A METHOD FOR THE GENERATION OF VARIOUS GHOST CORRECTION ALGORITHMS—THE EXAMPLE OF THE POSITIVITY METHOD AND THE EXPONENTIAL METHOD

P. VAN HOUTTE

Department of Metallurgy and Materials Engineering, Katholieke Universiteit Leuven, Belgium

(Received 14 July 1990)

A theoretical strategy is presented that can derive the algorithms of several existing ghost correction methods. The examples of the positivity method and the "GHOST" method are elaborated. A new method is derived as well: the "exponential" method. It can successfully replace the quadratic method as a method that yields an exactly non-negative complete C.O.D.F. from pole figure data. The theoretical scheme that can generate all these algorithms makes use of the fact, that several parameter sets can be defined in order to describe a C.O.D.F. The parameters of one set are then functions of those of the other. The algorithms are derived from Taylor series expansions of these functions.

KEY WORDS Ghost correction, exponential method, positivity method, "GHOST" method.

INTRODUCTION

According to the series expansion method (Bunge, 1982), a crystallite orientation distribution function (C.O.D.F.) can be written as a series expansion of generalized spherical harmonics $T_{l}^{\mu \nu}(g)$:

$$f(g) = \sum_{l=0}^{L} \sum_{\mu=1}^{M(l)} \sum_{\nu=1}^{N(l)} C_{l}^{\mu \nu} T_{l}^{\mu \nu}(g)$$

(1)

$g$ represents the crystallite orientation. The series expansion takes even and odd values of $l$. $L$ is the maximum degree of the series expansion. It is finite in practical calculations. This sets a limit to the angular resolution of the series expansion. The $C_{l}^{\mu \nu}$ depend on the texture; if they are known, the C.O.D.F. is known.

By X-ray diffraction or neutron diffraction, it is possible to measure the pole density of certain \{hkl\} lattice planes as a function of the direction of the normals on these planes. This leads to a density function on an orientation sphere, which can be graphically represented in a stereographic projection called "pole figure." Such measured pole density functions (called "pole figures" hereafter) are always centro-symmetric, because of a property of X-ray and neutron diffraction. These pole figures too can be expressed as series expansions; their series expansion
coefficients are called $F'_{\gamma}$. These coefficients can in principle be obtained from measured pole figures. Linear relationships exist between these $F'_{\gamma}$-coefficients and the $C^\mu_{\nu}$-coefficients that describe the C.O.D.F. If one has collected the $F'_{\gamma}$-coefficients that correspond to a certain number of pole figures of the same sample, each associated to different \{hkl\} reflections, then a system of linear equations can be set up from which the $C^\mu_{\nu}$ can be solved up to a certain degree $L$ that is linked to the number of measured pole figures (Bunge, 1982). Unfortunately, the $F'_{\gamma}$-coefficients for odd values of $\gamma$ are zero because of the centro-symmetry mentioned above. This means in practice that the pole figures are independent of the $C^\mu_{\nu}$-coefficients of odd rank $\gamma$. Thus, the solution of the system of linear equations (called "classical pole figure inversion" hereafter) only leads to the values of the even $C^\mu_{\nu}$-coefficients, the odd ones remaining unknown.

There is one property that still allows to estimate these odd $C^\mu_{\nu}$-coefficients in case of sufficiently sharp textures, namely:

$$f(g) \geq 0$$  \hspace{1cm} (2)

Indeed, the C.O.D.F. is a statistical distribution function, so it must be non-negative. For sufficiently sharp textures, a C.O.D.F. as described by Eq. (1) would typically violate Eq. (2) for some orientations $g$ if $L$ is not infinite, even if both even and odd $C^\mu_{\nu}$-coefficients are known. This is because such C.O.D.F. must feature some ranges in orientation space where it is zero. When $L$ is not infinite, the series expansion of Eq. (1) will oscillate around zero in this range, leading to (slightly) negative values here and there.

Such errors are small compared to the errors that would be made by ignoring the $C^\mu_{\nu}$-coefficients of odd rank $\gamma$ while calculating the C.O.D.F. This means that these $C$-coefficients are given the value 0 in Eq. (1). Such series expansion is called a "reduced C.O.D.F." and may contain positive and negative "ghost peaks" that do not correspond to existing texture components. These peaks usually appear in those ranges of orientation space where the C.O.D.F. should be zero. Several methods to overcome this problem have been published, all making use of Eq. (2); see for example the overview by Wenk et al. (1988). Note that the problem as described above also exists (in a different mathematical form) in methods for pole figure inversion that do not use series expansions; such methods also use Eq. (2) to find physically sound solutions.

It is the purpose of this paper to present a fairly general method for constructing algorithms that can be used to develop ghost correcting procedures (i.e., methods based on Eq. (2) to estimate the odd $C^\mu_{\nu}$-coefficients). First it will be used to derive two similar methods: the "positivity method" proposed by Dahms and Bunge (1988) and the "GHOST" method used by Van Houtte (1983) to find an initial guess for starting the iteration of the quadratic method. Then a new method called "exponential method" will be described. As in the quadratic method (Van Houtte, 1983), it generates a texture function that is strictly non-negative. The even $C^\mu_{\nu}$-coefficients that are associated to this non-negative texture function should be as close to the even $C^\mu_{\nu}$ obtained from the classical pole figure inversion as possible.

From a practical point of view, the quadratic method and the exponential method lead to similar results. It is however much easier to write a computer code for the exponential method than for the quadratic method. The memory
requirements of the former are much smaller, and it needs only approximately one third of the calculation time of the latter.

CONSTRUCTION OF ALGORITHMS ON THE BASIS OF A TAYLOR SERIES

Parameter Sets that Describe a C.O.D.F.

In practical applications, the true C.O.D.F. \(f_{\text{true}}(g)\) is never exactly known, not even after the measurement of perfect pole figures followed by a perfect mathematical processing. Only an approximation \(f(g)\) which is represented by a finite number of parameters can be known. Examples of such parameter sets are:

1. The \(C_l^{\mu\nu}\)-coefficients of Eq. (1) for a finite value of \(L\).
2. The values of \(f_{\text{true}}(g)\) on a 5° grid in Euler (=orientation) space. Let these numerical values be \(f_k\), with \(k = 1, K\). \(K\) is the number of grid points that must be considered for a complete description of the C.O.D.F.; \(K\) will depend on crystal and sample symmetry. The \(f_k\) form a finite set of parameters; the estimation \(f(g)\) then is an interpolation between the \(f_k\) on the grid points that surround the orientation \(g\) in Euler space.

It is seen that several sets of parameters can be used to construct estimations of the same \(f_{\text{true}}(g)\). Let \(x_i (i = 1, n)\) be such a set, and \(y_j (j = 1, m)\) another set. It is then logical to assume that there exists a relationship between the two sets under the form of functions such as:

\[
y_j = Y_j(x_i) \tag{3}
\]

\[
x_i = X_i(y_j) \tag{4}
\]

Algorithm for the Estimation of One Parameter Set from Another

Assume that the \(x_i\) are known and the \(y_j\) not. There is however a first estimate \(y_{0j}\) of \(y_j\). Let us now try to develop an algorithm that should find a better estimate of \(y_j\) from \(x_i\) and \(y_{0j}\). Let \(x_{0i}\) be defined as follows:

\[
x_{0i} = X_i(y_{0j}) \tag{5}
\]

The functions \(Y_j(x_i)\) can be represented by Taylor series expansions:

\[
Y_j(x_i) = y_{0j} + \sum_{i=1}^{n} \frac{\partial Y_j(x_{0k})}{\partial x_i} (x_i - x_{0i}) + \ldots \tag{6}
\]

A new estimate of \(y_j\) can now be made using the linear term of the series expansion only:

\[
y_{1j} = y_{0j} + \sum_{i=1}^{n} \frac{\partial Y_j(x_{0k})}{\partial x_i} (x_i - x_{0i}) \tag{7}
\]

The old \(y_{0j}\) are now replaced by the new \(y_{1j}\). The algorithm can be executed again starting with Eq. (5).

Algorithms like this can only be used if the functions \(X_i(y_j)\) are known and if
the values of $\partial Y_i(x_{0k})/\partial x_i$ can be evaluated. Besides, the exact values of $x_i$ must in principle also be known. The method will now be illustrated for two types of ghost correction algorithms.

THE EXAMPLE OF THE POSITIVITY METHOD

Assume that one has analyzed measured pole figures in a classical way, and that the even $C_i^{\mu\nu}$-coefficients have been obtained. It is now attempted to find the complete C.O.D.F., as well as the values of the odd $C_i^{\mu\nu}$-coefficients. One does not wish to change the values of the even $C_i^{\mu\nu}$-coefficients; so these are treated as constants.

Choice of the Parameter Sets

The two parameter sets are chosen as follows:

- The $y_j$ are the odd $C_i^{\mu\nu}$-coefficients; $m$ is the total number of such coefficients for the degree $L$ of the series expansion;
- the $x_i$ are the values of the complete C.O.D.F. on a $5^\circ$ grid in Euler space; $n$ is the total number of such points for the crystal and sample symmetry at hand.

The function $X_i(y_j)$ is given by Eq. (1), in which the $C_i^{\mu\nu}$ with odd rank $l$ are the $y_j$ and those with even rank $l$ are constants. Let $\tilde{f}(g)$ be the reduced C.O.D.F. (Eq. (1) in which $l$ takes even values only). Equation (1) can now be rewritten as:

$$X_i(y_j) = \tilde{f}(g_i) + \sum_{l=1(2)}^L \sum_{\mu=1}^{M(l)} \sum_{\nu=1}^{N(l)} T_{i}^{\mu\nu}(g_i)y_j$$

in which $g_i$ are the crystal orientations at the grid points in Euler space. $j$ corresponds to a particular combination of $l$, $\mu$, $\nu$.

The following formula is given by Bunge (1982):

$$C_i^{\mu\nu} = (2l + 1) \int f(g) \tilde{T}_{i}^{\mu\nu}(g) \, dg$$

The integral can be evaluated in a discrete way:

$$C_i^{\mu\nu} = (2l + 1) \sum_{i=1}^{n} w_i f(g_i) \tilde{T}_{i}^{\mu\nu}(g_i)$$

in which $w_i$ are the weight factors suitable for the numerical integration. With the conventions for $x_i$ and $y_j$ made above, this can be written as follows:

$$Y_j(x_i) = (2l + 1) \sum_{i=1}^{n} w_i \tilde{T}_{i}^{\nu\mu}(g_i)x_i$$

from which the partial derivative that is needed in the algorithm is derived:

$$\frac{\partial Y_i}{\partial x_i} = (2l + 1) w_i \tilde{T}_{i}^{\nu\mu}(g_i)$$
The Algorithm

The initial estimate $y_{0j}$ of the odd $C''_{i}$-coefficients is zero. Equation (8) is used for calculating the initial value of $x_{0i}$ from $y_{0j}$. The partial derivatives can be calculated once and for all by Eq. (12), since they are independent of $x_i$ in this case.

The C.O.D.F. (= $x_{0i}$ values) that corresponds to this initial estimate is of course the reduced C.O.D.F. It will contain negative zones if the texture is sufficiently sharp. It is then assumed that in these negative ranges, the true value of the C.O.D.F. ($x_i$) is zero. Nothing is known really about the true value of the C.O.D.F. in the non-negative ranges. For lack of better knowledge, it is assumed that in these ranges, $x_i$ is also zero unless $x_{0i}$ is larger or equal than some non-negative threshold $\alpha$, in which case $x_i$ would be chosen equal to $x_{0i}$. $\alpha$ can of course be chosen zero. This leads to the following formulas for $x_i$:

\[
x_i = x_{0i} \quad \text{if} \quad x_{0i} \geq \alpha \quad \text{(13a)}
\]
\[
x_i = 0 \quad \text{if} \quad x_{0i} < \alpha \quad \text{(13b)}
\]

Let $Z_+$ be the range in Euler space for which Eq. (13a) holds, and $Z_-$ the range for which Eq. (13b) holds. Using Eq. (12), Eq. (7) can now be written as follows:

\[
y_{lj} = y_{0j} - (2l + 1) \sum_{g_{i} \in Z_{-}} w_{i} \hat{T}_{l}^{n}(g_{i}) x_{0i} \quad \text{(14)}
\]

For the next iteration step, one substitutes the values $y_{0j}$ by these $y_{lj}$ values. Equation (8) is then used to find the new values of $x_{0i}$. Equations (13a, b) are then used to find $x_i$ and $Z_-$ and $Z_+$. Finally Eq. (14) can be used again. The algorithm can be repeated as often as necessary. At each stage, $x_{0i}$ is the current estimate of the complete C.O.D.F. One could monitor the evolution of this estimate by some error criterion, for example the integral of minus this estimate in the range where it is negative. The algorithm can be stopped when such error criterion becomes smaller than a set value, or when the number of iterations reaches a set maximum.

Positivity Method

Equation (14) may be seen as a numerical estimate of the integral

\[
y_{lj} = y_{0j} - (2l + 1) \int_{g_{i} \in Z_{-}} f_{0}(g) \hat{T}_{l}^{n}(g) \, dg \quad \text{(15)}
\]

in which $f_{0}(g)$ is a continuous function obtained by interpolation between the $x_{0i}$ values at the nodal points. It can be seen as the initial estimation of the complete C.O.D.F. at the beginning of the iteration step. Equation (15) shows, that the $(y_{lj} - y_{0j})$ are the odd $C''_{i}$-coefficients of a texture function which is equal to $-f_{0}(g)$ in the range $Z_-$ and zero outside it. In the case that $\alpha$ is zero, such texture function precisely is the “correction term” $(n)\hat{T}(g)$ proposed by Dahms and Bunge (1988) (see Eqs. (11–12) in this paper) when they describe their “positivity method”. Because of this, the algorithm described here is at least theoretically identical to the algorithm of the positivity method in the case $\alpha = 0$. 
“GHOST” Method

It is possible to rewrite Eq. (14) by expressing the \( y_{0j} \) in the right hand side as functions of the \( x_{0i} \) using Eq. (11). In Eq. (11), a summation is made for \( i \) from 1 to \( n \), which means that the nodal points in Euler space are taken in the combined \( Z_+ \) and \( Z_- \) ranges. As a result, Eq. (14) becomes:

\[
y_{1j} = (2l + 1) \sum_{\nu \in Z_+} w_i \hat{y}_{l \nu}(g_i) x_{0i}
\]

The algorithm “GHOST”, used by the present author to find a first estimate of the odd part of the C.O.D.F. in order to start off the quadratic method, is in principle based on Eq. (16). The procedure has been described in Appendix 2 of the paper about the quadratic method (Van Houtte, 1983). The method starts with a rather high level of the threshold \( \alpha \) (70% of the maximum value of the \( x_{0i} \)) which guarantees that no influences due to ghost peaks affect the estimates \( y_{1j} \) of the odd C-coefficients. At the second iteration, the ghost peaks have already been reduced, which allows to use a lower value of \( \alpha \) without having to fear “ghost” influences in Eq. (16). The last iteration steps are performed with fairly low values of \( \alpha \).

Convergence is accelerated by not substituting \( y_{0j} \) merely by \( y_{1j} \) but by \( \beta y_{1j} \cdot x_{0j} \), the new estimate of the complete C.O.D.F. obtained from \( y_{0j} \) through Eq. (8), will be then a function of \( \beta \). The latter is chosen in order to minimize an error criterion. The error criterion is minus the integral of the complete C.O.D.F. over the range where it is negative.

In its basic form (without \( \beta \)), and for \( \alpha = 0 \), the GHOST method is mathematically equivalent to the positivity-method described above, since the two methods are based on equivalent mathematical expressions: Eq. (16) and Eq. (14).

THE EXAMPLE OF THE EXPONENTIAL METHOD

Both the positivity method and the GHOST method described above conserve the even \( C_l^{p^*} \)-coefficients and estimate values of the odd \( C_l^{p^*} \)-coefficients. The resulting estimate of the complete C.O.D.F. is then generated by a series expansion (Eq. (1)) in which the maximum degree of the series expansion \( L \) is finite. Even when there would be no experimental or other errors on the values of the C-coefficients, small negative values can be expected here and there in the range where the C.O.D.F. should be zero. In cases where the texture is too sharp to be represented by a series expansion up to \( L \), much larger truncation errors can be expected. These truncation errors not only include undesired negative overshoots, but also underestimates of the peak heights. In short it can be stated that the estimates of the complete C.O.D.F. are still no true statistical distribution functions which are completely free from negative values. For this reason, Van Houtte (1983) presented the “quadratic method”. In this method, the even \( C_l^{p^*} \)-coefficients are first estimated by a classical pole figure inversion. Then a first estimate of the complete C.O.D.F. is made using the GHOST method (see previous section). The negative values of this function are then artificially made zero; the resulting function is then re-normalized and used as a
first estimate of a strictly non-negative C.O.D.F. This estimate (and the estimates in the next iterations) has its own even and odd C-coefficients which are calculated using Eq. (9). The even coefficients of the estimate \( C_{\mu}^{\nu} \) should be equal to those that were obtained after classical pole figure inversion \( C_{\mu}^{\nu} \). This can be judged by the following error criterion:

\[
\Delta = \sum_{l=0}^{L} \sum_{\mu=1}^{M(l)} \sum_{\nu=1}^{N(l)} \frac{1}{2l+1} (C_{\mu}^{\nu} - C_{\mu}^{\nu})^2
\]

(17)

An algorithm is then used that tries to minimize \( \Delta \) by making better estimates of the non-negative complete C.O.D.F. In practice it is found that \( \Delta \) does not always tend to zero as the iteration goes on but often tends to a value such as \( 10^{-1} \) for materials with texture indices around 10. Much smaller values are frequently obtained for weak textures. One of the reasons why \( \Delta \) does not always tend to zero is that in many cases a non-negative texture function of which the even C-coefficients \( C_{\mu}^{\nu} \) are exactly equal to the \( C_{\mu}^{\nu} \) obtained after classical pole figure inversion does not really exist. This can be due to truncation effects or to errors in the coefficients \( C_{\mu}^{\nu} \). Such errors may be due to erroneous estimation of the normalization coefficients in cases where incomplete pole figures have been analyzed (see e.g. Van Houtte (1984)), or to inconsistencies between the measured pole figures, or to erroneous background correction of the measured pole figures. They may have the effect that recalculated pole figures that have not been measured, or parts of recalculated pole figures that correspond to the unmeasured parts of incompletely measured pole figures, feature excessive negative values. The problem of these excessive negative values in the recalculated pole figures can be overcome to use the even coefficients \( C_{\mu}^{\nu} \) (of the non-negative C.O.D.F. which results from the quadratic method) in order to recalculate the pole figures.

The "exponential method" which will be described below also generates a non-negative C.O.D.F. with a \( \Delta \)-value as small as possible. The algorithm is much simpler than that of the quadratic method, and requires less calculation time.

**Choice of the Parameter Sets**

Let the estimate of the complete C.O.D.F. be defined through another function \( h(g) \):

\[
f_s(g) = e^{h(g)}
\]

(18)

For any real value of \( h(g) \), we have

\[
f_s(g) \geq 0
\]

(19)

Let the parameters \( y_j \) be the values \( h(g_j) \) on the grid points \( g_j \) in Euler space; \( m \) is the total number of such grid points to be considered. Let the parameters \( x_i \) be the \( C_{\mu}^{\nu} \)-coefficients: of even rank \( l \) for \( i = 1, n_1 \) and of odd rank \( l \) for \( i = n_1 + 1, n \). The function \( X_i(y_j) \) is derived from Eq. (10) and from Eq. (18):

\[
X_i(y_j) = (2l+1) \sum_{j=1}^{m} w_j \hat{T}_{\mu}^{\nu}(g_j)e^{y_j}
\]

(20)
From Eq. (1) and Eq. (18) and the choice of \( x_i \) it is seen that

\[
e^y = \sum_{l=0}^{L} \sum_{\mu=1}^{M(l)} \sum_{\nu=1}^{N(l)} x_i \dot{I}_{l}^{\mu\nu}(g_i)
\]

in which \( i \) corresponds to a combination of \( l, \mu \) and \( \nu \). Differentiation with respect to \( x_i \) leads to

\[
e^y \frac{\partial Y_i}{\partial x_i} = \dot{I}_{l}^{\mu\nu}(g_i)
\]

from which it follows that:

\[
\frac{\partial Y_i(x_{0i})}{\partial x_i} = e^{-y_{0i}} \dot{I}_{l}^{\mu\nu}(g_i)
\]

**The Algorithm**

Let \( f_0(g) \) be an estimate of the complete C.O.D.F. obtained by the GHOST method.† It is used to calculate \( y_{0j} \), which is used later to make the initial estimate of \( y_{0j} \):

\[
y_{0j}' = \ln[f_0(g_j)] \quad \text{if} \quad f_0(g_j) > a \tag{24a}
y_{0j}' = f_0(g_j) - a + \ln(a) \quad \text{if} \quad f_0(g_j) \leq a \tag{24b}
\]

\( a \) has for example the value 0.1. Equation (24b) avoids that the occasional negative values that still exist in \( f_0(g) \) would make the evaluation of the right hand side of Eq. (24a) impossible.

\( y_{0j}' \) represents a non-normalized C.O.D.F. through Eq. (18). Renormalization is achieved by first calculating

\[
x_i' = \sum_{j=1}^{m} w_j e^{y_{0j}'}
\]

(which is the \( C_{0j}^{11} \)-coefficient of \( \exp(y_{0j}') \)) and then

\[
y_{0j} = y_{0j}' - \ln(x_i') \tag{26}
\]

Equation (20) is now used to calculate the values \( x_{0j} \). These values represent the \( C_{0}^{2l} \) of the current estimate of the non-negative C.O.D.F. They can be used to calculate the error criterion \( \Delta \) using Eq. (17).

It is now possible to write Eq. (7) for this case, using the expression given by Eq. (23) for the partial derivative:

\[
y_{ij} = y_{0j} + \sum_{i=1}^{n} e^{-y_{0j}} \dot{I}_{l}^{\mu\nu}(g_j)(x_i - x_{0i})
\]

In this expression,‡ the values \( C_{l}^{2l} \) of the even \( C \)-coefficients obtained after

---

† The positivity method would do just as well.
‡ While evaluating this expression in a computer code, some precaution should be taken against exponent underflow or overflow. One method is to set an artificial lower limit such as \(-23\) to \( y_{0j} \), and to replace the exponential \( \exp(-y_{0j}) \) by a function with a more moderate behaviour such as \( [0.1 + \exp(y_{0j})]^{-1} \).
classical pole figure inversion are used for $x_i$, $(i = 1, n_1)$, so these terms do not vanish in Eq. (27). For lack of better knowledge, the current estimates $x_{oi}$ of the odd $C$-coefficients are used for $x_i$, $(i = n_1 + 1, n)$. The corresponding terms in Eq. (27) vanish.

The algorithm is repeated by using the values $y_{lj}$ as $y_{oi}$-values and restarting the procedure from Eq. (25) on. The iteration can be stopped either when $\Delta$ becomes smaller than some preset value, or simply after a preset number of iterations. The final non-negative C.O.D.F. is obtained by means of Eq. (18); the last $y_{oj}$-values are equal to $h(g_j)$ on the grid points in Euler space.

Discussion of Some Results

In cases where the reduced C.O.D.F. has no negative values (weak textures), this method would find a solution which is equal to the reduced C.O.D.F. itself. For moderately sharp textures, the method works very well. Figure 1 shows the non-negative C.O.D.F. for such texture. Table 1 gives the texture index of that sample, as well as the $\Delta$-values of the first four iteration steps. Figure 2 gives the complete C.O.D.F. of a texture obtained by a Taylor-type simulation of the cold rolling process. It is a case for which the true texture is known. Four pole figures have been recalculated from this texture. They were processed as if they were incomplete experimental pole figures, measured on a grid of $5^\circ$ up to a maximal inclination of $80^\circ$. The classical pole figure inversion, using Method II described by Van Houtte (1984) leads to the reduced C.O.D.F. of Figure 3. Figure 4 gives the result of 4 iteration steps of the “exponential” method. Table 1 gives the texture index and the evolution of $\Delta$ for this case as well.

From a comparison of Figures 3 and 4, it may be concluded that the ghost peaks have disappeared. However, the texture of Figure 4 is less sharp than that of Figure 2. This is partly due to the fact, that the recalculated pole figures used

Table 1 Some results related to the C.O.D.F. represented by Figures 1, 2 and 4.

<table>
<thead>
<tr>
<th>C.O.D.F. represented by</th>
<th>Figure 1</th>
<th>Figure 2</th>
<th>Figure 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum degree of series expansion $L$</td>
<td>22</td>
<td>34</td>
<td>34</td>
</tr>
<tr>
<td>Texture index</td>
<td>2.33</td>
<td>8.70</td>
<td>7.13</td>
</tr>
<tr>
<td>Values of $\Delta$ (see Eq. (17)) for:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>iteration 1</td>
<td>$0.41 \times 10^{-2}$</td>
<td>—</td>
<td>$0.41 \times 10^{-1}$</td>
</tr>
<tr>
<td>iteration 2</td>
<td>$0.17 \times 10^{-2}$</td>
<td>—</td>
<td>$0.23 \times 10^{-1}$</td>
</tr>
<tr>
<td>iteration 3</td>
<td>$0.99 \times 10^{-3}$</td>
<td>—</td>
<td>$0.17 \times 10^{-1}$</td>
</tr>
<tr>
<td>iteration 4</td>
<td>$0.69 \times 10^{-3}$</td>
<td>—</td>
<td>$0.14 \times 10^{-1}$</td>
</tr>
</tbody>
</table>

† The outer 2 circles of the recalculated pole figures were deliberately not used for the pole figure inversion.

‡ In this method, the number of data points on the input pole figures is artificially increased by means of interpolation in order to decrease the step size down $1^\circ$ along the circles and $1.6667^\circ$ between the circles. This greatly increases the accuracy of the numerical integrations.
Figure 1 Non-negative complete C.O.D.F. of a cold rolled copper sheet. Moderate reduction. It is not possible to visually distinguish this Euler plot from the one that represents the reduced C.O.D.F.
Figure 2  Theoretically obtained complete C.O.D.F. of a simulated deformation texture. Odd and even C-coefficients are exactly known. F.C.C. metals, 68% rolling reduction.
Figure 3  Reduced C.O.D.F. obtained after classical pole figure inversion. Four of the recalculated pole figures of the C.O.D.F. of Figure 2 were treated as incompletely measured pole figures.
Figure 4 Non-negative complete C.O.D.F. obtained after using the exponential method as ghost correction procedure on the reduced C.O.D.F. of Figure 3. This result must be compared to the theoretical C.O.D.F. of Figure 2.
as input data were presented on a 5° grid. An additional test in which the recalculated pole figures were obtained on a much finer grid, and presented in that format to the pole figure inversion program, led to a result of which the texture index was 7.73, which is intermediate between that of Figure 2 and that of Figure 4 (Table 1). The remaining loss of sharpness is intrinsic of the "exponential method" in the case of sharp textures.

It is seen in Table 1 that $\Delta$ decreases to much lower values in the case of the moderately sharp texture (Figure 1) than in the case of the sharp texture (Figure 4). In this latter case, it seems to converge to a value that is larger than zero. This problem has already been discussed above. The $\Delta$-values obtained after 4 iterations by the exponential method are much better than those obtained by the quadratic method.

Attempts that used a random texture as initial guess for the iteration of the exponential method did not give satisfactory results. Apparently the rough work of the ghost correction task must be performed by an algorithm such as the GHOST or the positivity method.

The ghost correction procedure (transition from Figure 3 to Figure 4) took 30 minutes on a personal computer (AT-386 and 80387-coprocessor, memory with 1 wait state), for 4 iteration steps and with $L = 34$. This is 2.5 times longer than for $L = 22$. These durations are about three times shorter than those required for the quadratic method (Van Houtte, 1983).

No particular problems have been encountered during application of the exponential method on a large number of steel, copper and aluminium samples.

CONCLUDING REMARKS

A theoretical strategy has been developed that can derive the algorithms of several existing ghost correction methods. The examples of the positivity method (Dahms and Bunge, 1988) and the GHOST method (see appendix 2 in the paper by Van Houtte (1983)) have been presented. The study is not only of theoretical interest, since it has also been used to derive the algorithm of the exponential method which can find a complete C.O.D.F. which is exactly non-negative. The exponential method needs the result of the positivity or the GHOST method as first guess. When compared to the quadratic method (Van Houtte, 1983), the exponential method is more accurate, faster and incomparably easier to implement in a computer code.

Finally it is worthwhile to note that it only proved possible to apply the new scheme for algorithm generation thanks to the fact that the series expansion method (Bunge, 1982) offers convenient formulas for $X_i(y_j)$ and the partial derivatives of $Y_i(x_i)$ in all cases studied above.

References