D ARRAY NOTATION FOR COMPUTATIONS WITH TENSORS

This appendix describes the handling of second and fourth-order tensors in finite element computer programs. It should be particularly helpful to those wishing to follow, in program HYPLAS, the implementation of techniques discussed in this book that have been presented almost exclusively in compact tensorial notation.

In finite element computer programs, the components of a symmetric second-order tensor are usually stored as a single column array, whereas fourth-order tensor components are stored in two-dimensional arrays. By arranging the relevant components consistently, operations such as internal products between tensors and products between fourth and second-order tensors can be conveniently carried out in the computer program as matrix products.

The order in which components of a tensor can be stored in array format is not unique. In the following, we show the convention adopted in many finite element programs and, in particular, in the program HYPLAS.

D.1. Second-order tensors

Let us start with second-order tensors. Expression (2.27) (page 21) shows the matrix representation of a generic tensor in terms of its Cartesian components. Here we shall be concerned only with symmetric tensors (which are of relevance for finite element computations). Second-order symmetric tensors will be converted into single column arrays and their actual single array representation will depend on whether the tensor is a stress-like or strain-like quantity. Let us start by considering the stress tensor, $\boldsymbol{\sigma}$, in plane stress and plane strain problems. In this case, the in-plane components of the matrix $[\boldsymbol{\sigma}]$ will be converted into a single column array $\boldsymbol{\upsigma}$ (the computer array representation will be denoted here by *upright* bold-faced symbols) according to the rule

$$[\boldsymbol{\sigma}] = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{22} \end{bmatrix} \longrightarrow \boldsymbol{\upsigma} = \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{bmatrix}. \tag{D.1}$$

Computational Methods for Plasticity: Theory and Applications EA de Souza Neto, D Perić and DRJ Owen
 ISBN: 978-0-470-69452-7

Only the three relevant independent components are stored in $\boldsymbol{\sigma}$. In axisymmetric problems we store, in addition to the above, the component σ_{33}; that is

$$[\boldsymbol{\sigma}] = \begin{bmatrix} \sigma_{11} & \sigma_{12} & 0 \\ \sigma_{12} & \sigma_{22} & 0 \\ 0 & 0 & \sigma_{33} \end{bmatrix} \longrightarrow \boldsymbol{\sigma} = \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \\ \sigma_{33} \end{bmatrix}, \tag{D.2}$$

where the index 3 is associated with the circumferential direction. It should be noted here that the σ_{33} stress is generally non-zero also under plane strain conditions. In this case, the stress may be stored as in the above (i.e. including σ_{33}), but the last element of the stress array will be ignored in product operations such as (D.5).

In three-dimensions (this case is not implemented in HYPLAS), we have the conversion rule

$$[\boldsymbol{\sigma}] = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{12} & \sigma_{22} & \sigma_{23} \\ \sigma_{13} & \sigma_{23} & \sigma_{33} \end{bmatrix} \longrightarrow \boldsymbol{\sigma} = \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \\ \sigma_{23} \\ \sigma_{13} \end{bmatrix}. \tag{D.3}$$

Let us now consider the strain tensor, $\boldsymbol{\varepsilon}$. The rule for storage in this case is

$$[\boldsymbol{\varepsilon}] = \begin{bmatrix} \varepsilon_{11} & \varepsilon_{12} & \varepsilon_{13} \\ \varepsilon_{12} & \varepsilon_{22} & \varepsilon_{23} \\ \varepsilon_{13} & \varepsilon_{23} & \varepsilon_{33} \end{bmatrix} \longrightarrow \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ 2\varepsilon_{12} \\ 2\varepsilon_{23} \\ 2\varepsilon_{13} \end{bmatrix}. \tag{D.4}$$

Note that the shear components have been multiplied by a factor of two; that is, $\boldsymbol{\varepsilon}$ is the array of *engineering* strains. The reason for this is that, in this way, the internal product between a stress- and a strain-like tensor can be computed as a matrix product

$$\boldsymbol{\sigma} : \boldsymbol{\varepsilon} = \boldsymbol{\sigma}^T \boldsymbol{\varepsilon}. \tag{D.5}$$

If we denote by $\delta\boldsymbol{\varepsilon}$ a virtual strain tensor, i.e. the symmetric gradient virtual displacement field, by applying the above conversion rules the corresponding virtual work reads

$$\boldsymbol{\sigma} : \delta\boldsymbol{\varepsilon} = \boldsymbol{\sigma}^T \delta\boldsymbol{\varepsilon}. \tag{D.6}$$

In plane problems, the general conversion rule for strains is

$$[\boldsymbol{\varepsilon}] = \begin{bmatrix} \varepsilon_{11} & \varepsilon_{12} \\ \varepsilon_{12} & \varepsilon_{22} \end{bmatrix} \longrightarrow \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ 2\varepsilon_{12} \end{bmatrix}. \tag{D.7}$$

However, note that in elastoplasticity under plane strain, even though the *total* strain component ε_{33} vanishes, the corresponding elastic and plastic components do not. Thus, we adopt the following storage rule

$$[\boldsymbol{\varepsilon}^e] = \begin{bmatrix} \varepsilon^e_{11} & \varepsilon^e_{12} & 0 \\ \varepsilon^e_{12} & \varepsilon^e_{22} & 0 \\ 0 & 0 & \varepsilon^e_{33} \end{bmatrix} \longrightarrow \boldsymbol{\varepsilon}^e = \begin{bmatrix} \varepsilon^e_{11} \\ \varepsilon^e_{22} \\ 2\varepsilon^e_{12} \\ \varepsilon^e_{33} \end{bmatrix} \tag{D.8}$$

for the elastic strain (and the plastic strain). Axisymmetric implementations follow the above rule also for the array conversion of the total strain tensor, $\boldsymbol{\varepsilon}$.

D.2. Fourth-order tensors

We now consider fourth-order tensors. Let D be a tangent modulus tensor with Cartesian components D_{ijkl} on the basis $\{\boldsymbol{e}_i\}$

$$\mathsf{D} = \mathsf{D}_{ijkl}\, \boldsymbol{e}_i \otimes \boldsymbol{e}_j \otimes \boldsymbol{e}_k \otimes \boldsymbol{e}_l. \tag{D.9}$$

In tensorial compact form, the tangent stress–strain relation reads

$$\mathrm{d}\boldsymbol{\sigma} = \mathsf{D} : \mathrm{d}\boldsymbol{\varepsilon}. \tag{D.10}$$

Fourth-order tensors will be stored in two-dimensional arrays. In its array form, the components of D in plane strain problems (i.e. with $i, j = 1, 2$) will be arranged as

$$\mathbf{D} = \begin{bmatrix} \mathsf{D}_{1111} & \mathsf{D}_{1122} & \mathsf{D}_{1112} \\ \mathsf{D}_{2211} & \mathsf{D}_{2222} & \mathsf{D}_{2212} \\ \mathsf{D}_{1211} & \mathsf{D}_{1222} & \mathsf{D}_{1212} \end{bmatrix}, \tag{D.11}$$

so that the tangential relation between the in-plane stress array can be represented as the matrix product

$$\mathrm{d}\boldsymbol{\sigma} = \mathbf{D}\, \mathrm{d}\boldsymbol{\varepsilon}. \tag{D.12}$$

That the above is equivalent to (D.10) is left as an exercise for the interested reader. Note that in elasticity and associative plasticity problems the tensor D has the symmetries

$$\mathsf{D}_{ijkl} = \mathsf{D}_{jikl} = \mathsf{D}_{jilk} = \mathsf{D}_{klij}. \tag{D.13}$$

In such cases, the conversion rule produces a *symmetric* two-dimensional matrix.

In axisymmetric problems we have

$$\mathbf{D} = \begin{bmatrix} \mathsf{D}_{1111} & \mathsf{D}_{1122} & \mathsf{D}_{1112} & \mathsf{D}_{1133} \\ \mathsf{D}_{2211} & \mathsf{D}_{2222} & \mathsf{D}_{2212} & \mathsf{D}_{2233} \\ \mathsf{D}_{1211} & \mathsf{D}_{1222} & \mathsf{D}_{1212} & \mathsf{D}_{1233} \\ \mathsf{D}_{3311} & \mathsf{D}_{3322} & \mathsf{D}_{3312} & \mathsf{D}_{3333} \end{bmatrix}, \tag{D.14}$$

and in three-dimensions,

$$\mathbf{D} = \begin{bmatrix} \mathsf{D}_{1111} & \mathsf{D}_{1122} & \mathsf{D}_{1133} & \mathsf{D}_{1112} & \mathsf{D}_{1123} & \mathsf{D}_{1113} \\ \mathsf{D}_{2211} & \mathsf{D}_{2222} & \mathsf{D}_{2233} & \mathsf{D}_{2212} & \mathsf{D}_{2223} & \mathsf{D}_{2213} \\ \mathsf{D}_{3311} & \mathsf{D}_{3322} & \mathsf{D}_{3333} & \mathsf{D}_{3312} & \mathsf{D}_{3323} & \mathsf{D}_{3313} \\ \mathsf{D}_{1211} & \mathsf{D}_{1222} & \mathsf{D}_{1233} & \mathsf{D}_{1212} & \mathsf{D}_{1223} & \mathsf{D}_{1213} \\ \mathsf{D}_{2311} & \mathsf{D}_{2322} & \mathsf{D}_{2333} & \mathsf{D}_{2312} & \mathsf{D}_{2323} & \mathsf{D}_{2313} \\ \mathsf{D}_{1311} & \mathsf{D}_{1322} & \mathsf{D}_{1333} & \mathsf{D}_{1312} & \mathsf{D}_{1323} & \mathsf{D}_{1313} \end{bmatrix}. \tag{D.15}$$

Note that, according to the above rule, the fourth-order symmetric identity tensor defined by (2.108) (page 31) is represented in plane problems as

$$\mathbf{I}_S = \begin{bmatrix} 1 & 0 & 0 \\ & 1 & 0 \\ \text{sym} & & \frac{1}{2} \end{bmatrix}. \tag{D.16}$$

In using the above representation in computations, account should be taken of the fact that

$$\mathbf{I}_S\,\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{12} \end{bmatrix} \neq \boldsymbol{\varepsilon}. \tag{D.17}$$

For axisymmetric problems,

$$\mathbf{I}_S = \begin{bmatrix} 1 & 0 & 0 & 0 \\ & 1 & 0 & 0 \\ \text{sym} & & \frac{1}{2} & 0 \\ & & 0 & 1 \end{bmatrix}. \tag{D.18}$$

In three-dimensions, we have

$$\mathbf{I}_S = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ & 1 & 0 & 0 & 0 & 0 \\ & & 1 & 0 & 0 & 0 \\ & & & \frac{1}{2} & 0 & 0 \\ & \text{sym} & & & \frac{1}{2} & 0 \\ & & & & & \frac{1}{2} \end{bmatrix}. \tag{D.19}$$

D.2.1. OPERATIONS WITH NON-SYMMETRIC TENSORS

Note (refer to expressions (C.18), page 755, and (C.25)) that the product between a fourth-order tensor (the material or spatial modulus) and a generally *non-symmetric* second-order tensor (the *full* material or spatial gradient of a vector field) arises naturally in the linearisation of the virtual work equation under large deformations. In such cases, the above representation cannot be used. Modifications in the array conversion rules are needed to allow such products to be carried out in the computer program as matrix products. Let us use the spatial version (C.32) of the linearised virtual work as an example. We start by defining the second-order tensors

$$\boldsymbol{T} \equiv \nabla_x \delta \boldsymbol{u}, \quad \boldsymbol{U} \equiv \nabla_x \boldsymbol{\eta}, \tag{D.20}$$

respectively, as the (full) gradients of $\delta \boldsymbol{u}$ and the virtual displacement field $\boldsymbol{\eta}$. In plane problems, we will adopt the following computer array representation:

$$\mathbf{T} = \begin{bmatrix} T_{11} \\ T_{21} \\ T_{12} \\ T_{22} \end{bmatrix}, \tag{D.21}$$

with the same rule applying for $\boldsymbol{U}$, and

$$\mathbf{a} = \begin{bmatrix} \mathsf{a}_{1111} & \mathsf{a}_{1121} & \mathsf{a}_{1112} & \mathsf{a}_{1122} \\ \mathsf{a}_{2111} & \mathsf{a}_{2121} & \mathsf{a}_{2112} & \mathsf{a}_{2122} \\ \mathsf{a}_{1211} & \mathsf{a}_{1221} & \mathsf{a}_{1212} & \mathsf{a}_{1222} \\ \mathsf{a}_{2211} & \mathsf{a}_{2221} & \mathsf{a}_{2212} & \mathsf{a}_{2222} \end{bmatrix}. \tag{D.22}$$

With the above notation, the integrand on the right-hand side of (C.25) has the representation

$$\mathsf{a} : \boldsymbol{T} : \boldsymbol{U} = \mathbf{U}^T \mathbf{a}\, \mathbf{T}. \tag{D.23}$$

Again, note that when the fourth-order tensor has the major symmetries

$$\mathsf{a}_{ijkl} = \mathsf{a}_{klij}, \tag{D.24}$$

which occurs in hyperelasticity and hyperelastic-based associative plasticity, the conversion rule gives a symmetric two-dimensional matrix representation.

In axisymmetric problems, we adopt

$$\mathbf{T} = \begin{bmatrix} T_{11} \\ T_{21} \\ T_{12} \\ T_{22} \\ T_{33} \end{bmatrix} \tag{D.25}$$

and

$$\mathbf{a} = \begin{bmatrix} \mathsf{a}_{1111} & \mathsf{a}_{1121} & \mathsf{a}_{1112} & \mathsf{a}_{1122} & \mathsf{a}_{1133} \\ \mathsf{a}_{2111} & \mathsf{a}_{2121} & \mathsf{a}_{2112} & \mathsf{a}_{2122} & \mathsf{a}_{2133} \\ \mathsf{a}_{1211} & \mathsf{a}_{1221} & \mathsf{a}_{1212} & \mathsf{a}_{1222} & \mathsf{a}_{1233} \\ \mathsf{a}_{2211} & \mathsf{a}_{2221} & \mathsf{a}_{2212} & \mathsf{a}_{2222} & \mathsf{a}_{2233} \\ \mathsf{a}_{3311} & \mathsf{a}_{3321} & \mathsf{a}_{3312} & \mathsf{a}_{3322} & \mathsf{a}_{3333} \end{bmatrix}. \tag{D.26}$$

An analogous conversion rule can be defined for three-dimensional problems.