



<i>Publication Year</i>	2023
<i>Acceptance in OA</i>	2023-04-17T12:51:07Z
<i>Title</i>	Reducing GIANO-B spectra with IRAF. The GIANOTOOLS handbook
<i>Authors</i>	MASSI, Fabrizio
<i>Handle</i>	http://hdl.handle.net/20.500.12386/34056

REDUCING GIANO-B SPECTRA with IRAF

The GIANOTOOLS HANDBOOK

v1.1

(January 2023)

F. Massi¹ (fabrizio.massi@inaf.it)

¹INAF - Osservatorio Astrofisico di Arcetri

Abstract

*GIANO is a near infrared echelle spectrograph operating at Nasmyth focus A of Telescopio Nazionale Galileo. It is a second generation instrument and allows obtaining spectra that cover most of the range 0.9–2.5 μm with a single exposure. The spectral interval is covered through 49 orders and only a few small windows fall outside the detector. **GIANO-B is an upgrade of GIANO and has been available since 2017. Unlike GIANO, it is not fibre-fed but light is now transferred to the slit through pre-slit optics. This document is the updated version of the first “Reducing GIANO Spectra” handbook and explains how to reduce and extract GIANO-B calibrated spectra by using routines available in any basic installation of IRAF, the Image Reduction and Analysis Facility developed by the NOAO, plus the suite GIANOTOOLS of ad-hoc scripts, the latest version of which can be obtained from the author of this handbook. IRAF is one of the most commonly used scientific suites of software packages for reducing and analysing images and spectra, freely downloadable, and offering many routines which are both fully optimised and tested. Its versatile Command Language also allows scripting and easy customisation.***

Reduction quick guide

Some IRAF basics for new users are dealt with in Sect. 2. The reduction pipeline described in this handbook is summarised below.

1. Install the small set of scripts GIANO_TOOLS as explained in Sect. 3.1.
2. Open an IRAF session and load GIANO_TOOLS (this will also have ASTUTIL, IMRED, CRUTIL, and ECHELLE loaded).
3. Move to your work directory where the data are stored.
4. Update the headers of the reference wavelength lamp frames, the target frames, the telluric star frames, and the spectro-photometric star frames (if any) using GIANO_HEADER, as explained in Sect. 3.3.
5. Average together the dark frames with the same integration time using IMCOMBINE.
6. Subtract the corresponding dark frame from the flat-field frames and the single science or calibration frames **which were not taken in AB cycles, i. e. by shifting alternatively the target between two different positions in the 6 arcsec-wide slit**, using IMARITH.
7. Perform bad-pixel correction of all flat-field frames, reference lamp frames, science frames, and calibrator frames with clean_up as explained in Sect. 3.4.
8. Average together the flat-field frames with IMCOMBINE.
9. Set the ECHELLE parameters as discussed in Sect. 3.9.
10. Find the traces with APFIND as explained in Sect. 3.10 and Appendix A. Set the parameters as in Fig. 8. See Figs. 21, 22, and 23 for a correct order numbering.
11. Fit the traces with APTRACE as explained in Sect. 3.11. Set the parameters as in Fig. 10. See Figs. 24 and 25 for examples of actual trace fits.
12. construct a normalised 2D flat-field frame from the average flat-field frame as explained in Sect. 3.12 and Appendix B. See Fig. 12 on how to set the parameters for APFLATTEN. See Figs. 13 and 14 for an example of a 2D normalised flat-field frame.
13. Divide all corrected reference lamp frames, science frames, and calibration frames by the normalised flat-field frame using IMARITH.
14. Subtract the corrected frame B in every AB cycle set from the corresponding corrected frame A using IMARITH.
15. Clean the subtracted frames from cosmic rays and other spikes with the task GIANO_COSMICRAYS, as explained in Sect. 3.13.
16. In case of more than 1 AB cycle on the same target, average or sum together the subtracted frames using IMCOMBINE. See Appendix E to compute the effective exposure time, gain, and read-out noise. Do the same with dark-corrected frames of sources not observed in AB cycles.

17. Extract and wavelength-calibrate the 1D spectra from the positive tracks in 2D A–B spectra as explained in Sect. 3.14. Set the parameters as in Fig. 16 for DOECSLIT and Fig. 17 for SPARAMS. Recentre the apertures on the target signal and resize them to fit the signal spatial profile as explained in Appendix A. See examples in Fig. 18. If you need to subtract a background (only frames where the target could be imaged in only one slit location), go on to the background selection session as explained in Sect. 3.14 and Appendix A, and shown in Fig. 19. Check that the apertures follow reasonably well the signal by changing the plotted detector column (note that the spectral order tracks are curved). Then go on to wavelength calibrate the spectra as explained in Appendix F and Appendix G. **It is mandatory that you use the reference lines shown in Figs. 32, 33, and 34 to obtain an accurate wavelength calibration.** See Fig. 29 for a correct wavelength solution.
18. Now extract and wavelength-calibrate the 1D spectra from the negative tracks in 2D A–B spectra multiplying them by -1 with IMARITH and using again DOECSLIT. Note that the task will skip the wavelength calibration session, using the results from the previous run. We advice that you select the same aperture widths for A and B spectra (see next item).
19. use SCOMBINE to combine A and B spectra as shown in Sect. 3.15. Set the various task parameters as in Fig. 20. This will further decrease the effects of faint background radiation underlying the target signal.

Troubleshooting

This handbook also deals with a few problems that may arise during the reduction steps; just look for specific boxes labelled as “Remark” or “Problem solving”. Here is a brief summary.

1. **The graphic or image cursor does not work** See the REMARK on page 3.
2. **Error when loading GIANO_TOOLS in IRAF** See the REMARK on page 6.
3. **Some ECHELLE tasks do not run using correct parameters** See the REMARK on page 13.
4. **The wavelength calibration is not correct in small intervals inside some orders** Use all the reference lines suggested in Appendix G for the preliminary wavelength fit with DOECSLIT.
5. After finding a good fit to the reference wavelength-calibration lamp lines in DOECSLIT and everything seems to have been ok, nevertheless some of (or all) the extracted spectra have not been wavelength-calibrated. Most probably you have used names too long for your reduced frames. Be aware that DOECSLIT produces auxiliary files whose names are built by joining names of two frames (target plus reference lamp). Some of these files are used as reference by the task and their names are input in the headers of extracted spectra. So, if the names are too long, they are stored truncated in the corresponding header field and cannot be retrieved by DOECSLIT when applying wavelength calibration.

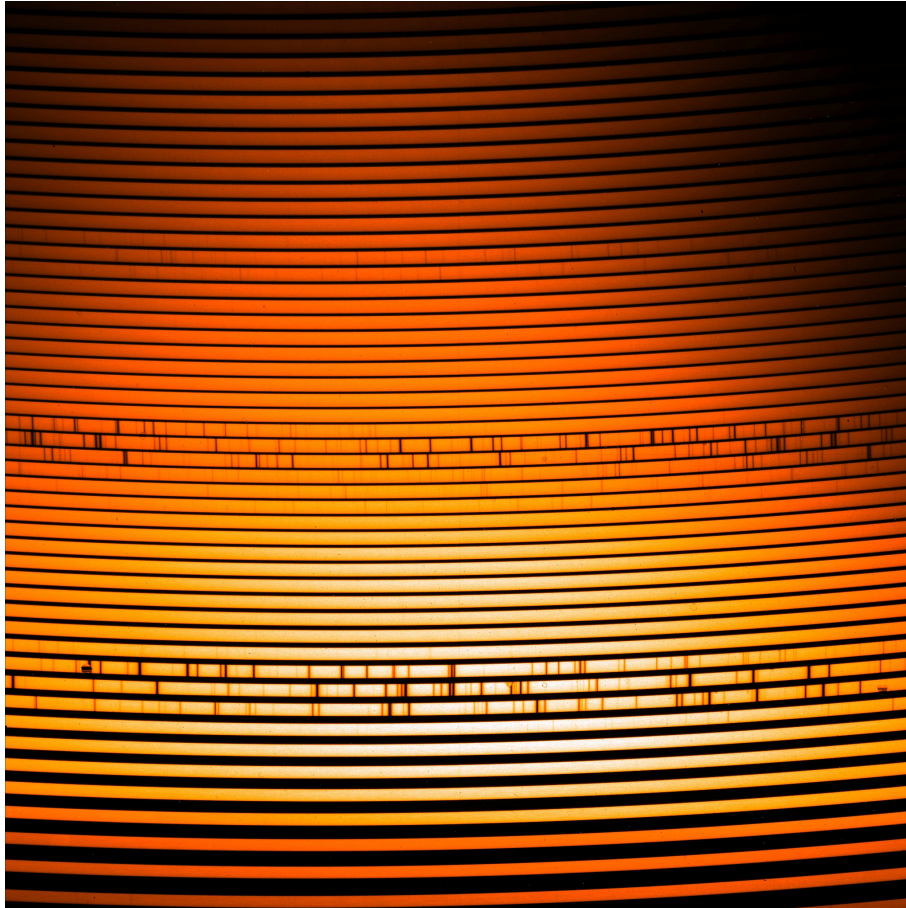


Figure 1: Frame with the spectrum (echellogram) of a continuum lamp obtained with GIANO-B.

1 Starting point: GIANO-B raw data

A typical 2D output from a GIANO-B integration is shown in Fig. 1. The frame contains 49 wide (~ 26 pixels) arc-shaped tracks (plus a couple of shorter segments actually). Each track represents a spectral order and contains a part of the spectrum falling inside a sub-range of the total pass-band. The wavelength intervals per order are listed in Appendix H. The slit width is ~ 6 arcsec on-sky, this is why the tracks are broad. The detector parameters are listed in Table 1.

The lowest order (at the bottom of the frame) is the 32nd and the highest one (at the top of the frame) is the 80st. Although you can count 50 arc-shaped tracks, which includes part of order 81, as well, only orders 32 through 80 are offered. Within each order, wavelength increases from right to left. In addition, wavelength decreases with increasing order (see Appendix H). One of the goals of the reduction procedure is to find the curve that best approximates each track shape and to count the signal as a function of position within “apertures” centred on these curves (called *traces*). More information on the instrument can be found in [1].

A run with GIANO-B will yield images recording the target spectra (2D spectra), as well as a few additional “calibration” frames. These are needed to properly extract the target spectra. Below, we list the sets of calibration frames necessary for accurate extraction and calibration.

DARK FRAMES these are obtained by integrating a signal while holding the shutter closed. Science targets are usually observed by nodding along the slit, i. e. by performing an integration with the target in a position A first and in a position B next (hereby, a cycle AB). After subtracting B from A, the resulting 2D spectra will both exhibit the target signal in all tracks and have most of the bias

Table 1: Main characteristics of the GIANO-B NIR detector (from Table 1 of [2]).

gain	2.2	$e^- \text{ ADU}^{-1}$
Read-Out Noise	5	e^-
Non-linearity	$1500 \times t_{\text{exp}}$	ADU

¹ t_{exp} total integration time in sec.

removed, so no dark subtraction is required in this case. On the other hand, 2D spectra not taken in AB pairs (flats, reference lamps, etc.) need being dark subtracted. Sets of dark frames must be acquired for all exposure times of non-AB 2D spectra.

FLAT FIELD FRAMES these are obtained by illuminating the slit with an intense continuum lamp. Flat field frames are needed not only to remove the pixel-to-pixel inhomogeneities in the detector response, but also to obtain a trace template that will be then used on the usually noisier and much fainter scientific 2D spectra to extract the 1D spectra.

REFERENCE LAMP SPECTRA reference lamp spectra are obtained by illuminating the slit with an intense EMISSION LINE lamp, (GIANO-B is currently equipped with a U-Ne calibration unit). These frames are required for wavelength-calibration of the spectra.

TARGET AND CALIBRATOR (TELLURIC OR SPECTRO-PHOTOMETRIC STARS) SPECTRA these are usually pairs of AB, nodded integrations, but you might also have 2D spectra taken with the target in only one position on the slit (possibly with corresponding 2D spectra with pure sky emission). If no sky frame is available, subtraction of dark frames with the same integration time are mandatory. See [2, 3] for more details on the observational techniques and the recommended integration times.

REMARK All GIANO-B observations are saved as FITS files.
--

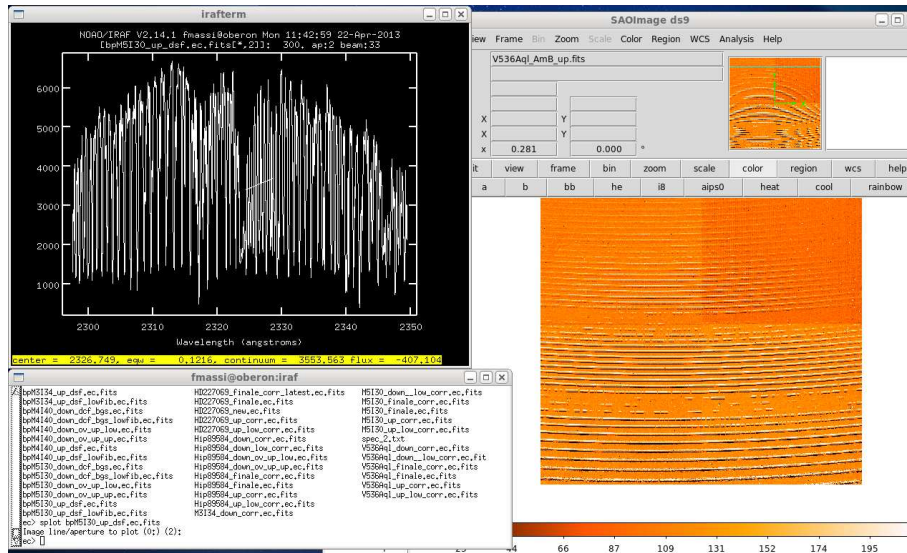


Figure 2: The three basic interfaces used in a typical IRAF session: an xgterm window for running commands and tasks (bottom left), an image tool (here ds9, right), and a graphic window (top left).

2 IRAF basics

This section is only intended as a quick introduction to allow new users to run a simple IRAF session. For more information, we recommend that you should read [4] and [5].

You need to have IRAF installed in your PC or network. We also suggest that you install x11iraf (providing you with xgterm terminal windows) and ds9 (to use as the image tool).

IRAF and x11iraf can be downloaded from
<http://iraf.noao.edu>
ds9 can be downloaded from
<http://hea-www.harvard.edu/RD/ds9/site/Home.html>.

Once IRAF is installed on your PC, you just need to create your IRAF home directory, the one from which you will always start IRAF, which should not coincide with your work directories. Then, run **mkiraf** on that directory and answer the questions. The most critical one is that about the terminal type. If you have x11iraf installed, choose “xgterm”. Now is all set up for running IRAF.

Basic IRAF session: first, open an xgterm terminal window. Then, start ds9. On the xgterm window, move to your IRAF home directory and type “cl”. The IRAF session will open up and you will be shown the prompt “cl>”. From the prompt you can run line commands and tasks. A typical IRAF session uses three windows (see Fig. 2): an xgterm one with the CL running, an image tool (ds9 in figure) and a graphic tool. The graphic tool will open up when you first run a task requiring it.

REMARK Lots of IRAF tasks are interactive and they can call either an image cursor or a graphic cursor. These allow you to point specific features inside the image tool (image cursor) or the graphic window (graphic tool) and issue interactive commands (usually by pressing sequences of keys).

BE CAREFUL: if you need to run interactive commands, first take care to click the mouse left button with the cursor inside the image tool or the graphic window, depending on whether you

are using the image or the graphic cursor. Then (and only then), issue the command. Remember to always have the proper window active every time you run an interactive command! When you need to return to the CL, use the interactive command “q”.

Once cl is running, you can move to your work directory by using “cd”. Some shell commands like ls, mkdir, etc. are recognised within cl. In any case you can run any external command or program by typing a “!” in front of the command line to be entered. Type in “logout” to quit cl.

IRAF is a bundle of layered programs divided into several packages. Only a limited number of basic packages will be available and running (i. e. , “loaded”) as soon as cl is opened. To run any task from any other package, you first need to load the whole package by simply typing its name and pressing enter. In this case, the prompt changes from “cl>” to another string with the first two characters of the package name. More than one package can be loaded simultaneously (this increases memory allocation; it should be less than a problem on modern PCs but IRAF may meet with some problems). The latest package loaded can be unloaded by typing “bye”. Here is an example of loading a package (in this case, noao). Note that as soon as the package is loaded the list of its packages and/or tasks will be shown. Note also that the prompt has changed:

```
cl> noao
      artdata.    digiphot.  nobsolete.    onedspec.
      astcat.     focas.     nproto.       rv.
      astrometry. imred.     observatory  surfphot.
      astutil.    mtlocal.   obsutil.     twodspect.
no>
```

A help facility is available, just type “help nametask” in cl to find out the help pages about the task NAMETASK.

Most IRAF tasks need a few parameters being set before running. The simplest way to do this is by using the IRAF parameter editor. You can access the NAMETASK parameter file through the editor by typing “epar nametask” in cl (for an example see Fig. 3). A number of fields will be shown in the cl window. You can select any of them by using the arrow keys and edit it (just type in and press enter). Press CTRL-d or type “:wq” to exit the parameter editor, or type “:go” to exit the parameter editor and make the task run.

Alternatively, you can run a task by simply typing its name and pressing enter. IRAF remembers the latest set of parameters entered for a given task *through the parameter editor*, so by default the task will run using that set. Some tasks can run in verbose mode; in this case you will be asked to confirm each parameter of the set. You will be able to either just confirm a parameter or enter a new value. Parameters can also be passed to a task via a line command (e. g., by typing

“nametask par1 par2 ...”,

or

“nametask field1=value field2=value ...”).

You can reset all the parameters of a given task to their default values by typing “unlearn nametask” at the cl prompt.

REMARK Most IRAF tasks can accept, as input and output fields: single file names, multiple file names separated by commas, single text file names including lists of files. The latter mode is useful to operate on whole blocks of spectra in a single shot. Files have to be listed one per line in

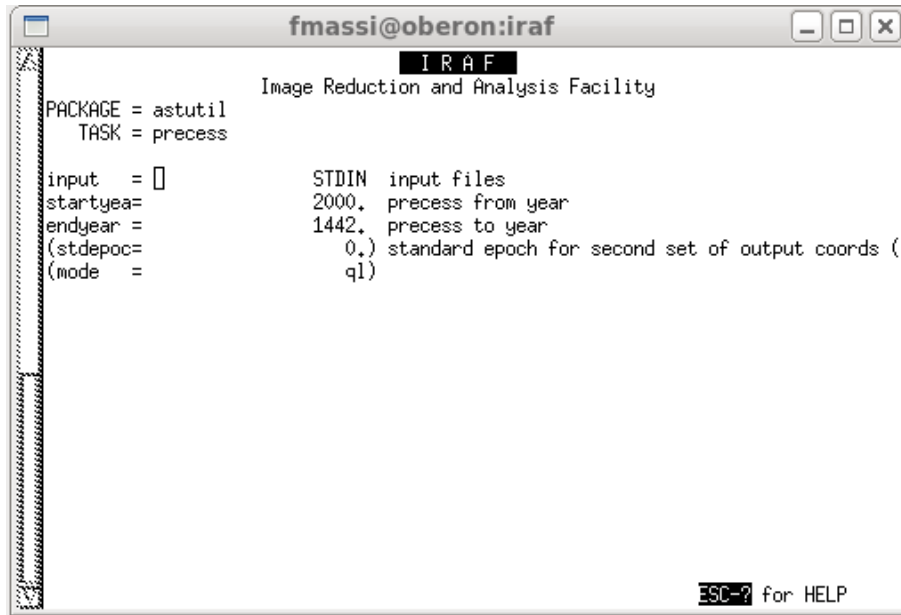


Figure 3: IRAF parameter editor opened on the parameter file of the task “precess”. You can move through the fields by using the arrow keys and you can input some text in the field evidenced by the cursor. The old text will automatically be replaced with the latest one after pressing ENTER.

the text file, then the task has to be informed that the given name refers to a list of files by typing a “@” in front of the file name. For example, by typing:

```
cl> imstatistics @image.list
```

the task IMSTATISTIC is run to yield the statistics of all images listed in the file image.list.

3 Data reduction with IRAF

3.1 Setting up an IRAF session to reduce data from GIANO-B

After running MKIRAF on your IRAF home directory (see page 3), you will be able to find a file called “login.cl” there, containing some basic paths, a few settings, and the tasks to be loaded whenever cl is run. Lines commented out have a hashtag # as the line first character, so they are not read by cl at start time.

Edit your command.cl file, find the line

```
set stdimage = imt...
```

check that it is **not** commented out and if necessary modify it to:

```
set stdimage = imt2048
```

Then check that the line

```
set imtype = “imh”
```

is **commented out**, so that the default image file format is fits and not imh. You will no longer need to type in the full name of an image including the extension “fits”, IRAF will now be able to recognise a fits file unless it is a fits file using a different extension (e. g., fts; see page 7 to change the extension).

You will also need to install a small package (called GIANO_TOOLS) containing a few scripts which we have developed to the purpose and is provided in the form of a tar file. Make a new directory to store the package, move the tar file there and untar it. One of the extracted files is called giano_tools.cl. Edit giano_tools.cl and change the line

```
set gianot = “put here the correct path to giano_tools”
```

by inserting the actual full path between quotes. **Make sure that the last character of the path is a “/”**. Finally, edit the login.cl file in your IRAF home directory and add the line

```
task $giano_tools = path/giano_tools.cl
```

where path is the full path to the directory where you have untarred the package. Now you will be able to load the package by simply typing “giano_tools” in cl. Note that the package also contains a fits file with an updated map of detector bad pixels (badpix_mask.fits), a fits file with an updated wavelength-calibrated lamp spectrum (cl_29_jul_UNe_300_ds_lowest.ec.fits) and its wavelength-solution file (eccl_29_jul_UNe_300_ds_lowest.ec), and a list of reference wavelengths for calibrating spectra using a U-Ne-Ar lamp (U_Ne_Ar_lines_list.cl.dat).

REMARK: INSTALLATION ERRORS Sometimes, after installing GIANO_TOOLS we got error messages when loading it, due to hidden characters inserted in giano_tools.cl by the text editor used to edit it. Should this happen, just try editing the original file with a different text editor.

3.2 Giano_tools

GIANO_TOOLS is a collection of IRAF scripts we have developed to adapt IRAF tasks and reduce GIANO and GIANO-B spectra. They are provided as a tar file and include the following files:

```
badpix_mask.fits
cl_29_jul_UNe_300_ds_lowest.ec.fits
clean_up.cl
copy_file.cl
eccl_29_jul_UNe_300_ds_lowest.ec
fits2text.cl
flat_1D.cl
giano_cosmicrays.cl
giano_find_trace.cl
giano_header.cl
giano_reidentify.cl
giano_tools.cl
split_file.cl
U_Ne_Ar_lines_list.cl.dat
```

3.3 First reduction steps

Each night run will yield a number of dark frames, flat-field frames, line reference lamp frames and scientific frames. We suggest that you should make a directory for each night and save the corresponding frames there. We assume that you have taken one complete set of calibration frames every night. If possible, keep a copy of each of these directories in your disk, since a few tasks you are going to use will modify the frames they operate on. If something goes wrong, it will sometimes be quicker to just erase the directory, e. g. with “rm -r /path-to-yourdirectory/*”, and copy there the original frames again.

Open an xgterm window, change the current directory to your IRAF home directory in the new window, and run cl. Within the cl prompt, change directory to the one where the data are stored. If their extension is not fits, e. g. fts, you need to change it to fits. This is easy done by the line command:

```
cl> rename *.fts fits field=extn
```

Now, you need to load the packages ASTUTIL, IMRED, ECHELLE, CRUTIL, and GIANO_TOOLS. But typing giano_tools is enough, since it will automatically load the other three packages.

Before starting to work on the spectra, you will also need to add a few fields to the headers of the reference lamp spectra, the target spectra, and any calibrator spectra. Just list all lamp reference frames in a text file (e. g., lamp.list), one per line. Then list all target and calibrator spectra in another one (e. g., target.list). Edit the parameter file of the GIANO_TOOLS task called GIANO_HEADER (running “epar giano_header”) and set *list_obj* to target.list, *list_com* to lamp.list, *obsvt* to lapalma, and *equi* to 2000. Therefore, the editor page must look as follows:

```
PACKAGE = giano_tools
TASK = giano_header
list_obj= target.list  File listing the target spectra:
list_com= lamp.list    File listing the comparison spectra:
obsvt =   lapalma      Observatory (from the IRAF observatory db):
equi =   2000.          Equinox which RA and DEC in the header are referred to:
(mode = q)
```

Exit starting the task by typing “:go”. Run the task twice, just to make sure everything is updated as needed.

3.4 Dark subtraction and bad-pixel correction

Dark subtraction is only needed by those frames that have not been taken in AB cycles (i. e., pairs of frames obtained by shifting the target along the slit will be subtracted from each other and do not need dark-subtraction). These include flat-fields, lamp reference spectra, and every target frame with spectra from only one position on the slit. **We stress again that the dark frames to subtract must have been taken with the same integration time as the image to be dark-corrected!** Thus, group together all dark frames according to their integration time. Check that none of the dark frames exhibit any kind of problems by displaying them (use the task DISPLAY) and having a look at the image statistics (use the task IMEXAMINE). Let us assume that all dark frames with an integration time of 300 s (e. g., dark_300_1.fits, dark_300_2.fits, dark_300_3.fits) are listed in a file called dark_300.list:

```
cl> display dark_300_1 1
cl> display dark_300_2 2
cl> display dark_300_3 3
cl> imstat @dark_300.list
```

The last number in each of the display line command is a frame identifier. E. g., in practice IRAF allows nine frames (1,2,3,4,5,6,7,8,9) to be displayed simultaneously on ds9. You can use the ds9 key “blink” (in the menu “frame” of ds9, see Fig. 4) to iteratively display all open frames, or you can shift from one frame to another by using “next” (in the menu “frame” of ds9). Drop any file from the dark_300.list that exhibits some strange patterns or statistics. Then combine the remaining ones together, using the task IMCOMBINE after updating the list of frames to be combined.

```
cl> imcombine @dark_300.list dark_300_combined combine=average
```

This will yield the file dark_300_combined.fits, which has to be subtracted from each frame with an integration time of 300 s needing dark correction.

The fastest way of dark-correcting all frames that need it, is by listing them (one per line) in a text file (e. g., to_be_dcorrected.list). Then do a copy of to_be_dcorrected.list (e. g., dcorrected.list), edit this new file, and change the name of each item listed (e.g., by adding the string “dc_” in front of each file name). Now you will be able to dark-subtract all files just by typing:

```
cl> imarith @to_be_dcorrected.list - dark_300_combined @dcorrected.list
```

This computes the dark-subtracted frames and saves them adding a “dc_” in front of their original file names (e. g., target_1.fits will yield dc_target_1.fits).

After dark subtraction, We suggest that you apply bad-pixel corrections to all flat-field frames, calibration-lamp frames, target frames and calibrator frames. Bad pixels cause a lot of annoyance when using tasks that perform fits to pixel rows or columns. The correction can be done by using the GIANO_TOOLS task CLEAN_UP. Just make a text file listing all frames to be bad-pixel corrected (e. g., call it to_be_flatted.list). Then open the clean_up parameter file by typing “epar clean_up”. Set the fields as below:

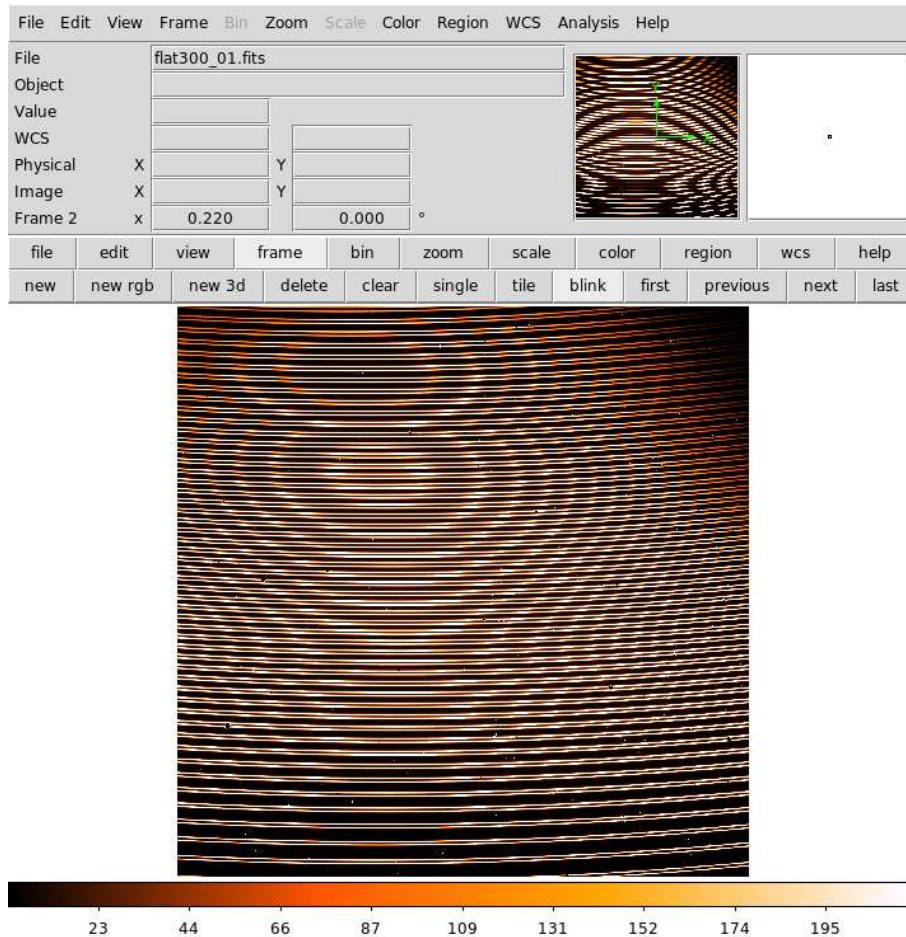


Figure 4: How to run “blink” on ds9: first click on *frame* on the upper part of the bar above the image. This will change the lower part of the bar and allow you to click on *blink*. Click on *single* to exit from blink mode.

```

PACKAGE = giano_tools
TASK = clean_up
  imlist=   to_be_flatted.list      File with list of images:
  outstr=   cl_                    String to be added to input image name:
  badima=   gianot$badpix_mask.fits Bad pixel image:
  xbox=     40                     Size of median-filter box along X:
  ybox=     1                      Size of median-filter box along Y:
  fix=      no                     Use fixpix?:
  fixfile =                               List of bad pixel masks for fixpix:
(mode = q)

```

When you have finished, exit by typing “:go” to start the task. It will construct bad-pixel corrected frames and save them adding the string “cl_” (you can choose any string you want, just update the field *outstr*) in front of the names of the input frames. The task uses a bad-pixel map already provided with GIANO_TOOLS (badpix_mask.fits) to replace each mapped bad pixel with an xbox-by-ybox median value around it. If you want, you can also have clean_up run the IRAF task FIXPIX (by setting *fix* to yes) to correct groups of pixels whose coordinates must be listed in a file (remember to enter its name in *fixfile*). See the IRAF help for further information on fixpix and the file format.

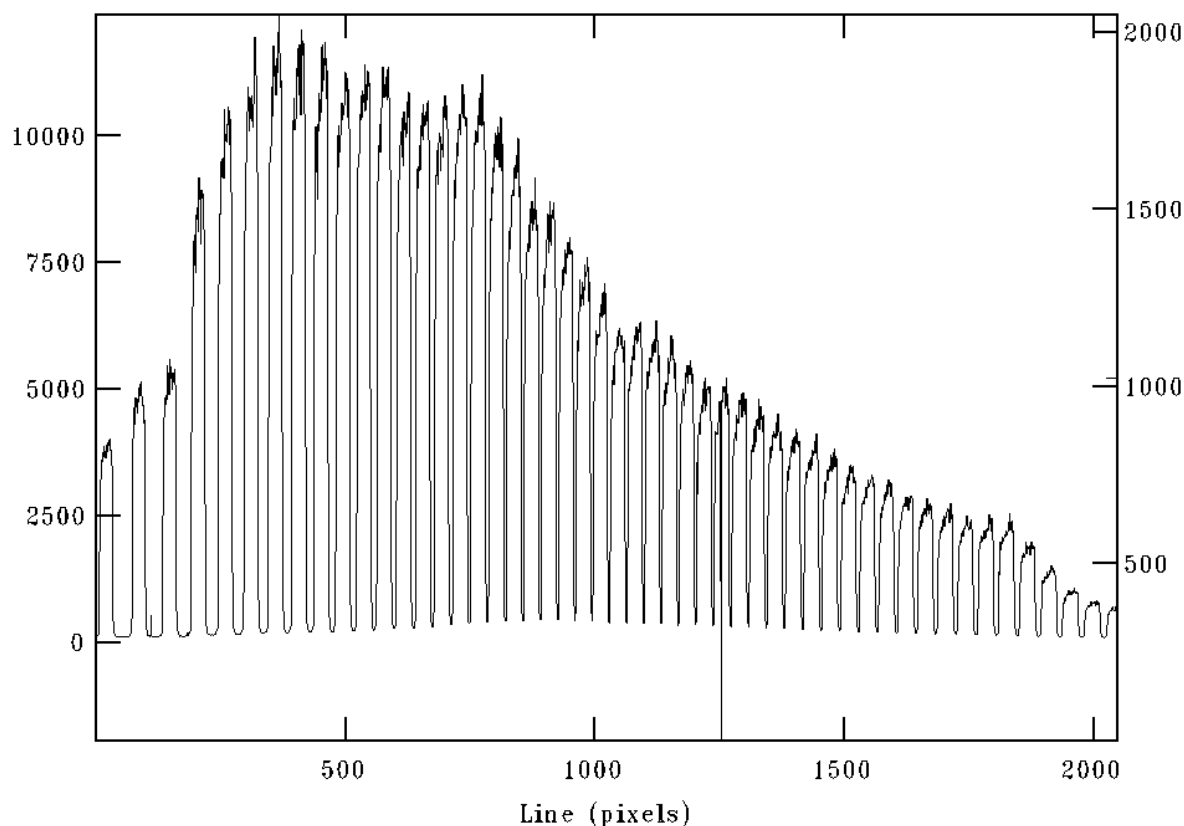


Figure 5: Cut along the middle column of a typical 100 s flat-field frame.

3.5 Learning more about flat-field frames

A typical flat-field frame is shown in Fig. 1. You can have a look at a 1D section of the frame by using the task `IMPLOT`. Just choose one of your flat image, e. g., `flat_1.fits`, and type “`implot flat_1.fits`” followed by enter. This will open the graphic window. Activate the window by clicking on it, and type “`:c 1024`” followed by enter. This will display an image cut along the middle column of the frame, like in Fig. 5. This flat was exposed for 100 s, thus the peak counts are roughly those you should expect when using the same integration time. Note how the spectrograph efficiency varies with the spectral order (remember that the lowest order corresponds to the bottommost track, hence the leftmost in figure).

To have a more careful look, just zoom-in on a smaller section: move the graphic cursor to the left bottom corner of the area you want to display and press “e”. Then move it to the right top corner and press “e”. You will be displayed something like in Fig. 6. To go back to the previous display, type “a”, and then “a” again. Type “q” to return to the command language prompt.

We note the scattered emission that causes the inter-order flux to be larger than 0 in Fig. 5 (see also Fig. 7), and the variable flux across each order in Fig. 6, which probably reflects difficulty in illuminating uniformly the slit.

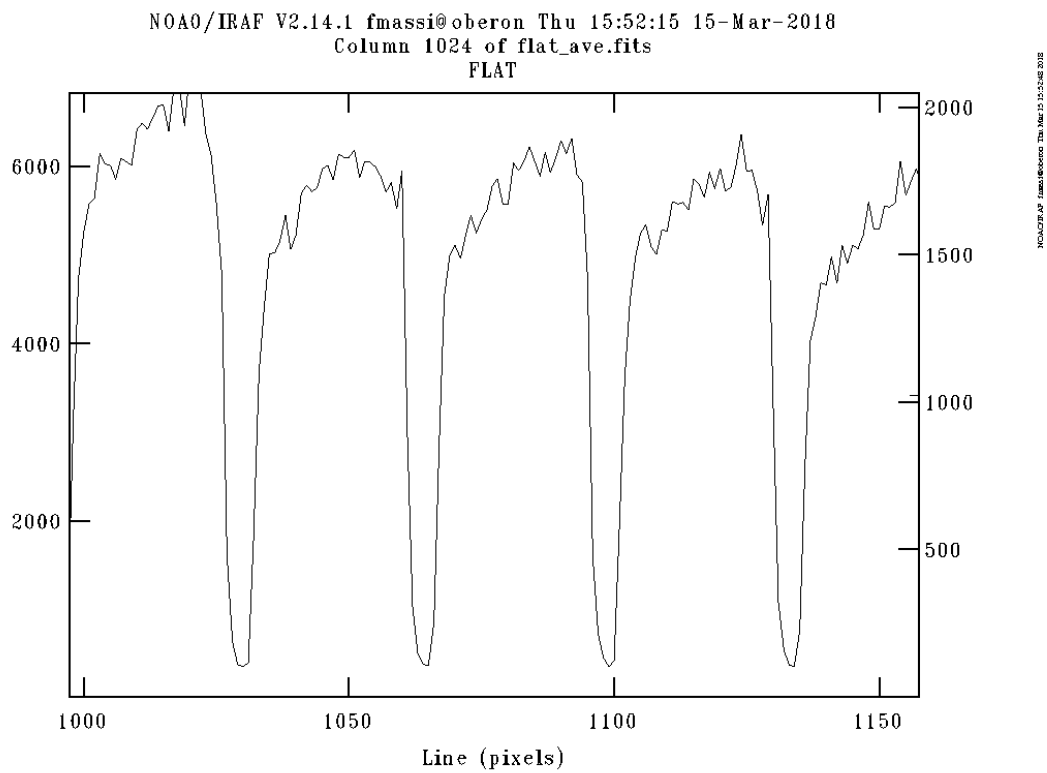


Figure 6: Zoom-in on a segment of the middle column of a typical 100 s flat-field frame.

3.6 2D flat field correction

CCDs and NIR detectors exhibit pixel-to-pixel variations in their quantum efficiency. These need being corrected for in order to increase the signal-to-noise ratio. The issue is more complicated for images of echelle spectra, due to the large fraction of the detector area (between orders) exposed to a very low flux. To avoid various degrading effects on the noise statistics, flat-field corrections should only be made inside each aperture (i. e., the detector areas impinged by the signal from the slit).

Since GIANO-B is not fibre-fed, we advise that you always use 2-D flat-field spectra. These can be constructed by deriving a flat-field correction map only inside the apertures (setting the pixels outside the apertures to 1). Then divide each frame by this map.

Due to the very high signal-to-noise ratio in all orders of the continuum lamp frames, they are also the best images to derive the traces that will be used to extract any other spectrum. This works fine since the spectrograph unit does not move during the observing run and the traces are then not expected to change. On the other hand, science target spectra are often faint in most orders and are thus not suitable to derive their own accurate traces.

MODAL NOISE AFFECTING FLAT-FIELD FRAMES As the light from the continuum lamp is transferred to the slit through optical fibres, flat-field frames are affected by modal noise. This may cause some concern. We have analysed a set of 7 subsequent 100-s flat-field exposures taken with the scrambler on, and found that modal noise in each frame cannot exceed $\sim 2\%$ of the signal, i. e. roughly the same order of pixel-to-pixel quantum efficiency variation. In fact, [6] indicate a typical modal noise r.m.s. $\sim 0.2\%$ in GIANO flat-field frames. Modal noise is proportional to wavelength; i. e., it is roughly three times larger in the *K* band than in the *Y* band. So, we can expect an r.m.s. less than 0.6% even in the *K* band. This figure will be of course further reduced after frame averaging. Clearly, modal noise in flat-field frames is less than a problem. As long as a flat-field correction yields a better signal-to-noise ratio, do it!

3.7 Averaging the flat-field frames

First of all, you have to combine together all flat-field frames (after dark subtraction and bad-pixel correction) to enhance the signal-to-noise ratio. Using the image tool and the task IMEXAMINE, check all flat-field frames and discard those displaying any problem. Then list the good files (one per line) in a text file (e. g., “to_be_flatted.list”). The frames can be combined together by using the task IMCOMBINE:

```
cl> imcombine @to_be_flatted.list flat_cmb combine=average
```

In this case, the averaged final frame will be called “flat_cmb.fits” (second argument in the line command). This is the image you will use not only to construct the 2D flat-field map, but also to find and fit the traces.

3.8 Handling spectra: IRAF packages to use

The package we will use to handle GIANO-B spectra (i. e., to construct flat-field frames, extract spectra, etc.) is called ECHELLE. So remember to load it. To do it, first you must load IMRED, then load ECHELLE (just type “imred”, enter, “echelle”, enter). However, if you already have GIANO_TOOLS loaded, ECHELLE has been automatically loaded, as well (see page 7).

REMARK IRAF has other packages that make use of some of the routines managed by ECHELLE (e. g., ONEDSPEC, TWODSPEC, etc.). Be careful, if you have loaded one of these packages after loading ECHELLE and run one of this common routines, IRAF will read its parameters from a different parameter file than the one associated with ECHELLE! Thus MAKE SURE that either GIANO_TOOLS or ECHELLE are the only spectrum analysis packages loaded when using ECHELLE routines.

3.9 Setting an ECHELLE session to extract GIANO-B spectra

After loading ECHELLE (or having GIANO_TOOLS already loaded), a few critical parameters need to be set once and for all. Hereafter, a list of the necessary steps are summarised. Be aware that many ECHELLE tasks output new fits files, write some new information in the frame headers, and save a lot of parameters in text files. These text files are saved in a subdirectory called (by default) “database” that is initialised by ECHELLE itself. This is why removing a newly-created frame is often not enough to go one step back in the reduction process. ECHELLE keeps memory of what has been done through header fields and text files, and ALWAYS uses this information any time you re-run a given task from the same directory.

Setting the dispersion axis (i. e., is dispersion along rows or columns ?) and changing the name of the subdirectory “database”: run “epar echelle” and set the field *disparis* to 1 (mandatory) and the field *database* to the name you want to assign to the database subdirectory. **In the following, we will always assume that the subdirectory name is “database” (recommended).**

Setting the number of apertures to be found. Run “epar apfind” and set the following fields as shown:

```
(nsum =    10)    Number of dispersion lines to sum or median
(nfind =    49)    Number of apertures to be found automatically
(minsep =   30.)    Minimum separation between spectra
(maxsep =  100.)    Maximum separation between spectra
```

Setting the parameters for background subtraction. Run “epar apdefault” and set the following fields as shown:

```
(lower = -16.)    Lower aperture limit relative to center
(upper = 16.)     Upper aperture limit relative to center
...
(b_funct =        chebyshev)                               Background function
(b_order =        2)                                       Background function order
...
(b_naver =       -100)                                     Background average or median
(b_niter =        3)                                       Background rejection iterations
(b_low =         1.)                                       Background lower rejection sigma
(b_high =         1.)                                       Background upper rejection sigma
```

Setting the aperture resize parameters. If *resize* is set to yes, the corresponding aperture boundaries along each column will be defined for each order by the pixels showing a number of counts equal to 10 % of the corresponding peak. However, we do not use this algorithm (it hardly works with our subtracted spectra), so run “epar apresize” and set the following fields as shown below:

```
(resize = no)     Resize apertures?
...
(nsum =    1)     Number of dispersion lines to sum or median
...
(peak =   yes)    Is ylevel a fraction of the peak?
```

Setting the parameter for the aperture editor. Run “epar apedit” and set the following fields as shown:

```
(nsum = 10)    Number of dispersion lines to sum or median
(width = 30.)  Profile centering width
(radius = 4.)  Profile centering radius
```

Setting the parameters for aperture tracing . Run “epar aptrace” and set the following fields as shown:

```
(line = 1000)   Starting dispersion line
(nsum = 10)     Number of dispersion lines to sum
(step = 10)     Tracing step
...            ...
(function= spline3) Trace fitting function
(order = 3)     Trace fitting function order
```

3.10 How to find traces

Following Sect. 3.7, you will have obtained a final, average flat-field image. Hopefully, you will have one per night. Flat-field images have by far the highest signal-to-noise ratio of all and none of the orders is depressed, as it happens to science frames due to the strong atmospheric absorption between the YJHK bands. Since, unlike the telescope, the spectrograph does not move at all during an observation run, flat-field images are the best choice to fit polynomials to the signal tracks with accuracy. These polynomials define the traces associated to each track. Usually, traces can be easily retrieved from any flat-field image. Scattered light follows a smooth distribution which peaks at the centre of the detector (see Fig 7), but is less than a problem when finding traces in GIANO-B frames.

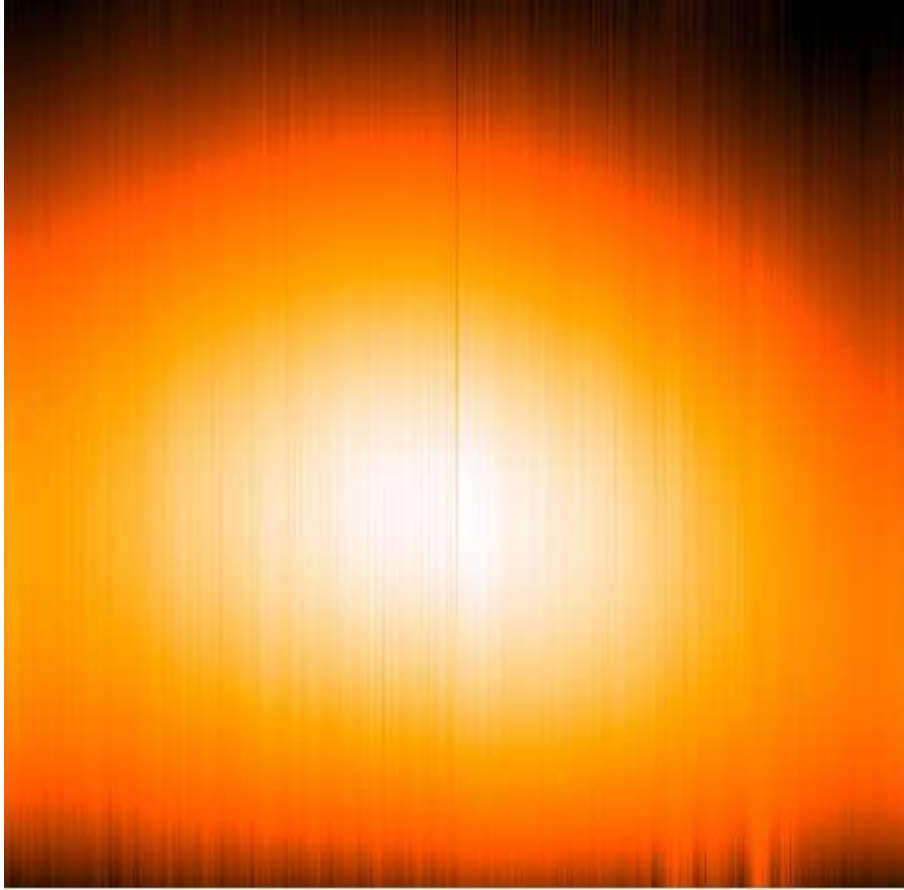


Figure 7: Map of scattered light over the GIANO-B detector area obtained by a simple spline3 fit of order 2 along columns (after discarding the signal inside the orders).

The only problem is the shape of the flux distribution inside each track along the y axis (see Fig. 6), which is not peaked. So we have to use a small trick, i. e. smoothing the flat-field frame to get each track become more peaked. This has proved to be precise enough to correctly locate all tracks, i. e. in the very first step that precedes fitting traces. We will use the iraf task MED to smooth the combined flat-field frame. Set the parameters as shown in Fig. 8. Set the field “input” to the name of the combined flat-field frame; the “output” frame could be named “flat_cmb_med.fits”.

```

      IRAF
Image Reduction and Analysis Facility

PACKAGE = imfilter
TASK = median

input =          flat_cmb  Input images to be filtered
output =        flat_cmb_med  Output images
xwindow = 5      5 X window size of median filter
ywindow = 20     20 Y window size of median filter
(zlorej= INDEF)  Lowside pixel value cutoff
(zhirej= INDEF)  High side pixel value cutoff
(boundar= nearest) Boundary (constant,nearest,reflect,wrap)
(constan= 0.)    Constant for boundary extension
(verbose= yes)   Print messages about actions taken by the task ?
(mode = ql)

```

Figure 8: Parameters to be set before running MED.

When MED is over, run APFIND on the med output frame, setting the field “input” to the smoothed combined flat-field frame (e. g., “flat_cmb_med.fits”). Check that all fields are set as in Fig. 9 and start the task. You will have the graphic window opened and an interactive graphic session ready. Check aperture locations and widths; if you want you can change them interactively (see Appendix A). However, we do not advise changing widths in this step, just check that each aperture is centred on the corresponding track and that all apertures follow the correct numbering scheme. The right aperture numbering sequence is shown in Figs. 21, 22, and 23. No aperture must be located between orders; this might happen due to bad-pixels. In case, just move to a nearby column (field “dispersion line”; a value from 1000 to 1200 should be ok) until you find one without bad-pixels, as explained in Appendix A, and recentre all apertures by typing “a” (to toggle the “all apertures” mode on) and then “c”. When satisfied, press “q” with the graphic window active.

```

      I R A F
Image Reduction and Analysis Facility

PACKAGE = echelle
TASK = apfind

input =          flat_cmb_med List of input images
(apertur= 1-49) Apertures
(referen=      ) Reference images

(interac=       yes) Run task interactively?
(find =        yes) Find apertures?
(recente=      no) Recenter apertures?
(resize =      no) Resize apertures?
(edit =        yes) Edit apertures?

(line =        1000) Dispersion line
(nsum =        10) Number of dispersion lines to sum or median
(nfind =       49) Number of apertures to be found automatically
(minsep =      30.) Minimum separation between spectra
(maxsep =      100.) Maximum separation between spectra
(order =       increasing) Order of apertures
(mode =        h)

ESC-? for HELP

```

Figure 9: Parameters to be set before running APFIND.

3.11 Using APTRACE to fit traces

REMARK The next step is critical, APTRACE will determine the traces that will then be used to extract the spectra from any other frame from the same night run.

Having found the positions, along the reference column, of the traces with APFIND, you have to “follow” them and derive the polynomial curve that best fits the corresponding signal along each track across the detector. This will be done by using APTRACE. The width of the apertures centred on the traces has been set following Sect. 3.9 (via APDEFAULT) to be 32 pixel in the direction orthogonal to the dispersion axis. The step described in Sect. 3.10 has resulted in APFIND to associate a text file listing the aperture locations (“database/apflat_cmb_med”) to its input image (“flat_cmb_med.fits”). The latter needs to be used as “reference image” in APTRACE. Once you have the aperture locations defined, the shape of the flux distribution along the y axis is no longer an issue, so the fit can directly be performed on the combined (unsmoothed) flat-field frame (i. e., the input image in APTRACE). Use “epar aptrace” to set the fields as shown in Fig. 10. Start APTRACE. Answer yes to all questions that are asked at the cl prompt. You will have the graphic window opened showing a plot similar to Fig. 21 and an interactive graphic session started. There is no need to centre again the apertures, but you have to set the width of each aperture now. Following Appendix A, you can do it either aperture by aperture (a bit time consuming), or by selecting the same lower and upper limits with respect to the centre for all apertures. E. g., we found that a lower limit of -16 and an upper limit of 14 is often an acceptable trade-off. The main criterion driving the choice of the aperture width is that inter-order regions must be left outside (this will be clearer in the next section). An optimal aperture should slightly exceed the top of the flux distribution along the y axis, including an area with a large enough signal, as shown in Fig. 11. When satisfied, press “q” with the graphic window active.

```

      IRAF
Image Reduction and Analysis Facility

PACKAGE = echelle
TASK = aptrace

input = []
flat_cmb List of input images to trace
  (apertur= 1-49) Apertures
  (referen= flat_cmb_med) List of reference images
  (interac= yes) Run task interactively?
  (find = no) Find apertures?
  (recente= yes) Recenter apertures?
  (resize = no) Resize apertures?
  (edit = yes) Edit apertures?
  (trace = yes) Trace apertures?
  (fittrac= yes) Fit the traced points interactively?

(line = 1000) Starting dispersion line
(nsum = 10) Number of dispersion lines to sum
(step = 10) Tracing step
(nlost = 5) Number of consecutive times profile is lost before quitting

(function= spline3) Trace fitting function
(order = 3) Trace fitting function order
(sample = *) Trace sample regions
(naverag= 1) Trace average or median
(niterat= 3) Trace rejection iterations
(low_rej= 3.) Trace lower rejection sigma
(high_re= 3.) Trace upper rejection sigma
(grow = 0.) Trace rejection growing radius
(mode = q)

ESC-? for HELP

```

Figure 10: Parameters to be set before running APTRACE. Check the list carefully.

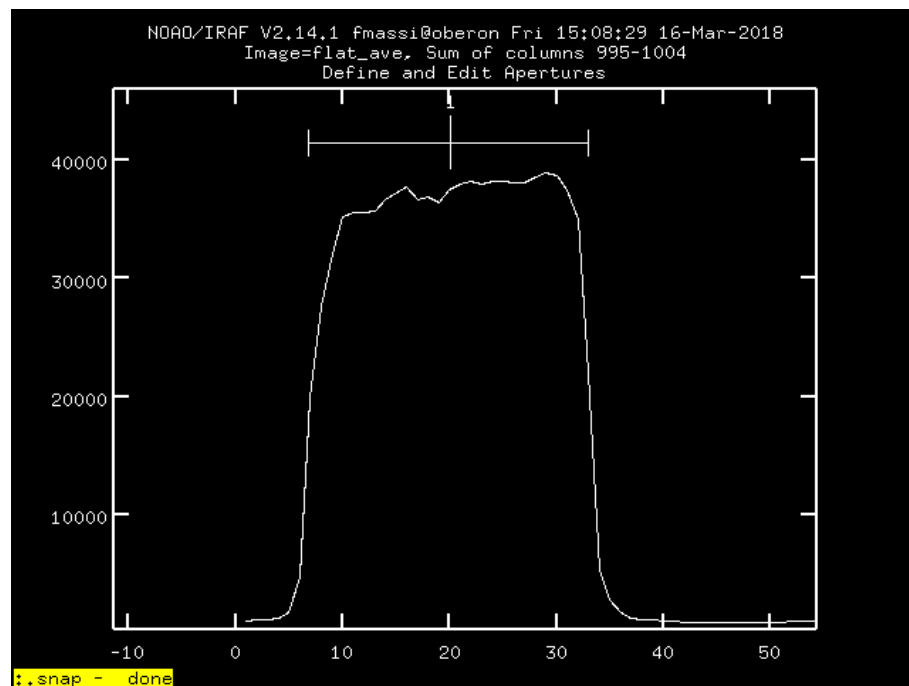


Figure 11: Example of an optimal aperture width.

You will be moved to the trace fitting session. Check the trace fits (one per track), if necessary you

can change the parameters interactively (see Appendix B). An example of correct fit for most of the 49 traces is shown in Fig. 24. Some of the highest number apertures may exhibit some more deviant points on the right, like in Fig. 25. Just delete the deviant points and redo the fit as explained in Appendix B. When done, press “q” with the graphic window active to exit the task.

Summary of APTRACE actions: Based on the aperture locations found by APFIND, it runs an algorithm which sums all image columns *nsum* by *nsum* and finds all peaks along every *step* columns following the signals along the various orders. Then, it fits a polynomial curve to each string of peaks corresponding to a given order.

It displays the fit results on the graphic window, and you can control this step interactively, as well.

3.12 Using APFLATTEN to construct a 2D flat-field frame

APFLATTEN allows you to construct a 2D flat-field image where the normalised values are only set inside the apertures, while the pixels outside them are set to 1. As explained in [7], this preserves the noise distribution all over a flat-field corrected frame, as needed by any bad-pixel removal algorithm based on a knowledge of the noise distribution. This is actually pointless in the NIR: NIR frames usually undergo several modifications and do not maintain the original information on the noise. So, we recommend that you do not use any of these algorithms. Nevertheless, given the length of the slit (~ 6 arcsec), a 2D flat is mandatory to keep the spatial information inside the apertures.

Summary of APFLATTEN actions: It extracts a spectrum from each trace found by APTRACE and fits a polynomial to it. You can review the fit results interactively. **These fits are used inside each trace to normalise the signal.**

To begin with, have a look at Fig. 12, which lists how to set the parameters (with *epar*) for APFLATTEN. Only five fields have to be adapted to your needs: *input* (set it to the name given to the final combined flat-field frame to normalise, i. e. the input file of APTRACE), *output* (set it to the name you want to give to the normalised flat-field frame), and *referen* (set it to the final combined flat-field frame to normalise, i. e. the input file of APTRACE). If you want to use a polynomial of order 1 (not recommended), also set *functio* to chebyshev and *order* to 1.

All other fields *must* be set as in figure. However, we note that, being the field *clean* set to “no”, there is no need to set the fields *saturat*, *readnoise*, and *gain*.

You are now able to start APFLATTEN. Answer yes to all questions that are asked at the cl prompt. You will have the graphic window opened and an interactive graphic session started. In the following, the session steps are summarised. Each one is further detailed in the appendix. Just follow the appendix instructions.

1. First, you are shown the aperture locations and widths as in Fig. 21. You can skip this step by pressing “q” with the graphic window active, you just did it with APFIND and APTRACE. Just press “q” with the graphic window active and say “yes” to all questions.
2. Check the spectrum fits (one per trace), you can change the parameters interactively (see Appendix C). When satisfied with the fit, press “q” and answer “yes”; you will be moved to the next aperture. Examples of good fits (and correctly extracted spectra) are shown in Fig. 26 (lower orders), Fig. 27 (central orders), and Fig. 28 (higher orders). Note that the signal peak shifts in position to the left when moving from the lower to the higher orders. When done, press “q” with the graphic window active.

```

                                IRAF
                                Image Reduction and Analysis Facility

PACKAGE = echelle
TASK = apflatten

input = []          flat_cmb List of images to flatten
output =          flat_cmb_norm List of output flatten images
(apertur=          1-49) Apertures
(referen=          flat_cmb) List of reference images

(interac=          yes) Run task interactively?
(find =           no) Find apertures?
(recente=         no) Recenter apertures?
(resize =         no) Resize apertures?
(edit =           yes) Edit apertures?
(trace =          no) Trace apertures?
(fittrac=         yes) Fit traced points interactively?
(flatten=         yes) Flatten spectra?
(fitspec=         yes) Fit normalization spectra interactively?

(line =           1000) Dispersion line
(nsum =           10) Number of dispersion lines to sum or median
(thresho=         10.) Threshold for flattening spectra

(pfit =           fit1d) Profile fitting type (fit1d/fit2d)
(clean =          no) Detect and replace bad pixels?
(saturat=         INDEF) Saturation level
(readnoi=         0.) Read out noise sigma (photons)
(gain =           1.) Photon gain (photons/data number)
(lsigma =         4.) Lower rejection threshold
(usigma =         4.) Upper rejection threshold

(funcio=          spline3) Fitting function for normalization spectra
(order =          5) Fitting function order
(sample =         *) Sample regions
(naverag=         1) Average or median
(niterat=         3) Number of rejection iterations
(low_rej=         3.) Lower rejection sigma
(high_re=         3.) High upper rejection sigma
(grow =           0.) Rejection growing radius
(mode =           ql)

ESC-? for HELP

```

Figure 12: Parameters to be set for APFLATTEN. Note that the fields “input”, “output”, and “referen” have to be adapted (see text). If necessary, change also “funcio” and “order”. All other fields must be set as shown.

A normalised flat frame obtained from APFLATTEN (and displayed with DISPLAY) will look like the one in Fig. 13. Note the perfectly flat areas between orders. This is why you need to use the same traces for normalised flat images and target 2D spectra, to make sure that all extracted signal is flat-field corrected. You can obtain a view along a column by using IMPLOT, which will look like that in Fig. 14. Note that the r.m.s. is of order of 0.05, which is ok.

3.13 Flat-fielding the targets in 2D, and final steps before spectrum extraction

Following the steps outlined in the above section yields a frame that is hereby named flat_cmb_norm.fits. This is the normalised 2D frame to use for flat-field correction. Now, list all dark-subtracted, bad-pixel cleaned comparison lamp spectra, and all bad-pixel cleaned science frames in a single text file (e.g., to-be-flatted.list). Make a new text file (e. g., call it flatted.list) and list, in the same order, the names you want to allot to the flat-field corrected frames. To flat-field correct them all, just type:

```
cl> imarith @to-be-flatted.list / flat_cmb_norm @flatted.list
```

This will divide (pixel-by-pixel) each frame listed in to-be-flatted.list by the normalised flat, assigning the corresponding name found in flatted.list to the output frame.

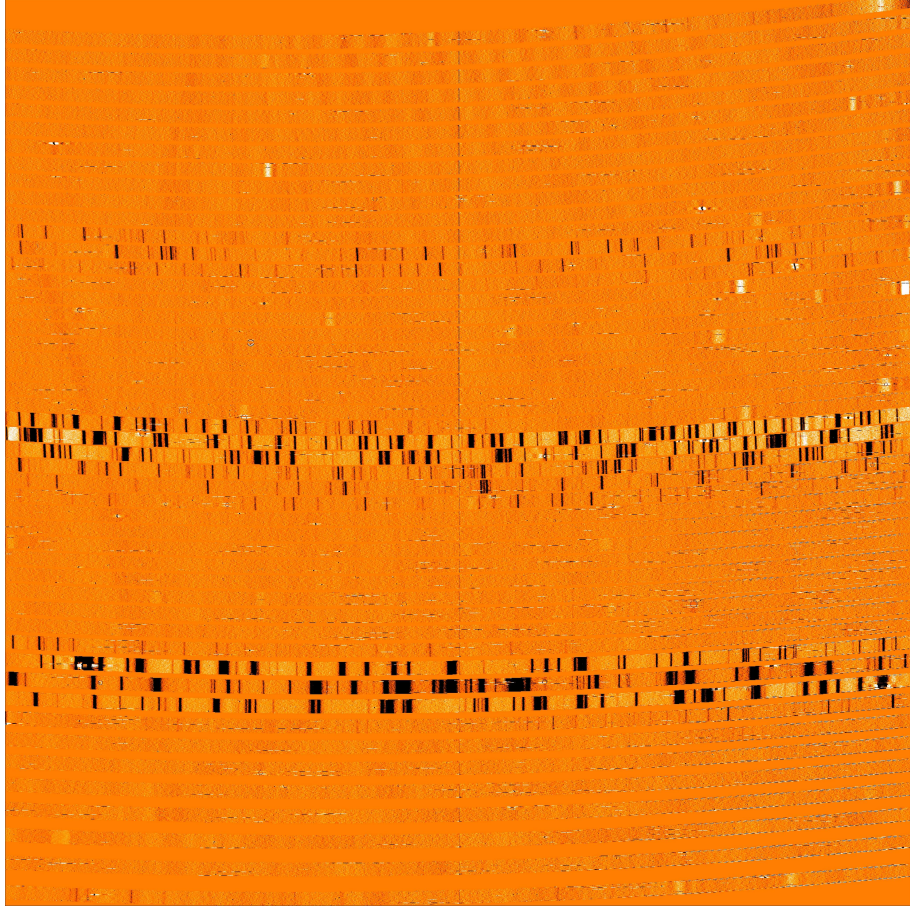


Figure 13: Normalised flat-field image obtained with APFLATTEN.

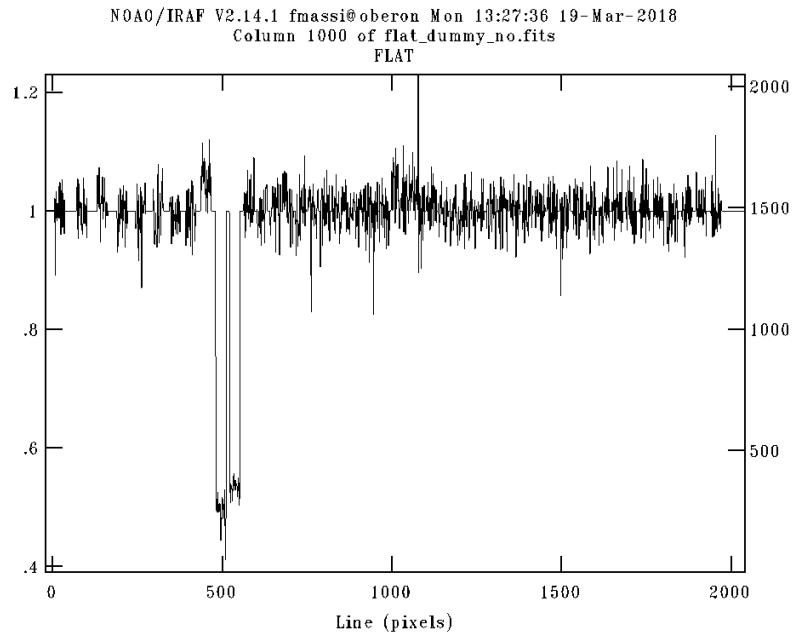


Figure 14: Cut along the central column of a normalised flat image obtained with APFLATTEN.

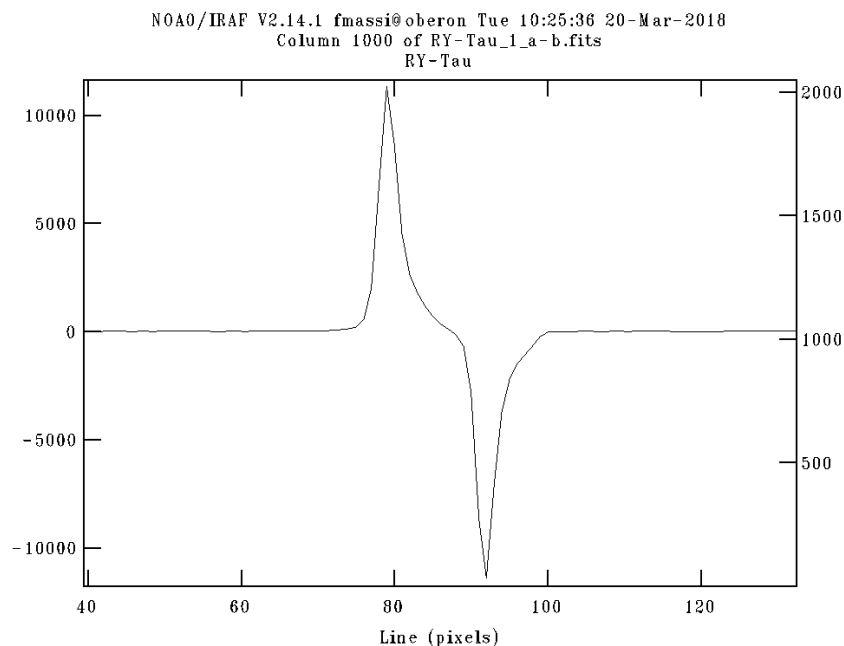


Figure 15: Zoom-in on a cut along the y axis of a subtracted target frame, showing the positive and negative peaks.

FRAME SUBTRACTION After you have corrected the science frames, subtract frame B from frame A for every AB cycles with the target alternatively in each position on the slit. This will remove any residual scattered and diffuse light contribution. Do this again using IMARITH, e. g.:

```
cl> imarith star1-A - star1-B star1_A-B
```

Of course, this can be done for all frames simultaneously by using three list files (and an @ in front of the file names in the command line, as usual). The frame `star1_A-B.fits` will have positive and negative peaks. The signal from frame A will be positive and the signal from frame B will be negative. This can be seen in Fig. 15, showing a zoom-in on a cut along the y axis. Note that the two peaks span the whole trace width obtained from the flat-field frame. The detector areas next to the peaks were not affected by radiation from the slit and quickly reduce in size with increasing orders, so they cannot be used to sample a background.

Thus, you need to multiply `star1_A-B.fits` by -1 before extracting the spectra from the negative tracks (see Sect. 3.14). You might also have either pairs of on-source and off-source frames or single target frames, i. e. with the source signal in only one position. In the former case, subtract the off-source frames from the on-source ones, as explained above. In the latter case, you have no other choice but a dark-subtracted, flat-field corrected, background contaminated spectrum.

COSMIC RAY CLEANING Despite bad-pixel cleaning, subtracted frames will still show residual spikes mostly due to cosmic rays. You can further clean your subtracted frames by using the GIANO_TOOLS task GIANO_COSMICRAYS. This is a script based on the iraf task COSMICRAYS; first, it uses COSMICRAYS on the subtracted frame. The cleaned frame is then multiplied by -1 and COSMICRAYS is again applied to this. Finally, the double-cleaned frame is again multiplied by -1 and output.

To use GIANO_COSMICRAYS, you just have to set the field *imlist* to the name of a text file listing the frames to be corrected; the field *outstr* to a string of characters that will be added in front of the name of each input frame to mark the cleaned frame; and the field *fix* to “yes” if you want an interactive graphic section. **We advise you to use this task on already subtracted (or dark-subtracted) frames only, as frame subtraction itself corrects for lots of bad-pixels.**

COMBINING SUBTRACTED FRAMES If you have more than one subtracted AB pair of frames for the same target, combine them together with IMCOMBINE before extracting the spectra. Alternatively, you could first extract the spectra from every single A–B frame and then combine the spectra from the same targets with SARITH as explained in Sect. 3.15. As already said, list them all in a text file (e. g., target1_A-B.list) and run IMCOMBINE:

```
cl> imcombine @target1_A-B.list target1_A-B_comb combine=average
```

this will yield the file target1_A-B_comb.fits as the average of all frames listed in target1_A-B.list.list.

HOW TO DEAL WITH SPECTRA FROM SINGLE SLIT LOCATIONS You might have to deal with 2D spectra from which you cannot subtract a sky. E. g., because you realise that the source is extended and after AB subtraction the spatial profiles would overlap. Or due to technical problems. In this case, you must subtract the corresponding dark frame before flat-field correction. Then use GIANO_COSMICRAYS to clean the dark-subtracted frame. If you have more than one frame, average them together. You can still extract a spectrum from this final frame using DOECSLIT with background subtraction enabled (see next section). We have tested this method by extracting a few spectra with and without AB subtraction. Comparing them we did not find any significant difference.

3.14 Spectrum extraction

The task that you will use for extracting spectra is DOECSLIT. Most of the work has already been done by APFIND and APEDIT. Nevertheless, you will have to set new apertures centred on the target spectrum inside the slit, although you must not fit the traces again. Then, DOECSLIT will wavelength-calibrate the spectra after extracting them and finally will save them in fits format. Flux calibration will not be performed with DOECSLIT.

Obviously DOECSLIT operates on the positive tracks, so you must construct new frames from the subtracted ones multiplying each by -1 and repeat the steps below on these new frames to also extract the spectra from the negative tracks. You can use IMARITH to do this. To start with, list all subtracted frames in a text file (e. g., target1_A-B.list) and make a file listing the names of the output frames (e. g. target1_B-A.list) in the same order. Then, run IMARITH:

```
cl> imarith @target1_A-B.list * -1 @target1_B-A.list
```

this will yield new frames whose names are listed in target1_B-A.list where the negative signal is now positive (and vice versa).

First, run doeslit on the original A–B frames (those listed in target1_A-B.list) to extract the spectra taken in the A position:

1. Run “epar doeslit” and set the fields as in Fig. 16. **the fields to adapt to your own setting are: objects, apref, arcs, readnoi, gain, datamax, and backgrou.** As for the first three fields, input the name you have assigned to the file listing the science frames, the reference frame for the traces (i. e., flat_cmb, the input file in APTRACE), and the name you have assigned to the

file listing the calibration lamp frames. Compute the values to input into fields *readnoi* (effective frame readout noise) and *gain* (effective frame gain) as explained in Appendix E. Compute the maximum data value to input into field *datamax* from Table 1. It is difficult to set inter-order apertures to sample a background: the distance between orders decreases with increasing orders, so they should be set individually. In addition, as the positive and negative signal peaks span the whole trace in each order, background windows could only sample the inter-order scattered radiation and not the residual sky emission. Since frame subtraction already cancel most of the sky signal and scattered light, we advise you to set *backgrou* to “none”. However, if you operate on unsubtracted frames you must use background subtraction. In this case, set *backgrou* to “fit”.

2. Note that the field *sparams* in Fig. 16 is empty. This is because SPARAMS does not require a value, it is just a link to another parameter file. You have to access it by entering “:e” (followed by return) into the field. Alternatively, you can exit the parameter editor and type “epar sparams” to access the file. Edit SPARAMS and set ONLY the fields shown in Fig. 17 as shown in figure, keeping the default value for the others. In fact, you can leave the parameters in section BACKGROUND AND SCATTERED LIGHT PARAMETER as they are if you operate on subtracted frames. In addition, there is no need to change any of the fields not shown in the figure. Only parameters in the “AUTOMATIC ARC ASSIGNMENT PARAMETERS” section can be adapted in alternative ways; see Appendix F.
3. Run DOECSLIT. A graphic session will be opened showing a cut along the y axis and the numbered apertures (a plot like the one displayed in Fig. 21). You *must* resize and recentre each aperture as in Fig. 18. To do that, follow the instructions given in Appendix A. In summary, zoom in on aperture 1, centre it by typing “c”, resize it by typing “l” and “u”, then move to the background fitting session (only if *backgrou* set to yes) and set the background sampling windows (see below). Once done, go back to the aperture setting session by typing “q” and move to aperture 2 by typing “+”. Repeat the steps for all apertures. Check how the aperture windows enclose the signal in other columns, as well. If necessary, choose trade-off widths. In addition, by comparing Fig. 18 and Fig. 21 you can note that the new aperture is smaller than those used to construct the flat-field frame and entirely included in the corresponding larger one obtained from the flat-field frame.
4. If *backgrou* is set to yes, by typing “b” you will be shown the background sampling windows of the aperture you are operating on. Type “t” to cancel the current windows and follow Appendix A to set new ones. In summary, by typing “s” at each end of the desired interval, set the windows like the ones shown in Fig. 19. **Note that you can set one window only if necessary (e. g., the one on the right in Fig. 19). Also note that having set the parameter *b_naver* to -100, the background is estimated as the median in each of the windows.** Then type “f” to compute the background. When satisfied, type “q” to move back to the aperture setting session. You need to repeat this step for all apertures (however, see Appendix A on how to do this one and for all).
5. After the last aperture has been set, you will be moved to the wavelength calibration session. Follow Appendix G to do that.
6. After wavelength calibration has been done, the task SPLOT will be invoked by DOECSLIT and you will be shown the final spectra. See Appendix D on how to use SPLOT, the 1D spectrum editor.
7. When DOECSLIT has finished, the extracted spectra will be saved in fits files having the same name as the input file plus the string “.ec.” at the end (e. g., input file “target1.fits”, output file “target1.ec.fits”).

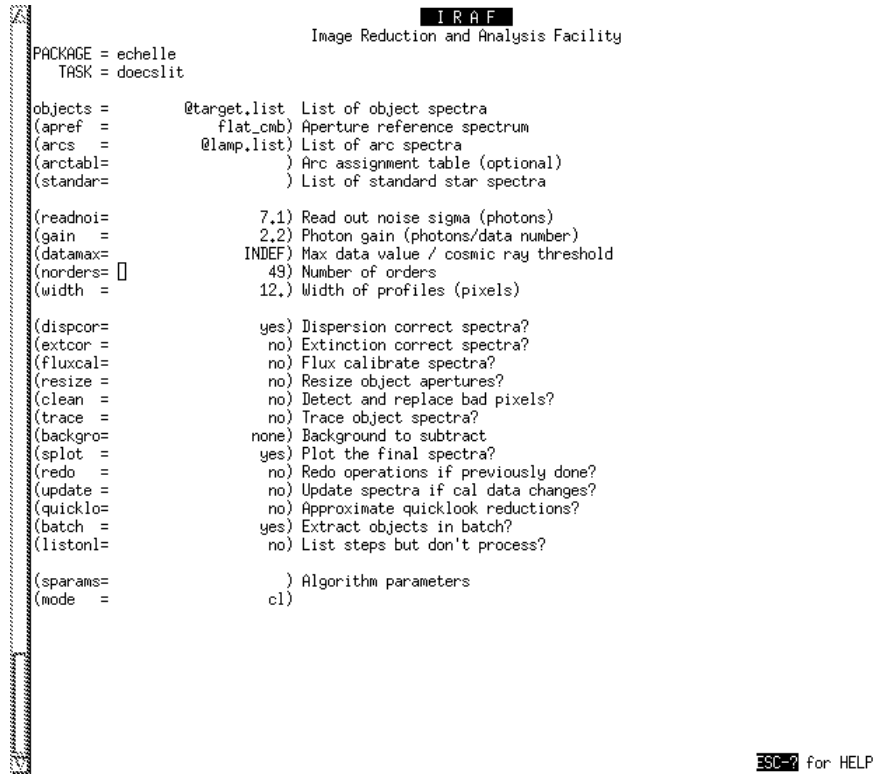


Figure 16: Parameters to be set for DOECSLIT. Note that the fields **objects**, **apref**, **arcs**, **readnoi**, **gain**, **datamax**, and **backgro** have to be adapted (see text).

Now repeat the same steps on the frames listed in the file `target1_B-A.list` to extract the spectra taken in the B position.

3.15 Combining the extracted 1D spectra together

At this point, you will have two different 1D spectra of every target (each per position on the slit). You need to combine them together to increase the signal-to-noise ratio. There is one more reason to combine them. The signal detected by pixel (i, j) is given by $T \times f(i, j) + b$, where T is the atmospheric transmission, $f(i, j)$ is the astronomical signal, and b is any other contribution that adds up to the astronomical signal (scattering, sky emission, etc.). By summing along the y axis, across a given aperture, one gets $T \times \sum_{j=1}^n f(i, j) + n \times b$. We can assume b almost constant across the aperture. Nevertheless, b can vary between the A integration and the B integration. Thus, extracting the X spectrum (where X is A or B and Y is B or A) from a subtracted frame will yield $T_X \times \sum_{j=1}^n f_X(i, j) + n \times (b_X - b_Y)$. We found that the term $n \times (b_X - b_Y)$ becomes not negligible in the upper right part of the detector, where the astronomical signal gets fainter and fainter due to the lower efficiency of the spectrograph. However, when summing the two extracted spectra, one gets $T_X \times \sum_{j=1}^n f_X(i, j) + T_Y \times \sum_{j=1}^n f_Y(i, j)$ and background differences cancel out (to 0 order). Hence, summing or averaging A and B spectra further minimises background variability, provided the trace width is almost the same for both.

Summing spectra can easily be done using the task **SCOMBINE**, which is the analogue of **IMCOMBINE** for spectra; **SCOMBINE** handles spectra and deals with issues such as differences in sampling intervals and wavelength ranges. First, have a look at the 1D spectra to sum up with **SPLIT** (see Appendix D) to identify the maximum and minimum (in principle, the minimum should be zero) signal level throughout the sample. Then, list all spectra to be combined, in a text file (e. g., call it `target1_ec.list`). **Now you**

```

-- BACKGROUND AND SCATTERED LIGHT PARAMETERS --
(b_func=      legendre) Background function
(b_order=      2) Background function order
(b_naver=     -100) Background average or median
(b_niter=      3) Background rejection iterations
(b_low =       1.) Background lower rejection sigma
(b_high =      1.) Background upper rejection sigma
(buffer =      2.) Buffer distance from apertures
(apscat1=      ) Fitting parameters across the dispersion
(apscat2=      ) Fitting parameters along the dispersion

-- APERTURE EXTRACTION PARAMETERS --
(weights=     none) Extraction weights (nonelvariance)
(pfit =       fit1d) Profile fitting algorithm (fit1d|fit2d)
(lsigma =      3.) Lower rejection threshold
(usigma =      3.) Upper rejection threshold

-- ARC DISPERSION FUNCTION PARAMETERS --
(thresh=      10.) Minimum line contrast threshold
(coordl=      gianot$U_Ne_Ar_lines_list.cl.dat) Line list
(match =       0.05) Line list matching limit in Angstroms
(fwidth =     13.) Arc line widths in pixels
(cradius=      5.) Centering radius in pixels
(i_func=      legendre) Echelle coordinate function
(i_xorde=      4) Order of coordinate function along dispersion
(i_yorde=      4) Order of coordinate function across dispersion
(i_niter=      3) Rejection iterations
(i_low =       3.) Lower rejection sigma
(i_high =      3.) Upper rejection sigma
(refit =       yes) Refit coordinate function when reidentifying

-- AUTOMATIC ARC ASSIGNMENT PARAMETERS --
(select =      nearest) Selection method for reference spectra
(sort =        jd) Sort key
(group =       1jd) Group key
(time =        no) Is sort key a time?
(timeura=     17.) Time wrap point for time sorting

-- DISPERSION CORRECTION PARAMETERS --
(lineari=      yes) Linearize (interpolate) spectra?
(log =        no) Logarithmic wavelength scale?
(flux =       yes) Conserve flux?

```

Figure 17: Parameters to set in SPARAMS. **Only for “AUTOMATIC ARC ASSIGNMENT PARAMETERS”** you might want to use a different set-up (see text).

will have to use the fits files of the extracted 1D spectra (the “ec.fits” files) obtained from DOECSLIT. Run “epar scombine” and set the parameters as in Fig. 20. **The only fields you may want to set are *lthresh* and *hthresh*.** You need to put a value slightly below the lowest signal level you have identified into *lthresh*, and a value slightly above the highest signal level you have identified into *hthresh*. This will clean off a few annoying spikes. The parameter fields listed by the parameter editor lying below *lthresh* are not used in our case, so you can leave them untouched. As for *combine*, we prefer using the average, **but see Appendix E to compute effective exposure time, gain, and read-out noise.** And, of course, *output* should be set to the name you want to give to the output combined 1D spectrum. Finally, run SCOMBINE.

3.16 Normalising the extracted 1D spectra to the continuum

You might want to have your spectra normalised to the continuum, i. e. to have the continuum level set to a constant value. You can do it by using the task CONTINUUM. Unfortunately, it is often difficult to identify the continuum level in spectra exhibiting lots of absorption lines. In principle, you could alternatively do it flattening your frames with a non-normalised flat-field frame. Taking into account that the flat-field spectra are those with the best signal-to-noise ratio, since all orders span a limited range in wavelength, the spectral shape in the flat-field apertures is mostly due to the spectrograph efficiency. The simplest way to operate is by fitting a Legendre polynomial of order 1 (i. e., a horizontal straight line) when constructing a normalised 2D flat (Sect. 3.12). After flat-field correction, your extracted spectra will already be approximately normalised to the continuum. **However, we discourage using this method. The scattered light on the detector is almost the same as the signal from the flat lamp in the upper right quadrant of the detector, where the spectrograph efficiency is the lowest. On the other hand, it is very difficult to correct flat-field frames for this as the tracks from subsequent orders are too near-by to sample inter-order regions. This causes**

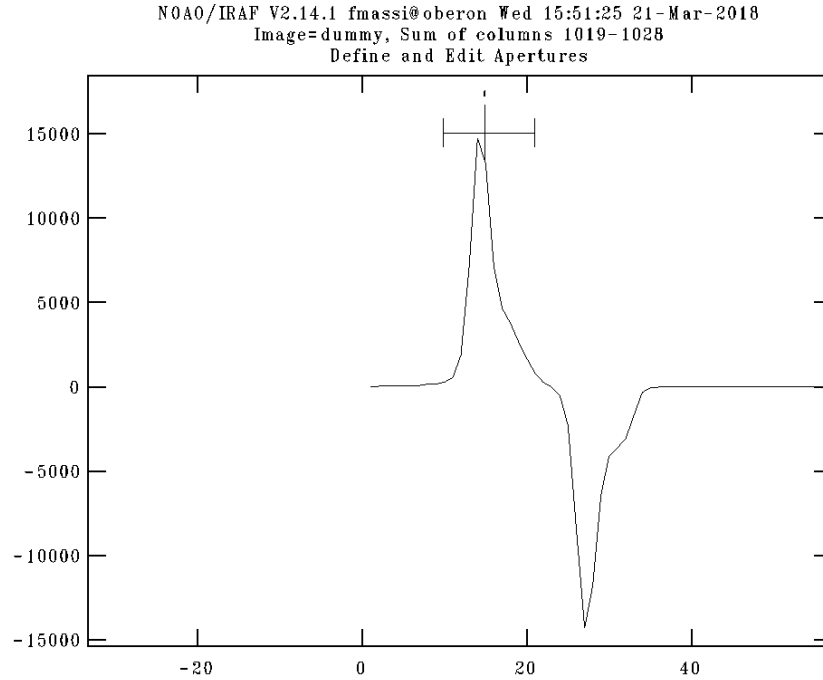


Figure 18: Zoom-in on aperture 1 from a cut along the y axis of a science subtracted frame. The plot shows an example of a correctly resized and recentred aperture (the horizontal line above the signal peak). By comparing with Fig. 21, you can note that the aperture is smaller than those used to construct the flat-field frame and entirely included in the corresponding larger one obtained from the flat-field frame.

the signal in the flat-field corrected spectra to drop rapidly in the lower wavelength end of the higher orders.

3.17 Viewing the final 1D spectra

To examine your final extracted 1D spectra you can use SPLOT, as explained in Appendix D. You can also use SPLOT to perform several operations on spectra (see the SPLOT help in IRAF). You might also want to handle 1D spectra after translating them into text format (a two-column list with wavelengths and counts). To do this, you can use the GIANO_TOOLS script called FITS2TXT, which is based on the iraf.onedspec task WSPECTEXT. We will explain how to use FITS2TXT first, and then how to use WSPECTEXT itself. Run “epar fits2txt”; the editor page will look like below:

```
PACKAGE = giano_tools
TASK = fits2txt
specin = @spec.list   Input fits spectrum
startnum = yes        Start output from 1(yes) or 32(no)?
sphed = no           Include header?
(mode = q)
```

The field *specin* has to be set either to the name of a single fits file or to a text file (remember to put “@” followed by the file name into the field) listing a number of 1D spectrum fits files. The task output consists of 49 new text files each with one spectral order, having the same name of the input fits file but ending in “.tN.txt” instead of “.fits”. Note that *N* is an integer number ranging from 1 (the lowest order, i. e. the largest wavelength) to 49 (the highest order, i. e. the smallest wavelength). If

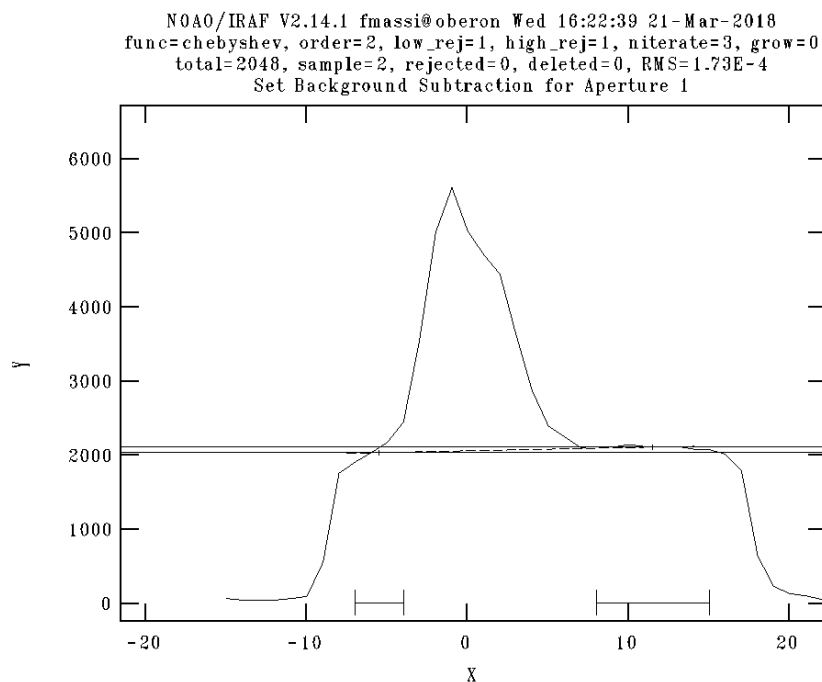


Figure 19: Zoom-in on aperture 1 from a cut along the y axis of a science **unsubtracted** frame, after moving to the background fitting session. The plot shows an example of background sampling windows correctly set (the two horizontal lines in the bottom). The horizontal solid lines show the median computed in each window, and the dashed line shows the background to subtract from the signal. By comparing with Fig. 18, you can note that being the frame unsubtracted, the science signal falls on a square pedestal given by the sky emission that fills the whole aperture. The background *must* be sampled inside the top of the pedestal. Also note that the pedestal is hardly visible at the higher orders.

you want, you can number the text files starting from 32 (i. e., the real number of the lowest order) just setting *startnum* to *no*. If you want to have the fits header saved in the text files, as well, set *sphed* to *no*. For the wavelength ranges covered in each order, see Appendix H.

Alternatively, you can use WSPECTEXT by yourself. You first have to load the package:

```
cl> onedspec
```

Note that *wspectext* allows you to convert only one order of a spectrum at a time. E. g., to convert the first trace of a spectrum named *target1.ec.fits* into a text file named *target1_n1.txt*, you will have to set the *wspectext* parameters as follows (note the *[*,1]* at the end of the input file name):

```
PACKAGE = onedspec
TASK = wspectext
input = target1.ec.fits[*,1] Input list of image spectra
output = target1_n1.txt Output list of text spectra
header = no Include header?
wformat = Wavelength format
(mode = q)
```

The good news is, you can make a script to convert all traces in one shot. Just make a text file with the following lines (one per aperture):

```
wspectext target1.ec.fits[*,1] target1_n1.txt header=no
wspectext target1.ec.fits[*,2] target1_n2.txt header=no
wspectext target1.ec.fits[*,3] target1_n3.txt header=no
...
```

Then run the script typing:

```
cl> cl < nomefile
```

Needless to say, there is also a task RSPECTEXT, which allows you to convert text spectra in fits files (see the help in IRAF). **After using wspectext, unload the package onedspec by typing “bye”** to avoid messing up the parameter files associated with the package ECHELLE.

```

only one output image will be created. In the first case the
image will contain only one spectrum and in the latter case
there will be a spectrum for each selected aperture. If the
grouping option is "images" then there will be one output
spectrum per input spectrum.

noutput = ""
List of output images to be created containing the number of
spectra combined. The number of images required is the same as
the noutput list. Any or all image names may be given as a null
string, i.e. "", in which case no output image is created.

logfile = "STDOUT"
File name for recording log information about the combining
operation. The file name "STDOUT" is used to write the
information to the terminal. If the null string is specified
then no log information is printed or recorded.

apertures = ""
                                IRAF
                                Image Reduction and Analysis Facility

PACKAGE = echelle
TASK = scombine

input = target_1A.ec,target_1B.ec List of input spectra
output = target_1_sum List of output spectra
(noutput= ) List of output number combined spectra
(logfile= ) Log file

(apertur= 1-49) Apertures to combine
(group = apertures) Grouping option
(combine= sum) Type of combine operation
(reject = none) Type of rejection

(first = yes) Use first spectrum for dispersion?
(w1 = INDEF) Starting wavelength of output spectra
(w2 = INDEF) Ending wavelength of output spectra
(dw = INDEF) Wavelength increment of output spectra
(nw = INDEF) Length of output spectra
(log = no) Logarithmic increments?

(scale = none) Image scaling
(zero = none) Image zero point offset
(weight = none) Image weights
(sample = ) Wavelength sample regions for statistics

(lthresh= INDEF) Lower threshold
(hthresh= INDEF) Upper threshold
(nlow = 1) minmax: Number of low pixels to reject
(nhigh = 1) minmax: Number of high pixels to reject
(nkeep = 1) Minimum to keep (pos) or maximum to reject (neg)
(mclip = yes) Use median in sigma clipping algorithms?

```

Figure 20: Parameters to be set in SCOMBINE. Fields lthresh and hthresh need being adapted as explained in the text.

4 Bibliography

- [1] Oliva, E., et al. 2012, SPIE, 8446
- [2] Origlia, L., Oliva, E., & Scuderi, S. GIANO cookbook for observers,
<http://www.tng.iac.es/instruments/giano>
- [3] Origlia, L., & Oliva, E. GIANO cookbook for proposers,
<http://www.tng.iac.es/instruments/giano>
- [4] Barnes, J. 1993, “A Beginner’s Guide to Using IRAF”,
<http://iraf.noao.edu/docs/recommend.html>
- [5] Shames, P., & Tody, D. 1986, “A User’s Introduction to the IRAF Command Language”,
<http://iraf.noao.edu/docs/recommend.html>
- [6] Iuzzolino, M., et al. 2014, SPIE, 9147, 914766 [7] Wilmarth, D., & Barnes, J. 1994, “A Users Guide to Reducing Echelle Spectra with IRAF”,
<http://iraf.net/irafdocs/ech.pdf>
- [8] Oliva, E., et al. 2018, SPIE, 2018, 10702, 74O
- [9] Oliva, E., et al. 2013, A&A 555, A78
- [10] ESO Sky Model Calculator: <http://www.eso.org/observing/etc/>

Appendix A: Checking the apertures

After starting APFIND, APTRACE, APFLATTEN, or DOECSLIT, you will be asked a few questions; always answer “yes”. Then, the graphic window will open up and the first thing you will be shown is a section of the frame along the central column, like in Fig. 21. You can see all orders as peaks and, on top of the panel, the aperture intervals marked by horizontal segments joining small vertical bars and labelled by numbers. You may want to check that the aperture intervals are correct, so you had better zoom in on parts of the image. In Fig. 22, the zoom-in shows the correct centring. Aperture interval segments must not fall in areas between subsequent orders, which might happen due to one or more bad-pixels. Conversely, all subsequent order peaks must be numbered (from 1 to 49), excepted the topmost two orders (see Fig. 23).

Hereafter, a list of the relevant graphic commands, **always to input with the graphic window active. Furthermore, note that the graphic commands that modify an aperture must be entered on the aperture indicated in the yellow bar below the plot (see below how to change aperture)!**

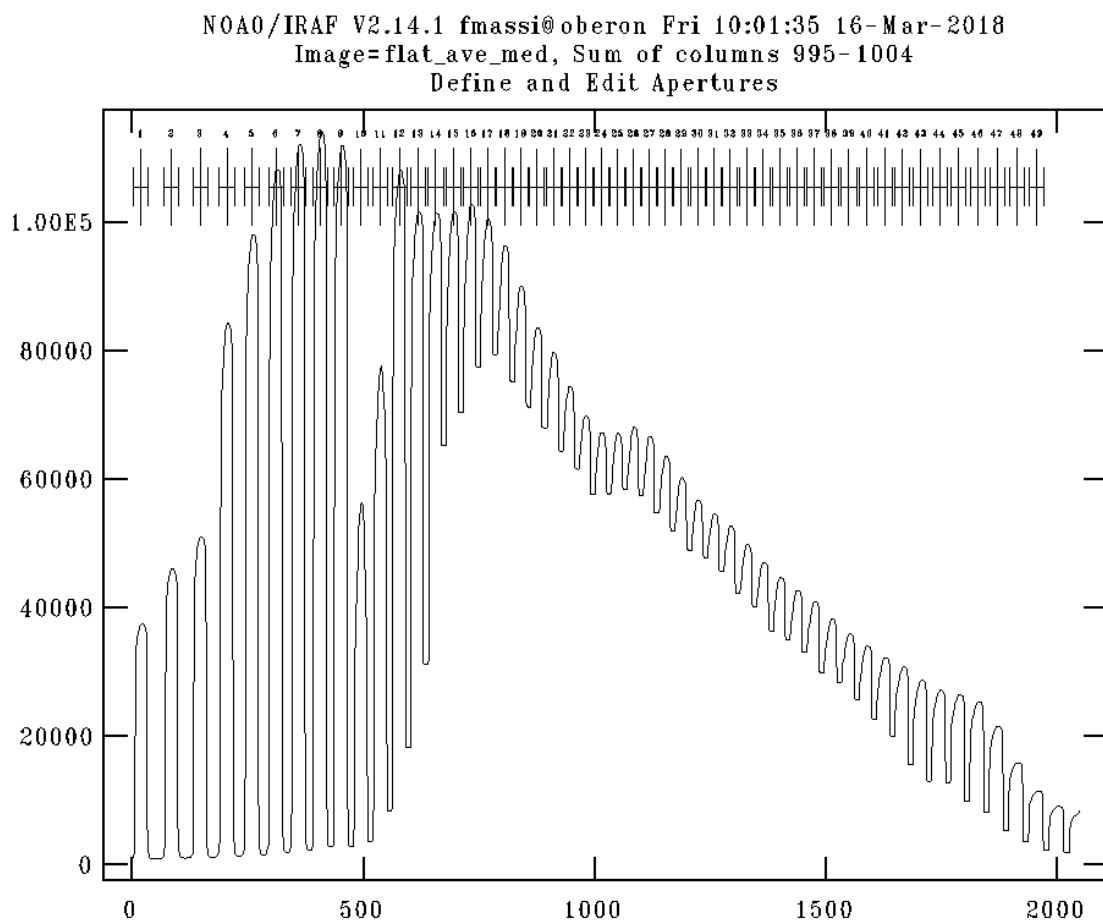


Figure 21: Graphic window showing the apertures selected by the algorithm. **We note that the image displayed is the smoothed combined flat-field frame.** The signal is plotted along the reference column; each peak represents a spectral order. The numbered segments on top of the panel mark the selected apertures and must always enclose the peaks.

1. to start the help page, type “?”. The cl window will display a list of commands that you can

scroll down. After you have finished, type first “q” and then “return” to return the cursor to the graphic window.

2. to enlarge the view, first type “w”; then move the cursor marker to the lower left corner of the area you want to enlarge and type “e”. Finally, move the cursor marker to the upper right corner of the area you want to enlarge and type again “e”. Figure 22 shows an example of the result of this sequence of commands.
3. to return to the largest view, type “w” and then “a”.
4. to enlarge the width of the current aperture (at the beginning, aperture number 1, i. e. order 32), move the cursor marker to the lower end of the interval and type “l”. Then move the cursor marker to the upper end of the interval and type “u”.
5. to move to the next aperture, type “+”. To move to the previous aperture, type “-”.
6. to enlarge the width of all apertures simultaneously, type “a” before entering the graphic commands for single apertures. This toggles the “all apertures” mode on. Type again “a” to go back to the “single aperture” mode.
7. to change the column shown in the plot, type “:line n”i and jenterj, where “n” is the column pixel coordinate (from 1 to 2048).

We advise that you do not display another frame column (which is possible) during the trace fitting session (i. e., when using APFIND and APTRACE), it may cause errors in the trace centring. On the other hand, you should do that when using APFLATTEN and DOECSLIT to be sure that the traces obtained from APTRACE actually fit the signal peaks.

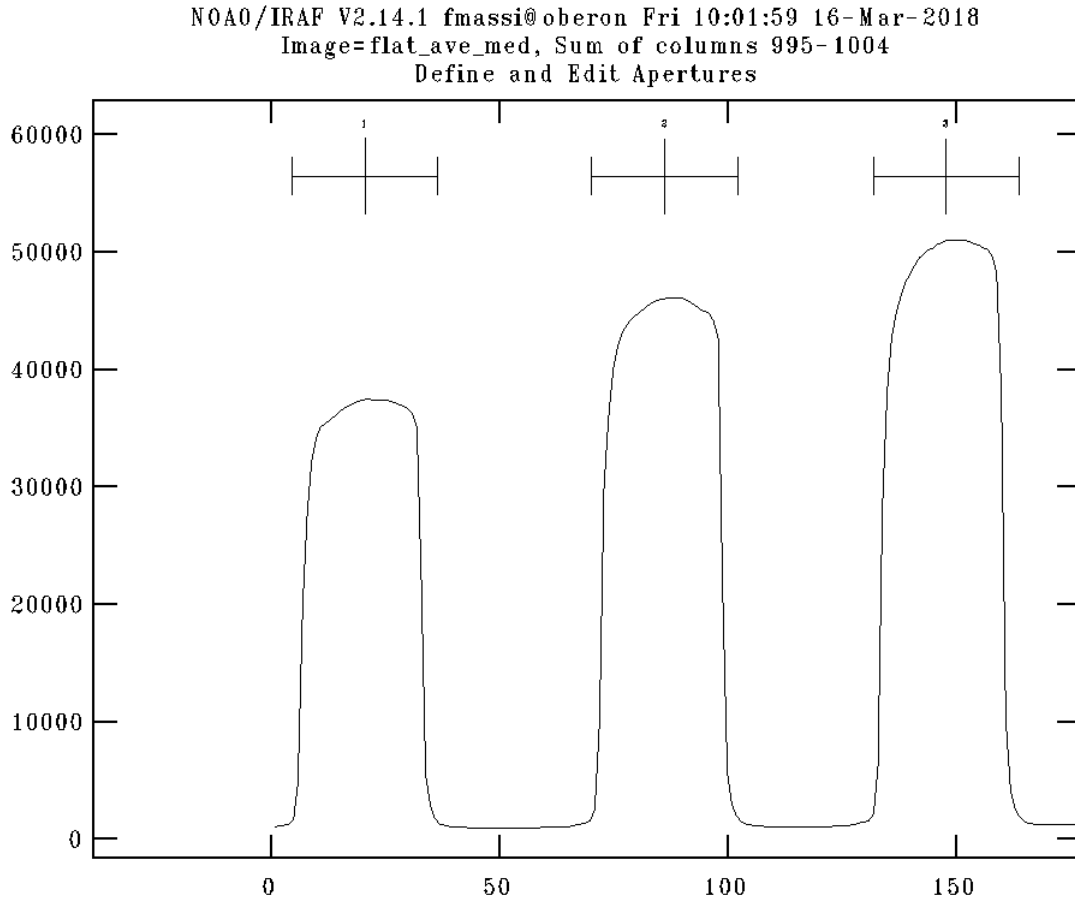


Figure 22: Graphic window showing the apertures found, after zooming in using the graphic commands as explained in the text. **We note that the image displayed is from the smoothed combined flat-field frame.**

If you are using DOECSLIT on unsubtracted frames, you will also have to review the two intervals outside each aperture that define the windows that will be used to estimate the background to subtract. This can be done for the current aperture by typing “b” (it will not work if *backgrou* is set to “none”). This will move the current session to the background fitting session and the graphic window will display the current background sampling windows (for the active aperture only). You can enlarge the view as explained above. In Fig. 19, the graphic window displays the background intervals (horizontal segments at the bottom of the panel) for aperture 1. Be aware that the intervals are defined as pixel distances from the aperture centre along a column (e. g., $-a : -b, c : d$). To change background intervals, type “t” to cancel the current windows and then type “:sample $-a : -b, c : d$ ”. Alternatively, you can move the graphic cursor to the ends of each desired interval and type “s”. If the “a” mode is active, this will set the background sampling windows for all the apertures. Then, go back to the aperture editing session by typing “q”. Note that you can only change aperture from the aperture editing session.

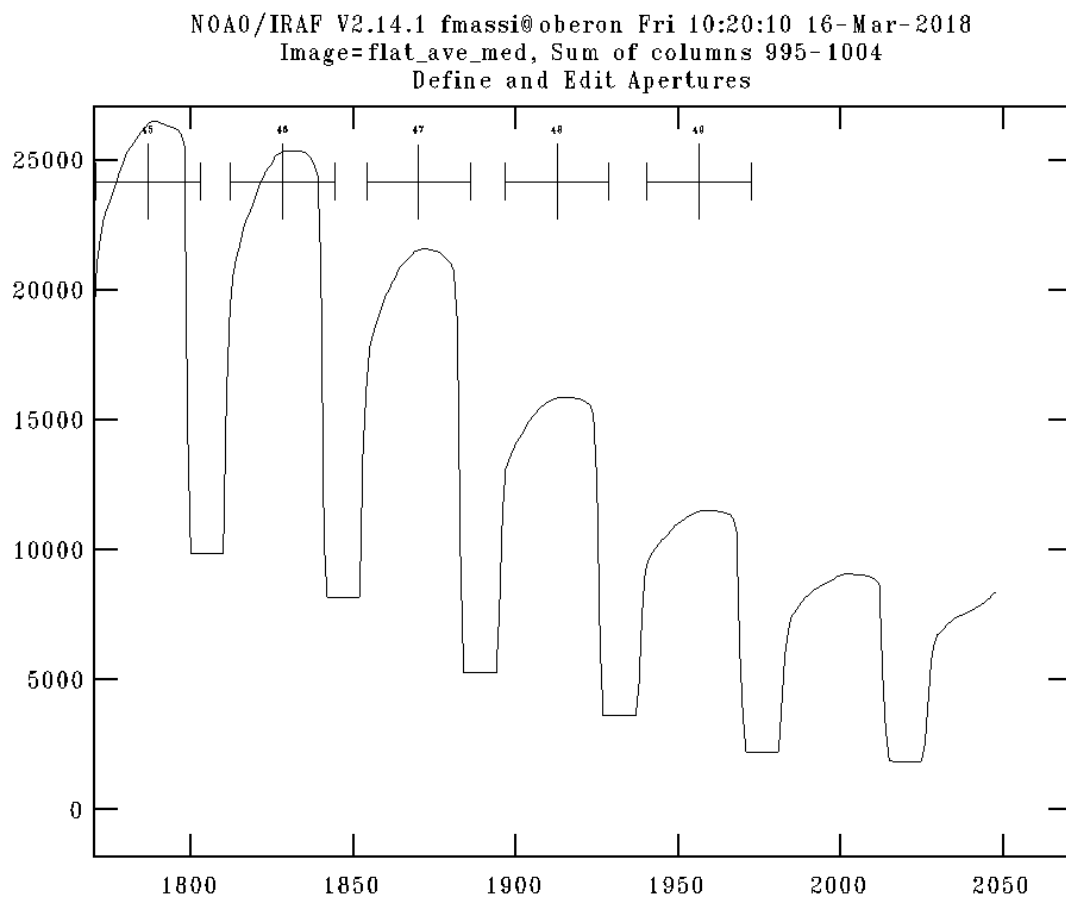


Figure 23: Graphic window showing the apertures found, after zooming in on the topmost orders using the graphic commands as explained in the text. **We note that the image displayed is the smoothed combined flat-field frame.**

Having finished with aperture editing, you need to move to the next step by typing “q”. You will be asked other questions, just answer yes to all.

Appendix B: Fitting traces

Once out of the aperture editing display of APTRACE, you will be shown the fits to the tracks (trace fits), one by one, in the graphic window; the first display will show the curve fit to aperture 1 (see Fig. 24). By typing “q”, it will move to the next aperture. The long and short of it is, an algorithm finds all peaks along columns, then polynomials are fitted to each string of peaks corresponding to the same track. The fit parameters are listed on top of the graphic window. Keep an eye on the RMS, it must be less than one tenth to few hundredths of a pixel for the fit to be good enough. Otherwise, you may need to increase the fit order. The graphic commands to do this are listed below. Be aware that after changing any parameter, you will have to redo the fit by typing “f”. A good fit looks like that in Fig. 24. At the highest orders you may find something like in Fig. 25; just delete the deviant points on the top right corner and redo the fit as explained below. This is ok and due to the low signal level at the highest orders near the right edge of the frame.

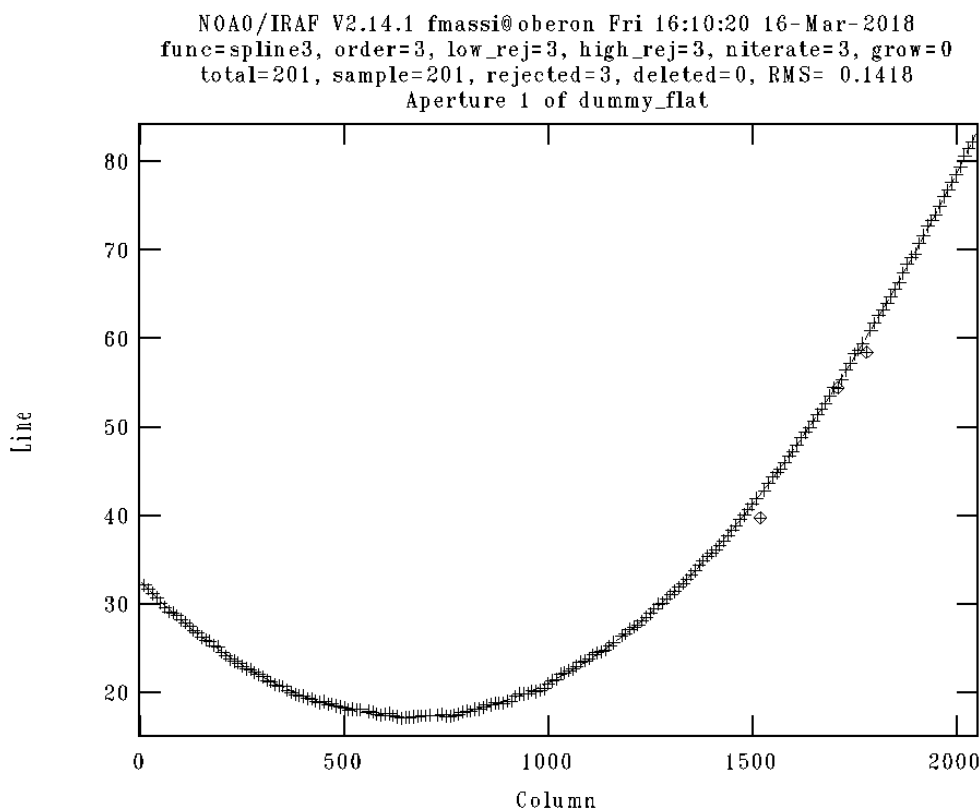


Figure 24: Graphic window showing a typical trace fit.

1. to start the help page, type “?”. The cl window will display a list of commands that you can scroll down. After you have finished, type first “q” and then “return” to return to the graphic window.
2. to enlarge the view, first type “w”; then move the cursor marker to the lower left corner of the area you want to enlarge and type “e”. Finally, move the cursor marker to the upper right corner of the area you want to enlarge and type again “e”.
3. to return to the largest view, type “w” and then “a”.

4. to delete a deviant point, put the cursor marker as near as possible to it and type “d”.
5. to undelete a deleted point, put the cursor marker as near as possible to it and type “u”.
6. to change function, type “:function *fname*”, where *fname* can be chebyshev, legendre, spline3, or spline1.
7. to change polynomial order, type “:order *value*” where *value* is the chosen order.
8. to change low (high) rejection threshold, type “:low_reject *value*” (“:high_reject *value*”), where *value* is the rejection threshold in sigma units.
9. to change the number of rejection iterations, type “:niterate *value*”, where *value* is the number of rejection iterations.
10. to redo the fit after changing any parameter, type “f”.

When you are satisfied with the current fit, type “q” to move to the next aperture (answer yes to every question). You will be shown all trace fits one by one. **Note that the fit parameters are global, so the last parameters input are used to update the fits to all traces.** After the last aperture has been fit, you will exit the task by typing “q”.

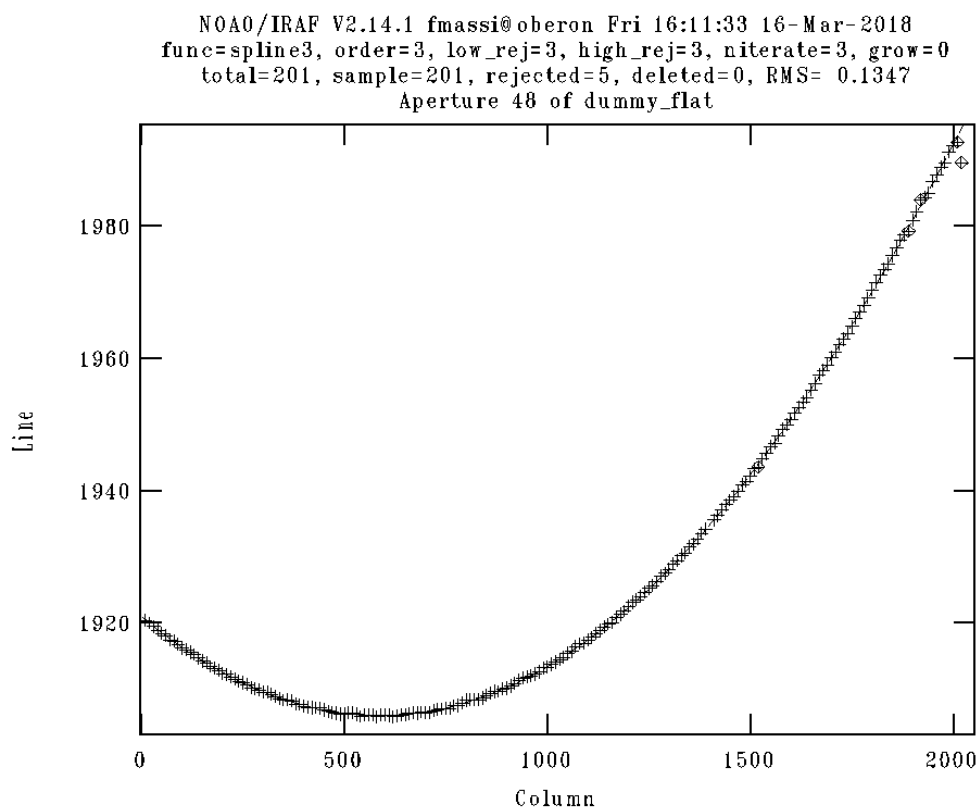


Figure 25: Graphic window showing a typical trace fit at high orders. Note the deviant points on the top right corner of the panel, which need being deleted and refitting.

Appendix C: Fitting spectra

When using APFLATTEN, after the trace fit session is over, the spectrum extraction process will start. For each trace, all pixels in a column within a given distance (according to the selected parameters) from the trace centre (i. e., falling inside the corresponding aperture)) are summed together, and a 1D spectrum is constructed. The graphic window will display the spectrum extracted from aperture 1 and the fit results. As can be seen in Figs. 26 and 27, the fit parameters are listed on top of the graphic window, whereas the polynomial fit is overlaid on the spectrum, and the rejected points are marked. If you are not satisfied with the fit results, you may want to change polynomial order and refit as explained below. Otherwise, by typing “q” the graphic session will move to the next aperture. Be aware that after changing any parameter, you will need to type “f” to perform a new fit.

IMPORTANT REMARK Flat-field spectra are only intended to remove the pixel-to-pixel response variations of the detector. So, you don’t have to fit every small oscillation of the spectrum. You will need to use a polynomial of order 5 to follow the spectrum at the highest orders, but be careful not to exceed this value. Alternatively, you can simply use an order 1 to also remove the spectrograph response (during the flat-fielding phase) normalising the target 1D spectra to the continuum (however, read the caveats in Sect. 3.16).

1. to start the help page, type “?”. The cl window will display a list of commands that you can scroll down. After you have finished, type first “q” and then “return” to return to the graphic window.
2. to enlarge the view, first type “w”; then move the cursor marker to the lower left corner of the area you want to enlarge and type “e”. Finally, move the cursor marker to the upper right corner of the area you want to enlarge and type again “e”.
3. to return to the largest view, type “w” and then “a”.
4. to delete a deviant point, put the cursor marker as near as possible to it and type “d”.
5. to undelete a deleted point, put the cursor marker as near as possible to it and type “u”.
6. to change function, type “:function *fname*”, where *fname* can be chebyshev, legendre, spline3, or spline1.
7. to change polynomial order, type “:order *value*” where *value* is the chosen order.
8. to change low (high) rejection threshold, type “:low_reject *value*” (“:high_reject *value*”), where *value* is the rejection threshold in sigma units.
9. to change the number of rejection iterations, type “:niterate *value*”, where *value* is the number of rejection iterations.
10. to redo the fit after changing any parameter, type “f”.

When you are satisfied with the current fit results, type “q” to move to the next aperture (answer yes to every question). You will be shown the curve fit to all extracted spectra one by one. **Note that the fit parameters are global, so the last parameters input are used to update the fits to all spectra.** After the last spectrum has been fit (aperture 49), typing “q” will start the next graphic session.

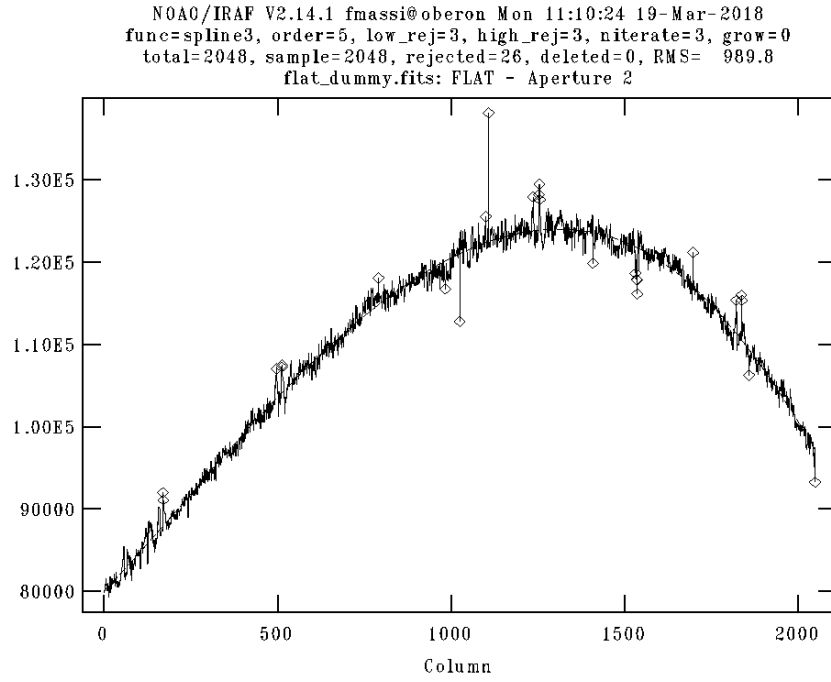


Figure 26: Graphic window showing a polynomial fit to the extracted continuum lamp spectra from one of the lowest spectral orders.

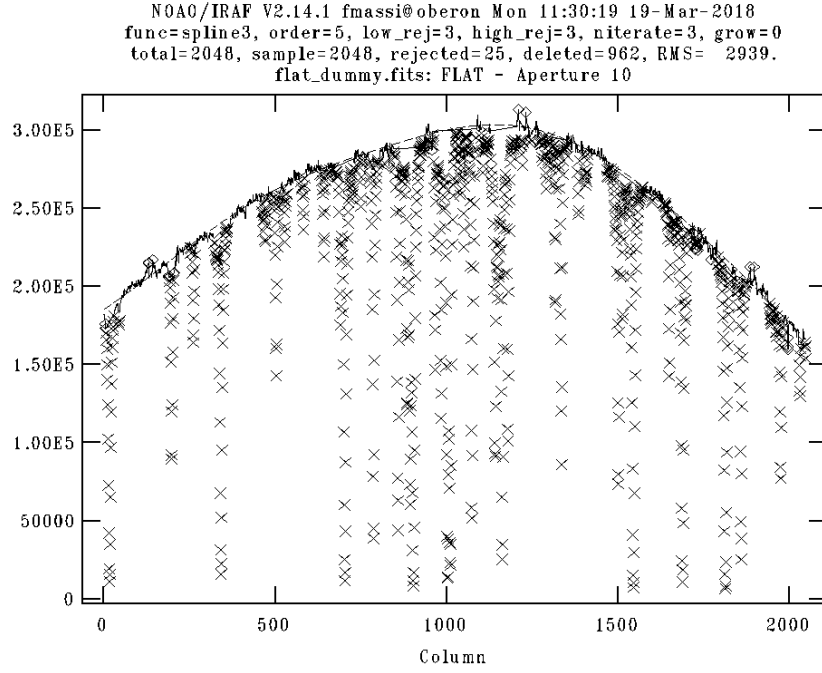


Figure 27: Graphic window showing a polynomial fit to the extracted continuum lamp spectra from one of the central spectral orders. The crosses mark deleted points corresponding to absorption lines. A few central orders exhibit narrow absorption lines that need to be excluded from the fit. However, these coincide with spectral regions of atmospheric absorption.

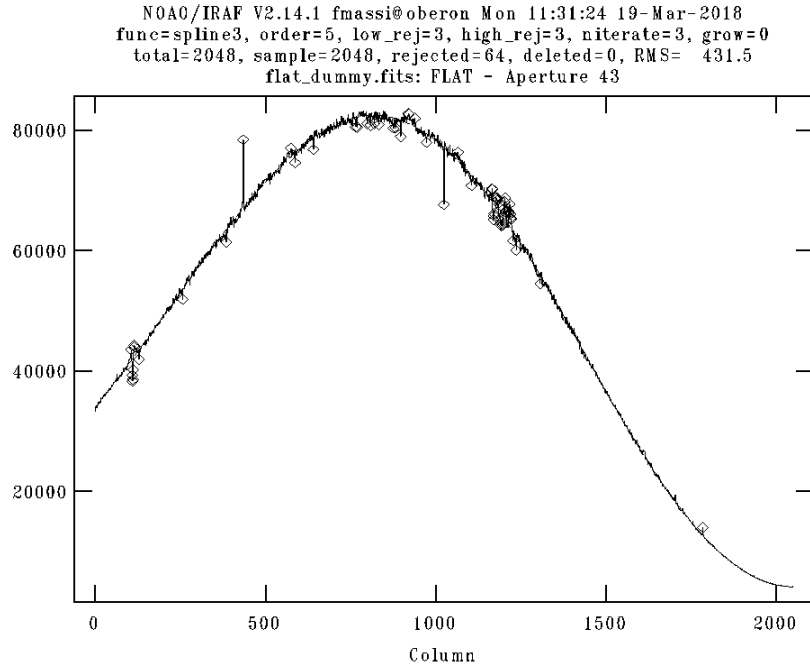


Figure 28: Graphic window showing the polynomial fit to aperture n. 43.

Appendix D: Editing spectra using SPLOT

To edit the extracted (wavelength-calibrated) 1D spectra, i. e. the fits files with a string “.ec.fits” added to the name, just use the task SPLOT. This task is also invoked by DOECSLIT at end of the job and allows you to revise the extracted spectra. To run SPLOT on a spectrum, just type “splot SPECNAME”, where specname is the name of the fits file with the extracted 1D spectra. You will be asked which aperture you want to review (of course, from 1 to 49). Input its number and enter “return”. Below, a list of basic graphic commands that allow handling SPLOT.

1. to start the help page, type “?”. The cl window will display a list of commands that you can scroll down. After you have finished, type first “q” and then “return” to return to the graphic window.
2. to enlarge the view, first type “w”; then move the cursor marker to the lower left corner of the area you want to enlarge and type “e”. Finally, move the cursor marker to the upper right corner of the area you want to enlarge and type again “e”.
3. to return to the largest view, type “w” and then “a”.
4. to change aperture, type “# value”, where value is the aperture number.
5. to quit, type “q”.

The task also allows you to perform some operations on spectra, see the help for more information.

Appendix E: Computing effective gain and readnoise

Readout noise and gain of a single raw GIANO-B frame are listed in Table 1. However, since a final GIANO-B 2D image gets somewhat modified, due to subtraction, sum, average, etc., its effective readnoise and gain (before 1D spectrum extraction) are also modified. To compute them, simply apply recursively (i. e., after each step) the following rules:

Sum of N frames:

effective exposure time = $N \times$ exposure time

effective gain = gain (e^-/ADU)

effective readout noise = $\sqrt{(N)} \times$ readout noise (e^-)

Subtraction of 2 frames:

effective exposure time = exposure time

effective gain = gain (e^-/ADU)

effective readout noise = $\sqrt{(2)} \times$ readout noise (e^-)

Average of N frames:

effective exposure time = exposure time

effective gain = $N \times$ gain (e^-/ADU)

effective readout noise = $\sqrt{(N)} \times$ readout noise (e^-)

Median of N frames:

effective exposure time = exposure time

effective gain = $2 \times N \times$ gain /3 (e^-/ADU)

effective readout noise = $\sqrt{(2 \times N/3)} \times$ readout noise (e^-)

The final subtracted (and combined) frames will then have exposure time t_{eff} , gain g_{eff} , and read-out noise R_{eff} . By applying again the formulae above when combining A and B spectra, one obtains for the final 1D spectra:

Single trace out of either A or B position

effective exposure time = t_{eff}

effective gain = g_{eff} (e^-/ADU)

effective readout noise = R_{eff} (e^-)

Summing A and B spectra

effective exposure time = $2 \times t_{\text{eff}}$

effective gain = g_{eff} (e^-/ADU)

effective readout noise = $1.4 \times R_{\text{eff}}$ (e^-)

Averaging A and B spectra

effective exposure time = t_{eff}

effective gain = $2 \times g_{\text{eff}}$ (e^-/ADU)

effective readout noise = $1.4 \times R_{\text{eff}}$ (e^-)

Appendix F: Automatic arc assignment parameters

The wavelength-calibration is performed by DOECSLIT. The task allows selecting the lamp spectra to use for calibrating each science spectrum based on several possible criteria (nearest in time, average, etc.). The field *arcs* must be set to a list of calibration lamp 2D spectra. The criterion to select (“assign”) a spectrum from the list is controlled by the value in the field *select*, in the “Automatic arc assignment” parameter section of the parameter file SPARAMS. Detailed assignments are possible (see the help), but the most useful values are the following. These are based on the observation date and time (in Julian day, having set *sort* to “jd”).

1. *average* The average of the first two spectra in the list is used. If only one reference spectrum is specified then it is assigned with a warning. If more than two reference spectra are specified then only the first two are used and a warning is given.
2. *following* Select the nearest following spectrum in the reference list based on the sorting parameter (in our case, time). If there is no following spectrum it uses the nearest preceding spectrum.
3. *interp* Interpolate between the preceding and following spectra in the reference list based on the sorting parameter. If there is no preceding and following spectrum it uses the nearest spectrum. The interpolation is weighted by the relative distances of the sorting parameter (in our case, time).
4. *nearest* Select the nearest spectrum in the reference list based on the sorting parameter (in our case, time).
5. *preceding* Select the nearest preceding spectrum in the reference list based on the sorting parameter (in our case, time). If there is no preceding spectrum it uses the nearest following spectrum.

Appendix G: Wavelength-calibration

After confirming the choice of the apertures, DOECSLIT will move to the wavelength-calibration session. There is no trace-fitting session, since we are using the traces from APTRACE. The graphic window will display a 1D spectrum extracted from one of the spectral orders of the calibration lamp frame (see Fig. 29). You need to identify a few lines and input their wavelength for a few orders. We advise you to use aperture number 1, 2, 10, 11, 33, 35, 38, 49 and to input the wavelengths of the lines shown in Fig. 32, 33, and 34. We have checked against the wavelengths of telluric absorption and emission lines ([7,8]) that using all the lines in the figures results in a wavelength accuracy < 0.01 nm on the whole passband (see Fig. 30; note that the pixel sampling interval is ~ 0.02 nm).

We performed another test to check the accuracy of wavelength calibration. [8] have derived a stable algorithm for wavelength calibration based on an analysis of sintetic spectra obtained from ray tracing. They found that the calibration can be approximated within ~ 0.03 pixel by a third degree polynomial, of which they provide coefficients. The wavelength calibration yielded by fitting this polynomial differs from that obtained following this handbook by less than 0.01 nm (in fact, less than 0.005 nm in most of the orders). The difference ranges from less than 1/4 the resolution element at the high wavelength end (K) to less than 1/3 the resolution element at the low wavelength end (YJ). An in-depth analysis indicates that the accuracy is limited by the precision achieved in deriving the centroid of the reference lamp lines rather than the polynomial fit used. The line fwhm ranges from 0.05 nm in K to 0.03 nm in Y , i. e. of the order of the spectral resolution element. Calibration lines are then poorly sampled. Thus, the error in the centre position is $\sim \text{fwhm}/\sqrt{N}$, where N is the number of sampling points. Assuming $N = 13$, the position error ranges from ~ 0.014 nm to 0.008 nm, as expected. Line blending and possible saturation of the brightest lines make the position computation even more biased. Therefore, one has no gain in precision by implementing the algorithm of [8] when using the U-Ne lamp.

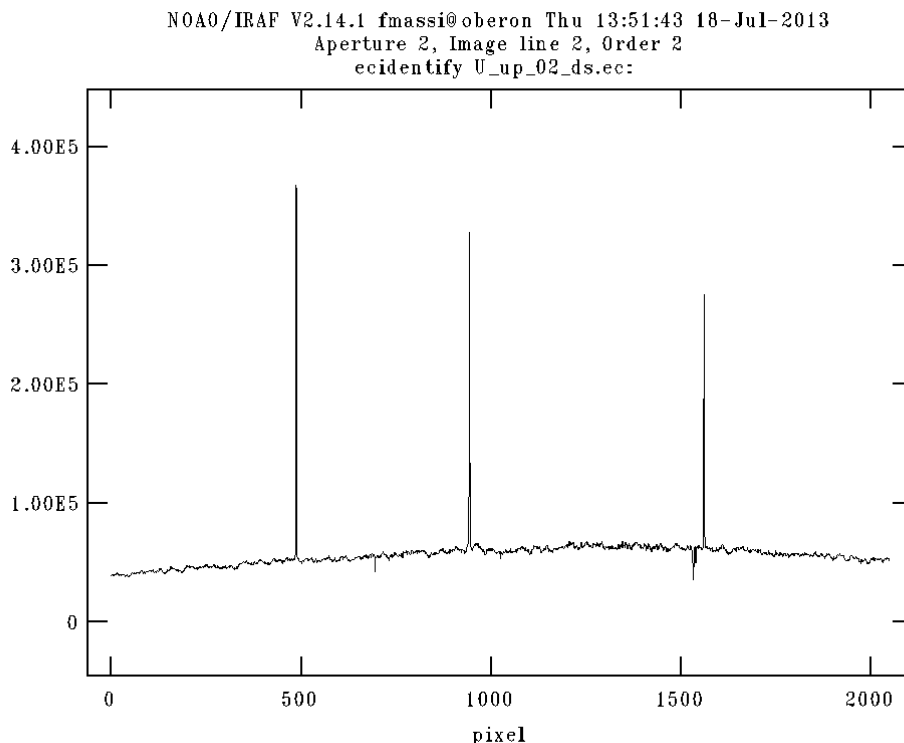


Figure 29: Graphic window showing a spectrum extracted from an aperture of the calibration lamp frame, waiting for the lines displayed to be identified.

The line wavelengths must be input in nm (as they are listed in the reference file list). Note also that DOECSLIT refers to the apertures or traces as orders. Actually, aperture 1 corresponds to order 32 of the echelle spectrum. This task uses the line catalogue file indicated in the field *coordli* of the SPARAMS file. This file is part of the GIANO-TOOLS package, so do not worry about it. **If a wavelength solution already exists then the graphic window will display a calibrated spectra with identified lines already marked. Skip the line identification stage and have a look at the fit results by pressing “f” (as a graphic command). Check the fit results as explained below. If not satisfied, you can unmark all lines and redo the line assignment. But usually you should be able to leave the session without any action at all.**

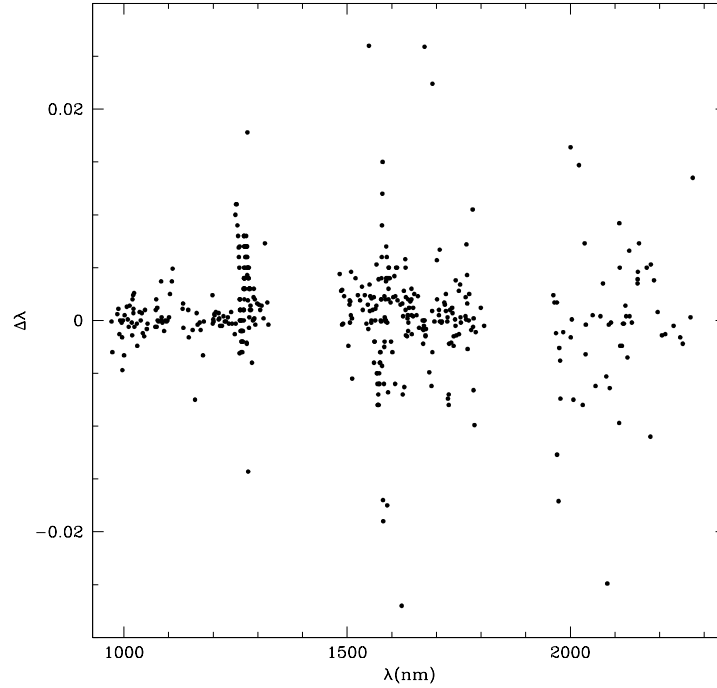


Figure 30: Wavelength difference (in nm) between observed absorption or emission telluric lines and their counterparts identified from [9] or [10]. We note that differences > 0.01 nm usually arise due to heavy blending or low s/n of the line in the spectrum, making it difficult to locate its centre. Note that the pixel sampling interval is ~ 0.02 nm).

In order to identify the lines and input their wavelengths, use the following graphic commands:

1. to start the help page, type “?”. The cl window will display a list of commands that you can scroll down. After you have finished, type first “q” and then “return” to return to the graphic window.
2. to enlarge the view, first type “w”; then move the cursor marker to the lower left corner of the area you want to enlarge and type “e”. Finally, move the cursor marker to the upper right corner of the area you want to enlarge and type again “e”.
3. to return to the largest view, type “w” and then “a”.
4. to enter the wavelength of a line, first move the graphic cursor marker onto the line, then type “m”. The line will be marked and the graphic window will ask for the wavelength. Type the wavelength (in nm) and enter “return”.

5. to delete an identified line, first move the graphic cursor marker onto the line, then type “u”.
6. to change aperture, type “o” and enter the aperture number.

After all the lines recommended have been identified, type “f” to perform a preliminary fit. The graphic window will display the fit results (fit residuals vs. pixel scale) as in Fig. 31. As usual, the fit parameters are listed on the top of the graphic window. It is mandatory that you check that the field *offset* displayed in the graphic windows is *offset=31* (i. e., aperture 1 corresponds to order $31 + 1 = 32$, as it must be).

If the offset is not 31, return to the previous step by typing “q”. Check the line identifications and, where needed, delete the incorrect ones by typing “u”, then mark them again with “m” and assign the correct wavelengths.

The next result to check is the rms, which must be less than few hundredth of a nm. To get a better fit, you can perform the following operations:

1. to start the help page, type “?”. The cl window will display a list of commands that you can scroll down. After you have finished, type first “q” and then “return” to return to the graphic window.
2. to change the *x*-axis scale from pixel to order, type “x” and then “o”.
3. to change the *x*-axis scale from pixel to wavelength, type “x” and then “w”.
4. to go back to the *x*-axis scale in pixel, type “x” and then “p”.
5. to delete a deviant point, put the cursor marker as near as possible to it and type “u”.
6. to change function, type “:function *fname*”, where *fname* can be chebyshev or legendre.
7. to change polynomial orders, type “:xorder *value*” or/and “:yorder *value*” where *value* means the order.
8. to change low (high) rejection threshold, type “:lowreject *value*” (“:highreject *value*”), where *value* means the rejection threshold in sigmas.
9. to change the number of rejection iterations, type “:niterate *value*”, where *value* means the number of rejection iterations.
10. to redo the fit after changing any parameter, type “f”.

Be aware that you need to redo the fit (by typing “f”) after any change of parameters. When you are satisfied with the fit, return to the line identification display by typing “q”.

Now you can run the automatic finding procedure that browses into the line catalogue and try to identify as many lines as possible. Just type:

“:maxfeatures 800” followed by ENTER and then “l”.

After this, you can move the display within orders (with “o” as explained above) to verify that many more lines have been marked. Perform a new fit by typing “f”. You will return to the fit result display. Follow the explanations above to obtain a better fit. **Check again that offset is equal to 31.** When satisfied, type “q”. You will be again moved to the line identification display (like Fig. 31). Type “q” again and you will be through.

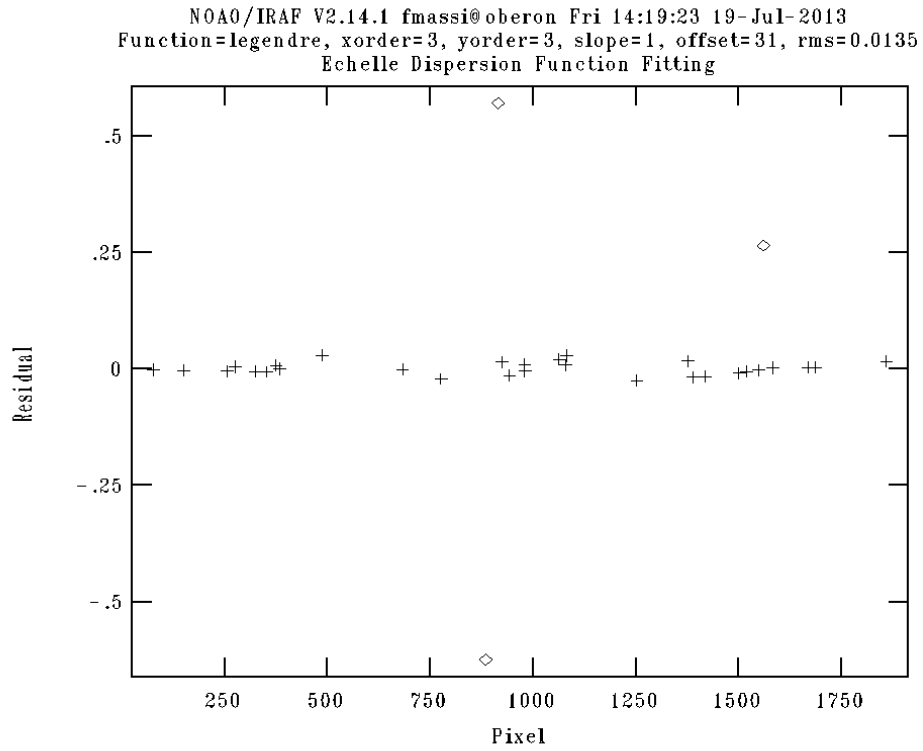


Figure 31: Graphic window showing the preliminary wavelength solution after the reference line identification step.

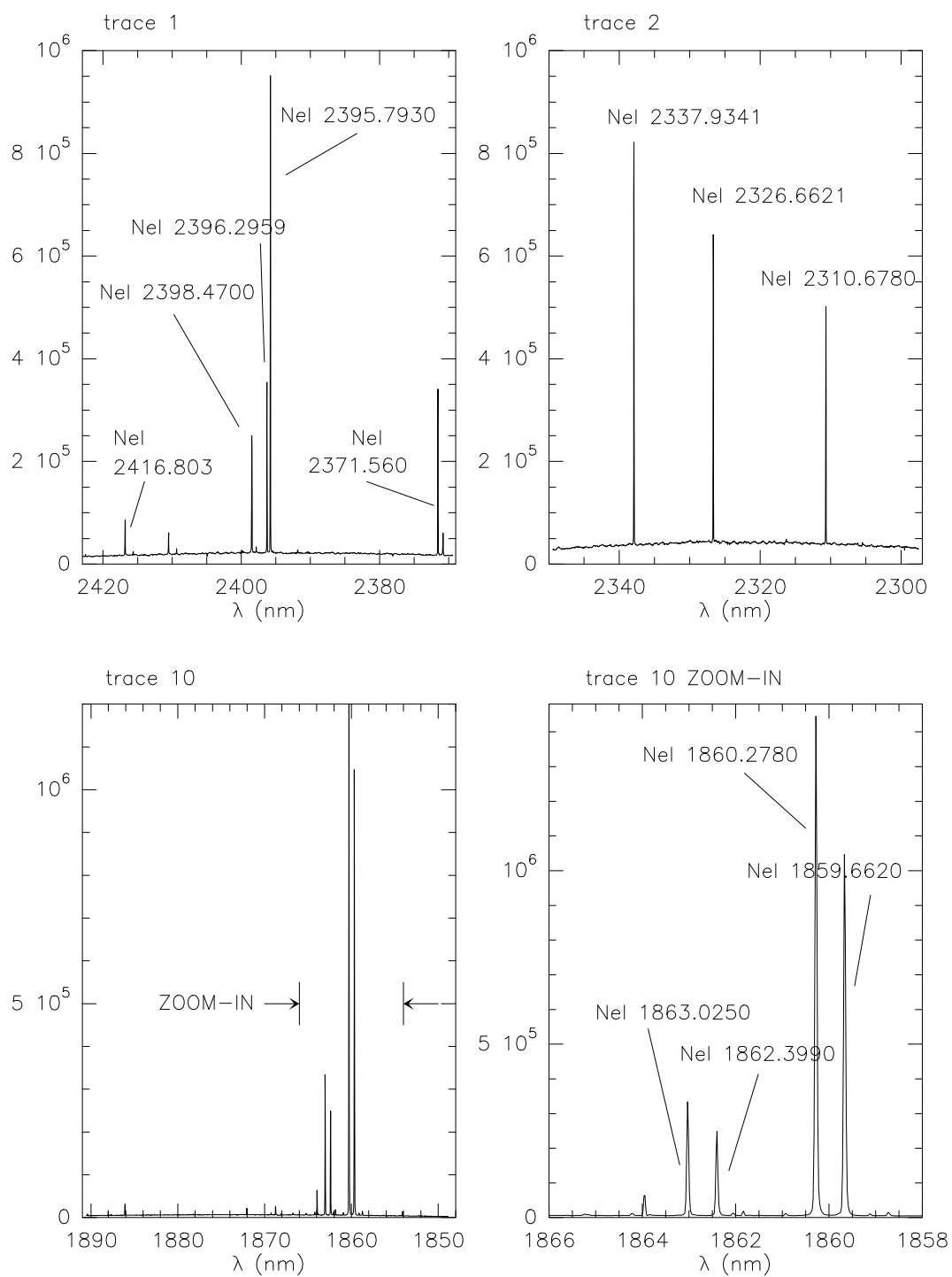


Figure 32: Calibration lamp lines to be used for wavelength calibration. **Note that the wavelength scale is reversed so that it mimics the pixel scale shown in the graphic windows.**

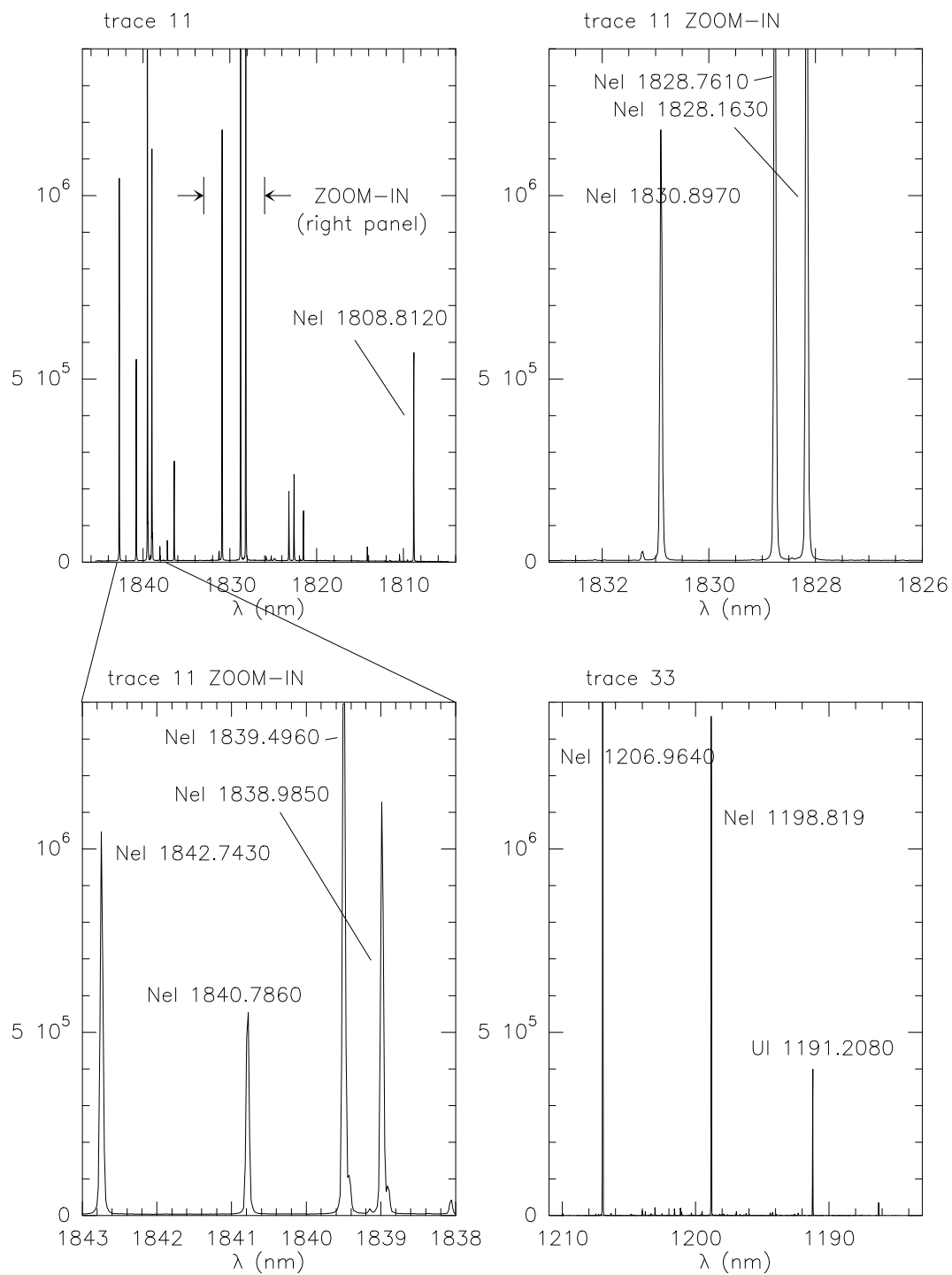


Figure 33: Calibration lamp lines to be used for wavelength calibration. **Note that the wavelength scale is reversed so that it mimics the pixel scale shown in the graphic windows.**

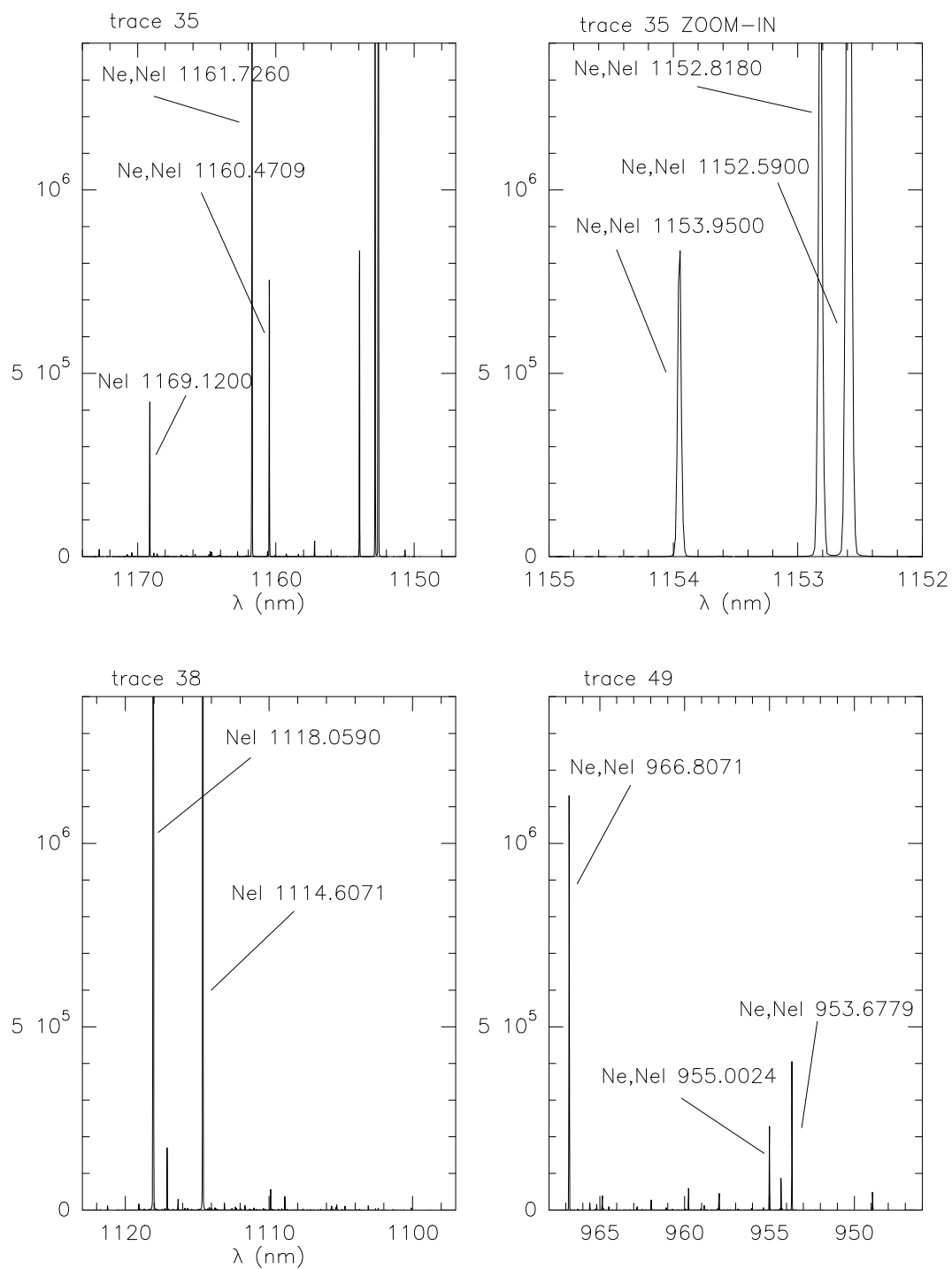


Figure 34: Calibration lamp lines to be used for wavelength calibration. **Note that the wavelength scale is reversed so that it mimics the pixel scale shown in the graphic windows.**

Appendix H: Wavelength range per order

Order 32	2.3699–2.4224 μm
Order 33	2.2979–2.3490 μm
Order 34	2.2302–2.2799 μm
Order 35	2.1664–2.2146 μm
Order 36	2.1062–2.1531 μm
Order 37	2.0492–2.0948 μm
Order 38	1.9952–2.0396 μm
Order 39	1.9440–1.9872 μm
Order 40	1.8953–1.9375 μm
Order 41	1.8490–1.9902 μm
Order 42	1.8049–1.8451 μm
Order 43	1.7629–1.8021 μm
Order 44	1.7228–1.7611 μm
Order 45	1.6844–1.7219 μm
Order 46	1.6478–1.6844 μm
Order 47	1.6126–1.6486 μm
Order 48	1.5790–1.6142 μm
Order 49	1.5467–1.5812 μm
Order 50	1.5157–1.5495 μm
Order 51	1.4860–1.5191 μm
Order 52	1.4573–1.4898 μm
Order 53	1.4298–1.4616 μm
Order 54	1.4033–1.4345 μm
Order 55	1.3777–1.4084 μm
Order 56	1.3531–1.3832 μm
Order 57	1.3293–1.3589 μm
Order 58	1.3063–1.3354 μm
Order 59	1.2841–1.3127 μm
Order 60	1.2627–1.2908 μm
Order 61	1.2419–1.2696 μm
Order 62	1.2219–1.2491 μm
Order 63	1.2024–1.2292 μm
Order 64	1.1836–1.2100 μm
Order 65	1.1654–1.1913 μm
Order 66	1.1477–1.1732 μm
Order 67	1.1305–1.1557 μm
Order 68	1.1138–1.1387 μm
Order 69	1.0977–1.1221 μm
Order 70	1.0819–1.1060 μm
Order 71	1.0667–1.0904 μm
Order 72	1.0518–1.0753 μm
Order 73	1.0374–1.0605 μm
Order 74	1.0233–1.0461 μm
Order 75	1.0096–1.0321 μm
Order 76	0.9963–1.0185 μm
Order 77	0.9833–1.0053 μm
Order 78	0.9707–0.9923 μm
Order 79	0.9584–0.9797 μm
Order 80	0.9464–0.9675 μm

Wavelengths are in vacuum.

(from Table 2 of *GIANO cookbook for proposers*)