# NTT pipeline

The NTT pipeline is a python module that should be added to the PYTHONPATH of the reducer.
If the reducer is not a python-user, probably there is not a PYTHONPATH defined on his machine.
The installed modules in python are usually in a directory called site-package in the lib/ directory where python is installed. Each user can decide to have a personal s-p directory where to put the python module he developed.
Setting as /home/user/site-package/ the directory where I want to have my private python module, I can add the following lines in my .bashrc

```
# SM Python
if [ -n "$PYTHONPATH" ]; then
   export PYTHONPATH=$PYTHONPATH:/home/user/site-package/
else
   export PYTHONPATH=/home/usere/site-package/
fi
```

The second step is to untar the NTT.tar directory in /home/usere/site-package/
At this point when ever I use python I can import the NTT module:

```
$ python
>> import NTT
>>
```

The NTT pipeline module is now working.
The python modules used in the pipeline are the following:

Pyraf
numpy
pyfits
pylab

You need to have these modules running on your machine, if you want to use the NTT pipeline.

# NTT STRUCTURE

The NTT directory contains the following files/directories:

| | |
|---|---|
| README | readmefile for installation |
| __init__.py | (needed to make the NTT directory a python module) |
| __init__.pyc | (needed to make the NTT directory a python module) |
| archive | (directories with archive data: flat, bias.arc) |
| standard | (dir. with calibration files: standards, extinction curve, ecc) |
| bin | (directory with the executable python script) |
| doc | (documentation) |
| login.cl | (login.cl file for iraf setup) |
| uparm | (iraf parameters files) |
| src | NTT pipeline sub-modules |

The most important directory for the reducer is the bin directory with the executable files:

-**PESSTOFASTSPEC.py** Fast spectroscopy for classification (no flat, bias correction, calibration from archive)

-**PESSTOEFOSCPHOT.py** (photometry efosc pre-reduction)

-**PESSTOEFOSC2dSPEC.py** (spectroscopy efosc pre-reduction 2D frames-wavelengh-calibrated)

-**PESSTOEFOSC1dSPEC.py** (spectroscopic extraction and flux calibration)

-**PESSTOSOFIPHOT.py** (photometry sofi pre-reduction)

-**PESSTOSOFI1dSPEC.py** (spectroscopy efosc pre-reduction 2D frames-wavelengh-calibrated)

-**PESSTOSOFI2dSPEC.py** (spectroscopic extraction and flux calibration)

- **etabase_file.py**

each of these modules will be explained in detail here:

## EFOSC PHOTOMETRY

```
$ /home/user/s-p/NTT/bin/PESSTOEFOSCPHOT.py
Usage: PESSTOEFOSCPHOT.py  -l listfile
> Fast photometry reduction of efosc images
Options:
  -h, --help            show this help message and exit
  -l INPUT, --list=INPUT
                        name list        [listfiles]
  -i, --interactive
  -f, --flat
  -F LISTFLAT, --listflat=LISTFLAT
                        name flat list   []
  -b, --bias
  -B LISTBIAS, --listbias=LISTBIAS
                        name bias list   []
  -m, --mask
  -M BADPIXELMASK, --maskbadpixel=BADPIXELMASK
                        name bad pixel mask      []
  -r FRINGINGMASK, --fringing=FRINGINGMASK
                        name fringing mask       []
  -a, --archive
```

The fundamental input to make the script running is:

```
$ /home/user/s-p/NTT/bin/PESSTOEFOSCPHOT.py  -l listfiles
```

The recommended syntax is:
$ /home/user/s-p/NTT/bin/PESSTOEFOSCPHOT.py  -l listfiles -i

**option –l listfiles  (**list of all the files)
**option –i** is to make some choice interactively
**option –f** is to skip the flat field correction
**option –F** listflat (list of flat or a single flat) is to use a specific flat
**option –b** is to skip the bias correction
**option –B** listbias (list of bias or a single bias) is to use a specific bias file
**option –m** is to skip the mask correction
**option –M** listmask (list of masks or a single mask) is to use a specific mask file
**option –a** is to copy the combined flatfield and bias images in the NTT archive, to be used for nights with missing flat and bias calibrations

*output running the program:*

1) **recognize the different types of files**:

*EFOSC_AcqRotSlit026_0001.fits 120.0005 -> not good*
*EFOSC_AcqRotSlit026_0002.fits 120.0005 -> not good*
*EFOSC_AcqSlit024_0001.fits 10.0005 -> not good*
*EFOSC_AcqSlit024_0002.fits 10.0006 -> not good*
*EFOSC_AcqSlit024_0003.fits 120.0006 -> not good*
*EFOSC_Dark025_0010.fits 0.0 -> bias*
*EFOSC_Dark025_0011.fits 0.0 -> bias*
*EFOSC_FlatIma023_0001.fits 16.0706 -> flat*
*EFOSC_FlatIma023_0002.fits 16.0705 -> flat*
*EFOSC_FlatIma023_0003.fits 16.0705 -> flat*
*EFOSC_FlatIntSpec024_0001.fits 127.0006 -> spectroscopic data*
*EFOSC_FlatIntSpec024_0002.fits 127.0006 -> spectroscopic data*
*EFOSC_FlatIntSpec024_0003.fits 127.0006 -> spectroscopic data*
*FOSC_Image024_0019.fits 60.0006 -> object*
*EFOSC_Image024_0020.fits 60.0005 -> object*
*EFOSC_Image024_0021.fits 60.0005 -> object*
*EFOSC_Image024_0025.fits 2.0006 -> object*
*EFOSC_Image024_0026.fits 2.0006 -> object*
*EFOSC_Image024_0027.fits 2.0005 -> object*
*EFOSC_Image024_0028.fits 2.0006 -> object*

2) **make bias**
- the program calculate the mean for each bias (Mi), and reject the bias files with a mean 1-sigma-out of mean(Mi)

*EFOSC_Dark023_0007.fits rejected*
*EFOSC_Dark023_0008.fits 9.793283*
*EFOSC_Dark023_0009.fits rejected*
*EFOSC_Dark023_0010.fits 11.25625*
*EFOSC_Dark023_0011.fits 11.02843*
*EFOSC_Dark023_0012.fits 10.25389*

- if **interactive** ask to the reducer if the other bias file are **good or bad**
*good/bad [[g]/b/G(all good)/s(stop) ] ?g*
*good/bad [[g]/b/G(all good)/s(stop) ] ? b*
*good/bad [[g]/b/G(all good)/s(stop) ] ? g*

*good/bad  [[g]/b/G(all good)/s(stop) ] ? g*

3) **make flat**

the program reject the flat saturated and with low counts automatically.
If **interactive** ask to the reducer if the other flat file are **good or bad**

*good/bad  [[g]/b/G(all good)/s(stop)]?*
*good/bad  [[g]/b/G(all good)/s(stop)]?*

4)  **object pre-reduction:**

TRIM correction
BIAS correction
Bad pixel mask correction
Flat Field correction

*EFOSC_Image026_0069.fits: Nov  3 18:22 Overscan section is [3:1010,1026:1029] with mean=215.9374*
*EFOSC_Image026_0069.fits: Nov  3 18:22 Zero level correction image is masterbias_20110125.fits*
*SN2009ls_20110126_R642_2.fits: Nov  3 18:22 Flat field image is masterflat_20110125_R642.fits with scale=39735.15*
*SN2009ls_20110126_R642_2.fits: Nov  3 18:22 Trim data section is [3:1010,1:1015]*
*this filter does not need fringing correction*
*FIXPIX: image SN2009ls_20110126_R642_2.fits with mask bad_pixel_mask.pl*
*EFOSC_Image026_0070.fits SN2009ls_20110126_R642_3.fits*
*bias  yes masterbias_20110125.fits*
*flat  yes masterflat_20110125_R642.fits*

5) **Fringing correction for I filter images:**

*do you want to correct for fringing ? [y/n] [y]*
*do you want to correct for fringing ? [y/n] [y]*

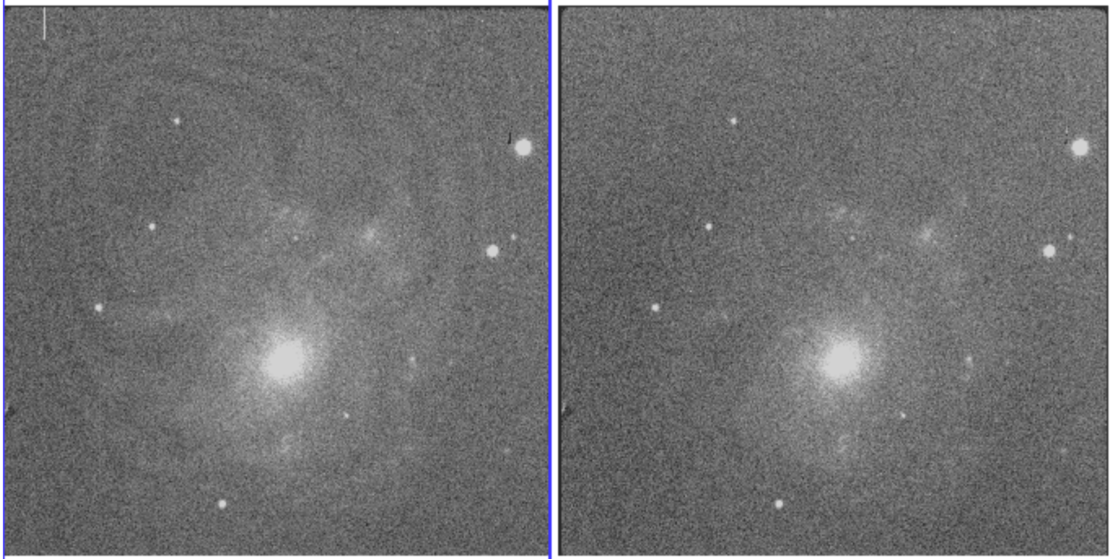
For the Fringing correction, the objects are masked and on the rest of the image the fringing image is scaled to minimize the difference with the object image. In the header FRICOR is reported how much the fringing mask is scaled:

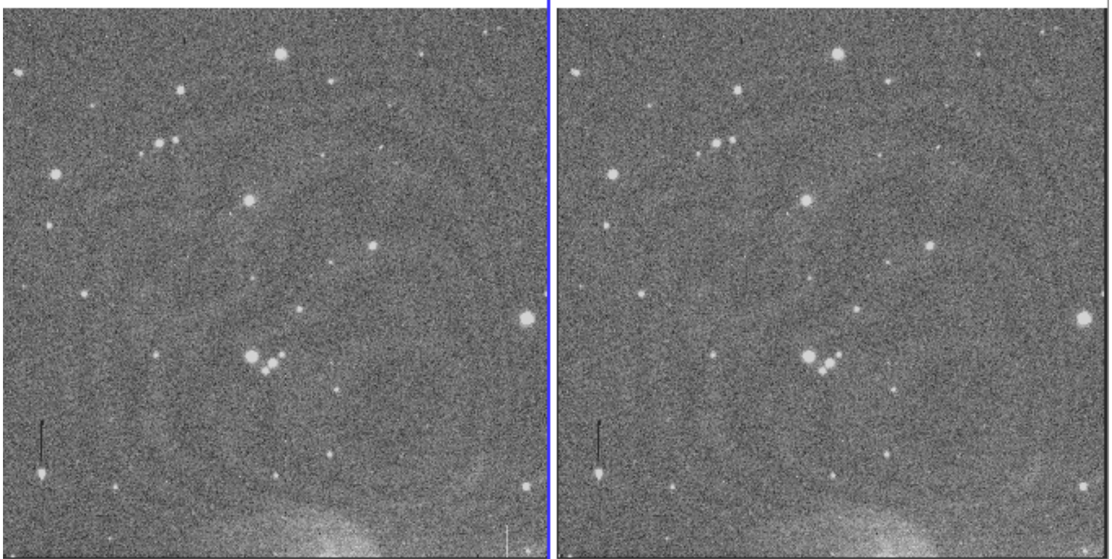
*fr_SN2010jp_20110126_i705_2.fits -3.53981685638 * (fmask_20110126.fits - 12.4882030487)*

some time the fringing correction give good result
other time there is not a big improvement.

6) **The output files:**

Object_DATE_FILTER_N.fits
FR_Object_DATE_FILTER_N.fits  (if fringing correction is applied)
…..
SN2010F_20110126_B639_4.fits
SN2010F_20110126_B639_5.fits
…..





# EFOSC SPECTROSCOPY

Usage: **PESSTOEFOSC2dSPEC.py** -l listfile
> Specroscopic pre-reduction of efosc data

Options:
```
  -h, --help            show this help message and exit
  -l INPUT, --list=INPUT
                        name list        [listfiles]
  -i, --interactive
  -f, --flat
  -F LISTFLAT, --listflat=LISTFLAT
                        name flat list   []
  -b, --bias
  -B LISTBIAS, --listbias=LISTBIAS
                        name bias list   []
  -a LISTARC, --listarc=LISTARC
                        name arc list    []
```

The fundamental input to make the script running is:
$ /home/user/s-p/NTT/bin/ PESSTOEFOSC2dSPEC.py -l listfile

The **recommended** syntax is:
$ /home/user/s-p/NTT/bin/ PESSTOEFOSC2dSPEC.py -l listfile -i

**option –l listfiles (**list of all the files)
**option –i** is to make some choice interactively
**option –f** is to skip the flat field correction
**option –F listflat** (list of flat or a single flat) is to use a specific flat
**option –b** is to skip the bias correction
**option –B listbias** (list of bias or a single bias) is to use a specific bias file
**option –a listarc** is to use a particular arc to wavelength calibration

*output running the program:*

**1) recognize the different types of files**
no print on the screen ..just wait

2) **make bias**
- the program calculate the mean for each bias (Mi), and reject the bias files with a mean 1-sigma-out of mean(Mi)

*EFOSC_Dark023_0007.fits rejected*
*EFOSC_Dark023_0008.fits 9.793283*
*EFOSC_Dark023_0009.fits rejected*
*EFOSC_Dark023_0010.fits 11.25625*
*EFOSC_Dark023_0011.fits 11.02843*
*EFOSC_Dark023_0012.fits 10.25389*

- if **interactive** ask to the reducer if the other bias file are **good or bad**
*good/bad [[g]/b/G(all good)/s(stop) ] ?g*
*good/bad [[g]/b/G(all good)/s(stop) ] ? b*
*good/bad [[g]/b/G(all good)/s(stop) ] ? g*

*good/bad [[g]/b/G(all good)/s(stop) ] ? g*

3) **reduction for each setup**

The program print the object to be reduced for each setup and ask to proceed:

*['EFOSC_Spectrum024_0002.fits', 'EFOSC_Spectrum024_0008.fits',*
*'EFOSC_Spectrum024_0010.fits']*
*('Gr16', 'OG530', 'slit1.0')*
*do you want to reduce this setup [[y],n] ?*

**4) make flat for each setup**

For each object the program display the image on DS9 and ask the position of the object in order to compute the normalized flat in a region around the object position.

*which is the column of the object (x axis) [500] ?*

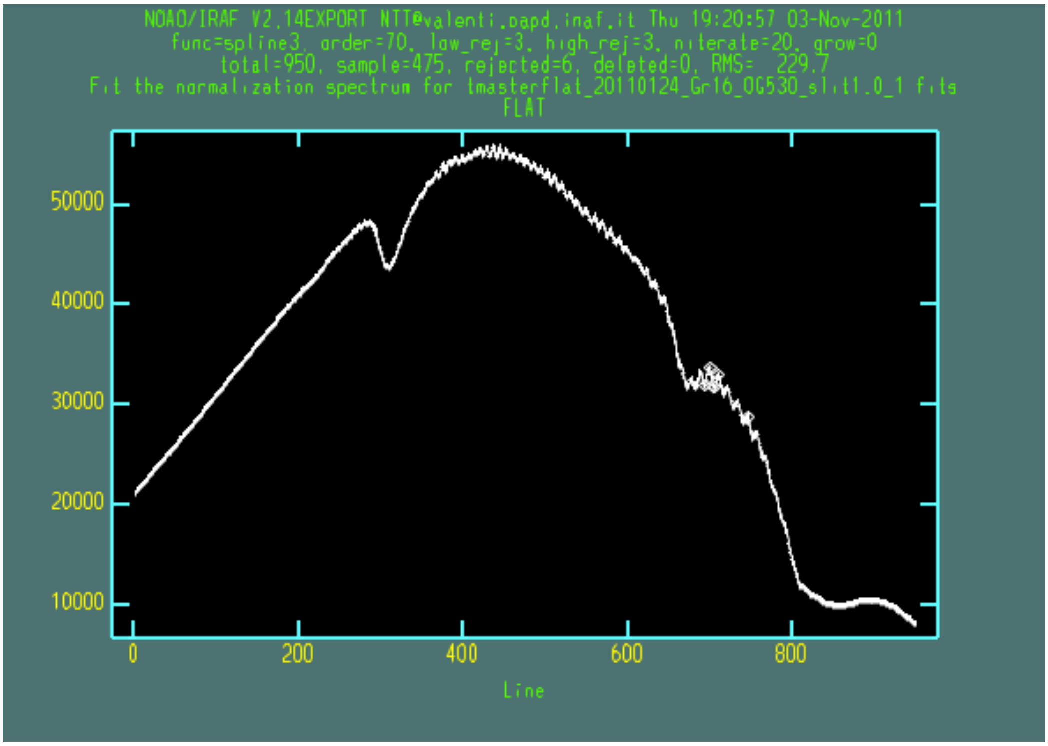
The list of all the flat available for this setup are shown with also the distance in time from the observation:

-0.00308334000147 EFOSC_FlatIntSpec024_0001.fits ('Gr16', 'OG530', 'slit1.0')
-0.0049469000005 EFOSC_FlatIntSpec024_0002.fits ('Gr16', 'OG530', 'slit1.0')
-0.00680999000178 EFOSC_FlatIntSpec024_0003.fits ('Gr16', 'OG530', 'slit1.0')
-0.175067060001 EFOSC_FlatIntSpec024_0010.fits ('Gr16', 'OG530', 'slit1.0')
-0.176937340002 EFOSC_FlatIntSpec024_0011.fits ('Gr16', 'OG530', 'slit1.0')
-0.178813049999 EFOSC_FlatIntSpec024_0012.fits ('Gr16', 'OG530', 'slit1.0')
-0.230789430003 EFOSC_FlatIntSpec024_0013.fits ('Gr16', 'OG530', 'slit1.0')
-0.232655649997 EFOSC_FlatIntSpec024_0014.fits ('Gr16', 'OG530', 'slit1.0')
-0.234529959998 EFOSC_FlatIntSpec024_0015.fits ('Gr16', 'OG530', 'slit1.0')

For reducer can chose with flat to use for each object (the closer in time)
*-0.00308334000147 EFOSC_FlatIntSpec024_0001.fits ('Gr16', 'OG530', 'slit1.0')*
*good/bad/stop(enough files, go on) [[g],b,s]*
*-0.0049469000005 EFOSC_FlatIntSpec024_0002.fits ('Gr16', 'OG530', 'slit1.0')*
*good/bad/stop(enough files, go on) [[g],b,s]*
*-0.00680999000178 EFOSC_FlatIntSpec024_0003.fits ('Gr16', 'OG530', 'slit1.0')*
*good/bad/stop(enough files, go on) [[g],b,s]*
*-0.175067060001 EFOSC_FlatIntSpec024_0010.fits ('Gr16', 'OG530', 'slit1.0')*
*good/bad/stop(enough files, go on) [[g],b,s]* ***s***

press **s** to stop the interactive choise after you have selected at least 3 good flat cose in time to the observations.
If the option interactive is not used the program chose the 3 flat closer
In time to the observations.

In the interactive mode the response on the combined flat is done interactively:
*masterbias_20110123.fits: Nov 3 19:17 Trim data section is [100:950,1:950]*
*masterflat_20110124_Gr16_OG530_slit1.0_1.fits: Nov 3 19:17 Trim data section is [100:950,1:950]*
*masterflat_20110124_Gr16_OG530_slit1.0_1.fits: Nov 3 19:17 Overscan section is [3:1010,1026:1029] with mean=229.1458*
*masterflat_20110124_Gr16_OG530_slit1.0_1.fits: Nov 3 19:17 Zero level correction image is tmasterbias_20110123.fits*
*Fit the normalization spectrum for tmasterflat_20110124_Gr16_OG530_slit1.0_1.fits interactively (yes):*




**5) object correction:**

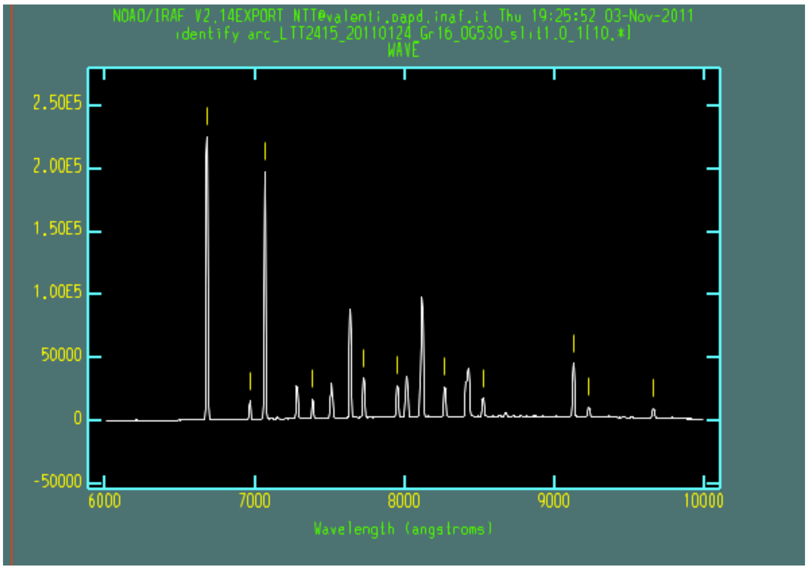
The object is corrected for TRIM, OVERSCAN, BIAS, FLAT
Also the closest arc_file is corrected as the object.

*EFOSC_Spectrum024_0002.fits: Nov 3 19:22 Trim data section is [100:950,1:950]*
*EFOSC_Spectrum024_0002.fits: Nov 3 19:22 Overscan section is [3:1010,1026:1029] with mean=221.8332*
*EFOSC_Spectrum024_0002.fits: Nov 3 19:22 Zero level correction image is tmasterbias_20110123.fits*
*EFOSC_Spectrum024_0002.fits: Nov 3 19:22 Flat field image is nmasterflat_20110124_Gr16_OG530_slit1.0_1.fits with scale=0.9961664*

**6) arc wavelength calibration**

The arc is calibrated in wavelength first along one column and automatically along the 2D image using as reference the closest arc in the archive

*['EFOSC_spec_HeAr023_0004.fits', 'EFOSC_spec_HeAr024_0004.fits', 'EFOSC_spec_HeAr025_0004.fits']*
*EFOSC_spec_HeAr023_0004.fits: Nov 3 19:22 Trim data section is [100:950,1:950]*
*EFOSC_spec_HeAr023_0004.fits: Nov 3 19:22 Overscan section is [3:1010,1026:1029] with mean=218.8736*
*EFOSC_spec_HeAr023_0004.fits: Nov 3 19:22 Zero level correction image is tmasterbias_20110123.fits*
*EFOSC_spec_HeAr023_0004.fits: Nov 3 19:22 Flat field image is nmasterflat_20110124_Gr16_OG530_slit1.0_1.fits with scale=0.9961664*
*arc_20110904_Gr16_OG530_slit1.0_3.fits arc_LTT2415_20110124_Gr16_OG530_slit1.0_1.fits /Users/sv/QUBA/s-p/NTT/archive/efosc/arc/Gr16/OG530/slit1.0/database/idarc_20110904_Gr16_OG530_slit1.0_3*
*cp /Users/sv/QUBA/s-p/NTT/archive/efosc/arc/Gr16/OG530/slit1.0/database/idarc_20110904_Gr16_OG530_slit1.0_3 database/*
*arc_LTT2415_20110124_Gr16_OG530_slit1.0_1[10,*] 11/11 11/11 1.82 7.68 9.64E-4 0.0811*
*Fit dispersion function interactively? (no|yes|NO|YES) ('yes'):*




*NOAO/IRAF V2.14EXPORT NTT@valenti.oapd.inaf.it Thu 19:27:11 03-Nov-2011*
*Longslit coordinate fit name is arc_LTT2415_20110124_Gr16_OG530_slit1.0_1.*
*Longslit database is database.*
*Features from images:*
*arc_LTT2415_20110124_Gr16_OG530_slit1.0_1*
*Map User coordinates for axis 2 using image features:*
*Number of feature coordnates = 935*
*Mapping function = legendre*
*X order = 6*
*Y order = 4*
*Fitted coordinates at the corners of the images:*
*(1, 1) = 6002.962  (851, 1) = 5995.915*
*(1, 950) = 9990.408  (851, 950) = 9983.62*

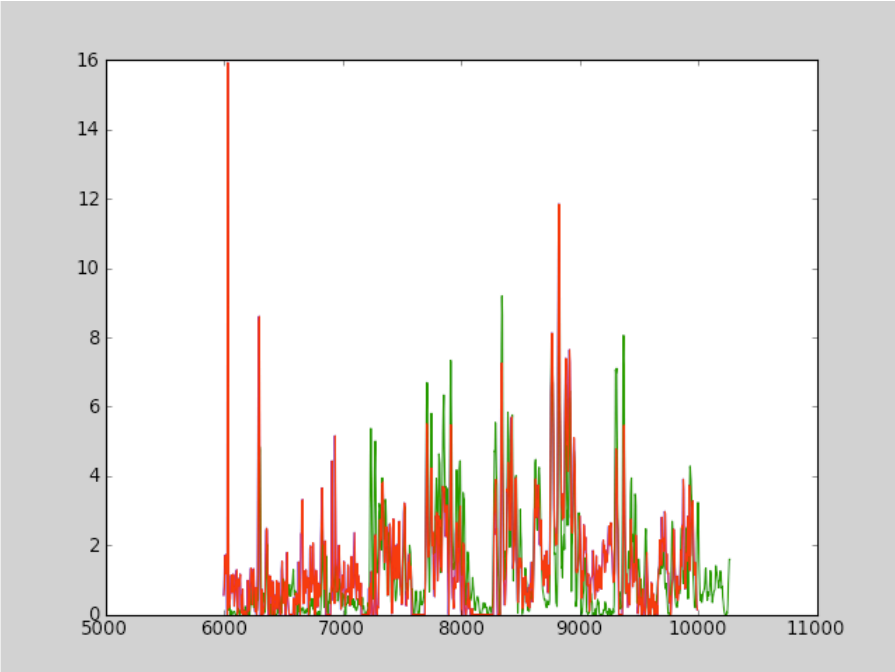
## 7) object 2D wavelength calibration

The 2D - image is wavelength calibrated using the arc file

*NOAO/IRAF V2.14EXPORT NTT@valenti.oapd.inaf.it Thu 19:27:13 03-Nov-2011*
*Transform LTT2415_20110124_Gr16_OG530_slit1.0_1.fits to tLTT2415_20110124_Gr16_OG530_slit1.0_1.fits.*
*Conserve flux per pixel.*
*User coordinate transformations:*
*arc_LTT2415_20110124_Gr16_OG530_slit1.0_1*
*Interpolation is spline3.*
*Using edge extension for out of bounds pixel values.*
*Output coordinate parameters are:*
*x1 =  1., x2 =  851., dx =  1., nx =  851, xlog = no*
*y1 =  5996., y2 =  9990., dy =  4.209, ny =  950, ylog = no*

## 8) wavelength calibration check

Wavelengh calibration is checked comparing the 2D-image sky with a sky from archive (for the same setup)




**1 to 8 Steps are repeated for all the objects and all the setup**

**9) output files are the 2D-images wavelength calibrated:**
the output file have this format: t+objectname+date+filter
tLTT2415_20110124_Gr16_OG530_slit1.0_1.fits

# Extraction and calibration

1) make a list with all the 2D-images (output of the previous command)
$ ls t*.fits > list2D

2) run the second stpe for efosc spectra reduction

~$ /home/user/site-packages/NTT/bin/**PESSTOEFOSC1dSPEC.py**
Usage: PESSTOEFOSC1dSPEC.py -l listfile

> Specroscopic reduction of efosc data

```
Options:
  --version            show program's version number and exit
  -h, --help           show this help message and exit
  -l INPUT, --list=INPUT
                       name list       [listfiles]
  -i, --interactive
  -s LISTSTAND, --standard=LISTSTAND
                       name standard list       []
  -t, --trace
  -d, --dispersion
```

The fundamental input to make the script running is:
$ /home/user/site-packages/NTT/bin/PESSTOEFOSC1dSPEC.py –l list2D

The **recommended** syntax is:
$ /home/user/site-packages/NTT/bin/PESSTOEFOSC1dSPEC.py –l list2D -i

**option –l listfiles (**list of 2D wavelength calibrated files)
**option –i** is to make some choice interactively
**option –s** use specific standard list (**NOT IMPLEMENTED YET** )
**option –t** For very faint object we do the trace with another object
**option –d** chose interactively the dispersion line will be plotted to chose the aperture and the background region.

*output running the program:*

**1) extraction**
print on the screen the list of objects for each setup that should be reduced, dividing the objects in standard (**std**) and object (**obj**).

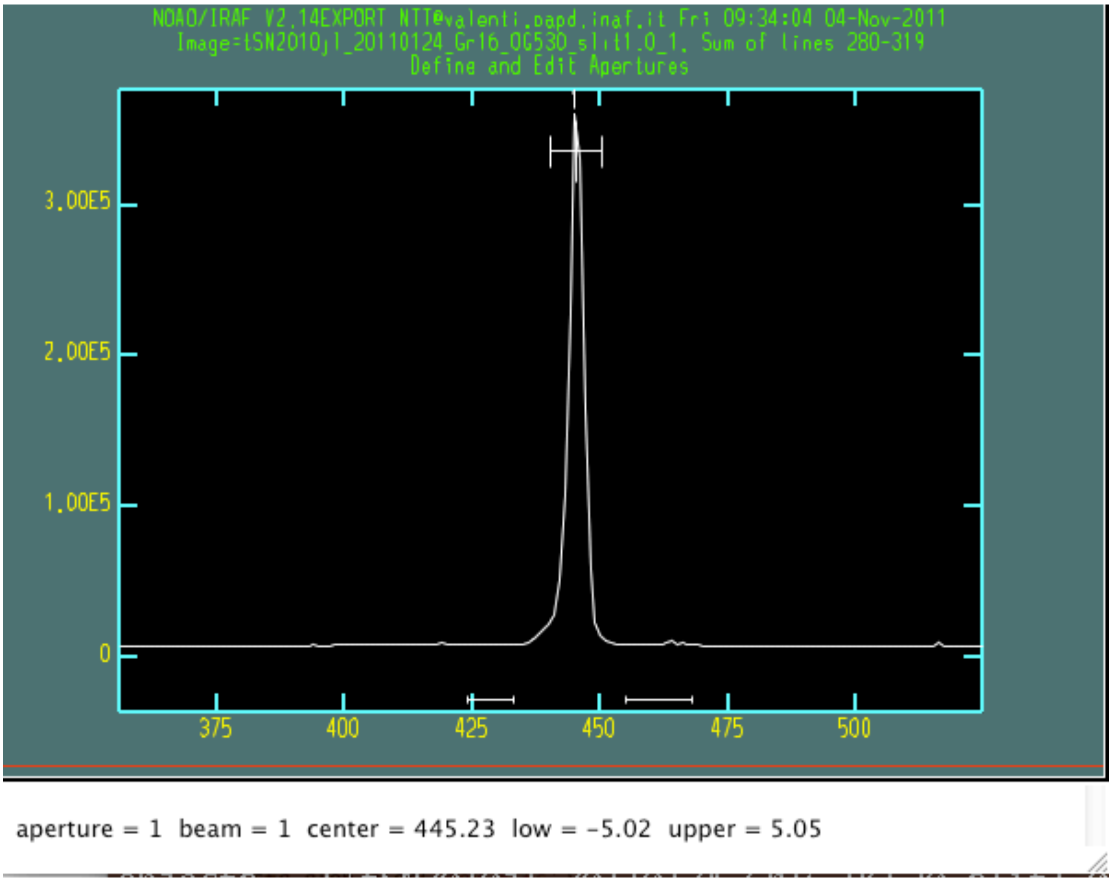
*std ('Gr16', 'OG530', 'slit1.0')*
*tLTT2415_20110124_Gr16_OG530_slit1.0_1.fits*
*tLTT2415_20110124_Gr16_OG530_slit1.0_2.fits*
*std ('Gr13', 'Free', 'slit1.5')*
*tLTT2415_20110125_Gr13_Free_slit1.5_1.fits*
*std ('Gr11', 'Free', 'slit1.0')*
*tLTT2415_20110124_Gr11_Free_slit1.0_1.fits*
*obj ('Gr16', 'OG530', 'slit1.0')*
*tSN2010jl_20110124_Gr16_OG530_slit1.0_1.fits*
*tSN2011A_20110124_Gr16_OG530_slit1.0_1.fits*
*obj ('Gr11', 'Free', 'slit1.0')*
*tSN2010jl_20110124_Gr11_Free_slit1.0_1.fits*
Start extracting the first object with the first setup. If the object has been already extracted, will ask the reducer if he wants to extract again the spectrum or skip this step. If the option interactive is used the exstraction will be interactive.

*setup= ('Gr16', 'OG530', 'slit1.0')*
*objects= ['tSN2010jl_20110124_Gr16_OG530_slit1.0_1.fits',*
*'tSN2011A_20110124_Gr16_OG530_slit1.0_1.fits']*

*next object= tSN2010jl_20110124_Gr16_OG530_slit1.0_1.fits SN2010jl*

*Find apertures for tSN2010jl_20110124_Gr16_OG530_slit1.0_1? ('yes'):*
*Number of apertures to be found automatically (1):*
*Resize apertures for tSN2010jl_20110124_Gr16_OG530_slit1.0_1? ('yes'):*
*Edit apertures for tSN2010jl_20110124_Gr16_OG530_slit1.0_1? ('yes'):*
*Trace apertures for tSN2010jl_20110124_Gr16_OG530_slit1.0_1?Fit traced positions for tSN2010jl_20110124_Gr16_OG530_slit1.0_1 interactively?Fit curve to aperture 1 of tSN2010jl_20110124_Gr16_OG530_slit1.0_1 interactivelyWrite apertures for tSN2010jl_20110124_Gr16_OG530_slit1.0_1 to databaseExtract aperture spectra for tSN2010jl_20110124_Gr16_OG530_slit1.0_1?Review extracted spectra from tSN2010jl_20110124_Gr16_OG530_slit1.0_1?Review extracted spectrum for aperture 1 from tSN2010jl_20110124_Gr16_OG530_slit1.0_1?*

*next object=  tSN2011A_20110124_Gr16_OG530_slit1.0_1.fits   SN2011A*
*……*
*all object with this setup extracted*




when all the objects with one setup have been extracted, the standard with the same setup will be extracted (if available). If an external standard is provided with the **option –s** (with the same setup) the standard extraction step will be skip. (NOT IMPLEMENTED YET). If more than one standard are available, the reducer has to chose interactively which one to use.

*standard for setup ('Gr16', 'OG530', 'slit1.0') =  tLTT2415_20110124_Gr16_OG530_slit1.0_1.fits LTT2415*

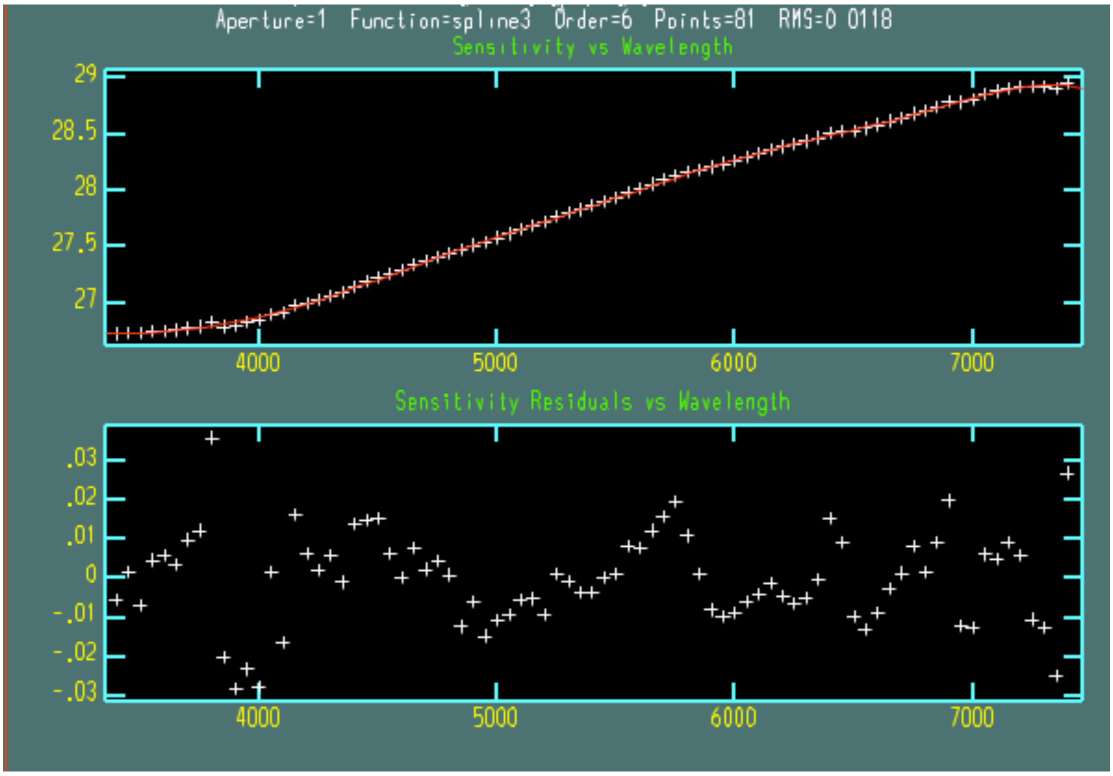
*file tLTT2415_20110124_Gr16_OG530_slit1.0_1_ex.fits already extracted*
*do you want to extract again [[y]/n] ? n*

*standard for setup ('Gr16', 'OG530', 'slit1.0') =  tLTT2415_20110124_Gr16_OG530_slit1.0_2.fits LTT2415*

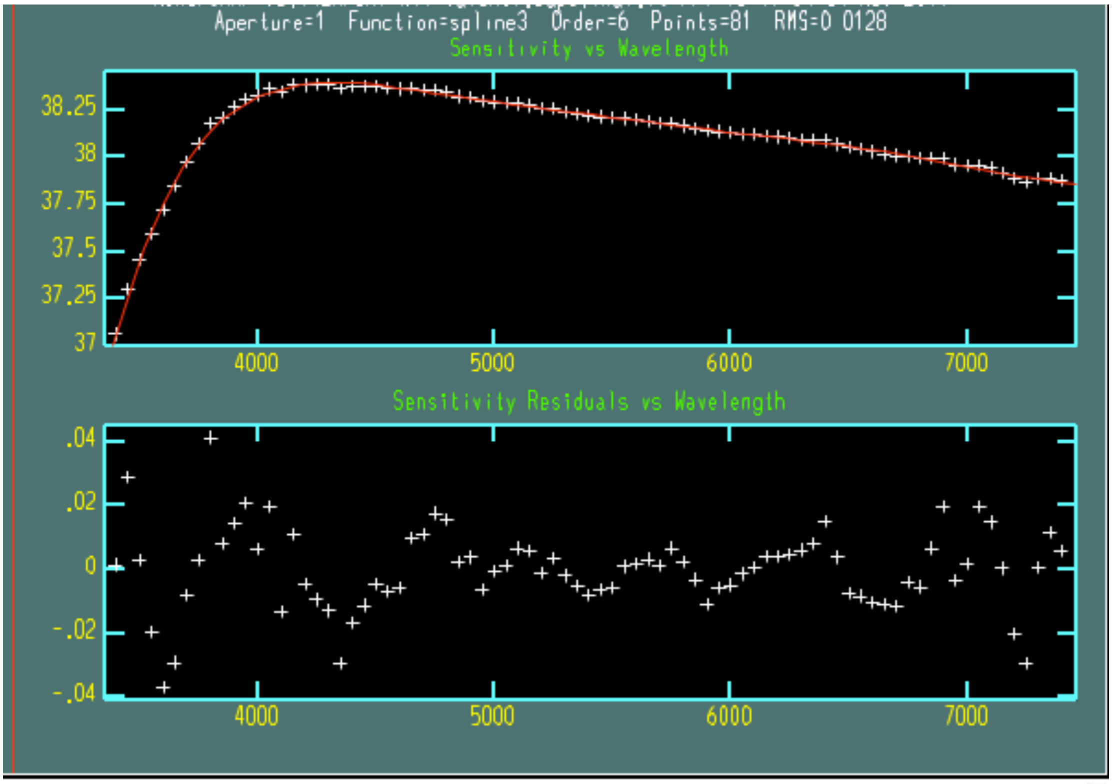
*Find apertures for tLTT2415_20110124_Gr16_OG530_slit1.0_2? ('yes'):*
*Number of apertures to be found automatically (1):*
*Edit apertures for tLTT2415_20110124_Gr16_OG530_slit1.0_2? ('yes'):*
*Trace apertures for tLTT2415_20110124_Gr16_OG530_slit1.0_2?Fit traced positions for tLTT2415_20110124_Gr16_OG530_slit1.0_2 interactively?Fit curve to aperture 1 of tLTT2415_20110124_Gr16_OG530_slit1.0_2 interactivelyWrite apertures for tLTT2415_20110124_Gr16_OG530_slit1.0_2 to databaseExtract aperture spectra for tLTT2415_20110124_Gr16_OG530_slit1.0_2?Review extracted spectra from tLTT2415_20110124_Gr16_OG530_slit1.0_2?Review extracted spectrum for aperture 1 from tLTT2415_20110124_Gr16_OG530_slit1.0_2?{('Gr16', 'OG530', 'slit1.0'):*

*['tLTT2415_20110124_Gr16_OG530_slit1.0_1_ex.fits',*
*'tLTT2415_20110124_Gr16_OG530_slit1.0_2_ex.fits']}*
*tLTT2415_20110124_Gr16_OG530_slit1.0_1_ex.fits*
*tLTT2415_20110124_Gr16_OG530_slit1.0_2_ex.fits*
*more than one standard for this setup, which one do you want to use ?*
*tLTT2415_20110124_Gr16_OG530_slit1.0_1_ex.fits*

The flux calibration is computed in 2 different ways:
**a)** dividing standard and object by a sensitivity function from archive in order that the sensitivity function is only a “f second order correction”.
(To be replaced, maybe, with the method suggested by Brian).




This is similar to the Brian method (improving the calibration on the edge), and it doesn’t have the following (possible) problem of the method suggested by Brian: If the standard used have low S/N ratio or the fringing has not been corrected in a proper way, this artifact are propagated in the SN spectrum.
**b)** This is the standard method, with the sensitivity function computed directly on the extracted standard spectrum




*# STANDARD: Observatory parameters for European Southern Observatory: La Silla*
*#           latitude = -29:15.4*
*tLTT2415_20110124_Gr11_Free_slit1.0_1_ex.fits:*
*tLTT2415_20110124_Gr11_Free_slit1.0_1_ex.fits[1]: Edit bandpasses? (no|yes|NO|YES|NO!|YES!) ('no'):*
*Fit aperture 1 interactively? (no|yes|NO|YES) (no|yes|NO|YES) ('yes'):*
*tLTT2415_20110124_Gr11_Free_slit1.0_1_f.fits: LTT2415*
*Extinction correction applied*
*Flux calibration applied*
*tSN2010jl_20110124_Gr11_Free_slit1.0_1_f.fits: SN2010jl*
*# CALIBRATE: Observatory parameters for European Southern Observatory: La Silla*
*#           latitude = -29:15.4*
*tSN2010jl_20110124_Gr11_Free_slit1.0_1_N.fits:   Extinction correction applied*
*Flux calibration applied*
*tSN2010jl_20110124_Gr11_Free_slit1.0_1_ff.fits: SN2010jl*
*# CALIBRATE: Observatory parameters for European Southern Observatory: La Silla*
*#           latitude = -29:15.4*
*tSN2010jl_20110124_Gr11_Free_slit1.0_1_ex.fits:   Extinction correction applied*
*Flux calibration applied*

The telluric features correction is performed grabbing the telluric features form the standard spectrum and dividing the flux calibrated spectrum by this telluric spectrum. This is applied only on the first dimension of the spectrum in order to keep the sky spectrum (for further wavelength check on the third dimension.

*tSN2010jl_20110124_Gr11_Free_slit1.0_1_f.fits[*,1,2] ->*
*tSN2010jl_20110124_Gr11_Free_slit1.0_1_e.fits[*,1,2]*
*tSN2010jl_20110124_Gr11_Free_slit1.0_1_f.fits[*,1,3] ->*
*tSN2010jl_20110124_Gr11_Free_slit1.0_1_e.fits[*,1,3]*
*tSN2010jl_20110124_Gr11_Free_slit1.0_1_f.fits[*,1,4] ->*
*tSN2010jl_20110124_Gr11_Free_slit1.0_1_e.fits[*,1,4]*

*tSN2010jl_20110124_Gr11_Free_slit1.0_1_ff.fits[*,1,2] ->*
*tSN2010jl_20110124_Gr11_Free_slit1.0_1_ee.fits[*,1,2]*
*tSN2010jl_20110124_Gr11_Free_slit1.0_1_ff.fits[*,1,3] ->*
*tSN2010jl_20110124_Gr11_Free_slit1.0_1_ee.fits[*,1,3]*
*tSN2010jl_20110124_Gr11_Free_slit1.0_1_ff.fits[*,1,4] ->*
*tSN2010jl_20110124_Gr11_Free_slit1.0_1_ee.fits[*,1,4]*

output product of 1D efosc spectra reduction:

tSN2010jl_20110124_Gr16_OG530_slit1.0_1.fits = 2D (input file)

tSN2010jl_20110124_Gr16_OG530_slit1.0_1_ex.fits
= 1D extracted spectrum

tSN2010jl_20110124_Gr16_OG530_slit1.0_1_N.fits
= extracted /archive sensitivity function (method a)

tSN2010jl_20110124_Gr16_OG530_slit1.0_1_f.fits
= flux calibrated (method a)
tSN2010jl_20110124_Gr16_OG530_slit1.0_1_ff.fits
= flux calibrated (method b)

tSN2010jl_20110124_Gr16_OG530_slit1.0_1_e.fits
=telluric features corrected (method a)
tSN2010jl_20110124_Gr16_OG530_slit1.0_1_ee.fits
=telluric features corrected (method b)

sens_Gr11_Free.fits = sensitivity function (method a)
sens2_Gr11_Free.fits = sensitivity function (method b)

# SOFI PHOTOMETRY

$ /home/users/site-package/NTT/bin/**PESSTOSOFIPHOT.py**
Usage: PESSTOSOFIPHOT.py -l listfile

> Fast photometry reduction of sofi images

```
Options:
  -h, --help            show this help message and exit
  -l INPUT, --list=INPUT
                        name list       [listfiles]
  -i, --interactive
  -f, --flat
  -F LISTFLAT, --listflat=LISTFLAT
                        name flat list  []
  -m, --doill
  -M LISTILL, --illuminationlist=LISTILL
                        name bad pixel mask     []
  -c, --docross
```

The fundamental input to make the script running is:
$ /home/users/site-package/NTT/bin/PESSTOSOFIPHOT.py –l listfile

The recommended syntax is:
$ /home/users/site-package/NTT/bin/PESSTOSOFIPHOT.py –l listfile -i

**option –l listfiles (**list of all the files)
**option –i** is to make some choice interactively
**option –f** is to skip the flat field correction
**option –F** listflat (list of flat or a single flat) is to use a specific flat

**option –m** is to skip the illumination correction
**option –M** listillumination (list of illumination correction files) is to use a specific illumination file
**option –c** is to skip the crosstalk correction

*output running the program:*
**1) splitting the files**
the files are splitted reading the setup and the OBs identification number. There is no output on the screen

**2) pre-reduction one by one the set of files (flatfield, crosstalk, llumination,trim)**

*next object*
*10jl_K_0001.fits 104.166667 104.166667*
*10jl_K_0002.fits 104.166667 -104.166667*
*10jl_K_0003.fits -104.166667 -104.166667*
*10jl_K_0004.fits -104.166667 104.166667*

*do you want to reduce this object SN2010jl_1 and filter Ks [[y],n] ?*

*10jl_K_0001.fits SN2010jl_20110125_Ks_1.fits SpecFlat_20110516_Ks.fits Illum_20110516_Ks.fits*

*CSN2010jl_20110125_Ks_1.fits: Nov  4 13:17 Trim data section is [1:1024,1:1007]*
*SpecFlat_20110516_Ks.fits: Nov  4 13:17 Trim data section is [1:1024,1:1007]*
*CSN2010jl_20110125_Ks_1.fits: Nov  4 13:17 Flat field image is SpecFlat_20110516_Ks.fits with scale=1.026512*
*CSN2010jl_20110125_Ks_1.fits: Nov  4 13:17 Illumination image is Illum_20110516_Ks.fits with scale=1.*

*10jl_K_0002.fits SN2010jl_20110125_Ks_2.fits SpecFlat_20110516_Ks.fits Illum_20110516_Ks.fits*
*CSN2010jl_20110125_Ks_2.fits: Nov  4 13:17 Trim data section is [1:1024,1:1007]*
*CSN2010jl_20110125_Ks_2.fits: Nov  4 13:17 Flat field image is SpecFlat_20110516_Ks.fits with scale=1.026512*
*CSN2010jl_20110125_Ks_2.fits: Nov  4 13:17 Illumination image is Illum_20110516_Ks.fits with scale=1.*

*10jl_K_0003.fits SN2010jl_20110125_Ks_3.fits SpecFlat_20110516_Ks.fits Illum_20110516_Ks.fits*
*CSN2010jl_20110125_Ks_3.fits: Nov  4 13:17 Trim data section is [1:1024,1:1007]*
*CSN2010jl_20110125_Ks_3.fits: Nov  4 13:17 Flat field image is SpecFlat_20110516_Ks.fits with scale=1.026512*
*CSN2010jl_20110125_Ks_3.fits: Nov  4 13:17 Illumination image is Illum_20110516_Ks.fits with scale=1.*

*10jl_K_0004.fits SN2010jl_20110125_Ks_4.fits SpecFlat_20110516_Ks.fits Illum_20110516_Ks.fits*
*CSN2010jl_20110125_Ks_4.fits: Nov  4 13:17 Trim data section is [1:1024,1:1007]*
*CSN2010jl_20110125_Ks_4.fits: Nov  4 13:17 Flat field image is SpecFlat_20110516_Ks.fits with scale=1.026512*
*CSN2010jl_20110125_Ks_4.fits: Nov  4 13:17 Illumination image is Illum_20110516_Ks.fits with scale=1.*

**3) sky-subtraction and merge of dithered images**

After all the set of images have been pre-reduced, the images are sky subtracted. For each set of images the observational strategy is recognize and the sky is computed. There are two observational strategies:

a) **DITHERING ON SOURCE:  dither smaller than 300 arcsec**

In this case the sky is computed using all the images of one cycle. Usually each cycle is formed by 4 images

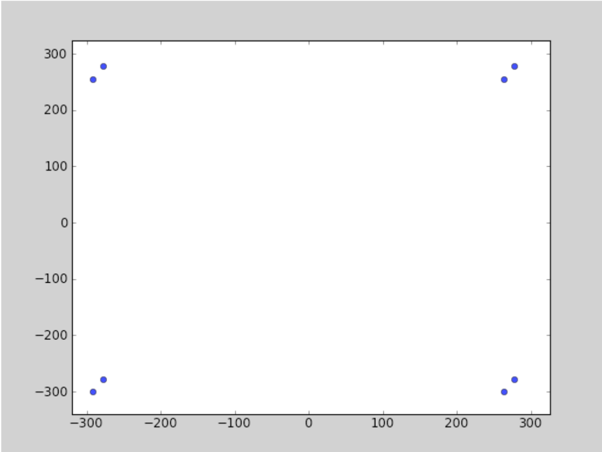
THE MASK USUALLY USED HAS 4 POSITIONS. Dither can be chosen following the observational strategy documentation.

REF

Anyhow, the mask positions are shown for each set of images (and printed on the screen) and the reducer has to confirm interactively the number of position used. For the faint objects the cycle will be repeated several time.  In the following example **the mask has 4 positions and the cycle has been repeated 2 times.** The reducer has to chose **4 positions.** In this way, the images will be split in two cycles and two sky images will be produced.

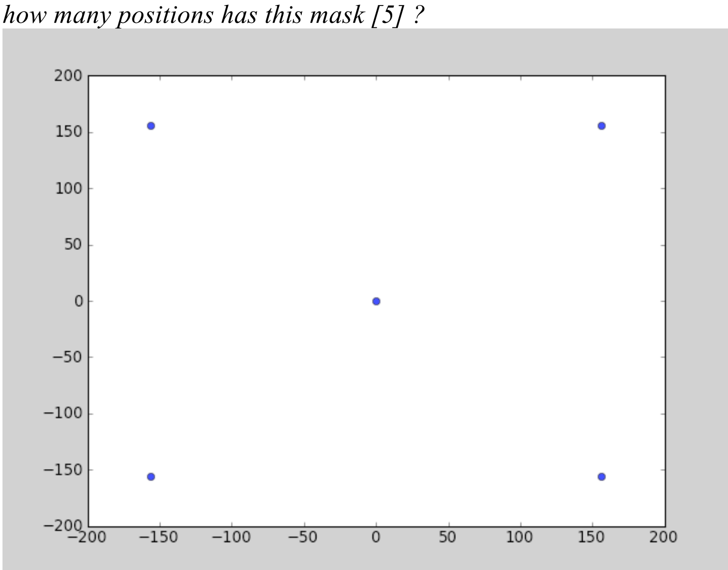
*Dithering on source*
*[277.77777800000001, 277.77777800000001, -277.77777800000001, -277.77777800000001, 263.77777800000001, 263.77777800000001, -291.77777800000001, -291.77777800000001]*
*[277.77777800000001, -277.77777800000001, -277.77777800000001, 277.77777800000001, 255.77777800000001, -299.77777800000001, -299.77777800000001, 255.77777800000001]*

*how many positions has this mask [8] ?* ***4***




In the second example the mask has 5 positions and one cycle. The reducer has to chose **5** positions

*STD_5 Ks*
*Dithering on source*
*[0.0, 156.25, -156.25, -156.25, 156.25]*
*[0.0, 156.25, 156.25, -156.25, -156.25]*
*how many positions has this mask [5] ?*




b) **ON-OFF: dither larger than 300 arcsec**

In this case the sky is computed with the images out of the field (off images). If the option is **interactive** the reducer has to confirm which are the images ON and OFF. Usually there are at least 3 images ON and 3 OFF.

*SN2010el_1 Ks*
*Warning: ON OFF*
*SN2010el_20100917_Ks_13.fits 0.0 0.0*
*is this ON[1] or OFF[2] ? [1]*
*SN2010el_20100917_Ks_14.fits -181.875042 1236.697808*
*is this ON[1] or OFF[2] ? [2]*
*SN2010el_20100917_Ks_15.fits 24.0 -6.0*
*is this ON[1] or OFF[2] ? [1]*
*SN2010el_20100917_Ks_16.fits 712.187918 -1027.272296*
*is this ON[1] or OFF[2] ? [2]*
*SN2010el_20100917_Ks_17.fits 19.0 23.0*
*is this ON[1] or OFF[2] ? [1]*
*SN2010el_20100917_Ks_18.fits 1241.737975 -143.481017*

*is this ON[1] or OFF[2] ? [2]*
*SN2010el_20100917_Ks_19.fits -17.0 -12.0*
*is this ON[1] or OFF[2] ? [1]*
*SN2010el_20100917_Ks_20.fits 841.883953 -923.975871*
*is this ON[1] or OFF[2] ? [2]*
*SN2010el_20100917_Ks_21.fits -20.0 27.0*
*is this ON[1] or OFF[2] ? [1]*
*SN2010el_20100917_Ks_22.fits 1172.49344 433.311819*
*is this ON[1] or OFF[2] ? [2]*
*SN2010el_20100917_Ks_23.fits -31.0 -31.0*
*is this ON[1] or OFF[2] ? [1]*
*SN2010el_20100917_Ks_24.fits 841.883953 -923.975871*
*is this ON[1] or OFF[2] ? [2]*

In both strategy a first sky is subtracted; The object are detected and an object mask is computed (for each image for the dither on source strategy and only for the OFF images in the ON/OFF strategy).
The mask images are used to mask the object in new sky images used to obtain again the final sky-subtracted images.
Using sextractor all the objects of the images are detected and aliened in order to combine the dithered images in a single image.

*----- SExtractor 2.5.0 started on 2011-11-05 at 00:27:51 with 1 thread*

*Measuring from: "SN2010hp" / 1024 x 1007 / 32 bits FLOATING POINT data*
*(M+D) Background: 1.79228 RMS: 3.76415 / Threshold: 5.64623*
*Objects: detected 111 / sextracted 90*
*> All done (in 0 s)*

***offsets -0.19 -0.03***
*----- SExtractor 2.5.0 started on 2011-11-05 at 00:27:52 with 1 thread*

*Measuring from: "SN2010hp" / 1024 x 1007 / 32 bits FLOATING POINT data*
*(M+D) Background: -0.137745 RMS: 3.34684 / Threshold: 5.02026*
*Objects: detected 154 / sextracted 120*
*> All done (in 0 s)*

***offsets -0.68 0.64***

The images are finally combined using the offset just obtained.
The output images are:

SN2010hp_20100917_Ks_19.fits trimmed image, illum cor., crosstalk
mask_SN2010hp_20100917_Ks_19.fits object mask for the image
sky_SN2010hp_20100917_Ks_19.fits sky-subtracted image
SN2010hp_20100917_Ks_merge_1.fits **merged image**

# SOFI SPECTROSCOPY

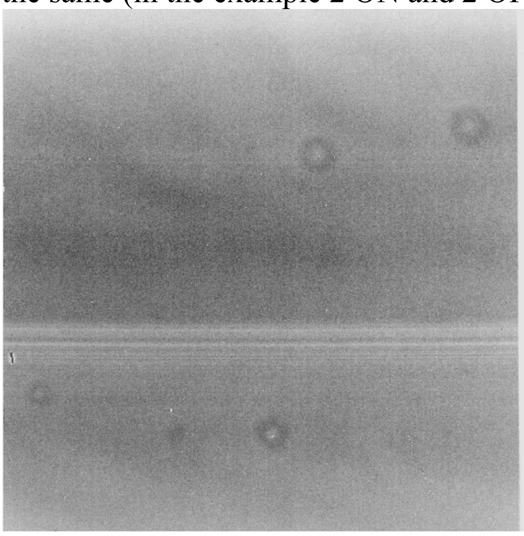
$ /home/user/site-packages/NTT/bin/**PESSTOSOFI2dSPEC.py**
Usage: PESSTOSOFI2dSPEC.py  -l listfile

> Specroscopic pre-reduction of sofi data

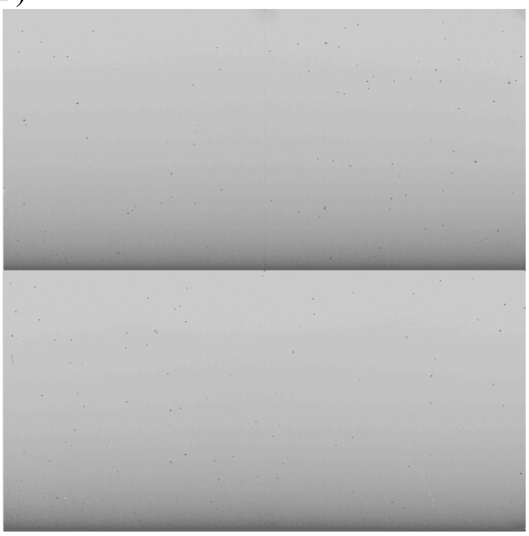
```
Options:
  -h, --help            show this help message and exit
  -l INPUT, --list=INPUT
                        name list       [listfiles]
  -i, --interactive
  -f, --flat
  -F LISTFLAT, --listflat=LISTFLAT
                        name flat list   []
  -c, --docross
```

**option –l listfiles (**list of all files)
**option –i** is to make some choice interactively
**option –f** is to skip the flat field correction
**option –F** listflat (list of flat or a single flat) is to use a specific flat
**option –c** is to skip the crosstalk correction

*output running the program:*
**1) spectroscopic flatfield:**
the reducer has to identify interactively, which flat filed are with the **Lamp ON** and **Lamp OFF** since there are no information in the header.
**Note:** the flat field will be computed only if the number of ON and OFF images are the same (in the example 2 ON and 2 OFF)



***FLAT LAMP ON***

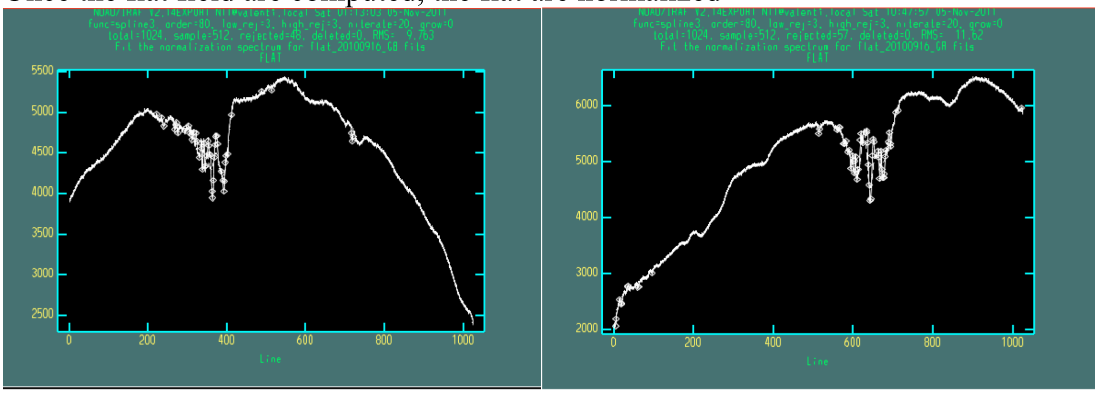


***FLAT LAMP OFF***

*SOFI.2010-09-16T20:50:11.441.fits*
*ON OFF*
*1 1*
*ON/OFF/REJECT/STOP [ON] ok [[y]/n/r/s]?*
*OFF GB*

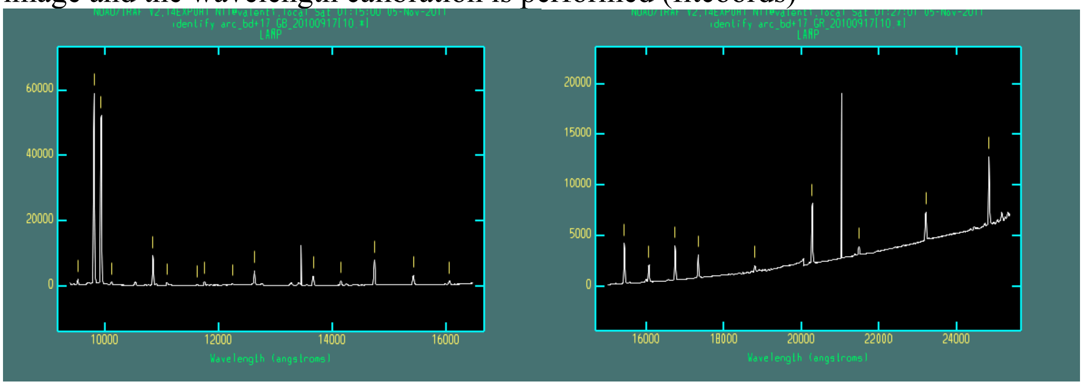
*SOFI.2010-09-16T20:50:37.753.fits*
*ON   OFF*
*2  1*
*ON/OFF/REJECT/STOP [OFF]  ok [[y]/n/r/s]?*
*ON GB*
*SOFI.2010-09-16T20:54:23.262.fits*
*ON   OFF*
***2  2***
*ON/OFF/REJECT/STOP [ON]*  ok [[y]/n/r/s]? **s**

Once the flat field are computed, the flat are normalized




## 2) arc wavelength calibration

For each set of images (each OBs), the closest ARC is selected and wavelength calibrated along one line. The line identification is then propagated along the 2D image and the wavelength calibration is performed (fitcoords)




*NOAO/IRAF V2.14EXPORT NTT@valenti.local Sat 10:49:29 05-Nov-2011*
*Longslit coordinate fit name is arc_SN2010hp_1_GB_20100916.*
*Longslit database is database.*
*Features from images:*
*arc_SN2010hp_1_GB_20100916*
*Map User coordinates for axis 2 using image features:*
*Number of feature coordnates = 1449*
*Mapping function = legendre*
*X order = 6*
*Y order = 4*
*Fitted coordinates at the corners of the images:*
*(1, 1) = 16466.51  (1024, 1) = 16472.26*
*(1, 1024) = 9375.287  (1024, 1024) = 9386.109*

## 3) object pre-reduction and wavelength calibration

The spectra of each OB are first corrected for CROSSTALK and than for FLATFIELD.

*add CSOFI.2010-09-17T03:57:37.961.fits,CROSSTALK = True*
*CSOFI.2010-09-17T03:57:37.961.fits updated*
*add CSOFI.2010-09-17T04:06:57.930.fits,CROSSTALK = True*
*CSOFI.2010-09-17T04:06:57.930.fits updated*
*add CSOFI.2010-09-17T04:16:08.065.fits,CROSSTALK = True*
*CSOFI.2010-09-17T04:16:08.065.fits updated*
*CSOFI.2010-09-17T03:57:37.961.fits: Nov 5 12:26 Flat field image is nflat_20100916_GB.fits with scale=1.*
*CSOFI.2010-09-17T04:06:57.930.fits: Nov 5 12:26 Flat field image is nflat_20100916_GB.fits with scale=1.*
*CSOFI.2010-09-17T04:16:08.065.fits: Nov 5 12:26 Flat field image is nflat_20100916_GB.fits with scale=1.*

For each exposure, the closest (in time) exposure is subtracted. If the option interactive is used the reducer has to confirm with is the image that should be subtracted.

*RCSOFI.2010-09-17T04:16:08.065.fits RCSOFI.2010-09-17T04:25:28.034.fits 94.88539 -86.350609 55456.6778711 55456.6778711*
*ok [[y]/n] ?*

The pre-reduced images are then wavelength calibrated using the calibrated arcs.

```
NOAO/IRAF V2.14EXPORT NTT@valenti.local Sat 12:58:43 05-Nov-2011
  Transform SN2010hp_1_20100917_GB_2_3.fits to tSN2010hp_1_20100917_GB_2_3.fits.
  Conserve flux per pixel.
  User coordinate transformations:
    arc_SN2010hp_1_GB_20100916
  Interpolation is spline3.
  Using edge extension for out of bounds pixel values.
  Output coordinate parameters are:
    x1 =        1., x2 =     1024., dx =        1., nx = 1024, xlog = no
    y1 =     9375., y2 =    16472., dy =     6.937, ny = 1024, ylog = no
Dispersion axis (1=along lines, 2=along columns, 3=along z) (1:3) (2):
```

**The output have the following form**:
t+name_object_date_filter_N_M.fits
where M is the number of the image subtracted to the image N.
tbd+17_20100917_GB_2_3.fits
tSN2010hp_1_20100917_GB_2_3.fits

# SOFI SPECTROSCOPY 1D

$ /home/user/site-packages/NTT/bin/**PESSTOSOFI1dSPEC.py**
Usage: PESSTOSOFI1dSPEC.py  -l listfile

~$ /home/user/site-package/NTT/bin/PESSTOSOFI1dSPEC.py
Usage: PESSTOSOFI1dSPEC.py  -l listfile

> Specroscopic reduction of sofi data (2D -> 1D)

```
Options:
  --version             show program's version number and exit
  -h, --help            show this help message and exit
  -l INPUT, --list=INPUT
                        name -l list     [listfiles]
  -i, --interactive
  -s LISTSTAND, --standard=LISTSTAND
                        name standard list       []
  -t, --trace
  -d, --dispersion
```

The fundamental input to make the script running is:
$/home/user/site-packages/NTT/bin/PESSTOSOFI1dSPEC.py –l list2D

The **recommended** syntax is:
~$/home/user/site-packages/NTT/bin/PESSTOSOFI1dSPEC.py –l list2D -i

**option –l listfiles (**list of  2D wavelength calibrated files)
**option –i** is to make some choice interactively
**option –s** use specific standard list  (**NOT IMPLEMENTED YET** )
**option –t**  For very faint object we do the trace with another object
**option –d** chose interactively the dispersion line will be plotted to chose the aperture and the background region.

*output running the program:*

The 2D wavelength-calibrated spectra are divided in
**Objects,**
**vega type standard**
**sun type standard**
**photometric type standard**

there are only (up to now) 3 photometric standard in the infrared

**gd71_stsci.dat  05 52 27.51     +15 53 16.6     13.03**
**gd153_003.dat   12 57 02.34     +22 01 52.7     13.40**
**bd174708003.dat 22 11 31.37     +18 05 34.174   9.45**
also available at this link
https://psweb.mp.qub.ac.uk/pesstowiki/index.php/File:Standard_phot.pdf

the standard sun type and vega type are mainly used for telluric corrections, while the photometric standards (if available) are used to check the flux calibration.

A list of telluric standard vega type and solar type is available at the following links:

**Sun type standard**:
https://psweb.mp.qub.ac.uk/pesstowiki/index.php/File:Standard_sun.pdf

**Vega type standard:**
https://psweb.mp.qub.ac.uk/pesstowiki/index.php/File:Standard_vega.pdf

*photometric standard in the list of object*
*telluric G standard (sun type) in the list of object*
*warning: not telluric A standard (vega type)*

**1) combining spectra**
all the spectra of one OB are combined in one:

*setup= ('GB', 'GBF', 'long_slit_1') name field= hip1467_1 merge image= hip1467_1_GB_merge.fits*
*setup= ('GB', 'GBF', 'long_slit_1') name field= SN2010hp_1 merge image= SN2010hp_1_GB_merge.fits*
*setup= ('GR', 'GRF', 'long_slit_1') name field= bd+17_2 merge image= bd+17_2_GR_merge.fits*

**2) extraction object spectrum, telluric standard spectrum**
If the option interactive is used, the spectrum is extracted interactively.
If the spectrum has been already extracted, there is the possibility to skip this step.

*next object*
*SN2010hp_1_GB_merge.fits SN2010hp*

*file SN2010hp_1_GB_merge_ex.fits already extracted*
*do you want to extract again [[y]/n] ?*
*Recenter apertures for SN2010hp_1_GB_merge? ('yes'):*
*Resize apertures for SN2010hp_1_GB_merge? ('yes'):*
*Edit apertures for SN2010hp_1_GB_merge? ('yes'):*
*Trace apertures for SN2010hp_1_GB_merge?Fit traced positions for SN2010hp_1_GB_merge interactively?Fit curve to aperture 1 of SN2010hp_1_GB_merge interactivelyWrite apertures for SN2010hp_1_GB_merge to databaseExtract aperture spectra for SN2010hp_1_GB_merge?Review extracted spectra from SN2010hp_1_GB_merge?Review extracted spectrum for aperture 1 from SN2010hp_1_GB_merge?*

*##### closer standard for telluric corrections ####*

*hip1467_1_GB_merge.fits 1.004*

*file hip1467_1_GB_merge_ex.fits already extracted*
*do you want to extract again [[y]/n] ? n*

**3) photometric calibration**
If a photometric standard has been observed, the flux calibration is checked with those of the standard.

*#####  photometric calibration   ######*

*file bd+17_3_GB_merge_ex.fits already extracted*
*do you want to extract again [[y]/n] ? n*

*file bd+17_2_GB_merge_ex.fits already extracted*
*do you want to extract again [[y]/n] ? n*

*['bd+17_3_GB_merge_ex.fits', 'bd+17_2_GB_merge_ex.fits', 'bd+17_1_GB_merge_ex.fits']*

*which one do you want to use [bd+17_3_GB_merge_ex.fits] ?*
*bd+17_3_GB_merge_ex.fits*
*bd+17_3_GB_merge_f.fits*

*flux calibrated spectrum=  SN2010hp_1_GB_merge_f.fits  with the standard=*
*bd+17_3_GB_merge_f.fits*

**3) output:**
The flux calibrated spectrum will end with **'_f.fits'**
If the photometric standard is not available the output spectrum will end with **'_ex.fits'**

This steps are repeated for each spectrum and setup.