Contents

Advanced data treatment

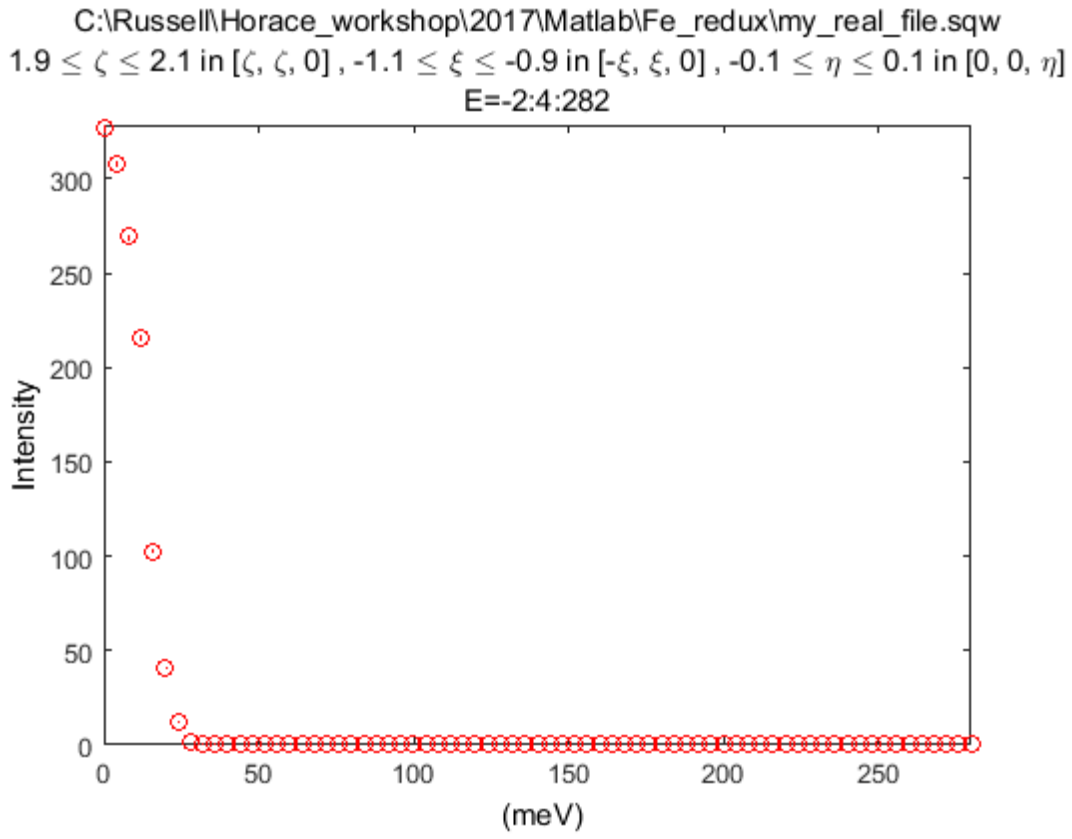
```
%How to subtract backgrounds, improve your statistics using symmetrisation,
%apply masking (e.g. of spurions), and other miscellaneous useful routines
```

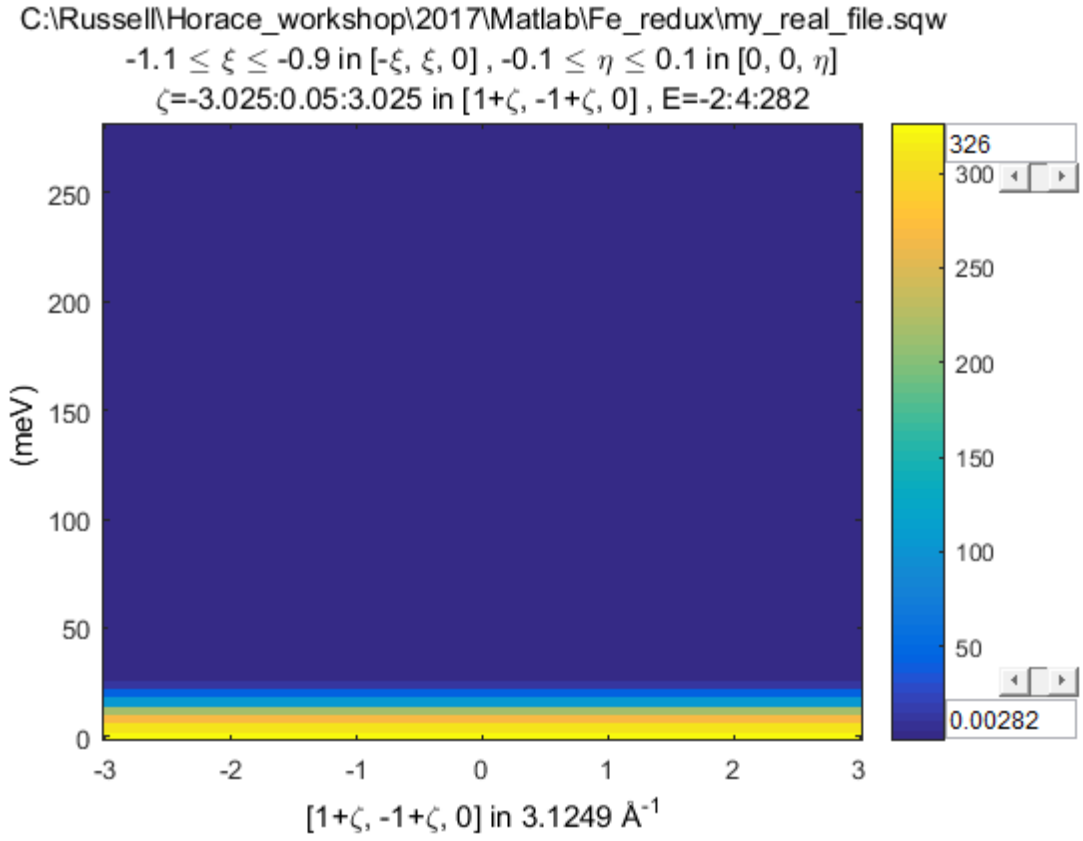
Background subtraction (including cuts, replication, binary operation)==

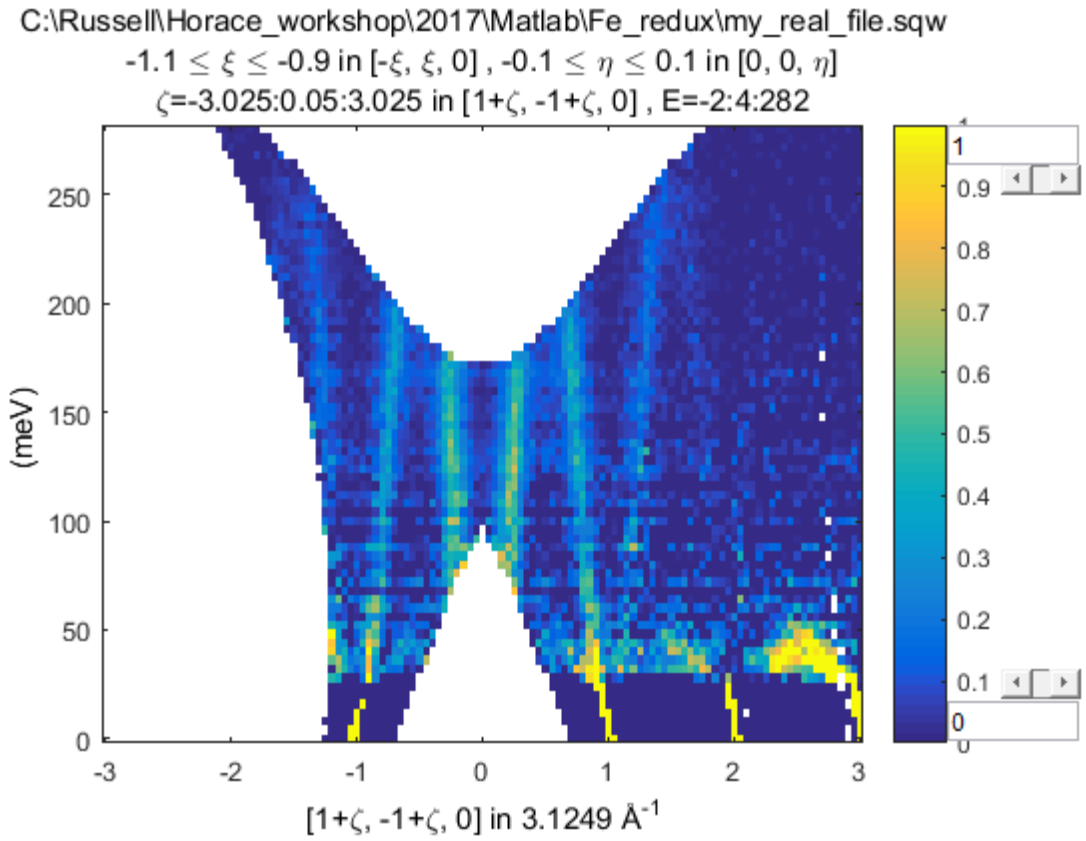
```
my_slice=cut_sqw(sqw_file,proj,[-3,0.05,3],[-1.1,-0.9],[-0.1,0.1],[0,4,280]);
my_bg=cut(my_slice,[1.9,2.1],[]);
plot(my_bg);
keep_figure;

my_bg_rep=replicate(d1d(my_bg),d2d(my_slice));
plot(my_bg_rep)
keep_figure;

my_slice_subtracted=d2d(my_slice) - my_bg_rep;
plot(my_slice_subtracted);
lz 0 1
keep_figure;
```

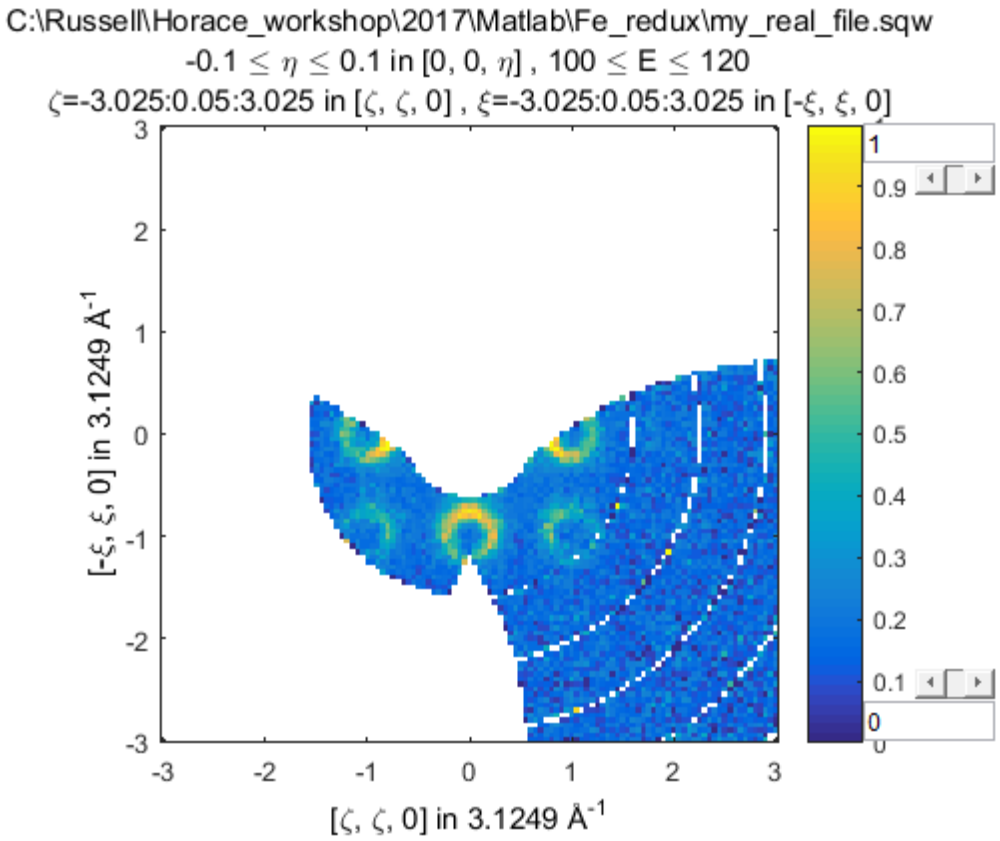

C:\Russell\Horace_workshop\2017\Matlab\Fe_redux\my_real_file.sqw
1.9 ≤ ζ ≤ 2.1 in [ζ, ζ, 0] , -1.1 ≤ ξ ≤ -0.9 in [-ξ, ξ, 0] , -0.1 ≤ η ≤ 0.1 in [0, 0, η]
E=-2:4:282
Intensity
(meV)
0
50
100
150
200
250
300


C:\Russell\Horace_workshop\2017\Matlab\Fe_redux\my_real_file.sqw
-1.1 ≤ ξ ≤ -0.9 in [-ξ, ξ, 0] , -0.1 ≤ η ≤ 0.1 in [0, 0, η]
ζ=-3.025:0.05:3.025 in [1+ζ, -1+ζ, 0] , E=-2:4:282
(meV)
[1+ζ, -1+ζ, 0] in 3.1249 Å-1
326
300
250
200
150
100
50
0.00282



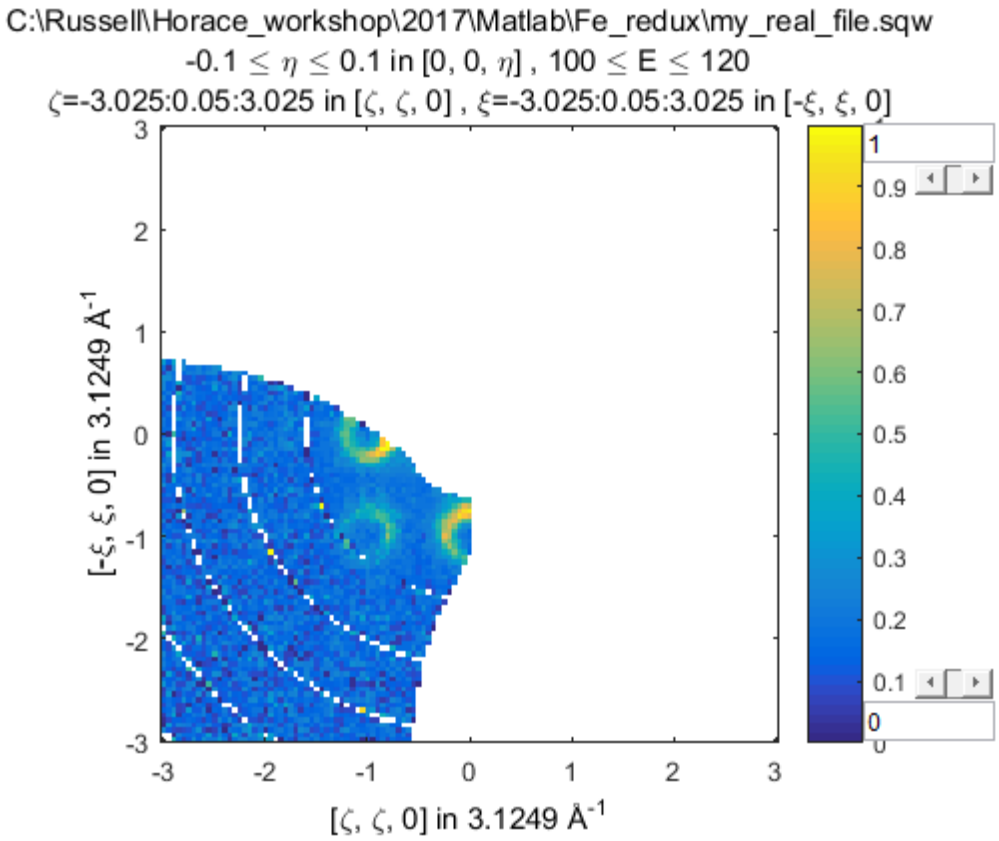
Symmetrisation

```
my_slice2=cut_sqw(sqw_file,proj,[-3,0.05,3],[-3,0.05,3],[-0.1,0.1],[100,120]);
plot(my_slice2);
lz 0 1
keep_figure;
```



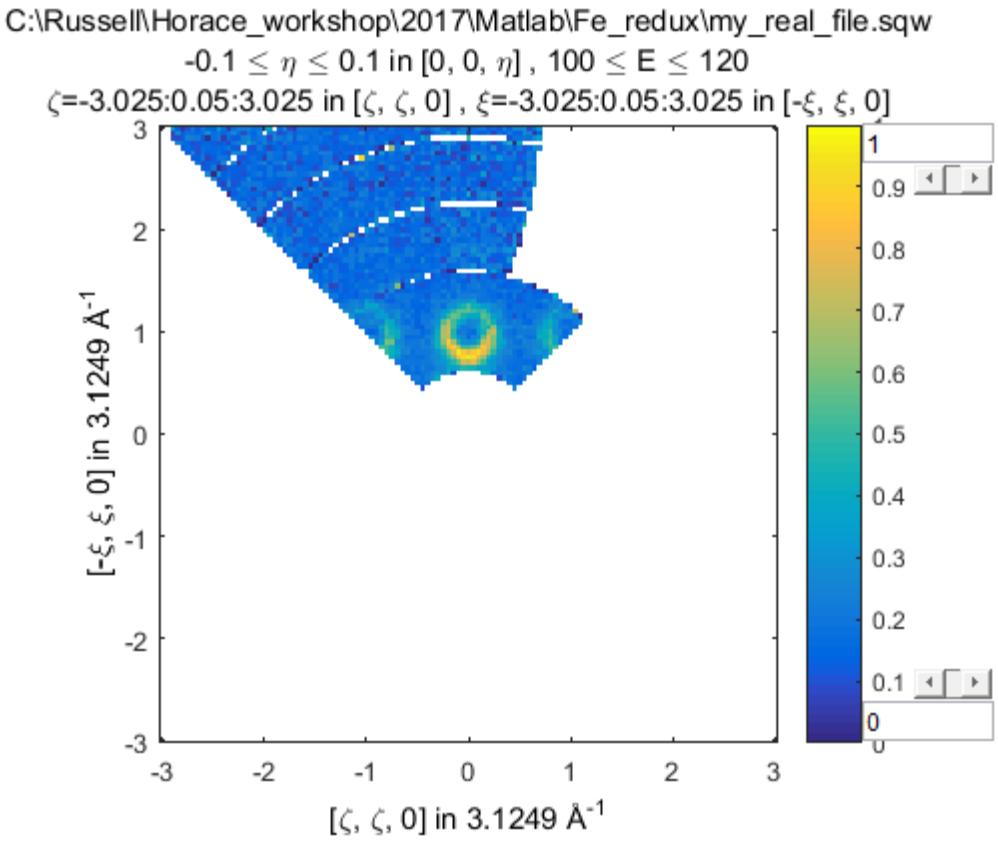
Fold along vertical:

```
my_sym=symmetrise_sqw(my_slice2,[-1,1,0],[0,0,1],[0,0,0]);
plot(my_sym);
lz 0 1
```



Two folds along diagonals

```
my_sym2=symmetrise_sqw(my_slice2,[1,0,0],[0,0,1],[0,0,0]);
my_sym2=symmetrise_sqw(my_sym2,[0,1,0],[0,0,1],[0,0,0]);
plot(my_sym2);
lz 0 1
```

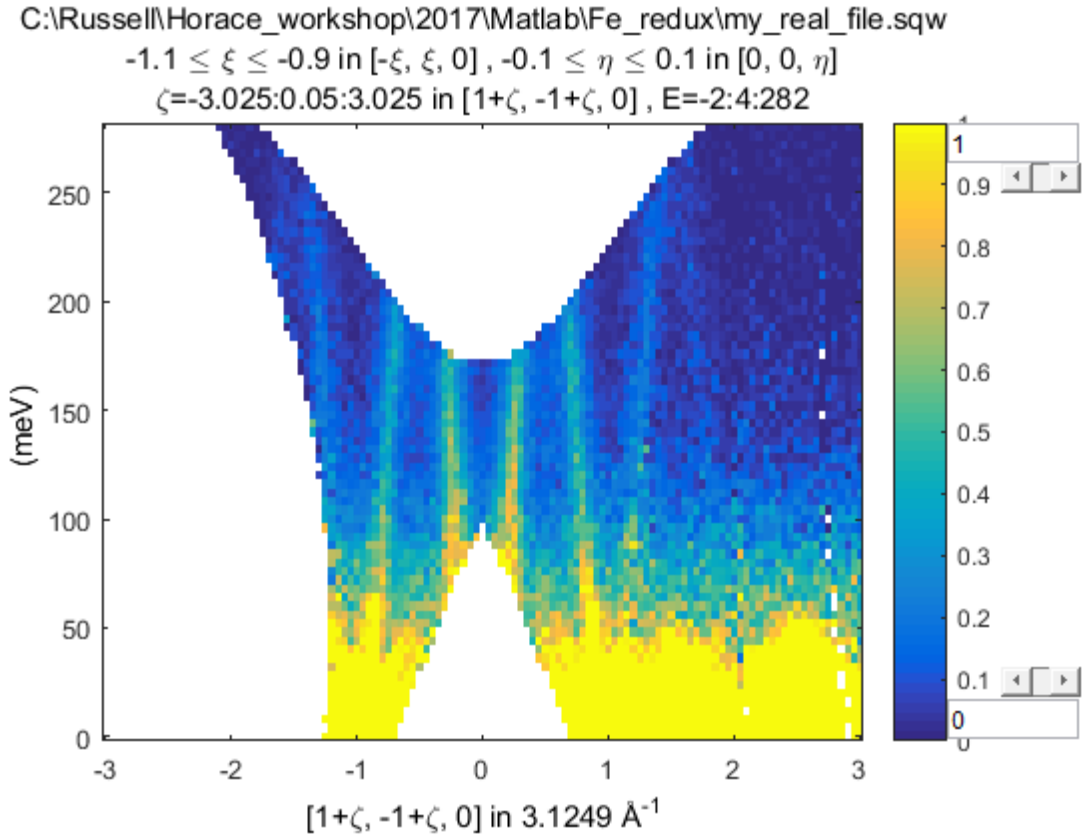


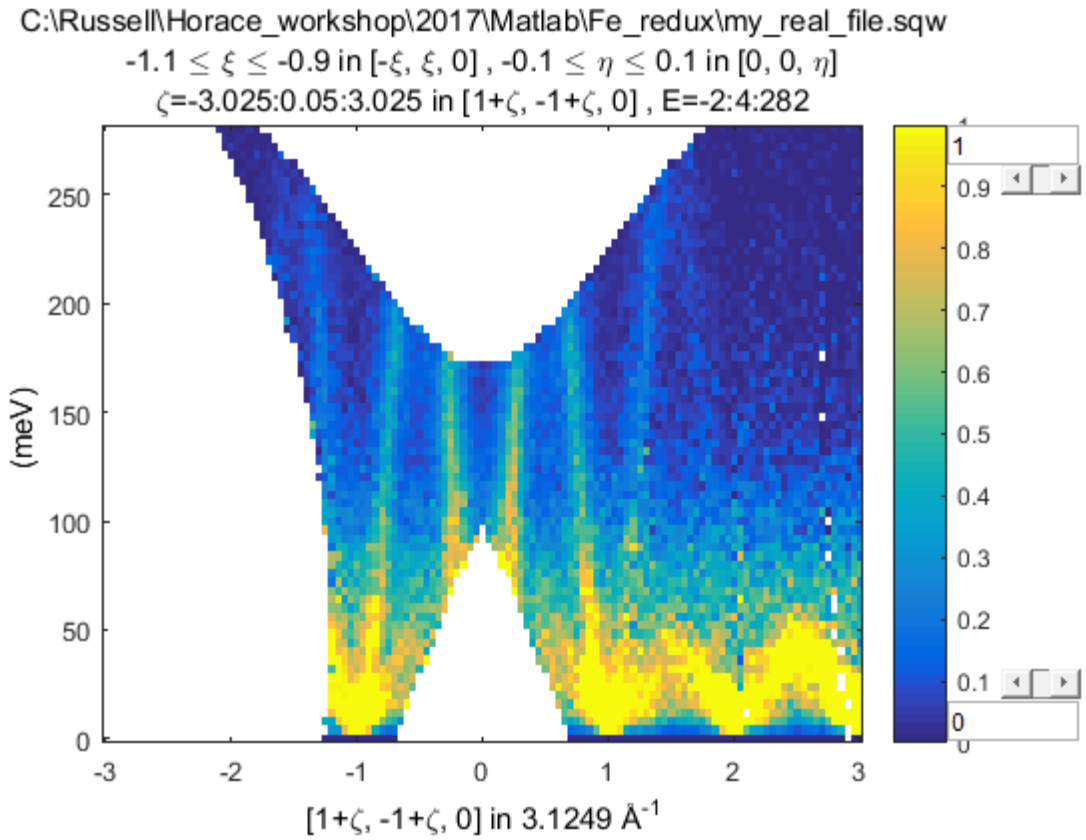
Rescaling data

```
%Bose correction function. NB it does not do much at these high energies, or course!

my_slice=cut_sqw(sqw_file,proj,[-3,0.05,3],[-1.1,-0.9],[-0.1,0.1],[0,4,280]);
plot(my_slice);
keep_figure;
lz 0 1

my_slice_bose=bose(my_slice,300);%pretend data taken at 300K...
plot(my_slice_bose);%can still see what this does
keep_figure
lz 0 1
```

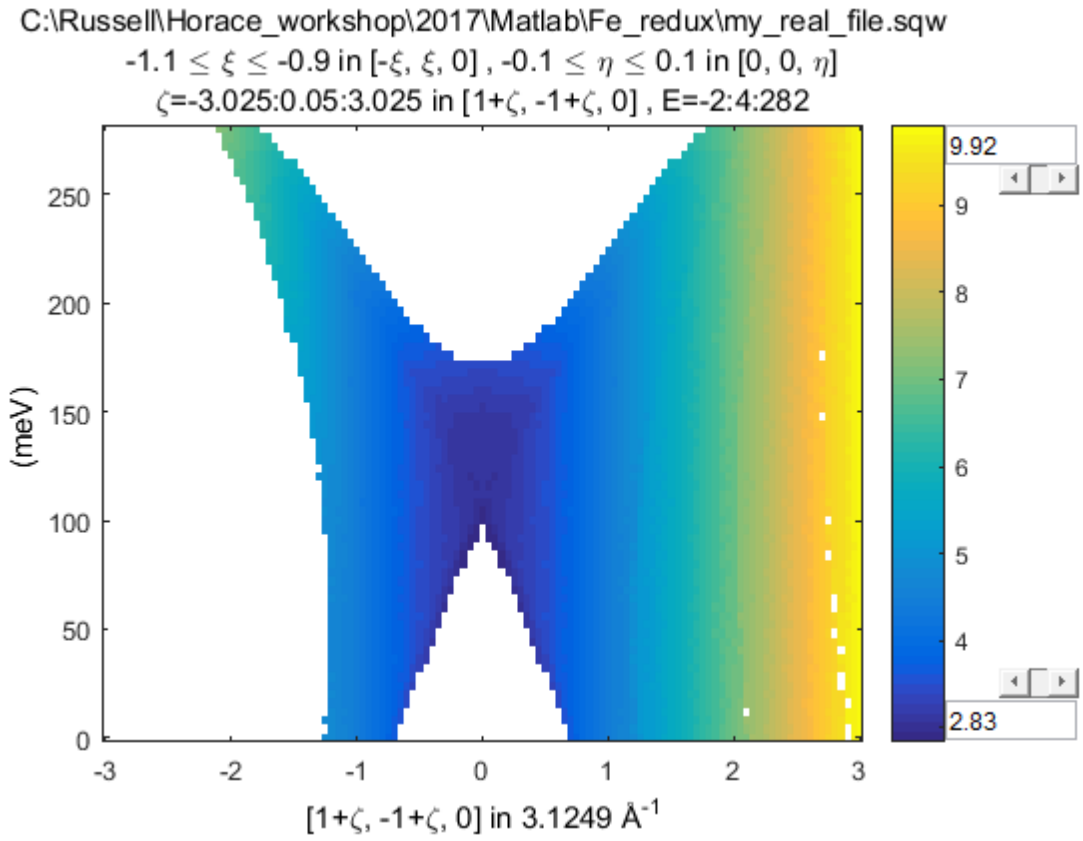




Misc

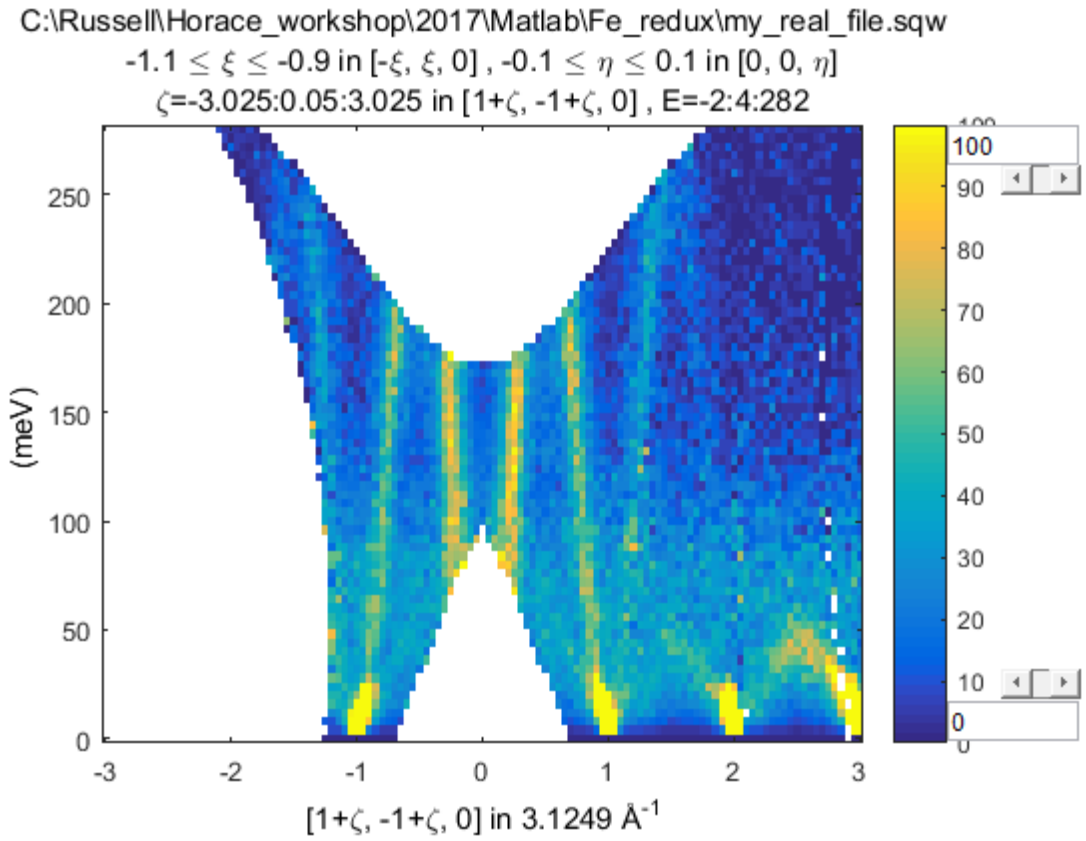
If you want to see how a certain parameter varies across a dataset (e.g. Q, energy, h, etc.)

```
w_sig=signal(my_slice,'Q');%mod Q in this case
plot(w_sig)
```



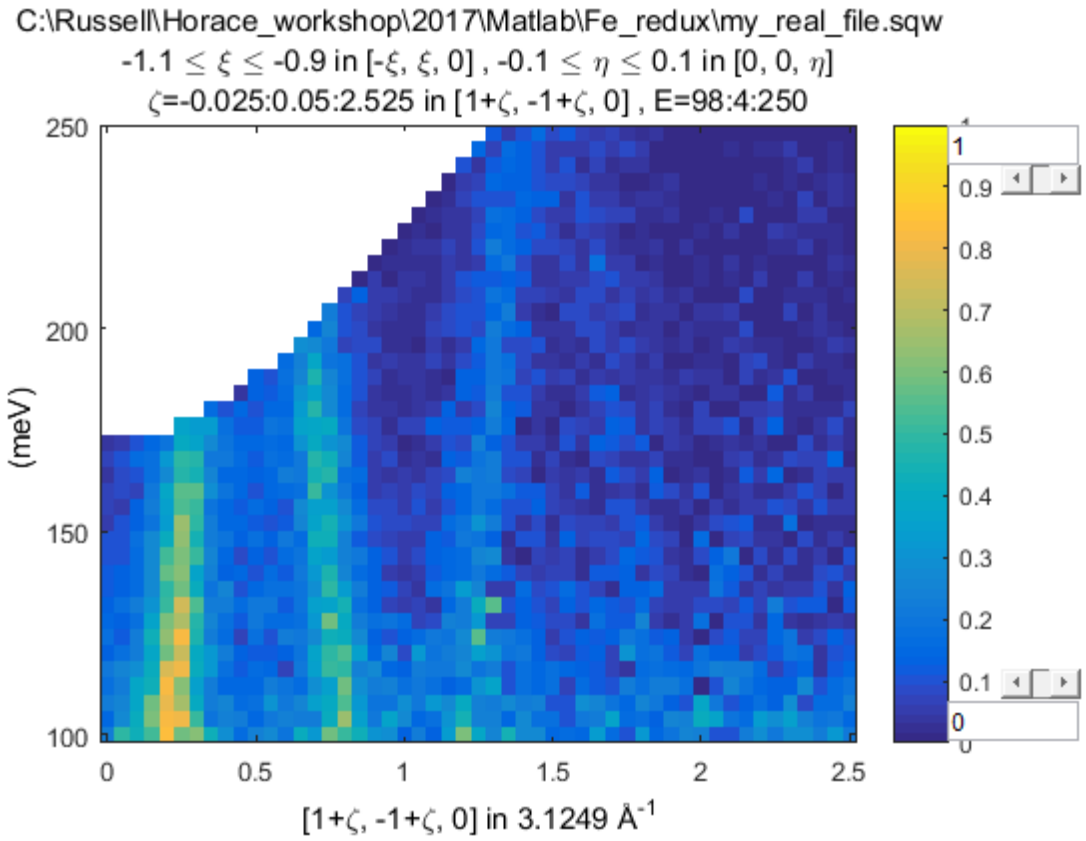
You can use this now to apply a scale factor to the data.

```
%Suppose you wish to multiply signal by energy:
w_sig=signal(my_slice,'E');
my_slice2=my_slice*w_sig;
plot(my_slice2)
lz 0 100
```



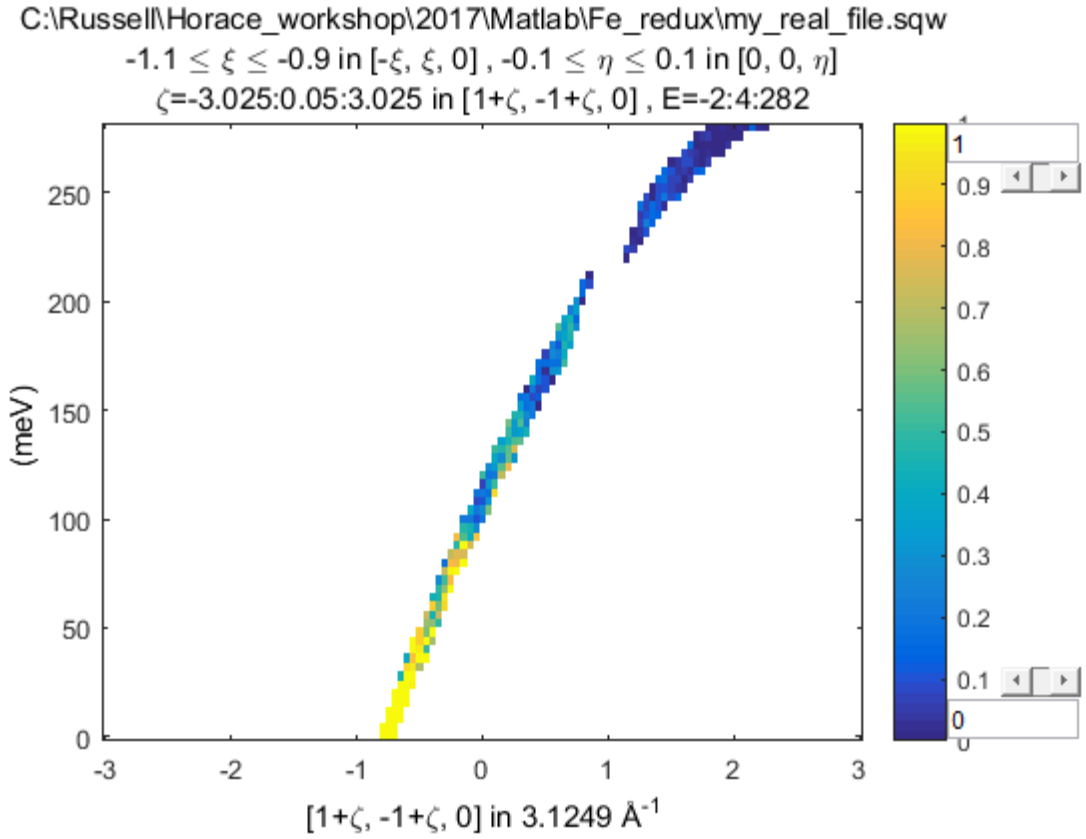
Take a section out of a dataset:

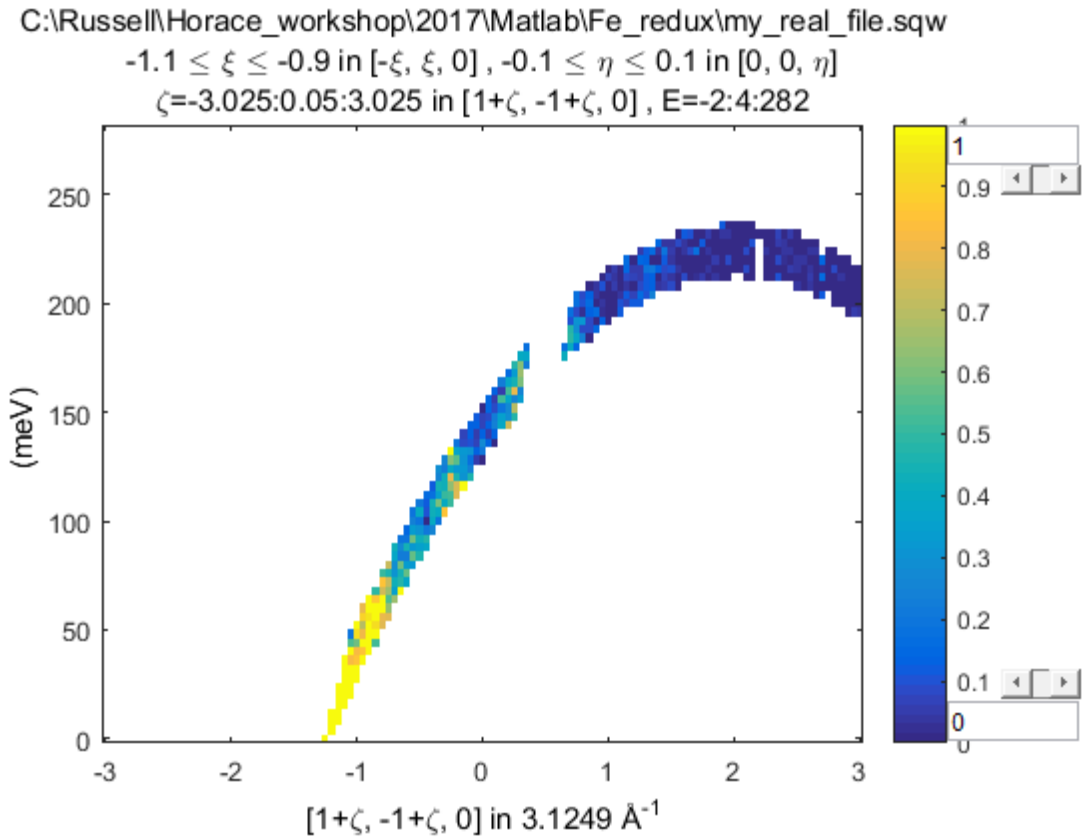
```
w_sec = section (my_slice, [0, 2.5], [100, 250]);%just 0 to 2.5 in Q, 100 to 250 in energy
plot(w_sec);
lz 0 1
```



Split a dataset up into its contributing runs

```
w_split=split(my_slice);%splits into an array of objects (recall indexing of arrays in Matlab)
%each element of the array corresponds to the data from a single
%contributing spe file
plot(w_split(1)); keep_figure; lz 0 1
plot(w_split(10));
lz 0 1
%etc.
%Allows you to determine if a spurious or strange signal is coming from a
%single run, or if it is from a collection of runs.
```





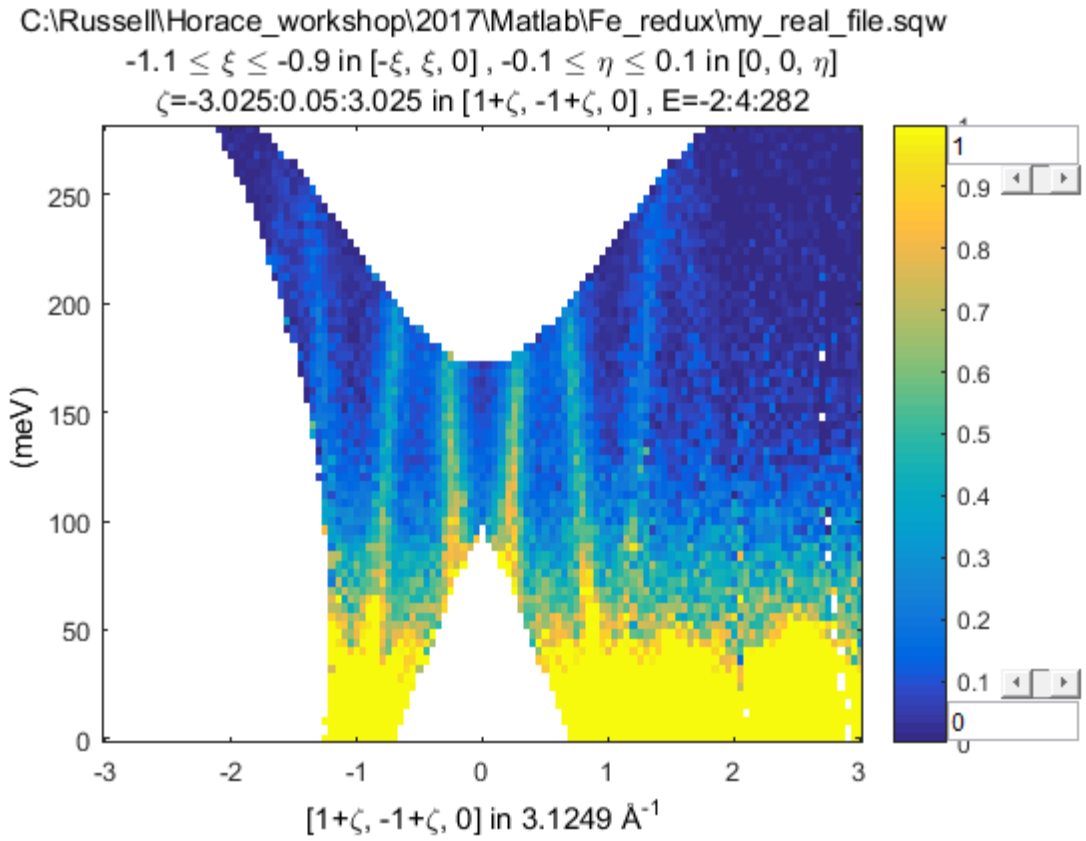
Mask parts of a dataset out, e.g. if there is a region with a spurion that you wish to remove before proceeding to fitting the data

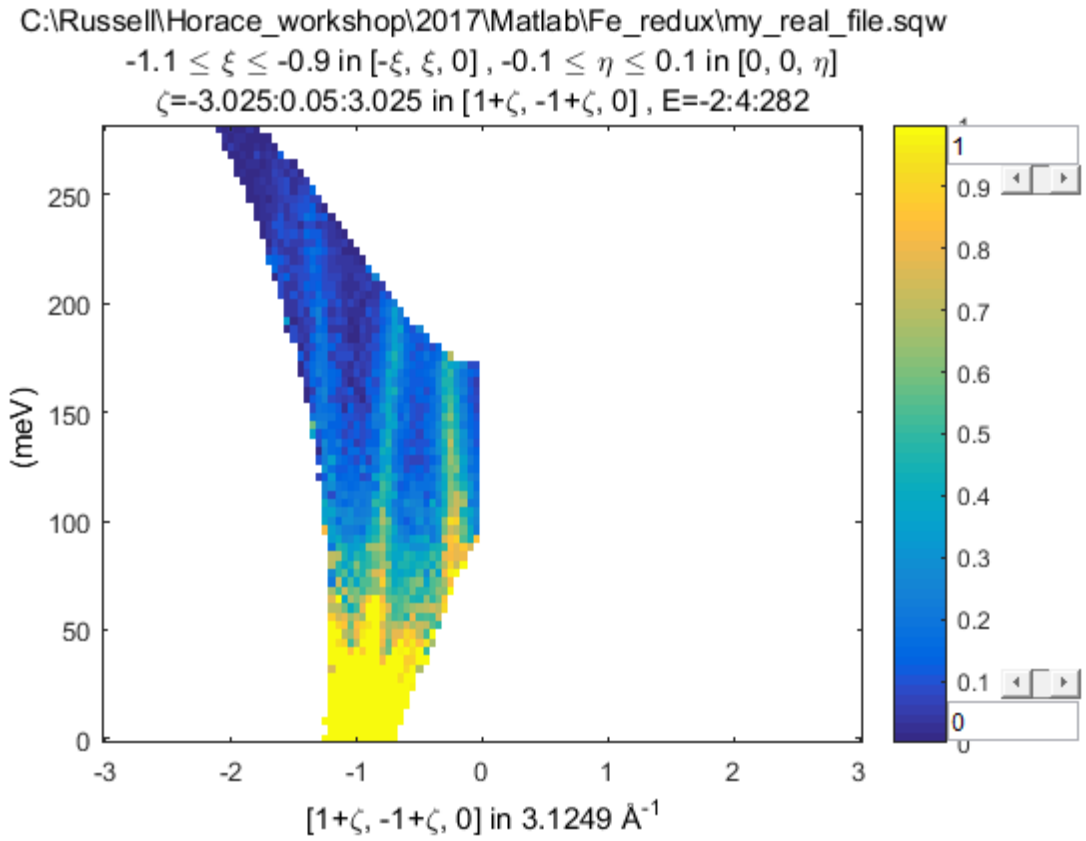
```
mask_arr=ones(size(my_slice.data.npix));%keeps everything
mask_arr2=mask_arr;
```

```
mask_arr2(61:121,:)=0;

my_slice_masked1=mask(my_slice,mask_arr);%should do nothing
my_slice_masked2=mask(my_slice,mask_arr2);

plot(my_slice_masked1); keep_figure; lz 0 1
plot(my_slice_masked2); keep_figure; lz 0 1
```





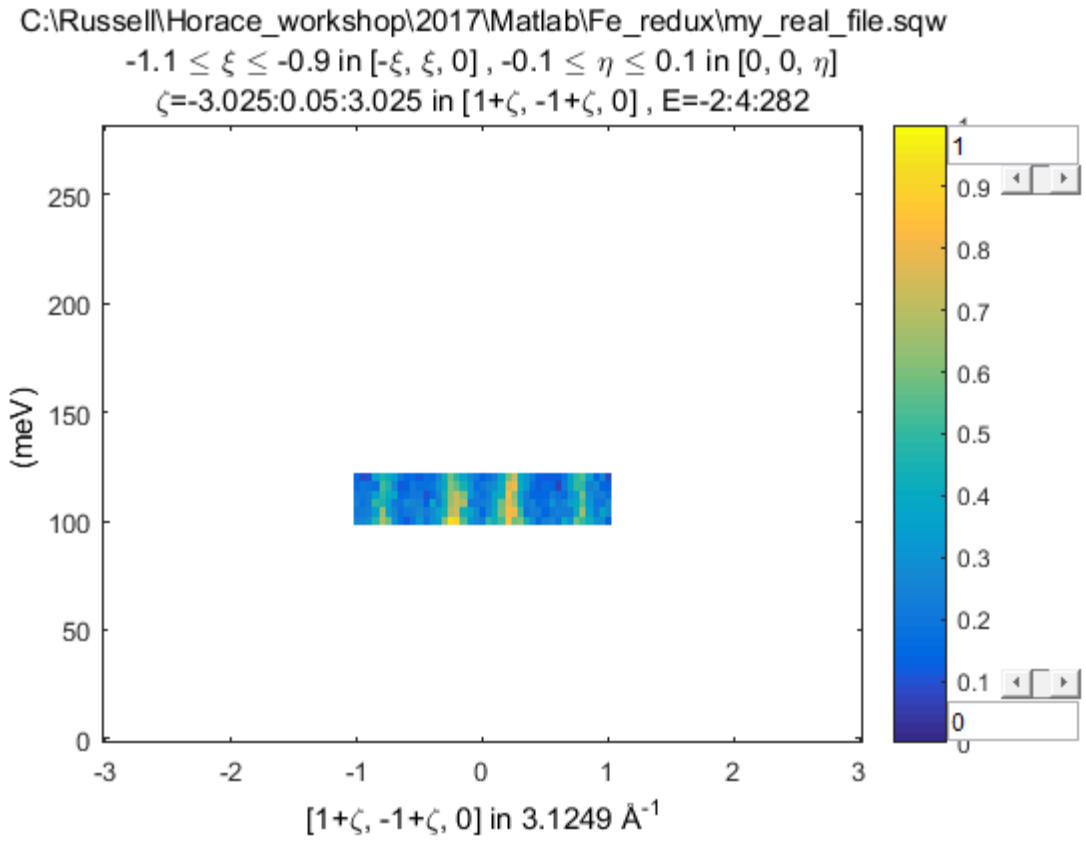
Mask out specific points, if the mask you need for the above is more complex:

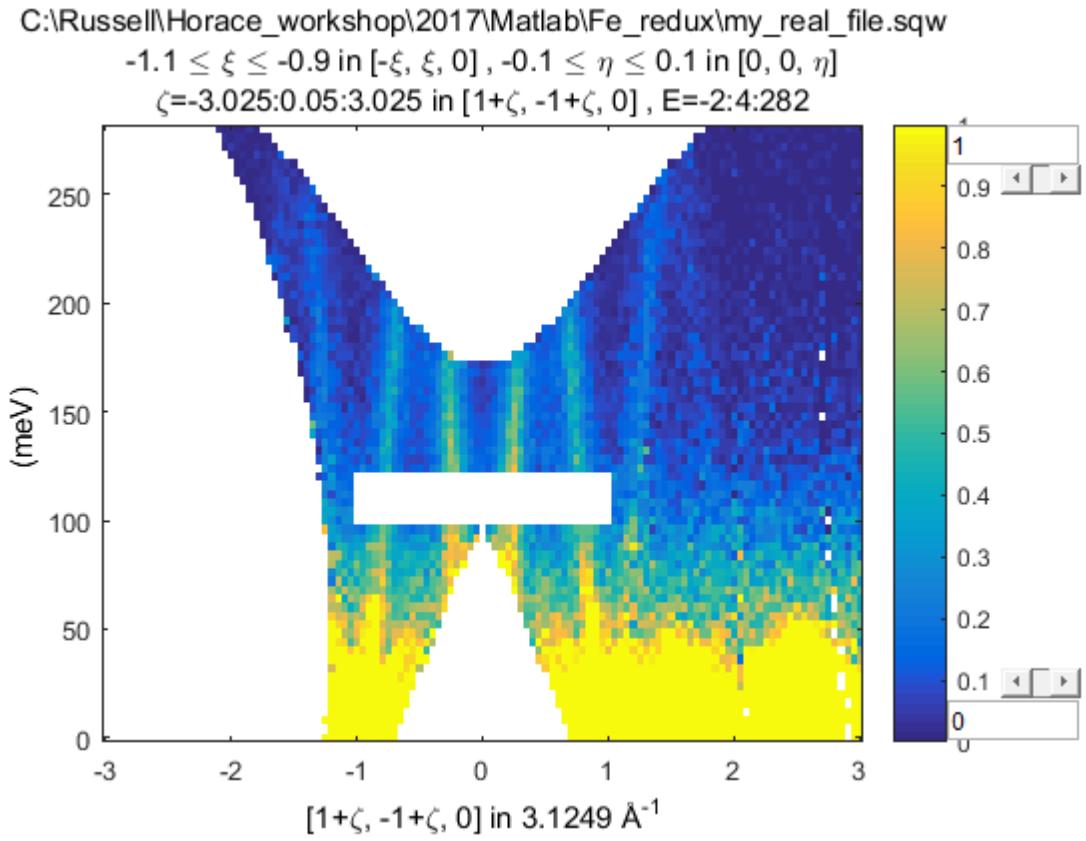
```
sel1 = mask_points (my_slice, 'keep', [-1,1,100,120]);%specify limits to keep

sel2 = mask_points(my_slice,'remove', [-1,1,100,120]);%specify limits to remove
```

```
my_slice_masked3=mask(my_slice,sel1);
my_slice_masked4=mask(my_slice,sel2);

plot(my_slice_masked3); keep_figure; lz 0 1
plot(my_slice_masked4); keep_figure; lz 0 1
```





Symmetrise entire file (see more detailed help for information about this possibility - only use

```
%if you know what you are doing, and the sample misalignment has been
%accurately characterised and corrected).
```

```
%gen_sqw (spefile, par_file, sqw_file, efix, emode, alatt, angdeg,...
%    u, v, psi, omega, dpsi, gl, gs, 'transform_sqw',@(x)(symm_all_data(x)));
```

Published with MATLAB® R2015b