# Excitations Data Analysis Training Course – Worksheet 5

## 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 $S(\boldsymbol{Q},\omega)$ 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].

## Fitting a single 1d cut with an S(Q,w) model (including coding your own model)

We will now fit part of the same 1d cut we used before with `multifit_func`, but with `multifit_sqw`, which is for the case where you have a model for S(Q,w). The function syntax (parameters, free/fixed, binding) is exactly the same. Make a new 1d cut that is the same as the previous one, with energy integration range 100 to 120 meV, but with a Q range of 0.5 to 1.5 in steps of 0.05.

1. Try fitting your new cut on a linear background using `multifit_sqw` to the model for S(Q,w) in `sr122_xsec`. Good starting parameters are the ones we used at the beginning of this session when simulating the Q-E slice, namely `[1, 0, 0, 35, -5, 15, 10, 0.1]`. Remember that parameters 2 and 3 are redundant, so fix these in the fit. Use a linear background model `linear_bg` (the same one you were using before) with appropriate background constant and gradient as starting parameters.

2. You will remember from when we simulated the Q-E slice that the model we have been using so far will not be all that good for globally fitting our data! Using the file `sr122_xsec.m` as a starting template, write your own Matlab file that codes the following model:

$$E_0 = \Delta + 8J(1 - \cos(\pi h)\cos(\pi k)\cos(\pi l))$$

$$B = \frac{E}{1 - \exp(-\frac{1.602E}{T})}$$

$$S(\mathbf{Q}, E) = AB\frac{4\Gamma E_0/\pi}{(E^2 - E_0^2)^2 + 4(\Gamma E)^2}$$

3. Where *A* is an amplitude scale factor, *T* is the temperature in Kelvin, *J* is a ferromagnetic exchange constant in meV, Γ is an energy broadening term in meV, and Δ is an energy gap in meV .
4. Run a simulation of our favourite Q-E slice using this new function, and play around with the input parameters to get something that looks similar to the data. Note that the temperature should be fixed at 10 K.
5. Now try fitting the 1d cut, allowing for a linear background model as well. Play with fixing different parameters in the fit. What sensitivity does the data have to the different parameters, and how might pairs of parameters be coupled in a fit?

## Fitting multiple cuts simultaneously with a single S(Q,w) model

One of the key features of Horace is the ability to fit many objects to a single model and single set of parameters, but with independent backgrounds. This can be done to an arbitrary collection of 1d, 2d, 3d and 4d objects, not just 1d cuts, although that is all we will do here.

1. Use the array of 1d cuts created at the start of this exercise that all cover the same Q-range for a series of energies.
2. Run `multifit_sqw` but this time initialise with the array of cuts. Make a series of plots of the results (i.e. plot the cut and add a line for its corresponding fit, and repeat for every cut). Also look at the structure array that contains the fit data, to understand how the information is now stored there.
3. For our final fitting exercise we are going to use different background functions for different cuts. To do this you need to create a cell array whose number of elements is equal to the number of cuts. Each element should be the handle to the background function you are going to use – use the `linear_bg` function (Herbert built-in) for the first three cuts, and `quad_bg` for the last two. The input parameters for the background functions similarly have to be contained in a cell array, each element of which is a vector of input parameters; likewise the free parameter list.
4. As an exercise in using bindings with multiple datasets, set the bindings so that the gradients are all the same for the linear backgrounds, and the x2 terms are all the same for the quadratic backgrounds.
5. Once the fit above has run and converged, plot the results of the cuts and fits. Also simulate the Q-E slice with the fit parameters you found, as a sanity check.

## Fitting with Resolution Convolution

Often the resolution function of the instrument can make a significant quantitative difference to the result of a fit. In this session we are going to repeat the simultaneous fit to multiple cuts but this time include the effects of resolution. The program which does this is the eponymous tobyfit, which has a very similar syntax to multifit and its variants. Before yu can use it, additional information needs to be added to the cuts that describes the configuration of the instrument – moderator and chopper pulse shapes, the size of the sample, information about the beam divergence and the size of the detectors. Use the on-line documentation for Tobyfit as a guide in what follows.

1. First create a sample description. Create an object of type IX_sample. The sample can be approximated as a cuboid 3cm x 3cm x 4cm high, with the sides parallel to a*, b* and c* respectively.

2. Create an instrument object using the function maps_instrument. This function contains much of the fixed instrument description, and just needs a few parameters specific to the particular experiment. For reference, there are equivalent instrument generation functions for maps, mari and let. You already know that the incident energy was 401 meV. The monochromating chopper was the 'Sloppy' chopper running at 600Hz. Have a browse through the output of this function.

3. Retaining your original cuts, set the sample and instrument descriptions using set_sample and set_instrument. Have a look at the header section of one of the cuts. Can you see where the information is stored?

4. Now create a fitting object using Tobyfit. This works just like multifit – type doc sqw/tobyfit to open a help window. If you look closely at the list of methods you'll see that there are a few extra ones that are specific to Tobyfit. We'll one or two of these later on. Set the cross-section model to the one you wrote for use with multifit_sqw, and set the starting parameter values (and which parameters are free) to the same ones you used in the previous exercise. To start with, just initialise Tobyfit with one of the cuts in your array, and set a linear background. Fit the parameters and compare the results with those you get using multifit_sqw.

5. Do the same, but now for all five datasets simultaneously. Follow a similar procedure as you did when you fitted all five cuts simultaneously with multifit_sqw earlier on, namely a single function for the S(Q,w) model but independent linear backgrounds for each cut.

6. One of the control parameters you have in Tobyfit is the number of Monte Carlo points per detector-energy pixel. Use the documentation doc sqw/tobyfit and navigate to the list of methods to work out how to set the number of Monte Carlo points. The default is 10. Why does the resolution convolution algorithm give useful results with such a small number?

7. There are various contributions to the resolution function of the instrument, and you can disable one or more of these contributions. Experiment with the effect of turning off the moderator contribution, the chopper contribution, and others too. Just to keep the fitting time reasonable, return to fitting just the second cut of the array. What are the major contributors to the resolution of the instrument?

8. As a final exercise, set the foreground function to be local and fit the data with the exchange constant to be constrained to be the same for all cuts, but the intensity and lifetimes to vary. Why would you want to do this? At this point you will have become an expert!

## A worked example using $CuGeO_3$ (Optional)

We've now gone through a full example using the bcc-iron dataset from generating the 4D S(***Q***,ω) file, making cuts and plotting them to simulating a model and including resolution convolution. As an optional exercise, to confirm what you've learnt, you also apply this analysis to $CuGeO_3$, a 1D spin-1/2 chain material with spinon excitations. The theory is described in an early work by Nagler et al. [2] and the first experimental inelastic neutron work is by Arai et al. [3]. The theoretical S(q, ω) model (the Muller Ansatz) is coded in a file /home/dl11170/edatc/scripts/Bethe_Ansatz_CuGeO3.m but you could equally write it from eqs (3), (4) and (7) of ref [2].

Because the system is 1D, the (magnetic) dispersion is only along a single Q direction (00L) so as long as this direction is perpendicular to the incident beam, it will be imaged in the detectors. As such the data was taken at a single rotation angle. The data file is:
`/home/dl11170/edatc/data/MER18519_Ei50.00meV_One2One.nxspe`

It was measured with the *b* axis parallel to the incident beam ($k_i$ || [010]) and with the *c* axis [001] horizontal and perpendicular to the incident beam. The lattice parameters are *a*=4.81Å, *b*=8.47 Å, *c*=2.94Å, and α=β=γ=90°. You can assume all the goniometer angles are zero.

The measurement (from an earlier in-person neutron training course) used the MERLIN spectrometer with the gadolinium ('G') chopper running at 250 Hz with Ei=50 meV.

1. Create an sqw file from the nxspe data file using the information given above.
2. Make a Q-E cut along the dispersion direction [00L] similar to that shown in Fig. 1 of [3].
3. Evaluate the theoretical (Muller Ansatz / `Bethe_Ansatz_CuGeO3.m`) model on your cut. The theoretical model function takes two parameters: [scale_factor, JS]. Note that the exchange parameter used in the model function is J*S which is actually half of the J value quoted in ref [3] because S=1/2.
4. Make some 1D cuts to compare the data and the model calculations – which part of the spectrum should you use to estimate the background?
5. Subtract the background and fit the data to the model. Plot the results – what do you notice about the widths of the data and model?
6. Simulate some spectra including resolution effect and compare that to the data.
7. (Optional) Try to fit the data to the model including instrument resolution effects

## 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] S. E. Nagler, D. A. Tennant, R. A. Cowley, T. G. Perring, and S. K. Satija, *Spin dynamics in the quantum antiferromagnetic chain compound $KCuF_3$*, Phys. Rev. B **44**, 12361 (1991).

[3] M. Arai, M. Fujita, M. Motokawa, J. Akimitsu, and S. M. Bennington, *Quantum Spin Excitations in the Spin-Peierls System $CuGeO_3$*, Phys. Rev. Lett. **77** 3649 (1996).