necessary to limit certain array sizes, or to establish overlay procedures.)

The programs were written with readability more in mind than running speed; they do not use the most efficient algorithms. On the other hand, computer processor time is no longer a significant issue for the analysis in question.

Since the set of programs is split into functionally sized routines, making use of human judgements during the analysis procedure, they can serve as the basis for an on-line, interactive 'subsystem'. The variety of 'time-sharing' systems, and the rapidly changing nature of that component of the computer field, leads us to believe that such implementations will probably continue to be facility-dependent, and hence a local task, for the near future.

The present versions of the programs have been tested, and the documentation and code have been used elsewhere without significant problems. An extensive report describing the use of the programs, listings of the programs and sample data, and punched-card decks of the programs and sample input for each program are available. The requester will be asked to defray the cost of making and sending copies, estimated to be about $50 at present.*

* Inquiries may be sent to Professor Roy Kaplow, Room 13-5106, M.I.T., Cambridge, Massachusetts 02139.

Reference


A new computer program for contour maps of texture functions. By H. W. BERGMANN and R. KERN, Institut für Metallkunde und Metalphysik, and W. BRUNN, Institut für Theoretische Physik B, Universität Clausthal, Clausthal-Zellerfeld, Germany (BRD)

(Received 8 August 1975; accepted 16 December 1975)

A Fortran program for plotting contour maps of texture functions is described. The program works at high speed and is very flexible in application. For input the values of the texture function, which have been measured or calculated on a grid of mesh points, are required. From these data, the single-level lines of a contour map are calculated and plotted. Format and caption of the contour map as well as the marking of the level lines can be individually chosen. In the present paper the program is applied to direct and inverse pole figures as well as to a three-dimensional orientation distribution function (ODF) of a silver sheet deformed by 99.5\%.

Introduction

In recent years research in texture determination and analysis has concentrated on the three-dimensional orientation distribution function (ODF) (Bunge, 1969). Programs were developed which calculate ODF, inverse pole figures or direct pole figures (Jura, Pospiech & Bunge, 1974). In these cases one obtains the actual texture function on certain mesh points which can be the intensities I_{\alpha\beta} of a direct pole figure as measured by an automatic texture analyser (ATEMA) (Grewen, Sauer & Wahl, 1970), or the calculated values of the ODF in Euler space.

Since the output of a numerical array on a line printer is not easy to handle, it is usual to present orientation densities graphically. For direct pole figures plotter programs already exist (Greven et al., 1970; Fornara, Mengelberg & Peltz, 1974).

In this paper a program is described which can plot all the texture functions, i.e. direct pole figures, inverse pole figures and ODF. This program shows flexibility in application.

Short description of the program

In the case of the three-dimensional analysis the problem is simple: one obtains sections of Euler space. In the case of direct or inverse pole figures the problem is more complex. Starting from the corrected and normalized experimental data, the computer ought to work out lines of equal orientation density on the surface of the unit sphere before the stereographic projection is carried out (Fornara et al., 1974). This technique is also applied here.

In Figs. 1-3 the application of the program PFIGUR is demonstrated. Fig. 1 shows a test plot of PFIGUR. Figs. 2 and 3 show measured and calculated direct pole figures, inverse pole figures and the ODF of a silver sheet which was heavily deformed (99-3\%). In Fig. 2 only one quarter of the direct pole figure is shown, but it is also possible to plot the complete figure. The measured and calculated direct pole figures in Figs. 2(a) and 2(b) show some differences in the number and form of the contour lines. This is for two reasons: first, the theoretical and numerical approximations in the calculation of the orientation densities cause a mismatch of

![Fig. 1. Plot of the function f(x,y)=\sin(kx)\sin(ky) with an area of constant value of f(x,y) as a test plot.](image)
Fig. 2. Silver cold rolled to 99.3%. (a) Experimental pole figures. (b) Calculated pole figures. (c) Inverse pole figures. (The difference in intensity between neighbouring contour lines has been chosen as 1 on the scale of arbitrary orientation density.)
maximum 10% (Jura et al., 1974); second, because of the widely spaced grid of mesh points (\(\Delta x = 5^\circ, \Delta \beta = 5^\circ\)), contour lines, especially, of sharp peaks can change their form in the calculated densities so that they drift as a whole into one mesh so that they cannot be found by the linear interpolation of PFIGUR and do not appear in the calculated pole figure. By comparing the contour maps of Fig. 2(a,b), and the values of the minimum and maximum orientation densities of each direct pole figure given in Table 1 both kinds of errors may be estimated.

Table 1. Maximum and minimum orientation densities

<table>
<thead>
<tr>
<th>Pole figure</th>
<th>Measured</th>
<th>Calculated</th>
<th>Measured</th>
<th>Calculated</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.0039</td>
<td>-0.5328</td>
<td>10.558</td>
<td>9.417</td>
</tr>
<tr>
<td>110</td>
<td>0.2401</td>
<td>-0.3562</td>
<td>9.821</td>
<td>9.042</td>
</tr>
<tr>
<td>111</td>
<td>0.0556</td>
<td>-0.2761</td>
<td>8.323</td>
<td>7.157</td>
</tr>
</tbody>
</table>

Title of the Program: PFIGUR.

Computer used: TR4 (Telefunken).

Program language: Fortran IV.

High speed storage required: 5800 words without arrays.

Peripherals required: line printer, plotter (PFIGUR only requires standard software for BENSON or CALCOMP plotter).

Number of cards: 2000 (with comment cards).

Code used: IBM 26 (BCD).

The program PFIGUR calculates and plots lines and planes of contour maps of a real function \(f(x,y)\) of two variables \(x\) and \(y\). If in the contour maps planes of equal orientation density appear, they will be marked by crosses at the mesh points (see Fig. 1 for example). Such a plane is determined as an area in which the deviation of the orientation density from the level in question is smaller than a fixed value. This value can be changed. The function \(f(x,y)\) must be given in an array \(F(I,K)\). By a specific selection of the parameter of PFIGUR the array \(F(I,K)\) is interpreted as a constant-\(\varphi\) section of an ODF or as a direct or inverse pole figure. The corresponding contour maps are plotted in the usual form. Moreover, real functions \(f(x)\) of a variable \(x\) can be plotted in a scaled Cartesian orthogonal coordinate system. The functions \(f(x)\) must also be stored in the array \(F(I,K)\). The general flow chart is given in Fig. 4.

References


