

OPTIONS AND ARGUMENTS OF THE SAXS PROGRAMS	3
Version: 2000-09-04	3
Introduction	3
Prompts	3
Arguments	4
Options	4
Reference Systems	4
Orientation	5
List of Options.....	5
Example:	6
Option Values	6
Arithmetic Expressions as Values.....	6
General Options Used in All SAXS Programs	8
General Options for Output and Input Files.....	8
Image Numbers and Memory Numbers	9
File Access	9
Loop Control	9
Transformation of Input and Output Images	10
Image Dimension	11
Reference System Parameters	12
SAXS Program Specific Options	12
binary2saxs	12
ccd2saxs	13
gas2saxs	13
gel2saxs	13
saxs_add	14
saxs_addcol	14
saxs_addrow	14
saxs_aff	15
saxs_ascii	15
saxs_ave	16
saxs_angle	17
saxs_bsl	18
saxs_cave	18
saxs_clip	18
saxs_col	19
saxs_const	20
saxs_csub	20
saxs_cut	20
saxs_div	21
saxs_divcol	21
saxs_divrow	22
saxs_gnaw	22
saxs_log	22
saxs_mac	23
saxs_mul	23
saxs_mulcol	23
saxs_mulrow	24
saxs_normn	24
saxs_patch	29

saxs_power.....	29
saxs_rot.....	29
saxs_row.....	30
saxs_scal.....	31
saxs_stat.....	31
saxs_sub.....	32
saxs_subcol.....	32
saxs_subrow.....	33
saxs_tiff.....	33
saxs_urol.....	36
OTHER ROUTINES.....	36
Programs	36
col2array	36
Macros	37
saxsdisp.mac	37
INDEX.....	38

Options and Arguments of the SAXS Programs

Version: 2000-09-04

Introduction

Parameters can be passed in three ways to the saxs programs:

- a) by prompts
- b) by arguments
- c) by options

The multiplication of the first three images of the input file "x.edf" with the factor 5.5 is used to demonstrate these possibilities. The output is written to "y.edf"

Prompts

The program prompts for some important parameters, but not for all.

```
> saxs_mac
```

```
Usage: saxs_mac [options]
```

```
<i1nam> <onam> <i1fst> <i1lst> <i1inc>
```

```
<odum> <odim1> <odim2> <ofac> <ocon> <shft1> <shft2>
```

```
saxs_mac saxs_mac 3.12 24-May-1995, Peter Boesecke
```

```
Input sequence 1 [input.edf] : x.edf <return>
```

```
Output sequence [output.edf] : y.edf <return>
```

```
First image of input sequence 1 [1] : <return>
```

```
Last image of input sequence 1 [1] : 3 <return>
```

```
Increment of input sequence 1 [1] : <return>
```

```
Output dummy [0.000000e+00] : -1 <return>
```

```
Output dimension 1 [1024] : <return>
```

```
Output dimension 2 [1024] : <return>
```

```
<output image> = <input image> * Factor + Const
```

```
Multiplication factor [1.000000e+00] : 5.5 <return>
```

```
<output image> = <input image> * Factor + Const
```

```
Addition constant [0.000000e+00] : <return>
```

Input reference system is Image

output shift 1 [0.000000e+00] :

Input reference system is Image

output shift 2 [0.000000e+00] :

...

...

Arguments

The same parameters can be passed as arguments in the same order as the program would prompt for them. "=" stands for an empty argument. The option -p is used to switch off the prompt mode. With +p the program will prompt and will show the arguments as defaults in parenthesis []. This can be used for checks.

```
> saxs_mac -p x.edf y.edf = 3 = -1 == 5.5 == =
```

Options

Usually, the default of all arguments can be set with options. While arguments must be passed in a specific order options can be passed in any order before the first argument. Options are preceeded by "-" or "+" and can be repeated. The last occurrence sets the value.

```
> saxs_mac -p -ilnam x.edf -onam y.edf -il1st 3 -odum -1 -ilfac 5.5
```

Reference Systems

Currently, six different reference systems are defined: array, image, center, real, normal, saxs. They are used to define the geometrical relation between two images. A reference system is chosen with the option -rsys <refsys> (see chapter Option Values);

Orientation

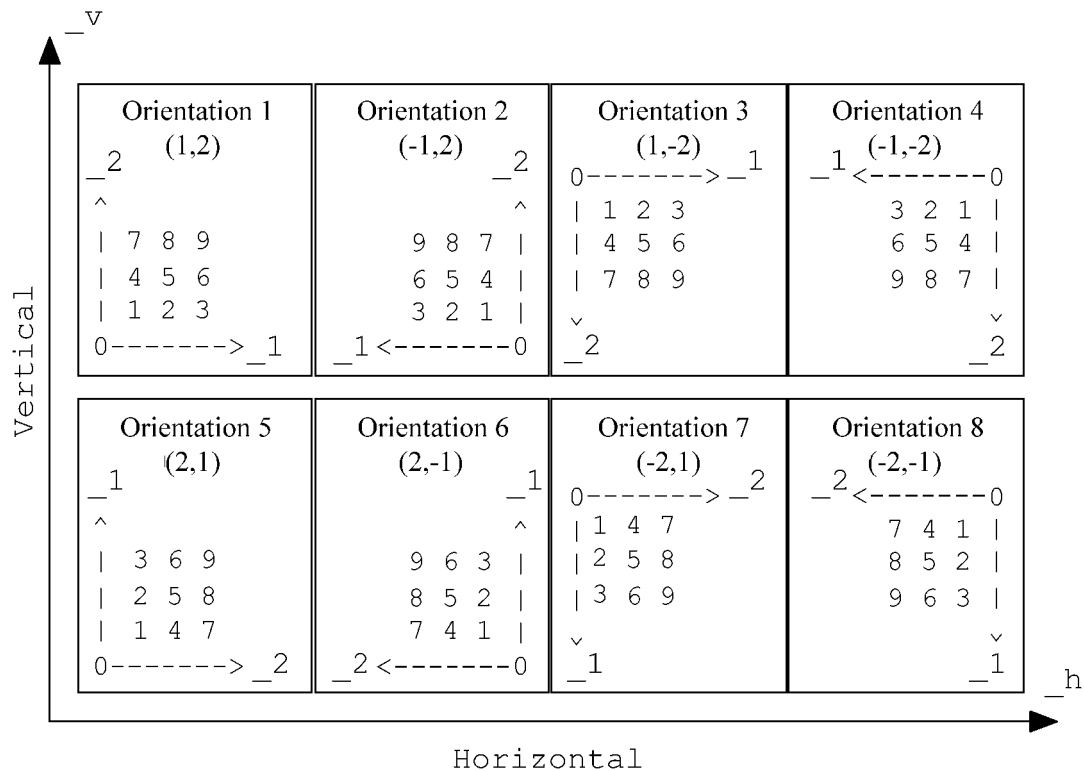


Fig. 1. Orientations of two-dimensional images. In orientation 1 (default for input and output) the geometrical origin of the image is at the lower left corner, index 1 corresponds to the horizontal axis and index 2 corresponds to the vertical axis.

The orientation of the image describes the relation between data indices and directions in real space. In orientation 1 index 1 corresponds to the horizontal axis and index 2 to the vertical axis. Internally, all images are converted to orientation 1. The conversion includes a reorientation of the image and a swapping of some header values, e.g. for dimension, pixel size and offset. The orientation of the input and output image can be chosen with the options `-ilori O` and `oori O`, where `O` is the orientation number shown in Fig. 1.

List of Options

Each option in the following list is followed by a parenthesis that shows the number of parameters and the parameter type:

option [`<num of parameters>`,`<typ>`]

The saxs programs are called in the following way:

saxs_xxx [option] {[option]} [argument] {[argument]}

Options are preceded by '-' or '+'. Options are read from left to right and are processed before the arguments are read. Options can be repeated. The last option sets the value.

Example:

```
saxs_normn +flat -i3nam "flat_field" -trm 0.78 -ofac 1/0.0003 \
image.edf == image.nrm
```

Each option can be followed by parameters. The number of parameters and the type of the parameter are shown in brackets. The following parameter types are used:

Option Values

- +: boolean value , e.g. +option and -option (no parameter)
- d: integer value, e.g. -ofst 3
- f: float value, e.g. -ofac 1/0.0003
- M: file opening mode (n+fp, n+ip, n, o+fp, o+ip, o, a+fp, a+ip, a)
 - n, n+fp : new (create new file and empty any old file, allow to overwrite existing images if header size and image size fit into the existing block structure)
 - n+ip : new + image protection (create new file, do not allow to overwrite existing images)
 - o, o+fp : old (use old file, error if it does not exist, allow to overwrite existing images if header size and image size fit into the existing block structure)
 - o+ip : old + image protection (use old file, do not allow to overwrite existing images)
 - a, a+fp : any (if the file does not exists create a new file, otherwise open it as an old file, allow to overwrite existing images if header size and image size fit into the existing block structure)
 - a+ip : any + image protection (like any but do not allow to overwrite existing images)
- R: reference system value
(array, image, center, real, normal, saxs), e.g. -rsys saxs
 - array: ARRAY coordinate p = pixel coordinate
 - image: IMAGE coordinate i = array coordinate + offset
 - center: CENTER coordinate i = image coordinate - center
 - real: REAL coordinate r = image coordinate * pixel size
 - normal: NORMAL coordinate i = CENTER coordinate * pixel size
 - saxs: SAXS coordinate s = (image coordinate - center) * (pixel size/sample distance) * WaveLength0/wavelength (WaveLength0=1e-9 m)
- s: string value, e.g. -otit "flatfield image"

Arithmetic Expressions as Values

Integer and float values can be expressed by C-type expressions. Additional functions and constants are defined. Values are usually converted to SI-units, e.g. keV is converted to Joule.

physical constants from:

Lawrence Berkeley Laboratory

University of California

Berkeley, California 94720

X-Ray data booklet, second printing, with corrections, April 1996

Constants

```

gamma = 0.577215664901532861;
kb  = 1.380662e-23; /* Boltzmann constant (J/K) */
me  = 9.109534e-31; /* electron rest mass (kg) */
mp  = 1.6726485e-27; /* proton rest mass (kg) */
na  = 6.022045e23; /* Avogadro number (1/mol) */
pi  = 3.1415926535897932384626;
re  = 2.8179380e-15; /* classical electron radius (m) */
c   = 2.99792458e8; /* velocity of light (m/s) */
e   = 1.6021892e-19; /* electron charge (C) */
h   = 6.626176e-34; /* Planck's number (J*s) */

```

Units

```

barn_ = 1e-28; /* barn (1/m^2) */
amu_  = 1.6605655e-27; /* atomic mass unit (kg) */
keV_  = e*1e3; /* _keV (J) */
eV_   = e; /* _eV (J) */
km_   = 1e3; /* _KM (m) */
m_    = 1.0; /* _M (m) */
cm_   = 0.01; /* _CM (m) */
mm_   = 1e-3; /* _MM (m) */
um_   = 1e-6; /* _UM (m) */
nm_   = 1e-9; /* _NM (m) */

```

Arithmetic Functions**Float Functions**

```

rad(x)  = pi/180.0 * x; /* transformation deg to rad */
deg(x)  = 180.0/pi * x; /* transformation rad to deg */
pi(x)   = pi;
sin(x)
cos(x)
tan(x)
asin(x)
acos(x)
atan(x)
atan2(x,y)
sinh(x)
cosh(x)
tanh(x)
floor(x)
ceil(x)
abs(x)
exp(x)
log(x)
log10(x)
pow(x)
sqrt(x)
round(x) = floor( x + 0.5 ); /* rounding */

```

Long integer functions

true = 11;

false = 01;

yes = 11;

no = 01;

(long int) x : transformation of x to long integer;

(round) x : rounding and transformation of x to long integer;

General Options Used in All SAXS Programs

An option can be followed one or more parameters. All options are described in the following way:

<option name> [<number_of_parameters>, <parameter type>]

- test [0, +]
switches a test modus on (+test) and off (-test)
- h [0, +]
displays long (+h) and short (-h) help
- p [0, +]
switches prompt mode on (+p) and off (-p)
- r [0, +]
do not use, undefined behaviour
- shift [2, f]
do not use, undefined behaviour
- rsys [1, R]
defines the reference system (-rsys array | image | real | saxs). The default value is image.
- usys [1, R]
defines the user reference system (-usys array | image | real | saxs). The default value is usys.
- gblk [1, +]
defines, whether the edf output file will be preceded by a general block (+gblk) or not (-gblk) (default: -gblk)
(see description of new edf format)
- bibo [1, d]
selects globally the input byte order of bsl files. All bsl files that are read by a single routine must have the same byte order, allowed values (not case-sensitive): low_byte_first, high_byte_first (default: internal byte order of the machine, e.g. i386 -> LOW_BYTE_FIRST, all others currently -> HIGH_BYTE_FIRST)
- mlw [1, d]
sets the maximum line width in the output file. Lines longer than mlw will be terminated with a backlash and continued in the next line, default: not set.

General Options for Output and Input Files

The following options can be used for output and for input files. An option for the output file starts with o, an option for the <n>-th input file starts with i<n>, e.g. -odum -1 -i1dum -1 -i2dum -1 etc. The number of input and output files can be viewed with the extended help option +h (block should be read as file):


```

...
Number of image blocks      : 3
Maximum number <n> of input blocks: 2
...

```

The file contains 1 output file (-onam <output file name>) and 2 input files (-i1nam <name of input file 1>, -i2nam <name of input file 2>).

Some conversion programs do not have standard input and output files, e.g. gas2saxs. In this case the number of image files ("blocks") is 0.

Image Numbers and Memory Numbers

Image numbers offer the possibility to store a sequence of (small) images in a single file. All images are numbered and can be stored in different memories. Image numbers and memory numbers are integer values. Positive memory values stand for image data, negative memory values for error data. Memory zero must not be used. The default input memory number is 1, the default output memory number is the input memory number.

How image numbers can be chosen is described in the section Loop Control. Memory numbers can be chosen with the options -i<n>mem and -omem.

- omem (i<n>mem) [1, d]
memory number of images in output (input) file, e.g. -omem -1.

File Access

- omod (i<n>mod) [1, M]
output (input) file opening mode, e.g. -i1mod o -omod n+ip: open an existing file for reading and create a new output file, protect the images of the output file
- onam (i<n>nam) [1, s]
output (input) file name, e.g. -i1nam "input.edf" -onam "output.edf"

The file "/dev/null" is used as dummy. Used as input, a saxs_XXX programs runs as if it would read from a file. All parameters can be defined explicitly by options, e.g. saxs_mac -i1dim 512 512 -i1con 5 /dev/null output.edf will create an image with 512 512 pixels and write it to output.edf. Each pixel has the value 5. Output to "/dev/null" suppresses the output to a real file.

Loop Control

All saxs_XXX programs read images sequentially from several input sequences (i1, i2, ...) and write the resulting images sequentially to a single output sequence (o). Per default the output images are described in the same way as the images of the first input sequence, e.g. title, pixel size etc. are copied from the input images of the first sequence. Only a single output image sequence is created: (Output[i01], Output[i02], ...). The output image is a function of the input images: Output[i0] = Function(image1[i1], image2[i2], image3[i3], ..., imageN[iN]). Per default, the start number (ofst), the increment (oinc) and the last number (olst) of the output image sequence are the same as start number (i1fst), increment (i1inc) and last number (i1lst) of the first input sequence. Thus, normally the output image number is equal to the number of the image in the FIRST input file.

i1fst, i1inc, i1lst -> maxloop1 = max(1, (i1lst-i1fst+1)/i1inc)

```

...
iNfst, iNinc, iNlst -> maxloopN = max(1,(iNlst-iNfst+1)/iNinc)
ofst, oinc, olst -> maxloop0 = max(1,(i0lst-i0fst+1)/i0inc)

```

```

maxloop = Min(loop1, .. , loopN, loop0)

```

If the increment of the output image is 0, maxloop0 is set to infinity and it is only written once at the end of the other loops.

If the increment of an input sequence J is zero the loop is stopped after the first loop (maxloopJ is set to 1).

The loop is stopped when the first incrementation loop has finished. Also negative increments are possible.

```

ImageLoop(...) {
  for (O=OFirst,I1=I1First,I2=I2First,...;
    O<OLast,I1<I1Last,I2<I2Last,...;O+=OInc,I1+=I1Inc,I2+=I2Inc) {
    Read(I1,I2,...);
    Function((CmdBlk * ) pcb, (ImgHeadBlk *) ihb, (int *) pstatus );
    Write(O); }
}

```

Missing images are skipped. All start images must exist.

- ofst (i<n>fst) [1, d]
first image number of output (input) file, e.g. -i1fst 10 -ofst 10
- olst (i<n>lst) [1, d]
last image number of output (input) file, e.g. -i1lst 20 -olst 20
- oinc (i<n>inc) [1, d]
increment of image number of output (input) file, e.g. -i1inc 2 -oinc 2, if oinc (i<n>inc) is 0 the output image is only written once at the end of the loop (the input image is only read once)

Transformation of Input and Output Images

After reading and before writing the pixel values of all images can be modified. The transformations are done in the following order after reading the input data and before writing to the output file. If the option is omitted nothing is done.

```

clip (-omin, -omax),
gnaw (-ogna),
rebin (-obin),
orientation (-oori)
transform (-ocon, -ofac))

```

Attention, each step can create new dummy pixels, e.g. -odum 10 -ofac 0 -ocon 10 would replace all output pixels with the value 10 which is the dummy value.

- omin (i<n>min) [1, f]
minimum value in output (input) image, e.g. -omin 0.01, lower values are replaced with dummies

- **omax (i<n>max) [1, f]**
maximum value in output (input) image, e.g. -omax 10000, larger values are replaced with dummies
- **ocon (i<n>con) [1, f]**
constant is added to output (input) image value before saving, e.g. -ocon 1000 (-ilcon 1000), adds 1000 to each non-dummy pixel of an image before saving (after reading). Multiplication and addition are done at the same time according to: $\text{value}' = \text{value} * \text{factor} + \text{constant}$.
- **ofac (i<n>fac) [1, f]**
factor is multiplied to output (input) image value before saving, e.g. -ofac 0.37 (-ilfac), multiplies each non-dummy pixel of an image with 0.37 before it is saved (after it was read). Multiplication and addition are done at the same time according to: $\text{value}' = \text{value} * \text{factor} + \text{constant}$.
- **obin (i<n>bin) [2, d]**
rebinning factor of the output (input) images, e.g. -obin 2 2 rebins the output image by a factor 2. The image dimension is automatically adapted.
- **oori (i<n>ori) [1, d]**
input: input image is transformed from orientation i<n>ori to
output: output image is transformed from orientation 1 to orientation oori
The value range of ori is from 1 to 8 (see Fig. 1)
- **ogna (i<n>gna) [2, d]**
rounding of the dummy area of the output (input) images, e.g. -ogna 2 4 enlarges each dummy pixel by 2 pixels horizontally and by 4 pixels vertically.

Image Dimension

- **odim (i<n>dim) [2, d]**
dimension of each output (input) image, e.g. -odim 720 1000 sets the output dimension horizontally to 720 and vertically to 1000. Unused pixels are filled with dummy. This option overwrites the header information of an input image.

In case of an input image this dimension is used instead of the value coming with the image.

Dummy Value

- **odum (i<n>dum) [1, f]**
dummy value of each output (input) image, e.g. -odum -1 sets the output dummy value to -1. The value 0 is never a dummy. A value is a dummy if $(\text{abs}(\text{value} - \text{dummy}) \leq \text{ddummy})$ (with $\text{ddummy} > 0.1$). Dummy values are ignored in calculations and are replaced by the output dummy value. This option overwrites the header information of an input image.

In case of an input image this dummy value is used instead of the value coming with the image.

- **oddum (i<n>ddum) [1, f]**
ddummy value, radius around a dummy value, in which the pixel values are ignored. This value is automatically set and must be always larger than 0.1. There should be no need to use this option. See odum. This option overwrites the header information of an input image.

In case of an input image this ddummy value is used instead of the value coming with the image.

Additional Information

- otit (i<n>tit) [1, s]
output (input) image title, e.g. -otit "black sample"

In case of an input image this title string is used instead of the title coming with the image.

- otim (i<n>tim)
(*reserved for time string*)

Reference System Parameters

The reference system for calculation is globally chosen with rsys. The reference systems are described in a separate document.

- ooff (i<n>off) [2, f]
offset of the output (input) image (in pixel)
This option overwrites the header information of an input image.
In case of an input image this offset is used instead of the offset coming with the image.
- ocen (i<n>cen) [2, f]
center of the output (input) image (in pixel). This option overwrites the header information of an input image.
In case of an input image this center is used instead of the center coming with the image.
- opix (i<n>pix) [2, f]
pixel size of the output (input) image (in meters). This option overwrites the header information of an input image.
In case of an input image this pixel size is used instead of the pixel size coming with the image.
- odis (i<n>dis) [1, f]
sample distance in the output (input) image (in meters). This option overwrites the header information of an input image.
In case of an input image this distance is used instead of the distance coming with the image.
- owvl (i<n>wvl) [1, f]
wavelength in the output (input) image (in meters). This option overwrites the header information of an input image.
In case of an input image this wavelength is used instead of the wavelength coming with the image.

SAXS Program Specific Options

binary2saxs

Number of image blocks : 2

Maximum number <n> of input blocks: 1

Arguments

binary2saxs [options] <fnam> <onam> <i1dim1> <i1dim2> <fskp> <ftyp> <odum>

Options

- fskip [1, d]
number of bytes to skip
- fnam [1, s]
binary file name
- ftyp [1, s]

binary type : type of a single element

- "float"
 - "short integer"
 - "integer"
 - "long integer"
 - "unsigned short integer"
 - "unsigned integer",
 - "unsigned long integer"
 - "byte"
- fend [0, +]
byte swapping on/off (+fend: swap bytes, -fend: keep byte order (default))

ccd2saxs

Number of image blocks : 0

Arguments

ccd2saxs [options] <i/p file> <dummy> <first> <last> <inc> <o/p file> <o/p first> <i/p hm>

Options

- ext [1, s]
extension of the edf output file (default: ".edf")

gas2saxs

Number of image blocks : 0

Arguments

gas2saxs [options] <i/p file> <dummy> <first> <last> <inc> <o/p file> <o/p first> <i/p hm>

Options

- ext [1, s]
extension of the edf output file (default: ".edf")

gel2saxs

Number of image blocks : 0

Arguments

gel2saxs [options] <i/p file> <dummy> <o/p file>

saxs_add

Addition of two image sequences

Arguments

```
saxs_add [options]
    <i1nam> <onam> <i1fst> <i1lst> <i1inc> <i2fst>
    <odum> <odim1> <odim2> <i1con> <i1fac> <i2con> <i2fac>
    <ocon> <ofac>
```

saxs_addcol

Average columns between col1 and col2 in image sequence 2 and add the results to each column of image sequence 1

Number of image blocks : 3

Maximum number <n> of input blocks: 2

Arguments

```
saxs_addcol [options]
    <i1nam> <onam> <i1fst> <i1lst> <i1inc> <i2fst>
    <odum> <odim1> <odim2> <i1con> <i1fac> <i2con> <i2fac>
    <ocon> <ofac>
```

Options

- col1 [1, f]
first column
- col2 [1, f]
last column

saxs_addrow

Average rows between row1 and row2 in image sequence 2 and add the result to each row of image sequence 1

Number of image blocks : 3

Maximum number <n> of input blocks: 2

Arguments

```
saxs_addrow [options]
    <i1nam> <onam> <i1fst> <i1lst> <i1inc> <i2fst>
    <odum> <odim1> <odim2> <i1con> <i1fac> <i2con> <i2fac>
    <ocon> <ofac>
```

Options

- row1 [1, f]
first row
- row2 [1, f]
last row

saxs_aff

Affine transformation of an image sequence

Transformation matrix T

$$T = (A, B); A = \begin{pmatrix} T00 & T10 \\ T01 & T11 \end{pmatrix}; B = \begin{pmatrix} T20 \\ T21 \end{pmatrix};$$

$$W' = A * W + B$$

with coordinates W (input image), W' (output image)

see also: saxs_rot

Number of image blocks : 2

Maximum number <n> of input blocks: 1

Arguments

saxs_aff [options]
 <i1nam> <onam> <i1fst> <i1lst> <i1inc>
 <odum> <odim1> <odim2>
 <t00> <t01> <t10> <t11> <t20> <t21>

Options

- t00 [1, f]
transformation matrix element T00 (default: 1.0)
- t10 [1, f]
transformation matrix element T10 (default: 0.0)
- t20 [1, f]
transformation matrix element T20 (default: 0.0)
- t01 [1, f]
transformation matrix element T01 (default: 1.0)
- t11 [1, f]
transformation matrix element T11 (default: 0.0)
- t21 [1, f]
transformation matrix element T21 (default: 0.0)

saxs_ascii

Transformation of an edf-file into ASCII format

Number of image blocks : 2

Maximum number <n> of input blocks: 1

Arguments

```
saxs_ascii [options]
    <i1nam> <onam> <i1fst> <i1lst> <i1inc>
    <odum> <odim1> <odim2>
    <header> <swap> <head_1> <head_2>
```

Options

- scf [1, f]
-scf <factor> : multiplication of coordinates with a scale factor, e.g. to transform s in nm to $q=2\pi*s$ in Angstrom (-scf "0.2*pi")
- pref [1, s]
-pref 'prefix' : specify the prefix of the output file
- ext [1, s]
-ext 'extension' : extension of the output ascii file (default: ".txt")
- head [0, +]
+-head : enables/disables the header printing
- swap [0, +]
+-swap : enables/disables swap of rows and lines
- hd1 [0, +]
+-hd1 : enables/disables the i_1 coordinate printing
- hd2 [0, +]
+-hd2 : enables/disables the i_2 coordinate printing
- abs [0, +]
+-abs : print absolute values of coordinates
- spc [1, s]
+-spc : spacer between data columns, default: tabulator

saxs_ave

Averaging of two image sequences

Number of image blocks : 3

Maximum number <n> of input blocks: 2

Arguments

```
saxs_ave [options]
    <i1nam> <onam> <i1fst> <i1lst> <i1inc> <i2fst>
    <odum> <odim1> <odim2> <i1con> <i1fac> <i2con> <i2fac>
    <ocon> <ofac>
```


saxs_angle

Transformation from cartesian coordinates to polar coordinates with pixel interpolation and azimuthal averaging.

x_1 of the output file corresponds to the radius

x_2 of the output file corresponds to the angle

The angle coordinate (x_2) uses always the real reference system (Offset, PixSiz). Center, WaveLength and SampleDistance are not used.

Number of image blocks : 2

Maximum number <n> of input blocks: 1

Arguments

saxs_angle [options]
 <i1nam> <onam> <i1fst> <i1lst> <i1inc> <odum>
 <i1cen1> <i1cen2> <r0> <dr> <dim1> <a0> <da> <dim2>

Options

- r0 [1, f]
 -r0 <minimum radius>: default 0.0
- dr [1, f]
 -dr <radial interval>: default is a value that corresponds to 1 pixel
- a0 [1, f]
 -a0 <azimuthal start angle [deg]>: default 0.0 deg
- da [1, f]
 -da <azimuthal interval [deg]>: default 1 deg
- s2 [0, +]
 +-s2 This option is used to multiply the input data with the square of the radius (in rsys units) before transformation. This allows the direct calculation of the scattering power $re^2 \cdot Q/V$ from the data by summation over i_1. (default: no multiplication)

The options asym and csym are used together with the option s2. They allow proper azimuthal averaging of intensities when the scattering pattern shows fiber symmetry and the fiber axis is lying in the detector plane. The angle of the fiber axis with the w_1 axis is defined by the option asym.

- asym [1, f]
 -asym <rotation of the symmetry axis relative to the w_1 axis [deg]>: default: 0.0 deg
- csym [0, +]
 +-csym switches on/off the multiplication of the input pattern with $Pi/2 \cdot \sin(\text{Alpha} - \text{ASym})$, where Alpha is the angle of the data point from the w_1 axis and ASym the angle of the symmetry axis from the w_1 axis (default: no multiplication).

saxs_bsl

Transformation of an edf image sequence into bsl format (daresbury otoko)

Number of image blocks : 2

Maximum number <n> of input blocks: 1

Arguments

```
saxs_bsl [options]
        <i1nam> <output header name> <i1fst> <i1lst> <i1inc>
        <odum> <odim1> <odim2>
```

Options

- hnam [1, s]
-hnam <output header name>: XNNMMM.EDF (default: automatically created from input filename)
- bsl [0, +]
+bsl : write into a single bsl binary file (default: -bsl)

saxs_cave

Averaging of a point mirrored sequence with itself.

Number of image blocks : 2

Maximum number <n> of input blocks: 1

Arguments

```
saxs_cave [options]
        <i1nam> <onam> <i1fst> <i1lst> <i1inc>
        <odum> <odim1> <odim2> <center1> <center2>
```

Options

- cen1 [1, f]
-cen1 <WCenter_1> : center 1 (input: user system coordinate, coordinate calculated according to first image)
- cen2 [1, f]
-cen2 <WCenter_2> : center 2 (input: user system coordinate, coordinate calculated according to first image)

saxs_clip

Sets all values above maxclip to maxvalue and all values that are smaller than minclip to minvalue. Works like -min and -max on the read data (after transformation and binning), but the excluded pixels are set to minimum and maximum (not to dummy).

Number of image blocks : 2

Maximum number <n> of input blocks: 1

Arguments

```
saxs_clip [options]
        <i1nam> <onam> <i1fst> <i1lst> <i1inc>
        <odum> <odim1> <odim2> <micl> <macl>
```

Options

- macl [1, f]
-macl : maximum
- micl [1, f]
-micl : minimum

saxs_col

Projection of a vertical band in a sequence of images to a single column. The output column is swapped to a line. The swapped column is written to a line of the output image.

see also saxs_row

Number of image blocks : 2

Maximum number <n> of input blocks: 1

Arguments

```
saxs_col [options]
        <i1nam> <onam> <i1fst> <i1lst> <i1inc>
        <odum> <odim1> <odim2> <col1> <col2>
```

Options

- col1 [1, f]
first column of the vertical band (input: user system coordinate, coordinate calculated according to first image)
- col2 [1, f]
last column of the vertical band (input: user system coordinate, coordinate calculated according to first image)
- row1 [1, f]
first row of the input image (input: user system coordinate, coordinate calculated according to first image)
- row2 [1, f]
last row of the input image (input: user system coordinate, coordinate calculated according to first image)
- +/-vsum [0, +]
switch between summation (+vsum) and integration/averaging (-vsum) of the data values (default: integration/averaging). The options +vint +vint cannot be set at the same time.
- +/-vint [0, +]

switch between integration (+vint) and averaging (-vint) of the data values (default: averaging).
The options +vint +vint cannot be set at the same time.

saxs_const

aliased to --> saxs_mac

saxs_csub

Subtraction of a point mirrored sequence from itself.

Number of image blocks : 2

Maximum number <n> of input blocks: 1

Arguments

saxs_csub [options]
 <i1nam> <onam> <i1fst> <i1lst> <i1inc>
 <odum> <odim1> <odim2> <center1> <center2>

Options

- cen1 [1, f]
 -cen1 <WCenter_1> : center 1 (input: user system coordinate, coordinate calculated according to first image)
- cen2 [1, f]
 -cen2 <WCenter_2> : center 2 (input: user system coordinate, coordinate calculated according to first image))

saxs_cut

Unroll an image around by radial cuts around a center. (Radial cuts through subsequent sectors).
The radial intensities are written line by line to the output file.

See also: saxs_urol

Number of image blocks : 2

Maximum number <n> of input blocks: 1

Arguments

saxs_cut[options]
 <i/p file> <o/p file> <first> <last> <inc>
 <psize1> <psize2> <center1> <center2> <Rmin>
 <DeltaR> <NR> <Alphamin> <DeltaAlpha> <NAlpha> <o/p file>

Options

- rmin [1, f]
 minimum radius (default: 0.0)

- `rste [1, f]`
radial step size (default: 1.0)
- `tmin [1, f]`
minimum azimuthal angle [deg] (default: 0.0)
- `tste [1, f]`
azimuthal step size (default: 1.0)
- `nrst [1, d]`
number of radial steps (default: 100)
- `ntst [1, d]`
number of azimuthal steps (default 360)

saxs_div

Division of two image sequences. To calculate the inverse of each image in an image sequence with the name x.edf type: `saxs_div -i1fac 0 -i1con 1 -i1nam x.edf -i2nam x.edf`

Number of image blocks : 3

Maximum number <n> of input blocks: 2

Arguments

`saxs_div [options]`
`<i1nam> <onam> <i1fst> <i1lst> <i1inc> <i2fst>`
`<odum> <odim1> <odim2> <i1con> <i1fac> <i2con> <i2fac>`
`<ocon> <ofac>`

saxs_divcol

Average columns between col1 and col2 in image sequence 2 and divide each column of image sequence 1 by the results

Number of image blocks : 3

Maximum number <n> of input blocks: 2

Arguments

`saxs_divcol [options]`
`<i1nam> <onam> <i1fst> <i1lst> <i1inc> <i2fst>`
`<odum> <odim1> <odim2> <i1con> <i1fac> <i2con> <i2fac>`
`<ocon> <ofac>`

Options

- `col1 [1, f]`
first column
- `col2 [1, f]`
last column

saxs_divrow

Average rows between row1 and row2 in image sequence 2 and divide each row of image sequence 1 by the result

Number of image blocks : 3

Maximum number <n> of input blocks: 2

Arguments

```
saxs_divrow [options]
    <i1nam> <onam> <i1fst> <i1lst> <i1inc> <i2fst>
    <odum> <odim1> <odim2> <i1con> <i1fac> <i2con> <i2fac>
    <ocon> <ofac>
```

Options

- row1 [1, f]
first row
- row2 [1, f]
last row

saxs_gnaw

Sets pixels in an ellipse around a dummy also to dummy. Similar to the general option +ogna <gnc> <gnr>.

Number of image blocks : 2

Maximum number <n> of input blocks: 1

Arguments

```
saxs_gnaw [options]
    <i1nam> <onam> <i1fst> <i1lst> <i1inc>
    <odum> <odim1> <odim2> <gnc> <gnr>
```

Options

- gnc [1, d]
horizontal radius
- gnr [1, d]
vertical radius

saxs_log

Logarithm (base 10) of the image. To calculate the logarithm to another base multiply the output with -ofac "log(<base>)/log(10)"

Number of image blocks : 2

Maximum number <n> of input blocks: 1

Arguments

saxs_log [options]
 <i1nam> <onam> <i1fst> <i1lst> <i1inc>
 <odum> <odim1> <odim2>

saxs_mac

Multiplication with a factor and addition of a constant to each image in a sequence. This routine performs a simple read and write of an image sequence. All general options are available and can be used to modify the values, e.g. to multiply with a factor and to add a constant: +ofac <fac> +ocon <con>. To convert an edf-file or bsl file to a standard edf-file with 512 byte block size and float binaries type saxs_mac old.edf new.edf or saxs_mac A01000.BSL new.edf.

Number of image blocks : 2

Maximum number <n> of input blocks: 1

Arguments

saxs_mac [options]
 <i1nam> <onam> <i1fst> <i1lst> <i1inc>
 <odum> <odim1> <odim2> <ofac> <ocon> <shft1> <shft2>

saxs_mul

Multiplication of two image sequences.

Number of image blocks : 3

Maximum number <n> of input blocks: 2

Arguments

saxs_mul [options]
 <i1nam> <onam> <i1fst> <i1lst> <i1inc> <i2fst>
 <odum> <odim1> <odim2> <i1con> <i1fac> <i2con> <i2fac>
 <ocon> <ofac>

saxs_mulcol

Average columns between col1 and col2 in image sequence 2 and multiply each column of image sequence 1 with the results

Number of image blocks : 3

Maximum number <n> of input blocks: 2

Arguments

saxs_mulcol [options]
 <i1nam> <onam> <i1fst> <i1lst> <i1inc> <i2fst>
 <odum> <odim1> <odim2> <i1con> <i1fac> <i2con> <i2fac>
 <ocon> <ofac>

Options

- col1 [1, f]
first column
- col2 [1, f]
last column

saxs_mulrow

Average rows between row1 and row2 in image sequence 2 and multiply each row of image sequence 1 with the result

Number of image blocks : 3

Maximum number <n> of input blocks: 2

Arguments

saxs_mulrow [options]
 <i1nam> <onam> <i1fst> <i1lst> <i1inc> <i2fst>
 <odum> <odim1> <odim2> <i1con> <i1fac> <i2con> <i2fac>
 <ocon> <ofac>

Options

- row1 [1, f]
first row
- row2 [1, f]
last row

saxs_normn

Normalization of an image sequence to absolute intensities (+ccd for ccd, -ccd for delay line detector, default: -ccd)

Number of image blocks : 4

Maximum number <n> of input blocks: 3

Arguments

saxs_normn [options]
 <i/p file> <scaler file> [<flatfield file>] <o/p file>
 <first image> <last image> <inc>


```

<first scaler header> <inc scaler header>
[<flatfield image>]
<o/p dummy> <o/p dim1> <o/p dim2>
<transmission>
<scaler I0 mon.> <scaler exposure time> <scaler anode counts>
<2nd scaler I0 mon.> <2nd scaler exposure time> <2nd scaler anode counts>

```

Method

The normalization is done in the following way:

$$I_{out} = \frac{I_{in}}{\text{sum}} * \frac{(anoc*anof)}{(monc-monz*time*timf)*monf} * \frac{rd}{dis} * \frac{rd^2}{pix1*pix2} * \frac{1}{trm} * \frac{1}{I_{flat}}$$

with $rd^2 = dis^2 +$
 $INDEX2R(i1,off1-cen1,pix1)^2 +$
 $INDEX2R(i2,off2-cen2,pix2)^2$

with $trm = (trmc-trmz*time*timf)*trmf / (monc-monz*time*timf)*monf$
or $trm = <option>$

Fig. 2. Normalization of a sequence of SAXS images to absolute values. The images are not divided by the flatfield image Iflat (default +flat) if the option -flat is set. The images are not corrected for dead time if the option +ccd is set. In that case the ratio anoc*anof/sum is set to 1.

sum	: sum of input image	[event]
time	: time counts	[ct]
timf	: time factor	[s]
anoc	: anode counts	[ct]
anof	: anode factor	[event/ct]
monc	: monitor counts	[ct]
monf	: monitor factor	[photon/ct]
monz	: monitor zero	[ct/s]
trmc	: transmission monitor counts	[ct]
trmf	: transmission monitor factor	[photon/ct]
trmz	: transmission monitor zero	[ct/s]
trm	: transmission	
rd	: distance sample-pixel	[m]
dis	: distance sample-detector (SampleDistance)	[m]
cen1	: horizontal beam center	[pixel]
cen2	: vertical beam center	[pixel]
off1	: horizontal offset	[pixel]
off2	: vertical offset	[pixel]
pix1	: horizontal pixel size (PixSiz_1)	[m/pixel]
pix2	: vertical pixel size (PixSiz_2)	[m/pixel]
re	: classical electron radius (RELECTRON)	= 2.81794e-15 m
Iin	: input image	[event/pixel]
Iflat	: flatfield image	[event/photon]
Iout	: output image = (1/Vol) * dSigma/dOmega	[1/m]

Fig. 3. Explanation of abbreviations

The values of time, timf etc. are stored as keywords in the header of the raw data file. To convert the output image to absolute values it has to be divided by the flat field file [event/photon].

The second scalers are only used if (count2*calib2>max_count*calib1) where max_count is defined as maxcount = 2^ScalerDepth. If the number of the second scaler is set to 0 it will not be used.

Options

- trm [1, f]
transmission (0..1)
- dpth [1, f]
eff. scaler depth in bits,
e.g. scaler depth=24 and 4 accumulations
=> -dpth "24+log(4)/log(2)" = 26
- monc [1, f]
absolute counts of I0 scaler
- monz [1, f]
zero value of I0 scaler
- monf [1, f]
factor of I0 scaler (conversion of counts to photons)
- monv [1, f]
number of incident photons

- `time [1, f]`
absolute counts of time scaler
- `timf [1, f]`
factor of time scaler (for conversion of counts to seconds)
- `timv [1, f]`
exposure time in seconds
- `anoc [1, f]`
absolute counts of anode scaler
- `anof [1, f]`
factor of anode counter (for conversion of counts to anode counts, usually 1)
- `anoz [1, f]`
zero value of anode scaler
- `anov [1, f]`
anode value in counts
- `trmc [1, f]`
absolute counts of I1 scaler
- `trmz [1, f]`
zero value of I1 scaler
- `trmf [1, f]`
factor of I1 scaler (conversion of counts to photons)
- `trmv [1, f]`
number of transmitted photons
- `mo2c [1, f]`
absolute counts of second I0 scaler
- `mo2f [1, f]`
factor of second I0 scaler (conversion of counts to photons)
- `ti2c [1, f]`
absolute counts of second time scaler
- `ti2f [1, f]`
factor of second time scaler (for conversion of counts to seconds)
- `an2c [1, f]`
absolute counts of second anode scaler
- `an2f [1, f]`
factor of second anode scaler (for conversion of counts to anode counts)
- `tr2c [1, f]`
absolute counts of second I1 scaler
- `tr2f [1, f]`
factor of second I1 scaler
- `sum [1, f]`
sum of input image 1. This option can be used to set the sum of the input image externally.

- bcf [1, f]
The option -ilbin a b averages over a*b pixels. To calculate afterwards the correct rejection the binning correction factor must be set to -bcf "a*b". The binning correction factor cancels out and has no effect on the output image data, but only on the displayed acceptance and rejection values.
- rdis [1, f]
relative distance of sample from homne position of detector displacement. If RelDis is set, the value of SampleDistance is ignored and replaced by the sum of DetectorPosition plus RelDis in input sequence 1 and 2 (i1 and i2) and in the output sequence (o). The value of DetectorPosition is read from input sequence 2.
- mons [1, d]
number of I0 monitor scaler
- tims [1, d]
number of exposure time scaler
- anos [1, d]
number of anode counts scaler
- trms [1, d]
number of I1 monitor scaler
- mo2s [1, d]
number of second I0 monitor scaler
- ti2s [1, d]
number of second exposure time scaler
- an2s [1, d]
number of second anode count scaler
- tr2s [1, d]
number of second I1 monitor scaler
- slen [1, d]
scaler length total number of available scalers (1..slen)
- flat [0, +]
flatfield correction/no floatfield correction
- ccd [0, +]
 - +ccd: ccd data (no dead time correction)
 - -ccd: delay line detector data (dead time correction)
 - default: -ccd
- tron [0, +]
 - +tron: use user supplied tranmission value (option: -trm <transmission>)
 - -tron: calculate transmission with scaler values (I1/I0)
 - default: -tron
- scal [0, +]
 - +scal: use scaler values to calculate I0, I1, exposure time and anode counts.
 - -scal: read I0, I1, exposure time and anode counts from keywords in the file
 - default +scal, automatically set to -scal if scaler information not found

saxs_patch

Replacement of the first image by the second image (where it is defined). The second image is patched into the first image.

Number of image blocks : 3
Maximum number <n> of input blocks: 2

Arguments

saxs_patch [options]
 <i1nam> <onam> <i1fst> <i1lst> <i1inc> <i2fst>
 <odum> <odim1> <odim2>

Options

- ex [0, +]
 - +ex: do not copy dummy values from image 2 to image 1
 - -ex: copy all pixels from image 2 to image 1
 - default: -ex
- dumo [0, +]
 - +dumo: only the dummy values of image 1 are patched with the values of image 2
 - -dumo: replace all values of image 1
 - default: -dumo

saxs_power

Calculate the n-th power of each image value

Number of image blocks : 2
Maximum number <n> of input blocks: 1

Arguments

saxs_power [options]
 <i1nam> <onam> <i1fst> <i1lst> <i1inc>
 <odum> <odim1> <odim2> <power>

Options

- pow [1, f]
 - power exponent

saxs_rot

Rotation of an image sequence by alfa

$$T = (A, B); A = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix}; B = \begin{pmatrix} 0.0 \\ 0.0 \end{pmatrix};$$

$$W' = A * W + B$$

with coordinates W (input image), W' (output image)

see also: saxs_aff

Number of image blocks : 2

Maximum number <n> of input blocks: 1

Arguments

saxs_rot [options] <i1nam> <onam> <i1fst> <i1lst> <i1inc> <odum> <odim1> <odim2> <alfa>

Options

- -alfa [1, f]
ccw-rotation of the image-in degree. The rotation center is the origin of the chosen reference system.

saxs_row

Projection of a horizontal band in a sequence of images to a single row. The row is written to a line of the output image.

see also: saxs_col

Number of image blocks : 2

Maximum number <n> of input blocks: 1

Arguments

saxs_row [options]
 <i1nam> <onam> <i1fst> <i1lst> <i1inc>
 <odum> <odim1> <odim2> <row1> <row2>

Options

- col1 [1, f]
first column of the input image (input: user system coordinate, coordinate calculated according to first image)
- col2 [1, f]
last column of the input image (input: user system coordinate, coordinate calculated according to first image)
- row1 [1, f]
first row of the horizontal band (input: user system coordinate, coordinate calculated according to first image)
- row2 [1, f]
last row of the horizontal band (input: user system coordinate, coordinate calculated according to first image)
- +/-vsum [0, +]
switch between summation (+vsum) and integration/averaging (-vsum) of the data values (default: integration/averaging). The options +vint +vint cannot be set at the same time.

- +/-vint [0, +]
switch between integration (+vint) and averaging (-vint) of the data values (default: averaging).
The options +vint +vint cannot be set at the same time.

saxs_scal

Read the value of a keyword from the headers of each picture and write it to a text file.

Number of image blocks : 2
Maximum number <n> of input blocks: 1

Arguments

saxs_scal [options]
 <i1nam> <onam> <i1fst> <i1lst> <i1inc> <keyword>

Options

- key [1, s]
 Keyword
- pref [1, s]
 Prefix of the output file
- ext [1, s]
 Extension of the output file
- fnam [1, s]
 Name of the output file

saxs_stat

Calculates statistics of the input image and write the result into a text file.

Number of image blocks : 2
Maximum number <n> of input blocks: 1

Arguments

saxs_stat [options]
 <i1nam> <i2nam> <o/p prefix> <i1fst> <i1lst> <i1inc>
 <selection>

Options

- sel [1, s]
 selection
 - inum Image number
 - dim1 Dimension 1
 - dim2 inferior limit 1
 - sup1 superior limit 1

- inf2 inferior limit 2
- sup2 superior limit 2
- ntot Number of data points
- nval Number of valid data points
- ndum Number of dummies
- max Maximum Value
- min Minimum Value
- amin Minimum absolute value
- dum Dummy value
- sum Sum of data values
- ssum Sum of squared data values
- mean Mean value = sum / nval
- smean Mean squared value = ssum / nval
- rms Root mean square value = sqrt(smean)
- devi Standard deviation value = smean - mean*mean
- gc1 Center of gravity 1
- pref [1, s]
Prefix of the output file
- ext [1, s]
Extension of the output file
- fnam [1, s]
Name of the output file
- show [0, +]
Print results to the standard output

saxs_sub

Subtraction of two image sequences

Number of image blocks : 3

Maximum number <n> of input blocks: 2

Arguments

saxs_sub [options]

<i1nam> <onam> <i1fst> <i1lst> <i1inc> <i2fst>
<odum> <odim1> <odim2> <i1con> <i1fac> <i2con> <i2fac>
<ocon> <ofac>

saxs_subcol

Average columns between col1 and col2 in image sequence 2 and subtract the results from each column of image sequence 1

Number of image blocks : 3

Maximum number <n> of input blocks: 2

Arguments

saxs_subcol [options]
 <i1nam> <onam> <i1fst> <i1lst> <i1inc> <i2fst>
 <odum> <odim1> <odim2> <i1con> <i1fac> <i2con> <i2fac>
 <ocon> <ofac>

Options

- col1 [1, f]
first column
- col2 [1, f]
last column

saxs_subrow

Average rows between row1 and row2 in image sequence 2 and subtract the result from each row of image sequence 1

Number of image blocks : 3

Maximum number <n> of input blocks: 2

Arguments

saxs_subrow [options]
 <i1nam> <onam> <i1fst> <i1lst> <i1inc> <i2fst>
 <odum> <odim1> <odim2> <i1con> <i1fac> <i2con> <i2fac>
 <ocon> <ofac>

Options

- row1 [1, f]
first row
- row2 [1, f]
last row

saxs_tiff

Transformation of an edf image sequence to 1byte/2byte/4byte tiff files. If more than 1 image is read from the input file the output files are numbered. The input file is first converted to an intermediate output file and then converted to a tiff file. The output dummy is never scaled. The title is written as comment into the tiff file. The output byte order is given by the machine byte order.

Hint: To convert a 16 bit unsigned integer edf-raw data file to a tiff files with 16 bps the options "-auto -bps 16" should be set. If the file contains dummy values the output dummy value should be set to a positive value inside the mapping range. If it lies outside the output range it is forced to the closest number inside the output range.

Number of image blocks : 2
 Maximum number <n> of input blocks: 1

Arguments

saxs_tiff [options]

<i1nam> <tiff file name -tnam> <i1fst> <i1lst> <i1inc>
 <odum> <odim1> <odim2> <8|16|32-bps> <1-8-ori>
 <auto> [<min> <max>]

Options

- bps [1, d]
 bits per sample in output tiff image
 <8|16|32> : 8, 16 and 32 bits in tiff image (default 16)
- ori [1, d]
 orientation of the output tiff image
 1: (x,y)
 2: (-x,y) x-axis inverted
 3: (x,-y) y-axis inverted (default)
 4: (-x,-y) both axes inverted
 5: (y,x) axes swapped
 6: (y,-x) axes swapped and x-axis inverted
 7: (-y,x) axes swapped and y-axis inverted
 8: (-y,-x) axes swapped and both axes inverted
- auto [1, +]
 -auto: As usual, the output dummy is set to -odum or its default. If not explicitly set by -min or -max, minimum and maximum of the input image are set 0 and $\text{pow}(2, \text{bps}) - 1$, all values in the range [minimum, maximum] are mapped to $[0, (\text{pow}(2, \text{bps}) - 1)]$
 +auto: The output dummy is set to upper range value $(\text{pow}(2, \text{bps}) - 1)$. If not explicitly set by -min or -max, minimum and maximum of the input image are calculated, all values in the range [minimum, maximum] are mapped to $[0, (\text{pow}(2, \text{bps}) - 2)]$ (default +auto)
 -auto should be used for format transformations, +auto for image display, e.g. with xv.
- min [1, f]
 minimum of mapping range, values smaller or equal to min will be set to 0, all values in the range [minimum, maximum] are mapped to $[0, (\text{pow}(2, \text{bps}) - 1)]$
- max [1, f]
 maximum of mapping range, value larger or equal to max will be set to $\text{pow}(2, \text{bps})$, all values in the range [minimum, maximum] are mapped to $[0, (\text{pow}(2, \text{bps}) - 1)]$

Example

saxs_tiff saxs_tiff 3.0 1999-11-12, Peter Boesecke

Input sequence 1 [input.edf] : flat.new230696

Output tiff file [flat.tif] :

First image of input sequence 1 [1] :

Last image of input sequence 1 [1] :
Increment of input sequence 1 [1] :
Output dummy [-1.000000e+00] :
Output dimension 1 [512] :
Output dimension 2 [512] :
Bits per sample (8, 16, 32) [16] :
Orientation of output image (1-8) [3] :
Range output image with minimum and maximum? [FALSE] :
Minimum value [0.000000e+00] :
Maximum value [6.553500e+04] :

input file : flat.new230696
output prefix : flat.tif
first image : 1
last image : 1
increment : 1
output dummy : -1.000000
output dimension 1 : 512
output dimension 2 : 512
bits per sample : 16
autorange : 0
minimum value : 0.000000
maximum value : 65535.000000

Reading image: flat.new230696, Image : 1 done

Calculating ihb[0, 1] = Function(ihb[1, 1])

Input file : flat.new230696
Input image number : 1
Output tiff file : flat.tif

End of /home/boesecke/programs/saxs/bin/saxs_tiff

saxs_urol

Unroll an image around a center. (Azimuthal averaging of subsequent sectors). The averaged radial intensities in each sector are written line by line to the output file.

see also: saxs_cut

Number of image blocks : 2
Maximum number <n> of input blocks: 1

Arguments

saxs_urol [options]
 <i/p file> <o/p file> <first> <last> <inc>
 <psize1> <psize2> <center1> <center2> <Rmin>
 <DeltaR> <NR> <Alphamin> <DeltaAlpha> <NAlpha> <o/p file>

Options

- rmin [1, f]
 minimum radius (default: 0.0)
- rste [1, f]
 radial step size (default: 1.0)
- tmin [1, f]
 minimum azimuthal angle [deg] (default: 0.0)
- tste [1, f]
 azimuthal step size (default: 1.0)
- nrst [1, d]
 number of radial steps (default: 100)
- ntst [1, d]
 number of azimuthal steps (default 360)

Other Routines

Programs

col2array

A single column is extracted from a set of numbered ascii-input files and are rearranged to rows of a single ascii-output file. The columns must be separated by tabs 't'. The columns are numbered with 1, 2, 3 etc. from the start and -1, -2, -3 etc. from the end. The parameters must be entered in the described order. For omitted parameters default values are used.

Arguments

col2array [-h] [-b <mode[0]>] <output> <input> <first[1]> <last[1]> <inc[1]>

<col[1]> <skip_lines[0]> <skip_chars[0]> <separator['\t']>

- output : output filename, e.g. output.txt
- input : input filename with optional #-character specifying the position of the number, e.g. input_###.txt. Leading positions are filled with zeros.
- first : number of the first file (default: 1)
- last : number of the last file (default: first)
- inc : increment (default: 1)
- col : column number (default: 1)
- skip_lines : number of lines to skip at the top (default: 0)
- skip_chars : number of characters to skip (default: 0).

Options

- -b <mode>: 0->text (default), 6->Signed32; 9->FloatIEEE32
- -h : show this help

Example

The command

```
>> col2array output.txt input_###.txt 105 095 -3 -1 2 0
```

will read the last column of the files input_105.txt, input_102.txt, input_099.txt and input_096.txt and write it to output.txt. It will skip 2 lines and 0 characters at the start of each file.

Macros

Macros can only be used if the path to the saxs programs is defined in the shell initialization script (e.g. ".cshrc"). The initialization script must not change the directory.

saxsdisp.mac

A single image of an edf file is converted into 8 bps tiff format and displayed with xv. The output image is automatically scaled between 0 and 255. Dummy values are set to 255.

Arguments

saxs_disp <input> [<num>]

- input: input file name
- num: image number (optional) (default: first image)

Index

/

/dev/null, 9

A

a, a+fp, 6
 a+ip, 6
 addition of images, 14
 additional information, 12
 affine transformation, 15
 argument, 4
 arithmetic expression, 6
 arithmetic function, 7
 array, 6

B

bibo (option), 8
 binary2saxs, 12
 boolean value, 6
 bsl format (daresbury otoko), 18

C

cartesian coordinate, 17
 ccd2saxs, 13
 center, 6
 col2array, 36
 constant, 7
 conversion to standard edf file, 23

D

dimension, 11
 display of an edf image with xv, 37
 dummy, 11
 dummy file, 9

E

extraction of a column, 36

F

file access, 9
 file opening mode, 6
 float function, 7
 float value, 6

G

gas2saxs, 13
 gblk (option), 8
 gel2saxs, 13
 general options, 8

H

h (option), 8

I

image, 6
 image number, 9
 integer value, 6

L

long integer function, 8
 loop control, 9

M

memory number, 9
 mlw (option), 8

N

n, n+fp, 6
 n+ip, 6
 normal, 6

O

o, o+fp, 6
 o+ip, 6
 obin (option)
 ilbin (option), 11
 ocen (option)
 ilcen (option), 12
 ocon (option)
 ilcon (option), 11
 oddum (option)
 ilddum (option), 11
 odim (option)
 ildim (option), 11
 odis (option)
 ildis (option), 12
 odum (option)
 ildum (option), 11
 ofac (option)
 ilfac (option), 11
 ofst (option)
 ilfst (option), 10
 oga (option)
 ilgna (option), 11
 oinc (option)
 ilinc (option), 10
 olst (option)
 illst (option), 10
 omax (option)
 ilmax (option), 11
 omem (option)
 ilmem (option), 9
 omin (option)
 ilmin (option), 10
 omod (option)
 ilmod (option), 9
 onam (option)
 ilnam (option), 9
 ooff (option)
 illoff (option), 12

oori (option)
 ilori (option), 11
 opix (option)
 ilpix (option), 12
 option, 4
 option value, 6
 otim (option)
 iltim (option), 12
 otit (option)
 ilit (option), 12
 owvl (option)
 ilwvl (option), 12

P

p (option), 8
 polar coordinate, 17
 projection, 30
 prompt, 3

R

r (option), 8
 real, 6
 reference system, 6
 reference system parameters, 12
 reference system value, 6
 rotation, 29
 rsys (option), 8

S

saxs, 6
 saxs_add, 14
 saxs_addcol, 14
 saxs_addrow, 14
 saxs_aff, 15
 saxs_angle, 17
 saxs_ascii, 14–15
 saxs_ave, 16
 saxs_bsl, 17–18
 saxs_cave, 18
 saxs_clip, 18
 saxs_col, 19
 saxs_const, 20

saxs_csub, 20
 saxs_cut, 20
 saxs_div, 21
 saxs_divcol, 21
 saxs_divrow, 21–22
 saxs_gnaw, 22
 saxs_log, 22
 saxs_mac, 23
 saxs_mul, 23
 saxs_mulcol, 23
 saxs_mulrow, 23–24
 saxs_normn, 24
 saxs_patch, 28–29
 saxs_power, 29
 saxs_rot, 29
 saxs_row, 29–30
 saxs_scal, 31
 saxs_stat, 31
 saxs_sub, 32
 saxs_subcol, 32
 saxs_subrow, 33
 saxs_tiff, 33
 saxs_urol, 32–36
 shift (option), 8
 specific options, 12
 string value, 6

T

test (option), 8
 tiff (tagged image file format), 33
 transformation of images, 10
 transformation to ASCII, 15
 transformation to bsl, 18
 transformation to tiff, 33

U

units, 7
 usys (option), 8

X

xv, 37