# Data processing in *CCP4 – iMosflm*

**Harry Powell**

IBMC, Strasbourg: Getting the Best from your Structural Data - Beyond Black Boxes

Thursday 6th October 2016



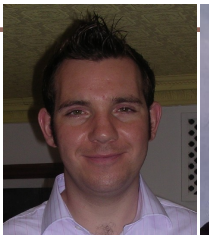
This is an introduction to both the ideas behind processing diffraction data and the way that we process the data within the CCP4 framework. It will cover the basic workflow of the programs used, and highlight those indications that show how well the whole process has proceeded.

The main focus will be on *iMosflm*, which is the graphical user interface for *Mosflm*, and which also uses the *CCP4* programs *Pointless*, *Aimless* and *cTruncate*. Although *iMosflm* can also use *Feckless*, its use is not covered in this talk; *Feckless* is used to process data from images containing diffraction from multiple crystals, which can also be integrated by *iMosflm*.

## Acknowledgements


Andrew Leslie


Phil Evans


Geoff Battye
2004 - 2007


Luke Kontogiannis
2008 - 2009


Owen Johnson
2010 - 2014


Graeme Winter

MRC Laboratory of Molecular Biology

CCP4

diamond

2/28

Andrew Leslie has been in charge of *Mosflm* development for many years; Phil Evans writes *Pointless*, *Aimless*, and other programs used in analysis.

Geoff Battye produced the original design for *iMosflm*, Luke Kontoggianis developed it to the point where it was a useful and robust tool. Owen Johnson added many features and continued to develop the program (almost) to the point it has reached today.

All five of the above have carried out their work on data processing at the Laboratory of Molecular Biology in Cambridge.

Graeme Winter worked (also at the LMB) on a predecessor of *iMosflm* that was never released, but the interest that he developed in diffraction data processing helped guide him towards developing *xia2* as a large part of his doctoral studies at Daresbury Laboratory and the University of Manchester.

*CCP4* funded *Mosflm* and *iMosflm* development (including my own position) from 1998 to 2016. The LMB provided us with somewhere to work. Diamond has employed Graeme since soon after the facility opened.

*iMosflm* and *Mosflm* can be downloaded either as an integral part of the *CCP4* suite, or from our website at

http://www.mrc-lmb.cam.ac.uk/harry

I can be contacted at harry@mrc-lmb.cam.ac.uk

Andrew can be contacted at andrew@mrc-lmb.cam.ac.uk

## Preamble – rationale for the experiment

We are measuring intensities of diffraction spots to obtain structure factor *amplitudes*

$$\left|F_{hkl}\right| = \left(\frac{KI_{hkl}}{Lp}\right)^{1/2} \quad (1)$$

$$\rho(x,y,z) = \frac{1}{V}\sum_{h}\sum_{k}\sum_{l} F_{hkl} e^{-2\pi i(hx+ky+lz)} \quad (2)$$

Careful data collection and careful measurement of intensities can be used to recover the phases (which are otherwise lost)

Data collection is the last *experimental* stage - if you collect bad data now you are stuck with it. No data processing program can rescue the irredeemable!

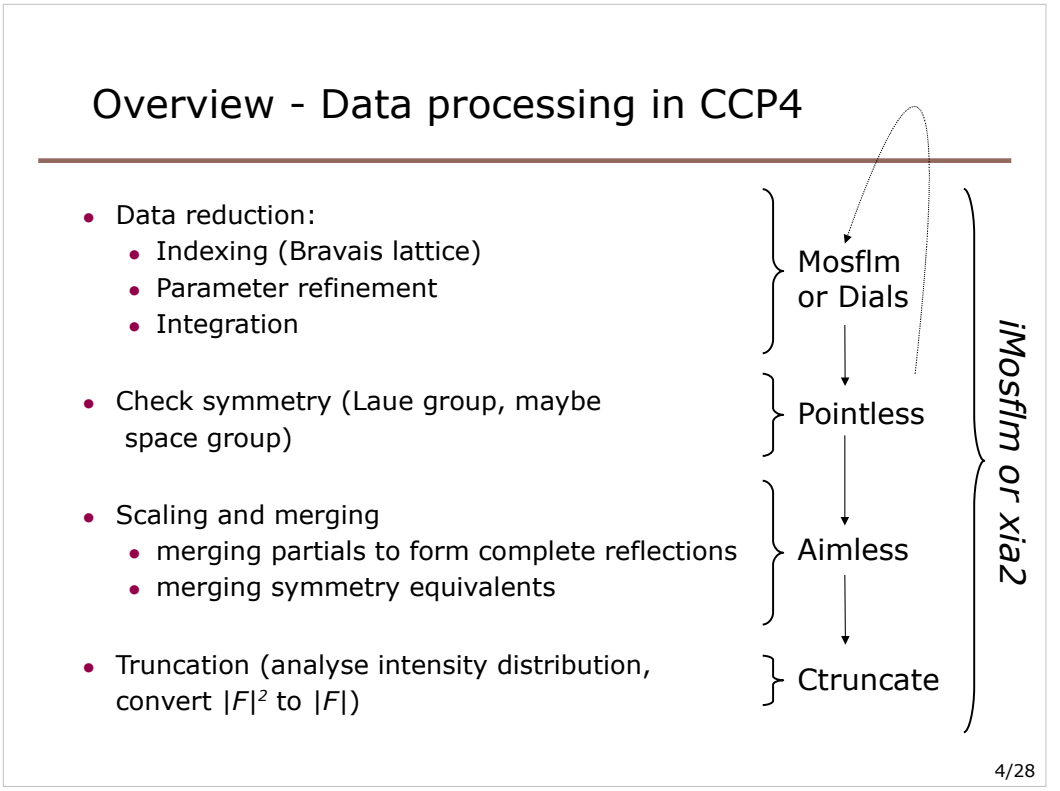
Don't necessarily do what your PI or post-doc (or even the beamline scientist) says – think! At Diamond or ESRF use Edna

3/28

The intensity of each reflection is related to the "structure factor amplitude" by equation (1) above. "*L*" is the Lorentz factor (which will be discussed later, but depends on, among other things, the data collection method), and "*p*" is the polarisation factor, related to the method of monochromation and the X-ray source. Both *L* and *p* depend on the diffraction angle of each reflection. "*K*" is usually a constant for a given crystal in an experiment, and depends on the crystal size, beam intensity and a number of other fundamental constants; since it is the same for every reflection in a dataset, it is usually applied as an overall scale factor to the measurements.

The "Structure Factor equation" (2) demonstrates why it is important to collect and measure the intensities as well as possible, since the electron density that gives us our structural model depends on the values we obtain for *F*. The electron density at every point in the cell depends on the intensity of every single reflection. Any badly measured or missing reflection will affect the maps we calculate.

Note that we are working with X-ray waves, and each diffracted ray has both an amplitude and a phase. The structure factor equation uses the structure factors *F*, not just the amplitudes |*F*|, but the phase information is lost in the data collection process. However, careful data collection and processing can allow us to obtain the phase information, usually by analysis of small differences in |*F*| between related reflections, *e.g.* in anomalous dispersion experiments like SAD or MAD, or in the classic heavy atom methods.




The process of converting the spots on a diffraction image to indexed and measured diffraction data that may be used in structural analysis consists of four basic parts, though in modern programs these tend to merge into a single workflow.

Measuring the intensity of spots on the images is “integration”. This can only be done well if the program knows the spot location, which is found approximately by indexing and then accurately by refinement of the crystal and detector parameters.

Once the measurements have been made, they are corrected for a variety of effects; purely geometrical effects are normally done by the integrating program – usually only Lorentz and polarisation effects. Other corrections, *e.g.* absorption by the crystal, differences between images (effective exposure, radiation damage, *etc.*) are either handled by the scaling and merging programs or by specialist programs devoted to particular aspects of the data.

Merging includes not only merging measurements of reflections that are equivalent by crystal symmetry, but also merging together the different components of reflections that are partially recorded over a number of adjacent images. This may be done either by the integration program (if it implements 3D profile fitting) or the scaling program (if the integration program performs a 2D analysis). Scaling attempts to put all of the observations onto a common scale, by accounting for errors and inconsistencies caused by the instrument or the crystal.

Truncation produces |F|s from these partially corrected $|F|^2$ measurements by taking account of expected statistical errors in measurement; analysing this process gives many of the diagnostics about twinning and also the Wilson statistics.

## *Mosflm*

- Indexes single crystal diffraction images
- Estimates the mosaic spread
- Refines the crystal and detector parameters
- Calculates a strategy for data collection
- Performs 2D integration of diffraction spots
- Produces extensive diagnostics and a large log file
- Outputs a file in *multirecord* MTZ format for further data analysis



Data integration involves these steps. Indexing is necessary so that Mosflm knows approximately where on the images the diffraction spots occur. Mosaicity estimation

## *iMosflm*

- Interface to *Mosflm*
- Has familiar file-browsing tools
- Displays the images
- Displays results graphically
- Connects to external programs *Pointless*, *Aimless* and *cTruncate* for further analysis *via QuickScale* option

Both *Mosflm* and *iMosflm* run natively on Windows, Mac OSX and Linux



*iMosflm* is the graphical user interface to the data processing program *Mosflm*. It is written in TclTk with its object-oriented extensions [incr]Tcl and [incr]Tk.

## Typical use of *iMosflm* for processing

| | |
|---|---|
| Image Task | • load images for processing |
| Indexing Task | • find spots on two images at ~90º to each other<br>• index (usually use *iMosflm*'s choice of solution)<br>• estimate mosaicity<br>• check the predictions match the spots |
| Integration Task (1) | • Integrate ~5 - 10º of images<br>• Run QuickSymm to check symmetry |
| If symmetries match from QuickScale and Indexing Task: | |
| Refinement Task | • refine crystal & detector parameters<br>• check the predictions match the spots |
| Integration Task (2) | • integrate dataset<br>• run QuickScale task |



This is a typical route through processing a straightforward dataset with *iMosflm*. The first integration step, where a few images are processed *before* refinement, allows the user to make themselves more sure of the true symmetry of the unit cell before they proceed further into the more time-consuming step of integration.

If the symmetries found by *QuickScale* and the one chosen by the user do not match, it is always best to go back to the Indexing task and choose a solution with a good penalty that does match *QuickScale*.

The first choice of the user should always be to use the solution in indexing chosen by *iMosflm* unless the user has a good reason to choose another.

## Practicalities of processing with *Mosflm*

- Reads and uses experimental information from image headers (*e.g.* detector type, wavelength, distance, oscillation range)
- Designed for macromolecular crystallography, but usable for small molecules
- Uses images in the current dataset to optimise processing parameters in all stages rather than default values
- Comprehensive set of commands for experienced users

*Therefore*

- No need to use configuration files for processing
- Straightforward for novices to process data effectively
- *Processing can be almost automatic - emphasised in iMosflm*



Over the years, the authors of *Mosflm* have put a lot of effort into making sure that it reads the "metadata" in the diffraction image headers correctly, and that the processing is automatically optimised for the current dataset. This means that integrating images after reading them in can be almost automatic, but it remains important for the user to use the feedback that *iMosflm* gives in the forms of graphs, images, overlays, *etc*.

It should be noted that *Mosflm* has been optimised for processing images collected from macromolecular crystals (which have close spots, and generally quite high background levels); by changing some of the processing parameters it is, however, straightforward to integrate small molecule datasets.

## Before processing

Load the dataset

- examine several images, e.g. first image,
  another 45º away
  another 90º away

- Use the program tools to mask the beamstop, cryostream, other shadows
- Set the resolution limit to ~0.5Å higher than visible spots for PAD, ~0.2Å higher for CCD
- Make sure the beam position is more or less correct
- Make sure other parameters (distance, wavelength, oscillation angle) are what you expect
  - Do they correspond with what is in your notebook
  - Did you make an independent note?

9/28

Although integration programs can make good attempts at measuring spots that are partially masked by obstructions such as the backstop, backstop arm or the cryostream, these reflections can cause severe difficulties in scaling.

For example, if there are only two measurements of symmetry-related equivalents of a reflection, and one is weak and the other strong, the scaling program cannot tell if one is masked or the other contains a zinger.

Usually, there is measurable intensity beyond the resolution limit visible to the eye; I find it is reasonable to integrate to about 0.5Å higher resolution for data collected using a Pixel Array Detector *(PAD*, like Pilatus or Eiger), and about 0.2Å higher if using a CCD.

The beam position is critical to successful indexing and further processing. It should be correct to less than half the minimum spot separation, or the calculated indices of the reflections could well be out by one or more, even if the cell is approximately correct.

Finally, make sure the parameters used by the program are what you expect, or remember from the data collection. Don't necessarily believe the information in the image headers - some beamline staff are less thorough in updating their set-up files than others.

## Indexing - what is it for?

Provides

approximate values

- indices (*h, k, l*) for each reflection
- unit cell dimensions (a, b, c, α, β, γ)
- crystal orientation (matrix)
- (first estimate of the Bravais lattice)

Knowledge of these allows us to find the positions of the diffraction spots on the image so their intensities can be measured.

Unit cell dimensions are used in structure solution, refinement, model building, analysis - so we need accurate values.



Indexing provides us with the information required to integrate the images in a dataset; the unit cell parameters and orientation of the crystal (in combination with known instrument parameters such as crystal to detector distance, wavelength of radiation, *etc.*) tell us where the diffraction spots occur on the detector for each image.

Further, the unit cell dimensions are used in many of the subsequent calculations in structure determination and refinement. Accurate values (obtained after refinement) will mean that the derived results have higher significance.

If we can determine the Bravais lattice, symmetry constraints can be applied in refinement to make the process more stable. Further, if we can determine the symmetry (or at least eliminate low symmetry solutions) we can run data collection strategy software and make sure we collect complete data with as small a rotation range as possible; in the case of crystals that suffer significantly from radiation damage this can be very important.

## Indexing in *Mosflm*

The program

- Finds spots on the image
- Converts 2D image co-ordinates to 3D scattering vectors (which correspond to reciprocal lattice coordinates)
- Indexes *via* Fast Fourier Transform
- Reduces the cell
- Lists the 44 characteristic lattices for this reduced cell, with a penalty value for each
- Picks a likely solution
- (estimates mosaic spread of the crystal)

The user should

- Check predicted position of spots on more than one image



Indexing involves several distinct processes, the main ones of which are listed here. They start with "spot finding", or locating likely diffraction spots on the image or images (indexing tends to be more robust when information from several images separated in phi are used, rather than just from a single image).

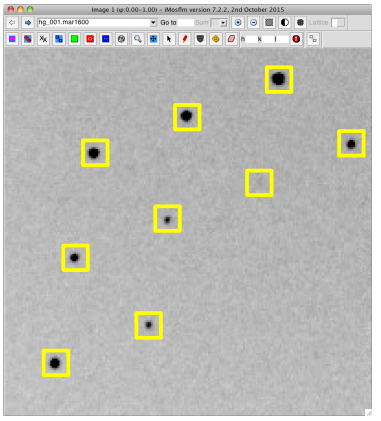
The two-dimensional co-ordinates can be mapped (using the Ewald sphere construction) to scattering vectors that correspond to (approximate) 3D reciprocal lattice co-ordinates.

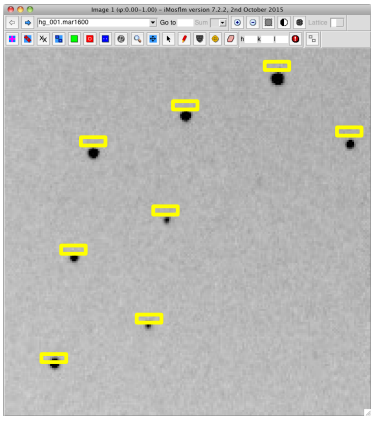
Indexing itself within Mosflm uses a “real-space” method (*i.e.* the real space unit cell dimensions are obtained directly, rather than via the reciprocal space unit cell) using an FFT-based method suggested by Gérard Bricogne in 1986 and implemented with a large set of 1D transforms by Steller *et al* (1997). An alternative formulation using a single 3D transform is used in HKL. XDS uses a method based on “difference vectors”, which will not be discussed further here.

The initial cell obtained may not be the “reduced cell”, *i.e.* with angles closest to 90º and the shortest cell edges, so “cell reduction” is performed next. At this point, the cell has triclinic symmetry; it can be transformed via a set of operations (listed in International Tables for Crystallography Vol. A) to 44 characteristic lattices (each of which corresponds to one of the 14 Bravais Lattices), and a distortion penalty calculated for each lattice. It is important to remember that the 44 solutions correspond to the single triclinic lattice obtained from indexing.

Having chosen a solution, the user should obtain an estimate of the mosaic spread of the crystal, prior to refinement. Mosflm uses an iterative integration routine to calculate a starting value.

## Check the predictions





Predictions should match spot

- positions
- size
- shape



The image on the left shows prediction boxes that are of the right size, shape and position for the image. The one on the right shows poorly shaped boxes that are not in quite the right places so there must be an error somewhere. This sort of situation can arise for a number of reasons (see the practical for examples).

## Indexing only gives the geometry of the cell

Indexing gives us a basis solution that is triclinic.

Applying symmetry transformations to give the *reduced bases* allows us to see how well this triclinic solution fits lattices with higher symmetry, *e.g.* monoclinic, orthorhombic etc.

*Mosflm* gives all 44 solutions, each corresponding to one of the 14 Bravais lattices (each of which may occur several times as a result of different transformations)

The unit cell geometry may not be the correct crystal symmetry, but it usually is.

The space group is only a hypothesis until after your structure is deposited in the PDB (or later...)

13/28

The cell dimensions derived from autoindexing usually give a good indication of the true symmetry of the crystal. For example, in the case that $a \neq b \neq c$, $\alpha \neq \gamma \neq \beta \neq 90$, the crystal system is most probably triclinic, unless the indexing has failed. If $a=b \neq c$, $\alpha=\beta=\gamma=90$, the crystal system may be tetragonal, but there are many examples where unit cells fit this but the true symmetry is orthorhombic or lower.

However, probably more than 95% of the time, the crystal symmetry derived from the unit cell geometry will be correct.

The practice of providing all 44 characteristic lattice solutions in *Mosflm* and *XDS* is to be preferred to that of *Denzo/HKL*; the latter only gives the "best guess"of each characteristic lattice as a choice. A small error in instrument parameters, or even in the choice of spots used for indexing, could easily give rise to the correct solution not being present in the list of results, *even though the program has actually calculated it.*

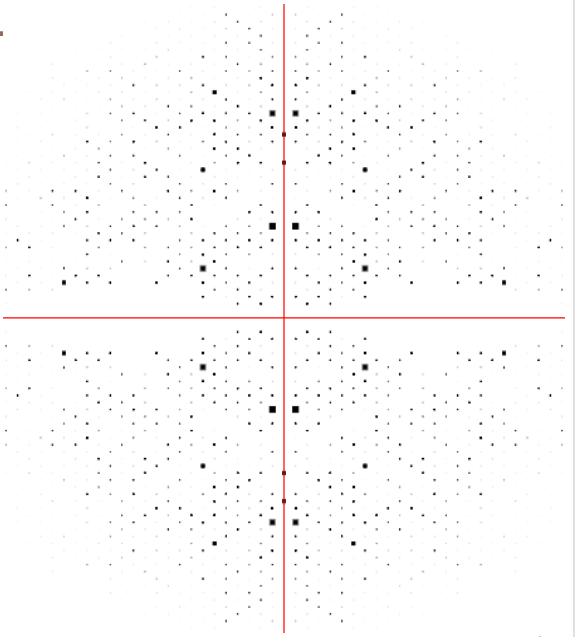
Note that *Dials* only lists the "best" solutions by default; this is okay provided that the indexing has worked correctly, but does mean that the user may not even be aware of other possibilities.

The 44 characteristic lattices and the transformations from the basis triclinic solution that correspond to the reduced bases are tabulated in *International Tables Volume A* pp 750 - 755. Each characteristic lattice (or lattice character) is associated with a Bravais lattice, *e.g. aP* is primitive triclinic ("anorthic Primitive"), *mC* is C-centred monoclinic *etc.*

## Bravais lattice – from intensities

The true Bravais Lattice symmetry can *only* be determined by analysing the intensities of symmetry equivalent reflections – *i.e.* after integration.

In *iMosflm*, we use the *QuickSymm* option in the Integration task to run *Pointless* and determine the true symmetry.

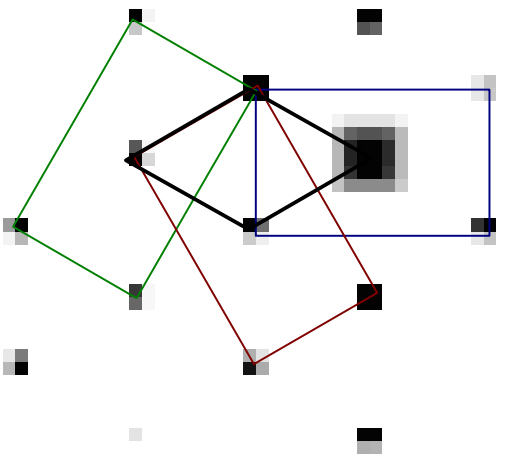


14/28

This is an example provided to Phil Evans where the metric symmetry indicated that the crystal was hexagonal, but the merging statistics showed that it was C-centred orthorhombic; the *mm* symmetry of the diffraction spots projected along the c* axis clearly illustrates this.

There are also two incorrect C-centred orthorhombic solutions at 120° to the correct solution, with identical cell parameters (see below); again, it can be seen that the reflections that should have the same intensity by hexagonal symmetry do not match.

It is interesting to note that autoindexing gave variously the hexagonal or one of the three orthorhombic solutions, depending on the choice of spots used in indexing – or only a one in four chance of the correct answer. Differentiating between the four solutions and picking the correct one can only be done *after* integrating at least some images; *iMosflm* includes a *QuickScale* task button in the *Integration* pane that runs *Pointless* to perform this analysis.



## Refining the parameters (1)

Optimise the fit of observed to predicted spot positions, so that the measurement boxes can be placed accurately over the spots.

Specifically, improve estimates of:

- Crystal parameters
- Instrument parameters

Accurate cell dimensions are important after data processing because they are also used in all subsequent stages of structure determination, refinement and analysis.

Can be performed by either (or both):

- Positional refinement using spot co-ordinates
- Post-refinement using intensity measurements



Indexing is based on approximations, and the fit of observed spots to their calculated positions can be improved by refinement. These approximations include the phi position of the centroid of each reflection and various parameters like crystal to detector distance and detector mis-setting angles. Provided that there are sufficient usable data at high enough resolution, refinement not only gives better information about where on the detector the spots occur, but also gives better estimates of both the crystal and instrument parameters.

Most integration programs use a “positional refinement” based on the spot positions on the detector surface; this is simple to calculate, but care must be taken because several parameters are closely correlated (*e.g.* cell edges and crystal to detector distance), especially at low resolution.

Mosflm combines positional refinement with another method, which is based on the relative intensities of the different parts of partial reflections across several images. Because this can only be done *after* the reflections have been integrated, it is called “post-refinement”. Using both methods together has distinct advantages over just using positional refinement, *e.g.* it is possible to de-couple the crystal parameter refinement from that of the crystal to detector distance, and it also gives (provided there are sufficient reflections for a stable refinement) more accurate cell parameters than those available from positional refinement.

Other processing packages delay post-refinement until a step following integration, and often combine it into the scaling and merging step.

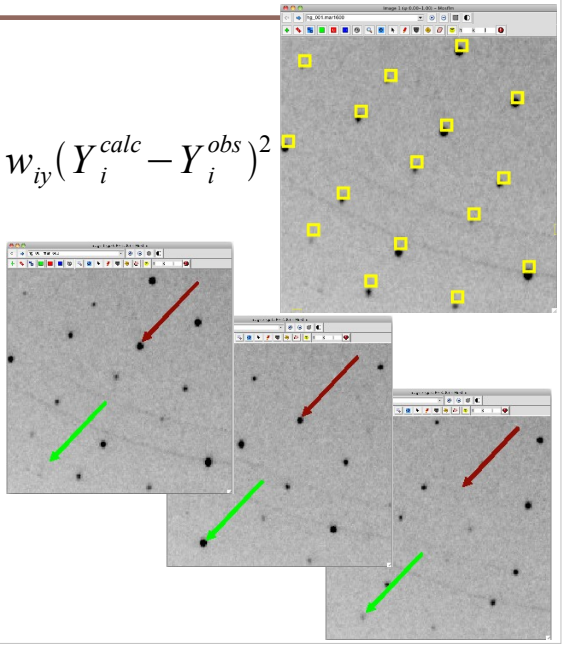
## Refining the parameters (2)

Positional refinement:

$$\Omega_1 = \sum_{i=1}^{n} w_{ix}(X_i^{calc} - X_i^{obs})^2 + w_{iy}(Y_i^{calc} - Y_i^{obs})^2$$

Postrefinement:

$$\Omega_2 = \sum_{i=1}^{n} w_i \left[ \frac{(R_i^{calc} - R_i^{obs})}{d_i} \right]^2$$



Positional refinement uses the positions of the spots on the detector to improve values of crystal and detector parameters. Some of these, for example unit cell dimensions and crystal to detector distance, may be very strongly correlated so they will not refine stably.

Postrefinement uses the intensity of parts of reflections sprad over multiple images to refine crystal and detector parameters, and gives the most accurate values for unit cell dimensions.

*Mosflm* uses both residuals, refining some parameters with one and other parameters with the other to remove the chance of unstable refinement and correlated residuals (*e.g.* cell is refined with postrefinement and distance with positional refinement).

## Integration

The process of measuring the intensities of the diffraction spots.

Two basic ways (*Mosflm* does both) -

- Summation integration; simple, fast, okay for all except weak, overloaded or partially overlapping reflections
- Profile fitting (only *intended* to improve weak spots); can be sub-divided into
  - two-dimensional (2D) – builds up reflections from profiles on individual images – *Mosflm, HKL*
  - three-dimensional (3D) – builds up profiles across several adjacent images – *XDS, DIALS,* not *Mosflm*

17/28

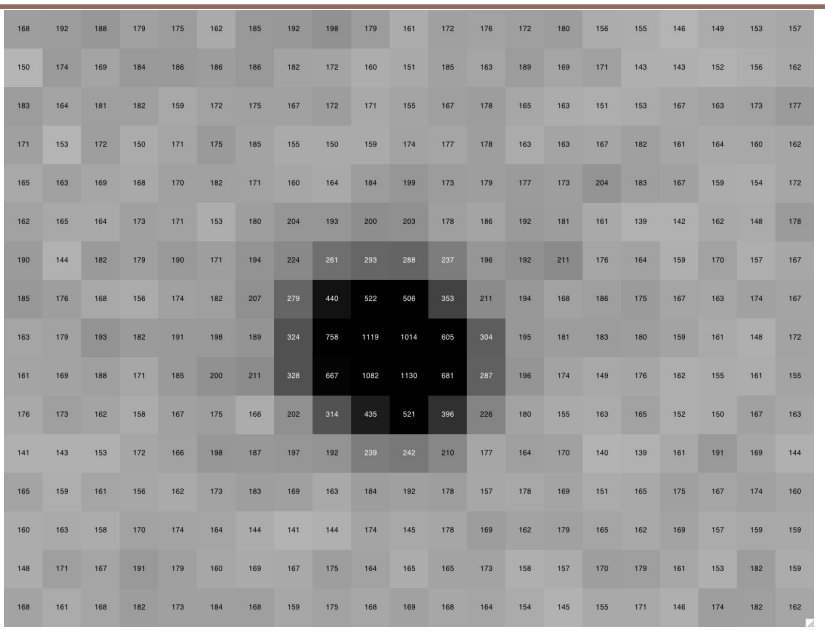
Integration is performed once the crystal and instrument parameters have been optimised by refinement.

The main difference between two-dimensional and three-dimensional integration is that the profiles used for partials over several images for 2D integration are the same for each part of the reflection, whereas for 3D integration, the profile for different parts of the same reflection can change significantly.

In principle, 3D profile fitting should give better results than 2D, but in practice the difference does not seem to be important, and other differences between programs (or even parts of the same program) tend to dominate.

Another major difference is that 2D integration will record both fully recorded reflections (those that ar complete on a single image) and partial reflecitons (those that are spread over multiple images); the partials need to be merged to form fully recorded reflections in the subsequent scaling step. This is not necessary for 3D integration.

## Measuring the intensity of a spot




18/28

In order to measure the intensity of the spots, we need to be able to (among other things):

- Identify where they are on the images (to within a fraction of a pixel)
- Determine how big they are on each image
- Work out how much background is underneath the spot
- Allow for the detector not being flat, and not perpendicular to the X-ray beam

## Summation integration

- In the absence of background, just add the pixel counts in the spot region together - but there is (always) background!
- Need to define spot and background regions - we cannot measure background directly under the spots, so we calculate a local background plane and slope from nearby non-spot pixels
- Use this to subtract the background under the spots
- Weak spots may have their shoulders under the background, so their measurement is impaired.

19/28

If the background intensity is negligible, the program doesn't even need to be very accurate in its placement of the integration boxes when using summation integration, provided they enclose all the spot intensity.

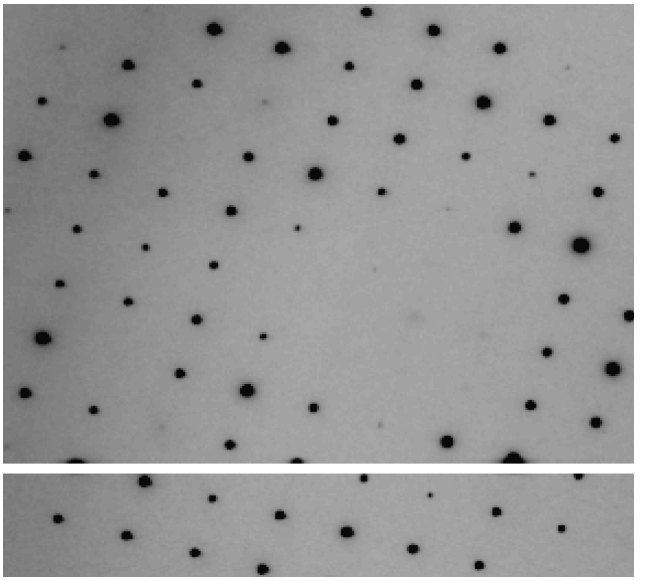
In practice, however, there is always some background, so this needs to be taken into account. It is impossible to measure the background directly under the spot, but its intensity can be inferred by assuming it to be a sloping plane in the neighbourhood of the spot. If the plane is steeper than some threshold value (*e.g.* because the spot is near an ice-ring), *Mosflm* will issue a warning.

With some newer detectors that have very low intrinsic noise levels and small point-spread functions, it is probably correct to integrate using summation integration (at least for the strong reflections), especially when the background is low. However, weak spots will still have their shoulders hidden by the background, and summation intensity will not measure their intensity optimally.

## Profile fitting

Based on the idea that spots in the same region of the detector (on the same and nearby images) will have similar profiles independently of their intensities.

This is not true if the size of the spots is similar to the size of the pixels (*e.g.* Pilatus, Eiger) because the sampling is too coarse.



20/28

The spot shape on a detector (including its intensity profile) is a function of several physical factors – the cross-section and divergence of the illuminating radiation, the size, shape and mosaic spread of the crystal (and its orientation relative to the beam), the direction the diffracted beams exit from the crystal, scatter from air in the beam path, the size and shape of the pixels on the detector, etc.

For a given image (or short series of images) most of these may be assumed to be constant in the diffraction experiment (or nearly constant); the biggest change between nearby (fully recorded) spots is in the direction of the diffracted rays from the crystal, and if the angle between these rays is small, this major difference is also small, so the idea that spots close to each other on the detector (even on different images) have similar profiles has some validity. However, if the physical spot size (determined by the cross-section of the diffracted rays) is similar to the pixel size on the detector, and the detector has a point-spread function that is small compared to the pixel size, this may not be true. There are other complicating factors which may occur to the reader!

## Analysing the results of integration

Check graphs - they should vary smoothly without obvious discontinuities.

- Large changes in parameters may indicate problems with the crystal or instrument.
- Look at any images corresponding to discontinuities in the graphs.
- I/σ(I) at (high resolution limit+0.2Å) should be >1

Check any warnings issued by the program; it may be best to re-process after following the advice given (all warnings given by *Mosflm* are accompanied by suggestions on how to improve the processing).



Before going on to scaling the data, it is sensible to check that the integration has not thrown up any errors. In particular, examine any graphs that the integrating program has produced. They should all vary smoothly from image to image, without any sharp discontinuities.

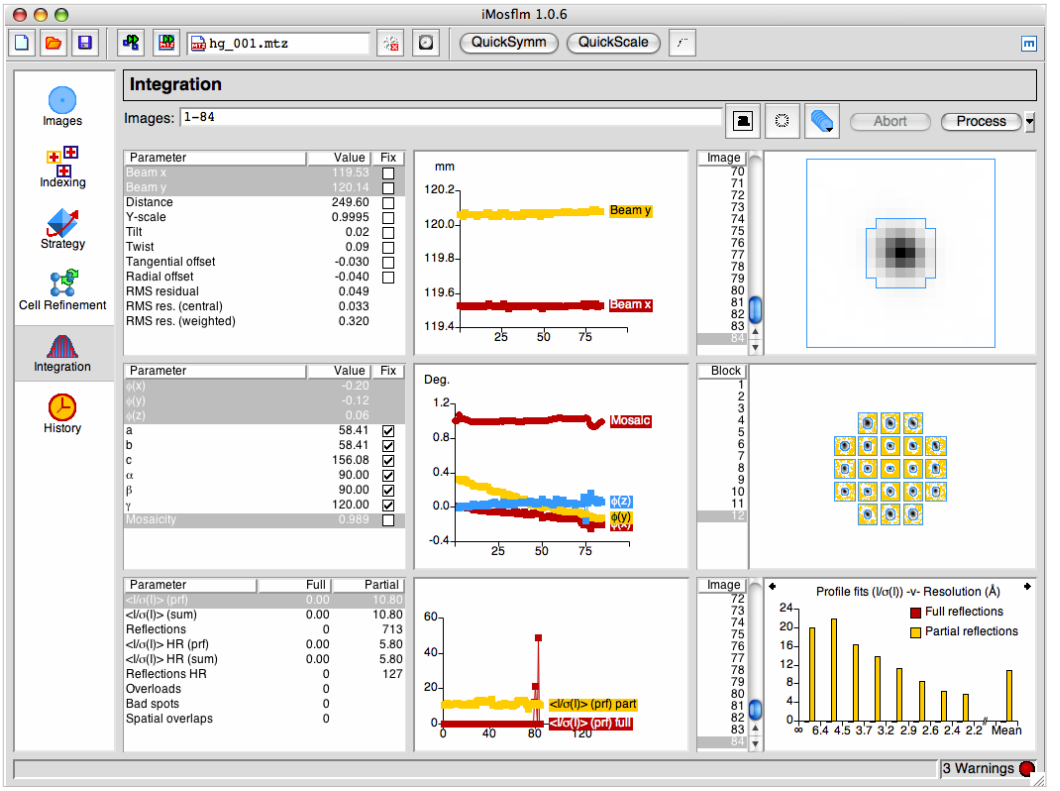
If there are discontinuities in the graphs, they often occur around the same images for different graphs. Look at any images in the region of the discontinuities and see if there is anything obviously wrong with them.

In the case that all the graphs look good until a certain point in the dataset, then the processing deteriorates, it is often an indication that too high a symmetry has been imposed on the integration, and the program cannot refine detector and/or crystal values sufficiently to keep the integration boxes well centred on the spots.

If the graphs corresponding to $I/\sigma(I)$ fall gradually to lower values towards the end of the dataset, it is usually an indication that the crystal is exhibiting radiation damage.

As a first check that the data have been integrated to their resolution limit, I make sure that the average I/σ(I) for the outermost but one resolution bin is at least 1; I usually find, particularly with fine phi-sliced data where there are no fully recorded reflections, that there is significant intensity (after scaling and merging) to around 0.2Å better than the results of integration itself suggest.

*Mosflm* will often issue several warnings at the end of processing. Each of these is accompanied by one or more suggestions (in the main "mosflm.lp" log file) to improve the data processing.




This is an example of data processing in Mosflm where things seem to have been okay except for the last few images. There is a dip in the mosaic spread and the mis-setting angles have jumped around image 75. Also, the I/sig(I) of the fully recorded reflections has jumped from 0 to ~30 - 50 for a couple of images (because the mosaic spread for these images is lower than the rotation angle, there are actually spots identified as fulls rather than partials – all other images only have partials).

In this case it seems that there is something “odd” about the images around image 75 – it is worthwhile looking at the images near here to see if there is any obvious reason for this problem.

## Scaling and merging

This is the next step following integration. It is important because -

- It attempts to put all observations on a common scale
- It attempts to make the data internally consistent, but systematic errors that are the same for symmetry-related reflections will remain.
- It provides the main diagnostics of data quality
  - were the data collection and data integration satisfactory?
  - are the data internally consistent?

Because of this diagnostic role, scale the data as soon as possible after (during) data collection, preferably while the crystal is still on the camera.

- Do not leave integration and scaling until you get home after a synchrotron visit - check the results of beamline autoprocessing immediately!

23/28

Pressing the "*QuickScale*" button in the Integration task in *iMosflm* runs default jobs of *Pointless*, *Aimless*, *cTruncate* and *Uniquefy* – this job is often sufficient to proceed further in the structural analysis with the default settings, but these can be changed in the "sort, scale and merge" settings under Processing options.

When run from *ccp4i2*, the *MTZ* file produced by *Mosflm* can be scaled and merged by in the *Data reduction* module.

## What is scaling?

An attempt to make symmetry related and duplicate measurements of a reflection equal by modelling the diffraction experiment, principally as a function of the incident and the diffracted beam directions in the crystal.

Scaling attempts to make the data internally consistent, by minimising the differences between the individual observations I and the weighted mean of all the symmetry-related equivalents of reflection I.

- However, systematic errors that are the same for symmetry-related reflections will remain.

24/28

Scaling is the process in which we try to minimise the difference between equivalent observations (*e.g.* symmetry mates, multiple observations of the same reflection) in the dataset.

Merging does two basic things:

It combines the parts of reflections spread over several images ("partially recorded reflections" or "partials") to form fully recorded reflections (or "fulls).

It combines fully recorded reflections that are equivalent by symmetry into single observations.

The second step may be omitted early in structure solution because some methods (and some programs, *e.g.* the SHELXC/D/E pipeline) give better results with unmerged reflections.

Together, scaling and merging give the best statistics about both data collection and data processing - far superior to the statistics from data integration alone, because they are calculated from the entire dataset. They should be good enough to allow the user to decide if it is necessary to collect more data from the same crystal - so should be performed while the crystal is still on the diffractometer if possible.

## Why are reflections on different scales?

Various physical factors lead to observed intensities being on different scales. Some geometric corrections are known, but others can only be determined from the data;

Factors related to
(1) the incident beam and the "camera"
(2) the crystal and the diffracted beam
(3) the detector

Scaling models should try to parameterise the experiment, so different experiments may require different models.

Understanding the effect of these factors allows sensible corrections and an understanding of what can go wrong

25/28

Geometrical factors like the *Lp* correction can be made prior to scaling, and should be known from the experimental set-up. The *Lp* correction is made up of

• The Lorentz factor (which can be thought of as related to the time a reciprocal lattice point takes to traverse the Ewald sphere, and is important for reflections close to the projection of the rotation axis on the image, where it tends to infinity)

•The polarisation of the X-ray beam (important at synchrotron sources)

Further correction factors from different physical sources are necessary following the integration step, including

(1) Factors related to the incident beam include things like variations in X-ray intensity, illuminated volume of the crystal (moving in and out of beam), variations in rotation speed

(2) Those factors related to the crystal include absorption and radiation damage

(3) Issues with the detector include non-linearity of response, "corner" problems (the corners of the modules in both CCD and PADs (like Pilatus) respond differently to the rest of the module, "dead" and "hot" pixels, regions between modules. Also, it is hard for a scaling program to treat reflections properly if they have been masked by external objects, *e.g.* beamstops or cryostreams (how does the program know which measurements have been affected?)..

## *Aimless*

A program to scale and merge reflections measured by an integration program. It reads *multirecord* reflection files output from integration programs

- Can read *MTZ* files directly from *Mosflm* (no *need* for *sortmtz* or *Pointless*)
- Can read *HKL* files from *XDS*, unmerged *.sca* files from *HKL*, *etc*.
- Merges the parts of partial reflections together
- Puts data onto a common scale
- Merges each set of symmetry equivalent reflections into single observations
- Writes an *MTZ* file that can be read by other *CCP4* programs
- Produces the classic "Table 1" which contains useful statistics about the quality of the dataset

26/28

Scaling and merging in *CCP4* is performed by the program *Aimless; i*t reads reflection files that have separate entries for each individual observation. 2D integration programs (like *Mosflm* and *Denzo*) write reflection files containing partials, and *Aimless* merges the separate partials together to form fully recorded observations. 3D integration programs (like *Dials*, *XDS*, *SAINT* and *d*Trek*) form fully recorded reflections as part of the integration process, so this does not need to be done by any subsequent program.

However, multiple observations of symmetry equivalent reflections do still need to be merged into single records for most current programs, and this is not done by any of the integration programs, either 2D or 3D. This step is performed by *Aimless*, after the data have been put onto a common scale.

*Aimless* provides the most useful diagnostics of the data processing; strictly speaking, the statistics produced (and reported in the classic "Table 1", such as *Rmeas, CC½, etc.*) are measures of the *internal consistency* of the data; they are *not* directly measures of the quality of the data. However, they are the best measure we have at this stage, and internally consistent data sets are often also of high quality.

## *cTruncate*

Takes the *MTZ* file written by *Aimless* (which contains $|F^2|$ for each reflection) and

- analyses scaled data according to an expected physical model
- gives statistics on intensity distribution - *e.g.*
  - Wilson statistics
  - twinning analyses
- writes an *MTZ* file containing $|F|$ values for use in subsequent CCP4 programs (for structrure solution, refinement, analysis...)

27/28

From equation (1) in slide 2, we can see that it should be possible to obtain $|F|$ straightforwardly once we have the scaled intensities :

$$\left|F_{hkl}\right| = \left(\frac{KI_{hkl}}{Lp}\right)^{1/2}$$

However, measured intensities have an associated error $\sigma(I)$, which may be larger than $I$ for weak reflections; small intensities can be recorded with negative values. Ctruncate estimates the value of $F$ based on the knowledge that the intensities physically cannot be negative (French & Wilson).This was addressed in S. French and K. Wilson, *Acta Cryst*. **A34**, 517-525 (1978) "On the treatment of negative intensity observations", and is based on using the average intensity in each resolution range, which gives the prior probability. The estimated value of $F$ is given by

$$E(F;I,\sigma(I)) = \int_0^\infty F\, p(I;J,\sigma(I))\, p(J)\, dJ$$

$E(F ; I,\sigma(I))$ is the estimate of $F$ given $I$ and $\sigma(I)$, *etc.*

## Finally

Remember -

- Don't expect software to correct for a badly performed experiment
- Take the time to look at your images and the results of integration and scaling
- Scaling and merging provide the best statistics on the quality of your data