Web-based flexible data reduction for neutron reflectometry

Brendan Rowan (Poolesville High School)
Mentor: Dr. Brian Maranville (NIST Center for Neutron Research)

Background and Importance

Neutron reflectometry refers to the act of firing a beam of neutrons at a flat surface while measuring the intensity of the reflected neutrons. The data collected when using a neutron reflectometer gives the user valuable information about the structure of any thin films on the surface being measured. However, it is difficult to confidently state conclusions about the structure of these films with only raw data, as they are recorded in instrument-specific coordinates. Converting instrument-specific data into a more usable form is commonly known as data reduction, and is the main basis for the problem at hand.

The proposed program, Dataflow, allows the user to make unique reduction recipes, regardless of which instrument is used. By only requiring the user to learn one interface, as opposed to the current standard of one program per instrument and experiment type, Dataflow makes it easy for the user to learn new reduction routines for many different types of instruments and experiments. Additionally, because the program resides in the user's web browser, a number of advantages are present such as platform and browser independence, version unity, and ease of access.

Caching Intermediate Results

As many data reductions take a large amount of time to compute, it makes sense to cache intermediate results. Therefore, when a user clicks on a wire to view a plot of the data at that point in the reduction, the program can short-circuit and retrieve the desired result if it has already been calculated.

The program that was used to stash results was Redis, a key-value datastore that has mechanisms for storing integers, strings, lists, and maps. Because the serialization modules for Python, the programming language used for the backend, are worthless in regards to most user-defined classes, methods were written that converted Dataflow-specific datatypes to and from strings.

```
getting result
retrieving cached value: Fingerprint:9935f99ed9d065d05d1cab925ef77b93b72ff295:output
Starting new converter
```

Dataflow must then be able to decide whether a certain calculation has been performed or not. To make this decision, a fingerprint can be created for a certain terminal of a certain node in the reduction diagram. However, certain factors have to be accounted for which include:

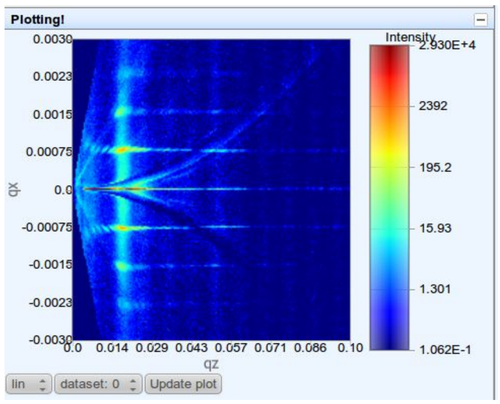
- Fingerprints of ancestors
- All arguments passed to the module
- The output terminal and node number

With this information, a SHA-1 hash, which creates a 40 character string of hex digits, can be created at each step in the reduction for maximum efficiency.

Example Output

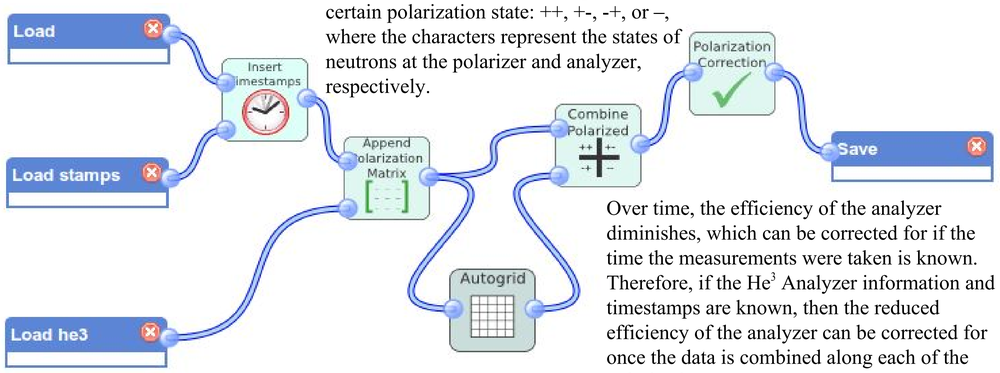
On the right, you can see a very small portion of raw data taken from AND/R at NIST. Below is a screen shot from Dataflow, which shows the plot of the final step in most offspecular reduction routines: converting to Q-space. The data on the right, along with ten other data files was used to produce the plot below.

```
'sabc2002.cg1' 'Mar 11 2008 18:44' 'I'          60.    1  'TIME'  201  'RAW'
  Filename          Date                Scan      Mon    Prf  Base   #pts  Type
sabc2p self-assembled block copolymer on Si gratin
 460    1  500  512 608 512   0   4.9926     0.00000 0.00000   0.00000    1
 Collimation       Mosaic     Wavelength   T-Start   Incr.   H-field #Det
  1        0.80000    0.00000    0.80000
  2        0.80000    0.00000    0.80000
  3       -4.00000    0.01000   -2.00000
  4        3.50000    0.00000    3.50000
  5       25.60000    0.00000   25.60000
  6        0.49940    0.00000    0.49940
Mot:      Start          Step          End
     A3        MIN     MONITOR      COUNTS
     -4.000    1.00      115594        1498
```



Polarized Reduction Template Offspecular Neutron Reflectometry

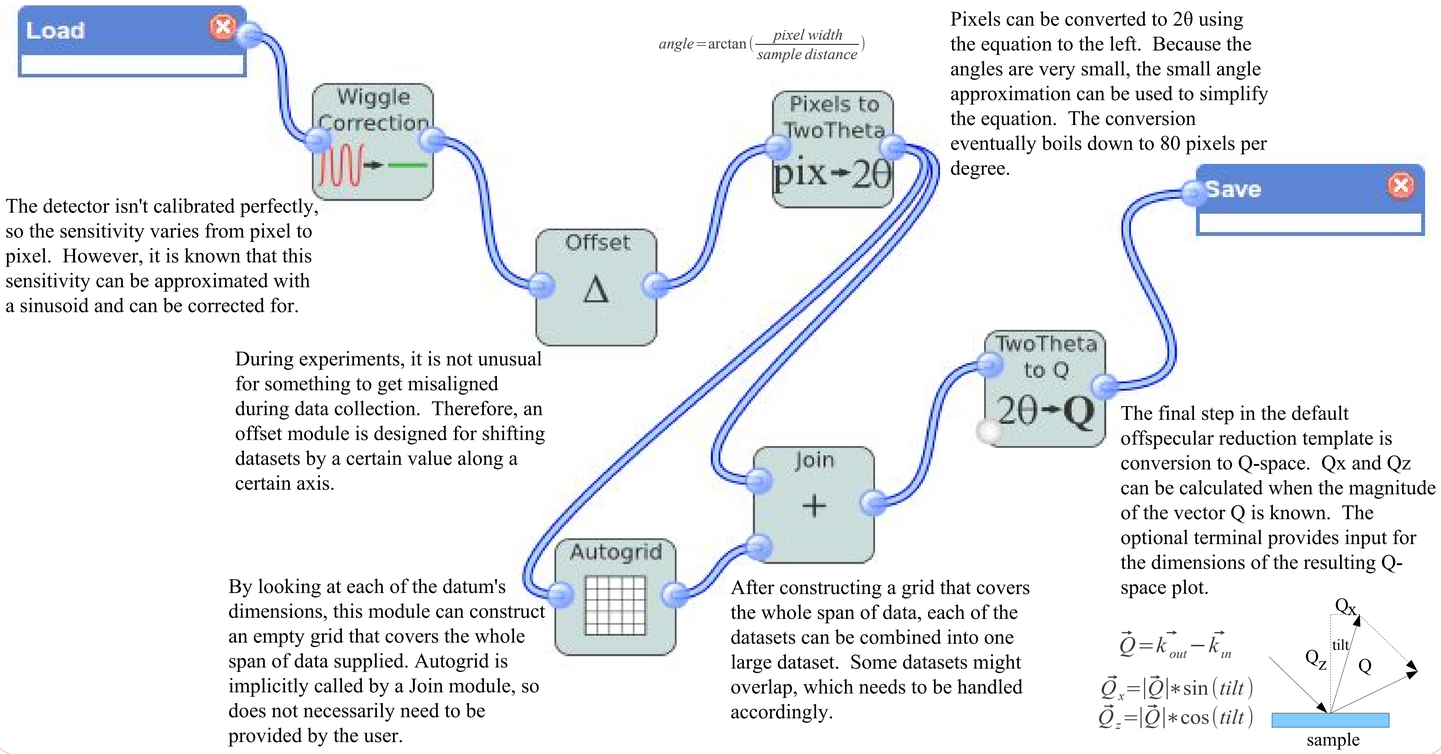
Polarized offspecular neutron data is different from regular offspecular neutron data in that a certain file corresponds to a certain polarization state: ++, +-, -+, or –, where the characters represent the states of neutrons at the polarizer and analyzer, respectively.



Over time, the efficiency of the analyzer diminishes, which can be corrected for if the time the measurements were taken is known. Therefore, if the He^3 Analyzer information and timestamps are known, then the reduced efficiency of the analyzer can be corrected for once the data is combined along each of the polarization states.

Above and to the right are the default templates for polarized and non-polarized offspecular neutron data. As shown, these “templates” are flow diagrams that the user can create at will which link together “modules” that perform a certain reduction. The user is allowed to drag and drop individual modules onto the editor and link them together to perform reductions, although these default templates are provided. Then, the user specifies which data files are needed for certain loaders. Finally, the user can click on a wire to view his or her data at that point in the reduction.

General Reduction Template Offspecular Neutron Reflectometry



$$angle = \arctan\left(\frac{pixel\ width}{sample\ distance}\right)$$

Pixels can be converted to 2θ using the equation to the left. Because the angles are very small, the small angle approximation can be used to simplify the equation. The conversion eventually boils down to 80 pixels per degree.

The detector isn't calibrated perfectly, so the sensitivity varies from pixel to pixel. However, it is known that this sensitivity can be approximated with a sinusoid and can be corrected for.

During experiments, it is not unusual for something to get misaligned during data collection. Therefore, an offset module is designed for shifting datasets by a certain value along a certain axis.

By looking at each of the datum's dimensions, this module can construct an empty grid that covers the whole span of data supplied. Autogrid is implicitly called by a Join module, so does not necessarily need to be provided by the user.

After constructing a grid that covers the whole span of data, each of the datasets can be combined into one large dataset. Some datasets might overlap, which needs to be handled accordingly.

The final step in the default offspecular reduction template is conversion to Q-space. Qx and Qz can be calculated when the magnitude of the vector Q is known. The optional terminal provides input for the dimensions of the resulting Q-space plot.

$$\vec{Q} = \vec{k}_{out} - \vec{k}_{in}$$

$$\vec{Q}_x = |\vec{Q}| * \sin(tilt)$$

$$\vec{Q}_z = |\vec{Q}| * \cos(tilt)$$