SPHERE/ZIMPOL IDL Pipeline Manual

A. Bazzon, E. Buenzli, H.M. Schmid, S. Hunziker, ETH Zürich
Version 1.5, 29.10.2018

main recipes in /home/sphere/szlib/szdef.pro
aux recipes in /home/sphere/szlib/szaux.pro
start file in /home/sphere/szlib/sz_start.pro

A) Installation

B) Help function

C) Standard data reduction

1. General comments (automatic vs. step-by-step reduction)
2. Main data reduction steps
3. Description of main recipes and standard calling sequence

D) Standard Options

1. FNAME
2. NORMALIZE even/odd frames
3. FRAMESELECT
4. bad pixel filtering
   3.1 median clipping,
   3.2 sigma clipping
5. centering options
   5.1 NCENTER
   5.2 DOCENTROID
   5.3 RADIUS
   5.3 GCNTRD
   5.4 MPFIT
6. skipping options
   6.1 NOPREPROC and FPOL
   6.2 SKIPCAL
   6.3 SKIPCLMEAN
   6.4 SKIPSIGMA
   6.5 SKIPCENTER
   6.6 SKIPMERGE
7. MERGE
   7.1 USEMEDIAN
   7.2 NOCOLLAPSE
8. Derotation options
   8.1 NODEROT
   8.2 WRITE_DEROT
9. PMETHOD

E) Advanced Options

1. parallel processing (PIDL and NPIDL)
2. frame transfer smearing correction
3. beamshift calculations

4. injecting fake signals
5. SUBFRAME
6. MASK
7. P1PUPIL and IGNOREALT
8. SDI (spectral differential imaging)
9. SADI (simple ADI)
10. PCA
11. daytime calibrations (CALIB)
12. INSTRPOL
13. CAMALIGN

F) Overview of file types

A) Installation

The pipeline is defined in three .pro files:
- szdef.pro (main definition file)
- szaux.pro (auxiliary files)
- sz_start.pro (used to compile all pipeline receipes and define global variables)

Furthermore, the pipeline requires the astrolib IDL library.

1. save szdef.pro and szaux.pro where IDL can find it (e.g. current directory or in a directory this is in your private IDL search path)
2. To be able to use the built-in help function, find sz_help in szaux.pro (very end) and adapt the "szdefdir" parameter to the directory of your szdef.pro.
3. pre-compile all recipes defined in szdef.pro and szaux.pro by running sz_start.pro

Note: the parallel processing features of the pipeline require an IDL startup file which contains the path information for szdef.pro and szaux.pro, and which executes sz_start.pro upon every start of IDL (see Section E.1 for info on parallel processing).

Note: the parallel processing uses IDL_IDLBRIDGE. It was reported that for some IDL installations there is a problem when calling IDL_IDLBRIDGE::EXECUTETIMER. Usually this file is stored in the IDL installation directory
(e.g. /usr/local/itt/idl71/lib/bridges/idl_idlbridge__executetimer.sav)
and we noticed that for some IDL installations this file is missing. A potential fix would be to copy this file from a working installation but this was never tested. Note that the pipeline is fully functional without parallel mode and will simply be slower.

B) Help function (sz_help)

```
IDL> sz_help, /recipes          (list main recipes)
IDL> sz_help, string            (short info about specific recipe, e.g. 'sz_science')
IDL> sz_help, /ALL              (short info about all main recipes)
IDL> sz_help, /FILETYPES        (list all produced file types)
IDL> sz_help, /VERSION          (show pipeline version)
```

C) Standard data reduction

1. General comments

There are 2 approaches:

- *step-by-step reduction for calibration files (recommended)*
- *fully automatic reduction*

The pipeline philosophy is the same as the ESO DRH philosophy. The files to be reduced need to be written into a .sof file, which is a textfile with two columns containing in each line the filename of the individual raw or master file and its corresponding filetag. This .sof file is then called by the pipeline recipes. Of course, the advanced user can also use some of the standalone sub-routines of the pipeline which act on a single file (e.g. sz_preproc_basic, 'file.fits'). It is also possible to first try to run the pipeline automatically, check the intermediate products, and re-run with adjusted keywords as necessary, skipping the steps that already worked well.

Step-by-step reduction for calibration files (recommended)
Master calibration files can be created using the individual calibration recipes before proceeding with the science recipe. This allows assigning a particular name to the master calibration files to assure that master calibration files will not be overwritten (for example if files for multiple filters are located and processed in the same directory). It also provides an option for sigma clipping (not available for calibration files in fully automatic reduction). After creation of the master calibration files, proceed with the automatic reduction but add the master calibration files instead of the raw calibration files to the .sof file for the science reduction.

Automatic reduction
There are two main science recipes: *sz_science* for data taken in polarimetric mode, and *sz_science_imaging* for data taken in imaging mode. The .sof file for these two recipes can contain raw or master calibration files and raw science files. First, the recipes check if master calibration files are available. If no corresponding master files are present they check for raw calibration files and create master files (with default names, e.g. mbias1.fits, etc) . If also no raw calibration files are present they give a warning but continue the data reduction without calibration files (except overscan bias subtraction).

The .sof files for science reduction can contain files with the following filetags:

imaging:
- ZPL_SCIENCE_IMAGING_RAW
- ZPL_MASTER_BIAS_CAM1 (or ZPL_BIAS_RAW)
- ZPL_MASTER_BIAS_CAM2 (or ZPL_BIAS_RAW)
- ZPL_MASTER_DARK_CAM1 (or ZPL_DARK_RAW)
- ZPL_MASTER_DARK_CAM2 (or ZPL_DARK_RAW)
- ZPL_INT_FLAT_FIELD_MASTER_CAM1 (or ZPL_INT_FLAT_FIELD_RAW)
- ZPL_INT_FLAT_FIELD_MASTER_CAM2 (or ZPL_INT_FLAT_FIELD_RAW)

polarimetry
- ZPL_SCIENCE_P1_RAW (if obs = P1)
- ZPL_SCIENCE_P23_RAW (if obs = P2)

ZPL_MASTER_BIAS_CAM1 (or ZPL_BIAS_RAW)
ZPL_MASTER_BIAS_CAM2 (or ZPL_BIAS_RAW)
ZPL_MASTER_DARK_CAM1 (or ZPL_DARK_RAW)
ZPL_MASTER_DARK_CAM2 (or ZPL_DARK_RAW)
ZPL_MODEM_EFF_CAM1 (or ZPL_MODEM_EFF_RAW)
ZPL_MODEM_EFF_CAM2 (or ZPL_MODEM_EFF_RAW)
ZPL_INT_FLAT_FIELD_MASTER_CAM1 (or ZPL_INT_FLAT_FIELD_RAW)
ZPL_INT_FLAT_FIELD_MASTER_CAM2 (or ZPL_INT_FLAT_FIELD_RAW)

Note: At the beginning of the line, a # can be used to comment out that line, for example if a whole exposure should be ignored

2. Main data reduction steps

These are the main data reduction steps applied by the **calibration** recipes:

1. preprocessing (sz_preproc, splits files into ndit and even/odd frames)
2. calculate polarization signal, for polarimetry mode only (sz_pol calculates twophase frames I, Q)
3. apply calibration and de-dither (sz_apply_calib, calib applied frame-by-frame)

optional:
- *median or sigma-clipping*

4. collapse (sz_collapse)

These are the main data reduction steps applied by the **science** recipes:

1. preprocessing (sz_preproc, splits files into ndit and even/odd frames)

optional:
- *insert fake signal*

2. calculate polarization signal, for polarimetry mode only (sz_pol calculates twophase frames I, Q)

optional:
- *apply a frame transfer smearing correction for intensity*
- *calculate calculate and correct beamshift*

3. apply calibration and de-dither (sz_apply_calib, calib applied frame-by-frame)

optional:
- *frame selection*
- *select subframe*
- *median or sigma-clipping*
- *centering* (optional but required for pupil stabilized imaging or P1 polarimetry!)
- *masking*
- *rotate P1 into pupil stabilized orientation*
- *merge into a single cube (recommended if frame selection is done and number of total frames is not too large, cube can also be used for further processing with external tools, in particular if pupil stabilized imaging was performed)*
- *simple ADI*

- *PCA*

4. collapse and de-rotation (sz_collapse, sz_derot)

optional:
- *camalign, (usually not needed if centering was done)*

Note:
Steps 1-3 are called *"basic data reduction steps"* and the **cal_*.fits** files are produced. These files are calibrated and still contain all frames and therefore they are also well suited as input for other advanced data reduction pipelines (e.g. ADI) or individual analysis.

Note:
Preprocessing only rearranges the data (splits the two cameras and the even/odd lines) into a more user friendly data format ([*,*,ndit, 2], whereas the 2 denotes even/odd for polarimetry and open/covered for standard imaging). For each camera a separate .fits file is created (<orig>_cam1.fits, <orig>_cam2.fits).
Note:
Most of the options create new files and updated internal .sof files which are used by the pipeline. The files have new filenames given a prefix corresponding to the option (e.g. sclip_<file> for sigma-clipping). When using the skipping options (see below) one has to be careful that all these files exist and have not been changed. Whenever, an optional step has been changed also all the subsequent steps need to be repeated (see execution tree below)

3. Description of main recipes and standard calling sequence

3.1 Create master calibration files:

1. sz_bias
2. sz_dark
3. sz_flat
4. sz_modem

Note: for all master calibration files, the sigma clipping options (keywords sigmaclip and sigmaval or clmean) can be used (see details in Section D. 4)

---

sz_bias

same recipe for imaging and polarimetry (type is recognized from header)

sof file:
file1.fits ZPL_BIAS_RAW
file2.fits ZPL_BIAS_RAW
...

standard calling sequence:
IDL> sz_bias, 'bias.sof' (,FNAME='<name>')

Default FNAME: mbias1.fits, mbias2.fits

For polarimetry the bias is applied on the 0even/0odd and PIeven/PIodd frames

---

sz_dark

same recipe for imaging and polarimetry (type is recognized from header)

sof file:
file1.fits ZPL_DARK_RAW
file2.fits ZPL_DARK_RAW
mbias1.fits ZPL_MASTER_BIAS_CAM1
mbias2.fits ZPL_MASTER_BIAS_CAM2
...

standard calling sequence:
IDL> sz_dark, 'dark.sof', (,FNAME='<name>')

Default FNAME: mdark1.fits, mdark2.fits
For Polarimetry the dark is applied on the 0even/0odd and PIeven/PIodd frames

---

sz_flat

same recipe for imaging and polarimetry (type is recognized from header)

sof file:
file1.fits ZPL_INT_FLAT_FIELD_RAW
file2.fits ZPL_INT_FLAT_FIELD_RAW
mbias1.fits ZPL_MASTER_BIAS_CAM1
mbias2.fits ZPL_MASTER_BIAS_CAM2
mdark1.fits ZPL_MASTER_DARK_CAM1
mdark2.fits ZPL_MASTER_DARK_CAM2
...

standard calling sequence:
IDL> sz_flat, 'flat.sof' (,FNAME='<name>')

Default FNAME: mflat1.fits, mflat2.fits

Note: for calculation in polarimetry mode, flat is only calculated from I-image which will also be used for Q.

---

sz_modem

only for polarimetry

sof file:
file1.fits ZPL_MODEM_EFF_RAW
file2.fits ZPL_MODEM_EFF_RAW
mbias1.fits ZPL_MASTER_BIAS_CAM1
mbias2.fits ZPL_MASTER_BIAS_CAM2
mdark1.fits ZPL_MASTER_DARK_CAM1
mdark2.fits ZPL_MASTER_DARK_CAM2

...

standard calling sequence:
IDL> sz_modem, 'modem.sof'(,FNAME='<name>')

Default FNAME: modem1.fits, modem2.fits

Note: optionally choose /DETMEAN to use a spatially constant mean efficiency.

3.2 Create science files:

---

sz_science_imaging (for imaging data)

sof file:
file1.fits ZPL_SCIENCE_IMAGING_RAW
file2.fits ZPL_SCIENCE_IMAGING_RAW
mbias1.fits ZPL_MASTER_BIAS_CAM1
mbias2.fits ZPL_MASTER_BIAS_CAM2
mdark1.fits ZPL_MASTER_DARK_CAM1
mdark2.fits ZPL_MASTER_DARK_CAM2
mflat1.fits ZPL_INT_FLAT_FIELD_MASTER_CAM1
mflat2.fits ZPL_INT_FLAT_FIELD_MASTER_CAM2
...

standard calling sequence:
IDL> sz_science_imaging_, 'science.sof' (,FNAME='<name>', … other keywords)

Defaults:
FNAME: sci1.fits, sci2.fits

Field stabilized mode or pupil stabilized mode is recognized from header. Use option /NODEROT to skip derotation for pupil stabilized mode.

Files produced:

<file*>_cam1.fits	(preprocessed files for each fits file cam1)
<file*>_cam2.fits	(preprocessed files for each fits file cam2)
cal_OBS*_cam1.fits	(calibrated frames for each fits file cam1)
cal_OBS*_cam2.fits	(calibrated frames for each fits file cam2)
sci1.fits	(collapsed science file cam1)
sci2.fits	(collapsed science file cam2)

optionally other intermediate files depending on keywords

---

sz_science (for polarimetric data)

sof file:
file1.fits ZPL_SCIENCE_P1_RAW (or for P2/P3: ZPL_SCIENCE_P23_RAW)
file2.fits ZPL_SCIENCE_P1_RAW (or for P2/P3: ZPL_SCIENCE_P23_RAW)

mbias1.fits ZPL_MASTER_BIAS_CAM1
mbias2.fits ZPL_MASTER_BIAS_CAM2
mdark1.fits ZPL_MASTER_DARK_CAM1
mdark2.fits ZPL_MASTER_DARK_CAM2
mflat1.fits ZPL_INT_FLAT_FIELD_MASTER_CAM1
mflat2.fits ZPL_INT_FLAT_FIELD_MASTER_CAM2
modem1.fits ZPL_MODEM_EFF_CAM1
modem2.fits ZPL_MODEM_EFF_CAM2
...

standard calling sequence:
IDL> sz_science, 'science.sof' (,FNAME='<name>', ... other keywords)

Default:
FNAME: sci1.fits, sci2.fits
P2 mode or P1 mode mode is recognized from header. Use option /NODEROT to skip derotation for P1 mode. Use option PMETHOD to set method for calculation of polarization (1, 2 or 3, default is 1, see Section 9 for details)

Files produced:

<file*>_cam1.fits	(preprocessed files for each fits file cam1)
<file*>_cam2.fits	(preprocessed files for each fits file cam2)
p_<file*>_cam1.fits	(2-phase polarization frames for each fits file cam1)
p_<file*>_cam2.fits	(2-phase polarization frames for each fits file cam2)
cal_OBS*_cam1.fits	(calibrated frames for each fits file cam1)
cal_OBS*_cam2.fits	(calibrated frames for each fits file cam2)
coll_Qplus_cam1.fits	(collapsed Stokes-Files)
coll_Qminus_cam1.fits	
coll_Uplus_cam1.fits	
coll_Uminus_cam1.fits	
coll_Qplus_cam2.fits	
coll_Qminus_cam2.fits	
coll_Uplus_cam2.fits	
coll_Uminus_cam2.fits	
sci1.fits	(collapsed science file cam1, plus/minus combined)
sci2.fits	(collapsed science file cam2, plus/minus combined)

optionally other intermediate files depending on keywords

3.3 Useful sub-recipes:

1. sz_preproc	(splits files into ndit and even/odd)
2. sz_pol	(for polarimetry, calculates twophase frames I, Q)

---

sz_preproc

same recipe for imaging and polarimetry (type is recognized from header)

sof file:
file1.fits
file2.fits
...

standard calling sequence:
IDL> sz_preproc, 'preproc.sof'

Default names: <file1>_cam1.fits, <file1>_cam2.fits, ...

---

sz_pol

only for polarimetry

sof file:
<file1>_cam1.fits
<file1>_cam2.fits
<file2>_cam1.fits
...

standard calling sequence:
IDL> sz_pol, 'pol.sof'

Default names: p_<file1>_cam1.fits, p_<file1>_cam2.fits, ...

D) Standard options

All the following options are available for the science recipes and where useful also for the calibration recipes. They are set as keywords when calling the science recipe

1. FNAME

Give a filename prefix (string!). Final name will be <FNAME>_1.fits, <FNAME>_2.fits
FNAME=string set string, e.g. 'test1'

Note: only final collapsed frames gets name fname, intermediate products still have standard names. When rerunning the pipeline with different options and new fname, intermediate output will still be overwritten and should therefore be renamed manually if it is desired to keep them.

2. NORMALIZE (polarimetry only)

Attempt to subtract the residual instrument polarization from each individual frame. The algorithm estimates the instrument polarization in an annulus around the central object according to

IP = total(odd - even)/total(odd+even)

and subsequently subtracts the instrument polarization from the even and odd frames according to

even += 0.5*(odd+even)*IP
odd -= 0.5*(odd+even)*IP

before continuing with the data reduction. The center of the annulus by default is set to [511, 511] for both cameras and the inner and outer radii are set to 30 and 200, respectively. However, if the NCENTER keyword is set, the algorithm will use the NCENTER values as the center of the annulus instead. The inner and outer radii of the annuls can also be customized by setting the IP_RADIUS = [r_in, r_out] keyword at the start of the science recipe.
In addition the instrument polarization algorithm writes the estimated instrument polarization for each exposure into text files with names <file>.ip0.
--> format: [IP_zero, IP_pi]
The last line of the text file represents the median instrument polarization over all exposures.

Note:
This algorithm should only be used if the overall polarization of the exposure can indeed be assumed to be zero, i.e. if the dominant source is unpolarized, and if the pixels included into the calculation of the instrument polarization are unsaturated.

/NORMALIZE use normalization algorithm

3. FRAMESELECT

Use frame selection if bad frames are present. This procedure makes sure that also the header is correctly updated.
For each file containing bad frames produce a textfile with name <file>.fits.bad with a 1-dimensional list (1 column) of the bad frames (start count at 0!).

/FRAMESELECT looks for *.fits.bad files and ignores/deletes the bad frames
/SAVEORIG save original exposure before frame selection is applied (name: orig_*.fits)
/STOPFS stop pipeline before frame selection

Notes:
Frame selection happens after basic calibrations, and is generally done on the cal_*.fits frames. If doing automatic reduction, it is therefore necessary to first run the pipeline without frame selection to produce these files, then create the cal_*.fits.bad text files that identify the bad frames. With the keyword /STOPFS the pipeline can automatically be stopped before frame selection to allow manual inspection of frames and creation of the textfiles. The pipeline can then be rerun using the keywords /NOPREPROC and /SKIPCAL (but without /STOPFS) which will make it proceed where stopped before

By default the bad frames are deleted without saving the original state. Use the /SAVEORIG keyword to save the original or re-run the preprocessing step.

The algorithm checks if the exposure had already been frame selected (according to the NDIT keyword in the header and the current number of frames). It only allows one execution of the FRAMESELECT option. Therefore, to use a different frame selection (i.e. more or less bad frames) restore the original (e.g. rerun the pipeline without skipping the step prior to frame selection or rename the orig_* files if /SAVEORIG was set) and re-apply frame selection.

4. bad pixel filtering

sigma clipping
Apply sigma clipping algorithm described in sz_sigma_clip.pro. Default sigma value is 10. **New filenames** for clipped files: **sclip_*.fits**

/SIGMACLIP	use algorithm
SIGMAVAL=value	set sigma value to be used (default: 10)

median clipping
Use a clean mean algorithm (see sz_median_clip.pro). **New filenames** for clipped files: **mclip_*.fits**
CAUTION: algorithm needs to be improved, be careful!

/CLMEAN	use algorithm

5. Centering options

Pupil stabilized and polarimetry P1 images are derotated by default. However, centering is not applied by default although usually it is preferred to center the images before derotation. To do this, the cal*.fits images can be centered using one's own procedures and then be used as input for the sz-pipeline, or one can use the built-in centering options.
Centering is applied on the cal_*.fits images (i.e. after de-dithering). **New filenames** for centerd files: **cen_*.fits**

NCENTER=[x1,y1, x2,y2]	new/estimated center for all cam1 and cam2 frames
/DOCENTROID	determine center frame by frame
RADIUS=value	radius of subimage for fine centering (default : 8pix)
/GCNTRD	use gcntrd.pro for centering (default: gauss2dfit.pro)
/MPFIT	use mpfit to fit a 2D Moffat function

Notes:
All the built-in centering recipes require a well defined point source (either source itself or coronagraphic spot). If this is not the case, more sophisticated centering algorithms are required (not supported by this pipeline).

NCENTER=[x1,y1,x2,y2] recenters all frames such that the NCENTER coordinates are the new image centers ([512,512]).

/DOCENTROID acts frame by frame. It first finds the dominant point source of the frame (by center of mass) and then applies a centroid routine in a subimage around the source. It then recenters and defines these centroid coordinates to be the image center ([512,512]). The radius of the subimage can be changed by the RADIUS keyword (e.g. this can be useful if more sources of similar brightness are close together).

/DOCENTROID, NCENTER=[x1,y1,x2,y2] acts frame by frame. . It applies a centroid routine in a subimage around the NCENTER coordinates. It then recenters and defines these centroid coordinates to be the image center ([512,512]). The radius of the subimage can be changed by the RADIUS keyword (e.g. this can be useful if more sources of similar brightness are close together).

For /DOCENTROID a text-file for each original cal*.fits file is written (cal*.center). These files can be used to analyze the centering accuracy or potential problems. Together with the SKIPCENTER option these files can also be used to correct bad centering results by the centroid recipes.

The default centering procedure is GAUSS2DFIT.pro and requires a clean unsaturated PSF. For the Beamshift calculation (required accuracy 1/100 pixel, see later) only gauss2dfit gives reasonable results. However, for slightly saturated PSFs or strong neighbouring sources gauss2dfit crashes. In this case for simple centering (e.g. for derotation) one can use the GCNTRD.pro algorithm which can tolerate messier PSFs but is only accurate to about 1/10 pixel. Alternatively, /MPFIT can be used, which fits a 2D Moffat function. It has not been thoroughly tested if and in which cases it works better than gauss2dfit or gcntrd.

Together with the SKIPCENTER option one can use own centering routines and save the centered files with the original filenames cen_<cal_*name>. The science recipes then produce the required internal sof-files but skip the built-in centering algorithms.

6. Skipping options

The pipeline can be re-run on the complete dataset or a sub-set without some of the most time consuming steps. In this case only the corresponding pipeline-internal sof files are produced to select the correct (existing!) files.

/NOPREPROC	skip preprocessing and sz_pol
/FPOL	force execution of sz_pol
/SKIPCAL	skip calibration
/SKIPCLMEAN	skip median clipping
/SKIPSIGMA	skip sigma clipping
/SKIPCENTER	skip centering
/SKIPMERGE	skip merging

Notes:
Each of the above steps writes a new file with a given prefix. When skipping steps make sure that all files produced by the skipped step exist. This is the order within the pipeline:

*_cam1 and *_cam2
cal_*
sub_*
mclip_*
sclip_*
cen_*
msk_*
p1pup*_
merg_*
sadi_*
pca_*

e.g.
/SKIPCAL expects files cal_*.fits
/SKIPCAL, /SIGMACLIP, /SKIPSIGMA expects sclip_cal_*.fits
/CLMEAN, /SKIPCLMEAN expects mclip_cal_*.fits
/CLMEAN, /SKIPCLMEAN, /DOCENTER, SKIPCENTER expects cen_mclip_cal*.fits

7. MERGE

Merge all the frames from different exposures into a single cube before collapsing to a final image. This is recommended especially if frame selection was done, because otherwise the collapse routine will first take the mean/median of each exposure before again taking the mean/median of these results. Thus, if not all exposures have the same number of frames because some frames were removed, the derived mean is technically not the true mean. However, the merge option currently works only for <10000 total frames and will slow down the pipeline for a large number of frames.

When the merge keyword is set, the merged cube is written out as a fits file. In addition, a text file is generated with the name *_params.dat which includes 6 columns. These files are particularly useful if external packages are to be used for PSF subtraction, especially in pupil stabilized imaging mode. In that case, the /NOCOLLAPSE option can be set in order to skip the collapsing of the cube to a final frame. The text file includes 5 colums: the parallactic angle, altitude, LST and UTC values and position angle for each frame, as well as the flux scale factor SFS used if the /SDI option was set (see advanced options). If /SDI is not set this value will always be zero. The position angle is relevant for field stabilized observations where rotations were set.

Note: If manually processing and derotating frames taken in pupil stabilized mode or P1 mode, derotating by the parallactic angle will not orient the image such that North is up. Consult the ZIMPOL documentation to relate the parallactic angle to the correct angle on the sky (or check sz_derot.pro).

The /USEMEDIAN keyword can be used to collapse the frames using the median instead of the mean.

8. DEROTATION OPTION

There are two additional option concerning derotation before collapse of the images:

NODEROT: Do not derotate pupil stabilized or polarimetry P1 images. Pupil stabilized and polarimetry P1 images are recognized from the header and are derotated by default.

/NODEROT use algorithm

WRITE_DEROT: save the final derotated cube before collapse

/WRITE_DEROT write out derotated images before collapsing

9. PMETHOD (polarimetry only)

Set the calculation method for the final polarization image.

PMETHOD = value, where value is 1, 2 or 3 (default: 1)

PMETHOD = 1 : p = 0.5 * (q_plus - q_minus)
PMETHOD = 2 : p = 0.5 (q_plus / i_plus - q_minus / i_minus) * 0.5 * (i_plus + i_minus)
PMETHOD = 3 : Tinbergen ratio method

E) Advanced options

All the following options are available for the science recipes and where useful also for the calibration recipes.

1. Parallel Processing

All recipes can be parallelized using the options:

/PIDL: use parallel IDL processes
NPIDL: number of parallel sub-processes (default: NPIDL=2)

Caution:
Currently error messages in sub-processes are not yet correctly forwarded to the main process. Therefore, if the files are not too large it is recommended to first run in single mode to check for errors or examine the log files for unexpected messages.

Parallel processing is mainly limited by reading/writing. On the ETH cluster bluesky (48 cores) substantial speed-up happens between NPIDL=1 and NPIDL=4. It makes no sense to use NPIDL larger than 12. (Useful number on bluesky around NPIDL=6)

2. Frame transfer semearing correction (polarimetry only)

For high-performance polarimetry correction for the frame transfer (ft) smearing is required. The frame transfer adds additional light to each frame, essentially the same for each row, and therefore also each polarization. The amount of the smearing depends on the ratio ft-time/DIT and is averaged over the entire detector less than 5%. Thus an averaged detector row flux is subtracted from each frame. Because the same flux is

subtracted from the odd and even row (parallel and perpendicular polarization), the polarized intensity is not affected, but the total intensity is reduced. There are issues with the ft-correction for the correction of the read-out stop column or if the frames are saturated. These problems have not been investigated yet.
A frame transfer correction is important for deep polarimetric data. The polarimetric normalization (option /normalize) may not be optimal (especially for regions at larger separation > 1 arcsec from a bright source in column direction), without a previous frame transfer smearing correction.

/FTCORR

2. Beamshift calculations (polarimetry only)

For high-performance polarimetry correction of the beamshift effect is required. It is foreseen that in the future it will be possible to determine the beamshift from the instrument setup (in particular filter and derotator and HWP1/2 orientations). However, for the moment it is only possible to measure the beamshift for each exposure and to re-run the science recipe to correct for the measured beamshift. For the measurement a good PSF is required (either non-coronagraphic or a clean coronagraphic dot not too close to the edge of the coronagraph). The measurement requires the NCENTER keyword for the measurement spot to an accuracy of at least ±1 pixel.

/APPLYBS, NCENTER=[x1,y1,x2,y2]	calculates and applies beamshifts
/APPLYBS, /NOBSCALC	applies median beamshifts defined in .bs0 files

The beamshifts per frame for each exposure are written into text files with names <file>.bs0
--> format: [deltaXzero, deltaYzero, deltaXpi, deltaYpi, LST, ALT]
The last line is always the median values for the delta* columns and the mean values for the LST and ALT columns.
/APPLYBS only uses the last line in the .bs0 files (by default the median) and therefore corrects for the median beamshifts per exposure.
After applying the beamshift the new beamshifts are calculated and stored in the .bs1 files. These files are only for monitoring of the correction quality.

Note:
The APPLYBS option searches for available .bs0 files and then applies the shifts given by the last line (*median* per default). Therefore, for more sophisticated beamshift analysis one can use the individual .bs0 files and change the last line which will then be applied by the pipeline.

3. Injecting fake signals

For field stabilized data (and no derotator offset, i.e. FIELD.POSANG=0) it is straight-forward to use the built-in option to inject a fake signal into the preprocessed frames.

FAKE = '<fake>.fits'

The fits file with the fake signal (and otherwise 0) must have dimensions 1024x1024. The fake signal will then be added to each frame.

Note:
For non-field stabilized data a rotation center is required. Therefore, one has to run the pipeline first without fake signal to create the .center files. Then one can re-run with the fake-option and the pipeline will search for the cal_*.center files. If you apply also /CLMAN or /SIGMACLIP one has to additionally specify the prefix of the .center file using the CENFAKE option, e.g.:

- sz_science_imaging, 'temp1.sof', FAKE='planet_1e-4.fits', CENFAKE='sclip_'
- sz_science_imaging, 'temp1.sof', FAKE='planet_1e-4.fits', CENFAKE='mclip_'
- sz_science_imaging, 'temp1.sof', FAKE='planet_1e-4.fits', CENFAKE='sclip_mclip_'

4. SUBFRAME

Selects a subframe after apply_calib (i.e. before applying SIGMACLIP and centering) to increase speed and/or decrease data size.

SUBFRAME=[xlow, xhigh, ylow, yhigh]

5. MASK

In particular for pupil stabilized imaging and polarimetry P1 mode it is sometimes useful to mask (set as NAN) the AO artifacts before derotation.

/MASK	use algorithm
/SKIPMASK	skip algorithm but produce correct pipeline internal sof files

The algorithm requires a file named 'mask.fits' in current directory with dimensions (1024,1024,2) consisting of the masks for cam1 and cam2. The masked files are then named **msk_*.fits.**

6. P1PUPIL and IGNOREALT (polarimetry only)

In order to use simple ADI or PCA for polarimetry P1 observations one can rotate the frames into a "pupil stabilized" orientation to calculate the mean PSF or the PCA's.

/P1PUPIL	use algorithm

When using this option the pipeline then does not need to de-rotate for the altitude parameter and therefore this option only makes sense together with the /IGNOREALT option:

/IGNOREALT	ignore altitude in P1 derotation law

7. SDI (imaging only)

For imaging observations taken with different filters in camera 1 and camera 2, it is in principle possible to use spectral differential imaging (SDI) to remove the speckle pattern. This is particularly useful when searching for a point source that may have significant emission in one filter but not the other. Using the /SDI option, the pipeline will automatically subtract the flux from the image in the filter taken with at the shorter wavelength from that with the longer wavelength before derotating and collapsing.

/SDI use algorithm
/SKIPSDI skip algorithm but produce correct pipeline internal sof file
/WLSCALE scale size of short wavelength image by wavelength ratio
FLUXSCALE = X set a fixed value X to scale the flux of one image to the other, default (not set): determine automatically for each image

Because of the wavelength difference, speckles move slightly outward at longer wavelengths. With the option /WLSCALE, the short-wavelength image is scaled up to match the longer-wavelength image. If /WLSCALE is not set, the images will not be scaled to match the speckles. This can be sufficient in particular when looking for close companions where the position difference for speckles is negligible.

Warning: The wavelength scaling currently only works properly for the filters BHa and Ha_Cnt, where the wavelength scaling results in an integer size difference. For other filters there may be rounding errors. This option has not been tested for any other filter combination than BHa / Ha_Cnt.

Because the images in the two filters will not generally have the same brightness, a flux scaling is also applied before subtraction. With the keyword FLUXSCALE, a value can be assigned which will be applied to scale the brightness of the shorter wavelength image. If the keyword is not set, the flux scaling factor will be determined automatically from the images and will be different for each frame.

Notes:
If scaling the size by wavelength, any astrophysical object will be in a slightly different position, while speckles are matched in position.

This option can also be combined with /SADI or /PCA (steps 8 and 9), these algorithms are then applied to the difference image.

8. simple ADI

For pupil stabilized imaging and "pseudo pupil stabilized" polarimetry P1 data (see P1PUPIL option above) one can use a simple ADI (angular differential imaging) algorithm. To use this option it is advised to also set the /MERGE keyword which will also automatically compute the appropriate median frames needed as the PSF for this step. If the /MERGE keyword is not set (if there are too many frames), appropriate files have to be manually saved in the directory prior to running this step. For imaging, the files 'mean_1.fits' and 'mean_2.fits' and required (mean image for camera1 and 2

respectively). For polarimetry, the file names are mean_X.fits where X is 1 / 2 for Qplus cam1/2, 3 / 4 for Qminus cam1/2, 5 / 6 for Uplus cam1/2 and 7/8 for Uminus cam1/2.

These frames are then subtracted from each frame before derotation.

/SADI	use algorithm
/SKIPSADI	skip algorithm but produce correct pipeline internal sof files

The pipeline can be re-run on the complete dataset or a sub-set. In this case only the corresponding pipeline-internal sof files are produced to select the correct (existing!) files.
The mean subtracted file names have names **sadi_*.fits**

Note:
This algorithm is only meant to be an "advanced quicklook" and therefore does not accept any additional options or fine tuning. It is recommended to use the merged data cube and more sophisticated PSF subtraction algorithms for proper subtraction of the stellar PSF for pupil stabilized imaging observations.

9. simple PCA

For pupil stabilized imaging and "pseudo pupil stabilized" polarimetry P1 data (see P1PUPIL option above) one can use a simple PCA algorithm. The algorithm calculates a number of modes (default 10) which are subtracted from each frame. For calculating the eigenvalues it uses a cube consisting of every 20th frame. However, using the option /PCAALL , all frames are used (only possible if dataset is not too large, e.g. <1000 frames?).

/PCA	use algorithm
/EIGENV	calculate Eigenvalues
/PCAALL	use all frames for calculation of Eigenvalues
PCANUM=X	X=number of modes to subtract, default is 10
/QPCA	apply algorithm on Stokes QU frames
/SKIPPCA	skip algorithm but produce correct pipeline internal sof files

The pipeline can be re-run on the complete dataset or a sub-set. In this case only the corresponding pipeline-internal sof files are produced to select the correct (existing!) files.
The subtracted file names have names **pca_*.fits**

Note:
This algorithm is only meant to be an "advanced quicklook" and therefore does not accept any additional options or fine tuning. It is recommended to use the merged data cube and more sophisticated PSF subtraction algorithms for proper subtraction of the stellar PSF for pupil stabilized imaging observations.

Calculated Eigenvalues are saved in 'pca_'+<sof>+'.sav' and only need to be calculated once.

Per default the algorithm is only applied on the intensity or Stokes I frames. This can be changed by the option /QPCA. However, in this case it is only applied on the QU frames but not on Stokes I. For both PCA on QU and I the pipeline has to be run twice (with and without the /QPCA option).

10. daytime calibrations (CALIB)

When using the science recipe for analyzing daytime calibration data (e.g. cross-talk measurements) the recipe gets confused by missing or additional polarimetry keywords (e.g. POL.STOKES = Vplus). Therefore, by using the /CALIB keyword all *plus* frames are treated as Qplus frames and all *minus* frames are treated as Qminus frames by the recipe.

Note:
This option was mainly used by ABz for daytime testing during commissioning.

11. INSTRPOL (polarimetry only)

If this keyword is set, the science recipe also calculates the instrumental polarization measured by the HWP2 switching. For the calculation the collapsed files are used. To find the instrumental polarization of different exposures run the science recipe on a subset of exposures.

Note:
This option was mainly used by ABz for daytime testing during commissioning. Therefore, for the moment no fits file is produced but the result is only stored in the common variables instr1 and instr2.

12. CAMALIGN

Align cam2 to cam1 (use sz_align.pro, requires good/unsaturated PSF). This option is applied at the very end on the already collapsed science frames. Only the cam2 frame is shifted. The shift is calculated from the intensity images and then the same shifts are applied for intensity and polarimetry images.

/CAMALIGN use algorithm

Note: this option is usually not needed if the centering of the individual cameras was done properly in the centering step and may actually worsen the alignment if frames were properly centered.

F) Overview of filetypes

A) preprocessed Files

<orig>_cam1.fits
<orig>_cam2.fits

Format: [1024, 1024, ndit, 2] , 2 defines even/odd

B) for Polarimetry, single double-phase Frames, I, Q:
---
p_<orig>_cam1.fits
p_<orig>_cam2.fits

Format: [1024, 1024, ndit/2, 2] , 2 defines I, Q
Format for bias, dark: [1024, 1024, ndit/2, 2] , 2 defines phase 0/Pi

C) (optional) subframes:
---
sub_*.fits

Format: [xlow:xhigh, ylow:yhigh, ndit, 2] , imaging
Format: [xlow:xhigh, ylow:yhigh, ndit/2, 2] , polarimetry

D) (optional) median or sigma-clipping:
---
mclip_*.fits (median clipping)
sclip_*.fits (sigma clipping)
sclip_mclip_*.fits (median+sigma clipping)

Format: [1024, 1024, ndit, 2] , imaging
Format: [1024, 1024, ndit/2, 2] , polarimetry

E) (optional) centering:
---
cen_*.fits

Format: [1024, 1024, ndit, 2] , imaging
Format: [1024, 1024, ndit/2, 2] , polarimetry

F) (optional) masking:
---
msk_*.fits

Format: [1024, 1024, ndit, 2] , imaging

Format: [1024, 1024, ndit/2, 2] , polarimetry

G) (optional) rotate P1 into pupil stabilized orientation:
---
p1pup_*.fits

Format: [1024, 1024, ndit, 2] , imaging
Format: [1024, 1024, ndit/2, 2] , polarimetry

H) (optional) merge
---
merg_*.fits

Format: [1024, 1024, ndit*nexp, 2] , imaging
Format: [1024, 1024, ndit/2*nexp, 2] , polarimetry

I) (optional) SDI, imaging only
---
sdi_*.fits

Format: [1024, 1024, ndit] , imaging

(if merged, 3rd dimension is ndit*exp)

J) (optional) simple ADI:
---
sadi_*.fits

Format: [1024, 1024, ndit, 2] , imaging
Format: [1024, 1024, ndit/2, 2] , polarimetry

(if merged, 3rd dimension is ndit*exp or ndit/2*exp)

K) (optional) PCA subtraction:
---
pca_*.fits

Format: [1024, 1024, ndit, 2] , imaging
Format: [1024, 1024, ndit/2, 2] , polarimetry

(if merged, 3rd dimension is ndit*exp or ndit/2*exp)

L) temporary collapsed calibration Files:
---
bias, dark, flat:
coll_tmp_cal_cam1.fits
coll_tmp_cal_cam2.fits

modem:

coll_tmp_modem_Qplus_cam1.fits
coll_tmp_modem_Qplus_cam2.fits
coll_tmp_modem_Qminus_cam1.fits
coll_tmp_modem_Qminus_cam2.fits

Format:
Imaging: [1024, 1024, 2] , 2 defines even, odd
Polarimetry: [1024, 1024, 2] , 2 defines I, Q

- no calibration applied
(for calibration recipes: calib applied on collapsed frames)

M) calibrated science frames:
---
cal_OBS224_00*_cam1.fits
cal_OBS224_00*_cam2.fits

- each frame separately calibrated (Bias, Flat, Modem).
- for Polarimetrie a double-phase frame is created first.
- optional: centering applied

Format:
Imaging: [1024, 1024, ndit, 2] , 2 defines even, odd
Polarimetry: [1024, 1024, ndit/2, 2] , 2 defines I, Q

- optional median or sigma clipping applied (names mclip_*.fits, sclip_*.fits)
- optional subframes selected (sub_*.fits)
- optional centered frames (cen_*.fits)
- any combination of options (e.g. cen_sub_sclip_*.fits)
- optional merged cube: merg_tmp_*.fits, and parameters, merg_tmp_*_params.dat

N) for Polarimetry, collapsed Stokes-Files:
---
coll_tmp_cal_Qplus_cam1.fits
coll_tmp_cal_Qminus_cam1.fits
coll_tmp_cal_Uplus_cam1.fits
coll_tmp_cal_Uminus_cam1.fits

coll_tmp_cal_Qplus_cam2.fits
coll_tmp_cal_Qminus_cam2.fits
coll_tmp_cal_Uplus_cam2.fits
coll_tmp_cal_Uminus_cam2.fits

- for P1 mode: derotation applied

O) collapsed science Files:

(plus/minus Q und plus/minus U combined)
sci_*_1.fits
sci_*_2.fits

Format:
Imaging: [1024, 1024, 2] , 2 defines even, odd
Polarimetry: [1024, 1024, 4] , 4 defines I_q, Q, I_u, U

- for P1 mode and pupil stabilized imaging: derotation applied
- optional subframe format: [subx, suby, 2] or [subx, suby, 4])
- optional: derotated frames before collapsing: derot_*.fits