1 Introduction

This document is the first go at writing some documentation for the FitAllB program, an add-on to the FABLE suite of programs for analysis of 3DXRD data (http://fable.wiki.sourceforge.net/). FitAllB is tailored to do centre-of-mass (COM) refinements of grain orientations, positions and strain tensors from far-field images of a polycrystalline material. The aim is to be able to handle several hundred illuminated grains and obtain the strain tensors to an accuracy of 10^{-4}. The strain tensors are output both in the Cartesian grain coordinate system relative to the grain orientation and in the sample system for overall comparisons, and if the components of the stiffness tensor **C** are provided the stress tensors in the same two representations will also be output. FitAllB includes an error propagation to give standard deviations of all refined parameters. Finally the relative volumes of the grains are refined, so in principle a 3D orientation and stress/strain map of the polycrystal can be obtained using tessellation.

2 Mathematical equations

2.1 Basic equations

We aim to minimise the following function using MINUIT (James, CERN):

$$FCN = \sum_{i,j(i)} \left| \overline{G}_{ij} - \frac{\lambda}{2\pi} \Gamma_{ij} U_i B_i \overline{G}_{hkl,ij} \right|^2 , \tag{1}$$

i.e. the deviation from the ideal diffraction equation. The sum is taken over i grains with $j(i)$ reflections in the i'th grain. The unrotated scatter vector in reciprocal space, $\overline{G}_{ij}$, can be expressed in terms of direct space quantities:

$$\overline{G}_{ij} = \left[\frac{\bar{d}_{ij}}{|\bar{d}_{ij}|} - \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \right] , \tag{2}$$

$$\bar{d}_{ij} = R \begin{pmatrix} 0 \\ y_{d,ij} \\ z_{d,ij} \end{pmatrix} + \begin{pmatrix} D \\ 0 \\ 0 \end{pmatrix} - \Gamma_{ij} \begin{pmatrix} x_{0,i} \\ y_{0,i} \\ z_{0,i} \end{pmatrix} , \tag{3}$$

$$y_{d,ij} = (y_{det,ij} - y_{det,0}) p_y, z_{d,ij} = (z_{det,ij} - z_{det,0}) p_z . \tag{4}$$

$$\Gamma_{ij} = \begin{pmatrix} 1 - 2q_2^2 - 2q_3^2 & 2q_1q_2 - 2q_3q_0 & 2q_1q_3 + 2q_2q_0 \\ 2q_1q_2 + 2q_3q_0 & 1 - 2q_1^2 - 2q_3^2 & 2q_2q_3 - 2q_1q_0 \\ 2q_1q_3 - 2q_2q_0 & 2q_2q_3 + 2q_1q_0 & 1 - 2q_1^2 - 2q_2^2 \end{pmatrix}_{ij} \tag{5}$$

$$q_{ij} = \left(\cos \frac{\omega_{ij}}{2} | \Phi_x \Phi_y \left(\begin{array}{c} 0 \\ 0 \\ \sin \frac{\omega_{ij}}{2} \end{array} \right) \right) \tag{6}$$

Refineable are 12 parameters per grain (3 positional, 3 rotational and 6 strain) and the 10 global parameters: tilt (R, 3 parameters), sample-to-detector distance (D), pixels sizes (p_y and p_z), beam centre on detector ($y_{det,0}$, $z_{det,0}$), and the tilt of the ω-axis around x and y.

The measured quantities are ω_{ij}, $y_{det,ij}$ and $z_{det,ij}$ and standard deviations of these: $\sigma(\omega_{ij})$, $\sigma(y_{det,ij})$, and $\sigma(y_{det,ij})$.

2.2 Error propagation

To include the propagated errors of ω_{ij}, $y_{det,ij}$ and $z_{det,ij}$ it is convenient to rewrite (1) as:

$$FCN = \sum_{i,j(i)} \left(\Gamma_{ij}^{-1} \overline{G}_{ij} - \frac{\lambda}{2\pi} U_i B_i \overline{G}_{hkl,ij} \right)^T V_{ij}^{-1} \left(\Gamma_{ij}^{-1} \overline{G}_{ij} - \frac{\lambda}{2\pi} U_i B_i \overline{G}_{hkl,ij} \right) \tag{7}$$

where V_{ij} is the covariance matrix of $\Gamma_{ij}^{-1}\overline{G}_{ij}$. Thus the problem boils down to determining V_{ij} from the components of the variance-covariance matrix of the measured quantities: $S_{\omega\omega,ij}$, $S_{yy,ij}$, $S_{zz,ij}$, $S_{yx,ij}$, $S_{\omega y,ij}$, and $S_{\omega z,ij}$. Assuming that the covariance matrix is diagonal, the elements of V_{ij} can be determined as the error propagated variance of $\Gamma_{ij}^{-1}\overline{G}_{ij}$:

$$\left(\begin{array}{c} V_{ij}^{11} \\ V_{ij}^{22} \\ V_{ij}^{33} \end{array} \right) = \left(\frac{d\Gamma_{ij}^{-1}\overline{G}_{ij}}{dy_{det,ij}} \right)^2 S_{yy,ij} + \left(\frac{d\Gamma_{ij}^{-1}\overline{G}_{ij}}{dz_{det,ij}} \right)^2 S_{zz,ij} + \left(\frac{d\Gamma_{ij}^{-1}\overline{G}_{ij}}{d\omega} \right)^2 S_{\omega\omega,ij} \tag{8}$$

The derivatives in (8) are calculated analytically from the equations in Section 2.1 before every refinement step. The most correct thing to do would of course be to update the values in each cycle of the refinement, but this is simply too time consuming. Note that the covariance terms have been omitted. It could be argued to add $S_{yz,ij}$, but the contribution is an order of magnitude smaller than that from $S_{yy,ij}$ and $S_{zz,ij}$ which again is an order of magnitude less than the contribution from $S_{\omega\omega,ij}$.

2.3 Determining experimental errors

The question is now how to obtain $S_{yy,ij}$, $S_{zz,ij}$ and $S_{\omega\omega,ij}$ from the experiments assuming that the spread of the intensity I_k in pixel k is $\sigma(I_k) = \alpha\sqrt{I_k}$, thus

$S_{I_k I_k} = \alpha^2 I_k$. Here it is probably convenient to set $\alpha = c\sqrt{I_{avg}}$, where I_{avg} is the intensity averaged over all peaks and c is some constant tuned to give a good estimation of the errors on the fitted parameters from simulated data.

2.3.1 $S_{\omega\omega,ij}$

ω_{ij} is obtained as a intensity weighted sum over the contributing frames k:

$$\omega_{ij} = \frac{\sum_k \omega_{ij,k} I_{ij,k}}{\sum_k I_{ij,k}} \tag{9}$$

Deriving this equation with respect to $I_{ij,k}$ and remembering that $I_{ij} = \sum_k I_{ij,k}$ one obtains:

$$\frac{d\omega_{ij}}{dI_{ij,k}} = \frac{I_{ij}\omega_{ij,k} - \sum_k \omega_{ij,k} I_{ij,k}}{I_{ij}^2} = \frac{\omega_{ij,k} - \omega_{ij}}{I_{ij}} \tag{10}$$

If $k = 1$ (the spot is only observed in a single frame) then:

$$S_{\omega\omega,ij} = \frac{(\Delta\omega)^2}{12} \tag{11}$$

where $\Delta\omega$ is the step size in ω. Otherwise $S_{\omega\omega,ij}$ is obtained using error propagation over the contributing frames:

$$S_{\omega\omega,ij} = \sum_k \left(\frac{d\omega_{ij}}{dI_{ij,k}}\right)^2 S_{I_{ij,k} I_{ij,k}} = \frac{\alpha^2}{I_{ij}^2} \sum_k I_{ij,k}(\omega_{ij,k} - \omega_{ij})^2 = \frac{\alpha^2}{I_{ij}} \sigma^2(\omega_{ij}) \tag{12}$$

2.3.2 $S_{yy,ij}$ and $S_{zz,ij}$

In the same way as above it can be derived that

$$S_{yy,ij} = \frac{\alpha^2}{I_{ij}} \sigma^2(y_{det,ij}) \tag{13}$$

and

$$S_{zz,ij} = \frac{\alpha^2}{I_{ij}} \sigma^2(z_{det,ij}) \tag{14}$$

in good accordance with for instance Withers et al. (2001), J. Appl. Cryst. 34: 737–743. In case of a single pixel peak the values are simply set to $S_{yy,ij} = 1$ and $S_{zz,ij} = 1$

3 Algorithm

```
Read files and set defaults
Initialise and perform substantial outlier rejection
(intensity, residual (all grains, user defined reslimit), merge, multi)
If refinement of global parameters requested
    Refine global parameters and reject outliers based on residual (start)
Refine Euler angles for each grain and reject outliers based on residual (euler)
Refine COM grain positions for each grain and reject outliers based on residual (xyz)
Refine strain tensors for each grain and reject outliers based on residual (eps)
If refinement of global parameters requested
    Refine global parameters and reject outliers based on residual (start0)
Refine Euler angles, COM positions and strain tensors for each grain (grain)
Reject outliers based on residual
While more outliers are being rejected
    If refinement of global parameters requested
        Refine global parameters and reject outliers based on residual (start1)
    For all grains from which outliers have just been rejected
        Refine Euler angles, COM positions and strain tensors for each grain (final)
        Reject outliers based on residual
        For the poor grains
            Perform substantial outlier rejection
            (residual (only poor, reslimit=1), intensity, merge, multi)
```

4 Input

file	**origin**	**contents**
grainspotter.log	GrainSpotter	number of grains, for each grain: **U**, and for all assigned peaks: h, k, l, 2θ, η, ω, peak_id
peaks_tXXXX.flt	peaksearch	filtered peaks, for each peak: peak_id, y_{det}, z_{det}, ω, σ_y, σ_y, σ_ω, I
detector.par	ImageD11	experimental parameters: unstrained unit cell, wavelength, tilt, sample-to-detector distance etc.
fitallb.inp	in time: GUI	name can be anything, see example below

4.1 Example of FitAllB input file

```
##### Mandatory input files
log_file S1_e0_050_grainspotter.0.35.log
flt_file S1_e0_050.flt
par_file frelon_fitted_july08.par
##### Additional file to be read for volume based outlier rejection
structure_file grethe.cif
##### on/off possibilities for fitting
w 0            # Fit omega stage tilt parameters wx and wy (1=yes/0=no)
tilt 0         # Fit detector tilt parameters tx, ty, tx (1=yes/0=no)
pixel 0        # Fit pixel size py and pz (1=yes/0=no)
center 0       # Fit detector centre (1=yes/0=no)
L 0            # Fit sample-to-detector distance (1=yes/0=no)
euler 1        # Fit orientations and thus Euler angles (1=yes/0=no)
xyz 1          # Fit cms positions (1=yes/0=no)
eps 1          # Fit B matrix and thus strain tensors (1=yes/0=no)
hesse 0        # Error estimation using full Hessian (1=yes/0=no)
##### Mandatory parameters
dety_size 2048 # detector dimension in pixels
detz_size 2048 # detector dimension in pixels
w_step 0.5     # step size in omega
w_limit -22.5 22.5 67.5 112.5   # limits of omega intervals
##### Possible parameters (ovelap, limit and tolerance defaults)
limit 5 10     # outlier rejection limits in terms of average peak contribution
overlap 0.5    # grain overlap tolerance
skip 31 45     # skip grains 31 and 45 in the GrainSpotter log_file
tol_start 1e-1
tol_euler 1e-1
tol_xyz 1e-1   # tolerances for the different refinement steps
tol_eps 1e-2
tol_grain 1e-3
title 'noglobals, test new tolerances'
##### Parameters for strain-to-stress conversion
crystal_system cubic
c11 23.7e10
c12 14.1e10
c44 11.6e10
##### Parameters used for resuming refinement
#resume grain
#res_file S1_e0_050/S1_e0_050_grain.txt
#rej_file S1_e0_050/S1_e0_050_rej.txt
```

The name of the file can be anything and the format is simply ascii. The commands can be in any order. Some of them are mandatory, the rest will be given default values if nothing is specified. # is used for commenting.

command	**usage**
log_file	the name of the log file from GrainSpotter to be read
flt_file	the name of the filtered peaks file from peaksearch to be read
par_file	the name of the detector.par file from ImageD11 to be read
structure_file	(possible formats .cif or .pdb) contains information about the crystal structure that is used for outlier rejection based on intensities (Section 6.2)
dety_size dety_size	detector size along y and z in pixels default values are 2048 and 2048
w_step	the step size in ω
w_limit	the limits of the ω-intervals collected (max and min for every interval must be given, order irrelevant). If the data are collected as only one interval w_limit can be omitted as the limits can be determined from the filtered peaks file.
limit	limits for residual based outlier rejection after each refinement step 2 values corresponding to Sections 6.3.1 and 6.3.2 respectively
overlap	criteria for merging grains, see section 6.4
skip	used to rule out specific grains from the GrainSpotter log file that for some reason or another are known to be poor
tol_step	where step can be start, euler, xyz, eps or grain fit tolerance used in the corresponding refinement step In the above example the default values are given Note that the program sets tol_final = tol_grain·0.1 The refinement is considered converged when the estimated distance to the minimum is less than 10^{-3} times the specified tolerance.
title	anything to specify the refinement will be given in the output log.log and rej.txt files to keep track of things NB! Remember to use quotation marks
crystal_system	used for setting up stiffness tensor, allowed possibilities as of present: isotropic, cubic, hexagonal, orthorhombic, monoclinic, triclinic NB! All crystal systems can be treated as triclinic.
c11, c12, etc.	stiffness constants, program will stop if the supplied c's are not enough to fulfil the requirements of the crystal system
resume	possible values: start, euler, eps, start0 or grain resume refinement from the specified step in the algorithm (Section 3)
res_file	read refined grain parameters from this file (mostly used with resume)
rej_file	read information about rejected peaks and skipped grains from this file NB! Information about further outlier rejection performed by the new run of the program will be appended to this file

5 Output

All output files will be saved in i directory with the name of the used input file up to the first punctuation, so in the example using fitallb.inp as input file, the output directory will be named fitallb. This directory name will also be used as the stem of the generated output files which are the following:

file	**contents**
fitallb_log.log	Main output file: Global parameter values and standard deviations Residual values, refinement times and number of outliers for each refinement step
fitallb_rej.txt	List of rejected peaks and skipped grains for each refinement step The reason to reject the peak, i.e. intensity or residual, is specified
fitallb_cor.txt	Correlation matrix of the refined parameters for each grain New matrices appended after each grain and final step
fitallb_cov.txt	As fitallb_cor.txt, except this is the covariance matrices
fitallb_global.txt	If global parameters are refined the correlation and covariance matrices of these are written to this file
fitallb_start.txt fitallb_euler.txt fitallb_xyz.txt fitallb_eps.txt fitallb_start0.txt fitallb_grain.txt fitallb_start1.txt fitallb_final.txt	For each step in the algorithm the values of the parameters for each grain is listed in these files. Parameter order: grainno grainsize grainvolume x y z phi1 PHI phi2 U11 U12 U13 U21 U22 U23 U31 U32 U33 eps11 eps22 eps33 eps23 eps13 eps12 eps11_s eps22_s eps33_s eps23_s eps13_s eps12_s sig11 sig22 sig33 sig23 sig13 sig12 sig11_s sig22_s sig33_s sig23_s sig13_s sig12_s NB! If more than one final step is required the output file is overwritten in each step
fitallb_errors.txt	Estimated standard deviation of all the grain parameters given in the same order as listed above for the values The file is made by appending after each grain and final step

The parameters eps11, eps22,... are the components of the strain tensor in the Cartesian grain coordinate system; eps11_s, eps22_s,... are the strain tensor components in the sample system; sig11, sig22,... and the the components of the stress tensor in the Cartesian grain coordinate system; and finally sig11_s, sig22_s,... are the strain tensor components in the sample system.

The MINUIT program puts certain limitations on how the function FCN to be minimised is implemented. To make sure that the function has the right specifications it is necessary to build it before every refinement step. The building is performed by the build_fcn.py script and the result is a file called fcn.py placed in the output directory (in our example fitallb) which contains the functions to be minimised in the correct syntax. fcn.py is read by the program during execution.

6 Outlier rejection

Several different outlier rejection schemes are implemented into FitAllB, and the philosophy behind each of these will be outlined below. It should be noted that grains with less than 12 peaks are skipped, as it is considered unreliable to refine 12 parameters using less than 12 observations. In the future it should be made possible for expert users to change this 12 peak limit.

6.1 Median Absolute Deviation

Median absolute deviation (MAD) is used a robust measure for outlier rejection within FitAllB. Given the data set X_1, X_2, ..., X_n the median absolute deviation is defined as:

$$\mathrm{MAD} = \mathrm{median}_i(|X_i - \mathrm{median}_j X_j|)$$

For normally distributed data more than 99% of the observations should fall between the median±5MAD. In FitAllB the method is used iteratively:

```
While more outliers are present in the data set
    Calculate median and MAD of data set
    Reject data point differing by more than madlimit*MAD from median
```

6.2 Based on intensities

This check is only carried out if a valid structure file is given in fitallb.inp. In this case the measure $\frac{I}{L|F^2|}$ is calculated for each peak. Here I is the intensity, L is the Lorentz factor and $|F^2|$ is the theoretical structure factor squared determined based on the structure file. This quantity is directly proportional to the grain volume and should thus be roughly the same for all peaks assigned to a given grain. Thus a MAD outlier rejection with a madlimit of 5 is performed on these volume measures within each grain. Perhaps it should be made possible to choose another madlimit in order to change the priority of the intensity based outlier rejection relative to the other outlier rejection schemes.

6.3 Based on residuals

The residual is the refined quantity and should thus be as small as possible. Sometimes a single peak contributes abnormally much to the residual. This could either be because the peak belongs to a grain for which the parameters are still poor (in which case usually all peaks assigned to this grain will contribute rather much to the residual) or because the peak is assigned to a wrong grain. It is the peaks of this second kind that the residual based outlier rejection is tuned to spot and get rid of. The weight of the residual based outlier rejection relative to the other types of outlier rejection is determined by the user specified values limit1 and limit2. The smaller these are the more peaks are rejected based on residuals.

6.3.1 For specific grains, many peaks

At any given time the contribution to the residual from each peak can be calculated. Within each grain the mean and median of these are determined and the peaks with the largest residual contributions are rejected until the mean is less than reslimit times the median. This outlier rejection is used in two different places: 1) Before the refinement is initiated for all grain using reslimit=limit1 (default=5), and 2) For the poor grains towards the end of the refinement using reslimit=1.

6.3.2 After a refinement step, all grains, few peaks from each grain

After each refinement step the peaks contributing more than limit2 (default=10) times the mean peak contribution to the residual within each grain are rejected. This type of outlier rejection is able to identify one or two peaks differing significantly from the rest, but if more peaks are incorrectly assigned chances are that refinement will not change the parameters and this scheme doesn't work. In this case the entire grain will eventually be marked as poor and be subjected to the more thorough residual rejection scheme described in 6.3.1.

6.4 Merging of grains

It sometimes happens that the GrainSpotter comes up with two grains that are said to be different but in fact have the same orientation and many of the same peaks assigned. FitAllB contains a routine for identifying and merging grains of this type. In fitallb.inp the overlap command tells how large a fraction of overlapping peaks is allowed before the grains are considered to be identical. The default value is 0.5. Grains where a larger fraction of the assigned peaks are identical are cornered out and the grain with the fewest assigned peaks is skipped. In case the two grains contain the same number of peaks, only the identical peaks are kept in the refinement.

6.5 Peaks assigned to multiple grains

A large number of peaks will be assigned to more than one grain when GrainSpotter is run on data collected for more than 100 grains. FitAllB identifies these peaks and looks at how they contribute to each of the assigned grains. If it is found that the peak gives rise to a higher residual in grain1 than in grain2, and at the same time the volume calculated for the peak is further from the mean volume in grain1 than in grain2, then the peak will be rejected from grain1.

6.6 Poor grains

After each refinement step the number of peaks assigned to each grain, the grain contribution to the residual and the residual contribution per peak within the grain are calculated and output in the dialog window. When the refinement has converged the average residual contribution per peak should be of the order 1 since this is nothing but the squared distance between the calculated and measured g-vectors divided by the measured variance. Grains for which the residual per peak is more than 25 times the MAD above the median are marked as poor, and as seen in the Algorithm section these will eventually be handled in a special way.

7 Installation and execution

FitAllB can be downloaded from:

```
https://fable.svn.sourceforge.net/svnroot/fable/FitAllB
```

Go to the trunk and install the program by:

```
python setup.py install
```

After setting the relevant paths the program can then be called from the directory containing all input files in the following way:

```
fitallb.py -i fitallb.inp
```

Besides generating the already describe output files (in fitallb directory), this will give a lot of info in the dialog window from which the program was called.
Provided test example: 20 Al grains simulated with random orientation, positions and strains using PolyXSim (true values in al20_0020.txt), input: al20_90.inp, output directory: al20_90_out

7.1 Required packages

python 2.4 or later
numpy (http://www.numpy.org/)
pyminuit (http://code.google.com/p/pyminuit/)
From FABLE: ImageD11, xfab, polyxsim