Z. Kristallogr. Suppl. **26** (2007) 197-202

EFLECH/INDEX – Another try of whole pattern indexing

J. Bergmann

Ludwig-Renn-Allee 14, D-01217 Dresden, Germany
Contact author: E-mail: email@jbergmann.de

Keywords: XRPD, indexing, whole-pattern, full exhaustive

Abstract. EFLECH/INDEX is a program pair for a new try of whole-pattern, full-exhaustive indexing of XPD patterns. EFLECH scans the pattern for peaks and gives additionally the full error information for all the peak parameters to INDEX. Therefrom, INDEX generates intensity and position weights of the peaks. INDEX does a random, full exhaustive search. 200000 trials per second are good trial rates. EFLECH/INDEX are checked against 17 real world laboratory patterns.

Introduction

In the past, a number of indexing algorithms were developed. For a summary see [1] and [2]. All these programs are searching for some significant combinations of peaks and try to reconstruct the lattice/reciprocal lattice from such combinations. Some more recent programs do a extended search ([3]: exhaustive, [4]: genetic algorithm). Until now, all exhaustive programs only count lines within/without some “acceptance windows”, which are set around the observed line positions. But, a full exhaustive search should take into account the different errors of all the lines (depending from intensity, line overlap etc.) and compute not only within/without counts, but weighted sums from all the lines residual errors. EFLECH/INDEX was developed for doing such a full exhaustive search.

Principles of EFLECH/INDEX

INDEX was designed to be an automatic trial-and error program based on a previous whole powder pattern peak fitting procedure. It is unable to run INDEX on common peak lists. The indexing algorithm needs the error information of the peak parameters. Therefore, a prior EFLECH run is necessary. EFLECH is using the ray-traced/learnt profile shape modelling of BGMN [5]. The program performs a whole pattern fitting and delivers a list of peak profile parameters plus their covariance matrix.
INDEX is using this information to estimate weights and acceptance widths for each line. In difference to [2], lines are not only counted as inside/outside the acceptance width, a continuous figure of merit will be used instead:

$$Q_{\text{partial}} = \sum_i \min\left(w_i, \left(\frac{x_i - \hat{x}_i}{\delta_i}\right)^2\right) \quad (1)$$

where w_i is the observed weight of line i, x_i , $\hat{x}_i$ are the corresponding observed and simulated line positions and δ_i the observed random error of line i. This partial figure of merit will then be multiplied by a reliability factor depending from general multiplicity of the simulated lattice (cubic. . . triclinic), from the unit cell volumina of the simulated lattice, from the real number of ignored peaks and, of course, from all the individual line's w_i, δ_i and x_i. Estimating this reliability factor needs to solve integrals for all the peaks and would slow down the search by several magnitudes if done for each trial. Therefore, during the first few seconds of the INDEX run, there are built interpolation tables depending from unit cell volumina and ignore peak number. Such, the total figure of merit Q considers a large amount of statistical information originating from the raw pattern. This is what "full exhaustive search" means.
Now, INDEX does a Monte Carlo search using random lattice parameters. Of course, testing all random lattice parameters for the final best solution will be too time-consuming. In principle, INDEX does a series of branch and bound searches. At a first level, a set of lattices with identical α, β, γ and identical $a : b : c$ ratios will be tested against a low-angle part of all lines. The number of these low-angle lines will be estimated together with the above mentioned interpolation tables at INDEX start-up, it will then be reported in the result listing. Each the best solution of this first level branch and bound will then refined by an incomplete regression algorithm, which will minimize Q, again using the above low-angle part of lines. The incompleteness consists in allowing only a small number of refinement steps, in general no final solution will be found. Another branch and bound will select a minor part of the first type regression results, these results undergo a second type, still incomplete regression over all lines. After another branch and bound, a third complete regression is done. A last branch and bound will decide if this solution will go into the result table.

EFLECH/INDEX details

In fact there is a third program called TEIL which must be run first in order to prepare the peak search by EFLECH. This is a very simple one: It looks for sufficient large "background areas" and defines some virtual parts of the pattern by writing their angular limits into the control file. EFLECH reads this information and, part by part, does an automatic search for peaks. Starting from scratch (if no starting parameters given starting from an empty pattern, this should be the default case), EFLECH performs an automated series of trial-and-error steps, on each step the count/composition of parameters will be altered as follows:

N adds a new peak,
+ enlarges the parameter count of a single peak,
– diminishes the parameter count of a single peak,
D deletes a peak,
U reduces the background polynomial by one degree.

These alterations are done automatically, on each step a significance analysis for all possible alterations is done. In following, the positive tested alterations are tried in order of decreasing significance, the pattern will be refined for the altered peak/parameter composition. This will done until a non-degenerated solution is found (the solution may degenerate e.g. by shifting peaks outside the angular area or setting a peak intensity to zero), the program will undo each degenerated solution. Concerning the + – operators: Peaks may have 2, 3 or 4 parameters. These are:

2 intensity and position,
3 additional Lorentzian width (often related to size values),
4 additional Gaussian-like width (often related to micro-stress).

The output files (one by each angular part) contain the whole error matrix at the end of the file, not only the "significant elements". INDEX uses this information (peak intensity, position and their error information) for calculation of w_i, x_i and δ_i in expression (1). For the optimizations done to speed up INDEX, see 'Principles of EFLECH/INDEX". Of course, the scan rate depends from the data. Good rates are 200000 indexing trials per second (monoclinic epidote, scanning in the lattice constant space for cubic to monoclinic) and 6000 indexing trials per second (common value for all triclinic samples, scanning in the index space).
During the development, the need for a "ignore peaks" feature became clear. Some patterns contain some spurious lines of other (sometimes unknown) phases. So INDEX was changed to deal with such phases, see 'Principles'. Ignored peaks may be restricted by number and by "total content". If nothing set manually, INDEX selects some useful values automatically (see figure 1, too) during the start-up-time, together with the number of low angle lines for the first type, incomplete regression.
As common, INDEX gives not only one solution, but a list of possible solutions from low unit cell volume/high total figure of merit Q to high cell volume/low total figure of merit Q. If there is a significant solution, it may pick off simply by observing this list.

INDEX test suite

Beginning with the development of EFLECH/INDEX, the author has requested R. Kleeberg for supporting some real-world mineral patterns. This are neither narrow lines nor high accuracy data, and due to the origin date (winter 1997/98) the peak search was performed without known tube tails (see [6]) in a "routine" quality. These 17 examples are listed in table 1 together with their respective calculation times.
Figure 1 shows a section from the test runs of gypsum, $K_2Cr_2O_7$, cassiterite. The three examples show some interesting details: a.) *Gypsum* contains some spurious peaks of halite impurity. Note the single 0.1% ignored peak in the right solution. b.) *$K_2Cr_2O_7$* is the triclinic example as distributed with the program on http://www.bgmn.de/related_download.html. c.) *Cassiterite* has an interesting structural detail: The heavy Sn atoms form a body-centered tetragonal lattice, and only the O atoms do break that symmetry. So, you will see the wrong centered solution with only one ignored peak and an reliability factor slightly above that of the right uncentered solution. This is due to the additional "ignore penalty" to the reliability factor.

```
**************************************************************************
GIPS
**************************************************************************
BGMN and related programs Copyright (C) J. Bergmann Dresden 1991-2006
BGMN is a registered trademark of J. Bergmann
Version 3.5.12
Developers version for internal use only
Cubic
+++ using 33 pre-scan peaks, WMAX=55.24, ignore=4/7.0%
+----+----+----+----+----+----+----+-----+

--- skipped because of its length ---

Monoclinic
+++ using 23 pre-scan peaks, WMAX=46.52, ignore=3/3.0%
+----+----+----+----+----+----+----+-----+
Q        V       ignore  A      B      C      ALPHA   BETA    GAMMA
 19.34%   0.101 P  2/3.0% 0.2561 0.5191 0.7617  90.000  92.397  90.000
 13.82%   0.103 P  2/3.0% 0.2604 0.5185 0.7621  90.000  94.016  90.000
 12.31%   0.108 P  1/1.4% 0.2747 0.5186 0.7675  90.000  98.103  90.000
  6.90%   0.124 P  1/1.4% 0.3143 0.7605 0.5251  90.000  99.033  90.000
  0.60%   0.496 A  1/0.1% 0.5680 1.5210 0.6286  90.000 114.088  90.000
  0.58%   1.487 A  1/0.1% 1.0044 1.5207 1.0499  90.000 112.034  90.000
default setting (the common has BASIS=1):
A=0.5680
B=1.5211
C=0.6286
BETA=114.098
BASIS=A
**************************************************************************
K2Cr2O7
**************************************************************************
BGMN and related programs Copyright (C) J. Bergmann Dresden 1991-2006
BGMN is a registered trademark of J. Bergmann
Version 3.5.12
Developers version for internal use only
Cubic
+++ using 57 pre-scan peaks, WMAX=55.28, ignore=2/7.0%
+----+----+----+----+----+----+----+-----+
Hexagonal
+++ using 57 pre-scan peaks, WMAX=55.28, ignore=1/5.0%
+----+----+----+----+----+----+----+-----+
Tetragonal
+++ using 57 pre-scan peaks, WMAX=55.28, ignore=1/5.0%
+----+----+----+----+----+----+----+-----+
Orthorhombic
+++ using 57 pre-scan peaks, WMAX=55.28, ignore=1/4.0%
+----+----+----+----+----+----+----+-----+
Q        V       ignore  A      B      C      ALPHA   BETA    GAMMA
 29.70%   1.470 P  0/0.0% 0.1723 2.0895 4.0829  90.000  90.000  90.000
Monoclinic
+++ using 37 pre-scan peaks, WMAX=38.51, ignore=0/0.0%
+----+----+----+----+----+----+----+-----+
Q        V       A      B      C      ALPHA   BETA    GAMMA
 26.31%   1.392 P 1.9818 0.3478 2.0613  90.000 101.457  90.000
Triclinic
+++ using 29 pre-scan peaks, WMAX=33.96, ignore=0/0.0%
+----+----+----+----+----+----+----+-----+
Q        V       A      B      C      ALPHA   BETA    GAMMA
 27.63%   0.602 P 0.7142 0.8525 1.0580  81.477  71.217  84.204
  7.68%   0.729 P 0.7394 0.7472 1.3408  96.243  98.065  90.845
A=0.7400 B=0.7520 C=1.3400
ALPHA= 96.2 BETA=98.0 GAMMA=90.8
**************************************************************************
KASSIT
**************************************************************************
BGMN and related programs Copyright (C) J. Bergmann Dresden 1991-2006
BGMN is a registered trademark of J. Bergmann
Version 3.5.12
Developers version for internal use only
Cubic
+++ using 16 pre-scan peaks, WMAX=66.17, ignore=2/7.0%
+----+----+----+----+----+----+----+-----+
Hexagonal
+++ using 16 pre-scan peaks, WMAX=66.17, ignore=2/5.0%
+----+----+----+----+----+----+----+-----+
Q        V       ignore  A      B      C      ALPHA   BETA    GAMMA
 21.97%   0.886 R  0/0.0% 0.8203 0.8203 1.5202  90.000  90.000 120.000
 18.46%   0.386 P  0/0.0% 0.3851 0.3851 3.0074  90.000  90.000 120.000
 17.91%   0.392 P  0/0.0% 0.3875 0.3875 3.0178  90.000  90.000 120.000
 13.87%   0.399 P  0/0.0% 0.3892 0.3892 3.0401  90.000  90.000 120.000
Tetragonal
+++ using 16 pre-scan peaks, WMAX=66.17, ignore=2/5.0%
+----+----+----+----+----+----+----+-----+
Q        V       ignore  A      B      C      ALPHA   BETA    GAMMA
 18.66%   0.030 P  0/0.0% 0.3353 0.3353 0.2646  90.000  90.000  90.000
  1.74%   0.072 I  1/0.3% 0.4745 0.4745 0.3190  90.000  90.000  90.000
  1.05%   0.072 P  0/0.0% 0.4744 0.4744 0.3190  90.000  90.000  90.000
A=0.4738
C=0.3187
```

Figure 1: Exemplary run of the INDEX program on 3 of the 17 real world patterns.

Table 1: 17 examples of the INDEX test suite plus their respective computation times

sample	*lattice*	*computation time*
anhydrite	orthorhombic	28min
corundum	hexagonal	10s
epidote	monoclinic	2h 7min
fluorite	cubic	2s
gypsum	monoclinic	3h 13min
$K_2Cr_2O_7$	triclinic	12h 44min
cassiterite	tetragonal	13s
kyanite	triclinic	6h 55min
magnetite	cubic	2s
muscovite	monoclinic	1h 3min
plagioclase	triclinic	11h 24min
quartz	hexagonal	13s
$PbSO_4$	orthorhombic	1min 48s
wollastonite	triclinic	5h 39min
Y_2O_3	cubic	4s
zincite	hexagonal	9s
zircon	tetragonal	23s

Status of the development

EFLECH/INDEX are at beta stage until now. You may download the programs for free at http://www.bgmn.de/related download.html, a short description of the usage of the programs is given under http://www.bgmn.de/related.html. Actually there is no intention to make the program for sale. In equivalence to the commercial BGMN package as mainly distributed on http://www.bgmn.de, EFLECH/INDEX come in versions for Winxx, OS/2 and Linux. But, in difference to BGMN, EFLECH/INDEX

- do not have any graphical user interface similar to BGMNwin for BGMN. They are pure command line programs.
- are not supported by an extended manual at now.
- are not delivered with any installer, even not for Winxx!

The development is done using a minor part of the authors past time, so the progress is only minor by time (the development reaches back to 1995). For that reason, EFLECH/INDEX are called beta until now.

Conclusion

The program pair EFLECH/INDEX is some of the first kind of full exhaustive indexing program. The performance was demonstrated by delivering meaningful indexing solutions in an indexing round robin, see [1]. The programs are useful mainly for people with computer skills, knowing how to use a command line program. But, under such circumstances, one may check EFLECH/INDEX for solving an indexing problem. If you have questions/ problems while testing/using EFLECH/INDEX, do not hesitate to contact the author on email@jbergmann.de.

References

1. Bergmann, J., Le Bail, A.,Shirley, R. & Zlokazov, V., 2004, *Z. Kristallogr*., **219**, 783.
2. Shirley, R., 1980, *NBS Spec. Publ*., **567**, 361.
3. Werner, P.E., Erikson, L. & Westdahl, M., 1985, *J. Appl. Cryst*., **18**, 367.
4. Kariuki, B.M., Belmonte, S.A., McMahon, M.I., Johnston, R.L., Harris, K.D.M. & Nelmes, R.J., 1999*, J. Synchrotron Rad*., **6** , 87.
5. Bergmann, J., Friedel, P. & Kleeberg, R., 1998, *CPD Newsletter*, **20**, 5.
6. Bergmann, J., Kleeberg, R., Haase, A. & Breidenstein, B., 2000, *Materials Science Forum*, **347–349**, 303.

Acknowledgements. The author thanks R. Kleeberg from the TU Bergakademie Freiberg for supporting the 17 exemplary patterns.