Textures and Microstructures, 1992, Vol. 19, pp. 147–167
Reprints available directly from the publisher.
Photocopying permitted by license only

TENSORIAL REPRESENTATION OF THE ORIENTATION DISTRIBUTION FUNCTION IN CUBIC POLYCRYSTALS

MAURIZIO GUIDI, BRENT L. ADAMS and E. TURAN ONAT

Department of Mechanical Engineering, Yale University, New Haven, CT 06520 USA

(22 July 1991)

A precise definition for the crystallite orientation distribution function (codf) of cubic polycrystals is given in terms of the set of distinct orientations of a cube. Elements of the classical Fourier representation of the codf, in terms of (symmetrized) generalized spherical harmonics, are reviewed. An alternative Fourier representation is defined in which the coefficients of the series expansion are irreducible tensors. Since tensors can be defined without the benefit of a coordinate frame, the tensorial representation is coordinate free. A geometrical association between irreducible tensors and a bouquet of lines passing through a common origin is discussed. Algorithms are given for computing the irreducible tensors and basis functions for cubic polycrystals.

KEY WORDS Crystallite orientation distribution function, Fourier representation, tensors.

INTRODUCTION

A traditional role of the scientist is to construct predictive models of physical systems. When solids are of interest, these models often attempt to relate material properties to a set of parameters which define the current state and orientation of the material. It is known that state variables relating to mechanical and physical behavior of materials are tensorial in character (cf. Geary and Onat, 1979; Adams *et al.*, 1991). An essential characteristic of tensorial state variables is that they accept a coordinate-free interpretation which is independent of the observers choice of coordinate system.

The well-known crystallite orientation distribution function (codf) contains first-order statistical information about the distribution of interatomic bonding within polycrystals (Bunge, 1965; Roe, 1965). It is therefore not surprising that the codf would correlate reasonably well with a number of important physical and mechanical properties. It is also no surprise that the codf is not adequately robust, as a set of state variables, when some finer aspects of material behavior are of concern. In particular, it is known that other aspects of microstructure (e.g. the shapes, sizes, geometry and states of dislocation of constituent crystallites) also affect properties. A deep understanding of microphysics and micromechanics of polycrystals is required to decide which additional features of microstructure are relevant for the more complete description of current state in the material.

In this paper we present a viewpoint of the codf as one important aspect of a much richer set of tensorial state variables required in a more faithful repre-

sentation of the relevant aspects of mechanical and physical behavior of polycrystals. A precise definition of the codf, useful in the developments which follow, is presented; for pedagogical clarity the focus is on cubic crystallite symmetry. Classical Fourier representation in terms of generalized spherical harmonic (gsh) functions is reviewed. A new tensorial representation is then introduced which derives from the theory of group representations. Thus, the codf is represented, to within any desired degree of accuracy, by the leading irreducible tensors of a sequence of tensorial coefficients of the Fourier series. A few examples are given which illustrate the use of this new representation in well-known applications. Algorithms are presented for constructing irreducible tensors with cubic symmetry and the basis functions of the tensorial representation.

THE CRYSTALLITE ORIENTATION DISTRIBUTION FUNCTION

Consider a macroscopically homogeneous polycrystal composed of very many crystallites. For definiteness assume that the lattice of each crystallite is cubic; in other words, the lattice of the crystallite remains unchanged after the application of any combination of the 24 rotations which take the cubical unit cell into itself. We shall not consider other geometrical aspects of the boundary of these crystallites or the state of dislocation which will not be invariant under these same rotations. (The 24 rotations of a cube are associated with a point-symmetry group better known as 432 in the International or O in the Schoenflies conventions. We shall use the Schoenflies convention in this paper.)

The orientation of a crystallite is defined by the orientation of its cubic unit cell. Thus we shall be interested in the set E_c composed of distinct orientations of a cube. (Refection will show that this set constitutes a smooth, three-dimensional manifold with no boundary (cf. Altmann, 1986). It is also isomorphic to the orbit of a fourth-rank tensor which has the symmetries of a cube. This fact will become more evident later in this paper.) Observe that given two orientations of a cube, x_0 and x, there exists a rotation $g \in SO(3)$ such that a cube in the orientation x_0 will move to the orientation x under the application of g. Here $SO(3)$ denotes the special orthogonal group of rotations. In a more formal language, there exists a mapping $P_g : E_c \to E_c$ such that for any pair of orientations $x, x_0 \in E_c$

$$x = P_g x_0 \quad \text{for some} \quad g \in SO(3). \tag{1}$$

It is easily seen that the mapping P_g satisfies the composition rule

$$P_g P_h = P_{gh}, \tag{2}$$

where $g, h \in SO(3)$.

Now consider the subgroup $H_0 \subset SO(3)$ defined such that

$$P_h x_0 = x_0 \quad \text{for } h \in H_0. \tag{3}$$

Thus H_0 is isomorphic with the O point-symmetry group, and consists of 24 rotations which leave a cube with orientation x_0 unchanged. Note that $P_{gh} = P_g$ when h belongs to the subgroup H_0.

Define the crystallite orientation distribution function

$$\hat{f} : E_c \to \mathbb{R}^+ \tag{4}$$

where $\hat{f}(x)$ measures the volume fraction density of crystallites exhibiting the orientation x. More precise definition of $\hat{f}(x)$ requires consideration of an infinitesimal neighborhood $\hat{N}(x)$ of x in E_c. Let dx be the *appropriately defined volume* of this neighborhood, and $V(\hat{N}(x))$ the volume of crystallites in a material sample of volume V which have orientations in the neighborhood $\hat{N}(x)$ of E_c. Exactly how dx is to be defined is taken up later. $\hat{f}(x)$ is defined by the expression

$$\hat{f}(x)\,dx = V(\hat{N}(x))/V. \tag{5}$$

It is clear from Eq. (5) that the normalization of the codf is

$$\int_{E_c} \hat{f}(x)\,dx = 1. \tag{6}$$

A codf is typically obtained from measurements performed on plane sections through the material element. Several smoothing operations are often applied to the data. We shall assume in this paper that the codf is continuous.

The preceding definition is not the typical definition of the codf found in the literature. A related function f, also called the codf, is defined on $SO(3)$ by choosing a reference element x_0 in E_c and by creating

$$f: SO(3) \to \mathbb{R}^+ \tag{7}$$

with the definition

$$f(g) = \hat{f}(x), \quad x = P_g x_0. \tag{8}$$

The advantage of using $f(g)$ over $\hat{f}(x)$ is that the representation of functions defined on $SO(3)$ is a well studied problem of group theory (cf. Gel'fand, Minlos and Shapiro, 1963). A seeming disadvantage of $f(g)$ is that it depends upon the choice of reference orientation x_0.

It is evident that there exists a neighborhood $N(g)$ of g in $SO(3)$ such that

$$\hat{N}(x) = \{P_{g'} x_0 : g' \in N(g)\}. \tag{9}$$

In fact, there will be 24 neighborhoods $N(gh)$, $h \in H_0$, all of which satisfy the same condition of relation (9). Let dg be the volume of $N(g)$ in $SO(3)$ which is *appropriate to invariant integration over* $SO(3)$. It follows (cf. Gel'fand, Minlos and Shapiro, 1963) that

$$\int_{SO(3)} f(gg_0)\,dg = \int_{SO(3)} f(g)\,dg \quad \text{for any} \quad g_0 \in SO(3). \tag{10}$$

Invariant integration in several parameterizations of $SO(3)$ is an established procedure (cf. Bunge, 1982).

Now return to the question of defining the volume element dx of the neighborhood $\hat{N}(x)$. We choose the volume dx as follows:

$$dx = dg. \tag{11}$$

With this definition it is clear from (8), (10) and (11) that there exists an invariant integration over E_c of the form:

$$\int_{E_c} \hat{f}(P_g x)\,dx = \int_{E_c} \hat{f}(x)\,dx \quad \text{for any} \quad g \in SO(3). \tag{12}$$

It is also evident from (3) and (6) that

$$\int_{SO(3)} f(g)\, dg = 24. \tag{13}$$

For the Euler angle parameterization of orientations it has been customary to partition $SO(3)$ into "fundamental regions" or "asymmetric domains" which are isomorphic with E_c (cf. Bunge, 1982; Hansen, Pospiech and Lucke, 1978). Integration over such regions has the value of unity. The topology of these asymmetric domains is complicated but well known, and we shall not discuss specific parameterizations in this paper.

Let us now observe some properties of the codf, $f(x)$. From Eq. (3) we see that

$$f(gh) = f(g) \quad \text{for} \quad h \in H_0 \quad \text{and} \quad g \in SO(3). \tag{14}$$

This relation associates the symmetry of the codf with lattice symmetry.

Now consider rotations of the material sample. Let k denote this rotation, and let $P_k f$ denote the codf of this rotated material element. It is known (cf. Altmann, 1986, p. 36) that

$$P_k f(g) = f(k^{-1}g) \quad \text{for} \quad k, g \in SO(3). \tag{15}$$

One can regard the above P_k as a map on the linear space $C(SO(3), \mathbb{R})$ of real valued functions on $SO(3)$ that takes f to $P_k f$. $\mathbb{R}$ here denotes the real numbers.

Now consider the often-encountered case where the material sample possesses symmetries with respect to rotations. Let K denote the symmetry subgroup of the material element. From (15) and the fact that $k^{-1} \in K$ it follows that

$$f(g) = f(kg) \quad \text{for} \quad k \in K \quad \text{and} \quad g \in SO(3). \tag{16}$$

For material elements possessing both lattice and sample symmetries the two effects can be combined:

$$f(g) = f(kgh) \quad \text{for} \quad k \in K, \quad h \in H_0 \quad \text{and} \quad g \in SO(3). \tag{17}$$

CLASSICAL REPRESENTATION OF THE CODF FOR CUBIC MATERIALS

Consider now any real-valued function f defined on $SO(3)$, of which the codf is one example. Let f be continuous or square integrable over $SO(3)$. Thus

$$f \in C(SO(3), \mathbb{R}) \quad \text{or} \quad f \in L^2(SO(3), \mathbb{R}). \tag{18}$$

It is known from group theory that f can be represented by the convergent Fourier series

$$f = f_0 + f_1 + f_2 + \cdots + f_l + \cdots \tag{19}$$

where $f_l \in M_l$ and M_l is a $(2l+1)^2$ dimensional subspace of the linear space $C(SO(3), \mathbb{R})$. Moreover, M_l is $SO(3)$ invariant; this means that if $f \in M_l$ then $P_g f \in M_l$ whenever $g \in SO(3)$. It is also known that the subspaces $\{M_l : l = 0, 1, 2, \ldots\}$ are mutually orthogonal with respect to the inner product

$$\langle f, f' \rangle = \int_{SO(3)} f(g) f'(g)\, dg \tag{20}$$

where an invariant integration is performed. The orthogonality has the form

$$\langle f, f' \rangle = 0 \quad \text{when} \quad f \in M_l, \qquad f' \in M_{l'} \quad \text{and} \quad l \neq l'. \tag{21}$$

It is well known that the gsh's $T_l^{mn}(g)$ constitute a basis for $C(SO(3), \mathbb{C})$ where $\mathbb{C}$ denotes the complex numbers. Real valued functions $t_l^{mn}(g)$ are readily fabricated from the complex valued ones (Bunge, 1982, p. 356). These form a set of $(2l+1)^2$ functions $\{t_l^{mn}(g)\colon m, n = -l, \ldots, 0, \ldots, l\}$ which are *mutually orthogonal* and constitute a basis for the linear space M_l. Thus, the Fourier series of (19) takes the form

$$f(g) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \sum_{n=-l}^{l} C_l^{mn} t_l^{mn}(g). \tag{22}$$

When it is desired that the codf carry the symmetries of the crystal lattice and the sample, as expressed in Eqs (14), (16) and (17), then it is discovered that functions in M_l which have the desired symmetries constitute an $SO(3)$ invariant subspace, $\hat{M}_l \subset M_l$. A basis for $\hat{M}_l$ can be constructed by considering linear combinations of the original basis $\{t_l^{mn}(g)\}$ (cf. Bunge, 1982, ch. 14). It can be shown that for cubic materials the Fourier series must have the form

$$f = \hat{f}_0 + \hat{f}_4 + \hat{f}_6 + \hat{f}_8 + \hat{f}_9 + \hat{f}_{10} + \hat{f}_{12} + \cdots \tag{23}$$

where $\hat{f}_l \in \hat{M}_l$.

Consider now the case where only lattice symmetry is accounted for according to (14). When $l = 0, 4, 6, 8, 9$, and 10, $\hat{M}_l$ is a $(2l+1)$ dimensional subspace of M_l. However, when $l = 12$ (for example) $\hat{M}_{12}$ is a $2(2 \times 12 + 1) = 50$ dimensional subspace of M_{12} which is a $(2 \times 12 + 1)^2 = 625$ dimensional space. All of these subspaces belong to $C(SO(3), \mathbb{R})$ and have the common property that their elements satisfy the symmetry condition expressed in (14). When symmetries of the material element are considered, in addition to lattice symmetry, the appropriate subspaces of M_l reduce further in their dimension.

TENSORIAL REPRESENTATION OF THE CODF FOR CUBIC MATERIALS

The major purpose of this paper is to define an alternative Fourier representation of the codf for cubic polycrystals in terms of a new set of basis functions. We shall demonstrate that these basis functions possess the lattice symmetry property expressed in (14). Further, the Fourier series representation of an arbitrary function in $C(SO(3), \mathbb{R})$ in this new basis has coefficients which are irreducible tensors. The codf, in particular, can be represented to within a desired degree of accuracy by the leading irreducible tensors of the sequence of tensorial coefficients of the Fourier series. In this form the state-variable character of the codf is properly emphasized. These results are achieved by pursuing steps suggested by the theory of group representations (cf. Gel'fand, Minlos and Shapiro, 1963; Bröcker and tom Dieck, 1985).

First we generate, with the help of an lth rank completely-symmetric and traceless tensor, a $(2l+1)$ dimensional irreducible subspace of $C(SO(3), \mathbb{R})$ whose elements satisfy the invariance requirement (14). (Notice that only the lattice symmetry is considered in creating these basis functions; sample symmetry

will be carried in the tensorial coefficients as described later.) A completely symmetric and traceless tensor of rank $l>1$, $\mathbf{t}\in \mathsf{T}^s_l$, has components in a rectangular frame which satisfy the relations

$$t_{i_1 i_2 \cdots i_l} = t_{p\{i_1 i_2 \cdots i_l\}}, \qquad t_{i_k i_k i_3 \cdots i_l} = 0 \tag{24}$$

where the indices i_k take on the values 1, 2, and 3, $p\{x\}$ denotes permutation of x, and repeated indices imply summation. It is easily seen from (24) that $\mathbf{t}$ has $(2l+1)$ independent components and the linear space T^s_l is therefore $(2l+1)$ dimensional.

Next recall the familiar linear tensor transformation

$$P_g : \{\mathsf{T}^s_l \rightarrow \mathsf{T}^s_l,\ g \in SO(3)\} \tag{25}$$

which takes the tensor $\mathbf{t}$ with components $t_{i_1 i_2 \cdots i_l}$ into the tensor $\mathbf{t}'$ with components

$$t'_{j_1 j_2 \cdots j_l} = g_{j_1 i_1} \cdots g_{j_l i_l} t_{i_1 i_2 \cdots i_l} \tag{26}$$

where g_{ij} are components of the orthogonal transformation $g \in SO(3)$. (Notice that g_{ij} are not the direction cosines which pertain to transformations of coordinate frame, such as is common in the classical texture analysis; rather, they are elements of the active tensor transformation.) It is known (cf. Gel'fand, Minlos and Shapiro, 1963, p. 58) that relation (25) is a $(2l+1)$ dimensional *irreducible representation* of $SO(3)$. Hence the name irreducible tensor for the elements of T^s_l.

Irreducible tensors have many useful properties. For instance, it has been known since the times of Maxwell (cf. Courant and Hilbert, 1937; Backus, 1970; Onat, 1986) that an irreducible tensor $\mathbf{t}\in \mathsf{T}^s_l$ defines uniquely a bouquet of l line segments of equal length, and an assignment of directions on these lines. This assignment is not unique. Simultaneous reversal of any two directions results in an acceptable set of directions.

Thus a traceless and symmetric second rank tensor has five components and will be associated with a bouquet of two lines. It is important to realize that the tensor will have the symmetries of this bouquet. Notice that rotations of π about the bisector of this 2-bouquet does not alter the tensor. Similarly, rotations of π about an axis perpendicular to the plane defined by the 2-bouquet, or about an axis orthogonal to these first two axes, does not alter the tensor. (These actions are equivalent to rotations of π about the three principal directions of the tensor.) Therefore second rank tensors have at least the symmetries of an orthotropic body. Notice, however, that the 2-bouquet does not carry the symmetry of a cube.

On the other hand, a fully symmetric and traceless fourth-rank tensor, which has nine independent components, is associated with a 4-bouquet. It follows that such a tensor would have no generic symmetry. Symmetric 4-bouquets can be envisioned which do represent a particular class of symmetries. In particular, the 4-bouquet composed of the body diagonals of a cube carry all of the symmetries of a cube. This is illustrated in Figure 1.

Now we come to a crucial step in the use of irreducible tensors. We pick a particular tensor $\mathbf{t}\in \mathsf{T}^s_l$. Observe (cf. Bröcker and tom Dieck, 1985, ch. 3) that as g takes on values in $SO(3)$, $P_g\mathbf{t}$ defines through its components, in the manner

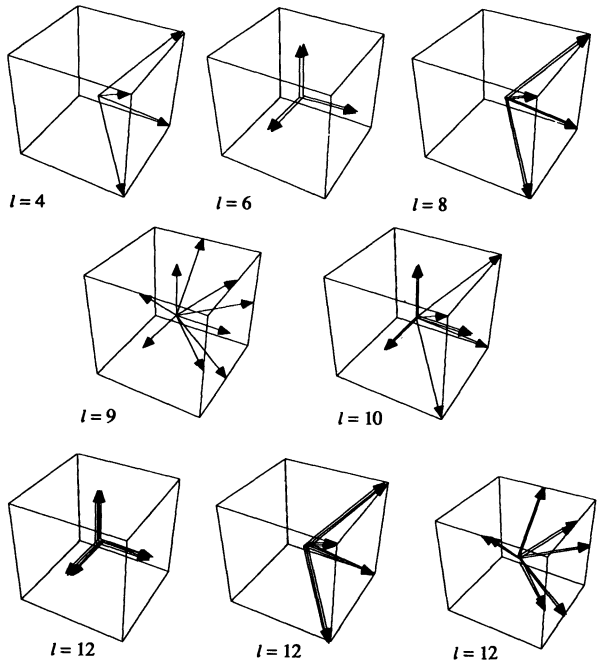

Figure 1 Bouquets of vectors associated with tensors possessing cubic symmetry.

described by Eq. (26), $(2l+1)$ linearly independent functions ϕ_i in $C(SO(3), \mathbb{R})$:

$$\phi_i : SO(3) \to \mathbb{R}, \qquad i = 1, 2, \ldots, (2l+1). \tag{27}$$

To illustrate this step consider the case of $l = 2$ as an example. Then $P_g\mathbf{t}$ would have components $g_{ip}g_{jq}t_{pq}$ in a selected rectangular coordinate frame. These components define the following nine functions on $SO(3)$:

$$f_{ij}(g) = g_{ip}g_{jq}t_{pq}, \qquad g \in SO(3). \tag{28}$$

Since $\mathbf{t}$ is traceless and symmetric only five of these functions, say $\{\phi_1 = f_{11}(g), \phi_2 = f_{22}(g), \phi_3 = f_{12}(g), \phi_4 = f_{23}(g), \phi_5 = f_{31}(g)\}$, are linearly independent.

Now if we *choose* $\mathbf{t} \in \mathsf{T}_l^s$ in such a way that

$$P_h\mathbf{t} = \mathbf{t} \quad \text{for} \quad h \in H_0, \tag{29}$$

where H_0 is the lattice symmetry subgroup that occurs in (14), then in view of the composition rule expressed in (2), the functions ϕ_i will satisfy the conditions

$$\phi_i(g) = \phi_i(gh), \qquad g \in SO(3), \qquad h \in H_0. \tag{30}$$

The span of $\{\phi_1, \phi_2, \ldots, \phi_{(2l+1)}\}$ defines a $(2l+1)$ dimensional irreducible subspace of $C(SO(3), \mathbb{R})$, and the elements of this subspace fulfill the invariance requirement of relation (14).

We must now realize that there may be a number of linearly independent tensors in T_l^s which have the desired property (29). Let $\{\mathbf{t}^{(l,t)}, \ldots, \mathbf{t}^{(l,Q)}\}$ be a maximal linearly-independent set in T_l^s satisfying (29), where $Q = Q(l)$ is the

number of linearly-independent tensors of rank l. (Later we discuss how to construct such a set with the help of geometric representations of irreducible tensors.) In the same spirit as Eq. (28) construct Q sets of functions from $P_g\{\mathbf{t}^{(l,1)}, \ldots, \mathbf{t}^{(l,Q)}\}$:

$$f^{(l,\alpha)}_{i_1\cdots i_l}(g) = g_{i_1 j_1}\cdots g_{i_l j_l} t^{(l,\alpha)}_{j_1\cdots j_l}, \qquad \alpha = 1, 2, \ldots, Q(l). \tag{31}$$

It follows from (30) and the theory of representation of Lie groups that the span of these functions, for specified rank l, is the $SO(3)$ invariant subspace $\hat{M}_l \subset M_l$ possessing the lattice symmetry:

$$\text{Span}\,\{f^{(l,\alpha)}_{i_1\cdots i_l}(g) : i_k = 1, 2, 3;\ \alpha = 1, 2, \ldots, Q(l)\} = \hat{M}_l. \tag{32}$$

We emphasize that the present basis differs from the symmetrized gsh.

Now consider the construction of irreducible tensors which have the property of relation (30). As previously noted an irreducible tensor of rank l defines uniquely a bouquet of l equal segments and an assignment of directions on these segments. This assignment is not unique; simultaneous reversal of any two directions results in an acceptable set of directions. A few minutes of contemplation of this situation leads the investigator to conclude that no tensors of rank 1, 2, 3, 5, 7, etc. can satisfy relation (30). However, when $l = 4$ a fourth-rank tensor $\mathbf{t} \in \mathrm{T}^s_4$ generated by the four vectors $\{\mathbf{v}^1, \mathbf{v}^2, \mathbf{v}^3, \mathbf{v}^4\}$ defined by the body diagonals of the unit cube, as illustrated in Figure 1, has the required symmetry and satisfies relation (30). The components of this tensor are

$$\begin{aligned} t^{(4,1)}_{ijkl} = &\sum_{p\{\alpha\beta\gamma\delta\}} v^\alpha_i v^\beta_j v^\gamma_k v^\delta_l - \frac{1}{7}(\mathbf{v}^\alpha \cdot \mathbf{v}^\beta)(\delta_{ij}v^\gamma_k v^\delta_l + \delta_{ik}v^\gamma_j v^\delta_l + \delta_{il}v^\gamma_j v^\delta_k \\ &+ \delta_{jk}v^\gamma_i v^\delta_l + \delta_{jl}v^\gamma_i v^\delta_k + \delta_{kl}v^\gamma_i v^\delta_j) \\ &+ \frac{1}{7\times 5}(\mathbf{v}^\alpha \cdot \mathbf{v}^\beta)(\mathbf{v}^\gamma \cdot \mathbf{v}^\delta)(\delta_{ij}\delta_{kl} + \delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}) \\ &i, j, k, l = 1, 2, 3; \qquad \alpha, \beta, \gamma, \delta = 1, 2, 3, 4. \end{aligned} \tag{33}$$

One can easily check that (33) has the properties of (24), and the symmetry of relation (30). Only one such tensor can be found; i.e., $Q(4) = 1$.

The geometric images of tensors of rank 6, 8, 9, and 10 which have the required symmetry can also be constructed using various directions defined in Figure 1. In each of these cases $Q(l) = 1$. After careful observation (Seibert, 1991), one obtains the following general expression for tensors satisfying relation (24):

$$\begin{aligned} t^{(l,\alpha)}_{i_1 i_2\cdots i_l} = &\sum_{p\{j_1 j_2\cdots j_l\}} \left\{ v^{j_1}_{i_1} v^{j_2}_{i_2}\cdots v^{j_l}_{i_l} + \sum_{k=1}^{l/2} \frac{(-1)^k}{\prod_{n=1}^k (2(l-n)+1)} \right. \\ &\left. \times \left\{ \left(\prod_{n=1}^{k} (v^{j_{2n}} \cdot v^{j_{2n-1}}) \right) \sum_{d_{2k}\{i_1 i_2\cdots i_l\}} \left[\left(\sum_{d_2\{i_1 i_2\cdots i_{2k}\}} \prod_{n=1}^{k} \delta_{i_{2n-1} i_{2n}} \right) \prod_{n=2k+1}^{l} v^{j_n}_{i_n} \right] \right\} \right\} \end{aligned} \tag{34}$$

where $p\{\ \}$ again denotes permutations, and $d_k\{i_1 \cdots i_l\}$ indicates the *disposition* of l objects taken k at a time regardless of the order (i.e., all possible distinct subsets of k elements taken from the set $\{i_1 \cdots i_l\}$).

When the rank $l = 12$ it is possible to construct three distinct tensors with the required symmetry. This is illustrated in Figure 1. Further work shows that only

two of these tensors are linearly independent, however, and thus $Q(12) = 2$. In fact, we have found that $Q(l)$ corresponds exactly with $M(l)$ in the classical formulation (cf. Bunge, 1982, Figures 4.4 and 14.1). It follows that a Fourier series in terms of the new basis must have the same form as the classical series, as expressed in relation (23).

From (24) and (31) it is clear that

$$f^{(l,\alpha)}_{i_1\cdots i_l}(g) = f^{(l,\alpha)}_{p\{i_1\cdots i_l\}}(g), \quad \text{and} \quad f^{(l,\alpha)}_{i_1 i_1 i_3\cdots i_l}(g) = 0. \tag{35}$$

Furthermore, $\hat{f}_l$ of relation (23) can be expressed as

$$\hat{f}_l(g) = \sum_{\alpha=1}^{Q(l)} V^{(l,\alpha)}_{i_1\cdots i_l} f^{(l,\alpha)}_{i_1\cdots i_l}(g), \tag{36}$$

where summation over the repeated (subscripted) indices is implied. Owing to the properties of the basis functions expressed in (35) it follows that the same symmetry can be applied to the coefficients:

$$V^{(l,\alpha)}_{i_1\cdots i_l} = V^{(l,\alpha)}_{p\{i_1\cdots i_l\}}, \quad \text{and} \quad V^{(l,\alpha)}_{i_1 i_1 i_3\cdots i_l} = 0. \tag{37}$$

When this choice has been made the coefficients $V^{(l,\alpha)}_{i_1\cdots i_l}$ will be uniquely determined from a knowledge of the codf, $f(g)$:

$$f(g) = \sum_{l=0,4,\ldots}^{\infty} \sum_{\alpha=1}^{Q(l)} V^{(l,\alpha)}_{i_1\cdots i_l} f^{(l,\alpha)}_{i_1\cdots i_l}(g). \tag{38}$$

An advantage of the tensorial representation is that the coefficients $V^{(l,\alpha)}_{i_1\cdots i_l}$ transform like tensors of rank l under rigid body rotations of the material sample. To see this combine expressions (15), (31) and (36):

$$\begin{aligned} P_k \hat{f}_l(g) = \hat{f}_l(k^{-1}g) &= \sum_{\alpha=1}^{Q(l)} V^{(l,\alpha)}_{i_1\cdots i_l} f^{(l,\alpha)}_{i_1\cdots i_l}(k^{-1}g) \\ &= \sum_{\alpha=l}^{Q(l)} V^{(l,\alpha)}_{i_1\cdots i_l} k_{j_1 i_1} g_{j_1 m_1} \cdots k_{j_l i_l} g_{j_l m_l} t^{(l,\alpha)}_{m_1\cdots m_l} \\ &= \sum_{\alpha=1}^{Q(l)} k_{j_1 i_1} \cdots k_{j_l i_l} V^{(l,\alpha)}_{i_1\cdots i_l} g_{j_1 m_1} \cdots g_{j_l m_l} t^{(l,\alpha)}_{m_1\cdots m_l} \\ &= \sum_{\alpha=1}^{Q(l)} k_{j_1 i_1} \cdots k_{j_l i_l} V^{(l,\alpha)}_{i_1\cdots i_l} f^{(l,\alpha)}_{j_1\cdots j_l}(g) \\ &= \sum_{\alpha=1}^{Q(l)} \tilde{V}^{(l,\alpha)}_{j_1\cdots j_l} f^{(l,\alpha)}_{j_1\cdots j_l}(g). \end{aligned} \tag{39}$$

The transformed coefficients $\tilde{V}^{(l,\alpha)}_{j_1\cdots j_l}$ are related to the original coefficients $V^{(l,\alpha)}_{i_1\cdots i_l}$ by the expression

$$\tilde{V}^{(l,\alpha)}_{j_1\cdots j_l} = k_{j_1 i_1} \cdots k_{j_l i_l} V^{(l,\alpha)}_{i_1\cdots i_l}. \tag{40}$$

The new tensorial basis functions of rank l can be expressed as linear combinations of the gsh of order l. They also satisfy the orthogonality relation

$$\int_{SO(3)} f^{(l,\alpha)}_{i_1\cdots i_l}(g) f^{(m,\alpha)}_{i_1\cdots i_m}(g)\, dg = 0 \quad \text{if} \quad l \neq m \quad (i_k = 1, 2, 3). \tag{41}$$

When $l = m$, however, the tensorial basis functions are no longer orthogonal.

A NUMERICAL APPROACH TO THE GENERATION OF THE CUBIC TENSORIAL FUNCTIONS

The numerical implementation of Eq. (34) is not practical beyond rank 12 with present day computers; the number of calculations required increases roughly one order-of-magnitude for each unit increase in the rank of the tensor. A more direct approach is much more efficient in the case of tensors possessing cubic (or orthorhombic) symmetry.

The desired tensors must satisfy the symmetry condition expressed in relation (29). In component form this is

$$t^{(l,\alpha)}_{m_1\cdots m_l} = h_{m_1 n_1}\cdots h_{m_l n_l} t^{(l,\alpha)}_{n_1\cdots n_l}, \qquad h \in H_0. \tag{42}$$

In a suitable reference frame, the matrices h_{ij} have a very simple structure: there is only one non-zero entry for each row and column, and its value is ± 1. Thus, only one term in the summation of each index is non-zero. The overall effect of the symmetry operation is an equivalence of one component of the tensor with another, coupled with a possible sign change. Every transformed component can be expressed in terms of the $2l+1$ original components, and for each symmetry element a system of linear equations is formed. The determinant of any such system is generally zero, but with the assistance of algorithms capable of manipulating algebraic expressions (we have used *Mathematica*®), we obtain a solution to the system in terms of one or more free parameters. Introducing this solution into Eq. (42), and then repeating the procedure until the list of symmetry elements has been exhausted, yields either the zero tensor or a tensor with (typically) a small number of free parameters. The number of free parameters is $Q(l)$ for tensors of rank l, which is found to be identical with $M(l)$ in the classical representation as we have previously indicated. Table 1 describes the results of this process for completely symmetric and traceless tensors possessing the 0 point-symmetry group.

Once tensors with appropriate symmetry have been generated, and a choice has been made for each of the arbitrary parameters, then an analytical expression for each basis function follows from relation (31). When more than one tensor occurs for any given rank (i.e., more than one free parameter appears in the numerical procedure), the free parameters can be chosen such that the resulting tensors are orthogonal among themselves. However, orthogonality is not a requirement of the formulation, and a casual choice of free parameters is also permitted.

We note that the same procedure is used to symmetrize the V coefficients according to the relevant sample symmetry, except that we use rotation matrices k_{ij} associated with elements of the material sample symmetry group K. This affects a reduction in the number of linearly independent coefficients associated with rank l to a number $< (2l+1)$. In particular, for the purpose of describing cubic-cubic functions which have been widely used to describe the grain-boundary misorientation distribution function in cubic polycrystals (cf. Bunge 1982, p. 282; Pospiech, Sztwiertnia and Haessner, 1986; Adams, Zhao and O'Hara, 1990), the symmetrized tensor of coefficients has the symmetry given in Table 1.

The general expression for the tensorial functions given in Eq. (31) requires a summation over 3^l components of the tensor, multiplied by products of components of the rotation tensor raised to a power. The presence of symmetry reduces the number of terms (very substantially in the case of cubic symmetry)

Table 1 Independent components of the cubic-symmetric irreducible tensors.

Rank l	*Tensor components* ($t_{1\ldots111}$, $t_{1\ldots112}$, $t_{1\ldots113}$, $t_{1\ldots122}$, $t_{1\ldots123}$, $t_{1\ldots222}$, . . . , $t_{2\ldots222}$, $t_{2\ldots223}$)
4	$2a$ 00 $-a$ 000 $2a$ 0
6	$2a$ 00 $-a$ 000 $-a$ 0 00 $2a$ 0
8	$2a$ 00 $-a$ 000 a 0 00 $-a$ 0 00 $2a$ 0
9	0 00 0 000 0 $-a$ 00 0 a 00 0 0 00
10	$10a$ 00 $-5a$ 000 a 0 00 a 0 00 $-5a$ 0 00 $10a$ 0
12	$4a$ 00 $-2a$ 000 $2b$ 0 00 $a-3b$ 0 00 $2b$ 0 00 $-2a$ 0 00 $4a$ 0
13	0 00 0 000 0 $-a$ 00 0 a 00 0 $-a$ 00 0 a 00 0 0 00
14	$14a$ 00 $-7a$ 000 $3a$ 0 00 $-a$ 0 00 $-a$ 0 00 $3a$ 0 00 $-7a$ 0 00 $14a$ 0
15	0 00 0 000 0 $-a$ 00 0 a 00 0 0 00 0 $-a$ 00 0 a 00 0 0 00
16	$20a$ 00 $-10a$ 000 $10b$ 0 00 $5a-15b$ 0 00 $16b-6a$ 0 00 $5a-15b$ 0 00 $10b$ 0 00 $-10a$ 0 00 $20a$ 0
17	0 00 0 000 0 $-a$ 00 0 a 00 0 $-a$ 00 0 a 00 0 $-a$ 00 0 a 00 0 0 00
18	$28a$ 00 $-14a$ 000 $14b$ 0 00 $7a-21b$ 0 00 $10b-4a$ 0 00 $10b-4a$ 0 00 $7a-21b$ 0 00 $14b$ 0 00 $-14a$ 0 00 $28a$ 0
19	0 00 0 000 0 $-2a$ 00 0 $2a$ 00 0 $-a$ 00 0 0 00 0 a 00 0 $-2a$ 00 0 $2a$ 00 0 0 00
20	$28a$ 00 $-14a$ 000 $14b$ 0 00 $7a-21b$ 0 00 $24b-10a$ 0 00 $11a-25b$ 0 00 $24b-10a$ 0 00 $7a-21b$ 0 00 $14b$ 0 00 $-14a$ 0 00 $28a$ 0
21	0 00 0 000 0 $-a$ 00 0 a 00 0 $-b$ 00 0 $2b-a$ 00 0 $a-2b$ 00 0 b 00 0 $-a$ 00 0 a 00 0 000

because many of the components of the tensor are zero. Consider for instance the generation of the fourth-order tensorial function $f^{(4,1)}_{1111}(g)$. From (31), and the fact that the basic tensors are completely symmetric, we observe that only 15 out of the $3^4 = 81$ components of the generating tensor have distinct values; the other components can be generated by permutation of the indices of the set of independent components. When symmetry is considered, only 6 are different from zero: $t^{(4,1)}_{1111} = 2$, $t^{(4,1)}_{1122} = -1$, $t^{(4,1)}_{1133} = -1$, $t^{(4,1)}_{2222} = 2$, $t^{(4,1)}_{2233} = -1$, $t^{(4,1)}_{3333} = 2$. The number of distinct permutations of their indices are 1, 6, 6, 1, 6 and 1, respectively (in general the number of distinct permutations of a set of three distinct objects a, b and c, each occurring in number n_a, n_b and n_c, is $(n_a + n_b + n_c)!/(n_a!n_b!n_c!)$), hence the complete expression for $f^{(4,1)}_{1111}(g)$ becomes

$$f^{(4,1)}_{1111}(g) = 2g^4_{11} - 6g^2_{11}g^2_{12} - 6g^2_{11}g^2_{13} + 2g^4_{12} - 6g^2_{12}g^2_{13} + 2g^4_{13}. \tag{43}$$

When the indices of the basis functions are not all equal, not all possible permutations of the indices of the tensor lead to the same term in the summation. Consider for instance the case of $f^{(4,1)}_{1112}(g)$; the six possible permutations of the tensor coefficient $t^{(4,1)}_{iijj}$ will generate terms of the form $g^2_{1i}g_{1j}g_{2j}$ and $g_{1i}g^2_{1j}g_{2i}$, both occurring three times. The expression for this function is

$$\begin{aligned} f^{(4,1)}_{1112}(g) = {} & 2g^3_{11}g_{21} - 3g^2_{11}g_{12}g_{22} - 3g_{11}g^2_{12}g_{21} - 3g^2_{11}g_{13}g_{23} \\ & - 3g_{11}g^2_{13}g_{21} + 2g^3_{12}g_{22} - 3g^2_{12}g_{13}g_{23} - 3g_{12}g^2_{13}g_{22} + 2g^3_{13}g_{23}. \end{aligned} \tag{44}$$

For higher orders, the reduction in the number of the terms appearing in the function is more substantial; for instance, the largest expression for rank 21 contains about 3000 terms out of the 3^{21} possible.

The feasibility of generating and evaluating the tensorial functions is thus seen to depend crucially upon our ability to directly generate the distinct terms in the summation, without having to browse over all of the 3^l terms. Algorithms for generating the tensorial functions have been incorporated in a computer program, written in ANSI C. This program is listed in the Appendix; its output is an ANSI C source code containing expressions of the form shown in Eqs (43) and (44), for all of the linearly independent tensorial functions exhibiting cubic symmetry, up to rank $l = 21$. This file must be compiled and linked with any other program making use of the tensorial functions. The storage requirement for the compiled program is approximately 27 MBytes. Benchmark comparisons between programs for generating the gsh and the tensorial functions indicate that up to order 12 the tensorial functions are more rapidly calculated than the cubic symmetric classical functions. Beyond order 12 a reversal of this result is observed.

SOME APPLICATIONS OF THE TENSORIAL REPRESENTATION

Estimation of Coefficients of the CODF from Single Orientation Measurements

The conventional procedure for obtaining the series expansion of the codf from a set of single orientation measurements was described by Bunge (1982). Each single orientation data point is considered to be delta function in the Euler space; these functions are expressed as a Fourier series. Coefficients for the complete codf are then estimated by averaging the coefficients over the entire set of measurements. Let the number of measurements be N; in terms of the gsh it is

found that

$$C_l^{\mu\nu} \cong \frac{2l+1}{N} \sum_{i=1}^{N} \dot{\bar{T}}_l^{\mu\nu *}(g_i), \tag{45}$$

where $\dot{\bar{T}}_l^{\mu\nu *}(g_i)$ is the value of the fully-symmetrized (for sample and lattice symmetry) complex conjugate of the gsh evaluated in the ith orientation.

In the case of the tensorial representation we begin with the series representation for the codf expressed in relation (38). Lattice symmetry is carried in the basis functions $f_{i_1\cdots i_l}^{(l,\alpha)}(g)$ and material sample symmetry is carried in the tensorial coefficients $V_{i_1\cdots i_l}^{(l,\alpha)}$.

Note the strong redundancy in expression (38) which arises on account of the completely symmetric and traceless character of the basis functions and coefficients expressed in (35) and (37). It is necessary to carry this redundancy when preservation of the tensorial character of the series is essential. An abbreviated, but equivalent series can be established in terms of a selected set of linearly independent basis functions. In other words, select from among the full set of basis functions $\{f_{i_1\cdots i_l}^{(l,\alpha)}(g) : i_k = 1, 2, 3\}$ an arbitrary, but linearly independent set. The indices of this independent set for rank l we shall denote as $\{i_1 i_2 \cdots i_l\}$. It is evident that Eq. (38) can be rewritten as

$$f(g) = \sum_{l=0,4,\ldots}^{\infty} \sum_{\alpha=1}^{Q(l)} \sum_{\{i_1 i_2 \cdots i_l\}} B_{i_1\cdots i_l}^{(l,\alpha)} f_{i_1\cdots i_l}^{(l,\alpha)}(g). \tag{46}$$

The coefficients $B_{i_1\cdots i_l}^{(l,\alpha)}$ are linear combinations of $V_{i_1\cdots i_l}^{(l,\alpha)}$; once the independent basis functions have been chosen they are uniquely determined. The B coefficients no longer possess the tensorial character that V coefficients have.

If we now multiply both sides of Eq. (46) by a particular basis function, and perform invariant integration over $SO(3)$, we obtain

$$\langle f_{j_1 j_2 \cdots j_l}^{(l,\alpha)}(g), f(g) \rangle = \sum_{\mu=1}^{Q(l)} \sum_{\{i_1 i_2 \cdots i_l\}} \{ f_{j_1 j_2 \cdots j_l}^{(l,\alpha)}(g), f_{i_1 i_2 \cdots i_l}^{(l,\mu)}(g) \rangle B_{i_1 i_2 \cdots i_l}^{(l,\mu)} \tag{47}$$

where the notation of relation (20) has been used. This is just a system of linear equations which can be solved for the coefficients $B_{i_1\cdots i_l}^{(l,\alpha)}$:

$$B_{i_1\cdots i_l}^{(l,\mu)} = \sum_{\alpha=1}^{Q(l)} \sum_{\{j_1\cdots j_l\}} \langle f_{j_1\cdots j_l}^{(l,\alpha)}(g), f_{i_1\cdots i_l}^{(l,\mu)}(g) \rangle^{-1} \langle f_{j_1\cdots j_l}^{(l,\alpha)}(g), \hat{f}(g) \rangle. \tag{48}$$

It follows that the expression equivalent to that of (45) in the new basis functions is

$$B_{i_1\cdots i_l}^{(l,\mu)} \cong \frac{1}{N} \sum_{\beta=1}^{N} \sum_{\alpha=1}^{Q(l)} \sum_{\{j_1\cdots j_l\}} \langle f_{j_1\cdots j_l}^{(l,\alpha)}(g), f_{i_1\cdots i_l}^{(l,\mu)}(g) \rangle^{-1} f_{j_1\cdots j_l}^{(l,\alpha)}(g_\beta). \tag{49}$$

Calculation of the tensorial V coefficients requires inversion of the original linear relationship between B and V.

We remark that expression (49) is more complicated than the quivalent relation for the classical Fourier coefficients given in (45). The reason is that orthogonality between different functions belonging to the same order does not exist. We would emphasize, however, that the matrix of the inner products, $\langle f_{j_1\cdots j_l}^{(l,\alpha)}(g), f_{i_1\cdots i_l}^{(l,\mu)}(g) \rangle$, is always small and easily invertible because of the linear independence of the functions and a highly regular block structure in the matrix.

Averaging of Single Crystal Properties

Simple averages of single crystal properties often occur in models of polycrystalline properties. These averages have the form

$$\langle \mathbf{E} \rangle = \int_{SO(3)} \mathbf{E}(g) f(g)\, dg \tag{50}$$

where $\mathbf{E}(g)$ denotes the local crystal property tensor referred to a reference crystal, $\mathbf{E}^0$, through the tensor transformation $\mathbf{E}(g) = P_g \mathbf{E}^0$. $f(g)$ is the codf, and $\langle \mathbf{E} \rangle$ denotes the average property tensor associated with the polycrystal. This average is often used as a first-order estimate of the effective property tensor for the polycrystal.

In the evaluation of property averages using the conventional approach (Bunge, 1982, ch. 13), the local property tensor is expressed as a linear combination of symmetrized gsh. This leads to an expression in the gsh, of finite order (equivalent to the rank of the local property tensor). The integrand of relation (50) then includes terms in products of the gsh and coefficients of the local property tensor and the codf. Due to the orthogonality properties of the gsh, the final expression for (50) appears in the form

$$\langle \mathbf{E} \rangle = \xi \cdot \mathbf{E}^0 \tag{51}$$

where ξ is a tensor of rank $2R$, and $\mathbf{E}^0$ is the property tensor of rank R associated with the reference crystal. It is evident that the tensor ξ depends upon the coefficients of the codf which carry both sample an crystal symmetries, even though $\langle \mathbf{E} \rangle$ evidently carries only the material sample symmetry.

We now consider the equivalent construction in terms of the tensorial representation. In this case relation (50) becomes

$$\begin{aligned} \langle \mathbf{E} \rangle &= \int_{SO(3)} \mathbf{E}(g) \sum_{l=0}^{\infty} \sum_{\alpha=1}^{Q(l)} V^{(l,\alpha)}_{i_1 \cdots i_l} f^{(l,\alpha)}_{i_1 \cdots i_l}(g)\, dg \\ &= \int_{SO(3)} P_g \mathbf{E}^0 \sum_{l=0}^{\infty} \sum_{\alpha=1}^{Q(l)} \mathbf{V}^{(l,\alpha)} : P_g \mathbf{t}^{(l,\alpha)}\, dg \\ &= \sum_{l=0}^{\infty} \sum_{\alpha=1}^{Q(l)} \mathbf{V}^{(l,\alpha)} : \int_{SO(3)} P_g \mathbf{t}^{(l,\alpha)} \otimes \mathbf{E}^0\, dg, \end{aligned} \tag{52}$$

where we have used the colon (:) symbol to denote contraction over the set of repeated indices. Notice that since $\mathbf{E}^0$ has rank R, the sum over l can be limited to span $[0, R]$. Also, the invariant integration of $P_g \mathbf{t}^{(l,\alpha)} \otimes \mathbf{E}^0$ clearly yields an isotropic tensor of rank $(l + R)$. This is obvious from the requirement that

$$\int_{SO(3)} P_{gh} \mathbf{t}^{(l,\alpha)} \otimes \mathbf{E}^0\, dg = \int_{SO(3)} P_g \mathbf{t}^{(l,\alpha)} \otimes \mathbf{E}^0\, dg, \qquad h \in SO(3). \tag{53}$$

Let $\Xi^{(l+R,\alpha)}$ represent the isotropic tensors defined by the invariant integration of relation (53); these tensors depend only upon the basic tensors $\mathbf{t}^{(l,\alpha)}$ and the crystal property tensor defined in the reference orientation, $\mathbf{E}^0$. It follows from (52) that

$$\langle \mathbf{E} \rangle = \sum_{l=0}^{R} \sum_{\alpha=1}^{Q(l)} \mathbf{V}^{(l,\alpha)} : \Xi^{(l+R,\alpha)}. \tag{54}$$

The reader should notice the remarkable simplicity which arises when the tensorial representation is used in forming simple averages. The average is expressed as the contraction of isotropic tensors formed from the local crystal tensor with the tensorial coefficients of the codf. It is perfectly clear, when presented this way, that the average tensor has the symmetries of the sample, since these are carried in the tensorial coefficients. This remarkable simplicity is a consequence of the tensorial representation; it illustrates why the tensorial representation may have some advantages in certain situations.

CONCLUSIONS

A tensorial representation of the codf has been presented. Tensorial basis functions can be expressed as linear combinations of the conventional generalized spherical harmonics of the same order. They exhibit orthogonality between ranks, but not between functions of the same rank. This property increases, somewhat, the complexity in working with them in comparison with the classical representation. Computation times are nominally equivalent in comparison with the conventional methods (faster at orders < 13, but slower at higher orders in the case of cubic lattice symmetry). Because of their coordinate-free interpretation, there will appear certain advantages in using the tensorial representation. An example of such is the remarkable simplicity that appears in simple averages of local properties.

Acknowledgement

The authors wish to acknowledge sponsorship of this work by the National Science Foundation under a Materials Research Groups Award.

APPENDIX

```
/*****************************************************************************/
/*----------------------Program Tens-funct-gen.c, written by M. Guidi, Yale University, January 1991----------------------*/
/*----------------generation of the expressions for the tensorial basis functions defined on SO(3) to real values----------------*/
/*****************************************************************************/
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define maxrank  8                                   /* highest rank processed:21 max for this program */
#define maxind(n) (((n)*(n) + 3*(n) + 2)/2)          /* # of distinct coeff in the tensor */
#define sign(n) (((n) % 2) ? -1 : 1)

/*-----------------------------------------FUNCTION S DECLARATIONS-----------------------------------------*/
void HeaderPreparation(void), FillBox(int, int), OutputTerm(void);
void GenIndep(int), GenDist(int), Express_by_Indep(char *);
int  find_nzero_coeff(int rank);
double    fac(int n);

/*-----------------------------------------GLOBAL VARIABLES DECLARATIONS-----------------------------------------*/
char OutputString[7*maxrank+1], **nzero;
char funct[2*maxrank+1][maxrank], index[maxind(maxrank)][maxrank];
int  ForBox[4],ForInd[4], Box[maxrank], coeff[2*maxrank+1];
int  m, rank, nind, ncf, dj, Run, BoxSize;OnTheLast, CharsOnLine, NLines;
FILE *out;
float    *tValue;

int  rk[20] = {4,6,8,9,10,12,12,13,14,15,16,16,17,18,18,19,20,20,21,21};

/*---------------values of the independent components of the cubic-symmetric tensors up to rank 21----------------*/
int tValues[22][45] ={
{2,0,0,-1,0,0,0,2,0},
{2,0,0,-1,0,0,0,-1,0,0,0,0,2,0},
{2,0,0,-1,0,0,0,1,0,0,0,-1,0,0,0,2,0},
{0,0,0,0,0,0,0,0,-1,0,0,0,1,0,0,0,0,0,0},
{10,0,0,-5,0,0,0,1,0,0,0,1,0,0,0,0,-5,0,0,0,0,10,0},
{4,0,0,-2,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,-2,0,0,0,4,0},
{0,0,0,0,0,0,0,2,0,0,0,-3,0,0,0,2,0,0,0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0,-1,0,0,0,1,0,0,0,-1,0,0,0,1,0,0,0,0,0,0},
{14,0,0,-7,0,0,0,3,0,0,0,-1,0,0,0,-1,0,0,0,3,0,0,0,-7,0,0,0,14,0},
{0,0,0,0,0,0,0,0,-1,0,0,0,1,0,0,0,0,0,0,0,-1,0,0,0,1,0,0,0,0,0,0},
{20,0,0,-10,0,0,0,0,0,0,0,5,0,0,0,-6,0,0,0,5,0,0,0,0,0,0,0,-10,0,0,0,20,0},
{0,0,0,0,0,0,0,10,0,0,0,-15,0,0,0,16,0,0,0,-15,0,0,0,10,0,0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0,-1,0,0,0,1,0,0,0,-1,0,0,0,1,0,0,0,-1,0,0,0,1,0,0,0,0,0,0},
{28,0,0,-14,0,0,0,0,0,0,0,7,0,0,0,-4,0,0,0,-4,0,0,0,7,0,0,0,0,0,0,0,-14,0,0,0,28,0},
{0,0,0,0,0,0,0,14,0,0,0,-21,0,0,0,10,0,0,0,0,10,0,0,0,-21,0,0,0,14,0,0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0,-2,0,0,0,2,0,0,0,-1,0,0,0,0,0,0,0,1,0,0,0,-2,0,0,0,2,0,0,0,0,0,0},
{28,0,0,-14,0,0,0,0,0,0,0,7,0,0,0,-10,0,0,0,0,11,0,0,0,-10,0,0,0,0,7,0,0,0,0,0,0,0,-14,
0,0,0,28,0},
{0,0,0,0,0,0,0,14,0,0,0,-21,0,0,0,24,0,0,0,-25,0,0,0,24,0,0,0,-21,0,0,0,14,0,0,0, 0,0,0,0,0,(
{0,0,0,0,0,0,0,0,-1,0,0,0,1,0,0,0,0,0,0,0,-1,0,0,0,1,0,0,0,0,0,0,0,-1,0,0,0,1,0,0, 0,0,0,0},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-1,0,0,0,2,0,0,0,-2,0,0,0,1,0,0,0,0,0,0,0,0,0,0, 0,0,0,0}};

/*----------values of the normalization factors for the functions (<f1111..> = 1 for each order) up to rank 21----------*/
double norm[20] ={2.8716323331596296, 5.1615697874749094, 2.2223734635886014,
                  4.2591220627765916, 17.449290667941380, 13.102697906460071,
                  13.890394782794614, 4.9922915579431564, 19.433885971651812,
                  13.235280035222852, 56.586542725083959, 62.336584907819223,
                  5.3849731080004419, 196.27812313619282, 372.53981179251281,
                  20.270473375123991, 73.432397168092621, 82.884027703764218,
                  41.336289950959682, 42.171950896747951};
```

```
/*****************************************************************************/
/*------------------------------------ beginning of the main module ------------------------------------*/
/*****************************************************************************/
void main(void) {
     int  j, ncoeff, fcount = 0;

     out = fopen("fdef.c","w");
     HeaderPreparation();
     rank = maxrank;
/*---------- generation the linear independent and distinct coeff of the biggest tensor ----------*/
     m = 0; nind = 2 * rank + 1; funct[0][0] = 1; GenIndep(0);
     m = 0; nind = maxind(rank); index[0][0] = 1; GenDist(0);
     for (Run = 0; (rank = rk[Run]) <= maxrank; Run++) {
          nind = 2 * rank + 1; dj = maxrank - rank;
/* ---------- generation of the non-zero permutationally distinct coefficients of the tensor ----------*/
          ncoeff = find_nzero_coeff(rank);
          m = 0; for (m = 0; m < nind; m++) {
               fprintf(out,"\ndouble f%i(double *a)\n{\n",fcount);
               fprintf(out,"\n\tdouble\tg;\n\t\g = 0 ");
               fcount++;
               ForBox[1] = ForBox[2] = ForBox[3] = 0;
               for (j = 0; j < rank; j++) ++ForBox[funct[m][j+dj]];
               for (ncf = 0; ncf < ncoeff; ncf++) {
                    ForInd[1] = ForInd[2] = ForInd[3] = 0;
                    for (j = 0; j < rank; j++) ++ForInd[nzero[ncf][j]];
                    OnTheLast = 0; BoxSize = (ForBox[1] ? ForBox[1] : ForBox[2]);
                    if (ForBox[3]) for (OnTheLast = 1; OnTheLast <= 3; OnTheLast++) {
                         if (ForInd[OnTheLast]) {
                              --ForInd[OnTheLast]; FillBox(1,0); ++ForInd[OnTheLast]; }
                    } else FillBox(1,0);
               }
               fprintf(out,";\n\treturn g /= %lf;\n}\n\n",norm[Run]);
               CharsOnLine = 0; NLines = 0;
          }
          for (j = 0; j < ncoeff; j++) free(nzero[j]); free(nzero); free(tValue);
     }
}
/*****************************************************************************/
/*---------- module finding ther non-zero entries in the tensor:returns the arrays nzero & tValue ----------*/
/*---------- containing the indices of the non-zero components, and their values ----------*/
/*****************************************************************************/
int find_nzero_coeff(int rank) {
     int  i, j, nzero_cnt = 0;
     float     tVal;
          /*---------- determination of the number of non-zero entries ---------- */
     for (i = 0; i < maxind(rank); i++) {
          Express_by_Indep(index[i]);
          tVal = 0; for (j = 0; j < nind; j++) {
               tVal += coeff[j] * tValues[Run][j]; coeff[j] = 0;
          } if (tVal) nzero_cnt++;
     }
     nzero = calloc(nzero_cnt, sizeof(char *));
     tValue = calloc(nzero_cnt, sizeof(float));
     for (i = 0; i < nzero_cnt; i++) nzero[i] = calloc(rank, sizeof(char));
          /*---------- determination of the number of non-zero entries ---------- */
     nzero_cnt = 0; for (i = 0; i < maxind(rank); i++) {
          Express_by_Indep(index[i]);
          tVal = 0; for (j = 0; j < nind; j++) {
               tVal += coeff[j] * tValues[Run][j]; coeff[j] = 0;
          } if (tVal) {
               for (j = 0; j < rank; j++) nzero[nzero_cnt][j] = index[i][dj + j];
               tValue[nzero_cnt] = tVal; nzero_cnt++;
          }
     } return nzero_cnt;
}
```

```
/*****************************************************************************/
/* generation of all the possible dispositions of the indices of the non-zeros coefficients with the indices of the function */
/*****************************************************************************/
void FillBox(int Index, int Depth) {
     int i, j, n[4];

     for (i = Index; i <= 3; i++) {
          if (ForInd[i] > 0) {
               Box[Depth] = i; --ForInd[i];
               if (Depth == BoxSize - 1) {
                    n[1] = ForInd[1]; n[2] = ForInd[2]; n[3] = ForInd[3];
                    for (j = BoxSize; j < rank - (OnTheLast != 0); j++){
                         if (n[1]-- > 0) Box[j] = 1;
                         else if (n[2]-- > 0) Box[j] = 2;
                         else if (n[3]-- > 0) Box[j] = 3;
                    }
                    if (OnTheLast) Box[j] = OnTheLast;
                    OutputTerm(); ++ForInd[i];
               } else {
                    FillBox(i,Depth + 1); ++ForInd[i];
               }
}    }    }
/*****************************************************************************/
/*---------- generation of a term in the expression of the function ----------*/
/*****************************************************************************/
void OutputTerm() {
     int  j, n, *gExp, *InBox, *OutBox;
     double    Mult;
     char *s;

     gExp = calloc(9, sizeof(int));
     InBox  = calloc(3, sizeof(int)); InBox--;
     OutBox = calloc(3, sizeof(int)); OutBox--;

     for (j = 0; j < BoxSize; j++) {
          ++gExp[3*funct[m][j+dj] + Box[j] - 4]; ++InBox[Box[j]];
     }
     for (j = BoxSize; j < rank - (OnTheLast != 0); j++) {
          ++gExp[3*funct[m][j+dj] + Box[j] - 4]; ++OutBox[Box[j]];
     }
     if (OnTheLast) ++gExp[(3*funct[m][j+dj] + Box[j] - 4)];
     Mult  = fac(BoxSize)/(fac(InBox[1])*fac(InBox[2])*fac(InBox[3]));
     Mult *= fac(rank-BoxSize-(!OnTheLast))/(fac(OutBox[1])*fac(OutBox[2])*fac(OutBox[3]));
     s = OutputString;
     for (j = 0; j < 9; j++) if (gExp[j]) {
               n = sprintf(s,"a[%i]*", (gExp[j]-1)*9 + j);
               s += n*sizeof(char); CharsOnLine += n;
     } s[-1] = '\0';
     fprintf(out,"%+g*%s ", Mult*tValue[ncf], OutputString);
     if (CharsOnLine   > 50) {
          if(++NLines == 100) { fprintf(out,";\n\tg += 0 ");NLines = 0;}
          else fprintf(out,"\n\t\t");
          CharsOnLine = 0;
}    }
/*****************************************************************************/
/*---------- calculation of n! ----------*/
/*****************************************************************************/
double fac(int n) {
     int i;
     double x = 1.0;

     if (n) { for (i = 1; i <= n; i++) x *= i; return x;
     } else  return 1.0;
}
```

```
/*****************************************************************************/
/*---- generation of the low-indices 2n+1 independent coefficients of a completely symmetric and traceless tensor------*/
/*****************************************************************************/
void GenIndep(int n) {
     int        i,j;

     if (n >= rank-2) {
          for (i = funct[m][n]; i <= 3 ; i++) {
               funct[m][n+1] = i;
     /*------------------- if it is not the next-to-last coeff,copy the last to the following one ----------------------------*/
               if (m < nind - 3)  {
                    for (j = 0; j < rank-1; j++) funct[m+1][j] = funct[m][j];
               } else {
                    for (j = 0;  j < rank;  j++) funct[m+1][j] = funct[m+2][j]
funct[m][j];
                    funct[m+1][0] = funct[m+2][0] = funct[m+1][rank-1] = 2;
               } m++;
          } return;
     }
     for (i = funct[m][n]; i <= 2; i++) {
          funct[m][n+1] = i;
          GenIndep(n+1);
     }
return;
}

/*****************************************************************************/
/*--------- generation of the permutationally disinct coefficients of a completely symmetric and traceless tensor---------*/
/*****************************************************************************/
void GenDist(int n) {
     int        i,j;

     if (n >= rank-2) {
          for (i = index[m][n]; i <= 3 ; i++) {
               index[m][n+1] = i;
     /*------------------- if it is not the next-to-last coeff,copy the last to the following one ----------------------------*/
               for (j = 0; j < rank-1; j++) index[m+1][j] = index[m][j];
               if (m >= nind - 2) index[m+1][0] = index[m+1][rank-1] = 3;
               m++;
          } return;
     }
     for (i = index[m][n]; i <= 3; i++) {
          index[m][n+1] = i;
          GenDist(n+1);
     }
     if (n == 0 && index[m][0] == 1) { index[m][0] = 2; GenDist(0);}
     return;
}

/*****************************************************************************/
/*-conversion of a generic element of a tensor into a linear combination of the 2l+1 linear independent components--*/
/*****************************************************************************/
void Express_by_Indep(char *el)
{
     int        i, n[4], c0, n3;

     n[1] = n[2] = n[3] = 0;
     for (i = maxrank - rank; i < maxrank; i++) ++n[el[i]];
     c0 = (((n[2] + (n[3] % 2)) == 0) ? 0 : (2 * n[2] + 3 * (n[3] % 2) - 1));
     n3 = n[3]/2; coeff[c0] = sign(n3);
     for (i = 1; i <= n3; i++)
          coeff[c0 + 4*i - (c0 == 0)] = sign(n3)*fac(n3)/(fac(i)*fac(n3-i));
}
```

```
/*****************************************************************************/
/*--------Introduction in the output file of the code for the routine fun_prepare that arranges the functions in a two ------*/
/*----------------------------------dimensional array f[][] ; generation of the include file fdecl.h----------------------------------*/
/*****************************************************************************/
void HeaderPreparation()
{
     int       i, j, nfun, chars_on_line,fcount = 0;
     FILE *declaration;

     fprintf(out,"#include <stdlib.h>\n#include <stdio.h>\n#include <math.h>");
     fprintf(out,"\n#include \"fdecl.h\"\n\nextern double (***f)(double *);\n\n");
     fprintf(out,"void  fun_prepare(void);\n\nvoid fun_prepare(void)\n{\tint \ti,
nfun;");
     fprintf(out,"\n\n\tf = calloc(maxrank - 3, sizeof(double (**)()));\n\tf -= 4;");
     fprintf(out,"\n\tfor (i = 4; i <= maxrank; i++) if ((i!=5)&&(i!=7)&&(i!=11))");
     fprintf(out," {\n\t\tnfun = (((i==12)||(i==16)||(i==18)||(i==20)||(i==21)||");
     fprintf(out,"(i==22)) ? 4*i+2 : 2*i+1);\n\t\tf[i] = calloc(nfun,");
     fprintf(out,"sizeof(double (*)()));\n\t}\n\t");
     for (i = 4; i <= maxrank; i++) if ((i!=5)&&(i!=7)&&(i!=11)) {
          nfun = (((i==12)||(i==16)||(i==18)||(i==20)||(i==21)||(i==22)) ? 4*i+2 :
2*i+1);
          chars_on_line = 0;
          fprintf(out,"\n\t");
          for (j = 0; j < nfun; j++){
               chars_on_line += fprintf(out, "f[%i][%i]=f%i; ", i, j, fcount);
               fcount ++; if (chars_on_line/70) {
                    fprintf(out,"\n\t"); chars_on_line = 0; }
          } if (chars_on_line) fprintf(out, "\n");
     }
     fprintf(out,"}\n"); chars_on_line = 0;
     declaration = fopen("fdecl.h","w");
     fprintf(declaration,"#define maxrank %i\n\n", maxrank);
     fprintf(declaration, "double\t");
     for (i = 0; i < fcount-1; i++) {
          chars_on_line += fprintf(declaration, "f%i(double *), ", i);
          if (chars_on_line/70) { fprintf(declaration,"\n\t\t"); chars_on_line = 0; }
     }
     fprintf(declaration,"f%i(double *);\n", i); fclose(declaration);
}
/*----------------------------------------------END OF THE PROGRAM Tens-funct-gen.c----------------------------------------------*/

/*#############################################################################*/
/*----------SAMPLE PROGRAM Sample.c EXEMPLIFYING THE USAGE OF THE TENSORIAL FUNCTIONS----------*/
/*#############################################################################*/
#include <math.h>
#include "fdecl.h"
/*-n_ind is a macro returning the number of independent functions for a given order (<=21)-*/
#define   n_ind(x)    ((x-12)*(x-16)*(x-18)*(x-20)*(x-21)   ?   2*x+1
4*x+2)*(x!=5)*(x!=7)*(x!=11)
double g[maxrank][9], (***f)(double *); /*maxrank is defined in fdecl.h and is the order of the last function defined*/

main() {
     double    fi1, fi, fi2, sf1, sf, sf2, cf1, cf, cf2, function_value;
     int  m, l, rank;

     fun_prepare();     /*---- the routine fun-prepare must be called once at the beginning of the program ----*/

     fi1 = fi = fi2 = .7854
        /*-------------assign some value to the three Euler angles and calculate the rotation matrix g ------------*/
     sf1=sin(fi1); cf1=cos(fi1); sf =sin(fi); cf =cos(fi); sf2=sin(fi2); cf2=cos(fi2);
     g[0][0] = cf1*cf2 - sf1*sf2*cf;  g[0][1] =-cf1*sf2 - sf1*cf2*cf; g[0][2] = sf1*sf;
     g[0][3] = sf1*cf2 + cf1*sf2*cf;  g[0][4] =-sf1*sf2 + cf1*cf2*cf; g[0][5] =-cf1*sf;
     g[0][6] = sf2*sf;                g[0][7] = cf2*sf;                g[0][8] = cf;
/*calculate the powers of the entries of the matrix g up to maxrank ; all the functions up to maxrank can then be evaluated */
     for (l = 0; l < 9; l++) for (m = 1; m < maxrank; m++) g[m][l] = g[m-1][l] * g[0][l];
               /*---------------------------------------- evaluation of the function ----------------------------------*/
     for (rank = 4, rank < maxrank; rank++) for (l = 0; l < n_ind(rank); l++) {
          function_value = (*f[rank][l])(&g[0][0]);
          fprintf ("%lf \t", function_value);
     }
}
```

References

Adams, B. L., Zhao, J. W. and O'Hara, D. (1990). *Acta Metall, Mater.*, **38**, 935–966.

Adams, B. L., Boehler, J. P., Guidi, M. and Onat, E. T. (1991) *J. Mech. Phys. Solids*, in press.

Altmann, S. L. (1986). *Rotations, Quaternions, and Double Groups*. 1st edition, Oxford: Clarendon Press.

Backus, G. (1970). *Rev. Geophys. Spacephys.* **8**, 633–671.

Bröcker, T., tom Dieck, T. (1985). *Representation of Compact Lie Groups*, 1st edition, pp. 133–142. New York: Springer Verlag.

Bunge, H. J. (1965). *Z. Metallkde.*, **56**, 872–874.

Bunge, H. J. (1982). *Texture Analysis in Material Science*, 1st edition, pp. 47–116, 294–401. London: Butterworth.

Courant, R. and Hilbert, D. (1989). *Methods of Mathematical Physics*, 1st English edition, pp. 514–522. New York: John Wiley and Sons.

Geary, J. E. and Onat, E. T. (1974). Representation of Nonlinear Hereditary Behavior. *Oak Ridge National Laboratory Report ORNL-TM-4525*.

Gel'fand, I. M., Minlos, R. A. and Shapiro, Z. Ya. (1963). *Representations of the Rotation and Lorentz Groups and their Applications*, 1st edition, pp. 13–156. Oxford: Pergamon Press.

Hansen, J., Pospiech, J., and Lücke, K. ((1978). *Tables for Texture Analysis of Cubic Materials*. Berlin: Springer-Verlag.

Onat, E. T. (1986). *Engineering Fracture Mechanics*, **25**, 605–614.

Pospiech, J., Sztwiertnia, K. and Haessner, F. (1986). *Textures and Microstructures*, **6**, 201.

Roe, R. J. (1965). Description of Crystallite Orientation in Polycrystalline Materials. III. General Solution to the Pole Figure Inversion. *J. Appl. Phys*, **36**, 2024–2031.

Seibert, D. (1991) *Z. angew Math. Mech.* **71**, 91–97.