FitAllB version 1.1.1

Jette Oddershede
DTU Physics
jeto@fysik.dtu.dk

July 19, 2012

Contents

1 Introduction

The FitAllB program, a part of the FABLE suite of programs for analysis of 3DXRD data (http://sourceforge.net/apps/trac/fable/wiki), 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 name FitAllB originates because the routine fits all $\mathbf{B}_i$ (1), the grain specific reciprocal space metrics, which contain information about the strain states of the individual grains.

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 $\mathbf{C}$ are provided, the stress tensors in the same two representations will also be output. FitAllB includes an error estimation routine to give standard deviations of all refined parameters. In addition the relative volumes of the grains (from the peak intensities with the possibility to take absorption effects into account) are refined, so in principle a 3D orientation and stress/strain map of the polycrystal can be obtained using tessellation. Lately an algorithm to extract the peak widths (median 2θ and η) for each grain has been added as an indicator of intragranular orientation and/or strain gradients.

The FitAllB program suite also contains calibration tools for refining the global parameters related to the experimental setup as described in Section 3.1.

2 Mathematical formalism

2.1 Basic equations

The refinements performed within FitAllB are done using the Variable Metric Minimisation routine of MINUIT [James, 1972] to minimise the following χ^2-function:

$$FCN = \sum_{i,j(i)} \left(\mathbf{\Gamma}_{ij}^{-1} \overline{G}_{ij} - \frac{\lambda}{2\pi} \mathbf{U}_i \mathbf{B}_i \overline{G}_{hkl,ij} \right)^T \mathbf{V}_{ij}^{-1} \left(\mathbf{\Gamma}_{ij}^{-1} \overline{G}_{ij} - \frac{\lambda}{2\pi} \mathbf{U}_i \mathbf{B}_i \overline{G}_{hkl,ij} \right) \quad (1)$$

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. $\overline{G}_{hkl,ij}$ is the assigned hkl of the reflection, $\mathbf{B}_i$ is the grain specific reciprocal space metric, which is connected to the strain tensor as described in Section 2.2, and $\mathbf{U}_i$ gives the orientation of the grain in the rotated system. $\mathbf{\Gamma}_{ij} = \mathbf{\Phi}_x \mathbf{\Phi}_y \mathbf{\Omega}(\omega_{ij})$ relates the rotated system to the laboratory system, $\mathbf{V}_{ij}$ is the covariance matrix of $\mathbf{\Gamma}_{ij}^{-1} \overline{G}_{ij}$, and the unrotated scatter vector

in reciprocal space, $\overline{G}_{ij}$, can be expressed in terms of direct space quantities:

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

where:

$$\bar{d_{ij}} = \mathbf{R} \begin{pmatrix} 0 \\ (y_{det,ij} - y_{det,0})p_y \\ (z_{det,ij} - z_{det,0})p_z \end{pmatrix} + \begin{pmatrix} L \\ 0 \\ 0 \end{pmatrix} - \mathbf{\Gamma}_{ij} \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}. \tag{3}$$

Refineable are 12 parameters per grain (3 positional (x_i y_i z_i) , 3 rotational ($\mathbf{U}_i$) and 6 strain ($\mathbf{B}_i$)) and the 10 global parameters: tilt ($\mathbf{R}$, 3 parameters), sample-to-detector distance (L), 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 ($\mathbf{\Phi}_x$ and $\mathbf{\Phi}_y$).

The measured quantities are ω_{ij}, $y_{det,ij}$ and $z_{det,ij}$, thus 3 observables per reflection.

2.2 Strain tensor definition

The grain specific reciprocal space metric of (1) relating reciprocal space to the Cartesian grain system is defined from the reciprocal lattice constants as [Poulsen, 2004]:

$$\mathbf{B}_i = \begin{pmatrix} a_i^* & b_i^* \cos(\gamma_i^*) & c^* \cos(\beta_i^*) \\ 0 & b_i^* \sin(\gamma_i^*) & -c_i^* \sin(\beta_i^*) \cos(\alpha_i) \\ 0 & 0 & c_i^* \sin(\beta_i^*) \sin(\alpha_i) \end{pmatrix} \tag{4}$$

There are several ways of defining the direct space counterpart $\mathbf{A}_i$ to $\mathbf{B}_i$. $\mathbf{A}_i$ relates any direct space coordinate system to the Cartesian grain system. Here it was chosen to define $\mathbf{A}_i$ to fulfill the following condition:

$$\mathbf{A}_i^T \mathbf{B}_i = \mathbf{I} \tag{5}$$

This implies that $\mathbf{A}_i$ is a lower triangular matrix and that the principal directions of $\mathbf{A}_i$ are the same as those for $\mathbf{B}_i$ such that the orientation matrix $\mathbf{U}_i$ relates both the reciprocal and the direct lattice to the rotated system.

Seing that the elastic strain tensor gives the perturbation of the local lattice in real space we define it as:

$$\epsilon_i = \frac{1}{2} \left(\mathbf{A}_i \mathbf{A}_0^{-1} + (\mathbf{A}_i \mathbf{A}_0^{-1})^T \right) - \mathbf{I}, \tag{6}$$

where $\mathbf{A}_0$ is (5) for the undeformed lattice and $\mathbf{I}$ is the identity matrix. From this it is clear that there is a one-to-one correspondence between the strain tensor components and the elements of $\mathbf{B}_i$ if the undeformed lattice constants are known.

The strain tensor as defined in (6) is sometimes termed the linear Lagrangian strain tensor [Schlenker et al., 1978].

2.2.1 Converting strain to stress

The stress is obtained from the strain via the stiffness tensor, **C**:

$$\sigma_i = \mathbf{C}\epsilon_i \tag{7}$$

It should be noted that one must check which direct lattice convention is used to define the strain tensor that **C** relates to. Here the direct lattice is defined to coincide with the standard definition of the reciprocal lattice, see above. For space groups with $\alpha = \beta = \gamma = 90°$ there is no ambiguity in how **C** is defined. The same holds for hexagonal space groups since hexagonal symmetry of the elastic properties implies cylindrical symmetry around the c-axis [Hosford, 1993]. However, special attention must be paid to trigonal, monoclinic and triclinic space groups.

2.2.2 Rotating the strain and stress tensors

The strain and stress tensors as defined above relate to the Cartesian grain system which is rotated by $\mathbf{U}_i$ relative to the sample system. Below is sketched how the different coordinate systems are related:

$$\begin{array}{lcl}
\text{undeformed} & \mathbf{U}_{i,0} & \text{undeformed} \\
\text{sample} & \longleftrightarrow & \text{Cartesian grain} \\
\text{coordinate system} & \mathbf{U}_{i,0}^T & \text{coordinate system} \\
 & & \\
 & & \mathbf{A}_0^{-1} \downarrow\uparrow \mathbf{A}_0 \\
 & & \\
 & & \text{direct space reference} \\
 & & \text{coordinate system} \\
 & & \\
 & & \mathbf{A}_i \downarrow\uparrow \mathbf{A}_i^{-1} \\
 & & \\
\text{deformed} & \mathbf{U}_i & \text{deformed} \\
\text{sample} & \longleftrightarrow & \text{Cartesian grain} \\
\text{coordinate system} & \mathbf{U}_i^T & \text{coordinate system}
\end{array} \tag{8}$$

From this it can be derived that if ϵ_i and σ_i are the strain and stress tensors in the Cartesian grain coordinate system describing the difference between the undeformed and deformed crystal lattices, then:

$$\epsilon_i^{sample} = \mathbf{U}_i \epsilon_i \mathbf{U}_i^T \tag{9}$$

$$\sigma_i^{sample} = \mathbf{U}_i \sigma_i \mathbf{U}_i^T \tag{10}$$

are the strain and stress tensors in the sample coordinate system, respectively. Here it has been assumed that $\mathbf{U}_i = \mathbf{U}_{i,0}$ since only $\mathbf{U}_i$ and not $\mathbf{U}_{i,0}$ can be obtained from the diffraction experiment. In the case of elastic deformation this is a good approximation, but it does not hold in the plastic deformation regime where grain rotations are non-negligible.

2.3 Error estimation

The covariance matrix of $\mathbf{\Gamma}_{ij}^{-1}\overline{G}_{ij}$, $\mathbf{V}_{ij}$, is assumed to be diagonal (thus no covariance) and equivalent for all reflections corresponding to a specific grain. Assuming that the experimental error on ω_{ij} is significantly larger than the error on the detector coordinates $y_{det,ij}$ and $z_{det,ij}$ and remembering that:

$$\mathbf{\Gamma}_{ij}^{-1} \approx \begin{pmatrix} \cos(\omega_{ij}) & \sin(\omega_{ij}) & 0 \\ -\sin(\omega_{ij}) & \cos(\omega_{ij}) & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{11}$$

we must have $\mathbf{V}_{ij}^{11} \approx \mathbf{V}_{ij}^{22} > \mathbf{V}_{ij}^{33}$. This relative weighting of the errors in x, y and z is taken into account by setting $\mathbf{V}_{ij}^{11} = \mathbf{V}_{ij}^{22} = 4 \cdot 10^{-8}/v_i$ and $\mathbf{V}_{ij}^{33} = 10^{-8}/v_i$, where v_i is the volume of grain i. The volume term is only significant for simultaneous refinements of multiple grains, thus only for fitgloball.py (Section 3).

If these error estimates are correct the contribution from grain i to (1) must be equal to the number of observation minus the number of refined parameters [Eadie et al., 1971], thus in most cases $3 \cdot j(i) - 12$. In case the contribution from grain i to (1) differs from the predicted value, the estimated experimental errors are scaled to make the observed contribution equal to the prediction. The errors on the refined parameters are then calculated by the MINUIT [James, 1972] to be the change in each parameter causing the value of (1) to increase by 1.

3 Algorithm

The FitAllB program package is written in python and contains three different scripts which basically run the same minimisation in different preset modes. The reason for this is that grain and global parameters cannot be refined simultaneously because of correlations, but in order to obtain accurate grain resolved strain tensors it is of the utmost importance that the global parameters relating to the experimental setup are well calibrated.

fitglobalgrain.py is used to refine the global parameters using one grain at the time. The final values and estimated errors are then obtained as the volume weighted average and spread over all grains.

fitgloball.py is used to refine the global parameters using all grains in one go. This procedure is more time consuming, but less affected by outlier grains. Large grains have more weight in the refinement than small ones, *c.f.* Section 2.3.

fitgloball_multidet.py is the newest variant of fitgloball.py where global parameters relating to several detectors can be refined simultaneously by taking reflections from all detectors into account in (1). The algorithm is described in more detail in Section 4.1.4.

fitallb.py is the main script for refining COM positions, orientation and strain tensors after having calibrated the global parameters. Only grain specific parameters are refined. If near-field diffraction data are available these are initially used to fit the grain positions, which are then kept fixed in the subsequent far-field refinement of orientations and strains.

3.1 Global parameter refinements

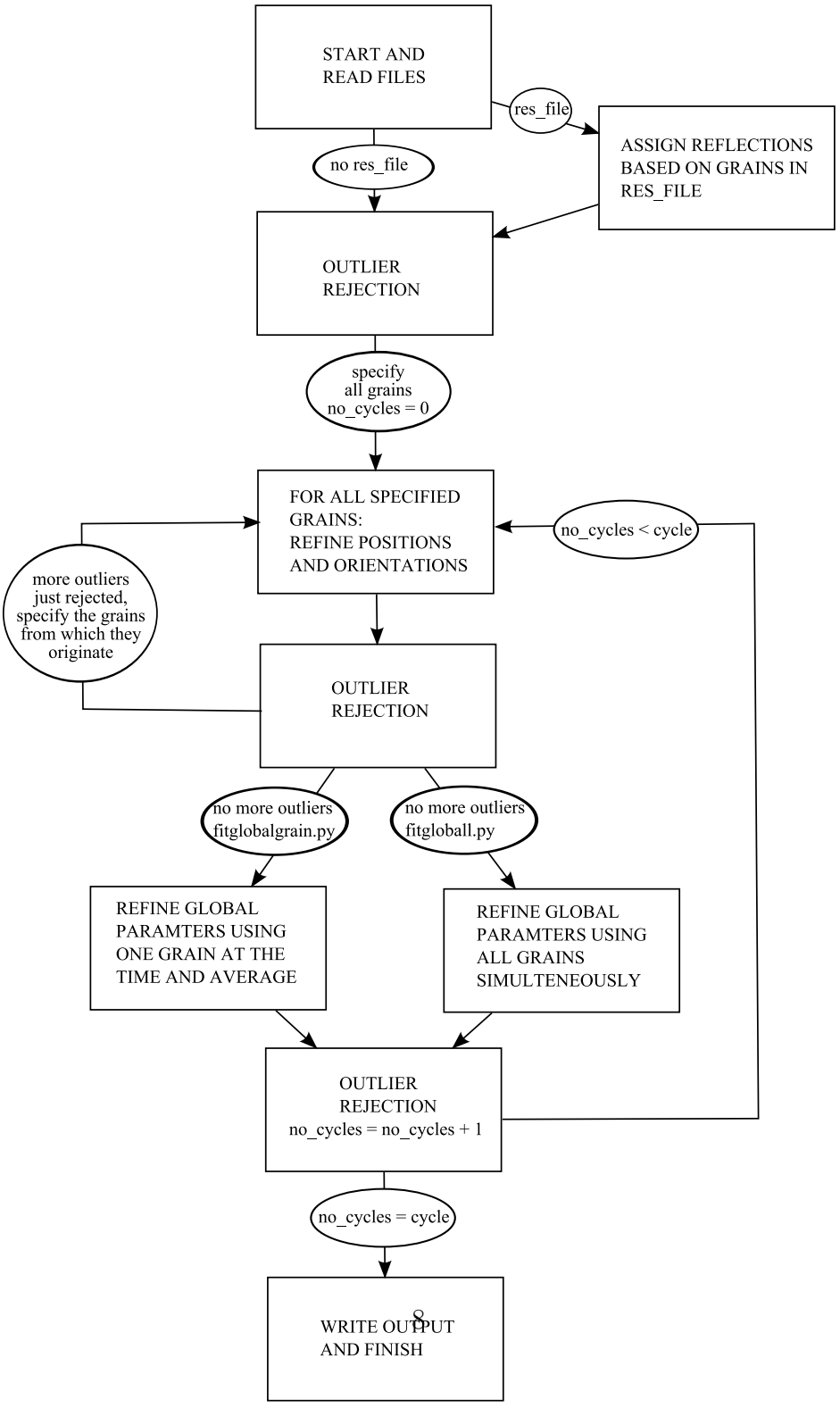
The calibration or refinement of global parameters must be performed using an undeformed polycrystalline sample. A good start guess on both the global parameters (ImageD11) and grain orientations and positions (GrainSpotter) is a prerequisite. The refinement is then carried out by alternately refining the global parameters and the grain parameters (positions and orientations) for a preset number of cycles to minimise correlations. An overview of the refinement procedure is given in Figure 1.

When fitting the global parameters the tilt of the sample stage corresponding to a rotation around the beam (in ImageD11 called chi, in FitAllB called wx, c.f. Table 3) is kept fixed as this correlates with the detector tilt component corresponding to a rotation around the same axis (ImageD11: tilt_x, FitAllB: tx). Furthermore the beam centre in the z-direction (ImageD11: z_center, FitAllB: cz) correlates with the grain positions along z. Therefore it is kept fixed for fitglobalgrain.py, while in fitgloball.py it is refined assuming that the center-of-mass of the assigned grains is zero along z.

Finally a number of options are only available in fitgloball.py. The first of these is the fitting of the unstrained reference lattice parameters (d0). Depending on space group the d0 fit covers from one (cubic) to six (triclinic) parameters. Please note that because of correlations it is only possible to refine one of the options d0, L (sample-to-detector distance) and pixel (detector pixel size). The other possibility is to restrain the grain position fits to only include a common COM translation along x (constr_x), y (constr_y) or z (constr_z). This is for instance useful if the grains of interest are known to be arranged in a plane or along a line, or if the rotation range is so small that the grain positions along the incoming beam direction are ill-determined.

After every cycle the refined global parameters are output in an ImageD11

Figure 1: Flowchart of global parameter refinements using an undeformed polycrystalline sample. Results from position and orientation step saved in stem_rotpos#.gff, from globals step in stem_globals#_fab.par, where #=no_cycle



format detector.par file. After the global parameter calibration this detector.par file can be used as input to the fitallb.py grain refinements so accurate grain parameters can be obtained.

3.1.1 Multiple detector screens

A new options to refine global parameters for several detector screens simultaneously using all indexed grains was implemented in fitgloball_multidet.py in April 2011. As of present (December 2011) the test directory contains an example input file (multidet.inp) for a Si single crystal and 3 detector screen. A successful refinement requires a good set of starting parameters for all detectors and one or more indexed grains. The peak assignments are then performed by forward projection as described in the next section one detector at the time. (1) is then built for all reflections of all detectors and refinements are carried out. In order to weight the data from the different screen the input parameters vars_scale#, # = 1,2,... are applied to scale the variances of the data corresponding to flt_file_# and par_file_# relative to those of flt_file and par_file. It should be noted that outlier rejections are only performed based on internal angles (Section 6.1) and residuals (Section 6.3). Volume based outlier rejection (Section 6.2) requires that similar scale factors (vol_scale#) are determined to take into account the difference in efficiency of the various screens, and this has not been done yet.

3.2 Grain parameter refinements

3.2.1 Far-field detector only

An overview of the grain parameter refinement procedure using only far-field data is given in Figure 2. Note that there are three different ways to start the refinement in terms of parameters of indexed grains and the corresponding diffraction peaks assignments:

1. Using the grain parameters and peak assignments from the grainspotter.log.

2. Reading the initial grain parameters from a .gff (Section 5) and using these to do a forward projection onto the far-field detector to harvest diffraction spots. Here it may happen that the number of assigned peak is higher than actually possible because very loose cuts are being used, but these falsely assigned peaks will eventually be removed as outliers. The tolerance for peak assignment in the forward projection is given by tol_fw_proj as an integer value telling how far away to search for peaks in terms of $\Delta\omega$ along ω and pixels along the detector y- and z-directions. The minimum step corresponding to tol_fw_proj = 1 is $(\Delta y, \Delta z)$ = (4,3) pixels. The default

values of tol_fw_proj is 2 and values up to 5 may be useful if the detector parameters are poor.

3. Reading the initial grain parameters from a .gff and the peak assignments from a .flt.new made by ImageD11/makemap.py (using the --no_sort option). If the supplied flt_file contains the labels column written by makemap this option will automatically be envoked, so make sure that the grain labels in the supplied res_file in fact correspond to those in the flt_file. makemap does not supply a .gff, but rather gives the grain information as .ubi, but xfab/ubi_to_gff.py can be used to do the required conversion.

3.2.2 Near- and far-field data

An overview of the grain parameter refinement procedure using both near- and far-field data is given in Figure 3. Here the far-field data is used to index the grains and the refinement is started from this information in one of the three possible ways described above followed by a forward projection onto the near-field detector. The comments given in the previous section regarding forward projection also holds here. For the near-field refinement all the near_ input option described in the next section apply.

4 Input

The main input file for fitallb.py/fitglobalgrain.py/fitgloball.py is an ascii file with any name containing a number of mandatory and optional commands with a given syntax. These commands contain information about all the remaining files to be read (Table 1), the experimental setup (Table 2), which parameters to fit and tolerances to use (Table 3), and what kind of outlier rejection to perform (Table 4).

Figure 2: Flowchart of fitallb.py for refinements using only far-field data. Results from first refinement step saved in stem_grain.gff, subsequent refinements in stem_final.gff

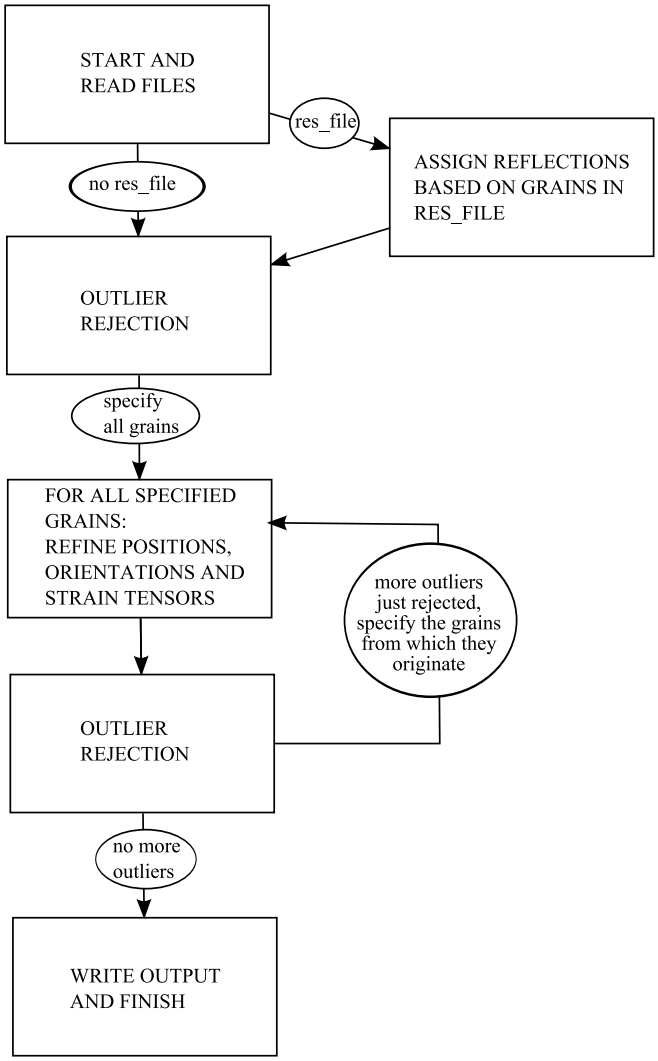


Figure 3: Flowchart of fitallb.py for refinements using both near- and far-field data. Results from near-field step saved in stem_near_rotpos.gff, from first far-field step in stem_grain.gff and subsequent steps in stem_final.gff

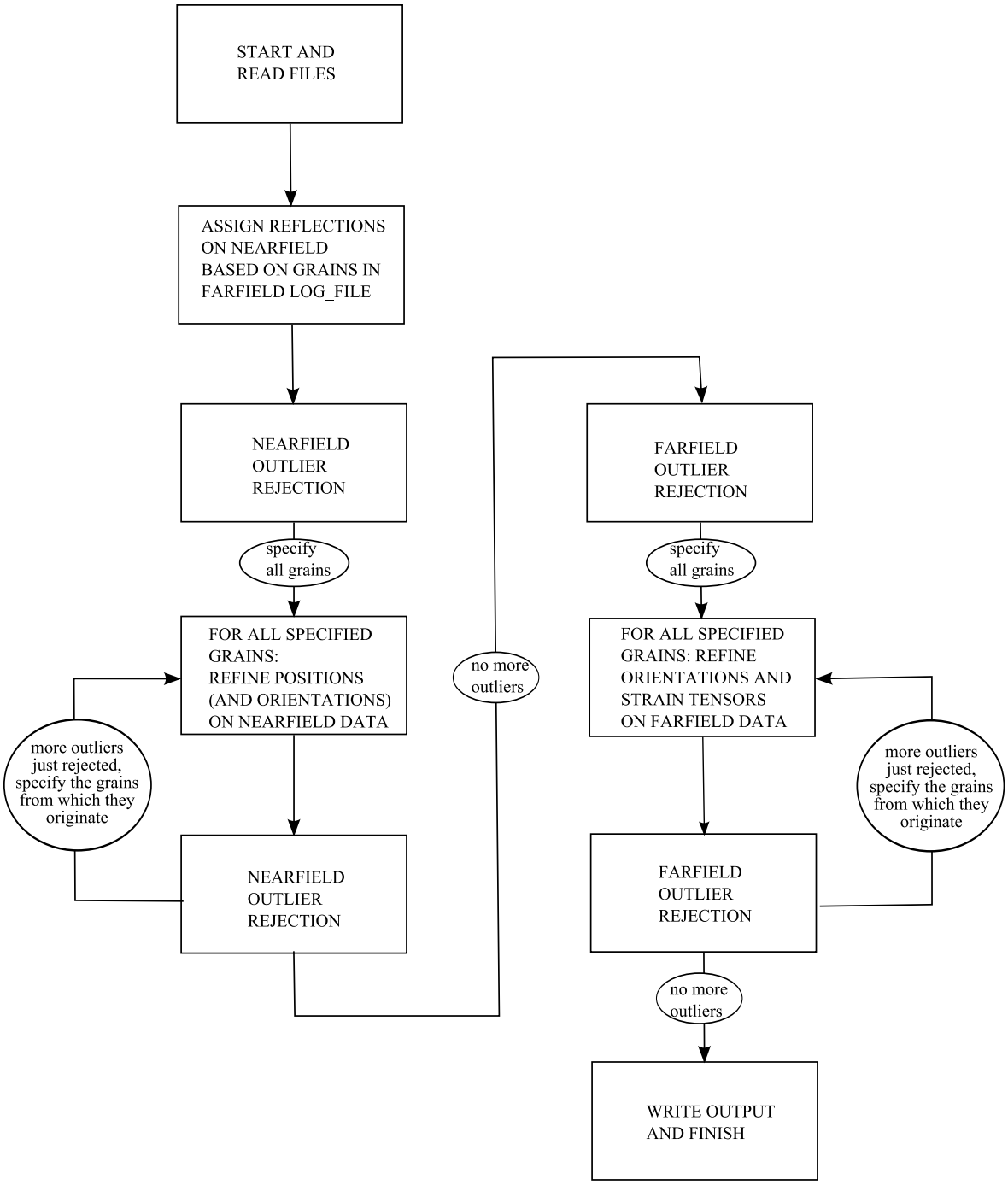


Table 1: List of input file commands. **Mandatory bold**.

command	**usage**
log_file	Name of the log file from GrainSpotter to be read Contains number of grains, and for each grain:
OR	**U**, (x, y, z), and for all assigned peaks:
	h, k, l, 2θ, η, ω, peak_id
res_file	Read starting values of grain parameters from this .gff (grain file format). If used the peaks assigned to each grain in the grainspotter.log are replaced by matches made by fitallb based on the grain parameters in res_file. NB! EITHER log_file OR res_file is needed, NOT both.
flt_file	Name of the filtered peaks file from peaksearch to be read Contains the filtered peaks, for each peak: peak_id, y_{det}, z_{det}, ω, I As of September 2011 the .flt.new from ImageD11/makemap (--no_sort) with grain labels for each peak can be used with the corresponding .gff as res_file
flt_file_#	# = 1,2,... for fitgloball_multidet.py (April 2011)
par_file	Name of the detector.par file from ImageD11 to be read Contains the experimental parameters: unstrained unit cell, wavelength, tilt, sample-to-detector distance etc.
par_file_#	# = 1,2,... for fitgloball_multidet.py (April 2011)
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)
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
near_flt_file	Name of the nearfield filtered peaks file
near_par_file	Name of the nearfield detector.par file
near_rej_file	Name of the nearfield rejection file

The commands can be given in any order, the optional ones can be left out if the default values should be used, and comments can be inserted in the input file by the use of # (see input examples in Section 4.1)

Table 2: List of experimental setup and sample commands. **Mandatory bold**.

title	Anything to specify the refinement NB! Remember to use quotation marks: 'my very own refinement specifying title'
w_step	Step size in ω
w_limit	Limits of the ω-intervals [deg] collected, e.g. -150 -30 30 150 min and max for every interval must be given, order irrelevant, thus 150 30 -150 -30 is equivalent to the above. 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. Used to reject peaks on the edges of the ω-intervals.
bg near_bg	Average background levels on far- and nearfield detectors. Used to reject overflown peaks, i.e. peaks with a maximum intensity above $2^{16}-$bg. Defaults: bg 1000, near_bg 100
dety_size **detz_size** dety_size# detz_size# **near_dety_size** **near_detz_size**	Farfield detector size along y and z in pixels default values are 2048 and 2048. # = 1,2,... for fitgloball_multidet.py (April 2011) Nearfield detector size along y and z in pixels default values are 1536 (y) and 1024 (z). Used to reject peaks close to the edges of the detector and to convert to the ImageD11 beam centre convention in case of a flipped detector, so it is important to check whether the defaults are correct.
crystal_system	Used for setting up stiffness tensor, allowed possibilities: isotropic, cubic, hexagonal, trigonal_high, trigonal_low, tetragonal_high, tetragonal_low, orthorhombic, monoclinic, triclinic. This information is also important for converting the orientation into the fundamental zone, therefore it is mandatory. isotropic is used for cubic with isotropic elasticity. trigonal_high: 32, 3m, $\overline{3}$m trigonal_low: 3, $\overline{3}$ tetragonal_high: 422, 4mm, $\overline{4}$2m, 4/mmm tetragonal_low: 4, $\overline{4}$, 4/m
stress	convert strain to stress, (1=yes/0=no), default: 0
c11, c12, etc.	Stiffness constants for strain to stress conversion (stress 1). The program will stop if the supplied c's are not enough to fulfil the requirements of the crystal system. A list of required c's for each crystal_system is given on the wiki: http://sourceforge.net/apps/trac/fable/wiki/FitAllB
beampol_factor	Beam polarisation factor, 1: fully plane polarised (default), 0: unpolarised
beampol_direct	Direction of the normal to the plane of the primary beam polarisation with respect to the sample rotation axis [deg], e.g. if the ω-axis is parallel to the z-axis the value is 0 (default), and if the ω-axis is along the y-axis it is 90.
abs_mu	Linear absorption coefficient in mm^{-1}, default: 0, abs_mu $\leq$ 0 no correction.
abs_xlim abs_ylim	xmin xmax, limits of rectangular sample cross section in mm, default: None. ymin ymax, limits of rectangular sample cross section in mm, default: None.

Table 3: Fit options: Parameters and tolerances. NB! The global parameters can only be refined when calling fitglobalgrain.py/fitgloball.py/fitgloball_multidet.py

command	**usage**
tilt	Fit detector tilt parameters tx, ty, tz (1=yes/0=no), default: 0
w	Fit wedge, omega stage tilt parameter wy (1=yes/0=no), default: 0 NB! chi, the omega stage tilt around x, correlates with tx.
center	Fit beam centre on detector (1=yes/0=no), default: 0 NB! For fitglobalgrain.py only cy, the centre along y is refined, while for fitgloball.py both cy and cz are fitted assuming that the COM of the assigned grains is zero along z.
pixel	Fit pixel size py and pz (1=yes/0=no), default: 0
L	Fit sample-to-detector distance (1=yes/0=no), default: 0
d0	Fit lattice parameters of unstrained reference cell (1=yes/0=no), default: 0 NB! This option is only possible for fitgloball and gives from one (cubic) to six (triclinic) independent parameters. Because of correlations only one of pixel, L and d0 can be non-zero.
constr_x constr_y constr_z	fitgloball.py option to restrain the position fits to include only a common COM parameter for all grains along x, y and/or z (plane or line geometry). (1=do restrain/0= refine freely), default: 0.
fixx fixy fixz	Fix grain position along x, y and/or z. (1 = do fix/0 = refine freely), default : 0. Useful for plane geometry (fixz) and limited scan ranges (fixx or fixy).
xyz	Fit positions on far-field (1=yes/0=no), default: 1
rod	Fit orientations and thus Rodrigues vectors on far-field (1=yes/0=no), default: 1
eps	Fit strain tensors on far-field (1=yes/0=no), default: 1 NB! For fitglobalgrain.py and fitgloball.py this must be 0.
near_xyz	Fit positions on near-field (1=yes/0=no), default: 0 NB! If near-field data are read (near_flt_file and near_par_file) it is important to set near_xyz 1, fit positions on near-field, and xyz 0, do not fit positions on far-field, because otherwise the accurate positions refined on the near-field detector will be overwritten in the far-field refinement step, c.f. Figure 2.
near_rod	Fit orientations and thus Rodrigues vectors on near-field (1=yes/0=no), default: 0
near_eps	Fit strain tensors on near-field (0=no only option)
tol_step	where step can be rotpos, grain or global, c.f. algorithm flowcharts. Fit tolerance used in the corresponding refinement step. Defaults are: tol_rotpos = tol_global = 1e-2 and tol_grain = 1e-3. 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.
tol_fw_proj	Tolerance of forward projection for peak assignment. Integer, default: 2. Indicates how far away to search for peak in steps of w_step along ω, and $(\Delta y, \Delta z)$ = (4,3) pixels in the detector coordinates. For near-field data maybe use near_tol_fw_proj = 10 and near_w_step = w_step/5?
cycle near_cycle	Number of cycles for fitglobalgrain.py/fitgloball.py, default=20

Table 4: Outlier rejection commands. Commands not marked by * can be used in a near_ option to specify different outlier rejection limits for the near-field data.

command	**usage**
rej_ia	Outlier rejection limit in degrees for gvector misorientation, Section 6.1. All peaks with misorientations larger than this are rejected. Default: 0.2.
rej_vol	Outlier limit for volume/intensity based rejection, Section 6.2. NB! Requires a valid structure_file command. Default: 5.
rej_resmedian	Outlier limit for robust residual based rejection, Section 6.3.1. Default: 5.
rej_resmean	Outlier rejection limit in terms of average peak contributions, Section 6.3.2. Default: 10.
rej_multi	Remove peaks assigned to more than one grain (1=yes/0=no), Section 6.4. If yes (default) the peak is removed from the grain where the fit (in terms of residual and volume/intensity) is poorest.
overlap	Criteria for merging grains, Section 6.5. Grain with a larger fraction of overlapping grains than specified are merged. Default: 1., i.e. no merging.
min_refl	If the number of reflection assigned to a grain drops below the specified value the grain is skipped in subsequent refinement steps. Defaults: min_refl 9, near_min_refl 3.
skip*	Used to rule out specific grains from the GrainSpotter log file or the .gff res_file that for some reason are known to be poor.
vars_scale#*	Scale factor of variances for detector screen # relative to far-field detector. Default: 1 (no scaling). Larger values (of the order 10) recommended to keep reflections from near-field detector in refinement

4.1 Example of FitAllB input files

4.1.1 Grain parameter fitting, far-field detector only

```
title 'IF steel, experimental data, undeformed layer 50'
##### Mandatory input files, farfield strain fitting
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 IF_steel.cif
##### Mandatory experimental parameters
dety_size 2048     # detector dimension [pixels]
detz_size 2048     # detector dimension [pixels]
w_step 0.5         # step size in omega [deg]
w_limit -22.5 22.5 67.5 112.5   # limits of omega intervals [deg]
crystal_system cubic
##### on/off possibilities for fitting
# Global parameters, do not fit.
# This is the default and only option, so no need to specify
# Grain parameters, fit all
rod 1              # Fit orientations and thus Rodrigues vectors (1=yes/0=no)
xyz 1              # Fit positions (1=yes/0=no)
eps 1              # Fit strain tensors (1=yes/0=no)
##### Outlier rejection information
rej_ia 0.2         # Outlier limit in deg for gvector misorientations
rej_vol 5          # Outlier limit for volume/intensity based rejection
rej_resmedian 10   # Outlier limit for robust residual based rejection
rej_resmean 5      # Outlier rejection limit in terms of average peak contribution
rej_multi 1        # Reject peaks assigned to more than one grain
min_refl 36        # Require 36 assigned reflections for good grain
overlap 1          # grain overlap tolerance
skip 31 45         # skip grains 31 and 45 in the GrainSpotter log_file
##### Tolerances
tol_grain 1e-3
##### Parameters for strain-to-stress conversion
stress 1
c11 23.7e10
c12 14.1e10
c44 11.6e10
```

4.1.2 Grain parameter fitting, near- and far-field data

```
title '10 grains of IF steel, 2 detectors'
##### Mandatory input files, farfield strain fitting
log_file if100_t50_10.log
flt_file if100_t50.flt
par_file if100.par
##### Additional files for nearfield position fitting
near_flt_file if100_near_t50.flt
near_par_file if100_near.par
##### Additional file to be read for volume based outlier rejection
structure_file IF_steel.cif
##### Mandatory experimental parameters
w_step 0.5              # step size in omega [deg]
w_limit -22.5 22.5 67.5 112.5   # limits of omega intervals [deg]
dety_size 2048          # farfield detector dimension [pixels]
detz_size 2048          # farfield detector dimension [pixels]
near_dety_size 1536     # nearfield detector dimension [pixels]
near_detz_size 1024     # nearfield detector dimension [pixels]
crystal_system cubic
##### Parameters for strain-to-stress conversion
stress 1
c11 23.7e10
c12 14.1e10
c44 11.6e10
##### Outlier rejection information
# Use default rej_ for both detectors, thus mo need to specify values in input
min_refl 36        # Require 36 assigned reflections for good grain on farfield
near_min_refl 30   # Require 30 assigned reflections for good grain on nearfield
##### on/off possibilities for fitting
# Global parameters, do not fit
# This is the default and only option, so no need to specify
# Grain parameters, farfield
xyz 0                # Do not fit positions on farfield
rod 1                # Fit orientations and thus Rodrigues vectors on farfield
eps 1                # Fit strain tensors on farfield
# Grain parameters, nearfield
near_xyz 1           # Fit positions on nearfield
near_rod 0           # Do not fit orientations and thus Rodrigues vectors on nearfield
near_eps 0           # Do not fit strain tensors on nearfield
##### Tolerances
tol_grain 1e-3
tol_rotpos 1e-2
```

4.1.3 Global parameter fitting, far-field only

```
title 'fitgloball 5 undeformed grains of IF steel, 20 cycles'
##### Mandatory input files
log_file if100_globals_wedge_5.log
flt_file if100_globals_wedge.flt
par_file if100_globals_wedge.par
##### Additional file to be read for volume based outlier rejection
structure_file IF_steel.cif
##### Mandatory experimental parameters
dety_size 2048     # detector dimension [pixels]
detz_size 2048     # detector dimension [pixels]
w_step 0.5         # step size in omega [deg]
w_limit -22.5 22.5 67.5 112.5   # limits of omega intervals [deg]
crystal_system cubic
##### Number of refinement cycles
cycle 20
##### Outlier rejection information
min_refl 45    # NB! Use only grains with high completeness
rej_ia 0.2
rej_resmedian 5
rej_resmean 10
rej_vol 5
##### Global parameters
w 1            # Fit omega stage tilt parameter wy
tilt 1         # Fit detector tilt parameters tx, ty, tx
pixel 0        # Do not fit pixel size py and pz
center 1       # Fit beam centre on detector along y-axis, cy
L 1            # Fit sample-to-detector distance
##### Grain parameters
xyz 1          # Fit grain positions
rod 1          # Fit orientations and thus Rodrigues vectors
eps 0          # Fit strain tensors
##### Tolerances
tol_global 1e-1
tol_rotpos 1e-2
```

4.1.4 Global parameter fitting, 3 detectors, test/multidet.inp

```
title 'fit of global parameters for 3 detectors'
res_file si.gff
no_det 3
flt_file frelon_t2000.flt
par_file frelon.par
flt_file_1 star1_t2000c.flt
par_file_1 star1.par
flt_file_2 star2_t500c.flt
par_file_2 star2.par
w_step 0.25
w_limit -100 100
dety_size 2048
detz_size 2048
dety_size1 2048
detz_size1 2048
dety_size2 2048
detz_size2 2048
vars_scale1 10
vars_scale2 10
crystal_system cubic
sgno 227
cycle 10
min_refl 45
rej_ia 0.2
rej_resmedian 10
rej_resmean 50
rej_vol 5
w 1            # Fit omega stage tilt parameter wy
tilt 1         # Fit detector tilt parameters tx, ty, tx
pixel 0        # Fit pixel size py and pz
center 1       # Fit detector centre along y-axis, cy
L 1            # Fit sample-to-detector distance
d0 0           # Fit unstrained lattice parameters
xyz 1          # Fit grain positions
rod 1          # Fit orientations and thus Rodrigues vectors
eps 0          # Fit strain tensors
tol_global 1e-2
tol_rotpos 1e-4
```

5 Output

All output files will be saved in a 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 given in Table 5.

Table 5: List of output files from fitallb.py, fitglobalgrain.py, fitgloball.py and fitgloball_multidet.py

file	**contents**
fitallb_log.log fitallb_near_log.log	Main output files for far- and nearfield: Global parameter values and standard deviations Residual values, refinement times and number of outliers for each refinement step
fitallb_rej.txt fitallb_near_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 fitallb_near_cor.txt	Correlation matrix of the refined parameters for each grain New matrices appended after each grain and final step
fitallb_cov.txt fitallb_near_cov.txt	As fitallb_cor.txt, except this is the covariance matrices
fitallb_global.txt	Correlation and covariance matrices of the refined global parameters are appended to this file after each cycle
fitallb_globals#_fab#.par	ImageD11 type detector.par file with refined global parameters For fitgloball_multidet.py fab0 contains the far-field parameter, and fab1, fab2, ... correspond to the additional screens.
fitallb.py: nearfield: fitallb_near_rotpos.gff farfield: fitallb_grain.gff fitallb_final.gff fitglobalgrain.py/ fitgloball.py: fitallb_rotpos#.gff	For each step in the algorithm the values of the parameters for each grain is listed in these (grain file format) files. Parameter order: grainno mean_IA grainvolume x y z rodx rody rodz 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.gff	Estimated standard deviation of all the grain parameters given in the same order as listed above for the values. The file is updated after each refinement step. Step specific output files are names fitallb_step_errors.txt
fitallb_cov_eps_sig.gff	Covariance matrices of strain and stress enabling the calculation of error bars along any direction, for instance for resolved shear stresses

The output grain format files contain the refined COM grain positions (x, y, z) and orientations in terms of the components (rodx, rody, rodz) of the Rodrigues

vectors [Frank, 1988] $\overline{r} = \tan\frac{\phi}{2}\overline{n}$ corresponding to a rotation of ϕ around the normal vector $\overline{n}$. Furthermore the components (U11, U12,...) of **U** are also given for convenience. 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 stress tensor components in the sample system. Stress tensor output requires that the stiffness constants are given in the input file and that the stress calculation is switched on (stress 1). The estimated errors on the refined parameters are calculated by MINUIT as described in Section 2.3, while the errors on the derived parameters are propagated from these.

In addition the relative grain volumes are given (requires a valid structure_file command in the input). These are calculated from measures $\frac{I\exp(\mu r)}{LP|F^2|}$ for each assigned reflection. Here I is the intensity, L is the Lorentz factor, P is the polarisation factor, $|F^2|$ is the theoretical structure factor squared determined based on the structure file, μ is the linear absorption coefficient and r is the total distance travelled inside the sample by the incoming and outcoming X-rays. This quantity is directly proportional to the grain volume and should thus be roughly the same for all peaks assigned to a given grain. The grainvolume is taken as the median of the single peak volume measures, while the median absolute deviation (MAD, page 23) is given as the estimated error.

Finally the peak widths along 2θ and η are calculated for each peak based on the peak shape information in the .flt (variances and covariances of sc and fc). The median of the single peak values for each grain are given as sig_tth and sig_eta [°] in the _final.gff, while the median absolute deviations are given in the _errors.gff. sig_eta reflects the intragranular orientation gradients, while sig_tth reflects both domain size effects and intraganular strain distributions. In a typical far-field experiment the domain size effects have been shown to be proportional to $\sqrt[6]{v}$, where v is the grain volume.

The python wrapper applied for the MINUIT variable metric minimisation, pyminuit (http://code.google.com/p/pyminuit/), puts certain limitations on how the function FCN to be minimised (1) 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, but deleted upon normal termination of the fitting.

Besides generating the described output files (in directory fitallb), the program will give a lot of info in the dialog window from which it was called (standard output). To start out with the number of reflections rejected based on the different

criteria are listed. Later on follows list of all grain with the number of assigned reflections, the total contribution to (1) from each grain and the average contribution per per reflection (should be roughly 3 according to the arguments in Section 2.3). While the program is running a counter specifies which grain is being refined now to give an idea of the refinement progress.

The quality of the fit can be judged from the fitallb_log.log and the standard output. Likewise the mean_IA given in the .gff can be used as an indication in this direction.

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 a user defined number of assigned peak (min_refl) will be skipped to avoid spending time on refining non-physical ghost grains. The fitallb_rej.txt file contains a record of all rejected reflections (which reflection rejected from which grain for what reason) and skipped grains. This file can be read and the information exploited if the refinement was stopped prematurely.

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:

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

For normally distributed data more than 99% of the observations should fall between the median$\pm$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

6.1 Based on internal angles

The internal angle (measured in degrees) is the angle between $\mathbf{\Gamma}_{ij}^{-1}\overline{G}_{ij}$ and $\frac{\lambda}{2\pi}\mathbf{U}_i\mathbf{B}_i\overline{G}_{hkl,ij}$ of (1). All reflections with an internal angle greater than rej_ia will be rejected as outliers. The default value for undeformed materials is 0.2, this must be increased for deformed samples.

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 $\frac{I\exp(\mu r)}{LP|F^2|}$ is calculated for each peak. Here I is the intensity, L is the Lorentz factor, P is the polarisation factor, $|F^2|$ is the theoretical structure factor squared determined based on the structure file, μ is the linear absorption coefficient and r is the total distance travelled inside the sample by the incoming and outcoming X-rays. 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 rej_vol is performed on these volume measures within each grain. The default value of 5 was chosen based on the arguments in MAD Section. Increasing the value will cause less reflections to be rejected. Reasonable values are found in [3;20].

6.2.1 Absorption correction

The directions of the incoming and outgoing X-ray beams in the unrotated sample coordinates system are given by:

$$\bar{r}_{ij}^{in} = \mathbf{\Gamma}_{ij}^{-1} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}. \tag{12}$$

$$\bar{r}_{ij}^{out} = \mathbf{\Gamma}_{ij}^{-1}\mathbf{R} \begin{pmatrix} 0 \\ (y_{det,ij} - y_{det,0})p_y \\ (z_{det,ij} - z_{det,0})p_z \end{pmatrix} + \mathbf{\Gamma}_{ij}^{-1} \begin{pmatrix} L \\ 0 \\ 0 \end{pmatrix} - \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}. \tag{13}$$

where all parameters are defined as in Section 2.1. In the unrotated sample system the boundaries of the rectangular sample cross section are found at x=x_{min}, x=x_{max}, y=y_{min} and y=y_{max}. $\bar{r}_{ij}^{out}$ points from the grain at position (x_i y_i z_i) towards the edges of the sample, and so does $-\bar{r}_{ij}^{in}$. If the first component of $\bar{r}_{ij}^{out}$ is negative the outgoing X-ray will cross the face determined by x=x_{min}, while the x=x_{max} face will be crossed if it is positive. Likewise if the second component of $\bar{r}_{ij}^{out}$ is negative the outgoing X-ray will cross the face determined by y=y_{min}, while the y=y_{max} face will be crossed if it is positive. Now the distances to the x- and y-faces, $r_{ij,x}^{out}$ and $r_{ij,y}^{out}$, can be calculated, and the true distance travelled by the outgoing X-ray inside the sample will be r_{ij}^{out}=min($r_{ij,x}^{out}$,$r_{ij,y}^{out}$). Using similar arguments the distance travelled by the incoming X-ray in the sample, r_{ij}^{in}, can be determined if one considers $-\bar{r}_{ij}^{in}$. The total distance travelled by the X-ray through the sample before giving rise to reflection j of grain i is thus $r_{ij} = r_{ij}^{in} + r_{ij}^{out}$.

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.

6.3.1 Based on residuals, robust

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 rej_resmedian times the median. The default value is 5, but any value in $]2;\infty[$ can be used, the larger the value the less reflections are rejected.

6.3.2 Based on average residuals

After each refinement step the peaks contributing more than rej_resmean (default=10) times the mean peak contribution to the residual within each grain are rejected. This specific outlier rejection scheme is, contrary to the others, not used until after the first refinement step as it is not very robust towards poor starting values of the grain parameters. Values in [5;20] are applicable in most cases.

6.4 Peaks assigned to multiple grains

Occasionally a number of peaks will be assigned to more than one grain, especially if many grains are illuminated simultaneously, the sample is textured or very loose cuts have been applied in the GrainSpotter indexing or the FitAllB forward projection (Section 3.2.1). 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. This feature can be turned off (default: on) by setting rej_multi to 0. This is for instance done for refinements of twinning samples where the different twin variants share a number of reflections.

6.5 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 1, i.e. do not merge anything. 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.

7 Installation and execution

The newest developement version of FitAllB can be obtained from the subversion (SVN) repository:

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

Go to the trunk and install the program by:

python setup.py install

NB! Most likely this has to be done with root privileges, e.g. put a sudo in front of the command.

Alternatively a packed version called FitAllB-1.1.1.zip can be downloaded from

https://sourceforge.net/projects/fable/files/FitAllB/1.1.1/

Unpack, go to FitAllB-1.1.1/trunk and install the program by:

python setup.py install

NB! Most likely this has to be done with root privileges, e.g. put a sudo in front of the command.

7.1 Required packages

python 2.4 or later (http://www.python.org)
pyminuit (http://code.google.com/p/pyminuit/)
PIL (http://www.pythonware.com)
numpy (http://www.numpy.org/)
scipy (http://www.scipy.org/Download)
PyCifRW (anbf2.kek.jp/CIF/)

From FABLE: ImageD11, xfab, PolyXSim, fabIO

7.2 Execution

After setting the relevant paths the program can then be called from the directory containing the main input files (here .inp) in the following way:

fitallb.py -i fitallb.inp
fitglobalgrain.py -i fitglobalgrain5_5.inp
fitgloball.py -i fitgloball5_5.inp
fitgloball_multidet.py -i multidet.inp

7.3 Provided test examples

All files required to perform the below describe test runs are found in trunk/test.

if10.inp: (fitallb.py, near- and far-field data)
10 grains of IF steel simulated using PolyXSim within an illuminated volume of $0.7 \times 0.7 \times 0.01$mm with random orientations, grain sizes and Gaussian strains with a mean of 0 and a spread of 0.001. The refinement is an example of using both near- and far-field data, and the runtime is approx. 1 min per grain on a 2.2 GHzm 3.5 GB RAM PC.

Zr_200MPa_noabs.fit and **Zr_200MPa_abs.fit:** (fitallb.py)
December 2011. Examples of reading .flt.new and the corresponding .gff from ImageD11/makemap.py. The examples show the required input to correct the calculated volumes of 10 indexed grains for the effects of absorption in the 1×1 mm^2 Zr sample at 80 keV (Zr_200MPa_abs.fit). The corresponding input for no absorption correction (Zr_200MPa_noabs.fit) is provided to allow the user to investigate the difference in the estimated error on the volumes.

if10_res.inp: (fitallb.py, far-field data only, resumed refinement)
Same data as for if10.inp, but only the far-field refinement is performed. The starting values of all grain parameters is read from the if10_near_rotpos.gff file generated in the if10 refinement to show how resuming a refinement can be done. Note that the grain positions refined on the near-field detector are kept fixed in this far-field refinement. Runtime approx. 40 sec per grain.

fitglobalgrain5_5.inp: (fitglobalgrain.py)
5 grains of undeformed IF steel simulated within a volume of $0.7 \times 0.7 \times 0.01$mm with random orientations and grain sizes. The input file of global parameters, if100_globals_wedge.par, has a slight offset in all of these as compared to the values used to simulate the diffraction images to demonstrate the powers of the fitglobalgrain.py. The true parameters used for the simulation are given in if100_globals_wedge_true.par for comparison. 5 refinement cycles performed. Runtime: 14 min.

fitgloball5_5.inp: (fitgloball.py)
Same input as for fitglobalgrain5_5, but intended for use with fitgloball.py to

demonstrate the different between the two. Runtime: 16 min.

fitgloball5_res2.inp: (fitgloball.py)
Resuming the fitgloball refinement of fitgloball5_5 using as input the final grain parameters, fitgloball5_5_rotpos4.gff, and globals, fitgloball5_5_globals4_fab.par, as input to do 2 more cycles. Runtime: 5 min.

multidet.inp: (fitgloball_multidet.py, 4.1.4)
Si single crystal, 3 detectors: Frelon (far-field for indexing) and the two screens of the Risø 3D detector; star2 (pixel size 1.5 μm, L=12 mm) and star1 (pixel size 4.5 μm, L=21 mm). 10 cycles using peaks from all 3 screens (final numbers 191, 23 and 48). Runtime: 40 min.

7.4 Contact

FitAllB has its own wiki with input examples and links to useful stuff:
(http://sourceforge.net/apps/trac/fable/wiki/FitAllB).
More information on the standard 3DXRD geometry can be found in the geometry document [Poulsen et al., 2010].
For publications using FitAllB please cite [Oddershede et al., 2010]. It reports the details of the FitAllB algorithm and some first validation experiments.

Questions, comments and suggestions are most welcome!

Jette Oddershede
DTU Physics
jeto@fysik.dtu.dk

7.5 Changes since version 1.1.0

Jan. 2012	Output of fitallb_cov_eps_sig.gff containing covariance matrices of strain and stress added to enable the calculation of error bars along any direction, for instance for resolved shear stresses.
June 2012	Added option to fix position along x, y and/or z. Useful for line beams and/or limited scan ranges.

References

[Eadie et al., 1971] Eadie, W. T., Dryard, D., James, F. E., Ross, M., and Sadoulet, B. (1971). *Statistical Methods in Experimental Physics*. North-Holland Publishing Company.

[Frank, 1988] Frank, F. C. (1988). *Metall. Trans. A*, 19(3):403–408.

[Hosford, 1993] Hosford, W. F. (1993). *The Mechanics of Crystals and Textured Polycrystals*. Oxford Science Publishers.

[James, 1972] James, F. (1972). http://www.cern.ch/minuit.

[Oddershede et al., 2010] Oddershede, J., Schmidt, S., Poulsen, H. F., Sørensen, H. O., Wright, J., and Reimers, W. (2010). *J. Appl. Cryst.*, 43(3):539–549.

[Poulsen, 2004] Poulsen, H. F. (2004). *Three-Dimensional X-ray Diffraction Microscopy*. Springer.

[Poulsen et al., 2010] Poulsen, H. F., Schmidt, S., Wright, J., Sørensen, H. O., Oddershede, J., and Alpers, A. (2010). 3DXRD and TotalCryst Geometry http://sourceforge.net/apps/trac/fable/attachment/wiki/WikiStart/Geometry_version_1.0.8.pdf.

[Schlenker et al., 1978] Schlenker, J. L., Gibbs, G. V., and Boisen Jr., M. B. (1978). *Acta Cryst.*, A34(1):52–54.