i^{3-6} & H_i^{4-6} \\ \cdot & \cdot & \cdot & \cdot \\ H_N^{1-6} & H_N^{2-6} & H_N^{3-6} & H_N^{4-6} \end{bmatrix} \quad \text{bzw.} \quad (3.4)$$

$$\mathbf{F}^T = \begin{bmatrix} H_1^{1-6} & H_2^{1-6} & H_3^{1-6} & \cdot & \cdot & H_i^{1-6} & \cdot & H_N^{1-6} \\ H_1^{2-6} & H_2^{2-6} & H_3^{2-6} & \cdot & \cdot & H_i^{2-6} & \cdot & H_N^{2-6} \\ H_1^{3-6} & H_2^{3-6} & H_3^{3-6} & \cdot & \cdot & H_i^{3-6} & \cdot & H_N^{3-6} \\ H_1^{4-6} & H_2^{4-6} & H_3^{4-6} & \cdot & \cdot & H_i^{4-6} & \cdot & H_N^{4-6} \end{bmatrix}$$

Der untere Index gibt hier die Zahl des Messpunktes an, der erste obere Index die Reihe des Hermite-Polynoms, während der letzte obere Index die Ordnung des Hermite-Polynoms bezeichnet.

Mit

- N_1 als Anzahl der Durchbiegungsmesswerte w_i , $i=1, N_1$ bzw.
- N_2 als Anzahl der Winkelneigungsmesswerte w_i' , $i=1, N_2$ und
- N_3 als Anzahl der Krümmungsmesswerte w_i'' , $i=1, N_3$ folgt:

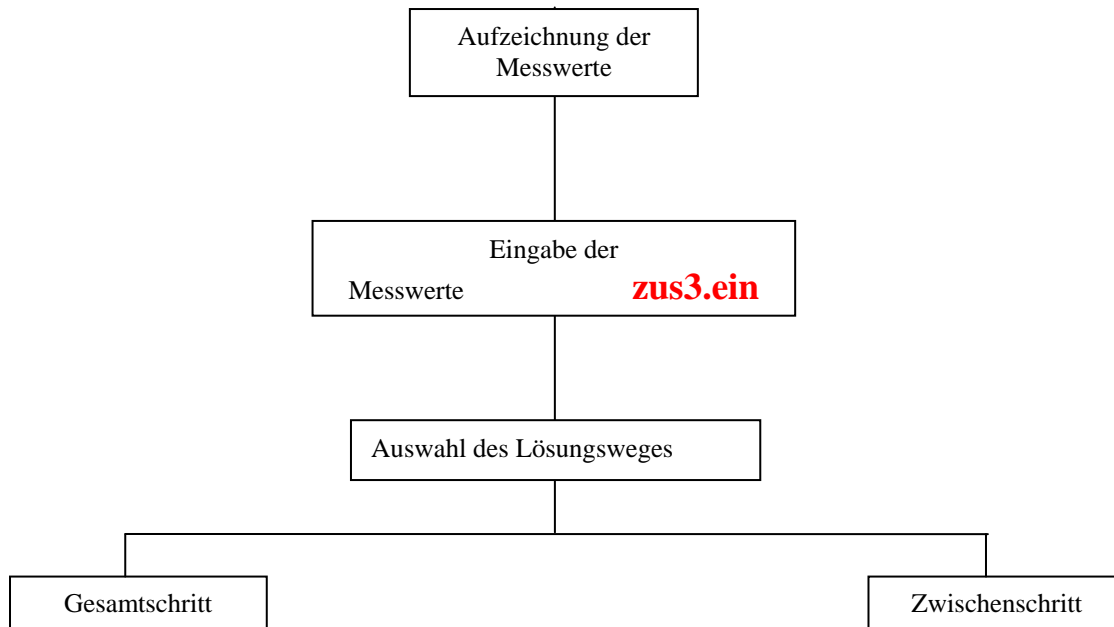
$$\mathbf{B}[\mathbf{w}] = 2 * \begin{bmatrix} \sum_i^{1, N_1} H_i^{1-6} H_i^{1-6} & \sum_i^{1, N_1} H_i^{1-6} H_i^{2-6} & \sum_i^{1, N_1} H_i^{1-6} H_i^{3-6} & \sum_i^{1, N_1} H_i^{1-6} H_i^{4-6} \\ \sum_i^{1, N_1} H_i^{2-6} H_i^{1-6} & \sum_i^{1, N_1} H_i^{2-6} H_i^{2-6} & \sum_i^{1, N_1} H_i^{2-6} H_i^{3-6} & \sum_i^{1, N_1} H_i^{2-6} H_i^{4-6} \\ \sum_i^{1, N_1} H_i^{3-6} H_i^{1-6} & \sum_i^{1, N_1} H_i^{3-6} H_i^{2-6} & \sum_i^{1, N_1} H_i^{3-6} H_i^{3-6} & \sum_i^{1, N_1} H_i^{3-6} H_i^{4-6} \\ \sum_i^{1, N_1} H_i^{4-6} H_i^{1-6} & \sum_i^{1, N_1} H_i^{4-6} H_i^{2-6} & \sum_i^{1, N_1} H_i^{4-6} H_i^{3-6} & \sum_i^{1, N_1} H_i^{4-6} H_i^{4-6} \end{bmatrix}$$

$$\mathbf{A}[\mathbf{w}] = 2 * \begin{bmatrix} \sum_i^{1, N_1} H_i^{1-6} * w_i \\ \sum_i^{1, N_1} H_i^{2-6} * w_i \\ \sum_i^{1, N_1} H_i^{3-6} * w_i \\ \sum_i^{1, N_1} H_i^{4-6} * w_i \end{bmatrix}$$

Entsprechend sind dann $\mathbf{B}[\mathbf{w}']$ und $\mathbf{A}[\mathbf{w}']$ sowie $\mathbf{B}[\mathbf{w}'']$ und $\mathbf{A}[\mathbf{w}'']$ zu bilden.

4 Lösungsweg

In Bild B-4-1 wird der prinzipielle Lösungsweg für alle Verfahren angegeben. Die dargestellten Lösungswege - das Gesamtschritt- bzw. Einzelschritt-Verfahren - werden anschließend mittels INTEL- Fortran- Compiler realisiert. Zwecks Vergleichbarkeit wird generell dieselbe Input-Datei verwendet.



- nur Abstandslösung
- mit Flächenausgleich

- nur Abstandslösung
- mit Flächenausgleich

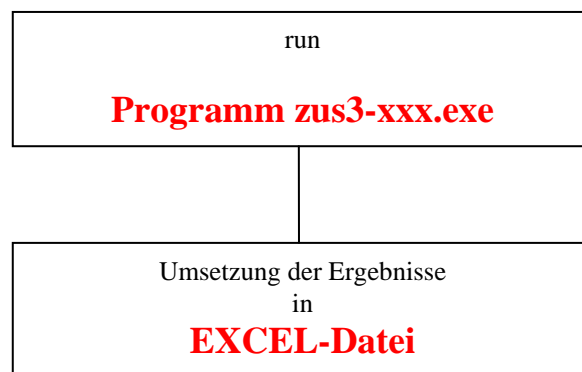
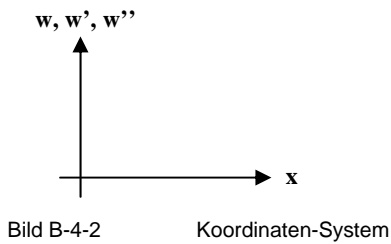


Bild B-4-1: Prinzipieller Lösungsweg

4.1 Programm-Input

Aus historischen Gründen wird dieser Input-File „zus3.ein“ bezeichnet. Zur Vorbereitung bzw. Generierung der Daten ist es erforderlich, das Positiv-Bild vorzugeben:

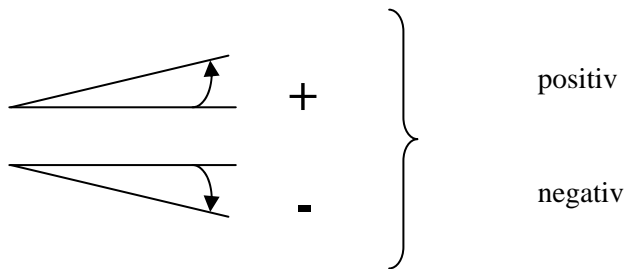


Damit sind die

1. Durchbiegungen w

nach oben positiv
nach unten negativ

2. Winkelneigungen w'



3. Krümmungen $w'' = \epsilon / z$

mit $z =$ Abstand von neutraler Achse
Zug = positiv
Druck = negativ

w'' dann positiv, falls ϵ_{unten} Zug mit z_{unten} positiv
 ϵ_{oben} Druck mit z_{oben} negativ

w'' dann negativ, falls ϵ_{unten} Druck mit z_{unten} positiv
 ϵ_{oben} Zug mit z_{oben} negativ

Diese Vorschrift ist bei der Eingabe der Messwerte in Tab. B-4-1 strikt zu beachten.

Nr:	Bezeichnung	Erläuterung	Format
1	'Text'	beliebiger Text	80A1
2	NFELD	NFELD Zahl der Felder bzw. Abschnitte (<10)	i10, F20.0
3	[kpolart(i), i=1,NFELD]	kpolart Polynom je Abschnitt =3 Polynom 3. Grades (Einzellast) =4 Polynom 4. Grades (Gleichlast)	5i10
4	[ksysw(i), i=1,NFELD - 1]	ksysw Funktionswert w je innerer Abschnitts grenze 0 nicht 1,2,3 Kopplung f, f', f''	5i10
5	[aalang(i), messw(i), messws(i) messd(i),mzzz(i), i=1,NFELD]	aalang Feldlänge messw Zahl der gemessenen Durchbiegungen je Feld / Abschnitt messws Zahl der gemessenen Winkelneigungen je Feld / Abschnitt messd Zahl der gemessenen Dehnungen je Feld / Abschnitt mzzz Zahl der Höhen z zur neutralen Linie (<10)	F20.0,5i10
	falls messw(i) ne 0: [xw(i,j), w(i,j), j=1,messw(i)]	xw Längenkoo x für gem. Durchbiegungen je Abschnitt w gemes. Durchbiegungen je Abschnitt (<50)	4F20.0
	falls messws(i) ne 0: [xws(i,j), ws(i,j), j=1,messws(i)]	xws Längenkoo x für gem. Winkelneigungen je Abschnitt ws gemes. Winkelneigungen je Abschnitt (<50)	4F20.0
	falls messd(i) ne 0: [xd(i,j), ebs(i,j), j=1,messd(i)]	xd Längenkoo x für gemessene Dehnungen je Abschnitt ebs*10^6 gemes. Dehnungen je Abschnitt (<50)	4F20.0
	[xz(i,j), z(i,j), j=1,mzzz(i)]	xz Längenkoo für z-Koordinaten je Abschnitt z Abstand von neutraler Linie (<50)	4F20.0
	Randbedingungen:		
	irandges	irandges Zahl aller Randbedingungen	8i10
	irandw, irandws, irandw2s	irandw Zahl der w -Bedingungen irandws Zahl der ws -Bedingungen irandw2s Zahl der w2s -Bedingungen	8i10
	falls irandw ne 0: [jrandw(ii), aaxxw(ii), ccw(ii), ii=1,irandw]	jrandw Feld-Nr. aaxxw bezog. Koordinate ξ ccw echte Durchbiegungen meist 0	i10,3F20.0
	falls irandws ne 0: [jrandws(ii), aaxxws(ii), ccws(ii), ii=1,irandws]	jrandws Feld-Nr. aaxxws bezog. Koordinate ξ ccws echte Winkelneigungen meist 0	i10,3F20.0
	falls irandw2s ne 0: [jrandw2s(ii), aaxxw2s(ii), ccw2s(ii), ii=1,irandw2s]	jrandw2s Feld-Nr. aaxxw2s bezog. Koordinate ξ ccw2s echte Krümmungen meist 0	i10,3F20.0

Tab. B-4-1: Eingabe-Werte in die Input-Datei „zus3.ein“

4.2 Quellcode zur Gesamtschritt-Lösung

Der Quellcode wird nur für den gängigsten Weg der Gesamtschritt-Lösung angegeben:

C Programm zus3-1-abstand.for

```
c 1 Polynom 4. Grades wie Balkenlinie fuer Gleichlast =4
c (5 Parameter)

c 1 Polynom 3. Grades wie Balkenlinie fuer Einzellast =3
c (4 Parameter)
```

```
C mit variablen Funktionsbedingungen bei den
C i n n e r e n Abschnitten

C Variation Messungsregression
C
C max. 10 Abschnitte bzw. Felder moeglich
```

```
=====
implicit real*8 (A-H,O-Z)

integer iafeld(100,7)
common/adressfeld/iafeld,izges
real*8 zzeil1(5),zzeil2(5)
common/besetzen1/zzeil1,wert1
common/besetzen2/zzeil2,wert2
common/indeex/index

real*8 A(300,300),B(300,2)
COMMON/gauss/ M,N,A,B
COMMON/hhh/xw,xws,xd,xz,aalang,messw,messws,messd,mzzz,
a w,ws,ebz,z,bb,ggs,gg,ww,wws,ee

COMMON/hhh1/fpol,isysw,ipolart
COMMON/hhh2/pl1,plr,ak2,ak2t
COMMON/hhh3/NFELD,MM1,M1
COMMON/hhh4/Feld
COMMON/hhh5/irandges,irandw,irandws,irandw2s
COMMON/hhh6/jrandw,jrandws,jrandw2s,
a aaxxw,aaxxws,aaxw2s,ccw,ccws,ccw2s

character*1 Feld(80)
real*8 bb(5,5),ggs(5,5),gg(5,5),ww(5),wws(5),ee(5)
real*8 xw(10,50),xws(10,50),xd(10,50),xz(10,50),
a w(10,50),ws(10,50),ebz(10,50),z(10,50)

real*8 aalang(10)
integer messw(20),messws(20),messd(20),mzzz(20),
a jrandw(20),jrandws(20),jrandw2s(20)
real*8 aaxxw(20),aaxxws(20),aaxw2s(20),
a ccw(20),ccws(20),ccw2s(20)

real*8 ak2(3,10),ak2t(10,3)
real*8 wneu(10,50,3),wstri(10,50,3),w2stri(10,50,3)
a,wsneu(10,50,3),ebzneu(10,50,3)
real*8 fpol(6,6),pl1(3,5),plr(3,5)
real*8 hhw(5),hhstri(5),hh2stri(5),eb2stri(5)

real*8 randwerte(5)

COMMON/hhh1a/ksysw,kpolart
integer ksysw(50),kpolart(50)

real*8 Ages(300,300)
COMMON/gauss1/ Ages

real*8 xx,zz
```

```
c Zuordnung der files:
c-----
character*80 pfadname,filename
character*160 gesamtname
character*1 vp(80),vf(80),vg(160)
equivalence(pfadname,vp)
equivalence(filename,vf)
equivalence(gesamtname,vg)

c-----
c zus3-alle.bslash ==> 52
open(Unit=52,File='zus3-alle.bslash',Status='OLD')
c-----
cpri
print*,'Pfadname='
read(52,5621) pfadname
5621 format(A80)
print*,'Pfadname'
pause

c zus3.ein ==> 55
c-----
filename='zus3.ein?'
cpri
```

```
print*,'Filename= ',filename
call nom(vp,vf,vg)
print*,'Gesamtname= ',gesamtname
pause
open(Unit=55,File=gesamtname,Status='OLD')

c zus3-1-abstand.aus ==> 66
c-----
filename='zus3-1-abstand.aus?'
cpri
print*,'Filename= ',filename
call nom(vp,vf,vg)
print*,'Gesamtname= ',gesamtname
pause
open(Unit=66,File=gesamtname,Status='UNKNOWN')

c zus3-1-abstand.out ==> 77
c-----
filename='zus3-1-abstand.out?'
cpri
print*,'Filename= ',filename
call nom(vp,vf,vg)
print*,'Gesamtname= ',gesamtname
pause
open(Unit=77,File=gesamtname,Status='UNKNOWN')
```

```
C Bezeichnungen
C-----
c M Zahl der DOF
c N Zahl der Lastfaelle max. = 2
c N=1

c NFELD Zahl der Felder bzw. Abschnitte
(<=10 !!!)

c kpolart(i),i=1,NFELD Polynom je Abschnitt

c ipolart Art der Ansatz-Funktionen
c =4 nur 1 Polynom 4. Grades
bis **4
c =3 nur 1 Polynom 3. Grades
bis **3

C ksysw(i),i=1,NFELD-1 Funktionswerte w an den i n n
e r e n Abschnittsgrenzen
C
c ksysw Einstellung fuer Bereich w = Durch-
biegungen
c =1 Zeile 1 beruecksichti-
gen =f
c =2 Zeilen 1+2 beruecksichti-
gen =f'
c =3 Zeilen 1+2+3 beruecksichti-
gen =f"

c aalang(i) Laenge Feld(i)

c messw(i) Zahl der gemess. Durchbiegungen w
(i,j)
c messws(i) Zahl der gemess. Winkel w'
(i,j)
c messd(i) Zahl der gemess. Dehnungen
ebs(i,j)
c mzzz(i) Zahl der Abstaende z von neutraler
Linie

c xw(i,j) Laengenkoo x fuer Durchbiegungen
c xws(i,j) Laengenkoo x fuer Winkel
c xd(i,j) Laengenkoo x fuer Dehnungen

c w(i,j) gemess. Durchbiegungen
c ws(i,j) gemess. Winkel
c ebs(i,j) gemess. Dehnungen * 10**(+6)

c xz(i,j) Laengenkoo x Hoehe z
c z(i,j) Hoehe z zur neutralen Linie

c irandges Gesamt-Zahl der Randbedingungen
c irandw Zahl der Rdbd fuer w
c irandws Zahl der Rdbd fuer w'
c irandw2s Zahl der Rdbd fuer w''

c mit irand-
ges=irandw+irandws+irandw2s

c w: i=1,irandw
c jrandw(i) Feld-Nr.
c aaxxw(i) Koordinate ksi
c ccw(i) Last-Bedingung meist 0.

c ws: i=1,irandws
c jrandws(i) Feld-Nr.
c aaxxws(i) Koordinate ksi
c ccws(i) Last-Bedingung meist 0.

c w2s: i=1,irandw2s
c jrandw2s(i) Feld-Nr.
c aaxxw2s(i) Koordinate ksi
c ccw2s(i) Last-Bedingung meist 0.

c Aufbau des Adressfeldes iafeld
c-----
c Index-Reihenfolge 1 alle Felder mit w
c-----
```

```

c   Index-Reihenfolge  2   alle Felder mit   w'
c-----
c   Index-Reihenfolge  3   alle Felder mit   w''
c-----
c-----
c   Index-Reihenfolge  4   innere Kopplung   w w' w''
Felder i,i+1
c   a) beide Felder w
falls nein
c   b) Feld i w // Feld i+1 w'
falls nein
c   c) Feld i w // Feld i+1 w''
falls nein
c   d) Feld i w' // Feld i+1 w''
c-----
c-----
c   Index-Reihenfolge  5   aeußere Kopplung w - w' -
Felder
c   Index-Reihenfolge  6   aeußere Kopplung w - w''-
Felder
c   Index-Reihenfolge  7   aeußere Kopplung w'- w''-
Felder
c-----
c-----
c   Index-Reihenfolge  8   alle Randbedingungen

c   Aufbau des Adressfeldes iafeld i=1,NFELD
c-----
c   iafeld(i,1)          Index-Nr.

c   iafeld(i,2)          bei Index-Wert 1,2,3      Feld-
Nr.
c   iafeld(i,2)          bei Index-Wert 5,6,7,8    Feld-
Nr.
c   iafeld(i,2)          bei Index-Wert 4          Feld
links

c   iafeld(i,3)          bei Index-Wert 1,2,3      Feld-
Nr.
c   iafeld(i,3)          bei Index-Wert 5,6,7,8    Feld-
Nr.
c   iafeld(i,3)          bei Index-Wert 4          Feld
rechts

c   iafeld(i,4)          jeweils Anfangszeile
c   iafeld(i,5)          jeweils Endzeile

c   iafeld(i,6)          Differenz der Adressen idel fuer
innere Kopplung

C   Parameter der Polynome
c-----
igewicht=0

c   einlesen+drucken der Input-Parameter
cpri
print*, ' vor EINAUS'
pause
CALL EINAUS

cpri
print*, ' nach EINAUS'
pause

cpri
print*, 'irandges =', irandges
print*, ' irandw =', irandw
print*, ' irandws =', irandws
print*, ' irandw2w =', irandw2s

C   Speicher-Kontrolle
C=====
cpri
print*, 'NFELD=', NFELD
if (NFELD.gt.10)
aprint*, ' stop da NFELD gt 10 !!!'
if (NFELD.gt.10) stop

nw =0
nws=0
nd =0
do 87 i=1,NFELD
if (messw(i).gt.nw) nw = messw (i)
if (messws(i).gt.nws) nws= messws(i)
if (messd(i).gt.nd) nd = messd (i)
87 continue

cpri
print*, 'nw=',nw, ' nws=',nws, ' nd=',nd
if (nw.gt.50)
aprint*, ' stop da nw gt 50 !!!'
if (nw.gt.50) stop

if (nws.gt.50)
aprint*, ' stop da nws gt 50 !!!'
if (nws.gt.50) stop

if (nd.gt.50)
aprint*, ' stop da nd gt 50 !!!'
if (nd.gt.50) stop

C   Berechnung des Adressfeldes iafeld
c=====

```

```

c   Index 1: alle Felder mit w
c-----
c   iz          Position im Adressenfeld

iz=0
izeile_a=0
izeile_e=0

jwa=0
jwe=0
do 5931 i=1,NFELD
if (messw(i).eq.0) go to 5931

iz=iz+1
iafeld(iz,1)=1

if (kpolart(i).eq.4) izeile_a=izeile_e + 1
if (kpolart(i).eq.4) izeile_e=izeile_e + 5

if (kpolart(i).eq.3) izeile_a=izeile_e + 1
if (kpolart(i).eq.3) izeile_e=izeile_e + 4

iafeld(iz,2)=i
iafeld(iz,3)=i

iafeld(iz,4)=izeile_a
iafeld(iz,5)=izeile_e

5931 continue
izges=iz
MMw =izeile_e

if (izges.ne.0) jwe = izges
if (izges.ne.0) jwa = 1

if (izges.eq.0) jwe = 0
if (izges.eq.0) jwa = 0

cpri
print*, 'index=1 iafeld'
print*, 'izges=', izges
print*, 'jwa =', jwa, ' jwe =', jwe
print*, 'MMw =', MMw

c   Index 2: alle Felder mit w'
c-----
jwsa=0
jwse=0
do 5932 i=1,NFELD
if (messws(i).eq.0) go to 5932

iz=iz+1
iafeld(iz,1)=2

if (kpolart(i).eq.4) izeile_a=izeile_e + 1
if (kpolart(i).eq.4) izeile_e=izeile_e + 5

if (kpolart(i).eq.3) izeile_a=izeile_e + 1
if (kpolart(i).eq.3) izeile_e=izeile_e + 4

iafeld(iz,2)=i
iafeld(iz,3)=i

iafeld(iz,4)=izeile_a
iafeld(iz,5)=izeile_e

5932 continue
izges=iz
MMws =izeile_e

if (izges.ne.jwe) jwse = izges
if (izges.ne.jwe) jwsa = jwe+1

if (izges.eq.jwe) jwse = jwe
if (izges.eq.jwe) jwsa = jwe

cpri
cpri
print*, 'index=2 iafeld'
print*, 'izges=', izges
print*, 'jwsa =', jwsa, ' jwse =', jwse
print*, 'MMws =', MMws

c   Index 3: alle Felder mit w''
c-----
jw2sa=0
jw2se=0
do 5933 i=1,NFELD
if (messd(i).eq.0) go to 5933

iz=iz+1
iafeld(iz,1)=3

if (kpolart(i).eq.4) izeile_a=izeile_e + 1
if (kpolart(i).eq.4) izeile_e=izeile_e + 5

if (kpolart(i).eq.3) izeile_a=izeile_e + 1
if (kpolart(i).eq.3) izeile_e=izeile_e + 4

iafeld(iz,2)=i
iafeld(iz,3)=i

iafeld(iz,4)=izeile_a
iafeld(iz,5)=izeile_e

5933 continue
izges=iz
MMw2s =izeile_e

if (izges.ne.jwse) jw2se = izges
if (izges.ne.jwse) jw2sa = jwse+1

```

```

        if(izges.eq.jwse)   jw2se = jwse
        if(izges.eq.jwse)   jw2sa = jwse
cpri
    print*, 'index=3   iafeld'
    print*, 'izges=', izges
    print*, 'jw2sa=', jw2sa, '   jw2se=', jw2se
    print*, 'MMw2s =', MMw2s

    print*, ' '
    print*, 'iafeld   iz=1,jw2se'
    do 24 jjj=1,jw2se
    print 25, (iafeld(jjj,kkjjm),kkjjm=1,5)
25   format(7I10)
24   continue

        if(jw2se.eq.0) print*, 'jw2se  NULL keine Besetzung ==>
stop'
        if(jw2se.eq.0) stop

cpri
    print*, ' '
    print*, 'index=4   iafeld   izges=', izges
c   Index 4: innere Kopplung
c-----
c   innere Kopplung der   w-Felder
c
c   nur 1 Feld
    if(NFELD.eq.1) go to 5801

    do 4334 i=1,NFELD-1
    ifind=0
c   Feld links   i   Feld rechts   i+1
    i1=i+1
cpri
    print*, 'i=', i, '   i1=', i1
c
c   1.   suchen   Feld   i
    do 4335 kk=1,jw2se
    if(iafeld(kk,2).eq.i )           go to 4336
4335 continue
c   i   nicht gefunden
cpri
    print*, 'Feld i =', i, '   nicht gefunden'
    go to 4334

4336 continue
c   i   gefunden
    ideli =iafeld(kk,4)

c
c   2.   suchen   Feld   i+1
    do 4345 ll=1,jw2se
    if(iafeld(ll,2).eq.i1)           go to 4346
4345 continue
c   i+1   nicht gefunden
cpri
    print*, 'Feld i+1=', i1, '   nicht gefunden'
    go to 4334

4346 continue
c   i+1   gefunden
    ideli1=iafeld(ll,4)

c
c   eintragen in   iafeld
    iz=iz+1
    iafeld(iz,1)=4

    iafeld(iz,2)=i
    iafeld(iz,3)=i+1

    jend=ksysw(i)

    izeile_a=izeile_e + 1
    izeile_e=izeile_e + jend

    iafeld(iz,4)=izeile_a
    iafeld(iz,5)=izeile_e

c
c   Differenz idel
    idel=ideli1 - ideli
    iafeld(iz,6)=idel

4334 continue
    izges=iz

cpri
c   print*, ' '
c   print*, 'iafeld   iz=jw2se+1,izges'
c   do 124 jjj=1,jw2se+1,izges
c   print 25, (iafeld(jjj,kkjjm),kkjjm=1,6)
c   cff 124 continue

5801 continue

c
c   Index 5: aeussere Kopplung der   w - w'-Felder
c-----
c   gesuchte Parameter sollen gleich sein !
    j5a1=iz

    do 5937 i=1,NFELD
c   Abfrage ob keine Kopplung ?
    if(messw(i).eq.0.or.messws(i).eq.0) go to 5937

    iz=iz+1

```

```

c      ==> wg. index=4 u. 6   keine Kopplung fuer Feld i (index
7)
      if (i7.eq. 1.and.i5.eq. 1.and.i6.eq. 1)      go to 9939

c      jetzt besetzen index=7

      iz=iz+1
      iafeld(iz,1)=7

      iafeld(iz,2)=i
      iafeld(iz,3)=i

      if (kpolart(i).eq.3)      jend=4
      if (kpolart(i).eq.4)      jend=5

      izeile_a=izeile_e + 1
      izeile_e=izeile_e + jend

      iafeld(iz,4)=izeile_a
      iafeld(iz,5)=izeile_e

9939 continue
      izges=iz

c      Index 8: alle Randbedingungen
c-----
      if (irandges.eq.0)      go to 5804

c      Randbedingungen w
      if (irandw.eq.0) go to 5940
      do 5941 i=1,irandw
      iz=iz+1
      iafeld(iz,1)=8

c      Feld-Nr.
      iafeld(iz,2)=jrandw(i)
      iafeld(iz,3)=jrandw(i)

      jend=1

      izeile_a=izeile_e + 1
      izeile_e=izeile_e + jend

      iafeld(iz,4)=izeile_a
      iafeld(iz,5)=izeile_e

5941 continue
5940 continue

c      Randbedingungen ws
      if (irandws.eq.0) go to 5942
      do 5943 i=1,irandws
      iz=iz+1
      iafeld(iz,1)=8

c      Feld-Nr.
      iafeld(iz,2)=jrandws(i)
      iafeld(iz,3)=jrandws(i)

      jend=1

      izeile_a=izeile_e + 1
      izeile_e=izeile_e + jend

      iafeld(iz,4)=izeile_a
      iafeld(iz,5)=izeile_e

5943 continue
5942 continue

c      Randbedingungen w"
      if (irandw2s.eq.0) go to 5944
      do 5945 i=1,irandw2s
      iz=iz+1
      iafeld(iz,1)=8

c      Feld-Nr.
      iafeld(iz,2)=jrandw2s(i)
      iafeld(iz,3)=jrandw2s(i)

      jend=1

      izeile_a=izeile_e + 1
      izeile_e=izeile_e + jend

      iafeld(iz,4)=izeile_a
      iafeld(iz,5)=izeile_e

5945 continue
5944 continue

5804 continue

c      Zeilengesamtzahl izges
      izges=iz

c      Ausdruck Adressenfeld   iafeld
      write(66,256) 1,izges
256 format(//,' Adressenfeld   iafeld von ',I1,' bis ',I4,/)
      write(66,257)
257 format(' I   iafeld(I,1) iafeld(I,2) iafeld(I,3)
      aiafeld(I,4) iafeld(I,5) iafeld(I,6) iafeld(I,7) ')
      do 258 i=1,izges
      write(66,259) I,iafeld(I,1),iafeld(I,2),iafeld(I,3)
      a      ,iafeld(I,4),iafeld(I,5),iafeld(I,6)

b      ,iafeld(I,7)

259 format( I3,5x,I3,10x,I3,10x,I3,10x,I3,10x,I3
a      ,10x,I3,10x,I3)
258 continue

c      berechnen+drucken   iafeld
cpri
      print*,' drucken   iafeld'
      print*,' izges      =',izges
      print*,' izeile_e    =',izeile_e

C      Berechnung M      Zahl der DOF
C      Matrix-Parameter
c-----
C      Last-Matrix B zunaechst generell
N=1

C      Gesamt-Matrix A
M=izeile_e

cpri
      print*,' Zahl der DOF   M =',M
      if (M.gt.300)
      aprint*,' M gt 300 ==> stop !!!'
      if (M.gt.300)      stop

      pause

      if (igewicht.eq.0) last=1

C=====
C      Aufbau der Gesamtmatrizen   A   B
C=====
C      INDEX=1
C=====
cpri
      print*,' '
      print*,' '
      print*,' INDEX=1'

      jstart1=0
      izeil=0

      MMw   =0
      MMws  =0
      MMw2s =0

      kzw=0
      do 1000 k=1,izges
c      Abfrage nach Index=1      w
      if (iafeld(k,1).ne.1)      go to 1000
      kzw=kzw+1

c      alle w-Felder
c-----
c      jstart1      letzte Zeile   fuer Matrizen bb   u.   ww
je Feld w
c      (Diagonale)

cpri
      print*,' Matrizen bb      ww'
cv      print*,' Feld-Nr.   iafeld(k,2)=' ,iafeld(k,2),' Grund +
Last'

c      Feld-Nr.   iafeld(k,2)
i=iafeld(k,2)

c      bestimmen ipolart
c-----
      ipolart=kpolart(i)

c      Parameter setzen
      if (ipolart.eq.4) jende=5
      if (ipolart.eq.3) jende=4

c      vorher jeweils Null setzen   ww(5)
c      bb(5,5)
c-----
      do 1091 kj=1,5
c      Last-Matrizen
      ww (kj) =0.
      do 1092 km=1,5
c      Grund-Matrizen
      bb (kj,km)=0.
1092 continue
1091 continue

C      Durchbiegungen
C      Aufbau der Lastmatrix ww
c-----
c      Last aus w:      ww
      call hww(i)

C      Aufbau der Grundmatrix bb
c-----
c      Grund aus w:      bb

```

```

call hbxxxw(i)
C Besetzen der Gesamt-Matrix A
c aus Grund-Matrizen bb
C Besetzen der Gesamt-Matrix B
c aus Last-Matrizen ww
c-----
cpri print*, 'besetzen A B i=',i
cv print*, ' '
cv print*, ' '
cv print*, 'INDEX=1'
cv print*, ' jstart1=',jstart1
c Durchbiegungen w
c-----
do 1004 kk=1,jende
kkk=jstart1+kk
c=====
c Faktor 2 fuer Grund+Lastmatrix
B(kkk,1)=2.*(ww(kk))
c=====
1004 continue
do 1008 ll=1,jende
lll=jstart1+ll
do 1009 nn=1,jende
nnn=jstart1+nn
cpri
cc print*, 'i lll nnn wert=',i, ' ',lll,nnn, '
',A(lll,nnn)
c=====
c Faktor 2 fuer Grund+Lastmatrix
A(lll,nnn)=2.*(bb(ll,nn))
c=====
1009 continue
1008 continue
jjstart1=kkk
cpri
print*, ' besetzen A B i=',i
print*, ' jjstart1=',jjstart1
c neue letzte Diagonale: jstart1
c-----
c neu fuer Feld(i+1)
jstart1=jjstart1
c-----
1000 continue
cpri
print*, 'jstart1=',jstart1
MMw =jstart1
c keine w kzw=0 ==> INDEX=2
if(kzw.eq.0) go to 9971
9971 continue
C INDEX=2
C=====
cpri
print*, ' '
print*, ' '
print*, 'INDEX=2'
kzws=0
do 2000 k=1,izges
c Abfrage nach Index=2 w' go to 2000
if(iafeld(k,1).ne.2)
kzws=kzws+1
c alle w'-Felder
c-----
c jstart1 letzte Zeile fuer Matrizen ggs u. wws
je Feld w'
c (Diagonale)
cpri
cv print*, ' '
cv print*, ' '
cv print*, 'INDEX=2'
cv print*, 'MMw =',MMw
cv print*, ' Matrizen ggs wws'
cv print*, ' Feld-Nr. iafield(k,2)=' iafield(k,2), ' Grund +
Last'
c Feld-Nr. iafield(k,2)
i=iafield(k,2)
c bestimmen ipolart
c-----
ipolart=kpolart(i)
c Parameter setzen
if(ipolart.eq.4) jende=5
if(ipolart.eq.3) jende=4
c vorher jeweils Null setzen wws(5)
c ggs(5,5)
c-----
do 2091 kj=1,5
Last-Matrizen
wws(kj) =0.
do 2092 km=1,5
Grund-Matrizen
ggs(kj,km)=0.
2092 continue
2091 continue
C Winkelneigungen
C Aufbau der Lastmatrix wws
c-----
c Last aus w': wws
call hwsw1(i)
C Aufbau der Grundmatrix ggs
c-----
c Grund aus w': ggs
call hbxxxwsw1(i)
c090708
itest=0
if(itest.eq.1) ggs(1,1)=1.
if(itest.eq.1) go to 3331
c bei Last 1. Wert bei Grund 1. Zeile
c-----
index=2
call besetzen_zeile1_ws(i)
c besetzen 1. Zeile Grundmatrix ggs
c-----
do 2262 jj=1,jende
ggs(1,jj)=zzei11(jj)
2262 continue
c besetzen 1. Zeile Lastmatrix wws
c-----
wws(1)=wert1
cpri
cv print*, 'zzei11'
cv print 31, (zzei11(klmn), klmn=1,5)
cv 31 format (5E15.6)
cv print*, 'wert1=',wert1
cv
3331 continue
c090708
C Besetzen der Gesamt-Matrix A
c aus Grund-Matrizen ggs
C Besetzen der Gesamt-Matrix B
c aus Last-Matrizen wws
c-----
cpri
print*, ' besetzen A B i=',i
print*, ' jstart1=',jstart1
c Winkelneigungen w'
c-----
do 2004 kk=1,jende
kkk=jstart1+kk
c=====
c Faktor 2 fuer Grund+Lastmatrix
B(kkk,1)=2.*(wws(kk))
c=====
2004 continue
do 2008 ll=1,jende
lll=jstart1+ll
do 2009 nn=1,jende
nnn=jstart1+nn
cpri
cc print*, 'i lll nnn wert=',i, ' ',lll,nnn, '
',A(lll,nnn)
c=====
c Faktor 2 fuer Grund+Lastmatrix
A(lll,nnn)=2.*(ggs(ll,nn))
c=====
2009 continue
2008 continue
jjstart1=kkk
cpri
print*, ' besetzen A B i=',i
print*, ' jjstart1=',jjstart1
c neue letzte Diagonale: jstart1
c-----
c neu fuer Feld(i+1)
jstart1=jjstart1
c-----

```

```

2000 continue
MMws =jstart1

c keine w' kzs=0 ==> INDEX=3
if(kzs.eq.0) go to 9972

9972 continue

C INDEX=3
C=====
cpri
print*, ' '
print*, ' '
print*, 'INDEX=3'

kzw2s=0
do 3000 k=1,izges
Abfrage nach Index=3 w'' go to 3000
if(iafeld(k,1).ne.3)
kzw2s=kzw2s+1

c alle w''-Felder
c-----
c jstart1 letzte Zeile fuer Matrizen gg u. ee
je Feld w''
c (Diagonale)

cpri
cv print*, ' '
cv print*, ' '
cv print*, 'INDEX=3'
print*, ' Matrizen gg ee'
cv print*, ' Feld-Nr. iafeld(k,2)=' iafeld(k,2), ' Grund +
Last'

c Feld-Nr. iafeld(k,2)
i=iafeld(k,2)

c bestimmen ipolart
c-----
ipolart=kpolart(i)

c Parameter setzen
if(ipolart.eq.4) jende=5
if(ipolart.eq.3) jende=4

c vorher jeweils Null setzen ee(5)
c gg(5,5)
c-----

c do 3091 kj=1,5
Last-Matrizen
ee(kj) =0.
do 3092 km=1,5
Grund-Matrizen
gg(kj,km)=0.
3092 continue
3091 continue

C Dehnungen
C Aufbau der Lastmatrix ee
c-----
c Last aus w'': ee
call hwel(i)

C Aufbau der Grundmatrix gg
c-----
c Grund aus w'': gg
call hbxxxbs1(i)

c bei Last 1. Wert bei Grund 1. Zeile
c bei Last 2. Wert bei Grund 2. Zeile
c-----
index=3
call besetzen_zeilelu2_w2s(i)

c besetzen 1.+2. Zeile Grundmatrix gg
c-----
do 3262 jj=1,jende
gg(1,jj)=zzeil1(jj)
gg(2,jj)=zzeil2(jj)
3262 continue

c besetzen 1.+2. Zeile Lastmatrix ee
c-----
ee(1)=wert1
ee(2)=wert2

C Besetzen der Gesamt-Matrix A
c aus Grund-Matrizen gg

C Besetzen der Gesamt-Matrix B
c aus Last-Matrizen ee

c-----
cpri
print*, ' besetzen A B i=' i
print*, ' jstart1=' jstart1

c Dehnungen w''
c-----
do 3004 kk=1,jende
kkk=jstart1+kk

```

```

c=====
c Faktor 2 fuer Grund+Lastmatrix
B(kkk,1)=2.*(ee(kk))
c=====
3004 continue

do 3008 ll=1,jende
lll=jstart1+ll
do 3009 nn=1,jende
nnn=jstart1+nn

cpri
cc print*, 'i lll nnn wert=' i, ' ',lll,nnn, '
',A(lll,nnn)

c=====
c Faktor 2 fuer Grund+Lastmatrix
A(lll,nnn)=2.*(gg(ll,nn))
c=====
3009 continue
3008 continue

jjstart1=kkk

cpri
print*, ' besetzen A B i=' i
print*, ' jjstart1=' jjstart1

c neue letzte Diagonale: jstart1
c-----
c neu fuer Feld(i+1)
jstart1=jjstart1
c-----

3000 continue
MMw2s =jstart1

c keine w' kzw2s=0 ==> INDEX=4
if(kzw2s.eq.0) go to 9973

9973 continue

C INDEX=4
C=====
cpri
print*, ' '
print*, ' '
print*, 'INDEX=4'
print*, 'jstart1 =',jstart1

koppinz1=jstart1
izeil=jstart1

c do 1010 k=1,izges
Abfrage nach Index=4 innere Kopplung w
if(iafeld(k,1).ne.4) go to 1010

c innere Kopplung der Felder k,k+1
c-----
c gefunden
ifei =iafeld(k,2) !Feld i
ifeil =iafeld(k,3) !Feld i
idel =iafeld(k,6) !Differenzadresse

cpri
print*, 'k=' k, ' ifei =' ifei, ' ifeil =' ifeil

c jstart1
c (Diagonale)

c koppinz1 Startzeile -1 jew. am Ende
c koppins1 Startspalte-1 jew. am Anfang

c neue koppinsli
c-----
do 1310 kj=1,izges
if(ifei .eq.iafeld(kj,2)) go to 1311
1310 continue
1311 continue
koppinsli =iafeld(kj,4) - 1

c neue koppinsli1
c-----
do 1410 kj=1,izges
if(ifeil.eq.iafeld(kj,2)) go to 1411
1410 continue
1411 continue
koppinsli1=iafeld(kj,4) - 1

cpri
cv print*, ' '
cv print*, ' '
cv print*, 'INDEX=4'
print*, ' innere Kopplung w'
cv print*, ' Feld links iafeld(k,2)=' iafeld(k,2)
cv print*, ' Feld rechts iafeld(k,3)=' iafeld(k,3)
cv print*, 'koppinz1 =' koppinz1
cv print*, 'koppinsli=' koppinsli, ' kop-
pinsli1=' koppinsli1

c Feld links iafeld(k,2) ==> i
i=iafeld(k,2)

c Feld rechts = i+1

```



```

C      Parameter-Werte
C-----
c      bei ipolart=4      5 Parameter
c      linkes Schnittufer pl1(5,5)
c      rechtes Schnittufer plr(5,5)

c      bei ipolart=3      4 Parameter
c      linkes Schnittufer pl1(4,4)
c      rechtes Schnittufer plr(4,4)

cpri
cv      print*, 'besetzen Kopplung ak2      i=',i
cv      print*, '                          kop-
pinz1=',koppinz1

c      koppeln Abschnitt i u. i+1
c      bestimmen ipolart
      ipolartli=kpolart(i )
      ipolartre=kpolart(i+1)

c      Parameter setzen
      if(ipolartli.eq.4) jendeli=5
      if(ipolartre.eq.4) jendere=5

      if(ipolartli.eq.3) jendeli=4
      if(ipolartre.eq.3) jendere=4

c      beide Felder jeweils Durchbiegung / Dehnung => ak2 ak2t
C-----
C      KOPPLUNG der jew. Abschnitte/Felder
c      Koppelmatrizen ak2      bzw. ak2t

      CALL KOPPEL(i)

c      jetzt Matrizen ak2 bzw. ak2t bekannt
C-----
c      Anfangs- bzw. Endzeile
      j1 =1
      jend=ksysw(i)

cpri
cv      if(ifei.le.2)
cv      aprint*, 'jendeli=',jendeli, ' jendere',jendere

      jzz=0
c      einfüegen ak2      ak2t
      do 1111 jz=j1,jend
      jzz=jzz+1
      do 1121 js=1,jendeli+jendere
      izeil=koppinz1+jzz

      if(js.le.jendeli) koppins1=koppinsli
      if(js.gt.jendeli) koppins1=koppinsli1 - jendeli

      ispal=koppins1+jz

      A(izeil,ispal)=ak2(jz,js)
      A(ispal,izeil)=A(izeil,ispal)

      1121 continue
      1111 continue

c      neue koppinz1
C-----
      koppinz1=koppinz1 + jend

      1010 continue

c      Ende interne Kopplung alle w
      jstart1 = izeil

cpri
      print*, 'nach interner Kopplung      jstart1=',jstart1
C=====
cv      pause

C      INDEX=5
C=====
cpri
      print*, ' '
      print*, ' '
      print*, 'INDEX=5'
      print*, 'MMw2s=',MMw2s, ' izges=',izges
      print*, 'Kopplung w ws'

      do 4010 k=1,izges
c      Abfrage nach Index=5      aeussere Kopplung w mit w'
      if(iafeld(k,1).ne.5)      go to 4010

c      hier w und w' gefunden: Feld-Nr.      iafeld(k,2 bzw. 3)
      ifeldnr = iafeld(k,2)

cpri
      print*, 'ifeldnr=',ifeldnr

c      suchen Anfangszeile Feld-Nr.      bei w + Feld-Position
      nw=0
      do 1620 kkjj=1,izges
      if(iafeld(kkjj,1).eq.1) nw=nw+1
      if(iafeld(kkjj,1).eq.1.and.iafeld(kkjj,2).eq.ifeldnr)
      a      go to 1621
      1620 continue
c      nicht gefunden ==> Fehler in iafeld ??
cpri
      print*, 'k=',k, '      kkjj=',kkjj

      print*, 'iafeld(k,1)=',iafeld(k,1), '
      iafeld(k,2)=',iafeld(k,2)
      print*, 'ifeldnr=',ifeldnr, ' nicht gefunden Fehler in
      iafeld ???'
      stop

      1621 continue

c      Anfangszeile w      iazz_w      nw gibt Position bei w an
      iazz_w = iafeld(kkjj,4)

cpri
cv      print*, 'kkjj=',kkjj, ' iazz_w =',iazz_w, ' nw =',nw

c      suchen Anfangszeile Feld-Nr.      bei w+ Feld-Position
      nws=0
      do 1622 nnjj=1,izges
      if(iafeld(nnjj,1).eq.2) nws=nws+1
      if(iafeld(nnjj,1).eq.2.and.iafeld(nnjj,2).eq.ifeldnr)
      a      go to 1623
      1622 continue
c      nicht gefunden ==> Fehler in iafeld ??
cpri
      print*, 'k=',k, '      nnjj=',nnjj
      print*, 'iafeld(k,1)=',iafeld(k,1), '
      iafeld(k,2)=',iafeld(k,2)
      print*, 'ifeldnr=',ifeldnr, ' nicht gefunden Fehler in
      iafeld ???'
      stop

      1623 continue

c      Anfangszeile ws      iazz_ws      nws gibt Position bei ws
      an
      iazz_ws = iafeld(nnjj,4)

cpri
cv      print*, 'nnjj=',nnjj, ' iazz_ws=',iazz_ws, ' nws=',nws

c      Differenz idel
      idel = iazz_ws - iazz_w

cpri
cv      print*, 'kkjj=',kkjj, ' idel =',idel

c      Besetzen A-Matrix
      if(kpolart(ifeldnr).eq.3) jend=4
      if(kpolart(ifeldnr).eq.4) jend=5

      do 4111 jz=1,jend
      izeil=jstart1 + jz

      ispal1=(nw-1)*jend + jz
      ispal =ispal1

cpri
cv      print*, 'jz=',jz, ' izeil=',izeil, ' ispal1=',ispal

      A(izeil,ispal)= 1.
      A(ispal,izeil)=A(izeil,ispal)

      ispal2=ispal1 + idel
      ispal =ispal2

cpri
cv      print*, 'jz=',jz, ' izeil=',izeil, ' ispal2=',ispal

      A(izeil,ispal)=-1.
      A(ispal,izeil)=A(izeil,ispal)

      4111 continue

c      neues jstart1
      jstart1=jstart1+jend

      4010 continue

C      INDEX=6
C=====
cpri
      print*, ' '
      print*, ' '
      print*, 'INDEX=6'
      print*, 'MMw2s=',MMw2s, ' izges=',izges
      print*, 'Kopplung w w2s'

      do 5010 k=1,izges
c      Abfrage nach Index=6      aeussere Kopplung w mit w"
      if(iafeld(k,1).ne.6)      go to 5010

c      hier w und w" gefunden: Feld-Nr.      iafeld(k,2 bzw. 3)
      ifeldnr = iafeld(k,2)

cpri
      print*, 'ifeldnr=',ifeldnr

c      suchen Anfangszeile Feld-Nr.      bei w + Feld-Position
      nw=0
      do 1630 kkjj=1,izges
      if(iafeld(kkjj,1).eq.1) nw=nw+1
      if(iafeld(kkjj,1).eq.1.and.iafeld(kkjj,2).eq.ifeldnr)
      a      go to 1631
      1630 continue
c      nicht gefunden ==> Fehler in iafeld ??
cpri
      print*, ' '
      print*, ' '
      print*, 'INDEX=6'
      print*, 'Kopplung w w2s'
      print*, 'k=',k
      print*, 'iafeld(k,1)=',iafeld(k,1), ' ia-
      feld(k,2)=',iafeld(k,2)
      print*, 'ifeldnr=',ifeldnr, ' nicht gefunden Fehler in
      iafeld ???'

```

```

stop
1631 continue

c  Anfangszeile w  iazz_w  nw gibt Position bei w an
   iazz_w  = iafeld(kkjj,4)
cpri
cv  print*,'kkjj=',kkjj,' iazz_w = ',iazz_w,' nw =
',nw

c  suchen Anfangszeile  Feld-Nr.  bei w" + Feld-Position
nw2s=0
do 1632 nnjj=1,izges
if (iafeld(nnjj,1).eq.3) nw2s=nw2s+1
if (iafeld(nnjj,1).eq.3.and.iafeld(nnjj,2).eq.ifeldnr)
a
1632 continue
cpri
print*,'k=',k
print*,'iafeld(k,1)=' ,iafeld(k,1),'
iafeld(k,2)=' ,iafeld(k,2)
print*,'ifeldnr=' ,ifeldnr,' nicht gefunden Fehler in
iafeld ???'
stop

1633 continue

c  Anfangszeile w2s  iazz_w2s  nw2s gibt Position bei
w" an
   iazz_w2s  = iafeld(nnjj,4)
cpri
cv  print*,'nnjj=',nnjj,' iazz_w2s =',iazz_w2s,' nw2s
=',nw2s

c  Differenz idel
idel = iazz_w2s - iazz_w

c  Besetzen A-Matrix
if (kpolar(ifeldnr).eq.3)  jend=4
if (kpolar(ifeldnr).eq.4)  jend=5

do 5211 jz=1,jend
izeil=jstart1 + jz

ispall=(nw-1)*jend + jz
ispal =ispall

cpri
cv  print*,'jz=' ,jz,' izeil=' ,izeil,'  ispal1=' ,ispal

A(izeil,ispal)= 1.
A(ispal,izeil)=A(izeil,ispal)

ispal2=ispall + idel
ispal =ispal2

cpri
cv  print*,'jz=' ,jz,' izeil=' ,izeil,'  ispal2=' ,ispal

A(izeil,ispal)=-1.
A(ispal,izeil)=A(izeil,ispal)

5211 continue

c  neues jstart1
jstart1=jstart1+jend

5010 continue

C  INDEX=7
C=====
cpri
print*,' '
print*,' '
print*,'INDEX=7'
print*,'MMw2s=',MMw2s,' izges=' ,izges
print*,'Kopplung ws  w2s'

do 6010 k=1,izges
Abfrage nach Index=7  aeussere Kopplung w' mit w"
if (iafeld(k,1).ne.7)  go to 6010

c  hier w' und w" gefunden: Feld-Nr.  iafeld(k,2 bzw. 3)
ifeldnr  = iafeld(k,2)
cpri
print*,'ifeldnr=' ,ifeldnr

c  suchen Anfangszeile  Feld-Nr.  bei w' + Feld-Position
nws=0
do 1640 kkjj=1,izges
if (iafeld(kkjj,1).eq.2) nws=nws+1
if (iafeld(kkjj,1).eq.2.and.iafeld(kkjj,2).eq.ifeldnr)
a
1640 continue
c  nicht gefunden ==> Fehler in iafeld ??
cpri
print*,' '
print*,' '
print*,'INDEX=7'
print*,'Kopplung ws  w2s'
print*,'k=',k
print*,'iafeld(k,1)=' ,iafeld(k,1),' ia-
feld(k,2)=' ,iafeld(k,2)
print*,'ifeldnr=' ,ifeldnr,' nicht gefunden Fehler in
iafeld ???'
stop

1641 continue

c  Anfangszeile w'  iazz_ws  nws gibt Position bei ws
an
   iazz_ws  = iafeld(kkjj,4)
cpri
cv  print*,'kkjj=',kkjj,' iazz_ws = ',iazz_ws,' nws
=',nws

c  suchen Anfangszeile  Feld-Nr.  bei w" + Feld-Position
nw2s=0
do 1642 nnjj=1,izges
if (iafeld(nnjj,1).eq.3) nw2s=nw2s+1
if (iafeld(nnjj,1).eq.3.and.iafeld(nnjj,2).eq.ifeldnr)
a
1642 continue
cpri
print*,'k=',k
print*,'iafeld(k,1)=' ,iafeld(k,1),'
iafeld(k,2)=' ,iafeld(k,2)
print*,'ifeldnr=' ,ifeldnr,' nicht gefunden Fehler in
iafeld ???'
stop

1643 continue

c  Anfangszeile w2s  iazz_w2s  nw2s gibt Position bei
w2s an
   iazz_w2s  = iafeld(nnjj,4)
cpri
cv  print*,'nnjj=',nnjj,' iazz_w2s =',iazz_w2s,' nw2s
=',nw2s

c  Differenz idel
idel = iazz_w2s - iazz_ws

c  Besetzen A-Matrix
if (kpolar(ifeldnr).eq.3)  jend=4
if (kpolar(ifeldnr).eq.4)  jend=5

do 6311 jz=1,jend
izeil=jstart1 + jz

ispall=MMw + (nws-1)*jend + jz
ispal =ispall

cpri
cv  print*,'jz=' ,jz,' izeil=' ,izeil,'  ispal1=' ,ispal

A(izeil,ispal)= 1.
A(ispal,izeil)=A(izeil,ispal)

ispal2=ispall + idel
ispal =ispal2

cpri
cv  print*,'jz=' ,jz,' izeil=' ,izeil,'  ispal2=' ,ispal

A(izeil,ispal)=-1.
A(ispal,izeil)=A(izeil,ispal)

6311 continue

c  neues jstart1
jstart1=jstart1+jend

6010 continue

C  INDEX=8
C=====
cpri
print*,' '
print*,' '
print*,'INDEX=8'

if (irandges.eq.0) go to 1256
c=====
C  kinematische Randbedingungen
c=====
M  =jstart1
lzeile=jstart1

cpri
print*,' '
print*,' '
print*,'INDEX=8'
print*,'Randbed.  lzeile=' ,lzeile
print*,'irandw  =' ,irandw
print*,'irandws  =' ,irandws
print*,'irandw2s =' ,irandw2s

if (irandw.eq.0)  go to 1402
c  w-Bedingungen
c-----
do 1501 ii=1,irandw
Zeilenr.  lzeile
lzeile=lzeile+1

cpri
print*,'w  ii=' ,ii,'  lzeile=' ,lzeile

iii=ii
kart=1
c  Abschnitt
kfeld=jrandw(ii)

CALL Randbeding(iii,kart,randwerte,randlast)

c  bestimmen ipolart
ipolart=kpolar(kfeld)

```

```

c      Parameter setzen
      if(ipolart.eq.4) jende=5
      if(ipolart.eq.3) jende=4

cpri
cv      print*,'ii =',ii , ' kart=',kart, ' kfeld=',kfeld
cv      print*,'jende=',jende, ' ipolart=',ipolart

c      Spaltenanfangsnr.-1  nspall  suchen  w
-----
c      ielnr  Elementnr.
      ielnr=jrandw(ii)

c      suchen nach Element-Nr. in iafeld
      call suchen_elnr(ielnr,nspall,MMw,MMws,jende)

cpri
cv      print*,'ielnr  nspall  MMw  MMws  jende'
cv      print*,'ielnr,nspall,MMw,MMws,jende

      do 8511 lljj=1,jende
      nspal = nspall + lljj

cpri
cv      print*,'lljj=',lljj, ' lzeile=',lzeile, ' nspal=',nspal

      A(lzeile,nspal) = randwerte(lljj)
      A(nspal,lzeile) = A(lzeile,nspal)

      8511 continue
      B(lzeile,1) = randlast
      1501 continue

      1402 continue
      if(irandws.eq.0) go to 1403
c      ws-Bedingungen
-----
      do 1502 ii=1,irandws
      lzeile=lzeile+1

cpri
      print*,'ws  ii=',ii, ' lzeile=',lzeile

      iii=ii
      kart=2
c      Abschnitt
      kfeld=jrandws(ii)

      CALL Randbeding(iii,kart,randwerte,randlast)

c      bestimmen ipolart
      ipolart=kpolart(kfeld)

c      Parameter setzen
      if(ipolart.eq.4) jende=5
      if(ipolart.eq.3) jende=4

cpri
cv      print*,'ii =',ii , ' kart=',kart, ' kfeld=',kfeld
cv      print*,'jende=',jende, ' ipolart=',ipolart

c      Spaltenanfangsnr.-1  nspall  suchen  ws
-----
c      ielnr  Elementnr.
      ielnr=jrandws(ii)

c      suchen nach Element-Nr. in iafeld
      call suchen_elnr(ielnr,nspall,MMw,MMws,jende)

      do 8711 lljj=1,jende
      nspal = nspall + lljj

cpri
cv      print*,'lljj=',lljj, ' lzeile=',lzeile, ' nspal=',nspal

      A(lzeile,nspal) = randwerte(lljj)
      A(nspal,lzeile) = A(lzeile,nspal)

      8711 continue
      B(lzeile,1) = randlast
      1502 continue

      1403 continue
      if(irandw2s.eq.0) go to 1404
c      w2s-Bedingungen
-----
      do 1503 ii=1,irandw2s
      lzeile=lzeile+1

cpri
      print*,'w2s  ii=',ii, ' lzeile=',lzeile

      iii=ii
      kart=3
c      Abschnitt
      kfeld=jrandw2s(ii)

      CALL Randbeding(iii,kart,randwerte,randlast)

c      bestimmen ipolart
      ipolart=kpolart(kfeld)

c      Parameter setzen
      if(ipolart.eq.4) jende=5
      if(ipolart.eq.3) jende=4

cpri
cv      print*,'ii =',ii , ' kart=',kart, ' kfeld=',kfeld
cv      print*,'jende=',jende, ' ipolart=',ipolart

c      Spaltenanfangsnr.-1  nspall  suchen  w2s
-----
c      ielnr  Elementnr.
      ielnr=jrandw2s(ii)

c      suchen nach Element-Nr. in iafeld
      call suchen_elnr(ielnr,nspall,MMw,MMws,jende)

      do 8911 lljj=1,jende
      nspal = nspall + lljj

cpri
cv      print*,'lljj=',lljj, ' lzeile=',lzeile, ' nspal=',nspal

      A(lzeile,nspal) = randwerte(lljj)
      A(nspal,lzeile) = A(lzeile,nspal)

      8911 continue
      B(lzeile,1) = randlast
      1503 continue

      1404 continue
      1256 continue

c      hier Ende der Matrix-Besetzung

c      Zahl der DOF m i t Randbedingungen
      M = M + irandges

cpri
      print*,'Zahl der DOF m i t Randbedingungen M=',M

c      Uebergabe A ==> Ages
      do 9521 i=1,M
      do 9522 j=1,M
      Ages(i,j)=A(i,j)
      9522 continue
      9521 continue

C      Ausdruck Matrizen-Besetzung
      if(M.lt.300) CALL MATBES(77)

C      Ausdruck Matrizen A und B
      CALL BDRUC(77)

c      Test vor Loesen der Matrix
      CALL TESTMATRIX

C      Loesung
      CALL GAJO

c      Test nach Loesen der Matrix
      CALL TESTMATRIX2(Q,dett,iinvers)

C      Ausdruck Matrizen A und B
      CALL BDRUC(77)
      CALL BDRUCB(66)

C      Berechnung + Ausdruck der Loesungen
-----
      CALL ENDOUTPUT

c      Ausgabe Q aus Testmatrix2
      write(66,9531) Q,dett,iinvers
      9531 format(/,' Q = Bt * A * B =',E14.6,/,
a      ' , Determinante dett =',E14.6,/,
b      ' , invers =',I5)
      write(77,9531) Q,dett,iinvers

cpri
      print*,' '
      print*,' '
      print*,' '
      print*,'pause      damit Fenster bleibt'
      pause

      STOP
      END

C      SUBROUTINE hbxxxxs1(i)
      Berechnung ggs(i,j)

c      Schritt 1: fuer w aus Winkeln
c      ggs      Grundmatrix

      implicit real*8 (A-H,O-Z)

      real*8 A(300,300),B(300,2)
      COMMON/gauss/ M,N,A,B
      COMMON/hhh/xw,xws,xd,xz,aalang,messw,messws,messd,mzzz,
a      w,ws,ebs,z,bb, ggs,gg,ww, wws,ee

      COMMON/hhh1/fpol,isysw,ipolart
      COMMON/hhh2/pl1,plr,ak2,ak2t
      COMMON/hhh3/NFELD,MM1,M1
      COMMON/hhh4/Feld

```

```

COMMON/hhh5/irandges, irandw, irandws, irandw2s
COMMON/hhh6/jrandw, jrandws, jrandw2s,
a      aaxxw, aaxxws, aaxxw2s, ccw, ccws, ccw2s

character*1 Feld(80)
real*8 bb(5,5), ggs(5,5), gg(5,5), ww(5), wws(5), ee(5)
real*8 xw(10,50), xws(10,50), xd(10,50), xz(10,50),
a      w(10,50), ws(10,50), ebs(10,50), z(10,50)

real*8 aalang(10)
integer messw(20), messws(20), messd(20), mzzz(20),
a      jrandw(20), jrandws(20), jrandw2s(20)
real*8 aaxxw(20), aaxxws(20), aaxxw2s(20),
a      ccw(20), ccws(20), ccw2s(20)

real*8 ak2(3,10), ak2t(10,3)
real*8 wneu(10,50,3), wstri(10,50,3), w2stri(10,50,3)
a, wsneu(10,50,3), ebsneu(10,50,3)
real*8 fpol(6,6), pll(3,5), plr(3,5)
real*8 hhw(5), hhstri(5), hh2stri(5), eb2stri(5)

c      Parameter setzen
      if(ipolart.eq.4) jende=5
      if(ipolart.eq.3) jende=4

C      Null setzen
      do 1 j=1, jende
      wws(j)=0.
      continue

1

C      besetzen wws(j)
      do 2 j=1, jende
      Null setzen
      wert=0.
      do 3 k=1, messws(i)
      cc=xws(i,k)/aalang(i)
      cc2=cc*cc
      cc3=cc*cc2
      aal=aalang(i)

C      Winkel-Anteile
      if(j.eq.1) hhh = 0.
      if(j.eq.2) hhh = 1. /aal
      if(j.eq.3) hhh = 2.*cc /aal
      if(j.eq.4) hhh = 3.*cc2 /aal
      if(j.eq.5) hhh = 4.*cc3 /aal

c      hier gemessene Winkel
      wert=wert+hhh*ws(i,k)
      continue
      wws(j)=wert
      continue
      return
      end

2
3
2

C      SUBROUTINE hbxxxeb1(i)
      Berechnung gg(i,j)

c      Schritt 1: fuer w aus ebs
      gg      Grundmatrix

      implicit real*8 (A-H,O-Z)

      real*8 A(300,300), B(300,2)
      COMMON/gauss/ M,N,A,B
      COMMON/hhh/xw, xws, xd, xz, aalang, messw, messws, messd, mzzz,
a      w, ws, ebs, z, bb, ggs, gg, ww, wws, ee

      COMMON/hhh1/fpol, isysw, ipolart
      COMMON/hhh2/pll, plr, ak2, ak2t
      COMMON/hhh3/NFELD, MM1, M1
      COMMON/hhh4/Feld
      COMMON/hhh5/irandges, irandw, irandws, irandw2s
      COMMON/hhh6/jrandw, jrandws, jrandw2s,
a      aaxxw, aaxxws, aaxxw2s, ccw, ccws, ccw2s

character*1 Feld(80)
real*8 bb(5,5), ggs(5,5), gg(5,5), ww(5), wws(5), ee(5)
real*8 xw(10,50), xws(10,50), xd(10,50), xz(10,50),
a      w(10,50), ws(10,50), ebs(10,50), z(10,50)

real*8 aalang(10)
integer messw(20), messws(20), messd(20), mzzz(20),
a      jrandw(20), jrandws(20), jrandw2s(20)
real*8 aaxxw(20), aaxxws(20), aaxxw2s(20),
a      ccw(20), ccws(20), ccw2s(20)

real*8 ak2(3,10), ak2t(10,3)
real*8 wneu(10,50,3), wstri(10,50,3), w2stri(10,50,3)
a, wsneu(10,50,3), ebsneu(10,50,3)
real*8 fpol(6,6), pll(3,5), plr(3,5)
real*8 hhw(5), hhstri(5), hh2stri(5), eb2stri(5)

c      Parameter setzen
      if(ipolart.eq.4) jende=5
      if(ipolart.eq.3) jende=4

C      Null setzen
      do 1 j=1, jende
      do 2 k=1, jende
      gg(j,k)=0.
      continue
      continue

2
1

C      besetzen gg(j,l)
      do 3 j=1, jende
      do 4 l=1, jende
      Null setzen
      wert=0.
      do 5 k=1, messd(i)
      cc=xd(i,k)/aalang(i)
      aa2=aalang(i)*aalang(i)

c      Dehnungsanteile
      if(j.eq.1) hhj = 0.
      if(j.eq.2) hhj = 0.
      if(j.eq.3) hhj = 2. /aa2
      if(j.eq.4) hhj = 6.*cc /aa2
      if(j.eq.5) hhj = 12.*cc*cc/aa2

      if(l.eq.1) hhl = 0.

```

```

if(1.eq.2) hhl = 0.
if(1.eq.3) hhl = 2. /aa2
if(1.eq.4) hhl = 6.*cc /aa2
if(1.eq.5) hhl = 12.*cc*cc/aa2

wert=wert+hhl*hh1
continue
gg(j,l)=wert

4 continue
3 continue
return
end

SUBROUTINE hwe1(i)
Berechnung ee(i)

c Schritt 1: fuer w aus ebs
c ee Lastmatrix

implicit real*8(A-H,O-Z)

real*8 A(300,300),B(300,2)
COMMON/gauss/ M,N,A,B
COMMON/hhh/xw,xws,xd,xz,aalang,messw,messws,messd,mzzz,
a w,ws,ebs,z,bb,ggs,gg,ww,wws,ee

COMMON/hhh1/fpol,isysw,ipolart
COMMON/hhh2/pll,plr,ak2,ak2t
COMMON/hhh3/NFELD,MM1,M1
COMMON/hhh4/Feld
COMMON/hhh5/irandges,irandw,irandws,irandw2s
COMMON/hhh6/jrandw,jrandws,jrandw2s,
a aaxxw,aaxxws,aaxxw2s,ccw,ccws,ccw2s

character*1 Feld(80)
real*8 bb(5,5),ggs(5,5),gg(5,5),ww(5),wvs(5),ee(5)
real*8 xw(10,50),xws(10,50),xd(10,50),xz(10,50),
a w(10,50),ws(10,50),ebs(10,50),z(10,50)

real*8 aalang(10)
integer messw(20),messws(20),messd(20),mzzz(20),
a jrandw(20),jrandws(20),jrandw2s(20)
real*8 aaxxw(20),aaxxws(20),aaxxw2s(20),
a ccw(20),ccws(20),ccw2s(20)

real*8 ak2(3,10),ak2t(10,3)
real*8 wneu(10,50,3),wstri(10,50,3),w2stri(10,50,3)
a,wsneu(10,50,3),ebsneu(10,50,3)
real*8 fpol(6,6),pll(3,5),plr(3,5)
real*8 hhw(5),hhstri(5),hh2stri(5),eb2stri(5)

COMMON/hhh1a/ksysw,kpolart
integer ksysw(50),kpolart(50)

real*8 xx,zz

C=====
==
C Lesen von zus3.ein
C
C Zeile 1 Text
read(55,101) (Feld(i),i=1,80)
101 format(80A1)
C Zeile 2 NFELD Zahl der Felder bzw. Abschnitte
READ(55,224) NFELD
224 format(I10,3F20.0)

c Einlesen Funktionstypen
c je Abschnitt
c-----
do 31 i=1,NFELD
READ(55,1) kpolart(i)
1 FORMAT(5I10)
if(kpolart(i).le.2.or.kpolart(i).ge.5) go to 8341
continue
go to 8342

cpri
8341 continue
print*,'i=',i,' kpolart(i)=' ,kpolart(i),' le 2 bzw.
ge 5' stop

8342 continue

c if(NFELD.eq.1) go to 8442
c Einlesen Funktionswerte der
c inneren Abschnittsgrenzen
c-----
do 41 i=1,NFELD-1
READ(55,1) ksysw(i)
if(ksysw(i).ge.4) go to 8441
41 continue
go to 8442

cpri
8441 continue
print*,'i=',i,' ksysw(i)=' ,ksysw(i),' ge 4'
stop

8442 continue

c Schleife alle Felder bzw. Abschnitte
c Einlesen Parameter je Feld
c-----
do 51 i=1,NFELD
READ(55,2) aalang(i),messw(i),messws(i),messd(i),mzzz(i)

cpri
cc print*,'aalang(i)=' ,aalang(i),
cc a' messw(i) =' ,messw(i),
cc b' messws(i) =' ,messws(i),
cc c' messd(i) =' ,messd(i),' mzzz(i) =' ,mzzz(i)
2 format(F20.0,5I10)
if(messw(i).eq.0) go to 52

c Einlesen Durchbiegungen // Messungen
do 53 j=1,messw(i)
READ(55,3) xw(i,j),w(i,j)

cpri
cc print*,' bei xw,w gelesen j=',j
3 format(4F20.0)
53 continue

52 continue

c if(messws(i).eq.0) go to 522
c Einlesen Winkel // Messungen

```

```

do 533 j=1,messws(i)
READ(55,3) xws(i,j),ws(i,j)
cpri
cc print*,' bei xws,ws gelesen j=',j
533 continue

522 continue

c if(messd(i).eq.0) go to 54
      Einlesen Dehnungen // Messungen
do 55 j=1,messd(i)
READ(55,3) xd(i,j),ebs(i,j)
cpri
cc print*,' bei xd,ebs gelesen j=',j
55 continue

54 continue

      if(mzzz(i).eq.0) go to 51
c      Einlesen Hoehe z von neutr. Linie
do 755 j=1,mzzz(i)
READ(55,3) xz(i,j),z(i,j)
cpri
cc print*,' bei z gelesen j=',j
755 continue

51 continue

C Randbedingungen
READ(55,223) irandges
223 format(8I10)
READ(55,223) irandw,irandws,irandw2s

isumm=irandw+irandws+irandw2s
cpri
print*,'isumm =',isumm
print*,'irandges=',irandges

if(isumm.ne.irandges) print*,'isum ne irandges'
if(isumm.ne.irandges) stop

if(irandges.eq.0) go to 61

c if(irandw.eq.0) go to 587
Randbedingungen fuer w einlesen
do 71 ii=1,irandw
READ(55,224) jrandw(ii),aaxxw(ii),ccw(ii)
cpri
cc print*,'jrandw(ii),' ',aaxxw(ii),' ',ccw(ii)
71 continue
587 continue

c if(irandws.eq.0) go to 588
Randbedingungen fuer ws einlesen
do 72 ii=1,irandws
READ(55,224) jrandws(ii),aaxxws(ii),ccws(ii)
72 continue
588 continue

c if(irandw2s.eq.0) go to 589
Randbedingungen fuer w2s einlesen
do 73 ii=1,irandw2s
READ(55,224) jrandw2s(ii),aaxxw2s(ii),ccw2s(ii)
73 continue
589 continue

61 continue
C=====
==

C=====
=====
C Ausgabe der bekannten Parameter
C-----
      write(66,201) NFELD,irandges
201 format(' PROGRAMM zus3-1-abstand: mit ',/,
a ' NFELD =',I3,/,
b ' irandges =',I3,/)

c Funktionstypen
c je Abschnitt
C-----
      write(66,129)
129 format(' Funktionstypen je Abschnitt',/)
do 131 i=1,NFELD
if(kpolart(i).eq.4) write(66,132) i,i,kpolart(i)
132 format('i=',i2,' kpolart(',i2,')=',i1,' 5 Parameter')
if(kpolart(i).eq.3) write(66,122) i,i,kpolart(i)
122 format('i=',i2,' kpolart(',i2,')=',i1,' 4 Parameter')
131 continue

c Funktionswerte der
c inneren Abschnittsgrenzen
C-----
      write(66,139)
139 format(/,' Funktionswerte der inneren Abschnitte',/)
do 141 i=1,NFELD-1
write(66,142) i,i+1,i,ksysw(i)
142 format('i=',i2,' i+1=',i2,' ksysw(',i2,')=',i1)
141 continue

      write(66,1081)
1081 format(/)

c Randbedingungen
C-----
if(irandges.eq.0) go to 161

c if(irandw.eq.0) go to 687
Randbedingungen fuer w einlesen
do 171 ii=1,irandw
write(66,2278) jrandw(ii),aaxxw(ii),ccw(ii)
2278 format(' Randbedingung jrandw =',i2,
a' ksi=',E12.5,' Wert=',E12.5)
171 continue
687 continue

c if(irandws.eq.0) go to 688
Randbedingungen fuer ws einlesen
do 172 ii=1,irandws

C - - - - -
c umrechnen der Winkel * Abschnittslaenge
aaall=aalang(jrandws(ii))
ccneuw=ccws(ii)*aaall
C - - - - -

      write(66,2279) jrandws(ii),aaxxws(ii),ccws(ii),ccneuw
2279 format(' Randbedingung jrandws =',i2,
a' ksi=',E12.5,' Wert=',E12.5,' neuer Wert=',E12.5)

C - - - - -
c ccws(ii)=ccneuw
C - - - - -

172 continue

688 continue

c if(irandw2s.eq.0) go to 689
Randbedingungen fuer w2s einlesen
do 173 ii=1,irandw2s

C - - - - -
c umrechnen der Dehnung * Abschnittslaenge*2
aaall2=aalang(jrandw2s(ii))*aalang(jrandw2s(ii))
ccneuw2s=ccw2s(ii)*aaall2
C - - - - -

      write(66,2280)
jrandw2s(ii),aaxxw2s(ii),ccw2s(ii),ccneuw2s
2280 format(' Randbedingung jrandw2s=',i2,
a' ksi=',E12.5,' Wert=',E12.5,' neuer Wert=',E12.5)

C - - - - -
c ccw2s(ii)=ccneuw2s
C - - - - -

173 continue
689 continue
161 continue

c write(66,11)
w + ws + ebs
write(66,304)
304 format('falls alle ksysw(i)=0 ohne Kopplung',/,
a'falls ksysw(i)=1 w + ws + ebs mit Zeile
1',/,
b'falls ksysw(i)=2 w + ws + ebs mit Zeile
1+2',/,
b'falls ksysw(i)=3 w + ws + ebs mit Zeile
1+2+3',/)

c if(igewicht.eq.0) write(66,1293)
1293 format(' o h n e Gewichtung !!!',/)
if(igewicht.eq.1) write(66,1294)
1294 format(' m i t Gewichtung !!!',/)

      write(66,101) (Feld(i),i=1,80)

do 202 i=1,NFELD
write(66,203)
i,aalang(i),messw(i),messws(i),messd(i),mzzz(i)
203 format(/,' i=',i2,' Laenge=',E12.5,/,6x,
a' Zahl der Durchbiegungsmesswerte=',I3,/,6x,
b' Zahl der Winkelmesswerte =',I3,/,6x,
c' Zahl der Dehnungsmesswerte =',I3,/,6x,
d' Zahl der z-Werte =',I3)

c if(messw(i).eq.0) go to 204
Ausgabe Durchbiegungen // Messungen
c write(66,206)
format(/,'Messwerte der Durchbiegungen')
do 207 j=1,messw(i)
write(66,208) i,j,xw(i,j),w(i,j)
208 format(' i=',i2,' j=',i3,' xw =',E12.5,' w
=',E12.5)
207 continue

204 continue

c if(messws(i).eq.0) go to 2204
Ausgabe Winkel // Messungen
c write(66,2206)
2206 format(/,'Messwerte der Winkel')
do 2207 j=1,messws(i)
write(66,2208) i,j,xws(i,j),ws(i,j)

```

```

2208 format(' i=',I2,' j=',I3,' xws =' ,E12.5,' ws
=' ,E12.5)
2207 continue

2204 continue

c if(messd(i).eq.0) go to 209
      Ausgabe      Dehnungen // Messungen
210 write(66,210)
      format(/,'Messwerte der Dehnungen')
      do 211 j=1,messd(i)
-----
      ebs(i,j)=0.000001*ebs(i,j)
-----

      write(66,212) i,j,xd(i,j),ebs(i,j)
212 format(' i=',I2,' j=',I3,' xd =' ,E12.5,' ebs
=' ,E12.5)
211 continue

209 continue

c      Ausgabe      Hoehe z von neutr. Linie
      write(66,213)
213 format(/,'Hoehe z von neutraler Linie')
      do 214 j=1,mzzz(i)
      write(66,215) i,j,xz(i,j),z(i,j)
215 format(' i=',I2,' j=',I3,' xz =' ,E12.5,' z
=' ,E12.5)
214 continue

202 continue
      write(66,11)
11 format(/)
C=====
=====

      return
      end

SUBROUTINE ENDOUTPUT
c Endoutput der Loesungen zum Vergleich

implicit real*8(A-H,O-Z)

integer iafeld(100,7)
common/adressfeld/iafeld,izges
real*8 zzeil1(5),zzeil2(5)
common/besetzen1/zzeil1,wert1
common/besetzen2/zzeil2,wert2
common/indeexx/index

real*8 A(300,300),B(300,2)
COMMON/gauss/ M,N,A,B
COMMON/hhh/xw,xws,xd,xz,aalang,messw,messws,messd,mzzz,
a w,ws,ebs,z,bb,gg,gg,ww,wws,ee

COMMON/hhh1/fpol,isysw,ipolart
COMMON/hhh2/pll,plr,ak2,ak2t
COMMON/hhh3/NFELD,MM1,M1
COMMON/hhh4/Feld
COMMON/hhh5/irandges,irandw,irandws,irandw2s
COMMON/hhh6/jrandw,jrandws,jrandw2s,
a aaxxw,aaxxws,aaxw2s,ccw,ccws,ccw2s

character*1 Feld(80)
real*8 bb(5,5),gg(5,5),gg(5,5),ww(5),wws(5),ee(5)
real*8 xw(10,50),xws(10,50),xd(10,50),xz(10,50),
a w(10,50),ws(10,50),ebs(10,50),z(10,50)

real*8 aalang(10)
integer messw(20),messws(20),messd(20),mzzz(20),
a jrandw(20),jrandws(20),jrandw2s(20)
real*8 aaxxw(20),aaxxws(20),aaxw2s(20),
a ccw(20),ccws(20),ccw2s(20)

real*8 ak2(3,10),ak2t(10,3)
real*8 wneu(10,50,3),wstri(10,50,3),w2stri(10,50,3)
a,wsneu(10,50,3),ebsneu(10,50,3)
real*8 fpol(6,6),pll(3,5),plr(3,5)
real*8 hh(5),hhstri(5),hh2stri(5),eb2stri(5)

real*8 we(10,50,1),wes(10,50,1)

COMMON/hhh1a/ksysw,kpolart
integer ksysw(50),kpolart(50)

real*8 xx,zz

c Kanal-Nr. ka
ka=66

C ENDOUTPUT
C=====
cout2

write(66,201) NFELD,irandges
201 format(' PROGRAMM zus3-1-abstand: mit ',/,/,
a ' NFELD =',I3,/,/,
b ' irandges =' ,I3,/)

c Funktionswerte der
c inneren Abschnittsgrenzen
c-----
      write(66,139)
139 format(' Funktionswerte der inneren Abschnitte',/)
      do 141 i=1,NFELD-1
      write(66,142) i,i+1,i,ksysw(i)
142 format('i=',i2,' i+1=',i2,' ksysw(' ,i2,')=',i1)
141 continue

c Randbedingungen
if(irandges.eq.0) go to 689

      write(66,11)

if(irandw.eq.0) go to 687

c Randbedingungen fuer w einlesen
do 171 ii=1,irandw
      write(66,2278) jrandw(ii),aaxw(ii),ccw(ii)
2278 format(' Randbedingung jrandw =' ,i2,
a ' ksi=',E12.5,' Wert=',E12.5)
171 continue
687 continue

      if(irandws.eq.0) go to 688
c Randbedingungen fuer ws einlesen
do 172 ii=1,irandws
      write(66,2279) jrandws(ii),aaxws(ii),ccws(ii)
2279 format(' Randbedingung jrandws =' ,i2,
a ' ksi=',E12.5,' Wert=',E12.5)
172 continue
688 continue

      if(irandw2s.eq.0) go to 689
c Randbedingungen fuer w2s einlesen
do 173 ii=1,irandw2s
      write(66,2280) jrandw2s(ii),aaxw2s(ii),ccw2s(ii)
2280 format(' Randbedingung jrandw2s=' ,i2,
a ' ksi=',E12.5,' Wert=',E12.5)
173 continue
689 continue

      if(igewicht.eq.0) write(ka,1293)
1293 format(' o h n e Gewichtung !!!',/)

      write(ka,101) (Feld(i),i=1,80)
101 format(80A1)

C Vergleich gegebene - neue Biegelinie
C=====
c igewicht = 0 ==> ilast=1 mnlk=1
      ilast=1
      mnlk =1

c neue Anfangsnr. - 1
cccc iwa=0
cpri

      print*,'w iwa=',iwa
c-----

do 500 i=1,NFELD
cpri
cc print*,'w mnlk=',mnlk,' i=',i
cc print*,'Anfangsnr.-1 iwa=',iwa

c suchen in iafeld Anfang - 1 =iwa !!
call suchen_anfangm1(i,iwa)

c hier Feld i nicht gemessen
c-----
if(iwa.ge.0) go to 5503

c hier iwa=-100 dieses Feld keine Messwerte !!!
write(66,5501) i,iwa
5501 format(/,' Feld ',i2,' keine Messwerte iwa=',i5,/)
go to 500

5503 continue

c Durchbiegungsanteile
c-----
C Loesungen
u1=B(iwa+1,ilast)
u2=B(iwa+2,ilast)
u3=B(iwa+3,ilast)
u4=B(iwa+4,ilast)
u5=B(iwa+5,ilast)

c bestimmen ipolart
ipolart=kpolart(i)

c Parameter setzen
if(ipolart.eq.4) jende=5
if(ipolart.eq.3) jende=4
if(ipolart.eq.3) u5=0.

c Durchbiegungen // Messungen

      write(ka,311)
311 format(

```

```

a'      i      j      x      wneu      wstri
w2stri  b      epsilon')
      dell=aalang(i)/10.
      xc=-dell
      DO 6000 j=1,11
      xc=xc+dell
c      Durchbiegungsanteile
      cc =xc/aalang(i)
      cc2=cc*cc
      cc3=cc*cc2
      cc4=cc*cc3
      aal=aalang(i)
      aa2=aalang(i)*aalang(i)
      wneu      = u1      + u2*cc
      a          + u3*cc2 + u4*cc3 + u5*cc4
c      Winkel      wstri
      wwstri      = ( u2      + u3*2.*cc
      a          + u4*3.*cc2 + u5*4.*cc3 )/aal
c      echte Kruemmung      w2stri
      ww2stri      = ( u3*2.+ u4*6.*cc + u5*12.*cc2)/aa2
c      festlegen Bezugshoehe zz
      xx=xc
      CALL zfestlegen(i,xx,zz)
      epsilon=ww2stri*zz
c      Vorzeichenwechsel
cc      w2stri(i,j,mnlk) = - w2stri(i,j,mnlk)
C=====
      write(ka,313) i,j,xc,wneu,wwstri,ww2stri,epsilon
      313 format(2i6,5E14.6)
C=====
6000 continue
      write(ka,11)
      11 format(/)
500 continue
      return
      end
c      SUBROUTINE zfestlegen(i,xx,zz)
c      festlegen z = Hoehe von neutraler Linie
      COMMON/hhh/xw,xws,xd,xz,aalang,messw,messws,messd,mzzz,
      a      w,ws,ebs,z,bb,ggs,gg,ww,wws,ee
      real*8 bb(5,5),ggs(5,5),gg(5,5),ww(5),wws(5),ee(5)
      real*8 xw(10,50),xws(10,50),xd(10,50),xz(10,50),
      a      w(10,50),ws(10,50),ebs(10,50),z(10,50)
      real*8 aalang(10)
      integer messw(20),messws(20),messd(20),mzzz(20)
      real*8 xx,zz
c      i      ist der Abschnitt
c      xx      ist die gesuchte, zug. Koordinate fuer zz
c      zz      ist die interpolierte Hoehe z
      kke=mzzz(i)
      if(mzzz(i).ne.1) go to 10
c      bei konst. Hoehe      mzzz(i)=1
c-----
      zz=z(i,kke)
cpri
cc      print*,'konst.      zz=',zz
cc      pause
cc      return
      10 continue
c      variable Hoehe      mzzz(i)>1
c=====
c      xx gleich vorgegebener z-Wert xz
c-----
      do 221 kk=1,kke
      if(xx.eq.xz(i,kk)) go to 222
      221 continue
c      nicht gefunden
      go to 199
      222 continue
c      gefunden
      zz = z(i,kk)
cpri
cc      print*,'= gegeb. z      zz=',zz
cc      pause
cc      return
199 continue
c      xx nicht gleich vorgegebener z-Wert xz
c-----
c      Frage:      xx      kleiner      xz(i,1) ? ja ==> 40
      if(xx.lt.xz(i,1)) go to 40
c      Frage:      xx      groesser      xz(i,kke) ? ja ==> 50
      if(xx.gt.xz(i,kke)) go to 50
c      hier liegt xx      zwischen den xz-Werten
c-----
      do 21 kk=1,kke
      if(xx.le.xz(i,kk)) go to 22
      21 continue
      22 continue
      kkli=kk-1
      kkre=kk
      dx = xz(i,kkre) - xz(i,kkli)
      dz = z(i,kkre) - z(i,kkli)
      sx = xx - xz(i,kkli)
      zz = z(i,kkli) + dz*(sx/dx)
cpri
cc      print*,'zwischen      zz=',zz
cc      pause
cc      return
40 continue
c      hier ist xx      kleiner      xz(i,1)
c-----
      kkli=1
      kkre=2
      dx = xz(i,kkre) - xz(i,kkli)
      dz = z(i,kkre) - z(i,kkli)
      sx = xz(i,kkre) - xx
      zz = z(i,kkre) - dz*(sx/dx)
cpri
cc      print*,'xx      kleiner      zz=',zz
cc      pause
cc      return
50 continue
c      hier ist xx      groesser      xz(i,kke)
c-----
      kkli=kke-1
      kkre=kke
      dx = xz(i,kkre) - xz(i,kkli)
      dz = z(i,kkre) - z(i,kkli)
      sx = xx - xz(i,kkli)
      zz = z(i,kkli) + dz*(sx/dx)
cpri
cc      print*,'xx      groesser      zz=',zz
cc      pause
cc      return
      end
c      SUBROUTINE KOPPEL(i)
c      Koppelmatrix
      implicit real*8(A-H,O-Z)
      common/indeex/index
      real*8 A(300,300),B(300,2)
      COMMON/gauss/ M,N,A,B
      COMMON/hhh/xw,xws,xd,xz,aalang,messw,messws,messd,mzzz,
      a      w,ws,ebs,z,bb,ggs,gg,ww,wws,ee
      COMMON/hhh1/fpol,isysw,ipolart
      COMMON/hhh2/pll,plr,ak2,ak2t
      COMMON/hhh3/NFELD,MM1,M1
      COMMON/hhh4/Feld
      COMMON/hhh5/irandges,irandw,irandws,irandw2s
      COMMON/hhh6/jrandw,jrandws,jrandw2s,
      a      aaxxw,aaxxws,aaxxw2s,ccw,ccws,ccw2s
      character*1 Feld(80)
      real*8 bb(5,5),ggs(5,5),gg(5,5),ww(5),wws(5),ee(5)
      real*8 xw(10,50),xws(10,50),xd(10,50),xz(10,50),
      a      w(10,50),ws(10,50),ebs(10,50),z(10,50)
      real*8 aalang(10)
      integer messw(20),messws(20),messd(20),mzzz(20),
      a      jrandw(20),jrandws(20),jrandw2s(20)
      real*8 aaxxw(20),aaxxws(20),aaxxw2s(20),
      a      ccw(20),ccws(20),ccw2s(20)
      real*8 ak2(3,10),ak2t(10,3)
      real*8 wneu(10,50,3),wstri(10,50,3),w2stri(10,50,3)
      a,wsneu(10,50,3),ebsneu(10,50,3)
      real*8 fpol(6,6),pll(3,5),plr(3,5)
      real*8 hhw(5),hhstri(5),hh2stri(5),eb2stri(5)
      COMMON/hhhla/ksysw,kpolart
      integer ksysw(50),kpolart(50)
c      koppeln Abschnitt i u. i+1

```



```

c      bestimmen ipolart
      ipolartli=kpolart(i )
      ipolartre=kpolart(i+1)

c      Parameter setzen
      if(ipolartli.eq.4) jendeli=5
      if(ipolartre.eq.4) jendere=5

      if(ipolartli.eq.3) jendeli=4
      if(ipolartre.eq.3) jendere=4

c      null setzen ak2
      do 4711 kk=1,3
      do 4712 ll=1,10
      ak2(kk,ll)=0.
4712 continue
4711 continue

c      null setzen ak2t
      do 4811 kk=1,10
      do 4812 ll=1,3
      ak2t(kk,ll)=0.
4812 continue
4811 continue

c      i      Anschnittsgrenze i/i+1

C      Bezugslaenge ist Feld l= aalang(1)
      ccc=aalang(1)
      cc2=ccc*ccc

C      Feld i
      aaa=aalang(i)
      aa2=aaa*aaa

C      Feld i+1
      bbb=aalang(i+1)
      bb2=bbb*bbb

c      voll besetzen ==>5 Parameter
      do 121 kk=1,5
c      Zeile 1
      pll(1,kk) =          1.

      if(kk.eq.1) plr(1,kk) ==-1.
      if(kk.ne.1) plr(1,kk) = 0.

c      Zeile 2
      pll(2,kk) =          (kk-1)

      if(kk.eq.2) plr(2,kk) ==-1.
      if(kk.ne.2) plr(2,kk) = 0.

c      Zeile 3
      if(kk.le.2) pll(3,kk) = 0.
      if(kk.eq.3) pll(3,kk) = 2.
      if(kk.eq.4) pll(3,kk) = 6.
      if(kk.eq.5) pll(3,kk) =12.

      if(kk.eq.3) plr(3,kk) ==-2.
      if(kk.ne.3) plr(3,kk) = 0.

121 continue

      do 21 kk=1,jendeli
c      Zeile 1
      ak2(1,kk)          = pll(1,kk)

c      Zeile 2
      ak2(2,kk)          = pll(2,kk) *(ccc/aaa)

c      Zeile 3
      ak2(3,kk)          = pll(3,kk) *(cc2/aa2)
21 continue

      do 212 kk=1,jendere
c      Zeile 1
      ak2(1,kk+jendeli) = plr(1,kk)

c      Zeile 2
      ak2(2,kk+jendeli) = plr(2,kk) *(ccc/bbb)

c      Zeile 3
      ak2(3,kk+jendeli) = plr(3,kk) *(cc2/bb2)
212 continue

      do 22 kk=1,jendeli+jendere
      do 23 jj=1,3
      ak2t(kk,jj)=ak2(jj,kk)
23 continue
22 continue

-----
c      in ak2      (iz,kk)   iz=Zeilen 1-3   kk=Par. 1-5
c      oder
c      in ak2      (iz,kk)   iz=Zeilen 1-3   kk=Par. 1-4
c      Zeilen der Koppelwerte

c      in ak2t      Transponierte ak2
-----
return
end

SUBROUTINE Randbeding(iii,kart,randwerte,randlast)
berechnen der Randbedingungswerte

implicit real*8(A-H,O-Z)

real*8 A(300,300),B(300,2)
COMMON/gauss/ M,N,A,B
COMMON/hhh/xw,xws,xd,xz,aalang,messw,messws,messd,mzzz,
a      w,ws,ebs,z,bb,ggs,gg,ww,wws,ee

COMMON/hhh1/fpol,isysw,ipolart
COMMON/hhh2/pll,plr,ak2,ak2t
COMMON/hhh3/NFELD,MM1,M1
COMMON/hhh4/Feld
COMMON/hhh5/irandges,irandw,irandws,irandw2s
COMMON/hhh6/jrandw,jrandws,jrandw2s,
a      aaxxw,aaxxws,aaxxw2s,ccw,ccws,ccw2s

character*1 Feld(80)
real*8 bb(5,5),ggs(5,5),gg(5,5),ww(5),wws(5),ee(5)
real*8 xw(10,50),xws(10,50),xd(10,50),xz(10,50),
a      w(10,50),ws(10,50),ebs(10,50),z(10,50)

real*8 aalang(10)
integer messw(20),messws(20),messd(20),mzzz(20),
a      jrandw(20),jrandws(20),jrandw2s(20)
real*8 aaxxw(20),aaxxws(20),aaxxw2s(20),
a      ccw(20),ccws(20),ccw2s(20)

real*8 ak2(3,10),ak2t(10,3)
real*8 wneu(10,50,3),wstri(10,50,3),w2stri(10,50,3)
a,wsneu(10,50,3),ebsneu(10,50,3)
real*8 fpol(6,6),pll(3,5),plr(3,5)
real*8 hhv(5),hhstri(5),hh2stri(5),eb2stri(5)

real*8 we(10,50,1),wes(10,50,1)
real*8 randwerte(5)

c      Parameter setzen
      if(ipolart.eq.4) jende=5
      if(ipolart.eq.3) jende=4

C      kinematische Randbedingungen
c      iii      Stelle z.B. jrand..(iii)
c      kart      Art der Bedingung
c      1=w      2=ws      3=w2s

c      alle Null setzen
      do 1 kkk=1,5
      randwerte(kkk)=0.
1 continue
      randlast=0.

      if(kart.eq.1) go to 10
      if(kart.eq.2) go to 20
      if(kart.eq.3) go to 30

10 continue
c      w-Randbedingungen
c      aksi =aaxxw(iii)
      do 12 jjj=1,jende
      randwerte(jjj)=aksi**(jjj-1)
12 continue
      randlast=ccw(iii)
      return

20 continue
c      ws-Randbedingungen
c      aksi =aaxxws(iii)
      randwerte(1)=0.
      do 22 jjj=2,jende
      randwerte(jjj)=(jjj-1)*aksi**(jjj-2)
22 continue
      randlast=ccws(iii)
      return

30 continue
c      w2s-Randbedingungen
c      aksi =aaxxw2s(iii)
      randwerte(1)=0.
      randwerte(2)=0.
      do 32 jjj=3,jende
      randwerte(jjj)=(jjj-1)*(jjj-2)*aksi**(jjj-3)
32 continue
      randlast=ccw2s(iii)
      return

end

SUBROUTINE hww(i)
Berechnung ww(i,j)

implicit real*8(A-H,O-Z)

```

```

real*8 A(300,300),B(300,2)
COMMON/gauss/ M,N,A,B
COMMON/hhh/xw,xws,xd,xz,aalang,messw,messws,messd,mzzz,
a w,ws,ebs,z,bb,ggs,gg,ww,wws,ee

COMMON/hhh1/fpol,isisw,ipolart
COMMON/hhh2/pl1,plr,ak2,ak2t
COMMON/hhh3/NFELD,MM1,M1
COMMON/hhh4/Feld
COMMON/hhh5/irandges,irandw,irandws,irandw2s
COMMON/hhh6/jrandw,jrandws,jrandw2s,
a aaxxw,aaxxws,aaxxw2s,ccw,ccws,ccw2s

character*1 Feld(80)
real*8 bb(5,5),ggs(5,5),gg(5,5),ww(5),wws(5),ee(5)
real*8 xw(10,50),xws(10,50),xd(10,50),xz(10,50),
a w(10,50),ws(10,50),ebs(10,50),z(10,50)

real*8 aalang(10)
integer messw(20),messws(20),messd(20),mzzz(20),
a jrandw(20),jrandws(20),jrandw2s(20)
real*8 aaxxw(20),aaxxws(20),aaxxw2s(20),
a ccw(20),ccws(20),ccw2s(20)

real*8 ak2(3,10),ak2t(10,3)
real*8 wneu(10,50,3),wstri(10,50,3),w2stri(10,50,3)
a,wsneu(10,50,3),ebsneu(10,50,3)
real*8 fpol(6,6),pl1(3,5),plr(3,5)
real*8 hhw(5),hhstri(5),hh2stri(5),eb2stri(5)

c Parameter setzen
if(ipolart.eq.4) jende=5
if(ipolart.eq.3) jende=4

C Null setzen
do 1 j=1,jende
ww(j)=0.
continue

1 besetzen ww(j)
do 2 j=1,jende
Null setzen
wert=0.
do 3 k=1,messw(i)
cc=xw(i,k)/aalang(i)

c Durchbiegungsanteile
hhh = cc**(j-1)

wert=wert+hhh*w(i,k)
continue
ww(j)=wert
continue
return
end

SUBROUTINE hbxxxw(i)
Berechnung bb(i,j)

implicit real*8(A-H,O-Z)

real*8 A(300,300),B(300,2)
COMMON/gauss/ M,N,A,B
COMMON/hhh/xw,xws,xd,xz,aalang,messw,messws,messd,mzzz,
a w,ws,ebs,z,bb,ggs,gg,ww,wws,ee

COMMON/hhh1/fpol,isisw,ipolart
COMMON/hhh2/pl1,plr,ak2,ak2t
COMMON/hhh3/NFELD,MM1,M1
COMMON/hhh4/Feld
COMMON/hhh5/irandges,irandw,irandws,irandw2s
COMMON/hhh6/jrandw,jrandws,jrandw2s,
a aaxxw,aaxxws,aaxxw2s,ccw,ccws,ccw2s

character*1 Feld(80)
real*8 bb(5,5),ggs(5,5),gg(5,5),ww(5),wws(5),ee(5)
real*8 xw(10,50),xws(10,50),xd(10,50),xz(10,50),
a w(10,50),ws(10,50),ebs(10,50),z(10,50)

real*8 aalang(10)
integer messw(20),messws(20),messd(20),mzzz(20),
a jrandw(20),jrandws(20),jrandw2s(20)
real*8 aaxxw(20),aaxxws(20),aaxxw2s(20),
a ccw(20),ccws(20),ccw2s(20)

real*8 ak2(3,10),ak2t(10,3)
real*8 wneu(10,50,3),wstri(10,50,3),w2stri(10,50,3)
a,wsneu(10,50,3),ebsneu(10,50,3)
real*8 fpol(6,6),pl1(3,5),plr(3,5)
real*8 hhw(5),hhstri(5),hh2stri(5),eb2stri(5)

c Parameter setzen
if(ipolart.eq.4) jende=5
if(ipolart.eq.3) jende=4

C Null setzen
do 1 j=1,jende
do 2 k=1,jende
bb(j,k)=0.
continue

2

```

```

1 continue
C besetzen bb(j,1)
do 3 j=1,jende
do 4 l=1,jende
Null setzen
wert=0.
do 5 k=1,messw(i)
cc=xw(i,k)/aalang(i)

c Durchbiegungsanteile
hhj = cc**(j-1)

hhl = cc**(l-1)

wert=wert+hhj*hhl
continue
bb(j,1)=wert

4 continue
3 continue
return
end

SUBROUTINE GAJO
GLEICHUNGSLÖSER GAUSS-JORDAN
implicit real*8(A-H,O-Z)
real*8 A(300,300),B(300,2)
COMMON/gauss/ M,N,A,B
C A = NAME DER LINKEN SEITE
C B = NAME DER RECHTEN SEITE
C M = ANZAHL DER UNBEKANNTEN
C N = ANZAHL DER RECHTEN SEITEN

DO 60 I=1,M
IMAX=I
AMAX=A(I,I)
DO 12 J=I,M
IF (ABS(A(J,I))-ABS(AMAX)) 12,12,0
AAA=ABS(A(J,I))-ABS(AMAX)
IF(AAA.LE.0.) GO TO 12
IMAX=J
AMAX=A(J,I)
CONTINUE
PIVOT=AMAX
ITAUSCH=IMAX
CC IF (ABS(PIVOT)) 0,70,0
AAA=ABS(PIVOT)
IF(AAA.EQ.0.) GO TO 70
DO 20 J=1,M
TAUSCH=A(ITAUSCH,J)/PIVOT
A(ITAUSCH,J)=A(I,J)
A(I,J)=TAUSCH
DO 30 J=1,N
TAUSCH=B(ITAUSCH,J)/PIVOT
B(ITAUSCH,J)=B(I,J)
B(I,J)=TAUSCH
DO 60 K=1,M
IF(K.EQ.I) GO TO 60
PIVI=A(K,I)
DO 40 J=I,M
A(K,J)=A(K,J)-A(I,J)*PIVI
DO 50 J=1,N
B(K,J)=B(K,J)-B(I,J)*PIVI
60 CONTINUE
RETURN
calt 70 WRITE(66,80)
calt 80 FORMAT(1X,'MATRIX SINGULÄER')
70 WRITE(66,80) I,J,M,AAA
80 FORMAT(1X,'MATRIX SINGULÄER',/,
a' I=',i3,' J=',i3,' M=',i3,' AAA=',E14.6)

print*, ' MATRIX SINGULÄER'
print*, ' I=',I,' J=',J,' M=',M,' AAA=',AAA

RETURN
END

SUBROUTINE BDRUC(ka)
implicit real*8(A-H,O-Z)
real*8 A(300,300),B(300,2)
COMMON/gauss/ M,N,A,B
C ka ist Druck-Kanal

WRITE(ka,5)
FORMAT(1H1)
WRITE(ka,11)
WRITE(ka,11)
FORMAT(/)
WRITE(ka,301)
301 FORMAT(' MATRIX A',/)

ICOL=3
MK=M/ICOL
MKK=MK*ICOL
IDIF=M-MKK
IF(IDIF.GT.0) MK=MK+1
DO 333 KKK=1,MK
WRITE(ka,11)
JA=(KKK-1)*ICOL+1
JE=KKK*ICOL
IF(IDIF.NE.0.AND.KKK.EQ.MK) JE=(KKK-1)*ICOL+IDIF
DO 12 I=1,M

```

```

WRITE(ka,13) (I,J,A(I,J),J=JA,JE)
13  FORMAT(I6,1x,I3,1x,E14.6,I6,1x,I3,1x,E14.6,
1I6,1x,I3,1x,E14.6)
12  CONTINUE
333  CONTINUE
WRITE(ka,11)
WRITE(ka,11)
WRITE(ka,11)
WRITE(ka,11)
WRITE(ka,302)
302  FORMAT('  MATRIX  B',/)
MK=N/ICOL
MKK=MK*ICOL
IDIF=N-MKK
IF(IDIF.GT.0) MK=MK+1
DO 444 KKK=1,MK
WRITE(ka,11)
JA=(KKK-1)*ICOL+1
JE=KKK*ICOL
IF(IDIF.NE.0.AND.KKK.EQ.MK) JE=(KKK-1)*ICOL+IDIF
DO 14 I=1,M
WRITE(ka,13) (I,J,B(I,J),J=JA,JE)
14  CONTINUE
444  CONTINUE
RETURN
END

SUBROUTINE BDRUCB(ka)
implicit real*8 (A-H,O-Z)
real*8 A(300,300),B(300,2)
COMMON/gauss/ M,N,A,B
C      ka ist Druck-Kanal
WRITE(ka,11)
11  FORMAT(/)
WRITE(ka,11)
WRITE(ka,302)
302  FORMAT('  MATRIX  B',/)
ICOL=3
MK=N/ICOL
MKK=MK*ICOL
IDIF=N-MKK
IF(IDIF.GT.0) MK=MK+1
DO 444 KKK=1,MK
WRITE(ka,11)
JA=(KKK-1)*ICOL+1
JE=KKK*ICOL
IF(IDIF.NE.0.AND.KKK.EQ.MK) JE=(KKK-1)*ICOL+IDIF
DO 14 I=1,M
WRITE(ka,13) (I,J,B(I,J),J=JA,JE)
13  FORMAT(I6,I3,1x,E14.6,I6,I3,1x,E14.6,
1I6,I3,1x,E14.6)
14  CONTINUE
444  CONTINUE
RETURN
END

SUBROUTINE MATBES(ka)
Ausdruck Matrizen-Belegung
implicit real*8 (A-H,O-Z)
real*8 A(300,300),B(300,2)
COMMON/gauss/ M,N,A,B

character*1 AF(300,300),BF(300,2),ee,ff
integer AS(300),AZ(300),AZ1(30)

C      ka ist Druck-Kanal
ee = '-'
ff = '|'

C      Test-Ausdruck Matrizen-Schema
c-----
c  AS      Spaltenvektor
c  AZ      Zeilenvektor

iz=0
do 21 i=1,M
AS(i) = i

ii =i/10
idif =i - ii*10

if(idif.eq.0) AZ(i) = 0
if(idif.ne.0) AZ(i) = idif

if(idif.eq.0) iz=iz+1
if(idif.eq.0) AZ1(iz)=iz

21  continue

do 30 i=1,M
do 31 j=1,M
if(A(i,j).eq.0.) AF(i,j)=ee
if(A(i,j).ne.0.) AF(i,j)=ff

31  continue
30  continue

do 40 i=1,M
do 41 j=1,N
if(B(i,j).eq.0.) BF(i,j)=ee
if(B(i,j).ne.0.) BF(i,j)=ff

41  continue
40  continue

write(ka,101)
101  format('  Matrizen-Schema  Matrix A',/)

write(ka,102) (AZ1(i),i=1,iz)
cccc 102  format(10x,10I10)
cccc 102  format(11x,20I10)
102  format(11x,30I10)
write(ka,103) (AZ(i),i=1,M)
cccc 103  format(10x,100I1)
cccc103  format(11x,200I1)
103  format(11x,300I1)
write(ka,111)
111  format(/)

do 200 i=1,M
write(ka,201) AS(i), (AF(i,j),j=1,M)
ccc 201  format(3x,I2,5x,100A1)
ccc 201  format(3x,I3,5x,200A1)
201  format(3x,I3,5x,300A1)
200  continue

write(ka,111)
write(ka,301)
301  format('  Matrizen-Schema  Matrix B',/)

write(ka,111)

do 300 i=1,M
write(ka,311) AS(i), (BF(i,k),k=1,N)
cccc 311  format(3x,I2,5x,10A1)
cccc 311  format(3x,I3,5x,20A1)
311  format(3x,I3,5x,30A1)
300  continue

return
end

SUBROUTINE TESTMATRIX
testen der Matrix A auf Zeilenwiederholung
implicit real*8 (A-H,O-Z)
real*8 A(300,300),B(300,2)
COMMON/gauss/ M,N,A,B

real*8 zeile(300)
A = NAME DER LINKEN SEITE
C B = NAME DER RECHTEN SEITE
C M = ANZAHL DER UNBEKANNTEN
C N = ANZAHL DER RECHTEN SEITEN

cpri
print*, 'SUB TESTMATRIX'
do 1 i=1,M-1

c  Uebergabe Zeile i ==> zeile
do 2 j=1,M
zeile(j)=A(i,j)
2  continue

c  Vergleich obere Zeilen mit zeile i
do 3 k=i+1,M

cpri
cc  print*, 'Vergleich Zeile i=',i,' Zeile k=',k

do 4 l=1,M
Abfrage Einzelmitglieder je Zeilen i bzw. k

c  Mitglied verschieden naechste Zeile
if(A(k,l).ne.zeile(l)) go to 3

4  continue
c  alle Mitglieder Zeilen i bzw. k gleich
c  also Zeilenwiederholung

cpri
print*, 'SUB TESTMATRIX'
print*, ' Zeile i=',i,' gleich Zeile k=',k
print*, ' '
print*, '==> STOP'

c100708  stop

3  continue

1  continue

return
end

SUBROUTINE TESTMATRIX2(Q,dett,iinvers)
testen der Matrix A auf Definitzuestand
implicit real*8 (A-H,O-Z)

```

```

real*8 A(300,300),B(300,2)
COMMON/gauss/ M,N,A,B
COMMON/gauss1/ Ages
real*8 AgesB(300)

c Berechnung Determinante
c=====
real*8 BB(300,300),CC(300,300)
real*8 Ages(300,300),zeile(300)

C A = NAME DER LINKEN SEITE
C B = NAME DER RECHTEN SEITE
C M = ANZAHL DER UNBEKANNTEN
C N = ANZAHL DER RECHTEN SEITEN

cpri
print*, 'SUB TESTMATRIX2'

c Multiplikation Ages*B
do 91 i=1,M
sum=0.
do 92 j=1,M
sum=sum + Ages(i,j)*B(j,1)
92 continue
AgesB(i)=sum
91 continue

c Multiplikation B * AgesB
Q=0.
do 93 k=1,M
Q=Q + B(k,1)*AgesB(k)
93 continue

cpri
print*, ' '
print*, ' Q=Bt * Ages * B =',Q

c Null setzen
do 66 i=1,M
do 67 k=1,M
BB(i,k)=0.
CC(i,k)=0.
67 continue
66 continue

C=====
C nach Dreieckszerlegung b ii-Werte,
c so dass Determinante det=Produkt b ii

c Uebergabe 1. Zeile ==> zeile
do 873 k=1,M
zeile(k)=Ages(1,k)
873 continue

iinvers=0
do 1 i=1,M
izeil=i

111 continue
do 2 k=1,M
if(i.gt.1) go to 20
222 continue
c i=1 1. Zeile b lk = a lk
c-----
c Frage: a 11 = Null ?
c falls ja ==> Zeilentausch bis Ages(1,1) ne 0.
cpri
cc do 55 lll=1,M
cc print*,i,' ',lll,' Ages(i,k)=' ,Ages(i, lll)
cc 55 continue
c if(Ages(1,1).ne.0.) go to 1
c welches Ages(jjj,1) ne 0. ??
do 741 jjj=i+1,M
if(Ages(jjj,1).ne.0.) go to 742
741 continue
742 continue
c falls i ne jjj: Zeilentausch i mit jjj
CALL ZEILTAUSCH(M,i,jjj)
iinvers=iinvers+1
ctest go to 222
go to 1

20 continue
c i>1 i. Zeile b ik bzw. c ik
c-----
c Berechnung c ik k<i b ik k>i
if(k.lt.i) nsum=k-1
if(k.ge.i) nsum=i-1
cpri
cc print*, 'i=',i,' k=',k,' nsum=',nsum
sum=Ages(izeil,k)
cpri
cc print*, 'sum=',sum
if(nsum.eq.0) go to 22
do 21 nn=i,nsum
cpri
cc print*, ' nn=',nn
cc print*, ' Ages(i,nn)=' ,Ages(i, nn), '
Ages(nn,k)=' ,Ages(nn,k)
sum=sum-zeile(nn)*Ages(nn,k)

cpri
print*, 'sum=',sum
cc 21 continue
22 continue

c Frage: b kk = Null ?
c falls ja ==> Zeilentausch bis b kk ne 0.
c-----
if(k.lt.i) zeile(k)=sum/Ages(k,k)
if(k.ge.i) zeile(k)=sum
c=====

cpri
cc print*,i,' ',izeil,izeil,' ',k,' zeile(k)=' ,zeile(k)
2 continue

c falls Hauptdiagonale zeile(i) ne 0. ==> fertig
if(zeile(i).ne.0.) go to 4711

c b kk-Glied ist Null
izeil=izeil+1
if(izeil.gt.M) go to 999
go to 111

4711 continue
c falls i ne izeil: Zeilentausch i mit izeil
if(i.ne.izeil) CALL ZEILTAUSCH(M,i,izeil)
if(i.ne.izeil) iinvers=iinvers+1
c Uebergabe von zeile ==> S
do 4712 jj=1,M
Ages(i,jj)=zeile(jj)
4712 continue

c Zeile i fertig
1 continue

c Uebergabe Matrizen ==> BB, CC
do 341 i=1,M
do 342 j=1,M
if(j.ge.i) BB(i,j)=Ages(i,j)
if(j.lt.i) CC(i,j)=Ages(i,j)
342 continue
341 continue

c Testausdruck Testmatrizen
cpri
cc print 401
cc write(66,401)
cc 401 format(/,'Matrix BB',/)
cc do 402 i=1,M
cpri
cff print 403,(BB(i,j),j=1,M)
cff write(66,403) (BB(i,j),j=1,M)
cff 403 format(5E12.4)
cc print 4053,i, BB(i,i)
cc write(66,4053) i, BB(i,i)
cc 4053 format(I5,E12.4)
cc 402 continue

cpri
cc print 501
cc write(66,501)
cc 501 format(/,'Matrix CC',/)
cc do 502 i=1,M
cpri
cc print 503,(CC(i,j),j=1,M)
cc write(66,503) (CC(i,j),j=1,M)
cc 503 format(5E12.4)
cc 502 continue

c Determinante aus Diagonalen b ii
dett=1.
do 47 i=1,M
dett=dett*BB(i,i)
47 continue
dett=dett*(-1)**(iinvers)

cpri
print*, ' '
print*, ' DETERMINANTE dett =',dett
print*, ' iinvers =',iinvers
print*, ' '

c Ausgabe aus Testmatrix2
write(66,9531) Q,dett,iinvers
9531 format(/,' Q = Bt * A * B =',E14.6,/,
a ' Determinante dett =',E14.6,/,
b ' iinvers =',I5)

RETURN

999 continue
cpri
print*, 'MATRIX singulaer !!!'
print*, 'i=',i,' itausch=',itausch
stop
END

```

```

real*8 zzeil1(5), zzeil2(5)
common/besetzen1/zzeil1, wert1
common/besetzen2/zzeil2, wert2
common/indeexx/index

SUBROUTINE ZEILTAUSCH(M,i,itausch)
C Zeilentausch bei Matrix Ages
implicit real*8(A-H,O-Z)

real*8 Ages(300,300)
COMMON/gauss1/Ages

C M Zahl der DOF
C i zu tauschende Zeile i==>itausch (da b ii =
0.)
C itausch getauschte Zeile itausch==>i

if(itausch.gt.M) go to 777

do 1 j=1,M
tausch = Ages(i,j)
Ages(i,j) = Ages(itausch,j)
Ages(itausch,j) = tausch
1 continue

c jetzt ist Zeile i mit itausch getauscht !!

cpri
cc print*, 'SUB ZEILTAUSCH i=', i, ' itausch=', itausch

RETURN

777 continue
cpri
print*, 'SUB ZEILTAUSCH '
print*, 'i=', i, ' itausch=', itausch
stop

END

=====
subroutine nom(vp,vf,vg)
c Bestimmung des Gesamtnamens

character*1 vp(80),vf(80),vg(160),frage,null

null = ' '
frage = '?'

c null setzen vg
c-----
do 55 i=1,160
vg(i)=null
55 continue

c Laenge von pfadname
c-----
i1=0
1 continue
i1=i1+1
if(vp(i1).eq.frage) go to 2
go to 1

c '?' gefunden
2 continue
np=i1-1

c Laenge von filename
c-----
i1=0
11 continue
i1=i1+1
if(vf(i1).eq.frage) go to 12
go to 11

c '?' gefunden
12 continue
nf=i1-1

c Eintrag Gesamtname
c-----
do 20 i=1,np
vg(i)=vp(i)
continue

do 21 i=1,nf
ii=i+np
vg(ii)=vf(i)
21 continue

return
end

=====
subroutine besetzen_zeile1_ws(i)
implicit real*8(A-H,O-Z)

real*8 A(300,300),B(300,2)
COMMON/gauss/ M,N,A,B
COMMON/hhh/xw,xws,xd,xz,aalang,messw,messws,messd,mzzz,
a w,ws,ebs,z,bb,ggs,gg,ww,wws,ee

COMMON/hhh1/fpol, isysw, ipolart
COMMON/hhh2/pl1,plr,ak2,ak2t
COMMON/hhh3/NFELD,MM1,M1
COMMON/hhh4/Feld
COMMON/hhh5/irandges, irandw, irandws, irandw2s
COMMON/hhh6/jrandw, jrandws, jrandw2s,
a aaxxw, aaxxws, aaxxw2s, ccw, ccws, ccw2s

character*1 Feld(80)
real*8 bb(5,5),ggs(5,5),gg(5,5),ww(5),wws(5),ee(5)
real*8 xw(10,50),xws(10,50),xd(10,50),xz(10,50),
a w(10,50),ws(10,50),ebs(10,50),z(10,50)

real*8 aalang(10)
integer messw(20),messws(20),messd(20),mzzz(20),
a jrandw(20),jrandws(20),jrandw2s(20)
real*8 aaxxw(20),aaxxws(20),aaxxw2s(20),
a ccw(20),ccws(20),ccw2s(20)

real*8 ak2(3,10),ak2t(10,3)
real*8 wneu(10,50,3),wstri(10,50,3),w2stri(10,50,3)
a,wsneu(10,50,3),ebsneu(10,50,3)
real*8 fpol(6,6),pl1(3,5),plr(3,5)
real*8 hhw(5),hhstri(5),hh2stri(5),eb2stri(5)

COMMON/hhh1a/ksysw,kpolart
integer ksysw(50),kpolart(50)

real*8 xx,zz

c Null setzen
do 1 kk=1,5
zzeil1(kk)=0.
zzeil2(kk)=0.
1 continue

C i ist Feld-Nr.

c bei Last 1. Wert bei Grund 1. Zeile
c mit w besetzen !!
c-----
c 1. gibt es Randbed. w im Feld i ==> besetzen
c 2. gibt es Werte w im Feld i ==> besetzen
c falls 1.+2. nein ==> stop
c-----

c 1. Bedingung
if(irandw.eq.0) go to 2140
do 2141 jj=1,irandw
if(jrandw(jj).eq.i) go to 2250
2141 continue
go to 2140

2250 continue
c Randbedingung im Feld i fuer w gefunden
calt w_wert = aaxw(jj)
calt si_wert = ccw(jj)
w_wert = ccw(jj)
si_wert = aaxw(jj)

si_wert2=si_wert*si_wert
si_wert3=si_wert*si_wert2
si_wert4=si_wert*si_wert3

go to 2260

2140 continue
c 2. Bedingung
do 2143 jj=1,NFELD
if(jj.eq.i.and.messw(jj).ne.0) go to 2255
2143 continue

c-----
c hier beide Bedingungen nicht erfuehlt
cpri
print*, 'subroutine besetzen_zeile1_ws(i)'
print*, 'Feld i=', i, ' index=', index
print*, 'keine Bedingung w gefunden ==> stop'
stop

2255 continue
c w im Feld i fuer w gefunden
w_wert = w(jj,1)
si_wert = xw(jj,1)/aalang(jj)

si_wert2=si_wert*si_wert
si_wert3=si_wert*si_wert2
si_wert4=si_wert*si_wert3

go to 2260

```

```

2260 continue
cpri
cv      print*,'in sub besetzen_zeile1_ws(i)'
cv      print*,'w_wert =',w_wert
cv      print*,'si_wert =',si_wert

zzeill(1)=1.
zzeill(2)=si_wert
zzeill(3)=si_wert2
zzeill(4)=si_wert3
zzeill(5)=si_wert4

wert1=w_wert

cpri
cv      print*,'zzeill'
cv      print 3l,(zzeill(klmn),klmn=1,5)
cv 31   format(5E15.6)
cv      print*,'wert1=',wert1
cv      print*,'ende sub besetzen_zeile1_ws(i) '

return
end

subroutine besetzen_zeilelu2_w2s(i)
implicit real*8(A-H,O-Z)

real*8 zzeill(5),zzeill2(5)
common/besetzen1/zzeill,wert1
common/besetzen2/zzeill2,wert2
common/indeexx/index

real*8 A(300,300),B(300,2)
COMMON/gauss/ M,N,A,B
COMMON/hhh/xw,xws,xd,xz,aalang,messw,messws,messd,mzzz,
a      w,ws,ebs,z,bb,ggs,gg,ww,wws,ee

COMMON/hhh1/fpol,isysw,ipolart
COMMON/hhh2/pll,plr,ak2,ak2t
COMMON/hhh3/NFELD,MM1,M1
COMMON/hhh4/Feld
COMMON/hhh5/irandges,irandw,irandws,irandw2s
COMMON/hhh6/jrandw,jrandws,jrandw2s,
a      aaxxw,aaxxws,aaxw2s,ccw,ccws,ccw2s

character*1 Feld(80)
real*8 bb(5,5),ggs(5,5),gg(5,5),ww(5),wws(5),ee(5)
real*8 xw(10,50),xws(10,50),xd(10,50),xz(10,50),
a      w(10,50),ws(10,50),ebs(10,50),z(10,50)

real*8 aalang(10)
integer messw(20),messws(20),messd(20),mzzz(20),
a      jrandw(20),jrandws(20),jrandw2s(20)
real*8 aaxxw(20),aaxxws(20),aaxw2s(20),
a      ccw(20),ccws(20),ccw2s(20)

real*8 ak2(3,10),ak2t(10,3)
real*8 wneu(10,50,3),wstri(10,50,3),w2stri(10,50,3)
a,wsneu(10,50,3),ebsneu(10,50,3)
real*8 fpol(6,6),pll(3,5),plr(3,5)
real*8 hhw(5),hhstri(5),hh2stri(5),eb2stri(5)

COMMON/hhh1a/ksysw,kpolart
integer ksysw(50),kpolart(50)

real*8 xx,xz

ikz=0

c      Null setzen
do 1 kk=1,5
zzeill(kk)=0.
zzeill2(kk)=0.
1      continue

C      i      ist Feld-Nr.

c      1. Zeile
c      bei Last 1. Wert      bei Grund 1. Zeile
c      mit w besetzen !!

c      2. Zeile
c      bei Last 1. Wert      bei Grund 1. Zeile
c      mit w oder w' besetzen !!

-----
c      1. gibt es Randbed. w im Feld i ==> besetzen
c      2. gibt es Werte w im Feld i ==> besetzen
c      falls 1.+2. nein ==> w' im Feld i ==> besetzen
-----

C      1. Zeile
C=====
c      1. Bedingung
if(irandw.eq.0) go to 2140
do 2141 jj=1,irandw
if(jrandw(jj).eq.i) go to 2250
2141 continue
go to 2140

2250 continue
jjrandalt=jj
Randbedingung im Feld i fuer w gefunden
w_wert = ccw(jj)
si_wert = aaxxw(jj)

si_wert2=si_wert*si_wert
si_wert3=si_wert*si_wert2
si_wert4=si_wert*si_wert3

go to 2260

2140 continue
c      2. Bedingung
do 2143 jj=1,NFELD
if(jj.eq.1.and.messw(jj).ne.0) go to 2255
2143 continue

-----
c      hier beide Bedingungen nicht erfuehlt
cpri
print*,'subroutine besetzen_zeilelu2_w2s(i)'
print*,'Feld i=',i,' index=',index
print*,'keine Bedingung w gefunden ==> stop'
stop

2255 continue
c      w im Feld i fuer w gefunden

ikz=1

w_wert = w(jj,1)
si_wert = xw(jj,1)/aalang(jj)

si_wert2=si_wert*si_wert
si_wert3=si_wert*si_wert2
si_wert4=si_wert*si_wert3

go to 2260

2260 continue
cpri
cv      print*,'in sub besetzen_zeilelu2_w2s(i)'
cv      print*,'Zeile 1 w'
cv      print*,'w_wert =',w_wert
cv      print*,'si_wert =',si_wert

zzeill(1)=1.
zzeill(2)=si_wert
zzeill(3)=si_wert2
zzeill(4)=si_wert3
zzeill(5)=si_wert4

wert1=w_wert

cpri
print*,'zzeill'
print 3l,(zzeill(klmn),klmn=1,5)
31   format(5E15.6)
print*,'wert1=',wert1
print*,'ende sub besetzen_zeile1_ws(i)'
print*,' '

C      2. Zeile
C=====
c      gibt es einen 2. w-Wert in Feld i ?

c      1. Bedingung
if(irandw.eq.0) go to 3140
do 3141 jj=1,irandw
if(jrandw(jj).eq.i.and.jj.ne.jjrandalt) go to 3250
3141 continue
go to 3140

3250 continue
c      Randbedingung im Feld i fuer w gefunden
w_wert = ccw(jj)
si_wert = aaxxw(jj)

si_wert2=si_wert*si_wert
si_wert3=si_wert*si_wert2
si_wert4=si_wert*si_wert3

go to 3260

3140 continue
c      2. Bedingung
do 3143 jj=1,NFELD
if(jj.eq.1.and.messw(jj).gt.1) go to 3255
3143 continue

-----
c      hier beide Bedingungen 2. Zeile w nicht erfuehlt
-----
c      hier beide Bedingungen w nicht erfuehlt
cpri
print*,'subroutine besetzen_zeilelu2_w2s(i)'
print*,'Feld i=',i,' index=',index
print*,'keine 2. w Bedingung gefunden ==> Winkel'
go to 777

3255 continue
c      w im Feld i fuer w gefunden

ikz=ikz+1

```

```

w_wert = w(jj,ikz)
si_wert = xw(jj,ikz)/aalang(jj)

si_wert2=si_wert*si_wert
si_wert3=si_wert*si_wert2
si_wert4=si_wert*si_wert3

go to 3260

3260 continue
cpri
cv print*, 'in sub besetzen_zeilelu2_w2s(i)'
cv print*, 'Zeile 2 w'
cv print*, 'w_wert =', w_wert
cv print*, 'si_wert =', si_wert

zzeil2(1)=1.
zzeil2(2)=si_wert
zzeil2(3)=si_wert2
zzeil2(4)=si_wert3
zzeil2(5)=si_wert4

wert2=w_wert
cpri
print*, 'zzeil2'
print 31, (zzeil2(klmn), klmn=1,5)
print*, 'wert2=', wert2
print*, 'ende sub besetzen_zeilelu2_w2s(i)'

return

subroutine suchen_anfangm1(i,iwa)
fuer Ausdruck: suchen Anfangsnr.-1 =iwa
implicit real*8 (A-H,O-Z)

integer iafeld(100,7)
common/adressfeld/iafeld,izges

c i = Feld-Nr.

c suchen 1. in w (iafeld(k,1))
c suchen 2. in w' (iafeld(k,1))
c suchen 3. in w" (iafeld(k,1))

c iwa=-100 dann Feld nicht besetzt !!!!
c dann weiter zum nächsten Feld

c 1. Suche nach w
do 1 k=1,izges
if(iafeld(k,1).eq.1.and.iafeld(k,2).eq.i) go to 100
continue
1 nicht gefunden bei w

c 2. Suche nach w'
do 2 k=1,izges
if(iafeld(k,1).eq.2.and.iafeld(k,2).eq.i) go to 100
continue
2 nicht gefunden bei w'

c 3. Suche nach w"
do 3 k=1,izges
if(iafeld(k,1).eq.3.and.iafeld(k,2).eq.i) go to 100
continue
3 nicht gefunden bei w"

c dann ist das Feld i nicht gemessen
c-----
iwa=-100
return

c Feld i gefunden
c-----
100 continue
iwa= iafeld(k,4) - 1
return
end

subroutine suchen_elnr(ielnr,nspall,MMw,MMws,jende)
implicit real*8 (A-H,O-Z)

integer iafeld(100,7)
common/adressfeld/iafeld,izges

c ielnr ist Element-Nr.

c Element in w vorhanden?
nw=0
do 1511 kkjj=1,izges
if(iafeld(kkjj,1).eq.1) nw=nw+1
if(iafeld(kkjj,1).eq.1.and.iafeld(kkjj,2).eq.ielnr) go to
1571
1511 continue
c nicht in w gefunden

c Element in w' vorhanden?
nws=0
do 1521 kkjj=1,izges
if(iafeld(kkjj,1).eq.2) nws=nws+1
if(iafeld(kkjj,1).eq.2.and.iafeld(kkjj,2).eq.ielnr) go to
1572
1521 continue
c nicht in w' gefunden

c Element in w" vorhanden?
nw2s=0
do 1531 kkjj=1,izges
if(iafeld(kkjj,1).eq.3) nw2s=nw2s+1
if(iafeld(kkjj,1).eq.3.and.iafeld(kkjj,2).eq.ielnr) go to
1573
1531 continue
c nicht in w" gefunden

w_wert = w(jj,ikz)
si_wert = xw(jj,ikz)/aalang(jj)

si_wert2=si_wert*si_wert
si_wert3=si_wert*si_wert2
si_wert4=si_wert*si_wert3

go to 3260

3260 continue
cpri
cv print*, 'in sub besetzen_zeilelu2_w2s(i)'
cv print*, 'Zeile 2 w'
cv print*, 'w_wert =', w_wert
cv print*, 'si_wert =', si_wert

zzeil2(1)=1.
zzeil2(2)=si_wert
zzeil2(3)=si_wert2
zzeil2(4)=si_wert3
zzeil2(5)=si_wert4

wert2=w_wert
cpri
print*, 'zzeil2'
print 31, (zzeil2(klmn), klmn=1,5)
print*, 'wert2=', wert2
print*, 'ende sub besetzen_zeilelu2_w2s(i)'

return

777 continue
c keinen 2. w - Wert gefunden
c Winkelneigung suchen
c-----
c 2. Zeile
c bei Last 2. Wert bei Grund 2. Zeile
c mit w' besetzen !!

c-----
c 1. gibt es Randbed. w' im Feld i ==> besetzen
c 2. gibt es Werte w' im Feld i ==> besetzen
c falls 1.+2. nein ==> stop
c-----

C 2. Zeile
C-----
c 1. Bedingung
if(irandws.eq.0) go to 4140
do 4141 jj=1,irandws
if(jrandws(jj).eq.i) go to 4250
4141 continue
go to 4140

4250 continue
c Randbedingung im Feld i fuer w' gefunden
w_wert = ccws(jj)
si_wert = aaxxws(jj)

si_wert2=si_wert*si_wert
si_wert3=si_wert*si_wert2
si_wert4=si_wert*si_wert3

go to 4260

4140 continue
c 2. Bedingung
do 4143 jj=1,NFELD
if(jj.eq.1.and.messws(jj).ne.0) go to 4255
4143 continue

c-----
c hier beide Bedingungen nicht erfuehlt
cpri
print*, 'subroutine besetzen_zeilelu2_w2s(i)'
print*, 'Feld i=', i, index=', index
print*, 'keine Bedingung ws gefunden ==> stop'
stop

4255 continue
c w im Feld i fuer w gefunden
w_wert = ws(jj,1)
si_wert = xws(jj,1)/aalang(jj)

si_wert2=si_wert*si_wert
si_wert3=si_wert*si_wert2
si_wert4=si_wert*si_wert3

go to 4260

4260 continue
aal=aalang(i)
cpri
cv print*, 'in sub besetzen_zeilelu2_w2s(i)'
cv print*, 'Zeile 2 ws'
cv print*, 'w_wert =', w_wert
cv print*, 'si_wert =', si_wert

zzeil2(1)=0.
zzeil2(2)=1. /aal

```

```
cpri
  print*, 'Element ielnr=', ielnr,
  a' nicht in iafeld gefunden ==> stop'

1571 continue
  nspall = (nw-1)*jende
  go to 1574
1572 continue
  nspall = MMw + (nws-1)*jende
  go to 1574
1573 continue
  nspall = MMws + (nw2s-1)*jende
  go to 1574

1574 continue
  return
  end
```