

Notes on MRC IMAGE PROCESSING software 2000

The MRC Image Processing package is currently freely available to academic workers, and although it is helpful if bugs are reported, no support or instruction is available. A list of users is kept, so please do not just copy the system without informing the MRC. Use of the programs is described in published papers, which are referred to in Crowther, Henderson and Smith, *J. Struct. Biol.* (1996) 116, 9-16.

1. Overview and general philosophy

The system of image processing programs developed by past and present members of the Laboratory is used in various forms worldwide for determining the structure of macromolecular assemblies. The approach has been to study specimens with some form of symmetry, such as crystals or helical and icosahedral particles. Programs suitable for many kinds of 2-D and 3-D analysis have been written. The philosophy has been to write largely stand-alone FORTRAN programs for carrying out the various steps of processing. At the present time (November 2000), there are about 70 stand-alone programs, varying in length from a few lines to many thousands of lines of code. The unifying principle has been to use a standard format (MAPFORMAT) for images and Fourier transforms and postscript format for graphics files. These are also used in the crystallographic CCP4 system. In the later stages of some of the procedures, for example in high resolution electron crystallography, amplitudes and phases are put into a third standard format, that of MTZ which allows direct access to the X-ray crystallographic software in CCP4. Libraries of standard subroutines are available for reading, writing and manipulating files. A user program written to accept files in these three formats and producing output in the same three formats thus fits directly into the system, with no need for any modifications to existing structures. The libraries can be invoked at the linking step on a DEC/Alpha or Silicon graphics under UNIX. FORTRAN user codes will run under either system without modification. MAPFORMAT involves each image or transform having an initial header block 1024 bytes long, which specifies the type and size of the file, the maximum, minimum and mean densities and information about origins. Most importantly it also contains a label field which provides a history of processing operations that have been applied to the image; the convention is that each program adds a label containing a one line description of the operation and time of running. Programs producing graphical output use PLOT2K routines and the resulting postscript format files can be sent to a laserprinter, or viewed on a terminal, using programs such as ghostscript or xview which may be installation dependent. Thus apart from getting digitized data into the system and hard copy out, the system of programs can be used as a coherent whole, independent of the particularities of the installation.

.1 Reading list

- Program documentation- to be found in /users/mrc/mrc/image/doc and /prog/V411/ccp4_OSF1/doc
- Amos *et al.*, 1982
- Henderson *et al.*, 1990
- Unger and Jaeger

2. The practical class

2.1 Aims and objectives

The aim of this class is to introduce you to the use of the MRC crystallographic image processing programs by working on a low-resolution projection image of an untilted, negatively stained crystal.

By the end of this session you should be able to:

- Display an image or Fourier transform on the screen using '*Ximdisp*'
- Select an image area for processing
- Understand the information contained in the image header
- Perform various image manipulations using '*label*'
- Calculate a Fourier transform
- Index a Fourier transform
- Filter an image
- Correct for lattice distortions in an image
- Extract amplitudes and phases from an image transform
- Correct for the effects of the phase CTF
- Determine plane group symmetry and phase origin by comparing phases
- Estimate the resolution and quality of your data
- Merge and average multiple measurements to produce a final list of amplitudes and phases
- Calculate a projection map using some programs from the X-ray crystallography programme suite, CCP4

2.2 A warning

The MRC programs are free, but the price you pay is that they are not supported and there is no user manual. NEVER assume that the data you output are correct without checking over them very carefully. Always check very carefully for error messages. The example we are using here is a well behaved one; other samples can be much more difficult to process.

2.3 Getting started

You will have to know some UNIX in order to run the programs. If you don't know any, don't panic- you should quickly pick up a few basic commands. Commands are shown in **bold**, filenames in *italic*.

2.3.1 Directory structure

You will need to be very organised with your book keeping if you come to process many images. It is a good idea to have a properly organised directory structure.

Type **ls -la** to show all your files

Various sub-directories should have been set up:

<i>./com</i>	contains scripts for processing
<i>./data</i>	contains various data files for reading and writing and keeping permanently
<i>/scr*/[username]</i>	your scratch directory, which contains the digitised image and various intermediate image and transform files

2.3.2 The raw image

Display your image by typing **Ximdisp** and responding to the prompts. **<return>** enters default values. Try entering different parameters to see the effect.

Type **header *image_name.mrc***

** What is the size of the image in pixels?

** Where is the image origin?

2.4 Preparing the image for processing

Using the online Fourier transform facility in Ximdisp, find the best diffracting area

Box out an image of 1000 X 1000 centred on the best area. Save as *image_name.pic*

Type **histo *image_name.pic***
shows *HISTO.PS*

The image has various pieces of dirt etc. that can be suppressed by removing density outliers.

** What density cutoffs would you use?

Type **cd ~/com**
nedit *fftir.com*

It should look like this:

```
#!/bin/csh -x
#
set image = "image_name"           # image name
set cutoffs = "68,98"              # OD cutoffs determined from HISTO output
set scratch = "/scr*/[username]"    # working directory
set red = "1,1,1"                  # pixel compression factor for large images
cd ${scratch}
#
# FFTIMGRED - taperededge, cutoff, and initial pixel averaging and FFT.
# taperededge is often needed to eliminate transform stripes arising
# from discontinuities at the image edge
# Cutoffs help to reduce the effects of bits of dust and hairs.
#
#
setenv IN  ${scratch}/${image}.pic
setenv OUT ${scratch}/${image}.tap
time taperedgek.exe << 'eot'
20,20,200 ! IAVER,ISMOOTH,ITAPER 20,20,200
20,20,200 ! JAVER,JSMOOTH,JTAPER
'eot'
/bin/rm -f YPLOT.PLT
#
label.exe << eot
${image}.tap
99          ! other options
3           ! cutoff over and underflows
${image}.cut
${cutoffs}
eot
#
#
/bin/rm -f ${scratch}/${image}.tap
label2:
label.exe << eot
${image}.cut
4          # average adjacent pixels
${image}.red
${red}     # compression factor
eot
#
#
#          FFTRANS
#
#          Calculate Fourier transform
#
#
#
setenv IN  ${scratch}/${image}.red
setenv OUT ${scratch}/${image}.fft
```

```

fftrans.exe
#
/bin/rm ${scratch}/${image}.red
/bin/rm ${scratch}/YPLOT.PS
#
# FFTPRELIM to allow the transform to be examined on graphics device.
#
label.exe << eot
${scratch}/${image}.fft
5          ! convert to intensities
${scratch}/${image}.int
0
eot
#

```

The script *fftir.com* runs a series of FORTRAN programs to prepare the image for processing. The parameters that usually have to be changed are set as shell variables at the top of the script.

Enter the cutoff values that you determined.

****** What do the various FORTRAN programs in this UNIX script do?

Finish editing the script

Run the script by typing **go fftir**

When all programs have finished you can examine the 'log' of the job, *fftir.log*

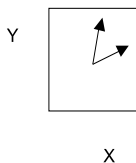
Some of the individual programs in the script can also be run interactively. Try typing **label** and seeing what you can do with it in response to the prompts.

2.5 Indexing a diffraction pattern

You should now have a Fourier transform of the compressed image, e.g. */scr2/david//image_name.fft*

Display this with **Ximdisp**, choosing the density limits to make the diffraction spots visible.

Select **lattice refinement** from the menu and try selecting half a dozen spots to calculate a lattice. Save the results to *image_name.lat*. Hint: the lattice should look something like this, with $<90^\circ$ between *h* and *k*



Display the lattice and write down all the spots visible above background in one half of the transform (*h,k* is equivalent to $-h,-k$)

In `./data` create a file called `image_name.spt` and enter all the spots thus:

```
0 3
0 4
0 5
0 6
1 3
...
...
etc.
```

2.6 Unbending an image

Edit the file `./com/job1000.com`

```
#!/bin/csh -x
#
#      2000 version
#      Image processing for uncompressed 1000 X 1000 image
#
#      NORMALLY ONLY NEED TO CHANGE THE FOLLOWING PARAMETERS FOR DIFFERENT IMAGES
#
set date      = "31.7.01"
set title     = "RC/LH1 form B first try"
set cell      = "45.677,6.941,-4.614,29.708" # lattice from Ximdisp
set image     = "Rcores31-0-3"
set filmno    = " 3"
set saved     = "~per/qian/data"
set spots     = "~per/qian/data"
set scratch   = "/scr1/PER/qian"
set resoll    = "600.0 15.0" # resolution for MMBBOX of compressed
image
set axes      = "127.9,196.5" # unit cell axes in Angstrom
set gamma     = "90.0"
set rad1      = "5" # 1st mask radius
set rad2      = "6" # 2nd mask radius
set smbox     = "487,512,487,512" # 26 x 26 box
set vert1     = "350 350" # coordinates for
reference box in first cycle
set vert2     = "650 350"
set vert3     = "650 650"
set vert4     = "350 650"
set vert1a    = "400 400" # coordinates for second
cycle
set vert2a    = "600 400"
set vert3a    = "600 400"
set vert4a    = "400 600"
set orig      = "500 500" # origin for compressed image
set orig2     = "500 500" # origin for uncompressed FFT
set size      = "1000 1000" # size of compressed image
set red       = "1,1,1" # compression factor
set nocells   = "-80,80,-80,80" # number of cells to search
set thresh    = "0.13" # correlation threshold for
unbending
```

```

cd ${scratch}
#
#
#      MMBOX - preliminary
time mmboxa.exe << eot
${image}.fft
${filmno}          ${title} ${date} preliminary
Y                  ! Use grid units?
Y                  ! Generate grid from lattice?
N                  ! Generate points from lattice?
2 0 0 40 40 15 15 ! IPIXEL,IOUT,NUMSPOT,NOH,NOK,NHOR,NVERT
${orig}           ! XORIG,YORIG
${resol1} 1 ${axes} 100.0 ${gamma} ! RINNER,ROUTER,IRAD,A,B,W,ABANG
${cell}          ! Lattice vectors
eot
#
#
#      JOBA
#
#      MASKTRAN - JOBA
#
#      Apply mask to FT of compressed image
#
setenv IN  ${image}.fft
setenv OUT ${image}.msk
setenv SPOTS ${spots}/${image}.spt
time masktrana.exe << eot
1 T T ! ISHAPE= 1(CIRC),2(GAUSS CIRC),OR 3(RECT) HOLE, IAMPLIMIT(T or F)
${rad1} ! RADIUS OF HOLE IF CIRCULAR, X,Y HALF-EDGE-LENGTHS IF RECT.
${cell} -20 20 -20 20 200 1 !A/BX/Y,IH/IKMN/MX,RMAX,ITYPE
eot
#
#      FFTMSK - JOBA
#
#      Reverse FT to calculate filtered image
#
setenv IN  ${image}.msk
setenv OUT ${image}.flt
time fftrans.exe
#
#      LABEL FOR SMALL BOX IN CENTRE - JOBA
#
#      Used to calculate reference correlation peak profile
#
time label.exe << eot
${image}.flt
1
box${image}.flt
${smbox}
eot
#
#      CALCULATE AUTOCORRELATION PEAK OF SMALL BOX, 20 TIMES OVERSAMPLED
setenv IN  box${image}.flt
setenv OUT auto${image}.cor
time autocorrl.exe << eot
20
eot
/bin/rm -f box${image}.flt
#
#
#      EXTRACT CENTRAL 100 X 100 BOX FROM PROFILE
#
time label.exe << eot

```

```

auto${image}.cor
1
auto${image}.map
210,310,210,310
eot
/bin/rm -f auto${image}.cor
#
#       BOXIMAGE - JOBA
#
#
#       Box out reference box for performing
#       cross-correlation search against filtered image
#
setenv IN  ${image}.flt
setenv OUT ref${image}.flt
time boximage.exe << eot
4       ! NOVERT, VERTEX COORDS GIVEN IN GRID UNITS RELATIVE TO (0,0) ORIGIN.
0 0     ! ORIGIN FOR LATER USE (E.G. IN FOURIER TRANSFORM)
${vert1}
${vert2}
${vert3}
${vert4}
eot
#
/bin/rm ${image}.flt
#
#       FFTREF - JOBA
#
#       Calculate FT of reference box
#
setenv IN  ref${image}.flt
setenv OUT ref${image}.fft
time fftrans.exe
/bin/rm -f ref${image}.flt
#
#       TWOFILE - JOBA
#       Multiply two files together      :      FILE1 * Complex Conjugate of FILE2
#       ICOMB = 2
#       First image on stream 1 (IN1)
#       Second image on stream 2 (IN2)
#       Product output on stream 3 (OUT)
#
#       Performs FFT(filtered image) X FFT(reference box)
#       equivalent to convolution in real space
#
setenv IN1 ${image}.msk
setenv IN2 ref${image}.fft
setenv OUT cor${image}.fft
time twofile.exe << eot
2       ! ICOMB = 2
2 0 0 ${orig}      ! IORIGIN,OXA,OYA,OXB,OYB  Origin shifts to FFT's.
eot
/bin/rm -f ${image}.msk
/bin/rm -f ref${image}.fft
/bin/rm -f boximage.tmp
#       FFTCOR - JOBA
#
#       Reverse FFT gives cross-correlation map
#
setenv IN  cor${image}.fft
setenv OUT cor${image}.cor
time fftrans.exe
/bin/rm -f cor${image}.fft

```



```

#
# QUADSERCHK - JOBA
#
# Use autocorrelation profile to search for cross-correlation peaks
#
setenv PROFILE auto${image}.map
setenv PROFDATA prof${image}.dat
time quadserchk.exe << eot
0,7 ! IPASS,NRANGE
cor${image}.cor
${size} ! SIZE OF TRANSFORM (ISIZEX, ISIZEY)
${cell},F ! Lattice vectors, LREAL
${nocells} ! NUMBER UNIT CELLS TO SEARCH
6,6 ! RADIUS OF CORR SEARCH
${orig} ! POSN OF START SEARCH ORIGIN 0,0 IS ORIGIN
N ! YES/NO FOR DETAILED PRINTOUT
28.0 28.0 45.0 ! RADLIM IN PROFILE GRID UNITS
eot
#
/bin/mv CCPLOT.PS QUAD1.PS
/bin/rm -f auto${image}.map
/bin/rm -f cor${image}.cor
# CCUNBEND - JOBA -- remember to change both titles on each run.
#
# Correct lattice distortions in raw image based on
# deviations of correlation peaks from predicted lattice positions
#
setenv CCORDATA prof${image}.dat
time ccunbendk.exe << eot
${image}.cut
0,1,60,25,F,40,F !ITYPE,IOUT,IMAXCOR,ISTEP,LTAPER,RTAPER,LTABOUT
30,52,0.00001,${thresh},45 !IKX,IKY,EPS,FACTOR,TLTAXIS
${title} SIMPLE UNBEND, ${date}, ${image} PASS 1
cor${image}.pic
UNBENT PASS 1
eot
/bin/mv CCPLOT.PS CCUNB1.PS
/bin/rm prof${image}.dat
#
#
# FFTCORRED
#
# Compress corrected image
#
label.exe << eot
cor${image}.pic
4 # average adjacent pixels
cor${image}.red
${red} # compression factor
eot
#
/bin/rm -f cor${image}.pic
#
setenv IN cor${image}.red
setenv OUT cor${image}.fft
fftrans.exe
/bin/rm -f cor${image}.red
#
# MMBOX - JOBA
time mmboxa.exe << eot
cor${image}.fft
${filmno} ${title} ${date} PASS 1
Y ! Use grid units?

```

```

Y                ! Generate grid from lattice?
N                ! Generate points from lattice?
2 0 0 40 40 15 15      ! IPIXEL, IOUT, NUMSPOT, NOH, NOK, NHOR, NVERT
${orig}            ! XORIG, YORIG
${resol1} 1 ${axes} 100.0 ${gamma} ! RINNER, ROUTER, IRAD, A, B, W, ABANG
${cell}           ! Lattice vectors
eot
#
#
#####
#####
#
#
#                CYCLE 2
#####
#####
#
#   MASKTRAN1 - JOBB- reference map
#
#   Calculate highly averaged image from 1st unbent image
#   giving high signal:noise
#
setenv IN cor${image}.fft
setenv OUT ref${image}.msk
setenv SPOTS ${spots}/${image}.spt
time masktrana.exe << eot
1 T T ! ISHAPE= 1(CIRC),2(GAUSS CIRC),OR 3(RECT) HOLE, IAMPLIMIT(T or F)
1     ! RADIUS OF HOLE IF CIRCULAR, X,Y HALF-EDGE-LENGTHS IF RECT.
${cell} -9 9 -9 9 200 1 !A/BX/Y, IH/IKMN/MX, RMAX, ITYPE
eot
#/bin/rm -f cor${image}.fft
#   FFTMSK - FFTRANS1 - JOBB
setenv IN ref${image}.msk
setenv OUT ref${image}.flt
time fftrans.exe
#/bin/rm -f ref${image}.msk
#   LABEL FOR SMALL BOX IN CENTRE - JOBB
time label.exe << eot
ref${image}.flt
1
box${image}.flt
${smbox}
eot
setenv IN box${image}.flt
setenv OUT auto${image}.cor
time autocorrl.exe << eot
20
eot
time label.exe << eot
auto${image}.cor
1
auto${image}.map
210,310,210,310
eot
#/bin/rm -f box${image}.flt
#/bin/rm -f auto${image}.cor
#   BOXIMAGE - JOBB
#
#
#   Normally select smaller reference box
#   to follow finer detail of image distortions
#   but gives less signal:noise
#

```

```

setenv IN ref${image}.flt
setenv OUT refb${image}.flt
time boximage.exe << eot
4 ! NOVERT, VERTEX COORDS GIVEN IN GRID UNITS RELATIVE TO (0,0) ORIGIN.
0 0 ! ORIGIN FOR LATER USE (E.G. IN FOURIER TRANSFORM)
${vert1a}
${vert2a}
${vert3a}
${vert4a}
eot
/bin/rm -f ref${image}.flt
# FFTREF - FFTRANS2 - JOBB
setenv IN refb${image}.flt
setenv OUT ref${image}.fft
time fftrans.exe
/bin/rm -f refb${image}.flt
# MASKTRAN2:
# MASKTRAN2 - original image to be searched - JOBB
#
# Less averaged version of original raw image
# shows more local distortions but has less
# signal:noise
#

setenv IN ${image}.fft
setenv OUT ${image}.msk
setenv SPOTS ${spots}/${image}.spt
time masktrana.exe << eot
1 T T ! ISHAPE= 1(CIRC),2(GAUSS CIRC),OR 3(RECT) HOLE, IAMPLIMIT(T or F)
${rad2} ! RADIUS OF HOLE IF CIRCULAR, X,Y HALF-EDGE-LENGTHS IF RECT.
${cell} -9 9 -9 9 200 1 !A/BX/Y,IH/IKMN/MX,RMAX,ITYPE
eot
# TWOFIELD - JOBB
# Multiply two files together : FILE1 * Complex Conjugate of FILE2
# ICOMB = 2
# First image on stream 1 (IN1)
# Second image on stream 2 (IN2)
# Product output on stream 3 (OUT)
#
setenv IN1 ${image}.msk
setenv IN2 ref${image}.fft
setenv OUT corel${image}.fft
time twofield.exe << eot
2 ! ICOMB = 2
2 0 0 ${orig} ! IORIGIN,OXA,OYA,OXB,OYB Origin shifts to FFT's.
eot
/bin/rm -f ${image}.msk
/bin/rm -f ref${image}.fft
/bin/rm -f boximage.tmp
# FFTCOR - FFTRANS3 - JOBB
setenv IN corel${image}.fft
setenv OUT cor${image}.cor
time fftrans.exe
/bin/rm -f corel${image}.fft
# QUADSERCHK - JOBB
setenv PROFILE auto${image}.map
setenv PROFDATA prof${image}.dat
time quadserchk.exe << eot
0,7 ! IPASS,NRANGE
cor${image}.cor
${size} ! SIZE OF TRANSFORM (ISIZEX, ISIZEY)
${cell},F ! Lattice vectors
${nocells} ! NUMBER UNIT CELLS TO SEARCH

```

```

6,6                ! RADIUS OF CORR SEARCH
${orig}            ! POSN OF START SEARCH ORIGIN  0,0 IS ORIGIN
N                  ! YES/NO FOR DETAILED PRINTOUT
28.0 28.0 45.0     ! RADLIM IN PROFILE GRID UNITS
eot
/bin/rm -f auto${image}.map
/bin/rm -f cor${image}.cor
/bin/mv CCPLOT.PS QUAD2.PS
#   CCUNBEND - JOBB -- remember to change both titles on each run.
setenv CCORDATA prof${image}.dat
time ccunbendk.exe << eot
${image}.cut
0,1,60,25,F,40,F  !ITYPE,IOUT,IMAXCOR,ISTEP,LTAPER,RTAPER - essential except
TAPER
30,52,0.00001,${thresh},45  !IKX,IKY,EPS,FACTOR,TLTAXIS -only FACTOR with
nonbicubic
  ${title} SIMPLE UNBEND, ${date}, ${image} PASS 2
cor${image}.pic
  UNBENT PASS 2, ccunbende, SIMPLE UNBEND ${date}
eot
/bin/mv CCPLOT.PS CCUNB2.PS
/bin/rm -f cor${image}.fft
/bin/rm -f prof${image}.dat
/bin/rm -f ${image}.fft
#
#
setenv IN cor${image}.pic
setenv OUT cor${image}.fft
time fftrans.exe
#   MMBOX - JOBB
#
#
time mmboxa.exe << eot
cor${image}.fft
${filmno}          ${title} ${date}
Y                  ! Use grid units?
Y                  ! Generate grid from lattice?
N                  ! Generate points from lattice?
2 0 0 40 40 15 15 ! IPIXEL,IOUT,NUMSPOT,NOH,NOK,NHOR,NVERT
${orig2}           ! XORIG,YORIG
${resol1} 1 ${axes} 100.0 ${gamma} ! RINNER,ROUTER,IRAD,A,B,W,ABANG
${cell}           ! Lattice vectors
eot
#

```

The parameters that most often need changing can be edited at the top of the script. You should have obtained X,Y values for your h,k axes similar to those entered for the variable 'cell'.

The script can be run in stages by activating any one of the 'goto' lines. A preceding '#' prevents the following text from being interpreted as a command. (The whole script can be run as one job by removing all 'exit' commands.)

Try running the job in stages by removing '#' for the stage in question and replacing '#' in front of all other goto commands.

Run the job each time by typing **go job1000**

Examine the output in *job1000.log*. The program *mmbbox.exe* at the start determines the amplitude and phase of each predicted spot.

** What is the average intensity peak height of all the spots relative to the background? Note this down.

** How many spots have IQ 1 and IQ 2? Note this down.

You will notice various image files are created in the scratch directory at different stages and some of them are then later deleted. Try examining the various files as you go along, using **Ximdisp**.

** What are the following images for: *image_name.msk*, *image_name.flt*, *image_name.flt*, *image_name.cor*, *image_name.pic*?

In 'stage4' you determine the lattice distortions by searching for peaks in the cross-correlation map (program *quadserchk.exe*). A graphical representation with distortion vectors magnified 20 times is shown in the two pages of *QUADI.PS*, which you can view on the screen or print out.

The programme *ccunbendk.exe* corrects for the distortions and the results can be seen in *CCUNBI.PS*

** By looking at the results of *mmbbox.exe* at the end of stage4, how much has the signal-to-noise been improved?

** Cycle2 performs a second round of unbending. How do the filtered image and reference image differ from the first cycle?

2.7 Determining the amplitudes and phases

Type **mmbox.com >& mmbox.log &**

From *mmbox.log* make a note of the central peak height and IQ values.

Try editing *mmbox.com* to try different resolution ranges.

```
#!/bin/csh -x
#
#
#       Output amplitudes and phases from uncompressed, unbent image
#
#
#       NORMALLY ONLY NEED TO CHANGE THE FOLLOWING PARAMETERS FOR DIFFERENT IMAGES
#
set date      = "16.7.01"
set title     = " Highresolution map"
set cell      = "70.786,28.962,10.465,75.572"      # lattice from Ximdisp
set image     = "highres"                        # filename
set filmno    = "101"                            # film number
set saved     = "../data"
set scratch   = "../scratch"
set resoll    = "600.0 4.0"                      # resolution
set axes      = "64,64"                          # unit cell axes in Angstrom
set gamma     = "120.0"                          # gamma
set orig2     = "2000,2000"                      # phase origin
cd ${scratch}
#
#
#       MMBOX
#
#
time mmboxa.exe << eot
cor${image}.fft
${filmno}          ${title} ${date}
Y                  ! Use grid units?
Y                  ! Generate grid from lattice?
N                  ! Generate points from lattice?
2 1 0 40 40 15 15 ! IPIXEL,IOUT,NUMSPOT,NOH,NOK,NHOR,NVERT
${saved}/raw${image}.aph
${orig2}           ! XORIG,YORIG
${resoll} 1 ${axes} 100.0 ${gamma} ! RINNER,ROUTER,IRAD,A,B,W,ABANG
${cell}           ! Lattice vectors
eot
#
```

**** Can you estimate the resolution of the data?**

Run *mmbbox.com* to output all data to the resolution limit. The data can be examined in *./data/raw [image_name].aph*.

**** What are the different columns in *raw[image_name].aph*?**

2.8 Determining the symmetry

Run *./com/allspace.com* in the same way that you ran all the other jobs above.

```
#!/bin/csh -x
#
#
#     ALLSPACEA
#
#     Perform internal phase comparison
#     between symmetry-related spots for
#     all plane groups
#
cd ../data
#
setenv IN [image_name].aph
time allspacea.exe << eot
ALL
T T T 4000                ! SEARCH,REFINE,TILT,NCYC
0. 0. 0. 0.              ! ORIGH,ORIGK,TILTH,TILTK
6.0 61                   ! STEPSIZE, PHASE SEARCH ARRAY SIZE
64.0 64.0 120.0 150.0 4.0 2.0 200 ! A, B, GAMMA, RIN, ROUT, CS, KV
F 0 7                    ! ILLIST,ROT180,IQMAX
eot
#
```

**** What is the symmetry of the crystal?**

**** By how much must the phase origin be corrected for in this symmetry group?**

2.9 Calculating a map

Typically you would want many independent measurements of each reflection before calculating a map, usually by processing several images. The script *fourp3.com* has been set up to calculate a projection map from just one image. However, it could be adapted to combine data from several images.

```
#!/bin/csh -x
#
#####
#
# COM-FILE TO CALCULATE A PROJECTION MAP WITH          #
# p3 SYMMETRY                                         #
#
#####
#
# usually need to change only parameters that follow.
#
#
set image      = "[image_name]"
set filmno    = "101"
set ctf       = "../data"
set title     = "[image_title]"
set saved     = "../data"
set scratch   = "../scratch"
set date      = "[date]"
set realuv    = "[a,b]"
set angle     = "[gamma]"
set grid      = "[number of sampling points along a,b,c e.g. 60 60 60]"
set uc        = "[range of calculation in sample points for one unit cell e.g. 0 59 0 59]"
set ucno      = "[range to plot e.g. 0 119 0 119] "
set resol     = "600.0 4.0"
set kv        = "200."
set cs        = "2.0"
#
#
cd ${scratch}
#
#####
#
# ORIGTILTK
#
# Sets image to correct phase origin and applies
# beam tilt correction
# For multiple images, can refine each image phase origin
```



```

#      against the phases of all the preceding images. Need to repeat cards
#      from FILM NUMBER to RESOLUTION LIMITS for each film. Check SCL is
#      set correctly
#
time origtiltk.exe << eot
13 0 F F 0 ${realuv} 100.  ${angle} 0 15 !ISPG,NPRG,NTL,NBM,ILST,A,B,W,ANG,IPL,MNRF
1001 0 30 7 61 F F          !IRUN,LHMN,LHMX,IQMX,IBXPHS,NREFOUT,NSHFTIN
      100 DUMMY
${ctf}/dummy.aph
      ${filmno}  ${title}          !Film number, title
${ctf}/ctf${image}.aph
      F
      45.0 0.0 1          ! TAXA,TANGL,IORIGT
-43.1 42.3 0.0 0.07 0 1.0 0 0 0          ! OH,OK,STEP,WIN,SGNXCH,SCL,ROT180,REVHK,CTFREV
${cs}  ${kv} -0.52 -1.84          ! cs,kv,tx,ty
${resol}          ! resolution limits
      -1
eot
#
/bin/mv fort.3 ${scratch}/${image}-merg.aph
#
#####
#  AVRGAMPHS
#
#      2000 version has a bug. This is an earlier version
#      Vector sum of structure factors
#
ln ${scratch}/${image}-merg.aph fort.1
time ../LOCAL/avrgamphs.exe << eot
F
1001,-0.01,0.01
8
${realuv}  ${angle}
eot
#
/bin/mv fort.2 ${scratch}/${image}-avrg.aph
/bin/rm SUMMARY
/bin/rm fort.1
#
#####
#  F2MTZ - to convert aph-file to mtz-file
#
#      All subsequent programs from CCP4
#      crystallography package
#
setenv HKLOUT ${scratch}/${image}-avrg.mtz
f2mtz << eof
TITLE  ${title}  ${filmno}  ${date}
CELL  ${realuv} 100.  90 90  ${angle}
SYMMETRY P3
LABOUT H K L AMP PHASE FOM
CTYPOUT H H H F P W
FILE  ${scratch}/${image}-avrg.aph
SKIP 1
END
eof
#
#####
#  FFTBIG - to calculate a map of unit cell
#
setenv HKLIN  ${scratch}/${image}-avrg.mtz
setenv MAPOUT ${scratch}/temp.map
fftbig << eot
LABIN F1=AMP PHI=PHASE W=FOM ##
PROJECTION

```

```

SCALE F1 1.0 0.0
RESOLUTION ${resol}
TITLE ${title} ${filmno} ${date}
GRID ${grid}
AXIS Y,X,Z
XYZLIM ${uc} 0 0
RHOLIM 250.0
HKLMAX ,,20
eot
#
#####
#
# EXTEND - to extend the map
#
extend MAPIN ${scratch}/temp.map MAPOUT ${scratch}/${image}.map << eof
xyzlim ${ucno} 0 0
eof
#
/bin/rm ${scratch}/temp.map
#####
#
setenv PUBLIC_FONT84 /software/ccp4-4.1.1/lib/font84.dat
# NPO - to create a contour plot
#
npo MAPIN ${scratch}/${image}.map PLOT ${scratch}/${image}.plt << eof-pluto
TITLE ${title} ${filmno}, ${date}
CELL 64.0 64.0 100.0 90.0 90.0 120.0
MAP
CONTRS -500 TO 500 BY 20
SECTNS 0 0
MODE BELOW 0 DASHED 1 0.15 0
GRID 0.1
COLOUR BLACK
PLOT Z
eof-pluto
#
#####
#
# PLOT - to create a ps-file of contour plot
#
pltdev -i ${scratch}/${image}.plt -o ${scratch}/${image}.ps
#
#####
#
exit
#
#####

```

You should have determined that the image has some internal symmetry meaning that each Fourier term has more than one measurement. These measurements can be averaged together. In order to do this, the correct phase origin and beam tilt must be set. The program *origtiltk.exe* does this and outputs a set of merged reflections for each symmetry-independent Fourier term. Edit the script, to set your image to the correct phase origin (OH, OK). Set the resolution to your highest estimate.

Run *fourp3.com* and examine the log file.

The program *avrgamphs.exe* averages the measurements for each Fourier term.

**** From the table of phase residuals, what is the overall phase residual?**

The remaining programs in *fourp3.com* are not from the MRC suite. They are from the CCP4 X-ray crystallography suite. These programs use a KEYWORD input of parameters. The job outputs a projection map *[image_name].ps*. Try displaying or printing this.

** The map has had symmetry applied, can you print the same map without symmetry averaging of the data?

Try printing the map at different resolutions such as 35, 25, or 20 Å.

** What can you infer about the structure of this protein complex?