VisionElasticFit

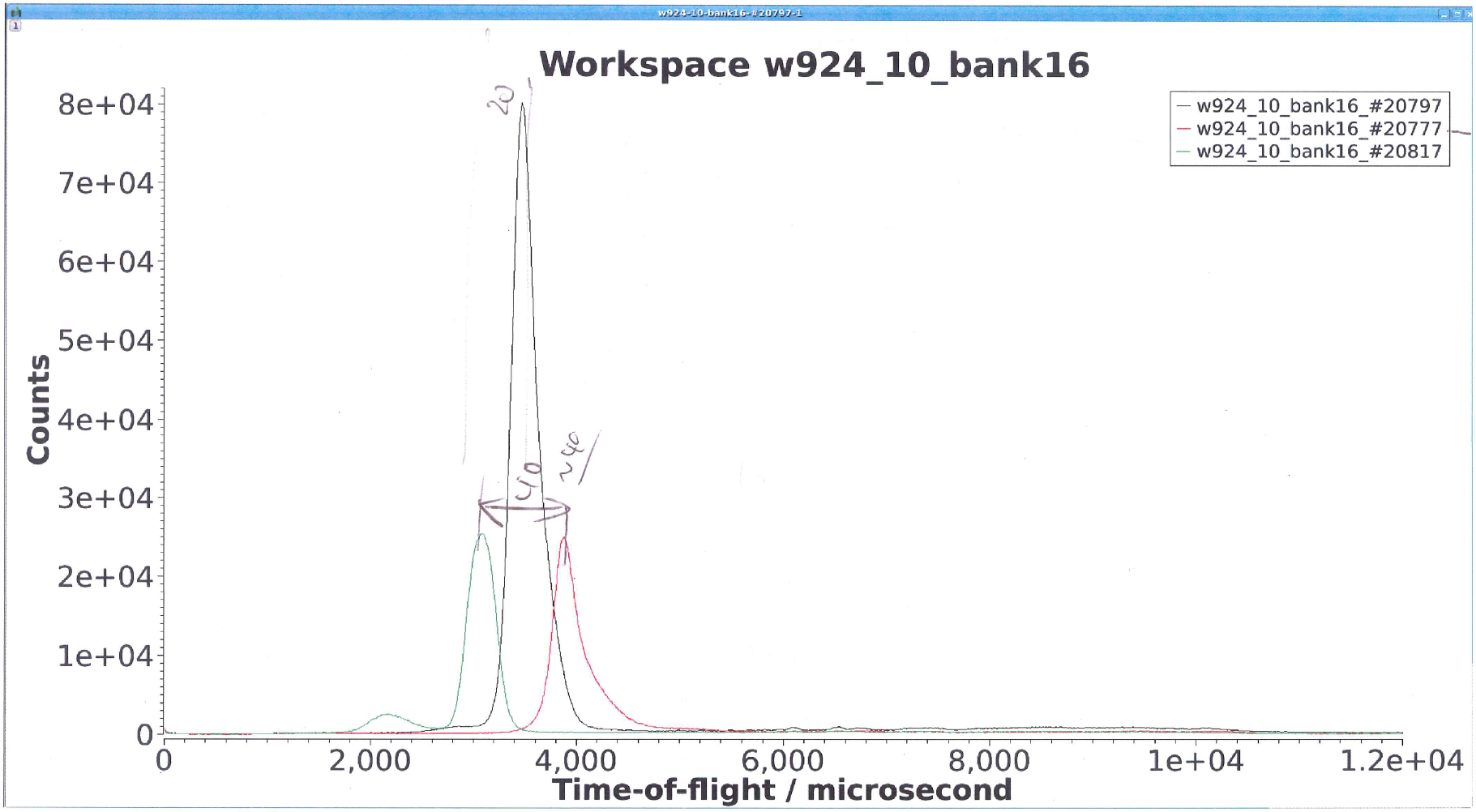
**Summary before Meeting Uli:**

* 7 modules for inelastic scattering at 35o, 7 modules for inelastic scattering at 135o, 6 modules for equatorial diffraction detectors, and 10 modules for backscattering diffraction detectors.
* Each inelastic detector: 8 tubes/detector x 32pixels/tube = 256 pixels/detector

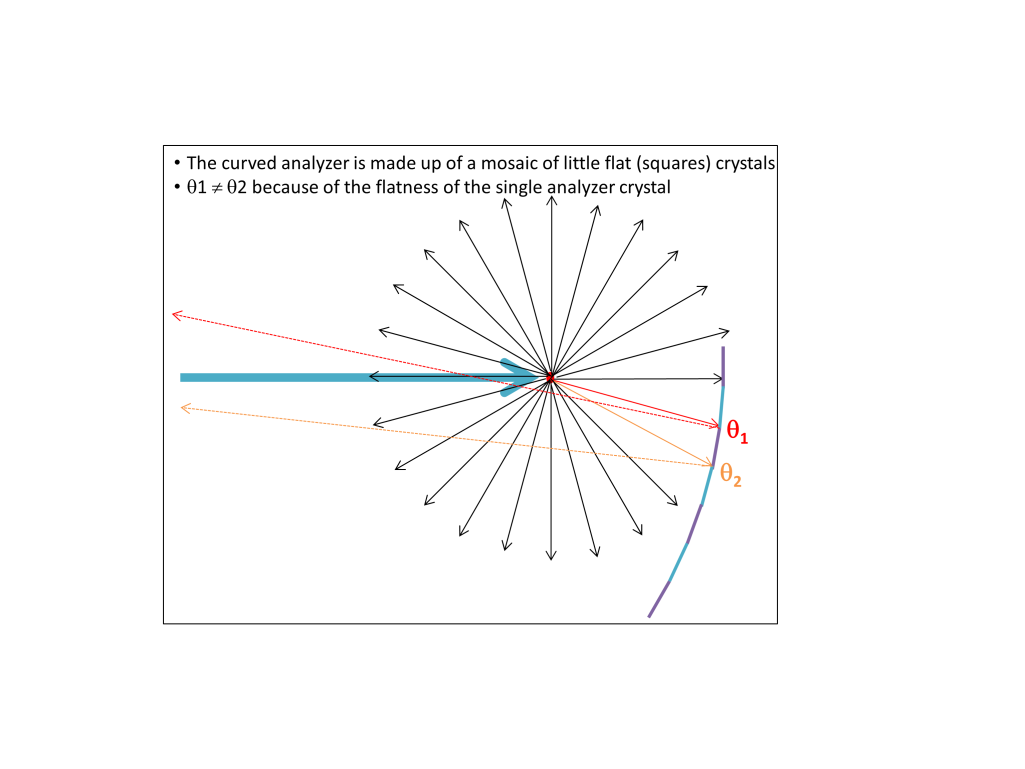
August 14 2012

**Meeting Uli:**

* The problem: for every pixel in a tube, the elastic line peaks at a slightly different time-of-flight (*tof*). See below picture. The sample is 15mgr of caffeine with a 5mm x 5mm geometry. Compared with the scale of the flight path, the sample can be considered as point in space. The picture was generated with data from proposal **IPTS 6966, run 924, bank 4**.



* The picture shows the same elastic line recorded by three different pixels in a particular tube. The middle, strongest peak corresponds to the pixel in the center of the tube. The left peak corresponds to the pixel located 20 pixels “down” the central pixel, and the right peak corresponds to the pixel located 20 pixels “up” the central pixel. As we see, the three peaks do not superimpose for at the same *tof*.
* The lack of superimposition poses a problem. If we simply add the intensities of all the pixels in the tube, we’ll end up with a smeared elastic line, and it will be difficult to extricate the inelastic signal from the total spectrum.
* The origin of the shift in the peak center with pixel ID is mainly due of a geometrical effect of the disposition of the analyzer crystals. Here’s a cartoon of the scattering of a point sample that scatters equally in all directions:



The cartoon shows that the finite dimensions of the single flat analyzer crystal produce a distribution of Bragg angles, θ1 to θ2. The particular [θ1, θ2] range is different for every crystal, hence the slight difference in *tof* for each crystal.

* Uli decided to fit the peak found in the spectrum of each analyzer crystal with the sum of two Gaussians and a background:



However the heights *h* and *h’* are constrained to obey a quadratic equation across pixel ID *i* in a single tube (remember there are 128 pixels, and a fraction of them are masked):

, and 

note that we do not necessarily require *c=m* and *c’=m’*.

Pixel *i=0* here corresponds to the pixel in the tube receiving the maximum neutron count, located around the center of the tube.

* Fitting becomes a 2-dimensional fitting in time and pixel ID. Thus, let *υ* be the real coordinate across pixel ID. The intensity of the whole tube can be modeled now as:



We have six free parameters per pixel and 12 parameters for the whole tube.

We can substitute the delta Dirac functions for peaked Gaussians. Moreover, we can neglect the  prefactor in the Gaussians since we are effectively integrating the intensity along the *υ*-coordinate when we evaluate it at the pixel ID’s. The model becomes:



, and 

Since we do not require *c=m* and *c’=m’*, we can do two independent fittings, one for and one for .

* A sensible strategy is to find initial values for the parameters is:
* Fit first the spectrum of each pixel setting *h* and *h’* as free parameters, obtaining 
* Fit  obtaining . Do analogously to find , , and .

August 16 2012

**Fitting strategy for Mantid:**

* We fit a whole ‘pack’ made up of eight tubes. Each tube may have its own *FunctionDomain1D* because the spectrum of each tube may have different values for the *tof* bins. If we do rebinning then the same *FunctionDomain1D* could be used by all tubes (*t*-coordinate). However, the current implementation of *MultiDomainFunction* requires different domains, not a common one.
* We will create a Mantid algorithm with a *MultiDomainFunction* as its core engine for fitting. the CompositeFunction will be made up of as many functions as spectra, with the unit function given by the expression:



, and 

Parameter *i* is an *attribute* of the function, and denotes the spectrum index.

We will impose ties for the parameters for *hi and h’i* among all the unit functions

* Initial values for the parameters will be obtained in the previously describe fashion, that is:
* Fit each spectrum of each pixel with *h* and *h’* as free parameters, obtaining 
* Fit  obtaining. Do analogously to find,, and .