EUROPEAN SOUTHERN OBSERVATORY

Organisation Européenne pour des Recherches Astronomiques dans l'Hémisphère Austral

Europäische Organisation für astronomische Forschung in der südlichen Hemisphäre

VERY LARGE TELESCOPE

FORS2 Spectroscopy Reflex Tutorial and Cookbook

VLT-MAN-ESO-19500-....

Issue 2.2

Date 7th July 2025

Prepared:	S. Moehler, Z. Kostrzewa	7th July 2025	
	Name	Date	Signature
Approved:	W. Freudling		
	Name	Date	Signature
Released:	M. Sterzik		
	Name	Date	Signature

This page was intentionally left blank

Change record

Issue/Rev.	Date	Section/Parag. affected	Reason/Initiation/Documents/Remarks
1.0	02/05/2013	All	First public release 4.9.22
1.1	01/04/2013	All	Changes for release 4.9.23
1.2	01/10/2013	All	Changes for release 4.10.8
1.3	06/11/2013	All	Changes for release 4.11.11
1.4	10/04/2014	All	Changes for release 4.12.8
1.5	18/08/2014	All	Changes for release 4.12.21
1.6	02/10/2014	All	Changes for release 5.0.0
1.8	01/04/2015	All	Changes for release 5.0.11
1.9	01/04/2016	All	Changes for release 5.3.4
1.10	01/04/2017	All	Changes for release 5.3.23
1.20	01/10/2017	All	Changes for release 5.3.32
1.21	April 2019	All	Changes for release 5.4.3
1.23	June 2020	All	Changes for release 5.5.4
1.24	May 2021	All	Changes for release 5.5.7
1.25	May 2022	All	Changes for release 5.6.2
1.26	August 2022	Section 9	Changes for release 5.6.3
1.27	17/05/2023	Section 11	Changes for release 5.6.5
2.0	12/03/2025	Sections 3,5,6,8,9	Changes for release 5.8.1
2.1	24/04/2025	Sections 10,11,12	Changes for release 5.8.2
2.2	07/07/2025	Section 4	Update for release 5.8.2

This page was intentionally left blank

Contents

1 Spectroscopy with the FORS2 instrument: an introduction

FORS is the visual and near-UV FOcal Reducer and low dispersion Spectrograph for the Very Large Telescope (VLT) of the European Southern Observatory (ESO). Two versions of FORS have been built and mounted at the VLT, but in April 2009, FORS1 was removed to make room for X-shooter, so only FORS2 is now in operation. FORS is designed as an all-dioptric instrument for the wavelength range from 330 nm to 1100 nm and provides an image scale in the standard mode of $0''.25$/pixel with the readout mode (2×2 binning).

FORS2 offers three spectroscopic modes: long-slit spectroscopy (LSS) using a mask with $6.8'$ long slits of different widths, and multi-object spectroscopy with movable slit blades (MOS, slit length about $20''$) and with masks (MXU, arbitrary slit length, curved and tilted slits possible). The spectrophotometric standard stars for flux calibration are always taken in MOS mode, where the slitlets are used to create a long slit of $5''$ width at the position of the long-slit used for the science data or at the center of the field-of-view for multi-object spectroscopy. FORS2 has several volume-phase holographic grisms, for which the response depends on the psoition on the CCD. This requires specific steps during the flux calibration, which are handled correctly by the pipeline.

The Longitudinal Atmospheric Dispersion Corrector corrects the effects of atmospheric refraction up to an airmass of about 1.5, thereby enabling observations with arbitrary angles on sky without flux losses due to atmospheric refraction.

The `FORS2` supports the correction of telluric absorption using the `molecfit` algorithm.

This document explains how to reduce data taken in any of the spectroscopic modes of `FORS1` and `FORS2`. It does not cover the other modes, which are described in different cookbooks and tutorial documents.

Information on FORS2 is given at `eso.org/sci/facilities/paranal/instruments/fors.html`, which also provides access to manuals. In particular, we recommend that you read the User Manual.

2 Data reduction of FORS2 spectroscopy

The reduction of FORS2 spectroscopy consists of essentially 4 steps:

1. Creating a master bias frame
2. Determining the dispersion relation and spatial distortion, together with the slit limits and creating a normalized flat field
3. Determining the instrumental response (optional)
4. Calibrating and extracting the spectra, including sky subtraction and flux calibration
5. Telluric correction of science spectra (optional)

The calibration frames can be obtained at the same time when downloading the science data from the ESO Science Archive with the `CalSelector` option.

Reducing and calibrating FORS2 spectrocopic observations requires the following frames:

- Science frames obtained during the night
- Calibration frames
  - Bias frames
  - Internal screen flats
  - Arc lamp exposures
  - Spectrophotometric standard star observations (optional)
  - Static calibrations, e.g. line catalog

2.1 Bias frames

Bias frames are taken with an exposure time of 0 seconds and a closed shutter. They thus record only the signal that is added during the read-out of the CCD to avoid negative numbers. A sequence of 5 or 20 bias frames[1] are taken the day following the observations as part of the FORS2 calibration plan. Using the full bias frame instead of just the pre-/overscan allows the user to correct for potential offsets between the bias level on the detector and the pre-/overscan regions.

2.2 Flat field and wavelength calibration

The main purpose of the flat-field is to remove the pixel-to-pixel sensitivity variations across the detector. As these variations act as a noise source, the precision of the flat-field correction will have direct consequences on the accuracy that can be achieved and on the signal-to-noise ratio of the reduced observations.

[1] The calibration plan was changed in Nov. 2014, and 20 bias frames are now taken per setting instead of 5.

For the spectroscopic modes one will use internal screen flats in most cases. These flats are taken during daytime with the telescope pointing to zenith and the instrument in calibration position.

For the wavelength calibration one may use the He, HgCd, and Ar lamps (at the lowest spectral resolution with grism `GRIS_150I`) and in addition the Ne lamp (at higher resolution). Wavelength calibration exposures are taken together with the flat field data to record the same mask position. They are observed during daytime only.

Flat field and arc lamp frames are used together to trace the slits and determine spatial distortion corrections. Therefore they are observed within one template to ensure that the mask positions does not change between the exposures.

2.3 Spectrophotometric standard stars

These are stars with a well known flux distribution, which are used to determine the instrumental response. They are usually observed during twilight within ± 3 days of the science data.

2.4 Static Calibration Data

In addition to calibration frames taken regularly, the FORS pipeline also uses static calibration tables related to spectroscopy. In the following list, the data organizer category is given in parenthesis.

1. Arc lamp line wavelengths (`MASTER_LINECAT`). This table contains the reference wavelengths (in Å) for the arc lamps and grism used.
2. Grism table (`GRISM_TABLE`). This table contains grism-specific parameters, like the dispersion in Å/pixel, the start and end wavelength, the polynomial degree to be used for the wavelength calibration, fit parameters for the response curve, etc.
3. Distortion table (`GLOBAL_DISTORTION_TABLE`). This table provides the correction of the spatial distortion for long-slit spectroscopy. It exists for most, but not all grisms, and is an optional input.
4. Spectrophotometric standard star table (`STD_FLUX_TABLE`). This file is needed to determine the instrumental response and contains the reference flux of the stars together with regions of strong stellar absorption that are by default ignored when fitting the response curve.
5. Telluric region table (`TELLURIC_CONTAMINATION`). This table contains for each grism affected by telluric absorption the wavelength regions that are by default ignored when fitting the response curve.
6. Extinction table describing the atmospheric extinction at the Paranal observatory (`EXTINCT_TABLE`). This table is needed for the creation and application of the response curve.
7. Earth Observation Parameter table (`EOP_PARAM`) containing the Earth Orientation Parameter as a function of time (MJD-OBS). This file is needed for the barycntric correction. An updated version of this table may be downloaded by using the recipe `fors_eop`.
8. Flux standard star catalog (`STD_FLUX_CATALOG`). This file provides the reference data (model spectra) for the 7 flux standard stars, that are used by the recipe `fors_response`) to determine the response.

9. Fit point catalog (`FIT_POINT_CAT`) providing the wavelength points per grism, filter and star. These points are used to fit a response curve to the raw response (ratio of reference to observed spectrum)

10. Table with wavelength regions for telluric absorption (`WAVE_INCLUDE`). This table contains for each combination of grism and filter the wavelength regions to be used by the `fors_molecfit_model`) recipe to fit the telluric absorption.

11. `MOLECULES` table. This table contains the list of molecules used by the `fors_molecfit...` recipes.

A description of these tables is given in the FORS Pipeline User Manual. We refer to this manual for detailed information on the recipes and on what parameters can be configured. These tables are provided together with the FORS pipeline and located in the directory <INSTALL-DIR>/calib/.

3 Running the FORS2 pipeline

There are three ways to run the ESO pipelines, in all cases the executed recipes are, however, the same. The differences are in the user interfaces.

1. Reflex is the **recommended** environment to reduce ESO data. It automatically organizes input files according to their category and runs the entire reduction chain at the push of a button. It supports break points in the reduction sequence in order to inspect and interact with intermediate and final products and rerun the corresponding step if necessary. A more detailed description on how to use Reflex to reduce FORS2 spectroscopic data is provided in sections 6 to 10.
2. Gasgano is a Java-based data file organizer developed and maintained by ESO. It can be used to manage and organize in a systematic way the astronomical data observed and produced by all VLT compliant telescopes. Gasgano offers functionalities for data viewing, grouping, sorting, classification, searching, and filtering of data. And, of course, Gasgano will run any requested CPL recipe on the selected data. Gasgano is automatically installed when installing the stand-alone FORS pipeline kit available from `http://www.eso.org/sci/software/pipelines/`, but not as part of the Reflex installation.
3. Esorex, a command-line utility for running pipeline recipes is also available (and is used also by Reflex to run the pipeline recipes). Esorex may be embedded by users into data reduction scripts for the automation of processing tasks. See `http://www.eso.org/sci/software/cpl/esorex.html` for more information.

The underlying algorithms and recipes are the same for a given instrument pipeline, irrespective if Reflex, Gasgano or Esorex is used.

The pipeline itself can be accessed from the web at `http://www.eso.org/sci/software/pipelines/`.

To reduce FORS2 spectroscopic data, one can just execute the following recipes in succession:

1. `fors_bias`: process all bias files
2. `fors_calib`: processes all lamp calibration files (flat fields and arcs).
3. `fors_science`: processes the standard star observations applying the calibrations processed in the previous step and creates a response curve for the stars with reference data in `STD_FLUX_TABLE`).
4. `fors_molecfit_model, fors_molecfit_calctrans, fors_molecfit_correct`: correct the telluric absorption for the standard stars with reference data (model spectra) in `STD_FLUX_CATALOG` used by the recipe `fors_response`
5. `fors_response`: determine the instrumental response from the spectra of the 7 standard stars, which are supported by this recipe, using either directly the results of `fors_science` (Step 3) or the telluric-corrected spectra (Step 4).
6. `fors_science`: processes the science data, applying the calibrations processed in the previous step.
7. `fors_molecfit_model, fors_molecfit_calctrans, fors_molecfit_correct`: correct the telluric absorption in the reduced science spectra

More information of the FORS2 pipeline is available in the FORS Pipeline User Manual.

The association of calibration files with the science data is based on the so-called OCA rules (see Sect. 9.3.1), also used by CalSelector to retrieve the correct calibrations from the archive with your science data. The OCA rules associate files by header keywords.

Gasgano (and also Reflex, see Sect. 9.3.1) allow to display files. For example, to check a wavelength calibration file, select one of category “LAMP_MOS” and choose the “Display...” option under the “Selected files” menu.

The following sections describe the data reduction using Reflex.

4 Introduction to `EsoReflex`

This document is a tutorial designed to enable the user to to reduce his/her data with the ESO pipeline run under an user-friendly environmet, called `EsoReflex`, concentrating on high-level issues such as data reduction quality and signal-to-noise (S/N) optimisation.

`EsoReflex` is the ESO Recipe Flexible Execution Workbench, an environment to run ESO VLT pipelines which employs a workflow engine to provide a real-time visual representation of a data reduction cascade, called a workflow, which can be easily understood by most astronomers. The basic philosophy and concepts of Reflex have been discussed by Freudling et al. (2013A&A...559A..96F). Please reference this article if you use Reflex in a scientific publication.

Reflex and the data reduction workflows have been developed by ESO and instrument consortia and they are fully supported. If you have any issue, please have a look to https://support.eso.org to see if this has been reported before or open a ticket for further support.

A workflow accepts science and calibration data, as downloaded from the archive using the CalSelector tool[2] (with associated raw calibrations) and organises them into DataSets, where each DataSet contains one science object observation (possibly consisting of several science files) and all associated raw and static calibrations required for a successful data reduction. The data organisation process is fully automatic, which is a major time-saving feature provided by the software. The DataSets selected by the user for reduction are fed to the workflow which executes the relevant pipeline recipes (or stages) in the correct order. Full control of the various recipe parameters is available within the workflow, and the workflow deals automatically with optional recipe inputs via built-in conditional branches. Additionally, the workflow stores the reduced final data products in a logically organised directory structure employing user-configurable file names.

The FORS2 `Reflex` workflow `fors_spec` described in this tutorial supports the reduction of FORS2 spectroscopic observations, only LSS/MOS/MXU, no PMOS. The user is referred to the FORS User Manual [3] for more information on the instrument itself, and the FORS Pipeline User Manual[4] for the details of the spectroscopic FORS2 pipeline recipes. It is possible to reduce data from FORS1 with a dedicated `Reflex` workflow `fors_spec_fors1`.

The FORS2 pipeline can provide good quality results for most of the spectroscopic data (LSS/MOS/MXU), but may require fine-tuning of parameters to achieve the best results. Data with slits distributed across a large range along the dispersion axis can be difficult to process with one set of parameters for all slits. The workflow provides interactive windows for most recipes that allow the user to verify the quality of the reduction and tweak the parameters if necessary. The FORS2 pipeline underwent a thorough evaluation in 2013-2014, which greatly improved the quality of the results, esp. with respect to error propagation, distortion correction, and flux calibration.

Optionally, the 1D spectra in IDP format can be corrected for telluric absorption using the `molecfit` algorithm. For each dataset, instead of a telluric standard, the extracted spectrum with the higher signal-to-noise ratio is used as reference to compute the atmosphere absorption. Only spectra that extend further than 6800 Å are considered by `esoreflex` for the telluric correction.

The quick start section (see Section 6) describes the minimum effort to get started, and it makes up only two

[2] *https://www.eso.org/sci/archive/calselectorInfo.html*
[3] available at *http://www.eso.org/sci/facilities/paranal/instruments/fors/doc*
[4] available at *https://www.eso.org/sci/software/pipelines/index.html* (Documentation)

pages of text in this tutorial.

Sections 9.4 (p. 43), 10 (p. 57), 11.1 (p. 66), and 12.2 (p. 73) provide useful information for FORS2 data reduction irrespective of the use of Reflex workflows.

5 Software Installation

`Esoreflex` and the workflows can be installed in different ways: via package repositories, via the `install_esoreflex` script or manually installing the software tar files.

The recommended way is to use the package repositories if your operating system is supported. The pipelines and Reflex can be installed from the ESO `macports` repositories that support macOS platforms, the and the `rpm/yum` repositories that support Fedora and CentOS platforms. For any other operating system it is recommended to use the `install_esoreflex` script.

The installation from package repository requires administrative privileges (typically granted via sudo), as it installs files in system-wide directories under the control of the package manager. If you want a local installation, or you do not have sudo privileges, or if you want to manage different installations on different directories, then use the `install_esoreflex` script. Note that the script installation requires that your system fulfill several software prerequisites, which might also need sudo privileges.

Reflex 2.11.x needs java JDK 11 to be installed.

Please note that in case of major or minor (affecting the first two digit numbers) Reflex upgrades, the user should erase the `$HOME/KeplerData`, `$HOME/.kepler` directories if present, to prevent possible aborts (i.e. a hard crash) of the `esoreflex` process.

5.1 Installing `Esoreflex` workflows via `macports`

This method is supported for the macOS operating system. It is assumed that macports (*https://www.macports.org*) is installed. Please read the full documentation at *https://www.eso.org/sci/software/pipelines/installation/macports.html*, which also describes the versions of macOS that are currently supported.

5.2 Installing `Esoreflex` workflows via `rpm/yum/dnf`

This method is supported for Fedora and CentOS platforms and requires sudo rights. Please read the full documentation at *https://www.eso.org/sci/software/pipelines/installation/rpm.html*, which also describes the versions of Fedora and CentOS that are currently supported.

5.3 Installing `Esoreflex` workflows via `install_esoreflex`

This method is recommended for operating systems other than what indicated above, or if the user has no sudo rights. Software dependencies are not fulfilled by the installation script, therefore the user has to install all the prerequisites before running the installation script.

The software pre-requisites for `Reflex 2.11` may be found at:
https://www.eso.org/sci/software/pipelines/reflex_workflows

To install the `Reflex 2.11` software and demo data, please follow these instructions:

1. From any directory, download the installation script:

```
wget https://eso.org/sci/software/pipelines/install_esoreflex
```

2. Make the installation script executable:

```
chmod u+x install_esoreflex
```

3. Execute the installation script:

```
./install_esoreflex
```

   and the script will ask you to specify three directories: the download directory `<download_dir>`, the software installation directory `<install_dir>`, and the directory to be used to store the demo data `<data_dir>`. If you do not specify these directories, then the installation script will create them in the current directory with default names.

4. Follow all the script instructions; you will be asked whether to use your Internet connection (recommended: yes), the pipelines and demo-datasets to install (note that the installation will remove all previously installed pipelines that are found in the same installation directory).

5. To start `Reflex`, issue the command:

```
<install_dir>/bin/esoreflex
```

   It may also be desirable to set up an alias command for starting the `Reflex` software, using the shell command `alias`. Alternatively, the `PATH` variable can be updated to contain the `<install_dir>/bin` directory.

5.4 Demo Data

Together with the pipeline you will also receive a demo data set, that allows you to run the `Reflex FORS2` workflow without any changes in parameters. This way you have a data set to experiment with before you start to work on your own data.

Note that you will need a minimum of ∼0.5 GB, ∼0.6 GB and ∼1.7 GB of free disk space for the directories `<download_dir>`, `<install_dir>` and `<data_dir>`, respectively. The FORS2 demo data have been retrieved with the CalSelector tool[5].

The data sets were carefully selected to illustrate potential problems and Sect. 10.1 describes how to improve the results obtained for these data sets.

[5] *http://www.eso.org/sci/archive/calselectorInfo.html*

6 Quick Start: Reducing The Demo Data

For the user who is keen on starting reductions without being distracted by detailed documentation, we describe the steps to be performed to reduce the science data provided in the FORS2 demo data set supplied with the `esoreflex 2.11` release. By following these steps, the user should have enough information to perform a reduction of his/her own data without any further reading:

1. First, type:

   ```
   esoreflex -l
   ```

   If the `esoreflex` executable is not in your path, then you have to provide the command with the executable full path `<install_dir>/bin/esoreflex -l` . For convenience, we will drop the reference to `<install_dir>`. A list with the available `esoreflex` workflows will appear, showing the workflow names and their full path.

2. Open the FORS2 Spectroscopy by typing:

   ```
   esoreflex fors_spec&
   ```

   Alternatively, you can type only the command `esoreflex` the empty canvas will appear (Figure 6.1) and you can select the workflow to open by clicking on `File -> Open File`. Note that the loaded workflow will appear in a new window. The FORS2 Spectroscopy workflow is shown in Figure 6.2.

3. To aid in the visual tracking of the reduction cascade, it is advisable to use component (or actor) highlighting. Click on `Tools -> Animate at Runtime`, enter the number of milliseconds representing the animation interval (100 ms is recommended), and click `OK`.

4. Change directories set-up. Under "Setup Directories" in the workflow canvas there are seven parameters that specify important directories (green dots).

   By default, the `ROOT_DATA_DIR`, which specifies the working directory within which the other directories are organised. is set to your `$HOME/reflex_data` directory. All the temporary and final products of the reduction will be organized under sub-directories of `ROOT_DATA_DIR`, therefore make sure this parameter points to a location where there is enough disk space. To change `ROOT_DATA_DIR`, double click on it and a pop-up window will appear allowing you to modify the directory string, which you may either edit directly, or use the `Browse` button to select the directory from a file browser. When you have finished, click `OK` to save your changes.

   Changing the value of `RAW_DATA_DIR` is the only necessary modification if you want to process data other than the demo data

5. Click the  button to start the workflow

6. The workflow will highlight the `Data Organiser` actor which recursively scans the raw data directory (specified by the parameter `RAW_DATA_DIR` under "Setup Directories" in the workflow canvas) and constructs the datasets. Note that the raw and static calibration data must be present either

in `RAW_DATA_DIR` or in `CALIB_DATA_DIR`, otherwise datasets may be incomplete and cannot be processed. However, if the same reference file was downloaded twice to different places this creates a problem as `esoreflex` cannot decide which one to use.

7. The `Data Set Chooser` actor will be highlighted next and will display a "Select Datasets" window (see Figure 6.3) that lists the datasets along with the values of a selection of useful header keywords[6]. The first column consists of a set of tick boxes which allow the user to select the datasets to be processed. By default all complete datasets which have not yet been reduced will be selected. A full description of the options offered by the `Data Set Chooser` will be presented in Section 9.3.2.

8. Click the `Continue` button and watch the progress of the workflow by following the red highlighting of the actors. A window will show which dataset is currently being processed.

9. For each execution of the *ForsCalib* actor, an interactive window will appear which shows some plots to evaluate the quality of the slit tracing, the wavelength calibration and the master flat creation. Figure 6.4 shows the look of that window. The first and fifth data set will create an error message during that step. See p. 57 how to proceed.

   This window will appear as many times as inputs of the calibration recipe exist. Note that some datasets contain calibrations for the standard star as well. For the time being just click on `Continue` to proceed to the next steps. Also the *Response Curve* (Fig. 9.3, p. 46) *ForsScience* (Fig. 9.6, p. 51), and *Telluric-Correction* (Fig. 9.7, p. 53; Fig. 9.8, p. 55) actors have associated interactive windows, where you should proceed in the same way.

10. Once the reduction of all datasets has finished, a pop-up window called *Product Explorer* will appear, showing the datasets which have been reduced together with the list of final products. This actor allows the user to inspect the final data products, as well as to search and inspect the input data used to create any of the products of the workflow. Figure 6.5 shows the Product Explorer window. A full description of the *Product Explorer* will be presented in Section 9.3.3.

11. After the workflow has finished, all the products from all the datasets can be found in a directory under `END_PRODUCTS_DIR` named after the workflow start timestamp. Further subdirectories will be found with the name of each dataset.

Well done! You have successfully completed the quick start section and you should be able to use this knowledge to reduce your own data. However, there are many interesting features of `Reflex` and the FORS2 workflow that merit a look at the rest of this tutorial.

[6]The keywords listed can be changed by double clicking on the `DataOrganiser` Actor and editing the list of keywords in the second line of the pop-up window. Alternatively, instead of double-clicking, you can press the right mouse button on the `DataOrganiser` Actor and select `Configure Actor` to visualize the pop-up window.

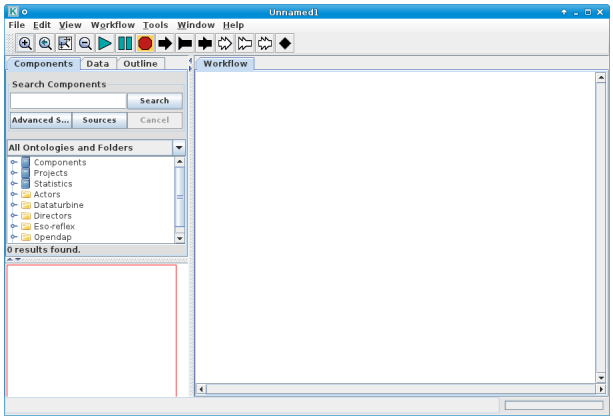


Figure 6.1: *The empty* `Reflex` *canvas.*

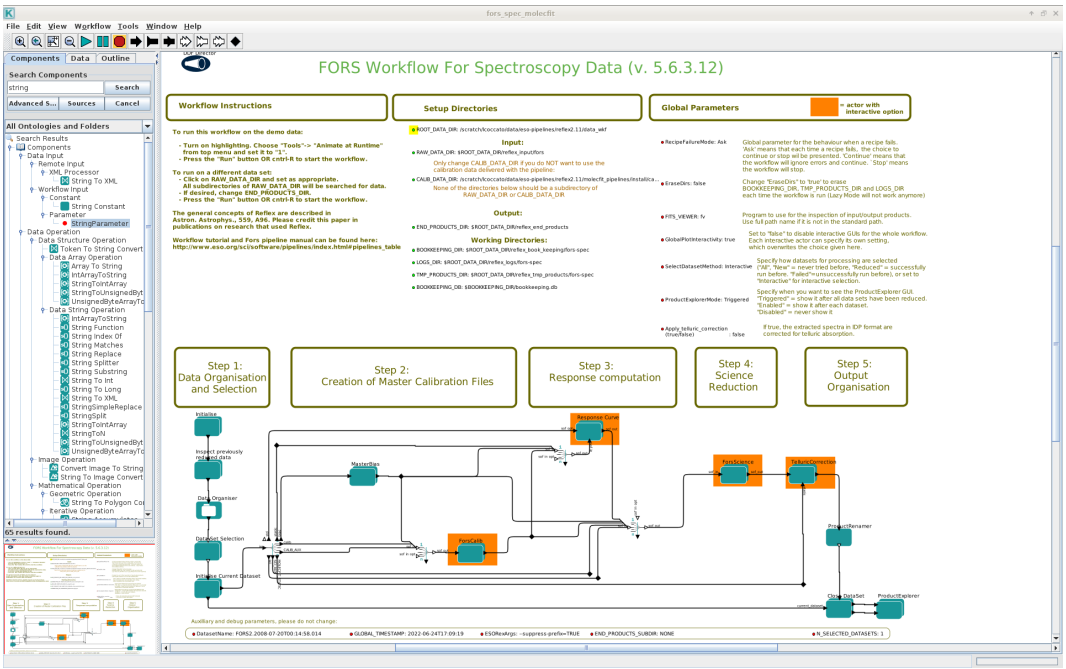


Figure 6.2: *FORS2 workflow general layout.*

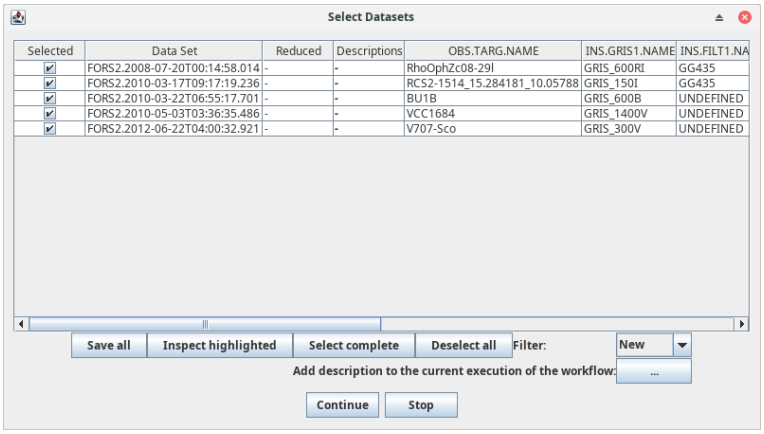


Figure 6.3: *The "Select Datasets" pop-up window.*

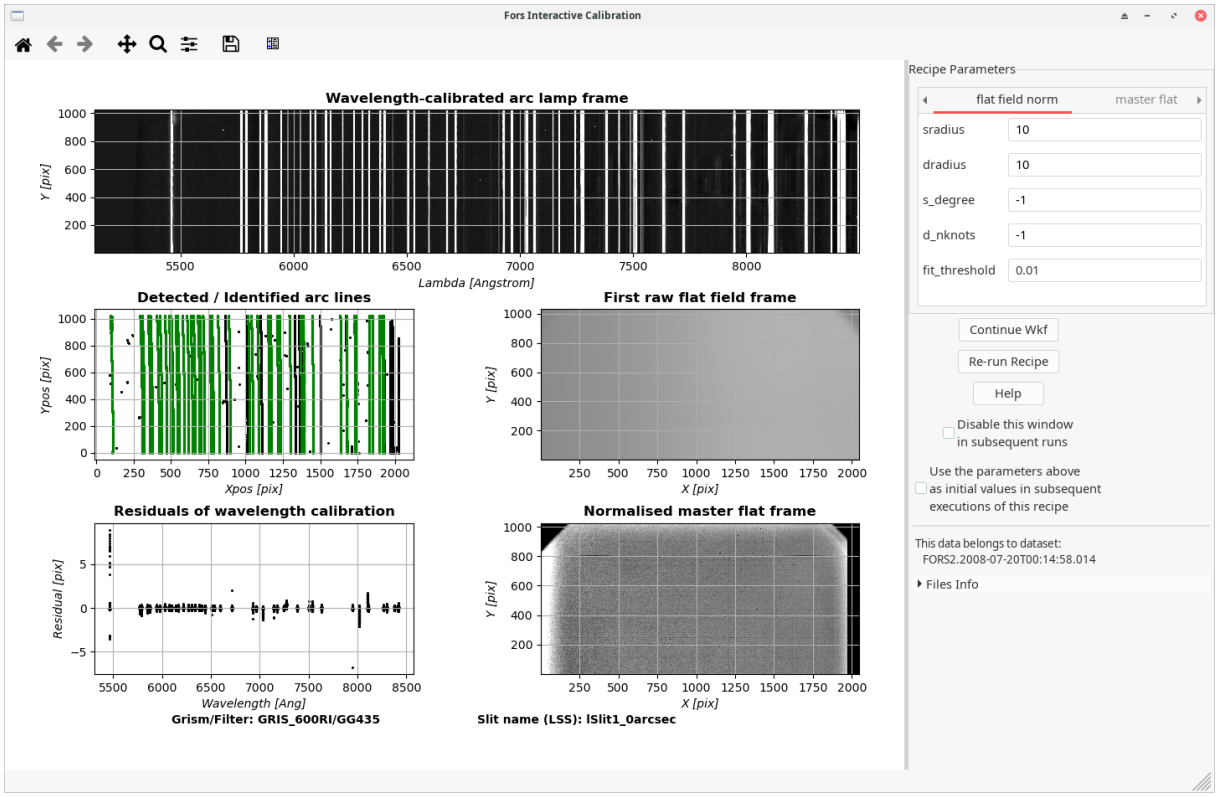


Figure 6.4: *The interactive window of the* `fors_calib` *actor for the first demo DataSet. First the calibration data of the flux standard star are processed.*

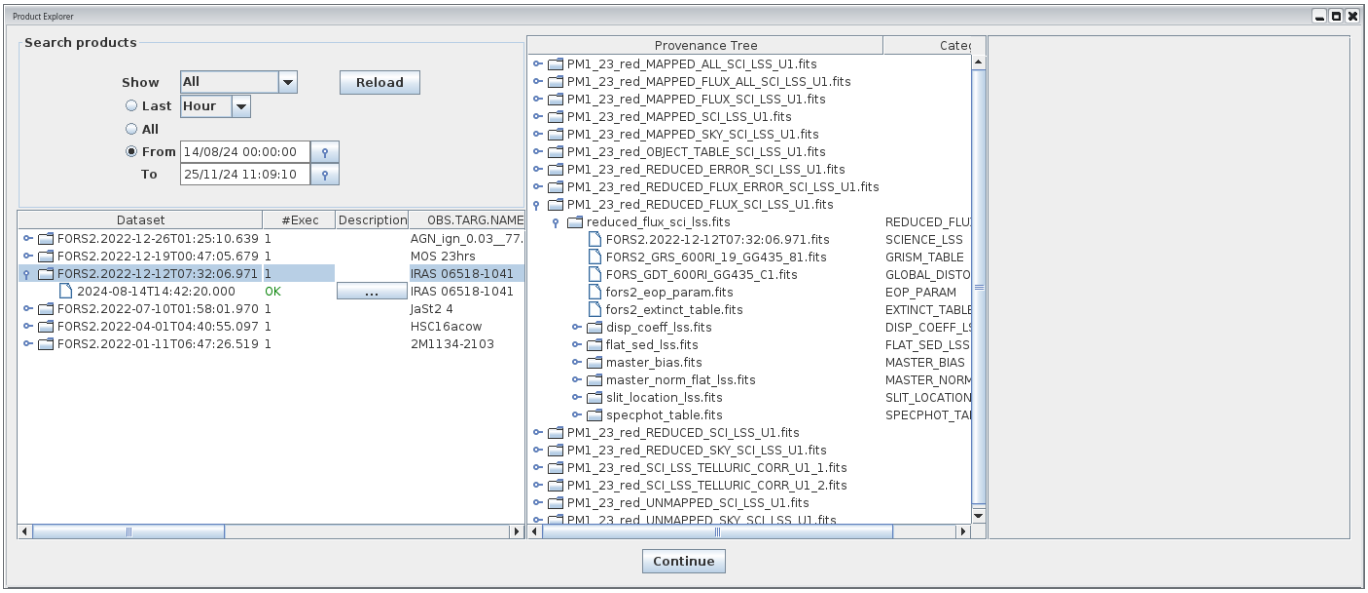


Figure 6.5: *The Product Explorer shows all datasets reduced in previous executions together with the full reduction chain for all the pipeline products.*

7 About the main `esoreflex` canvas

7.1 Saving And Loading Workflows

In the course of your data reductions, it is likely that you will customise the workflow for various data sets, even if this simply consists of editing the `ROOT_DATA_DIR` to a different value for each data set. Whenever you modify a workflow in any way, you have the option of saving the modified version to an `XML` file using `File -> Export As` (which will also open a new workflow canvas corresponding to the saved file). The saved workflow may be opened in subsequent `esoreflex` sessions using `File -> Open`. Saving the workflow in the default Kepler format (.kar) is only advised if you do not plan to use the workflow with another computer.

7.2 Buttons

At the top of the `esoreflex` canvas are a set of buttons which have the following functions:

-  - Zoom in.
-  - Reset the zoom to 100%.
-  - Zoom the workflow to fit the current window size (Recommended).
-  - Zoom out.
-  - Run (or resume) the workflow.
-  - Pause the workflow execution.
-  - Stop the workflow execution.

The remainder of the buttons (not shown here) are not relevant to the workflow execution.

7.3 Workflow States

A workflow may only be in one of three states: executing, paused, or stopped. These states are indicated by the yellow highlighting of the  ,  , and  buttons, respectively. A workflow is executed by clicking the  button. Subsequently the workflow and any running pipeline recipe may be stopped immediately by clicking the  button, or the workflow may be paused by clicking the  button which will allow the current actor/recipe to finish execution before the workflow is actually paused. After pausing, the workflow may be resumed by clicking the  button again.

8 The FORS2 Workflow

The FORS2 workflow canvas is organised into a number of areas. From top-left to top-right you will find general workflow instructions, directory parameters, and global parameters. In the middle row you will find five boxes describing the workflow general processing steps in order from left to right, and below this the workflow actors themselves are organised following the workflow general steps.

8.1 Workflow Canvas Parameters

The workflow canvas displays a number of parameters that may be set by the user. Under "Setup Directories" the user is only required to set the `RAW_DATA_DIR` to the working directory for the dataset(s) to be reduced, which, by default, is set to the directory containing the demo data. The `RAW_DATA_DIR` is recursively scanned by the `Data Organiser` actor for input raw data. The directory `CALIB_DATA_DIR`, which is by default within the pipeline installation directory, is also scanned by the `Data Organiser` actor to find any static calibrations that may be missing in your dataset(s). If required, the user may edit the directories `BOOKKEEPING_DIR`, `LOGS_DIR`, `TMP_PRODUCTS_DIR`, and `END_PRODUCTS_DIR`, which correspond to the directories where book-keeping files, logs, temporary products and end products are stored, respectively (see the Reflex User Manual for further details; Forchì (2012)).

There is a mode of the `Data Organiser` that skips the built-in data organisation and uses instead the data organisation provided by the CalSelector tool. To use this mode, click on `Use CalSelector associations` in the `Data Organiser` properties and make sure that the input data directory contains the XML file downloaded with the CalSelector archive request (note that this does not work for all instrument workflows).

Under the "Global Parameters" area of the workflow canvas, the user may set the `FITS_VIEWER` parameter to the command used for running his/her favourite application for inspecting FITS files. Currently this is set by default to `fv`, but other applications, such as `ds9`, `skycat` and `gaia` for example, may be useful for inspecting image data. Note that it is recommended to specify the full path to the visualization application (an alias will not work).

By default the `EraseDirs` parameter is set to `false`, which means that no directories are cleaned before executing the workflow, and the recipe actors will work in Lazy Mode (see Section 8.2.4), reusing the previous pipeline recipe outputs if input files and parameters are the same as for the previous execution, which saves considerable processing time. Sometimes it is desirable to set the `EraseDirs` parameter to `true`, which forces the workflow to recursively delete the contents of the directories specified by `BOOKKEEPING_DIR`, `LOGS_DIR`, and `TMP_PRODUCTS_DIR`. This is useful for keeping disk space usage to a minimum and will force the workflow to fully re-reduce the data each time the workflow is run.

The parameter `RecipeFailureMode` controls the behaviour in case that a recipe fails. If set to `Continue`, the workflow will trigger the next recipes as usual, but without the output of the failing recipe, which in most of the cases will lead to further failures of other recipes without the user actually being aware of it. This mode might be useful for unattended processing of large number of datasets. If set to `Ask`, a pop-up window will ask whether the workflow should stop or continue. This is the default. Alternatively, the `Stop` mode will stop the workflow execution immediately.

The parameter `GlobalPlotInteractivity` controls whether the interactive windows will appear for those windows which are *enabled* by default. The possible values are `true, false`. Take into account that some

windows are disabled in the default configuration and therefore are not affected by this parameter.

The parameter `ProductExplorerMode` controls whether the `ProductExplorer` actor will show its window or not. The possible values are `Enabled`, `Triggered`, and `Disabled`. `Enabled` opens the ProductExplorer GUI at the end of the reduction of each individual dataset. `Triggered` (default and recommended) opens the ProductExplorer GUI when all the selected datasets have been reduced. `Disabled` does not display the ProductExplorer GUI.

Under "Telluric Correction" the user can first decide if a telluric correction should be applied by setting the parameter `Apply_telluric_correction` (possible values are `true`, `false`) accordingly. Next one can request a minimum signal-to-noise ratio (SNR) for the modelling of the telluric absorption via the parameter `min_snr_to_model`. If no spectrum with an SNR above this threshold is available in a dataset no telluric correction will be performed. Additionally one can set a threshold for the SNR below which no telluric correction is applied (`min_snr_to_correct`). This value should never be higher than the one specified in `min_snr_to_model`.

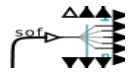
8.2 Workflow Actors

8.2.1 Simple Actors

Simple actors have workflow symbols that consist of a single (rather than multiple) green-blue rectangle. They may also have an icon within the rectangle to aid in their identification. The following actors are simple actors:

- 

- The `DataOrganiser` actor.
- 

- The `DataSetChooser` actor (inside a composite actor).
- 

- The `FitsRouter` actor Redirects files according to their categories.
- 

- The `ProductRenamer` actor.
- 

- The `ProductExplorer` actor (inside a composite actor).

Access to the parameters for a simple actor is achieved by right-clicking on the actor and selecting `Configure Actor`. This will open an "Edit parameters" window. Note that the `Product Renamer` actor is a jython script (Java implementation of the Python interpreter) meant to be customised by the user (by double-clicking on it).

8.2.2 Composite Actors

Composite Actors have workflow symbols that consist of multiple-layered green-blue rectangles. They generally do not have a logo within the rectangle. A Composite Actor represents a combination of more Simple or Composite Actors which hides over-complexity from the user in the top-level workflow. In the FORS2 workflow, the following actors are composite actors:

- 

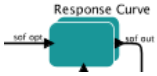
- The `Initialise` actor.
- 

- The `Data Set Selection` actor.
- 

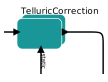
- The `Product Explorer` actor (contains the simple actor).
- 

- The `Initialise Current DataSet` actor.
- 

- The `MasterBias` actor.
- 

- The `ForsCalib` actor.
- 

- The `ResponseCurve` actor.
- 

- The `ForsScience` actor.
- 

- The `TelluricCorrection` actor.
- 

- The `Close DataSet` actor.

Composite Actors may also be expanded for inspection. To do this, right-click on the actor and select `Open Actor`, which will expand the Composite Actor components in a new `Reflex` canvas window. If the Composite Actor corresponds to a pipeline recipe, then the corresponding `RecipeExecuter` actor will be present as a Simple Actor, and its parameters are accessible as for any other Simple Actor. Alternatively you may still find Composite Actors, on which you need to repeat the first step to access the `Recipe Executer`.

8.2.3 Recipe Execution within Composite Actors

The FORS2 workflow contains Composite Actors to run pipeline recipes. This is in the most simple case due to the `SoF Splitter`/`SoF Accumulator`[7], which allow to process calibration data from different settings within one given DataSet (e.g. lamp frames taken with different slits/masks). More complex Composite Actors contain several actors (e.g. `Recipe Executer`).

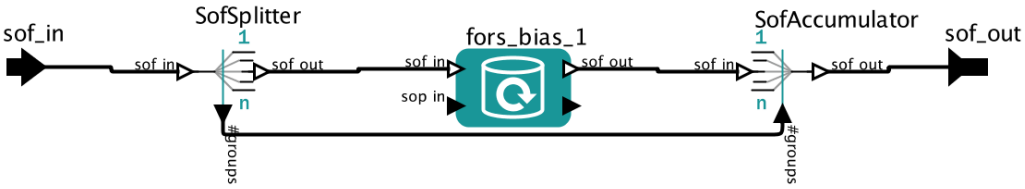


Figure 8.1: *This is the window you get when you choose* `Open Actor` *for the Composite Actor* `MasterBias`. *This is the most simple case for a Composite Actor. Using* `Configure Actor` *on* `fors_bias_1` *gives you Fig. 8.2.*

The central elements of any Reflex workflow are the `RecipeExecuter` actors that actually run the recipes. One basic way to embed a `RecipeExecuter` in a workflow is shown in Fig 8.1, which is the most simple version of a Composite Actor. The `RecipeExecuter` is preceded by an `SofSplitter`, and followed by an `SofAccumulator`. The function of the `SofSplitter` is to investigate the incoming SoFs, sort them by "purpose", and create separate SoFs for each purpose. The `RecipeExecuter` then processes each of the SoFs independently (unless they are actually the same files). Finally, the `SofAccumulator` packs all the results into a single output SoF. The direct relation between the `SofSplitter` and `SofAccumulator` is used to communicate the number of different SoFs created by the `SofSplitter`. A workflow will only work as intended if the purpose of all the files a recipe needs as input is identical. The only exception to this rule is that a purpose can also be "default". In this case, the file is included in any output SoF created by the `SoFsplitter` and `SofAccumulator`.

The reason for this scheme is best explained by an example. For a complex DataSet, the `Data Organiser` might have selected a large number of individual raw lamp frames (arc and flat field). The different lamp frames are to be used to calibrate different frames, e.g. the science frames and the standard star frames. The `Data Organiser` determines and records this "purpose" of each lamp frame, and this information is included in the DataSet and each SoF created from this DataSet. The `FitsRouter` directs all raw lamp frames to the `ForsCalib` Composite Actor. The `SofSplitter` then creates SoFs, one for the lamp frames to be used for the science frames, and (probably) separate ones for the lamp frames to be used for the standard star observations. The `fors_calib` recipe creates one master flat field (and other products) for each SoF, and the `SofAccumulator` then creates a SoF that contains all the products.

A `RecipeExecuter` actor is used in the workflow to run a single FORS2 pipeline recipe (e.g: in the `MasterBias` actor the recipe `fors_bias` is executed). In order to configure the `RecipeExecuters`, one has to first use `Open Actor` to get to the level of the recipe executors (see Fig. 8.1).

[7] SoF stands for Set of Files, which is an ASCII file containing the name (and path) of each input file and its category (e.g. `BIAS`).

Table 8.1: The FORS2 pipeline actors and their contents

actor	recipes	description
MasterBias	`fors_bias`	create master bias
ForsCalib	`fors_calib`	create master flat, determine coefficients for wavelength calibration and correction of spatial distortion
ResponseCurve	`fors_science`	determine response function (for stars with low-resolution reference data, mostly observed until 2020)
	`fors_response`	determine response function (for stars with high resolution model spectra as reference data)
ForsScience	`fors_science`	reduce science data

TelluricCorrection	`fors_molecfit_model`
	`fors_molecfit_calctrans`
	`fors_molecfit_correct`
		model and correct telluric correction in science and standard star spectra

In Figure 8.2 we show the "Edit parameters" window for a typical `RecipeExecuter` actor, which can be displayed by right-clicking on the actor and selecting `Configure Actor`. In the following we describe in more detail the function of some of the parameters for a `RecipeExecuter` actor:

- The "recipe" parameter states the FORS2 pipeline recipe which will be executed.
- The "mode" parameter has a pull-down menu allowing the user to specify the execution mode of the actor. The available options are:
  - `Run`: The pipeline recipe will be executed, possibly in Lazy mode (see Section 8.2.4). This option is the default option.
  - `Skip`: The pipeline recipe is not executed, and the actor inputs are passed to the actor outputs.
  - `Disabled`: The pipeline recipe is not executed, and the actor inputs are not passed to the actor outputs.
- The "Lazy Mode" parameter has a tick-box (selected by default) which indicates whether the `RecipeExecuter` actor will run in Lazy mode or not. A full description of Lazy mode is provided in Sect. 8.2.4.
- The "Recipe Failure Mode" parameter has a pull-down menu allowing the user to specify the behaviour of the actor if the pipeline recipe fails. The available options are:
  - `Stop`: The actor issues an error message and the workflow stops.
  - `Continue`: The actor creates an empty output and the workflow continues.
  - `Ask`: The actor displays a pop-up window and asks the user whether he/she wants to continue or stop the workflow. This option is the default option.
- The set of parameters which start with "recipe param" and end with a number or a string correspond to the parameters of the relevant FORS2 pipeline recipe. By default in the `RecipeExecuter` actor, the pipeline recipe parameters are set to their pipeline default values. If you need to change the default

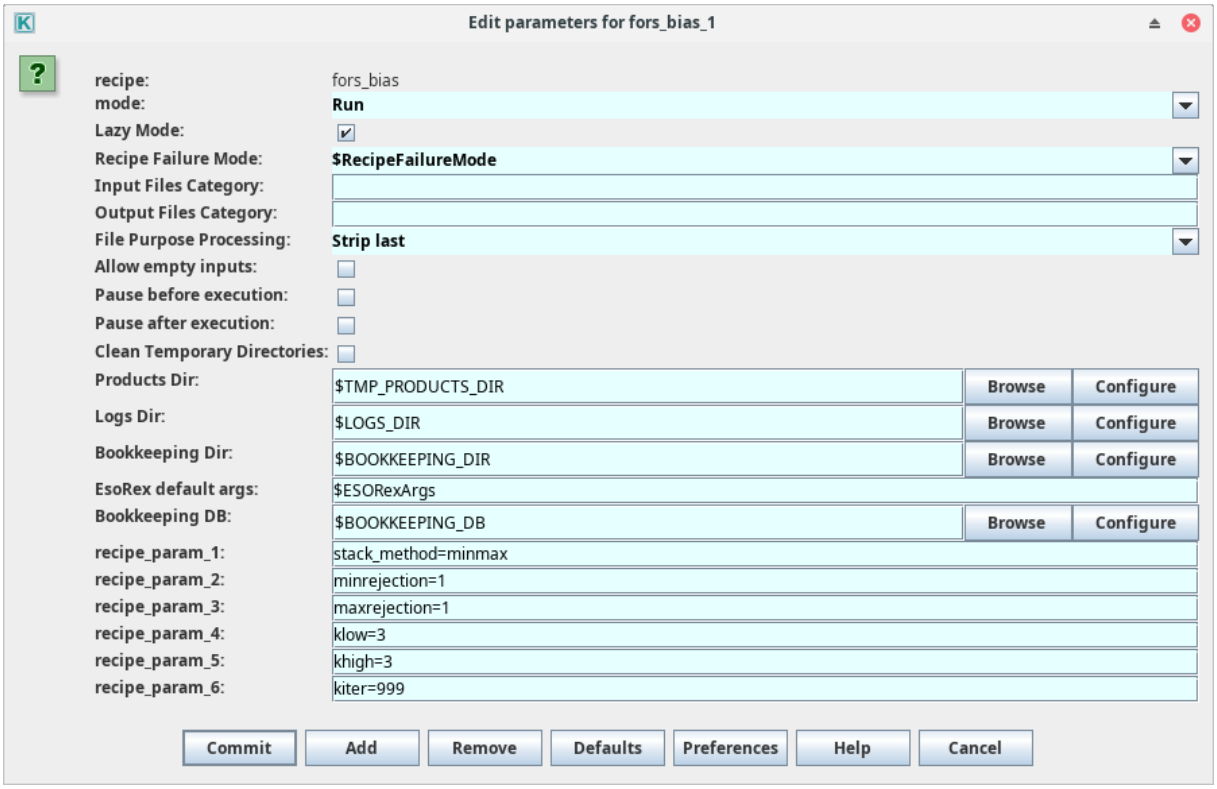


Figure 8.2: *The "Edit parameters" window for a typical* `RecipeExecuter` *actor, the* `fors_bias_1` *actor which runs the* `fors_bias` *pipeline recipe.*

parameter value for any pipeline recipe, then this is where you should edit the value[8]. A special case are parameters that can be changed in the interactive windows (see Sect. 9.4). For more information on the FORS2 pipeline recipe parameters, the user should refer to the FORS Pipeline User Manual[9]).

The description of the remainder of the `RecipeExecuter` actor parameters are outside the scope of this tutorial, and the interested user is referred to the Reflex User Manual for further details (Forchì 2012). Any changes that you make in the "Edit parameters" window may be saved in the workflow by clicking the `Commit` button when you have finished.

8.2.4 Lazy Mode

By default, all `RecipeExecuter` actors in a pipeline workflow are "Lazy Mode" enabled. This means that when the workflow attempts to execute such an actor, the actor will check whether the relevant pipeline recipe has already been executed with the same input files and with the same recipe parameters. If this is the case, then the actor will not execute the pipeline recipe, and instead it will simply broadcast the previously generated

[8] Some of the pipeline parameters are read from the `GRISM_TABLE`s, which contain grism-specific parameters. These cannot be changed here.

[9] Available at *https://ftp.eso.org/pub/dfs/pipelines/fors/fors-pipeline-manual-5.18.pdf*

products to the output port. The purpose of the Lazy Mode is therefore to minimise any reprocessing of data by avoiding data re-reduction where it is not necessary.

One should note that the actor's Lazy Mode depends on the contents of the directory specified by the parameter `BOOKKEEPING_DIR` and the relevant FITS file checksums. Any modification to the directory contents and/or the file checksums will cause the corresponding actor to run the pipeline recipe again when executed, thereby re-reducing the input data.

The re-reduction of data at each execution may sometimes be desirable. To force a re-reduction of data for any single `RecipeExecuter` actor in the workflow, right-click the actor, select `Configure Actor`, and uncheck the Lazy mode parameter tick-box in the "Edit parameters" window that is displayed. For many workflows the `RecipeExecuter` actors are actually found inside the composite actors in the top level workflow. To access such embedded `RecipeExecuter` actors you will first need to open the sub-workflow by right-clicking on the composite actor and then selecting `Open Actor`.

To force the re-reduction of all data in a workflow (i.e. to disable Lazy mode for the whole workflow), you must uncheck the Lazy mode for every single `RecipeExecuter` actor in the entire workflow. It is also possible to change the name of the bookkeeping directory, instead of modifying any of the Lazy mode parameters. This will also force a re-reduction of the given dataset(s). A new reduction will start (with the lazy mode still enabled), but the results of previous reduction will not be reused. Alternatively, if there is no need to keep any of the previously reduced data, one can simply set the `EraseDirs` parameter under the "Global Parameters" area of the workflow canvas to `true`. This will then remove all previous results that are stored in the bookkeeping, temporary, and log directories before processing the input data, in effect, starting a new clean data reduction and re-processing every input dataset. *Note: The option* `EraseDirs = true` *does not work in* `esoreflex` *version 2.9.x and makes the workflow to crash.*

9 Reducing your own data

In this section we describe how to reduce your own data set.

First, we suggest the reader to familiarize with the workflow by reducing the demo dataset first (Section 6), but it is not a requirement.

9.1 The `esoreflex` command

We list here some options associated to the `esoreflex` command. We recommend to try them to familiarize with the system. In the following, we assume the `esoreflex` executable is in your path; if not you have to provide the full path `<install_dir>/bin/esoreflex`

To see the available options of the `esoreflex` command type:

```
esoreflex -h
```

The output is the following.

```
-h | -help              print this help message and exit.
-v | -version           show installed Reflex version and pipelines and exit.
-l | -list-workflows    list available installed workflows and from
                        ~/KeplerData/workflows.
-n | -non-interactive   enable non-interactive features.
-e | -explore           run only the Product Explorer in this workflow
-p <workflow> | -list-parameters <workflow>
                        lists the available parameters for the given
                        workflow.
-config <file>          allows to specify a custom esoreflex.rc configuration
                        file.
-create-config <file>   if <file> is TRUE then a new configuration file is
                        created in ~/.esoreflex/esoreflex.rc. Alternatively
                        a configuration file name can be given to write to.
                        Any existing file is backed up to a file with a '.bak'
                        extension, or '.bakN' where N is an integer.
-debug                  prints the environment and actual Reflex launch
                        command used.
```

9.2 Launching the workflow

We list here the recommended way to reduce your own datasets. Steps 1 and 2 are optional and one can start from step 3.

1. Type: `esoreflex -n <parameters>` FORS2 Spectroscopy to launch the workflow non interactively and reduce all the datasets with default parameters.

<parameters> allows you to specify the workflow parameters, such as the location of your raw data and the final destination of the products.

For example, type (in a single command line):

```
esoreflex -n
  -RAW_DATA_DIR /home/user/my_raw_data
  -ROOT_DATA_DIR /home/user/my_reduction
  -END_PRODUCTS_DIR $ROOT_DATA_DIR/reflex_end_products
  fors_spec
```

to reduce the complete datasets that are present in the directory /home/user/my_raw_data and that were not reduced before. Final products will be saved in /home/user/my_reduction/reflex_end_products, while book keeping, temporary products, and logs will be saved in sub-directories of /home/user/my_reduction/. If the reduction of a dataset fails, the reduction continues to the next dataset. It can take some time, depending on the number of datasets present in the input directory. For a full list of workflow parameters type esoreflex -p FORS2 Spectroscopy. Note that this command lists only the parameters, but does not launch the workflow.

Once the reduction is completed, one can proceed with optimizing the results with the next steps.

2. Type:

   esoreflex -e fors_spec

   to launch the Product Explorer. The Product Explorer allows you to inspect the data products already reduced by the FORS2 Spectroscopy esoreflex workflow. Only products associated with the workflow default bookkeeping database are shown. To visualize products associated to given bookkeeping database, pass the full path via the BOOKKEEPING_DB parameter:

   esoreflex -e BOOKKEEPING_DB <database_path> fors_spec

   to point the product explorer to a given <database_path>, e.g., /home/username/reflex/reflex_bookkeeping/test.db

   The Product Explorer allows you to inspect the products while the reduction is running. Press the button Refresh to update the content of the Product Explorer. This step can be launched in parallel to step 1.

   A full description of the Product Explorer will be given in Section 9.3.3

3. Type:

   esoreflex fors_spec &

   to launch the FORS2 Spectroscopy esoreflex workflow. The FORS2 Spectroscopy workflow window will appear (Fig. 6.2). Please configure the set-up directories ROOT_DATA_DIR, RAW_DATA_DIR, and other workflow parameters as needed. Just double-click on them, edit the content, and press OK. Remember to specify the same <database_path> as for the Product Explorer, if it has been opened at step #2, to synchronize the two processes.

4. (Recommended, but not mandatory) On the main esoreflex menu set Tools –> Animate at Runtime to 1 in order to highlight in red active actors during execution.

5. Press the button  to start the workflow. First, the workflow will highlight and execute the Initialise actor, which among other things will clear any previous reductions if required by the user (see Section 8.1).

Secondly, if set, the workflow will open the Product Explorer, allowing the user to inspect previously reduced datasets (see Section 9.3.3 for how to configure this option).

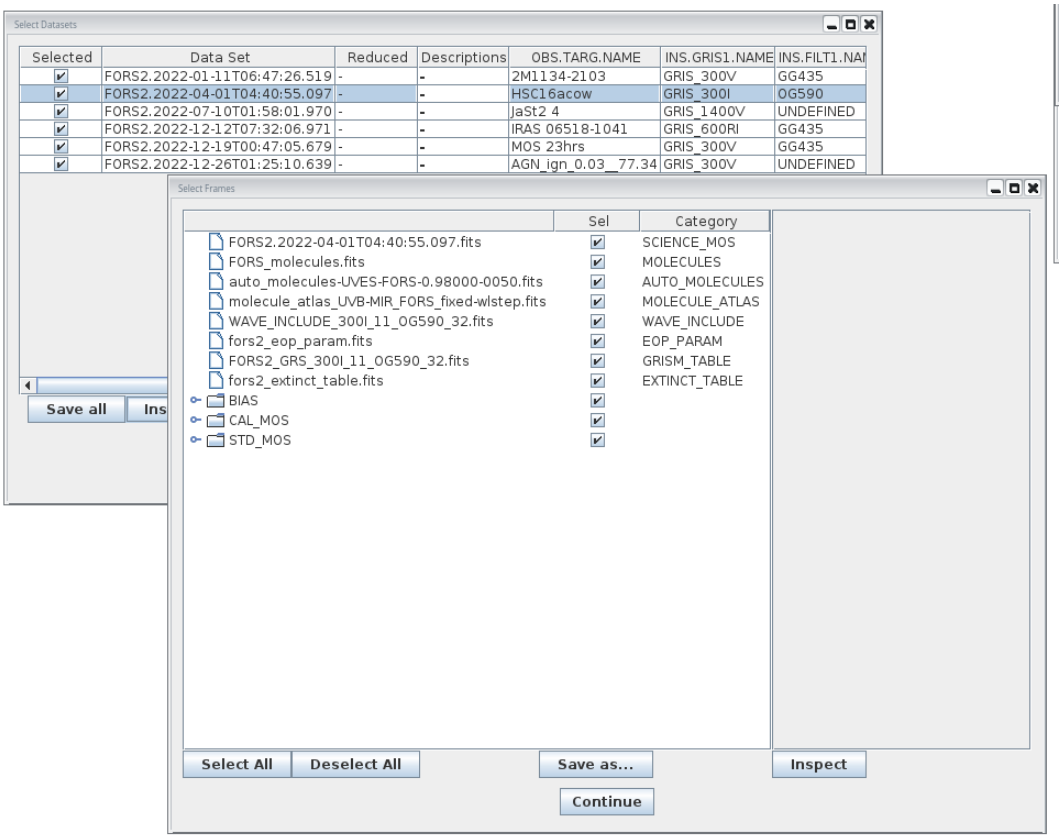


Figure 9.1: *The "Select Frames" window with a single file from the current Data Set highlighted in blue, and the corresponding FITS header displayed in the text box on the right. Hidden partially behind the "Select Frames" window is the "Select DataSets" window with the currently selected DataSet highlighted in blue.*

9.3 Workflow Steps

9.3.1 Data Organisation And Selection

The `DataOrganiser` (DO) is the first crucial component of a Reflex workflow. The DO takes as input `RAW_DATA_DIR` and `CALIB_DATA_DIR` and it detects, classifies, and organises the files in these directories and any subdirectories. The output of the DO is a list of "DataSets". A DataSet is a special Set of Files (SoF). A DataSet contains one or several science (or calibration) files that should be processed together, and all files needed to process these data. This includes any calibration files, and in turn files that are needed to process these calibrations. Note that different DataSets might overlap, i.e. some files might be included in more than one DataSet (e.g., common calibration files).

A DataSet lists three different pieces of information for each of its files, namely 1) the file name (including the path), 2) the file category, and 3) a string that is called the "purpose" of the file. The DO uses the OCA[10] rules to find the files to include in a DataSet, as well as their categories and purposes. The file category identifies different types of files, and it is derived by information in the header of the file itself. A category could for example be `RAW_CALIBRATION_1`, `RAW_CALIBRATION_2` or `RAW_SCIENCE`, depending on the instrument. The purpose of a file identifies the reason why a file is included in a DataSet. The syntax is `action_1/action_2/action_3/ ... /action_n`, where each `action_i` describes an intended processing step for this file (for example, creation of a `MASTER_CALIBRATION_1` or a `MASTER_CALIBRATION_2`). The actions are defined in the OCA rules and contain the recipe together with all file categories required to execute it (and predicted products in case of calibration data). For example, a workflow might include two actions `action_1` and `action_2`. The former creates `MASTER_ CALIBRATION_1` from `RAW_CALIBRATION_1`, and the later creates a `MASTER_CALIBRATION_2` from `RAW_CALIBRATION_2`. The `action_2` action needs `RAW_CALIBRATION_2` frames and the `MASTER_ CALIBRATION_1` as input. In this case, these `RAW_CALIBRATION_1` files will have the purpose `action_ 1/action_2`. The same DataSet might also include `RAW_CALIBRATION_1` with a different purpose; irrespective of their purpose the file category for all these biases will be `RAW_CALIBRATION_1`.

The Datasets created via the `DataOrganiser` will be displayed in the `DataSet Chooser`. Here the users have the possibility to inspect the various datasets and decide which one to reduce. By default, DataSets that have not been reduced before are highlighted for reduction. Click either `Continue` in order to continue with the workflow reduction, or `Stop` in order to stop the workflow. A full description of the `DataSet Chooser` is presented in Section 9.3.2.

Once the `Continue` is pressed, the workflow starts to reduce the first selected DataSet. Files are broadcasted according to their purpose to the relevant actors for processing.

The categories and purposes of raw files are set by the DO, whereas the categories and purpose of products generated by recipes are set by the `RecipeExecuter`. The file categories are used by the `FitsRouter` to send files to particular processing steps or branches of the workflow (see below). The purpose is used by the `SofSplitter` and `SofAccumulator` to generate input SoFs for the `RecipeExecuter`. The `SofSplitter` and `SofAccumulator` accept several SoFs as simultaneous input. The `SofAccumulator` creates a single output SoF from the inputs, whereas the `SofSplitter` creates a separate output SoF for each purpose.

9.3.2 DataSetChooser

The `DataSetChooser` displays the DataSets available in the "Select Data Sets" window, activating vertical and horizontal scroll bars if necessary (Fig. 6.3).

Some properties of the DataSets are displayed: the name, the number of files, a flag indicating if it has been

[10] OCA stands for OrganisationClassificationAssociation and refers to rules, which allow to classify the raw data according to the contents of the header keywords, organise them in appropriate groups for processing, and associate the required calibration data for processing. They can be found in the directory `<install_dir>/share/esopipes/<pipeline-version>/reflex/`, carrying the extension `.oca`. The variable `<install_dir>` depends on the operative system and installation procedure. For installation through `rpm`: `<install_dir>=/usr`; for installation through macport `<install_dir>=/opt/local`; for installation through the installation script install_esoreflex it depends on the path specified during installation, e.g. `<install_dir>=<specified_path>/install`

successfully reduced (a green OK), if the reduction attempts have failed or were aborted (a red FAILED), or if it is a new dataset (a black "-"). The column "Descriptions" lists user-provided descriptions (see below), other columns indicate the instrument set-up and a link to the night log.

Sometimes you will want to reduce a subset of these DataSets rather than all DataSets, and for this you may individually select (or de-select) DataSets for processing using the tick boxes in the first column, and the buttons `Deselect All` and `Select Complete` at the bottom, or configure the "Filter" field at the bottom left. Available filter options are: "New" (datasets not previously reduced will be selected), "Reduced" (datasets previously reduced will be selected), "All" (all datasets will be selected), and "Failed" (dataset with a failed or aborted reduction will be selected).

You may also highlight a single DataSet in blue by clicking on the relevant line. If you subsequently click on `Inspect Highlighted`, then a "Select Frames" window will appear that lists the set of files that make up the highlighted DataSet including the full filename[11], the file category (derived from the FITS header), and a selection tick box in the right column. The tick boxes allow you to edit the set of files in the DataSet which is useful if it is known that a certain calibration frame is of poor quality (e.g: a poor raw flat-field frame). The list of files in the DataSet may also be saved to disk as an `ASCII` file by clicking on `Save As` and using the file browser that appears.

By clicking on the line corresponding to a particular file in the "Select Frames" window, the file will be highlighted in blue, and the file FITS header will be displayed in the text box on the right, allowing a quick inspection of useful header keywords. If you then click on `Inspect`, the workflow will open the file in the selected FITS viewer application defined by the workflow parameter `FITS_VIEWER`.

To exit from the "Select Frames" window, click `Continue`.

To add a description of the reduction, press the button `...` associated with the field "Add description to the current execution of the workflow" at the bottom right of the Select Dataset Window; a pop up window will appear. Enter the desired description (e.g. "My first reduction attempt") and then press `OK`. In this way, all the datasets reduced in this execution, will be flagged with the input description. Description flags can be visualized in the `SelectFrames` window and in the `ProductExplorer`, and they can be used to identify different reduction strategies.

To exit from the "Select DataSets" window, click either `Continue` in order to continue with the workflow reduction, or `Stop` in order to stop the workflow.

9.3.3 The ProductExplorer

The ProductExplorer is an interactive component in the `esoreflex` workflow whose main purpose is to list the final products with the associated reduction tree for each dataset and for each reduction attempt (see Fig. 6.5).

Configuring the ProductExplorer

You can configure the ProductExplorer GUI to appear after or before the data reduction. In the latter case you can inspect products as reduction goes on.

1. To display the ProductExplorer GUI at the end of the datareduction:

[11] keep the mouse pointer on the file name to visualize the full path name.

- Click on the global parameter “ProductExplorerMode” before starting the data reduction. A configuration window will appear allowing you to set the execution mode of the Product Explorer. Valid options are:
  - "Triggered" (default). This option opens the ProductExplorer GUI when all the selected datasets have been reduced.
  - "Enabled". This option opens the ProductExplorer GUI at the end of the reduction of each individual dataset.
  - “Disable”. This option does not display the ProductExplorer GUI.
- Press the  button to start the workflow.

2. To display the ProductExplorer GUI “before” starting the data reduction:

- double click on the composite Actor "Inspect previously reduced data". A configuration window will appear. Set to "Yes" the field "Inspect previously reduced data (Yes/No)". Modify the field "Continue reduction after having inspected the previously reduced data? (Continue/Stop/Ask)". "Continue" will continue the workflow and trigger the DataOrganizer. "Stop" will stop the workflow; "Ask" will prompt another window deferring the decision whether continuing or not the reduction after having closed the Product Explorer.
- Press the  button to start the workflow. Now the ProductExplorer GUI will appear before starting the data organization and reduction.

Exploring the data reduction products

The left window of the ProductExplorer GUI shows the executions for all the datasets (see Fig. 6.5). Once you click on a dataset, you get the list of reduction attemps. Green and red flags identify successfull or unsucessfull reductions. Each reduction is linked to the “Description” tag assigned in the “Select Dataset” window.

1. To identify the desired reduction run via the “Description” tag, proceed as follows:

- Click on the symbol at the left of the dataset name. The full list of reduction attempts for that dataset will be listed. The column Exec indicates if the reduction was succesful (green flag: "OK") or not (red flag: "Failed").
- Click on the entries in the field "Description" to visualize the description you have entered associated to that dataset on the Select Dataset window when reducing the data.
- Identify the desired reduction run. All the products are listed in the central window, and they are organized following the data reduction cascade.

You can narrow down the range of datasets to search by configuring the field "Show" at the top-left side of the ProductExplorer (options are: "All", "Successful", "Unsuccessful"), and specifying the time range (Last, all, From-to).

2. To inspect the desired file, proceed as follows:

- Navigate through the data reduction cascade in the ProductExplorer by clicking on the files.
- Select the file to be inspected and click with the mouse right-hand button. The available options are:
  - Options available always:
    - Copy full path. It copies the full name of the file onto the clipboard. Shift+Ctr+v to past it into a terminal.
    - Inspect Generic. It opens the file with the fits viewer selected in the main workflow canvas.
    - Inspect with. It opens the file with an executable that can be specified (you have to provide the full path to the executable).
  - Options available for files in the `TMP_PRODUCTS_DIR` directory only:
    - command line. Copy of the environment configuration and recipe call used to generate that file.
    - Xterm. It opens an Xterm at the directory containing the file.
  - Options available for products associated to interactive windows only:
    - Display pipeline results. It opens the interactive windows associated to the recipe call that generated the file. Note that this is for visualization purposes only; the recipe parameters cannot be changed and the recipe cannot be re-run from this window.

9.3.4 Creation Of Master Calibration Files

In this step of the workflow, the following FORS2 recipes are executed in the order listed below. Please refer to the FORS Pipeline User Manual (Sections 9 and 10) for the details of each recipe and the algorithms employed:

1. The `MasterBias` actor will execute the FORS2 pipeline recipe `fors_bias` in order to create a combined master bias frame from the set of raw bias frames
2. The `ForsCalib` actor will execute the FORS2 pipeline recipe `fors_calib` in order to create from the set of raw flat and arc frames a combined master flat frame as well as coefficients for wavelength calibration and correction of spatial distortions. Sect. 10.2 provides additional information on this step and its interactive window is described in Sect. 9.4.1 (p. 43).

9.3.5 Response Computation

The `ResponseCurve` actor will create an instrument response curve from the observation of a standard star (if its reference data are found in the static calibrations), which will subsequently be used to flux-calibrate the science observation. Two recipes are available to perform this task:

`fors_science` This recipe integrates the observed flux, corrected for gain, exposure time and atmospheric extinction, over the intervals indicated in the `STD_FLUX_TABLE` corresponding to the observed star. This table is identified by matching the name recorded in the keyword `HIERARCH ESO OBS TARG NAME`. The reference flux values for the star are then divided by the integrated observed flux values to create the so-called raw response. Features like strong stellar lines or regions of telluric absorption are masked before a polynomial or a spline fit is performed to define the final response curve. The recipe also provides the extracted 1-dimensional spectrum of the standard star in Phase3 format, which is needed for the recipe `fors_response`.

`fors_response` This recipe uses model spectra provided in the `STD_FLUX_CATALOG` as reference data. Because the reference data do not include telluric absorption observed spectra extending to wavelengths above 6800 Å are first corrected using the FORS-specific molecfit recipes `fors_molecfit_model`, `fors_molecfit_calctrans`, and `fors_molecfit_correct` (see Sect. 9.3.7 for more details). For this correction the Phase3 format provided by `fors_science` is needed.

The reference model spectrum is then aligned in wavelength to the telluric-corrected observed spectrum or – for bluer settings – the original Phase3 spectrum so that its velocity matches that of the observed spectrum and it has the same step size in wavelength. Next it is divided by the observed spectrum (corrected for gain exposure time and atmospheric extinction) to create a raw response. The median value of this raw response is then determined at predefined wavelength (read from `FIT_POINTS_CATALOG`) and interpolated with a spline fit to define the response.

Sect. 10.3 provides additional information on this step and its interactive windows are described in Sect. 9.4.2 (p. 45). Please refer to the FORS Pipeline User Manual (Sections 9 and 10) for the details of this recipe.

Note that this actor will be skipped if there are no observations of a standard star in the current DataSet. calSelector usually includes a standard star observation only if it was taken within ±7 nights of the science observation. For data observed after January 1, 2015, calSelector wil also deliver a master response curve for most grisms.

9.3.6 Science Reduction

The `ForsScience` actor will execute the FORS2 pipeline recipe `fors_science` to apply sky subtraction and extract the spectra. Sect. 10.5 provides additional information on this step and its interactive window is described in Sect. 9.4.4 (p. 50). Please refer to the FORS Pipeline User Manual (Sections 9 and 10) for the details of this recipe and the extraction algorithms employed.

The FORS2 workflow will flux-calibrate the science observation using the master response curve (preferred if available) or the instrument response curve derived from the standard star observation in the current DataSet. If neither standard star observation nor master response curve exist in the current DataSet, then the science observation will not be flux-calibrated.

9.3.7 Telluric correction

To enable the computation of telluric absorption and its correction in FORS spectroscopic data, the user must set the workflow parameter `Apply_telluric_correction` to anything but `false`. Only spectra in Phase3 format that extend beyond 6800 Å will be corrected. The telluric correction is performed using the FORS-specific Molecfit recipes `fors_molecfit_model`, `fors_molecfit_calctrans` and `fors_molecfit_correct` based on the `molecfit` algorithm. More details on Molecfit can be found in Smette et al. (2015, A&A 576, 77) and in the pipeline manual at the pipeline overview page. The interactive windows are described in Sect. 9.4.5 and 9.4.6.

For each observation, the workflow selects the 1D spectrum with the highest signal-to-noise ratio and uses it as reference spectrum to be fitted by the recipe `fors_molecfit_model`. This recipe computes the abundances of various molecules taking into account also the instrument set-up. Then, the best fit parameters obtained in `fors_molecfit_model` are used to construct the atmospheric transmission for each of the extracted spectra

in the observation. This step is performed by the recipe `fors_molecfit_calctrans`. Last, each spectrum is corrected for telluric absorption using the appropriate transmission by the recipe `fors_molecfit_correct`.

It is strongly recommended to enable the interactive features for `fors_molecfit_model` and `fors_molecfit_correct` when executing the telluric correction.

For FORS2 data, it is generally sufficient to fit O_2 and H_2O molecules. Some general tips can also be found in Section 7.2 of the `molecfit` tutorial.

Note: At the moment, if the barycentric correction in ForsScience is activated, it is not possible to perform the telluric correction. This issue will be solved in future releases. If the user desires to have both telluric corrected data and barycentric correction, then we recommend to perform the telluric correction with the FORS2 spectroscopic workflow, and correct the final spectra with the recipe `esotk_barycorr` in the toolkit ESOTK.

9.3.8 Output Organisation

After having processed the input data for a DataSet, the workflow highlights and executes the `Product Renamer` actor, which, by default, will copy the defined final products of the `ForsScience` actor to the directory specified by `END_PRODUCTS_DIR` and rename them with names derived from the values of certain FITS header keywords. Specifically, final products are renamed by default with names of the form `<HIERARCH.ESO.OBS.NAME>_<HIERARCH.ESO.PRO.CATG>.fits`, with `<HIERARCH.ESO.OBS.NAME>` and `<HIERARCH.ESO.PRO.CATG>` representing the values of the corresponding FITS header keywords (`<HIERARCH.ESO.OBS.NAME>` is the name of the OB and `<HIERARCH.ESO.PRO.CATG>` is the category of the product file). These names are fully configurable by right-clicking on the `Product Renamer` actor, selecting `Configure Actor`, and then editing the string as appropriate. In some cases the keyword `<HIERARCH.ESO.OBS.TARG.NAME>` (target name) may be more useful than `<HIERARCH.ESO.OBS.NAME>`.

For `MOS`[12] data the final products that are copied and renamed are (for better readbility we replace `<HIERARCH.ESO.OBS.NAME>` by `<OB_NAME>`):

- **1-dimensional extracted spectra** (Phase3 compliant)
  - `<OB_NAME>_REDUCED_IDP_SCI_MOS.fits` flux-calibrated spectra
  - `<OB_NAME>_SCI_MOS_TELLURIC_CORR.fits` flux-calibrated spectra, corrected for telluric absorption. The first extension is identical to the corresponding `<OB_NAME>_REDUCED_IDP_SCI_MOS` product, the second extension contains information on the telluric correction parameters, and the third extension provides the telluric corercted spectrum (column `cflux`. Please note that the wavelength in the third extension (column `lambda`) is in μm and not in Å.
- **1-dimensional extracted spectra** (image format, `<OB_NAME>_REDUCED_*`, created only if spectra are identified and can be extracted).
  The individual spectra are provided as rows in a FITS file. The correspondence between these rows and the 2-dimensional frames and/or slit identifications can be obtained from `<OB_NAME>_OBJECT_TABLE_SCI_MOS.fits`. All extracted spectra have the same format.
  - `<OB_NAME>_REDUCED_SCI_MOS.fits` spectra

[12] The product filenames for the other modes are similar but with the `MOS` suffix replaced by `MXU`, `LONG_MOS` or `LSS` as appropriate.

  - <OB_NAME>_REDUCED_ERROR_SCI_MOS.fits error of spectra
  - <OB_NAME>_REDUCED_FLUX_SCI_MOS.fits flux-calibrated spectra
  - <OB_NAME>_REDUCED_FLUX_ERROR_SCI_MOS.fits error of flux-calibrated spectra
  - <OB_NAME>_REDUCED_SKY_SCI_MOS.fits sky spectra

- **2-dimensional wavelength calibrated and distortion corrected frames** (<OB_NAME>_MAPPED_*)
  For LSS data the distortion is corrected only if an appropriate GLOABL_DISTORTION_TABLE was provided[13]

  - <OB_NAME>_MAPPED_ALL_SCI_MOS.fits 2-dimensional SCIENCE frame without sky subtraction
  - <OB_NAME>_MAPPED_SCI_MOS.fits 2-dimensional SCIENCE frame, sky-subtracted
  - <OB_NAME>_MAPPED_FLUX_ALL_SCI_MOS.fits 2-dimensional SCIENCE frame, flux-calibrated, without sky subtraction
  - <OB_NAME>_MAPPED_FLUX_SCI_MOS.fits 2-dimensional SCIENCE frame, sky-subtracted and flux-calibrated
  - <OB_NAME>_MAPPED_SKY_SCI_MOS.fits 2-dimensional frame with fitted sky background

- **2-dimensional frames, neither wavelength calibrated nor distortion corrected** (<OB_NAME>_UNMAPPED_*)

  - <OB_NAME>_UNMAPPED_SCI_MOS.fits (not for LSS) 2-dimensional SCIENCE frame, sky-subtracted, neither wavelength calibrated nor distortion corrected
  - <OB_NAME>_UNMAPPED_SKY_SCI_MOS.fits 2-dimensional frame with fitted sky background, neither wavelength calibrated nor distortion corrected

- **table with position information for detected spectra** <OB_NAME>_OBJECT_TABLE_SCI_MOS.fits

If **sky alignment** is requested (skyalign≥0) the following products are provided in addition to the ones listed above:

- <OB_NAME>_DISP_COEFF_SCI_MOS.fits dispersion coefficients after adjusting to sky line positions
- <OB_NAME>_SKY_SHIFTS_SLIT_SCI_MOS.fits shifts in wavelength derived from sky line positions (for MOS-like data). For long-slit and LSS-like MOS (LONG_MOS) data this is <OB_NAME>_SKY_SHIFTS_LONG_SCI_MOS.fits.
- <OB_NAME>_WAVELENGTH_MAP_SCI_MOS.fits 2-dimensional frame with pixel value=wavelength of pixel

All products <HIERARCH.ESO.OBS.NAME>_<type>_FLUX_* are created only if appropriate flux standard star observations for the upper chip are provided and the standard star flux table is available (<type> being REDUCED or MAPPED).

[13] This table currently does not exist for GRIS_150I (all filters), GRIS_200I (no filter), GRIS_300I (no filter), GRIS_300V (GG375, GG435), GRIS_600R (GG435), and GRIS_600I (FILT_465_250).

The following actors in this step of the workflow are concerned with the termination of the data flow for the current DataSet and will highlight briefly as they are executed.

Finally, the `Product Explorer` window will appear as shown in Fig. 6.5 with a list of datasets on the left menu. By unfolding the menu under each dataset, all the renamed products appear, and if one is interested in the files, including all intermediate steps, that are used to produce that final product, just click on it and a dependency tree will show the whole reduction chain.

9.4 Interactive Windows

The FORS2 workflow contains three interactive windows that allow the user to iterate on the processing of their data. They are described below. For troubleshooting and tips how to improve the results see Sects. 12.2 and 10, respectively.

9.4.1 fors_calib

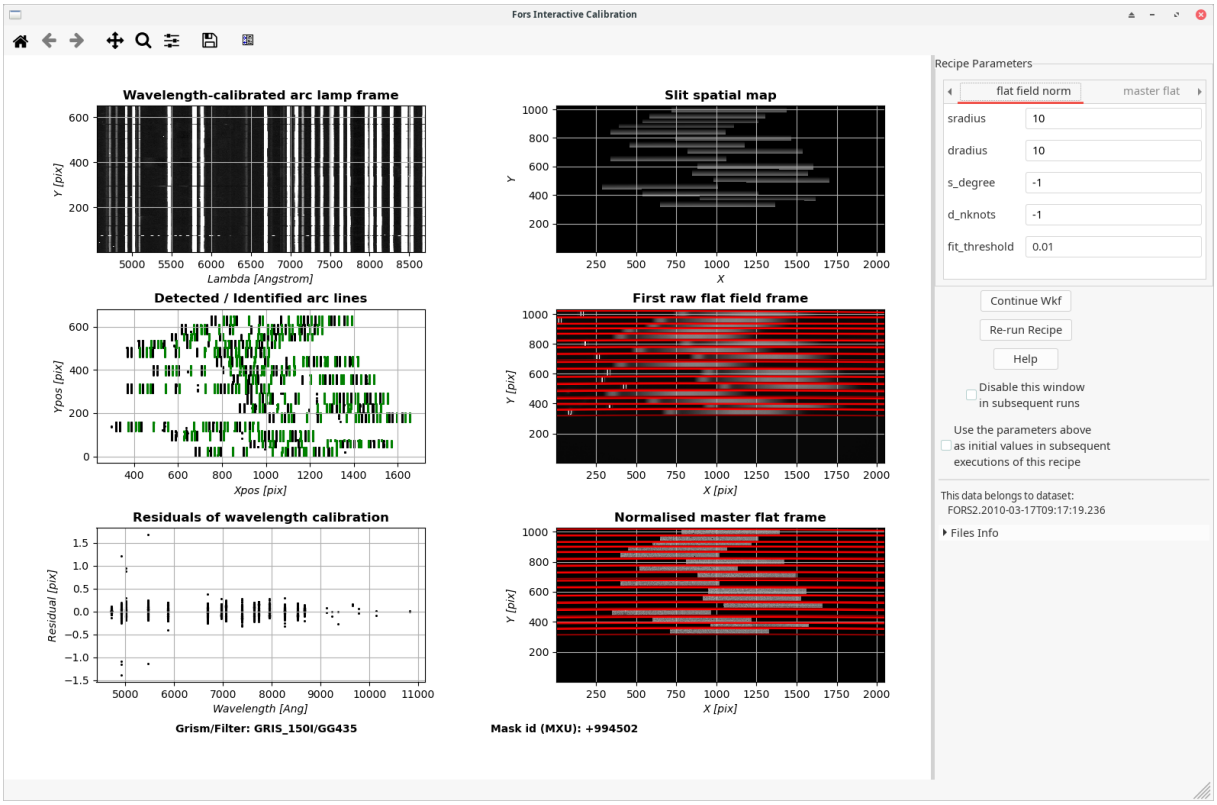


Figure 9.2: *The interactive window of the* `fors_calib` *actor for the MXU calibrations of the second demo DataSet.*.

The interactive windows shown in Figs. 6.4 (LSS, p. 22) and 9.2 (MXU, p. 43) provide information about the quality of the wavelength calibration (left column), distortion correction (top right and center plot, not applicable to LSS and LSS-like data) and the flat field combination and normalization (bottom right plot). The plots contain in detail:

Top Left *Wavelength-calibrated arc lamp frame*: In this plot the arc lamp lines should run straight from top to bottom without any empty rows between them. Particular attention should be given to lines at the blue and red ends of each spectrum, where the polynomial fit is more sensitive to small variations of the signal. Some arc lines may show gaps due to the placement of the slits, but empty rows without any lines point

towards problems with the detection of the arc lamp lines. For LSS and LSS-like data this plot covers the full top row.

Center Left *Positions of detected arc lines in rectified frame* Here the x/y positions of the detected (black) and identified (green) arc lines in the rectified frame are plotted. Dark/light green mark lines identified in the first attempt and during the search around identified lines (`wradius`>0). Again, there should be no empty rows and most of the detected lines should be identified. Unidentified lines at the edges of the frame are especially problematic, as they cause unnecessary extrapolations of the dispersion relation.

Bottom Left *Residuals between predicted and detected arc line position* The residuals should generally be below 0.5 pixel. If they show systematic varitions the polynomial degree used to fit the dispersion relation may be too low (or in rare cases too high). If the scatter appears very large one should zoom in, because there are often only a few outliers and the majority of the residuals are within ± 0.5 pixels. Middle-clicking with the mouse on a line will add the catalogue line nearest to this position to `ignore_lines`. The plot shows all the residuals, regardless on whether they are rejected or not.

Top Right *Spatial map* The spatial map has as pixel values the distance of a pixel from the bottom of the respective slitlet. The regions of the slitlets should not be strongly curved nor should regions of different slitlets overlap with each other.

Center Right *First raw flat* This plot is mostly of interest in comparison to the **Bottom Right** one, as the number of slitlets and the areas covered by them should be identical. The red lines show the traces of the slitlet edges (in case of MOS/MXU data). They should therefore follow the slit edges and not cut across slitlets. All slitlets should be detected and there should be no spurious detections (e.g. one slitlet detected as several). For LSS and LSS-like data the illuminated region along the y-axis has to be identical between this plot and the bottom right one. Otherwise part of the exposed area is lost.

Bottom Right *Normalized master flat* The normalized master flat field should have the same number of slitlets as the first raw flat and their areas should also be identical. Depending on the normalization method residual gradients may be validly present (e.g. for `sradius`$=-1$ as recommended for LSS data).

The right hand side of the GUI indicates the recipe parameters divided into 3 tabs: FLAT FIELD NORM, MASTER FLAT, WAVE CALIB / DISTORSIONS, which are described below.

FLAT FIELD NORM This tab offers the recipe parameters that deal with the normalization of the combined flat field along the dispersion and/or spatial axis. The fitting or smoothing is performed on 1-dimensional data created by collapsing the 2-dimensional frame along the other axis.

- `sradius` and `dradius` provide the half-width of the smoothing boxes along the spatial and dispersion axis, respectively. The smoothing is performed by calculating the median across the full box width. A value greater than -1 for `dradius_aver` defines the half-width of a smoothing box along the dispersion axis, using the average instead of the median. This is useful for spectra covering wavelength below 4000 Å, where the detector shows signficant pixel-to-pixel variations, that should be kept in the normalized flat field to enable their correction in the science data.
- `s_degree` provides the degree of a polynomial to be fit to the average variation along the spatial axis.

- `d_knots` provides the number of knots for a spline to be fit to the average variation along the dispersion axis.

MASTER FLAT

- `stack_method` offers various methods to combine the flat fields: a simple sum (`sum`), an average (`mean`), a median (`median`), or an average with $\kappa - \sigma$ clipping (`ksigma`). The default value `sum` is usually fine.
- `kiter`: number of iterations (only for `stack_method` = `ksigma`)
- `ksigma`: value for κ (only for `stack_method` = `ksigma`)
- `slit_ident`: If activated the pipeline tries to matching the detected slitlets with slit position information from the header (not applicable to long-slit data). Activating this matching may improve the accuracy of the preliminary spectra detection. If the slit identfication fails the parameter is forced to `false`.

WAVE CALIB / DISTORSIONS

- `startwavelength` and `endwavelength` define the maximum wavelength range covered by the extracted spectra.
- `wdegree` defines the maximum order of the polynomial fitted to the dispersion relation.
- `wdradius` defines the search radius for detected lines around predicted positions after the first run of the pattern matching. A larger radius allows to find more lines, but also increases the risk for false or spurious identifications.
- `wreject` sets the maximum acceptable difference between the calculated wavelength for a line and the value from the line catalog during the fit of the dispersion relation.
- `wmode` (long slit data only) defines the interpolation mode along the spatial axis to cover missing rows. A value of `0` means no interpolation and is useful to check the spatial coverage of identified lines. A value of `1` will interpolate locally across gaps, while a value of `2` will perform a global fit.
- `wmosmode` (multi-object spectroscopy data only) defines the interpolation mode along the spatial axis to cover missing rows. A value of `0` means no interpolation and is useful to check the spatial coverage of identified lines. A value of `1` will interpolate locally (within a given slit only) across gaps, while a value of `2` will perform a global fit.
- `dispersion` provides the first guess of the dispersion in Å/pixel (for binning 1×1) that is used for the pattern matching.
- `peakdetection` sets the threshold to detect spectral lines. If there are issues with the detection of slits or the number of lines varying this parameter will often solve them.
- `ignore-lines` enables the removal of deviating lines from the fit. Please keep in mind that the solution may look better without such lines, but may not be a closer approximation to the real dispersion solution.
- `used_linesets` For some grisms the arc lamp lines are clustered in certain regions. In order to avoid a fit to the dispersion relation that is dominated by these regions only a subsample of the arc lamp lines is used if `used_linesets` = `standard`. Setting instead `used_linesets` = `standard, extended` will include all lines in the catalog.

9.4.2 fors_science (Response)

The interactive window shown in Fig. 9.3 provides information about the quality of the response curve fit:

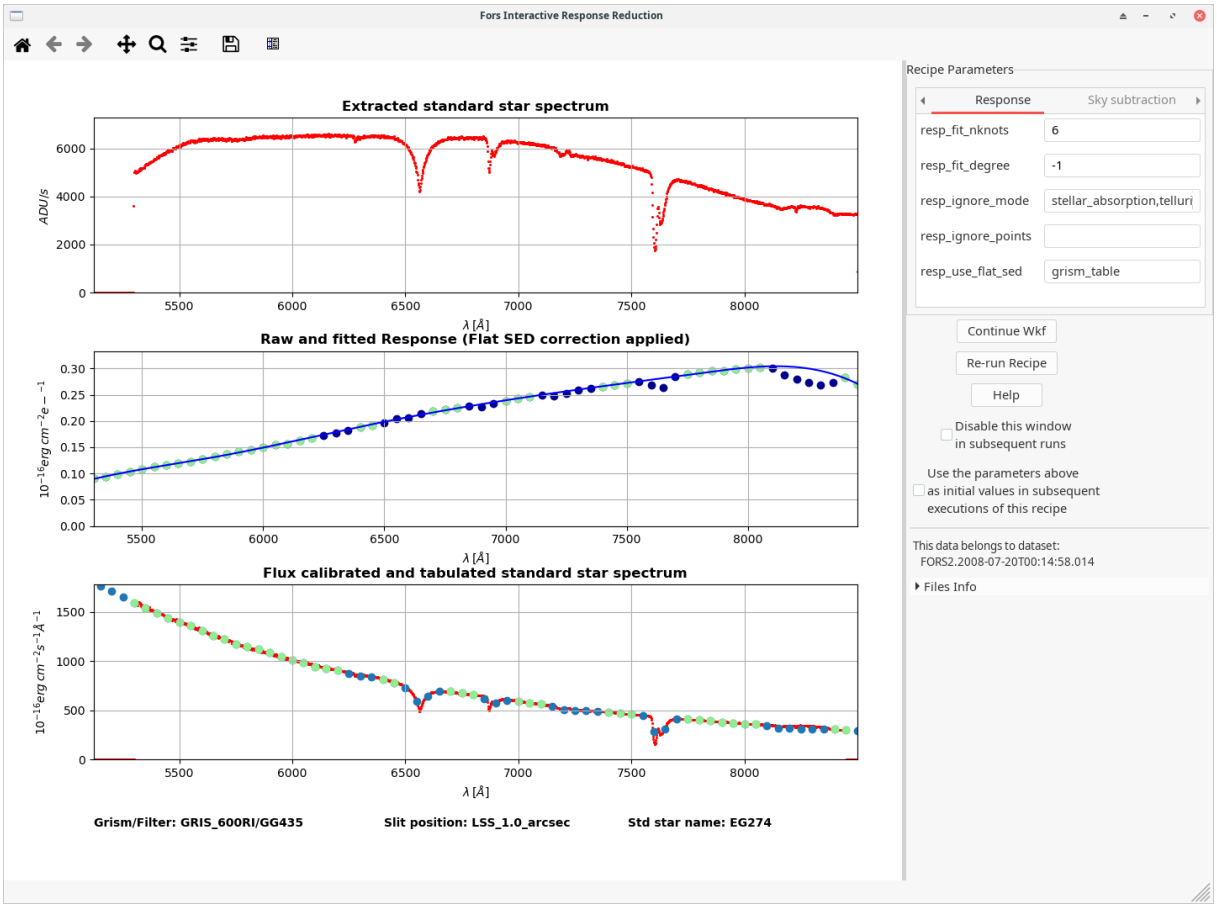


Figure 9.3: *The interactive window of the* `Response Curve` *actor for the first demo DataSet.*.

Top *Extracted standard star spectrum*: The extracted standard star spectrum should shows no jumps or sky emission lines. Strong gradients due to order separating filters are valid but may cause problems with the fit of the response curve.

Center *Raw response and fit*: The dots show the raw response (ratio of reference spectrum and observed spectrum integrated over same bins as reference spectrum) and the blue line shows the corresponding fit. Blue dots are masked (`ignore_resp_mode, ignore_resp_points`) and not used for the fit.

Bottom *Flux-calibrated standard star spectrum and reference*: The red line marks the observed standard star spectrum calibrated with its own response curve and the green and blue points indicate the reference data (blue points were masked during the fitting of the response). Differences between the green points and the red line indicate a problem with the flux calibration, usually features on a scale smaller than the bins of the reference data. Differences between the red line and blue points may indicate problems with inter- and/or extrapolation, but may also be valid, for instance due to different amounts of telluric absorption or to differences in resolution between the reference spectra and the observed data.

To verify the quality of the response determination the user should check the following displays in the interactive

window, which shows up after the execution of the `fors_science` recipe in the `ResponseCurve` actor (see Fig. 9.3):

Raw and fitted response The blue curve (fit) in the central plot should follow the green dots (unmasked raw response) closely. If the fit deviates from the data in the masked regions in unexpected ways (blue dots) the user may try to remove `stellar_absorption` and/or `telluric` from the parameter `resp_ignore_mode` and mask only smaller regions via `resp_ignore_points`[14]. Alternatively or in addition one may change the fit method and/or degree via `resp_fit_degree` (polynomial degree) or `resp_fit_nknots` (number of spline knots).

One should keep in mind that interpolating across a larger region (e.g. covered by telluric lines) provides at best a probable response curve, but may also show strong deviations. This is even more true for extrapolations, e.g. if a response curve has telluric regions at its edges.

Flux-calibrated standard star The red curve (flux-calibrated standard star spectrum) in the lower plot should coincide with the green dots (tabulated standard star flux; blue dots were ignored during the response fit). Deviations at the very blue end may results from problems with atmospheric extinction and deviations at wavelengths above 6000 Å may point to problems with telluric absorption.

If there is no `STD_FLUX_TABLE` associated to the observations of a flux standard star the interactive window will show only the extracted spectrum (see Fig. 9.4). In most cases the reference data for the standard will be available in the `STD_FLUX_CATALOG`, which contains model spectra for the stars. These standard stars are first corrected for telluric absorption (if their spectra extend beyond 6800 Å) and then processed with the recipe `fors_response` (see Sect. 9.3.5 for more details).

RESPONSE
- `resp_fit_nknots` specifies the number of knots to be used for a spline fit of the response curves (`resp_fit_degree` = `-1` required)
- `resp_fit_degree` sets the degree of a polynomial for the fitting of the response curve (`resp_fit_nknots` = `-1` required)
- `resp_ignore_mode` defines which features should be masked: `stellar_absorption` masks strong absorption lines in the stellar spectra, which may cause residuals. `telluric` excludes wavelength regions affected by telluric absorption from the fit. `command_line` excludes the wavelength points or regions specified by `resp_ignore_points`.
- `resp_ignore_points` provides user-defined wavelengths to be masked
- `resp_use_flat_sed` defines if the spectra should be corrected by the spectral energy distribution of the flat field lamp. This is necessary for data observed with holographic grisms, whose response depends on the position on the detector along the dispersion axis. By default it is read from the `GRISM_TABLE`.

SKY SUBTRACTION `skylocal` and `skyglobal` work on 2-dimensional images, which have been corrected for bias and flat field, but are neither wavelength calibrated nor distortion corrected. `skymedian` works on 2-dimensional image which have been wavelength calibrated and distortion corrected.

[14] Please note than an empty field for `resp_ignore_mode` the default value `stellar_absorption,telluric,command_line` is taken. As a work-around, set `resp_ignore_mode` to `command_line` and `resp_ignore_points` to empty, in order to express that no masking should be applied.

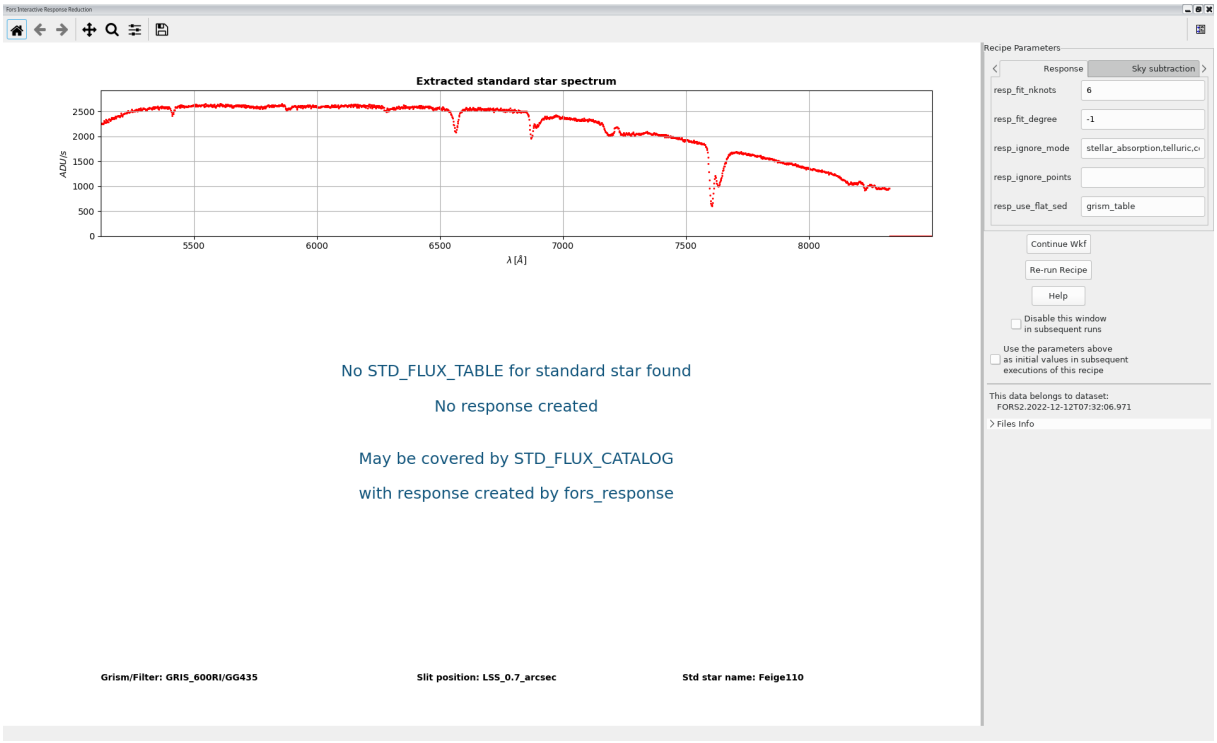


Figure 9.4: *The interactive window of the* `Response Curve` *actor for the fourth demo DataSet, for which the reference data of the standard star are available on in the* `STD_FLUX_CATALOG`..

- `skyglobal` activates the sky determination using the curvature of the sky lines to create a super-sampled sky spectrum. It does not work well in the presence of significant gradients in flux or resolution along the slit, but may give good results for ling-slit data.
- `skylocal` activates the modelling of the sky along each column within a slit in the unmapped data (neither wavelength calibrated nor distortion corrected). It works best for short slits (a few tens of arc seconds) with negligible curvature of sky lines. For long-slit data it is therefore aliased to `skymedian`
- `skymedian` determines the sky by taking the median value along the slit for each wavelength column, which is obviously not a good choice in the case of significant gradients in flux or resolution along the slit. `skymedian` is the most robust method of the three.
- `cosmics` performs a cosmic ray rejection on the 2-dimensional frame (only if `skylocal` or `skyglobal` are activated), which is not very robust. Usually the optimum extraction removes all cosmic ray hits. If `cosmics` is used the results should be compared to the results without using `cosmics` to check for artefacts.

SPECTRA EXTRACTION The values for the recipe parameters in this tab are all provided in unbinned pixels.

- `slit_margin` defines the number of pixels to be excluded at the upper and lower slit edge during detection and extraction of objects.

- `ext_radius` sets the maximum extraction radius for detected spectra. The default value corresponds to an aperture of 3″, which might not be sufficient to cover the spatial extent of an object. For the optimum extractions this parameter can be set to a large value as long as no other spectra are inside the radius.
- `cont_radius` sets the minimum distance at which 2 objects of similar brightness do not contaminate each other's specrum. This parameter is important in crowded regions.
- `ext_mode` sets the extraction mode of the 1-dimensional spectra (`0` for a straight sum, `1` optimum extraction using the Horne algorithm)

WAVE CALIB
- `skyalign` activates the refinement of the wavelength calibration using sky lines, which is by default switched off (`-1`). A value greater than −1 gives the order of the polynomial to fit the wavelength offsets. Standard star spectra have wide sky lines because of the wide slit with which they are observed. The alignment using sky lines will therefore not work well on these data.

9.4.3 fors_response

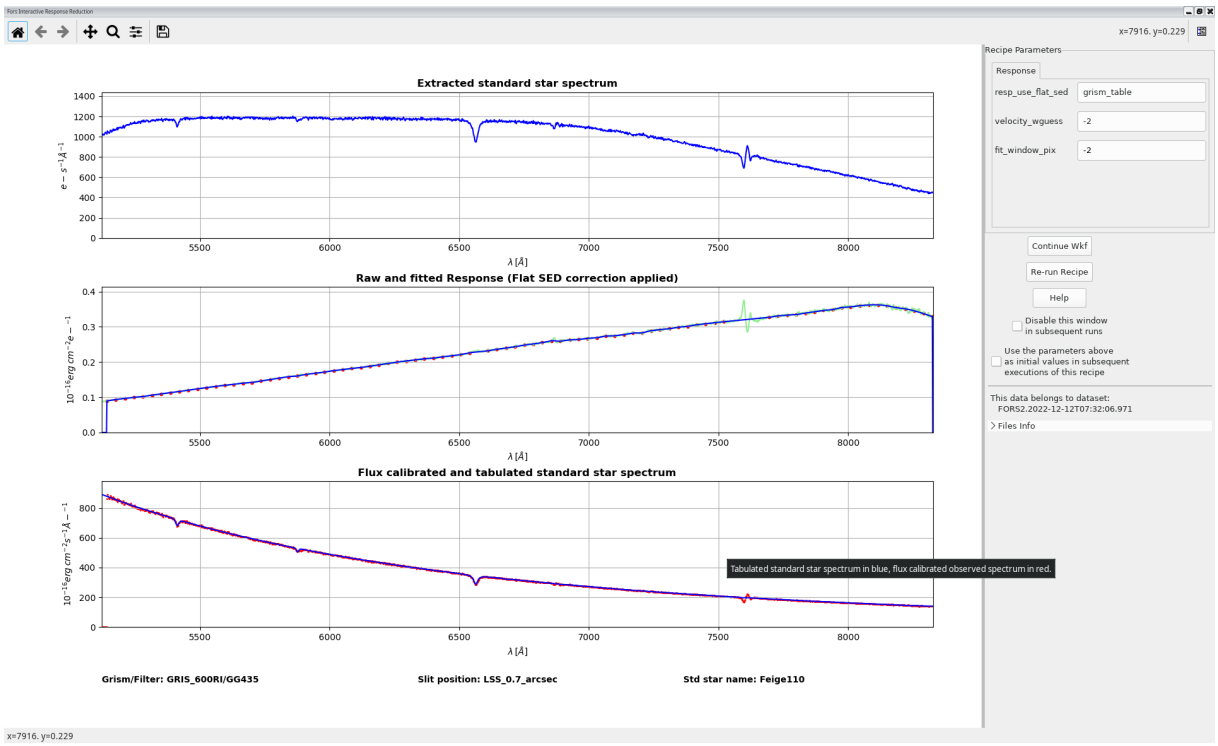


Figure 9.5: *The interactive window of the* `ForsResponse` *actor for the fourth demo DataSet.*.

The interactive window shown in Fig. 9.5 provides information about the quality of the telluric correction and the fit of the response curve.

Top *Extracted standard star spectrum (corrected for telluric absorption if necessary)*: The extracted standard star spectrum should shows no jumps or sky emission lines. Strong gradients due to order separating

filters are valid but may cause problems with the fit of the response curve. The O_2 absorption feature at 7594 Å often shows residuals after the telluric correction due to saturation. The saturation is not visible in FORS2 spectra due to the low resolution, which smears out the flux.

Center *Raw response and fit*: The green line shows the raw response (ratio of reference spectrum and observed spectrum) and the dots mark the median of the raw response calculated at predefined wavelengths over predefined windows. The blue line shows the corresponding fit.

Bottom *Flux-calibrated standard star spectrum and reference*: The red line marks the observed standard star spectrum calibrated with its own response curve and the blue line indicates the reference data. Differences between the blue and the red line point towards problems with the fit of the response curve.

To verify the quality of the response determination the user should check the following displays in the interactive window (see Fig. 9.5):

Raw and fitted response The blue curve (fit) in the central plot should follow the red dots closely. For wavelengths above about 8800 AA the detector shows fringing, which may require larger fit windows to smear out the effect of the fringe pattern, which is generally not the same between standard star and science observations.

Flux-calibrated standard star The red curve (flux-calibrated standard star spectrum) in the lower plot should coincide with the blue one (tabulated standard star flux). Deviations at the very blue end may results from problems with atmospheric extinction and deviations at wavelengths above 6500 Å may point to problems with telluric absorption.

RESPONSE
- `resp_use_flat_sed` defines if the spectra should be corrected by the spectral energy distribution of the flat field lamp. This is necessary for data observed with holographic grisms, whose response depends on the position on the detector along the dispersion axis. By default it is read from the `GRISM_TABLE`.
- `velocity_wguess` provides the wavelength (in Å) of the stellar line from which the radial velocity of the standard star is determined. By default the value is read fom the `GRISM_TABLE`.
- `fit_window_pix` sets the window (in unbinned pixels) across which the median of the raw response is taken at predefined wavelengths recorded in `FIT_POINTS_CATALOG`. By default the value is read fom the `GRISM_TABLE`. A large value may provide a smoother response, especially in wavelength regions affected by fringing.

9.4.4 fors_science (Science)

The interactive window shown in Fig. 9.6 provides information about the quality of the sky subtraction and spectrum extraction. It shows the spectra in ADU/sec, i.e. not flux-calibrated:

Top *Mapped sky-subtracted 2-dimensional spectrum*: The wavelength-calibrated, rectified frame is shown after sky subtraction. The yellow and red lines mark the lower/upper extraction limits of the detected spectra. Right-clicking on such a range will plot the extracted spectrum in the **bottom** plot. For `LSS` data you

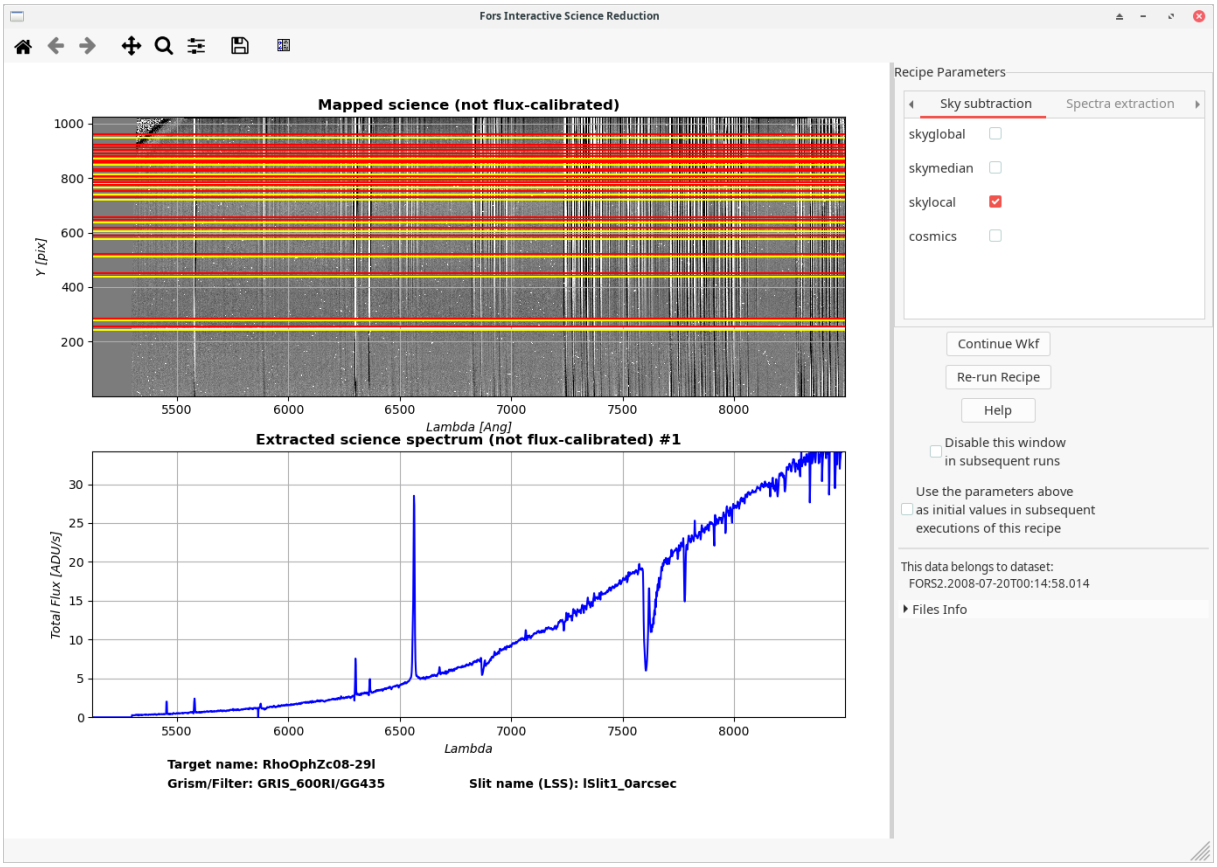


Figure 9.6: *The interactive window of the* `ForsScience` *actor for the first demo DataSet.*.

should check in this window that your spectra are horizontal (see also Frequently Asked Questions, p. 70). This is most easily done by zooming onto a spectrum, selecting the full range in wavelength but only a small range along the y-axis.

Bottom *Extracted science spectrum*: The spectrum should not show strong residuals of sky lines.

A quick check on sky subtraction can be made by examining the sky subtracted frame `MAPPED_SCI_<mode>`. The spectra should have a generally smooth look, and will only appear to be noisier in those regions where bright sky lines were subtracted.

The best way to ensure that the sky was subtracted optimally, at least at the positions of the objects to extract, is to check that the residual noise is compatible with the statistical error associated to the extracted object spectra. The extracted spectra are contained in the `REDUCED_SCI_<mode>` image (one extracted spectrum for each row). Their error spectra (at a 1-σ level) are contained in the `REDUCED_ERROR_SCI_<mode>` image. The regions of the extracted spectra corresponding to a (bright) sky line will include a few noisier points, whose deviation from the spectral continuum should (almost) never pass the 3-σ deviation. If this condition is fulfilled, the sky subtraction is probably as good as it can get.

The sky spectra extracted from the modeled sky frames in exactly the same way as the object spectra from

the sky-subtracted frames can be found in `REDUCED_SKY_SCI_<mode>`.

Note that if the barycentric correction is applied to the products, the wavelengths are re-computed to match the desired reference system. No interpolation is done on the spectrum itself, only wavelengths are changed. This means that spectra of the same target might be defined at different wavelengths, depending on the velocity correction. This has to be taken into account when combining spectra for further analysis.

SKY SUBTRACTION `skylocal` and `skyglobal` work on 2-dimensional images, which have been corrected for bias and flat field, but neither wavelength calibrated nor distortion corrected. `skymedian` works on 2-dimensional image which have been wavelength calibrated and distortion corrected.

- `skyglobal` activate the sky determination using the curvature of the sky lines to create a super-sampled sky spectrum. It does not work well in the presence of significant gradients in flux or resolution along the slit, but may give good results for ling-slit data.
- `skylocal` activates the modelling of the sky along each column within a slit in the unmapped data (neither wavelength calibrated nor distortion corrected). It works best for short slits (a few tens of arc seconds) with negligible curvature of sky lines. For long-slit data it is therefore aliased to `skymedian`
- `skymedian` determines the sky by taking the median value along the slit for each wavelength column, which is obviously not a good choice in the case of significant gradients in flux or resolution along the slit. `skymedian` is most robust method of the three.
- `cosmics` performs a cosmic ray rejection on the 2-dimensional frame (only if `skylocal` or `skyglobal` are activated), which is not very robust. Usually the optimum extraction removes all cosmic ray hits. If `cosmics` is used the results should be compared to the results without using `cosmics` to check for artefacts.

SPECTRA EXTRACTION The values for the recipe parameters in this tab are all provided in unbinned pixels.

- `slit_margin` defines the number of pixels to be excluded at the upper and lower slit edge during detection and extraction of objects.
- `ext_radius` sets the maximum extraction radius for detected spectra. The default value corresponds to an aperture of $3''$, which might not be sufficient to cover the spatial extent of an object. For the optimum extractions this parameter can be set to a large value as long as no other spectra are inside the radius.
- `cont_radius` sets the minimum distance at which 2 objects of similar brightness do not contaminate each other's specrum. This parameter is important in crowded regions.
- `ext_mode` sets the extraction mode of the 1-dimensional spectra (`0` for a straight sum, `1` optimum extraction using the Horne algorithm)
- `generate_idp` activates the creation of VO/Phase3 compliant FITS tables for each extracted spectrum.

WAVE CALIB
- `skyalign` activates the refinement of the wavelength calibration using sky lines, which is by default switched off (`-1`), because it should be used only with careful checking. A value greater than -1 gives the order of the polynomial to fit the wavelength offsets.

9.4.5 **fors_molecfit_model:** *Modelling the atmosphere*

Before executing any Molecfit recipe, a GUI pops up and shows the input spectrum with the pre-defined wavelength regions where the fit will be performed on the left hand side. Figure 9.7 illustrates the interactive window. The right hand side of the GUI indicates the recipe parameters divided into 3 tabs: MOLECULES, EXPERIENCED, and EXPERT, which are described below.

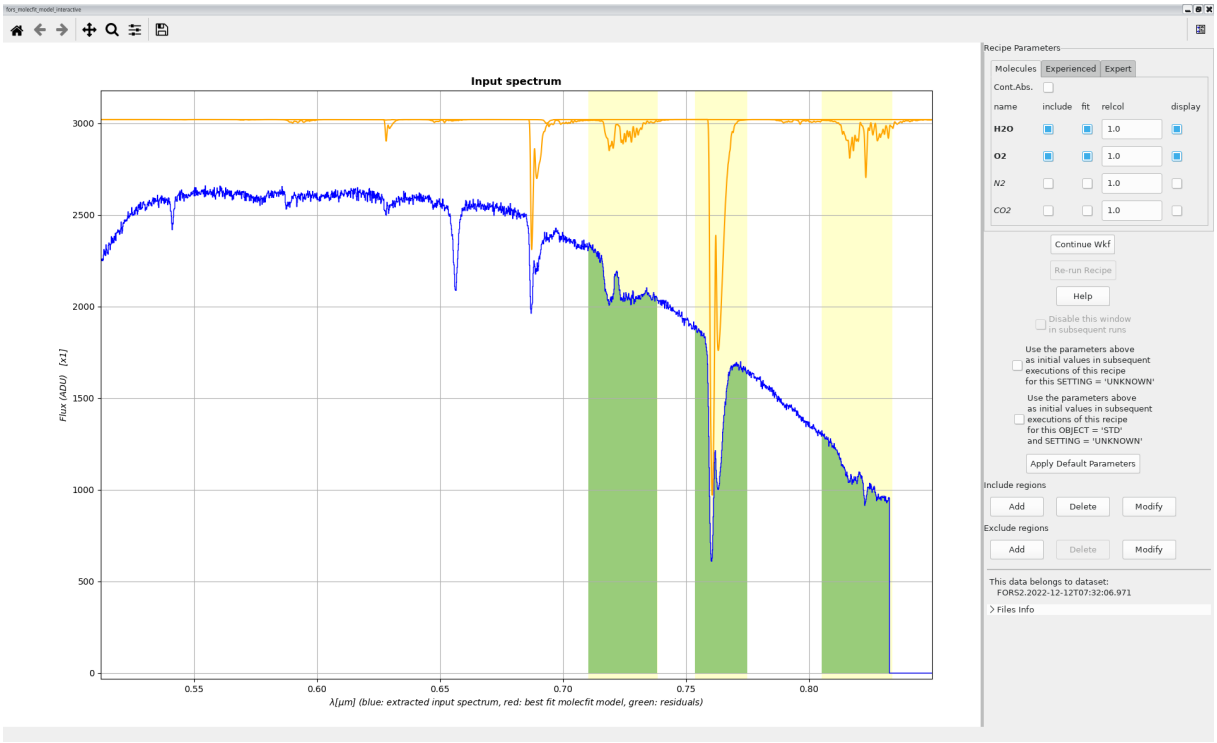


Figure 9.7: *The pre-molecfit window. The spectrum to fit is shown in blue, the wavelength regions to fit are colored in green. The Orange spectrum shows the typical atmospheric transmission of the highlighted molecules (H_2O and O_2 in this case, activated by clicking the button under "display").*.

MOLECULES This tab allows the user to select the molecules to consider in the model, whether to fit or only to include the absorption features for a fixed abundance. The column under "display" adjacent to each molecule provides plots of typical absorption features for that molecule. In this way, one can judge if the molecule is relevant or if the fitting regions have to be adjusted accordingly. The relative abundance, which is used as starting point for the fit, can be also changed. A molecule can also be "included" in the computation of the model, but not fitted. The molecules and their usage in the modeling can also be inserted in the EXPERT tab by setting the recipe parameters: `LIST_MOLEC`, `FIT_MOLEC`, and `REL_COL`.

The fitting or exclusion regions can be added, removed, or modified by pressing the corresponding buttons on the right hand side of the interactive GUI (under "Include regions" or "Exclude regions", respectively). You may have to enlarge the window to see these buttons. Typically, one has to define only inclusion

regions (everything that is not in an inclusion region will be ignored) [15].

To add a region, press the Add button, define the region with the mouse on the spectrum panel, and then press Add again. Same procedure for Delete and Modify. Note: depending on your version of matplotlib, the button Modify can be absent. The definition of inclusion or exclusion regions can be given also in the tab EXPERT by inserting a comma separated list of values in the field of the recipe parameters WAVE_INCLUDE (for inclusion regions) or WAVE_EXCLUDE and PIXEL_EXCLUDE (for exclusion regions).

In order to facilitate the decision which molecules to include and the definition of the wavelength regions, one can click on the relevant molecules to show in the bottom part of the GUI (Select molecules, non-selected molecules, other molecules). The typical telluric absorption spectrum for the corresponding molecule will be shown in the plot panel of the GUI.

We recommend to avoid wavelength regions which include absorption features that belong to the object.

EXPERIENCED This tab contains a few parameters that can be adjusted to improve the fit. The most relevant are:

- FIT_CONTINUUM: it determines if the continuum has to be fitted (1=fit, 0=do not fit). A single entry implies that the decision is applied for all the inclusion regions. A comma-separated series of 1 and 0 enables or disables the continuum fit for each of the inclusion regions. In this latter case, the number of entries must correspond to the number of entries of CONTINUUM_N and the number of inclusion regions.
- CONTINUUM_N: the order of polynomial for the continuum fit. A single number implies that all the include regions are fitted with the same order. A comma-separated list of values assigns a polynomial degree to each include region. In this case, the number of elements in CONTINUUM_N must agree with the number of elements of FIT_CONTINUUM and the number of inclusion regions.
- FIT_WLC and WLC_N. It determines whether the wavelength solution has to be adjusted (FIT_WLC = 1) to match the nominal position of the telluric features and which polynomial order to use for the fit (WLC_N).
- Parameters for the spectral resolution. One can determine which model to use for the line spread function, either a box-car function (FIT_RES_BOX) with initial width guess specified by RES_BOX (in units of slit widths); a Gaussian function (FIT_RES_GAUSS) with initial guess for the amplitude specified by RES_GAUSS (in units of pixels), a Lorentzian function (FIT_RES_LORENTZ) with initial guess amplitude specified by RES_LORENTZ (in units of pixels) or a combination of these. The fit can be done assuming the amplitude of the convolving kernel in pixels to be constant or variable with wavelength (tick on VARKERN); in this case the spectral resolution $\lambda/\Delta\lambda$ is kept constant, whereas the kernel amplitude $\Delta\lambda$ varies with wavelength. FORS2 has constant $\Delta\lambda$ and variable spectral resolution.

Please, consult the molecfit manual for other parameters.

EXPERT This tab contains in addition to parameters that are also available via the MOLECULES tab (LIST_MOLEC, FIT_MOLEC, REL_COL, WAVE_INCLUDE, EAVE_EXCLUDE) the possibility to define the parameters

[15] Typically, each inclusion region has its continuum fit (but see the CONTINUUM_N parameter description in the description of the tab EXPERIENCED for details). However, one might define an exclusion region "inside" an inclusion region: in this case the inclusion region also has only one continuum component, despite its split into 2 parts.

`PIXEL_EXCLUDE`, `DEFAULT_ERROR`, and `PWV`. Because the values are typed one has to be careful to provide the correct format.

After pressing the `Continue` button (and eventually the buttons to re-use the configured parameters for other datasets) the fit starts. The fit results is displayed on a second GUI (see Figure 9.8, very similar to Fig. 9.7).

The new GUI shows the fit results in red (defined only in the wavelength regions used in the fit), the residuals (in green). Moreover, one can also overplot the continuum fit, the full telluric absorption and the telluric corrected spectrum to judge the goodness of the entire process outside the fitting regions by clicking on `Overplot continuum fit`, `Overplot calctrans transmission`, and `Overplot calctrans corrected spectrum`, respectively. If needed, the recipe can be re-executed with adjusted parameters by pressing the button `Re-run recipe`. The parameters that can be adjusted are the same as those described above.

The window also displays some useful fit results, such as the χ^2.

If happy with the result, press the `Continue` Button. Eventually, the parameters can be saved for the next runs by clicking on one of the two `Use the parameters above...` options.

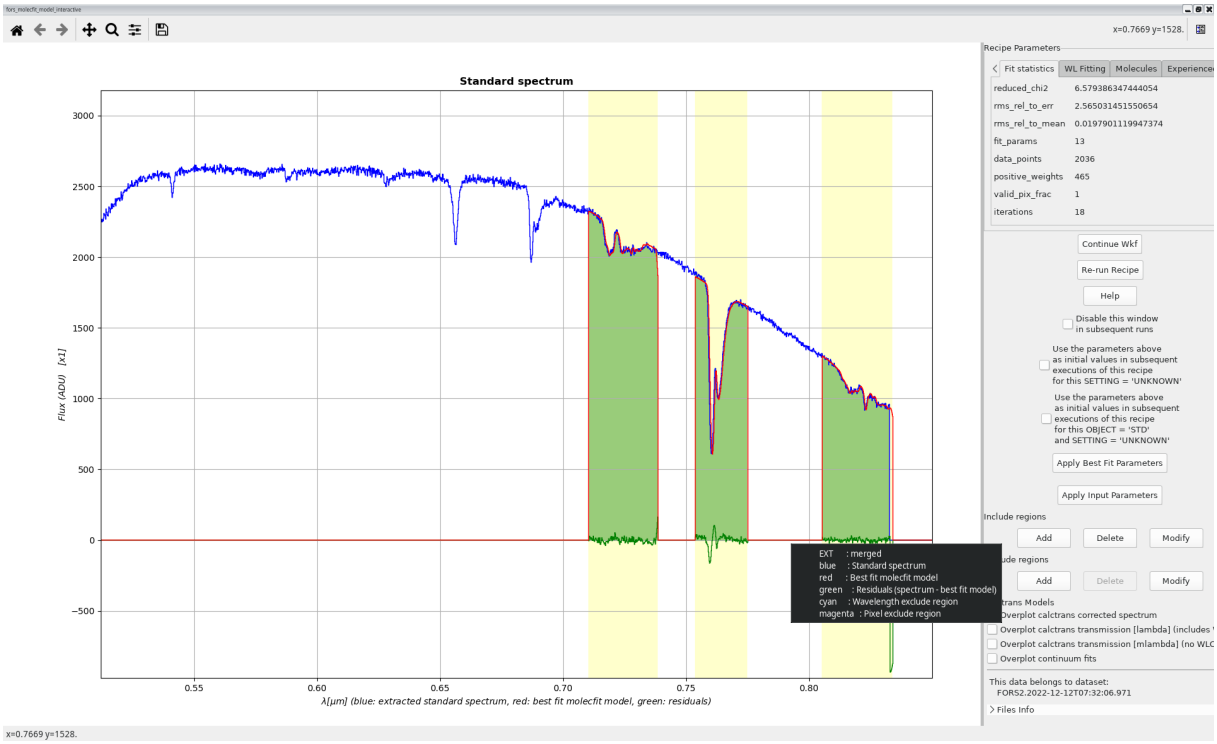


Figure 9.8: *The window showing the fit result of molecfit. The spectrum to fit is shown in blue, the wavelength regions to fit are colored in green/yellow. The best fit model is shown in red. The green spectrum shows the residuals. .*

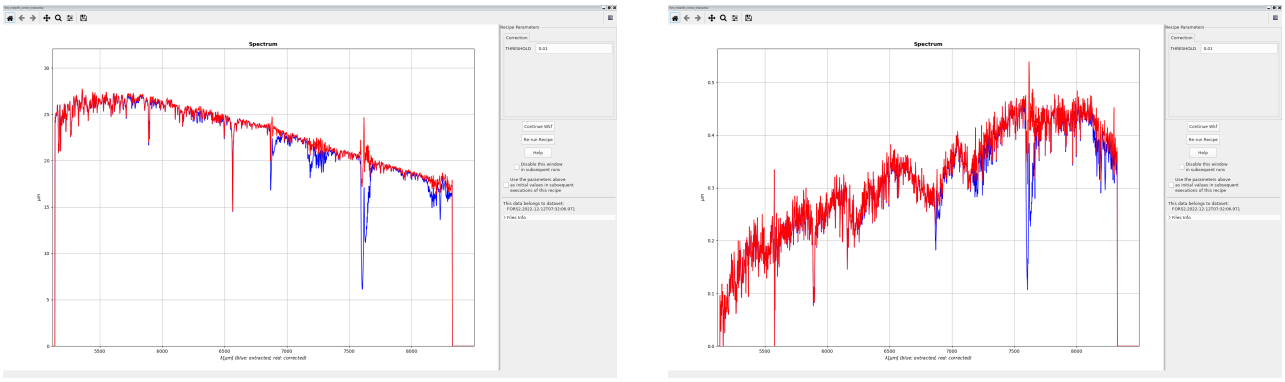


Figure 9.9: Left panel: the first telluric corrected spectrum from the fourth demo-dataset. The signal-to-noise ratio of the input spectrum is high enough to highlight the benefits of the telluric correction. The original spectrum is shown in blue, the telluric corrected spectrum in red. Right panel: the third telluric corrected spectrum from the fourth demo-dataset. Contrarily to the example shown in the left panel, the spectrum has rather low signal-to-noise. The benefits of telluric correction are much smaller than in the left panel.

9.4.6 fors_molecfit_calctrans / fors_molecfit_correct: *Computing the atmospheric transmission and correcting the spectrum*

The next steps compute the full telluric correction for all the input science spectra using the best fit model parameters determined at the first step. The correction is then applied to the data. Note: although the science spectra share the same reference spectrum used to determine the best fit parameters, each individual science spectrum must have its own correction because they might differ in wavelength extension (e.g., in the case of MOS or MXU observations)

The correction GUI window (see Figure 9.9) allows to inspect the science spectrum under analysis and to change the threshold below which the correction is not applied. A non-zero threshold is useful to avoid small number division in case of saturated telluric absorption.

Press the `Continue` button to correct the next spectrum in the list.

Note: Only 1-dimensional spectra in Phase3 format are corrected, not the various 2D frames produced by the FORS2 science recipe.

10 Improving Your Results

In this section we provide information on how to improve your results by changing the parameters of the `fors_calib` and `fors_science` recipes. For troubleshooting in case of pipeline failures see Sect. 12.2.

10.1 Demo Data

DataSet 1: `MXU GRIS_300V GG435` The interactive window of the `ForsResponse` actor for `fors_response` shows a wavy pattern in the fitted response above about 7000 Å, which is caused by fringing. Increasing the value of `fit_window_pix` to `60` creates a smoother fitted response. Because the fringe pattern differs between science and standard star observations it is not useful to try fitting it in the response curve.

The interactive window of the `ForsScience` actor shows that a value of `25` for `ext_radius` provides cleaner spectra.

DataSet 2: `MOS GRIS_300I OG590` In the interactive window of the `ForsCalib` actor one sees a gap in the wavelength-calibrated `MOS` arc lamp frame shown in the top left plot. Reducing the `peakdetection` to `200` solves this issue, because the slit edges are better traced.

The interactive window of the `ForsScience` actor shows that a value of `30` for `ext_radius` provides again cleaner spectra.

The first interactive window for the `TelluricCorrection` actor shows that the brightest spectrum has an emission line at about 0.9110 μm in the default fit range for the telluric absorption. The blue end of the reddest fit range should be changed from 0.8906 μm to about 0.9150 μm. This can be done either using the `Modify` button for the "Include regions" at the bottom right of the interactive window or by changing manually the values listed in the Tab EXPERT at `WAVE_INCLUDE`.

DataSet 3: `LSS GRIS_1400V free` The wavelength calibration can be improved by increasing the value of `peakdetection` to `90` in the ForsCalib interactive window, thereby removing spurious line detections.

In the interactive window of the `ForsScience` actor the extracted spectrum can be improved by enabling `skyglobal` and disabling `skylocal`, together with an increase of `ext_radius` to `20`.

DataSet 4: `LSS GRIS_600RI GG435` In the interactive window of the `ForsScience` actor the extracted spectrum can be improved by enabling `skyglobal` and disabling `skylocal`. This leaves some probably spurious detections in the lower part of the detector. Moreover, the brightest spectrum, especially its emission lines, is much wider than the default extraction window. Increasing `ext_radius` to `110` covers the full spatial extent of the brightest spectrum. Setting `cont_radius` to `1` removes all spurious detections, but unfortunately also a valid one at $y = 487$.

DataSet 5: `MOS GRIS_300V GG435` In order to detect both slits in the `MOS` spectra one has to set the value of `peakdetection` in the interactive window of the `ForsCalib` actor to `300`.

The interactive window of the `ForsScience` actor shows two spurious detections of spectra. Setting `cont_radius` to `1` will remove them.

DataSet 6: `LSS GRIS_300V free` Switching from `skylocal` to `skyglobal` in the interactive window of the `ForsScience` actor will improve the sky subtraction. Varying the value for `ext_radius` from

`12` to `60` will change the relative intensity of the three emission features between 5000 Å and 5500 Å. Setting in addition `cont_radius` to `1` will remove spurious detections

10.2 fors_calib

10.2.1 Slit Tracing

If you find that the slits in your data were not well traced you may want to change the parameter `cdegree`. Currently you can do so only on the command line or by changing the value in the `GRISM_TABLE`, which is listed under *Files Info* at the bottom right of the interactive window. In order to force a re-run of the recipe after changing the value in the `GRISM_TABLE` you may change a parameter which has no influence on your result (for instance `kiter`, which is not used for `stack_method` = `sum`). Keep in mind that the changed `GRISM_TABLE` will be used also for all future reductions.

Alternatively the slitlets in your data may be too tightly packed. As a default, the `fors_calib` recipe tries to recover untraceable edges by interpolating a global curvature model based on other traceable edges (if they are available). Using this global description of the spectral curvature helps to extract also those spectra whose edges cannot be traced. In some cases however the recipe may find and accept a bad tracing as if it were good, producing a bad global curvature model, and therefore a bad spectral extraction. Setting `cmode` to `0` will suppress the usage of the global curvature model. In this case the recovery strategy of lost spectral edges will consist in replicating the trace of the other available spectral edge (opportunely shifted) of the same slit spectrum. This may improve the results in some cases: however, if a tracing is missing for both edges of a slit spectrum, the spectrum will not be extracted.

10.2.2 Wavelength Calibration

To verify the quality of the wavelength calibration the user should check the displays in the interactive window, which shows up after the execution of the `fors_calib` recipe (see Sect. 9.4.1 and Figs. 6.4, p. 22 and 9.2 (p. 43) for examples).

While the parameters in the grism tables should generally provide good results some grisms are sensitive to the positions of the slits on the CCD, the line intensities, etc. In these cases it may make sense to try changing certain parameters to improve the results. For problematic cases setting `wmode` (LSS/LSS-like) or `wmosmode` (MOS/MXU) to `0` can be helpful as then only lines that are actually found and detected are used, without interpolation along the spatial axis.

Below we list suggestions to solve problems that may occur during wavelength calibration:

Problems with line detection and identification

Setting `wradius` to `0` will force the pipeline to use only those lines for wavelength calibration that were identified by pattern-matching. This can sometimes help to understand why some lines were not or not correctly identified.

Faint arc lamp lines If some of the reference lines listed in the catalog do not peak above a given threshold, they are not used by the pattern-matching task. This can be solved by lowering the value for `peakdetection`. If the faint lines are too close to the noise level you may remove them by specifying their wavelengths in the parameter `ignore_lines`.

Incorrect spectral dispersion The actual mean spectral dispersion can differ significantly from the default read from the `GRISM_TABLE`, for instance due to temperature effects or extreme slit positions. In general the pattern-recognition algorithm is quite robust against changes of the spectral dispersion of up to 20%, but for some grisms good results can only be obtained within a very close first-guess. Check for information on **Individual Grisms** on p. 60.

Spectra at very large offsets The observation may include spectra at large offsets along the dispersion axis, so that only part (red or blue) of their full wavelength range is observed. If the line catalog contains too few reference lines in this region (say, less than 5), they might not be enough to define an unambiguous pattern to detect. Setting `used_linesets` to `standard,extracted` will make additional lines available for a more complete coverage of the bluest/reddest parts of the complete spectral range, which might solve this problem. If there are no extra lines to be used as a reference (as is the case for some grisms), the truncated spectra will be definitely lost.

Valid reference lines are rejected

Sometimes the peak detection algorithm may return inaccurate positions of the detected reference arc lamp lines. Outliers are automatically rejected by the fitting algorithm, but if those lines were properly identified, not rejecting their positions may really improve the overall accuracy of the wavelength calibration.

In such cases increase the value of the `wreject` parameter, but check the results carefully: a tolerant line identification may provide an apparently good fit, but if this is based on misidentified lines the calibration will include unknown systematic errors.

Systematic trends in the wavelength calibration residuals

This problem may be solved by increasing `wdegree`[16], but one should be careful to avoid overfitting. This happens preferably at the red and blue ends of the spectra, where the polynomial is so poorly constrained that it also fits the uncertainties of the line positions, incorporating this noise into the solution: the corresponding residuals will be very small, but the calibrated spectra will look very noisy at the edges of the wavelength range. An extreme case of overfitting is, for instance, fitting 4 points with a 3rd degree polynomial: the residuals will be exactly zero, and yet the obtained model will be highly inaccurate. To reduce the influence of noise on the solution one may set `wmosmode` to 1 or 2 (for MOS/MXU data). `wmode` is already set to 2 by default for LSS and LSS-like data.

The calibrated spectra look "noisy" at their ends

This is a consequence of overfitting (see above for details). Here `wdegree` should be reduced. If this introduces systematic residuals one may try to increase `wradius`.

[16] Currently the pipeline does not allow `wdegree` > 5

The calibrated LSS spectrum looks distorted

This happens if the global wavelength calibration model is bad, either due to some bad local solutions or due to too few local solutions for global modeling. Rerunning the recipe with `wmode` set to `0` will show the result without any interpolation. The situation can be improved by varying `dispersion` and/or `peakdetection` to maximise the number of obtained local solutions and to get more uniform results in the `reduced_lamp_lss.fits` image. If the problems persist setting `wradius` to `0` will limit the lines used to those detected solely by pattern matching. Once a new set of parameters is found that yields good results set `wmode` again to `2` and rerun the recipe. Also `wradius` may now be set back to its default value of `4`.

Individual Grisms

Here we list tips for individual grisms. The grisms are sorted by increasing resolution. Keep in mind that the pipeline does not allow `wdegree` > 5.

GRIS_150I+OG590 This grism may need tailored parameter settings for each setup. Increasing `peakdetection` to `300–400` and/or changing `wradius` to higher or lower values may provide better results for certain masks. Changing the `dispersion` within the range `2.9` to `3.8` may be helpful as well.

GRIS_300V+GG435 Reducing `wdegree` from `5` to `4` produces in some cases more stable results. In some cases increasing `peakdetection` to `350` can improve the slit tracing.

GRIS_300I+OG590 Bad slit tracing can often be improved by increasing `peakdetection` to `400`. Changing `wdegree` to higher or lower values can also help.

GRIS_600B Setting `wdegree` to `5` sometimes gives better results for both wavelength calibration and slit tracings. Also changing `wradius` to lower or higher values can help.

GRIS_600RI+GG435 In some cases the solution can be improved by changing `wdegree` to `3` or `5`.

GRIS_1200B For this grism it is sometimes impossible to find a solution for all slitlets, especially if the slitlets are widely distributed along the dispersion direction. The user may have to process the data twice, once optimizing for the redder slitlets and once for the bluer ones. In that case varying `dispersion` between `0.32` (blue slits) and `0.38` (red slits) may give good results for extreme slits. The wavelength calibration reacts very sensitively to the presence or absence of the line 4077 Å. If results stay unstable the user should ignore one or more of the following lines via the `ignore_lines` parameter: `3466`, `3611`, `3965`, `4026`, `5331`, `5401`.

GRIS_1400V Decreasing `wradius` to `2` and/or increasing `peakdetection` to `90` or `120` can help to improve unstable solutions. Some solutions especially for MOS/MXU data can be improved by increasing `wdegree` to `4`.

GRIS_1028z+OG590 The solution can be improved in some cases by increasing `wdegree` to 4. Decreasing `wradius` to `2` may improve unstable solutions.

10.2.3 Flat Field Normalization

In general the user will want to remove all traces of the flat field lamp but keep the slit profile (spatial illumination) together with the pixel-to-pixel variation of the detector. The default parameters of the flat field normalization are optimized for this situation (`sradius`, `s_degree`,`d_knots` set to `-1`, `dradius` set to `10`). The spline fit along the dispersion axis `d_knots>0` is not well suited for data with very steep slopes due to order separation filters (e.g. `GRIS_150I+OG590`).

Avoiding normalization along the slit (i.e. setting both `sradius` and `s_degree` to `-1`) keeps the slit illumination and thus allows the user to correct the illumination profile of the science data. This facilitates the fitting of the sky background, which should then be flat along the spatial axis.

10.3 Response Determination

10.3.1 fors_science

To verify the quality of the response curve determined by the `fors_science` recipe the user should check the displays in the interactive window, which shows up after the execution of the `fors_science` recipe (see Sect. 9.4.2 and Fig. 9.3, p. 46).

The reference data for the flux standard stars have entries that identify the presence of strong stellar features (column `STLLR_ABSORP` set to `T`(rue)). These regions will be ignored during the response fit if `resp_ignore_mode` is set to `stellar_absorption`. Depending on the resolution of the grism such masking may or may not be necessary. If additional regions need to be ignored `resp_ignore_mode` set to `stellar_absorption, command_line` allows the user to specify the additional points (e.g. `4750, 4900`) and/or regions (e.g. `8900-9850`) with `resp_ignore_points`

In order to avoid the fitting of regions affected by telluric absorption (whose strength may differ between reference spectra and FORS2 data) the static calibration table `fors2_telluric_regions.fits` contains such regions (per grism, as their width depends on spectral resolution). These can be masked by setting `resp_ignore_mode` to `telluric`. If additional regions need to be ignored `resp_ignore_mode` set to `telluric, command_line` allows to specify the additional points/regions with `resp_ignore_points`. This is especially useful for data with very strong variations at the edges of the response (e.g. the 4$^{\text{th}}$ demo dataset, see Sect. 10.1).

By default `resp_ignore_mode` is set to `stellar_absorption, telluric, command_line`.

In order to take small scale variations into account when fitting the response the pipeline uses a spline fit with the number of knots defined with `resp_fit_nknots`. For grisms with a small wavelength range and/or only moderate large-scale variation in their response a small number of knots can be sufficient. For grisms with small-scale variations a higher number of knots may be necessary (e.g. fifth demo dataset, see above). One should keep in mind that the maximum number of knots for the spline is the number of unmasked data points – 2 (entering a higher number will cause the pipeline to reduce it to the allowed maximum). Since the knots are distributed at equal distances this means that the distance between two knots is always larger than the distance between two data points. This explains why even at a maximum number of knots the fit may not go through all data points.

10.3.2 fors_response

To verify the quality of the response curve determined by the `fors_response` recipe the user should check the displays in the interactive window, which shows up after the execution of the `fors_response` recipe (see Sect. 9.4.2 and Fig. 9.5, p. 49).

The reference data used by `fors_response` are model spectra at laboratory wavelengths and without telluric absorption. This requires two corrections to be applied before the response can be calculated:

1. The observed data have to be corrected for telluric absorption if they cover wavelengths above 6800 Å (see Sect. 10.4 below).
2. The model spectra have to be aligned in wavelength to the observed spectra to avoid pseudo P Cygni profiles in the raw response. This is done using a strong absorption line to determine the radial velocity of the observed spectrum, which is then applied to the model spectra.

The response curves may show a wavy pattern at wavelengths above about 7000 Å, which is caused by fringing. Because the fringe pattern differs between science and standard star observations it is not useful to try fitting it in the response curve. Substantially increasing the value of `fit_window_pix` helps to create a smoother fitted response. This can also be helpful in the case of noisy standard star observations.

If substantial P Cygni profiles are visible in the raw response the wavelength of the line used to determine the radial velocity can be changed with the recipe parameter `velocity_wguess`.

10.4 Telluric Correction

The two strong O_2 features at about 6870 Å and 7630 Å often show residuals after the telluric correction. This is due to the fact that they are often saturated, which is not noticeable in the low-resolution FORS spectra because the signal is smeared out.

10.5 Science Data Reduction

Also for science data some useful checks can be made, for which the `ProductExplorer` can be very helpful.

- **Were all spectra properly wavelength calibrated?**

  The overall quality of the wavelength calibration can be examined in the `MAPPED_ALL_SCI_<mode>` image[17]. This image contains the scientific spectra from each slit after removing the optical and spectral distortions. The visible sky lines should all appear perfectly aligned and vertical.

- **Were all spectra extracted?**

  For very faint sources or sources with emission lines only the pipeline may not extract the spectra. In some cases changing the sky subtraction from `skylocal=true` and `skyglobal=false` to `skyglobal=true`

[17] This file is accessible either via the `ProductExplorer` (use the `Inspect with` button) or via the *Files Info* at the bottom right of the interactive window (copy the file path and display it with your favourite display tool (e.g. `skycat`))

and `skylocal=false` helps. Alternatively you can try to increase the extraction radius via `ext_radius` or change the masking of the slit edges via `slit_margin`. In the worst case you will have to extract the spectra manually with some other software of your choice.

- **Spurious objects detected at the slit edges**

  As a default the *fors_science* recipe excludes objects that are detected within 3 pixels from the slit ends. This might not be enough in some cases and increasing `slit_margin` might help.

- **Sky subtraction for resolved sources**

  In case of extended objects filling most or all of the slit, the evaluation of the sky may be strongly biased by the inclusion of signal that actually belongs to the object to extract. Subtracting this contaminated background would actually destroy the object spectrum.

  For extended objects one should set `skyglobal` to `true` and `skylocal` to `false`. Then the pipeline subtracts a supersampled model of the median sky spectrum observed in all slits from all spectra. This method requires very similar spectral resolution for all slits and will not work well if this is not the case. It is always possible to process the scientific exposures in both ways, one for processing point-like sources and the other for processing spatially resolved sources.

- **Sky subtraction for curved or tilted slits**

  Obvious residuals related to the sky subtraction are visible on the extracted slit spectra. In this case one should set `skymedian` to `true` and `skylocal` to `false`. While `skylocal` subtracts the sky *before*, `skymedian` subtracts *after* the rectification of the spectral data. The second method performs poorly in comparison to the first, but in the case of curved or slanted slits there is at the moment no other choice than using it.

11 Frequently Asked Questions

- **The error window fills the whole screen - how can I get to the `Continue`/`Stop` buttons?**

  Press the Alt key together with your left mouse button to move the window upwards and to the left. At the bottom the `Continue`/`Stop` buttons will be visible. This bug is known but could not yet be fixed.

- **I tried to `Open` (or `Configure`) an `Actor` while the workflow is running and now it does not react any more. What should I do?**

  This is a limitation of the underlying Kepler engine. The only way out is to kill the workflow externally. If you want to change anything while a workflow is running you first need to pause it.

- **After a successful reduction of a data set, I changed this data set in some way (e.g. modified or removed some files, or changed the rules of the Data Organizer). When I restart Reflex, the Data Set Chooser correctly displays my new data set, but marks it as "reduced ok", even though it was never reduced before. What does this mean?**

  The labels in the column "Reduced" of the Data Set Chooser mark each dataset with "OK", "Failed" or "-". These labels indicate whether a data set has previously successfully been reduced at least once, all previous reductions failed, or a reduction has never been tried respectively. Data sets are identified by their name, which is derived from the first science file within the data set. As long as the data set name is preserved (i.e. the first science file in a data set has not changed), the Data Organizer will consider it to be the same data set. The Data Organizer recognizes any previous reductions of data sets it considers to be the same as the current one, and labels the current data set with "OK" if any of them was successful, even if the previously reduced data set differs from the current one.

  Note that the Product Explorer will list all the previous reductions of a particular data set only at the end of the reduction. This list might include successful and/or unsuccessful reduction runs with different parameters, or in your case with different input files. The important fact is that these are all reductions of data sets with the same first raw science file. By browsing through all reductions of a particular raw science file, the users can choose the one they want to use.

- **Where are my intermediate pipeline products?** Intermediate pipeline products are stored in the directory `<TMP_PRODUCTS_DIR>` (defined on the workflow canvas, under Setup Directories) and organised further in directories by pipeline recipe.

- **Can I use different sets of bias frames to calibrate my flat frames and science data?** Yes. In fact this is what is currently implemented in the workflow(s). Each file in a DataSet has a purpose attached to it (Forchì (2012)). It is this purpose that is used by the workflow to send the correct set of bias frames to the recipes for flat frame combination and science frame reduction, which may or may not be the same set of bias frames in each case.

- **Can I run `Reflex` from the command line?** Yes, use the command:

  ```
  esoreflex -n <workflow_path>/<workflow>.xml
  ```

  The -n option will set all the different options for Kepler and the workflows to avoid opening any GUI elements (including pipeline interactive windows).

  It is possible to specify workflow variables (those that appear in the workflow canvas) in the command line. For instance, the raw data directory can be set with this command:

```
esoreflex -n -RAW_DATA_DIR <raw_data_path> \
             <workflow_path>/<workflow>.xml
```

You can see all the command line options with the command `esoreflex -h`.

Note that this mode is not fully supported, and the user should be aware that the path to the workflow must be absolute and even if no GUI elements are shown, it still requires a connection to the window manager.

- **How can I add new actors to an existing workflow?** You can drag and drop the actors in the menu on the left of the `Reflex` canvas. Under `Eso-reflex -> Workflow` you may find all the actors relevant for pipeline workflows, with the exception of the recipe executer. This actor must be manually instantiated using `Tools -> Instantiate Component`. Fill in the "Class name" field with `org.eso.RecipeExecuter` and in the pop-up window choose the required recipe from the pull-down menu. To connect the ports of the actor, click on the source port, holding down the left mouse button, and release the mouse button over the destination port. Please consult the Reflex User Manual (Forchì (2012)) for more information.

- **How can I broadcast a result to different subsequent actors?** If the output port is a multi-port (filled in white), then you may have several relations from the port. However, if the port is a single port (filled in black), then you may use the black diamond from the toolbar. Make a relation from the output port to the diamond. Then make relations from the input ports to the diamond. Please note that you cannot click to start a relation from the diamond itself. Please consult the Reflex User Manual (Forchì (2012)) for more information.

- **How can I manually run the recipes executed by Reflex?** If a user wants to re-run a recipe on the command line he/she has to go to the appropriate reflex_book_keeping directory, which is generally reflex_book_keeping/<workflow>/<recipe_name>_<number> There, subdirectories exist with the time stamp of the recipe execution (e.g. 2013-01-25T12:33:53.926/). If the user wants to re-execute the most recent processing he/she should go to the `latest` directory and then execute the script `cmdline.sh`. Alternatively, to use a customized `esorex` command the user can execute

```
ESOREX_CONFIG="INSTALL_DIR/etc/esorex.rc"
PATH_TO/esorex --recipe-config=<recipe>.rc <recipe> data.sof
```

where INSTALL_DIR is the directory where Reflex and the pipelines were installed.

If a user wants to re-execute on the command line a recipe that used a specific raw frame, the way to find the proper data.sof in the bookkeeping directory is via `grep <raw_file> */data.sof`. Afterwards the procedure is the same as before.

If a recipe is re-executed with the command explained above, the products will appear in the directory from which the recipe is called, and not in the reflex_tmp_products or reflex_end_products directory, and they will not be renamed. This does not happen if you use the `cmdline.sh` script.

- **If I enter "-" into an empty integer parameter of an interactive window it is automatically completed to "-1". Why?**

  The parameters are validated for correctness according to their type (e.g. string, integer, float). In the case of an integer or float parameter "-" alone is considered an invalid input and is therefore automatically completed to "-1". This is part of the validation of input done by the WxPython library.

- **Can I reuse the bookkeeping directory created by previous versions of the pipeline?**

  In general no. In principle, it could be reused if no major changes were made to the pipeline. However there are situations in which a previously created bookkeeping directory will cause problems due to pipeline versions incompatibility. This is especially true if the parameters of the pipeline recipes have changed. In that case, please remove the bookkeeping directory completely.

- **How to insert negative values into a textbox?**

  Due to a bug in wxPython, the GUI might appear to freeze when attempting to enter a negative number in a parameter's value textbox. This can be worked around by navigating away to a different control in the GUI with a mouse click, and then navigating back to the original textbox. Once focus is back on the original textbox the contents should be selected and it should be possible to replace it with a valid value, by typing it in and pressing the enter key.

- **I've updated my Reflex installation and when I run esoreflex the process aborts. How can I fix this problem?**

  As indicated in Section 5, in case of major or minor (affecting the first two digit numbers) Reflex upgrades, the user should erase the `$HOME/KeplerData`, `$HOME/.kepler` directories if present, to prevent possible aborts (i.e. a hard crash) of the esoreflex process.

- **How can include my analysis scripts and algorithms into the workflow?**

  EsoReflex is capable of executing any user-provided script, if properly interfaced. The most convenient way to do it is through the Python actor. Please consult the tutorial on how to insert Python scripts into a workflow available here: `www.eso.org/sci/data-processing/Python_and_esoreflex.pdf`

11.1 FORS specific questions

1. **Some datasets are greyed out and a mouse-over yields "Missing GLOBAL_DISTORTION_TABLE". What can I do?**

   First of all, you can process the data also without this calibration by clicking on the box in the "Select" column of the `Data Set Chooser` or using the `Select all` button. Then the spatial distortion of your LSS data is not corrected, which is of importance mainly if you are interested in the spatial information about your targets. The extraction quality of the spectra is generally affected only at the very edges of the field.

   The `GLOBAL_DISTORTION_TABLE`s do not exist for the FORS2 grisms `GRIS_200I`, `GRIS_600V`, `GRIS_600R` nor for any FORS1 grism. It exists for the grisms `GRIS_300V` and `GRIS_300I` that have been in use in FORS2 since April 2009. To use the `GLOBAL_DISTORTION_TABLE`s of these two grisms for older data you need to change the OCA rules (located at `<install_dir>/share/esopipes/<pipeline-version>/reflex/fors_spec_wkf.oca`). You have to change the part that associates the `GLOBAL_DISTORTION_TABLE`

   ```
   select file as GLOBAL_DISTORTION_TABLE from calibFiles where
   PRO.CATG=="GLOBAL_DISTORTION_TABLE" and
   inputFile.INS.GRIS1.ID==INS.GRIS1.ID and
   ```

```
inputFile.DET.CHIP1.X==DET.CHIP1.X and
inputFile.DET.CHIP1.Y==DET.CHIP1.Y and
inputFile.DET.CHIP1.PSZX==DET.CHIP1.PSZX and
inputFile.DET.CHIP1.PSZY==DET.CHIP1.PSZY and
inputFile.INS.COLL.NAME==INS.COLL.NAME and
inputFile.INSTRUME==INSTRUME;
```

replacing

```
inputFile.INS.GRIS1.ID==INS.GRIS1.ID
```

by

```
inputFile.INS.GRIS1.NAME==INS.GRIS1.NAME
```

While the old and new grisms should be very similar there is, however, no guarantee and you should check your data carefully.

2. **Some of the slitlets in my science data look badly traced**

   This is most likely caused by an imperfect tracing of the slitlets in the calibration data. Usually an incompletely traced slit can well be seen in normalized master flat field. Because the pipeline detects the slit edges first in the arc lamp frame a probable culprit is the `peakdetection` recipe parameter of the `fors_calib` recipe, which may be too low or too high.

3. **My long-slit spectra show curved sky lines**

   This is caused by a bad wavelength calibration. Most likely the recipe parameter `peakdetection` in the `fors_calib` recipe was set too low so that spurious lines were detected.

4. **My reduced spectra do not cover the wavelength range I expected**

   Some low-resolution grisms (e.g., GRIS_300V) are subject to second order contamination, even if used with an order separation filter (e.g., GG435). The pipeline will limit the wavelength range of the reduced spectra to the range not affected by the second order contamination (=twice the bluest wavelength present in the observed data).

   Users can adjust the wavelength range following the steps below:

   (a) If the **old response determination** was used (recipe `fors_science`).
   One can change the wavelength range in the interactive window of the `fors_calib` actor, by editing the `startwavelength` and/or `endwavelength` values of the file classified as "GRISM_TABLE", or by specifying these parameters on the command line.

   (b) If the **new response determination** was used (recipe `fors_response`).
   The response is computed by using fit-points defined in the table with the tag `FIT_POINT_CATALOG`, and they cover only the wavelength region unaffected by second order contamination. To extend the wavelength range it is therefore necessary to edit the table before starting the reduction and insert extra fit points according to the needs.

When extending the wavelength range for the flux standard star (to get flux-calibrated spectra across the extended wavelength range) please note that many flux standard stars are hot stars, for which the second order contamination may distort the response shape.

Note also, that if master response curves are used, only the wavelength range without second order contamination is covered.

5. **Why do I see blank (flux=0) regions in the left and right side of my mapped files?**

   The nominal wavelength range used by default by the pipeline has been chosen to allow covering the registered range in the CCD for slits placed both to the very left or to the very right. For high resolution grisms and no filter (Free), the registered wavelength range in the CCD is in fact smaller than the default values, therefore the pipeline pads with 0 values to complete up to nominal range.

6. **Although I am experimenting with different flux standard stars for my observation the flux calibrated spectra do not change. Why?**

   If your dataset contains a master response curve (MASTER_SPECPHOT_TABLE) the pipeline will use that one instead of the response curves derived from individual standard stars. In order to see the effect of the different standard stars you have to remove the master response curve from the input data pool.

7. **There is no master response curve in my datasets - why?**

   There are currently no master response curves for the grisms GRIS_1200g, GRIS_200I+OG590, GRIS_600I+OG590, GRIS_600R+GG435, GRIS_600V+GG435, nor for any spectra observed before January 1, 2015.

8. **There is no flux standard star for my LSS observations - why?**

   calSelector delivers for LSS data only those flux standard stars that were observed at the same position on the detector as your long-slit spectrum to ensure the same wavelength coverage. In earlier times this rule was not followed strictly and sometimes the standard star was observed at the center of the field-of-view instead. You can look for observations at the instrument specific query forms for FORS2 and FORS1 by setting the following constraints:

   `Start` The date of your observations - 5 days
   `End` The date of your observations + 5 days
   `DPR.CATG` `CALIB`
   `INS.MODE` `MOS`
   `INS.FILT1.NAME` the order-separating filter of your observations (use `Any` if you had no filter)
   `INS.GRIS1.NAME` The grism of your observations

   and selecting for display in addition

   `DP.ID`
   `OBS.TARG.NAME`
   `DPR.TYPE` to identify the standard star (`STD`) and its calibrations (`FLAT,LAMP` and `WAVE,LAMP`)
   `DET.CHIP1.ID` to identify the correct chip for mosaic data
   `SEQ.SPEC.TARG` position at which a MOS standard star has been taken

For FORS1 data after April 1, 2007 and FORS2 data after April 1, 2002, you will always find 2 files per observations. You need the ones with `DET.CHIP1.ID` = `CCID20-14-5-3` or `Norma III`.

9. **Part of my flux calibrated spectrum has flux=-1. Why?**

   The response curve may cover a different wavelength region than the science spectrum. Pixels not covered by the response curve are set to -1 in the flux-calibrated data.

10. **If I click on "Use the parameters above as initial values in subsequent executions of this recipe", some changes are ignored**

    The FORS pipeline has a special way to handle the parameters which are in the grism tables. Basically, the initial values are always taken from the grism table. Still, during the loop execution, the parameters can be changed to the desired values.

    If you want to use different values as "initial values" for those parameters, edit the grism tables (`FORS2_GRS_*`) to create your own version. The paremeters affected are: `dispersion`, `peakdetection`, `wdegree`, `cdegree`, `reference`, `startwavelength`, `endwavelength`, `resp_use_flat_sed`, `resp_fit_degree`, `resp_fit_nknots`, and `dradius_aver`.

    Please note that the grism tables can be found in the `CALIB_DIR` and that they may also be present in the `RAW_DATA_DIR`. In the latter case the files will have names like `M.FORS2*`.

11. **I am using the maximum number of knots for the spline fit of the response but the fit still does not go through all data points.**

    The maximum number of knots for the spline is the number of unmasked data points – 2 (entering a higher number will cause the pipeline to reduce it to the allowed maximum). Since the knots are distributed at equal distances this means that the distance between two knots is always larger than the distance between two data points. This explains why even at a maximum number of knots the fit may not go through all data points. A polynomial of very high degree might achieve that, but is rather unstable.

12. **The fit to the response curve shows very strong deviations in the masked regions - how can I avoid that?**

    Some grisms have a very steep slope and/or cover small-scale variations of the detector response so that a spline fit with a high number of knots is needed to fit their response. This fit is unconstrained in the masked regions, which can result in very strong deviations. The masking regions have been defined very conservatively. It may well be that the response in the masked regions does not show any significant deviations or only at very few points. In such a case set `resp_ignore_mode` to `command_line` only and add the deviating points at `resp_ignore_points`. Alternatively you can try to exclude the steep part of the response via `resp_ignore_points` and/or use a polymonial fit of lower order, that will not take into account small-scale variations of the response (via `resp_fit_degree` instead of `resp_fit_nknots`).

13. **I used `-use_flat_sed`=true for my `GRIS_1200B` data but the flux-calibrated data look very bad.**

    Unfortunately the spectral energy distribution of the flat field lamp use for `GRIS_1200B` data can vary considerably on short timescales. In such cases the `FLAT_SED_<mode>` used to correct the effect of the position-dependent response will differ even if they were taken at the same position on the CCD.

So it will be impossible to correct the effect. If you have LSS data it is very likely that the flux standard star was observed at the same position on the CCD as your science target, so that you may use `-use_flat_sed=false`. In the case of MOS/MXU data there is unfortuntaely no solution for this problem. With `-use_flat_sed`=true you can correct the small scale variations of the detector response at the price of introducing spurious flux variations. Setting `-use_flat_sed`=false, however, will not correct for the position dependency of the response, which can severely distort the SED of your targets, up to reversing the slope of the spectra (i.e. making a hot spectrum look cool and vice versa).

14. **There are no extracted spectra - why?**

    While this may happen for observations of extremely faint targets the most common case for this behaviour are LSS observations from the lower chip, as the main target is usually placed close to the centre of the field-of-view, which is on the upper chip.

15. **The coordinates reported by the pipeline for my extracted spectra show offsets - why?**

    There can be several causes for such offsets:

    (a) Imprecise coordinates for the guide star, which introduce an offset in the world coordinate system recorded in the header of the raw data. This effect causes a rigid shift in right ascension and/or declination.

    (b) An incorrect value of the reference wavelength recorded in the keyword `HIERARCH ESO INS GRIS1 WLEN`. The pipeline assumes that this keyword gives the value of the undeviated wavelength for the corresponding grism. An incorrect value will introduce an offset of

    $$\frac{\texttt{INS GRIS1 WLEN} - \lambda_{true}}{\text{step\_size}} \cdot \text{pixel\_scale}$$

    An offset 10.5 Å for `GRIS_1200B` for instance corresponds to 15 steps [0.7Å/px] in wavelength and thus – for binning 2×2 and standard collimator – to an offset of 15 px·0.25″/px, i.e 3.8″, on sky. Depending on the angle on sky with which you data were taken this offset along the dispersion axis will affect right ascension and/or declination.

    (c) An incorrect wavelength calibration. Errors in the wavelength calibration should be below 1 pixel and thus sually not be noticeable in the derived coordinates. For multi-object spectroscopic data, however, such errors might introduce small shifts between spectra from different slits, which may again affect right ascension and/or declination, depending on the angle on sky of the observations.

16. **The spectra in my rectified LSS data are tilted - why?**

    In rare cases the use of the `GLOBAl_DISTORION_TABLE` results in tilted instead of horizontal spectra in the `MAPPED...` data. We are still looking into the causes for this behaviour, so please report the problem (and the data for which you see it) by opening a ticket at https://support.eso.org.

17. **Why do my flux-calibrated data not agree with independent photometric measurements?**

    In order to have a true absolute flux calibration several requirements need to be fulfilled:

    (a) The fraction of the total flux of an object that is contained in the slit depends on the shape of the object, the width and orientation of the slit, and the seeing. Absolute flux calibration requires that all the flux of both the object and the standard star has been collected.

(b) The flux that arrives at the telescope depends on the transparency of the sky. Absolute flux calibration requires the same transparency for the observations of the target and the standard star.

A change between the flux standard observation and the science object observations of any of the parameters mentioned above will change the flux scale in the final spectrum. To compare the flux calibrated spectrum to other measurements, differences in slit losses and atmospheric conditions have to be taken into account.

With respect to the faction of flux contained within the slit one should keep in mind that the flux standard stars are observed with a 5" wide slit, while science data are typically observed with slit widths of 0.8" to 1". For a point source a slit width of 0.8"/1.0" used with a seeing of 0.8" means that some 33%/24%, respectively, of the target flux are lost (see also Fig. 17). This results in a too low flux for the flux calibrated spectrum of the target.

If the standard star or the target or both are observed under non-photometric conditions (e.g. CLR or THN) their observed flux will be lower than it should be. If the standard star is observed under photometric conditions but the science target is not the flux in the flux calibrated target spectrum will be too low. The opposite happens if the target is observed under photometric conditions but the standard star is not. CLR/THN conditions allow for transparency variations of 10%/20%, respectively.

18. **Does the pipeline combine different detectors chips into a common product?**

No. The spectroscopic pipeline and Reflex workflow works only on files from the same detector chip. Files from different detectors must not be mixed in the same sof.

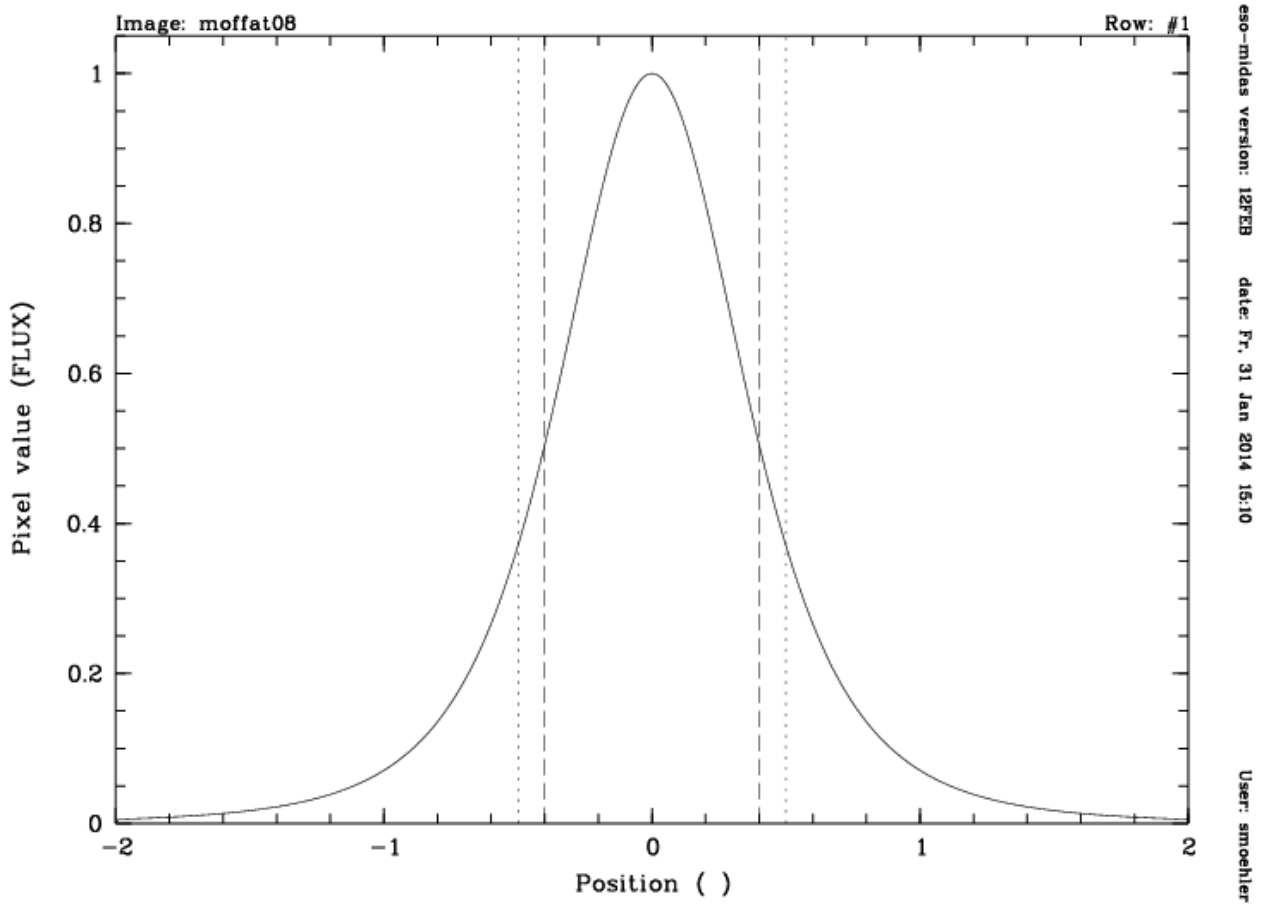


Figure 11.1: *Here a Moffat profile with a FWHM of 0.8" is shown. The dashed and dotted lines mark the limits of a 0.8"/1.0" slit..*

12 Troubleshooting

12.1 Reflex

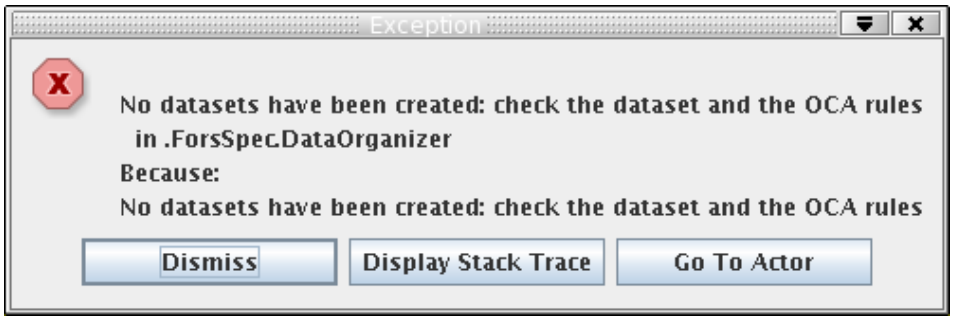


Figure 12.1: *TheDataOrganizer interactive window reports an error ":No DataSets have been created, check the data set and the OCA rules.".*

1. **I downloaded the data from the ESO archive, put them into a new directory, tried to run `Reflex` on them, but the DataOrganiser fails with the error message ":No DataSets have been created, check the data set and the OCA rules."(see Figure 12.1.)**

   This error may be due to the fact that the data provided by the ESO archive are compressed (`<filename>.fits.Z`). Please remember to uncompress the data before running the workflow in `Reflex`.

   Also, please remember that the FORS2 workflow supports only spectroscopic data (LSS/MOS/MXU, no PMOS). It is possible that your data consists entirely of IMG/IPOL/PMOS observations, in which case the `Data Organiser` actor will not construct any DataSets, showing the mentioned error message.

2. **The "Select DataSets" window displays my DataSets, but some/all of them are greyed out. What is going on?**

   If a DataSet in the "Select DataSets" window is greyed out, then it means that the DataSet which was constructed is missing some key calibration(s) (i.e. the DataSet is incomplete). To find out what calibration(s) are missing from a greyed out DataSet, click on the DataSet in question to highlight it in blue, and then click on the button `Inspect Highlighted`. The "Select Frames" window that appears will report the category of the calibration products that are missing (e.g. `MASTER_BIAS`). From this the user has then to determine the missing raw data (in this case bias frames). If static calibrations are missing the mechanism unfortunately does not work, but such data should be found by `reflex` in `<install_directory>/calib/<pipeline_version>/cal`

12.2 FORS2

fors_calib

3. **Few slitlets** For MOS/MXU observations with few (usually three or fewer) slitlets the slit identification

may fail. Using `slit_ident=FALSE` allows the user to process such data.

4. **The pipeline fails with the error message "The wavelength solution at row <number> does not increase monotonically, which is physically impossible. Try with new parameters"**.

   Non-monotonic dispersion relations are often due to spurious detections. In such cases try to decrease `wdegree` and/or increase `peakdetection`. A further possibility is to decrease `wradius`.

5. **The pipeline fails for wide slit data**

   If very wide slits are used, the reference lines become accordingly wider (and display a box-like, flat-top profile). The calibration recipe can handle this in case of well isolated lines, but if nearby lines blend together it is impossible to safely determine their positions. There is no solution for such data.

6. **The pipeline fails with error "No slits could be detected".**

   The recipe cannot detect the slit edges with default value of `peakdetection`, stored in the GRISM_TABLE. In such cases try to change `peakdetection`.

fors_science (Response)

7. **If I set parameter `resp_ignore_mode` to empty in the response interactive window, the value `stellar_absorption,telluric,command_line` is taken**

   This is a known bug in Reflex that will be solved in next release. As a work-around, set `resp_ignore_mode` to `command_line` and `resp_ignore_points` to empty, in order to express that no masking has to be done.

8. **What can I do if I get a problem with fors_calib recipe stating that there is no good wavelength solution for a list?**

   This is a side effect of a wrong slit position determination. In fact, this problem occurs during the refinement of the slit positions if slit identification has been performed. The problem will likely be addressed in future releases. For a workaround in the meantime, try running the pipeline with parameter `-slit_ident` set to `false`.

Acknowledgements

The `Reflex` team in alphabetical order consists of Pascal Ballester, Lodovico Coccato, Vincenzo Forchì, Wolfram Freudling, César Enrique García, Maurice Klein Gebbinck, Martino Romaniello.

Forchì V., 2012, Reflex User Manual, VLT-MAN-ESO-19000-5037, Issue 0.7, *https://ftp.eso.org/pub/dfs/reflex/ReflexUserManual-3.1.pdf* 25, 30, 64, 65

FORS User Manual, VLT-MAN-ESO-13100-1543
https://www.eso.org/sci/facilities/paranal/instruments/fors/doc.html 15

FORS Pipeline User Manual, VLT-MAN-ESO-19500-4106
https://www.eso.org/sci/software/pipelines/index.html 12, 14, 15, 30, 38, 39