MSLICE

Matlab Visualisation Software for Single Crystal and Powder Time-of-Flight Neutron Data

version 10-Oct-2000
written by Radu Coldea

1. INTRODUCTION

Programme files are are a collection of Matlab functions (source code ASCII.m files) plus a C file (source code ASCII ffind.c) and fortran files (source code ASCII slice_df.f, sort1d_df.f, avpix_df.f, load_spe_df.f) and the compiled versions of the C and the Fortran codes (in binary .dll format for PC). The (.m,.c,.f) source code should be portable between Windows, VMS and UNIX/Linux platforms. For VMS and Unix/Linux the c/Fortran source codes need to be compiled on the local computer they will actually run (use matlab command mex, i.e. >> mex filename).

1.1. FILE TRANSFER USING FTP FROM ISIS/RISOE

The complete MSlice package (+ example data sets) is currently available for ftp from either the ISIS alpha VMS cluster (ftp to isisa.nd.rl.ac.uk) or RISOE (ftp to fys-hp-2.risoe.dk), or email from the author. Can ftp a single zip file of all mslice files (recommended) produced using WinZip7 for Windows PC. Ftp as BINARY to your own PC, unzip using WinZip and follow instructions from Section 1.2. Unzipped mslice files also available, transfer as follows

ASCII: *.m, *txt, *dat, *.c, *.f files

BINARY:

ffind.dll	c i/o reading file compiled using visual c++
slice_df.dll	fortran slicing/binnig routine compiled using Digital Visual Fortran 5.0B for Windows PC
sort1d_df.dll	fortran routine to sort an array of pixels for subsequent binning, compiled using Digital Visual Fortran 6.0 for Windows98/NT
avpix_df.dll	fortran routine to average a quantity over groups of pixels, compiled using Digital Visual Fortran 6.0 for Windows98/NT for PC
load_spe_df.dll	fortran routine to load ASCII spe file Digital Visual Fortran 6.0 for Windows98/NT

Location of files:

RISOE: fys-p-2:/users/lip/oxford/radu/matlab/mslice/lastupdate_zip/
mslice_08feb2000.zip file

RISOE: fys-hp-2:/users/lip/oxford/radu/matlab/mslice/lastupdate/
+ subdirectories /HET, /MARI, /IRIS with all mslice files

ISIS: towser$disk2:[map.radu.mslice.lastupdate_zip]
mslice_08feb2000.zip file

ISIS: towser$disk2:[map.radu.mslice.lastupdate]
+ subdirectories [.HET], [.MARI], [.IRIS] with all mslice files

1.2. INSTALLATION FROM FLOPPY DISK

Installation disk (floppy 3 1/2") contains the complete version of the Mslice programme for PC, written in MATLAB Version 5.3 on PCWIN + all c and fortran code already compiled. Installation procedure:

Copy all files in the a:\mslice directory to a directory on the hard disk, for example c:\mslice. Start Matlab and change directory >> cd c:\mslice, then start the programme with command >> mslice, the Control Window should appear on the screen (On a 19 inch diagonal viewable area monitor with screen resolution 1024x768 Small Fonts True Color 85 Hertz the Control Window fills the screen height).To avoid having to change directory every time you wish to start mslice, add the mslice directory to the matlab path,for example in the function startup.m (in the matlab\bin directory or c:\matlabr11\work ver5.3) insert the line path('c:\mslice\',path), start matlab and then when you issue the command >> mslice, matlab will know where to pick up the functions from. Start typing in the editable white boxes parameters necessary for your particular data file and spectrometer settings as described in Section 3.

1.3. DOWNLOAD FROM ISIS PC NT NETWORK

The complete package is located in

\\Ndacromwell\BABYLON4\ Scratch\radu\mprogs.

Copy all directory structure onto c:\mprogs. Copy file startup.m from directory mprogs/ into c:\matlabr11\work directory to set up correct paths to mslice and mfit applications.

2. MSLICE OVERVIEW

The programme currently runs in four different configurations:

(a) Single crystal sample with PSD (or area) detectors (default mode) - data is described by projections onto three orthogonal axes in the 4D wavevector-energy space (requires information about the
lattice parameters and crystal orientation)

(b) Single crystal sample with conventional detectors - data is described by projections onto two orthogonal axes in the 4D wavevector-energy space (requires information about the lattice parameters and crystal orientation)

(c) Single crystal sample analysed as a powder- data is described in terms of two projections, which can be selected from |Q|, Energy, 2Theta and Detector Group Number, but calculates also (h,k,l,energy) values for each pixel (requires information about the lattice parameters and crystal orientation)

(d) Powder sample - data is described in terms of two projections, which can be selected from |Q|, Energy, 2Theta and Detector Group Number (does not require information about the lattice parameters). A single crystal data set can also be described in powder modeif no access to h,k,l projections is required.

Each of these modes has its own initialisation file with typical parameters:

(a) crystal_psd.msp
(b) crystal_no_psd.msp
(c) crystal_as_powder.msp
(d) powder.msp

To start MSlice with a particular parameter file type

>> mslice <parameter file>

If the parameter file is in the current matlab search path then only the filename is necessary, otherwise full path is required. The default extension is .msp and can be omitted, for example >> mslice powder will start MSlice in powder mode with the initialisation file powder.msp

3. MSLICE CONTROL WINDOW

3.1. TOP MENU

Exit — Exit MSlice.

Parameters — Use this menu option to load/save parameters typed in the Mslice ControlWindow. These are special formatted ASCII .msp files that assign a value to all parameters in the Control Window. The functionality of Mslice and the options available in the ControlWindow may change in future versions. To check compatibility and convert your old .msp parameter files to the latest version of Mslice run the conversion programme

>>updatemsp('directory of .msp files')

Type

>> help updatemsp

for more information and examples. 'List Parameters' displays in the command window the layout of all parameters stored in the Mslice ControlWindow including complete paths to data files. 'Save List to File' creates an ascii image (.txt) of the Control Window parameters to a file for subsequent printing (If using PFE editor on PC- see end of help.txt file for more info – select A4 paper size and Font Courier 8pt (9pt) regular for a convenient size logbook printout). Note that parameter files for loading in Mslice have to be in the special .msp format, Command line operation:

>>ms_load_msp('C:\mprogs\mslice\HET\HET_spe250_psd_la2cuo4_example.msp

Background — Use this menu to subtract an estimate of the energy-dependent 'background' signal from the current data. The 'background' can be generated by a cut along the energy axis for a group of detectors away from the magnetic/phonon signal. In the cut menu press button 'Store bkg(E)' to generate this signal. Then select 'Subtract stored bkg(E) from data' in the Control Window 'Background' top menu - error bars are changed accordingly to reflect the uncertainty in the background level. Select 'Add stored bkg(E) to data' to go back to the original data set. Select 'Display' to find out how the stored background was generated. See '4.Extending capabilities of Mslice' in file help.txt for how to use a background level generated from other sources.

Help — Displays in the matlab command window extracts from the mslice manual. It is recommended to print out a hardcopy of the complete manual for easy reference (ASCII file help.txt located in the mslice directory, on pc can use the text editor PFE, for download information see end of help.txt file). For a printout with 2 pages per A4 sheet landscape can ftp (ASCII mode) file help.txt to ISIS VMS network and then use the text print command

$ laser0/para=number=2 help.txt for the SYS$LSR0 printer in R3 1st floor.

3.2. CONTROL WINDOW OPERATION

3.2.1. SPECTROMETER:

efixed(meV) Type in value in meV of the incident(final) energy on direct(indirect) geometry spectrometers.

geometry Select between 'direct'- (HET,MARI) and 'indirect'-geometry (IRIS) neutron time-of-flight spectrometers.

DataFile(.spe) Push 'Browse' button on the same line to select data file by browsing through the directory structure (the data file should be in ASCII .spe format=typical output from the HOMER programme on the alpha/VMS ISIS cluster, for detailed file format look up matlab functions load_spe.m or save_spe.m in the MSlice directory). The .spe file needs to be transferred from the VMS platform onto the PC using FTP as ASCII (transfer as BINARY will corrupt end-of-line characters and produce an error at loading). Once the directory of .spe files is selected, the filename can be edited directly in the DataFile(.spe) editable box without the need to do 'Browse' again. When using MSlice only to simulate 'Detector Trajectories' leave this box blank.

Recipe for converting IRIS data into the spe format:

Can use a genie2 command to rebin data along the energy axis and create some intermediate files, which are then read in matlab and finally saved in a single ASCII file in .spe format.

1. Convert the IRIS raw file into the .ipg format (genie 2 binary file) using the igis/icon programme available on the ISIS alpha openVMS cluster (this corrects intensities for the wavelength-dependence of the incident flux and for the detector efficiencies using a vanadium calibration measurement).

   Run igis Program/calib with the vanadium run say 19171.

   Then igis Program/icon,
   Option/Many files,
   Res Function,
   RAW Files ->
   Continue ->
   Use Calibration File Yes,
   Run 19151 (press ENTER key here)
   Extension RAW,
   Spectra 3 53,
   Analyser PG(002),
   Individual Spectra, Energy 1.8463
   -> Run irs19151.raw -(igis/icon)-> irs19151.ipg

2. Put file g2a.com from the mslice directory into an area on the ISIS alpha openVMS cluster (ftp as ASCII). This is a genie2 command file to read .ipg data files, rebin data along energy and save in genie ASCII format. In this file edit the definitions

$ area = "scratch$disk:[iris]" directory of your .ipg files
$ p2 = "-0.2" Emin
$ p3 = "0.005" deltaE
$ p4 = "1.6" Emax

with the appropriate file location and binning information.

On the ISIS alpha openVMS cluster start $genie and execute

>>@g2a

and will be prompted for the run number. This will then create a number of ascii files in some genie-type format, one file for each detector (in the current format g2a.com works for the PG002 mode with 51 detectors, but can be adapted to incorporate other settings as well). Genie2 cannot save data directly in the ASCII .spe format, so matlab will be used to read these separate ascii files and put all the information into a single file.

irs19151.ipg-(genie/g2a)->(several files)irs19151.1,...irs19151.51

3. Ftp (ASCII) all detector files onto the platform running matlab/mslice. At the matlab prompt go to the directory with all these intermediate files and issue command

   >>a2spe

   and will be prompted for a run number, or simply type

   >>a2spe('19151') or >>a2spe(19151) or >>a2spe 19151.

This will read all detector files and will produce the final ASCII file irs19151.spe. All intermediate files created in step 2 can now be deleted. To work from or save in different directories/files use the more flexible commands from any directory

```
>>data=load_ipgascm('directory1\irs19151');
>>save_spe(data,'directory2\irs19151.spe');
```

DetFile(.phx) Push 'Browse' button on the same line to select the file with the detector information layout for the particular .spe file selected(format of detector file the same as required by the fortran PHOENIX programme, load function is load_phx.m in the MSlice directory).When using the programme only to simulate 'Detector Trajectories' ave 'DataFile' blanc, select 'DetFile' and push 'LoadData'. Example format of .phx file for the HET 2.5m West bank (2m_32.phx)

```
32
10.0   0.0  9.31   0.0  0.63  6.9 1
10.0   0.0  9.93   0.0  0.63  6.9 1
10.0   0.0 10.56  0.0  0.63  6.9 1
10.0   0.0 11.18  0.0  0.63  6.9 1
```

first line contains n, number of detector groups then angles twotheta, psi, dtwotheta, dpsi (all in deg) are in a columns 3,4,5,6 of subsequent table (the other columns 1,2,7 contain redundant parameters retained for historical reasons of compatibility with other programmes like PHOENIX)

twotheta = total scattering angles always >0

psi = azimuthal angle, defined such that on HET 2.5m bank has psi=0(W), 90(N), 180(E) and 270(S)on IRIS all PG002 detectors have psi=0.psi=0 defines the principal scattering plane.

dtwotheta = thickness of detector in twotheta (enlarged compared to the physical detector thickness to cover small gaps between adjacent detectors

dpsi = azimuthal angular width

IntensityLabel — Type in label for the intensity axis in all cuts. Can contain latex-type coding, for example:

k_i/k_f d^2\sigma/d\Omega dE (mbarn meV^{-1} sr^{-1} f.u.^{-1})

TitleLabel — Type in extra label for the title on all plots, for example name of sample and temperature, like La_2CuO_4 8K (value of fixed energy and sample rotation angle if single crystal are automatically added to the title label).

LoadData — Push this button to load the data file and the detector layout.

A message should appear in the matlab command window like:

Loading .spe file : m:\matlab\anal_spe\spe\spe750.spe
832 detector(s) and 87 energy bin(s)
Unmasked : masked detector groups = 525 : 307
Loading .phx file : m:\matlab\anal_spe\phx\pix_w_981.phx

which confirms loading the .spe file, tells how many detectors and energy bins are in total, and how many useful versus masked detectors are in the data file. Then confirms loading the detector layout file .phx. If the two files are incompatible (like number of detectors in .spe file not equal to number of detectors in .phx file an error will be given). If the 'DataFile' box is empty then only the detector layout file is read and a warning message is given at the command prompt, like

No data loaded.
Loading .phx file : m:\matlab\anal_spe\phx\pix_981.phx

In this case mslice can only be used for simulating detector trajectories. Note that data files are ascii and need to be transfered as ASCII not binary or auto from a VMS platform (ftp transfer in binary mode of an ascii file corrupts the end of line characters). Command line operation:

```
>> ms_load_data;
```

Requires functions: load_spe.m (matlab load routine) or load_spe_df.f, .dll

(fortran load routine)

SaveDataAs — Save current .spe data to a file. Used for example after backgroundsubtraction or more detector masking.

Requires function: save_spe.m

3.2.2. SINGLE CRYSTAL AND POWDER ANALYSIS MODE

Sample — Select 'single crystal' or 'powder' sample to specify if data is to be described as intensity as a function of projections in wavevector-energy space (h,k,l,E) or as intensity as a function of any two parameters from (|Q|,E,2Theta,DetectorNumber).

Subsequent menu buttons in the MSlice ControlWindow should change when the sample type is modified.

Single crystal (unit cell and orientation) — For 'single crystal' sample type in unit cell parameters a,b,c (Å) and then angles aa,bb,cc (deg) of the crystal lattice (can put in mathematical expressions recognised in matlab like 2*pi*sqrt(2)/1.16). The reciprocal lattice is calculated then according to the general rules set out in Ashcroft and Mermin, Solid State Physics p86. The orientation of the crystal with respect to the spectrometeris defined according to the standard practice used for tripleaxes spectrometers: one specifies two reciprocal crystal axes in the principal scattering plane of the spectrometer and then one more angle (Psi(deg)) uniquely identifies the rotation of the crystal in this plane.

EXAMPLES:

instrument HET, a*||k_i, b* in the horizontal scattering plane (which for HET is defined as the principal scattering plane) is described by
ux=1 uy=0 uz=0, or u=[100]
vx=0 vy=1 vz=0
Psi=0

HET, (a*,b*) in horizontal scattering plane, b*||k_i
u=[100]
v=[010]
Psi=-90

More specifically, one defines a spectrometer reference frame (Qx,Qy,Qz) such that Qx || k_i (incident beam direction) (Qx,Qy) define the principal scattering plane of the spectrometer (horizontal plane on HET, IRIS and vertical on MARI). Axis Qy always corresponds to the azimuthal angle of detectors being zero in the .phx detector file)

EXAMPLES:

HET Qx||k_i, Qy horizontal and Qz vertically up
MARI Qx||k_i, Qy vertically down and Q_z in horizontal plane. u and v are two crystal axes (reciprocal space hkl directions) in the principal scattering plane (Qx,Qy) and Psi is the angle beween k_i and u and is positive if (k_i x u) || Q_z (on HET and IRIS Qz is vertically up). u and v do not have to be orthogonal wavevector directions, they only define a plane and a sense of rotation (u cross v).

Analysis Mode

The 'powder' option was introduced to allow data for 'Single Crystal' samples to be analysed in terms of projections on any two of the axes (|Q|,E,2Theta,DetectorNumber) (as for a 'powder' sample), but such that for each pixel there is also a well-defined (H,K,L) and Energy projection and therefore cuts can be saved in typical 3-axis (H,K,L,E) .hkl format. Use 'single-crystal' mode if data is to be analysed in terms of projections onto wavevector direction.

Detectors

For 'single crystal' sample with 'single crystal' analysis mode choose between position-sensitive (PSD) and conventional (non-PSD) detectors. The single crystal data can be described as intensity measured along some trajectories in wavevector-energy space.For PDS (or area) detectors the origins of these trajectories define a 2d grid and so the trajectories will define a volume (and consequently three axes are needed to describe the data). In this case to produce a 2d colour intensity map of the data one needs to take a SLICE through the data volume, or plot the data onto two axes and integrate along the thirdin some restricted range. To plot PSD data along one direction, or do a CUT, one needs to specify the integration ranges in the other two orthogonal directions. For a conventional array of detectors the origins of the detector trajectories define a 1d grid and the trajectories will define a surface and so only two axes are needed to describe it. In this case one can simply DISPLAY a 2d colour intensity map of the data onto the plane of the two viewing axes. To plot the data along one direction, or do a CUT, one needs to specify the integration range in the other orthogonal direction.

Viewing Axes (Single Crystal)

The single crystal data is described in terms of projections onto [hkl] wavevector directions and energy. These are called viewing axes and for the purpose of plotting they should be chosen to be orthogonal (En error message is given at the matlab command prompt if the viewing axes are not orthogonal. For monoclinic/triclinic lattices suggestions are given to choose orthogonal wavevector axes). The energy axis is by definition orthogonal to any wavevector direction. The axes are specified in the 4-d space with basis (a*,b*,c*,Energy). For example in an HET experiment on a 2d antiferromagnet with the 2-d plane (a*,c*) perpendicular to k_I it is useful to project the data onto the [100], [001] and Energy axes and one defines:

```
u1=[1 0 0 0 ]    and type in label for this direction like Q_h
u2=[0 0 1 0 ]            Q_l
u3=[0 0 0 1 ]             E
```

For projecting data onto the diagonal [1,0,-1], [1,0,1] and Energy the axes are defined as:

```
u1=[1 0 -1 0]    and put some label Q_1
u2=[1 0  1 0]            Q_2
u3=[0 0  0 1]            E
```

The programme will then combine the label with the wavevector direction and so the u1 axis label will be [Q_1,0,-Q_1] in units of x Å^{-1} where x is length in Å^{-1} of the [1,0,-1] vector.

The projections along the viewing axes are now given in units of the length of the viewing axes vectors. For example a=5.4165, b=13.173, c=5.4164u1=[1 0 0 0], u2=[0 0 1 0] u3=[0 0 0 1] the units along u1,u2,u3 are 1.16 Å^{-1},1.16 Å^{-1} and 1 meV.

The menus for representing the data as 2d colour intensity plots (SLICE) and 1d plots (CUT) should be updated every time one types in a new label for the viewing axes.

Powder

Select 'powder' sample if data is to be described in terms of two axes out of (Energy,|Q|,2Theta,Azimuth,DetGroupNumber).

Viewing Axes (Powder)

Powder data can be described in terms of two axes out of (Energy,|Q|,2Theta,Azimuth,DetGroupNumber) (also data from single crystal samples can be looked at in powder mode). Select the viewing axes u1/u2. There are default labels for each axis but these can be changed by typing in new names under 'Labels'. Calculate Projections will update the labels and the projections on the viewing axes.

Calculate Projections

Push this button to calculate or refresh after more masking the projections of data points onto the viewing axes. A message
confirming the calculations should appear in the command line window :

'Calculating projections of data points onto the viewing basis' (for 'single crystal' mode with PSD detectors)

or

'Calculating projections of data points and bin boundaries onto the viewing basis' (for 'single crystal' mode with conventional detectors)

or

'Powder mode: calculating projections of data points and bin boundaries onto the viewing basis' (for 'powder' mode).

The original .spe data is given as a table of Intensity and error values as a function of detector number and energy. This information is combined with the detector layout from the .phx file and given the kinematics of the scattering for direct/indirect geometry and the value of the fixed energy, the data is now transformed into Intensity as a function of (Qx,Qy,Qz,E) in the spectrometer reference frame (see help on ':Single crystal: unit cell and orientation :' for the convention used in defining the (Qx,Qy,Qz) reference frame). In 'single crystal mode' the information on the crystal lattice (a,b,c,aa,bb,cc) and the crystal orientation (u,v,Psi) is then used to projected the (Qx,Qy,Qz,E) points onto the chosen viewing axes u1, u2 (and u3 for PSD detectors). For 'powder' mode each (Qx,Qy,Qz,E) point is 'projected' onto the viewing axes chosen from |Q|, E, 2Theta and Detector Group Number.

Command line operation: >> ms_calc_proj;

Requires functions:

calcproj.m (single crystal PSD detectors)
calcprojb.m (single crystal conventional detectors) calcprojpowder.m (data analysed in powder mode)

3.2.3. DISPLAY 2D INTENSITY COLOUR MAPS

Display — Use this menu to display 2d intensity colour maps of the data projected onto the plane defined by two viewing axes (in 'powder' mode the axes are any two out of (|Q|,E,2Theta,DetectorNumber) and in 'single crystal' mode with 'conventional' detectors the axes are wavevector or energy directions. The 'horizontal' and 'vertical' axes refer to the x and y axes of the plot and the values are expressed in units of the viewing axes (for example rlu along a* and c* for 'single crystal' mode and viewing axes u1=[1 0 0 0] and u2=[0 0 1 0]). The default plotting range is chosen such that all available data fits in the graph.

Intensity range — has optional parameters used to restrict the colour intensity axis (if not given (editable boxes left blanc) by default min=0 and max=max(intensity)).

Colour map — black->red (as in PHOENIX)
log(black->red) logarithmic (10^n) colour table
blue->red (matlab jet map)

Smoothing level — The results can be 'smoothed' by convoluting over the displayed plot a matrix like

```
A=[0.0, 0.0, 0.0;
   0.1, 0.4, 0.1;
   0.0, 0.0, 0.0]
```

which averages the intensity of a central point with the intensitiesof the nearby bins on the displayed grid with the most weight given to the original value of intensity and smaller weights to intensity of nearby points. For the particular matrix given above only nearby energy bins of the same detector are averaged; nearby detectors are not averaged. In the general case, the smoothing algorithm is such that only bins with data points contribute to the smoothing and for bins with no original data, no data is put after smooting. The matrix A is normalised over the number of contributing neighbours. The smoothing algorithm is in the matlab function smooth_spe.m in the mslice directory and the matrix A is defined at the beginning (edit this if you wish a different weighting, there are a number of options commented out in the source code).The default smoothing matrix will be displayed in the command lineevery time smoothing is used. The smoothing level gives how manytimes the results should be smoothed (if absent - field left blanc- or if 0 no smoothing).

Shading — flat = no bin boundaries
faceted = with black lines as bin boundaries
interp = no bin boundaries and colour in each bin is interpolated between colours of nearby bins

Display — Push this button to plot the 2d intensity colour plot of the data projected onto the two viewing axes. Automatic labels of the viewing axes and units should appear. The title gives all the information about the data file, smoothing and display ranges and the format is: filename (any TitleLabel if given), Efixed, Psi (for 'single crystal' sample), s= (if smoothing)display range. The figure can be resized with the mouse by dragging one of the corners. If

the two display axes are wavevector directions the aspect ratio is chosen such that the equivalent length of 1 Å^{-1} along one axis is the same as the equivalent length of 1 Å^{-1} along the other. Choose menu option 'Keep' to 'keep' current plot and direct new displayed plots to a new figure (otherwise these new plots will appear in the same figure window and old plots will be lost). To save graph to a file in various formats (including eps) in the figure window go to the 'File' menu, then select 'Export'. To print a hardcopy, go to 'File' and push 'Print' button (the aspect ratio will be preserved on the hardcopy). For a Postscript printer can also use menu option ColourPrint (this prints graph to an .eps file mslice.eps which is then sent directly to the printer using MS-DOS commands, a few printers for the ISIS printer network are already defined, the path for the 'default printer' has to be defined by editing function ms_printc. For some old printers like
the HP(PS11PS) in the HET cabin only the second method works. To add more printers under the 'ColourPrint' list follow instructions in mslice plotting function disp_spe.m.

Command line operation:

>>ms_disp

3.2.4. DISPLAY VOLUMETRIC DATA FROM PSD DETECTORS : SLICES

Slice plane: For single crystal PDS data this menu is used to plot the data as a 2d intensity colour map on the plane defined by two of the viewing axes with data integrated along the third axis in some restricted range. One selects the third axis (perpendicular to the slice plane) from the popup menu list 'perp to axes', say u3=Energy and specifies the integration thickness range, say 250 to 300. The choice for the perpendicular axis automatically defines the slice plane axes as the remaining two from the 3 viewing axes and subsequent menus for the horizontal and vertical label directions should change accordingly. For example this menu would look like horizontal range Q_h vertical range Q_lfor one of the previous given examples at help for 'Viewing axes'. Type in values for the limits along the Q_h and Q_l axes with 'step' as the bin width. All numbers given here are in units ofthe viewing axes (reciprocal units in the case of u1=[1 0 0 0], u2=[0 0 1 0] and meV for u3=[0 0 0 1]). The binning algorithm (the same as used by PHOENIX 4 or later versions) is generalisation of the binning routine for one-dimensional cuts and is as follows: a grid of bins of size step_x times step_y is constructed with the lower left corner at (x_min-step_x/2,y_min-step_y/2) and not exceeding x_max+step_x/2 along x and y_max+step_y/2 along y. The x,y and intensity value for each bin is calculated by taking the simple average (mean) of the x, y and intensity values of all pixels in that bin, where the limits for including pixels are taken as [) along both x and y axes, i.e. inclusive to the left and exclusive to the right. The limits are [], i.e. inclusive at both ends along the integration direction transverse to the plane (plane thickness). The error is calculated as the error of the resulting average of so many intensity values where each point has a given error e, but is not actually plotted (in future versions will be included if the output is to be saved to a file for subsequent fittings). The 2d binning algorithm is in the matlab function slice_spe.m with the option (by default on pc/linux, but not on vms) to call the optimised fortran routine in function slice_df.f.

Intensity range — has optional parameters used to restrict the colour intensity axis (if not given (editable boxes left blanc) by default min=0 and max=max(intensity)).

Colour map — black->red (as in PHOENIX)
log(black->red) logarithmic (10^n) colour table
blue->red (matlab jet map)

Smoothing level — The results can be 'smoothed' by convolving over the slice grid a matrix like

```
A=[0.1, 0.2, 0.1;
   0.2, 0.8, 0.2;
   0.1, 0.2, 0.1];
```

which averages the intensity of a central point with the intensities of the nearby points on the slice grid with the most weight given to the original value of intensity and smaller weights to intensity of nearby points. Only bins with data points contribute to the smoothing and for bins with no original data, no data is put after smooting. The matrix A is normalised over the number of contributing neighbours. The smoothing algorithm is in the matlab function smooth_slice.m in the \mslice directory and the matrix A is defined at the beginning (edit this if you wish a different weighting, there are a number of options commented out in the source code).The default smoothing matrix will be displayed in the command line every time smoothing is used. The smoothing level gives how many times the results should be smoothed (if absent - field left blanc - or if 0 no smoothing).

Shading — flat = no bin boundaries
faceted = with black lines as bin boundaries
interp = no bin boundaries and colour in each bin is interpolated between colours of nearby bins

Plot Slice — Push this button to plot the 2d intensity colour plot of the data projected onto the slice plane. Automatic labels of the slice axes and units should appear. The title gives all the information about the slice axes, range and bin widths and the format is: filename (any TitleLabel if given), Efixed, Psi, s=(if smoothing) slice 250<E<300, Q1=-1.1:0.045:0.30, Q2=-11:0.045:0.30The figure can be resized with the mouse by dragging one of the corners. If the two slice axes are wavevector directions the aspect ratio is chosen such that the equivalent length of 1 Å^{-1} along one axis is the same as the equivalent length of 1 Å^{-1} along the other. Choose menu option 'Keep' to 'keep' current plot and direct new slice plots to a new figure (otherwise these new plots will appear in the same figure window and old plots will be lost). To save graph to a file in various formats (including eps) in the figure window go to the 'File' menu, then select 'Export'. To print a hardcopy, go to 'File' and push 'Print' button (the aspect ratio will be preserved on the hardcopy).

For a Postscript printer can also use menu option ColourPrint (this prints graph to an .eps file mslice.eps which is then sent directly to the printer using MS-DOS commands, a few printers for the ISIS printer network are already defined, the path for the 'default printer' has to be defined by editing function ms_printc).

For some old printers like the HP(PS11PS) in the HET cabin need to use this method for b/w hardcopy also. To add more printers under the 'ColourPrint'

list follow instructions in mslice plotting function plot_slice.m.

Command line operation :
>>ms_slice;

Requires functions:
slice_spe.m
slice_df.f, .dll (fortran binning routine)

Surf Slice — Push this button to produce a 3D mountain intensity plot of the slice data with intensity both as colour and as the vertical axis. Smoothing is recommended. Shading: faceted plots a mesh and makes shapes more visible. With mouse can rotate the viewing perspective, can toggle the option to enable/disable 3d mouse rotations using figure menu option 'Rotate'.

3.2.5. ONE-DIMENSIONAL PLOTS : CUTS

Cut — Use this menu to plot the data along one of the viewing axes with integration along the other two directions for 'single crystal' PDS mode (or other one direction for other modes). The plotted intensities are the average values over the integration volume (or thickness) and not the total integrated counts. Choose 'along axis' direction (the x-direction of the final plot) and then the subsequent menus with labels of the integration direction(s) should change accordingly. Specify plot range along the projecting axis and bin width (step), then the integration range(s) in the other two (one) transverse direction(s). Put 'step=0' to bin data in each contributing detector along the energy axis only and project result onto the x-axis with no further binning - this will result in as many data points as contributing detectors (useful for 'single-crystal' conventional detectors or 'powder' mode to cut along a constant energy line - usually a line of constant background - and project data points along a wavevector direction or |Q| with no further binning). The binning algorithm (the same as used by PHOENIX 4 or later versions) is as follows: the x-range is divided into bins of length 'step' starting at x_min-step/2 and not exceeding x_max+step/2. The x and y value for each bin is calculated by taking the simple average (mean) of the x and y values of all pixels in that bin, where the limits are taken as [), i.e. inclusive to the left and exclusive to the right. The pixel inclusion criterion is [] for the transverse direction(s) (thickness of cut), i.e. inclusive at both ends. The errorbar is calculated as the error of the resulting average of so many y-values where each point has a given error e. When saving 'single crystal' data in .hkl format the (H,K,L,Energy) per each bin is again calculated by taking a simple average (mean) of the (h,k,l,energy) values of all contributing pixels per bin. The cut algorithm is in the matlab function cut_spe with the option (by default on pc/linux, but not on vms) to call the optimised fortran routines for the binning in functions sort1d_df.f and avpix_df.f.

Intensity range — Has optional parameters used to restrict the vertical intensity axis (if not given (editable boxes left blanc) by default axes are chosen automatically to get all data points with whole errorbars in). Can choose to plot the variation of other parameterslike Qx,Qy, Energy, |Q| ... (or H,K,L if 'single crystal') along the cut direction and for the same pixels which contribute to the cut. The errorbar in this case is the standard deviation (variance) of the y-values

for all pixels in each bin according to standard statistical textbooks (errorbar=0 if just one pixel in the bin).

Symbol

Choose plot data symbol (13 options): empty circle (default), square, ...
Choose colour (8 options) : white (default), red ..
Choose no line (default), solid line, dashed line ...

To avoid using the same symbol for two plots on the same figure, in the case of overplots the plotting symbol is chosen automatically as the first from the given list that is different from the symbols already used on the plotting graph (it is recommended to choose different colours for overplots).

OutputFile

Give a filename if the data in the cut is to be saved to a file, by default OutputFile is 'none'. If the required output is simply x,y,error (3-column) ASCII format (for other applications - fitting in Mfit without resolultion convolution and plotting in SigmaPlot, Origin ...) select format '.xye'.To save data in a format to keep all the pixel information (currently used as input by the resolution-convolution Fortran fitting programs available at ISIS-tobyfit or bobfit, for detailed file format see matlab functions \mslice\save_cut.m,load_cut.m) select option '.cut' (the option 'Mfit .cut' is an extension of this format which also saves the plot title, axes labels and single crystal orientation + unit cell parameters at the end of the file). Push 'Browse' button to select a particular directory the file should be saved to. A message like

>> Saving cut (56 points and 1125 pixels) in .cut format to file :
>> C:\Mprogs\mslice\temp.cut

confirms saving the cut to a file and specifies format.

Use option '.smh' to save data in a format required by the resolution convolution fitting programme smhfit (Stephen Hayden).

In single crystal mode could also select '.hkl' format, which is an extension of the simple '.xye' format which also includes H, K, L and energy values for each point in the cut and also axes labels and title in the header (for this multi-column format see functions mslice\save_cut.m and load_hkl.m).

Plot Cut

Push button to plot the 1d cut in a matlab window.

The y-axis will be labeled with what was typed in the Spectrometer Menu : Intensity label. The x-axis will have a combined label of the cut direction and the title willcontain all information about the integration ranges. To keep current plot and create new graphs in other figures choose menu option 'Keep' (otherwise the next plot will appear in the same figure and old plot(s) will be lost)and subsequently the 'keep' menu option of that figureis turned off and the 'Make Current' option is enabled (choose this option to make an old plot figure the current figure for the next plot or overplot). To save figure to a file in various formats (including .eps) use the figure menu 'File' then 'Export'. For color hardcopy output on postscript printers can use menu 'ColorPrint' (this prints graph to an .eps file mslice.eps which is then sent directly to the printer using MS-DOS commands, a few printers for the ISIS printer network are already defined, the path for the 'default printer' has to be defined by

editing function ms_printc following instructions given there. For some old printers like the HP (PS11PS) in the HET cabin need to use this method also to produce b/w plots. To add more printers under the 'ColourPrint' list follow instructions in mslice plotting function plot_cut.m.

Command line operation:

```
>>ms_cut;
```

Requires functions:

cut_spe.m
sort1d_df.f, .dll (fortran sorting routine)
avpix_df.f, .dll (fortran pixel averaging routine)

If the above compiled fortran functions are not available then matlab routines (sort1d_m.m, avpix_m.m) will be used instead.

Plot Cut Over — Overplot this cut on top of the previous cuts in the current plot cut figure. This updates figure title to have legends of all plots and changes data symbols (but not colours) automatically to be different from previous plots. Recommended change of colour befor pushing Plot Cut Over button. To save figure to a file in various formats (including .eps) use the figure menu 'File' then 'Export'. For color hardcopy output use menu 'ColorPrint' (default printer can be changed by editing mslice file ms_printc.m).For some old printers like the HP in the HET cabin need to use this method also to produce b/w plots.

Cut: x-axis: — The default x axis for the plot is the cut axis. However, can choose to plot binned data along other axes too. For example average data in a few detectors and plot along energy in the normal way. Now choose x-axis = |Q| or a wavevctor direction to see the same cut (using the same bins, same pixels etc) but now plotted as intensity vs. |Q| or wavevector direction. Useful for determining dispersions, i.e. finding centre of a peak along both wavevector and energy.

To Mfit — This button sends the cut data as an x,y,error file directly to the matlab fitting programme Mfit (version for matlab5) (download from http://www.ldv.univ-montp2.fr/matprgs.html).Axis labels and the title are transferred as well. The mfit programme needs to be installed and running before a file could be sent (a warning message should appear if this is not the case). The cut data will not be sent to the mslice plot cut window as well.

Store bkg(E) — Use this button to store present cut as 'background', a signal that can later be subtracted from the data set. Only energy-dependent backgrounds are supported. Use top menu 'Background', 'Subtract ..' to subtract stored background from the data.

3.2.6. SIMULATE DETECTOR TRAJECTORIES

Detector Trajectories: Use this section to plot/simulate the phase space covered by the detector trajectories on various axes. Can choose two plotting axes (x,y) from (Qx,Qy,Qz,H,K,L,u1,u2,u3,Energy,|Q|,2Theta,Azimuth,DetectorGroupNumber)for 'single crystal' mode (only two axes u1,u2 if conventional detectors,and axes u1,u2 missing altogether if analysis mode is 'powder') or (Energy,|Q|,2Theta,Azimuth,DetectorGroupNumber) for 'powder' samples. For the definition of the spectrometer reference frame (Qx,Qy,Qz) see help on :Single crystal: unit cell and orientation. The z-axis (if not 'none') can be used to restrict the range of plotting, for example DetGroupNumber from 1 to 94 plots the trajectories of the lower angle banks on MARI. For HET with PSD detectors can for example plot the extent of wavevector space covered in a certain energy window z-axis = Energy from 50 to 100 meV and project it onto the wavevector plane perpendicular to ki, i.e. (x=Qy horizontal perpendicular to ki, y=Qz-vertically up). 'Contours' of constant energy transfer can be overlayed on top of the graph starting at E_min and not exceeding E_max with step1 between consecutive lines. If step2 is given then thick lines are drawn at E_min and at subsequent values step2 apart up to E_max. A matlab plotting function can be entered in the 'Command' editable box and will be executed (if found in path) after plotting the trajectories, this can be used for example to overplot a dispersion relation on top of the simulated phase space covered in an experiment. For 'single crystal' mode and if both plotting directions x and y are wavevector axes tick '(hkl) points' and '(hkl) labels' to draw (hkl) lattice points and their labels (only some points will be labelled to avoid text overlap). All (hkl) lattice points in the given z-range will be projected onto the (x,y) plane.

4. EXTENDING CAPABILITIES OF MSLICE. MATLAB COMMAND LINE OPERATIONS

Adding .spe files

Use function add_spe(weights,spedir,files,fileout).

For an example

```
>> type add_spe_example
```

More files in memory simultaneously:

Can store several data files in memory at the same time and select the one to plot without having to reload it or re-calculate projections. Use the ControlWindow to load in the first data file, calculate projections and produce various plots. Then put the data into a variable for example <data1> using the commnad line :

```
>> data1=fromwindow
```

this extracts the .spe data and all the various labels and projections into the matlab data structure called 'data1'

Now load second .spe data file using the ControlWindow, calculate projections and produce plots, then put this data set in another variable called 'data2'

>> data2=fromwindow;

If you now wish to make more plots from the first data set just put it back into the ControlWindow

>> towindow(data1);

The ControlWindow fields Efixed(meV), direct/indirect-geometry, DataFile(.spe), DetFile(.phx), Intensity Label and TitleLabel are automatically updated with the values already stored for 'data1'. It is best to do CalculateProjections again just in case the viewing axes or analysis mode have changed since the data was extracted into the variable 'data1', howeverthis is not necessary if no changes have been made in the meantime. After this go directly to display/slice or cut.

Saving data in binary .mat format

The matlab binary format .mat is much more compact and faster to read/write than the ASCII format. The data can be saved to the file <filename> in binary matlab format following

>>data=fromwindow;
>>save <filename> data

At a later date, load data with

>>load <filename>
>>towindow(data);

<data> is the name of the variable also stored in <filename>

If <filename> has an extension different from the matlab default .mat then use

>>load <filename> -mat

to force loading as a binary mat file (the default format for extensions different from .mat is table ASCII).

If data is saved after projections have been calculated then after subsequent loading projections need not be calculated again provided the same viewing axes are used and one can go directly to diplay/slice or cut.

Masking detectors

>> data=fromwindow

extract data structure from ControlWindow

>> data=mask(data,[detector numbers])

mask certain detectors, i.e. detectors numbers can be [95:300] or [14] for a single detector

>> towindow(data)

put data back into the ControlWindow. Press 'Calculate Projections' to update projections and the masked detectors

Simulating scattering

Load in detector layout file only (no data file) and put in lattice parameters, crystal orientation and calculate projections.

>>ms_simulate(emin,emax,de,<crosssection number>,[p1 p1 p3 ...])

where the cross-section function should be added by editing the file ms_sqw.m. The detector layout and spectrometer/sample information is extracted from the ControlWindow and the cross-section is calculated for all detectors and for the energy range emin:de:emax for the given crosssection number and parameter values [p1 p2 ...]. No convolution with the spectrometer resolution is performed. When calculation is completed the resulting simulated data set is put back into the ControlWindow. Calculate projections then do slice/display/cut on the simulated set.

Subtracting backgrounds

>> background=getb

get background in the command line

>> putb(background)

put background structure in the ControlWindow required fields
x(1,ne): Energy bin centres (meV)
y(1,ne): Intensities
e(1,ne): Errors

linear interpolation of (y,e) over the actual energy bin centres of the spe data gives the 'background' level that will be subtracted

optional fields

x_label: usually 'Energy (meV)'
y_label: 'Intensity(abs.units)'
axis: [-10 599 0 8.08] for the plot
datafile: 'spe750_sw_msk.spe'
title: contains history, can be 'string' or {'string1','string2',...}

Displaying 3d/4d data from other sources

The ControlWindow can also be used to pass commands to cut or display/slice 3d/4d data produced by other applications, for example a monte-carlo simulation of the two-magnon scattering cross-section in a S=1/2 square-lattice Heisenberg antiferromagnet

>> S=1/2;
>> deltaS=0.196;
>> data1=mc2spe('swq_1m.dat',(S-deltaS)(2*deltaS+1))
>> towindow(data1);

do cuts

```
>> data2=mc2spe('swq_2m.dat',deltaS*(deltaS+1))
>> towindow(data2);
```

5. USEFUL SOFTWARE

Mfit fitting programme for MATLAB:

http://www.ldv.univ-montp2.fr/matprgs.html

is already downloaded on the ISIS PC NT network at

//ndacromwell/BABYLON4/Scratch/radu/mprogs all files in subdirectories

/funcs
/load,
/mfit4
/nllsq