

1

Einleitung

Dieses Kapitel hat vor allem zwei Ziele: Es soll zum einen die Leserinnen und Leser mit der Notation vertraut machen, die in diesem Buch durchgängig verwendet wird. Zum anderen werden wichtige Grundlagen aus der Mathematik (insbesondere der Linearen Algebra) und der Technischen Mechanik wiederholt. Was die Notation anbelangt, werden wir in der Regel den Matrix-Vektor-Formalismus benutzen. Die Tensorschreibweise – mit Indizes oder direkt – wird seltener benötigt. Für diejenigen, die mit der Tensorschreibweise weniger vertraut sind, wurde ein kleiner Abschnitt hierüber angefügt. Zunächst aber stellen wir ein anschauliches Beispiel für nichtlineares Verhalten in einem Bauteil vor – ein einfaches Trägerelement –, das sich analytisch lösen lässt. Dabei werden die verschiedenen, in diesem Buch beschriebenen Techniken zur Behandlung nichtlinearer Prozesse vorgestellt; ebenso wird das korrekte Lösen der sich daraus ergebenden Anfangs- bzw. Randwertprobleme beschrieben.

1.1

Ein einfaches Beispiel für nichtlineares Verhalten

Viele Eigenschaften von Lösungsverfahren lassen sich anhand eines einfachen Trägerelements – eventuell in Kombination mit Federn – diskutieren. Nichtlinearitäten können in diesem Fall sowohl von geometrischen als auch von Material-Nichtlinearitäten herrühren. Wir nehmen in diesem Abschnitt an, dass die Verschiebungen und Rotationen beliebig groß sein können, die auftretenden Dehnungen jedoch klein bleiben, z. B. kleiner als 5 %. Diese Beschränkung werden wir erst in Teil IV des Buches aufheben und dann die Diskussion auf beliebig große elastische und inelastische Dehnungen ausweiten.

Wenden wir uns nun dem flachen Trägerelement in Abb. 1.1 zu. Für den verformten Zustand erhalten wir aus elementaren Gleichgewichtsbetrachtungen den folgenden Ausdruck für die wirkenden Kräfte in einer der zueinander symmetrischen Hälften des flachen Trägers:

$$F_{\text{int}} = -A\sigma \sin \phi - F_{\text{Fed}} \quad (1.1)$$

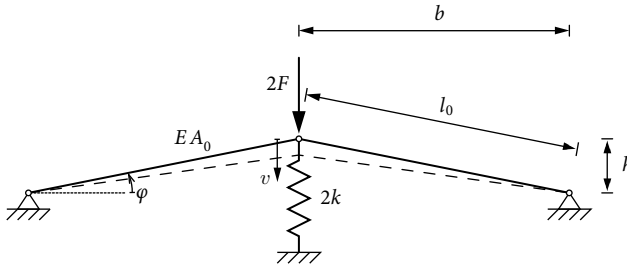


Abb. 1.1 Ebenes flaches Trägerelement.

Dabei steht σ für die axiale Spannung im Element, F_{Fed} ist die halbe auf die Feder wirkende Kraft und ϕ der Winkel, den das verformte Trägerelement mit der Horizontalen einnimmt. Unter der oben angesprochenen Annahme kleiner Dehnungen kann die Differenz zwischen der Querschnittsfläche A im verformten Zustand und A_0 im Ursprungszustand vernachlässigt werden. Aus dem gleichen Grund können wir den Unterschied zwischen der Länge des Balkens im Ursprungszustand

$$\ell_0 = \sqrt{b^2 + h^2} \quad (1.2)$$

und der Balkenlänge im verformten Zustand

$$\ell = \sqrt{b^2 + (h - v)^2} \quad (1.3)$$

im Nenner des Ausdrucks für die Dehnung ϵ vernachlässigen:

$$\epsilon = \frac{\ell - \ell_0}{\ell_0} \quad (1.4)$$

Der Neigungswinkel ϕ errechnet sich entsprechend näherungsweise

$$\sin \phi = \frac{h - v}{\ell} \approx \frac{h - v}{\ell_0} \quad (1.5)$$

Die Strecken b und h sind in Abb. 1.1 definiert. Die vertikale Verschiebung v soll bei einer Bewegung nach unten positiv gezählt werden. Die halbe Kraft auf die Feder ergibt sich dann zu

$$F_{\text{Fed}} = -kv \quad (1.6)$$

mit der Federkonstante k und der axialen Spannung σ im Balken, für die gilt:

$$\sigma = E\epsilon \quad (1.7)$$

E ist der Elastizitätsmodul. Einsetzen der Ausdrücke für σ , die Federkraft F_{Fed} und den Winkel ϕ in die Gleichgewichtsbedingung (1.1) ergibt

$$F_{\text{int}}(v) = -EA_0 \sin \phi \frac{\ell - \ell_0}{\ell_0} + kv \quad (1.8)$$

Gleichung (1.8) beschreibt die interne (innere) Kraft in der Struktur als eine nicht-lineare Funktion der vertikalen Verschiebung v . Normalerweise ist die externe (äußere) Kraft zur Zeit $t + \Delta t$, $F_{\text{ext}}^{t+\Delta t}$, gegeben. Die Verschiebung v muss dann aus

$$F_{\text{ext}}^{t+\Delta t} - F_{\text{int}}^{t+\Delta t} = 0 \quad (1.9)$$

bestimmt werden. Dies geschieht iterativ, beispielsweise mit dem Newton-Verfahren (auch Newton-Raphson-Verfahren):

$$F_{\text{ext}}^{t+\Delta t} = F_{\text{int}}(v_j) + \frac{dF_{\text{int}}}{dv} dv + \frac{1}{2} \frac{d^2 F_{\text{int}}}{dv^2} dv^2 + \mathcal{O}(dv^3) \quad (1.10)$$

In linearer Näherung beträgt die iterative Korrektur von v im j -ten Iterationsschritt

$$dv = \left(\frac{dF_{\text{int}}}{dv} \right)_j^{-1} \left(F_{\text{ext}}^{t+\Delta t} - F_{\text{int}}(v_j) \right) \quad (1.11)$$

Die Iteration endet, wenn ein Konvergenzkriterium $\|F_{\text{ext}}^{t+\Delta t} - F_{\text{int}}(v_j)\| < \varepsilon$ erfüllt ist (ε ist eine kleine Zahl). Im vorliegenden Fall können wir die Ableitung $\frac{dF_{\text{int}}}{dv}$ (in der Sprache der numerischen Mechanik: den Tangentenmodul) aus (1.8) folgendermaßen bestimmen:

$$\frac{dF_{\text{int}}}{dv} = \frac{A_0 \sin^2 \phi}{\ell_0} \left(E + \frac{dE}{d\ell} (\ell - \ell_0) \right) + \left(k + \frac{dk}{dv} v \right) + \frac{A_0 \sigma}{\ell_0} \quad (1.12)$$

Dabei haben wir den allgemeinen Fall angenommen, dass sowohl die Steifigkeit des Trägerelements als auch die der Feder von der Auslenkung abhängt. Ohne diese sog. Material-Nichtlinearität verschwinden die Terme mit $\frac{dE}{d\ell}$ und $\frac{dk}{dv}$. Der letzte Term in (1.12) beschreibt die Effekte großer Auslenkungen oder Rotationen (geometrische Nichtlinearität) und ist linear in der Spannung. Dieser Term ist von entscheidender Bedeutung, wenn die Stabilität von schlanken Konstruktionen berechnet werden soll. Abb. 1.2 zeigt das Verhalten des Trägerelements bei verschiedenen Werten der Federkonstante k . Die Kurven ergeben sich direkt aus der analytischen Formel für die interne Kraft (1.8) und der Gleichgewichtsbedingung (1.9). Das Iterationsverfahren kann nur angewendet werden, wenn die Federkonstante k groß genug ist, d. h., wenn die Last-Verschiebungs-Kurve kein lokales Maximum besitzt.

1.2

Wiederholung: Grundlagen der Linearen Algebra

Bei der rechnergestützten Behandlung der Mechanik von Festkörpern tauchen sehr häufig Vektoren und Matrizen auf. Unter einem Vektor versteht man dabei ein eindimensionales Feld von Skalaren. Ein Skalar wiederum ist eine physikalische Größe, deren Wert vom jeweiligen Bezugssystem unabhängig ist. Skalare bezeichnen wir mit kursiven Symbolen, Vektoren mit fett-kursiven Kleinbuchstaben.

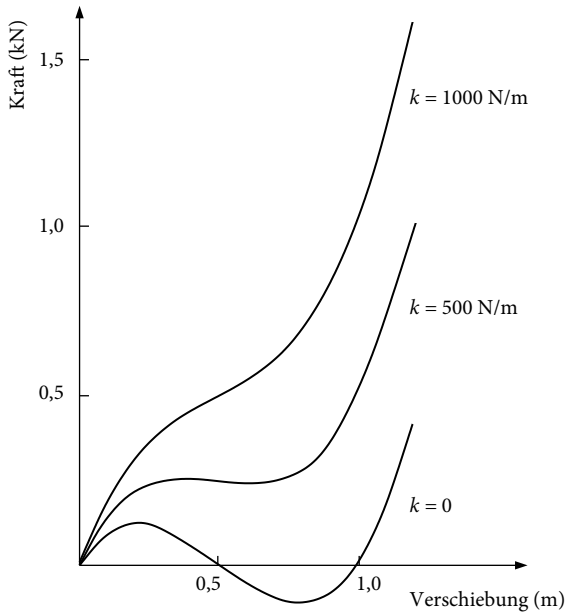


Abb. 1.2 Kraft-Verschiebungs-Diagramm für das flache Trägerelement mit unterschiedlichen Werten der Federkonstante k ($b = 10\text{ m}$, $h = 0,5\text{ m}$ und $EA_0 = 5\text{ MN/m}^2$).

Der Vektor \boldsymbol{v} hat somit n skalare Komponenten v_1, \dots, v_n :

$$\boldsymbol{v} = \begin{pmatrix} v_1 \\ \dots \\ \dots \\ v_n \end{pmatrix} \quad (1.13)$$

In (1.13) haben wir die Komponenten in Spaltenform notiert (Spaltenvektor). Alternativ ist auch die Schreibweise als Zeilenvektor möglich. Dieser entsteht aus einem Spaltenvektor durch Transposition, der transponierte Vektor wird wie folgt geschrieben:

$$\boldsymbol{v}^T = (v_1, \dots, v_n)$$

Die Addition von Vektoren ist definiert als Addition der Komponenten;

$$\boldsymbol{w} = \boldsymbol{u} + \boldsymbol{v} \quad (1.14)$$

bedeutet also, dass $w_i = u_i + v_i$ für alle $i = 1, \dots, n$. Die Multiplikation eines Vektors mit einem Skalar λ ist definiert durch

$$\boldsymbol{w} = \lambda \boldsymbol{u} \quad (1.15)$$

mit den Komponenten $w_i = \lambda u_i$.

Für die Multiplikation zweier Vektoren gibt es verschiedene Definitionen. Beim inneren Produkt (Skalarprodukt) werden zwei Vektoren mit jeweils n Komponenten multipliziert, indem die Produkte korrespondierender Komponenten addiert werden; das Skalarprodukt der n -komponentigen Vektoren \mathbf{u} und \mathbf{v} ist somit definiert als

$$\mathbf{u}^T \mathbf{v} = \sum_{i=1}^n u_i v_i \quad (1.16)$$

Das Skalarprodukt genügt dem Kommutativgesetz, d. h., es gilt $\mathbf{u}^T \mathbf{v} = \mathbf{v}^T \mathbf{u}$, wie Sie an der Definition dieses Produkts direkt ablesen können. Mithilfe des inneren Produkts lässt sich weiterhin die Norm eines Vektors definieren. Hierfür gibt es verschiedene Möglichkeiten, in der Theoretischen und Technischen Mechanik ist die am häufigsten verwendete Definition die sog. euklidische oder L_2 -Norm:

$$\|\mathbf{v}\|_2 = \sqrt{\mathbf{v}^T \mathbf{v}} \quad (1.17)$$

Der Index „2“ wird oft weggelassen.

Das Kreuz- oder Vektorprodukt zweier Vektoren \mathbf{a} und \mathbf{b} ist kein Skalar, sondern ein Vektor \mathbf{c} :

$$\mathbf{c} = \mathbf{a} \times \mathbf{b} \quad (1.18)$$

Dieser steht orthogonal (senkrecht) auf den Vektoren \mathbf{a} und \mathbf{b} im dreidimensionalen Raum. Seine Richtung wird mit der Rechte-Hand-Regel bestimmt, daraus folgt, dass das Kreuzprodukt antisymmetrisch ist: $\mathbf{b} \times \mathbf{a} = -\mathbf{a} \times \mathbf{b}$. Die Komponenten von $\mathbf{c} = \mathbf{a} \times \mathbf{b}$ berechnen sich nach der Vorschrift

$$\mathbf{c} = \begin{pmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{pmatrix} \quad (1.19)$$

Eine weitere Art, zwei Vektoren zu multiplizieren, das sog. dyadische Produkt, wird weiter unten beschrieben, vgl. (1.25).

Mit den Komponenten eines Vektors lassen sich skalare Funktionen bilden. Beispiele aus der Mechanik sind die Invarianten des Spannungs- oder des Dehnungstensors oder die Fließfunktion in der Plastizitätstheorie. Häufig wird auch der Gradient einer Funktion gebildet, wobei man wiederum einen Vektor erhält. Betrachten wir eine skalarwertige Funktion f der Komponenten a_i eines Vektors \mathbf{a} . Dann erhält man den Gradienten \mathbf{b} durch Ableiten von f nach \mathbf{a} :

$$\mathbf{b} = \frac{\partial f}{\partial \mathbf{a}} \quad (1.20)$$

oder komponentenweise:

$$b_i = \frac{\partial f}{\partial a_i} \quad (1.21)$$

Die Operation der Gradientenbildung bewirkt, dass \mathbf{b} orthogonal auf der Hyperfläche steht, welche im n -dimensionalen Vektorraum durch $f = c$ definiert wird, wobei die Konstante c gewöhnlich gleich null gesetzt wird.

Ein ebenfalls für die numerische Mechanik sehr nützliches mathematisches Konzept sind Matrizen. Wenn man Vektoren in einer einfachen Beschreibung als eindimensionale Felder von Skalaren versteht, sind Matrizen zweidimensionale Felder von Skalaren. Man spricht daher von den m Zeilen und n Spalten einer Matrix, im Allgemeinen sind m und n verschieden. Ein m -komponentiger (Spalten-) Vektor ist in diesem Sinne eine Matrix mit nur einer Spalte, also eine $m \times 1$ -Matrix. Ein Zeilenvektor mit n Komponenten entspricht einer $1 \times n$ -Matrix.

In diesem Buch werden Matrizen in der Regel mit fett-kursiven Großbuchstaben bezeichnet. Die Komponenten (häufiger Elemente genannt) einer $m \times n$ -Matrix \mathbf{A} werden, analog zu den Vektorkomponenten, mit a_{ij} bezeichnet, mit $i = 1, \dots, m$ und $j = 1, \dots, n$.

Ein Vektor \mathbf{b} der Länge n kann mit einer $m \times n$ -Matrix \mathbf{A} wie folgt linksmultipliziert werden:

$$\mathbf{c} = \mathbf{A}\mathbf{b} \quad (1.22)$$

Der sich ergebende Vektor \mathbf{c} hat m Komponenten:

$$c_i = \sum_{j=1}^n a_{ij} b_j \quad (1.23)$$

Die Addition zweier $m \times n$ -Matrizen \mathbf{A} und \mathbf{B} zu einer Matrix \mathbf{C} erfolgt ganz analog zur Vektoraddition, also komponentenweise: $c_{ij} = a_{ij} + b_{ij}$. Auch die Multiplikation einer Matrix mit einem Skalar geschieht wie bei der skalaren Multiplikation eines Vektors für jedes Element der Matrix gleich: $c_{ij} = \lambda a_{ij}$.

Das Produkt zweier Matrizen ist ähnlich definiert wie das Produkt von einer Matrix und einem Vektor. Sei \mathbf{A} eine $m \times k$ -Matrix und \mathbf{B} eine $k \times n$ -Matrix. Das Ergebnis der Matrixmultiplikation $\mathbf{A} \cdot \mathbf{B}$ ist dann eine $m \times n$ -Matrix \mathbf{C} mit den Elementen

$$c_{ij} = \sum_{e=1}^k a_{ie} b_{ej} \quad (1.24)$$

Eine Spezialfall ergibt sich, wenn die Matrix \mathbf{A} ein Spalten- und die Matrix \mathbf{B} ein Zeilenvektor ist ($k = 1$ in (1.24)). In diesem Fall kann man die beiden Matrizen auch als Vektoren \mathbf{a} und \mathbf{b}^T schreiben. Das Produkt ist aber immer noch eine $m \times n$ -Matrix:

$$\mathbf{C} = \mathbf{a}\mathbf{b}^T \quad (1.25)$$

mit den Elementen $c_{ij} = a_i b_j$. Diese Operation nennt man das dyadische oder äußere Produkt der beiden Vektoren \mathbf{a} und \mathbf{b} .

Die Transposition einer Matrix (das Transponieren) geschieht wie oben für Vektoren beschrieben, also bedeutet $\mathbf{B} = \mathbf{A}^T$, dass $b_{ij} = a_{ji}$; mit anderen Worten:

beim Transponieren werden Spalten und Zeilen vertauscht. Bei der Herleitung von Finite-Elemente-Gleichungen kommt es öfters zur folgenden Operation: Zwei Matrizen werden multipliziert und das Produkt dann transponiert. Hierfür gilt die Beziehung

$$(AB)^T = B^T A^T \quad (1.26)$$

Der häufigste Typ von Matrizen sind quadratische Matrizen, also solche mit $m = n$. Unter bestimmten Bedingungen, die im Folgenden eingehend diskutiert werden, kann man die Inverse $B = A^{-1}$ einer Matrix definieren, sodass

$$AB = I \quad (1.27)$$

mit der Einheitsmatrix I . Für letztere gilt, dass alle Elemente verschwinden mit Ausnahme der Diagonalelemente, welche alle den Wert 1 haben: $I = \text{diag}[1, \dots, 1]$. Die Matrixinversion wird bei der Lösung großer linearer Gleichungssysteme benötigt, die sich aus der Finite-Elemente-Diskretisierung ergeben. Solche Systeme haben die Form

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n &= b_n \end{aligned} \quad (1.28)$$

Wenn man die bekannten Koeffizienten a_{11}, \dots, a_{nn} zu einer Matrix A , die bekannten Komponenten b_1, \dots, b_n zu einem Vektor \mathbf{b} und die Unbekannten x_1, \dots, x_n zu einem Vektor \mathbf{x} zusammenfasst, kann man das Gleichungssystem (1.28) sehr kompakt wie folgt schreiben:

$$A\mathbf{x} = \mathbf{b} \quad (1.29)$$

Formal ergibt sich der Vektor \mathbf{x} der Unbekannten durch Multiplikation mit der inversen Matrix von A – vorausgesetzt natürlich, dass A^{-1} existiert:

$$\mathbf{x} = A^{-1}\mathbf{b} \quad (1.30)$$

In der Mechanik von Festkörpern ist die Matrix A oft symmetrisch, d. h., $a_{ij} = a_{ji}$, was die Berechnung von A^{-1} vereinfacht. Diese Symmetrien können allerdings verloren gehen, wenn Nichtlinearitäten in die Modelle eingeführt werden.

Ein effizientes Vorgehen beim Invertieren von Matrizen ist das Zerlegen der Matrix A in ein Matrixprodukt

$$A = LDR \quad (1.31)$$

mit einer unteren Dreiecksmatrix

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ l_{21} & 1 & 0 & \dots & 0 \\ l_{31} & l_{32} & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \dots & 1 \end{bmatrix} \quad (1.32)$$

einer oberen Dreiecksmatrix

$$\mathbf{R} = \begin{bmatrix} 1 & r_{12} & r_{13} & \dots & r_{1n} \\ 0 & 1 & r_{23} & \dots & r_{2n} \\ 0 & 0 & 1 & \dots & r_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix} \quad (1.33)$$

und einer Diagonalmatrix

$$\mathbf{D} = \text{diag}[d_{11}, \dots, d_{nn}] \quad (1.34)$$

Für symmetrische Matrizen gilt die Beziehung $\mathbf{R} = \mathbf{L}^T$.

Diese LR-Zerlegung (Dreieckszerlegung) basiert auf dem Gauß'schen Eliminationsverfahren. Wenn die Matrix \mathbf{A} eine Bandmatrix ist (was in der Finite-Elemente-Analyse normalerweise der Fall ist), bleibt deren Struktur in der Regel erhalten, d. h., die Dreiecksmatrizen \mathbf{L} und \mathbf{R} haben dann ebenfalls eine Bandstruktur. Wegen

$$\mathbf{x} = (\mathbf{LDR})^{-1}\mathbf{b} = \mathbf{R}^{-1}(\mathbf{LD})^{-1}\mathbf{b} = \mathbf{R}^{-1}\mathbf{D}^{-1}\mathbf{L}^{-1}\mathbf{b}$$

lässt sich das System nach \mathbf{x} auflösen:

$$\begin{aligned} \mathbf{c} &= \mathbf{L}^{-1}\mathbf{b} \\ \mathbf{d} &= \mathbf{D}^{-1}\mathbf{c} \\ \mathbf{x} &= \mathbf{R}^{-1}\mathbf{d} \end{aligned} \quad (1.35)$$

Diese Gleichung führt uns auf eine weitere interessante Beobachtung. Während die Ausdrücke $\mathbf{L}^{-1}\mathbf{b}$ und $\mathbf{R}^{-1}\mathbf{d}$ nur Multiplikationen enthalten und uns daher keine arithmetischen Schwierigkeiten bereiten können, stecken in dem Term $\mathbf{D}^{-1}\mathbf{c}$ Divisionen, denn $\mathbf{D}^{-1} = \text{diag}[d_{11}^{-1}, \dots, d_{nn}^{-1}]$. Wenn nun eines der Diagonalelemente von \mathbf{D} , auch Pivotelemente genannt, verschwindet, kann \mathbf{x} nicht mehr berechnet werden. In diesem Fall nennt man \mathbf{A} eine singuläre Matrix und es existiert keine eindeutige LR-Zerlegung mehr. Wir betrachten nun drei Fälle: Alle Pivotelemente von \mathbf{D} sind positiv, ein oder mehrere Pivotelemente von \mathbf{D} sind null oder ein oder mehrere Pivotelemente von \mathbf{D} sind negativ. Wenn alle Diagonalelemente positiv sind, heißt \mathbf{A} positiv definit. Ein Beispiel ist die Steifigkeitsmatrix \mathbf{A} ,

die sich aus der verschiebungsbasierten Finite-Elemente-Diskretisierung eines linear-elastischen Körpers ergibt. Für positiv definite Matrizen ist die LR-Zerlegung eindeutig und auftretende Rundungsfehler werden bei der Berechnung nicht verstärkt. Wenn nichtlineare Effekte auftreten, kann die tangentiale Steifigkeitsmatrix A singular (ein oder mehrere verschwindende Pivotelemente) oder indefinit werden (ein oder mehrere negative Pivotelemente). Wie oben dargestellt kann eine singuläre Matrix nicht zerlegt werden und die Berechnung führt auf keine physikalisch sinnvollen Ergebnisse. Wenn es allerdings zwar negative, aber keine verschwindenden Pivotelemente gibt, ist eine eindeutige LR-Zerlegung doch möglich. In jedem Fall kann man bei indefiniten Matrizen nicht ausschließen, dass Rundungsfehler, die bei der Zerlegung auftreten, bei der weiteren Berechnung verstärkt werden. Für die nichtlineare Finite-Elemente-Analyse bedeutet dies, dass die iterativen Algorithmen, mit denen die auftretenden nichtlinearen algebraischen Gleichungen gelöst werden, möglicherweise divergieren.

Die Frage, ob eine Matrix singular ist, hängt eng mit der Determinante dieser Matrix zusammen. Die Determinante einer Matrix ist definiert gemäß (Golub und van Loan, 1983; Noble und Daniel, 1969; Ortega, 1987; Saad, 1996)

$$\det A = \sum_{j=1}^n (-1)^{i+j} a_{ij} \det A_{ij} \quad (1.36)$$

wobei A_{ij} eine $(n-1) \times (n-1)$ -Matrix ist, die entsteht, wenn man die i -te Zeile und die j -te Spalte von A entfernt. Diese rekursive Beziehung schließt sich mit $\det A = a_{11}$ für $n=1$. Eine nützliche Eigenschaft der Determinante ist $\det(AB) = \det A \cdot \det B$. Mit (1.31) haben wir $\det A = \det L \cdot \det D \cdot \det R$, woraus wir mit Definition (1.36) ableiten, dass $\det L = \det R = 1$. Es ergibt sich der nützliche Zusammenhang

$$\det A = \prod_{i=1}^n d_i \quad (1.37)$$

woraus direkt folgt, dass die Determinante einer Matrix gleich null ist, wenn ein oder mehrere Pivotelemente null sind. In diesem Fall ist die Matrix singular.

Ebenfalls praktisch ist der folgende Satz über das Invertieren eines speziellen Typs von Matrizen, bekannt als Sherman-Morrison-Formel. Es seien A eine nicht singuläre $n \times n$ -Matrix und \mathbf{u} und \mathbf{v} zwei n -komponentige Vektoren. Dann gilt

$$(A + \mathbf{u}\mathbf{v}^T)^{-1} = A^{-1} - \frac{A^{-1}\mathbf{u}\mathbf{v}^T A^{-1}}{1 + \mathbf{v}^T A^{-1}\mathbf{u}} \quad (1.38)$$

Weiterhin hilfreich und bekannt ist der Gauß'sche Integralsatz, mit dessen Hilfe man ein Volumenintegral in ein Integral über die Oberfläche des betreffenden Körpers umwandeln kann:

$$\int_V \operatorname{div} \mathbf{v} dV = \int_S \mathbf{n}^T \mathbf{v} dS \quad (1.39)$$

Dabei ist \mathbf{n} der nach außen weisende Normalenvektor auf der Oberfläche des Körpers; das Symbol div steht für den Divergenzoperator:

$$\text{div} \mathbf{v} = \frac{\partial v_1}{\partial x_1} + \frac{\partial v_2}{\partial x_2} + \frac{\partial v_3}{\partial x_3} \quad (1.40)$$

In den vorangehenden Abschnitten haben wir das Summationssymbol Σ verwendet. Eine weit verbreitete, kompaktere Schreibweise verzichtet auf das Σ -Symbol mit der Konvention, das über alle Indizes zu summieren ist, die in einem Ausdruck zweimal vorkommen. Mit dieser sog. Einstein'schen Summenkonvention hätte beispielsweise (1.24) die folgende einfache Form:

$$c_{ij} = a_{ie} b_{ej} \quad (1.41)$$

In diesem Fall wäre also die Summation über den doppelt auftretenden Index „ e “ auszuführen. Man nennt einen Summationsindex wie hier das e auch „Dummy-Index“, da es egal ist, welcher Buchstabe als Symbol für den Summationsindex gewählt wird. Beispielsweise hätte der Ausdruck $c_{ij} = a_{iq} b_{qj}$ exakt dieselbe Bedeutung wie (1.41). Natürlich dürfen die Indizes i und j nicht ausgetauscht werden, es sei denn, dies erfolgt auf beiden Seiten der Gleichung. Unter Berücksichtigung der Summenkonvention erhält der Gauß'sche Integralsatz die folgende Form:

$$\int_V \frac{\partial v_i}{\partial x_i} dV = \int_S n_i v_i dS$$

Eine wichtige tensorielle Größe ist das Kronecker-Delta, definiert als

$$\begin{cases} \delta_{ij} = 1, & \text{wenn } i = j \\ \delta_{ij} = 0, & \text{wenn } i \neq j \end{cases} \quad (1.42)$$

Beispielsweise ist $a_{ij} = a_{ik} \delta_{kj}$. Ähnlich oft verwendet wird der Permutationstensor (Levi-Civita-Symbol, total antisymmetrischer Tensor) e_{ijk} , welcher für e_{123} und gerade Permutationen der Indizes (z. B. e_{231}) den Wert $+1$ hat. Bei ungeraden Permutationen der drei Indizes ist $e_{ijk} = -1$, also etwa für e_{213} . Wenn zwei oder drei Indizes gleich sind, verschwindet e_{ijk} . (In anderen Werken wird der Permutationstensor auch mit ε_{ijk} bezeichnet.)

In den vergangenen Jahren wurde die Indexnotation schrittweise durch die direkte Tensornotation ersetzt, welche auf den ersten Blick der Matrix-Vektor-Schreibweise ähnelt. Damit liest sich (1.24) als:

$$\mathbf{C} = \mathbf{A} \cdot \mathbf{B} \quad (1.43)$$

Der „Malpunkt“ in der Mitte der Matrizenmultiplikation steht also für die Summation über den Dummy-Index, in der Sprache der Tensornotation wird dies eine

einfache Kontraktion genannt. Entsprechend schreibt sich eine doppelte Kontraktion mit einem Doppelpunkt:

$$c = A : B \quad (1.44)$$

oder in Indexschreibweise: $c = a_{ie} b_{ei}$.

Für den Gradienten einer Größe verwenden wir das Nabla-Symbol ∇ , also

$$\mathbf{b} = \nabla f \quad (1.45)$$

mit dem in (1.20) definierten Gradienten. Dieser Operator kann auch auf Vektoren angewendet werden, und der Gauß'sche Integralsatz nimmt die Form

$$\int_V \nabla \cdot \mathbf{v} dV = \int_S \mathbf{n} \cdot \mathbf{v} dS$$

an. Das dyadische Produkt zweier Vektoren \mathbf{a} und \mathbf{b} wird zu

$$\mathbf{C} = \mathbf{a} \otimes \mathbf{b} \quad (1.46)$$

mit den Komponenten $c_{ij} = a_i b_j$. Schließlich definieren wir noch für einen Tensor zweiter Stufe \mathbf{A} den Divergenzoperator

$$\mathbf{a} = \nabla \cdot \mathbf{A} \quad (1.47)$$

sodass

$$a_j = \frac{\partial A_{ij}}{\partial x_i} \quad (1.48)$$

Die Spur eines Tensors ist definiert als die Summe seiner Diagonalelemente:

$$c = \text{Sp}(\mathbf{A}) \quad (1.49)$$

1.3

Vektoren und Tensoren

Bis jetzt haben wir Vektoren nur als bloße mathematische Hilfsmittel eingeführt: Felder, die eine Anzahl skalarer Größen in einer gegebenen Anordnung enthalten. Vektoren lassen sich aber durchaus auch physikalisch interpretieren. Betrachten wir etwa als Beispiel eine physikalische Kraft. Kräfte haben nicht nur eine Größe (einen Betrag), sondern auch eine Richtung. Oft ist es von Interesse zu wissen, wie sich die Komponenten einer Kraft verhalten, wenn sie in einem anderen Koordinatensystem beschrieben wird. Eine Translation addiert einfach zu allen Kraftkomponenten einen konstanten Betrag. Eine Drehung des Bezugssystems, etwa von einem x, y -Koordinatensystem zu einem \bar{x}, \bar{y} -Koordinatensystem (Abb. 1.3), verändert die Komponenten eines Vektors auf kompliziertere Weise.

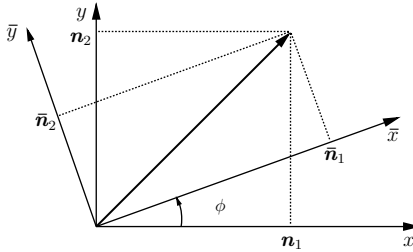


Abb. 1.3 x, y -Koordinatensystem und um den Winkel ϕ rotiertes \bar{x}, \bar{y} -System

Wir erhalten die Komponenten des Vektors $\bar{\mathbf{n}}$ im \bar{x}, \bar{y} -System aus denjenigen im x, y -System, zusammengefasst zum Vektor \mathbf{n} , durch die Transformation

$$\bar{\mathbf{n}} = \mathbf{R} \mathbf{n} \quad (1.50)$$

mit der Transformationsmatrix \mathbf{R} . Da die volle dreidimensionale Diskussion des Problems recht mühevoll ist (und keine wesentlich neuen Einsichten bringt), beschränken wir uns auf Drehungen in der Ebene. Es sei ϕ der Drehwinkel des Übergangs vom x, y -System zum \bar{x}, \bar{y} -System. Die beiden Vektoren, die im x, y -System die Darstellung $\mathbf{n}_1 = [1, 0]^T$ bzw. $\mathbf{n}_2 = [0, 1]^T$ haben, werden im rotierten Koordinatensystem als $\bar{\mathbf{n}}_1 = [\cos \phi, -\sin \phi]^T$ und $\bar{\mathbf{n}}_2 = [\sin \phi, \cos \phi]^T$ dargestellt. Daraus lässt sich die Form der zweidimensionalen Transformationsmatrix \mathbf{R} ableiten:

$$\mathbf{R} = \begin{bmatrix} \cos \phi & \sin \phi \\ -\sin \phi & \cos \phi \end{bmatrix} \quad (1.51)$$

Diese Transformationsmatrix, die auch Rotationsmatrix genannt wird, hat eine spezielle Struktur. Nachrechnen zeigt, dass

$$\mathbf{R}^{-1} = \mathbf{R}^T \quad (1.52)$$

was auch im allgemeineren dreidimensionalen Fall zutrifft. Matrizen, welche die Bedingung (1.52) erfüllen, heißen orthogonale Matrizen. Für diese Matrizen ist $\det(\mathbf{R}) = 1$.

Mithilfe der Regeln für die Koordinatentransformation bei Vektoren können wir entsprechende Regeln für Tensoren aufstellen. Tensoren, genauer gesagt Tensoren zweiter Stufe, sind physikalische Größen, welche zwei Vektoren miteinander in Beziehung setzen. Beispielsweise beschreibt der Spannungstensor eine Beziehung zwischen der Kraft auf einer Ebene und dem Normalenvektor dieser Ebene. Eine natürliche Darstellung eines Tensors zweiter Stufe ist eine Matrix. Es sind jedoch nicht alle Matrizen (Darstellungen von) Tensoren: Nur diejenigen Matrizen, welche die Regeln für die Koordinatentransformation erfüllen, können eine tensorielle Größe repräsentieren. Nehmen wir an, dass der Tensor zweiter Stufe \mathbf{C} die beiden Vektoren bzw. Tensoren erster Stufe \mathbf{t} und \mathbf{n} verbindet (mathematisch gesprochen: aufeinander abbildet):

$$\mathbf{t} = \mathbf{C} \mathbf{n} \quad (1.53)$$

Im \bar{x}, \bar{y} -Koordinatensystem vermittelt dann entsprechend der Tensor zweiter Stufe \bar{C} eine Beziehung zwischen $\bar{\mathbf{t}}$ und $\bar{\mathbf{n}}$:

$$\bar{\mathbf{t}} = \bar{C} \bar{\mathbf{n}} \quad (1.54)$$

Als nächstes setzen wir (1.50) und eine identische Beziehung für \mathbf{t} , d. h., $\bar{\mathbf{t}} = \mathbf{R} \mathbf{t}$, in (1.54) ein. Der Vergleich mit (1.53) zeigt, dass jeder Tensor zweiter Stufe sich beim Übergang in ein anderes Koordinatensystem nach der folgenden Regel transformiert:

$$\bar{C} = \mathbf{R} \mathbf{C} \mathbf{R}^T \quad (1.55)$$

Mit (1.51) leiten wir hieraus für den zweidimensionalen Fall ab:

$$\begin{aligned} \bar{c}_{11} &= c_{11} \cos^2 \phi + (c_{12} + c_{21}) \cos \phi \sin \phi + c_{22} \sin^2 \phi \\ \bar{c}_{22} &= c_{11} \sin^2 \phi - (c_{12} + c_{21}) \cos \phi \sin \phi + c_{22} \cos^2 \phi \\ \bar{c}_{12} &= -c_{11} \cos \phi \sin \phi + c_{12} \cos^2 \phi - c_{21} \sin^2 \phi + c_{22} \cos \phi \sin \phi \\ \bar{c}_{21} &= -c_{11} \cos \phi \sin \phi - c_{12} \sin^2 \phi + c_{21} \cos^2 \phi + c_{22} \cos \phi \sin \phi \end{aligned} \quad (1.56)$$

Bei symmetrischen Tensoren zweiter Stufe, mit denen wir uns hier ausschließlich beschäftigen, gilt $c_{21} = c_{12}$, und daher auch $\bar{c}_{21} = \bar{c}_{12}$.

Offensichtlich ändern sich die Komponenten eines Tensors, wenn er seine räumliche Orientierung wechselt (d. h., wenn wir eine Koordinatentransformation auf ihn anwenden). Es ist oft von Interesse, wie groß oder klein die Werte der Komponenten \bar{c}_{11} und \bar{c}_{22} werden können und in welcher Ebene, also bei welchem Wert des Drehwinkels ϕ , diese extremalen Werte der Diagonalelemente angenommen werden. Bei symmetrischen Tensoren zweiter Stufe gibt es zwei aufeinander senkrecht stehende Ebenen, auf denen \bar{c}_{11} und \bar{c}_{22} maximal bzw. minimal werden. Die Werte in diesen Koordinatensystemen nennt man gewöhnlich die Hauptwerte des Tensors. Da \bar{c}_{11} und \bar{c}_{22} Funktionen des Winkels ϕ sind, müssen die Hauptwerte die Bedingungen

$$\frac{\partial \bar{c}_{11}}{\partial \phi} = 0 \quad \text{bzw.} \quad \frac{\partial \bar{c}_{22}}{\partial \phi} = 0 \quad (1.57)$$

erfüllen. Für symmetrische Tensoren zweiter Stufe führt dies auf die Aussage, dass die diagonalen Komponenten des Tensors ihre extremalen Wert bei

$$\tan(2\phi) = \frac{2c_{12}}{c_{11} - c_{22}} \quad (1.58)$$

annehmen. Um daraus die Hauptwerte zu bestimmen, schreiben wir zunächst die ersten beiden Gleichungen aus (1.56) um:

$$\bar{c}_{11} = \frac{1}{2} (c_{11} + c_{22}) + \frac{1}{2} (c_{11} - c_{22}) \cos(2\phi) + c_{12} \sin(2\phi) \quad (1.59)$$

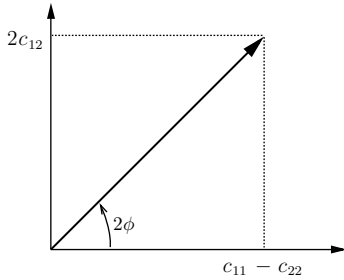


Abb. 1.4 Hauptrichtungen eines Tensors zweiter Stufe

Aus Abb. 1.4, vgl. (1.58), entnehmen wir, dass

$$\begin{aligned}\sin(2\phi) &= \pm \frac{2c_{12}}{\sqrt{(c_{11} - c_{22})^2 + 4c_{12}^2}} \\ \cos(2\phi) &= \pm \frac{c_{11} - c_{22}}{\sqrt{(c_{11} - c_{22})^2 + 4c_{12}^2}}\end{aligned}\quad (1.60)$$

woraus sich schließlich der folgende geschlossene Ausdruck für die Hauptwerte ergibt:

$$\begin{cases} \bar{c}_{11} = \frac{1}{2}(c_{11} + c_{22}) - \frac{1}{2}\sqrt{(c_{11} - c_{22})^2 + 4c_{12}^2} \\ \bar{c}_{22} = \frac{1}{2}(c_{11} + c_{22}) + \frac{1}{2}\sqrt{(c_{11} - c_{22})^2 + 4c_{12}^2} \end{cases}\quad (1.61)$$

Es ist eine Eigenschaft symmetrischer Tensoren zweiter Stufe (auf die wir uns hier beschränken), dass bei diesem Neigungswinkel die Nichtdiagonalelemente des Tensors verschwinden: $\bar{c}_{12} = 0$. Dies sehen Sie am einfachsten, wenn Sie die erste Gleichung aus (1.56) als

$$\bar{c}_{12} = -\frac{1}{2}(c_{11} - c_{22})\sin(2\phi) + c_{12}\cos(2\phi)\quad (1.62)$$

schreiben. Einsetzen der Beziehungen (1.60) beweist die Vermutung.

Das Koordinatensystem, in welchem die Hauptwerte in der Diagonale des Tensors stehen, kann man auch noch anders interpretieren. Seien C eine Matrixdarstellung eines symmetrischen Tensors zweiter Stufe und e ein Vektor. In der Regel wird das Produkt Ce nicht parallel zu e sein. Es gibt jedoch für jeden solchen Tensor ein Koordinatensystem, in dem der resultierende Vektor tatsächlich parallel zum ursprünglichen steht:

$$Ce = \lambda e\quad (1.63)$$

Den Skalar λ nennt man dann einen Eigenwert des Tensors. Wir können nun (1.63) umformen zu

$$(C - \lambda I)e = 0\quad (1.64)$$

mit der Einheitsmatrix $I = \text{diag}[1, \dots, 1]$. Eine nichttriviale Lösung ($\boldsymbol{e} \neq \mathbf{0}$) existiert dann und nur dann, wenn die Determinante von $\mathbf{C} - \lambda \mathbf{I}$ verschwindet:

$$\det[\mathbf{C} - \lambda \mathbf{I}] = 0 \quad (1.65)$$

Umformen von (1.65) ergibt dann exakt (1.58). Es ist also das Koordinatensystem, in dem c_{11} und c_{22} extreme Werte annehmen, genau dasjenige, in dem das Produkt $\mathbf{C}\boldsymbol{e}$ ein skalares Vielfaches von \boldsymbol{e} ist, also parallel zu \boldsymbol{e} steht. Da die Eigenwerte λ_i den Hauptwerten entsprechen, weisen die Eigenvektoren \boldsymbol{e}_i in die Hauptrichtungen. Für den Fall eines symmetrischen Tensors zweiter Stufe ist die Rechnung in Kasten 1.1 ausgeführt. Ähnlich wie bei den Pivotelementen, (1.37), ist das Produkt der Eigenwerte gleich der Determinanten der Matrix:

$$\det \mathbf{C} = \prod_{i=1}^n \lambda_i \quad (1.66)$$

Hierfür wird manchmal die Bezeichnung „Vieta-Regel“ verwendet. Sie gilt für symmetrische und für nichtsymmetrische Matrizen. Aus (1.66) können wir schließen, dass bei einer singulären Matrix nicht nur die Determinante und mindestens ein Pivotelement verschwinden, sondern auch mindestens ein Eigenwert gleich null ist.

Invertieren von (1.55) ergibt

$$\mathbf{C} = \mathbf{R}^T \bar{\mathbf{C}} \mathbf{R} \quad (1.67)$$

mit – in den Hauptachsen –

$$\bar{\mathbf{C}} = \begin{bmatrix} \bar{c}_{11} & 0 \\ 0 & \bar{c}_{22} \end{bmatrix} \quad (1.68)$$

und mit $\bar{c}_{11} = \lambda_1$ und $\bar{c}_{22} = \lambda_2$ als Hauptwerte bzw. Eigenwerte von \mathbf{C} . Auflösen von (1.67) mithilfe von (1.51) für \mathbf{R} in zwei Dimensionen gibt:

$$\mathbf{C} = \lambda_1 \begin{bmatrix} \cos^2 \phi & \cos \phi \sin \phi \\ \cos \phi \sin \phi & \sin^2 \phi \end{bmatrix} + \lambda_2 \begin{bmatrix} \sin^2 \phi & -\cos \phi \sin \phi \\ -\cos \phi \sin \phi & \cos^2 \phi \end{bmatrix}$$

oder

$$\mathbf{C} = \lambda_1 \begin{pmatrix} \cos \phi \\ \sin \phi \end{pmatrix} (\cos \phi, \sin \phi) + \lambda_2 \begin{pmatrix} \sin \phi \\ -\cos \phi \end{pmatrix} (\sin \phi, -\cos \phi)$$

Wir identifizieren $\boldsymbol{e}_1^T = (\cos \phi, \sin \phi)$ und $\boldsymbol{e}_2^T = (\sin \phi, -\cos \phi)$ als die Eigenvektoren, und stellen \mathbf{C} durch eine spektrale Zerlegung dar:

$$\mathbf{C} = \sum_{i=1}^n \lambda_i \boldsymbol{e}_i \otimes \boldsymbol{e}_i \quad (1.69)$$

Kasten 1.1: Eigenwerte eines symmetrischen Tensors zweiter Stufe

Für eine symmetrische Matrix C lässt sich die Bedingung $\det[C - \lambda I] = 0$ wie folgt umformen:

$$\begin{vmatrix} c_{11} - \lambda & c_{12} \\ c_{12} & c_{22} - \lambda \end{vmatrix} = 0 \quad \text{oder} \quad (c_{11} - \lambda)(c_{22} - \lambda) - c_{12}^2 = 0$$

Auflösen nach den Eigenwerten λ ergibt

$$\lambda_{1,2} = \frac{1}{2}(c_{11} + c_{22}) \pm \frac{1}{2}\sqrt{(c_{11} - c_{22})^2 + 4c_{12}^2}.$$

Dies sind gerade die Hauptwerte des Tensors C , vgl. (1.61). Die Richtungen der Vektoren e können nun berechnet werden, indem die Hauptwerte des Tensors C in eine der Gleichungen

$$(c_{11} - \lambda)e_1 + c_{12}e_2 = 0 \quad \text{oder} \quad c_{12}e_1 + (c_{22} - \lambda)e_2 = 0$$

eingesetzt werden; e_1, e_2 sind die Komponenten von e . Nehmen wir beispielsweise die erste Gleichung, so führt Einsetzen der Hauptwerte $\lambda_{1,2}$ auf

$$\left(\frac{1}{2}(c_{11} - c_{22}) \pm r\right)e_1 + c_{12}e_2 = 0, \quad \text{wobei} \quad r = \frac{1}{2}\sqrt{(c_{11} - c_{22})^2 + 4c_{12}^2}$$

Wir bringen den re_1 -Term auf die rechte Seite und quadrieren:

$$\frac{e_1 e_2}{e_1^2 - e_2^2} = \frac{c_{12}}{c_{11} - c_{22}}$$

Elementare Trigonometrie zeigt, dass

$$\tan 2\phi = \frac{2 \tan \phi}{1 - \tan^2 \phi} = \frac{2e_1 e_2}{e_1^2 - e_2^2}$$

womit bewiesen ist, dass (1.58) nicht nur die Hauptrichtungen, sondern ebenso die Richtungen der Eigenvektoren e festlegt. Die Begriffe Eigenvektor und Hauptrichtung bzw. Eigenwerte und Hauptwerte von symmetrischen Tensoren zweiter Stufe fallen also zusammen.

wobei wir die Aussage in diesem Fall auf n Dimensionen verallgemeinert haben. Wir definieren die Eigenprojektionen

$$E_i = e_i \otimes e_i \tag{1.70}$$

und können dann die spektrale Zerlegung eines symmetrischen Tensors zweiter Stufe wie folgt angeben:

$$C = \sum_{i=1}^n \lambda_i E_i \tag{1.71}$$

1.4

Spannungs- und Dehnungstensor

Die Grundaufgabe in der Mechanik fester Körper ist es, die Reaktion eines Körpers auf externe Kräfte zu bestimmen. Beispielsweise möchte man wissen, welche Kräfte von einer Seite eines gedachten Querschnitts durch den Körper auf die andere Seite dieser Fläche ausgeübt werden (Abb. 1.5). Das übliche Vorgehen in diesem Fall besteht darin, eine kleine Fläche innerhalb der gedachten Querschnittsfläche, ΔS , zu betrachten und die dort angreifende Kraft zu untersuchen. Diese Kraft wird mit Δf bezeichnet. Im Grenzfall $\Delta S \rightarrow 0$ führt dies auf den Spannungsvektor \mathbf{t} :

$$\mathbf{t} = \lim_{\Delta S \rightarrow 0} \frac{\Delta \mathbf{f}}{\Delta S} = \frac{d\mathbf{f}}{dS} \quad (1.72)$$

In jeder Ebene kann der Spannungsvektor \mathbf{t} in zwei Anteile zerlegt werden: eine Komponente parallel zum Normalenvektor der Ebene, die Normalspannung, und in zwei aufeinander senkrecht stehende Vektoren, die eine Basis der Ebene bilden und Schubspannungen (Scherspannungen) genannt werden. Wir wählen nun unser Koordinatensystem so, dass der Normalenvektor der betrachteten Ebene mit der x -Achse zusammenfällt. Die Normalkomponente von \mathbf{t} (die Normalspannung) bezeichnen wir als σ_{xx} , die in der Ebene liegenden Schubspannungen nennen wir σ_{xy} und σ_{xz} . σ_{xy} ist dabei der Anteil der Spannung, der in y -Richtung wirkt, σ_{xz} wirkt entsprechend in z -Richtung. Gemäß den Vorzeichenkonventionen der Mechanik fester Körper zählen wir die Normalspannung σ_{xx} positiv, wenn sie in Richtung der positiven x -Achse zeigt und auf eine Ebene wirkt, deren Normalenvektor ebenfalls in positive x -Richtung weist. In ähnlicher Weise nehmen wir σ_{xy} als positiv an, wenn sie in positive y -Richtung weist und auf eine Ebene mit Normalenvektor in positiver x -Richtung wirkt. Analog wird das Vorzeichen der anderen Schubspannung, σ_{xz} , gewählt. Beachten Sie: Mit diesen Konventionen ist die Normalspannung σ_{xx} auch dann positiv, wenn sie in negative x -Richtung auf eine Ebene wirkt, deren Normalenvektor in negative x -Richtung zeigt! Ebenso ist die Schubspannung σ_{xy} positiv, wenn sie in negative y -Richtung auf eine Ebene mit Normalenvektor in negativer x -Richtung wirkt.

In drei Dimensionen gibt es neun Spannungskomponenten (Abb. 1.6). Diese neun Größen bestimmen vollständig den Spannungszustand in einem Punkt eines Körpers. Sie bilden die Komponenten des Spannungstensors $\boldsymbol{\sigma}$. Dieser ist ein

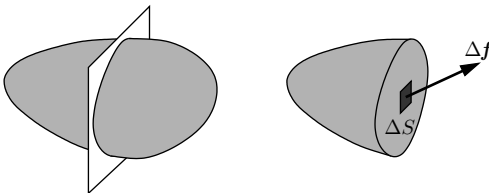


Abb. 1.5 Kraft auf eine gedachte Schnittfläche in einem festen Körper

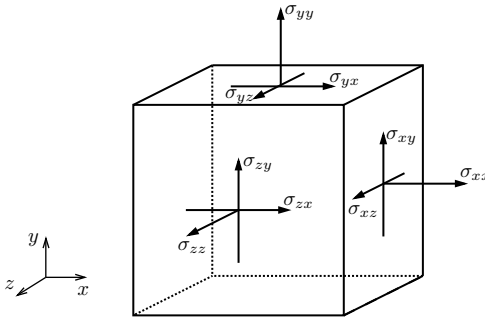


Abb. 1.6 Spannungskomponenten im dreidimensionalen Kontinuum

Tensor zweiter Stufe. Die natürliche Darstellung in Matrixschreibweise ist

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_{xx} & \sigma_{yx} & \sigma_{zx} \\ \sigma_{xy} & \sigma_{yy} & \sigma_{zy} \\ \sigma_{xz} & \sigma_{yz} & \sigma_{zz} \end{bmatrix} \quad (1.73)$$

Der Spannungstensor $\boldsymbol{\sigma}$ hängt unmittelbar mit dem oben definierten Spannungsvektor \boldsymbol{t} zusammen, der auf eine Ebene mit Normalenvektor \boldsymbol{n} einwirkt. In Matrixvektor-Notation gilt

$$\boldsymbol{\Sigma} \boldsymbol{n} = \boldsymbol{t} \quad (1.74)$$

Sie können diese Beziehung leicht überprüfen, wenn sie den Normalenvektor jeweils parallel zur x -Achse ($\boldsymbol{n}^T = [1, 0, 0]$), zur y -Achse ($\boldsymbol{n}^T = [0, 1, 0]$) und zur z -Achse ($\boldsymbol{n}^T = [0, 0, 1]$) wählen. Für den späteren Gebrauch führen wir das Analogon von (1.74) auch in Indeschreibweise

$$n_i \sigma_{ij} = t_j$$

und in direkter Tensornotation auf:

$$\boldsymbol{n} \cdot \boldsymbol{\sigma} = \boldsymbol{t} \quad (1.75)$$

Für ein nichtpolares oder Boltzmann-Kontinuum folgt aus der Drehimpulserhaltung in den drei Richtungen, dass nicht alle Komponenten des Spannungstensors unabhängig sind. Insbesondere gilt für die Schubspannungskomponenten

$$\begin{aligned} \sigma_{xy} &= \sigma_{yx} \\ \sigma_{yz} &= \sigma_{zy} \\ \sigma_{zx} &= \sigma_{xz} \end{aligned} \quad (1.76)$$

(in Kapitel 2 finden Sie einen formalen Beweis hierfür). Daher gibt es nur sechs unabhängige Komponenten des Spannungstensors und die Matrixdarstellung des

symmetrischen Spannungstensors ist

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{zx} \\ \sigma_{xy} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{yz} & \sigma_{zz} \end{bmatrix} \quad (1.77)$$

Da der Spannungstensor nur sechs unabhängige Komponenten enthält, kann man ihn auch als einen (sechskomponentigen) Vektor darstellen, was als Voigt-Notation bezeichnet wird:

$$\boldsymbol{\sigma}^T = (\sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{xy}, \sigma_{yz}, \sigma_{zx}) \quad (1.78)$$

Beachten Sie, dass für die Vektordarstellung des Spannungstensors das Symbol $\boldsymbol{\sigma}$ steht, während für die Matrixdarstellung $\boldsymbol{\Sigma}$ verwendet wird.

Oft ist es – etwa in der Geotechnik – praktisch, die Normalspannungen σ_{xx} , σ_{yy} und σ_{zz} in zwei Anteile aufzuspalten, nämlich einen konstanten „hydrostatischen“ (isotropen) Anteil und einen „Deviator“-Teil, genannt Spannungsdeviator („Spannungsunterschiedstensor“), der die jeweilige Abweichung vom Mittelwert beschreibt. Der Deviatoranteil bewirkt bei einem würfelförmigen Volumenelement Formänderungen, der hydrostatische Druck eine Volumenänderung. Der hydrostatische Druck ist hier definiert als der Mittelwert der Normalspannungen:

$$p = \frac{1}{3}(\sigma_{xx} + \sigma_{yy} + \sigma_{zz}) \quad (1.79)$$

Mit dieser Definition von p können wir den Spannungsdeviator angeben. Seine Matrixdarstellung lautet

$$\boldsymbol{S} = \boldsymbol{\Sigma} - p \boldsymbol{I} \quad (1.80)$$

In Voigt-Notation ergibt sich die folgende Formel:

$$\boldsymbol{s} = \boldsymbol{\sigma} - p \boldsymbol{i} \quad (1.81)$$

wobei

$$\begin{aligned} \boldsymbol{s}^T &= (s_{xx}, s_{yy}, s_{zz}, s_{xy}, s_{yz}, s_{zx}) \\ \boldsymbol{i}^T &= (1, 1, 1, 0, 0, 0) \end{aligned} \quad (1.82)$$

Spannungsinvarianten sind wichtige Größen in nichtlinearen Stoffmodellen. Man versteht darunter Spannungskomponenten, die invariant unter einem Wechsel des Bezugssystems sind. Diese ergeben sich auf natürliche Weise, wenn man die Hauptspannungen in einem dreidimensionalen Kontinuum berechnet. Wir haben im letzten Abschnitt gezeigt, wie man die Hauptwerte λ eines Tensors zweiter Stufe aus der Bedingung

$$\det(\boldsymbol{\Sigma} - \lambda \boldsymbol{I}) = 0 \quad (1.83)$$

ableitet. Komponentenweise entspricht dies

$$\begin{vmatrix} \sigma_{xx} - \lambda & \sigma_{xy} & \sigma_{zx} \\ \sigma_{xy} & \sigma_{yy} - \lambda & \sigma_{yz} \\ \sigma_{zx} & \sigma_{yz} & \sigma_{zz} - \lambda \end{vmatrix} = 0 \quad (1.84)$$

Mit den Größen

$$\begin{aligned} I_1 &= \sigma_{xx} + \sigma_{yy} + \sigma_{zz} \\ I_2 &= \sigma_{xx}\sigma_{yy} + \sigma_{yy}\sigma_{zz} + \sigma_{zz}\sigma_{xx} - \sigma_{xy}^2 - \sigma_{yz}^2 - \sigma_{zx}^2 \\ I_3 &= \sigma_{xx}\sigma_{yy}\sigma_{zz} + 2\sigma_{xy}\sigma_{yz}\sigma_{zx} - \sigma_{xx}\sigma_{yz}^2 - \sigma_{yy}\sigma_{zx}^2 - \sigma_{zz}\sigma_{xy}^2 \end{aligned} \quad (1.85)$$

nimmt (1.84) die folgende Form an:

$$\lambda^3 - I_1\lambda^2 + I_2\lambda - I_3 = 0 \quad (1.86)$$

Eine entscheidende Einsicht ist nun, dass diese Gleichung in jedem Bezugssystem dieselbe Lösung hat und daher I_1 , I_2 und I_3 invariant unter Koordinatentransformation sein müssen. Aus diesem Grund nennt man I_1 , I_2 und I_3 Invarianten des Spannungstensors oder kurz Spannungsinvarianten. Das Konzept der Hauptwerte und -richtungen lässt sich auf beliebige Tensoren zweiter Stufe übertragen. Darum kann man auch für jeden Tensor zweiter Stufe Invarianten definieren, insbesondere auch für den Dehnungstensor, den wir als nächstes behandeln werden.

Jede Funktion der Invarianten ist selbst eine Invariante. Solche modifizierten Invarianten ergeben sich ganz natürlich, wenn die Hauptwerte des Spannungsdeviators bestimmt werden, also wenn man die kubische Gleichung

$$\lambda^3 - J_2\lambda - J_3 = 0 \quad (1.87)$$

aufföst. Dabei sind

$$J_2 = -s_{xx}s_{yy} - s_{yy}s_{zz} - s_{zz}s_{xx} + s_{xy}^2 + s_{yz}^2 + s_{zx}^2 \quad (1.88)$$

und

$$J_3 = s_{xx}s_{yy}s_{zz} + 2s_{xy}s_{yz}s_{zx} - s_{xx}s_{yz}^2 - s_{yy}s_{zx}^2 - s_{zz}s_{xy}^2 \quad (1.89)$$

(Die erste Invariante des Spannungsdeviators, J_1 , verschwindet per definitionem.) Mit den obigen Definitionen der Invarianten von Spannungstensor und Spannungsdeviator lässt sich zeigen, dass (Fung, 1965)

$$\begin{aligned} J_2 &= \frac{1}{3}I_1^2 - I_2 \\ J_3 &= I_3 - \frac{1}{3}I_1I_2 + \frac{2}{27}I_1^3 \end{aligned} \quad (1.90)$$

Wir betrachten nun ein würfelförmiges Volumenelement, das wir nur in der x , y -Ebene verformen. Wir bezeichnen die Seitenflächen des Würfels mit Δx , Δy und Δz ($\Delta x = \Delta y = \Delta z$). Weiter nehmen wir an, dass Punkt A die Verschie-

bungen u und v erfährt, die Punkte B und C verschieben sich entsprechend um $[u + \Delta u^B, v + \Delta v^B]$ und $[u + \Delta u^C, v + \Delta v^C]$ (Abb. 1.7). Im Grenzfall $\Delta x \rightarrow 0$ und $\Delta y \rightarrow 0$ werden die Dehnungen in x - und y -Richtung zu

$$\begin{aligned}\epsilon_{xx} &= \lim_{\Delta x \rightarrow 0} \frac{\Delta u^B}{\Delta x} = \frac{\partial u}{\partial x} \\ \epsilon_{yy} &= \lim_{\Delta y \rightarrow 0} \frac{\Delta v^C}{\Delta y} = \frac{\partial v}{\partial y}\end{aligned}\quad (1.91)$$

(unter Vernachlässigung von Termen zweiter Ordnung). Die Verzerrung des quadratischen Flächenelements in der x, y -Ebene ist gegeben durch:

$$\gamma_{xy} = \lim_{\Delta x \rightarrow 0, \Delta y \rightarrow 0} \frac{\Delta u^C}{\Delta y} + \frac{\Delta v^B}{\Delta x} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}$$
(1.92)

die Rotation durch:

$$\omega_{xy} = \lim_{\Delta x \rightarrow 0, \Delta y \rightarrow 0} \frac{1}{2} \left(\frac{\Delta v^B}{\Delta x} - \frac{\Delta u^C}{\Delta y} \right) = \frac{1}{2} \left(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \right)$$
(1.93)

Hierbei ist γ_{xy} die totale Winkelverzerrung des würfelförmigen Volumenelements in der x, y -Ebene. Sie wird bei technischen Anwendungen oft als Maß für die Scherdehnung verwendet und daher auch technische Scherdehnung genannt. Für theoretische Untersuchungen ist es üblicher, die tensorielle Scherdehnungskomponente $\epsilon_{xy} = \frac{1}{2} \gamma_{xy}$ zu verwenden. Ähnlich wie bei den Normaldehnungen ϵ_{xx} und ϵ_{yy} sowie der technischen Scherdehnung γ_{xy} können wir auch die Normaldehnung ϵ_{zz} und die Scherdehnungen γ_{yz} und γ_{zx} einführen, indem wir die Verformung des würfelförmigen Volumenelements jeweils in der y, z - bzw. der z, x -Ebene betrachten. In Einklang mit den Definitionen (1.91) und (1.92) werden diese Dehnungskomponenten definiert als

$$\begin{aligned}\epsilon_{zz} &= \frac{\partial w}{\partial z} \\ \gamma_{yz} &= \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \\ \gamma_{zx} &= \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z}\end{aligned}\quad (1.94)$$

Hierbei ist w die Verschiebung in z -Richtung.

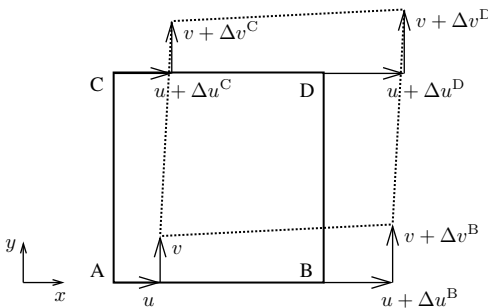


Abb. 1.7 Unverzerrte und verzerrte Konfiguration eines viereckigen Flächenelements

Die Bedeutung der Indizes ist beim Dehnungstensor exakt die gleiche wie beim Spannungstensor: ϵ_{xx} ist die Normaldehnungskomponente in x -Richtung und γ_{xy} steht für die Scherdehnungskomponente in der x, y -Ebene. Auch die Vorzeichenkonvention ist die gleiche, eine Dehnungskomponente hat also einen positiven Wert, wenn sie der positiven Verschiebung einer Ebene mit Normalenvektor in positive Rechnung entspricht usw. Dies bedeutet, dass eine Verlängerung – entsprechend dem Sprachgebrauch – als eine positive Dehnung angesehen wird.

Damit können wir jetzt den Dehnungstensor formal einführen, ganz analog zum Vorgehen beim Spannungstensor. Auch sind Matrix- und Vektordarstellung gleichermaßen möglich. Im allgemeinen dreidimensionalen Fall haben wir die Matrixdarstellung

$$\mathcal{E} = \begin{bmatrix} \epsilon_{xx} & \epsilon_{yx} & \epsilon_{zx} \\ \epsilon_{xy} & \epsilon_{yy} & \epsilon_{zy} \\ \epsilon_{xz} & \epsilon_{yz} & \epsilon_{zz} \end{bmatrix} \quad (1.95)$$

und, da auch der Dehnungstensor symmetrisch ist, die Vektordarstellung (Voigt-Notation)

$$\boldsymbol{\epsilon}^T = (\epsilon_{xx}, \epsilon_{yy}, \epsilon_{zz}, \gamma_{xy}, \gamma_{yz}, \gamma_{zx}) \quad (1.96)$$

Während die Größen γ_{xy} etc. in der Voigt-Notation bevorzugt werden, findet man bei der Matrixdarstellung normalerweise die tensoriellen Scherdehnungen ϵ_{xy}, \dots . Für die spätere Diskussion bemerken wir, dass die Volumendichte der inneren Energie sich in äquivalenter Weise in Voigt-Notation, Matrixdarstellung oder Indexschreibweise ausdrücken lässt:

$$\dot{\mathcal{W}}_{\text{int}} = \dot{\boldsymbol{\epsilon}}^T \boldsymbol{\sigma} = \dot{\boldsymbol{\epsilon}} : \boldsymbol{\sigma} = \dot{\epsilon}_{ij} \sigma_{ji} \quad (1.97)$$

Bei der Diskussion des Spannungstensors haben wir den hydrostatischen Druck p eingeführt. Ähnlich dazu führen wir nun die skalare relative Volumenänderung ϵ_{Vol} als Summe der Normaldehnungen ein:

$$\epsilon_{\text{Vol}} = \epsilon_{xx} + \epsilon_{yy} + \epsilon_{zz} \quad (1.98)$$

Mithilfe der relativen Volumenänderung ϵ_{Vol} definieren wir – dem Spannungsdeviator entsprechend – den sog. Dehnungsdeviator:

$$\boldsymbol{E} = \boldsymbol{\mathcal{E}} - \frac{1}{3} \epsilon_{\text{Vol}} \boldsymbol{I} \quad (1.99)$$

in Voigt-Notation:

$$\boldsymbol{e} = \boldsymbol{\epsilon} - \frac{1}{3} \epsilon_{\text{Vol}} \boldsymbol{i} \quad (1.100)$$

mit

$$\mathbf{e}^T = (\epsilon_{xx}, \epsilon_{yy}, \epsilon_{zz}, \gamma_{xy}, \gamma_{yz}, \gamma_{zx}) \quad (1.101)$$

In einem vorhergehenden Abschnitt haben wir die Transformationsregel für Tensoren zweiter Stufe hergeleitet, vgl. (1.56). In der Voigt-Notation können wir diese Regel im zweidimensionalen Fall folgendermaßen schreiben:

$$\bar{\boldsymbol{\sigma}} = \mathbf{T}_\sigma \boldsymbol{\sigma} \quad (1.102)$$

mit

$$\mathbf{T}_\sigma = \begin{bmatrix} \cos^2 \phi & \sin^2 \phi & 2 \sin \phi \cos \phi \\ \sin^2 \phi & \cos^2 \phi & -2 \sin \phi \cos \phi \\ -\sin \phi \cos \phi & \sin \phi \cos \phi & \cos^2 \phi - \sin^2 \phi \end{bmatrix} \quad (1.103)$$

und dem Spannungstensor $\boldsymbol{\sigma}^T = [\sigma_{xx}, \sigma_{yy}, \sigma_{xy}]$ für planare Spannungsbedingungen. Einsetzen von $(-\phi)$ in die letzte Gleichung zeigt, dass $\mathbf{T}_\sigma^{-1} = \mathbf{T}_\sigma^T$, woraus

$$\boldsymbol{\sigma} = \mathbf{T}_\sigma^T \bar{\boldsymbol{\sigma}} \quad (1.104)$$

folgt. Da die technische Scherdehnung γ_{xy} normalerweise in Voigt-Notation angegeben wird, haben wir als Dehnungstransformationsgleichung

$$\bar{\boldsymbol{\epsilon}} = \mathbf{T}_\epsilon \boldsymbol{\epsilon} \quad (1.105)$$

mit

$$\mathbf{T}_\epsilon = \begin{bmatrix} \cos^2 \phi & \sin^2 \phi & \sin \phi \cos \phi \\ \sin^2 \phi & \cos^2 \phi & -\sin \phi \cos \phi \\ -2 \sin \phi \cos \phi & 2 \sin \phi \cos \phi & \cos^2 \phi - \sin^2 \phi \end{bmatrix} \quad (1.106)$$

und $\boldsymbol{\epsilon}^T = [\epsilon_{xx}, \epsilon_{yy}, \gamma_{xy}]$. Wie bei der Transformation von Spannungen gilt

$$\boldsymbol{\epsilon} = \mathbf{T}_\epsilon^T \bar{\boldsymbol{\epsilon}} \quad (1.107)$$

1.5

Elastizität

Was haben wir bisher kennengelernt? Wir haben den Spannungs- und den Dehnungstensor eingeführt und wir haben kinetische Beziehungen untersucht, d. h. Beziehungen zwischen Verschiebungen und Dehnungen. In Kapitel 2 werden wir die Bewegungsgleichungen einführen. Um bis dahin die Feldgleichungen zu vervollständigen, brauchen wir noch Spannungs-Dehnungs-Relationen, auch Stoffgesetze genannt, welche zwischen dem Spannungs- und dem Dehnungstensor

vermitteln. Im einfachsten Modell, dem isotropen linearen Elastizitätsgesetz (Hooke'sches Gesetz), vermittelt ein Tensor vierter Stufe, der elastische Nachgiebigkeitstensor \mathbf{C}^e , die Beziehung zwischen dem Dehnungstensor $\boldsymbol{\epsilon}$ und dem Spannungstensor $\boldsymbol{\sigma}$:

$$\boldsymbol{\epsilon} = \mathbf{C}^e : \boldsymbol{\sigma} \quad (1.108)$$

oder in der inversen Form

$$\boldsymbol{\sigma} = \mathbf{D}^e : \boldsymbol{\epsilon} \quad (1.109)$$

mit dem elastischen Steifigkeitstensor \mathbf{D}^e . In Voigt-Notation wird die Nachgiebigkeitsrelation etwas übersichtlicher:

$$\begin{pmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{pmatrix} = \frac{1}{E} \begin{bmatrix} 1 & -\nu & -\nu & 0 & 0 & 0 \\ -\nu & 1 & -\nu & 0 & 0 & 0 \\ -\nu & -\nu & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2(1+\nu) & 0 & 0 \\ 0 & 0 & 0 & 0 & 2(1+\nu) & 0 \\ 0 & 0 & 0 & 0 & 0 & 2(1+\nu) \end{bmatrix} \begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{zx} \end{pmatrix} \quad (1.110)$$

Hier stehen E für den Elastizitätsmodul und ν für die Poisson-Zahl. Noch kompakter lässt sich (1.110) in Matrix-Vektor-Schreibweise fassen:

$$\boldsymbol{\epsilon} = \mathbf{C}^e \boldsymbol{\sigma} \quad (1.111)$$

mit der elastische Nachgiebigkeitsmatrix \mathbf{C}^e .

In (1.110) ist der Dehnungstensor $\boldsymbol{\epsilon}$ eine Funktion des Spannungstensors $\boldsymbol{\sigma}$. Um die umgekehrte Beziehung zu erhalten, schreiben wir zunächst die ersten drei Gleichungen von (1.110) als

$$\begin{aligned} \epsilon_{xx} &= \frac{1+\nu}{E} \sigma_{xx} - 3\nu E p \\ \epsilon_{yy} &= \frac{1+\nu}{E} \sigma_{yy} - 3\nu E p \\ \epsilon_{zz} &= \frac{1+\nu}{E} \sigma_{zz} - 3\nu E p \end{aligned} \quad (1.112)$$

Als nächstes addieren wir diese Gleichungen und erhalten mit (1.79) und (1.98)

$$\epsilon_{\text{vol}} = K^{-1} p \quad (1.113)$$

Dabei ist der Kompressionsmodul K das Verhältnis aus hydrostatischem Druck p und relativer Volumenänderung ϵ_{vol} ($K = \frac{p}{\epsilon_{\text{vol}}}$). Es gilt mit der Poisson-Zahl ν

$$K = \frac{E}{3(1-2\nu)} \quad (1.114)$$

Sukzessives Einsetzen von (1.113) in (1.112) und anschließendes Invertieren führt auf die elastische Steifigkeitsrelation:

$$\begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{zx} \end{pmatrix} = \begin{bmatrix} \lambda + 2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda + 2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda + 2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix} \begin{pmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{pmatrix} \quad (1.115)$$

wobei λ und μ die sog. Lamé-Konstanten sind:

$$\begin{aligned} \lambda &= \frac{\nu E}{(1 + \nu)(1 - 2\nu)} \\ \mu &= \frac{E}{2(1 + \nu)} \end{aligned} \quad (1.116)$$

Die Größe μ wird Schubmodul genannt und des öfteren auch mit dem Symbol G notiert. Die obige Relation schreibt sich in kompakter Form als

$$\boldsymbol{\sigma} = \mathbf{D}^e \boldsymbol{\epsilon} \quad (1.117)$$

mit der elastischen Nachgiebigkeitsmatrix \mathbf{D}^e . Ein alternativer Ausdruck für diese Matrix in Abhängigkeit vom Elastizitätsmodul und der Poisson-Zahl ergibt sich, wenn man (1.116) in (1.115) einsetzt:

$$\mathbf{D}^e = \frac{E}{(1 + \nu)(1 - 2\nu)} \begin{bmatrix} 1 - \nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1 - \nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1 - \nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} \quad (1.118)$$

1.6

Die PyFEM-Finite-Elemente-Bibliothek

Eine Anzahl von Modellen, die in diesem Buch besprochen werden, sind in einem kleinen Finite-Elemente-Code namens PyFEM implementiert, welcher in der Programmiersprache Python geschrieben ist. Um einige Features dieser Programmiersprache vorzustellen, werden wir im Folgenden eine PyFEM-Simulation des einfachen nichtlinearen Problems aus Abschnitt 1.1 vorstellen. Die

entsprechende Datei heißt `ShallowTruss.py` und findet sich im Verzeichnis `examples/ch01` von PyFEM („shallow truss“ heißt so viel wie „flacher Träger“).

Anstelle eines vollständigen Code-Listings werden wir uns bei der Diskussion auf eine Notation beschränken, die an das sog. Literate Programming („beschreibendes Programmieren“) angelehnt ist, vgl. z. B. Ramsey (1994). Ein kurzer Überblick über diese Notation wurde bereits im Abschnitt *Formeln und Abkürzungen* gegeben. Einige spezielle Besonderheiten werden in diesem Abschnitt noch einmal herausgestellt.

Beim Literate Programming wird der komplette Programmcode durch eine Hierarchie von Codefragmenten repräsentiert:

Fragment F1.1

(*Beispiel flaches Trägerelement*) ≡

```
<Initialisieren der Berechnung F1.2>
<Schrittweise Berechnung des Gleichgewichtspfads F1.7>
<Ergebnisausgabe F1.10>
```

Das Fragment F1.1 wird im Literate Programming also einfach als eine Abfolge von drei weiteren, untergeordneten Fragmenten definiert. Deren Funktion innerhalb des Programms wird bereits aus dem jeweiligen Namen ersichtlich und die grundlegende Struktur von F1.1 tritt deutlich hervor.

Die Zahl hinter dem Fragmentnamen gibt die Nummer des Fragments an, in welchem diese Routine definiert wird. Wenn Sie im Detail die Funktionsweise verstehen wollen, müssten Sie den originalen Sourcecode untersuchen.

Im ersten Fragment unseres Beispiels werden zunächst die Variablen, welche die räumliche Begrenzung der Simulation festlegen, deklariert:

Fragment F1.2

(*Initialisieren der Berechnung*) ≡

F1.1

```
b = 10.
h = 0.5
```

b und h stehen für die Systemabmessungen b und h , vgl. Abb. 1.1. Die Angabe am rechten Seitenrand verweist auf das übergeordnete Fragment, in dem dieses Fragment aufgerufen wird bzw. zuletzt diskutiert wurde.

Offensichtlich erfordert die Initialisierung des Beispiels mit dem flachen Trägerelement mehr als nur die Systemabmessungen. Wir erweitern daher das Fragment, indem wir die Steifigkeiten k und $EA0$ definieren, vgl. (1.8):

Fragment F1.3

(*Initialisieren der Berechnung*) +≡

F1.2

```
k = 1000.
EA0 = 5.0e6
```

Das Symbol „+≡“ hinter dem Fragmentnamen weist dabei darauf hin, dass dieses Fragment ein bereits zuvor definiertes Fragment erweitert. Wir können die Initialisierung weiter ergänzen, indem wir die inkrementelle externe Kraft DF , die Zahl der Schritte N , die Konvergenztoleranz tol und die maximal erlaubte Zahl von Iterationen bis zur Konvergenz, $iterMax$, angeben.

Fragment F1.4

⟨Initialisieren der Berechnung⟩ +≡ F1.3

```
DF      = 50
N       = 30
tol     = 1e-6
iterMax = 5
```

Das eigentliche Modell wird mithilfe der `lambda`-Funktion von Python definiert:

Fragment F1.5

⟨Initialisieren der Berechnung⟩ +≡ F1.4

```
from math import sqrt

l  = lambda v : sqrt(b**2+(h-v)**2)
F  = lambda v : -EA0*(h-v)/l(v)*(l(v)-l(0))/l(0)+k*v
dFdV= lambda v : (EA0/l(v))*((h-v)/l(v)**2+k+/
                 (EA0/l(v))*(l(v)-l(0))/l(0))
```

Bitte beachten Sie, dass wir in diesem Fragment eine mathematische Funktion verwendet haben, nämlich den Quadratwurzel-Operator. Diese Funktion wird aus dem `math`-Modul importiert. In PyFEM werden wir häufig Funktionen aus den Modulen `math`, `numpy` und `scipy` einsetzen. Aus Gründen der Übersichtlichkeit werden wir meistens die wichtige Import-Anweisung `from ... import ...` auslassen, es sei denn, die Herkunft der importierten Funktion ist unklar.

Als nächstes werden die Balkenlänge l und die Reaktionskraft des Systems F als Funktionen der Unbekannten v definiert, vgl. (1.3) und (1.8). Die nächste Zeile gibt die Ableitung der Kraft nach der Unbekannten v als Funktion von v an, vgl. (1.12). Zum Schluss werden noch die für die Simulation benötigten internen Parameter initialisiert:

Fragment F1.6

⟨Initialisieren der Berechnung⟩ +≡ F1.5

```
v      = 0.
Dv     = 0.
Fext   = 0.
output = [ [0.,0.] ]
```

Die Variablen `v`, `Dv` und `Fext` stehen für die Verschiebung v in der letzten konvergierten Lösung, die Schrittweite Δv sowie die totale externe Kraft F_{ext} . Die Variable

output ist eine „List of Lists“ und dient dazu, die Ergebnisse zu speichern, die dann als Last-Verschiebungs-Kurve geplottet werden, vgl. Abb. 1.2.

Die Simulation selbst besteht aus einer Schleife über N Lastschritte, wobei i die Nummer des aktuellen Schrittes angibt. Zunächst wird ein Header ausgegeben, welcher den aktuellen Lastschritt anzeigt. Dann wird die Iteration im Fragment F1.8 vorbereitet. Schließlich wird die Berechnung im Fragment F1.9 ausgeführt:

Fragment F1.7

(Schrittweise Berechnung des Gleichgewichtspfad) \equiv F1.1

```
for i in range(N):
    print '====='
    print '  Load_step%i' % i
    print '====='
    print '  NR_iter: |Fext-F(v)|'
    <Iteration vorbereiten F1.8>
    <Iteration F1.9>
```

Eine wichtige Besonderheit der Programmiersprache Python ist, dass Code-Blöcke lediglich durch Einrückung und nicht durch (zusätzliche) Klammern definiert werden. Code-Blöcke sind Abfolgen von Statements (Anweisungen), die innerhalb eines **if**-Statements, eines **for**-Statements oder einer **while**-Schleife ausgeführt werden. Ein Block endet dort, wo die Einrückung auf den vorherigen Wert zurückgesetzt wird. Diese Python-typische Einrückungsregel gilt ebenso für unsere Literate-Programming-Notation. Im obigen Fragment wird das neue Fragment F1.8 innerhalb der Schleife **for i in range(N):** ausgeführt, da der Aufruf dieses Fragments gegenüber dem **for**-Statement eingerückt ist. Dasselbe gilt für das Fragment F1.9.

Das Vorbereiten der Iteration besteht aus der Aktualisierung des Wertes der externen Kraft `Fext` sowie dem Zurücksetzen der Variablen `error` (siehe unten) und des Iterationszählers `iter`:

Fragment F1.8

(Iteration vorbereiten) \equiv F1.7

```
Fext = Fext + DF
error = 1.
iter = 0
```

Die Iteration selbst wird im folgenden Fragment ausgeführt:

Fragment F1.9

(Iteration) \equiv F1.7

```
while error > tol:
    iter += 1
    dv = ( 1. / dFdV(v+Dv) ) * ( Fext - F(v+Dv) )
    Dv += dv
```

```

error = abs( Fext - F(v+Dv) )
print 'Iter', iiter, ':', error

if iiter == iterMax:
    raise RuntimeError('Iterations did not converge!')

print 'Converged solution'
v += Dv
Dv = 0.
output.append( [ v, F(v) ] )

```

Das Fragment erhöht zunächst den Iterationszähler `iiter` um 1. Die Änderung der Verschiebung in diesem Iterationsschritt, `dv`, wird in der zweiten Zeile berechnet und dann zur totalen Änderung `Dv` addiert. Der Variablen `error` wird die betragsmäßige Abweichung zwischen `Fext` und `F(v+Dv)` zugewiesen und dieser Wert dann zusammen mit dem Wert des Iterationszählers ausgegeben. Um zu verhindern, dass das Programm in einer Endlosschleife landet, wird abgefragt, ob der Iterationszähler die maximal erlaubte Zahl von Schritten erreicht hat. Wenn ja, dann wird ein Runtime-Error ausgegeben. Eine konvergierte Lösung ist gefunden, wenn der Wert von `error` kleiner als die Konvergenz-Toleranz `tol` wird. Die Verschiebung `v` wird in diesem Fall aktualisiert und zusammen mit der aktuellen internen Kraft `F(v)` zur Output-Liste hinzugefügt.

Das letzte Fragment, das von F1.10 aufgerufen wird, gibt die Ergebnisse aus:

Fragment F1.10

Ergebnisausgabe ≡

F1.1

```

from pylab import plot, show, xlabel, ylabel

plot( [x[0] for x in output], [x[1] for x in output], 'ro' )

```

Das `output`-Feld wird geplottet mithilfe von `pylab`, einem Teil des `Matplotlib`-Pakets. In diesem Fragment wird die in Python integrierte „List Comprehension“ angewendet, um zwei neue Listen zu erzeugen, in welchen die Daten für die horizontale und die vertikale Achse enthalten sind.

In unserem Beispiel ist die exakte Lösung `F` bekannt und kann dem Plot hinzugefügt werden:

Fragment F1.11

Ergebnisausgabe +≡

F1.10

```

from numpy import arange

vrange = arange(0,1.2,0.01)
plot ( vrange, [F(vval) for vval in vrange], 'b-' )
xlabel ( 'v[m]' )
ylabel ( 'F[N]' )

show()

```

Die exakte Lösung wurde im Bereich $0,0 \leq F \leq 1,2$ mit einer Schrittweite von 0,01 berechnet. Nachdem die Labels für die horizontale und die vertikale Achse ausgegeben wurden, erscheint der Graph auf dem Bildschirm durch Aufrufen des Befehls `show()`.