- **Resolution Function in 2D**

27 MAY 2010 SRK
revised FEB 2012 SRK

This help file contains a description of the 2D resolution function, and also a discussion of the proper treatment of per-pixel counting statistics, with special consideration to the low count values often encountered in SANS, where Poisson statistics do not apply. See the section:

Propagation of Uncertainties in 2D

Effect of Gravity on 2D Resolution Smearing

The effect of gravity is described in full detail in :
"The effect of gravity on the resolution of small-angle neutron diffraction peaks"
D.F.R Mildner, J.G. Barker & S.R. Kline *J. Appl. Cryst.* (2011). **44**, 1127-1129.
[doi:10.1107/S0021889811033322]

See also Boothroyd, *J. Appl. Cryst*. **22** (1989) 252.

The effect of gravity on a neutron is the usual downward (y) fall with a parabolic trajectory, leading to a downward shift of the detected neutron from the “optical axis” (= no gravity). If the neutrons are monochromatic then every neutron, transmitted or scattered, falls the same distance due to gravity. The effect is that the entire scattering pattern shifts down by the same amount, including the beam center. If the measured (fallen) beam center is used, then all is fine, and gravity has no effect on the resolution of the instrument. Use of the optical center is incorrect and will affect the apparent resolution. If, however, there is a distribution of wavelengths, then the neutrons will fall differing distances, ygr, and a mean beam center will be used rather than the correct beam center for each incident wavelength. The distance a neutron of mean wavelength, λo, falls with gravity is:

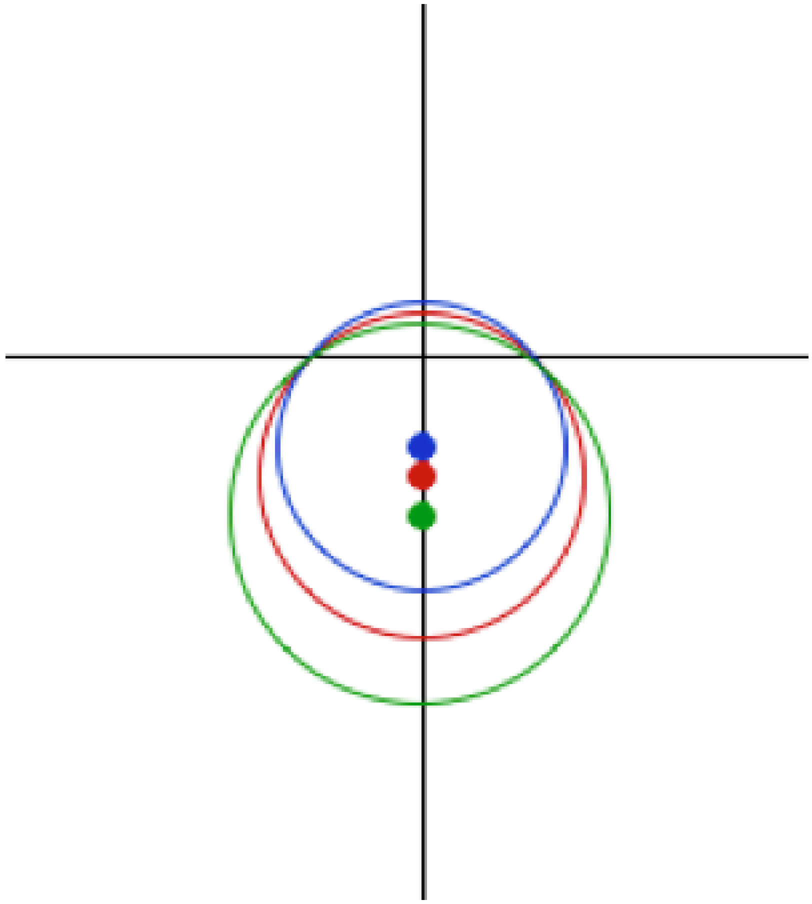
$$y_{gr}(\lambda_o) = -\frac{1}{2v_n^2} g L_2 (L_1 + L_2)$$

or

$$y_{gr}(\lambda_o) = -A\lambda^2 = -\frac{1}{2} g L_2 (L_1 + L_2) \left(\frac{m}{h}\right)^2 \lambda^2$$

with

$$g = 981\ cm/s^2$$
$$m = 1.6749e-27\ kg$$
$$h = 6.626e-34\ m^2 kg/s$$
$$(m/h) = 252.77\ s \cdot cm^{-2}$$
$$A = -\frac{1}{2} g L_2 (L_1 + L_2) \left(\frac{m}{h}\right)^2$$



The circles represent the scattering at equal q-values for three different wavelengths.
Red = mean wavelength
Blue = shorter
Green = longer

The q-value of the scattering is identical for each wavelength as dictated by the sample. The scattering angle and the location of detection, however, depend on the incident wavelength. The beam center is affected by gravity, and so for each wavelength is different. So using the beam center as measured is correct for the mean wavelength, and is incorrect for the other wavelengths, introducing a smearing effect as drawn in Figure 3 in Boothroyd. The effects of the wavelength distribution and gravity shift partially cancel at positions above the beam center, and add below the beam center.

It is important to note that the effect of gravity is only dependent on wavelength (for a given L2 distance. This sets ygr. Then only the y-shift needs to be accounted for in the variance in q, and the definition of the gravity correction can be in terms of directions parallel and perpendicular to q. The SANS instrumental resolution is naturally described in terms of a 2D gaussian in directions parallel and perpendicular to the q-vector at any point on the detector. Since there is no wavelength spread component to the resolution in the perpendicular (or theta) component, only the parallel component needs to be corrected for gravity. So decompose the parallel wavelength component into x and y components, with the usual definition of phi as the azimuthal angle. The x-component is not affected by gravity, so then the y-component is calculated with an adjusted qy, noting that the qy is a vector and ygr is negative. R is the distance from the beam center (the measured, gravity-corrected one) to the desired pixel. Then correcting the qy component gives the variances:

$$\sigma_{Q_\parallel}^2 = \frac{k^2}{12}\left\{3\left(\frac{R_1}{L_1}\right)^2 + 3\left(\frac{R_2}{L'}\right)^2 + \left(\frac{\sigma_d}{L_2}\right)^2\right\}$$

$$+\sigma_\lambda^2\left(\frac{k}{L_2}\right)^2\left[R^2\cos^2(\phi) + (R\sin(\phi) - 2A\lambda_0^2)^2\right]$$

$$\sigma_{Q_\perp}^2 = \frac{k^2}{12}\left\{3\left(\frac{R_1}{L_1}\right)^2 + 3\left(\frac{R_2}{L'}\right)^2 + \left(\frac{\sigma_d}{L_2}\right)^2\right\}$$

$$A = -\frac{1}{2}gL_2(L_1 + L_2)\left(\frac{m}{h}\right)^2$$

$\sigma_\lambda^2 = \frac{1}{6}\left(\frac{\Delta\lambda}{\lambda}\right)^2$ (Assumes triangular distribution, never mind the units)

The detector element is square, so:
$\sigma_d = \Delta x\cos\phi + \Delta y\sin\phi$

If x and y dimensions of the pixels are different, then
$\sigma_{d,\parallel} = \Delta x\cos\phi + \Delta y\sin\phi$
$\sigma_{d\perp} = \Delta x\sin\phi + \Delta y\cos\phi$

Sigma (parallel) is typically (0.01 to 0.05) 1/A, while sigma (perpendicular) is typically (0.001 to 0.003) 1/A. So then the integration is performed as 2D quadrature over Q_parallel and Q_perpendicular. By definition, Q_perpendicular = 0, but sigma(perpendicular) is not zero.

$$\int_{q_\perp}\int_{q_\parallel} R(q_o, q_\parallel, q_\perp) I(q_x, q_y) dq_\parallel dq_\perp$$

$$R(q_o, q_\parallel, q_\perp) = \frac{1}{2\pi\sigma_\parallel\sigma_\perp}\exp\left\{-\frac{1}{2}\left(\frac{(q_\parallel - q_o)^2}{\sigma_\parallel^2} + \frac{(q_\perp)^2}{\sigma_\perp^2}\right)\right\}$$

So the grid of evaluation points is defined in terms of Q_parallel and Q_perpendicular. At each point, the Q_parallel and Q_perpendicular is converted to (Qx, Qy) since the model intensity is defined in

(Qx, Qy). To do this conversion, first, a new |q| is found, and a new angle, phi.

$$q'_{\parallel} = |q| = \sqrt{q_{\parallel}^2 + q_{\perp}^2}$$

$$\phi' = \phi_o + \theta$$

$$\theta = \frac{q_{\perp}}{q'_{\parallel}}$$

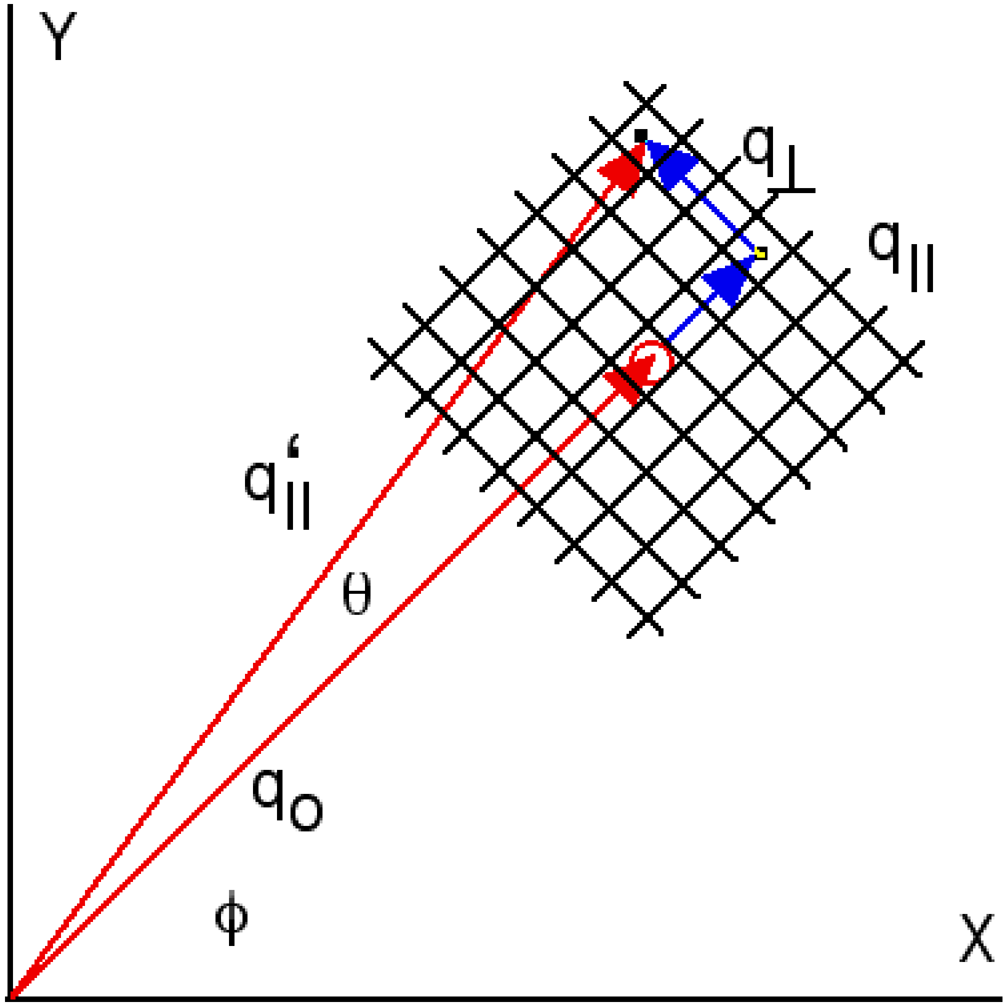
Since the Q_perp distances are small, the angle change can be well approximated using the vector distances, then adjusting the nominal phi.

Then from any point (Q,phi), (Qx, Qy) can be found from:

$$q_x = \sqrt{\frac{q_o^2}{1+\tan^2\phi}}$$

$$q_y = q_x \tan\phi$$

The signs of Qx and Qy are found by knowing the quadrant that phi lies in, based on phi starting at the positive x-axis and moving CCW.



Examples of the resolution calculation:

The resolution function was integrated to ensure that it does correctly integrate to V_avg = 0.995 for 3 standard deviations. This check is done simply by removing the function value from the quadrature. Then the integration is simply the resolution function.

So it is clear that gravity corrections are only of concern in the cases:

1) large wavelength spread (> 20%)
2) long SDD and/or long wavelength, so that ygr is large
3) very sharp features (and measurement) near q*

As noted in many references, TOF SANS usually does not need to include gravity corrections since the wavelength distribution is binned into narrow slices and the gravity contribution to the resolution naturally drops out. From the examples, the wavelength spread needs to be rather wide to see any effect at all, typically at the largest values used (> 25%) in steady-state SANS.

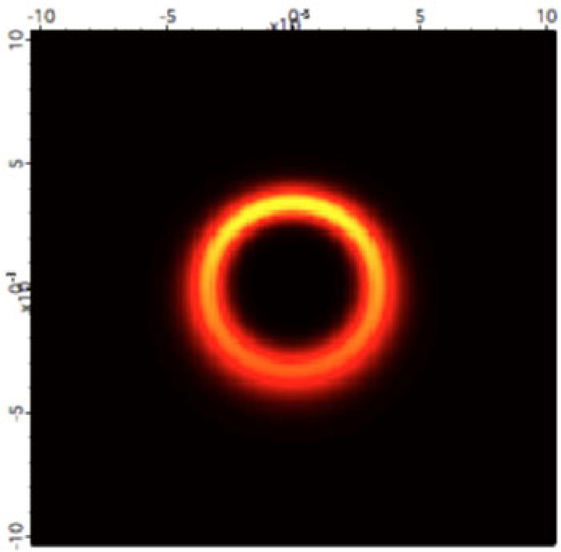
Example calculations:

Smeared Gauss Peak 2D:
Scale = 100
Peak position = 0.0034

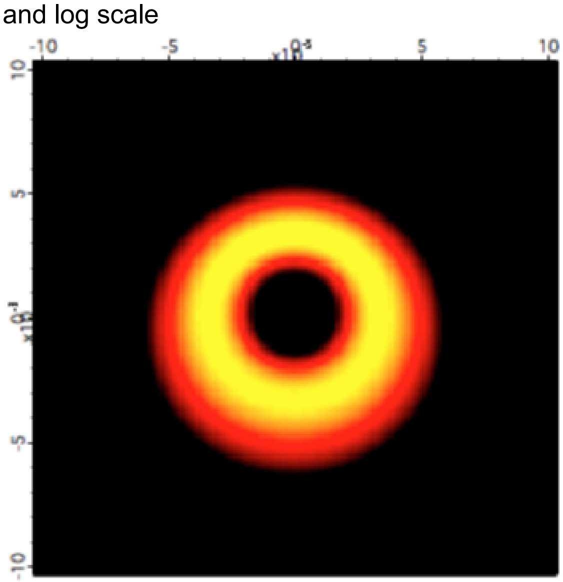
Std dev = 0.0003
Bkg = 0

1) 15A, 13m, 24% peak @ 2q* = 2*0.0017=0.0034
ygr = -2.73 cm
There is very visible distortion at these conditions:

linear scale:



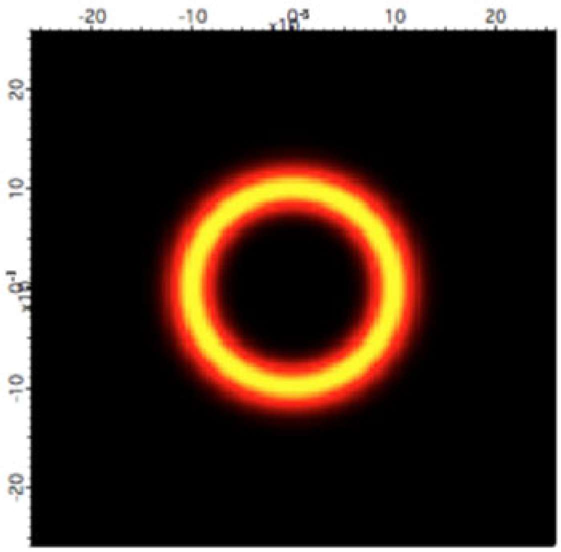
and log scale



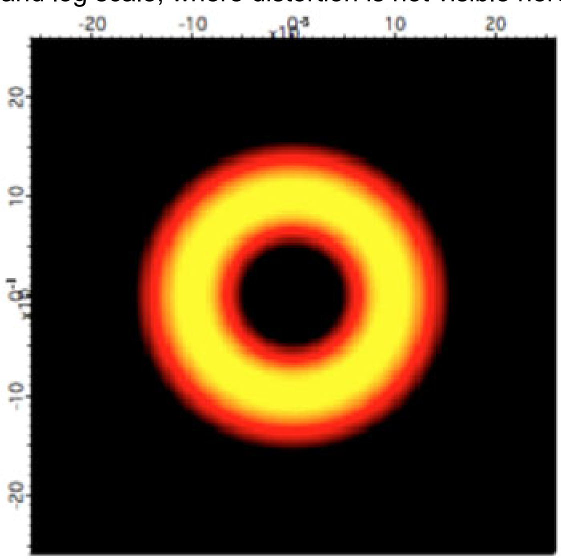
2) 6A, 13m 12.5%, peak @ 0.01 (q* = 0.0007 and is not accessible)
ygr = -0.44 cm

These are typical conditions for SANS, and the gravity effect is not visible, even for very sharp scattering features.

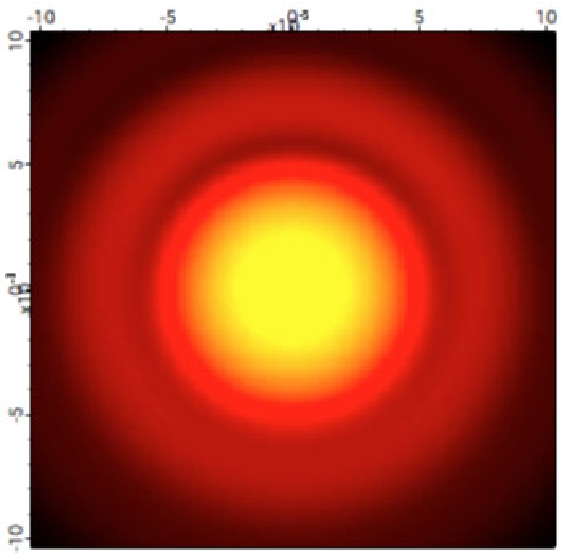
linear scale



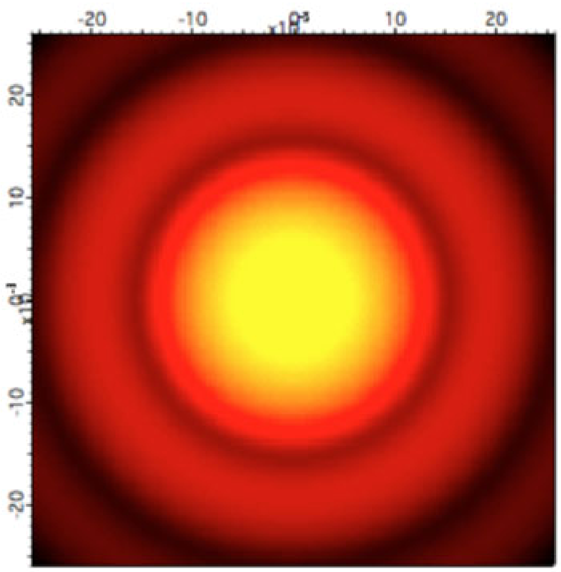
and log scale, where distortion is not visible here either:



3) For sphere scattering (R=800), the effect is very subtle, even for the 15A conditions. This is an effect that would probably not be visible in a real experiment.



For R=300A and the 6A conditions, the effect is not visible at all:



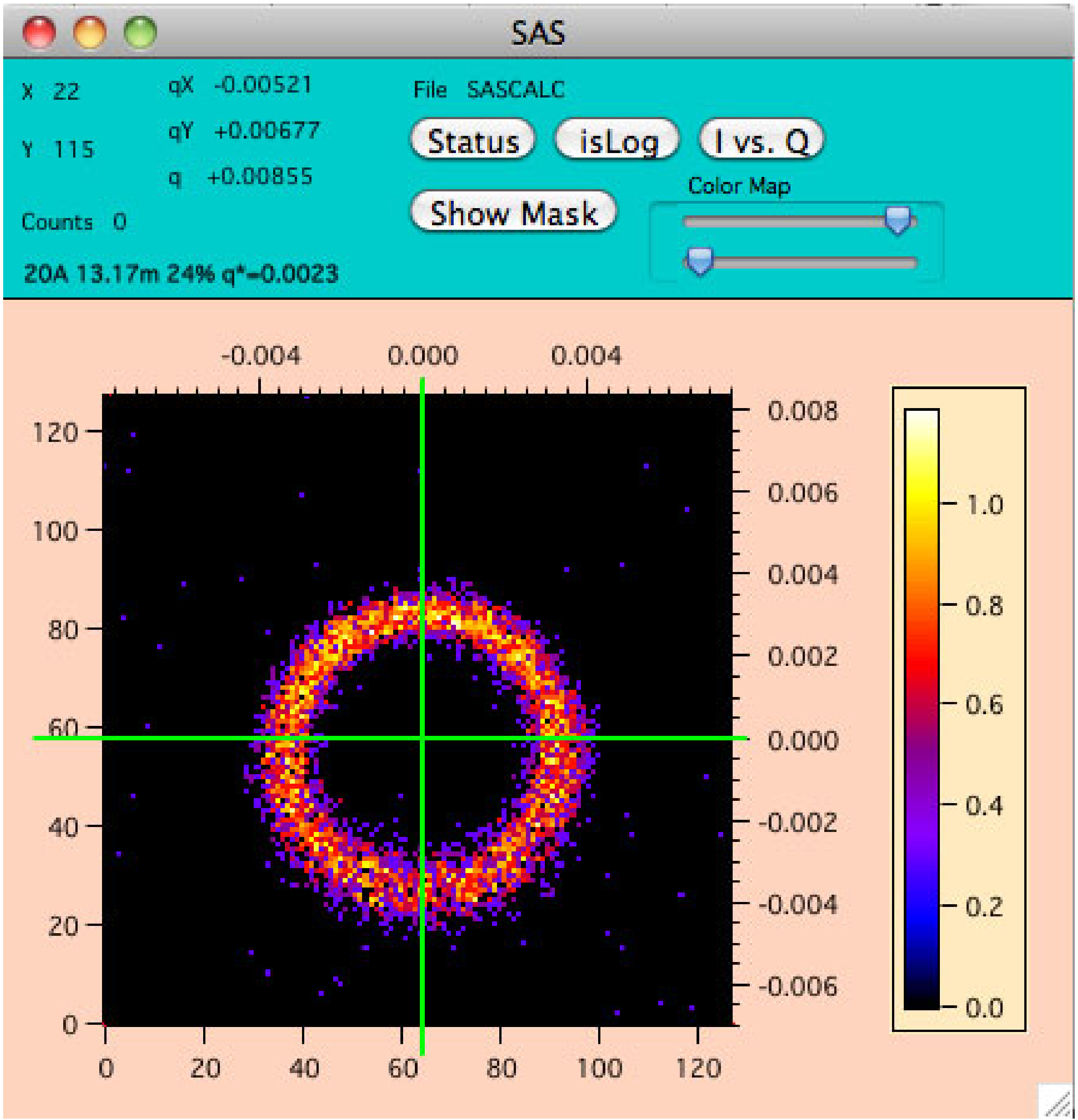
Resolution vs. Position on the Detector

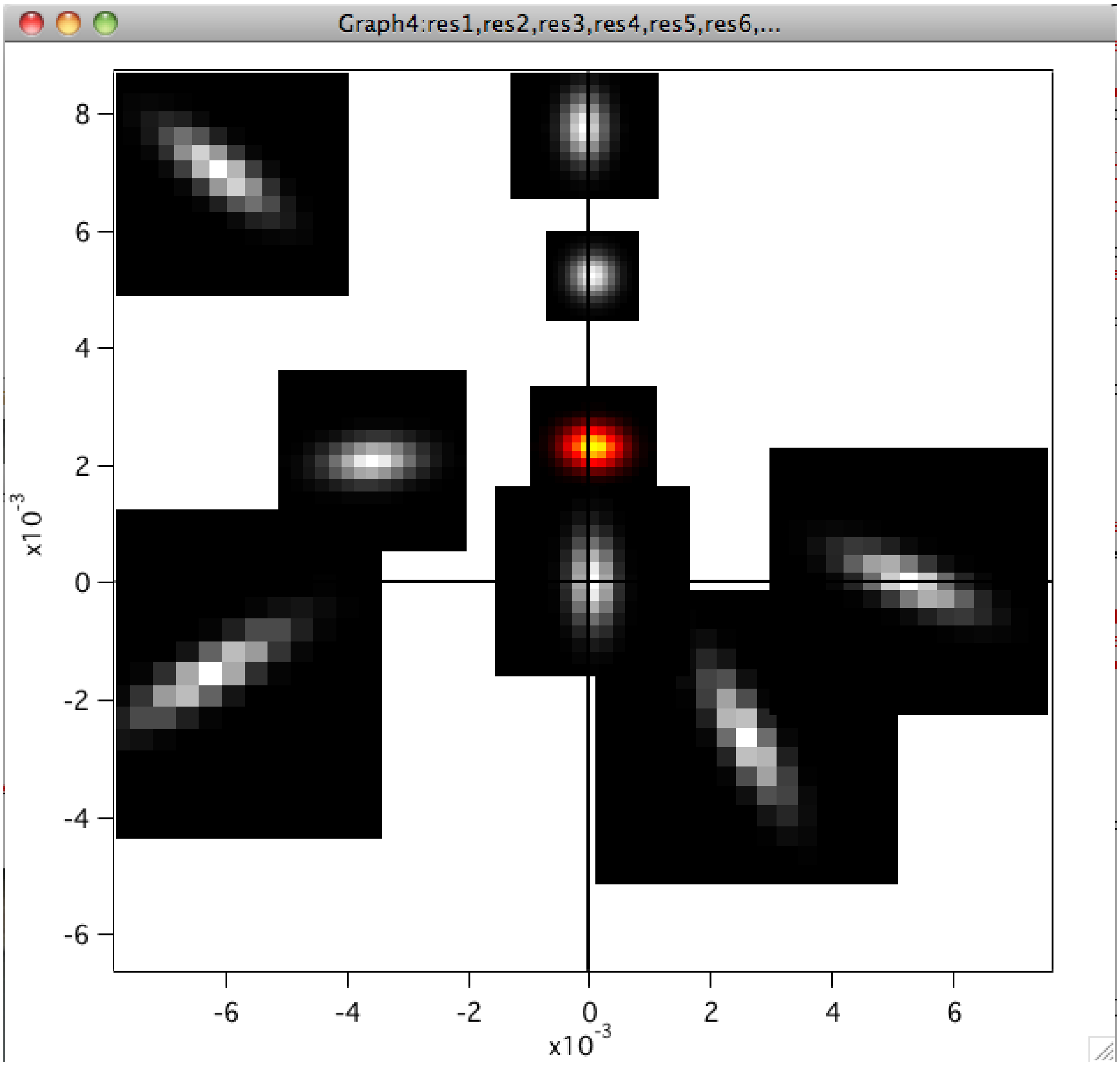
The function:

PlotResolution_atPixel("RAW",pcsr(A),qcsr(A))

was used to plot the resolution function as a 2D Gaussian at selected points on the detector. The resolution was calculated as components parallel and perpendicular to Q. For display, the elliptical Gaussian was rotated into the XY plane.

The 2D Gaussian is normalized at each point, and integrates to ~0.997 using a box of +/- 3 sigma in the parallel and perpendicular directions. The 2D SANS pattern below is simulated data at lambda = 20 A, dL/L = 24%, and SDD = 13 m to show the maximum effect of gravity on the resolution. The second plot is the resolution function at various points on the detector showing that the resolution function points back to q* = 0.0023 (1/A) rather than the measured beam center. As expected, the resolution function has a larger variance in the direction of Q due to the wavelength spread, and becomes more symmetric as Q=0 is approached. Gravity has been included in these calculations and is seen as an extra "skew" of the resolution function downward (y).





****

For ***visualization only***, it is simpler to have the resolution function rotated into an (x,y) coordinate system, rather than in terms of Q1 and Q2. The rotation of a general elliptical Gaussian was taken from Wikipedia:

http://en.wikipedia.org/wiki/Gaussian_function

NOTE: their definition of theta starts at the +Y axis, and proceeds CW. Our definition of angle = phi starts at +X, and proceeds CCW. In the code, setting theta = -phi gives the proper results... so there's an implicit pi/2 shift in there somewhere that I'm not seeing. I think it's because I'm calling x parallel rather than y. But anyhow, it's correct.

(straight from the web page)

In general, a two-dimensional elliptical Gaussian function is expressed as

$$f(x,y) = Ae^{-\left(a(x-x_o)^2+2b(x-x_o)(y-y_o)+c(y-y_o)^2\right)}$$

where the matrix

$$\begin{bmatrix} a & b \\ b & c \end{bmatrix}$$

is positive-definite.

Meaning of parameters for the general equation

For the general form of the equation the coefficient A is the height of the peak and (x_0, y_0) is the center of the blob.

If we set

$$a = \frac{\cos^2\theta}{2\sigma_x^2} + \frac{\sin^2\theta}{2\sigma_y^2}$$

$$b = -\frac{\sin 2\theta}{4\sigma_x^2} + \frac{\sin 2\theta}{4\sigma_y^2}$$

$$c = \frac{\sin^2\theta}{2\sigma_x^2} + \frac{\cos^2\theta}{2\sigma_y^2}$$

then we rotate the blob by an angle θ.

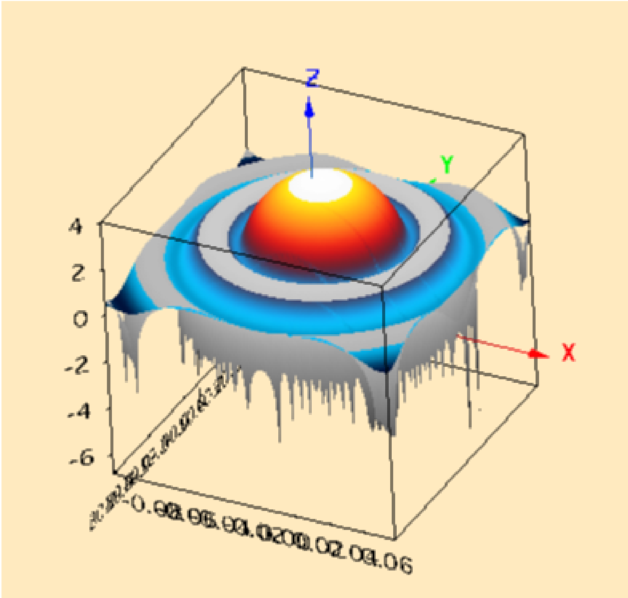
****

Examples of the 2D smearing at work

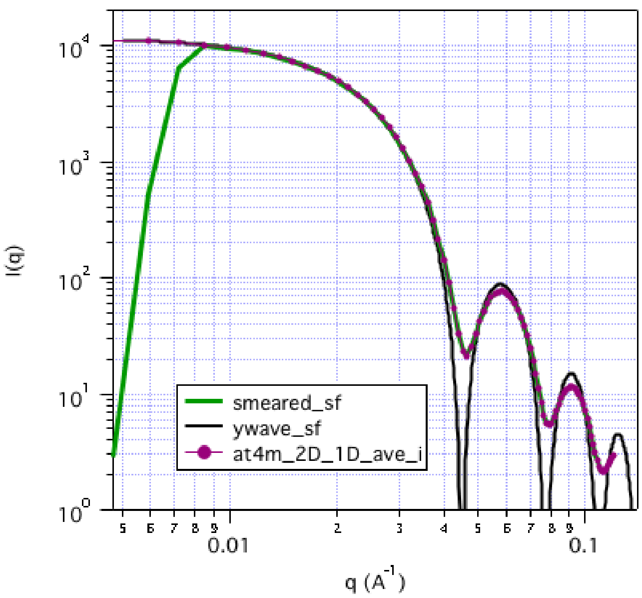
To test the smearing, the silica data from the tutorial was saved as **I**(Q) data and as **I**(Qx, Qy) data. A sphere form factor of R=100 was plotted, smearing in either 1D or 2D. To compare the 1D plots, the 2D matrix data was copied into a work folder with the appropriate header information. Then the 2D data was averaged to 1D. The results below are for SDD = 1m and SDD = 4m.

For (128,128) detector array and 10 Gauss points in each direction (100 function evaluations for each pixel), the calculation takes about 20 sec on a 1 GHz processor. Threading gives a speedup of Np. For 2 processors, the calculation takes 10 sec. For 4 processors @ 2.8 GHz, the calculation takes approx. 0.8 s.

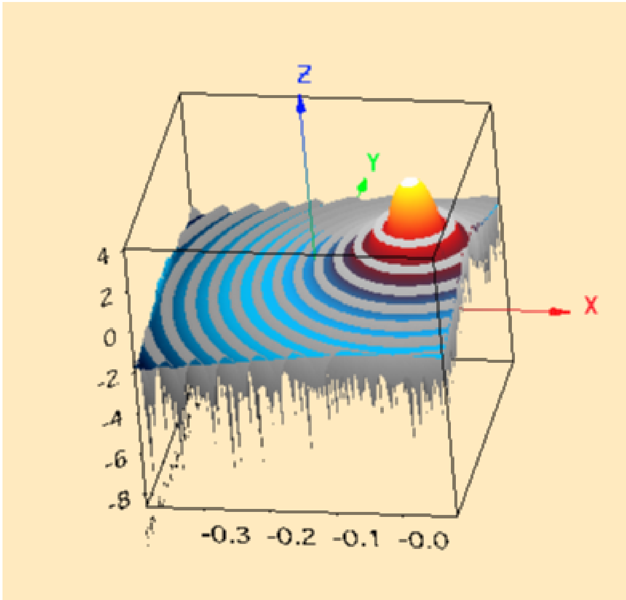
At 4m:
(grayscale is the unsmeared model, coldWarm is the smeared model)

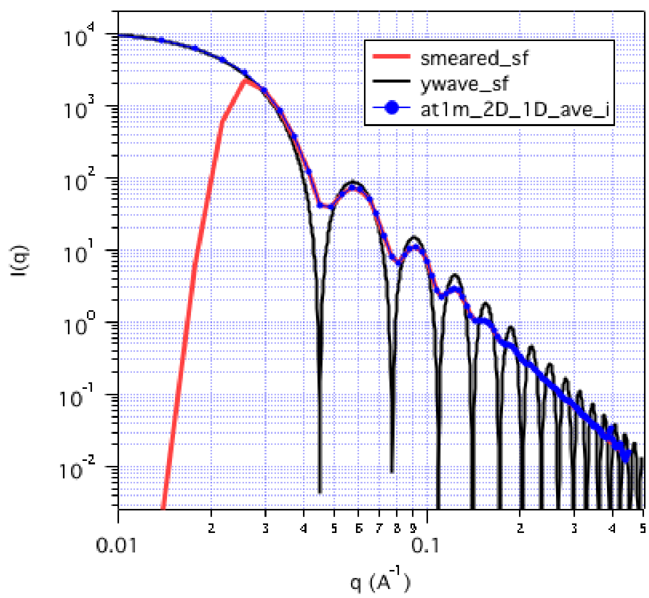


In 1D - the "_sf" are the unsmeared and the 1D smearing calculations. "at4m_2D_1D" is the above 2D data averaged to 1D. The match is excellent. The beamstop shadowing is not included in the 2D resolution function, but may be in the future.



At 1m:





- **Propagation of Uncertainties in 2D**

18 MAR 2011 SRK

Proper error for Poisson statistics, good for low count values too, rather than just sqrt(n) can be found in:

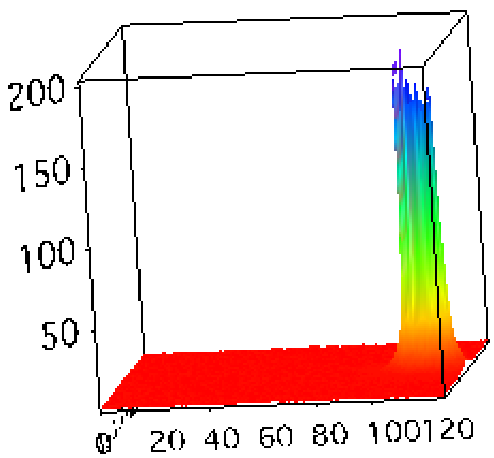
N. Gehrels, Astrophys. J., 303 (1986) 336-346, equation (7)
--for S = 1 in eq (7), this corresponds to one sigma error bars

=> linear_data_error = 1 + sqrt(linear_data + 0.75)

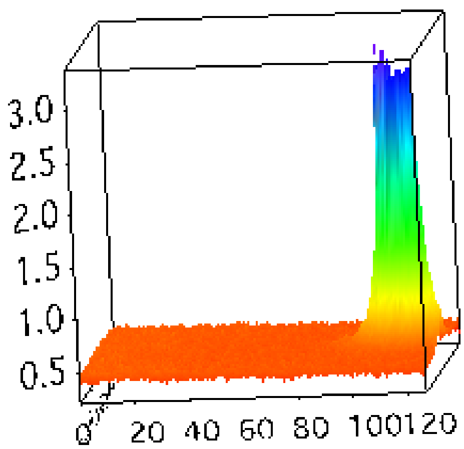
A separate document details all of the equations regarding the manipulation of the 2D data as is is loaded and subsequently flows through ExecuteProtocol(). During the SANS reduction, the per-pixel errors are automatically propagated through each mathematical operation of reduction, with the final per-pixel error written out (only) when the data is saved in the 2D QxQy ASCII format, suitable for 2D analysis.

Results of the Calculations:
The results of the error calculation are stored in every work folder. Data manipulations are carried out on the linear_data, and uncertainties are propagated in the linear_data_error wave.

For the BEAD185 data: the data looks like (ABS):



and the uncertainty in 2D is:

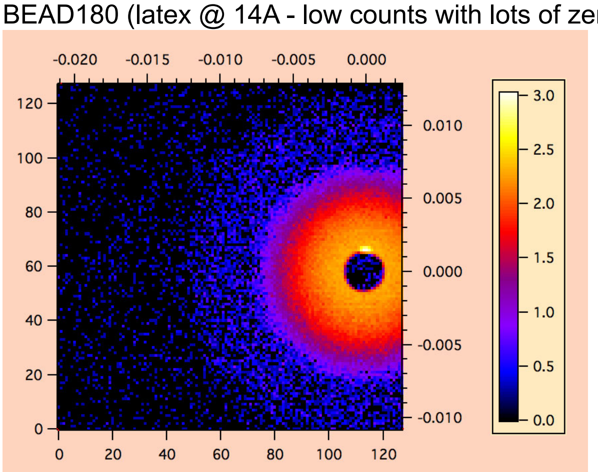


The uncertainty, oddly enough, has a shape that is proportional to the original data (duh). It can be seen that the error is not simply the square root - sqrt(200) != 3. But rather the error is derived from (mostly) the counting statistics of the sample measurement, and the relative uncertainty that arises from those counts.

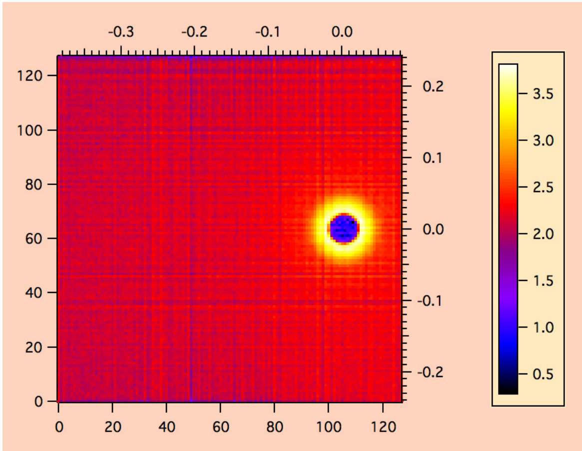
Results after averaging:

Results from three different files are shown.

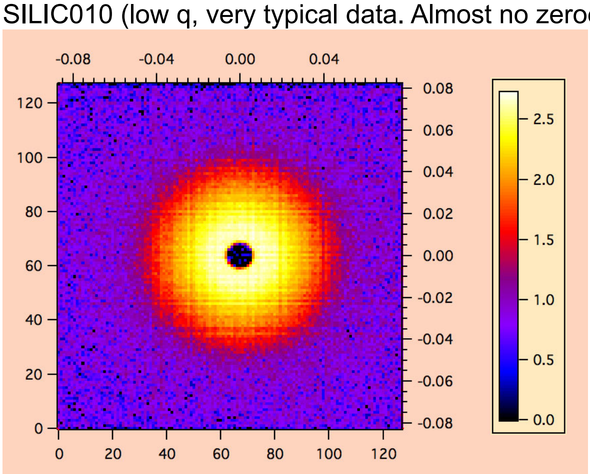
BEAD180 (latex @ 14A - low counts with lots of zero values)



SILIC009 (high q, lots of counts everywhere)



SILIC010 (low q, very typical data. Almost no zeroes, but lots of low count pixels)

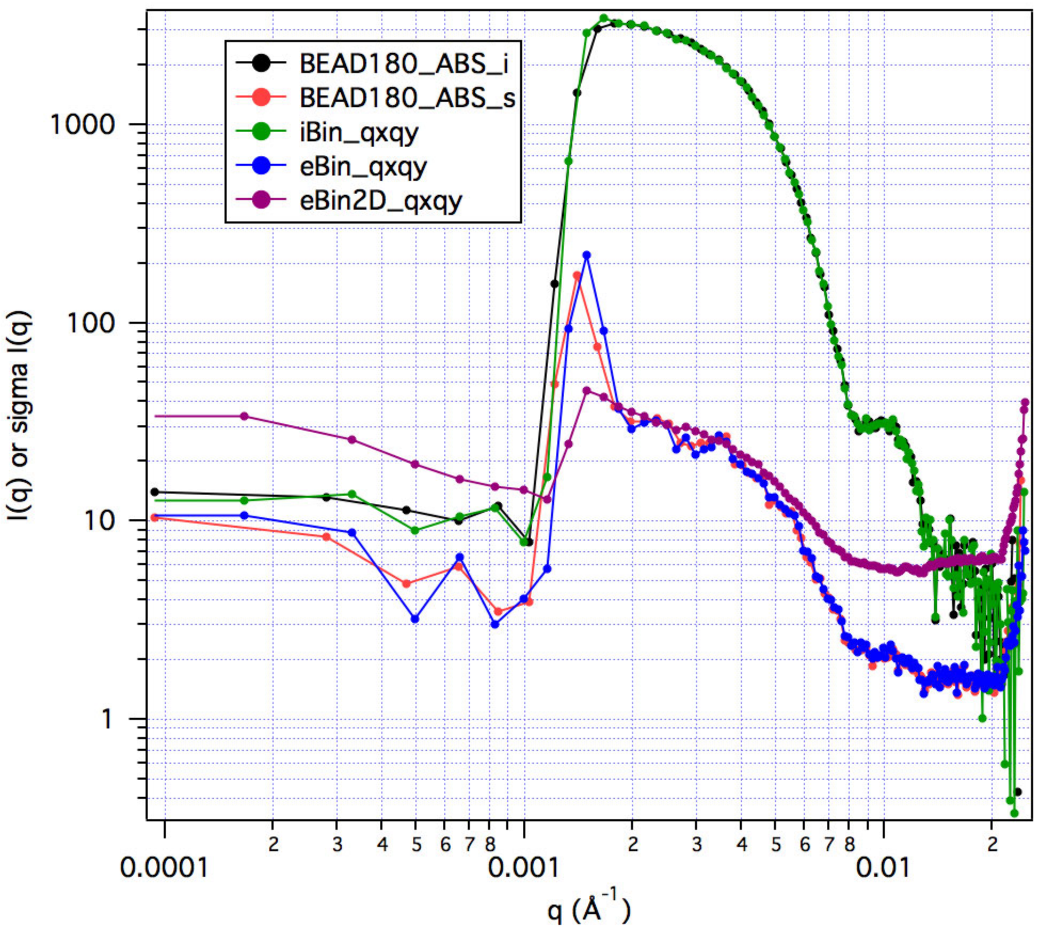


And the results averaged to 1D are compared by taking the QxQy ASCII output and binning that back to 1D. The waves shown are:

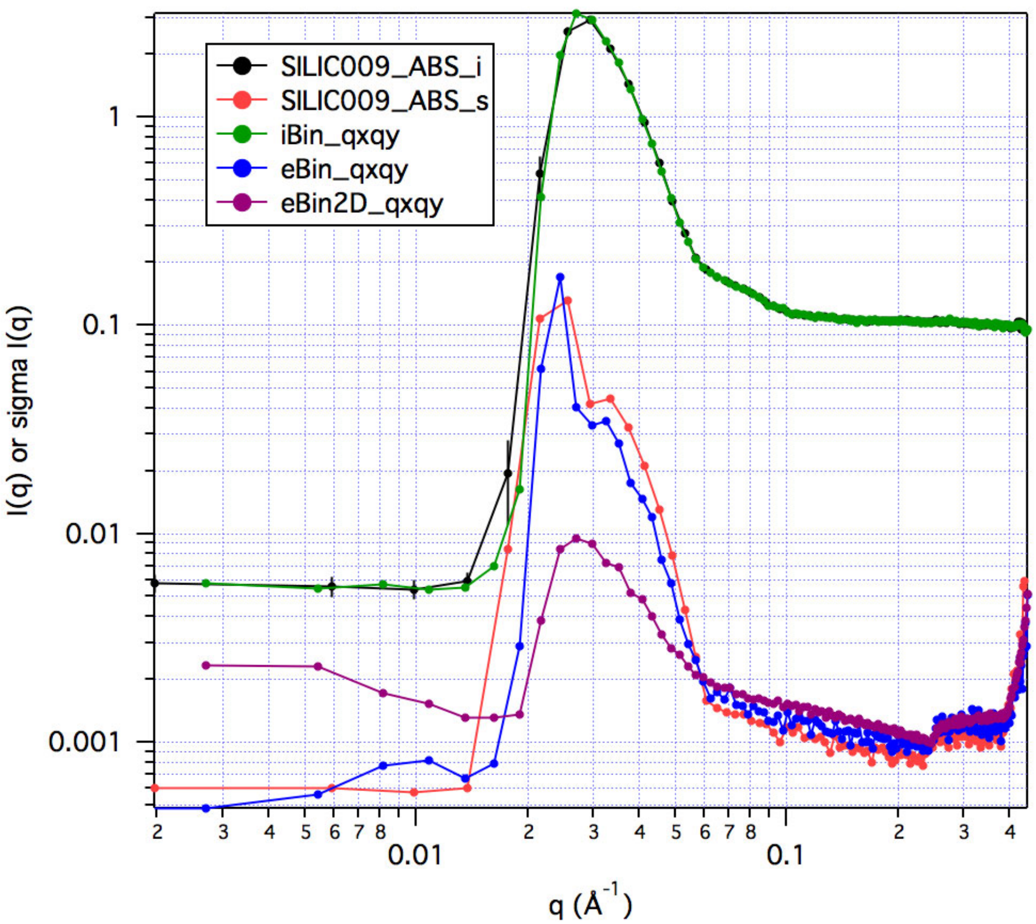
DATAnnn_ABS_i - from the normal processing
DATAnnn_ABS_s - errors as deviation from mean (the normal use)
iBin_qxqy - I(q) re-binned from QxQy
eBin_qxqy - errors rebinned, deviation from mean
eBin2D_qxqy - per pixel error propagated in each bin

Errors were derived from the QxQy data since it was far too cumbersome to modify the CircSectAve routines to do the averaging there. For all three examples, the intensity from the normal 1D average and the re-binning of the QxQy data are identical (but are at different q-values). The uncertainties are somewhat different, as seen below.

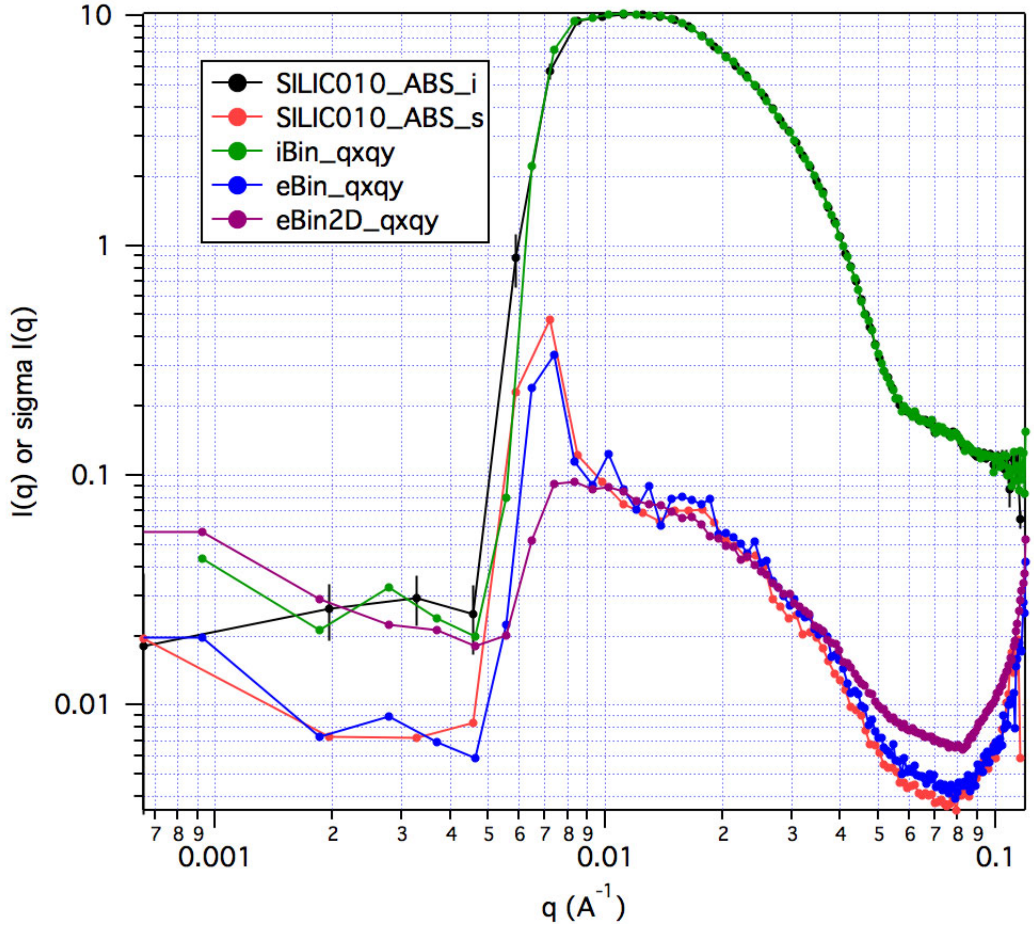
For data with lots of zeros, BEAD180, the uncertainties are similar in the mid-q range. At low q there is a flare at 12:00, leading to a very large deviation from the mean (As expected), and a smoother response for the 2D errors. At higher q, the 2D error is larger since the data is dominated here by count values of 0/1/2 and are now handled in a correct manner.



For data with good counts everywhere, SILIC009, the errors are all very similar. The 2D error is somewhat lower (about 4x) at the region near the beam stop. Why this is, I'm not entirely sure, but the data here is of very high counts, so the relative error per pixel is probably very small, maybe less than the variation from the mean going around the whole annulus. Not that much different, for data that's going to be thrown out anyways.



And for "typical" data with a range of counts, SILIC010, the results are similar to the BEAD180 data, but with not so much of a difference at the higher q values, since there are not as many zeros in the data set. Here, the 1D and 2D propagation are so close that either method will give a good value for the uncertainty.



What's missing:
- If ABS scaling is done with values from a secondary standard, the uncertainty in I(q=0) must be entered manually. The value defaults to zero and provides no contribution to the overall uncertainty if it is not entered.
- The uncertainty from the DIV operation is not included. The uncertainties from the DIV data are typically more than an order of magnitude smaller than the data uncertainties. So there is no reason to bother with the calculation. In addition, there is no mechanism to carry the DIV file errors except in another file. [I see no good reason to carry the DIV errors by hacking the current VAX format. Once the switch to a new data file format is done, then the newly-defined DIV files could (and should) carry their own uncertainties, as a processed 2D data set].

Notes:
- For some more of the plots and details, see the experiment file: 2DError_Verificaton.
- Error propagation has been added to the WorkFileMath operations. These operations work with ASCII (detector pixel) data, and as such do not have any error information associated with them in the data file, so the data as loaded in is treated as raw counts and errors are generated as when loading RAW data files. If the data is not raw counts, then the errors may not be representative of the true errors.
- Zero events in any pixel are not discarded at any point in the reduction, as this would skew the results.
- Low count events are not "resampled" when rescaling to monitor counts for the same reasons.
