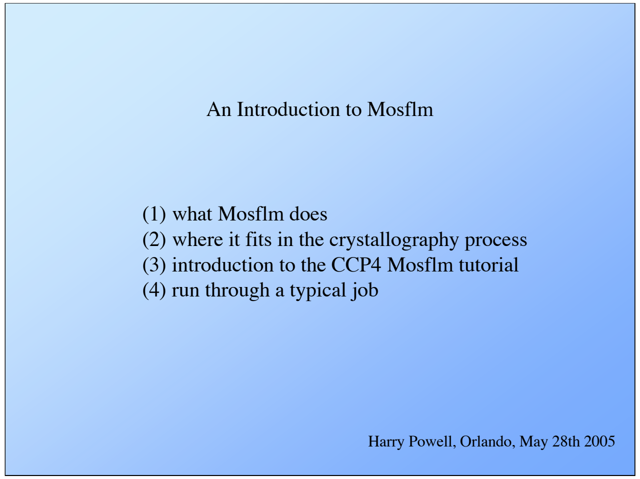


An Introduction to Mosflm

(1) what Mosflm does
(2) where it fits in the crystallography process
(3) introduction to the CCP4 Mosflm tutorial
(4) run through a typical job

Harry Powell, Orlando, May 28th 2005

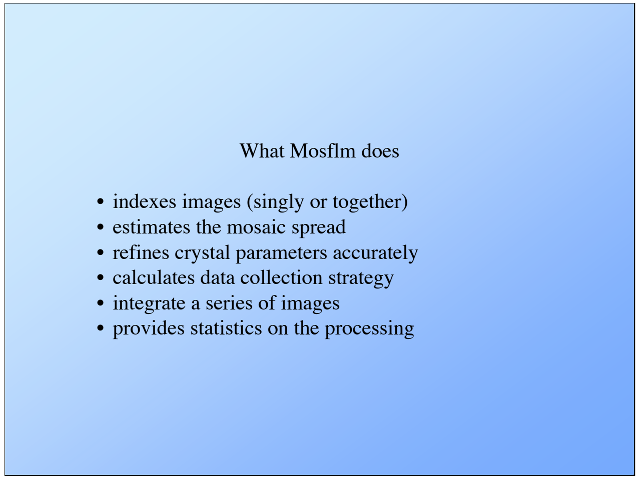
`Mosflm` is available free of charge to the end user; it has all necessary functionality to process diffraction images obtained on a wide variety of different detectors, and runs on all common UNIX-based computers used in crystallographic laboratories. A version which will run under Windows is currently under development.

The primary source of information about the program is

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

Advice can be obtained from the program's authors;

    andrew@mrc-lmb.cam.ac.uk
    harry@mrc-lmb.cam.ac.uk



What Mosflm does

- indexes images (singly or together)
- estimates the mosaic spread
- refines crystal parameters accurately
- calculates data collection strategy
- integrate a series of images
- provides statistics on the processing

`Mosflm` is more than a program for integrating diffraction images.

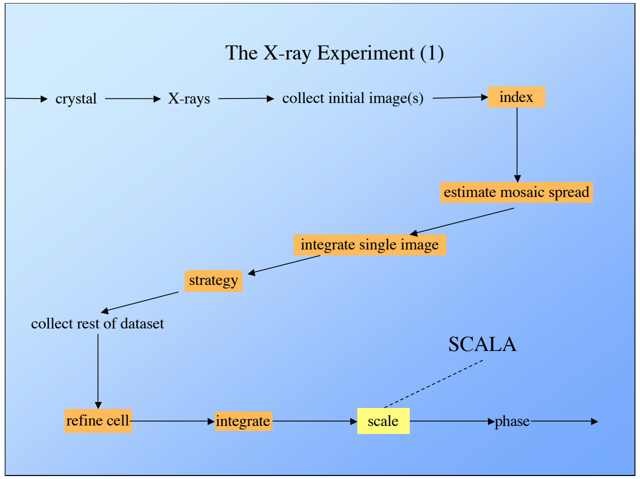
It includes two different component programs for autoindexing images - a method based on difference vectors (`REFIX`) and a method based on the Fast Fourier Transform (`DPS`); the latter is used by default as it is extremely reliable, but occasionally the alternative procedure may crack an otherwise intractable problem.

A process based on iterative integration of an image using incremental values for the mosaic spread has been implemented which gives a good estimate of the mosaic spread.

Accurate values for cell parameters and the mosaic spread are determined from postrefinement in `Mosflm` itself. Most other integration programs leave this for other post-processing programs.

Data collection strategies can be determined rapidly in `Mosflm` once the orientation and mosaic spread of the crystal are known. Information from previously collected datasets can be included in the calculation - this may be of use in cases where crystal decay precludes collection of a complete dataset from a single crystal.

Of course, it also integrates images and provides statistics on the processing - which can be viewed in a text log file or using the CCP4 program `loggraph`.



The X-ray Experiment (1)

crystal → X-rays → collect initial image(s) → index → estimate mosaic spread → integrate single image → strategy → collect rest of dataset → refine cell → integrate → scale → phase

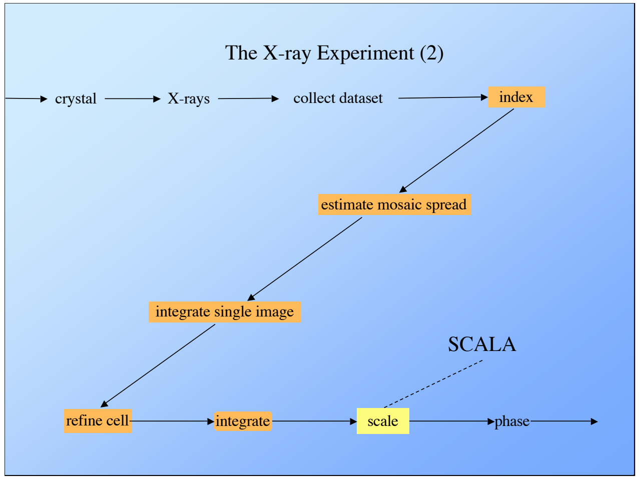
SCALA

There are two different "typical' ways of approaching data integration. The first (illustrated here) is where a crystal is mounted on a diffractometer and has to be characterized. The sequence shown is a careful approach to data collection and processing of an unknown sample; the orange boxes are steps performed by `Mosflm`.

Once an initial image has been collected, it makes good sense (in most cases) to characterize it by indexing and estimating the mosaic spread, then determining the optimum strategy before trying to collect the full dataset. At some point early in data processing the user should try to determine the point group, since this affects the strategy calculations. Various tools are available for this, *e.g.* `Pointless` by Phil Evans can give a good indication with a partial dataset.

Integrating a single image gives good information about the maximum resolution to which the crystal diffracts under the current experimental conditions, and so should be performed prior to running the strategy option (experience shows that, following scaling, there is often usable information at ~0.1 - 0.2Å higher resolution than the limit determined from integrating a single image). `Mosflm` (from version 6.2.5) will also be able to output files suitable for the analytical strategy determination program `BEST` (Popov & Bourenkov), which can give extra information on suitable exposure times and maximum likely resolution.

The second common approach is to take an existing dataset and process it (illustrated



The X-ray Experiment (2)

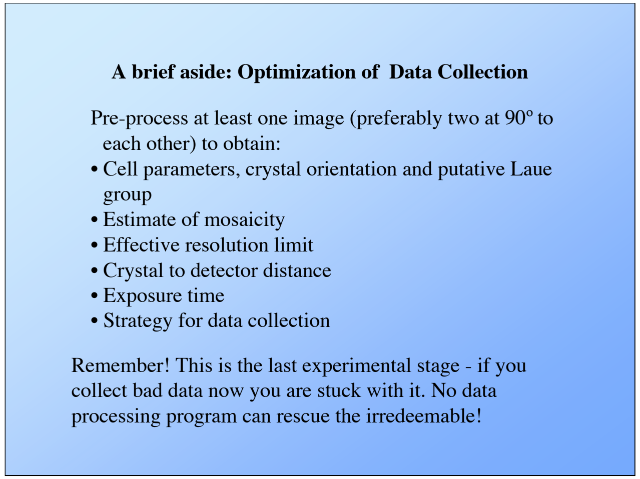
crystal → X-rays → collect dataset → index → estimate mosaic spread → integrate single image → refine cell → integrate → scale → phase

SCALA

in the next slide). The major difference between the two approaches is that there is probably no need to run the strategy option in the latter case, though it can still be somewhat comforting to run the option anyway and learn that the data have been collected correctly!

In either case, `Mosflm` integrates the images and writes the Lorentz and polarization corrected intensities to a multi-record MTZ file. Both profile fitted and summation integration intensities are written to the file, and both can be used by `SCALA` to produce the best scaled set of data.

Each image in a dataset probably has its recorded intensities on a different scale for a variety of reasons, *e.g.* variation in intensity of the incident radiation, absorption of diffracted rays, change in the diffracting volume, etc. In the process of merging symmetry equivalent reflections from different images, we need to take this into account and apply scales accordingly. Following a step which sorts the reflections in the output MTZ file, this process is performed by `SCALA`. The best statistics on an integration run are provided by scaling, so this should be performed as soon as possible after data collection so that any experimental problems can be addressed while the crystal is still mounted.



A brief aside: Optimization of Data Collection

Pre-process at least one image (preferably two at 90° to each other) to obtain:

- Cell parameters, crystal orientation and putative Laue group
- Estimate of mosaicity
- Effective resolution limit
- Crystal to detector distance
- Exposure time
- Strategy for data collection

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

It is always worthwhile spending some time prior to the full data collection to determine sensible parameters for the data collection. For example;

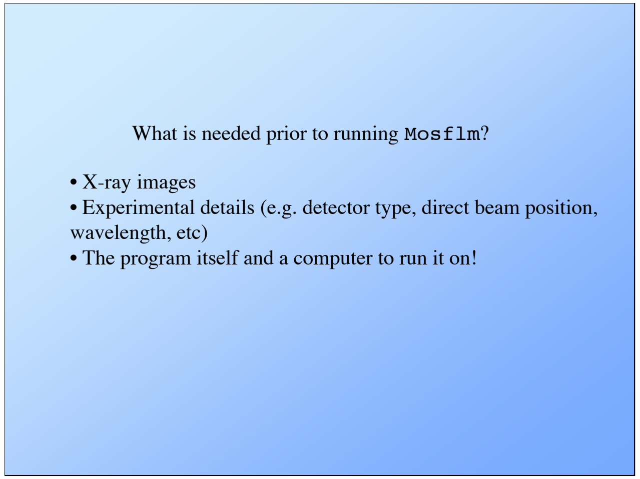
- are you using the full area of the detector?

does useful diffraction go beyond the edge of the detector (should it be moved closer)? Does it stop halfway to the edge (move the detector further away)?

- check for overloads - are there a lot (reduce exposure or perhaps only process higher resolution spots)? Are you using the full dynamic range of the detector (perhaps increase exposure time)? Consider a low and a high-resolution pass. Increase or decrease the exposure time.
- is there significant overlap of spots (decrease oscillation angle)?
- check that the predicted spots do coincide with their positions on the image(s); is your initial estimate of the mosaicity realistic?

Remember to use prior information! If you have experience of your particular sample or experimental setup, use your knowledge. If something looks odd, investigate it.

Whatever integration program you are using, there will be an option (or an external program) which can calculate the optimum data collection strategy for you. The two most important pieces of information which any strategy program will give are (a) the maximum oscillation angle to avoid overlap and (b) the best start and end angles.



What is needed prior to running `Mosflm`?

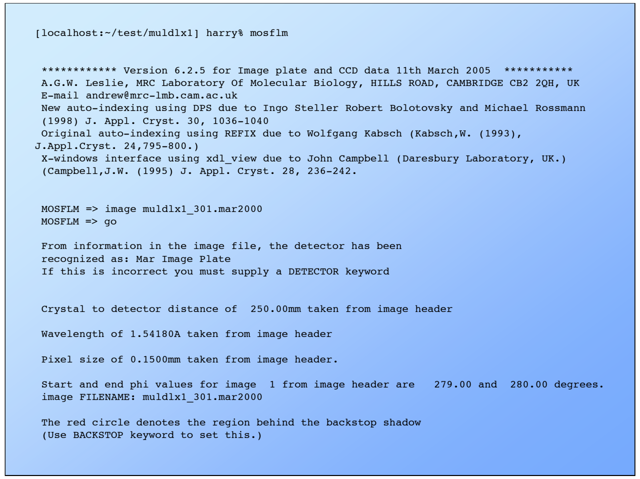
- X-ray images
- Experimental details (e.g. detector type, direct beam position, wavelength, etc)
- The program itself and a computer to run it on!

Before running `Mosflm`, the user needs to have access to oscillation images and a copy of the program (and the CCP4 software libraries) which will run on their computer. Pre-compiled binaries which will run “out-of-the-box” are available from our web- and ftp sites, and a “build-it-yourself” version is also available. Full installation details are included with the downloads.

However, it should be noted that the current version (6.2.4) is only available as part of the CCP4 suite; this is because the underlying library functions in CCP4 were changed, and these new functions are required by version 6.2.4.

The computer itself must be UNIX based (but a version of `Mosflm` which will run under Windows is under development) and have at least 32Mb RAM and 128 Mb swap space available. A screen resolution of 1280 x 1024 pixels is recommended (a small-screen version is available which is suitable for 1024 x 768 displays).

Many experimental details are written to image headers, and `Mosflm` can read and use this information. All values supplied in the headers can be over-ridden by the user during processing. Further, `Mosflm` will determine suitable processing parameters based on analysis of the images during the processing itself, *e.g.* integration box dimensions.



```
[localhost:~/test/muldlx1] harry% mosflm


 ************ Version 6.2.5 for Image plate and CCD data 11th March 2005  ***********
 A.G.W. Leslie, MRC Laboratory Of Molecular Biology, HILLS ROAD, CAMBRIDGE CB2 2QH, UK
 E-mail andrew@mrc-lmb.cam.ac.uk
 New auto-indexing using DPS due to Ingo Steller Robert Bolotovsky and Michael Rossmann
 (1998) J. Appl. Cryst. 30, 1036-1040
 Original auto-indexing using REFIX due to Wolfgang Kabsch (Kabsch,W. (1993),
J.Appl.Cryst. 24,795-800.)
 X-windows interface using xdl_view due to John Campbell (Daresbury Laboratory, UK.)
 (Campbell,J.W. (1995) J. Appl. Cryst. 28, 236-242.


 MOSFLM => image muldlx1_301.mar2000
 MOSFLM => go

 From information in the image file, the detector has been
 recognized as: Mar Image Plate
 If this is incorrect you must supply a DETECTOR keyword


 Crystal to detector distance of  250.00mm taken from image header

 Wavelength of 1.54180A taken from image header

 Pixel size of 0.1500mm taken from image header.

 Start and end phi values for image  1 from image header are   279.00 and  280.00 degrees.
 image FILENAME: muldlx1_301.mar2000

 The red circle denotes the region behind the backstop shadow
 (Use BACKSTOP keyword to set this.)
```

To run `Mosflm`, you need to have the program in your path and you need to know what it's called on your system - often it's "`ipmosflm`" rather than "`mosflm`".

If you run `Mosflm` and encounter difficulties and need help, make sure that you report which version of the program you are running - this is the first line of output to the screen, and also the first line in the standard log file, usually called `mosflm.lp`.

The first people to ask for help are Andrew Leslie (e-mail above) and Harry Powell (`harry@mrc-lmb.cam.ac.uk`); usually it's much quicker to ask them directly than to post questions on the CCP4 bulletin board!

In this example, using a Mar image plate, the detector type is automatically detected so no `DETECTOR` keyword is necessary. From version 6.2.3, other detector types (*e.g*. ADSC, Rigaku, DIP, Mar CCD), are also recognized from the data in the image header, but it's also possible to specify by using the `DETECTOR` keyword.

The `GO` keyword in conjunction with `IMAGE` tells the program to use the X-windows GUI for further processing.

The header information is read; in this case the crystal to detector distance, the wavelength, the pixel size of the detector, and the oscillation range of this image.

live demo here

Once the header information has been read and appropriate parameters set up, the image file is opened a second time and the image data read. Following this, the X-window GUI is displayed on the screen and further processing can be performed.

`Mosflm` checks to see whether the image is the right size for the named detector, and also whether the data has been written on a big-endian or little-endian computer. Byte swapping is performed automatically.

Initial processing of images (*i.e.* indexing, estimating mosaicity, refining the cell etc.) is best done using the GUI; since there is a computational overhead associated with updating the GUI, integration is probably best performed as a background job (using the CCP4i `Mosflm` task or a shell script); typically, integration without the `Mosflm` GUI is around 3 or 4 times faster than with it.

`Mosflm` can write a "save file" which contains details of the experiment to date and can be read by the CCP4i `Mosflm` task or used for input to a background job.

A fully-worked tutorial;

$CCP4/examples/tutorial/html/dataproc-tutorial.html

A tutorial on running Mosflm is included with the CCP4 suite, in the file

`$CCP4/examples/tutorial/html/dataproc-tutorial.html`

The sample dataset is available from the CCP4 ftp site; note, however, that the images are in the ftp directory /`autostruct/testdata/mosflm`, not /pub/`autostruct/testdata/mosflm` as stated in the current (May 2005) version of the tutorial. Note that the dataset is contained in a 128Mb tar file.

Checking the output

There are two useful log files;

- `SUMMARY`; this is of most use when viewed with the CCP4 graph viewer `LOGGRAPH`, as it contains graphs of parameters which have varied through the data processing.
- `mosflm.lp`; this can be very large, and contains a complete record of the experiment.

Things to check in `SUMMARY`

Quickly check through all the graphs to make sure that there are no sharp discontinuities, and that all the graphs vary smoothly through the data processing run. Some parameters will not change at all (e.g. cell parameters should not be refined during the integration run), while others will drift slightly.

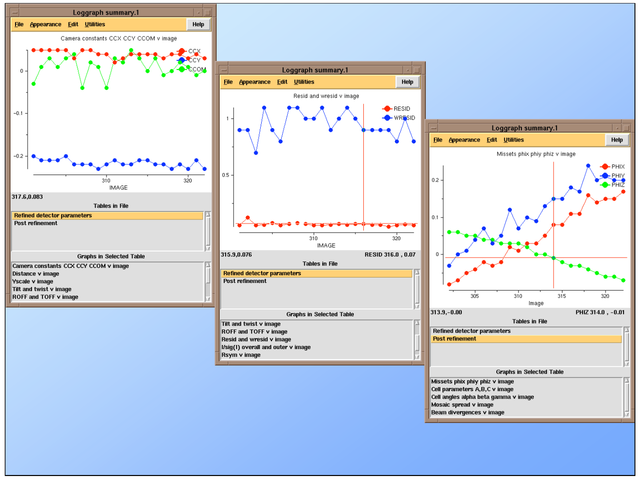
Check carefully that `YSCALE` is not changing by more than 1 part in 1000. Likewise, `DISTANCE` should be constant for a properly centred crystal; a large random variation in either of these indicates that postrefinement was not performed correctly - the cell parameters may be wrong, and the machine parameters may be unreliable. A regular variation (following an approximate sine curve) in `DISTANCE` indicates that the crystal is precessing about the true axis of rotation.

Check the `Residual` and `Weighted Residual`; both should remain roughly constant, but the weighted residual should be ~1.0 for a laboratory source; values higher than this indicate an incorrect `GAIN` for the detector, while smaller values are common for data collected at synchrotrons.

Check the I/sig(I) graphs; is the crystal dying? Does it diffract at the highest resolution specified in processing?

Things to check in `mosflm.lp`

Provided that there are no apparent problems from examining the `SUMMARY` file, the main thing to check in `mosflm.lp` is the list of warnings at the end of the file. Mosflm provides many warnings during its processing, but these should be taken as



they are intended, *i.e.* indications that some parameter has refined to a value outside its nominal limits and that Mosflm has adapted the processing to take account of this.

The list of warnings at the end of `mosflm.lp,` however, summarize the problems that may affect the data quality. Each warning is accompanied by an analysis of the problem, and a suggested course of action to remedy the problem. One reason why the corrective action is not taken part way through the integration is because this would mean that the processing might change significantly between two adjacent frames or at an inappropriate point (*e.g*. several frames after the point where the problem originally arose).

These are examples from `loggraph` of some of the plots from the `SUMMARY` file. The main point to draw from these is that refinement seems stable and no parameters are swinging from one value to another.

The "residual" plot in the center shows the use of the cursor in `loggraph` to pick out a particular value from the graph; in this instance, the r.m.s. residual for spots on image 316 is 0.07mm (*cf* pixel size of 0.15mm).

The apparent drift in the missetting angles is probably due to the rotation axis not being exactly perpendicular to the X-ray beam.

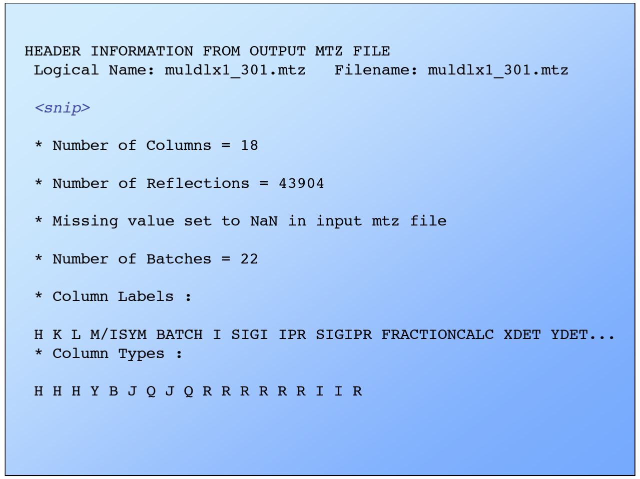
Further Processing with Mosflm

(1) what do you do about the warnings?
(2) what if there are real problems?
(3) the `ccp4i Mosflm` task

We will be going over the output produced by `Mosflm` and decoding what the warning messages mean, and how to alter the input parameters so as to improve the overall data processing.

We will also look at a real case of how to deal with problems arising when the default inputs are incorrect, and finish by introducing the ccp4i Mosflm task.

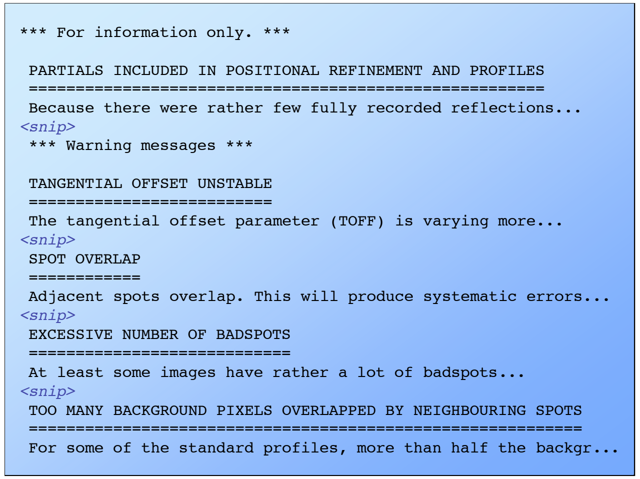
It should be remembered that `Mosflm` will do its best to determine appropriate parameters for the integration process, but occasionally fails to make the best choice. Sometimes this is because the user has supplied a value erroneously, but often it is because the images being processed are, in some way, sub-optimal. There may be diffuse scatter, multiple lattices, excessively close diffraction spots, etc.



```
HEADER INFORMATION FROM OUTPUT MTZ FILE
 Logical Name: muldlx1_301.mtz   Filename: muldlx1_301.mtz

 <snip>

 * Number of Columns = 18

 * Number of Reflections = 43904

 * Missing value set to NaN in input mtz file

 * Number of Batches = 22

 * Column Labels :

 H K L M/ISYM BATCH I SIGI IPR SIGIPR FRACTIONCALC XDET YDET...
 * Column Types :

 H H H Y B J Q J Q R R R R R R I I R
```

Details of what has been written to the MTZ file are printed out on screen at the end of processing; it's always worth checking this before looking at the `mosflm.lp` file. Things to look out for are the number of frames processed (prior to version 6.2.3, `Mosflm` quietly shut down if it encountered an error and gave no clue that it had done so!), that the resolution limit is correct, and that there are about the right number of reflections stored in the file. The MTZ file itself can be examined conveniently using the ccp4i GUI, or with one of the utility programs supplied with the suite - either `mtzdump` or `mtzdmp`.

Also, as mentioned earlier, it's well worthwhile examining the `SUMMARY` file using the `loggraph` utility.



```
*** For information only. ***

 PARTIALS INCLUDED IN POSITIONAL REFINEMENT AND PROFILES
 =======================================================
 Because there were rather few fully recorded reflections...
<snip>
 *** Warning messages ***

 TANGENTIAL OFFSET UNSTABLE
 ==========================
 The tangential offset parameter (TOFF) is varying more...
<snip>
 SPOT OVERLAP
 ============
 Adjacent spots overlap. This will produce systematic errors...
<snip>
 EXCESSIVE NUMBER OF BADSPOTS
 ============================
 At least some images have rather a lot of badspots...
<snip>
 TOO MANY BACKGROUND PIXELS OVERLAPPED BY NEIGHBOURING SPOTS
 ===========================================================
 For some of the standard profiles, more than half the backgr...
```

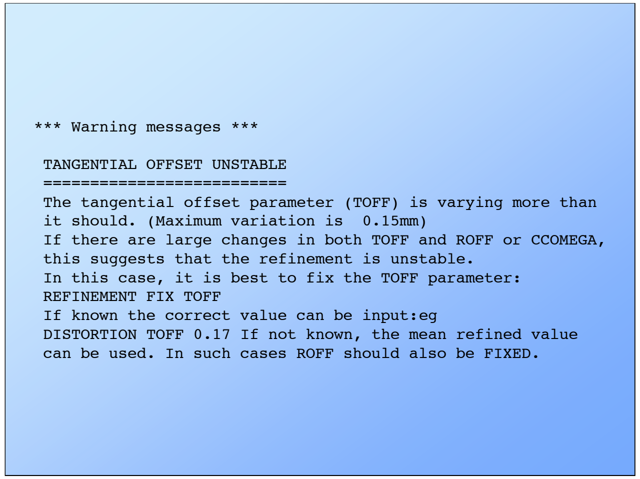
Near the end of the mosflm.lp log file is a list of information and warning messages; if this is missing it is nothing to worry about - in fact the opposite! If there are no informational or warning messages, it suggests that there is little that can be done to improve the data processing.

If they exist, it is important to take heed of the warnings, in particular; changing some of the processing parameters in the light of this information can lead to considerably improved processing.

The warnings above were generated following the processing of the 22 images used in the test dataset in the first talk. They indicate that the integration has proceeded reasonably well but that there are a few points that should be examined. It may be worthwhile re-processing the data with modified input to see if the warnings can be removed and the dataset quality improved.

Before any action is taken to correct the behaviour which gave rise to the warnings, it's a good idea to check through *all* of them and decide which is the most important. Some of them may be due to incorrect processing parameters while others may be symptomatic of detector error. The nature of data integration also means that many parameters are correlated, so an error in one part of input may give rise to warnings about others.

For reference, the full text of the above errors is included in Appendix A.

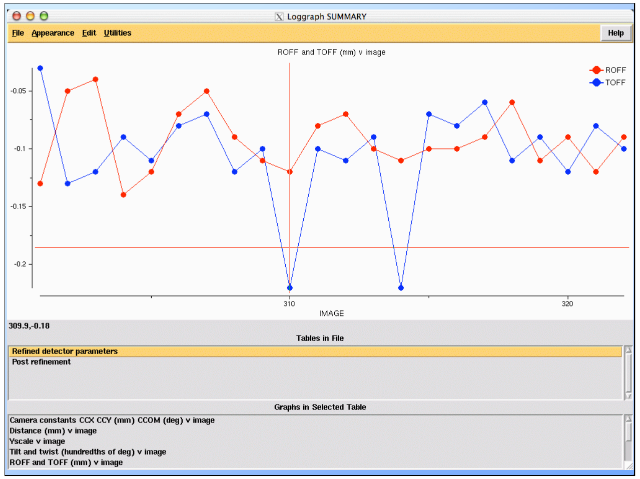


```
*** Warning messages ***

 TANGENTIAL OFFSET UNSTABLE
 ==========================
 The tangential offset parameter (TOFF) is varying more than
 it should. (Maximum variation is  0.15mm)
 If there are large changes in both TOFF and ROFF or CCOMEGA,
 this suggests that the refinement is unstable.
 In this case, it is best to fix the TOFF parameter:
 REFINEMENT FIX TOFF
 If known the correct value can be input:eg
 DISTORTION TOFF 0.17 If not known, the mean refined value
 can be used. In such cases ROFF should also be FIXED.
```

There is helpful advice printed alongside each warning. It’s well worthwhile checking against the loggraph plot to see if this is a smoothly varying value or if the plot has obvious discontinuities.

In this case, because there are no coincident changes in ROFF and CCOMEGA, the indication is that the refinement is stable and there is a real physical cause for this problem. The maximum variation is 0.15mm, and it is no coincidence that this is the pixel size for a Mar image plate; the locking mechanism on the detector appears to be faulty and has offset the scanner by one pixel for two of the images.

If you do the experiment and FIX ROFF and TOFF, you will find that the missetting angles (which currently refine stably) undergo a large change for the same two images that are involved here. This is a further indication that this is a real effect and not an artefact of processing.



309.9,-0.18

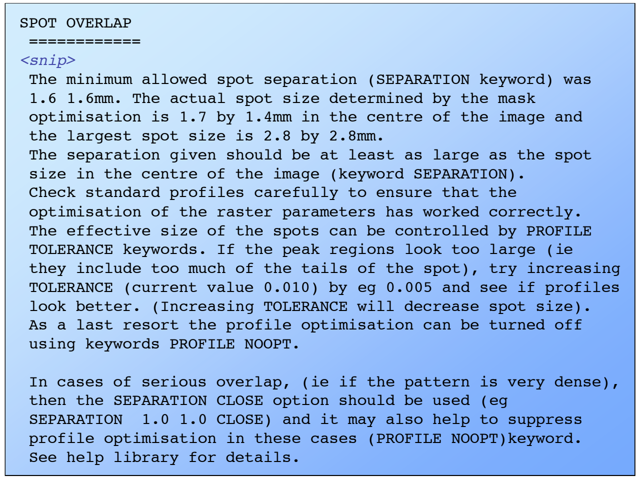
Tables in File

Refined detector parameters
Post refinement

Graphs in Selected Table

Camera constants CCX CCY (mm) CCOM (deg) v image
Distance (mm) v image
Yscale v image
Tilt and twist (hundredths of deg) v image
ROFF and TOFF (mm) v image

The `loggraph` output shows plainly that there are two images with TOFF in error by 0.15mm.

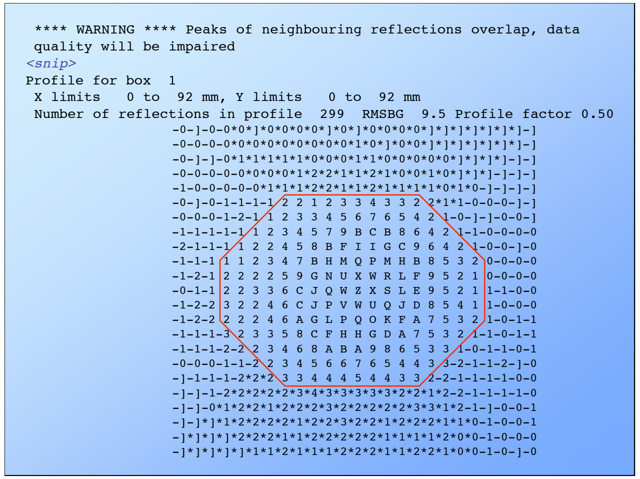


```
SPOT OVERLAP
 ============
<snip>
 The minimum allowed spot separation (SEPARATION keyword) was
 1.6 1.6mm. The actual spot size determined by the mask
 optimisation is 1.7 by 1.4mm in the centre of the image and
 the largest spot size is 2.8 by 2.8mm.
 The separation given should be at least as large as the spot
 size in the centre of the image (keyword SEPARATION).
 Check standard profiles carefully to ensure that the
 optimisation of the raster parameters has worked correctly.
 The effective size of the spots can be controlled by PROFILE
 TOLERANCE keywords. If the peak regions look too large (ie
 they include too much of the tails of the spot), try increasing
 TOLERANCE (current value 0.010) by eg 0.005 and see if profiles
 look better. (Increasing TOLERANCE will decrease spot size).
 As a last resort the profile optimisation can be turned off
 using keywords PROFILE NOOPT.

 In cases of serious overlap, (ie if the pattern is very dense),
 then the SEPARATION CLOSE option should be used (eg
 SEPARATION  1.0 1.0 CLOSE) and it may also help to suppress
 profile optimisation in these cases (PROFILE NOOPT)keyword.
 See help library for details.
```

There is a lot of information in the second warning, but it may not be as serious as it seems. Bear in mind that the minimum spot separation reported here is almost the same size as the largest spot size - so the separation parameter worked out by the program is probably not too far from ideal.

It is always worth checking the standard profiles in any case; they may be too large in this example, so modifying PROFILE TOLERANCE may help. Under very rare circumstances the optimisation can be turned off, but this is *absolutely* a last resort if all else fails.

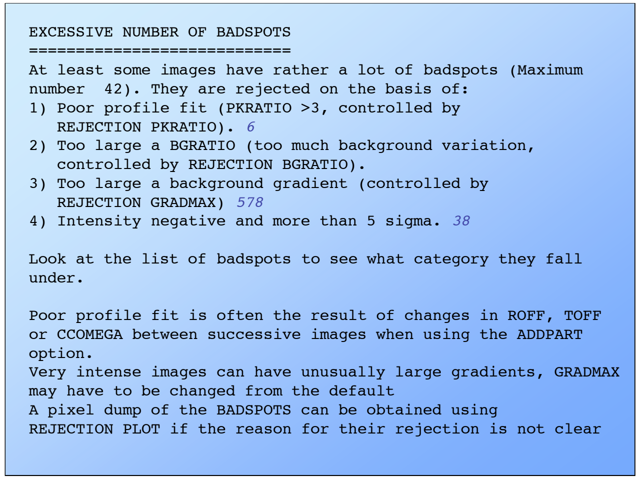
The clue to how we should proceed here is given in the second paragraph; are the spots really close?



```
 **** WARNING **** Peaks of neighbouring reflections overlap, data
 quality will be impaired
<snip>
Profile for box  1
 X limits    0 to  92 mm, Y limits   0 to  92 mm
 Number of reflections in profile  299  RMSBG  9.5 Profile factor 0.50
                  -0-]-0-0*0*]*0*0*0*0*]*0*]*0*0*0*0*]*]*]*]*]*]-]
                  -0-0-0-0*0*0*0*0*0*0*1*0*]*0*0*]*]*]*]*]*]-]
                  -0-]-]-0*1*1*1*1*1*0*0*0*1*1*0*0*0*0*0*]*]*]-]-]
                  -0-0-0-0-0*0*0*0*1*2*2*1*1*2*1*0*0*1*0*]*]*]-]-]-]
                  -1-0-0-0-0-0*1*1*1*2*2*1*1*2*1*1*1*1*0*1*0-]-]-]-]
                  -0-]-0-1-1-1-1 2 2 1 2 3 3 4 3 3 2 2*1*1-0-0-0-]-]
                  -0-0-0-1-2-1 1 2 3 3 4 5 6 7 6 5 4 2 1-0-]-]-0-0-]
                  -1-1-1-1-1 1 2 3 4 5 7 9 B C B 8 6 4 2 1-1-0-0-0-0
                  -2-1-1-1 1 2 2 4 5 8 B F I I G C 9 6 4 2 1-0-0-]-0
                  -1-1-1 1 1 2 3 4 7 B H M Q P M H B 8 5 3 2 0-0-0-0
                  -1-2-1 2 2 2 2 5 9 G N U X W R L F 9 5 2 1 0-0-0-0
                  -0-1-1 2 2 3 3 6 C J Q W Z X S L E 9 5 2 1 1-1-0-0
                  -1-2-2 3 2 2 4 6 C J P V W U Q J D 8 5 4 1 1-0-0-0
                  -1-2-2 2 2 2 4 6 A G L P Q O K F A 7 5 3 2 1-0-1-1
                  -1-1-1-3 2 3 3 5 8 C F H H G D A 7 5 3 2 1-1-0-1-1
                  -1-1-1-2-2 2 3 4 6 8 A B A 9 8 6 5 3 3 1-0-1-1-0-1
                  -0-0-0-1-1-2 2 3 4 5 6 6 7 6 5 4 4 3 3-2-1-1-2-]-0
                  -]-1-1-1-2*2*2 3 3 4 4 4 5 4 4 3 3 2-2-1-1-1-1-0-0
                  -]-]-1-2*2*2*2*2*3*4*3*3*3*3*3*2*2*1*2-2-1-1-1-1-0
                  -]-]-0*1*2*2*1*2*2*2*3*2*2*2*2*2*3*3*1*2-1-]-0-0-1
                  -]-]*]*1*2*2*2*2*1*2*2*3*2*2*1*2*2*2*1*1*0-1-0-0-1
                  -]*]*]*]*2*2*2*1*1*2*2*2*2*2*1*1*1*1*2*0*0-1-0-0-0
                  -]*]*]*]*]*1*1*2*1*1*1*2*2*2*1*1*2*2*1*0*0-1-0-]-0
```

The spot profiles for different areas of the detector (by default, 9 profiles if the high resolution limit is lower than 2.5Å, 25 if the data is to higher resolution) are printed in the `mosflm.lp` log file. The red octagon has been superimposed here merely to emphasize the spot measurement area. The following warning and information precede the spot profiles:

```
**** WARNING **** Peaks of neighbouring reflections overlap, data
quality will be impaired

In the following profiles, background pixels are indicated by a "-"
sign and rejected background pixels (due to overlap by a neighbouring
spot) are indicated by a "*". Pixels with negative values are denoted
by a "]".
Note that a background pixel is rejected if it is overlapped by a
neighbouring spot for ANY reflection in that area of the detector.
If too many background pixels are being rejected try the SEPARATION
CLOSE keywords (see help library)

If SEPARATION CLOSE is used, the warning changes accordingly.
```



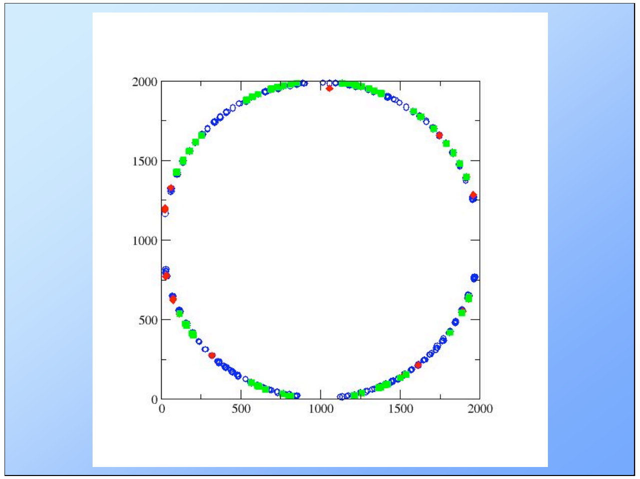
```
EXCESSIVE NUMBER OF BADSPOTS
============================
At least some images have rather a lot of badspots (Maximum
number  42). They are rejected on the basis of:
1) Poor profile fit (PKRATIO >3, controlled by
   REJECTION PKRATIO). 6
2) Too large a BGRATIO (too much background variation,
   controlled by REJECTION BGRATIO).
3) Too large a background gradient (controlled by
   REJECTION GRADMAX) 578
4) Intensity negative and more than 5 sigma. 38

Look at the list of badspots to see what category they fall
under.

Poor profile fit is often the result of changes in ROFF, TOFF
or CCOMEGA between successive images when using the ADDPART
option.
Very intense images can have unusually large gradients, GRADMAX
may have to be changed from the default
A pixel dump of the BADSPOTS can be obtained using
REJECTION PLOT if the reason for their rejection is not clear
```

If there are many bad spots, it indicates that there is something systematically wrong with the integration; a good starting point is to use a shell script to strip out the bad spots from mosflm.lp and see if there is anything about the parts of the image that they are from; very often, spots with a bad background plane gradient are close to the edge of shadows caused by the backstop or cryostream nozzle.

e.g.

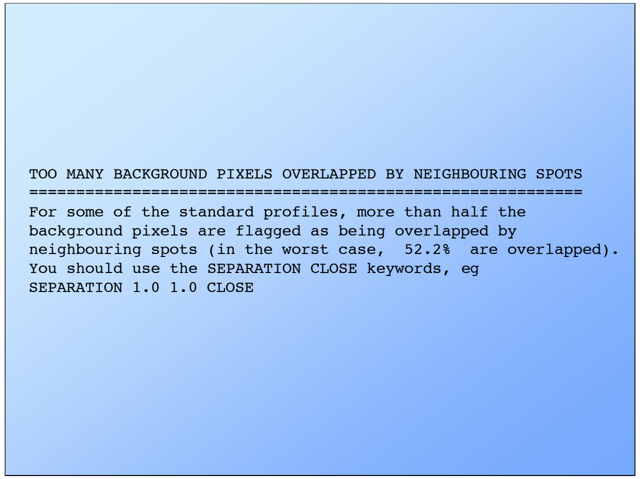
```
grep ‘Background plane gradient too steep’ mosflm.lp | sed ‘s/-/ -/g’
| awk ‘{print $7,$8}’ > plotfile
```

This will copy the pixel co-ordinates for each bad spot with the background plane too steep to a file, which can then be plotted with `xmgr` or `gnuplot`.



The blue circles correspond to the spots which lie on a bad background gradient, the red diamonds to those spots with an intensity which is too negative, and green squares those which fail both tests. Quite plainly, all the bad spots in this dataset are at the edge of the detector, and at this point we should remember that we did not set the high resolution limit! The resolution limit calculated by `Mosflm` based on the crystal to detector distance, detector size and wavelength is slightly overoptimistic in this case.

We can therefore remove all warnings about these bad spots by *not* integrating them; setting the resolution limit to ~3.00Å would probably be adequate, but an alternative would be to reject any spots which include a pixel below the noise level of the background - in this case, `NULLPIX 250` would work.



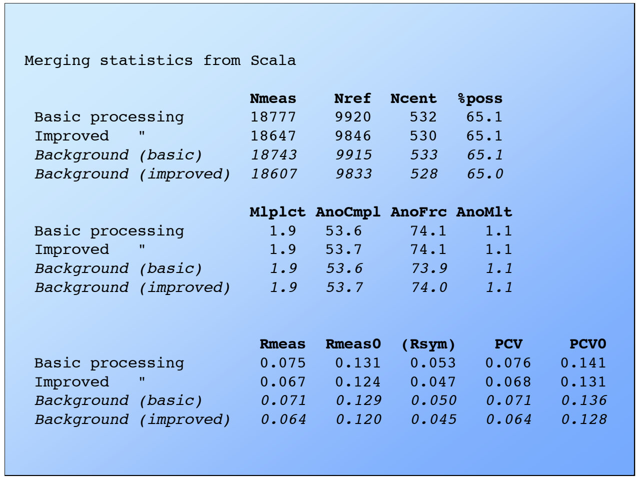
```
TOO MANY BACKGROUND PIXELS OVERLAPPED BY NEIGHBOURING SPOTS
===========================================================
For some of the standard profiles, more than half the
background pixels are flagged as being overlapped by
neighbouring spots (in the worst case,  52.2%  are overlapped).
You should use the SEPARATION CLOSE keywords, eg
SEPARATION 1.0 1.0 CLOSE
```

The final warning answers the question that we asked a couple of pages ago. We are now in a position to modify our command file and see if we can eliminate all the warnings in a sensible way; we will add these two lines to the “integrate” file created from the initial processing.

```
SEPARATION CLOSE
NULLPIX 250
```

The `NULLPIX` command is preferred here as it will only exclude those spots which are affected by the null background close to the detector edge; setting a resolution limit in this case could exclude some real spots which are close to the edge but not actually on it.

When this job is run, the only warning that arises is due to the unstable TOFF, and that is probably a reason to call the Mar engineer!



```
Merging statistics from Scala

                         Nmeas    Nref  Ncent  %poss
 Basic processing        18777    9920    532   65.1
 Improved   "            18647    9846    530   65.1
 Background (basic)      18743    9915    533   65.1
 Background (improved)   18607    9833    528   65.0

                         Mlplct AnoCmpl AnoFrc AnoMlt
 Basic processing          1.9   53.6     74.1    1.1
 Improved   "              1.9   53.7     74.1    1.1
 Background (basic)        1.9   53.6     73.9    1.1
 Background (improved)     1.9   53.7     74.0    1.1


                          Rmeas  Rmeas0  (Rsym)    PCV    PCV0
 Basic processing         0.075   0.131   0.053   0.076   0.141
 Improved   "             0.067   0.124   0.047   0.068   0.131
 Background (basic)       0.071   0.129   0.050   0.071   0.136
 Background (improved)    0.064   0.120   0.045   0.064   0.128
```

The overall effect of changing the processing parameters to take the warnings into account can be judged initially from the merging statistics produced by `SCALA`, but can only really be quantified by analysis of electron density maps and examination of a final, refined model.
Note that there will be still probably be some bad spots, but not enough to trigger the warning messages.
The “basic” and “improved” processing figures were obtained by initial use of the GUI for indexing and postrefinement, followed by background integration. The background (basic) run gave the program the correct spacegroup, but otherwise indexed, refined and integrated without user intervention, whereas the “improved” background run used the known `BACKSTOP`, `SEPARATION CLOSE` and `NULLPIX` values.
It can be seen from the Nmeas figures that those datasets which have been processed with the improvements contain slightly fewer reflections (~0.7% fewer for each style of processing), but Rmeas and other indicators of data consistency are significantly lower.
PCV is the “pooled coefficient of variation” and is a multiplicity-weighted RMS Rmerge.

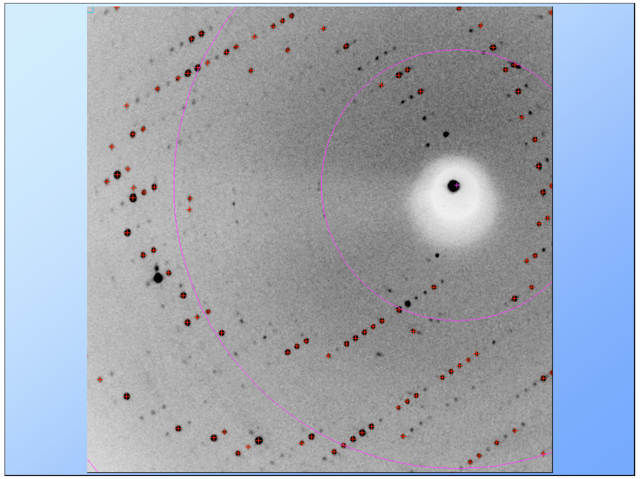
Further Processing with Mosflm

(1) what do you do about the warnings?
(2) what if there are real problems?
(3) the `CCP4i Mosflm` task

Like any program, `Mosflm` sometimes fails to give the expected results using default parameters. In these circumstances, it may be necessary to intervene in order to override the default or derived parameters.

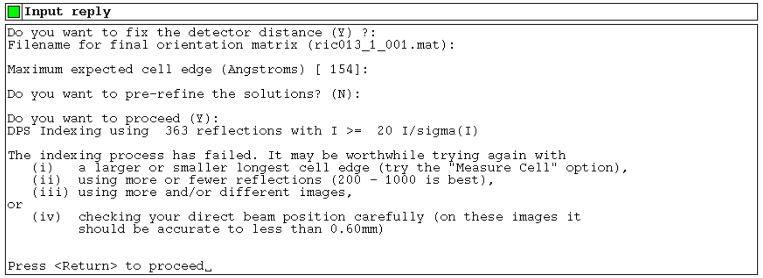
The three most common points of difficulty occur in the spot finding, autoindexing and postrefinement. Normally, once these three steps have been taken successfully, integration proceeds without difficulty.

Spot finding usually fails when the shape or distribution of spots on the image is in some way unusual. In most circumstances, examination of the spots will show how to make the routine work. In some exceptional cases, it may be necessary to pick spots by hand. The user is referred to the mosflm_user_guide, which is available as plain text, HTML and as a hyperlinked PDF on the Mosflm web site.



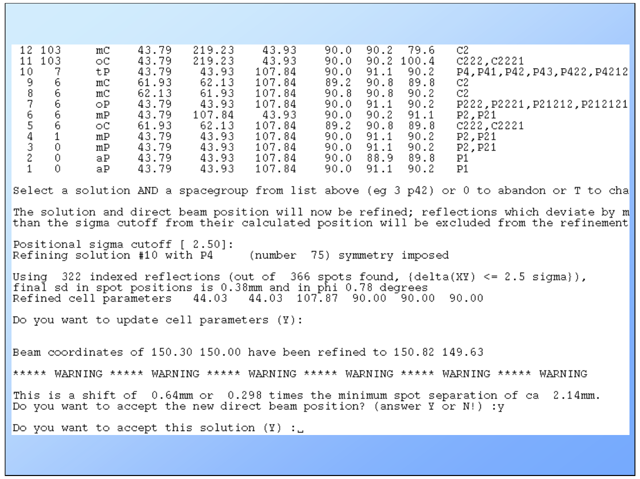
Provided that there is a “good” spot list, autoindexing normally fails when incorrect parameters have been given - usually the beam centre, crystal to detector distance or the wavelength.

In this example, the beam centre has been set to the centre of the image; as there is a “leaky” backstop, it can be seen that this is not too far out. The spot finding has worked and there is a reasonable selection of spots. A number of lunes are visible, so when the autoindexing failed it was rather surprising.



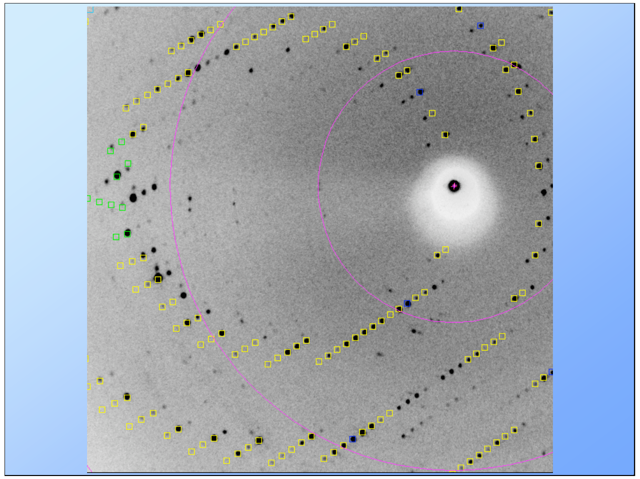
🟩 Input reply

```
Do you want to fix the detector distance (Y) ?:
Filename for final orientation matrix (ric013_1_001.mat):

Maximum expected cell edge (Angstroms) [ 154]:

Do you want to pre-refine the solutions? (N):

Do you want to proceed (Y):
DPS Indexing using  363 reflections with I >=  20 I/sigma(I)

The indexing process has failed. It may be worthwhile trying again with
   (i)   a larger or smaller longest cell edge (try the "Measure Cell" option),
   (ii)  using more or fewer reflections (200 - 1000 is best),
   (iii) using more and/or different images,
or
   (iv)  checking your direct beam position carefully (on these images it
         should be accurate to less than 0.60mm)


Press <Return> to proceed_
```



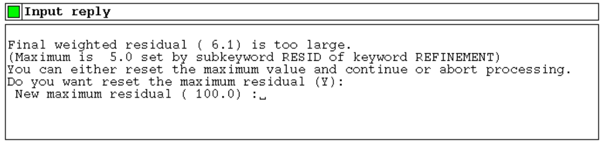
```
 12 103      mC    43.79   219.23    43.93     90.0  90.2  79.6   C2
 11 103      oC    43.79   219.23    43.93     90.0  90.2 100.4   C222,C2221
 10   7      tP    43.79    43.93   107.84     90.0  91.1  90.2   P4,P41,P42,P43,P422,P4212
  9   6      mC    61.93    62.13   107.84     89.2  90.8  89.8   C2
  8   6      mC    62.13    61.93   107.84     90.8  90.8  90.2   C2
  7   6      oP    43.79    43.93   107.84     90.0  91.1  90.2   P222,P2221,P21212,P212121
  6   6      mP    43.79   107.84    43.93     90.0  90.2  91.1   P2,P21
  5   6      oC    61.93    62.13   107.84     89.2  90.8  89.8   C222,C2221
  4   1      mP    43.79    43.93   107.84     90.0  91.1  90.2   P2,P21
  3   0      mP    43.79    43.93   107.84     90.0  91.1  90.2   P2,P21
  2   0      aP    43.79    43.93   107.84     90.0  88.9  89.8   P1
  1   0      aP    43.79    43.93   107.84     90.0  91.1  90.2   P1

Select a solution AND a spacegroup from list above (eg 3 p42) or 0 to abandon or T to cha

The solution and direct beam position will now be refined; reflections which deviate by m
than the sigma cutoff from their calculated position will be excluded from the refinement

Positional sigma cutoff [ 2.50]:
Refining solution #10 with P4     (number  75) symmetry imposed

Using  322 indexed reflections (out of  366 spots found, {delta(XY) <= 2.5 sigma}),
final sd in spot positions is 0.38mm and in phi 0.78 degrees
Refined cell parameters   44.03   44.03  107.87  90.00  90.00  90.00

Do you want to update cell parameters (Y):


Beam coordinates of 150.30 150.00 have been refined to 150.82 149.63

***** WARNING ***** WARNING ***** WARNING ***** WARNING ***** WARNING ***** WARNING

This is a shift of  0.64mm or  0.298 times the minimum spot separation of ca  2.14mm.
Do you want to accept the new direct beam position? (answer Y or N!) :y

Do you want to accept this solution (Y) :␣
```

Correcting the beam centre to the exact centre of the direct beam position on the backstop immediately gave these solutions to the autoindexing. Following normal guidelines, the highest symmetry solution with lowest penalty (#10, primitive tetragonal) was chosen. Refinement of this solution (using ~89% of the reflections used in indexing) gave an SD in the spot positions of 0.38mm, or about 2.5x the pixel size; this is rather larger than ideal, and the error in PHI is also quite large. The refinement of the beam position also has a large shift, so this is also a little worrying. However, as the student who collected the dataset had already encountered problems, the presence of bad statistics should not be a surprise!



Estimating the mosaicity gave a value of ~0.7°, and overlaying a prediction showed that the cell was probably okay, but the mosaicity should perhaps be higher.

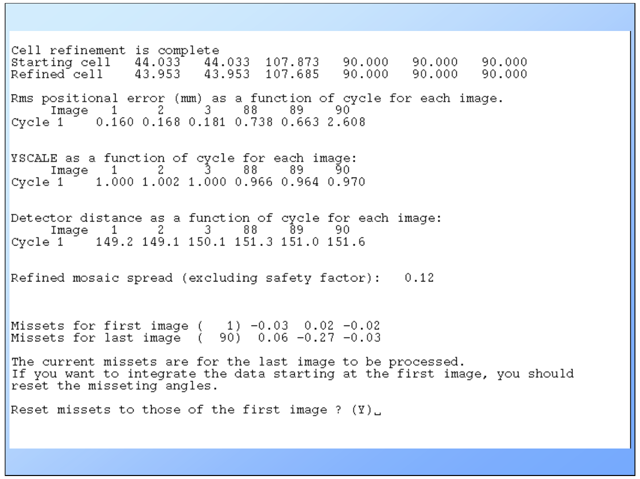
Although postrefinement can converge satisfactorily with one segment of images for tetragonal cells, using two segments separated by ~90° gives greater confidence in the results. However, the following message appeared after a couple of cycles of refinement;



```
Final weighted residual ( 6.1) is too large.
(Maximum is  5.0 set by subkeyword RESID of keyword REFINEMENT)
You can either reset the maximum value and continue or abort processing.
Do you want reset the maximum residual (Y):
 New maximum residual ( 100.0) :␣
```

Since the weighted residual should be roughly unity, this is a clear indication that there is something wrong.

Still, if we accept this allowed maximum residual and let the refinement run to completion, we can see what happens.



```
Cell refinement is complete
Starting cell   44.033   44.033  107.873   90.000   90.000   90.000
Refined cell    43.953   43.953  107.685   90.000   90.000   90.000

Rms positional error (mm) as a function of cycle for each image.
     Image   1     2     3    88    89    90
Cycle 1    0.160 0.168 0.181 0.738 0.663 2.608


YSCALE as a function of cycle for each image:
     Image   1     2     3    88    89    90
Cycle 1    1.000 1.002 1.000 0.966 0.964 0.970


Detector distance as a function of cycle for each image:
     Image   1     2     3    88    89    90
Cycle 1    149.2 149.1 150.1 151.3 151.0 151.6


Refined mosaic spread (excluding safety factor):   0.12



Missets for first image (   1) -0.03  0.02 -0.02
Missets for last image  (  90)  0.06 -0.27 -0.03

The current missets are for the last image to be processed.
If you want to integrate the data starting at the first image, you should
reset the misseting angles.

Reset missets to those of the first image ? (Y)_
```

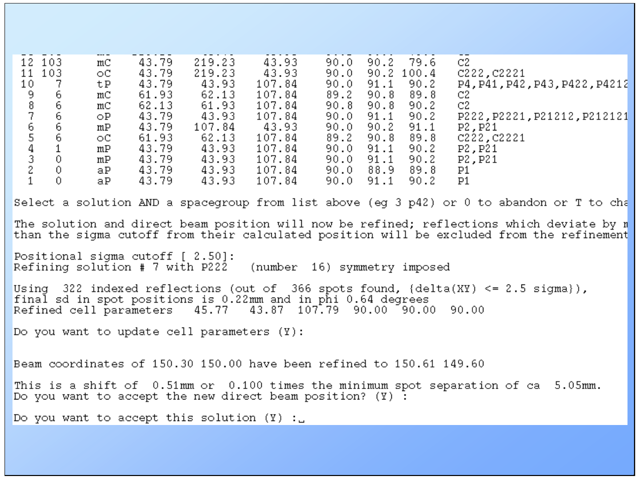
Something is obviously wrong here. The only good thing is that the final cell is close to the starting cell, and in view of the other parameter shifts this is probably an artefact.

The r.m.s. positional error, which should improve with the progress of refinement, and which for a good solution should be ~0.25 - 0.5x the pixel size, is unreasonably large. Similarly, YSCALE has become unreasonable, the distance is changing wildly from image to image, and the mosaic spread (which we knew was already too low at 0.7°) has refined to 0.12°.

It is also worrying that the refinement has stopped after only one cycle, when it seems unlikely that it has converged.

Plainly, the parameters are being refined to unrealistic values to compensate for some other parameters which are being fixed. The first thing to try is to relax the symmetry constraints - we could allow a ≠ b (*i.e.* process as orthorhombic), or we could go further and try to process as monoclinic or even triclinic.

It is normally best (from a practical point of view) to reduce the symmetry constraints one or two at a time. While it is relatively common for an orthorhombic cell to be metrically tetragonal, it is much less common for monoclinic or triclinic cells. Eliminate the obvious first!

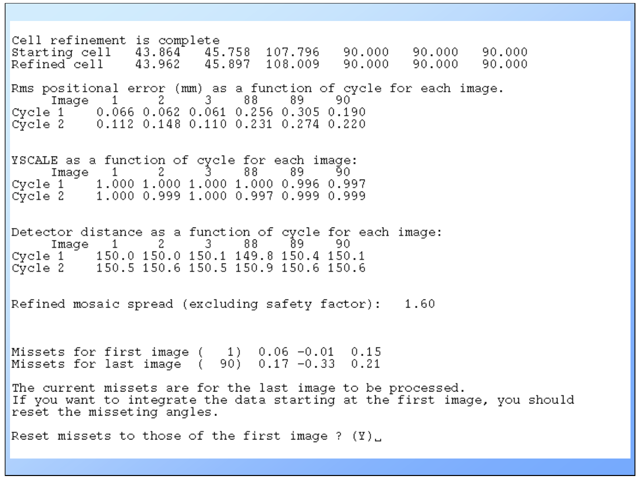


```
 12 103     mC    43.79   219.23    43.93     90.0  90.2  79.6   C2
 11 103     oC    43.79   219.23    43.93     90.0  90.2 100.4   C222,C2221
 10   7     tP    43.79    43.93   107.84     90.0  91.1  90.2   P4,P41,P42,P43,P422,P4212
  9   6     mC    61.93    62.13   107.84     89.2  90.8  89.8   C2
  8   6     mC    62.13    61.93   107.84     90.8  90.8  90.2   C2
  7   6     oP    43.79    43.93   107.84     90.0  91.1  90.2   P222,P2221,P21212,P212121
  6   6     mP    43.79   107.84    43.93     90.0  90.2  91.1   P2,P21
  5   6     oC    61.93    62.13   107.84     89.2  90.8  89.8   C222,C2221
  4   1     mP    43.79    43.93   107.84     90.0  91.1  90.2   P2,P21
  3   0     mP    43.79    43.93   107.84     90.0  91.1  90.2   P2,P21
  2   0     aP    43.79    43.93   107.84     90.0  88.9  89.8   P1
  1   0     aP    43.79    43.93   107.84     90.0  91.1  90.2   P1

Select a solution AND a spacegroup from list above (eg 3 p42) or 0 to abandon or T to cha

The solution and direct beam position will now be refined; reflections which deviate by m
than the sigma cutoff from their calculated position will be excluded from the refinement

Positional sigma cutoff [ 2.50]:
Refining solution # 7 with P222   (number  16) symmetry imposed

Using  322 indexed reflections (out of  366 spots found, {delta(XY) <= 2.5 sigma}),
final sd in spot positions is 0.22mm and in phi 0.64 degrees
Refined cell parameters   45.77   43.87  107.79  90.00  90.00  90.00

Do you want to update cell parameters (Y):


Beam coordinates of 150.30 150.00 have been refined to 150.61 149.60

This is a shift of  0.51mm or  0.100 times the minimum spot separation of ca  5.05mm.
Do you want to accept the new direct beam position? (Y) :

Do you want to accept this solution (Y) :_
```

Choosing the next lowest symmetry solution with a good penalty (orthorhombic P - the C-centred orthorhombic cell is a supercell of the original tetragonal solution) gives us more reasonable statistics which are actually very similar to those obtained if we choose the triclinic basis solution instead;

```
Refining solution # 1 with P1     (number   1) symmetry imposed

Using  325 indexed reflections (out of  366 spots found, {delta(XY) <=
final sd in spot positions is 0.23mm and in phi 0.65 degrees
Refined cell parameters   45.73   43.88  107.76  90.00  90.10  90.14

Do you want to update cell parameters (Y):


Beam coordinates of 150.30 150.00 have been refined to 150.61 149.63

This is a shift of  0.49mm or  0.096 times the minimum spot separation
Do you want to accept the new direct beam position? (Y) :

Do you want to accept this solution (Y) :_
```

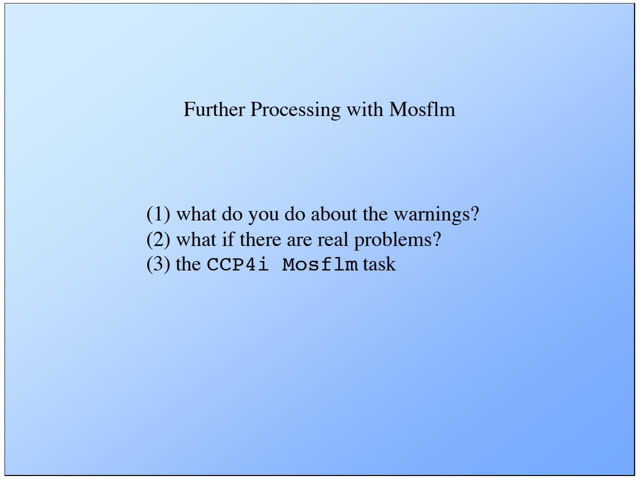
These are still not good statistics, but at least they are better than those we had before, and fall within the range that would normally be considered as acceptable.



```
Cell refinement is complete
Starting cell   43.864   45.758  107.796   90.000   90.000   90.000
Refined cell    43.962   45.897  108.009   90.000   90.000   90.000

Rms positional error (mm) as a function of cycle for each image.
     Image   1      2      3     88     89     90
Cycle 1    0.066 0.062 0.061 0.256 0.305 0.190
Cycle 2    0.112 0.148 0.110 0.231 0.274 0.220


YSCALE as a function of cycle for each image:
     Image   1      2      3     88     89     90
Cycle 1    1.000 1.000 1.000 1.000 0.996 0.997
Cycle 2    1.000 0.999 1.000 0.997 0.999 0.999


Detector distance as a function of cycle for each image:
     Image   1      2      3     88     89     90
Cycle 1    150.0 150.0 150.1 149.8 150.4 150.1
Cycle 2    150.5 150.6 150.5 150.9 150.6 150.6


Refined mosaic spread (excluding safety factor):   1.60



Missets for first image (   1)  0.06 -0.01  0.15
Missets for last image  (  90)  0.17 -0.33  0.21

The current missets are for the last image to be processed.
If you want to integrate the data starting at the first image, you should
reset the missetting angles.

Reset missets to those of the first image ? (Y)_
```

Performing a two segment postrefinement shows that this solution is more acceptable than the previous tetragonal attempt, and the refinement converges after two cycles. However, there still seems to be a problem with the later images (*e.g*. the r.m.s. errors are larger for images 88, 89 and 90); examination shows that the spots are much larger later in the data collection - a common effect when radiation damage has started to be significant.

However, this seems stable enough to proceed with a full integration run. Problems which are highlighted by warnings can be dealt with in a systematic way similar to that outlined earlier.

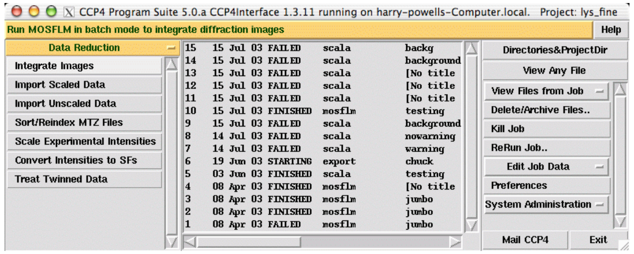


Further Processing with Mosflm

(1) what do you do about the warnings?
(2) what if there are real problems?
(3) the `CCP4i Mosflm` task

Over the past few years, the CCP4i GUI has been developed to make processing datasets following integration into a simpler task for new users of the CCP4 suite. We have recently added a Mosflm task so that, following indexing and postrefinement with the current Mosflm GUI, integration can be optimized using a familiar interface rather than having to edit a command file.

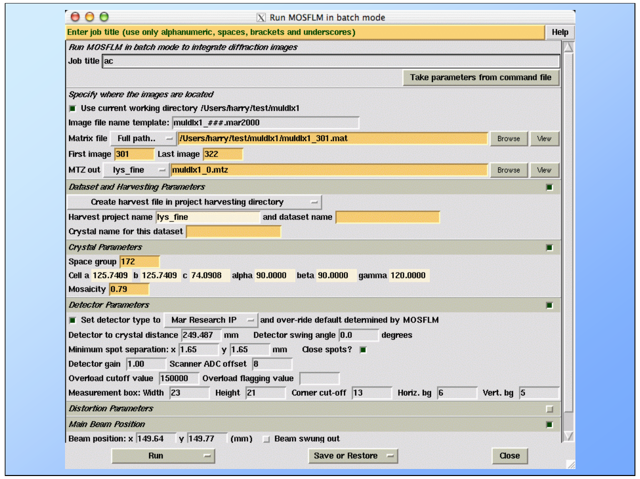
The option can be found under the ‘Data Reduction’ menu on the CCP4i startup screen;



CCP4 Program Suite 5.0.a CCP4Interface 1.3.11 running on harry-powells-Computer.local.   Project: lys_fine

Run MOSFLM in batch mode to integrate diffraction images

```
15    15 Jul 03 FAILED    scala     backg
14    15 Jul 03 FAILED    scala     background
13    15 Jul 03 FAILED    scala     [No title
12    15 Jul 03 FAILED    scala     [No title
11    15 Jul 03 FAILED    scala     [No title
10    15 Jul 03 FINISHED  mosflm    testing
9     15 Jul 03 FAILED    scala     background
8     14 Jul 03 FAILED    scala     nowarning
7     14 Jul 03 FAILED    scala     warning
6     19 Jun 03 STARTING  export    chuck
5     03 Jun 03 FINISHED  scala     testing
4     08 Apr 03 FINISHED  mosflm    [No title
3     08 Apr 03 FINISHED  mosflm    jumbo
2     08 Apr 03 FINISHED  mosflm    jumbo
1     08 Apr 03 FAILED    mosflm    jumbo
```



Enter job title (use only alphanumeric, spaces, brackets and underscores)

Run MOSFLM in batch mode to integrate diffraction images

Job title: ac

Specify where the images are located

☒ Use current working directory /Users/harry/test/muldlx1

Image file name template: muldlx1_###.mar2000

Matrix file: Full path.. /Users/harry/test/muldlx1/muldlx1_301.mat

First image: 301 Last image: 322

MTZ out: lys_fine muldlx1_0.mtz

Dataset and Harvesting Parameters ☒

Create harvest file in project harvesting directory

Harvest project name: lys_fine and dataset name:

Crystal name for this dataset:

Crystal Parameters ☒

Space group: 172

Cell a 125.7409 b 125.7409 c 74.0908 alpha 90.0000 beta 90.0000 gamma 120.0000

Mosaicity: 0.79

Detector Parameters ☒

☒ Set detector type to Mar Research IP and over-ride default determined by MOSFLM

Detector to crystal distance 249.487 mm Detector swing angle 0.0 degrees

Minimum spot separation: x 1.65 y 1.65 mm Close spots? ☒

Detector gain 1.00 Scanner ADC offset 8

Overload cutoff value 150000 Overload flagging value

Measurement box: Width 23 Height 21 Corner cut-off 13 Horiz. bg 6 Vert. bg 5

Distortion Parameters □

Main Beam Position ☒

Beam position: x 149.64 y 149.77 (mm) □ Beam swung out

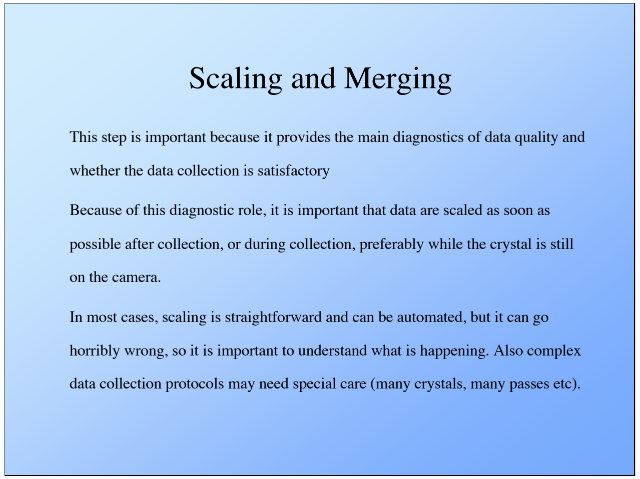
Possibly the most useful feature for new users is the ability to read and interpret a command file saved from an interactive run of Mosflm. Note that several of the boxes which contain information have orange fields, indicating that the user may want to change their contents. Some fields, for example “dataset name” and “crystal name” are blank, but for the purposes of data harvesting it is recommended that they should be completed.

The task has been included in the current release of CCP4 (version 5).

Appendix A: warning messages printed at the end of an integration run

```
*** For information only. ***

 PARTIALS INCLUDED IN POSITIONAL REFINEMENT AND PROFILES
 =======================================================
 Because there were rather few fully recorded reflections in the
 central region of the first image, partials have been included both in
 the positional refinement and in profile formation.
 (The minimum number (eg 20) is set by keywords: REFINEMENT NREF 20 )
 This is equivalent to including keywords:
 REFINEMENT INCLUDE PARTIALS 0.50
 PROFILE PARTIALS


 *** Warning messages ***

 TANGENTIAL OFFSET UNSTABLE
 ==========================
 The tangential offset parameter (TOFF) is varying more than it should.
 (Maximum variation is  0.15mm)
 If there are large changes in both TOFF and ROFF or CCOMEGA, this
 suggests that the refinement is unstable.
 In this case, it is best to fix the TOFF parameter: REFINEMENT FIX
 TOFF
 If known the correct value can be input:eg DISTORTION TOFF 0.17 If not
 known, the mean refined value can be used.
 In such cases ROFF should also be FIXED.

 SPOT OVERLAP
 ============
 Adjacent spots overlap. This will produce systematic errors in the
 intensities.
 Note that this warning will arise even if only one pair of spots in
 one area of the detector overlap. Look at the standard profiles to see
 how serious the overlap is.
 The minimum allowed spot separation (SEPARATION keyword) was
 1.6 1.6mm. The actual spot size determined by the mask optimisation is
 1.7 by 1.4mm in the centre of the image and the largest spot size is
 2.8 by 2.8mm.
 The separation given should be at least as large as the spot size
 in the centre of the image (keyword SEPARATION).
 Check standard profiles carefully to ensure that the optimisation of
 the raster parameters has worked correctly. The effective size of the
 spots can be controlled by PROFILE TOLERANCE keywords. If the peak
 regions look too large (ie they include too much of the tails of the
 spot), try increasing TOLERANCE (current value 0.010) by eg 0.005 and
 see if profiles look better.
 (Increasing TOLERANCE will decrease spot size).
 As a last resort the profile optimisation can be turned off using
 keywords PROFILE NOOPT.

 In cases of serious overlap, (ie if the pattern is very dense), then
 the SEPARATION CLOSE option should be used (eg SEPARATION  1.0 1.0
 CLOSE) and it may also help to suppress profile optimisation in these
 cases (PROFILE NOOPT)keyword. See help library for details.
```

```
EXCESSIVE NUMBER OF BADSPOTS
============================
At least some images have rather a lot of badspots (Maximum number 42)
They are rejected on the basis of:
1) Poor profile fit (PKRATIO >3, controlled by REJECTION PKRATIO).
2) Too large a BGRATIO (too much background variation, controlled by
   REJECTION BGRATIO).
3) Too large a background gradient (controlled by REJECTION GRADMAX)
4) Intensity negative and more than 5 sigma.
Look at the list of badspots to see what category they fall under.

Poor profile fit is often the result of changes in ROFF, TOFF or
CCOMEGA between successive images when using the ADDPART option.
Very intense images can have unusually large gradients, GRADMAX may
have to be changed from the default
A pixel dump of the BADSPOTS can be obtained using REJECTION PLOT
if the reason for their rejection is not clear

TOO MANY BACKGROUND PIXELS OVERLAPPED BY NEIGHBOURING SPOTS
===========================================================
For some of the standard profiles, more than half the background
pixels are flagged as being overlapped by neighbouring spots (in the
worst case,  52.2% are overlapped).
You should use the SEPARATION CLOSE keywords, eg SEPARATION 1.0 1.0
CLOSE
```



Scaling and Merging

This step is important because it provides the main diagnostics of data quality and whether the data collection is satisfactory

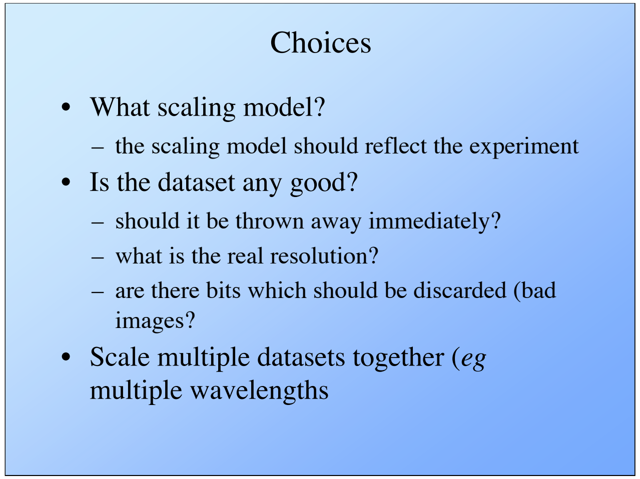
Because of this diagnostic role, it is important that data are scaled as soon as possible after collection, or during collection, preferably while the crystal is still on the camera.

In most cases, scaling is straightforward and can be automated, but it can go horribly wrong, so it is important to understand what is happening. Also complex data collection protocols may need special care (many crystals, many passes etc).

The simple way to run SCALA is to use the CCP4i interface. Sensible defaults have been set, so there is often no need to delve into the manual to find out how to "tweak" the processing. Currently, the only change I make for routine datasets is to switch the "tails correction" on after examination of the normal probability plot, if the plots show that the summed partial intensity is significantly greater than for fully recorded reflections (*i.e*. there is *negative* partial bias).

Check through the SCALA output graphs for discontinuities between adjacent batches/images; sharp changes usually indicate that there are problems with the images, with integrating, or with the scaling model. It is straightforward to check the output files and graphs via CCP4i.

Be wary of using other people's scripts for running SCALA or any other CCP4 program; they will often have been modified to deal with a specific case and may not reflect the nature of your experiment. Often, the person you obtain the script from will have inherited it from someone else and not know why the protocols have been used.

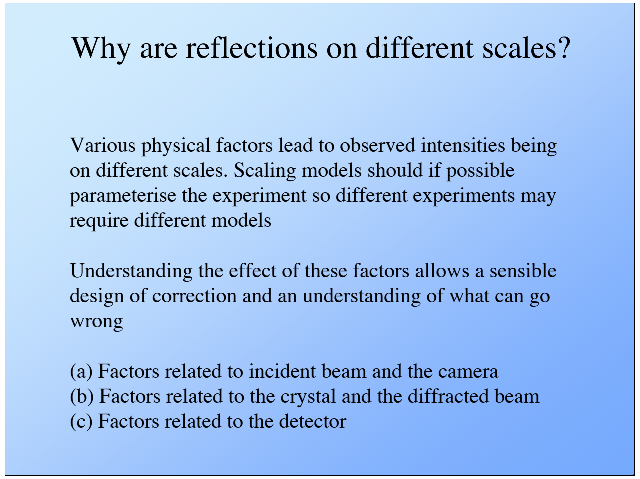


Choices

- What scaling model?
  - the scaling model should reflect the experiment
- Is the dataset any good?
  - should it be thrown away immediately?
  - what is the real resolution?
  - are there bits which should be discarded (bad images?
- Scale multiple datasets together (*eg* multiple wavelengths

For "quick and dirty" diagnostics, the default parameters for running `SCALA` from `CCP4i` are perfectly adequate. The results will show in a few minutes whether the experiment has succeeded or if a more complicated protocol is necessary. Provided that everything looks more or less okay, it is possible to decide whether to collect data on another crystal, or continue collecting on the current one, or even move directly to structure solution (or, if you are lucky enough to be at that stage, refinement).

A detailed analysis may take somewhat longer if the processing has to be tailored to an individual case.



Why are reflections on different scales?

Various physical factors lead to observed intensities being on different scales. Scaling models should if possible parameterise the experiment so different experiments may require different models

Understanding the effect of these factors allows a sensible design of correction and an understanding of what can go wrong

(a) Factors related to incident beam and the camera
(b) Factors related to the crystal and the diffracted beam
(c) Factors related to the detector

(a) Factors related to incident Xray beam

(i) incident beam intensity: variable on synchrotrons and not normally measured. Assumed to be constant during a single image, or at least varying smoothly and slowly (relative to exposure time). If this is not true, the data will be poor

(ii) illuminated volume: changes with ϕ if beam smaller than crystal

(iii) absorption in primary beam by crystal: indistinguishable from (ii)

(iv) variations in rotation speed and shutter synchronisation. These errors are disastrous, difficult to detect, and impossible to correct for: we **assume** that the crystal rotation rate is constant and that adjacent images exactly abut in ϕ. Shutter synchronisation errors lead to partial bias which may be **positive**, unlike the usual negative bias

(b) Factors related to crystal and diffracted beam

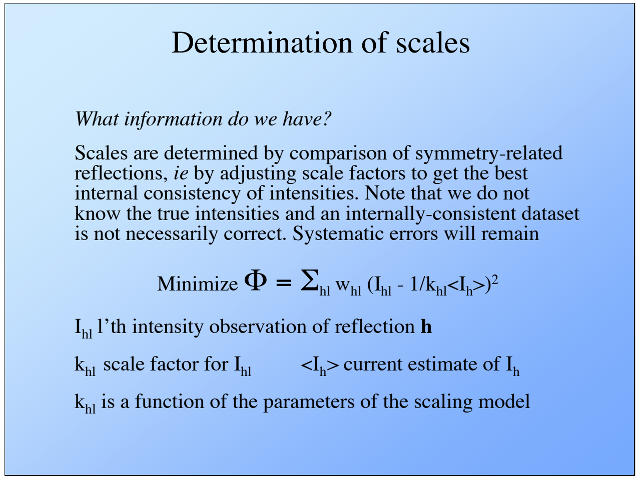
(i) Absorption in secondary beam - serious at long wavelength (including CuKα), worth correcting for MAD data

(ii) radiation damage - serious on high brilliance sources. Not correctable unless small as the structure is changing

The relative B-factor is largely a correction for radiation damage

(c) Factors related to the detector

- The detector should be properly calibrated for spatial distortion and sensitivity of response, and should be stable. Problems with this are difficult to detect from diffraction data.
- The useful area of the detector should be calibrated or told to the integration program
  - – Calibration should flag defective pixels and dead regions *e.g*. between tiles
  - – The user should tell the integration (or scaling program) program about shadows from the beamstop, beamstop support or cryocooler (define bad areas by circles, rectangles, arcs etc.)



Determination of scales

What information do we have?

Scales are determined by comparison of symmetry-related reflections, *ie* by adjusting scale factors to get the best internal consistency of intensities. Note that we do not know the true intensities and an internally-consistent dataset is not necessarily correct. Systematic errors will remain

$$\text{Minimize } \Phi = \Sigma_{hl}\, w_{hl}\, (I_{hl} - 1/k_{hl}\langle I_h\rangle)^2$$

I_{hl} l'th intensity observation of reflection **h**

k_{hl} scale factor for I_{hl} $\langle I_h\rangle$ current estimate of I_h

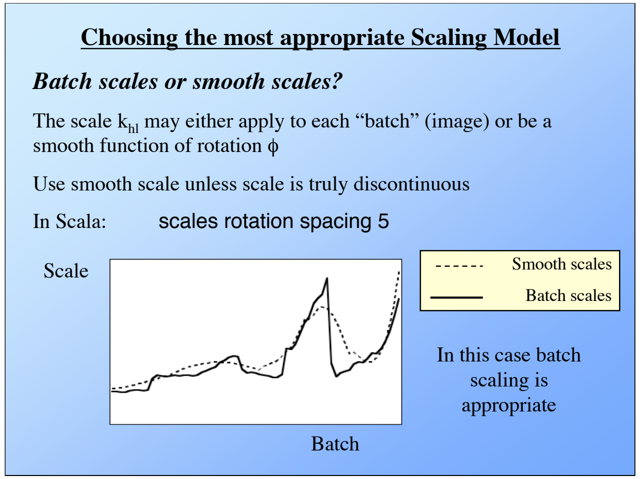
k_{hl} is a function of the parameters of the scaling model

There are a number of different common scaling models of which the user should be aware.

(1) Batch scales or smooth scales? Unless there are true discontinuities between images (batches), smooth scales should be used.

(2) Relative B-factor scaling. This is principally a correction for radiation damage, and unless this is significant *and* you have data to a resolution higher than ~3.0Å it should not be used.

(3) Secondary beam correction. This is mainly an absorption correction (similar to a ψ-scan), and is useful at longer wavelengths (including Cu-Kα). Its use is recommended, particularly for higher symmetry spacegroups.



Choosing the most appropriate Scaling Model

Batch scales or smooth scales?

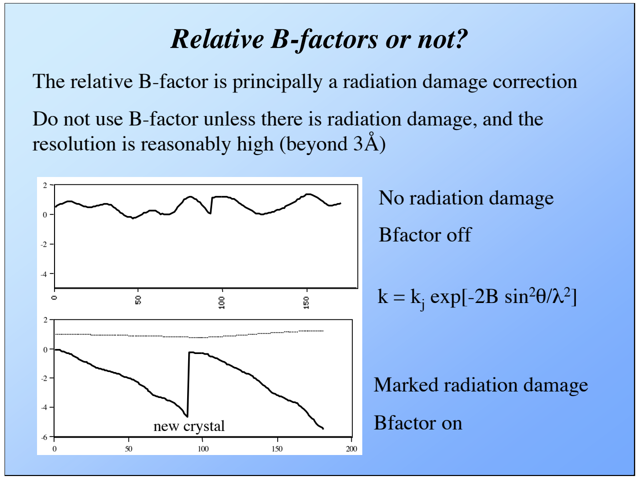
The scale k_{hl} may either apply to each “batch” (image) or be a smooth function of rotation ϕ

Use smooth scale unless scale is truly discontinuous

In Scala: scales rotation spacing 5

In this case batch scaling is appropriate

The necessity for batch scaling is often associated with experimental problems, *e.g*. an instability in the detector or goniostat, or possibly a shutter malfunction. If there has been a loss of X-radiation during data collection (*e.g*. due to beam dump or cooling water failure), this can also indicate that simple smooth scaling is inappropriate, but there may be better ways to deal with the problem.



Relative B-factors or not?

The relative B-factor is principally a radiation damage correction

Do not use B-factor unless there is radiation damage, and the resolution is reasonably high (beyond 3Å)

No radiation damage

Bfactor off

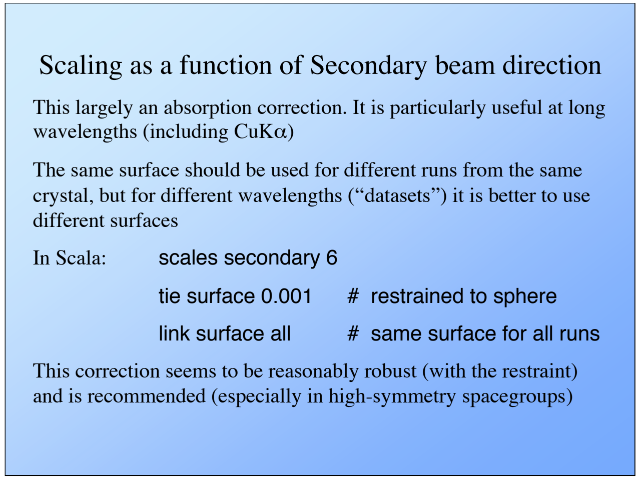
$k = k_j \exp[-2B \sin^2\theta/\lambda^2]$

Marked radiation damage

Bfactor on

Data collected on a cryocooled crystal at a home source should probably not use the B-factor scaling, since there is unlikely to be much radiation damage and inn many cases the resolution will not be very high.

However, for higher temperature data collections, or those carried out at synchrotrons it is probably worthwhile checking to see if it is appropriate.



Scaling as a function of Secondary beam direction

This largely an absorption correction. It is particularly useful at long wavelengths (including CuKα)

The same surface should be used for different runs from the same crystal, but for different wavelengths (“datasets”) it is better to use different surfaces

```
In Scala:    scales secondary 6
             tie surface 0.001     #  restrained to sphere
             link surface all      #  same surface for all runs
```

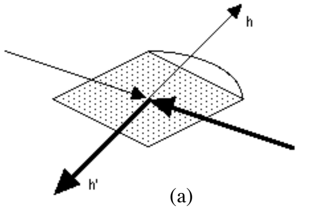
This correction seems to be reasonably robust (with the restraint) and is recommended (especially in high-symmetry spacegroups)

From a typical experiment, we do not have enough data to determine the true absorption, since the absorption generally follows the crystal symmetry. A complete correction requires data from rotating about more than one axis. But we can correct for the absorption differences between symmetry-related observations

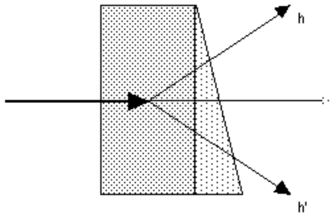
Note that collection strategies aimed at minimising absorption differences between Bijvoet-related reflections assume that the absorption follows the symmetry of the experiment: this is not generally true because of irregular crystal shape and liquid around the crystal

(a) inverse beam: only same absorption if absorption surface has a centre of symmetry

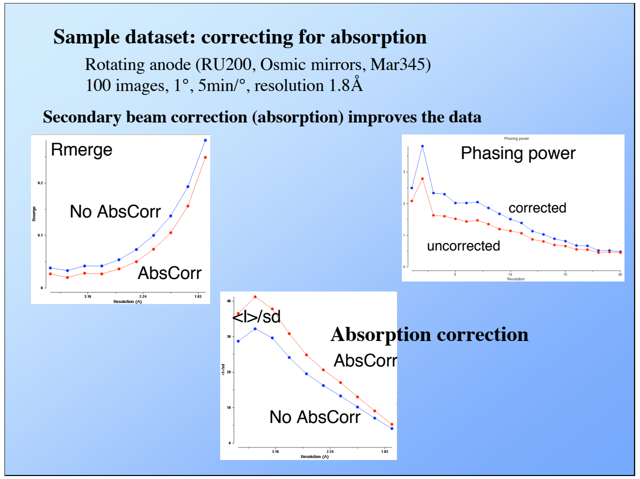
(b) pairs on same image: only same absorption if absorption surface has a mirror plane of symmetry



(a)



(b)

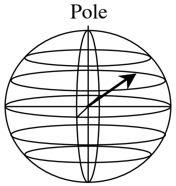


Sample dataset: correcting for absorption

Rotating anode (RU200, Osmic mirrors, Mar345)
100 images, 1°, 5min/°, resolution 1.8Å

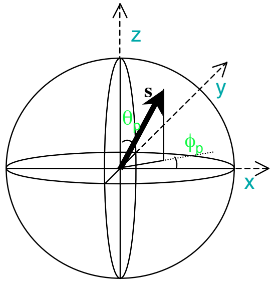
Secondary beam correction (absorption) improves the data

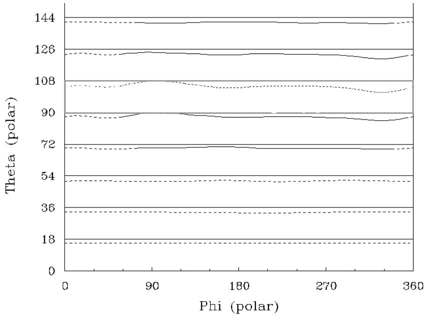
Absorption correction



$$A(\mathbf{s}) = \Sigma_{lm}\, c_{lm} Y_{lm}(\theta_p,\phi_p)$$

Circles of constant latitude θ



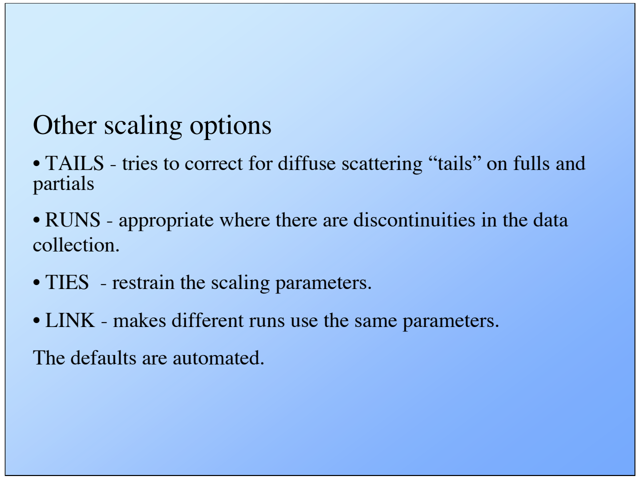


Graph along lines of latitude
(dashed lines where there are no data)

Expand A(**s**) (the *absorption surface*) as sum of spherical harmonic terms, with linear coefficients c_{lm} determined as parameters. The direction of **s** can be expressed as two polar angles θ_p and ϕ_p

Note the surface is not centrosymmetric (see *e.g.* equator θ = 90°)

i.e. different corrections are applied to I+ & I-



Other scaling options

- TAILS - tries to correct for diffuse scattering “tails” on fulls and partials
- RUNS - appropriate where there are discontinuities in the data collection.
- TIES - restrain the scaling parameters.
- LINK - makes different runs use the same parameters.

The defaults are automated.

• TAILS

This correction tries to compensate for the different sampling of diffuse scattering “tails” on fulls and partials.

It should be tried if the Partial Bias is significant

• RUNS

Datasets should be split into separate “runs” where there are discontinuities in the data collection (*e.g.* stop & restart; different crystals), to allow smooth scaling within the runs. This will be automated in future

• TIES are restraints on the scaling parameters

```
Syntax:     tie <restraint_type>        <standard error>
```

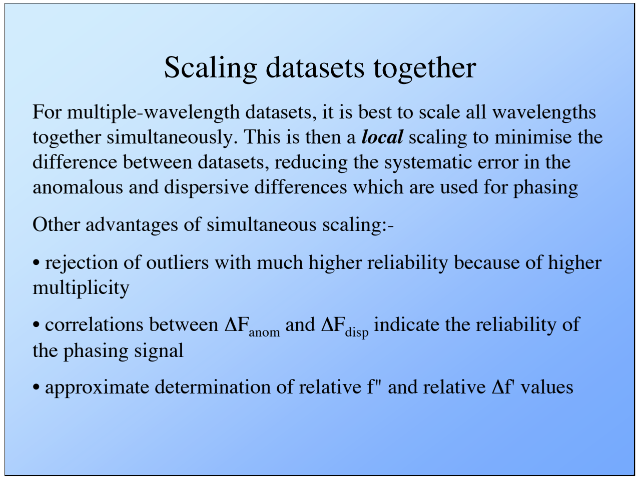
The most useful is TIE SURFACE to restrain the SECONDARY correction (the default is TIE SURFACE 0.001)

Scales and Bfactors may also be restrained

```
            tie rotation 0.1
            tie bfactor 0.5             #   version 3 only
```

• LINKS make different runs use the same parameters

```
            link tails all              # default to use same TAILS values
            link surface all            # same surface
```

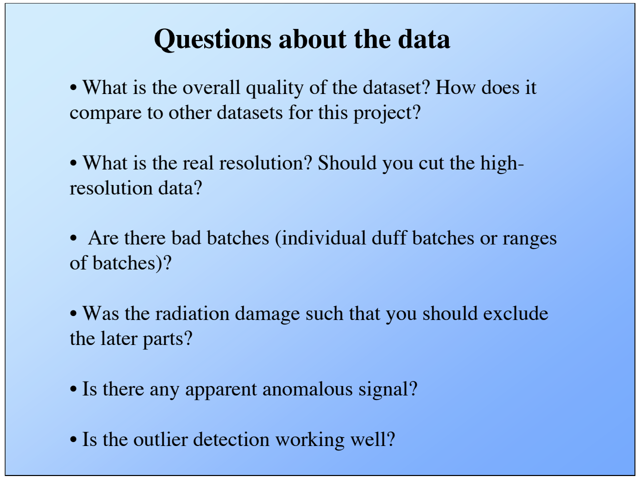


Scaling datasets together

For multiple-wavelength datasets, it is best to scale all wavelengths together simultaneously. This is then a ***local*** scaling to minimise the difference between datasets, reducing the systematic error in the anomalous and dispersive differences which are used for phasing

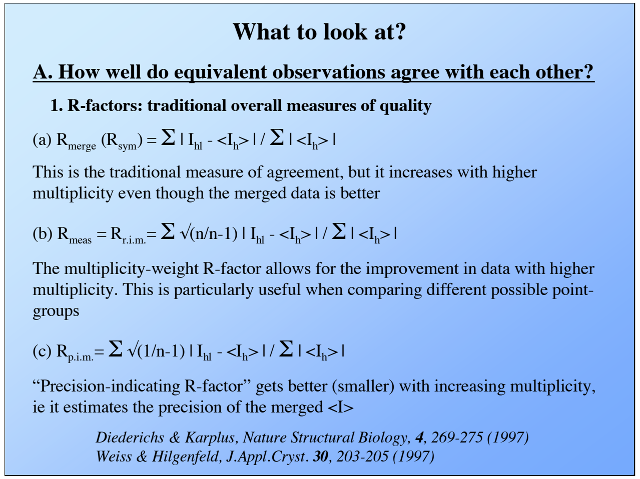
Other advantages of simultaneous scaling:-

- rejection of outliers with much higher reliability because of higher multiplicity
- correlations between ΔF_{anom} and ΔF_{disp} indicate the reliability of the phasing signal
- approximate determination of relative f" and relative Δf' values



Questions about the data

- What is the overall quality of the dataset? How does it compare to other datasets for this project?
- What is the real resolution? Should you cut the high-resolution data?
- Are there bad batches (individual duff batches or ranges of batches)?
- Was the radiation damage such that you should exclude the later parts?
- Is there any apparent anomalous signal?
- Is the outlier detection working well?



What to look at?

A. How well do equivalent observations agree with each other?

1. R-factors: traditional overall measures of quality

(a) R_{merge} $(R_{sym}) = \Sigma \mid I_{hl} - <I_h> \mid / \Sigma \mid <I_h> \mid$

This is the traditional measure of agreement, but it increases with higher multiplicity even though the merged data is better

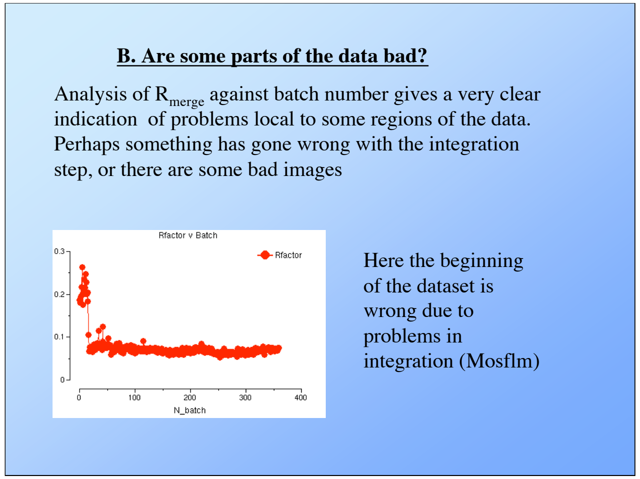
(b) $R_{meas} = R_{r.i.m.} = \Sigma \sqrt{(n/n-1)} \mid I_{hl} - <I_h> \mid / \Sigma \mid <I_h> \mid$

The multiplicity-weight R-factor allows for the improvement in data with higher multiplicity. This is particularly useful when comparing different possible point-groups

(c) $R_{p.i.m.} = \Sigma \sqrt{(1/n-1)} \mid I_{hl} - <I_h> \mid / \Sigma \mid <I_h> \mid$

“Precision-indicating R-factor” gets better (smaller) with increasing multiplicity, ie it estimates the precision of the merged <I>

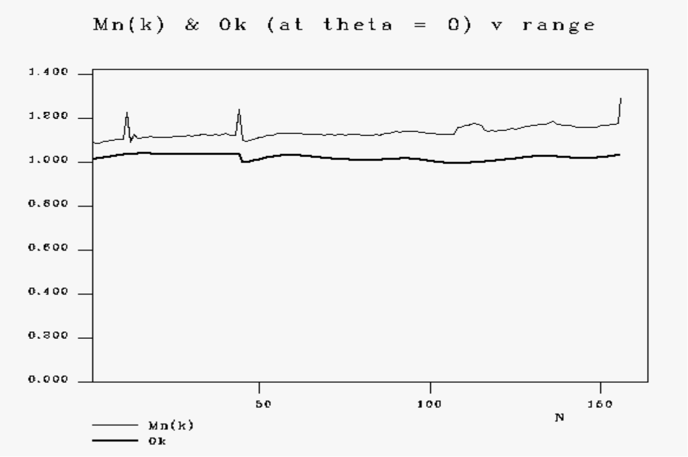
*Diederichs & Karplus, Nature Structural Biology, **4**, 269-275 (1997)*
*Weiss & Hilgenfeld, J.Appl.Cryst. **30**, 203-205 (1997)*



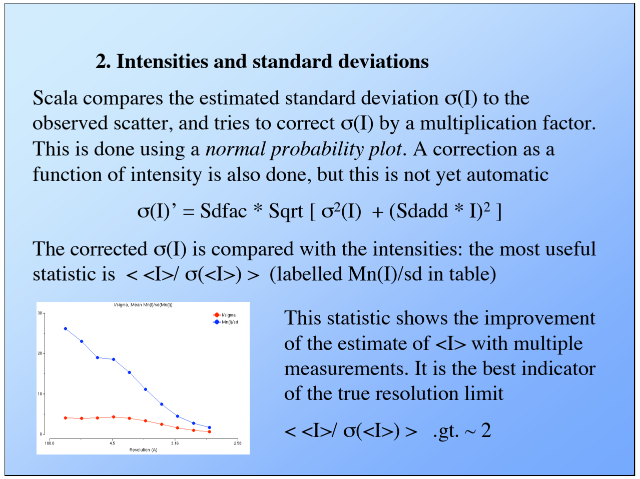
B. Are some parts of the data bad?

Analysis of R_{merge} against batch number gives a very clear indication of problems local to some regions of the data. Perhaps something has gone wrong with the integration step, or there are some bad images

Here the beginning of the dataset is wrong due to problems in integration (Mosflm)



Bad regions may also be indicated by wild variations of scale-factor with batch; here there are a few bad images



2. Intensities and standard deviations

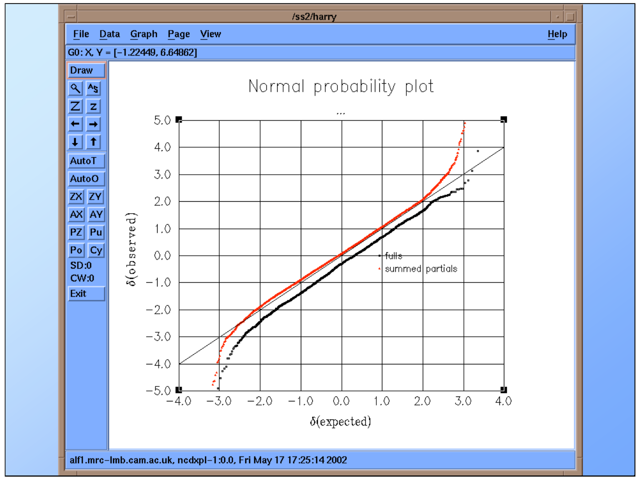
Scala compares the estimated standard deviation σ(I) to the observed scatter, and tries to correct σ(I) by a multiplication factor. This is done using a *normal probability plot*. A correction as a function of intensity is also done, but this is not yet automatic

$$\sigma(I)' = \text{Sdfac} * \text{Sqrt} [\ \sigma^2(I) + (\text{Sdadd} * I)^2\ ]$$

The corrected σ(I) is compared with the intensities: the most useful statistic is < <I>/ σ(<I>) > (labelled Mn(I)/sd in table)

This statistic shows the improvement of the estimate of <I> with multiple measurements. It is the best indicator of the true resolution limit

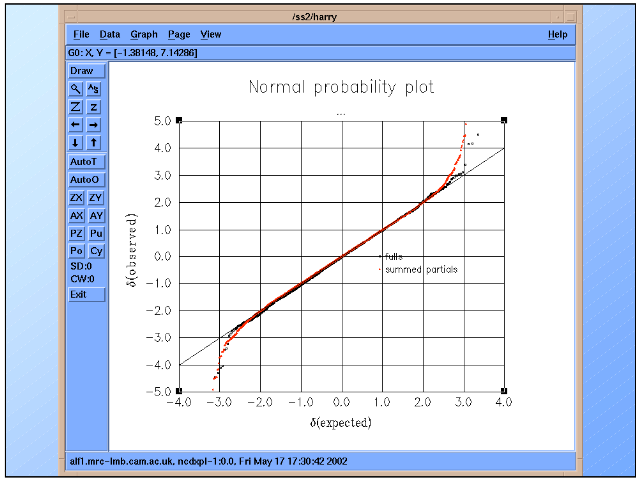
< <I>/ σ(<I>) > .gt. ~ 2



G0: X, Y = [−1.22449, 6.64862]

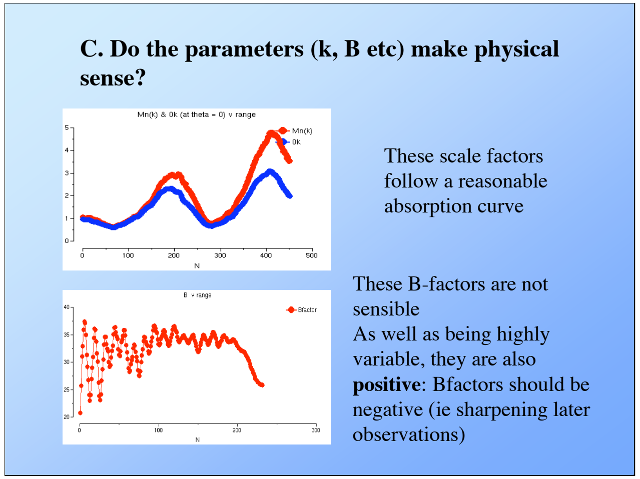
alf1.mrc−lmb.cam.ac.uk, ncdxpl−1:0.0, Fri May 17 17:25:14 2002

This is an example of a normal probability plot (displayed in `XMGR`) which shows significant negative partial bias. This is usually due to diffuse scatter around the Bragg peaks on the diffraction image. The TAILS correction in `SCALA` can be used to try to correct for this; in this case it is successful and the plots for partials and fulls become almost coincident.



G0: X, Y = [−1.38148, 7.14286]

alf1.mrc−lmb.cam.ac.uk, ncdxpl−1:0.0, Fri May 17 17:30:42 2002



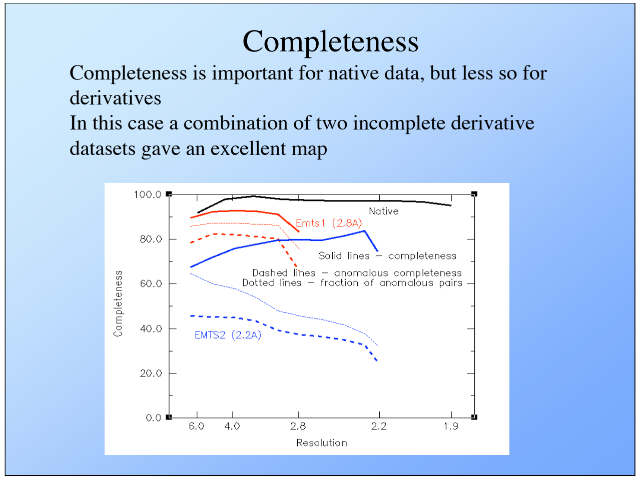
C. Do the parameters (k, B etc) make physical sense?

These scale factors follow a reasonable absorption curve

These B-factors are not sensible
As well as being highly variable, they are also **positive**: Bfactors should be negative (ie sharpening later observations)

As always, sharp discontinuities should raise suspicions. It is unlikely that refined parameters will remain static during the course of a physical experiment, but they should be expected to vary only in a gradual way, so that values from preceding and subsequent images should be similar to those for the current image.

As well as wild variations, watch out for values which are physically unreasonable, or which indicate that something may have gone wrong during the processing.



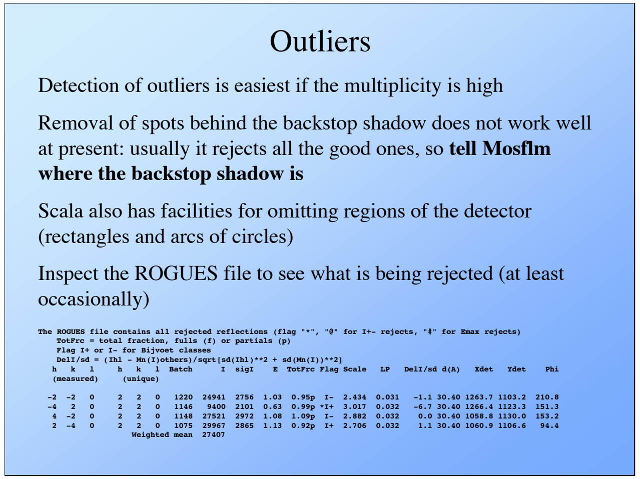
Completeness

Completeness is important for native data, but less so for derivatives

In this case a combination of two incomplete derivative datasets gave an excellent map

While it is good to have 100% complete data with high redundancy, this is usually not achieved in practice - so data collection makes a compromise between completeness and multiplicity.

In general, provided that the experiment has been performed well and there is little evidence of crystal deterioration, higher multiplicity means that the reliability of the measurements is improved. In addition, the chance of spotting outliers is increased, so individual bad measurements can be suitably treated in the data processing (often by omitting them).



Outliers

Detection of outliers is easiest if the multiplicity is high

Removal of spots behind the backstop shadow does not work well at present: usually it rejects all the good ones, so **tell Mosflm where the backstop shadow is**

Scala also has facilities for omitting regions of the detector (rectangles and arcs of circles)

Inspect the ROGUES file to see what is being rejected (at least occasionally)

```
The ROGUES file contains all rejected reflections (flag "*", "@" for I+- rejects, "#" for Emax rejects)
    TotFrc = total fraction, fulls (f) or partials (p)
    Flag I+ or I- for Bijvoet classes
    DelI/sd = (Ihl - Mn(I)others)/sqrt[sd(Ihl)**2 + sd(Mn(I))**2]
   h   k   l      h   k   l  Batch      I  sigI     E  TotFrc Flag Scale    LP   DelI/sd d(A)    Xdet    Ydet     Phi
   (measured)     (unique)

  -2  -2   0      2   2   0   1220  24941  2756  1.03  0.95p  I-  2.434  0.031   -1.1 30.40 1263.7 1103.2  210.8
  -4   2   0      2   2   0   1146   9400  2101  0.63  0.99p *I+  3.017  0.032   -6.7 30.40 1266.4 1123.3  151.3
   4  -2   0      2   2   0   1148  27521  2972  1.08  1.09p  I-  2.882  0.032    0.0 30.40 1058.8 1130.0  153.2
   2  -4   0      2   2   0   1075  29967  2865  1.13  0.92p  I+  2.706  0.032    1.1 30.40 1060.9 1106.6   94.4
                        Weighted mean  27407
```

In this example the second measurement (-4 2 0) is much weaker than the other three symmetry equivalents; while `SCALA` will omit it (as indicated by the * in the Flag column), it might be worthwhile checking that this is the correct choice and that there are no other problems with these reflections.

Reasons (cures) for outliers

- outside reliable area of detector (*e.g.* behind shadow)

  specify backstop shadow, calibrate detector
- ice spots

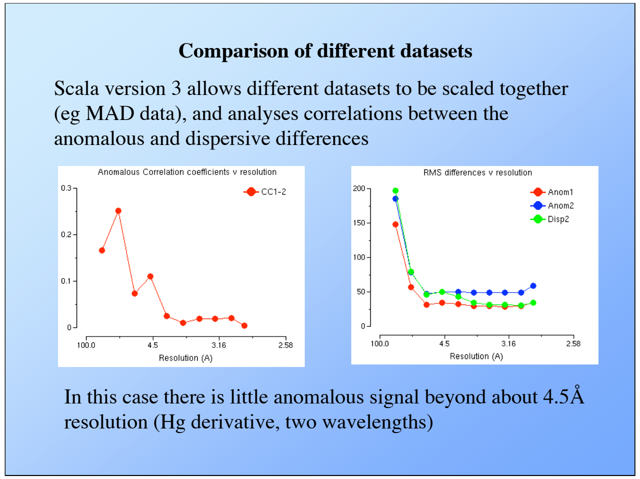
  do not get ice on your crystal!
- zingers
- bad prediction (spot not there)

  improve prediction
- spot overlap

  lower mosaicity, smaller slice, move detector back

  deconvolute overlaps
- multiple lattices

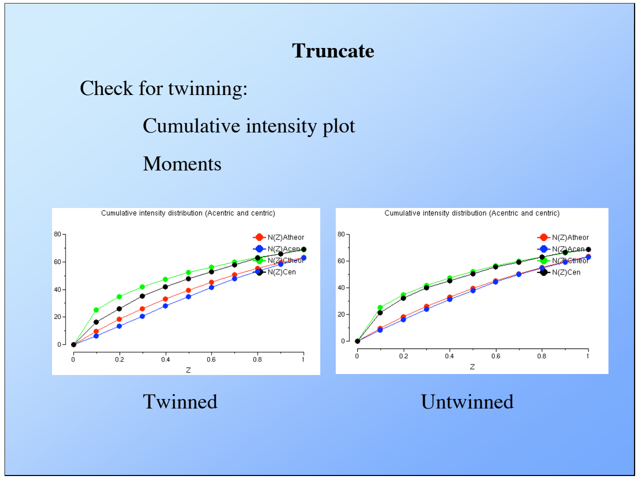
  find single crystal



Comparison of different datasets

Scala version 3 allows different datasets to be scaled together (eg MAD data), and analyses correlations between the anomalous and dispersive differences

In this case there is little anomalous signal beyond about 4.5Å resolution (Hg derivative, two wavelengths)



Truncate

Check for twinning:

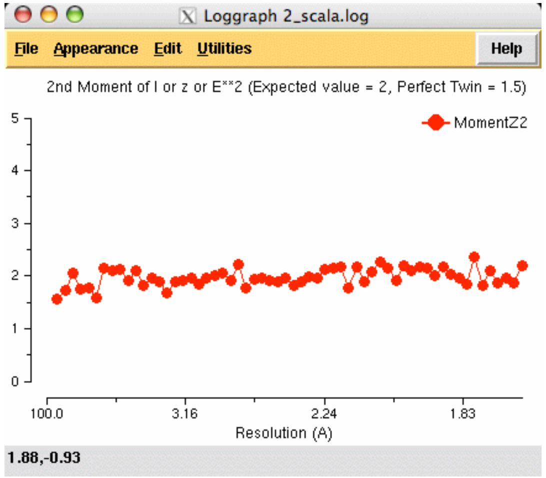
Cumulative intensity plot

Moments

Twinned

Untwinned

The graphs for the moments can be viewed using `loggraph` directly from `CCP4i`; the expected values for both twinned and untwinned crystals are written to the title line for each graph, *e.g.*



Loggraph 2_scala.log
File
Appearance
Edit
Utilities
Help
2nd Moment of I or z or E**2 (Expected value = 2, Perfect Twin = 1.5)
MomentZ2
5
4
3
2
1
0
100.0
3.16
2.24
1.83
Resolution (A)
1.88,-0.93