

Excitations Data Analysis Training Course – Worksheet 2

Horace Introduction

Horace is a program to view and analyse single crystal inelastic neutron scattering data from time-of-flight (ToF) spectrometers. These machines have a large area detector and measure the neutron's ToF in order to determine their velocity (energy). Combined with a way of defining either the neutron's energy before or after it scatters from a sample the ToF information allow the neutron's energy transfer to or from the sample to be determined. The location of the detected neutron on the 2D area detector then allows two of the three components of the momentum transfer vector to be determined.

Horace is based on the observation that depending on the orientation of a single crystal sample with respects to the incident beam, the projection of the 2D detector surface on the 3D momentum transfer space changes. Thus a 4D $S(\mathbf{Q}, \omega)$ dataset can be built up from multiple measurements of a single crystal sample at different orientations. Horace is designed to perform such a recombination and then to quickly access and rebin the data in 3D, 2D, or 1D for further analysis. The data structures which permits this quick access is describe in more detail in the Horace paper [\[1\]](#).

Horace Planner

To visualise the 2D projection of the detector surface, we will use a GUI tool called the Horace Planner. To run it, open up a Matlab session (Applications → Software → Matlab 2020a) and type:

```
horace_on  
horace_planner
```

You should see a GUI with several empty fields. First click “Browse” and navigate to `/home/dl11170/edatc/data` and select the file `4to1_102.par`. This is contains a definition of the detector array of the MAPS spectrometer. Then click “Load”.

Next we have to define the run parameters. We will use the lattice parameters of bcc-Iron, which will also be used for the next part of this practical: (type the contents of the quoted strings *without the quotes*)

```
u – “1 0 0”  
v – “0 1 0”  
psi min – “0”  
psi max – “0”  
lattice pars – “2.87 2.87 2.87”  
lattice angles – “90 90 90”  
Ei – “401”  
eps – “0”
```

Now click “Calculate” – you should see points defining trajectories in the 3D momentum transfer (\mathbf{Q}) space. The top right graph shows the plane defined by the \mathbf{u} and \mathbf{v} vectors which is usually defined to

be in the horizontal plane with **u** parallel to the incident beam direction. (The **psi** angles are measured with respect to **u** with **psi**=0 parallel to **u**). The two bottom graphs show the planes defined perpendicular to the plane of **u** and **v**. In this case we defined **u**=[100] and **v**=[010] so the perpendicular direction is [001].

Now change the **psi max** value to “10” and click “Calculate”. You should see that the single trajectory has changed into a fan. The blue edge of the fan represents the 2D plane of detectors at **psi**=0, and the yellow edge the detector plane at **psi**=10. Thus if you were to measure several scans at orientations between **psi**=0 and **psi**=10 you would cover the **Q** range filled in the graph.

Now change **psi max** to 90 degrees.

With this **psi** range (0-90°), what is the largest accessible **Q** parallel to the [100] direction?

In the above we have used **eps**=0. “**eps**” denotes the energy transfer, and **eps**=0 means we have been simulating for elastic scattering. Revert **psi max** and **psi min** to 0 and change **eps** to “50” and click “Calculate”. You should see a single trajectory again. Change **eps** to “0” and click “Calculate”.

What do you notice about the difference between the trajectories at **eps**=0 and **eps**=50 meV?

Hint: What is the lowest **Q** which is reached by the trajectories?

What happens if you increase **Ei**? What happens if you decrease **Ei**?

The purpose of the Horace Planner is to determine what range of rotation angles to measure your data, but the accessible **Q** range is more restricted at non-zero energy transfer so it is important determine what you want to measure and what sample rotation angle **psi** and/or **Ei** to measure to achieve this.

Let’s try an example. We are interested in scattering around the (200) position in bcc-iron.

What is the maximum energy transfer at which we still have data at this point, given the values of **Ei**=401meV and **psi**=0-90°?

We want to have measurements up to 250 meV at (200), (020) and (0,-2,0).

Find a value of **Ei** and the minimum range of **psi** that will allow us to achieve this.

Often you wish to minimise the range of angles scanned in an experiment, since this is directly proportional to the measurement time.

Horace GUI

We will now look at data measured on bcc-Iron using MAPS. In the main Matlab command window, type:

```
horace
```

You should see a GUI window with three buttons at the top “Data on File”, “Data in Memory” and “Generate SQW File”.

Generating a combined 4D $S(Q,\omega)$ datafile

Click on “Generate SQW File”. Click on the “Browse” button directly beneath this (connected to the “SPE files” label although this may be obscured). Change the “Type of File” to “All files” and select all the files starting with “map” and ending with “_ei400.nxspe”. Click “Open”. You should see the “SPE File” populated.

In the “Psi angles” text box type “0:2:90” and click “Refresh list”. Scroll to the bottom of the list and check that all files have equivalent angles, and that the run numbers are in order (from 15052 to 15097 inclusive).

Now click on “Browse” next to “SQW filename” and pick a location in your home folder (it is recommended to create a working folder using the folder-with-start icon (third from right in the top row)).

Leave the “PAR filename” box empty (we don’t need a par file because the detector information in it is included in the .nxspe files).

For the crystal orientation use “1 0 0” for “[uh,uk,ul]” and “0 1 0” for “[vh,vk,vl]”, and “401” meV for the incident energy.

In the “Geometry” dropdown box, select “Direct”.

Use “2.87 2.87 2.87” for the lattice parameters and “90 90 90” for the lattice angles.

Use “0 0 0 0” for the “Offset angles”.

Click “Generate SQW file” – and wait about 5 min for the file to be generated. You can switch to the main Matlab command window to see the progress of this generation.

Viewing the $S(Q,\omega)$ datafile

Now click the “Data on File” button at the top of the GUI.

Browse to the file you just created.

Set **u** as “1 1 0” and **v** as “-1 1 0” and leave **w** empty (note that these vectors define the *viewing* projection and do not need to be the same as the values of **u** and **v** you used to create the SQW file in the previous part. Those **u** and **v** vector define the physical orientation of the crystal with respects to the incident beam).

Set the units as “rlu” (reciprocal lattice units).

Set the “Axis 1 (u)” binning to “-3 0.05 3”

Set the “Axis 2 (v)” binning to “-3 0.05 3”

Set the “Axis 3 (w)” binning to “-0.1 0.1”

Set the “Energy” binning to “0 4 360”

For the binnings, three digits specify that the relevant axis should be rebinned with those values (min) (step) (max) – e.g. in this case we rebinned the u ([110]) and v ([-110]) axis from -3 to +3 with a step size of 0.05 r.l.u. Two digits specify that the program should integrate over that axis – in this case we integrate between -0.1 and 0.1 in the w ([001]) axis. Since we specified three sets of 3-digit bins, we will get a 3D object (we have just integrated over a single axis out of the 4D dataset).

Type “w1” (or whatever you like) in the “Output object name” and click “Cut”, and wait ~15s.

Now click on the “Data in Memory” button at the top of the GUI.

Click “Refresh List” and then select “w1.....d3d” from the list.

Click “Plot” – a new window will appear with a set of axes.

In the edit box below and to the right of the “Iso surface controller” text, change the value from ~4000 to 1. This is the intensity scale of the cuts which will be produced. Click on one of the side panel bar (they are labelled “E” [right side], “ ξ ” [left side] and “ ζ ” [top]) – you should see a 2D colormap surface showing the counts in a plane cut out of the 3D data. In principle you can drag the green arrow to move this plane around but the latency of the online connection makes this difficult. It might be easier to make a series of cuts (e.g. at E=50meV, 100meV, 150meV, etc) overlapping each other.

Explore the dataset and see if you can pick out the spin wave excitations (iron is ferromagnetic).

The dispersion is clearer if we picked one of the 2D planes.

Click “Data in Memory” and select “w1” again and click “Cut” (instead of “Plot”).

Type “0” for the “ ζ , ζ , 0” axis (u)

Type “-1.1 -0.9” for the “ ξ , ξ , 0” axis (v)

Type “0” for the “Energy axis”

In this case “0” indicates to keep the original binning, and “-1.1 -0.9” means to integrate between $-1.1 < \xi < -0.9$.

What happens if you typed “-3 0.05 3” or some other 3-digit binning in the other axes instead of “0”?

You should see an error message saying “Object is dnd”.

Horace contains two main types of data classes: `dnd` and `sqw`. A `dnd` object contains only the rebinned data – for example the “w1” object above is a 3D `dnd` (“d3d”) object and has a 3D array of data. An `sqw` object has this rebinned data (the 3D array) in addition to a list of all “pixels” (detector-energy bins) from the original datafiles which contribute to each rebinned points.

In order to avoid re-binning (rebinning data which has already been rebinned), Horace only supports integration and not rebinning for `dnd` objects – hence the error message.

Click on the “Data on File” button again, and this time check the “Retain all pixel information” radio button. Then click “Cut”. Click the “Data in Memory” button and click “Refresh List” – verify that “w1” is now a “sqw” rather than a “d3d”.

Click “Cut” and verify that you can use “3 -0.05 3” (or another 3 digit binning specifier) for the $zeta$ (u) axis.

Background subtraction

Make a d2d cut called “d2” rebinned “3 -0.05 3” along the zeta (**u**) direction, integrating over “-1.1 -0.9” in the xi (**v**) direction and “-0.1 0.1” in the eta (**w**) direction, and rebinned “0 4 360” along energy. Plot the cut.

What do you notice about the scattering at low $|Q|$ (around 0) compared to high $|Q|$?

Hint: Remember the magnetic form factor

Now made a d1d cut called “d1” integrating over “2.8 3.0” in zeta (**u**), “-1.1 -0.9” in xi (**v**) and “-0.1 0.1” in eta (**w**), and rebinned “0 4 360” along energy.

Select “d1” from the “Data object” dropdown box (you may have to click “Refresh List”), and then click “Replicate”. Select “d2” as the “Reference Object” and type “r2” as the “Output object name”, and click “Replicate”.

What this does is that it tiles the 1D energy cut “d1” along the zeta axis to create a new 2D cut “r2” which has the same bins as the reference object “d2”.

Now select “d2” from the “Data object” dropdown box, and click “Binary ops”. Select “r2” as the “Operate #2 name” and “minus” as the “Function”. Give the output name as “w3”, and click “Operate”.

Now plot “w3” and set the color scale from 0 to 1.

Is this a good background subtraction? How can you improve it?

The GUI is a little limiting, but we will use a similar approach with the command line on Thursday which will be a bit more flexible.

Fitting the dispersion

Finally, a traditional way of fitting spin wave dispersion is actually fit the peak positions to obtain a dispersion relation $\omega(\mathbf{Q})$ and to then fit these to a model. Since bcc-Fe is a ferromagnet, we can use the textbook dispersion relation [2]:

$$E(q_h, q_k, q_l) = 4JS(1 - \cos(\pi q_h) \cos(\pi q_k) \cos(\pi q_l))$$

Make 1D cut along energy at some \mathbf{Q} point – e.g. integrating over “0.7 0.75” in zeta, “-1.1 -0.9” in xi and “-0.1 0.1” in eta and plot it.

Can you see a clear magnon peak? What happens if you decrease the integration range?

The very steep dispersion unfortunately means that performing a constant-Q energy cut actually sums over a large dispersion so the peaks appear very broad. For fitting to the above equation it is better to perform a set of constant *energy* cuts instead.

Make a 1D cut along *eta* (**u**) with binning “0.5 0.02 1” integrating over “-1.1 -0.9” in xi (**v**) and “-0.1 0.1” in eta (**w**) and “45 55” in energy. Plot the cut – can you see a magnon peak?

Select the cut you just made and click “1d peak fitting” at the bottom of the GUI. Select “Gaussians” from the “Peak Functions” dropdown menu and “Constant background” from the “Background Functions” dropdown. Then put “0.5” for “Amplitude(s)”, “0.9” for “Centre(s)” and “0.05” for “Width(s)” (note that if you want to fit multiple peaks enter more than one numbers for each of these fields). Then put “0.7” for “Background” and click “Simulate”.

Now click “Refresh list” and you should see another workspace with “SIM” in it appear. Plot the original and the simulation and check they are ok. Select the original data `d1d` object and click “Fit”. You should see a “FIT” workspace appear (after clicking “Refresh List”), and the fit parameters printed in the message window (if this clears, you can also see the fit parameters in the main Matlab command window).

Make cuts at other energy transfers and fit the location of the magnon peak in them.

Plot the mean energy of the cuts vs $(1-\cos(\pi h) \cdot \cos(\pi k) \cdot \cos(\pi l))$ and determine the value of J (take $S=2$).

References

[1] R.A. Ewings, A. Buts, M.D. Lee, J. van Duijn, I. Bustinduy, T.G. Perring, *Horace: Software for the analysis of data from single crystal spectroscopy experiments at time-of-flight neutron instruments*, [Nucl. Instr. Methods Phys. Res. A, **834** \(2016\) 132](#).

[2] C. Kittel, *Introduction to Solid State Physics*, Wiley and Sons, 2005, Chapter 12, eq 22, extended to 3D.