

“Nika” manual

Version 1.43

Igor 6.10 and higher compatible ONLY

Date: Tuesday, February 23, 2010

Jan Ilavsky

Ilavsky@aps.anl.gov

@2009

This is manual for set of macros developed for Igor Pro (Wavemetrics, Inc, www.wavemetrics.com) version 6.10 and higher. These macros are designed to process 2D (CCD and other area detectors) data from small-angle and wide-angle scattering instruments. The purpose is to process the 2D data and convert these into 1D “line outs” data – providing correctly calibrated Intensity, q , and errors.

From version 1.42 Nika was redesigned to provide more capabilities in options how to extract the data:

1. (Original) Method of Sector and circular averages
2. Intensity along linear and elliptical path (vertical/horizontal lines, line under and angle and ellipse of arbitrary aspect ratio)
3. Intensity along linear path but for Grazing incidence geometry
4. Intensity vs azimuthal angle image intended for manual inspection of geometry.

Options 2 and 3 are new in version 1.42.

Version 1.43 additions:

New file formats, display Q_x/Q_y axes around CCD image, changed uncertainty calculations options.

Content:

INSTALLATION AND LOADING MACROS	4
Installing macros	4
Loading the macros	5
Removing the macros	5
Configure default fonts and names & Errors	5
Selecting data	10
Data Types which Nika can load	10
Igor native loaders	10
BSL/SAXS and BSL/WAXS data format	11
General Binary data loader	12
Other loaders with panels...	13
Using XOP to load	14
Fit2D data files	14
Invert 0,0 corner	14
Bottom controls	15
Geometry/processing controls	18
Main	18
Param	20
Mask	23
Emp/Dark	24
Sectors	25
Prev	27
LineProf	31
Main 2D image	39
Polarization correction	40
Implementation	40
UNCERTAINTIES (“ERRORS”)	42
BEAM CENTER AND GEOMETRY REFINEMENT TOOLS	43
Beam center using attenuated beam	44
Beam center using “help circle”	45
Calibrant & refinement	47
MASK	51
OUTPUT DATA	52
Internal	53
The wave names	53
ASCII external	54
INSTRUMENT SUPPORT	54
DND CAT (APS 5ID) SAXS camera	55
CREATE FLOOD FIELD	55
GUI	57

Installation and loading macros

Installing macros

There two methods of distribution available to users.

1. Distribution using Igor Experiment as installer and ftp server. Igor experiment, capable of using ftp server is available from my web site. This experiment is cross platform compatible and the most convenient method of installing the Nika (and other) package. I strongly encourage users to use this method. It requires access to the web for regular ftp (NOT sftp), so it may be challenge to use this method on some highly protected networks.
2. Old method of package distributed as zip file. The zip file includes paths as appropriate with respect to Wavemetrics folder. This is actually problem for Igor 6.10 and higher, as there are now two places, where Macros can be installed... Using pre-Igor 6.10 method (which works for now) macros are installed in Igor Pro Folder. When installed, Igor pro is installed in:
Disk:/XXX/Wavemetrics/Igor Pro Folder
Where on Windows the XXX = “Program Files” by default, On Mac this is in “Applications” folder. User can change defaults on both (Windows & Mac) systems.
These folders MAY NOT be writable for user, if he/she is NOT admin of the computer. Therefore Igor 6.10 and higher has preferred location, which is created in users home folder/directory. This place can be easiest accessed from “Help” menu in Igor.

When upgrading manually using zip file – you MUST remove old version of Nika and other included packages (there is one from Irena and one from Indra, which are shared) from both old (in Igor Pro Folder) and new (Igor Pro 6 user files) locations to avoid conflicts. The Installer Experiment does this for you, so preferably use this method...

Important is, that the folder structure within the Igor Pro Folder/Igor Pro 6 user files is required and must be kept for Igor to properly find all components.

There are two places for user macros:

...../Igor Pro Folder/Igor Procedures

Here are placed macro files which need to be loaded every time Igor starts
..../Igor Pro Folder/User Procedures

Here are placed macro files which may be loaded sometimes through “include” statement.

For Nika package to work right, single file (“loader”) needs to be included in Igor Procedures and rest of files (all others) need to be included in subfolder “Nika” in User Procedures. The files are properly stored in these folders in zip file. Please, place the files in corresponding folders.

Optional xop for MarIP (image plate) format

Mar Image plate format is really cumbersome and cannot be efficiently read by Igor code, therefore I have initiated development of xop to read this image file. This xop was developed by Jonathan Guyer and the code is available from :
<http://www.igorexchange.com/project/CCP4XOP> . It is, however, due to licensing of one library distributed as source code and needs to be compiled by the user. If you do not want to compile the code your self, send me an e-mail request and I will send you compiled xop for your platform. I believe this complies with the licensing of this ccp4 library.

This xop belongs to folder :
...../Igor Pro Folder/Igor Extensions

Comment version 1.42 (10/27/2009) – I was unable to recompile this xop for latest version of OSX and long file names again. Therefore this xop is likely not available for Mac at this time.

Loading the macros

Install macros (see above)
Start Igor Pro
In menu “Macros” select “Load Nika 2D SAS macros”.
New menu “SAS 2D” appears. This is where all the Nika macros are controlled from.

Removing the macros

There are two items on may wish to do to unload the macros from any experiment. First to remove the large lookup tables associated with the methods Nika is using to convert 2D to 1D data. This is done by selecting “HouseKeeping” from “SAS 2D” menu. This will make the Igor experiment much smaller (often by 60 Mb or even more).

Then removing the macros itself is achieved by selecting “Remove Nika 1 macros” in the “SAS 2D” menu. This will unload macros and put back in the “Macros” menu command to load Nika macros, if necessary.

Configure default fonts and names & Errors

“Configure default fonts and names” in the SAS menu will create panel with some controls common for all tools, like font type & size and how legend names are handled.
NOTE: Panel controls are applied immediately to all existing panels, graph controls are applied ONLY to the newly created graphs (and only those which were upgraded to this behavior).

Panels font and font sizes

These controls enable user to customize font used on control panels therefore this enables customization for a given platform. This is necessary as more and more control is provided on each platform to user and therefore default fonts and font sizes may not be appropriate any more for the panels I design. These settings are actually saved on a given machine as well as the experiment. This has some interesting features, so please, read carefully:

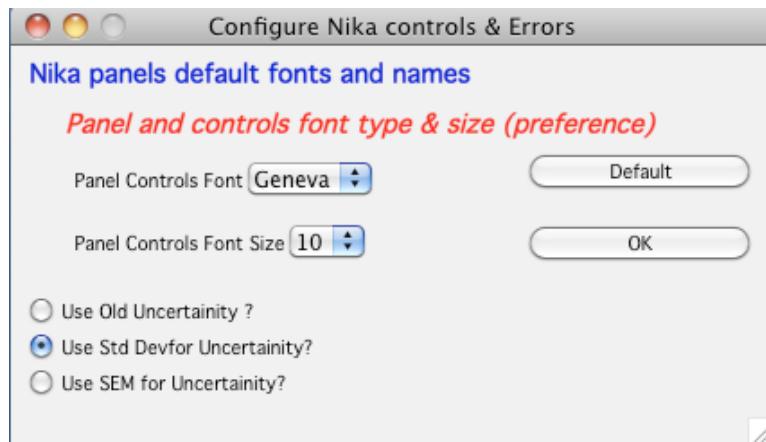
When these controls are run (and user is forced to run them if the Irena is loaded and preferences are not found), they save preferences in special folder Igor maintains for users. At the same time, the settings are applied to the current experiment.

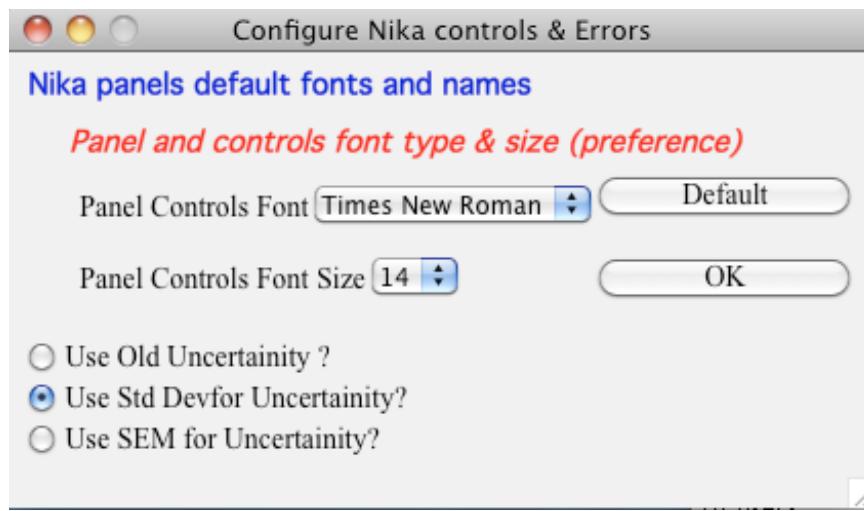
When this experiment is opened on another computer, the preferences from that computer are not reloaded, so the experiment will use preferences from the original computer. When the “Configure GUI and Graph defaults” is run, it will reload the computer defaults and apply them to the given experiment. Then user can change the fonts and font sizes as they wish. The new settings are saved on the computer – and within the experiment.

Note, that Panel font and font size are platform specific, so same experiment may present differently looking panels on Mac and PC.

Note, not all controls actually follow these settings, I have been changing some buttons to specific font and font size and those are not affected by these settings.

If there are any issues with the behavior, please, let me know and I'll see if I can make it more logical.





Note the difference in Configure GUI and Graph defaults panels when different fonts are used. Left is using Geneva font size 9, right is using Times New Roman size 14, both on Mac platform. You can mess up the panels really well by wrong choices!

Defaults button returns the panel font choices to platform specific default state (Mac: Geneva size 9 and PC Tahoma size 12). Note, that there is no guarantee that these were your choices before. But these should be reasonable choices for most setups.

Error (uncertainty) estimates for points

Up to version 1.42 Nika used “error” calculation, which seemed to produce reasonable numbers inmost cases. But one of the users realized, that it has error in it (bug) and fails in cases of really low intensities, which can be assumed to be Poisson distribution.

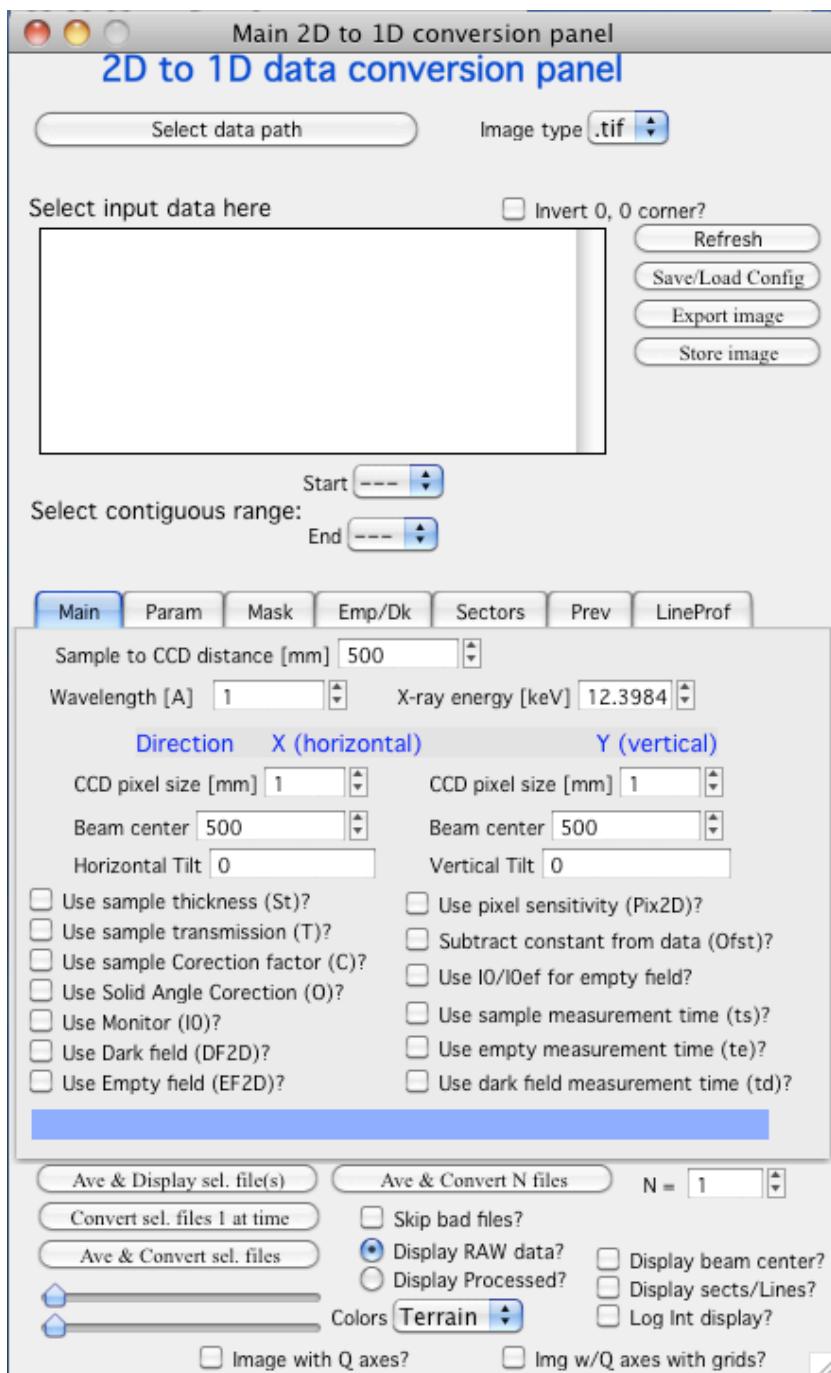
Therefore, from version 1.43 Nika has three options for this calculation.

1. Old method (default for compatibility reasons). As I said, it has a bug, but seem to work most of the time well enough.
2. Standard deviation
3. Standard error of mean

Please note, that Standard error of mean (SEM) is miserably small for high-intensity instruments and generally is too small. At least in my experience. Standard deviation was what I was trying to get with the old method – there should not be much difference between them most of the time.

4. Main control window

Select “Main panel” from the “SAS 2D” menu. This will present the following panel:



The panel has three major parts:

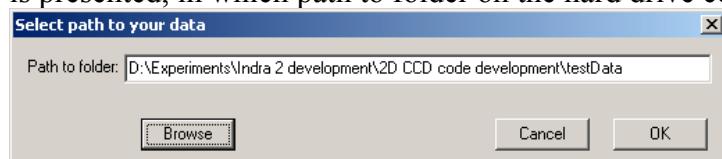
1. Top is designed for 2D data selection. Here user selects which 2D image will be processed.
2. Middle (tabbed area) is designed for controls of processing. This is the busiest area of the panel and each tab will be explained later.

3. Bottom contains buttons for main controls and 2D image controls.

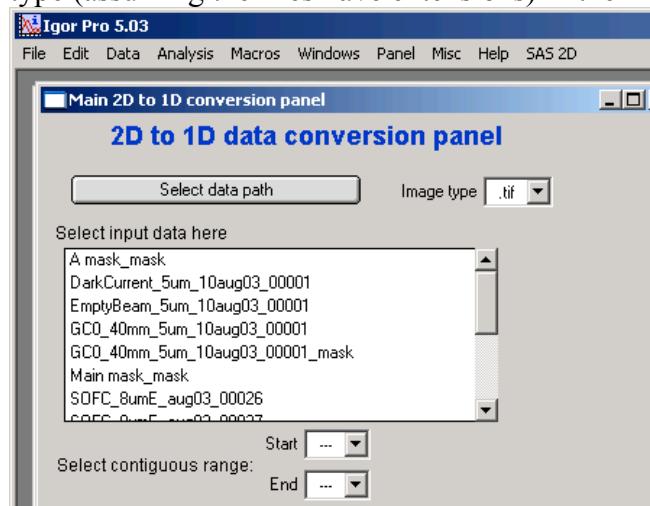
Selecting data

The macros can load selected data types. If appropriate file type is not found in the “Image type” popup menu at the top right corner, you will have to contact me so I add appropriate loader for your data.

Select appropriate type of data you have and then push “Select data path” button, dialog is presented, in which path to folder on the hard drive containing 2D images is selected:



When valid path is selected, the Igor will check the folder and list all files of appropriate type (assuming the files have extensions) in the ListBox below the button:



Here user can select one files, more files (by holding down shift key on Windows) and continuing selection (using the two pull down menus below the list Box)...

Note, the files ending with “_mask”. These are mask files created by Nika package, these are always tiff files... Separate chapter explains how mask is created.

Data Types which Nika can load

Igor native loaders

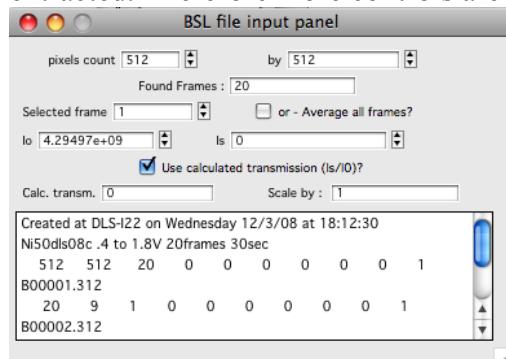
These are file types, which Nika can load natively. These can be loaded on any platforms:

tif	tif file
GeneralBinary	flexible binary data loader (see description below)
Pilatus	Should load NON COMPRESSED Pilatus data. Opens new screen with options
BrukerCCD	bruker SMART software for CCD
mpa	The software is MPA-NT (or just MPANT), version 1.48. It is from FAST ComTec, a German company that supplies multi-channel, multiparameter data collection and analysis tools. The hardware is the MPA-3 Dual-parameter multichannel analyzer (from

	FAST ComTec). That hardware provides the interface to multiwire 2D gas-filled X-ray detector from Molecular Metrology (recently purchased by Rigaku/Osmic).
DND/txt	This is loader specifically for data from DND CAT and APS. The data are tif files, but Nika will read text processing record, extract appropriate parameters and use those for reprocessing of the data. There is specific chapter in the manual on use of data from DND CAT.
mp/bin	mp binary format. for software above, Has header and binary data behind it.
mp/asc	mp format, ASCII column of data. Assumes squared shape (N x N pixels).
BSRC/Gold	BESSERC 1536x1536 Gold detector binary format. It has header and 16 bit binary data
RIGK/Raxis	This is for Rigaku file “86” format. The code should be able to handle any size images, but tested was on 1k x 1k and 1.5k x 1.5 sizes. I tested the code against Fit2D and based the code on Rigaku C-code.... NOTE: known issue is with newer Raxis file format “100”. Need to rewrite this reader.
ADSC	binary file with header. Header contains “HEADER_BYTES” as first element.
WinView spe	Princeton WinView file format.
ASCII	ASCII matrix file format. If the file has extension “mtx” the code assumes there is accompanying file with extension “prm” which is read and parameters from that file are placed in the right Nika variables.
ASCII 512 x 512	one column ASCII data, present for 512 x 512 pixels.
Ibw	Igor binary wave format. This may be useful if the data are produced by Igor.
BSL/SAXS and BSL/WAXS	BSL/OTOKO file format. See http://srs.dl.ac.uk/ncd/computing/manual.bsl.html for description. Note: you need at least 3 files – file with name Xnn000.mdd which is header file, Xnn001.mdd contains one or more SAXS image4s with Xnn002.mdd calibration file, and/or Xnn003.mdd file with one or more WAXS images with Xnn004.mdd calibration file. The set is listed only one – with “proper” file name in the list of samples. Note, that without the inf file the loader fails and gives error. Read details below, please...
Fuji/imp	Fuji image plate reader (BAS2000 and BAS2500). Nightmare. Anyway, this image plate reader should be able to read 8 bit and 16 bit data from these two readers. There is a lot of variability of this code and I was unable to get test data for 16 bit images. Only 8 bits are tested to work at this time. If you have data from other settings, please, send them for testing to me.
ESRF/idf	ESRF ID2 “edf” file format. Should read also other edf formats but has not been tested. Reads ONLY files with ONE image per file – principally the format enables multiple frames (images) in one file, but this is really difficult to support. What to do with all those images???

BSL/SAXS and BSL/WAXS data format

Please note, that this “container” with up to 5 files has more information, which can be extracted. Therefore more controls are needed and separate panel opens up, see below:



Following are loader details:

At the top are sixes of pixels for selected images, in this case the BSL/SAXS container contained images with 512 x 512 pixels. It contained 20 images, as indicated by “Found Frames” value. User can select to evaluate average of all 20 images (select checkbox “Average”) or individual frames.

I0 and Is are extracted from the associated calibration file. In this case I0 has some value but there are no numbers for Is. I0 is ion chamber before the sample (incoming flux, monitor...) and Is are after the sample. Principally, when present, the ratio should be transmission, which is calculated for user and placed in “Calc. transm.” value – note, this value is *always* Is/I0. Here it is 0, since Is is = 0.

In case the ion chambers had different sensitivity or Is is actually different detector type, user can also scale the Is/I0 by some ScalingFactor to correct for sensitivity.

Some of these values can be transferred to Nika variables:

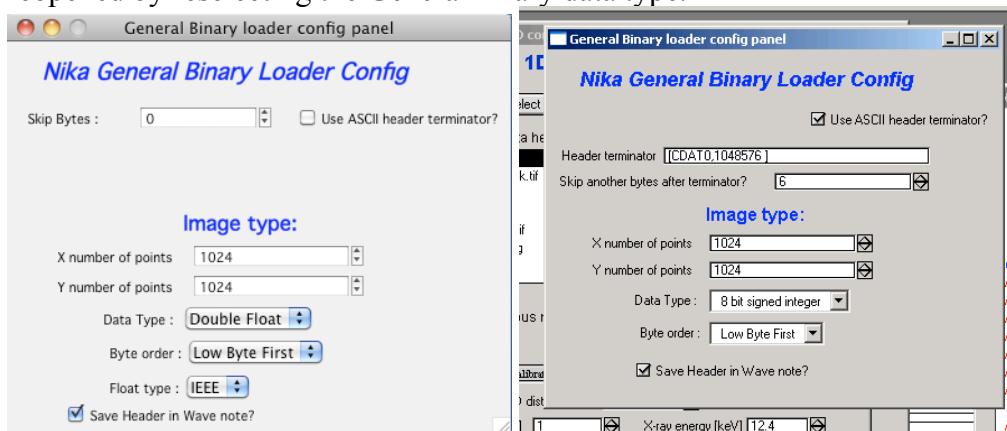
Always the I0 is transferred to Nika calibration value for I0. User can use it by selecting the “Use Monitor?”.

On the BSL panel is checkbox “Use calculated transmission”? – when selected, value of (ScalingFactor*Is/I0) is transferred to Nika’s “Sample transmission” value for calibration purposes. It is used when checkbox “Use sample Transmission” is selected, see later Calibration discussion in this manual.

General Binary data loader

This is basically interface to Igor GBLoadWave function, customized for this application. Most parameters are copied from the GBLoadWave and so it is suggested user looks up this function in the manual for details.

Selecting GeneralBinary type on any window will pull up control screen. The configuration is same for all NIKA! The window can be safely dismissed (closed) and reopened by reselecting the GeneralBinary data type.



Top part:

Select number of bytes to skip – or check the “Use ASCII header terminator” if there is ASCII header with known ASCII separator before binary data. The known terminator (separator) in ASCII header- very useful, if the header bytes vary - input search term to look for. NOTE: Only first 40kbytes of data are searched, so the header cannot be longer than 40kbytes. For longer headers use the “Skip Bytes” option. Sometimes, you may need to skip few more bytes behind the terminator. Add those in the field below.

Image type part:

Size of image (rows x columns), type of data, for integer types byte order, and for floats type (IEEE or VAX), check the manual what this means... Save Header in Wave Note will add the skipped ASCII header to wave note, which gets propagated through the package into the final data... Very useful...

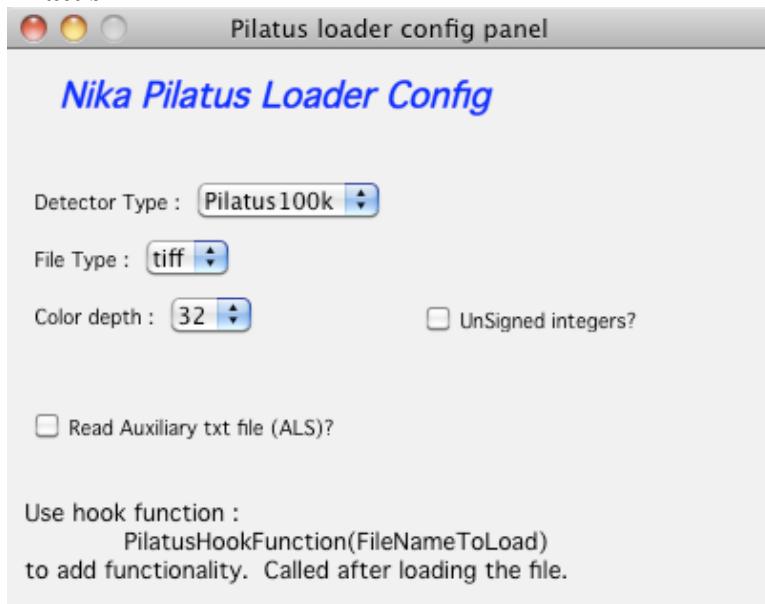
Other loaders with panels...

Some other loaders need user parameters to be included. There are two options, which are possible:

Panel. This is for example BSL/SAXS or BSL/WAXS format, where one file can contain multiple images. The panel enables selection of one or average of these images and provides some more information.

Function. Fuji image plate readers (BAS2000 or BAS2500) may be attached to hardware which saves data with varying “endiness”. A note is printed in history area which instructs user how to change endiness of the reader, if needed. This is set for the particular Igor experiment.

Pilatus



Can read: “tiff”, “edf”, “img”, and “tiff with floating numbers” (used for background subtracted images). Should manage 100k, 1M and 2M files, but tested on only 100k files (send files for testing, please). Can read Auxiliary txt files from ALS.

NOTE: You can design function, called “PilatusHookFunction(“FileNameToLoad”)” which is run after each image is loaded and which can customize behavior of the reader. For example, the function can read the wave note - which contains the header from the Pilatus file – and extract usable information. Or it can read auxiliary text file and extract information there. It is called with the name of file currently loaded as string for parameter.

Using XOP to load

Only when MarIP (CCP4) xop is present:

MarIP/xop this will use xop to load MarIP data As noted above, this ccp4.xop can be downloaded from igorexchange.com (and compiled by user) or is available on request from me. Please, send me e-mail with platform you need xop for.

Fit2D data files

Fit2D can be used as “loader” for number of file types, which it can recognize. This can be done for now **on Windows and Mac OSX version 10.4 and before**. Current Mac OSX version of Fit2D (14.101) cannot save usable tiff files on OSX 10.5 and unless I make changes in my code, Fit2D cannot be used as loader.

Fit2D method is somehow cumbersome, basically I create script file, which starts Fit2D, loads file in Fit2D and saves it immediately as tiff file. This tiff file is then read by Igor and immediately erased.

Note, that since I do not have these files to test, I cannot really verify function of all of these files... I have tested Mar image plate for now and it works... If you find one that does not work, please send me test file and I will fix the bug.

ADSC	ADSC Detector Format : Keyword-value header and binary data
Bruker	Bruker format : Bruker area detector frame data format
BSL	BSL format : Daresbury SAXS format, based on Hamburg format
Diffract	Compressed diffraction data : Compressed diffraction data
DIP2000	DIP-2000 (Mac science) : 2500*2500 Integer*2 special format
ESRF	ESRF Data format : ESRF binary, self describing format
Fit2D	Fit2D standard format: Self describing readable binary
BAS	FUJI BAS-2000 : Fuji image plate scanners (aslo BAS-1500)
GAS	GAS 2-D Detector (ESRF) : Raw format used on the beam-lines
HAMA	HAMAMATSU PHOTONICS : C4880 CCD detector format
IMGQ	IMAGEQUANT : Imagequant TIFF based format (molecular dynamics)
KLORA	KLORA : Simplified sub-set of "EDF" written by Jorg Klora
MarIP	MAR RESEARCH FORMAT : "image" format for on-line IP systems
MarPck	MAR-PCK FORMAT : Compressed old Mar format
MarIP	NEW MAR CODE : Same as MAR RESEARCH FORMAT
PDS	PDS FORMAT : Powder diffraction standard format file
PHOTOM	PHOTOMETRICS CCD FORMAT : X-ray image intensifier system
PMC	PMC Format : Photometrics Compressed XRII/CCD data
PRINC	PRINCETON CCD FORMAT :X-ray image intensifier system
RIGK	RIGAKU R-AXIS : Rigaku image plate scanner format

Invert 0,0 corner

As default Igor displays 0,0 of the image in the top left corner. This seems to be distressing for some users, so if checked, images will have 0,0 in the left bottom corner. Nothing else is changed, so the orientation of sectors WRT original image is preserved

and reduced data are the same as without this checkbox checked. Simply, the processing of this packes is independent of this checkbox, it is ONLY cosmetic...

Bottom controls



These controls have following functions:

“Ave & Display selected file” will average all selected files, which are selected in the list box, and display them as one figure. The program will just load and display the CCD images, including some processing (dezinging), if selected.

Note, if more than 1 image is selected, the images are first AVERAGED – that is intensities for each pixel as summed together and then divided by number of images.

“Convert selected files 1 at time” will load one after another the files selected in the list box and process them according to selection in the tabbed area.

“Ave & Convert selected files” will average all selected files in the list box and process them according to selection in the tabbed area.

Note, if more than 1 image is selected, the images are first AVERAGED – that is intensities for each pixel as summed together and then divided by number of images.

“Save displayed image” will save displayed image into tiff file for future use. This is method, how to for example average number of images and save them for single empty or blank image.

“Skip Bad files” Enables to skip automatically processing of files, which have too low intensity (SetVariable control with limiting value appears when selected). Used to skip files which were accidentally NOT exposed in case of failing shutters or other issues.

“Display RAW data” will display in the image right of the panel the UNCORRECTED data file as loaded in. Values for the pixels are raw counts from the detector.

“Display Processed” will display in the image right of the panel the fully CORRECTED and CALIBRATED data. The values for the pixels should be directly absolute intensity in this case. This choice is not available, if image was loaded through using “**Ave & Display sel. Files(s)**”. In this case no processing of the image was done. Use button **“Convert sel. Files 1 at time”** or the other buttons.... Just remember, that only the last image is available for display.

“Display beam center” will add circles in the image showing where beam center is set

“Display sectors/Lines” will add lines showing sectors or lines, which are selected for data analysis (if any)

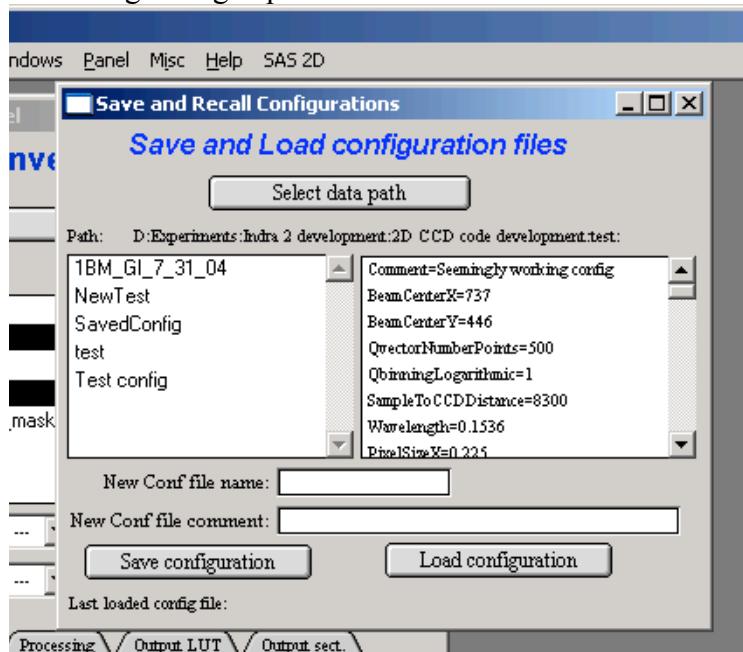
“Log Int display” will switch displayed image into log (intensity) or linear (Intensity).

“Image with Q axes” Appends Qx/Qy (or Qz/Qy) axes to displayed image. Note, when unchecked, it has to recreate the image, since these Q axes cannot be removed any other way.

“Image w/ Q axes with grid” Appends Qx/Qy (or Qz/Qy) axes to displayed image – with grid lines. Note, when unchecked, it has to recreate the image, since these Q axes cannot be removed any other way.

“Save & recall config” will allow user to save current settings – or load saved settings– in the tabbed area as “configuration”. User can save the configuration file in any place on the hard drive he/she wishes – ideally with the data!

Following dialog is presented:



Explanation of controls:

“Select data path” – select path to folder with the configuration files

Path is displayed below

Left window – shows the names of configuration files found in this location

Right window – shows content (note, first line is user comment) in the selected configuration file. If new configuration file is selected, content of the first one is shown.

New Conf file name – User input for new configuration file to be created

New Conf file comment – place to store info about what this conf file contains!

Save configuration button – save current setting of the tabbed area

Load configuration button – load stuff from config file into the program. Note, your current configuration will be overwritten and there is no way back, if you did not save your configuration...

Note: names of dark field, empty beam, mask, and pix2D sensitivity are not saved and are not reloaded, when configuration is reloaded. This would really be very complicated...

Geometry/processing controls

Note, that if images are averaged, they are first averaged during loading, and then – during processing to create lineouts / square matrix are corrected as described below. Therefore all parameters here related to single (if possibly averaged) image!

These are controls in the tabbed area.

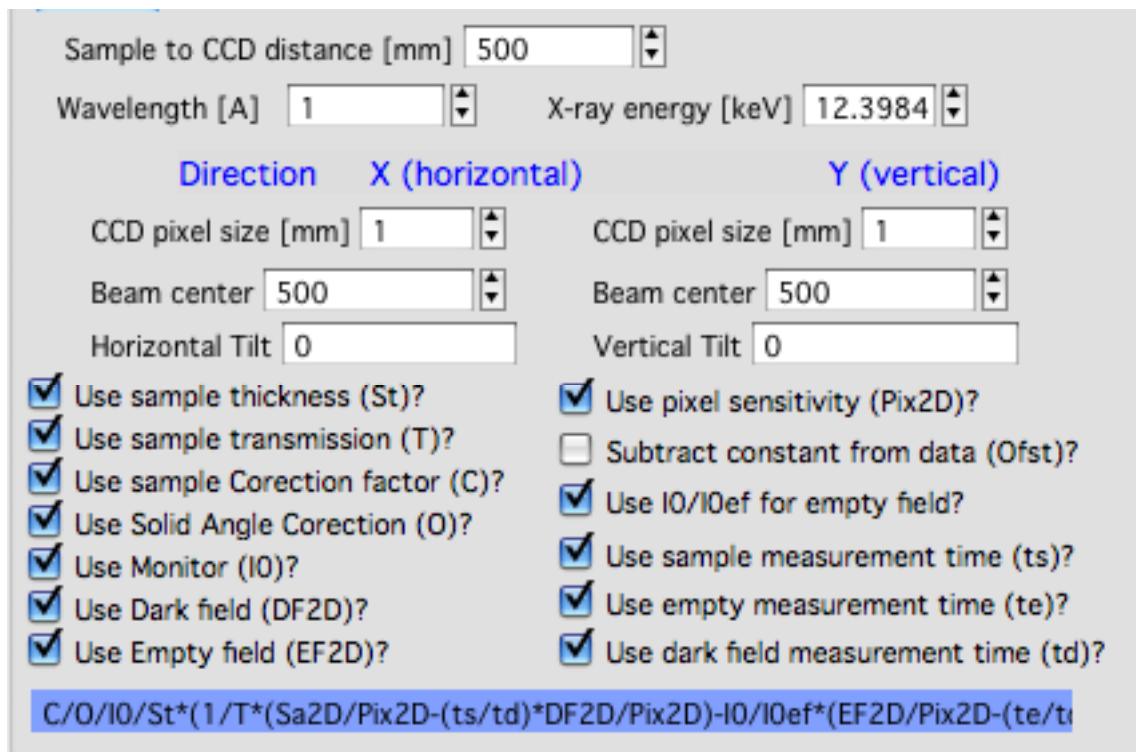
We will now go through each tab separately

Main

Here are some very clear parameters, related to SAXS camera geometry:

Sample to CCD distance in millimeters, Wavelength/Xray energy (these windows are linked), CCD image pixel size in mm (in X and Y directions). Note, X direction is horizontally, Y direction vertically. And Beam center position. Note, one can display beam center (to check it) in the graph by checkbox below the tab area.

And further there is pile of checkboxes, which describe method how to calibrate the data. Note, that formula used for calibration appears below to avoid any misunderstanding of the method. Select method needed for processing – and following tabs will have the appropriate controls available.



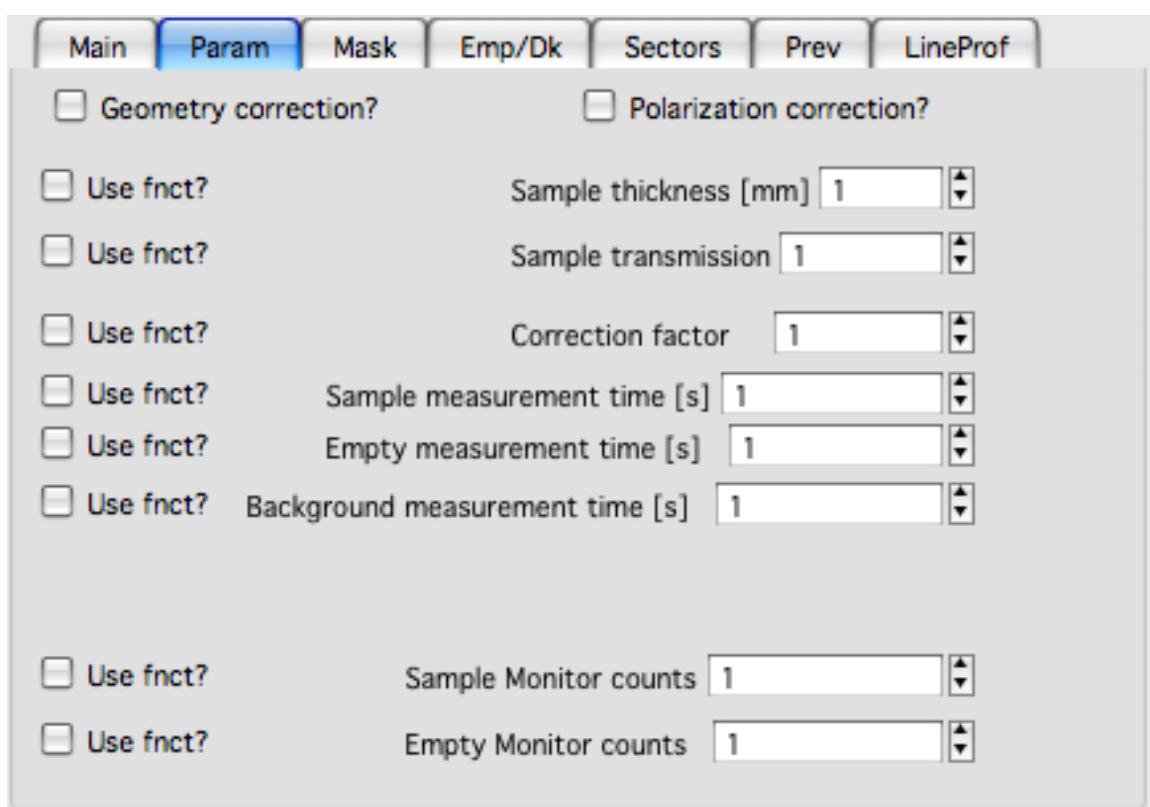
Note, that “**Use of Dark field**” and “**Subtract constant from Data**” cannot be used at the same time (they are effectively the same type correction)...

Note, only the appropriate controls will appear, so seeing all of these at the same time should be VERY unusual...

Comment for Use of **Solid Angle Correction**: When selected, the data are divided by solid angle of the central pixel (same value for all pixels). To correct for change in pixel solid angle as function of scattering angle, use Geometrical correction. Most of the time we do not bother with this option – if you use secondary calibration standard (like Glassy carbon or water) solid angle correction is included in the Calibration constant. If you do not use calibration and have relative data, you do not care also. The real need for this option is when you use data obtained in different sample to detector distances and want to combine the data together. Then this is necessary option.

Just remember, if you have obtained calibration constant, it is linked with the choice of the Solid angle correction.

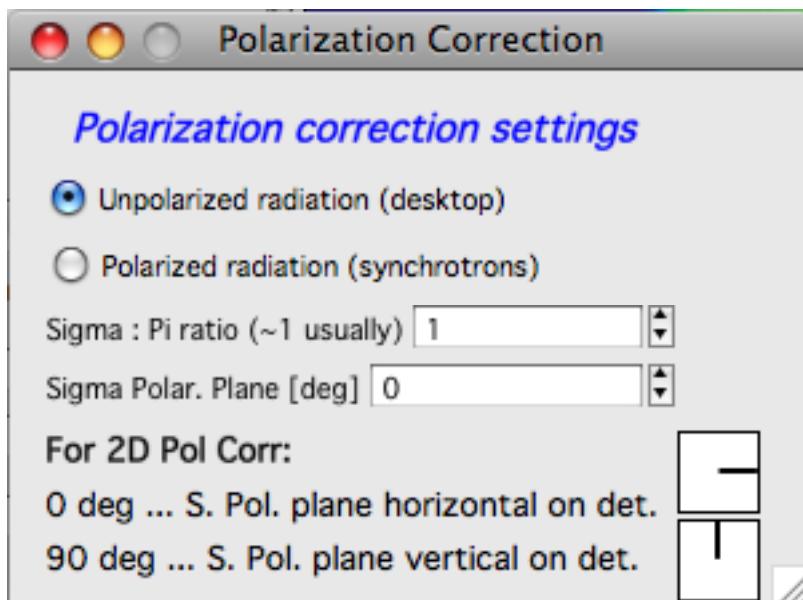
Param



Here are standard controls (self explaining I hope):

“Geometry correction” – fixes the **VARIATION** of solid angle projection of the pixels on planar CCD detector. Mostly negligible for SAXS data... Just for completeness, this divides the intensity at each pixel by $(\cos(2\Theta))^3$. And for those, who do not understand this formula, it took me maybe 3 weeks to check it (I stole it from NIST data reduction). Very simplified, one $\cos(2\theta)$ corrects for change of pixel radial direction as function of scattering angle, second $\cos(2\theta)$ comes from change in distance between sample and detector as function of scattering angle in radial direction, third $\cos(2\theta)$ comes from the same correction for tangential direction. Tangential size of pixel does not change as function of scattering angle.

“Polarization Correction” – Correction for either unpolarized radiation (desktop instruments with tube sources for example) or for Linearly polarized X-ray sources (synchrotrons). Opens up a new panel.



For unpolarized radiation use “Unpolarized radiation”. This is applicable ONLY to unpolarized radiation, the intensity data are corrected by formula:

$$\text{Intensity_corrected} = \text{Intensity_measured} / (0.5 * (1 + \cos((2\theta)))^2)$$

For linearly polarized radiation use “Polarization radiation”, see separate chapter on Polarization correction little bit further in this manual.

By the way, for small-angle scattering each of these corrections is negligible.

Sample thickness in millimeters, **transmission** as fraction.

Correction factor is for secondary calibration factor.

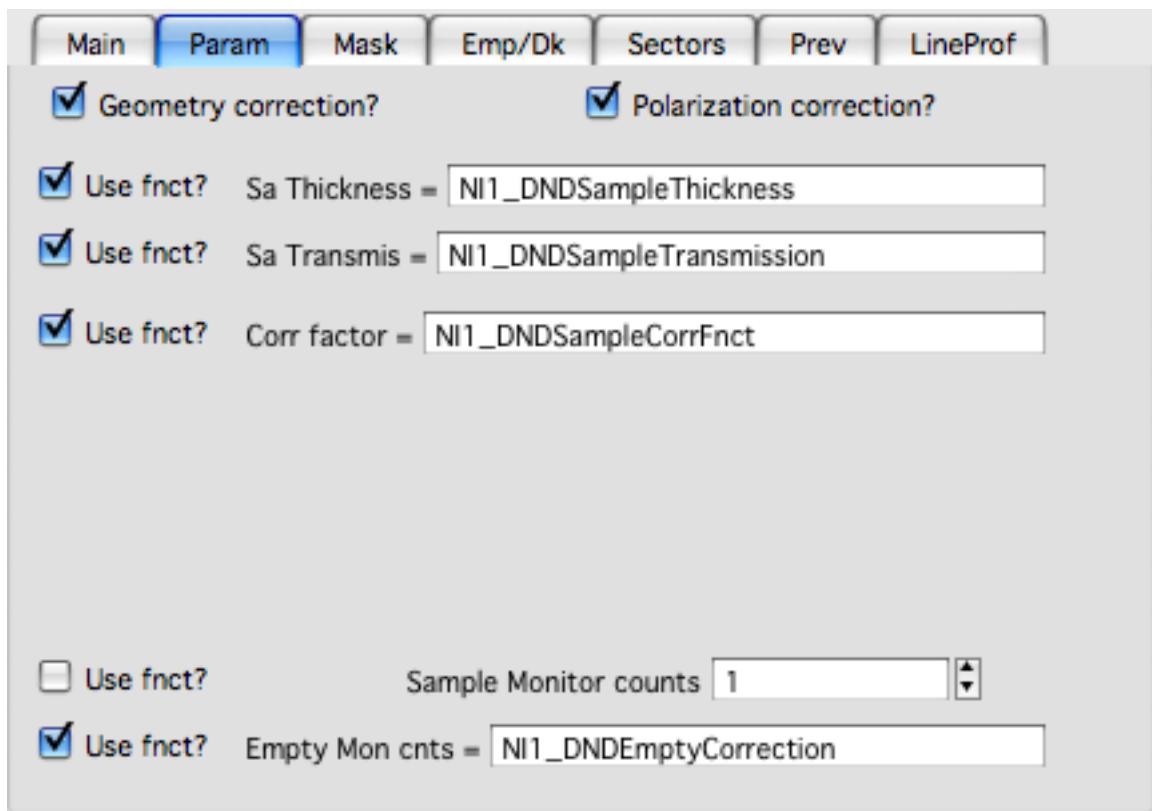
Measurements times in seconds, for each image.

Sometime one wants to use measurement time to correct images collected at different time exposures. While not suggested, it is possible to do here. I strongly discourage this. Monitor counts allow scaling data by using monitor on incoming intensity.

“**Fixed offset for CCD images**” this is single value to be **subtracted** from each pixel of image to be processed.

“**Monitor counts**” use monitor counts to scale images (Sample/Empty)... This makes no sense for dark field...

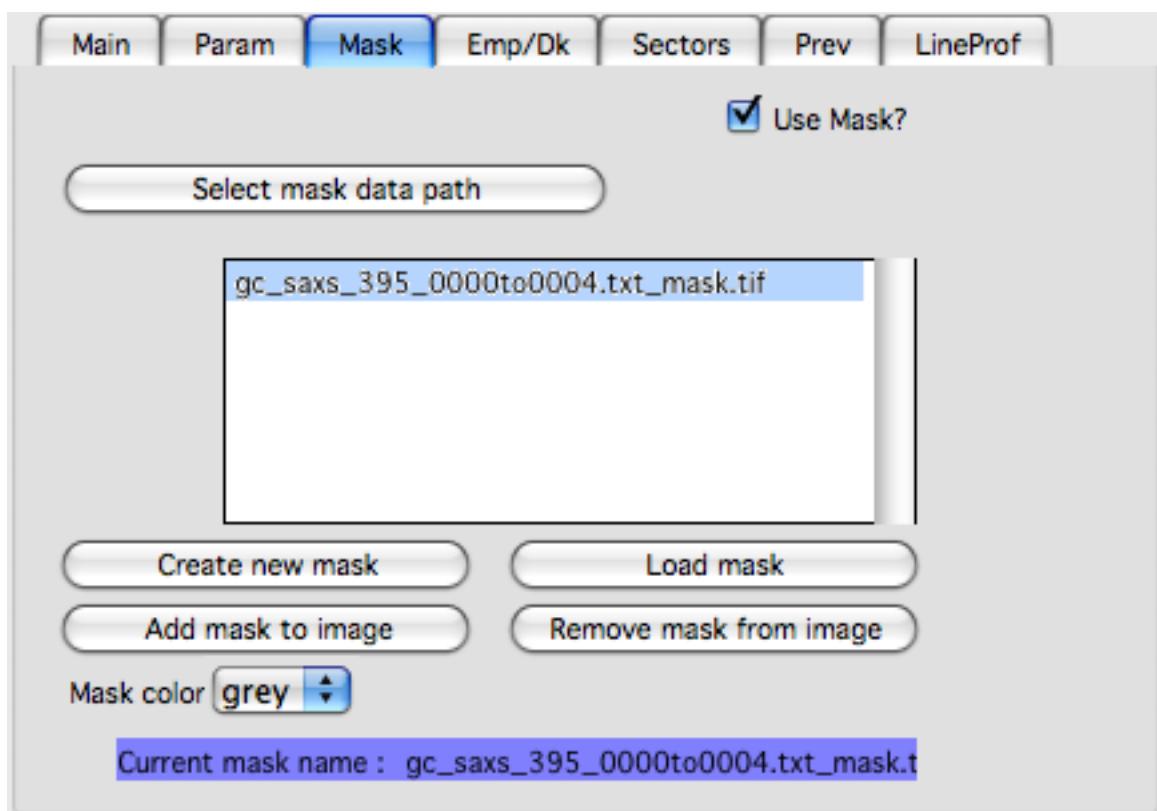
Each of these values can be inserted by user as number, or using function:



These function need to be “look up” functions, which are called with image name as parameter (FunctionName(“ImageName”)) and must return single real number. The real use is to provide automatic look up of parameters from some records written by instrument. Above example is from included special support for DND CAT instrument.

Let me point out once more here, that using some of these corrections together makes no sense... Choose wisely.

Mask



First checkbox, if Mask should be used (did not fit on the front tab...), button to select path to files with masks. Note, mask files created by Nika are always tiff files, with name in following manner: UserName_mask.tif

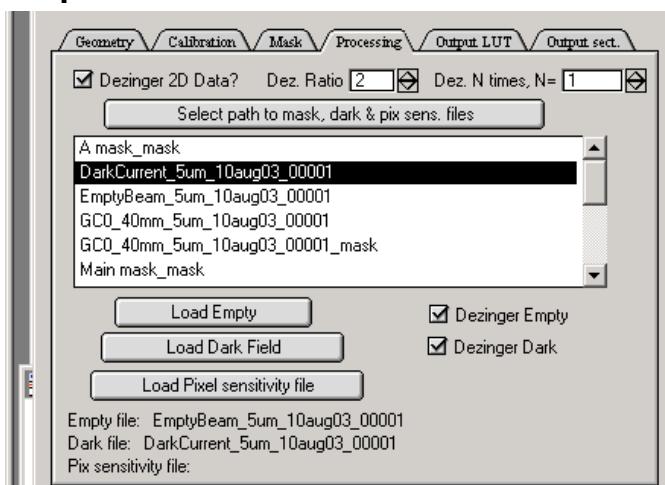
Following are function of the buttons:

1. Create New mask – calls tool to create mask (see later in the manual)
2. Load mask – load file selected above in the list box as mask
3. Add mask to image – adds mask into the 2D image from the image
4. Remove mask from image – removes the mask from the image

Mask color – allows to change color (red, green, blue, black) of the displayed mask...

Current mask name – shows name of last loaded mask file

Emp/Dark



Here are controls for Empty/Dark field/pixel sensitivity (aka flood) images.

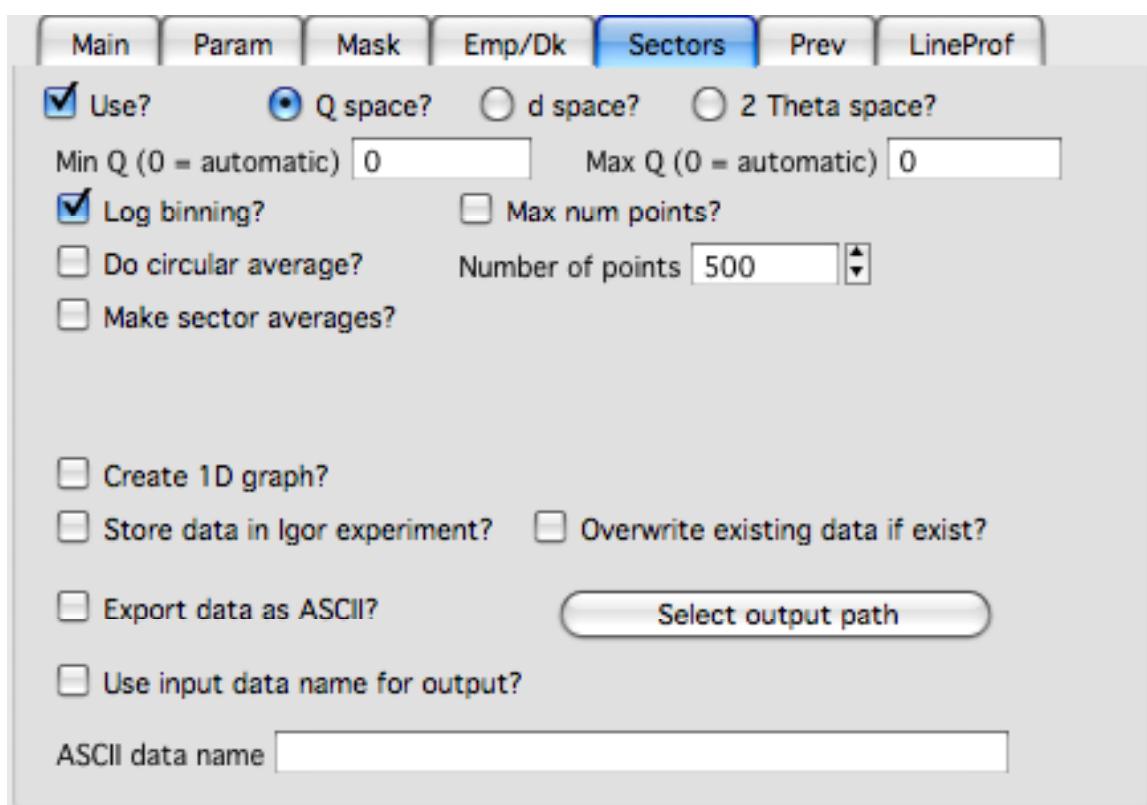
Note, Data, Empty, and Dark field images can be “dezingered” during loading. In this procedure each point is compared to surrounding pixels and if it is significantly larger (that is the dezinger ratio, if 2 then if the pixel is 2x larger than average of surrounding pixels) it is replaced with the average of the surrounding pixels. This is to remove spurious very high intensity points, which occur on some instruments.

It is possible to dezinger each image multiple times, in case the “zingers” are larger than single pixel.

Button “Select path to mask, dark & pix sens, files” Selects path to data with the Empty, Dark field etc. I believe the files need to be the same type as data file (I need to check this).

Further buttons load the Empty/Dark/Pixel sensitivity, allow Dezingering of these (same method as the sample dezingering as selected above). And at the bottom are listed the file names of the files loaded...

Sectors



This tab controls how data are processed when method using “reverse Lookup tables” is used. This is the more suggested method for regular data processing. In this method Nika creates first lookup table for each sector defined and then can process much faster subsequent data files with the same geometry...

Controls:

Q space/d space/ 2 theta space – Output as function of Q, d, or 2 theta...

Min/Max (Q, d, 1 theta) range of evaluated Q, d, 2 theta. Set to 0 for automatic – automatic means, that the min/max is set for first q/d/2 theta which has non zero intensity

“Log binning” – check yes if Q/d/2 theta binning should be in logarithmic.

“Number of points” – number of points in Q/d/2 theta which should be created.

Do circular average – self-explanatory.

Make sector averages – do sector averages. Controls below control orientation and sizes of sectors. To see how the sectors are places, check the checkbox at the bottom of the control panel.

Create 1D graph – if checked, 1d graph with output is created (if necessary) and data added. Note, the graph may be crowded very fast, since data are added, and added...

Store data in Igor experiment – keep data (as qrs triplets) in current Igor experiment. Overwrite existing data if exist – if data with the same name exist, overwrite without asking. Otherwise, you will be asked.

Export data – export ASCII data

Select output path – select where data are to be placed.

Use input data name for output – automatically name 1D data (with sector information added as DataName_Angle_width) by input data name.

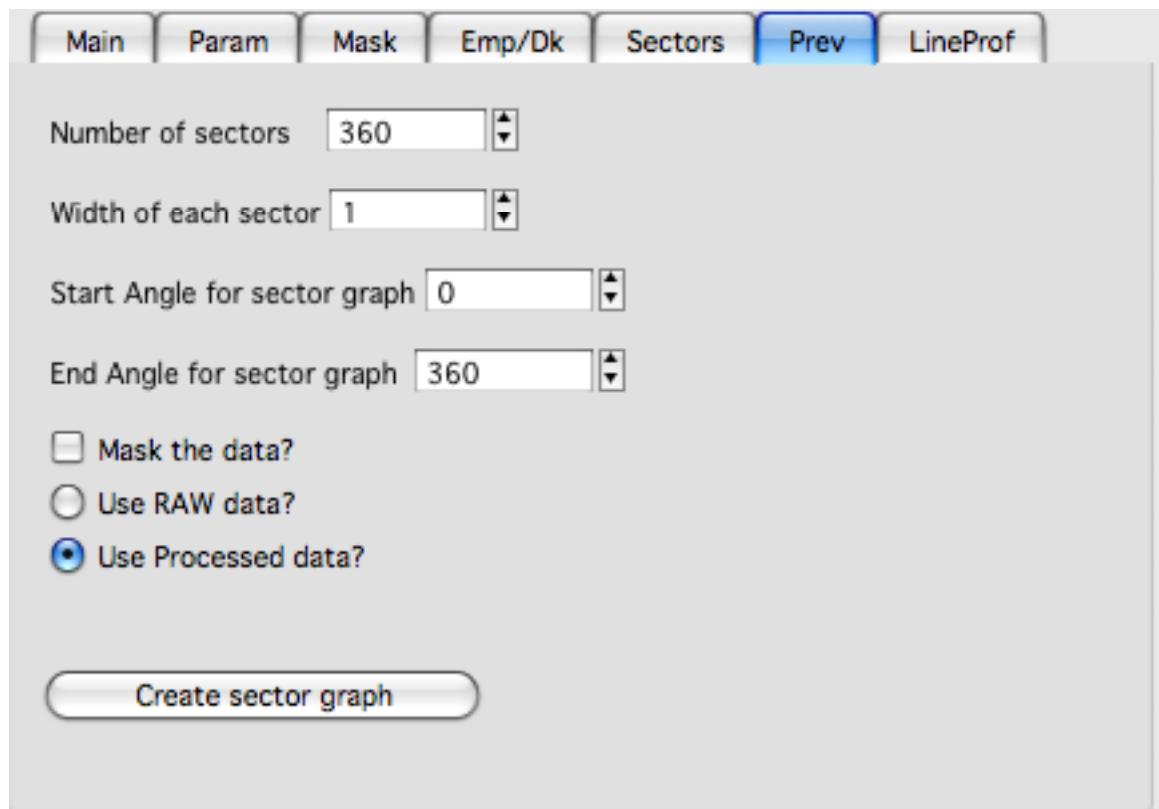
ASCII data name – if the above is not selected, this is place to place name for output file. Note, if there is nothing available for the code as sample name, it will ask for some...

Prev

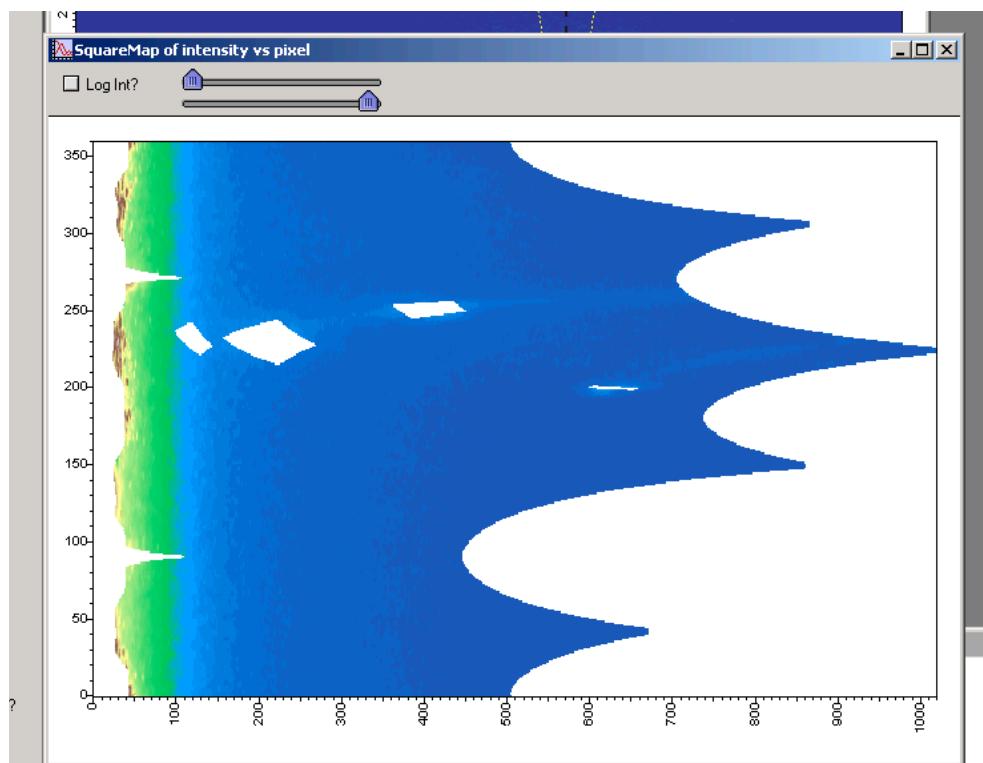
This means: "Preview" which is the intended use of this tool...

First:

This tool can use the calibrated data set (as well as RAW data set, depending on checkbox setting) so same calibration procedure is used as for the other processing. This tool is, however, less precise and does NOT produce useable errors. Be warned, this tool is meant as quick look on the data in different directions and not for final data processing...

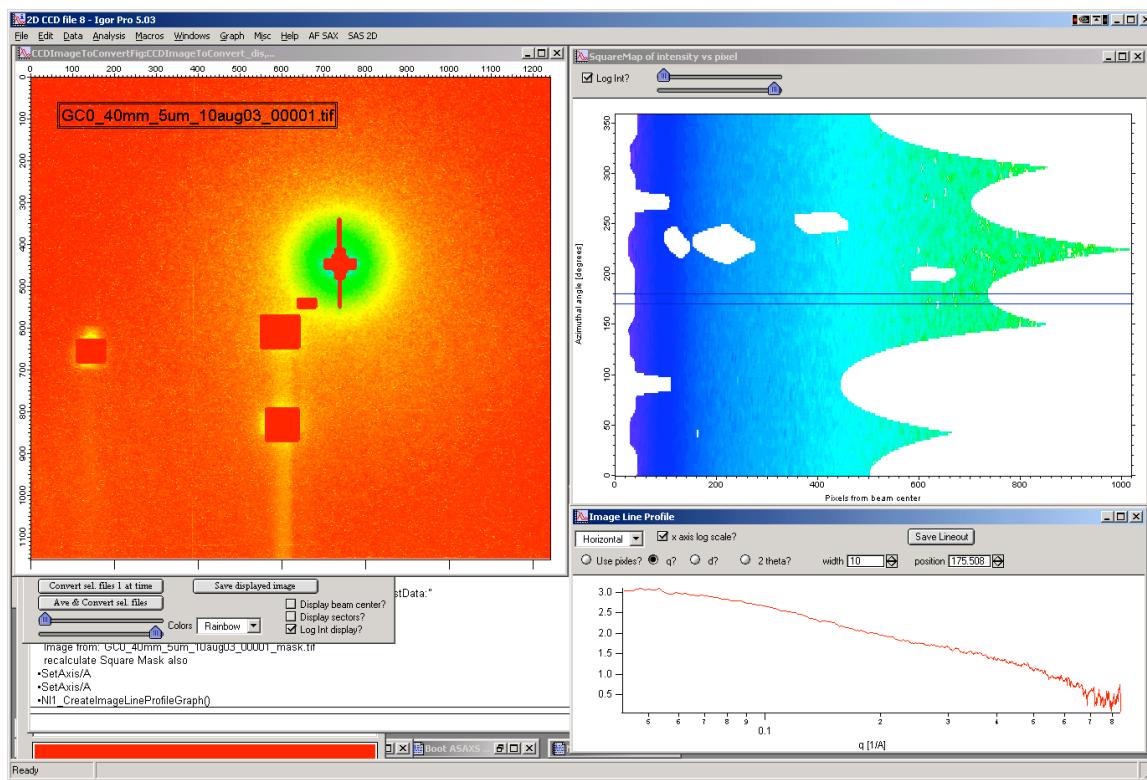


This is second method to get output data from this tool. In here, one can produce rectangular graph:



On vertical axis is angle from 0 degrees axis (horizontally right from the beam center) and on horizontal axis is pixels distance from beam center. This is effectively set of lineouts in all azimuthal angles. It should be noted, that the code works very well for relatively small widths – may be up to 5 degrees, then the code becomes less precise, so keep angles small. Suggested is 1 -5 degrees.

These data then can be processed further by use of “Image line profile” tool. This tool for now has it’s own “mindset” and does not properly update always. The dependencies are quite complex. If it does not update, close the tool and reopen...



The “SquareMap of Intensity vs pixel” graph on the top right above shows the intensity in linear/log (checkbox left top corner) as function of pixel (bottom axis) and azimuthal angle (left axis). The lineout plot at the right bottom shows the intensity from this plot (note, the log/lin scaling in the image translates here!) as function of pixels/q/d/2 theta. Note, that this produces “natural” binning with every step in pixel is assigned single q/d/2theta position.

Note, the controls:

Number of sectors

Width of each sector - it is possible to have width such, that bins overlap, touch or do not touch... Default here is to have them touching.

Start Angle (0 = right horizontally from beam center)

End angle (wrt to start angle, most likely 360 degrees, or 180 degrees for only top half).

Mask data this tool does not mask, unless selected here...

Note, that by selecting larger width here, one can get very good and reliable sector average and manually move this average through the different azimuthal angles. Very useful, when hunting for particular azimuthal orientation...

Use RAW data if selected unprocessed image is used.

Use Processed data if selected processed image is used, available ONLY if the last image was loaded using one of the “Convert...” buttons, unavailable if the last image was loaded using “Ave & display sel. files(s). If the data were loaded using “Ave & display...” button, processed data do not exist.

Controls on Lineout tool:

Orientation of line profile (Horizontal/vertical)

X axis linear/log scale

Use: pixels/q/d/2 theta

Width and position

Save lineout – this saves “qrs” data in SAS folder in current Igor experiment. Suggested folder/data name is offered through dialog and user can modify as needed. Note, that errors are simple $\text{sqrt}(\text{intensity})$ – another words, these errors are not very useful.

LineProf

This tool calculates Intensity profile along curve on the detector. It uses different method than **Sectors** tool. Therefore, there are some important differences in how to use this tool...

The differences:

“**Sectors**” use inverse lookup method and can be set to create multiple different sectors on one image at once. Since this tool caches the lookup tables, it is slower first time, but much faster on subsequent images. This tool can be used ONLY by setting the data reduction parameters and then using buttons “**Convert...**”. You cannot manually evaluate any sector and no preview is provided. This tool causes high memory sizes of the Igor experiments with Nika package – the lookup tables are large. But it is fast for what it does.

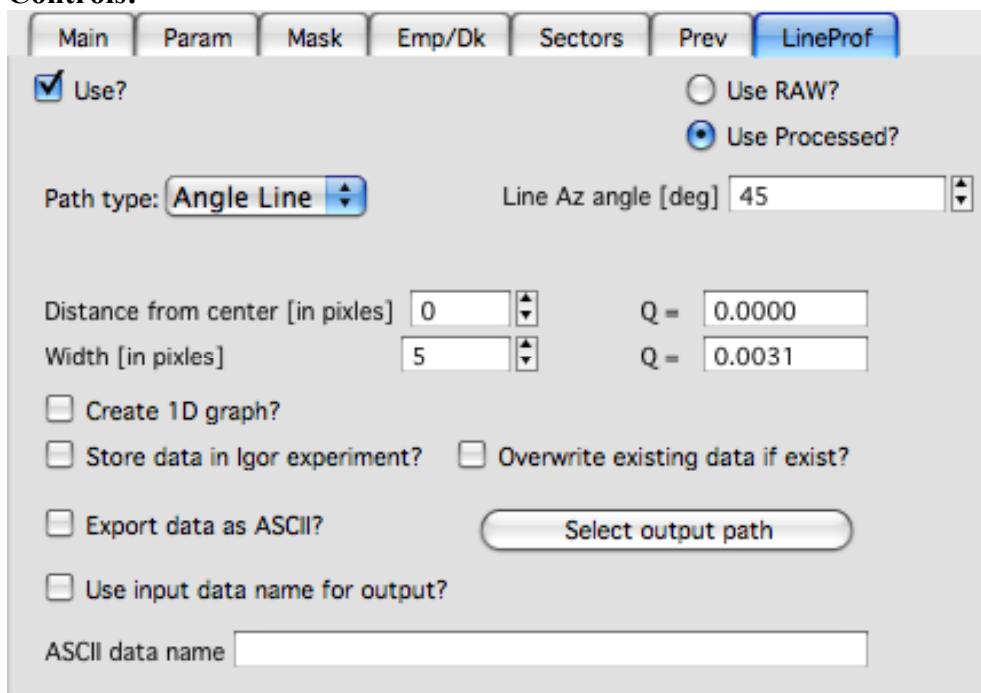
And you can setup multiple sectors to be evaluated at once.

“**LineProf**” uses built in Igor Line Profile tool. It can be set ONLY to process one line profile at a time. This tool does not cache anything, so it takes the same time to process for each image. However, it is relative fast and can be used manually on Converted image. So, there are two methods to use it:

- Set one line profile parameters, choose how to save data and push one of buttons “**Convert..**”
- Do not set any conversion parameters, but use one of the buttons “**Convert..**”, set the **LineProf** tool to use Processed data and then set parameters for the

You can only set one line profile at a time, unless you manually create multiple profiles on each converted image.

Controls:



New controls here:

“**Use?**” – switches on this tool.

“**Use Raw?**” – and “**Use Processed?**” – choices which image the tool will be used on.

User Processed is not available if the last data set was loaded using “**Ave & Display..**” button (no Processed data are created in this case). NOTE: if you hit any button “**Convert..**” and this tool is enabled, it is set to “**Use Processed**” automatically.

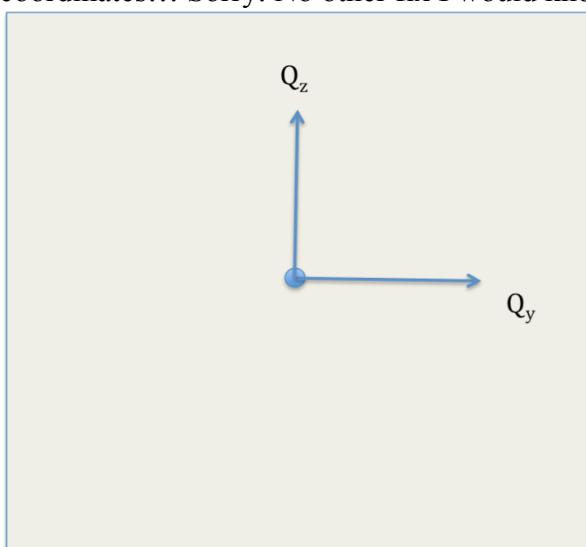
“**Distance from Center [in pixels]**” – user control to move the object to specific q . The q where the data will be calculated is displayed next to this control and is the appropriate q (q_y or q_z) for give shape. See Ellipse definition for specific there. NOTE: you must control the pixel position. Positive direction is to the right of the beam center (horizontally) or up from the beam center (vertically). Lines are drawn to help user figure this out.

“**Width [in pixels]**” – width of the profile (minimum used one is 1 even if 0 is set by user) in pixels. This is the control to use to change how wide stripe is averaged. Next to it is control which shows this in q units. NOTE: the q width is calculated simply by subtracting Q values for the sides of the stripe. Intensity is averaged at each point perpendicularly to the direction of the line (curve). If more than 1 pixel is used for averaging, standard deviation of average is provided as error, if only 1 pixel is used, square root is used (which may be seriously WRONG)... You were warned.

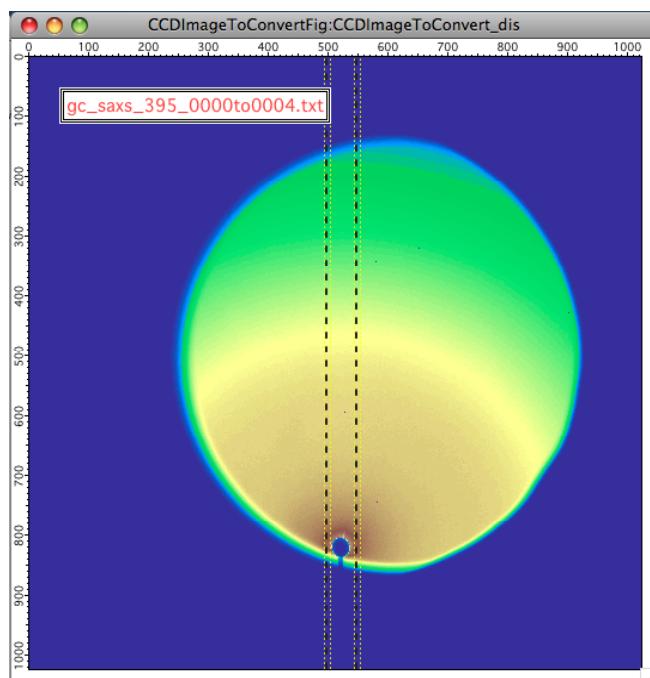
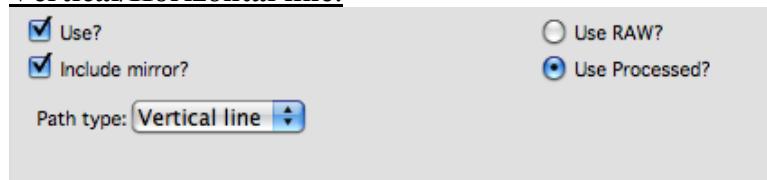
This tool calculate intensity, intensity uncertainty and q , q_y , and q_z values. If one of GI profiles is used, it will calculate q , q_y , q_z , and q_x values. See below.

IMPORTANT:

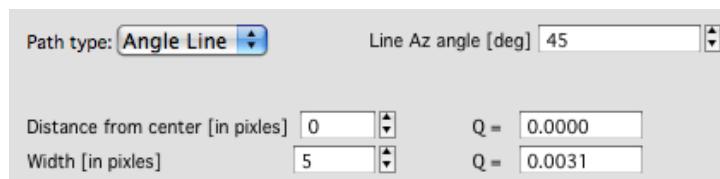
Of course, GISAXS community had to adopt different definition of Q_x , Q_y , and Q_z than I did years ago, and therefore, this tool uses somehow different definitions than rest of Nika. So the horizontal direction (x-direction for Nika) is the Q_y direction. Vertical direction on the detector is “y” direction for Nika, but it is direction of Q_z . Please, keep this in mind... For those adventurous souls, who actually read my code, keep in mind at some point the code switches on your the x-y image coordinates to y-z-(x) GISAXS coordinates... Sorry. No other fix I would know about.

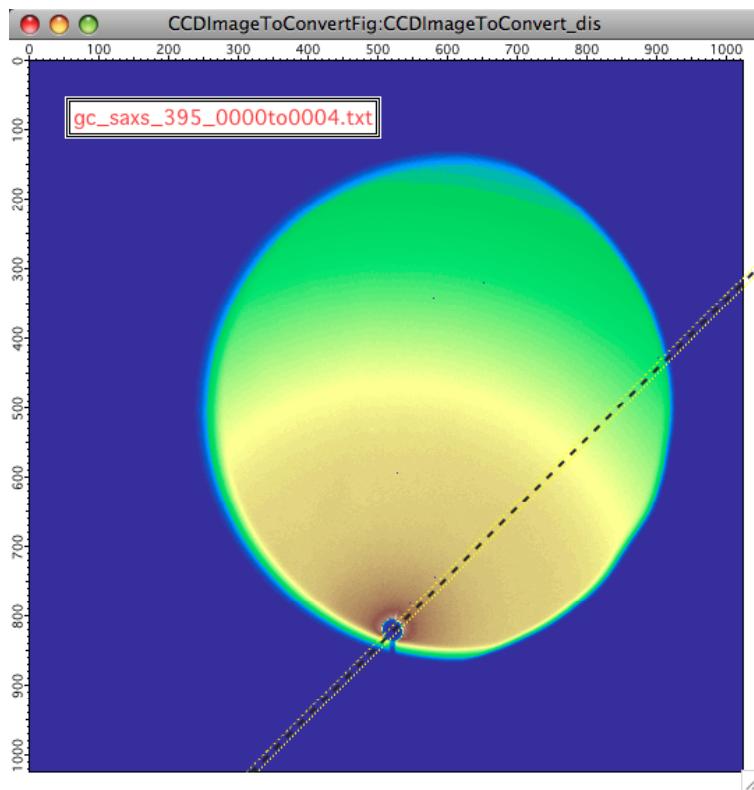


For now these are the available profiles:

Vertical/Horizontal line:

There is one more control available – “**include mirror**” (above the popup). If this is selected, mirror line over the beam center is included in calculations, see above. This is line profile for transmission geometry.

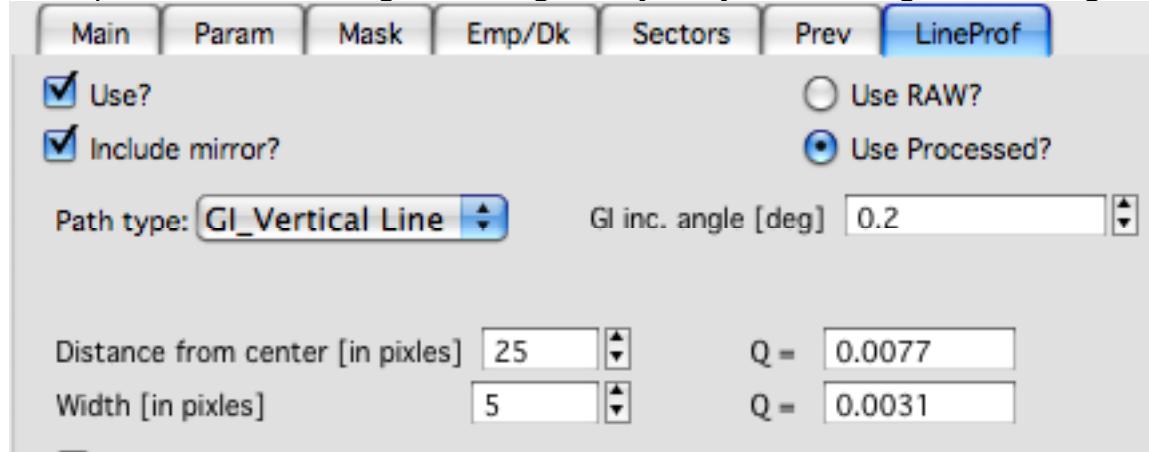
Angle line:



This is also for transmission geometry.

GI Vertical line & GI Horizontal line

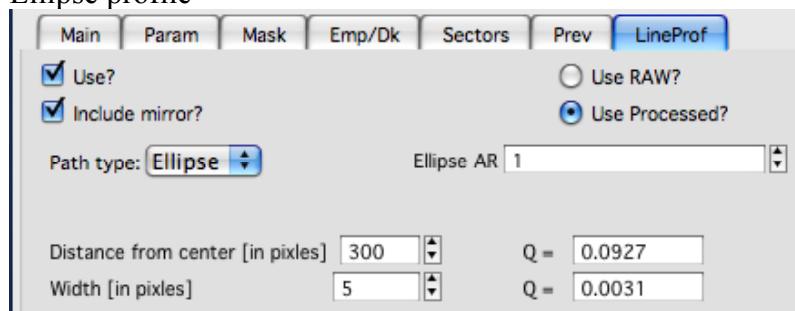
These profiles are for Grazing incidence geometry. They need Grazing incidence angle:



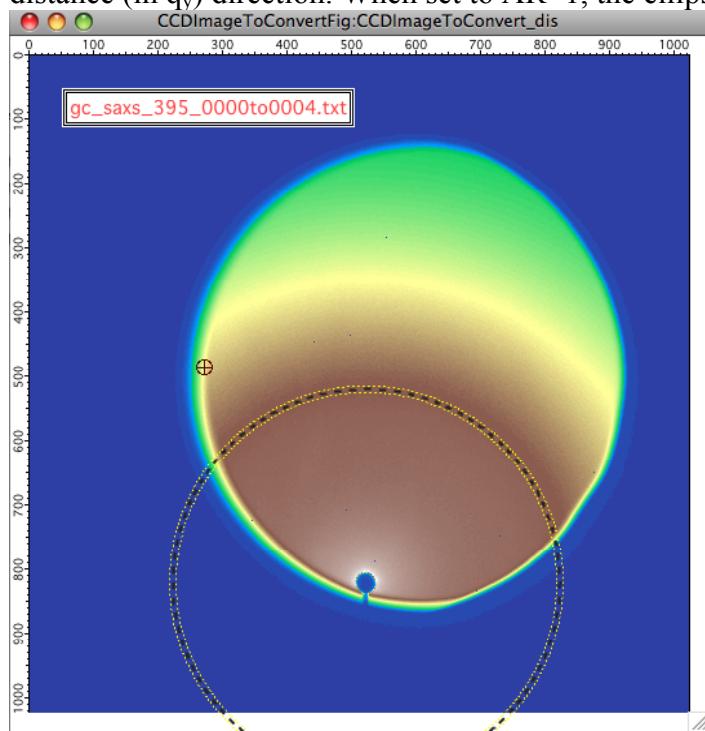
Both can include mirror image line across the beam center.

Note, that the position is defined in pixels as before, but the Q values are corrected according to the Grazing incidence geometry corrections, see Gilles Renaud, Remi Lazzari, and Frederic Leroy, Probing surface and interface morphology with GISAXS, Surface Science Reports 64(2009) 255-380, formula (1).

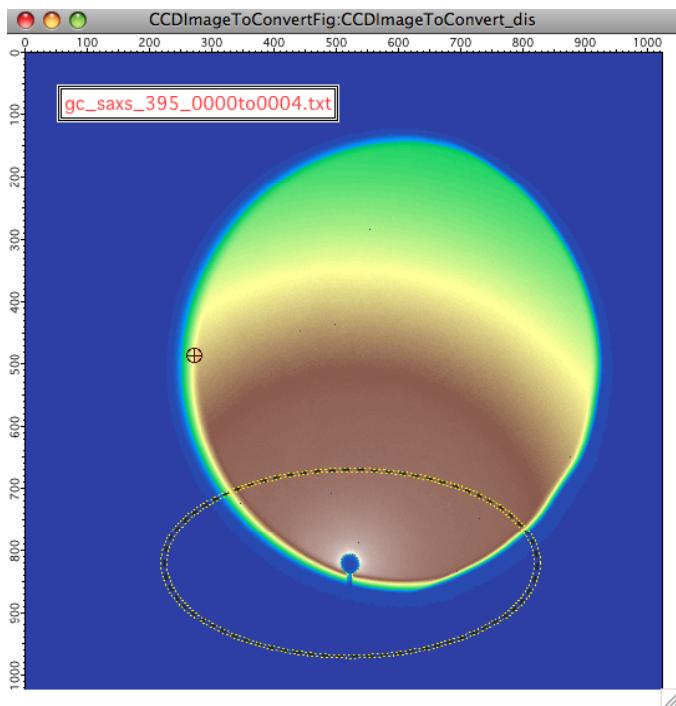
Ellipse profile



Note, that there is aspect ratio control here and the Distance from center here is horizontal distance (in q_y) direction. When set to AR=1, the ellipse becomes circle.



For AR<1, the ellipse is this way:



Note, that this tool has one major problem – it is practically impossible to display the data in any sensible way. Neither q_x , q_y , or q_z makes any sense here. In some way one needs to get angle of the intensity position. At this moment I do not produce such data within Nika. User can produce them by himself (the step is 0.25 degree, starting from 0 degrees azimuthal angle on the detector [note: I hope, I got turned around so many times, that this requires some data to test on]).

The other option is to use q_y and q_z to generate this angle. **If anyone will ever use this tool, please, contact me and tell me, how you want to use it and I will modify the tool to suit needs of users.**

Finally : More shapes.... I can imagine broadening capabilities of this tool with other shapes. If you have such need, talk with me and I'll add line profile shape for your needs.

Controls for saving data are the same (really, these are the same controls, showing on second screen also) as in the **Sectors** tab:

Create 1D graph – if checked, 1d graph with output is created (if necessary) and data added. Note, the graph may be crowded very fast, since data are added, and added...

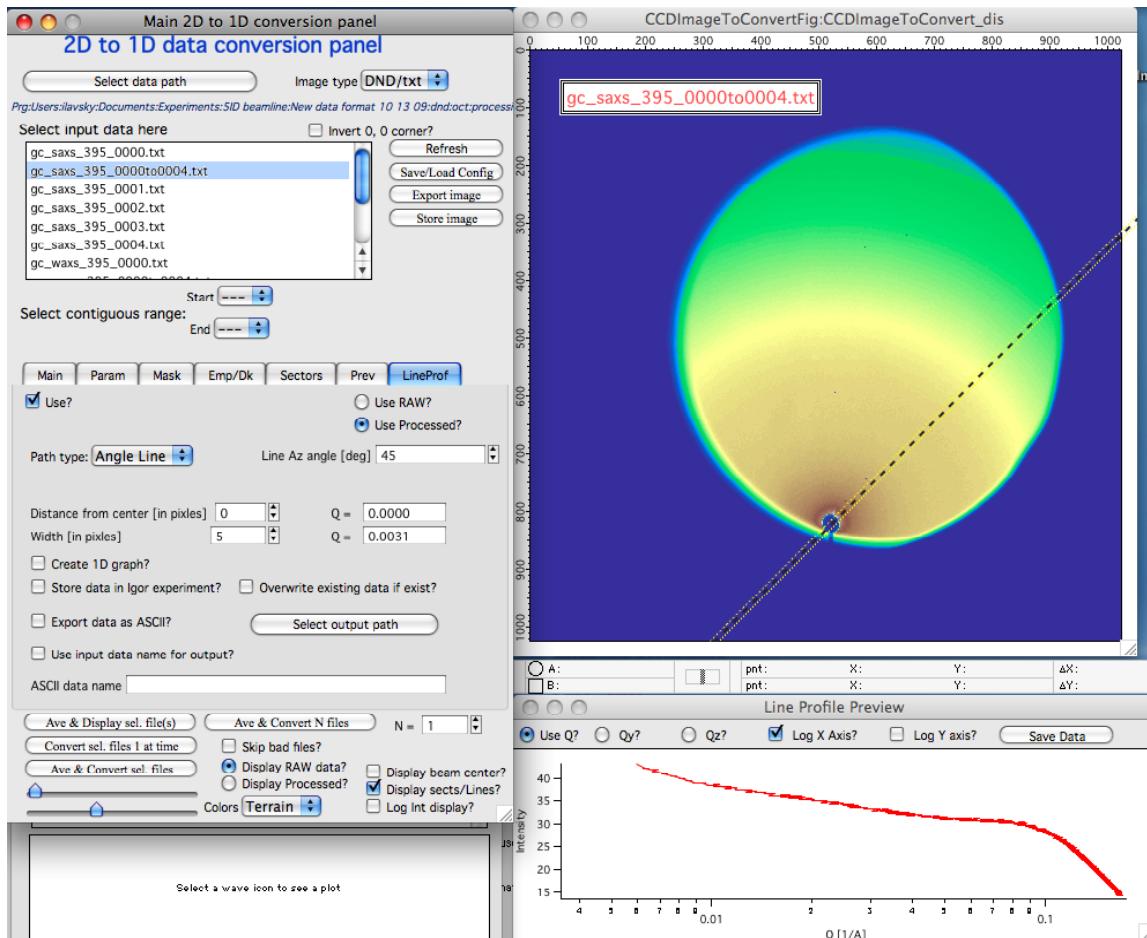
Store data in Igor experiment – keep data (as qrs triplets) in current Igor experiment. Overwrite existing data if exist – if data with the same name exist, overwrite without asking. Otherwise, you will be asked.

Export data – export ASCII data

Select output path – select where data are to be placed.

Use input data name for output – automatically name 1D data (with sector information added as DataName_Angle_width) by input data name.

ASCII data name – if the above is not selected, this is place to place name for output file. Note, if there is nothing available for the code as sample name, it will ask for some...



Note, that the LineProf tool uses another “graph” window (“Line Profile Preview”) under the main image. This window contains some controls that are very useful.

The data are automatically updated as the parameters for the profile are changed. This gives user live update (but can take time, if it takes too much time for anyone, let me know and I'll add controls to avoid the updates “live”).

User can display the data as function of q , q_y or q_z and on lin-lin, log-lin, lin-log and log-log scales. Note, that negative values cannot be displayed on log scale, so since q values for lower part of detector (below beam center) are defined as negative, you may not see

them if you choose log scale. Also the q values look sometimes really weird, but generally they should be correct. If there are any issues with definitions of negative directions, let me know.

User can also save the data displayed in this window, which enables user to create multiple line profiles from existing image – this is manual method. NOTE that save parameters are taken from the setting of the controls for this purpose in the tab in the main panel (“Create 1D graph”, “Store data in Igor experiment”...). If you choose “Overwrite existing data” and do not change the name, you may get in troubles.

When data are being saved some cryptic description to indicate what profile was used and which q was used will be attached to the name used. More full description is attached to wave note.

For example for GI_Verical line in my test case, this was the name:

gc_saxs_395_GI_VLp_0.0077
 “gc_saxs_395”.... Part of the name of used image
 GI_VLp..... GI_Verical Line
 0.0077 q_y value at which the data were calculated.

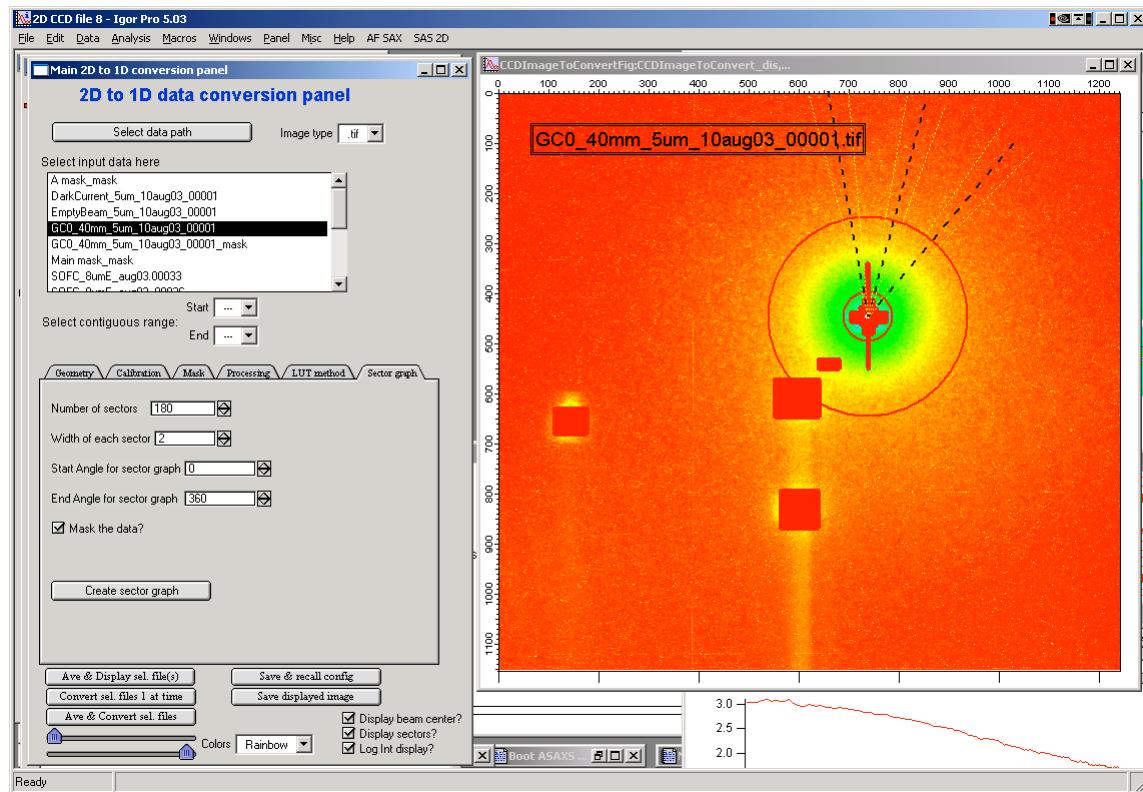
Exported data are Int, error, Q, qx, qy, qz columns with header and column names

Saved data in Igor are

r_gc_saxs_395_GI_VLp_0.0077	intensity
q_gc_saxs_395_GI_VLp_0.0077	q
s_gc_saxs_395_GI_VLp_0.0077	error
qy_gc_saxs_395_GI_VLp_0.0077	qy
qz_gc_saxs_395_GI_VLp_0.0077	qz
qx_gc_saxs_395_GI_VLp_0.0077	qx (generated ONLY if GI... profile is used)

Note: next release of Irena package will have capabilities to use not only qrs data , but also qxrs, qyrs, and qzrs data.

Main 2D image



This is image of the 2D image. Notice, this is in log scale, mask is displayed, , sectors and beam center are also added. User can use sliders and Colors controls on the main panel to change image to see what is necessary:

Note, that name of loaded file is in the image, if this was average from many images, all of the names would be listed.

Polarization correction

Two types are available.

Unpolarized radiation

$$Int_{cor} = \frac{Int}{0.5(1 + \cos(2\theta)^2)}$$

This is generally accepted formula.

Linearly polarized radiation

This is polarization correction for linearly polarized radiation, such as produced by double-crystal monochromators on synchrotrons.

There are two polarization orientations, sigma (linear part) and pi. Most synchrotrons will be linearly sigma polarized, with sigma fraction may be 0.99 or so. Depending on the way the detector is read, the sigma polarization plane may be horizontal or vertical. The panel enables setting the sigma polarization plane orientation.

The final formula is:

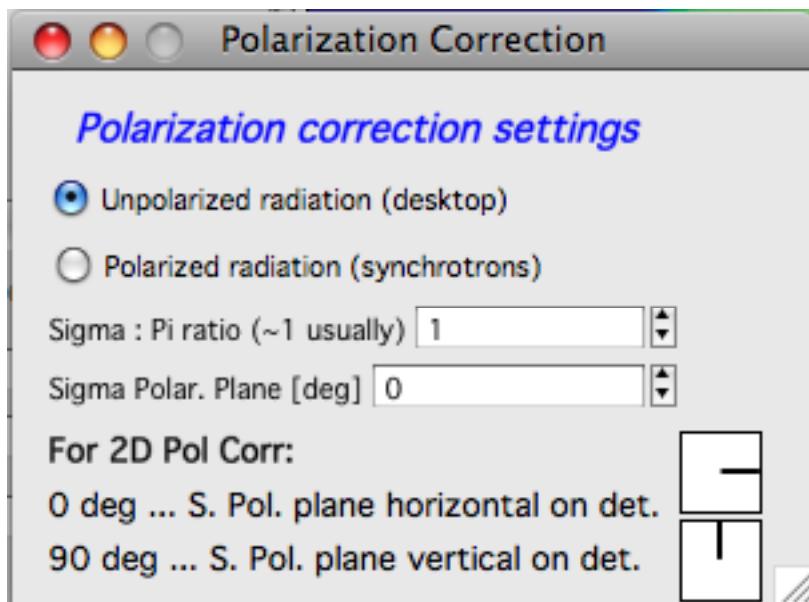
$$Int_{cor} = \frac{Int}{f_\sigma (\cos(2\theta)^2 \cos(\alpha)^2 + \sin(\alpha)^2) + (1 - f_\sigma) (\cos(2\theta)^2 \sin(\alpha)^2 + \cos(\alpha)^2)}$$

where f_σ is fraction of sigma polarization, 2θ is 2 theta angle, and α is azimuthal angle from the plane of polarization plane.

Implementation

All of the Polarization corrections (from version 1.42) in Nika are applied by scaling the 2D data by the formulas above after all of the corrections (including background and dark current subtraction).

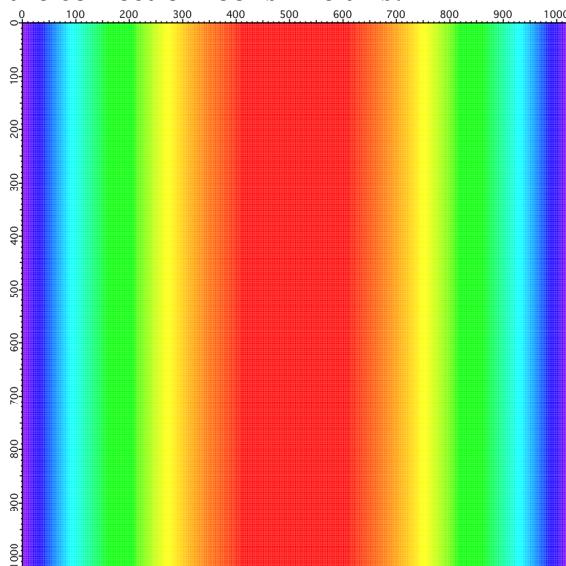
In the following panel which shows after selecting “Polarization correction” on the main panel:



After selecting Polarized radiation you need to make further choice...

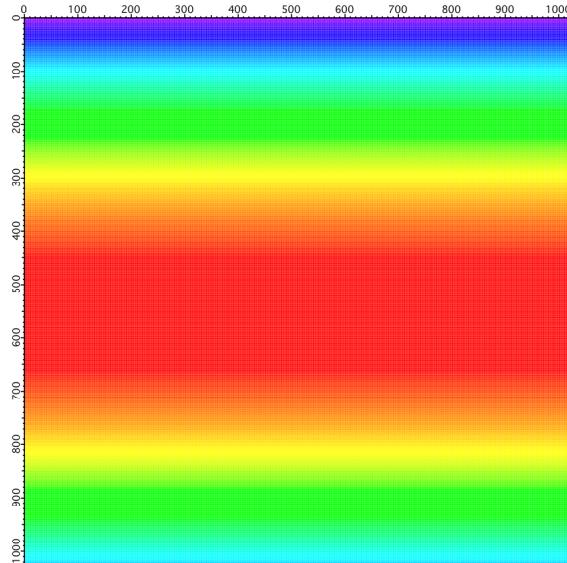
If the Sigma Polarization Plane is 0 degrees, then the detector orientation is such, that the polarization plane is horizontal in the Nika image of the detector. Note that horizontal is Nika's definition of 0 degrees on the detector.

This has nothing to do with the orientation of polarization in real World, this is an orientation between the polarization plane and the way detector is read. In this case the correction looks like this:



with largest correction (increase of intensity) where the color is blue.

For case, when polarization plane is vertical in Igor image (perpendicular to Nika's definition of 0 degrees on detector), the correction looks like this:



with maximum correction (blue color).

Uncertainties (“Errors”)

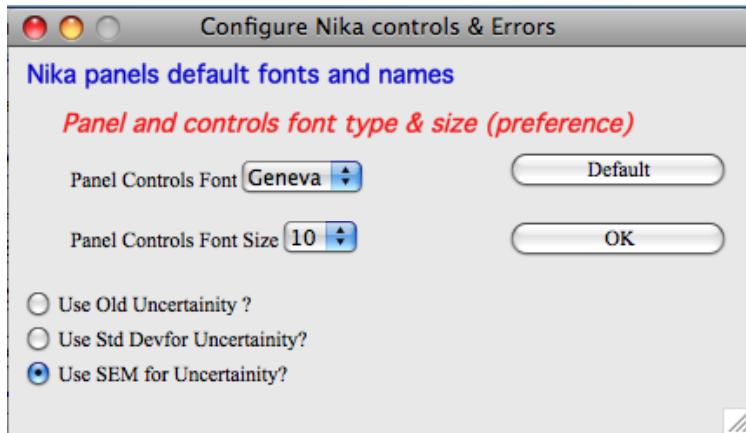
Uncertainty estimate in 2D data reduction is sore point and I have not yet found correct solution for it. As far as I know there is really no good way to get meaningful estimates.

To complicate the matter is, that prior version 1.43 (1.42 and before) there is bug in the uncertainty (error) calculation, which results in overestimate of the values. My intention was to provide standard deviation of the values averaged into the pixel, but simply, I made typo, which resulted in somehow higher values.

Therefore for version 1.43 I provide now three different methods for uncertainty calculations, Standard deviation is default. For compatibility purposes user can choose old (incorrect) version and also standard error of mean – SEM - (standard deviation / $\sqrt{\text{number of points}}$).

Please note, that the line profile calculations provide ONLY standard deviation or SEM, since they never used the old method (they use Igor internal method for standard deviation). They default to standard deviation if old method is selected.

The Uncertainty method can be changed in the “Configuration panel” available from menu.

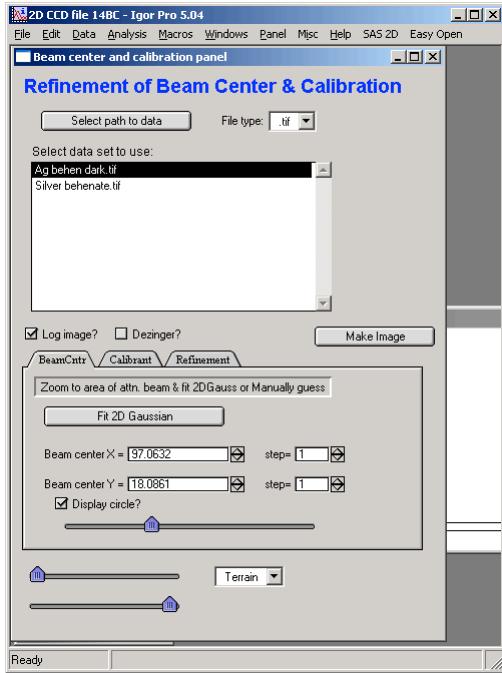


Beam Center and Geometry refinement tools

Included tool for finding beam center and refining geometry parameters allows at this time to:

1. Locate beam center when image with attenuated beam is collected by fitting 2D Gaussian profile
2. Locate beam center with help of graphical tools when diffraction lines are available
3. Refine beam center using least square fitting when diffraction lines are available
4. Refine Sample to detector distance, wavelength and beam centers when calibrant image is available (Silver behenate for example for SAXS and CeO standard for diffraction, but user defined is available).

Main tool is located in the menu under name: "Beam center and geometry cor." This tool creates control panel:

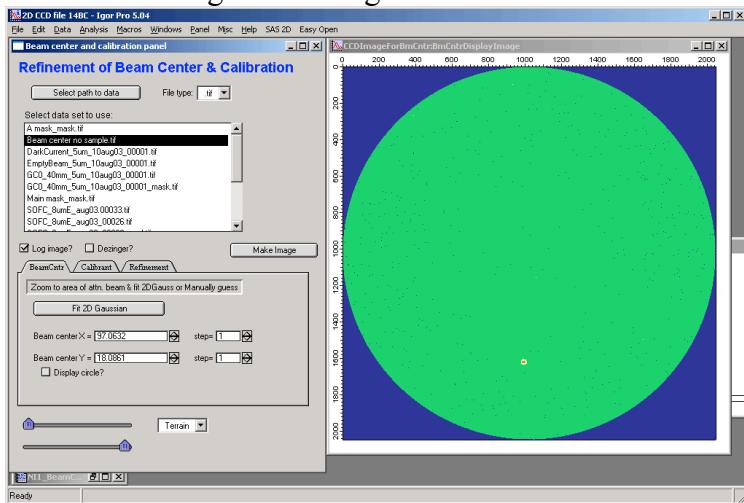


First select path to data, as using the other tools... Select appropriate type of data. Select “dezinger” if needed and number of passes of this process. Check log image if you wish to see log of intensity, but all calculations are done with original intensity... Create image by pushimg “Make Image” button.

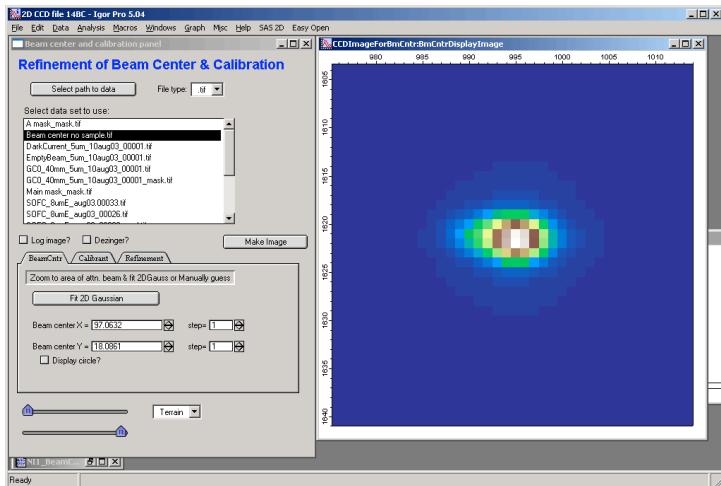
The first tab is for locating (at least roughly) the beam center.

Beam center using attenuated beam

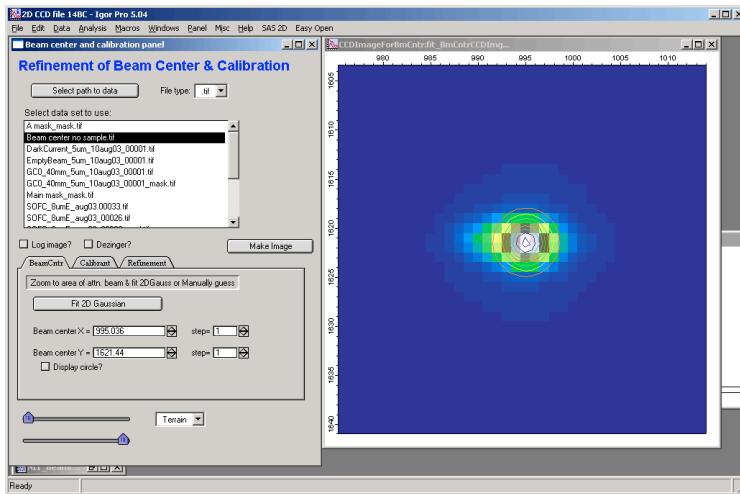
Load in the image containing attenuated beam :



Zoom in the area with the beam using Igor controls (select the area and right-click on Windows, select Expand. Select reasonably small area, fitting over large areas takes a long time.



Push “Fit 2D Gaussian” button:

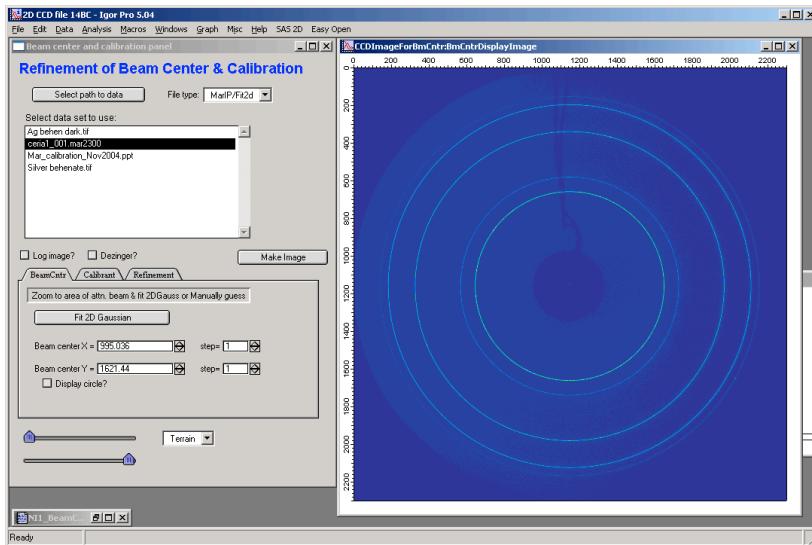


Note, that Contours are appended to the image showing how the Gaussina fit looks like. Results from fitting with beam center values are pushed into right variables.

Beam center using “help circle”

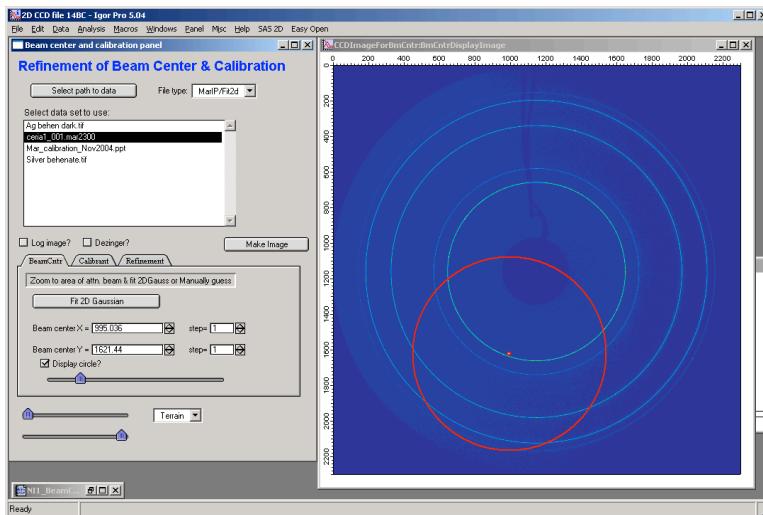
If image with attenuated beam is not available, following method may help to get relatively good estimate for beam center. Needed is image with material which has diffraction rings – this is usually no problem for diffraction, where number of standard exists. For SAXS usual material is silver behenate.

This is image with CeO powder standard collected on 2D area detector:

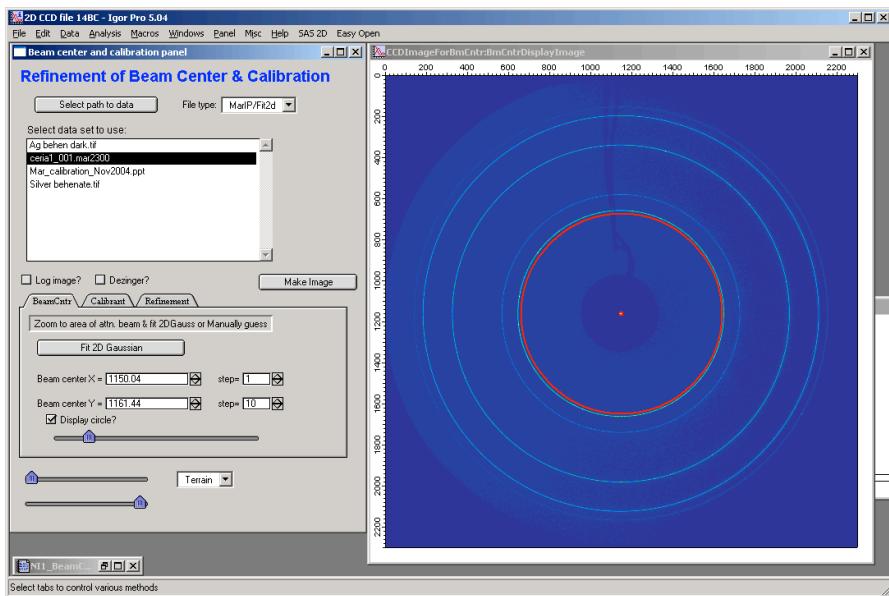


Check the “Display circle” checkbox and use slider to scale the circle to size close to one of the rings. Then change beam center (set useful step size using the “step” variables) to match the circles as good as possible:

So from this:



Get to this:

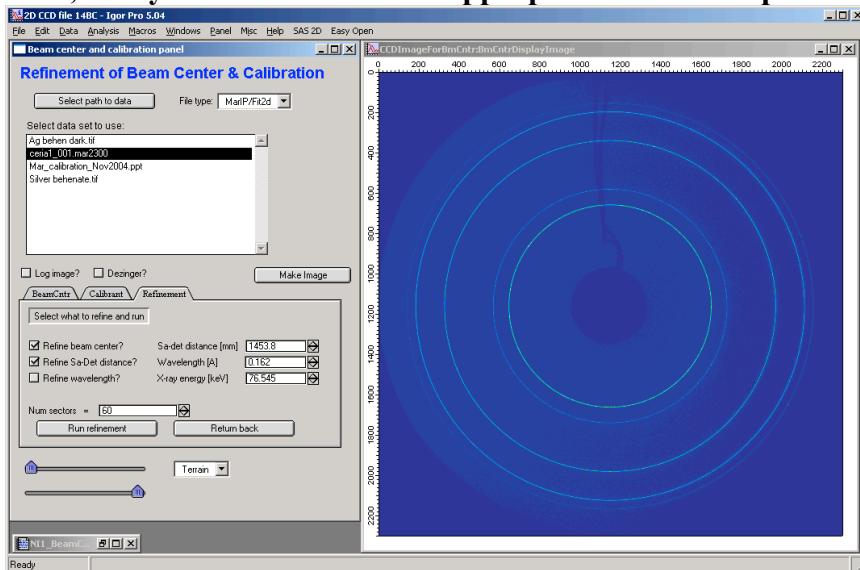


This is already a good estimate of the beam center...

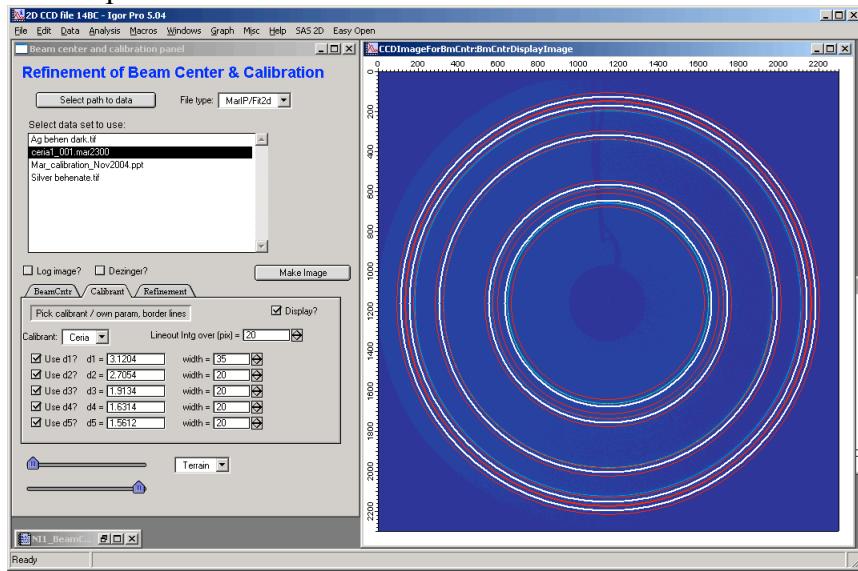
Calibrant & refinement

On the next tab pick the calibrant to use and in tab refinement insert reasonably good estimates of the sample-to-detector distance and wavelength. Pick the predefined calibrant (I have now only CeO and Ag behenate, but can add any number of others). The list of d spacings is filled in... The code can use up to 5 lines for any calibrant material – just overwrite the d spacings on the “Calibrant tab” with own values. User needs to know the d spacing for this material. D spacing cannot be optimized!

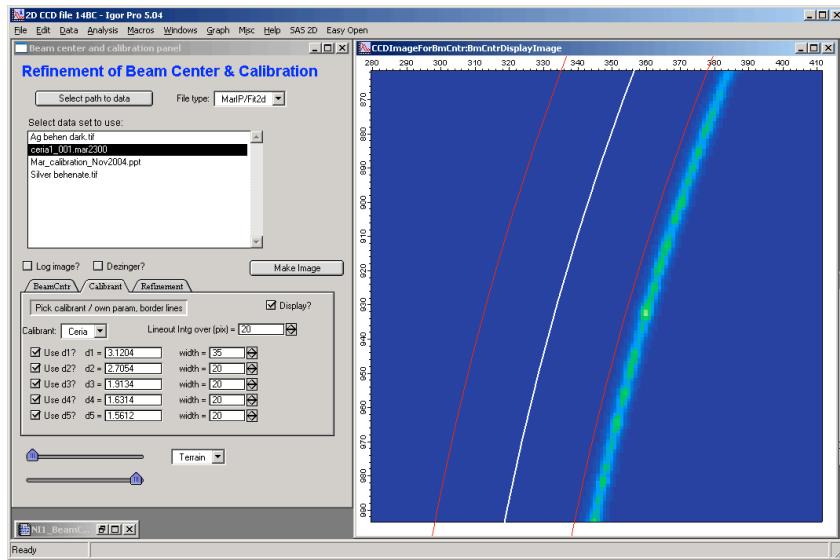
Note, that you have to have also appropriate size of the pixels set in the main panel:



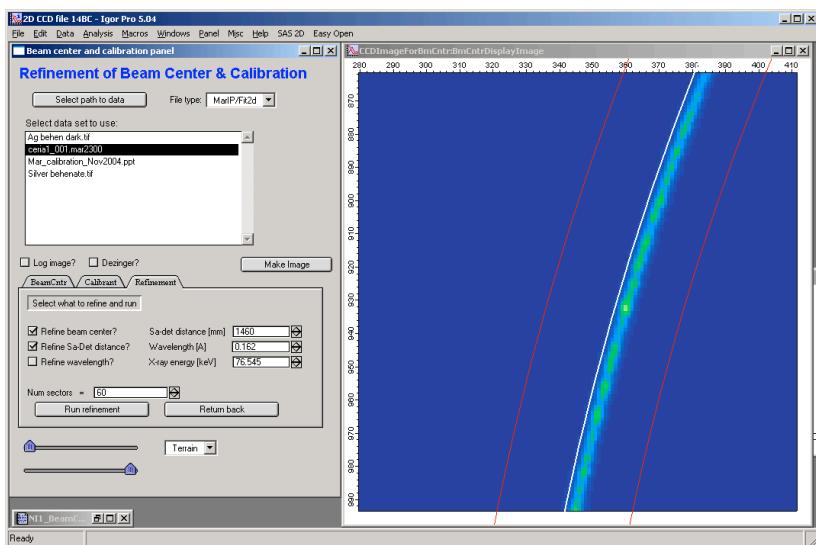
In the tab “Calibrant” now select “Display?” Checkbox. This will add circles where using current parameters should be the lines and two lines around each of this line.



Note detail here:

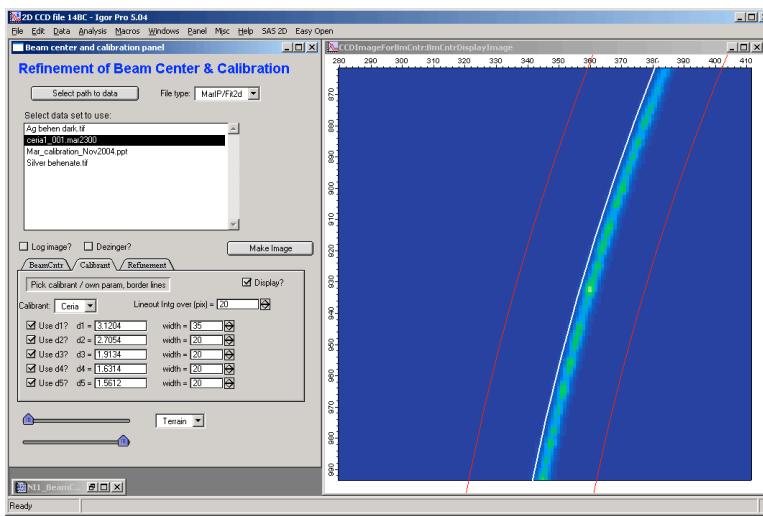


The white line is calculated position of the diffraction from current parameters, greenish fuzzy line below is the diffraction line and red lines indicate the width which will be used by the code to search for the line positions. In order for the code to work, the diffraction line has to be within the two red lines all way around the circle. It has to be single line within this area – therefore no overlapping lines are possible here.... To do this, change wavelength and sample to detector distance, possibly beam center positions. See here:



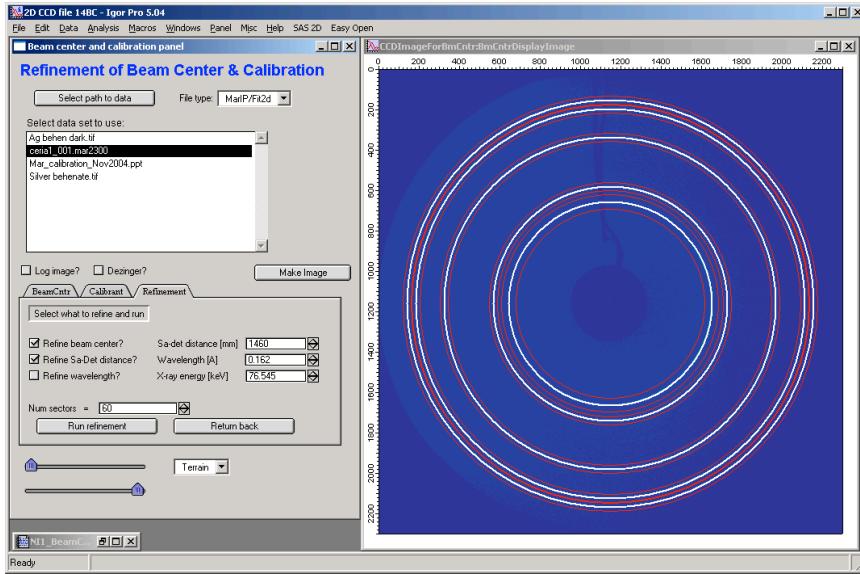
If needed make the width between the two red lines wider as necessary – it is line specific, so each diffraction line can have different width. Note, that the peak position is found by fitting Gaussian profile on intensity profile in the radial direction between the two red lines, so they need to contain some flat background around the diffraction line, but now too much.

The line profile is taken over width (number of pixels on image) perpendicular to the radial direction as set in “Lineout Intg. Over (pix)” on “Calibrant” tab. This value is ONLY one for all diffraction lines. Depending on quality of the lines this may be narrow or broad. If the lines are broken up, with spots, wider will help, but too wide will reduce precision.



In the “Refinement” tab select which parameters to refine – beam center, Sample-to-detector distance and/or wavelength. Note, that to refine wavelength AND sample-to-detector distance together you need at least 2 diffraction lines.

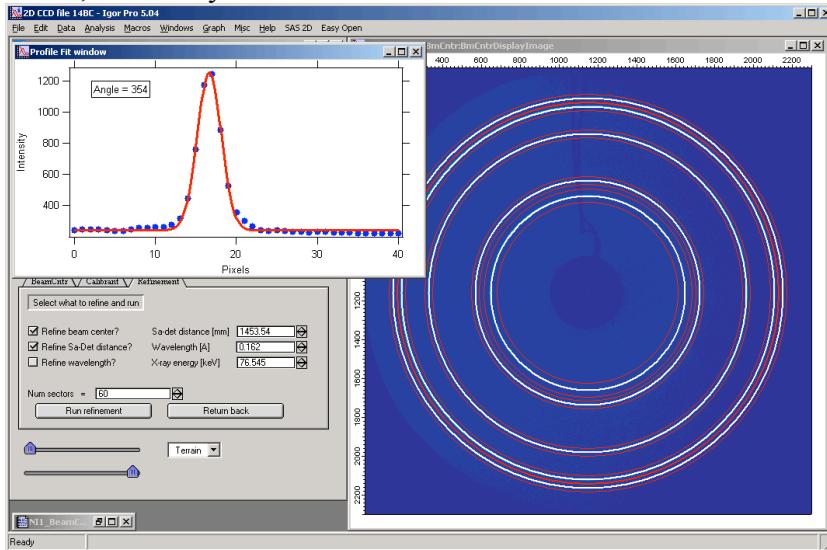
Select number of “sectors” to use (see below is set to 60).



This is how many directions for each ring will be evaluated. If the direction falls out of image, it is skipped. Note, too many may take lot of time...

When ready, push “Run refinement” and observe:

As refinement progresses, dotted red line on the image indicates which direction/line are being evaluated and “Profile fit window” graph shows the intensity vs pixel data there and fitted Gaussian profile. Observe and judge quality. If the quality is poor and data are misfit, it is likely that results of refinement will be bad...



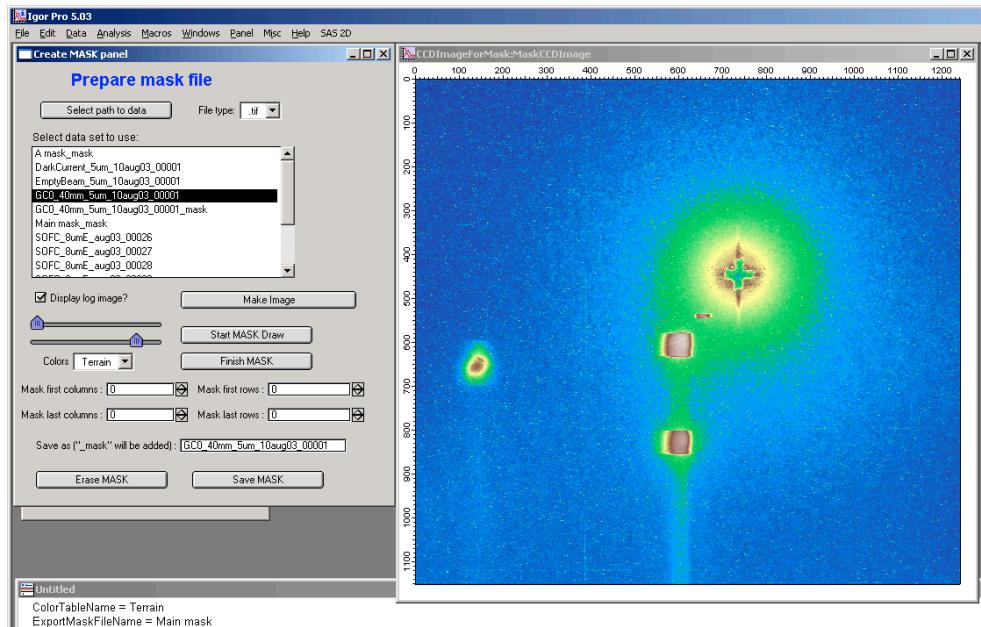
If the refinement at the end fails, you get error message and no change to original parameters is made. If refinement is successful but you still do not like the result, you can recover the previous parameters by pushing button “Return back”.

Otherwise, if successful, the results are pushed into the right variables in the main panel and all is done.

Note, with Silver behenate for SAXS, there is only one line, so the processes is easier. But one cannot refine wavelength AND sample-to-detector distance. Note, the line width for Silver behenate needs to be significantly larger and also it is likely that the “Lineout Intg over “ needs to be larger...

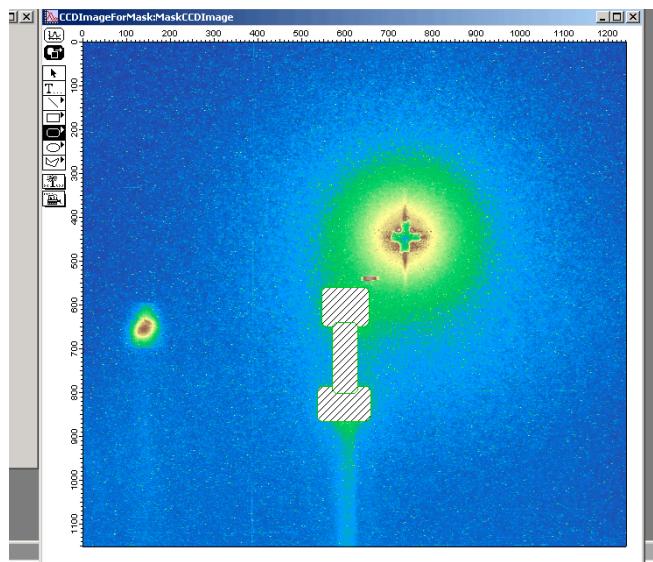
Note: Next update will be to add detector orientation and use of mask. At this time you cannot mask this image, which may be major problem under some circumstances...

Mask



To create mask use tool “Create mask” from either “SAS 2D” or Mask tab on main panel. Select image which you want to use to build the mask from.

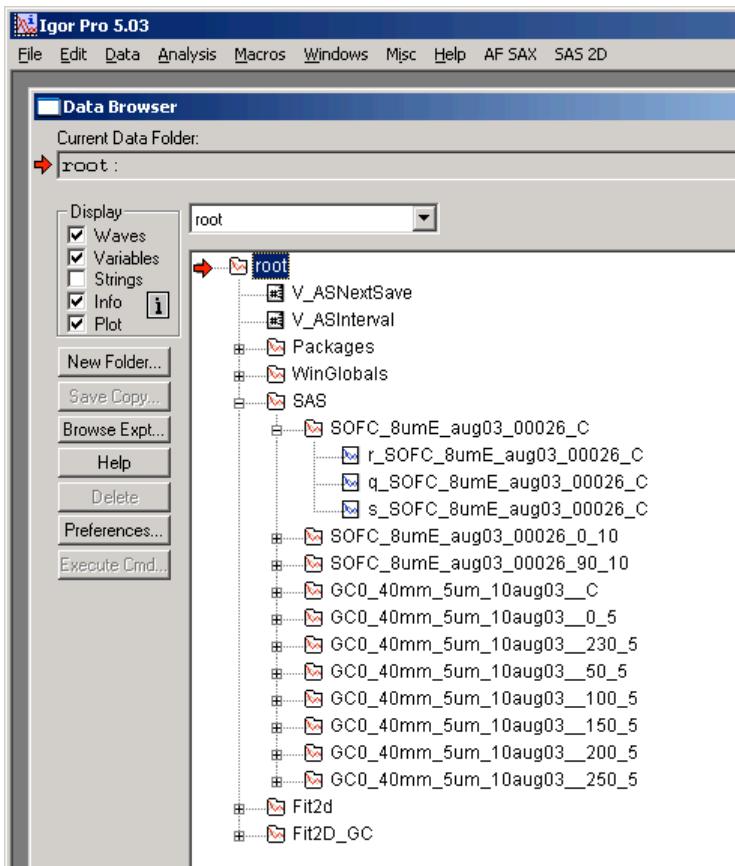
Push “Make image” to create image. Modify by Display log display checkbox, sliders and Colors popup to see details on the image as necessary. Then push “Start MASK draw” to start drawing mask. This brings tools up on the graph.



Use Igor tools (rectangle and other tools) to cover area to be masked OFF. If first/last few lines need to be covered, use the Mask first/last columns/rows below. When done push “Finish mask” and the SAVE the mask, giving it easy to understand name. Code will add `_mask.tif` to the name. This mask can then be loaded as mask to be used.

Output data

Internal



Data are internally stored (if selected) within Igor experiment in folder root:SAS: in folders with

`nameOfSample_C` being the circular average

`nameOfSample_Angle_halfWidth` being sector average around direction Angle with sector half-width.

The wave names

X-axis data

`q_NameOfSample_C (or _Angle_halfWidth)`

q vector in A⁻¹

`t_NameOfSample_C (or _Angle_halfWidth)`

2 theta, if output with respect to 2

`theta`

`d_NameOfSample_C (or _Angle_halfWidth)`

d for output wrt d

y axis data

`r_NameOfSample_C (or _Angle_halfWidth)`

intensity (if calibrated in whatever

units – thickness is converted to cm, so it should be cm⁻¹)

error

`s_NameOfSample_C (or _Angle_halfWidth)`

error for intensity

other

`w_NameOfSample_C (or _Angle_halfWidth)`

width of each bin of Q/d.2 theta.

This is for LUT output, and provides data for bin-width smearing. Smaller number of bins, larger width of each. For linear binning, this is same number and is (Max-Min/numOfPoits), but for log binning this is varying function of bin position.

For Line profile data:

For example for GI_Verical line in my test case, this was the name:

gc_saxs_395_GI_VLp_0.0077

“gc_saxs_395_”.... Part of the name of used image

GI_VLp_.... GI_Verical Line

0.0077 q_y value at which the data were calculated.

Exported data are Int, error, Q, qx, qy, qz columns with header and column names

Saved data in Igor are

r_NameOfSample_ProfileIndicator_Qvalue	intensity
q_NameOfSample_ProfileIndicator_Qvalue	$q [A^{-1}]$
s_NameOfSample_ProfileIndicator_Qvalue	error
qy_NameOfSample_ProfileIndicator_Qvalue	$qy [A^{-1}]$
qz_NameOfSample_ProfileIndicator_Qvalue	$qz [A^{-1}]$
qx_NameOfSample_ProfileIndicator_Qvalue	$qx [A^{-1}]$ (generated ONLY if GI... profile is used)

Note, intensity wave has attached wave note, containing some useful information:

CalibrationFormula=1*((Sa2D));CurrentMaskFileName=A

mask_mask;QvectorNumberPoints=300;CircularAverage=1;

ASCII external

ASCII files with following data are stored in the selected folder:

```
# CalibrationFormula=1*((Sa2D))
# CurrentMaskFileName=A mask_mask
# QvectorNumberPoints=300
# AngularSector=150
# AngularHalfWidth=5
```

0.01601654	0	0
0.0163735	1537	39.20459
0.01655496	1467	38.30144
0.01673842	1416	999.0073
0.01692392	1505	38.79433

The columns contain first q, second intensity and third error...

Instrument support

Instrument support are packages of specific additions in Nika to support special instrument. Depending on the instrument, these functions add capabilities and modify settings to make support of specific instrument easy.

DND CAT (APS 5ID) SAXS camera

DND CAT provides users with data, which are organized in specific folder structure. The data are reduced using scripts based on fit2d at the beamline. However, if users wants to process data later in different manner, they have to contact beamline staff and whole process is cumbersome.

Nika DND support is build on presence of evaluated data in text file, where header contains all necessary information for data reduction. Therefore, user opens this text file and the Tiff file with the processed image is found automatically (if user did not change the folder structure). Alternatively, user can point the Nika to the image files, when asked.

The data can then be reprocessed – for example different sectors can be analyzed etc. Note, that the user needs to make a new mask, but other parameters (beam center, wavelength, calibration constant s well as sample transmission and thickness) are loaded from the header.

To use:

1. select DND/txt as file type and point find data in the right folder. It is likely something like:.../APSCycle/YourName/Month/processing/data/plot_files
2. select and display one or more of the text files which contains current configuration and display. If the folder structure is correct, tiff image is found automatically. If not, Nika will ask for the image location. It should be necessary only once, unless the images are in different places.
3. Select “Instrument configurations” >> “DND CAT”. Select configuration from the listed names of the text file(s) which were loaded already. This will set wavelength, pixel sizes, distance, and centers... Also it will set proper calibration configuration and functions, which should be used. Note: Dark was already subtracted from these files, so you need only the few parameters listed (thickness, calibration constant and transmission).
4. Create and use mask.
5. If you have empty run measurements, select the checkbox for “**subtract empty**”. Do not change values for “**Use I0/I0emp**” or value of 1 for I0. This is important to scale properly Empty and Sample incoming intensities and measurement times.
6. Select proper reduction parameters (circular, sector etc...).

For other instrument scientists:

Other instrument setups can be added on request. Provide me with enough data and description and I can write support for your instrument.

Create Flood field

This simple routine will help user to create “Flood field” – or as NIKA is calling Pix2D sensitivity field.

Definition: This field is matrix of real values such, that when intensity measured in any given pixel is DIVIDED of its pix2D value, the intensity is scaled to correct intensity.

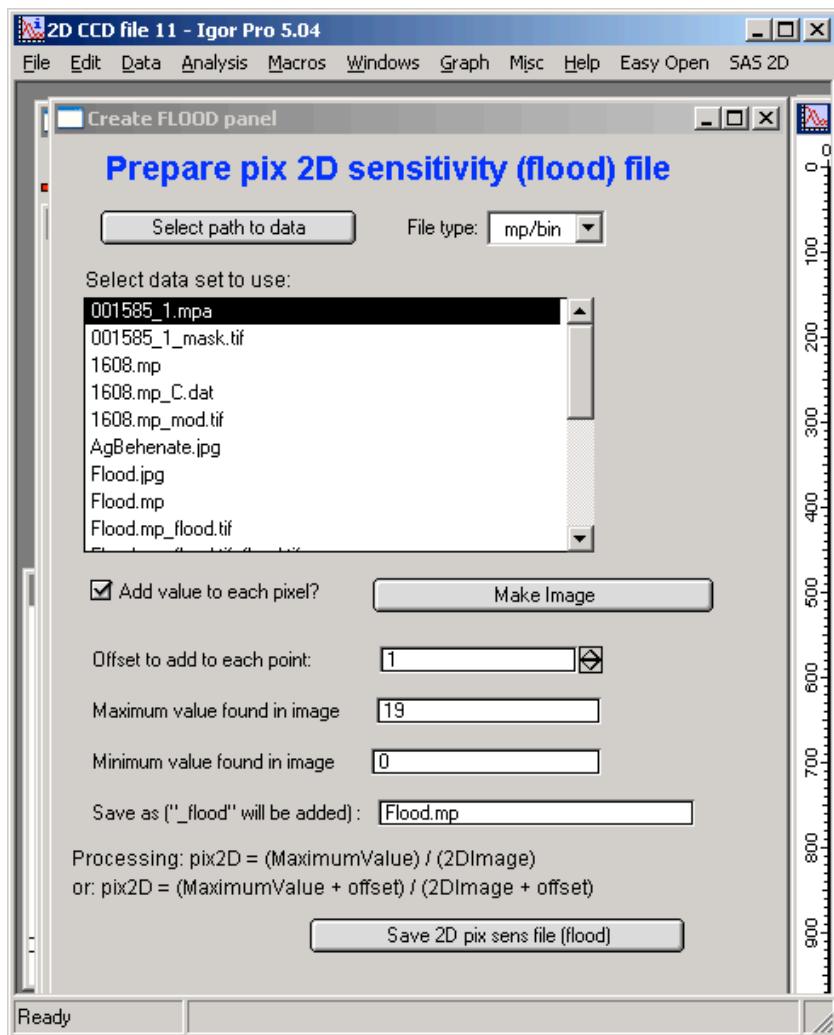
Corrected value = Measured value / pix2D sensitivity value

To create this pixel sensitivity map one has to expose all pixels to the same incoming flux of photons (neutrons) – aka flood field, flat field etc... With enough counts measured for each pixel – and assuming the incoming field is really “flat” – the intensity measured for each pixel is now inversely proportional to pixel sensitivity.

If you have such image, this routine can help to convert this image into NIKA-type pix2D sensitivity image. Note, that this image is saved as tiff file with single precision real number in each pixel – this image may not be readable by other packages!!!!

To create this pix2D sensitivity map, one needs to load in image and for each pixel divide measured value for that pixel by the maximum intensity measured in the image. If any pixel has intensity = 0 (WRONG... You need to collect at least 1 count in each point) it is possible to add 1 to each point intensity to avoid having 0 as corrections in the pix2D sensitivity file, which would cause dividing by 0 and therefore would remove this pixel from any calculations in the future....

GUI



The GUI is very similar to Mask GUI:

“*select path to data*” Select path, where measured flood field is located. Note, resulting pix2D sensitivity file will also be there.

“*File type*” well, select type of data you have. Note, resulting pix2D sensitivity file is tiff file.

Note, at this time this tool can read ONLY file file at time. If you have multiple files to process, you need to first use the main tool to average the measurements, save them as averaged file and then load the file in this tool.

“*Select data set to use*” select the image to use

“*Make Image*” Loads data in and Creates image

“*Add value to each point*” – select, if you need to add some number of counts to each point in image. This will open new control “*Offset to add to each point*”.

Note, the offset will be selected automatically, if during loading the data this tool detects, that there are points with 0 intensity in them. Conversely, this option will be deselected, if minimum intensity in the image is more than 0....

Further controls will be populated during loading of the data set:

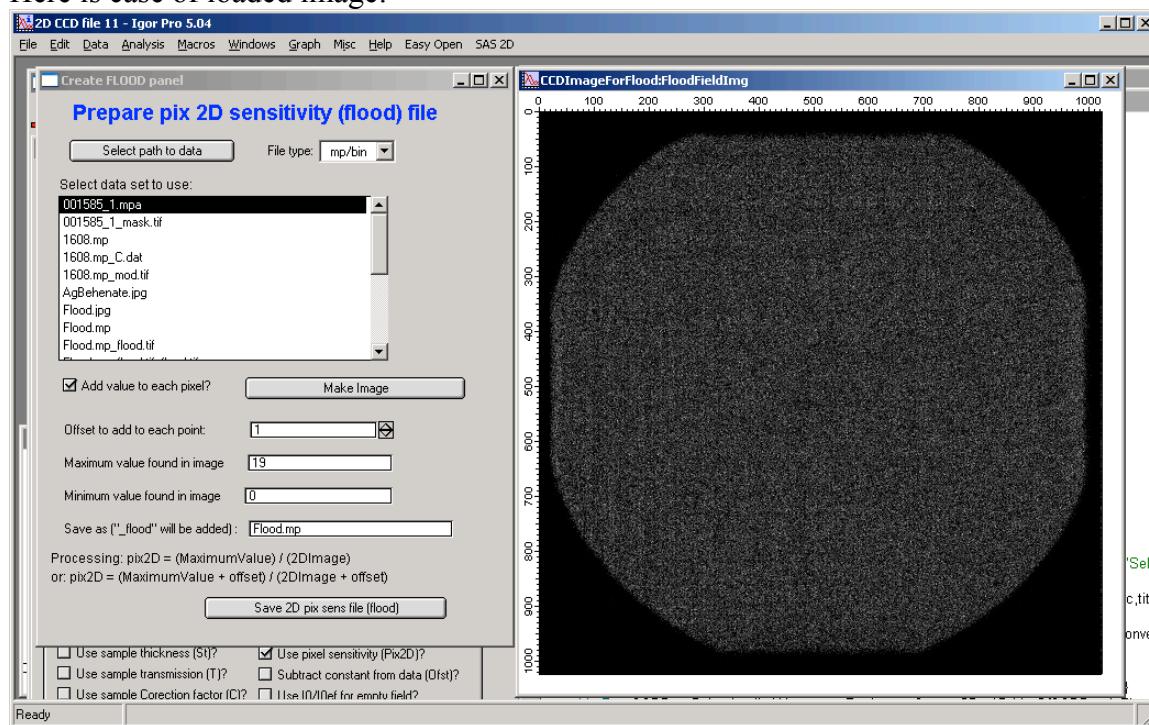
“Maximum value found in image” and *“Minimum value found in image”*. These are here mainly for information.

However, note, that it is possible to overwrite the *“Maximum value found in image”* and this new value will be used instead of the original maximum value for correction described above. It is therefore possible to create pix2D sensitivity file with different range of values. Note, that the code does not check for range of pix2D sensitivity values...

Warning

The code really does not check for values here, and it is possible to set offset negative, for example, and generate negative pix2D sensitivity map... This will cause all sorts of problems later on – **you were warned!**

Here is case of loaded image:



“Save 2D pix sensitivity file (flood)” will process the file as described above and save it. It will be saved in place where original file came from and _flood.tif will be added to text in the “Save as (“_flood” will be added)” control. The text will be first checked to be acceptable name for OS.