Texture of Crystalline Solids
1978, Vol. 3, pp. 1-25

THE DETERMINATION OF ORIENTATION DISTRIBUTION FUNCTION FROM INCOMPLETE POLE FIGURES - AN EXAMPLE OF A COMPUTER PROGRAM

JERZY JURA and JAN POSPIECH
Polish Academy of Sciences, Institute of Metal Research, Cracow, Poland

(Received June 7, 1977)

Abstract: The use of incomplete pole figure data for defining the orientation distribution function (ODF) in a polycrystalline material is of great practical importance, because it enables the use of experimental data from a simplified measurement. The present paper provides the source text of a computer program for calculating the coefficients of ODF series expansion, $C_l^{\mu\nu}$. The data for computations are in the form of incomplete pole figures of rhombic symmetry as determined by the back reflection or transmission technique for crystalline solids of the cubic system. Also described is the numerical method of determining the coefficients $C_l^{\mu\nu}$, and the results so obtained are discussed.

INTRODUCTION

Quantitative texture analysis is of fundamental importance in investigations of the properties of a polycrystalline material and its behaviour during recrystallization or deformation. A method which procides an unambiguous representation of the orientation distribution of crystallites in the polycrystalline aggregate is known as three-dimensional texture analysis. It defines a way of determining, on the basis of experimental data (pole figures or sets of single orientations), the three-dimentionsal orientation distribution function (ODF) of the crystallites, from which the relations between texture and the anisotropic properties of the material can also be derived.[1]

The method of three-dimensional texture analysis now finds wide applications for solving various kinds of problems involving textures. Hence methodical analyses of the

applicability of the three-dimensional technique with various kinds of experimental data becomes necessary and desirable. Among these, how the incomplete pole figure data can be used for orientation distribution analysis is obviously needed.[2,3]

We present here a source text of a computer program to calculate the coefficients $C_\ell^{\mu\nu}$ of the series expansion of an ODF. The data for the computation are in the form of incomplete pole figures of rhombic symmetry determined for materials having cubic crystal structure.

First, we shall describe the numerical method of finding $C_\ell^{\mu\nu}$; and the results obtained by executing the program will be discussed. Next, we shall present the structure of the program, the way it functions and the range of computations it can carry out.

The presented program is a supplement to the standard system of computer routines to compute the coefficients $C_\ell^{\mu\nu}$ using complete pole figures, and to determine the values of ODF's, pole figures and inverse pole figures.[4-6]

Method of Determining Coefficients $C_\ell^{\mu\nu}$ on the Basis of Incomplete Pole Figures

When deriving a three-dimensional orientation distribution on the basis of pole figures we take advantage of the fact that the sought ODF, f(g), and the general axis distribution function, A(h,y) [pole figures $P_h(y)$ correspond to sections A(h=const,y)], presented in the form of series of spherical functions have identical expansion coefficients, $C_\ell^{\mu\nu}$,[1]

$$A(h,y) = \sum_{\ell=0}^{\infty} \sum_{\mu=1}^{M(\ell)} \sum_{\nu=1}^{N(\ell)} C_\ell^{\mu\nu} \frac{4\pi}{2\ell+1} \dot{k}_\ell^{\mu}(h)\ \dot{k}_\ell^{\nu}(y) \qquad (1)$$

$$f(g) = \sum_{\ell=0}^{\infty} \sum_{\mu=1}^{M(\ell)} \sum_{\nu=1}^{N(\ell)} C_\ell^{\mu\nu}\ \ddot{T}_\ell^{\mu\nu}(g). \qquad (2)$$

Symmetrical spherical functions, of the surface version $\dot{k}_\ell^{\mu}(h)$, $\dot{k}_\ell^{\nu}(y)$ and the generalized ones $\ddot{T}_\ell^{\mu\nu}(g)$, form complete orthogonal systems. Hence, if we adopt the condition of the mean-square approximation of a given function A(h,y) by the series (1),

$$R = \oint\!\!\oint \left[A(h,y) - \sum_{\ell\mu\nu}^{\ell_{max}} C_\ell^{\mu\nu} \frac{4\pi}{2\ell+1} \dot{k}_\ell^{\mu}(h)\ \dot{k}_\ell^{\nu}(y) \right]^2 dydh = \min \qquad (3)$$

then the value of $C_\ell^{\mu\nu}$ is given by the relation

$$C_\ell^{\mu\nu} = \frac{2\ell+1}{4\pi} \oint\!\!\oint A(h,y)\ \dot{k}^{*\mu}_\ell(h)\ \dot{k}^{*\nu}_\ell(y)\ dydh. \qquad (4)$$

Conditions of measurements define the set $\{h_i, y_j\}$ for which the values $\hat{P}_{h_i}(y_j)$ are determined. As is known, it is restricted. For instance, for a metal crystallizing in the cubic system only $I_p = 3$ or 4 pole figures ($i = 1, 2, 3, 4$) can be measured with sufficient accuracy. In turn, measurements made by one technique along (back-reflection or transmission) provide incomplete figures (Figure 1). Therefore,

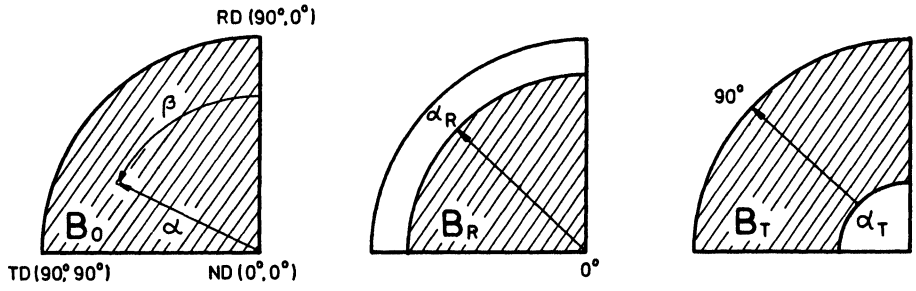

Figure 1. Pole figure regions accessible for back-reflection (B_R) and transmission (B_T) technique.

we must replace equation (3) by the expression

$$R = \sum_{i=1}^{I} \int_{B_i} \left[N_{h_i}\ \hat{P}_{h_i}(y) - \sum_{\ell\mu\nu}^{\ell_{max}} C_\ell^{\mu\nu}\ \frac{4\pi}{2\ell+1}\ \dot{k}^{\mu}_\ell(h)\ \dot{k}^{\nu}_\ell(y) \right]^2 dy = \min \qquad (5)$$

where N_{h_i} denotes the coefficients normalizing the measured values of the pole figure, $\hat{P}_{h_i}(y)$.

Let us assume the case of incomplete figures $B_i < B_0$ (Figure 1). In order to determine the values of $C_\ell^{\mu\nu}$ now, we make use of the fact that it follows from the normalization condition of the functions $f(g)$ and $A(h,y)$ that $C_0^{11} = 1$. Next, differentiating equation (5) with respect to $C_{\ell'}^{\mu'\nu'}$ ($\ell' \neq 0$) and N_{h_i}, yields after some manipulations the following set of equations[2]

$$\sum_{\ell=2}^{\ell_{max}} \sum_{\mu\nu} \Gamma^{\mu\mu'\nu\nu'}_{\ell\ell'} C^{\mu\nu}_{\ell} = - \Gamma^{1\mu'1\nu'}_{0\ell'} \tag{6}$$

where

$$\Gamma^{\mu\mu'\nu\nu'}_{\ell\ell'} = \sum_{i=1}^{I} \frac{q^{\mu\nu}_{\ell}(h_i)\ q^{*\mu'\nu'}_{\ell'}(h_i)}{P(h_i)} - \zeta^{\nu\nu'}_{\ell\ell'}(i)\ \alpha^{\mu\mu'}_{\ell\ell'}(h_i) \tag{7}$$

$$q^{\mu\nu}_{\ell}(h_i) = \frac{4\pi}{2\ell+1}\ \dot{k}^{\mu}_{\ell}(h_i)\ \rho^{\nu}_{\ell}(h_i) \tag{8}$$

$$\rho^{\nu}_{\ell}(h_i) = \int_{B_i} \hat{P}_{h_i}(y)\ \dot{k}^{\nu}_{\ell}(y)\ dy \tag{9}$$

$$\alpha^{\mu\mu'}_{\ell\ell'}(h_i) = \frac{16\pi^2}{(2\ell+1)(2\ell'+1)}\ \dot{k}^{*\mu'}_{\ell'}(h_i)\ \dot{k}^{\mu}_{\ell}(h_i) \tag{10}$$

$$\zeta^{\nu\nu'}_{\ell\ell'}(i) = \int_{B_i} \dot{k}^{\nu}_{\ell}(y)\ \dot{k}^{*\nu'}_{\ell'}(y)\ dy \tag{11}$$

$$P(h_i) = \int_{B_i} [\hat{P}_{h_i}(y)]^2\ dy. \tag{12}$$

The coefficient normalizing the measured pole figure values $\hat{P}_{h_i}(y)$ is equal

$$N_{h_i} = \frac{1}{P(h_i)} \sum_{\ell=0}^{\ell_{max}} \sum_{\mu\nu} C^{\mu\nu}_{\ell}\ q^{\mu\nu}_{\ell}(h_i). \tag{13}$$

The high degree of the system of equation (6) makes it impossible to solve without the use of a computer. For example, for cubic symmetry of the crystallographic lattice and rhombic symmetry of the measured pole figures we have for the orders of series expansion ℓ_{max} 18, 20 and 22 the respective numbers of 78, 100 and 124 equations.

This method [relations (6) to (13)] is used to determine the values of $C^{\mu\nu}_{\ell}$ and N_{h_i} in the computer program included in this paper.

Another way of finding $C^{\mu\nu}_{\ell}$ values is similar to that

which is used in the case of complete pole figures. It consists of the following procedures.

(a) Calculation of the coefficients $F_\ell^\nu(h_i)$ of the series expansion of the surface spherical functions $\dot{k}_\ell^\nu(y)$ of the pole $P_{h_i}(y)$,

$$P_{h_i}(y) = \bar{N}_{h_i}\hat{P}_{h_i}(y) = \sum_{\ell=0}^{\infty}\sum_{\nu=1}^{N(\ell)} F_\ell^\nu(h_i)\ \dot{k}_\ell^\nu(y). \tag{14}$$

The mean-square approximation condition now has the form

$$r = \int_{B_i}\left[\bar{N}_{h_i}\hat{P}_{h_i}(y) - \sum_{\ell\nu}^{\ell_{max}} F_\ell^\nu(h_i)\ \dot{k}_\ell^\nu(y)\right]^2 dy = \min. \tag{15}$$

As before, we make use of the fact that the pole figure normalization condition imposes the value $F_0^1(h_i) = \sqrt{4\pi}$. Differentiating the expression (15) with respect to $F_{\ell'}^{\nu'}(h_i)$ $(\ell' \neq 0)$ and with respect to $\bar{N}_{h_i}$ yields the system of equations

$$\sum_{\ell=2,\nu}^{\ell_{max}} \gamma_{\ell\ell'}^{\nu\nu'}(h_i)\ F_\ell^\nu(h_i) = -\sqrt{4\pi}\ \ \gamma_{0\ell'}^{1\nu'}(h_i) \tag{16}$$

where

$$\gamma_{\ell\ell'}^{\nu\nu'}(h_i) = \frac{\rho_\ell^\nu(h_i)\ \rho_{\ell'}^{*\nu'}(h_i)}{P(h_i)} - \zeta_{\ell\ell'}^{\nu\nu'}(i) \tag{17}$$

and $\rho_\ell^\nu(h_i)$ and $\zeta_{\ell\ell'}^{\nu\nu'}(i)$ are given by equations (9) and (11), respectively. The normalization coefficient $\bar{N}_{h_i}$ is

$$\bar{N}_{h_i} = \frac{1}{P(h_i)}\sum_{\ell=0,\nu}^{\ell_{max}} F_\ell^\nu(h_i)\ \rho_\ell^\nu(h_i). \tag{18}$$

(b) Determination of the values of $C_\ell^{\mu\nu}$ from the coefficients $F_\ell^\nu(h_i)$ (just as in the case when pole figures measured in the complete range are available.[1,5]

This way of computing the values of $C_\ell^{\mu\nu}$ takes account of the specific form of the data which are used for approximating the general function of axis distribution, i.e. pole figures or, in other words, the cross sections (A(h=const,y). Such data bear the name of nested data.[7]

Some Comments on the Precision of Coefficient $C_l^{\mu\nu}$ *Determination*

We obtain the values of $C_l^{\mu\nu}$ by taking the mean-square approximation of measured pole figure data by employing the finite functional series (1). The error $\Delta^2 C_l^{\mu\nu}$ which burdens the values of $C_l^{\mu\nu}$ determined in this way is affected by the following factors:

(1) experimental errors of the measured results, i.e. pole figures [the effect of measurement error can be accounted for by introducing into equation (5) the weight function $w(h,y)$, assuming, e.g., that $w(h,y) = 1/S^2(P_{hy})$, where $S^2(P_{hy})$ denotes the variance estimator of the pole figure of plane h at point y];

(2) termination of the series expansion at a finite value of l_{max};

(3) numerical integration of the pole figure [the integrals $\rho_l^{\nu}(h_i)$ and $P(h_i)$];

(4) the accuracy of the used tabular data and algebraic rounding-offs which are made by the computer during program execution.

Let us assume that the first two of these factors bear a dominating effect on the precision of the determined value of $C_l^{\mu\nu}$. This is justifiable on the grounds of experience. Frequently, it is assumed that all weights are identical, i.e., $w(h,y)$ = constant.

We can then evaluate the values of $\Delta^2 C_l^{\mu\nu}$, which correspond to the coefficients $C_l^{\mu\nu}$ determined on the basis of the system of equations (6), by means of the relation

$$\Delta^2 C_l^{\mu\nu} = (\Gamma^{-1})_{ll}^{\mu\mu\nu\nu} \frac{R_{min}}{J - K} \tag{19}$$

where

$(\Gamma^{-1})_{ll}^{\mu\mu\nu\nu}$ denotes the diagonal term of the inversed matrix of coefficients of the system of equations (6), R_{min} is the value of expression (5) which is obtained after the calculated values of $C_l^{\mu\nu}$ and N_{h_i} are inserted, $J = \sum_{i=1}^{I} J_i$, J_i being the number of points at which the values of the h_i plane's pole figure were measured, and $K = K(l_{max}) = \sum_{l=2\mu\nu}^{l_{max}} 1$ is the number of unknown coefficients $C_l^{\mu\nu}$.

Considering the fact that the nested data are used, we must be careful when using formula (19) that the following conditions are fulfilled.

$$\min (J_i) > k \quad \text{and} \quad I > M(\ell_{max}), \text{ where } k = \sum_{\ell=2\nu}^{\ell_{max}} 1 .$$

In a similar way we assess the precision of the coefficients $F_\ell^\nu(h_i)$ calculated by means of the set of equations (16). We then have

$$\Delta^2 F_\ell^\nu(h_i) = (\gamma_{h_i}^{-1})_{\ell\ell}^{\nu\nu} \frac{r_{min}}{J_i - k} \tag{20}$$

where:

$(\gamma_{h_i}^{-1})_{\ell\ell}^{\nu\nu}$ denotes the diagonal term of the inversed matrix of coefficients of the system of equations (16), and r_{min} is the value of expression (15) which is obtained after the calculated values of $F_\ell^\nu(h_i)$ and $\bar{N}_{h_i}$ are inserted. The values of $\Delta^2 F_\ell^\nu(h_i)$ determined thus can then be used as weights when determining the coefficients $C_\ell^{\mu\nu}$.

Discussion of Results of Processing Incomplete Pole Figures

A general idea about the precision of orientation distribution determination on the basis of incomplete pole figures is also obtained by comparing the ODF's. To this aim, let us introduce the coefficient $\delta_{I_o}(I,\alpha_o)$ which defines the relative root-mean-square deviation of an ODF determined on the basis of I incomplete pole figures within the range described by angle α_o from the function determined on the basis of I_o complete pole figures,

$$\delta(I,\alpha_o) = \frac{\oint [f_{I\alpha_o}(g) - f_{I_o}(g)]^2 dg}{\oint [f_{I_o}(g) - f_r(g)]^2 dg} =$$

$$\frac{\sum_{\ell\mu\nu}^{\ell_{max}} \frac{1}{2\ell+1} [C_\ell^{\mu\nu}(I,\alpha_o) - C_\ell^{\mu\nu}(I_o)]^2}{\sum_{\ell\mu\nu}^{\ell_{max}} \frac{1}{2\ell+1} [C_\ell^{\mu\nu}(I_o)]^2 - 1} \tag{21}$$

where $f_r(g) \equiv 1$ represents the equally-probable orientation distribution.

The characteristic plot of δ as a function of α_o and I has been described in detail in Ref. 5. Figure 2a presents the δ plots obtained for I = 3 and I = 4 (I_o = 4); they

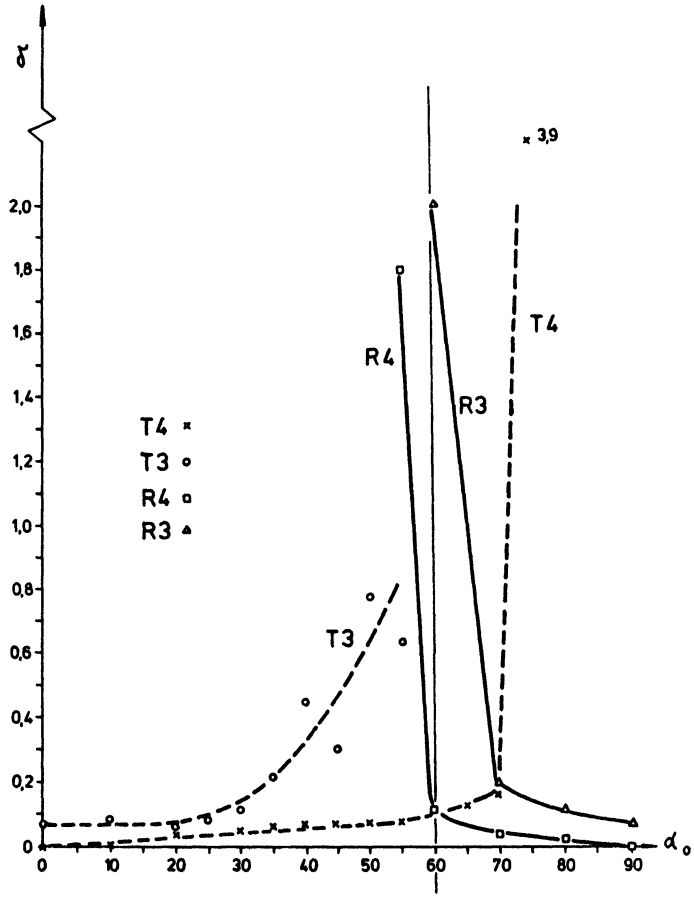

Figure 2a. Curves $\delta_4(3,\alpha_o)$ and $\delta_4(4,\alpha_o)$ representing root-mean-square deviation of ODF's calculated on the basis of pole figures determined in back-reflection technique region (R3 and R4) or in transmission technique region (T3 and T4). Pole figure values measured for silver of rolling reduction 90.9%.[8]

correspond to the pole figure region accessible to back-reflection techniques (curves R3 and R4). These are supplemented by curves (labelled T3 and T4) which correspond to the area accessible to the transmission technique. It may be considered that the value of angle α_o beyond which there follows a steep rise of $\delta_{I_o}(I,\alpha_o)$ marks out the limits of the measurement of I incomplete pole figures necessary for deriving the three-dimensional distribution, In turn Figure 2b shows analogous plots, but obtained for figures featuring

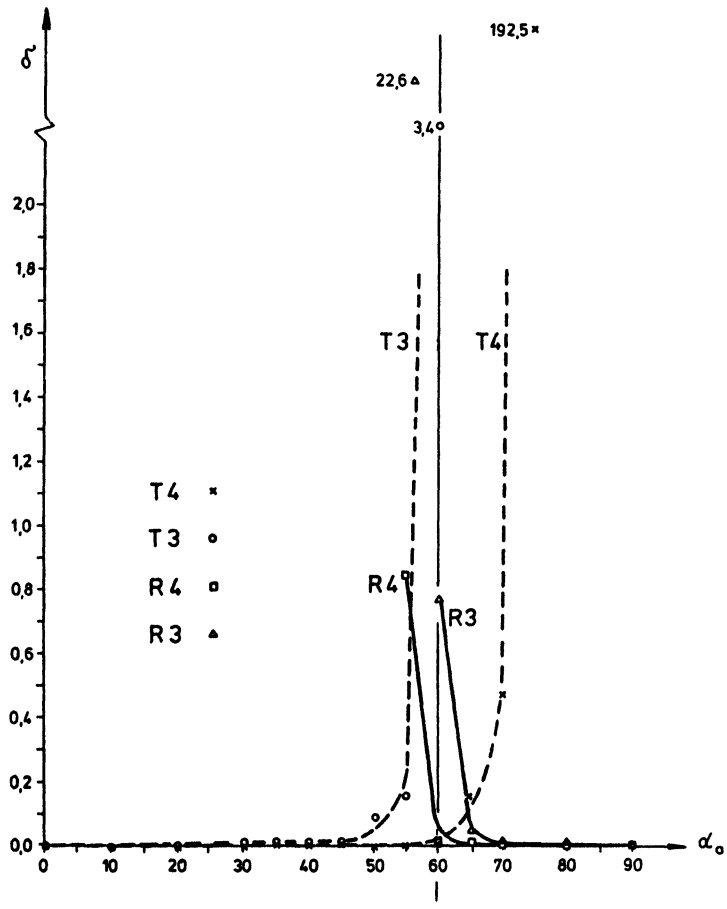

Figure 2b. Curves $\delta_4(3,\alpha_0)$ and $\delta_4(4,\alpha_0)$ representing root-mean-square deviation of ODF's calculated on the basis of pole figures determined in back-reflection technique region (R3 and R4) or in transmission technique region (T3 and T4). Pole figure values calculated from single orientations measured in copper of rolling reduction 95%.[9]

total mutual compatibility (the pole figures have been found on the basis of measurements of single orientations.

If we compare the values of $\bar{\alpha}_0$ for both measuring techniques, we find that when the transmission technique is used we can waive performing measurements in a range of changes of angle α_0 almost twice as large as in the case of the back-reflection method. This is associated with the different sizes of spherical surface elements to which the measured values of the pole figures are related, $\Delta y \approx \sin\alpha\Delta\alpha\Delta\beta$. For angles α near zero we have $\Delta y \approx 0$, hence, values in the vicinity of the center of the figure are strongly attenuated. Therefore, omission of data from the central region

of the pole figure affects less drastically, $C_L^{\mu\nu}$ and ODF values than would stem from the omission of any values from the region where α is nearly 90°.

As follows from the values of the integrals

$$\int_0^{\pi/2}\int_0^{\pi/3} \sin\alpha d\alpha d\beta = \int_0^{\pi/2}\int_{\pi/3}^{\pi/2} \sin\alpha d\alpha d\beta = \pi/4 \qquad (22)$$

the surface area of the spherical segment described by angles $0° \leqslant \alpha \leqslant 60°$ is equal to the area of the spherical surface described by angles $60° \geqslant \alpha \geqslant 90°$. Let us take the angle $\alpha_o = 60°$ as a hypothetical limit of range equivalence of the two measuring techniques for the ideal case (equally-probable crystallite distribution; no errors of experimental data, series termination, numerical integration, tabular data and algebraic rounding). As the number of measured pole figures decreases, the texture components appear, and their sharpness increases, and owing to the unavoidable occurrence of the errors just listed, the curves $\delta_{I_o}(I,\alpha_o)$ should feature an increasingly higher $\bar{\alpha}_o$ for the back-reflection technique and and increasingly lower $\bar{\alpha}_o$ for the transmission technique. Analysis of the provided δ plots corroborates this hypothesis. This is particularly well seen in Figure 2b, where the data were in the form of figures featuring total mutual compatibility. The curves of Figure 2a were obtained for measured figures which were therefore burdened with error of measurement. In this case we observe a rather pronounced deviation from the predicted traces, especially for the curve labelled T3. For the curve T4 the steep rise in δ value begins only when $\alpha_o > 70°$, but this can be explained as being due to random compensation of errors.

Apart from the parameters I and α_o, the shape of the δ plot is undoubtedly affected also by the number and type of the components in the texture of the analyzed sample (their arrangement in orientation space), and also by the degree of texture sharpness. The plots presented here are just illustrative, and only give a general idea about how the δ function behaves.

CONCLUSION

The presented examples of δ function plots allow us to presume that in the case when rolling textures of cubic metals are examined the coefficients $C_\ell^{\mu\nu}$ can be determined on the basis of incomplete pole figures measured in the ranges as outlined in the table.

Up-to-date diffractometric equipment adapted for pole figure measurements[10] allows the performance of measurements in ranges fully qualifying the figures to be processed by the method presented in the present paper.

I	$\bar{\alpha}_o$	
	back-reflection method	transmission method
3	70°	30°
4	60°	60°

INCOMPLETE POLE FIGURE PROCESSING - THE INCPOL COMPUTER PROGRAM

The INCPOL computer program determines the values of coefficients $C_l^{\mu\nu}$ of the series expansion of orientation distribution functions (ODF).

The current version of the program allows the coefficients $C_l^{\mu\nu}$ up to an order of $l_{max} = 22$ to be computed. The data for computations are in the form of three or four incomplete pole figures measured every 5° by the back-reflection or transmission technique. The ranges of measurement of all figures must be identical, and the step along both figure coordinates, the radial angle α and the horizontal angle β, must be 5°.

All parameters and data are entered by the card reader (logical unit number 2), while the output is via the line printer (unit number 3). The INCPOL program also makes use of tabular data files prepared by the INPLIB program. The device for inputting the tabular quantities (magnetic tape or disk unit) is assigned logical unit number 7. Moreover, during computation the program forms a local file on magnetic tape or disk; the assigned logical unit number is 1.

The successive stages of data processing are accomplished in subroutines united with the master program through COMMON statements. The functions of the various subroutines are as follows.

Subroutine METO

Reads the values of the general parameters characterizing the prepared data.

Parameters[1]

I) LFMAX - number of measured pole figures (LFMAX ≙ I).
The FORMAT statement is I2.
Permissible values of LFMAX: 3 or 4.

II) XMETH, XBETA - identify the pole figure measuring technique used and the method by which the figure's value is measured for the horizontal coordinate β.
FORMAT statement: A8,2X,F10.1.
Permissible values are:

[1]In this and subsequent sections the positions I, II, etc. correspond to the successive records (cards) input by the card reader.

$$\text{XMETH} = \begin{cases} \text{REFL} - \text{back-reflection measurements} \\ \text{TRAN} - \text{transmission measurements} \end{cases}$$

$$\text{XBETA} = \begin{cases} 0.0 - \text{pole figure values measured for angles } \beta = 0\cdot\Delta\beta + n\cdot\Delta\beta,\ n = 0,1,\ldots \\ 0.5 - \text{pole figure values measured for angles } \beta = 0.5\cdot\Delta\beta + n\cdot\Delta\beta,\ n = 0,1,\ldots \end{cases}$$

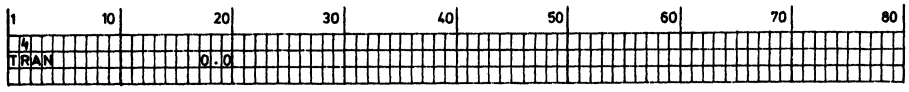

Figure 3. Example of cards with parameters to METO subroutine.

Subroutine CORINP

Reads values of pole figure, $\hat{P}_{h_i}(\alpha,\beta)_{exp}$ and corrects them, i.e., the effect of measurement geometry is considered. The value of a corrected figure is

$$\hat{P}_{h_i}(\alpha,\beta) = c(\alpha)\cdot\hat{P}_{h_i}(\alpha,\beta)_{exp} \tag{23}$$

where $c(\alpha)$ denotes the corrective factor.

Parameters

I) HKL, DF, DFM - identify the Miller indices $\{hkl\} \equiv h_i$ of the pole figure, and the steps of the radial coordinate $\Delta\alpha$ and the horizontal coordinate $\Delta\beta$, respectively.
FORMAT statement: 3F10.0.
Permissible values are:
HKL $\in$ [100, 110, 111, 102, 112, 122, 103, 113][2]
DF = DFM = 5.

II) GR - identifies the range of the radia angle α (in degrees) within which pole figure measurements were made (GR $\hat{=}$ α_o).

$$\text{GR} = \begin{cases} \alpha_R \rightarrow \alpha \in [0,\alpha_R] \text{ for back-reflection technique} \\ \alpha_T \rightarrow \alpha \in [\alpha_T,90°] \text{ for transmission technique} \end{cases}$$

FORMAT statement: F10.0.

Data - Corrective factors $c(\alpha)$ to values $\hat{P}_{h_i}(\alpha,\beta)_{exp}$, stored in array TR(I). They are introduced from 0° to α_R for back-reflection measurements or from α_T to 90° for transmission measurements.
FORMAT statement: 8F10.4.
Values of pole figure $\hat{P}_{h_i}(\alpha,\beta)_{exp}$ (stored in array R(J,I), I $\hat{=}$ α, J $\hat{=}$ β), introduced for successive values of angle β counted from rolling direction to transverse direction towards increasing values of angle α (Figure 1).
FORMAT statement: 20F4.0.

[2]Cf. description if INPLIB program.

Results - The CORINP subroutine rewrites all input values of parameters and data, and prints the corrected values of the pole figure, $\hat{P}_{h_i}(\alpha,\beta)$ [stored in array R(J,I)]. Also, it records the corrected figure on magnetic tape or disk for subsequent computations (normalization[3]).

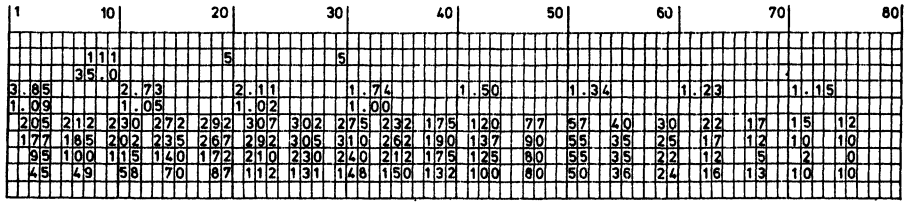

Figure 4. Example of cards with parameters and data to CORINP subroutine.

Subroutine TWOINC

Uses a simple method of numerical integration (formula of trapezoids) to determine values of integrals $q_o^{11}(h_i)$, $P(h_i)$ and $\rho_\ell^\nu(h_i)$ according to relations (8), (12) and (9), respectively.

Parameters and data: h_i, $\Delta\alpha$, $\Delta\beta$, α_o and values of corrected pole figure $\hat{P}_{h_i}(\alpha,\beta)$ are transferred via COMMON statement from subroutine CORINP. The subroutine TWOINC also employs tabular data prepared by program INPLIB, viz., the values of Legendre polynomials $P_1^n(\alpha)$ or their Fourier coefficients $a_1'^{ns}$.

Results - The computed values of integrals $q_o^{11}(h_i)$, $P(h_i)$ and $\rho_\ell^\nu(h_i)$ are stored in arrays P1(I), P2(I) and FEX(L,N,I); $\ell \hat{=} L$, $\nu \hat{=} N$, $(h_i) \hat{=} I$.

Subroutine CINPOL

Determines matrix of coefficients and column of free-standing terms for set of equations (6), in which the unknowns are $C_\ell^{\mu\nu}$, i.e. the coefficients of ODF expansion into a series.

A standard library subroutine solves the system (6).

Parameters and data: I, α_o and values of integrals $q_o^{11}(h_i)$, $P(h_i)$ and $\rho_\ell^\nu(h_i)$ are transferred by COMMON statements. The integrals $\zeta_{\ell\ell'}^{\nu\nu'}(i)$ and $\zeta_{0\ell}^{1\nu'}(i)$ and the values of spherical functions $\dot{k}_\ell^\mu(h_i)$ (stored in arrays MCC, CFW and XK, respectively) are read from the library file prepared by the INPLIB program.

[3]Cf. NOPOLF subroutine.

Results - The values $\Gamma^{\mu\mu'\nu\nu'}_{\ell\ell'}$ and $\Gamma^{1\mu'\nu'}_{0\ell'}$ are arranged in arrays MCC(J,I) and CFW(J); J $\hat{=}$ (l'μ'ν'), I $\hat{=}$ (lμν). After the system of equations (6) is solved the computed values of $C^{\mu\nu}_{\ell}$ are stored in successive cells of array CFW(J).

The values of $C^{\mu\nu}_{\ell}$ are printed out by the line printer, Figure 5.

ℓ	μ	ν = 1	2	3	4	5	6	7	8	9	10	11	12
4	1	-.989	-.062	-2.205									
6	1	-.751	3.965	-2.004	-.495								
8	1	-.815	.124	.982	2.064	1.014							
10	1	1.425	-1.472	-.498	.211	.030	1.535						
12	1	-.055	-.603	-.015	.026	-1.057	-1.212	.414					
12	2	-1.537	.293	-2.149	1.752	.615	-.757	.475					
14	1	-1.226	-.109	.554	-.766	.229	-.896	-.160	-.736				
16	1	1.235	.264	-.535	.109	-.174	-.034	.915	.358	.088			
16	2	.406	-.715	1.663	.937	-.408	.654	.178	.444	.604			
18	1	.049	.353	.489	.025	-.399	-.295	.133	.315	.528	-.196		
18	2	.670	-.470	.103	-.811	.404	-.045	-.482	.460	.318	-.379		
20	1	.165	.064	.041	-.060	-.020	.022	-.015	-.018	-.021	-.000	-.025	
20	2	.162	-.128	.086	.024	.006	-.049	.042	.002	-.021	.007	-.030	
22	1	-.002	.259	-.003	-.026	-.025	-.037	.000	-.001	.011	.035	-.006	.008
22	2	-.391	.193	-.036	-.012	.061	0.23	.023	-.010	.019	.031	.001	-.016

Figure 5. Array of ODF coefficients $C^{\mu\nu}_{\ell}$ printed by CINPOL subroutine.

Subroutine NOPOLF

On the basis of the $C^{\mu\nu}_{\ell}$ values determined by the subroutine CINPOL it computes according to formula (13) the normalization coefficients N_{hi} of the corrected pole figures $\hat{P}_{hi}(\alpha,\beta)$. Next, there is normalization of the pole figure,

$$P_{hi}(\alpha,\beta) = N_{hi}\cdot\hat{P}_{hi}(\alpha,\beta). \tag{24}$$

Parameters and data: I_p, α_o, $C^{\mu\nu}_{\ell}$, $q^{11}_{o}(h_i)$, $P(h_i)$, $\rho^{\nu}_{\ell}(h_i)$, $\dot{k}^{\mu}_{\ell}(h_i)$ are transferred via COMMON statements. The corrected values of the pole figures are read in from magnetic tape or disk, where they had previously been recorded by the subroutine CORINP.

Results - The line printer outputs the normalization coefficients and the normalized pole figures.

Comments on the Range of Computations of the INCPOL Program

The INCPOL program is written in the FORTRAN language. The text included here is the standard version of the program processing incomplete pole figures. It occupies some 33 kilowords of the computer's central memory, while the execution time is about 40 central processor seconds (in the case of the CDC-CYBER 72 computer).

The admissible values of parameters, i.e. the Miller indices of the pole figures being processed, their number and the figure coordinate steps, are dictated by experience in such measurements. Likewise, the accepted order of series expansion $\ell_{max} = 22$ to which values of $C_{\ell}^{\mu\nu}$ have to be determined, corresponds to the standards of calculations made when processing pole figures of metals crystallizing in the cubic system. The possibility of a slight modification (a change in the dimensions of arrays declared in the DIMENSION and COMMON statements) lets it be used for analyzing any set of incomplete pole figures measured in the same area B (Figure 1). For instance, the step of the coordinate $\Delta\alpha$ or $\Delta\beta$ can be any submultiple of the number 90. It is then only necessary to change the dimensions of the arrays R, TR, D, P, DESCRI and WORK, namely, R(90/$\Delta\beta$ + 1, 90/$\Delta\alpha$ + 1), TR(90/$\Delta\alpha$ + 1), D(ℓ_{max}/2 + 1, 90/$\Delta\alpha$ + 1), DESCRI(90/$\Delta\beta$ + 1), P(90/$\Delta\beta$ + 1). The new size of the array WORK stems from the relation WORK = R + FEX + P1 + P2 + GR.

The root-mean-square approximation method used for determining the values of $C_{\ell}^{\mu\nu}$ implies a symmetrical matrix of coefficients of the system of equations (6) resolved in subroutine CINPOL [MCC(J,I) = MCC(I,J)]. Therefore, it is possible to considerably reduce the area of computer memory occupied by the INCPOL program if the set of equations is solved by a subroutine utilizing only a triangular section of the coefficient matrix. The MCC matrix recorded column by column then represents a vector possessing only 7750 elements, whereas the whole matrix has 15376.

Library Program INPLIB

Since the INCPOL program is used in routine computations, it is advantageous (regarding the shorter time of computer operation) to record any used tabular data on magnetic tape or disk. Thus a permanent file becomes formed. The INPLIB program serves this purpose, and the file is utilized in every INCPOL program execution.

The parameters and data are input by the card reader (logical unit number 2), while the results are output onto magnetic tape or disk (unit number 7). Also, the line printer (number 3) rewrites the data.

Parameters

XGRL, XGRU, XGRS identify the lower α_{ℓ} and upper α_{u} limits of radial angle changes and the size of the step $\Delta\alpha$ (in degrees).

FORMAT statement: 3F10.1.

Standard parameter values are: 0, 90 and 5.

Data - Values of Fourier coefficients a'^{ns}_{ℓ} of the associated Legendre polynomials $P^{n}_{\ell}(\alpha)$ for:

ℓ every 2 from 4 to 22,
n every 2 from 0 to 1,
s every 2 from 0 to 1.

Values of surface spherical functions of cubic symmetry $\dot{k}^{\mu}_{\ell}(h_i)$ for:

$h_i = \{hkl\}_i$ - {100},{110},{111},{102},{112},{122},{103},{113

ℓ every 2 from 4 to 22,
μ every 1 from 1 to M(1).

FORMAT statement: 8F10.6.
Each value of h_i and ℓ and n opens a new record, hence, a new punchcard. The values of a'^{ns}_{ℓ} and $\dot{k}^{\mu}_{\ell}(h_i)$ can be found in Refs. 1 and 3.

Results - The line printer rewrites all input values of a'^{ns}_{ℓ} and $\dot{k}^{\mu}_{\ell}(h_i)$. On magnetic tape or disk a library file is recorded, containing: associated Legendre polynomials $P^{n}_{\ell}(\alpha)$ for ℓ every 2 from 4 to 22, n every 2 from 0 to 1, and α every $\Delta\alpha$ from α_{ℓ} to α_u; values of function $\dot{k}^{\mu}_{\ell}(h_i)$; coefficients a'^{ns}_{ℓ}; integrals $\zeta^{\nu\nu'}_{\ell\ell'}(\alpha_o)$ and $\sqrt{4\pi}\ \zeta^{1\nu'}_{0\ell'}(\alpha_o)$ [according to relation (11)] for ℓ every 2 from 4 to 22, n every 2 from 0 to 1, ℓ' every 2 from 4 to 22, n' every 2 from 0 to ℓ', and α_o every $\Delta\alpha$ from α_{ℓ} to α_u.

The program INPLIB is written in FORTRAN. The program takes up some 31 kilowords, and the time of execution for standard values of the parameters is about 350 central processor seconds (CDC-CYBER 72 computer).

```
C
C
C     PROGRAM  INCPOL(TAPE7,TAPE1,INPUT,OUTPUT,TAPE2=INPUT,TAPE3=OUTPUT)
C
      COMMON WORK(850)
      COMMON /SEI/ IN,INP,LIB,IOUT,LTP,LMB
      COMMON /DAI/ HEAD(22),ACT(10)
      COMMON /PAR/ LMAX,LFMAX
      COMMON /ORG/ LIFI,LOK,J,I
      COMMON /EVD/ CFW(124),V(10),XK(10,2,8),NRH(4)
      COMMON /MET/ XMETH,IBETA
      IN=2
      INP=2
      IOUT=3
      LMB=1
      LIB=7
      LMAX=22
      LMAX=LMAX/2
      LOK=10
      CALL METO
```

```
      DO 1 LIFI=1,LFMAX
      CALL CORINP
      CALL TWOINC
    1 CONTINUE
      CALL CINPOL
      CALL NOPOLF
      STOP
      END


C
      SUBROUTINE METO
C
      COMMON /SEI/ IN,INP,LIB,IOUT,LTP,LMB
      COMMON /MEI/ XMETH,IBETA
      DATA RMETH/8HREFL    /,TMETH/8HTRAN
      READ(IN,1001) LFMAX
      READ(IN,1000) XMETH,XBETA
      IF(XBETA.EQ.0.) IBETA=1
      IF(XBETA.EQ.0.5) IBETA=2
      IF(XMETH.EQ.RMETH) WRITE(IOUT,5000)
      IF(XMETH.EQ.TMETH) WRITE(IOUT,5500)
      RETURN
 1000 FORMAT(A8,2X,F10.1)
 1001 FORMAT(I2)
 5000 FORMAT(1H ////1H ,20X,79HPOLE   FIGURES   WERE   MEASURED   BY   M
     *EANS   OF   BACK   REFLECTION   METHOD/1H ,20X,80(1H*)/////)
 5500 FORMAT(1H ////1H ,23X,74HPOLE   FIGURES   WERE   MEASURED   BY   M
     *EANS   OF   TRANSMISSION   METHOD/1H ,23X,74(1H*)/////)
      END


C
      SUBROUTINE CORINP
C
      DIMENSION TR(19)
      COMMON R(19,19),REX(10,12,4),P1(4),P2(4),GR
      COMMON /SEI/ IN,INP,LIB,IOUT,LTP,LMB
      COMMON /DAI/ HEAD(22),ACT(10)
      COMMON /PAR/ LMAX,LFMAX
      COMMON /ORG/ LIFI,LOK,JJ,II
      COMMON /MEI/ XMETH,IBETA
      DATA XREFL/8HREFL    /,XTRAN/8HTRAN
      READ(IN,1000) HKL,DF,DFM
      READ(IN,1000) GR
      HEAD(LOK+1)=HKL
      HEAD(LOK+2)=DF
      HEAD(LOK+3)=DFM
      IF(IBETA.EQ.1) GM=90.
      IF(IBETA.EQ.2) GM=87.5
      J2=INT((GM+.0001)/DF)+1
      JJ=J2
      IF(XMETH.EQ.XREFL) GO TO 100
      I1=INT((GR+.0001)/DF)+1
      I2=INT((90.+.0001)/DF)+1
      II=I1
      GO TO 101
  100 CONTINUE
      I1=1
      I2=INT((GR+.0001)/DF)+1
      II=I2
  101 CONTINUE
      READ(INP,1100)(TR(I),I=I1,I2)
      DO 10 I=I1,I2
      READ(INP,1200)(R(J,I),J=1,J2)
   10 CONTINUE
      WRITE(IOUT,3100) HKL
      WRITE(IOUT,2200) HKL,GR,DF,GM,DFM
      WRITE(IOUT,4000)(TR(I),I=I1,I2)
      IF(XMETH.EQ.XREFL) WRITE(IOUT,5000)
      IF(XMETH.EQ.XTRAN) WRITE(IOUT,5500)
      DO 11 I=I1,I2
   11 WRITE(IOUT,3001)(R(J,I),J=1,J2)
      DO 20 I=I1,I2
      DO 20 J=1,J2
```

B

```
      R(J,I)=R(J,I)*TR(I)
   20 CONTINUE
      Z=0.
      IF(IBETA.EQ.2) Z=DFM/2.0
      DO 30 J=1,J2
   30 TR(J)=DFM*FLOAT(J-1)+Z
      WRITE(IOUT,3003) HKL
      WRITE(IOUT,3001)(TR(J),J=1,J2)
      DO 40 I=I1,I2
      XPS=DF*FLOAT(I-1)
      WRITE(IOUT,3002) XPS,(R(J,I),J=1,J2)
   40 CONTINUE
      WRITE(LMB) R
      LOK=LOK+3
      RETURN
 1000 FORMAT(8F10.0)
 1100 FORMAT(8F10.4)
 1200 FORMAT(20F4.0)
 2200 FORMAT(1H ,25HINCOMPLETE  POLE  FIGURE ,F8.0//32H ANGULAR  LIMITS
     * AND  STEPS    //7H    GR=,F6.1,7H    DR=,F6.1/7H    GH=,F6.1,7H
     *  DH=,F6.1)
 3001 FORMAT(1H //1H ,6X,19F6.0)
 3002 FORMAT(1H /1H ,2UF6.0)
 3003 FORMAT(1H1///1H ,40H CORRECTED   INCOMPLETED   POLE   FIGURE,F6.0)
 3100 FORMAT(1H1////38X,38H EVIDENCE   FOR   POLE   FIGURE   HKL=,F5.0//
     *)
 4000 FORMAT(1H ///1H ,27H  CORECTION   COEFFICIENTES///6(1H ,8F10.4//))
 5000 FORMAT(1H //1H ,18H REFLECTION   PART)
 5500 FORMAT(1H //1H ,20H TRANSMISSION   PART)
      END


C
      SUBROUTINE TWOINC
C
      DIMENSION NOR(12)
      DIMENSION D(12,19),P(19),A(40)
      COMMON R(19,19),FEX(10,12,4),P1(4),P2(4),GR
      COMMON /SEI/ IN,INP,LIB,IOUT,LTP,LMB
      COMMON /DAT/ HEAD(22),ACT(10)
      COMMON /PAR/ LMAX,LFMAX
      COMMON /ORG/ LIFI,LOK,J,I
      COMMON /MET/ XMETH,IBETA
      DATA RMETH/8HREFL    /,TMETH/8HTRAN
      DATA NOR/1,2,3,4,5,6,7,8,9,10,11,12/
      XB=0.01745329
      LP1=LMAX+1
      DF=HEAD(LOK-1)
      DFM=HEAD(LOK)
      FLI=FLOAT(IBETA)
      BEX=8.*DF*DFM*XB*XB
      XKK=2.0*DFM*XB/FLI
      XLL=2.0*DF*XB
      JP1=J
      IF(XMETH.EQ.RMETH) GO TO 100
      IP1=I
      IP2=INT((90.+.0001)/DF)+1
      GO TO 101
  100 CONTINUE
      IP1=1
      IP2=I
  101 CONTINUE
      Q=0.0
      DO 8 IM=IP1,IP2
      Y=SIN(FLOAT(IM-1)*DF*XB)
      JP2=JP1-1
      Z=.5*FLI*R(1,IM)**2
      DO 2 JX=2,JP2
    2 Z=Z+R(JX,IM)**2
      Z=Z+.5*FLI*R(JP1,IM)**2
      IF(IM.EQ.IP1.OR.IM.EQ.IP2) Z=.5*Z
      Q=Q+Z*Y
      DO 9 JM=1,JP1
    9 P(JM)=R(JM,IM)*Y
      P(1)=.5*FLI*P(1)
```

```
      P(JP1)=.5*PLI*P(JP1)
      DO 10 N=1,LP1
      X=0.0
      XNN=FLOAT(N-1)*XKK
      DO 11 JM=1,JP1
   11 X=X+COS(FLOAT(IBETA*JM-1)*XNN)*P(JM)
      IF(IM.EQ.IP1.OR.IM.EQ.IP2) X=.5*X
      D(N,IM)=X
   10 CONTINUE
    8 CONTINUE
      P2(LIFI)=Q*BEX
      X=0.0
      DO 13 IM=IP1,IP2
   13 X=X+D(1,IM)
      P1(LIFI)=X*BEX
      IF(DF.EQ.5.) GO TO 710
      DO 711 K=1,76
  711 READ(LIB)
  710 CONTINUE
      DO 14 LM=2,LMAX
      LN1=LM-1
      LMP1=LM+1
      DO 15 N=1,LMP1
      IF(DF.NE.5.) GO TO 713
      READ(LIB) A
      Y=0.0
      DO 16 IM=IP1,IP2
      Y=Y+A(IM)*D(N,IM)
   16 CONTINUE
      GO TO 714
  713 READ(LIB)A
      Y=0.0
      DO 715 IM=IP1,IP2
      X=A(1)
      XII=FLOAT(IM-1)*XLL
      DO 17 IS=2,LMP1
   17 X=X+A(IS)*COS(FLOAT(IS-1)*XII)
      Y=Y+X*D(N,IM)
  715 CONTINUE
  714 CONTINUE
      Y=0.564189584*Y*BEX
      IF(N.NE.1) GO TO 18
      Y=Y*0.707106781
   18 CONTINUE
      FEX(LN1,N,LIFI)=Y
   15 CONTINUE
   14 CONTINUE
      REWIND LIB
      RETURN
      END


C
      SUBROUTINE CINPUL
C
      REAL MCC
      DIMENSION MCC(124,124)
      DIMENSION NO(12),MO(12),MODHKL(8)
      DIMENSION XK1(4),XK2(4),YF1(4)
      COMMON R(19,19),FEX(10,12,4),P1(4),P2(4),GR
      COMMON /SEI/ IN,INP,LIB,IOUT,LTP,LMB
      COMMON /DAI/ HEAD(22),ACT(10)
      COMMON /PAR/ LMAX,LFMAX
      COMMON /ORG/ LIFI,LOK,LIMJ,LIMI
      COMMON /EVD/ CFW(124),V(10),XK(10,2,8),NRH(4)
      COMMON /MEI/ XMETH,IBETA
      DATA RMETH/8HREFL    /,TMETH/8HTRAN    /
      DATA MO/1,0,1,1,1,1,2,1,2,2,2,2/
      DATA NO/1,2,3,4,5,6,7,8,9,10,11,12/
      DATA MODHKL/100,110,111,102,112,122,103,113/
      DO 99 I=1,75
   99 READ(LIB)
      READ(LIB) XK
      DO 77 I=1,75
   77 READ(LIB)
```

```
      FI=90.-GR
      KLA1=INT((FI+.0001)/5.)+1
      DO 88 KLA=1,KLA1
      READ(LIB) CFW,MCC
   88 CONTINUE
      REWIND LIB
      LOKX=LOK
      DO 4 K=1,LFMAX
      K1=LFMAX-K+1
      LOKX=LOKX-3
      NHK=INT(HEAD(LOKX+1))
      DO 5 K2=1,8
      IF(MODHKL(K2).EQ.NHK) GOTO 6
    5 CONTINUE
    6 NRH(K1)=K2
    4 CONTINUE
      DO 9 LO=2,LMAX
      LR=LO-1
    9 V(LR)=1./(4.*FLOAT(LO)+1.)
      DO 8 IPF=1,LFMAX
      P2(IPF)=1./P2(IPF)
    8 P1(IPF)=P2(IPF)*P1(IPF)/12.5663706
      CON=1.0
      IF(XMETH.EQ.KMETH) CON=-1.0
      J=0
      LL=LMAX+1
      DO 10 LI=3,LL
      LT1=MO(LI)
      LI2=LI-2
      VLI2=V(LI2)
      DO 20 MI=1,LT1
      FX=.0
      DO 11 IPF=1,LFMAX
      NUM=NRH(IPF)
      XKK=XK(LI2,MI,NUM)
      XK1(IPF)=XKK
   11 FX=FX+XKK
      FX=0.282094792*FX
      DO 30 NI=1,LI
      FY=.0
      DO 12 IPF=1,LFMAX
      YKK=FEX(LI2,NI,IPF)*XK1(IPF)
      YF1(IPF)=YKK
   12 FY=FY+YKK*P1(IPF)
      I=J
      J=J+1
      CFW(J)=VLI2*(FX*CON*CFW(J)-FY)
      DO 1 LN=LI,LL
      LTK=MO(LN)
      LN2=LN-2
      VV=VLI2*V(LN2)
      IF(LN.NE.LI) GO TO 111
      MIX=MI
      GO TO 222
  111 MIX=1
  222 CONTINUE
      DO 2 MK=MIX,LTK
      X=.0
      DO 13 IPF=1,LFMAX
      NUM=NRH(IPF)
      XLL=XK(LN2,MK,NUM)
      XK2(IPF)=XLL
   13 X=X+XLL*XK1(IPF)
      IF(MK.NE.MI.OR.LN.NE.LI) GO TO 333
      NIX=NI
      GO TO 444
  333 NIX=1
  444 CONTINUE
      DO 3 NK=NIX,LN
      Y=.0
      DO 14 IPF=1,LFMAX
   14 Y=Y+XK2(IPF)*P2(IPF)*FEX(LN2,NK,IPF)*YF1(IPF)
      I=I+1
      UMCC=MCC(J,I)
      IF(XMETH.EQ.TMETH) GO TO 500
```

```
      ZX=.0
      IF(LI.EQ.LN.AND.NI.EQ.NK) ZX=1.0
      UMCC=ZX-UMCC
  500 CONTINUE
      ARA=VV*(Y-X*UMCC)
      MCC(J,I)=ARA
      IF(I.EQ.J) GO TO 440
      MCC(I,J)=ARA
  440 CONTINUE
    3 CONTINUE
    2 CONTINUE
    1 CONTINUE
   30 CONTINUE
   20 CONTINUE
   10 CONTINUE
C
C     ------------------------------------------------------------------
C         CALL  HERE  YOUR  SUBROUTINE  FOR  SOLVING  OF  A  SET  OF
C         SIMULTANEOUS  LINEAR  EQUATIONS  WITH
C               MCC  -  MATRIX  OF  COEFFICIENTS
C               CFW  -  VECTOR  OF  RIGHT-HAND  SIDE
C                 J  -  NUMBER  OF  EQUATIONS
C         RESULTS  SHOULD  BE  STORED  IN  CFW  ARRAY
C     ------------------------------------------------------------------
C
      WRITE(IOUT,3300)
      WRITE(IOUT,1002)
      WRITE(IOUT,1000)(NO(II),II=1,LL)
      J=0
      DO 101 L=3,LL
      LTL=NO(L)
      L2=2*(L-1)
      DO 101 MI=1,LTL
      J1=J+1
      J2=J+L
      WRITE(IOUT,1001) L2,MI,(CFW(K),K=J1,J2)

      J=J2
  101 CONTINUE
      RETURN
 1000 FORMAT(1H /10X,12I8)
 1001 FORMAT(1H ,2I5,12F8.3)
 1002 FORMAT(1H ///76H  MATRIX   OF   C - COEFFICIENTS     INCOMPLETE
     *POLE   FIGURES   PROBLEM  ///)
 3300 FORMAT(1H1///////)
      END


C
      SUBROUTINE NUPOLF
C
      DIMENSION DESCRI(19)
      DIMENSION NO(12)
      COMMON R(19,19),HEX(10,12,4),P1(4),P2(4),GR
      COMMON /SEI/ IN,INP,LIB,IOUT,LTP,LMB
      COMMON /DAI/ HEAD(22),ACT(10)
      COMMON /PAR/ LMAX,LFMAX
      COMMON /ORG/ LIFI,LOK,LIMJ,LIMI
      COMMON /EVO/ CFW(124),V(10),XK(10,2,8),NRH(4)
      COMMON /MEI/ XMETH,IBETA
      DATA RMETH/8HREFL    /,TMETH/8HTRAN
      DATA PI4/12.5663706/
      DATA NO/1,0,1,1,1,1,2,1,2,2,2,2/
      DO 500 IPF=1,LFMAX
      BACKSPACE LMB
  500 CONTINUE
      LOK=10
      LL=LMAX+1
      DO 60 IPF=1,LFMAX
      IH=NRH(IPF)
      LOK=LOK+3
      DF=HEAD(LOK-1)
      DFM=HEAD(LOK)
      IF(XMETH.EQ.RMETH) GO TO 100
      LIMP=INT((GR+.0001)/DF)+1
```

```
      LIMI=INT((90.+.0001)/DF)+1
      GO TO 101
  100 CONTINUE
      LIMP=1
      LIMI=INT((GR+.0001)/DF)+1
  101 CONTINUE
      IF(IBETA.EQ.1) GM=90.
      IF(IBETA.EQ.2) GM=87.5
      LIMJ=INT((GM+.0001)/DFM)+1
      I=0
      Z=P1(IPF)/P2(IPF)
      DO 61 L=3,LL
      LTL=MO(L)
      LN2=L-2
      VL=V(LN2)
      DO 61 M1=1,LTL
      X=XK(LN2,M1,IH)
      DO 61 N=1,L
      I=I+1
      Z=Z+X*FEX(LN2,N,IPF)*CFW(I)*VL
   61 CONTINUE
      Z=Z*P2(IPF)*PI4
      HKL=HEAD(LUK-2)
      WRITE(IOUT,6666) HKL,Z
      READ(LMB) R
      DO 62 I=LIMP,LIMI
      DO 62 J=1,LIMJ
   62 R(J,I)=Z*R(J,I)
      Z=.0
      IF(IBETA.EQ.2) Z=DFM/2.0
      DO 59 I=1,LIMJ
   59 DESCRI(I)=DFM*FLOAT(I-1)+Z
      WRITE(IOUT,6667) HKL,(DESCRI(I),I=1,LIMJ)
      DO 63 I=LIMP,LIMI
      Z=DF*FLOAT(I-1)
   63 WRITE(IOUT,6667) Z,(R(J,I),J=1,LIMJ)
   60 CONTINUE
      RETURN
 6666 FORMAT(1H1///1H ,34H INCOMPLETE   POLE   FIGURE   HKL=,F5.0///1H
     *,34H NORMALIZATION   COEFFICIENT   NI=,E16.6///)
 6667 FORMAT(1H ,20F6.1///)
      END


C
C
C     PROGRAM INPLIB(TAPE7,INPUT,OUTPUT,TAPE2=INPUT,TAPE3=OUTPUT)
C
      INTEGER SP,S1,SS,SZ,SJ
      REAL MCC
      DIMENSION MCC(124,124),CFW(124)
      DIMENSION XK(10,2,8),ANS(10,12,12)
      DIMENSION A(40),B(40),MO(12),MODHKL(8)
      DATA MCC/15376*0.0/,CFW/124*0.0/
      DATA XK/160*0.0/,A/40*0.0/,B/40*0.0/
      DATA MO/1,0,1,1,1,1,2,1,2,2,2,2/
      DATA MODHKL/100,110,111,102,112,122,103,113/
      DATA YB/0.01745329/
      IN=2
      IOUT=3
      LIB=7
      WRITE(IOUT,1100)
      READ(IN,1002) XGRL,XGRU,XGRS
      YI=2.0*YB*XGRS
      IGRL=INT((XGRL+.0001)/XGRS)
      JGRL=IGRL+1
      IGRU=INT((XGRU+.0001)/XGRS)
      JGRU=IGRU+1
      DO 11 L=2,11
      LP1=L+1
      LN1=L-1
      DO 11 N=1,LP1
      READ(IN,1000)(A(IS),IS=1,LP1)
      DO 101 IS=1,LP1
  101 ANS(LN1,N,IS)=A(IS)
```

```
      DO 110 I=JGRL,JGRU
      X=A(1)
      XI=YI*FLOAT(I-1)
      DO 112 IS=2,LP1
  112 X=X+A(IS)*COS((IS-1)*XI)
      B(I)=X
  110 CONTINUE
      WRITE(LIB) B
   11 CONTINUE
      WRITE(IOUT,3000)
      DO 120 J=1,8
      WRITE(IOUT,2500) MODHKL(J)
      DO 121 L=1,10
      LTL=MO(L+2)
  121 READ(IN,1000)(XK(L,MI,J),MI=1,LTL)
      DO 122 L=1,10
      LTL=MO(L+2)
      WRITE(IOUT,2200)
  122 WRITE(IOUT,1001)(XK(L,MI,J),MI=1,LTL)
  120 CONTINUE
      WRITE(LIB) XK
      WRITE(IOUT,4000)
      DO 150 I=1,40
  150 A(I)=0.0
      DO 151 L=2,11
      LP1=L+1
      LN1=L-1
      DO 151 N=1,LP1
      DO 153 IS=1,LP1
  153 A(IS)=ANS(LN1,N,IS)
      WRITE(IOUT,2200)
      WRITE(IOUT,1001)(A(IS),IS=1,LP1)
      WRITE(LIB) A
  151 CONTINUE
      IF(XGRL.NE.0.0) GO TO 100
      WRITE(LIB) CFW,MCC
      IGRL=IGRL+1
  100 CONTINUE
      DO 888 IGR=IGRL,IGRU
      FI=XGRS*FLOAT(IGR)
      FI0=FI*YB
      J=0
      DO 10 LI=3,12
      LT1=MO(LI)
      LI2=LI-2
      DO 20 MI=1,LT1
      DO 30 NI=1,LI
      I=J
      J=J+1
      IF(MI.NE.1) GO TO 31
      YX=.0
      IF(NI.NE.1) GO TO 300
      IF(FI.EQ.90.0) GO TO 300
      Z=.0
      Q=1.0
      DO 301 SP=1,LI
      YY=2.*FLOAT(SP-1)
      XX=1.-YY
      ZZ=1.+YY
      Z=Z+ANS(LI2,NI,SP)*(SIN(ZZ*FI0)/ZZ+SIN(XX*FI0)/XX)*Q
      Q=-Q
  301 CONTINUE
      YX=0.707106781*Z
  300 CONTINUE
      GO TO 32
   31 CONTINUE
      JJ=J-LI
      YX=CFW(JJ)
   32 CONTINUE
      CFW(J)=YX
      DO 1 LN=LI,12
      LTK=MO(LN)
      LN2=LN-2
      IF(LN.NE.LI) GO TO 111
      MIX=MI
```

```
      GO TO 222
  111 MIX=1
  222 CONTINUE
      DO 2 MK=MIX,LTK
      IF(MK.NE.MI.OR.LN.NE.LI) GO TO 333
      NIX=NI
      GO TO 444
  333 NIX=1
  444 CONTINUE
      DO 5 NK=NIX,LN
      I=I+1
      IF(MI.NE.1) GO TO 4
      IF(MK.NE.1) GO TO 5
      YX=.0
      IF(NI.NE.NK) GO TO 400
      IF(FI.EQ.90.0) GO TO 500
      Z=.0
      DO 401 SP=1,LI
      T=.0
      DO 402 SI=1,LN
      SS=SI-SP
      SZ=SI+SP-2
      Z1=2.*FLOAT(SS)
      Z2=2.*FLOAT(SZ)
      Q1=1.
      Q2=1.
      IF(SS/2*2.NE.SS) Q1=-1.
      IF(SZ/2*2.NE.SZ) Q2=-1.
      PZ1=1.+Z1
      PZ2=1.+Z2
      ZM1=1.-Z1
      ZM2=1.-Z2
  402 T=T+ANS(LN2,NK,SI)*(Q1*(SIN(PZ1*FI0)/PZ1+SIN(ZM1*FI0)/ZM1)+Q2*(SIN
     *(ZM2*FI0)/ZM2+SIN(PZ2*FI0)/PZ2))
      Z=Z+ANS(LI2,NI,SP)*T
  401 CONTINUE
      YX=0.5*Z
      GO TO 400
  500 IF(LI.EQ.LN) YX=1.0
  400 CONTINUE
      GO TO 6
    4 CONTINUE
      YX=MCC(JJ,I)
      GO TO 6
    5 CONTINUE
      II=I-LN
      YX=MCC(J,II)
    6 CONTINUE
      MCC(J,I)=YX
      MCC(I,J)=YX
    3 CONTINUE
    2 CONTINUE
    1 CONTINUE
   30 CONTINUE
   20 CONTINUE
   10 CONTINUE
      WRITE(LIB) LFW,MCC
  888 CONTINUE
      ENDFILE LIB
      REWIND LIB
      STOP
 1000 FORMAT(8F10.6)
 1001 FORMAT(1H ,12X,8F12.6,12X)
 1002 FORMAT(3F10.1)
 1100 FORMAT(1H1/////1H ,56X,49HCONSTANT    TABLES    TO    TEXTURE    CAL
     *CULATIONS)
 2200 FORMAT(1H )
 2500 FORMAT(1H1//////1H ,27X,6HHKL = ,I5////)
 3000 FORMAT(1H1/////1H ,56X,48HTABLE    OF    CUBIC    SYMETRICAL    FUNCTI
     *ONS    K /////)
 4000 FORMAT(1H1/////1H ,45X,33HTABLE    OF    A,LNS    COEFFICIENTS/////)
      END
```

REFERENCES

1. H. J. Bunge, *Mathematische Methoden der Texturanalyse*, Akademie-Verlag, Berlin, 1969.
2. J. Pospiech and J. Jura, *Z. Metallkunde*, 65, 324 (1974).
3. P. R. Morris, *Advances in X-Ray Analysis*, 18, 514 (1975).
4. J. Jura, J. Pospiech and H. J. Bunge, *Texture*, 1, 201 (1974).
5. J. Jura, J. Pospiech and H. J. Bunge, Papers of Commission of Metallurgy and Foundry (Polish Academy of Sciences), *Metallurgy*, No. 24, pp. 111-176 (1976).
6. J. Jura and J. Pospiech, *Texture*, 2, 81 (1976).
7. D. Cuthbert and F. S. Wood, *Fitting Equations to Data*, Wiley-Interscience, New York, 1971.
8. W. Truszkowski, J. Pospiech, J. Jura and B. Major, *Proceedings of "3^e colloque europeen sur les textures"*, Pont a Mousson, 1973, pp. 235-258.
9. J. Pospiech and K. Lücke, *Acta Met.*, 23, 997 (1975).
10. K. Lücke, G. Burmeister, R. Alan and M. D. Mengelberg, in press.