

DUBBLE

BM26B (SAXS/WAXS)

USER'S SURVIVAL MANUAL

V. 2.41

Last update: 22 Nov. 2010



Table of Contents

Preface.....	3
Useful SPEC commands	4
SAXSGUI troubleshooting	5
Pilatus 300K-W as WAXS detector.....	9
Data reduction	11

Preface

Dear BM26B user,

This little manual is intended to help you in running the beamline during your experiments at DUBBLE.

In it, you can find a basic description of the SPEC commands you need for samples alignment and some commands for using the detectors available at DUBBLE. Some useful tricks about how to solve small problems like software crashes are described in it as well.

The beam line, sample environments and other auxiliary equipment are in a continuous evolution. This also means that this manual is in continuous evolution and maybe not always fully updated. If in doubt, please check timely with the beam line scientists.

However good, bad or not updated a manual might be, the use of common sense is always a good thing.

Good luck with your measurements,

The beamline scientist

Useful SPEC commands

List and examples on how to use SPEC commands in the SAXS session at DUBBLE:

- **shopen** = opens the safety shutter to have X-ray beam
- **shclose** = closes the safety shutter
- **wa** = shows all the motors positions
- **wm** motor_name = show position of a single motor
- **mv** motor_name absolute_position = move the selected motor to a well defined position. Positions are usually in mm or degrees.

Ex. mv samplex 7

Ex. mv samplex CEN (it moves the motor to the estimated centre by using a Gaussian fit. DO NOT USE IF THE OUTPUT OF THE SCAN IS NOT EASILY DESCRIBED BY A GAUSSIN or BOX FUNCTION).

- **mvr** motor_name quantity = move relatively the selected motor of the specified quantity from the position you are.

Ex. mvr sampley -5 (moves sampley down of 5 mm)

- **dscan** motor_name from to nb_points time_per_point (in seconds) = perform a relative scan of the selected motor of the defined quantities in negative and positive directions. The resolution of the scan is defined by the number of points.

Ex. dscan sampley -1 1 50 0.1

- **ct** [time in seconds] = counts all the counters for the selected time

Available counters usually used are:

Seconds → time in seconds

i1 → incident intensity

i2 → transmitted intensity

intens → scattered intensity at the SAXS 2D multiwire detector

photo → transmitted intensity measured with photodiode integrated in the beamstop

- **plotslect** counter = plots the values of the selected counter

Ex. plotslect i2; plot

- **timescan** [time_per_point] = counts in time the selected counter values

SAXSGUI troubleshooting

This section explains in a few lines how to start and to quit the program in a proper way, and proposes do-it-yourself solutions for the most common problems using SAXSGUI.

It is assumed that you log in on dubble3 (the HP machine) as 'opd26'.

SAXSGUI basics

1. Proper starting up the program

Note: SAXSGUI is an IDL application and needs an IDL licence program to run simultaneously in a separate terminal/console. Don't worry about this now. If there is something wrong with the licence, you will notice it, since SAXSGUI will most probably crash e.g. when you click on "Inhouse" in the very first window. In such a case, refer to the "troubleshooting" section.

Starting SAXSGUI

- In a new (or available) terminal, type `saxsgui` in the command line.

ATTN: you will not be able to use that terminal for other actions afterwards. It will just display information about the evolution of the program.

- Click "Inhouse" in the first "SAXS/WAXS" window that rises up.
- Click "Saxs 2D" in the second "SAXS/WAXS" window.

This will open the main window for "SAXS 2D".

ATTN: don't close the small window called "Managed Widgets" that appears at the same time. Just ignore it. It will anyway be hidden by the main window once you start using the program. If you press "Kill Widget", the program will be killed immediately.

- Clear system memory (menu *Run / Clear memory*).

The operation is done when the cursor blinks and in the xterm you see something like:

```
Timeout: 295
0, 0, 0, 254, 0, 0, 0, 260, 261, 0, 0, 295, 256, -2147483648, 0, 0, 0, 0, 0
```

- Start the online display (menu *Online / Online 2D*);
- In ImageDisplay, go in "*Command / Colormap Editor*" and choose "*Logarithmic*";
- Back in the main window, go in "*Colors / Select colortable*" and choose "STERN SPECIAL", the click "*Done*". The system is now ready to use.
- Check (menu *File / Select OUTPUT directory*) that the output directory corresponds to your user identifier (e.g. `/users/opd26/data/inhouse/users/user512`).

- Check (menu *Fetch / Set OUTPUT filename*) the prefix for the output files.
- Menu "*Setup / Timeframe editor*" allows one to indicate the time scheme (dead time and life time) for each frame (or set of frames) to be acquired.
- Start acquisition with "*Run / Start*".

2. Proper exiting the program:

- First, quit the ImageDisplay (menu *File /Quit*),
- Then exit the main program (menu *File / Exit* in window SAXS 2D). This should end all operations in the xterm from which the program started, and give you back the access to the command line.

SAXSGUI troubleshooting

Here is a list of the most common problems together with their solution.

P: SAXSGUI crashes when I click on "Inhouse" in the very first window.

S: The IDL licence server is probably off. You can check this (on dubble3) by typing:

```
ps -ef | grep lmgrd
```

If there is no licence running, you will see only the "grep" process:

```
opd26 10528 10505 1 11:33:19 ttyp2 0:00 grep lmgrd
```

The meaning of the columns is: user, pid (process identifier), ppid (parent process identifier), cpu, starting time, terminal, time, process (with arguments).

Start the IDL licence:

- in a new (or available) terminal, go to your home directory (/users/opd26/).
Note that you can check at any time the current directory with "pwd".
- type: IDLLICENSE

ATTN: you will not be able to use that terminal for other actions afterwards.

Minimize it and ignore it.

Note: in normal operation, the `ps -ef | grep lmgrd` command yields two (and only two) `lmgrd`-related processes: "`lmgrd`" itself and "`idl_lmgrd`", e.g.:

```
opd26 7343 7342 0 May 2 ? 0:02 /usr/local/rsi/idl/bin/bin.hp/idl_lmgrd -T
Dubble3 6.1 3
opd26 7342 1 0 May 2 pts/6 0:00 /usr/local/rsi/idl/bin/bin.hp/lmgrd
opd26 10528 10505 1 11:33:19 ttyp2 0:00 grep lmgrd
```

Note that process 7342 is the parent of 7343.

If for some reason one of these processes is missing or if there are more than two `lmgrd`-related processes, kill them all, one by one, with "kill":

```
kill -TERM 7343
```

then restart the licence manager properly as explained above.

If `kill -TERM` doesn't work, use `kill -9`.

P: When I try to start the online display in Saxs 2D (*Online / Online 2D*), I get the error message "Access to memory is disabled, try again..."

S:

- click OK in the Error window,
- click "Kill Widget" in the small window that appeared at the same time
- clear system memory (*Run / Clear memory*, then 'Yes' in the *Question* window)

The operation is done when the cursor blinks and in the xterm you see something like:

```
Timeout: 295
```

```
0, 0, 0, 254, 0, 0, 0, 260, 261, 0, 0, 295, 256, -2147483648, 0, 0, 0, 0, 0
```

- start again the online display (*Online / Online 2D*)

P: SAXSGUI doesn't react anymore, possibly with an error message like

"COMMUNICATION ERROR WITH DEVICE SERVER !"

S: Open a terminal window on `dubble9` and Connect remotely on `dubble3` via telnet:

```
dubble9:~ % telnet dubble3
```

```
login: opd26
```

```
Password: ton....
```

On `dubble3`, look for all the `idl`-related processes:

```
ps -ef | grep idl
```

You will get a list like:

```
opd26 27685      1  0 12:53:04 pts/2      0:00
/usr/local/rsi/idl/bin/bin.hp/lmgrd
opd26 27763      1  0 13:00:53 pts/0      0:00 dtterm -iconic -tn vt100 -e
/usr/local/bin/idl /users/dubble
opd26 27686 27685  0 12:53:04 ?          0:00
/usr/local/rsi/idl/bin/bin.hp/idl_lmgrd -T Dubble3 6.1 3
opd26 27765 27763  0 13:00:53 pts/1      1:18
/usr/local/rsi/idl/bin/bin.hp/idl /users/dubble/idl/saxswaxs
opd26 27691 27494 17 12:53:12 pts/0      6:21
/usr/local/rsi/idl/bin/bin.hp/idl /users/dubble/idl/saxswaxs
opd26 27913 27890  0 14:56:26 tty2      0:00 grep idl
```

Kill the "saxswaxs" process having the smaller identifier (27691 in this example):

```
kill -TERM 27691
```

Note: this process is the "parent" of the other "saxswaxs" process (27765).

Killing the parent process will kill the child process as well. If it's not the case, kill each process separately.

Note: if "kill -TERM" doesn't work, use "kill -9".

A new check with "ps -ef | grep idl" should yield only 3 lines:

one with "lmgrd", another with "idl_lmgrd", and the last with "grep idl".

ATTN: don't kill 'lmgrd' nor 'idl_lmgrd' processes. If this happens, restart the IDL licence server as explained at the beginning of this troubleshooting section.

If this cleaning action is not enough to make SAXSGUI run properly, and/or if the top LED of the HIST module (in the uppermost VME crate in rack #10 in the control cabin) is red, reboot the crate (press the RESET button on the MVME 162-552A card).

You can follow the reboot sequence on the 15" screen placed inside rack #11.

Among the last lines of the sequence one can see:

```
Maxe server 5 started
+74
Dld server started
+76
Vtdc4 server started
----- start.ds done -----
etc.
```

Once "unit D262" is operational again, start SAXSGUI as explained above.

Pilatus 300K-W as WAXS detector

In order to use the Pilatus 300K-W as wide angle (WAXS) detector together with the acquisition system for SAXS data (GDA or SAXSGUI), make sure that the fast shutter is closed and follow the steps indicated below.

Start the Pilatus 300K-W software

1. Start remotely the acquisition software on PC B:

- on PC *dubble9*, go to Desktop 3, and open a new shell (Terminal or Konsole)
- connect first to Pilatus PC A: `ssh -X det@dec058A`
- (password: Pilatus2)
- connect to PC B: `ssh -X det@dec058B` (use the same password)
- `cd p2_det`
- `runtvx`

Wait till you get in the *tvx* window the message *disconnect*. The two detector modules driven by PC B are now ready to use, and two calibration images (all green and all blue) are displayed.

2. Start the acquisition software on PC A:

- on PC *dubble9*, go to Desktop 2, and open a new shell (Terminal or Konsole)
- connect to PC A: `ssh -X det@dec058A`
- (password: Pilatus2)
- `cd p2_det`
- `runtvx`

Wait till you get in the *tvx* window the message *disconnect*. The three detector modules are now loaded and calibrated. The corresponding green and blue images are also displayed.

N.B. the Pilatus detector has to be started without X-rays on it. The fast shutter should be closed.

3. Select the output image folder on PC A, both in camserver and in tvx windows:

- in *camserver* of (PC A): `imgpath [folderpath/foldername]`
- in *tvx* (PC A): `imagepath [folderpath/foldername]`

Data acquisition

When using the Pilatus 300K-W, a fast shutter has to be installed as well. Both the fast shutter and the Pilatus 300K-W are triggered by the GDA (or SAXSGUI) software.

Note: all the commands to the detector have to be given to PC A in Desktop 2.

PC B is automatically configured by PC A.

In the *camserver* window of PC A (Desktop 2), set the number of images to acquire (*ni*) and the *dma_hold* value if any.

Remember: $dma_hold = runtime + deadtime - 1 \text{ or } 2 \text{ seconds}$

To start the acquisition, type in *camserver*

exte filename_.edf

Ex. Say you want to acquire multiframe data and you set in the timeframe generator of GDA (or SAXSGUI) 20 images with a runtime of 55s and a deadtime of 5s between the images. In this case, type in *camserver*:

ni 20

dma_hold 58

exte exp1_.edf

(start the SAXS acquisition immediately after the command *exte*)

Once the *exte* command is sent, the detector will wait the TTL pulse from the SAXS software. Start the SAXS acquisition and enjoy the measurements.

Remember: fast shutter: opening and closing time 7ms

Pilatus: reading time 4ms in normal acquisition mode

To stop the data acquisition on the Pilatus system, type *cam k* and press Enter in *tvx* of PC A.

DO NOT PRESS *Ctrl+C* otherwise you'll kill the windows.

Pilatus troubleshooting

1) If once you set *dma_hold*, the acquisition fails for some reason, it is possible that PC B is not synchronized anymore. You eventually will get a dma timeout error. In this case close the Pilatus applications in Desktop 2 and Desktop 3 and restart them from the beginning.

2) sometimes the problem is merely that you give commands to PC B instead of PC A.

Data reduction

To make your data processing easier, we provide the essential software tools for data reduction.

SAXS data reduction

The processing of the raw SAXS patterns consists of a series of corrections, followed by an azimuthal integration between specified limits (this operation is outsourced to program *fit2d*). Specific Matlab functions have been created for *BSL*, *edf*, and *tiff* data.

The following corrections are applied to the SAXS images:

- normalization of the image with respect to the intensity of the incident beam,
- correction for sample transmission and background subtraction,
- masking of unwanted pixels,
- azimuthal integration (using *fit2d*),
- intensity calibration in cm^{-1} (if used),
- invariant calculation.

In order to use the programs, as a preliminary,

- connect to a *coral* machine with your visitor login and password (given on the A form)
e.g. `ssh -X b262500@coral`
- open a Matlab session: `matlab -nodesktop`
- type: `path(path, '/users/apostol/matlab')`

- For *BSL* images acquired with the GDA system, use function *azi_avg_bsl_gda*.

Syntax: `azi_avg_bsl_gda(infoFile,sample,ext,[bunch])`

Type `help azi_avg_bsl_gda` to get explanations about the input parameters.

Square brackets indicate optional parameter(s).

The local contact can provide you with an example *infoFile* that you can customize.

Example: `azi_avg_bsl_gda('par_saxs_user600.info','A13','708')`

This starts the processing of all the frames collected for sample *A13000.708*, image by image.

If you want to bunch a few (e.g. two) consecutive images together, use:

```
azi_avg_bsl_gda('par_saxs_user600.info','A13','708',2)
```

Note that string parameters are given between ‘’, whereas numerical values are entered as such.

- For *BSL* images acquired with the old Saxsgui system (running on the HP machine *dubble3*), use:

```
azi_avg_bsl(infoFile,sample,ext,[bunch])
```

As you can see, the syntax is practically the same as for the GDA version.

- Note that it is possible to perform azimuthal integration on all the *bsl* files in a folder using:

azi_avg_bsl_folder(infoFile,gda,[bunch])

- For *edf* images (acquired e.g. with the FReLoN camera), use:

azi_avg_edf_series(infoFile,EdfFile1,[EdfFile2,bunch])

In this case, *EdfFile1* and *EdfFile2* are the first (and the last) *edf* SAXS image(s) to be processed.

It is also possible to perform azimuthal integration on all the *edf* files in a folder using:

azi_avg_edf_folder(infoFile)

- for *tiff* images (collected e.g. with a MarCCD detector), use:

azi_avg_tif_series(infoFile,TifFile1,[TifFile2,bunch])

The same conventions apply as for the *edf* format.

One can process all the *tiff* files in a folder using:

azi_avg_tif_folder(infoFile)

The program will generate three types of output:

- a *.chi* file for each image, that contains q and $I(q)$ (2 columns, ASCII),
- a *.out* file that contains q and $I(q)$ for all the images in the series (multi-column, ASCII),
- a *_Inv.txt* file that contains the invariant values for each frame (2 columns, ASCII).

For WAXS data the file ends with *_Int.txt* and contains the integrated intensity.

Your local contact will help you with the first steps in using the programs.

WAXS data reduction

WAXS data are processed using the XOP package in order to correct the geometrical distortion due to detector's position, and to calculate the azimuthal integral. XOP supports *edf* and *tiff* data formats.

As a preliminary, with the help of your local contact,

- - connect to a *coral* machine with your visitor login and password (given on the A form)
e.g. *ssh -X b262500@coral*
- start XOP as: */scisoft/xop2.3/xop -d*

This starts the graphical application and also gives access to an IDL terminal.

- In this terminal, compile the following scripts (note the '.' before *run*):

.run readInfoFile readBslHead azi_integ azi_integ_series azi_integ_folder

Program *azi_integ* performs the calibration and the azimuthal integration of a single WAXS image:

azi_integ, infoFile, imaFile, [bslPref, bslExt, frame]

imaFile is –of course– the WAXS pattern to be analyzed.

The *infoFile* should indicate the following parameters: *direc*, *energy*, *beamX*, *beamY*, *sampleX*, *sampleY*, *sampleZ*, [*darkFile*, *waxsBkgd*, *saxsBgPref*, *saxsBgExt*, *bkgdCoef*], *IntenCal*, *outFormat*.

- *direc* is the data folder
- *beamX* and *beamY* give the position of the direct beam on the detector (in pixel units);
- *sampleX*, *sampleY*, and *sampleZ* indicate the sample position in detector reference system (in pixels);
- [*darkFile*] is the filename of the dark current image (if any);
- [*waxsBkgd*] is the filename of the WAXS background image (if any);
- [*saxsBgPref*] is the prefix of the SAXS background image (if any);
- [*saxsBgExt*] is the extension of the SAXS background image (if any);
- [*bkgdCoef*] is a coefficient for the background image (default: 0);
- *outFormat* is to be chosen among *Int*, *q-Int*, *2th-Int*, and *all*. Thus, the output file will contain: only intensity values (one column), or q (in \AA^{-1}) and intensity values (two columns), two-theta (in deg.) and intensity values (two columns), or the set two-theta, d -spacing (in \AA^{-1}), q and intensity (four columns).

The last three parameters [*bslPref*, *bslExt*, *frame*] are used to specify the SAXS image acquired at the same time as the WAXS one. From that BSL frame, the program reads the corresponding intensity of the incident beam (I1), and normalizes the WAXS image with respect to it.

To process a series of images use *azi_integ_series*:

azi_integ_series, infoFile, file1, [file2, nDig, bslPref, bslExt]

where: - *infoFile* is as explained above,

- *file1* and *file2* are the first and the last files in the series,

- *nDig* is the number of digits in the filenames,

- *bslPref* and *bslExt* specify the corresponding SAXS (BSL) data.

If all the optional parameters (*file2*, *nDig*, *bslPref*, *bslExt*) are missing, *azi_integ_series* just calls *azi_integ* for *file1*, without any normalization by I1.

It is also possible to process all the files in folder *direc* that have the extension *ext* (without '.'):

azi_integ_folder, infoFile, ext

direc is the data folder indicated in *infoFile*. As for *ext*, remember that XOP accepts *edf* or *tiff* data.

Note however that the normalization by I1 is not possible in this case.