



**HAL**  
open science

## Analysis of images in traverse topographs.

Yves Epelboin, Alain Soyer

► **To cite this version:**

Yves Epelboin, Alain Soyer. Analysis of images in traverse topographs.. Application of X-ray topographic methods to materials science, University Denver, Aug 1983, Snowmass; Colorado, United States. hal-01229002

**HAL Id: hal-01229002**

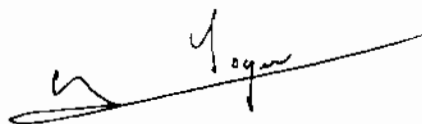
**<https://hal.science/hal-01229002>**

Submitted on 17 Nov 2015

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Snowmass - 1983

A handwritten signature, likely 'Yves Epelboin', written in dark ink. The signature is stylized and appears to be written over a horizontal line.

## ANALYSIS OF IMAGES IN TRAVERSE TOPOGRAPHS

Yves Epelboin and Alain Soyer

Laboratoire Minéralogie-Cristallographie, associé CNRS,  
Universités Pierre et Marie Curie et Paris VII  
4 Place Jussieu, tour 16, 75230 PARIS CEDEX 05, France

Among the various X-ray topographic methods traverse topography is certainly the most widely used. In one single experiment it is possible to study a great volume of crystal and it is much easier to perform and to interpret than section topography. However in most cases it is not possible to obtain quantitative results. This method is mainly used to visualize defects. Sometimes, for planar defects or dislocations, the fault vector may be determined by searching for extinction conditions.

Authier<sup>1</sup> attempted to explain the traverse contrast as the integration, during the scanning of the crystal, of the direct image of the defect. Interpretation of the images based on this theory have been very helpful but cannot give a complete quantitative understanding (see II-6 in this volume). When a full characterization is needed the only means is to make section topographs. Unfortunately the images are difficult to analyze and the method is irrelevant for thin wafers. Besides, a small part of the crystal only may be studied in one experiment, losing one of the big advantages of traverse topography.

Since in most cases no analytical solutions exist, numerical methods must be used. A first test has been done by Epelboin<sup>2</sup>. It shows that the precision of the numerical algorithm is a crucial factor and the time of computation was too long. Petrashen et al<sup>3</sup> suggested using the reciprocity theorem (Kato<sup>4</sup>).

The introduction of fast specialized computers and the development of new algorithms to integrate the Takagi-Taupin (TT) equations now permits the simulation of traverse topographs. In this paper we will discuss the basic principles needed to compute

such simulations. We will then give a brief highlight of a program which has been written to simulate the image of a dislocation. In the last part, we shall give some examples and show how this program may help in understanding the contrast of dislocations and stacking fault.

## PRINCIPLES OF THE CALCULATION

### Theoretical considerations

A traverse topograph may be calculated simulating the real experiment i.e. moving a point source along the entrance surface and adding the intensities of all calculations. To compute the intensity at a given point P along the exit surface (Fig. 1), it is necessary to scan the source from B to A, thus :

$$I_h(P) = \int_{BA} I_h(\xi) d\xi \quad (1)$$

where  $\xi$  is the position of each incident spherical wave along the entrance surface.

Since most of the contrast originates from the interaction of the defects with the wavefields propagating near the  $s^0$  direction, the precision of the numerical algorithm in these areas<sup>0</sup> is very critical. Epelboin<sup>2</sup> has shown that a constant step algorithm (CSA) is not precise enough to compute equations (1). It is necessary to use a varying step algorithm (VSA)

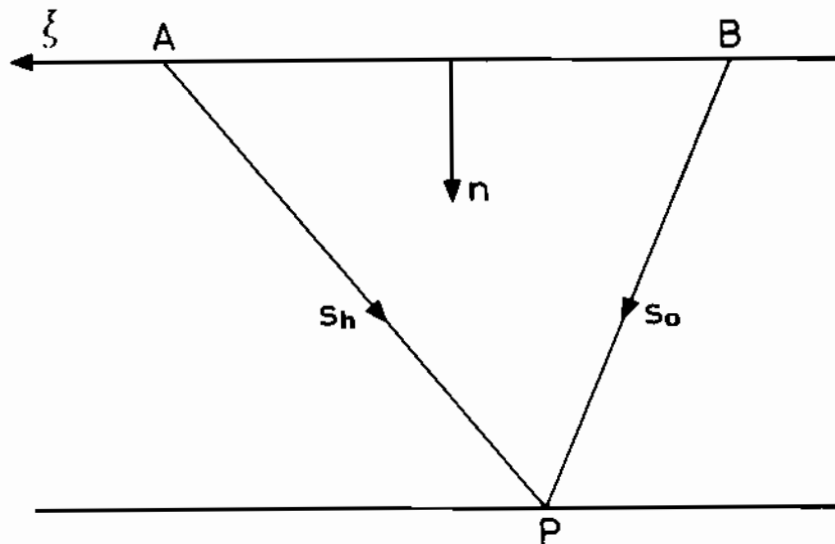


Fig. 1

The basic principles for numerical integration of TT equations have been explained elsewhere (Epelboin and Authier, II-6 in this volume). The calculation is performed along a set of characteristic lines parallel to the  $\vec{s}_o$  and  $\vec{s}_h$  directions.

The amplitudes of the waves at point A depend on the amplitudes of the waves at points B and C (Fig. 2).

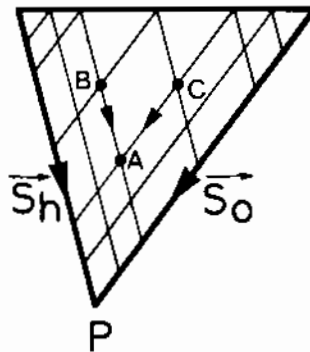


Fig. 2

Petrashen and al.<sup>3</sup> have suggested using the reciprocity theorem to speed up the calculation of equations (1).

In fact, the difference with a direct integration, as previous explained, is not significant. Moreover, the application of the reciprocity theorem to numerical integration seems to be rather delicate, so we have rejected Petrashen's method. Detailed explanations will be given elsewhere<sup>5</sup>.

Let us call p and q the elementary steps of integration AC and AB. The basic principles of VSA are the following :

- 1.- p and q are very small near the edges of the Borrmann fan to follow the fast oscillations of the waves.
- 2.- In order to save computation time, the steps of integration are increased in the middle of the Borrmann fan where the amplitudes of the waves vary slowly.
- 3.- In the direction of the direct image when a great intensity is scattered, the density of nodes, in the network of integration, is increased.

Fig. 3 is a schematic drawing of VSA as explained in details in (5).

It has been shown that the accuracy of the integration of TT equations is drastically improved, using such an algorithm. It also permits much faster calculations.

The computation of each image is very long. Using an IBM 370/168 computer, the calculation of the intensity for one single section topograph in one incident plane needs about 2 s for a silicon wafer  $400\mu\text{m}$  thick, 220 reflection. Let suppose that a resolution of  $1.6\mu\text{m}$  is satisfactory and that the image is  $320\mu\text{m}$  large ; it means that each line of the image is simulated in 600 s about. To speed up the computation we usually compute each third line in the image, the two missing ones being computed by linear interpolation. If the dimensions of the image are  $320 \times 320\mu\text{m}^2$  the complete image is simulated in 11 hours about. It is thus worthwhile to try to decrease this enormous time as much as possible. This may be achieved by the programming itself and also by the means of an array-processor.

#### Minimizing the number of elementary calculations

To compute an image of width  $ab$  (Fig. 4), the point source is scanned from  $O$  to  $O'$ . It is obvious that near the edges of the image a small part only of the nodes in the network of integration contributes to the calculation. For instance, when the point source is at  $O$ , it is sufficient to integrate the equations along  $Oa$  only. The computation may be restricted to the nodes contained in  $OO'ab$  only. Since the integration is performed along characteristic lines parallel to  $s_0$ , starting from the refracted direction, it is easy to introduce such limitations.

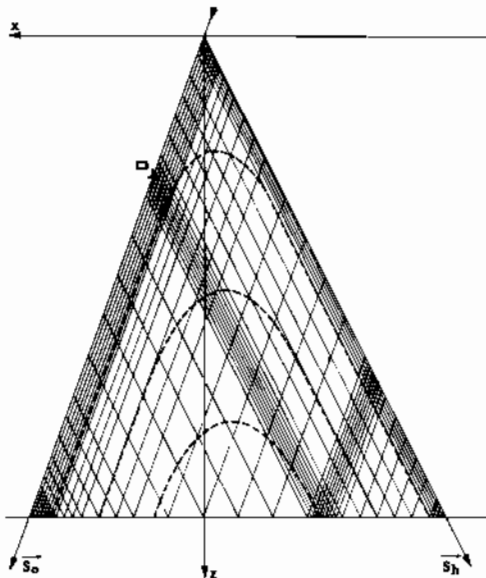


Fig. 3

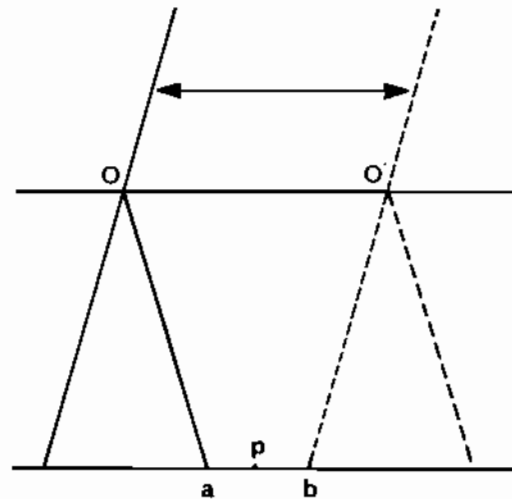


Fig. 4

### Automatic restart of the computation

5

In most computing centers it is not possible to continuously run a program for hours. It is thus necessary to include an automatic restart. This is done in the following manner.

A first program reads the initial data and creates save files which contains all data needed for the integration. A second program does the integration itself. After each integration, corresponding to one plane of incidence, the results and intermediary data are written in the save files. When the program starts it reads the save files and understands whether it is a new calculation or a previous one. In this case it starts at the point where it had been previously interrupted.

### Using an array-processor

An array-processor is a dedicated machine attached to a host computer. Thus one executes a program and on request asks the array-processor to perform specialized calculations. The host may wait till they are finished and continues its work. The array-processor may be started either by use of a specialized FORTRAN mathematical library or writing programs in its assembler language. It can be attached to a small computer and runs at speeds comparable to giant computers. This speed is achieved with a special hardware designed for scientific calculations.

For our purpose, the simulation program, ADELE, was first written in FORTRAN. Then the integration subroutines were rewritten in APAL, assembler language for FPS array-processors (Floating Point Systems Co.). All the integration is performed in the array-processor and the host writes the results in a file. Our tests have shown that, for a given image, a line is computed in 6 hours, using a small machine (Bull MINI6/53, comparable to a DEC 11/70) with the FORTRAN version, in 40 minutes using an IBM 370/168, in 4 minutes using a CRAY 1 and in only 13 minutes using a FPS AP100 array-processor.

The use of the array-processor permits to compute the images on a local small machine. It must be emphasized that, since most calculations are performed in the array-processor, the load for the host computer is negligible. It waits for results from the array-processor most of the time. The program can run day and night without influencing on the performances of the host computer as long as the AP100 is not needed by an other user. As explained, it is possible to stop on request and restart automatically.

## STUDY OF THE CONTRAST OF DISLOCATIONS

6

Fig. 5 is the traverse topograph of a straight dislocation in silicon (220 reflection MoK $\alpha$ ).

Fig. 6 shows the corresponding simulation. The agreement is quite satisfactory. Fringes appear in the left part of the image which are not visible in the experiment because the resolution of the simulation (0.8 $\mu\text{m}$ ) is much better than in the real experiment and because there is no noise in the background. The output device is a raster picture system PERICOLOR 2000. The main features are the same although the real dislocation is slightly curved. Notice a strong white contrast where the dislocation intersects the exit surface of the crystal.

Fig. 7 shows the simulation of the same dislocation except that the Burgers vector which has been reversed from  $1/2[10\bar{1}]$  to  $1/2[\bar{1}01]$ . The white contrast, near the exit surface vanishes and we may conclude that the topograph (Fig. 5) corresponds to the simulation in Fig. 6. This is in agreement with the simulation of section topographs.

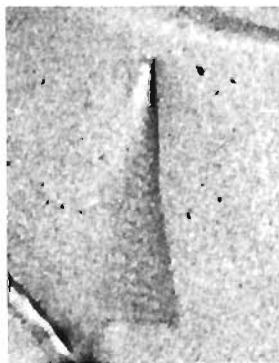


Fig. 5

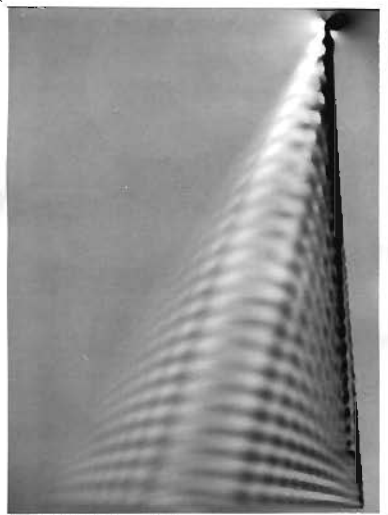


Fig. 6

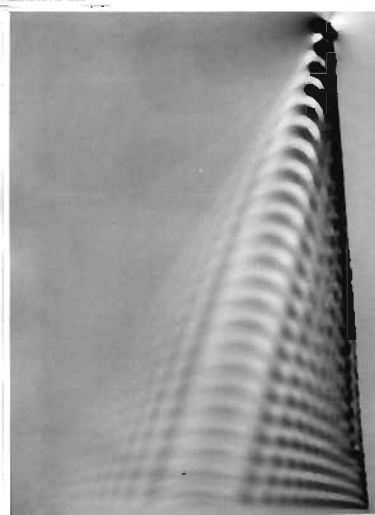


Fig. 7

Fig. 8 shows the images of rather straight dislocations in a silicon wafer (333 reflection  $\text{MoK}\alpha$ ) (courtesy M. Lefeld-Sosnowska). The angle between the surfaces of the crystal and the defect is rather small ; this explains why the contrast is poor : a black line for the direct image and a faint dynamical image which permits to distinguish the entrance surface from the exit surface. Fig. 9 is the corresponding simulation. We have also simulated the image with opposite Bürgers vector but no difference could be found. This is not surprising since there is very little information in this image.

Fig. 10 shows dislocations in KDP. These dislocations make an angle of 30 degrees about with the entrance surface of the crystal. The sense of their Bürgers vector has been found by surface effects<sup>6</sup> and by simulations of section topographs. Their Bürgers vector is  $[001]$  which is also the normale to the surface. Simulations of Figs 11 and 12 permit us to distinguish easily both cases : a first black fringe near the entrance surface corresponds to  $b = [001]$ . An additional black direct image appears in the simulation. It might be because of incorrect elastic stiffnesses values or to a relaxation in the core of the dislocations.

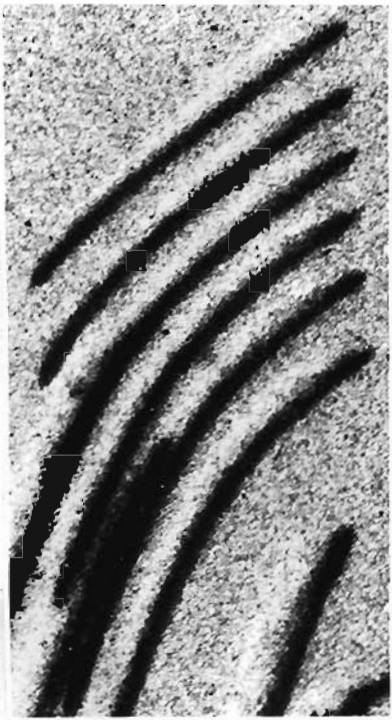


Fig. 8



Fig. 9



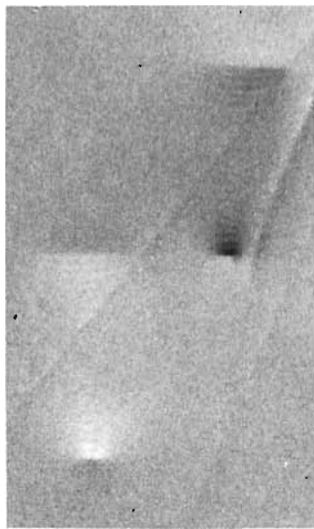


Fig. 10

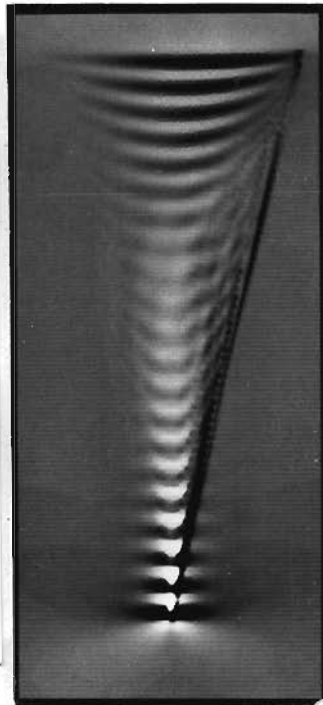


Fig. 11

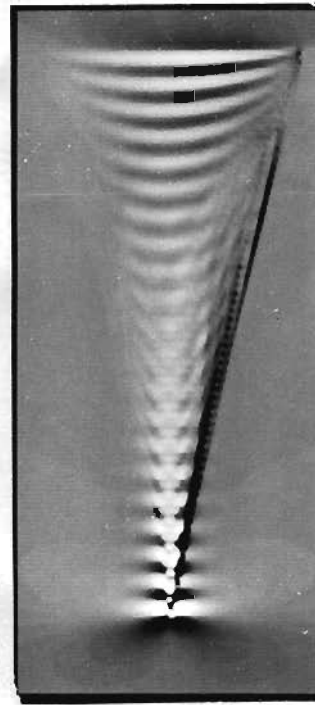


Fig. 12

These few examples show that quantitative measurements may be done using traverse topographs when the images contain enough information. The failure in Fig. 8 is due only to the lack of details in this image. Obviously this black line can just give an idea of the orientation of the defect.

It would be necessary to use an other reflection, where the image contains more information to be able to quantitatively characterize this defect.

#### CONTRAST OF PLANAR DEFECTS

Authier<sup>7</sup> has studied the contrast of stacking faults using a decomposition of the incident spherical wave in its plane wave components. The results are satisfactory but it is a first order approximation because he had to use the stationary phase method to integrate the propagation equations. Recently Patel<sup>8</sup>, studying the contrast of stacking faults in silicon (Fig. 13), found a disagreement between traverse topographs and the predictions made by Authier when the intersection of the fault with the surfaces is parallel to the incident plane. Capelle et al.<sup>9</sup> also notices, in section topographs, that a direct integration of TT equations using CSA matched better their experiments than Authier's theory especially along the direct image. This is not surprising because of the approximations.

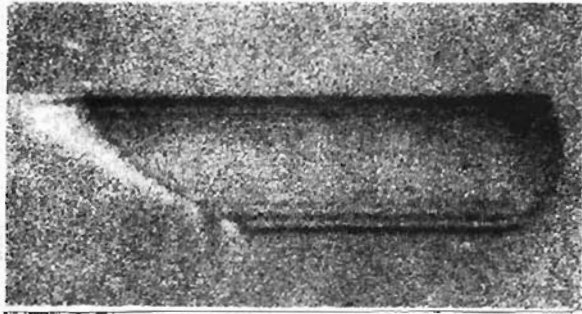


Fig. 13

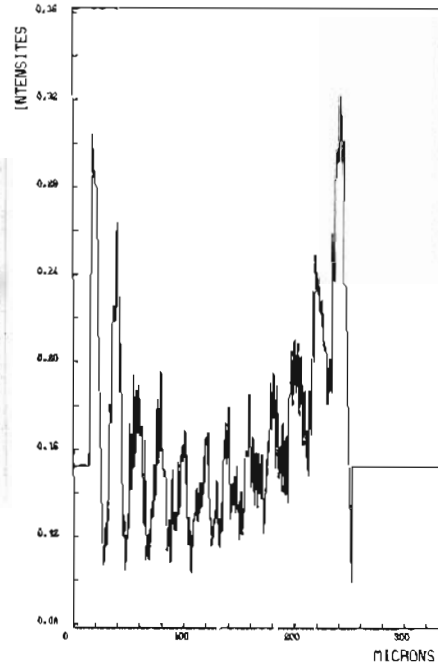


Fig. 14

The traverse topographs is much easier to simulate for planar defects than for dislocations. All the information is contained in one section topograph and the traverse is obtained by summing the intensities along lines parallel to the intersection of the fault with the exit surface. We have written a VSA program to simulate section topographs, EMPV, then we have computed a profile of intensity through the image of a stacking fault as shown in Fig. 13 (CuK $\alpha$  111 reflection). Fig. 14 is the computed intensity along a vertical line in the middle of the image. The agreement is good : a strong black fringe appears at both ends of the profile when Authier's theory predicted a white fringe near either the entrance or the exit surface (depending on the sign of the phase shift). The contrast in the middle is weaker and this also was not found in the previous theory.

#### CONCLUSION

The accuracy of VSA permits now to simulate traverse topographs. This can be applied to any kind of defect whenever a model for the deformation exists. Information such as the orientation and length of the Burgers vector of dislocations or the phase shift

of stacking fault can now be measured comparing the real and the simulated images. Simulations can be used in experimental work to quantitatively characterize defects. This is possible using either giant modern computers or array-processors linked to a small machine.

It should be especially useful when section topography cannot be used as, for instance, for thin crystals.

We are very indebted to F. Morris whose participation has been absolutely necessary when using the array-processor.

#### REFERENCES

1. A. Authier, Adv. X-Ray Anal. 10:9 (1967)
2. Y. Epelboin, Acta Cryst. A33:758 (1977)
3. P.V. Petrashen, F.N. Chukovskii and I.L. Shulpina, Acta Cryst. A36:287 (1980)
4. N. Kato, Acta Cryst. A24:157 (1968)
5. Y. Epelboin and A. Soyer, submitted to Acta Cryst.
6. E. Dunia, C. Malgrange and J.F. Petroff, Phil. Mag. A41:291 (1980)
7. A. Authier, Phys. Stat. Sol. 27:77 (1968)
8. J.R. Patel, Bell Laboratories Internal Report (1980)
9. B. Capelle, Y. Epelboin and C. Malgrange, J. Appl. Phys. 53:6767 (1982)