Convenient tensor representations for micromechanics (internal report)

Maarten Moesen

June 21, 2011

Abstract

This report presents the $\sqrt{2}$ matrix notation for second and fourth order symmetric tensors. Due to their high rank and the large numbers of indices involved, algorithms involving such tensors tend to be confusing, inefficient and prone to errors. The $\sqrt{2}$ matrix notation overcomes these problems by representing tensors by matrices such that all tensors and matrices are treated in the same consistent manner, making $\sqrt{2}$ matrix notation less error-prone than other matrix notations. Isomorphisms between the involved tensor and matrix algebra are derived, providing a toolbox for problem solving in micromechanics.

1 Introduction

1.1 Engineering notation

In micromechanics the constitutive behaviour of linear elastic materials is expressed by Hooke's law. Hooke's law relates a second order strain tensor ϵ to a second order strain tensor σ by a fourth order stiffness tensor $\mathbf{C}$: $\sigma = \mathbf{C} : \epsilon$. Expressing Hooke's law in index notation as $\sigma_{ij} = C_{ijkl}\epsilon_{kl}$ reveals that σ and ϵ have $3 \times 3 = 9$ components in three dimensions and $\mathbf{C}$ has $3 \times 3 \times 3 \times 3 = 81$ components. However, in elasticity theory it is proven that $\sigma_{ij} = \sigma_{ji}$ and $\epsilon_{ij} = \epsilon_{ji}$, that is the stress and strain tensors are symmetric and therefore have only 6 independent components. This implies for the stiffness tensor that $C_{ijkl} = C_{jikl} = C_{ijlk}$, which reduces the number of components to $6 \times 6 = 36$ components. It is further proven in elasticity theory that $C_{ijkl} = C_{klij}$, the effect of which is discussed below.

Because working with four indices is cumbersome, commonly a matrix notation is adopted which represents the stress and strain tensors as 6×1 vectors, commonly called the engineering stress vector $[\mathbf{s}]$ and engineering strain vector $[\mathbf{e}]$. The bracket notation $[\cdot]$ denotes that the enclosed tensor is represented in matrix form.

$$[\mathbf{s}] = \begin{bmatrix}\sigma_{11} & \sigma_{22} & \sigma_{33} & \sigma_{23} & \sigma_{31} & \sigma_{12}\end{bmatrix}^T, \tag{1}$$

$$[\mathbf{e}] = \begin{bmatrix}\epsilon_{11} & \epsilon_{22} & \epsilon_{33} & 2\epsilon_{23} & 2\epsilon_{31} & 2\epsilon_{12}\end{bmatrix}^T. \tag{2}$$

Note the factors 2 which occur in the shear components of strain vector $[\mathbf{e}]$, but not in the stress vector $[\mathbf{s}]$. These factors account for the symmetric terms in the tensor being represented only once, e.g. $a_{23} = a_{32} = [a]_4$, and they ensure that the deformation energy is equal in tensor and matrix notation: $\frac{1}{2}(\sigma : \epsilon) = \frac{1}{2}[\mathbf{s}]^T[\mathbf{e}]$.

Consequently the stiffness tensor C is represented by a 6×6 matrix

$$[\mathbf{C}] = \begin{bmatrix} C_{1111} & C_{1122} & C_{1133} & C_{1123} & C_{1131} & C_{1112} \\ C_{2211} & C_{2222} & C_{2233} & C_{2223} & C_{2231} & C_{2212} \\ C_{3311} & C_{3322} & C_{3333} & C_{3323} & C_{3331} & C_{3312} \\ C_{2311} & C_{2322} & C_{2333} & C_{2323} & C_{2331} & C_{2312} \\ C_{3111} & C_{3122} & C_{3133} & C_{3123} & C_{3131} & C_{3112} \\ C_{1211} & C_{1222} & C_{1233} & C_{1223} & C_{1231} & C_{1212} \end{bmatrix}. \tag{3}$$

Note that in case of additional symmetry $C_{ijkl} = C_{klij}$ this matrix is symmetric, reducing the number of independent components to 21.

Although much more practical than higher order tensors, this matrix notation is still prone to errors, because one must carefully keep track of these factors, especially when combining vectors and matrices. It becomes even more tedious when computing the compliance tensor as the inverse of the stiffness matrix. One may check that factors 2 and 4 are required as follows:

$$[\mathbf{S}] = \begin{bmatrix} S_{1111} & S_{1122} & S_{1133} & 2S_{1123} & 2S_{1131} & 2S_{1112} \\ S_{2211} & S_{2222} & S_{2233} & 2S_{2223} & 2S_{2231} & 2S_{2212} \\ S_{3311} & S_{3322} & S_{3333} & 2S_{3323} & 2S_{3331} & 2S_{3312} \\ 2S_{2311} & 2S_{2322} & 2S_{2333} & 4S_{2323} & 4S_{2331} & 4S_{2312} \\ 2S_{3111} & 2S_{3122} & 2S_{3133} & 4S_{3123} & 4S_{3131} & 4S_{3112} \\ 2S_{1211} & 2S_{1222} & 2S_{1233} & 4S_{1223} & 4S_{1231} & 4S_{1212} \end{bmatrix}. \tag{4}$$

The remainder of this report discusses an alternative matrix notation that treats these factors more consistently, keeping the matrix notation equally practical but making it less error-prone. It will be shown how to do tensor operations using matrix algebra, and implementation details will be discussed when appropriate.

2 $\sqrt{2}$ matrix notation

2.1 Tensor to matrix conversion

The idea behind $\sqrt{2}$ matrix notation is to represent symmetric tensors using matrices such that the factors are always the same for each kind of tensor. This is achieved by representing a symmetric second order tensor $a_{ij} = a_{ji}$ by the 6×1 vector

$$[\mathbf{a}] = \begin{bmatrix} a_{11} & a_{22} & a_{33} & \sqrt{2}a_{23} & \sqrt{2}a_{31} & \sqrt{2}a_{12} \end{bmatrix}^T, \tag{5}$$

and a symmetric fourth order tensor $A_{ijkl} = A_{jikl} = A_{ijlk}$ by the 6×6 matrix

$$[\mathbf{A}] = \begin{bmatrix} A_{1111} & A_{1122} & A_{1133} & \sqrt{2}A_{1123} & \sqrt{2}A_{1131} & \sqrt{2}A_{1112} \\ A_{2211} & A_{2222} & A_{2233} & \sqrt{2}A_{2223} & \sqrt{2}A_{2231} & \sqrt{2}A_{2212} \\ A_{3311} & A_{3322} & A_{3333} & \sqrt{2}A_{3323} & \sqrt{2}A_{3331} & \sqrt{2}A_{3312} \\ \sqrt{2}A_{2311} & \sqrt{2}A_{2322} & \sqrt{2}A_{2333} & 2A_{2323} & 2A_{2331} & 2A_{2312} \\ \sqrt{2}A_{3111} & \sqrt{2}A_{3122} & \sqrt{2}A_{3133} & 2A_{3123} & 2A_{3131} & 2A_{3112} \\ \sqrt{2}A_{1211} & \sqrt{2}A_{1222} & \sqrt{2}A_{1233} & 2A_{1223} & 2A_{1231} & 2A_{1212} \end{bmatrix}. \tag{6}$$

To relate tensor to matrix elements in a formal way, define the index function ι as

$$\begin{aligned} \iota : \{1..3\} \times \{1..3\} \to \{1..6\} : (i,j) \to m : \{&(1,1) \to 1, (2,2) \to 2, (3,3) \to 3, \\ &(2,3) \to 4, (3,1) \to 5, (1,2) \to 6, \\ &(3,2) \to 4, (1,3) \to 5, (2,1) \to 6\}, \text{ and} \end{aligned} \tag{7}$$

which maps symmetric tensor indices to vector indices. Accordingly define the factor function ϕ as

$$\phi : \{1..3\} \times \{1..3\} \to \{1, \sqrt{2}\} : (i,j) \to f : f(i,j) = \begin{cases} 1 & i = j, \\ \sqrt{2} & i \neq j. \end{cases} \tag{8}$$

Using these functions, second order tensor a_{ij} can be represented by vector $[a]_m$ as

$$[a]_m = [a]_{\iota(i,j)} = \phi(i,j) a_{ij}. \tag{9}$$

Likewise, fourth order tensor A_{ijkl} is represented by matrix $[A]_{mn}$ as

$$[A]_{mn} = [A]_{\iota(i,j)\iota(k,l)} = \phi(i,j)\phi(k,l) A_{ijkl}. \tag{10}$$

Because index conversions may occur frequently, it is important to implement the maps ι and ϕ efficiently. Depending on whether fast branching is supported(left) or not(right) this is done as follows:

```
function m = getSIndex(i,j)
m = bitxor(i,j);
if (m == 0)
    m = i;
else
    m = m + 3;
end
return
```

```
function m = getSIndex(i,j)
m = bitxor(i,j);
hydro = (m == 0);
m = m + hydro*i + ~hydro*3;
return
```

```
function f = getSFactor(i,j)
if (i == j)
    f = 1.0;
else
    % f = sqrt(2.0);
    f = 1.414213562373095e+00;
end
return
```

```
function f = getSFactor(i,j)
f = 1.0 + (i ~= j)*0.414213562373095e+00;
return
```

2.2 Identity tensors

The second order identity tensor $\mathbf{i}$, defined in component form as $i_{ij} = \delta_{ij}$ with δ_{ij} the Kronecker delta, is represented by the vector

$$[\mathbf{i}] = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}^T. \tag{11}$$

The fourth order identity tensor $\mathbf{I}$, defined in component form as $I_{ijkl} = \frac{1}{2}[\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}]$, has symmetries $I_{ijkl} = I_{jikl} = I_{ijlk} = I_{klij}$. The non-zero components with respect to symmetry are $I_{1111} = I_{2222} = I_{3333} = 1$ and $I_{2323} = I_{3131} = I_{1212} = \frac{1}{2}$. In matrix notation $[\mathbf{I}]$ becomes the identity matrix

$$[\mathbf{I}] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{12}$$

Note that the neutral element I_{ijkl} for double contraction of second order tensors coincides with the neutral element for matrix multiplication.

2.3 Dyadic product

The dyadic product of symmetric second order tensors $a_{ij} = a_{ji}$ and $b_{kl} = b_{lk}$ is a symmetric fourth order tensor $C_{ijkl} = a_{ij}b_{kl}$, and is written in short form as $\mathbf{C} = \mathbf{a} \otimes \mathbf{b}$. The following proves the isomorphism $[\mathbf{a} \otimes \mathbf{b}] = [\mathbf{a}] * [\mathbf{b}]^T$ between the dyadic tensor product $\otimes$ and the matrix multiplication $*$:

$$\begin{aligned}
[\mathbf{C}] &= [C]_{mn} \\
&= [a]_m [b]_n \\
&= \phi(i,j)a_{ij}\phi(k,l)b_{kl} \\
&= \phi(i,j)\phi(k,l)a_{ij}b_{kl} \\
&= \phi(i,j)\phi(k,l)C_{ijkl} \\
&= [\mathbf{C}] \,.
\end{aligned} \tag{13}$$

As a consequence, computing dyadic products in $\sqrt{2}$ matrix notation is straightforward.

2.4 Tensor contraction

The double contraction of symmetric second order tensors $a_{ij} = a_{ji}$ and $b_{kl} = b_{lk}$ is a scalar $c = a_{ij}b_{ij}$, and is written in short form as $c = \mathbf{a} : \mathbf{b}$. The double contraction is computed in $\sqrt{2}$ matrix notation using a vector dot product, and is written formally as $[\mathbf{a} : \mathbf{b}] = [\mathbf{a}]^T * [\mathbf{b}]$

$$\begin{aligned}
a_{ij}b_{ij} &= a_{11}b_{11} + a_{12}b_{12} + a_{13}b_{13} \\
&+ a_{21}b_{21} + a_{22}b_{22} + a_{23}b_{23} \\
&+ a_{31}b_{31} + a_{32}b_{32} + a_{33}b_{33} \\
&= a_{11}b_{11} + a_{22}b_{22} + a_{33}b_{33} + 2a_{23}b_{23} + 2a_{31}b_{31} + 2a_{12}b_{12} \\
&= [a]_1 [b]_1 + [a]_2 [b]_2 + [a]_3 [b]_3 + [a]_4 [b]_4 + [a]_5 [b]_5 + [a]_6 [b]_6 \\
&= [a]_m [b]_m \,.
\end{aligned} \tag{14}$$

The double contraction $c_{ij} = A_{ijkl}b_{kl}$ between fourth order tensor $A_{ijkl} = A_{jikl} = A_{ijlk}$ and second order tensor $a_{ij} = a_{ji}$ yields second order tensor $c_{kl} = c_{lk}$, and is written in short form as $\mathbf{c} = \mathbf{A} : \mathbf{b}$. This double contraction is equivalent to a matrix-vector product in $\sqrt{2}$ matrix notation as $[\mathbf{A} : \mathbf{b}] = [\mathbf{A}] * [\mathbf{b}]$:

$$\begin{aligned}
[c]_m &= [A]_{mn} [b]_n \\
&= [A]_{m1} [b]_1 + [A]_{m2} [b]_2 + [A]_{m3} [b]_3 + [A]_{m4} [b]_4 + [A]_{m5} [b]_5 + [A]_{m6} [b]_6 \\
&= \phi(i,j)A_{ij11}b_{11} + \phi(i,j)A_{ij22}b_{22} + \phi(i,j)A_{ij33}b_{33} \\
&+ 2\phi(i,j)A_{ij23}b_{23} + 2\phi(i,j)A_{ij31}b_{31} + 2\phi(i,j)A_{ij12}b_{12} \\
&= \phi(i,j)(A_{ij11}b_{11} + A_{ij12}b_{12} + A_{ij13}b_{13} \\
&+ A_{ij21}b_{21} + A_{ij22}b_{22} + A_{ij23}b_{23} \\
&+ A_{ij31}b_{31} + A_{ij32}b_{32} + A_{ij33}b_{33}) \\
&= \phi(i,j)A_{ijkl}b_{kl} \\
&= \phi(i,j)c_{ij} \\
&= [c]_m
\end{aligned} \tag{15}$$

The double contraction between symmetric fourth order tensors $A_{ijkl} = A_{jikl} = A_{ijlk}$ and $B_{ijkl} = B_{jikl} = B_{ijlk}$ yields a symmetric fourth order tensor $C_{ijkl} =$

$A_{ijpq}B_{pqkl}$, and is written in short form as $\mathbf{C} = \mathbf{A} : \mathbf{B}$. This contraction can be computed in $\sqrt{2}$ matrix notation using a matrix-matrix product $[\mathbf{A} : \mathbf{B}] = [\mathbf{A}] * [\mathbf{B}]$:

$$\begin{aligned}
[C]_{mn} &= [A]_{mr}\,[B]_{rn} \\
&= [A]_{m1}\,[B]_{1n} + [A]_{m2}\,[B]_{2n} + [A]_{m3}\,[B]_{3n} + [A]_{m4}\,[B]_{4n} + [A]_{m5}\,[B]_{5n} + [A]_{m6}\,[B]_{6n} \\
&= \phi(i,j)\phi(k,l)A_{ij11}B_{11kl} + \phi(i,j)\phi(k,l)A_{ij22}B_{22kl} + \phi(i,j)\phi(k,l)A_{ij33}B_{33kl} \\
&+ 2\phi(i,j)\phi(k,l)A_{ij23}B_{23kl} + 2\phi(i,j)\phi(k,l)A_{ij31}B_{31kl} + 2\phi(i,j)\phi(k,l)A_{ij12}B_{12kl} \\
&= \phi(i,j)\phi(k,l)(A_{ij11}B_{11kl} + A_{ij12}B_{12kl} + A_{ij13}B_{13kl} \\
&+ A_{ij21}B_{21kl} + A_{ij22}B_{22kl} + A_{ij23}B_{23kl} \\
&+ A_{ij31}B_{31kl} + A_{ij32}B_{32kl} + A_{ij33}B_{33kl}) \\
&= \phi(i,j)\phi(k,l)A_{ijpq}B_{pqkl} \\
&= \phi(i,j)\phi(k,l)C_{ijkl} \\
&= [C]_{mn}
\end{aligned} \tag{16}$$

The full contraction between symmetric fourth order tensors $A_{ijkl} = A_{jikl} = A_{ijlk}$ and $B_{ijkl} = B_{jikl} = B_{ijlk}$ yields a scalar $c = A_{ijkl}B_{ijkl}$, and is written in short form as $c = \mathbf{A} :: \mathbf{B}$. This contraction is computed in $\sqrt{2}$ matrix notation using an elementwise matrix-matrix product followed by a summation over all elements

$$\begin{aligned}
A_{ijkl}B_{ijkl} &= A_{1111}B_{1111} + A_{1112}B_{1112} + A_{1113}B_{1113} \\
&+ A_{1121}B_{1121} + A_{1122}B_{1122} + A_{1123}B_{1123} \\
&+ A_{1131}B_{1131} + A_{1132}B_{1132} + A_{1133}B_{1133} \\
&+ \dots \\
&+ A_{1211}B_{1211} + A_{1212}B_{1212} + A_{1213}B_{1213} \\
&+ A_{1221}B_{1221} + A_{1222}B_{1222} + A_{1223}B_{1223} \\
&+ A_{1231}B_{1231} + A_{1232}B_{1232} + A_{1233}B_{1233} \\
&+ A_{2111}B_{2111} + A_{2112}B_{2112} + A_{2113}B_{2113} \\
&+ A_{2121}B_{2121} + A_{2122}B_{2122} + A_{2123}B_{2123} \\
&+ A_{2131}B_{2131} + A_{2132}B_{2132} + A_{2133}B_{2133} \\
&= A_{1111}B_{1111} + A_{1122}B_{1122} + A_{1133}B_{1133} \\
&+ 2A_{1123}B_{1123} + 2A_{1131}B_{1131} + 2A_{1112}B_{1112} \\
&+ \dots \\
&+ 2A_{1211}B_{1211} + 2A_{1222}B_{1222} + 2A_{1233}B_{1233} \\
&+ 4A_{1223}B_{1223} + 4A_{1231}B_{1231} + 4A_{1212}B_{1212} \\
&= [A]_{11}\,[B]_{11} + [A]_{12}\,[B]_{12} + [A]_{13}\,[B]_{13} \\
&+ [A]_{14}\,[B]_{14} + [A]_{15}\,[B]_{15} + [A]_{16}\,[B]_{16} \\
&+ \dots \\
&+ [A]_{61}\,[B]_{61} + [A]_{62}\,[B]_{62} + [A]_{63}\,[B]_{63} \\
&+ [A]_{64}\,[B]_{64} + [A]_{65}\,[B]_{65} + [A]_{66}\,[B]_{66} \\
&= \sum_{m=1}^{6}\sum_{n=1}^{6} [A]_{mn}\,[B]_{mn}
\end{aligned} \tag{17}$$

2.5 Tensor inversion

Concerning inversion of higher order symmetric tensors, one has to distinguish between two different problems. The first problem is finding the symmetric second order tensor $x_{ij} = x_{ji}$ such that $b_{ij} = A_{ijkl}x_{kl}$ with $A_{ijkl} = A_{jikl} = A_{ijlk}$ and $b_{ij} = b_{ji}$. This problem can be solved most efficiently by solving the matrix system $[\mathbf{A}]\,[\mathbf{x}] = [\mathbf{b}]$. Most modern solvers allow to solve for multiple righthandsides $[\mathbf{b}_i]$ simultaneously.

The second problem is finding the symmetric fourth order tensor $A^{-1}_{ijkl} = A^{-1}_{jikl} = A^{-1}_{ijlk}$ such that $A^{-1}_{ijpq}A_{pqkl} = I_{ijkl} = A_{ijpq}A^{-1}_{pqkl}$ for a given fourth order tensor $A_{ijkl} = A_{jikl} = A_{ijlk}$. This problem can be solved most efficiently by solving the matrix system $[\mathbf{A}]\,[\mathbf{x}] = [\mathbf{I}]$, which requires a significantly higher amount of computation time than the first problem.

Because of the higher computational cost, but also because of additional inaccuracies introduced in computing $\mathbf{A}^{-1}$, it is inadvisable to solve the first problem by directly computing $[\mathbf{x}] = \left[\mathbf{A^{-1}}\right][\mathbf{b}]$.

2.6 Transformation of reference frame

The transformation formula for tensors is given by $a'_{ij} = n_{ik}n_{jl}a_{kl}$ with n_{ij} the cosine of the angle between the 'new' axis x'_i and the 'old' axis x_j. Converted into matrix notation the transformation formula becomes

$$\begin{aligned}
[a']_{\iota(i,j)} = \phi(i,j)a'_{ij} = \phi(i,j)n_{ik}n_{jl}a_{kl}& \\
&= \phi(i,j)(n_{i1}n_{j1}a_{11} + n_{i1}n_{j2}a_{12} + n_{i1}n_{j3}a_{13} \\
&+ n_{i2}n_{j1}a_{21} + n_{i2}n_{j2}a_{22} + n_{i2}n_{j3}a_{23} \\
&+ n_{i3}n_{j1}a_{31} + n_{i3}n_{j2}a_{32} + n_{i3}n_{j3}a_{33}) \\
&= \phi(i,j)\{n_{i1}n_{j1}a_{11} + n_{i2}n_{j2}a_{22} + n_{i3}n_{j3}a_{33} \\
&+ (n_{i2}n_{j3} + n_{i3}n_{j2})a_{23} + (n_{i3}n_{j1} + n_{i1}n_{j3})a_{31} + (n_{i1}n_{j2} + n_{i2}n_{j1})a_{12}\} \\
&= \phi(i,j)\,\{n_{i1}n_{j1}\,[a]_1 + n_{i2}n_{j2}\,[a]_2 + n_{i3}n_{j3}\,[a]_3\} \\
&+ \frac{\sqrt{2}}{2}\phi(i,j)(n_{i2}n_{j3} + n_{i3}n_{j2})\,[a]_4 \\
&+ \frac{\sqrt{2}}{2}\phi(i,j)(n_{i3}n_{j1} + n_{i1}n_{j3})\,[a]_5 \\
&+ \frac{\sqrt{2}}{2}\phi(i,j)(n_{i1}n_{j2} + n_{i2}n_{j1})\,[a]_6\,,
\end{aligned} \tag{18}$$

which can also be written as $[\mathbf{a}'] = [\mathbf{N}] * [\mathbf{a}]$ with $[\mathbf{N}]$ the transformation matrix

$$\begin{bmatrix}
n_{11}^2 & n_{12}^2 & n_{13}^2 & \sqrt{2}n_{12}n_{13} & \sqrt{2}n_{11}n_{13} & \sqrt{2}n_{11}n_{12} \\
n_{21}^2 & n_{22}^2 & n_{23}^2 & \sqrt{2}n_{22}n_{23} & \sqrt{2}n_{21}n_{23} & \sqrt{2}n_{21}n_{22} \\
n_{31}^2 & n_{32}^2 & n_{33}^2 & \sqrt{2}n_{32}n_{33} & \sqrt{2}n_{31}n_{33} & \sqrt{2}n_{31}n_{32} \\
\sqrt{2}n_{21}n_{31} & \sqrt{2}n_{22}n_{32} & \sqrt{2}n_{23}n_{33} & n_{22}n_{33} + n_{23}n_{32} & n_{23}n_{31} + n_{21}n_{33} & n_{21}n_{32} + n_{22}n_{31} \\
\sqrt{2}n_{31}n_{11} & \sqrt{2}n_{32}n_{12} & \sqrt{2}n_{33}n_{13} & n_{32}n_{13} + n_{33}n_{12} & n_{33}n_{11} + n_{31}n_{13} & n_{31}n_{12} + n_{32}n_{11} \\
\sqrt{2}n_{11}n_{21} & \sqrt{2}n_{12}n_{22} & \sqrt{2}n_{13}n_{23} & n_{12}n_{23} + n_{13}n_{22} & n_{13}n_{21} + n_{11}n_{23} & n_{11}n_{22} + n_{12}n_{21}
\end{bmatrix}. \tag{19}$$

Because the transformation matrix n_{ij} is orthogonal, its inverse is readily obtained by taking the transpose: $n^{-1}_{ij} = n_{ji}$. Observe that $[\mathbf{N}(n_{ij})]^T = [\mathbf{N}(n_{ji})]$ from which follows that $[\mathbf{N}]$ is also orthogonal:

$$[\mathbf{N}(n_{ij})]^{-1} = \left[\mathbf{N}(n_{ij}^{-1})\right] = [\mathbf{N}(n_{ji})] = [\mathbf{N}(n_{ij})]^T . \tag{20}$$

It is not difficult to show that the symmetric fourth order tensor $A_{ijkl} = A_{jikl} = A_{ijlk}$ can be transformed in matrix notation using $[\mathbf{N}]$ as $[\mathbf{A}'] = [\mathbf{N}] * [\mathbf{A}] * [\mathbf{N}]^T$.

3 Conclusion

3.1 Summary

The $\sqrt{2}$ matrix notation allows to represent symmetric higher order tensors using matrices and performing higher order algebra using matrix algebra. The following table summarizes the results of this report.

Tensor notation	$\sqrt{2}$ matrix notation
$\mathbf{i}$	$\begin{bmatrix}1 & 1 & 1 & 0 & 0 & 0\end{bmatrix}^T$
$\mathbf{I}$	$[\mathbf{I}]$
$\mathbf{a} \otimes \mathbf{b}$	$[\mathbf{a}] * [\mathbf{b}]^T$
$\mathbf{a} : \mathbf{b}$	$[\mathbf{a}]^T * [\mathbf{b}]$
$\mathbf{A} : \mathbf{b}$	$[\mathbf{A}] * [\mathbf{b}]$
$\mathbf{A} : \mathbf{B}$	$[\mathbf{A}] * [\mathbf{B}]$
$\mathbf{A} :: \mathbf{B}$	$\sum_{m=1}^{6} \sum_{n=1}^{6} [A]_{mn} [B]_{mn}$
$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$	Solve $[\mathbf{A}] [\mathbf{x}] = [\mathbf{b}]$
$\mathbf{A}^{-1}$	Solve $[\mathbf{A}] [\mathbf{x}] = [\mathbf{I}]$
$a'_{ij} = n_{ik} n_{jl} a_{kl}$	$[\mathbf{a}'] = [\mathbf{N}] * [\mathbf{a}]$
$n_{ij}^{-1} = n_{ji}$	$[\mathbf{N}]^{-1} = [\mathbf{N}]^T$
$A'_{ijkl} = n_{ip} n_{jq} n_{kr} n_{ls} A_{pqrs}$	$[\mathbf{A}'] = [\mathbf{N}] * [\mathbf{A}] * [\mathbf{N}]^T$

3.2 Getting started

Implementing microstructural models is not easy, and here's some advice for getting started. Consider using MATLAB, OCTAVE or another linear algebra environment of your choice. Using a reliable and user-friendly environment will save lots of time and frustration. If you nevertheless wish to use high-level programming languages such as FORTRAN or C++, do not write a single linear algebra routine yourself, but rather use *quality* libraries such as BLAS, LAPACK, ATLAS, NAG or MTL to name a few. This will make your code run faster, your programming easier and your results more accurate and reliable.