Part I

Basic Concepts and Solution Techniques

1

Preliminaries

This chapter is primarily intended to familiarise the reader with the notation we have adopted throughout this book and to refresh some of the required background in mathematics, especially linear algebra, and applied mechanics. As regards notation, we remark that most developments have been carried out using matrix-vector notation, and tensor notation is less often needed, either in indicial form or in direct form. For the benefit of readers who are less familiar with tensor notation, we have added a small section on this topic. But, first, we will give an example of non-linearity in a structural member. This example involving a simple truss element can be solved analytically, and serves well to illustrate the various procedures that are described in this book for capturing non-linear phenomena in solids and structures, and for accurately solving the ensuing initial/boundary-value problems.

1.1 A Simple Example of Non-linear Behaviour

Many features of solution techniques can be demonstrated for simple truss structures, possibly in combination with springs, where the non-linear structural behaviour can stem from geometrical as well as from material non-linearities. In this section we shall assume that the displacements and rotations can be arbitrarily large, but that the strains remain small, say less than 5%. This limitation will be dropped in Part IV of this book, where the extension will be made to large elastic and inelastic strains.

We consider the shallow truss structure of Figure 1.1. From elementary equilibrium considerations in the deformed configuration, the following expression for the force can be deduced that acts in a symmetric half of the shallow truss:

$$F_{\text{int}} = -A\sigma \sin\phi - F_s \tag{1.1}$$

where σ is the axial stress in the member, F_s is half of the force in the spring, and ϕ is the angle of the truss member with the horizontal plane in the deformed configuration. Owing to the small-strain assumption, the difference between the cross section in the current configuration, A, and that in the original configuration, A_0, is negligible. For the same reason, the difference

Non-linear Finite Element Analysis of Solids and Structures, Second Edition.
René de Borst, Mike A. Crisfield, Joris J.C. Remmers and Clemens V. Verhoosel.

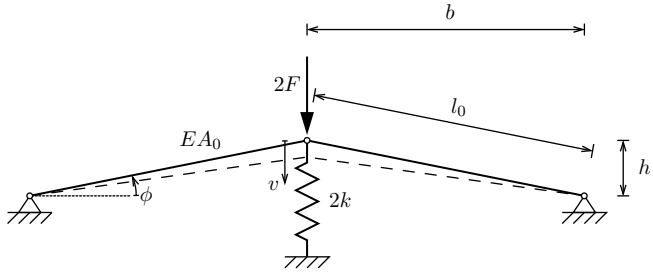

Figure 1.1 Plane shallow truss structure

between the length of the bar in the original configuration,

$$\ell_0 = \sqrt{b^2 + h^2} \tag{1.2}$$

and that in the current configuration,

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

can be neglected in the denominator of the expression for the strain:

$$\epsilon = \frac{\ell - \ell_0}{\ell_0} \tag{1.4}$$

or when computing the inclination angle ϕ:

$$\sin\phi = \frac{h - v}{\ell} \approx \frac{h - v}{\ell_0} \tag{1.5}$$

The dimensions b and h are defined in Figure 1.1. The vertical displacement v is taken positive in the downward sense. For half of the force in the spring we have

$$F_s = -kv \tag{1.6}$$

with k the spring stiffness, and the axial stress in the bar reads:

$$\sigma = E\epsilon \tag{1.7}$$

with E the Young's modulus. Substitution of the expressions for the stress σ, the force in the spring F_s and the angle ϕ into the equilibrium condition (1.1) yields:

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

Equation (1.8) expresses the internal force that acts in the structure as a non-linear function of the vertical displacement v. Normally, the external force at time $t + \Delta t$, $F_{\text{ext}}^{t+\Delta t}$, is given. The displacement v must then be computed such that

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

The correct value of v is computed in an iterative manner, for instance using the Newton–Raphson method:

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

with j the iteration counter. In a linear approximation we have for the iterative correction to the displacement v:

$$\mathrm{d}v = \left(\frac{\mathrm{d}F_{\text{int}}}{\mathrm{d}v}\right)_j^{-1} \left(F_{\text{ext}}^{t+\Delta t} - F_{\text{int}}(v_j)\right) \tag{1.11}$$

The iterative process is terminated when a convergence criterion has been met, $\|F_{\text{ext}}^{t+\Delta t} - F_{\text{int}}(v_j)\| < \varepsilon$, with ε a small number. For the present case the derivative $\frac{\mathrm{d}F_{\text{int}}}{\mathrm{d}v}$, or in computational mechanics terminology, the tangential stiffness modulus, can be evaluated from Equation (1.8) as:

$$\frac{\mathrm{d}F_{\text{int}}}{\mathrm{d}v} = \frac{A_0 \sin^2\phi}{\ell_0}\left(E + \frac{\mathrm{d}E}{\mathrm{d}\ell}(\ell - \ell_0)\right) + \left(k + \frac{\mathrm{d}k}{\mathrm{d}v}v\right) + \frac{A_0\sigma}{\ell_0} \tag{1.12}$$

where, for generality, it has been assumed that the stiffness of the truss as well as that of the spring depend on how much they have been extended. If this so-called material non-linearity is not present, the terms that involve $\frac{\mathrm{d}E}{\mathrm{d}\ell}$ and $\frac{\mathrm{d}k}{\mathrm{d}v}$ cancel. The last term in Equation (1.12) is due to the inclusion of large displacement/rotation effects (geometrical non-linearity), and is linear in the stress. This term is of crucial importance when computing the stability of slender structures. Figure 1.2 shows the behaviour of the truss for different values of the spring stiffness k. The graphs directly follow from application of the closed-form expression (1.8) for the internal force, in combination with the equilibrium condition (1.9). The iterative procedure can only be applied for larger values of the spring stiffness k, i.e. when there is no local maximum in the load–displacement curve.

1.2 A Review of Concepts from Linear Algebra

In computer oriented methods in the mechanics of solids frequent use is made of the concepts of a vector and a matrix. Herein, we shall denote by a vector a one-dimensional array of scalars. A scalar is a physical quantity that has the same value, irrespective of the choice of the reference frame. When we denote scalars by italic symbols and vectors by roman, bold-faced, lower-case symbols, the vector $\mathbf{v}$ has n scalar entries $v_1, \ldots, v_n$, so that:

$$\mathbf{v} = \begin{pmatrix} v_1 \\ \cdots \\ \cdots \\ v_n \end{pmatrix} \tag{1.13}$$

In Equation (1.13) the scalar entries are written in a column format. Alternatively, it is possible to write the scalar quantities $v_1, \ldots, v_n$ as a row. This row of scalars is named the transpose

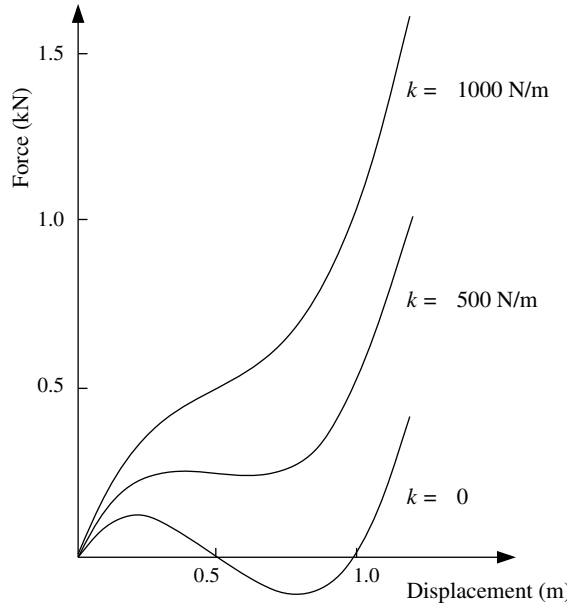

Figure 1.2 Force–displacement diagram for the shallow truss structure for different values of the spring stiffness k (b = 10 m, h = 0.5 m and EA_0 = 5 MN/m^2)

of the vector $\mathbf{v}$ and is written as:

$$\mathbf{v}^{\mathrm{T}} = (v_1, \ldots, v_n)$$

Addition of vectors is defined as the addition of their components, so that

$$\mathbf{w} = \mathbf{u} + \mathbf{v} \tag{1.14}$$

implies that $w_i = u_i + v_i$ for $i = 1, \ldots, n$. The multiplication of a vector by a scalar, say λ, is defined as:

$$\mathbf{w} = \lambda \mathbf{u} \tag{1.15}$$

with the components $w_i = \lambda u_i$.

An important operation between two vectors $\mathbf{u}$ and $\mathbf{v}$, each with n entries, is the inner product, also named scalar product:

$$\mathbf{u}^{\mathrm{T}}\mathbf{v} = \sum_{i=1}^{n} u_i v_i \tag{1.16}$$

The scalar product of two vectors possesses the commutativity property, i.e. $\mathbf{u}^{\mathrm{T}}\mathbf{v} = \mathbf{v}^{\mathrm{T}}\mathbf{u}$ as can be verified easily from the definition (1.16). The inner product can also be useful for the definition of the norm of a vector. Several definitions of the norm of a vector are possible, but in

theoretical and applied mechanics the most customary definition is the Euclidian or L_2-norm:

$$\|\mathbf{v}\|_2 = \sqrt{\mathbf{v}^{\mathrm{T}}\mathbf{v}} \tag{1.17}$$

where the subscript is often omitted. The cross product of two vectors **a** and **b**, also named the vector product, forms a vector **c**:

$$\mathbf{c} = \mathbf{a} \times \mathbf{b} \tag{1.18}$$

that is orthogonal to **a** and **b** in the three-dimensional space, and has a direction that is given by the right-hand rule, and as a consequence, is anti-symmetric: $\mathbf{b} \times \mathbf{a} = -\mathbf{a} \times \mathbf{b}$. The components of $\mathbf{c} = \mathbf{a} \times \mathbf{b}$ read:

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

The entries (or components) of a vector may be used to form a scalar function. Examples in mechanics are the invariants of the stress and strain tensors, or the yield function in plasticity. An operation that is often used is the calculation of the gradient of a function. Let the scalar-valued function f be a function of the components a_i of the vector **a**. Then, the gradient **b** is obtained by differentiation of f with respect to **a**

$$\mathbf{b} = \frac{\partial f}{\partial \mathbf{a}} \tag{1.20}$$

or in component form:

$$b_i = \frac{\partial f}{\partial a_i} \tag{1.21}$$

The gradient operation is such that **b** is orthogonal to the hypersurface in the n-dimensional vector space that is described by $f = c$, with c a constant that usually is taken equal to zero.

Matrices are another suitable mathematical vehicle that can be used in computational mechanics. While vectors in their most simple description are denoted as one-dimensional arrays of scalars, matrices are two-dimensional arrays of scalars. A matrix is said to have m rows and n columns. In general m does not have to be equal to n. If we think of vectors as matrices with only one column, a vector with m components can be termed a $m \times 1$ matrix. Similarly, a row vector with n entries can be named an $1 \times n$ matrix.

In this book we shall consistently denote a matrix by a bold-faced, upper-case symbol. The entries or components of the matrix **A** are, in a similar fashion as the components of a vector, denoted as a_{ij}, where, for an $m \times n$ matrix $i = 1, \ldots, m$ and $j = 1, \ldots, n$. A vector **b** of length n can be premultiplied by an $m \times n$ matrix **A**, as follows:

$$\mathbf{c} = \mathbf{A}\mathbf{b} \tag{1.22}$$

The resulting vector **c** has m components:

$$c_i = \sum_{j=1}^{n} a_{ij} b_j \tag{1.23}$$

The addition of two $m \times n$ matrices $\mathbf{A}$ and $\mathbf{B}$ is exactly analogous to the addition of vectors, as we have for each entry: $c_{ij} = a_{ij} + b_{ij}$, while the multiplication of a matrix by a scalar, say λ, is also defined similarly: $c_{ij} = \lambda a_{ij}$.

The product of two matrices is defined similar to the product of a matrix and a vector. Let $\mathbf{A}$ be an $m \times k$ matrix and $\mathbf{B}$ be a $k \times n$ matrix. The result of multiplying $\mathbf{A}$ and $\mathbf{B}$ is an $m \times n$ matrix $\mathbf{C}$, with components:

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

A special matrix multiplication occurs when the number of columns of $\mathbf{A}$, and consequently also the number of rows of $\mathbf{B}$, is set equal to 1 ($k = 1$). Now, $\mathbf{A}$ and $\mathbf{B}$ reduce to vectors, say $\mathbf{a}$ and $\mathbf{b}^{\mathrm{T}}$. The resulting product is still an $m \times n$ matrix,

$$\mathbf{C} = \mathbf{a}\mathbf{b}^{\mathrm{T}} \tag{1.25}$$

with components $c_{ij} = a_i b_j$. This operation is named the dyadic or outer product of two vectors $\mathbf{a}$ and $\mathbf{b}$. The transpose operation for matrices is identical to that for vectors, i.e. $\mathbf{B} = \mathbf{A}^{\mathrm{T}}$ implies that $b_{ij} = a_{ji}$. An operation that is frequently carried out in the derivation of finite element equations is taking the transpose of a product of two matrices. For such a transpose the following relationship holds:

$$(\mathbf{AB})^{\mathrm{T}} = \mathbf{B}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}} \tag{1.26}$$

The most common type of matrices are square matrices, for which $m = n$. Under certain conditions, to be discussed in the following pages, an inverse $\mathbf{B} = \mathbf{A}^{-1}$ can be defined, such that

$$\mathbf{AB} = \mathbf{I} \tag{1.27}$$

with $\mathbf{I}$ the unit matrix, i.e. all entries of $\mathbf{I}$ are zero with exception of the diagonal entries of $\mathbf{I}$ which are equal to 1: $\mathbf{I} = \mathrm{diag}[1, \ldots, 1]$. The inversion of matrices is required for the solution of large systems of linear equations which arise as a result of finite element discretisation. Such systems have the form

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

When the known coefficients $a_{11}, \ldots, a_{nn}$ are assembled in a matrix $\mathbf{A}$, the known components $b_1, \ldots, b_n$ in a vector $\mathbf{b}$, and the unknowns $x_1, \ldots, x_n$ in a vector $\mathbf{x}$, the system (1.28) can be written in a compact fashion

$$\mathbf{Ax} = \mathbf{b} \tag{1.29}$$

Formally, the vector of unknowns $\mathbf{x}$ can be obtained from

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b} \tag{1.30}$$

provided, of course, that $\mathbf{A}^{-1}$ exists. In solid mechanics the matrix $\mathbf{A}$ is often symmetric, i.e. $a_{ij} = a_{ji}$, which facilitates the computation of $\mathbf{A}^{-1}$. However, when non-linearities are incorporated in computational models, symmetry can be lost.

An efficient manner to carry out the above operation is to decompose the matrix $\mathbf{A}$ as

$$\mathbf{A} = \mathbf{LDU} \tag{1.31}$$

with $\mathbf{L}$ a lower triangular matrix

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ l_{21} & 1 & 0 & \dots & 0 \\ l_{31} & l_{32} & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ l_{n1} & l_{n2} & l_{n3} & \dots & 1 \end{bmatrix} \tag{1.32}$$

$\mathbf{U}$ an upper triangular matrix,

$$\mathbf{U} = \begin{bmatrix} 1 & u_{12} & u_{13} & \dots & u_{1n} \\ 0 & 1 & u_{23} & \dots & u_{2n} \\ 0 & 0 & 1 & \dots & u_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix} \tag{1.33}$$

and

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

a diagonal matrix. For symmetric matrices the identity $\mathbf{U} = \mathbf{L}^{\mathrm{T}}$ holds.

This LDU decomposition is based on Gauss elimination, and can preserve bandedness in the sense that if the matrix $\mathbf{A}$ has a band structure, as is normally the case in finite element applications, the lower and upper triangular matrices $\mathbf{L}$ and $\mathbf{U}$ also have a banded structure. Since

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

we can now solve for $\mathbf{x}$:

$$\begin{aligned} \mathbf{c} &= \mathbf{L}^{-1}\mathbf{b} \\ \mathbf{d} &= \mathbf{D}^{-1}\mathbf{c} \\ \mathbf{x} &= \mathbf{U}^{-1}\mathbf{d} \end{aligned} \tag{1.35}$$

This equation reveals another interesting fact. While the operations $\mathbf{L}^{-1}\mathbf{b}$ and $\mathbf{U}^{-1}\mathbf{d}$ only involve multiplications, and cannot result in arithmetic problems, the operation $\mathbf{D}^{-1}\mathbf{c}$ consists of divisions, since $\mathbf{D}^{-1} = \text{diag}[d_{11}^{-1}, \dots, d_{nn}^{-1}]$. Hence, as soon as one of the diagonal entries, named pivots, of $\mathbf{D}$ is zero, $\mathbf{x}$ can no longer be computed. In such a case the matrix $\mathbf{A}$ is said to be singular and a unique decomposition no longer exists. We distinguish between three cases: all

pivots of **D** are positive, one or more pivots of **D** are zero, and finally, one or more pivots of **D** are negative. When the diagonal matrix **D** has only positive pivots, the matrix **A** is called positive definite. An example is the stiffness matrix **A** which results from a displacement-method based finite element discretisation of a linear-elastic body. For positive-definite matrices the LDU decomposition is unique and round-off errors which arise are not amplified. When non-linear effects are introduced, the tangential stiffness matrix **A** can become singular (one or more zero pivots) during the loading process and eventually become indefinite (one or more negative pivots). As argued above, a singular matrix cannot be decomposed and meaningful answers cannot be obtained. However, a unique LDU decomposition can again be obtained if one or more pivots have turned negative, but are non-zero. Nevertheless, for indefinite matrices it cannot be ensured that round-off errors which arise during the decomposition are not amplified. In a non-linear analysis this observation implies that the iterative process that is necessary to solve the set of non-linear algebraic equations which then arises, can diverge.

Singularity of a matrix is also closely related to its determinant. The determinant of a matrix is defined as (Golub and van Loan 1983; Noble and Daniel 1969; Ortega 1987; Saad 1996)

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

where $\mathbf{A}_{ij}$ is an $(n-1) \times (n-1)$ matrix obtained by deleting the ith row and the jth column of **A**. This recursive relation is closed by $\det\mathbf{A} = a_{11}$ for $n = 1$. A useful property is that $\det(\mathbf{AB}) = \det\mathbf{A} \cdot \det\mathbf{B}$. In view of Equation (1.31) we have $\det\mathbf{A} = \det\mathbf{L} \cdot \det\mathbf{D} \cdot \det\mathbf{U}$ and from definition (1.36) we deduce that $\det\mathbf{L} = \det\mathbf{U} = 1$. We thus obtain the useful result that

$$\det\mathbf{A} = \prod_{i=1}^{n} d_i \tag{1.37}$$

which implies that the determinant of a matrix equals zero if one or more pivots are zero. In view of the discussion on pivots the matrix is then singular.

A useful result on the inversion of a special type of matrices is the Sherman–Morrison formula. Let **A** be a non-singular $n \times n$ matrix and let **u** and **v** be two vectors with n entries each. Then, the following identity holds:

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

A further useful result involving vectors is Gauss' divergence theorem. Using this theorem a volume integral can be transformed into a surface integral:

$$\int_V \mathrm{div}\mathbf{v}\,\mathrm{d}V = \int_S \mathbf{n}^{\mathrm{T}}\mathbf{v}\,\mathrm{d}S \tag{1.39}$$

where **n** is the outward normal to the bounding surface of the body, and div is the divergence operator:

$$\mathrm{div}\mathbf{v} = \frac{\partial v_1}{\partial x_1} + \frac{\partial v_2}{\partial x_2} + \frac{\partial v_3}{\partial x_3} \tag{1.40}$$

In the preceding, use has been made of the summation symbol Σ. A short-hand notation is to omit the Σ symbol and to suppose that a summation is implied whenever a subscript occurs twice in an expression. For instance, we can replace the summation in Equation (1.24) by the abbreviated notation (called the Einstein summation convention)

$$c_{ij} = a_{ie} b_{ej} \tag{1.41}$$

where summation with respect to the repeated index e is implied. Such an index is often called a 'dummy' index, since it is irrelevant which letter we take for this index. Indeed, the expression $c_{ij} = a_{iq} b_{qj}$ is identical. Of course, the indices i and j may not be replaced by other letters unless it is done on both sides of the equation. When rewriting Gauss' theorem in index notation, the result is:

$$\int_V \frac{\partial v_i}{\partial x_i} \mathrm{d}V = \int_S n_i v_i \mathrm{d}S$$

An important tensorial quantity is the Kronecker delta, defined as:

$$\begin{cases} \delta_{ij} = 1 & \text{if} \quad i = j \\ \delta_{ij} = 0 & \text{if} \quad i \neq j \end{cases} \tag{1.42}$$

As an example we note that $a_{ij} = a_{ik}\delta_{kj}$. Also useful is the permutation tensor e_{ijk}, which equals $+1$ for e_{123} and for even permutations thereof (e.g. e_{231}), and equals -1 for odd permutations (e.g. e_{213}). If two subscripts are identical, then $e_{ijk} = 0$.

In more recent years index notation has been gradually replaced by direct tensor notation, which, at first sight, somewhat resembles the matrix-vector notation. Now, the multiplication of Equation (1.24) is denoted as:

$$\mathbf{C} = \mathbf{A} \cdot \mathbf{B} \tag{1.43}$$

where the central dot denotes a single contraction, i.e. the summation over the dummy index. In a similar fashion, a double contraction is denoted as:

$$c = \mathbf{A} : \mathbf{B} \tag{1.44}$$

or using index notation: $c = a_{ie} b_{ei}$. Taking the gradient of a quantity is done using the ∇ symbol, as follows,

$$\mathbf{b} = \nabla f \tag{1.45}$$

which equals the gradient vector defined in Equation (1.20). This operator can also be used for vectors, and Gauss' theorem is now written as:

$$\int_V \nabla \cdot \mathbf{v} \mathrm{d}V = \int_S \mathbf{n} \cdot \mathbf{v} \mathrm{d}S$$

The dyadic product of two vectors $\mathbf{a}$ and $\mathbf{b}$ is now written as:

$$\mathbf{C} = \mathbf{a} \otimes \mathbf{b} \tag{1.46}$$

with components $c_{ij} = a_i b_j$. Finally, we define for a second-order tensor $\mathbf{A}$ the divergence operator

$$\mathbf{a} = \nabla \cdot \mathbf{A} \tag{1.47}$$

such that

$$a_j = \frac{\partial A_{ij}}{\partial x_i} \tag{1.48}$$

and its trace:

$$c = \mathrm{tr}(\mathbf{A}) \tag{1.49}$$

through $c = a_{ii}$.

1.3 Vectors and Tensors

So far, vectors have been introduced and treated as mere mathematical tools, arrays which contain a number of scalar quantities in an ordered fashion. Nonetheless, vectors can be given a physical interpretation. Take for instance the concept of force. A force not only has a magnitude, but also has a direction. It is often of interest to know how the components of a force change if the force is represented in a different coordinate system. A translation only adds the same number to all force components. A rotation of the reference frame, for instance from the x, y-coordinate system to a $\bar{x}$, $\bar{y}$-coordinate system, Figure 1.3, changes the components of a vector in a more complicated manner.

The components of a vector $\bar{\mathbf{n}}$ in the $\bar{x}$, $\bar{y}$-coordinate system can be obtained from those in the x, y-coordinate system, assembled in $\mathbf{n}$, by the transformation

$$\bar{\mathbf{n}} = \mathbf{R}\mathbf{n} \tag{1.50}$$

with $\mathbf{R}$ a transformation matrix. Since a full three-dimensional treatment is quite cumbersome, and hardly adds anything to the understanding, we will elaborate $\mathbf{R}$ only for planar conditions. Let the angle from the x, y-coordinate system to the $\bar{x}$, $\bar{y}$-coordinate be ϕ. For $\mathbf{n} = [1, 0]^{\mathrm{T}}$ and $\mathbf{n} = [0, 1]^{\mathrm{T}}$, respectively, the representations in the rotated coordinate system are $\bar{\mathbf{n}} = [\cos\phi, -\sin\phi]^{\mathrm{T}}$ and $\bar{\mathbf{n}} = [\sin\phi, \cos\phi]^{\mathrm{T}}$, respectively. It follows that in two dimensions the

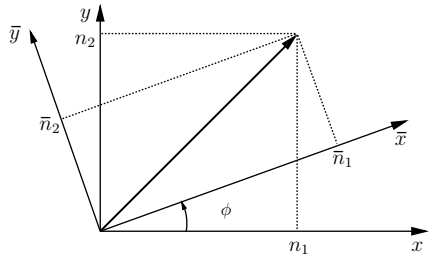

Figure 1.3 Original x, y-coordinate system and rotated $\bar{x}$, $\bar{y}$-coordinate system

transformation matrix $\mathbf{R}$ is given by

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

The transformation, or rotation matrix $\mathbf{R}$ has a special structure. Inspection shows that

$$\mathbf{R}^{-1} = \mathbf{R}^{\mathrm{T}} \tag{1.52}$$

which also holds true for the general three-dimensional case. Matrices that satisfy requirement (1.52) are called orthogonal matrices, for which $\det(\mathbf{R}) = 1$.

With the aid of the transformation rules for vectors we can derive transformation rules for tensors. Tensors, or here, more precisely, second-order tensors, are physical quantities that relate two vectors. For instance, the stress tensor sets a relation between the force on a plane and the normal vector of that plane, see also the next section. A natural representation of a second-order tensor is a matrix. However, not all matrices are tensors: only matrices that obey certain transformation rules can represent tensorial quantities. Suppose that the second-order tensor $\mathbf{C}$ relates the vectors, or first-order tensors, $\mathbf{t}$ and $\mathbf{n}$:

$$\mathbf{t} = \mathbf{C}\mathbf{n} \tag{1.53}$$

In the $\bar{x}$, $\bar{y}$ frame the second-order tensor $\bar{\mathbf{C}}$ sets a similar relation between $\bar{\mathbf{t}}$ and $\bar{\mathbf{n}}$:

$$\bar{\mathbf{t}} = \bar{\mathbf{C}}\bar{\mathbf{n}} \tag{1.54}$$

We next substitute Equation (1.50) and an identical relation for $\mathbf{t}$, i.e. $\bar{\mathbf{t}} = \mathbf{R}\mathbf{t}$, into Equation (1.54). Comparison with Equation (1.53) shows that any second-order tensor transforms according to:

$$\bar{\mathbf{C}} = \mathbf{R}\mathbf{C}\mathbf{R}^{\mathrm{T}} \tag{1.55}$$

Using Equation (1.51) this identity can be elaborated for two dimensions as

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

For symmetric second-order tensors, which will be employed here exclusively, $c_{21} = c_{12}$, and consequently also: $\bar{c}_{21} = \bar{c}_{12}$.

We observe that the components of a second-order tensor change from orientation to orientation. It is often of interest to know the extremal values of the tensor components $\bar{c}_{11}$ and $\bar{c}_{22}$, and on which plane they are attained, i.e. for which value of ϕ. For symmetric second-order tensors, there exist two mutually orthogonal planes on which $\bar{c}_{11}$ and $\bar{c}_{22}$ have a maximum and a minimum, respectively. The values in this coordinate system are commonly named the principal values. Since $\bar{c}_{11}$ and $\bar{c}_{22}$ are functions of the inclination angle ϕ these extremal values

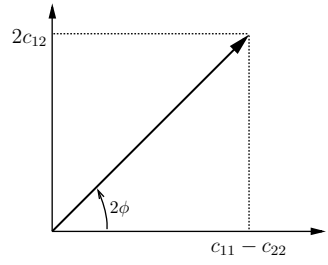

Figure 1.4 Principal directions of a second-order tensor

are obtained by requiring that

$$\frac{\partial \bar{c}_{11}}{\partial \phi} = 0 \quad \text{or} \quad \frac{\partial \bar{c}_{22}}{\partial \phi} = 0 \tag{1.57}$$

Elaborating these identities for symmetric second-order tensors we obtain that the diagonal tensor components attain extremal values for

$$\tan 2\phi = \frac{2c_{12}}{c_{11} - c_{22}} \tag{1.58}$$

To derive the principal values we first rewrite the first two equations of (1.56) as

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

From Figure 1.4, cf. Equation (1.58), we infer that

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

whence we obtain the following closed-form expression for the principal values:

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

It is a property of symmetric second-order tensors (to which the treatment will be limited) that for this inclination angle also the off-diagonal tensor components are zero: $\bar{c}_{12} = 0$. This is shown most simply by rewriting the first equation of (1.56) as:

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

whereupon substitution of the identities (1.60) proves the assertion.

Another interpretation can be given to the coordinate system in which the principal values of the diagonal tensor components attain a maximum. Let $\mathbf{C}$ be the matrix representation of a symmetric second-order tensor. Let $\boldsymbol{e}$ be a vector. As a rule, the product $\mathbf{C}\boldsymbol{e}$ will not be parallel with $\boldsymbol{e}$. However, for every such tensor there exists a coordinate system for which the resulting vector is indeed parallel with the original vector:

$$\mathbf{C}\boldsymbol{e} = \lambda \boldsymbol{e} \tag{1.63}$$

with λ the scalar-valued eigenvalue. We can rewrite Equation (1.63) as

$$(\mathbf{C} - \lambda \mathbf{I})\boldsymbol{e} = \mathbf{0} \tag{1.64}$$

with $\mathbf{I} = \text{diag}[1, \ldots, 1]$ the unit matrix. A non-trivial solution ($\boldsymbol{e} \neq \mathbf{0}$) then exists if and only if the determinant of $\mathbf{C} - \lambda \mathbf{I}$ vanishes:

$$\det[\mathbf{C} - \lambda \mathbf{I}] = 0 \tag{1.65}$$

Elaborating Equation (1.65) then yields exactly Equation (1.58). Thus, the coordinate system in which c_{11} and c_{22} attain extremal values is the same coordinate system in which a vector $\boldsymbol{e}$ multiplied by a tensor $\mathbf{C}$ results in a vector that is a multiple of $\boldsymbol{e}$. Since the eigenvalues λ_i correspond to the principal values, the eigenvectors $\boldsymbol{e}_i$ point in the principal directions. An elaboration for a symmetric second-order tensor is given in Box 1.1. Similar to pivots, Equation (1.37), a direct relationship can be established between the product of all eigenvalues and the determinant of a matrix:

$$\det\mathbf{C} = \prod_{i=1}^{n} \lambda_i \tag{1.66}$$

which is known as Vieta's rule, and is valid for symmetric and non-symmetric matrices. From Equation (1.66) we infer that the singularity of a matrix not only implies that the determinant and one or more pivots vanish, but also that at least one eigenvalue is equal to zero.

Inverting Equation (1.55) yields

$$\mathbf{C} = \mathbf{R}^{\mathrm{T}} \bar{\mathbf{C}} \mathbf{R} \tag{1.67}$$

with, in the principal axes,

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

and $\bar{c}_{11} = \lambda_1$ and $\bar{c}_{22} = \lambda_2$ the principal values or eigenvalues of $\mathbf{C}$. Elaboration of Equation (1.67) using expression (1.51) for $\mathbf{R}$ in two dimensions yields:

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

or

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

Box 1.1 Eigenvalues of a symmetric second-order tensor

For a symmetric matrix $\mathbf{C}$ the condition $\det[\mathbf{C} - \lambda\mathbf{I}] = 0$ can be elaborated as follows:

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

Solving for the eigenvalues λ yields: $\lambda_{1,2} = \frac{1}{2}(c_{11} + c_{22}) \pm \frac{1}{2}\sqrt{(c_{11} - c_{22})^2 + 4c_{12}^2}$, which are exactly the principal values of the tensor $\mathbf{C}$, see Equation (1.61). The directions of $\boldsymbol{e}$ can be computed by inserting the principal values of the tensor $\mathbf{C}$ in either

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

with e_1, e_2 the components of $\boldsymbol{e}$. Taking the first equation as an example, we can derive that substitution of the principal values $\lambda_{1,2}$ yields:

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

Bringing the re_1 term to the right-hand side, and squaring gives:

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

Simple goniometry shows that

$$\tan 2\phi = \frac{2\tan\phi}{1 - \tan^2\phi} = \frac{2e_1e_2}{e_1^2 - e_2^2}$$

which proves that Equation (1.58) also defines the directions of the eigenvectors $\boldsymbol{e}$. The notions of eigenvectors and principal directions, and of eigenvalues and principal values of symmetric second-order tensors coincide.

Identifying $\boldsymbol{e}_1^{\mathrm{T}} = (\cos\phi, \sin\phi)$ and $\boldsymbol{e}_2^{\mathrm{T}} = (\sin\phi, -\cos\phi)$ as the eigenvectors, we can represent $\mathbf{C}$ through the spectral decomposition

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

where a generalisation to n dimensions has been made. Defining the eigenprojections

$$\boldsymbol{E}_i = \boldsymbol{e}_i \otimes \boldsymbol{e}_i \tag{1.70}$$

the spectral decomposition of a symmetric, second-order tensor can also be written as:

$$\mathbf{C} = \sum_{i=1}^{n} \lambda_i \boldsymbol{E}_i \tag{1.71}$$

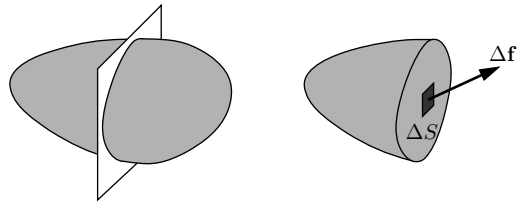

Figure 1.5 Force acting on an imaginary cut in a solid body

1.4 Stress and Strain Tensors

The basic problem of solid mechanics is to determine the response of a body to forces that are exerted onto that body. For instance, we want to know which forces act from one side of an imaginary cut in the body on the other side (Figure 1.5). It has become customary to consider a small area in that cut, say ΔS, and to investigate which force works on that area. This force is called $\Delta\mathbf{f}$. When we take the limiting case that $\Delta S \to 0$ the stress vector $\mathbf{t}$ is obtained:

$$\mathbf{t} = \lim_{\Delta S \to 0} \frac{\Delta \mathbf{f}}{\Delta S} = \frac{\mathrm{d}\mathbf{f}}{\mathrm{d}S} \tag{1.72}$$

On each plane the stress vector $\mathbf{t}$ can be decomposed in a component that acts along the normal to that plane and in two mutually orthogonal vectors which form a vectorial basis of the plane. We now choose the normal vector of this plane to coincide with the x-axis. The normal component of $\mathbf{t}$ is denoted by σ_{xx}, while the two components that lie in the plane are labelled as σ_{xy} and σ_{xz}. σ_{xy} is the stress component which acts in the direction of the y-axis and σ_{xz} is the stress component which acts in the direction of the z-axis. In accordance with the sign convention in solid mechanics the normal stress component σ_{xx} is considered positive when it points in the direction of the positive x-axis and works on a plane with a normal vector that points in the positive x-direction. In a similar fashion the shear stress σ_{xy} is taken positive when it points in the positive y-direction and acts on a plane with its normal in the positive x-direction. The definition of the other shear stress, σ_{xz}, is analogous. Along this line of reasoning the normal stress σ_{xx} is also called positive if it acts in the negative x-direction on a plane with its normal in the negative x-direction, while a positive shear stress σ_{xy} is also obtained when a shear stress acts on a plane with its normal in the negative x-direction and is directed along the negative y-axis.

In three dimensions there are nine stress components (Figure 1.6). These nine stress components fully determine the state of stress in a point of a body, and are components of the stress tensor. The stress tensor $\boldsymbol{\sigma}$ is a second-order tensor. It can be naturally expressed in matrix notation:

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

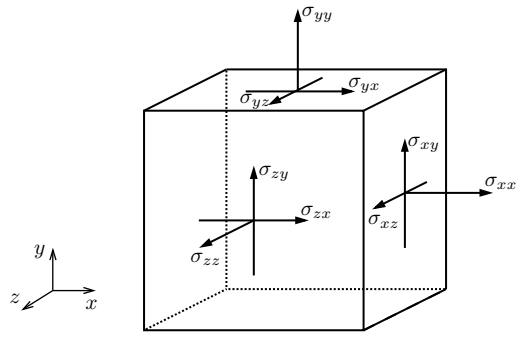

Figure 1.6 Stress components in a three-dimensional continuum

The stress tensor $\boldsymbol{\sigma}$ is related to the stress vector $\mathbf{t}$ which acts on a plane with normal $\mathbf{n}$. In matrix-vector notation, the relationship between $\boldsymbol{\Sigma}$, $\mathbf{t}$ and $\mathbf{n}$ is:

$$\boldsymbol{\Sigma}\mathbf{n} = \mathbf{t} \tag{1.74}$$

The validity of this relationship can be verified easily if the normal vector is chosen to be parallel to the x-axis ($\mathbf{n}^{\mathrm{T}} = [1, 0, 0]$), the y-axis ($\mathbf{n}^{\mathrm{T}} = [0, 1, 0]$), and the z-axis ($\mathbf{n}^{\mathrm{T}} = [0, 0, 1]$), respectively. For future use the analogue of Equation (1.74) is also given in index notation:

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

and in direct tensor notation:

$$\mathbf{n} \cdot \boldsymbol{\sigma} = \mathbf{t} \tag{1.75}$$

For a non-polar or Boltzmann continuum, the balance of moment of momentum in the three directions shows that not all the stress components are independent. In particular we find for the shear stress components that

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

(see Chapter 2 for a formal proof). Accordingly, there are six independent stress components and the matrix representation of the symmetric stress tensor $\boldsymbol{\sigma}$ can be written as

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

The observation that there are only six independent stress components makes it also feasible to write the stress tensor in a vector form (the so-called Voigt notation):

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

Note that for the vector representation the stress tensor is symbolically written as $\boldsymbol{\sigma}$ instead of $\boldsymbol{\Sigma}$ which is used for the matrix representation.

Often, for instance in geotechnical applications, it is convenient to decompose the normal stresses σ_{xx}, σ_{yy} and σ_{zz} into a deviatoric and a hydrostatic part. The deviatoric part then causes changes in the shape of an elementary cube, while the hydrostatic pressure causes a change in volume of the cube. The hydrostatic pressure is here defined as

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

With the aid of the definition of p we can define the deviatoric stress tensor. In matrix representation we have

$$\mathbf{S} = \boldsymbol{\Sigma} - p\mathbf{I} \tag{1.80}$$

while in Voigt's notation the following formula is obtained:

$$\mathbf{s} = \boldsymbol{\sigma} - p\mathbf{i} \tag{1.81}$$

where

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

Stress invariants are important quantities in non-linear constitutive theories. These are functions of the stress components that are invariant with respect to the choice of the reference frame. They arise naturally if the principal stresses in a three-dimensional continuum are computed. From the previous section it is known that the principal values λ of a second-order tensor are computed from the requirement that

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

or, in component form:

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

When we introduce the identities

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

Equation (1.84) can be reformulated as:

$$\lambda^3 - I_1\lambda^2 + I_2\lambda - I_3 = 0 \tag{1.86}$$

A crucial observation is that, since this equation has the same solution in each reference frame, I_1, I_2 and I_3 must have the same value irrespective of the choice of the reference frame. Thus,

the coefficients I_1, I_2 and I_3 must be invariant under a coordinate transformation. For this reason, I_1, I_2 and I_3 are called invariants of the stress tensor. The concept of principal values and principal directions exists for any second-order tensor, and invariants can be defined for any second-order tensor, also for the strain tensor to be treated next.

Any function of invariants is an invariant itself. Such modified invariants arise naturally if the principal values of the deviatoric stress tensor are computed. These quantities are obtained by solving the cubic equation:

$$\lambda^3 - J_2\lambda - J_3 = 0 \tag{1.87}$$

where

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

and

$$J_3 = s_{xx}s_{yy}s_{zz} + 2\sigma_{xy}\sigma_{yz}\sigma_{zx} - s_{xx}\sigma_{yz}^2 - s_{yy}\sigma_{zx}^2 - s_{zz}\sigma_{xy}^2 \tag{1.89}$$

The first invariant of the deviatoric stress tensor vanishes by definition. With the above definitions for the invariants of the stress tensor and the deviatoric stress tensor it can be shown that (Fung 1965):

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

We now consider an elementary cube which we deform only in the x, y-plane. The sides of the cube are denoted by Δx, Δy and Δz ($\Delta x = \Delta y = \Delta z$). Suppose that point A undergoes the displacements u and v and that points B and C displace as $[u + \Delta u^B, v + \Delta v^B]$ and $[u + \Delta u^C, v + \Delta v^C]$, respectively (Figure 1.7). In the limiting case that $\Delta x \to 0$ and $\Delta y \to 0$

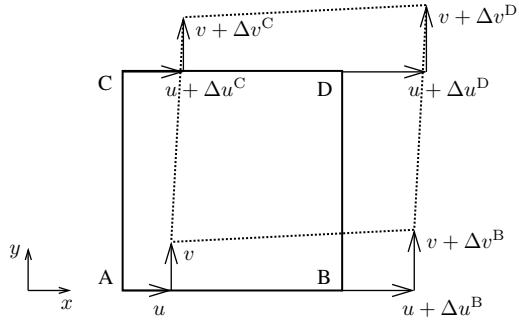

Figure 1.7 Undeformed and deformed configuration of an elementary quadrilateral

the strains in the x- and y-directions become (neglecting second-order terms):

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

The distortion of the elementary square in the x, y-plane is given by:

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

while its rotation is given by:

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

Here γ_{xy} is the total angular distortion of the elementary cube in the x, y-plane. This measure for the shear strain is often used in engineering applications. For theoretical investigations it is more customary to adopt the tensorial shear strain component $\epsilon_{xy} = \frac{1}{2}\gamma_{xy}$. In a similar fashion to which we have introduced the normal strains $\epsilon_{xx}, \epsilon_{yy}$ and the engineering shear strain γ_{xy} we can introduce the normal strain ϵ_{zz} and the shear strains γ_{yz} and γ_{zx} by considering deformations of the elementary cube in the y, z- and the z, x-planes, respectively. In accordance with the definitions (1.91) and (1.92) these strain components are defined as:

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

where w is the displacement in the z-direction. The convention for subscripts in the strain components is exactly the same as for stress components, e.g. ϵ_{xx} defines a normal strain component in the x-direction and γ_{xy} represents a shear strain component in the x, y-plane. Also the sign convention is identical: a strain component is called positive if it is related to a positive displacement of a plane with normal in the positive direction, etc. This implies for instance that elongation is considered positive.

Similar to the stress tensor we can now introduce the strain tensor. Again, matrix and vector representations are possible. For the fully three-dimensional case we have the matrix representation

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

or, noting that the strain tensor has been defined such that it is symmetric, we can write in Voigt notation:

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

While the use of the total distortion γ_{xy} etc. is more common in the Voigt notation, the tensorial shear strain ϵ_{xy} is normally used in the matrix representation. For future use we note that the rate of the internal energy per unit volume can be expressed equivalently in Voigt notation, direct tensor notation and index notation as:

$$\dot{\mathcal{W}}_{\mathrm{int}} = \dot{\boldsymbol{\epsilon}}^{\mathrm{T}}\boldsymbol{\sigma} = \dot{\boldsymbol{\epsilon}} : \boldsymbol{\sigma} = \dot{\epsilon}_{ij}\sigma_{ji} \tag{1.97}$$

In the treatment of the stress tensor the hydrostatic pressure p was introduced. Similarly, we can introduce the volumetric strain ϵ_{vol} as the sum of the normal strains:

$$\epsilon_{\mathrm{vol}} = \epsilon_{xx} + \epsilon_{yy} + \epsilon_{zz} \tag{1.98}$$

With the aid of the volumetric strain ϵ_{vol} we can define the so-called deviatoric strain tensor in a manner similar to the introduction of the deviatoric stresses:

$$\mathbf{E} = \boldsymbol{\mathcal{E}} - \frac{1}{3}\epsilon_{\mathrm{vol}}\mathbf{I} \tag{1.99}$$

or using Voigt's notation,

$$\mathbf{e} = \boldsymbol{\epsilon} - \frac{1}{3}\epsilon_{\mathrm{vol}}\mathbf{i} \tag{1.100}$$

with

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

In a preceding section the transformation rule for second-order tensors was derived, cf. Equation (1.56). Using Voigt notation, these transformation rules can, for the two-dimensional case, be written as:

$$\bar{\boldsymbol{\sigma}} = \mathbf{T}_{\sigma}\boldsymbol{\sigma} \tag{1.102}$$

with

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

with the stress tensor $\boldsymbol{\sigma}^{\mathrm{T}} = [\sigma_{xx}, \sigma_{yy}, \sigma_{xy}]$ for plane-stress conditions. By substituting $(-\phi)$ in the latter equation it is seen that $\mathbf{T}_{\sigma}^{-1} = \mathbf{T}_{\sigma}^{\mathrm{T}}$, whence

$$\boldsymbol{\sigma} = \mathbf{T}_{\sigma}^{\mathrm{T}}\bar{\boldsymbol{\sigma}} \tag{1.104}$$

Since the engineering shear strain γ_{xy} is normally used in Voigt's notation, we have for the strain transformation:

$$\bar{\boldsymbol{\epsilon}} = \mathbf{T}_{\epsilon}\boldsymbol{\epsilon} \tag{1.105}$$

with

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

and $\boldsymbol{\epsilon}^{\mathrm{T}} = [\epsilon_{xx}, \epsilon_{yy}, \gamma_{xy}]$. As for the stress transformation it holds that:

$$\boldsymbol{\epsilon} = \mathbf{T}_\epsilon^{\mathrm{T}} \bar{\boldsymbol{\epsilon}} \tag{1.107}$$

1.5 Elasticity

So far, we have introduced the stress tensor and we have considered kinematic relations, i.e. relations between displacements and strains. In Chapter 2 we will introduce the equations of motion. To complete the field equations we need stress–strain relations, or constitutive equations, which set a relation between the stress tensor and the strain tensor. For the simplest constitutive model, namely isotropic, linear elasticity (Hooke's law), the fourth-order elastic compliance tensor $\mathbf{C}^{\mathrm{e}}$ sets the relation between the strain tensor $\boldsymbol{\epsilon}$ and the stress tensor $\boldsymbol{\sigma}$:

$$\boldsymbol{\epsilon} = \mathbf{C}^{\mathrm{e}} : \boldsymbol{\sigma} \tag{1.108}$$

or in its inverse form:

$$\boldsymbol{\sigma} = \mathbf{D}^{\mathrm{e}} : \boldsymbol{\epsilon} \tag{1.109}$$

with $\mathbf{D}^{\mathrm{e}}$ the elastic stiffness tensor. In Voigt notation the compliance relation can be elaborated as:

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

with E the Young's modulus and ν the Poisson's ratio. Equation (1.110) can be written compactly as:

$$\boldsymbol{\epsilon} = \mathbf{C}^{\mathrm{e}} \boldsymbol{\sigma} \tag{1.111}$$

with $\mathbf{C}^{\mathrm{e}}$ the elastic compliance matrix.

Equation (1.110) gives the strain tensor $\boldsymbol{\epsilon}$ as a function of the stress tensor $\boldsymbol{\sigma}$. To obtain the inverse relation we rewrite the first three equations of (1.110) as

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

Next, we add these equations and, using Equations (1.79) and (1.98) we obtain:

$$\epsilon_{\text{vol}} = K^{-1} p \tag{1.113}$$

where the bulk modulus K, which sets the relation between the volumetric strain and the hydrostatic pressure, has been introduced:

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

Subsequent substitution of Equation (1.113) into Equations (1.112) and inversion yields the elastic stiffness relation:

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

where the two Lamé constants have been introduced:

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

The latter quantity is conventionally defined as the shear modulus and is also often denoted by the symbol G. The above stiffness relation can be written as

$$\boldsymbol{\sigma} = \mathbf{D}^{\text{e}} \boldsymbol{\epsilon} \tag{1.117}$$

with $\mathbf{D}^{\text{e}}$ the elastic stiffness matrix. An alternative expression for this matrix in terms of the Young's modulus and the Poisson's ratio can be obtained by inserting Equations (1.116) into

Equation (1.115)

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

1.6 The PyFEM Finite Element Library

A number of the models that are discussed in this book have been implemented in a small finite element code named PyFEM, which is written in the programming language Python. In order to demonstrate some features of this programming language in a numerical simulation, the implementation of the simple non-linear calculation in Section 1.1 is discussed in this section. The file is called `ShallowTruss.py` and can be found in the directory `examples/ch01` of PyFEM.

Instead of giving complete code listings, we will use a notation that is inspired by literate programming, see e.g. Ramsey (1994). A concise overview of this notation has been given earlier in the book on page xix. Some specific details of the notation will be highlighted in this section as well.

In literate programming the complete script can be represented as a collection of code fragments:

⟨*Shallow truss example* ⟩≡

⟨*Initialisation of the calculation* 25⟩
⟨*Step-wise calculation of the equilibrium path* 27⟩
⟨*Print results* 28⟩

This defines a fragment ⟨*Shallow truss example*⟩. The fragment itself refers to three other fragments, which are executed one after the other. Their function within the program can be deduced from their names. At this moment, this is the appropriate abstraction level.

The number behind the name of the fragment indicates the page number in this book where this fragment is discussed. Accordingly, the fragment ⟨*Initialisation of the calculation*⟩ is discussed on page 25. The absence of a page number indicates that the fragment is not discussed explicitly. One has to study the original source code to understand its functionality,

In the first fragment of this example, the variables that set the dimensions of the simulation are declared:

⟨*Initialisation of the calculation* ⟩≡ 25

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

`b` and `h` represent the dimensions of the system, b and h, as specified in Figure 1.1. The number in the right margin indicates the page number on which the fragment is mentioned before.

Obviously, the initialisation of the shallow truss example requires more than the system dimensions. We therefore extend the fragment by defining the stiffnesses k and EA_0, Equation (1.8), by writing:

⟨*Initialisation of the calculation* ⟩+≡ 25

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

The +≡ symbol after the fragment name indicates that this fragment augments a fragment defined before. We can further extend the fragment by specifying the magnitude of the incremental external force in the simulation, `DF`, the number of steps `N`, the convergence tolerance `tol`, and the maximum number of iterations to reach convergence, `iterMax`.

⟨*Initialisation of the calculation* ⟩+≡ 26

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

The actual model is defined through use of the **`lambda`** function of Python:

⟨*Initialisation of the calculation* ⟩+≡ 26

```
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)
```

Please note that in this fragment we have used a function, namely the square root operator. This function is imported from the `math` module. In PyFEM, we will often use functions from the `math`, `numpy` and `scipy` modules. In order to limit the amount code listing in this book, we will omit these import statements (**`from .. import ..`**) when possible. When the origin of a function is not exactly clear, the import statement will be listed.

Subsequently, the length of the beam l and the reaction force F of the system are defined as functions of the unknown v, see Equations (1.3) and (1.8). In the last line the derivative of the force with respect to the unknown is given as a function of the unknown v, Equation (1.12).

Finally, the parameters that are needed during the simulation are initialised:

⟨*Initialisation of the calculation* ⟩+≡ 26

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

The variables `v`, `Dv` and `Fext` represent the displacement v in the last converged solution, the incremental displacement Δv and the total external force F_{ext}, respectively. A list of lists `output` is created to store the variables that are plotted in the load–displacement curve, see Figure 1.2.

The simulation consists of a loop over `N` load steps, where `i` is the current step number. First a header is printed to denote the current load step. Then, the iterations are prepared in the fragment ⟨*Prepare iteration*⟩. Finally, the calculations are done in the fragment ⟨*Iteration*⟩:

⟨*Step-wise calculation of the equilibrium path* ⟩≡ 25

```
for i in range(N):
  print '================================'
  print '  Load step %i' % i
  print '================================'
  print '  NR iter : |Fext-F(v)|'
  ⟨Prepare iteration 27⟩
  ⟨Iteration 28⟩
```

An important feature of Python is that code blocks are defined by indentation. Code blocks are collections of statements that are executed within an **`if`** statement, within a **`for`** statement, or within a **`while`** loop. A block starts by indenting the code. A block ends by a reset to the original indent. The specific Python indentation rules also apply in the literate programming notation. In the above fragment, the new fragment ⟨*Prepare iteration*⟩ is executed within the loop **`for`** `i` **`in`** `range(N):`, since the fragment name is indented with respect to the **`for`** statement. The same holds for the fragment ⟨*Iteration*⟩.

The preparation of the iteration consists of the update of the magnitude of the external force `Fext`, and resetting the error and the iteration counter `iiter`:

⟨*Prepare iteration* ⟩≡ 27

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

The iteration itself is described in the following fragment:

⟨*Iteration* ⟩≡ 27

```
while error > tol:
  iiter += 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) ] )
```

In this fragment, the iteration counter `iiter` is initially increased by 1. The new displacement increment `dv` is calculated in the second line and added to the total increment of this step `Dv`. The error is calculated and printed in the following lines. In order to prevent the program from entering an infinite loop, a runtime error occurs when the number of iterations exceeds the maximum number of iterations. When the error is smaller than a certain tolerance, a converged solution has been found. The total displacement `v` is updated, and this displacement and the current internal force are added to the output list.

The last fragment of the ⟨*Shallow truss example*⟩ program prints the results:

⟨*Print results* ⟩≡ 25

```
from pylab import plot, show, xlabel, ylabel

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

The `output` array is plotted using `pylab` (a part of the `MatPlotLib` package). In this fragment, list comprehension has been used to create two new lists that contain the data for the horizontal and the vertical axes, respectively.

For this example problem, the exact solution `F` is known and can be added to the plot:

⟨*Print results* ⟩+≡ 28

```
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()
```

The exact solution has been calculated in the range $0.0 \leq F \leq 1.2$, with increments of 0.01. After printing the labels for the horizontal and vertical axes, the graph appears on the screen by invoking the command `show()`.

References

Fung YC 1965 *Foundations of Solid Mechanics*. Prentice-Hall.
Golub GH and van Loan CF 1983 *Matrix Computations*. The Johns Hopkins University Press.
Noble B and Daniel JW 1969 *Applied Linear Algebra*. Prentice-Hall.
Ortega JM 1987 *Matrix Theory – A Second Course*. Plenum Press.
Ramsey N 1994 Literate programming simplified. *IEEE Software* **11**, 97–105.
Saad Y 1996 *Iterative Methods for Sparse Linear Systems*. International Thomson Publishing.