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

FORS Pipeline User Manual

VLT-MAN-ESO-19500-4106

Issue 2.0

Date 2008-09-09

Prepared:	C. Izzo, J. M. Larsen	2008-09-09	
	Name	Date	Signature
Approved:	P.Ballester		
	Name	Date	Signature
Released:	M. Peron		
	Name	Date	Signature

This page was intentionally left blank

Change record

Issue/Rev.	Date	Section/Parag. affected	Reason/Initiation/Documents/Remarks
1.0	10/10/2006	All	First version
1.1	04/04/2007	Sections 3, 7.5, and Appendix	FORS1 blue mosaic upgrade
1.2	05/05/2007	Minor changes in Sections 1, 4, and 3	Public release
2.0	09/09/2008	All	Documenting new imaging pipeline

This page was intentionally left blank

Contents

1 Introduction

1.1 Purpose

The FORS pipeline is a subsystem of the *VLT Data Flow System* (DFS). It is used in two operational environements, for the ESO *Data Flow Operations* (DFO), and for the *Paranal Science Operations* (PSO), in the quick-look assessment of data, in the generation of master calibration data, in the reduction of scientific exposures, and in the data quality control. Additionally, the FORS pipeline recipes are made public to the user community, to allow a more personalised processing of the data from the instrument. The purpose of this document is to describe a typical FORS data reduction sequence with the FORS pipeline.

This manual is a complete description of the imaging and spectroscopic data reduction recipes reflecting the status of the FORS pipeline as of 09.09.2008 (version 4.3.5). In this release the direct imaging reduction recipes are offered for the first time, replacing the old (MIDAS-based) pipeline.

FORS1 polarimetric and FORS2 echelle data reduction are still unsupported.

1.2 Acknowledgements

The FORS pipeline is based on the CPL developed by the ESO/DFS.

The contribution of several people was essential for the definition and the implementation of the pipeline, together with its application to the reduction of FORS1 and FORS2 data.

Sabine Moehler (ESO-DFO) extensively tested the data reduction procedures by carefully examining their products. Her patient work and great insight have been vital to this project.

Discussions with Harald Kuntschner (ST-ECF), Martino Romaniello (ESO-DFO), Pascal Ballester (ESO-DFS), Marguerite Pierre (CEA, Saclay, France), Christophe Adami (LAM, Marseille, France), Stefano Cristiani (INAF – Osservatorio Astronomico di Trieste), have been an unvaluable and continuous source of useful ideas for improving the spectroscopy related recipes.

Emanuela Pompei, Kieran O'Brien, Emmanuel Jehin, Stefano Bagnulo, and Gianni Marconi (ESO Paranal Observatory), also offered suggestions for promptly addressing the problems of automatic data reduction.

1.3 Scope and references

This document describes the FORS pipeline used at ESO-Garching and ESO-Paranal for the purpose of data assessment and data quality control.

Updated versions of the present document may be found on [16]. For the work of data reduction, it is very useful to read also [11]. For general information about the current instrument pipelines status see [3]. Quality control information are at [4].

Additional information on the Common Pipeline Library (CPL), *Esorex* and *Gasgano* can be found at [1], [2], [5], [14], and [15]. A description of the instrument is in [7]. The FORS instrument user manuals [6], [11], [12], and [13], can be found in http://www.eso.org/instruments/fors/doc/.

[1] *Common Pipeline Library User Manual*. VLT-MAN-ESO-19500-2720.

[2] *CPL home page*. http://www.eso.org/cpl/.

[3] *Current pipeline status*. http://www.eso.org/observing/dfo/quality/pipeline-status.html.

[4] *ESO-Data Flow Operation home page*. http://www.eso.org/observing/dfo/quality/.

[5] *ESOREX home page*. http://www.eso.org/cpl/esorex.html.

[6] *FIMS User Manual*. VLT-MAN-ESO-13100-2308.

[7] *FORS instrument home page*. http://www.eso.org/instruments/fors.

[8] *FORS1 and FORS2 Instruments Description*. http://www.eso.org/sci/facilities/paranal/instruments/fors/inst/.

[9] *FORS1 and FORS2 Overview*. http://www.eso.org/sci/facilities/paranal/instruments/fors/overview.html.

[10] *FORS1 blue upgrade*. [8]/FactSheet/fors1_E2V_upgrade.html.

[11] *FORS1+2 Data Reduction Cookbook*. VLT-MAN-ESO-13100-4030.

[12] *FORS1+2 Template User Manual*. VLT-MAN-ESO-13100-2309.

[13] *FORS1+2 User Manual*. VLT-MAN-ESO-13100-1543.

[14] *GASGANO home page*. http://www.eso.org/gasgano.

[15] *Gasgano User's Manual*. VLT-PRO-ESO-19000-1932.

[16] *Pipelines Web Page*. http://www.eso.org/pipelines.

[17] E. Bertin. *SExtractor v2.5*. http://astroa.physics.metu.edu.tr/MANUALS/sextractor/sextractor.pdf.

[18] E. Bertin, S. Arnouts. *SExtractor: Software for source extraction*. (1996), A&A Supp. 117, 393-404.

[19] B. W. Holwerda. *Source Extractor for Dummies*. http://astroa.physics.metu.edu.tr/MANUALS/sextractor/Guide2source_extractor.pdf.

[20] K. Horne. *An Optimal Extraction Algorithm for CCD Spectroscopy*. (1986), PASP 98, 609.

[21] A. U. Landolt. $UBVRI$ *Photometric Standard Stars in the Magnitude Range* $11.5 < V < 16.0$ *Around the Celestial Equator*. (1992), AJ 104, 340.

[22] M. Fukugita et al. *The Sloan Digital Sky Survey photometric system*. (1996), AJ 111, 1748.

[23] P. B. Stetson. *Homogeneous Photometry for Star Clusters and Resolved Galaxies. II. Photometric Standard Stars*. (2000), PASP 112, 925-931.

[24] P. B. Stetson. *Stetson Photometric Standard Fields*. http://www3.cadc-ccda.hia-iha.nrc-cnrc.gc.ca/community/STETSON/standards/, August 2007.

[25] W. Freudling, M. Romaniello, F. Patat, P. Møller, E. Jehin, K. O'Brien. *Absolute Photometry with FORS: The FORS Absolute Photometry Project*. VLT-TRE-ESO-13100-4006.

[26] W. H. Press et al. *Numerical Recipes in C*. Second Edition (1992).

2 Overview

In collaboration with instrument consortia, the Data Flow Systems Department (DFS) of the Data Management and Operation Division is implementing data reduction pipelines for the most commonly used VLT/VLTI instrument modes. These data reduction pipelines have the following three main purposes:

Data quality control: pipelines are used to produce the quantitative information necessary to monitor instrument performance.

Master calibration product creation: pipelines are used to produce master calibration products (e.g., combined bias frames, super-flats, wavelength dispersion solutions).

Science product creation: using pipeline-generated master calibration products, science products are produced for the supported instrument modes (e.g., optimally extracted spectra, bias-corrected and flat-fielded images, wavelength-calibrated spectra). The accuracy of the science products is limited by the quality of the available master calibration products and by the algorithmic implementation of the pipelines themselves. In particular, adopted automatic reduction strategies may not be suitable or optimal for all scientific goals.

Instrument pipelines consist of a set of data processing modules that can be called from opportune front-end applications, such as the automatic data management tools available on Paranal.

ESO offers two front-end applications for launching pipeline recipes, *Gasgano* [15] and *Esorex* [5], both included in the pipeline distribution. These applications can also be downloaded separately from the ESO web pages (see [14] and [5]). An illustrated introduction to Gasgano is provided in the "Quick Start" Section of this manual (see page 15).

The FORS1 and FORS2 instruments and the different types of raw frames and auxiliary data are described in Sections 3, 6, and 7.

A brief introduction to the usage of the available reduction recipes using *Gasgano* or *Esorex* is presented in Section 4, and in Section 5 known data reduction problems are listed, providing also possible solutions; but it is strongly suggested to read also the pipeline related sections in the FORS1+2 Data Reduction Cookbook [11], which goes even deeper into that.

More details on what are inputs, products, quality control measured quantities, and controlling parameters of each recipe is given in Section 9.

More detailed descriptions of the data reduction algorithms used by the individual pipeline recipes can be found in Section 10.

In Appendix A a list of used abbreviations and acronyms is given.

3 FORS instruments description

The FORS instruments have been developed under ESO contract by the Landessternwarte Heidelberg, the University Observatory of Göttingen, and the University Observatory of Munich.

FORS1 and FORS2 have been made available to the community and started operations in Paranal respectively on April 1^{st}, 1999, and April 1^{st}, 2000. The new mosaic CCD detector for FORS2 is used since March 22, 2002. The new blue mosaic CCD detector for FORS1 is used since April 7, 2007.

Only a brief description of the instruments is given here. For more details please refer to [8] and [9].

3.1 Instruments overview

FORS is the visual and near UV **FO**cal **R**educer and low dispersion **S**pectrograph for the VLT.

FORS1 is a multi mode (imaging, polarimetry, long slit and multiobject spectroscopy) optical instrument placed at the UT2 Cassegrain focus. FORS1 works in the wavelength range 3300-11000 Ångstrom. Two different magnifications can be used with pixel scales of 0.1"/pixel (with the High Resolution collimator) and 0.2"/pixel (with the Standard Resolution collimator), on a 2k x 2k Site detector with 24 μm pixels. The corresponding field sizes are 3.4' x 3.4' and 6.8' x 6.8' respectively. The two different magnifications are chosen by selecting one of two different collimators, hence each magnification has to be calibrated independently.

A new CCD mosaic with blue optimised E2V detectors for FORS1 has been successfully commissioned the first week of April 2007. The new detector system consists of two 2k x 4k CCDs (15 μm pixel size). With respect to the FORS2 MIT mosaic, the E2V CCDs provide much higher response in the blue and UV wavelength range below 6000 Ångstrom, but suffers from strong fringing above 6500 Ångstrom. The image scale in the default readout mode (2x2 binning) is 0.125"/pixel in the high resolution (HR) mode and 0.25"/pixel in the standard resolution (SR) mode. The field of view in these two modes is, respectively, 4.25' x 4.25' and 6.8' x 6.8' (note that the detector area is larger than the field of view). An increased wavelength coverage is achieved in the spectroscopic modes, thanks to the larger CCD and the more flexibile mask preparation for multiobject spectroscopy. Unbinned CCD readout modes are only offered for applications that specifically require it, and that are therefore explicitly requested in the proposal. Please find more detail in [10].

FORS2 is offered at UT1 with a detector consisting of a mosaic of two 2k x 4k MIT CCD (15 μm pixels). The FORS2 mosaic provides greatly improved red sensitivity. The geometrical properties are the same as for the new FORS1 CCD mosaic.

Apart from its detector system, FORS2 is mostly identical to FORS1. The important differences are:

- FORS2 has no polarimetric capability.
- FORS2 offers an extended set of high-throughput volume phased holographic grisms (FORS1 offers only the 1200B grism).
- FORS2 offers, in addition to the MOS unit of movable slitlets, a mask exchange unit (MXU) which can accommodate up to 10 exchangeable slit masks for multiobject spectroscopy with approximately 80 slits each.

- FORS2 offers High Time resolution (HIT) mode in imaging and spectroscopy.

4 Quick start

This section describes the most immediate usage of the FORS pipeline recipes. For a complete list of the available recipes, please see Section 8, page 50.

4.1 An introduction to Gasgano and Esorex

Before being able to apply pipeline recipes to a set of data, the data must be opportunely classified, and associated with the appropriate calibrations. The *Data Classification* consists of tasks such as: "What kind of data am I?", e.g., BIAS, "To which group do I belong?", e.g., to a particular Observation Block or template. *Data Association* is the process of selecting appropriate calibration data for the reduction of a set of raw science frames. Typically, a set of frames can be associated if they share a number of properties, such as instrument and detector configuration. As all the required information is stored in the FITS headers, data association is based on a set of keywords (called "association keywords") and is specific to each type of calibration.

The process of data classification and association is known as data organisation. The *DO Category* is the label assigned to a data type as a result of data classification.

An instrument pipeline consists of a set of data processing modules that can be called from different host applications, either from the command line with *Esorex* [5], from the automatic data management tools available at Paranal, or from the graphical *Gasgano* tool [14].

Gasgano is a data management tool that simplifies the data organisation process, offering automatic data classification and making the data association easier (*even if automatic association of frames is not yet provided*). *Gasgano* determines the classification of a file by applying an instrument specific rule, while users must provide this information to the recipes when they are executed manually using *Esorex* from the command line. In addition, *Gasgano* allows the user to execute directly the pipeline recipes on a set of selected files.

4.1.1 Using Gasgano

To get familiar with the FORS pipeline recipes and their usage, it is advisable to begin with *Gasgano*, because it provides a complete graphic interface for data browsing, classification and association, and offers several other utilities such as easy access to recipes documentation and preferred data display tools.

Gasgano can be started from the system prompt in the following way:

```
gasgano &
```

The *Gasgano* main window will appear. On Figure 4.1.1 (page 17), a view on a set of FORS2 MXU data is shown as an example. *Gasgano* can be pointed to the directories where the data to be handled are located using the navigation panels accessible via the *Add/Remove Files* entry of the *File* menu (shown on the upper left of the figure).

The data are hierarchically organised as preferred by the user. After each file name are shown the observation date, the classification, the target of the observation, and the grism/filter combination that was used.

More information about a single frame can be obtained by clicking on its name: the corresponding FITS file header will be displayed on the bottom panel, where specific keywords can be opportunely filtered and searched. Images and tables may be easily displayed using the viewers specified in the appropriate *Preferences* fields.

Frames can be selected from the main window for being processed by the appropriate recipe: on Figure 4.1.2, an MXU arc lamp exposure with a sequence of flat field exposures, bias frames, a catalogue of reference lines, and a configuration table, are all selected and sent to the *fors_calib* recipe. This will open a *Gasgano* recipe execution window (see Figure 4.1.3), having all the specified files listed in its *Input Frames* panel.

Help about the recipe may be obtained from the *Help* menu. Before launching the recipe, its configuration may be opportunely modified on the *Parameters* panel (on top). The window contents might be saved for later use by selecting the *Save Current Settings* entry from the *File* menu, as shown in figure.

At this point the recipe can be launched by pressing the *Execute* button. Messages from the running recipe will appear on the *Log Messages* panel at bottom, and in case of successful completion the products will be listed on the *Output Frames* panel, where they can be easily viewed and located back on the Gasgano main window.

Please refer to the *Gasgano User's Manual* [15] for a more complete description of the *Gasgano* interface. See also [14].

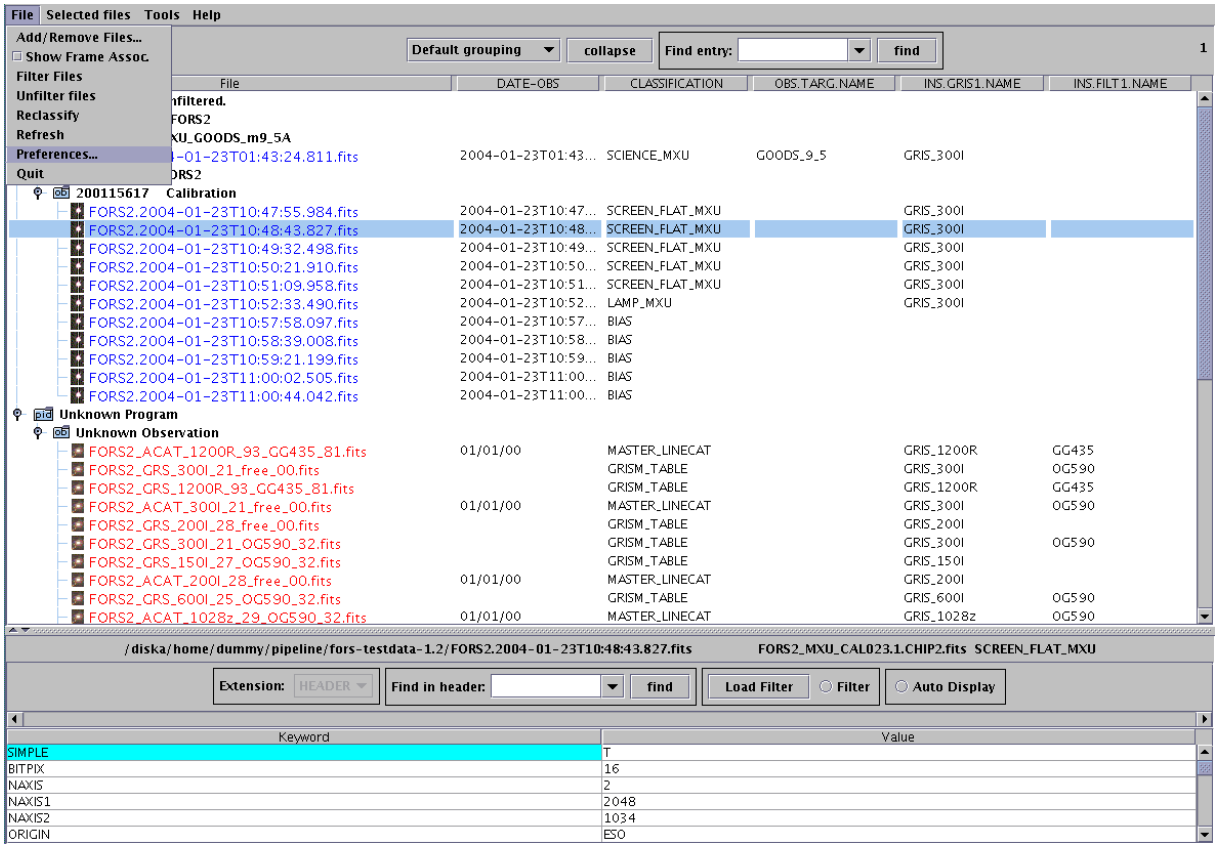

Figure 4.1.1: *The Gasgano main window.*

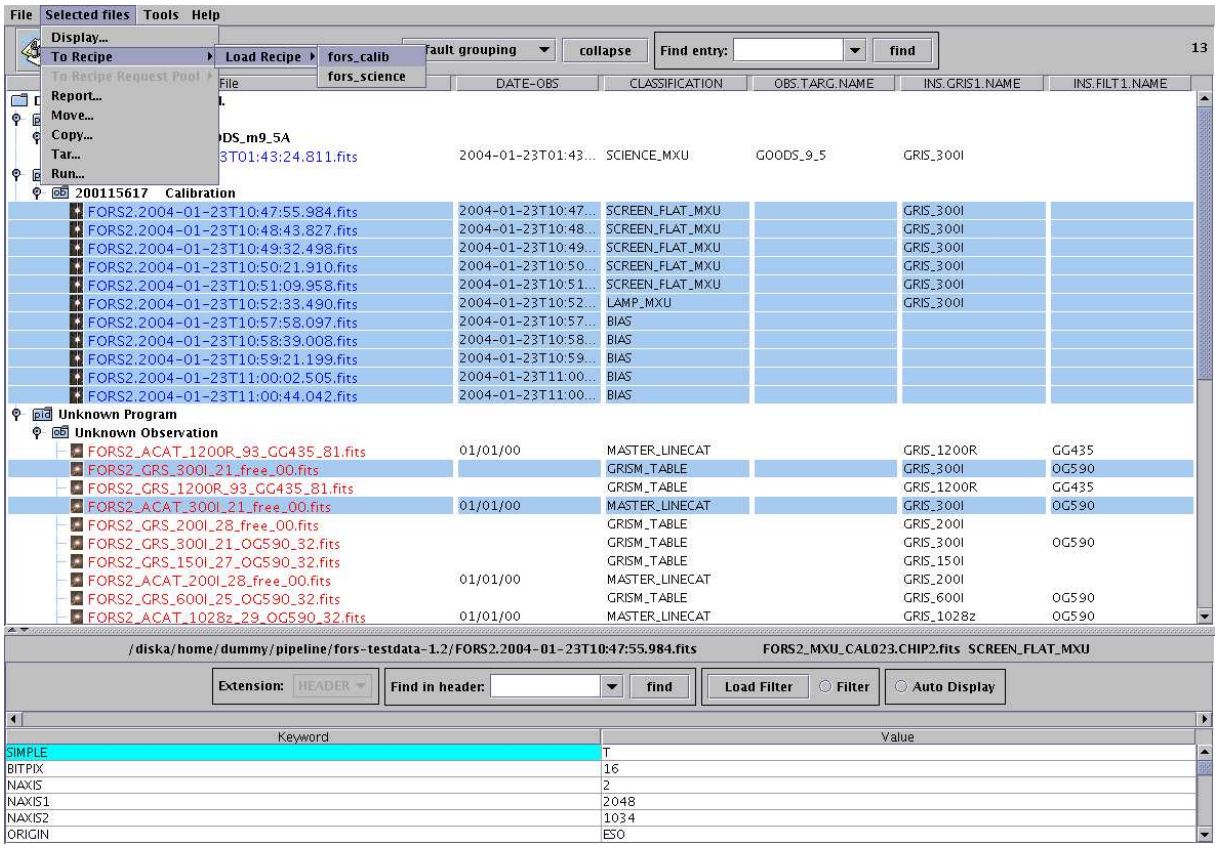

Figure 4.1.2: *Selecting files to be processed by a FORS pipeline recipe.*

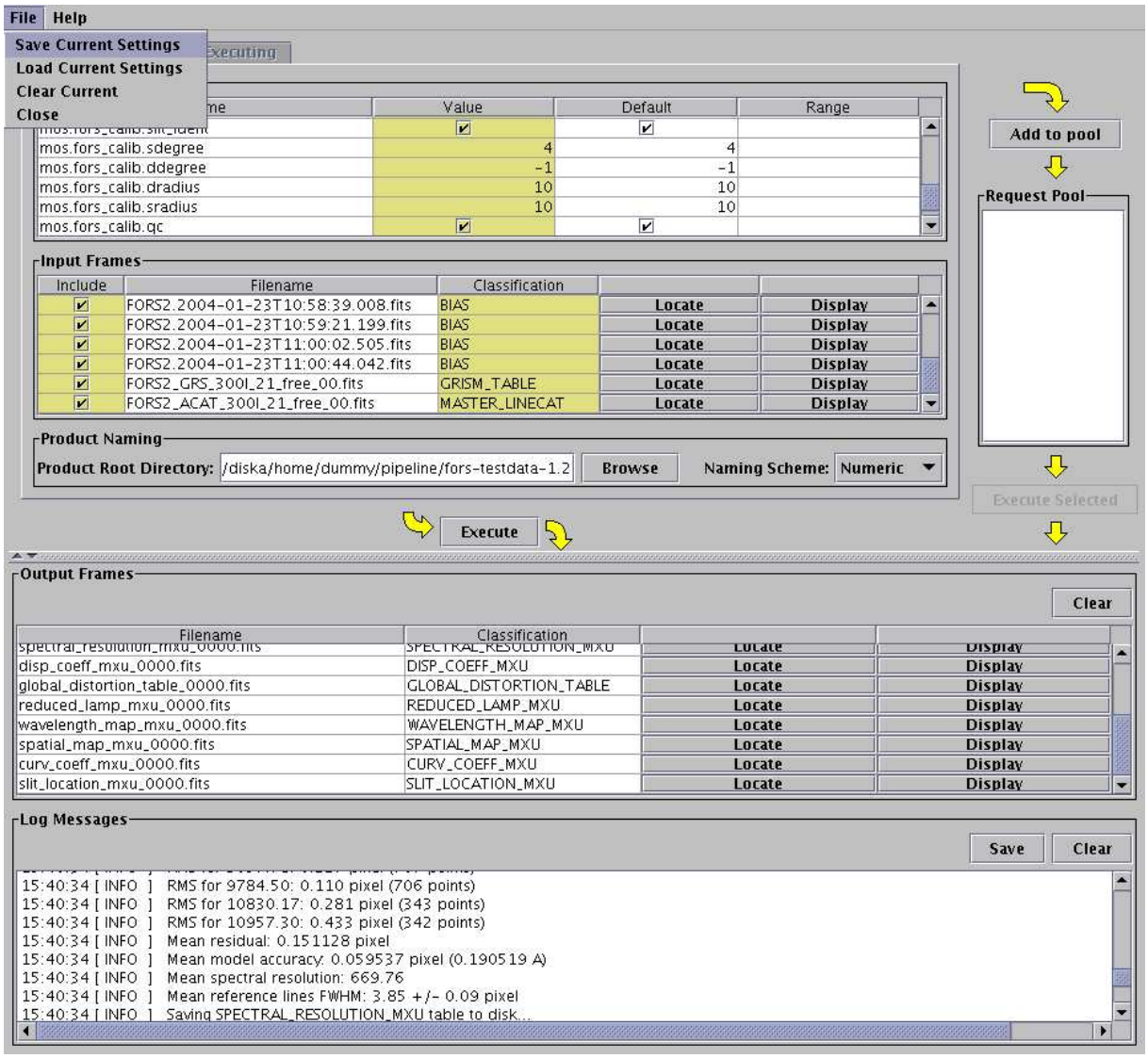

Figure 4.1.3: *The Gasgano recipe execution window.*

4.1.2 Using Esorex

Esorex is a command line utility for running pipeline recipes. It may be embedded by users into data reduction scripts for the automation of processing tasks. On the other side, *Esorex* does not offer all the facilities available with *Gasgano*, and the user must classify and associate the data using the information contained in the FITS header keywords (see Section 6.1, page 35). The user should also take care of defining the input set-of-frames and the appropriate configuration parameters for each recipe run:

The set-of-frames: Each pipeline recipe is run on a set of input FITS data files. When using *Esorex* the filenames must be listed together with their DO category in an ASCII file, the *set-of-frames* (SOF), that is required when launching a recipe. [1]

Here is an example of SOF, valid for the *fors_calib* recipe:

```
FORS1.2006-05-10T12:58:27.122.fits SCREEN_FLAT_MOS
FORS1.2006-05-10T12:59:45.326.fits SCREEN_FLAT_MOS
FORS1.2006-05-10T13:00:20.930.fits SCREEN_FLAT_MOS
FORS1.2006-05-10T13:01:17.711.fits SCREEN_FLAT_MOS
FORS1.2006-05-10T13:02:14.559.fits SCREEN_FLAT_MOS
FORS1.2006-05-10T13:03:37.926.fits LAMP_MOS
../../cal/FORS1_MBIAS.fits MASTER_BIAS
../../cal/FORS1_ACAT_300I_11_OG590_72.fits MASTER_LINECAT
```

This file contains the name of each input frame, and its DO category. The launched pipeline recipe will access the listed files when required by the reduction algorithm.

Note that the FORS pipeline recipes do not verify in any way the correctness of the *DO Category* specified by the user in the SOF. The reason of this lack of control is that the FORS recipes are just the DRS component of the complete pipeline running on Paranal, where the task of data classification and association is carried out by separate applications. Moreover, using *Gasgano* as an interface to the pipeline recipes will always ensure a correct classification of all the data frames, assigning the appropriate DO category to each one of them (see Section 4.1.1, page 15).

A recipe handling an incorrect SOF may stop or display unclear error messages at best. In the worst cases, the recipe would apparently run without any problem, producing results that may look reasonable, but are actually flawed.

Esorex syntax: The basic syntax to use *Esorex* is the following:

```
esorex [esorex_options] recipe_name [recipe_options] set_of_frames
```

To get more information on how to customise *Esorex* (see also [5]) run the commands:

```
esorex -man
esorex -help
esorex -par
```

[1] The set-of-frames corresponds to the *Input Frames* panel of the *Gasgano* recipe execution window (see Figure 4.1.3, page 19).

To generate a configuration file esorex.rc in the directory $HOME/.esorex run the command:

```
esorex -create-config
```

A list of all available recipes, each with a one-line description, can be obtained using the command:

```
esorex -recipes
```

All recipe parameters (aliases) and their default values can be displayed by the command

```
esorex -params recipe_name
```

To get a brief description of each parameter meaning execute the command:

```
esorex -help recipe_name
```

To get more details about the given recipe use the commands:

```
esorex -man recipe_name
esorex -help recipe_name
esorex -par recipe_name
```

Recipe configuration: Each pipeline recipe may be assigned an *Esorex* configuration file, containing the default values of the parameters related to that recipe.[2] The configuration files are normally generated in the directory `$HOME/.esorex`, and have the same name as the recipe to which they are related, with the filename extension `.rc`. For instance, the recipe *fors_calib* has its *Esorex* default configuration file named `fors_calib.rc`, generated with the command:

```
esorex -create-config fors_calib
```

If a number of recipe parameters are specified on the command line, the given values will be used in the created configuration file.

The definition of one parameter in the configuration file may look like this:

```
# --slit_ident
# Attempt slit identification for MOS or MXU.
mos.fors_calib.slit_ident=TRUE
```

[2]The *Esorex* recipe configuration file corresponds to the *Parameters* panel of the *Gasgano* recipe execution window (see Figure 4.1.3, page 19).

In this example, the parameter `mos.fors_calib.slit_ident` is set to the value `TRUE`. In the configuration file generated by *Esorex*, one or more comment lines are added containing information about the possible values of the parameter, and an alias that could be used as a command line option.

The recipes provided by the FORS pipeline are designed to implement a cascade of macro data reduction steps, each controlled by its own parameters. For this reason and to prevent parameter name clashes it is specified as parameter prefix not only the instrument name but also the name of the step they refer to. Shorter parameter aliases are made available for use on the command line.

A description of the recipe parameters is provided in Section 9, page 59.

Recipe execution: A recipe can be run by specifying its name to *Esorex*, together with the name of a set-of-frames. For instance, the following command line would be used to run the recipe *fors_calib* for processing the files specified in the set-of-frames `cal.sof`:

```
esorex fors_calib cal.sof
```

The recipe parameters can be modifyed either by editing directly the used configuration file, or by specifying new parameter values on the command line using the command line options defined for this purpose. Such command line options should be inserted after the recipe name and before the SOF name, and they will supersede the system defaults and/or the configuration file settings. For instance, to set the *fors_calib* recipe *slit_ident* parameter to `false`, the following may be typed:

```
esorex fors_calib -slit_ident=false cal.sof
```

For more information on *Esorex*, see [5].

4.2 Example of imaging data reduction using Esorex

In the following, a typical FORS1 imaging data reduction procedure is described.[3] It is assumed that the following data are available:

One scientific exposure:

```
FORS1.2007-08-30T23:41:28.745.fits          SCIENCE_IMG
```

Four sky field exposures:

```
FORS1.2007-08-30T23:01:46.059.fits          SKY_FLAT_IMG
FORS1.2007-08-30T23:02:22.453.fits          SKY_FLAT_IMG
FORS1.2007-08-30T23:02:59.897.fits          SKY_FLAT_IMG
FORS1.2007-08-30T23:03:38.441.fits          SKY_FLAT_IMG
```

[3] The procedure using *Gasgano* is conceptually identical.

One standard star field exposure:

```
FORS1.2007-08-30T23:19:47.455.fits          STANDARD_IMG
```

Five bias exposures:

```
FORS1.2007-08-30T10:03:01.111.fits          BIAS
FORS1.2007-08-30T10:03:29.714.fits          BIAS
FORS1.2007-08-30T10:03:56.637.fits          BIAS
FORS1.2007-08-30T10:04:24.850.fits          BIAS
FORS1.2007-08-30T10:04:51.253.fits          BIAS
```

All the listed data are meant to be obtained from the same FORS1 chip, with the same filter in use, same binning, readout method, etc.. In this example it is assumed that filter R_BESS is in use on chip "Norma III" (this is important for the association of the appropriate static calibration tables to the raw input data to be processed).

In the following, it is also assumed for simplicity that, in the *Esorex* configuration file, the flag *suppress-prefix* is set to `TRUE`, so that the product file names will just be identical to their product categories, with an extension `.fits`. Moreover, it is assumed that all the handled files (inputs and products) are located in the current directory. The only exception is represented by the standard calibration tables (e.g., line catalogues), that here are assumed to be located under `/cal/fors1`.

In order to produce a master bias calibration, the recipe *fors_bias* should be used (see Section 9.1, page 59). The input SOF may be defined as follows:

File: *bias.sof*

```
FORS1.2007-08-30T10:03:01.111.fits          BIAS
FORS1.2007-08-30T10:03:29.714.fits          BIAS
FORS1.2007-08-30T10:03:56.637.fits          BIAS
FORS1.2007-08-30T10:04:24.850.fits          BIAS
FORS1.2007-08-30T10:04:51.253.fits          BIAS
```

This is the list of the raw bias frames meant to be used for the production of the master bias calibration.

The following command line can now be given at the shell prompt:

```
esorex fors_bias bias.sof
```

Just one product is created on disk:

master_bias.fits: master bias calibration.

Different stacking methods are available and may be specified on the command line: the default is to stack frames rejecting the highest and lowest pixel values averaging the rest.

A twilight sky flat master calibration is produced using the recipe *fors_img_sky_flat* (see Section 9.4, page 64). The input SOF may be defined as follows:

File: *flat.sof*

```
FORS1.2007-08-30T23:01:46.059.fits      SKY_FLAT_IMG
FORS1.2007-08-30T23:02:22.453.fits      SKY_FLAT_IMG
FORS1.2007-08-30T23:02:59.897.fits      SKY_FLAT_IMG
FORS1.2007-08-30T23:03:38.441.fits      SKY_FLAT_IMG
master_bias.fits                        MASTER_BIAS
```

This is the list of the twilight sky exposures meant to be used for the production of the flat field calibration, and the bias master calibration previously created with *fors_bias*.

The following command line can now be given at the shell prompt:

```
esorex fors_img_sky_flat flat.sof
```

Just one product is created on disk:

master_sky_flat_img.fits: twilight sky master calibration.

Different stacking methods are available and may be specified on the command line: the default is to median-stack the input frames (after normalisation of each frame to its own median value).

At this point, with a flat and a bias, it is possible to reduce a scientific exposure. This is done with the recipe *fors_img_science* (see Section 9.5, page 64), defining the following input set-of-frames:

File: *sci.sof*

```
FORS1.2007-08-30T23:41:28.745.fits   SCIENCE_IMG
master_bias.fits                     MASTER_BIAS
master_sky_flat_img.fits             MASTER_SKY_FLAT_IMG
/cal/fors1/fors1_Norma_phot.fits     PHOT_TABLE
```

The so-called photometric table (PHOT_TABLE) carries information about the atmospheric extinction and spectral corrections for any of the available standard filters the instrument is equipped with. This recipe uses only the atmospheric extinction, for correcting measured *instrumental* magnitudes of detected objects to airmass zero.

The following command line can now be given at the shell prompt:

```
esorex fors_img_science sci.sof
```

A number of products are created on disk, mainly for check purposes:

science_reduced_img.fits: reduced scientific frame.

phot_background_img.fits: background map.

object_table_sci_img.fits: properties of detected objects.

sources_sci_img.fits: complete source detection output (SExtractor [18]).

If a standard star field exposure is available it is possible to transform the measured instrumental magnitudes into physical magnitudes. The recipe *fors_zeropoint* (see Section 9.6, page 68), with the following input set-of-frames, would produce a zeropoint for the standard star frame (hopefully valid for the scientific frame too):

File: *std.sof*

```
FORS1.2007-08-30T23:19:47.455.fits   STANDARD_IMG
master_bias.fits                     MASTER_BIAS
master_sky_flat_img.fits             MASTER_SKY_FLAT_IMG
/cal/fors1/fors1_Norma_phot.fits     PHOT_TABLE
/cal/fors1/landolt_std_UBVRI.tfits   FLX_STD_IMG
/cal/fors1/stetson_std_BVRI.fits     FLX_STD_IMG
```

The same master and static calibration files used for the scientific data reduction are used here. In addition to that, at least one photometric standard stars catalogue (FLX_STD_IMG) must be specified.

Currently the Landolt and the Stetson catalogues are supported: if both are specified in input, as in this example, the information will be merged into a single table, eliminating common entries by choosing the ones with the smallest magnitude error. It should be noted that using simultaneously two different catalogues carries a significant risk to produce inconsistent results, as the magnitudes are not derived with the same method in both catalogues, and as a consequence common stars can differ by up to 0.2 magnitudes. The use of two catalogues is therefore strongly discouraged.

The following command line will execute the recipe:

```
esorex fors_zeropoint std.sof
```

A number of products are created on disk, mainly for check purposes:

standard_reduced_img.fits: reduced standard field exposure.

phot_background_img.fits: background map.

aligned_phot.fits: properties of detected standard stars.

sources_std_img.fits: complete source detection output (SExtractor [18]).

debug.fits: image with marked detected sources *vs* expected standard stars.

4.3 Example of MOS/MXU data reduction using Esorex

The processing of both MXU and MOS FORS1/2 data is identical: the only difference lays in the suffix, either _MXU or _MOS, assigned to the DO categories of input and output files. It should also be noted that a MOS observation might be performed setting the same offet to all the slitlets: in this specific case, the aligned slitlets are perfectly equivalent to a single long-slit, which is processed in a slightly different way from what is described in this Section. A description of this specific case, namely the processing of LSS and LSS-like observations, is given in the next Section.

In the following, a typical FORS2 MXU data reduction procedure is described.[4]

It is assumed that the following data are available:

One scientific exposure:

```
FORS2.2004-09-27T02:39:11.479.fits          SCIENCE_MXU
```

Three flat field exposures obtained with the mask used for the scientific exposure:

```
FORS2.2004-09-27T18:59:03.641.fits          SCREEN_FLAT_MXU
FORS2.2004-09-27T19:00:07.828.fits          SCREEN_FLAT_MXU
FORS2.2004-09-27T19:01:14.252.fits          SCREEN_FLAT_MXU
```

One arc lamp exposure obtained with the mask used for the scientific exposure:

```
FORS2.2004-09-27T19:13:03.631.fits          LAMP_MXU
```

Five bias exposures:

```
FORS2.2004-09-27T08:00:27.821.fits          BIAS
FORS2.2004-09-27T08:01:05.604.fits          BIAS
FORS2.2004-09-27T08:01:44.091.fits          BIAS
FORS2.2004-09-27T08:02:22.070.fits          BIAS
FORS2.2004-09-27T08:03:01.042.fits          BIAS
```

All the listed data are meant to be obtained from the same FORS2 chip, with the same grism and filter in use. In this example it is assumed that respectively grism 300I and filter OG590 are in use (this is important for the association of the appropriate static calibration tables to the raw input data to be processed).

In the following, it is also assumed for simplicity that, in the *Esorex* configuration file, the flag *suppress-prefix* is set to `TRUE`, so that the product file names will just be identical to their product categories, with an extension `.fits`. Moreover, it is assumed that all the handled files (inputs and products) are located in the current directory. The only exception is represented by the standard calibration tables (e.g., line catalogues), that here are assumed to be located under `/cal/fors2`.

[4] The procedure using *Gasgano* is conceptually identical.

In order to process the calibration exposures available for the scientific observation, the recipe *fors_calib* is used (see Section 9.8, page 76). The input SOF may be defined as follows:

File: *cal.sof*

```
FORS2.2004-09-27T08:00:27.821.fits          BIAS
FORS2.2004-09-27T08:01:05.604.fits          BIAS
FORS2.2004-09-27T08:01:44.091.fits          BIAS
FORS2.2004-09-27T08:02:22.070.fits          BIAS
FORS2.2004-09-27T08:03:01.042.fits          BIAS
FORS2.2004-09-27T18:59:03.641.fits          SCREEN_FLAT_MXU
FORS2.2004-09-27T19:00:07.828.fits          SCREEN_FLAT_MXU
FORS2.2004-09-27T19:01:14.252.fits          SCREEN_FLAT_MXU
FORS2.2004-09-27T19:13:03.631.fits          LAMP_MXU
/cal/fors2/FORS2_ACAT_300I_21_OG590_32.fits MASTER_LINECAT
/cal/fors2/FORS2_GRS_300I_21_OG590_32.fits  GRISM_TABLE
```

The input BIAS frames are used to generate a median bias frame that is internally subtracted from all the input raw images, and eventually written to disk for further use. A master bias frame may also be produced using the recipe *fors_bias* (described in the imaging data reduction Section 4.2), or by other means (taking care of trimming the overscan regions from the final result). This own-produced master bias frame may be specified in input instead of the sequence of raw BIAS frames, using the tag MASTER_BIAS:

File: *cal.sof*

```
master_bias.fits                            MASTER_BIAS
FORS2.2004-09-27T18:59:03.641.fits          SCREEN_FLAT_MXU
FORS2.2004-09-27T19:00:07.828.fits          SCREEN_FLAT_MXU
FORS2.2004-09-27T19:01:14.252.fits          SCREEN_FLAT_MXU
FORS2.2004-09-27T19:13:03.631.fits          LAMP_MXU
/cal/fors2/FORS2_ACAT_300I_21_OG590_32.fits MASTER_LINECAT
/cal/fors2/FORS2_GRS_300I_21_OG590_32.fits  GRISM_TABLE
```

The MASTER_LINECAT and the GRISM_TABLE are static calibration tables that are available in the calibration directories delivered with the pipeline recipes. The file FORS2_ACAT_300I_21_OG590_32.fits is the default catalogue of reference arc lamp lines for grism 300I and filter OG590 of the FORS2 instrument. This catalogue may be replaced with alternative ones provided by the user, if found appropriate.

The FORS2_GRS_300I_21_OG590_32.fits table contains the default *fors_calib* recipe configuration parameters for grism 300I and filter OG590 of the FORS2 instrument. If this file is not specified, appropriate values for the parameters must be set in the command line or in the *Esorex* configuration file.

The following command line can now be given at the shell prompt:

```
esorex fors_calib cal.sof
```

Several products are created on disk, mainly for check purposes. The products which are necessary for the scientific data reduction are the following:

master_bias.fits: master bias frame, produced only in case a sequence of raw BIAS exposures was specified in input.

master_norm_flat_mxu.fits: normalised flat field image.

slit_location_mxu.fits: slit positions on the CCD.

curv_coeff_mxu.fits: coefficients of the spatial curvature fitting polynomials.

disp_coeff_mxu.fits: coefficients of the wavelength calibration fitting polynomials.

Products for checking the quality of the result are:

master_screen_flat_mxu.fits: sum of all the input flat field exposures.

curv_traces_mxu.fits: table containing the y CCD positions of the detected spectral edges at different x CCD positions, compared with their modeling.

delta_image_mxu.fits: deviation from the linear term of the wavelength calibration fitting polynomials.

disp_residuals_mxu.fits: residuals for each wavelength calibration fit.

disp_residuals_table_mxu.fits: table containing different kinds of residuals for a sample of wavelength calibration fits.

global_distortion_table.fits: table containing the modeling of the coefficients listed in the curv_coeff_mxu.fits and disp_coeff_mxu.fits tables, only produced if more than 12 slits are available.

reduced_lamp_mxu.fits: rectified and wavelength calibrated arc lamp image.

spectra_detection_mxu.fits: result of preliminary wavelength calibration applied to the input arc lamp exposure, produced only if the recipe configuration --check is set.

wavelength_map_mxu.fits: map of wavelengths on the CCD.

spatial_map_mxu.fits: map of spatial positions on the CCD.

slit_map_mxu.fits: map of the grism central wavelength.

spectra_resolution_mxu.fits: mean spectral resolution for each reference arc lamp line.

Now the scientific frame can be processed, and for this the recipe *fors_science* is used (see Section 9.9, page 92). The following set-of-frames file may be created:

File: *sci.sof*

```
FORS2.2004-09-27T02:39:11.479.fits          SCIENCE_MXU
master_bias.fits                            MASTER_BIAS
master_norm_flat_mxu.fits                   MASTER_NORM_FLAT_MXU
disp_coeff_mxu.fits                         DISP_COEFF_MXU
curv_coeff_mxu.fits                         CURV_COEFF_MXU
slit_location_mxu.fits                      SLIT_LOCATION_MXU
/cal/fors2/FORS2_GRS_300I_21_OG590_32.fits  GRISM_TABLE
```

Note that the same (optional) GRISM_TABLE specified in the `cal.sof` file is used here. This is advisable, even if not really mandatory.

With the following command:

```
esorex fors_science sci.sof
```

the following products are created on disk:

mapped_all_sci_mxu.fits: image with rectified and wavelength calibrated slit spectra.

mapped_sci_mxu.fits: image with rectified, wavelength calibrated, and sky subtracted slit spectra.

mapped_sky_sci_mxu.fits: image with rectified and wavelength calibrated slit sky spectra.

unmapped_sci_mxu.fits: image with the sky subtracted scientific spectra on the CCD.

unmapped_sky_sci_mxu.fits: image with the modeled sky spectra on the CCD.

object_table_sci_mxu.fits: slit positions on the CCD, on the mapped images, and positions of the detected objects within the slits.

reduced_sci_mxu.fits: image with extracted objects spectra.

reduced_sky_sci_mxu.fits: image with sky corresponding to the extracted objects spectra.

reduced_error_sci_mxu.fits: image with the statistical errors corresponding to the extracted objects spectra.

sky_shifts_slit_sci_mxu.fits: table containing the observed sky lines offsets that were used for adjusting the input wavelength solutions, only created if the sky line alignment was requested.

wavelength_map_sci_mxu.fits: map of wavelengths on the CCD, only created if the sky line alignment was requested.

disp_coeff_sci_mxu.fits: wavelength calibration polynomials coefficients after alignment of the solutions to the position of the sky lines, only created if the sky line alignment was requested.

Currently there is no support for a spectro-photometric calibration. Standard star spectra are reduced in a similar fashion, applying the long-slit data reduction strategy[5] that is described in the next Section.

[5] Standard star observations are typically performed in LSS mode, or in MOS mode with all slitlets at the same offset.

4.4 Example of FORS2 long-slit data reduction using Esorex

Long-slit observations are those which are performed either in LSS mode or in MOS mode with the same offset applied to all slitlets, and are used both for scientific and calibration (standard star) purposes.

The algorithms applied for data processing are slightly different from those applied in the case of a generic MOS/MXU observation: for instance, more robust methods can be used for the alignment of the dispersion solution to the sky line positions, thanks to the availability of a larger and more homogeneous statistical sample. Moreover, since the slit is long, its ends are far apart and typically not visible in the detector, and therefore they cannot be used to determine a reliable spatial curvature solution: for this reason the spatial curvature related products are not created.[6].

In the following example a FORS2 LSS observation of a standard star is processed.[7]

It is assumed that the following data are available:

One standard star exposure:

```
FORS2.2004-09-27T03:12:12.006.fits          STANDARD_LSS
```

Three flat field exposures obtained with the mask used for the standard star exposure:

```
FORS2.2004-09-27T19:22:22.308.fits          SCREEN_FLAT_LSS
FORS2.2004-09-27T19:23:14.722.fits          SCREEN_FLAT_LSS
FORS2.2004-09-27T19:24:52.651.fits          SCREEN_FLAT_LSS
```

One arc lamp exposure obtained with the mask used for the standard star exposure:

```
FORS2.2004-09-27T19:33:44.097.fits          LAMP_LSS
```

Five bias exposures:

```
FORS2.2004-09-27T08:00:27.821.fits          BIAS
FORS2.2004-09-27T08:01:05.604.fits          BIAS
FORS2.2004-09-27T08:01:44.091.fits          BIAS
FORS2.2004-09-27T08:02:22.070.fits          BIAS
FORS2.2004-09-27T08:03:01.042.fits          BIAS
```

All the listed data are meant to be obtained from the same FORS2 chip, with the same grism and filter in use (respectively 300I and OG590 in this example).

In the following, it is also assumed for simplicity that, in the *Esorex* configuration file, the flag *suppress-prefix* is set to `TRUE`, so that the product file names will just be identical to their product categories, with an extension `.fits`. Moreover, it is assumed that all the handled files (inputs and products) are located in the

[6] In a future release it will be possible to import a curvature solution obtained from appropriate calibration masks

[7] In case of a scientific observation the DO categories would be the same, just replacing STANDARD with SCIENCE, and STD with SCI.

current directory. The only exception is represented by the standard calibration tables (e.g., line catalogues), that here are assumed to be located under `/cal/fors2`.

In order to process the calibration exposures available for the standard star observation, the recipe *fors_calib* is used (see Section 9.8, page 76). The input SOF may be defined as follows:

File: *cal.sof*

```
FORS2.2004-09-27T08:00:27.821.fits          BIAS
FORS2.2004-09-27T08:01:05.604.fits          BIAS
FORS2.2004-09-27T08:01:44.091.fits          BIAS
FORS2.2004-09-27T08:02:22.070.fits          BIAS
FORS2.2004-09-27T08:03:01.042.fits          BIAS
FORS2.2004-09-27T19:22:22.308.fits          SCREEN_FLAT_LSS
FORS2.2004-09-27T19:23:14.722.fits          SCREEN_FLAT_LSS
FORS2.2004-09-27T19:24:52.651.fits          SCREEN_FLAT_LSS
FORS2.2004-09-27T19:33:44.097.fits          LAMP_LSS
/cal/fors2/FORS2_ACAT_300I_21_OG590_32.fits MASTER_LINECAT
/cal/fors2/FORS2_GRS_300I_21_OG590_32.fits  GRISM_TABLE
```

The input BIAS frames are used to generate a median bias frame that is internally subtracted from all the input raw images, and eventually written to disk for further use. A master bias frame may be also produced using other means (taking care of trimming the overscan regions from the final result). This own-produced master bias frame may be specified in input instead of the sequence of raw BIAS frames, using the tag MASTER_BIAS:

File: *cal.sof*

```
master_bias.fits                            MASTER_BIAS
FORS2.2004-09-27T19:22:22.308.fits          SCREEN_FLAT_LSS
FORS2.2004-09-27T19:23:14.722.fits          SCREEN_FLAT_LSS
FORS2.2004-09-27T19:24:52.651.fits          SCREEN_FLAT_LSS
FORS2.2004-09-27T19:33:44.097.fits          LAMP_LSS
/cal/fors2/FORS2_ACAT_300I_21_OG590_32.fits MASTER_LINECAT
/cal/fors2/FORS2_GRS_300I_21_OG590_32.fits  GRISM_TABLE
```

The MASTER_LINECAT and the GRISM_TABLE are static calibration tables that are available in the calibration directories delivered with the pipeline recipes. The file FORS2_ACAT_300I_21_OG590_32.fits is the default catalogue of reference arc lamp lines for grism 300I and filter OG590 of the FORS2 instrument. This catalogue may be replaced with alternative ones provided by the user, if found appropriate.

The FORS2_GRS_300I_21_OG590_32.fits table contains the default *fors_calib* recipe configuration parameters for grism 300I and filter OG590 of the FORS2 instrument. If this file is not specified, appropriate values for the parameters must be set in the command line or in the *Esorex* configuration file.

The following command line can now be given at the shell prompt:

```
esorex fors_calib cal.sof
```

Several products are created on disk, mainly for check purposes. The products which are necessary for the scientific data reduction are the following:

master_bias.fits: master bias frame, produced only in case a sequence of raw BIAS exposures was specified in input.

master_norm_flat_lss.fits: normalised flat field image.

slit_location_lss.fits: slit positions on the CCD.

disp_coeff_lss.fits: coefficients of the wavelength calibration fitting polynomials.

Products for checking the quality of the result are:

master_screen_flat_lss.fits: sum of all the input flat field exposures.

delta_image_lss.fits: deviation from the linear term of the wavelength calibration fitting polynomials.

disp_residuals_lss.fits: residuals for each wavelength calibration fit.

disp_residuals_table_lss.fits: table containing different kinds of residuals for a sample of wavelength calibration fits.

reduced_lamp_lss.fits: wavelength calibrated arc lamp image.

wavelength_map_lss.fits: map of wavelengths on the CCD.

spectra_resolution_lss.fits: mean spectral resolution for each reference arc lamp line.

Now the scientific frame can be processed, and for this the recipe *fors_science* is used (see Section 9.9, page 92). The following set-of-frames file may be created:

File: *sci.sof*

```
FORS2.2004-09-27T03:12:12.006.fits          STANDARD_LSS
master_bias.fits                            MASTER_BIAS
master_norm_flat_lss.fits                   MASTER_NORM_FLAT_LSS
disp_coeff_lss.fits                         DISP_COEFF_LSS
slit_location_lss.fits                      SLIT_LOCATION_LSS
/cal/fors2/FORS2_GRS_300I_21_OG590_32.fits  GRISM_TABLE
```

Note that the same (optional) GRISM_TABLE specified in the `cal.sof` file is used here. This is advisable, even if not really mandatory.

With the following command:

```
esorex fors_science sci.sof
```

the following products are created on disk:

mapped_all_std_lss.fits: image with wavelength calibrated slit spectra.

mapped_std_lss.fits: image with wavelength calibrated and sky subtracted slit spectra.

object_table_std_lss.fits: slit positions on the CCD, on the mapped images, and positions of the detected objects within the slits.

reduced_std_lss.fits: image with extracted objects spectra.

reduced_sky_std_lss.fits: image with sky corresponding to the extracted objects spectra.

reduced_error_std_lss.fits: image with the statistical errors corresponding to the extracted objects spectra.

sky_shifts_long_std_lss.fits: table containing the observed sky lines offsets that were used for adjusting the input wavelength solutions, only created if the sky line alignment was requested.

wavelength_map_std_lss.fits: map of wavelengths on the CCD, only created if the sky line alignment was requested.

disp_coeff_std_lss.fits: wavelength calibration polynomials coefficients after alignment of the solutions to the position of the sky lines, only created if the sky line alignment was requested.

Currently no comparison is made with catalogue standard star spectra for the computation of a spectroscopic response curve.

5 Known Problems

Development work is still ongoing, as these recipes do not yet support important tasks such as the combination of a sequence of scientific exposures, or the determination of spectral response curves.

The automatic spectral sky subtraction is not guaranteed to always work well, especially in the case of LSS or LSS-like data, or with too short MXU slits (dominated by the emission of the target): the presence of extended objects on slit would make it difficult to determine the sky signal to subtract.

Occasionally spectra may be lost to the pattern-matching algorithm: such cases will require some iteration, running the data reduction recipes with different parameter settings or editing the reference arc lamp lines catalogue (adding more lines, if available).

The alignment of the wavelength calibration solution to the sky-lines does not give good results at very low spectral resolution ($R < 300$, which is always the case with grism 150I in FORS1 and FORS2, even at slit widths less than 1").

The spatial curvature is not modeled for long-slit data, because it is not directly observable. This will be partially solved in the next pipeline release by applying predefined (but alas, fixed) spatial curvature models. A complete solution to this problem will probably never be available.

FORS1 polarimetric and FORS2 echelle data reduction are still unsupported.

6 Instrument Data Description

This Section mainly deals with the appropriate classification of data used by the pipeline. This is done automatically by *Gasgano* [14], but not by *Esorex* [5]. If the pipeline user has already classified their FITS files (in biases, flat fields, science frames, etc.), there is no need to use the formal classification rules described in this Section.

The data handled by the FORS pipeline can be separated into *raw* frames and *calib* frames. *Raw* frames are the unprocessed output of the FORS instrument observations, while *calib* frames are anything else: either the result of the FORS pipeline processing (as reduced frames, master calibration frames, etc.), or obtained in other ways (as standard stars catalogues, lists of grism characteristics, etc.).

Any *raw* or *calib* frame can be classified on the basis of a set of keywords read from its header. Data classification is typically carried out by the DO or by *Gasgano*, which apply the same set of classification rules. The association of a raw frame with calibration data (*e.g.*, of a science frame with a master bias frame) can be obtained by matching the values of appropriate sets of header keywords.

Each kind of *raw* frame is typically associated to a single FORS pipeline recipe, i.e., the recipe assigned to the reduction of that specific frame type. In the pipeline environment this recipe would be launched automatically. In some cases two recipes are assigned, one meant for the reduction of a single frame of that type, and the other for the reduction of a *stack* of frames of the same type, as happens in the case of jittered science observations.

A *calib* frame may be input to more than one FORS pipeline recipe, but it may be created by just one pipeline recipe (with the same exceptions mentioned above). In the automatic pipeline environment a *calib* data frame alone would not trigger the launch of any recipe.

In the following all raw FORS data frames are listed, together with the keywords used for their classification and correct association. The indicated *DO category* is a label assigned by the online pipeline system to any data type after it has been classified, which is then used to identify the frames listed in the *Set of Frames* (see Section 4.1.2, page 20).

The *calib* frames produced by the pipeline are listed in the description of the individual recipes producing them. The *calib* frames which are used by the pipelines, indicated also as static calibration data, are described in section 7, page 44.

Raw frames can be distinguished in *general* frames, *IMG* frames, *MXU* frames, *MOS* frames, and *LSS* frames. Their intended use is implicitly defined by the assigned recipe.

6.1 General frames

These are data that are in principle independent of the instrument mode (direct imaging, spectroscopy), as is the case for bias exposures.

- **Bias:**
  DO category: `BIAS`
  Processed by: `fors_bias, fors_calib`

Classification keywords:	Association keywords:	Note:
`DPR CATG = CALIB`	`DET READ CLOCK`	Readout mode
`DPR TYPE = BIAS`	`DET WIN1 BINX`	x-binning
	`DET WIN1 BINY`	y-binning
	`DET OUTPUTS`	No of outputs
	`DET WIN1 STRX`	Window start in x
	`DET WIN1 STRY`	Window start in y
	`DET WIN1 NX`	No of pixels in x
	`DET WIN1 NY`	No of pixels in y
	`DET CHIP1 ID`	Chip identifier

- **Dark current:**
  DO category: `DARK`
  Processed by: `fors_dark`

Classification keywords:	Association keywords:	Note:
`DPR CATG = CALIB`	`DET READ CLOCK`	Readout mode
`DPR TYPE = DARK`	`DET WIN1 BINX`	x-binning
	`DET WIN1 BINY`	y-binning
	`DET OUTPUTS`	No of outputs
	`DET WIN1 STRX`	Window start in x
	`DET WIN1 STRY`	Window start in y
	`DET WIN1 NX`	No of pixels in x
	`DET WIN1 NY`	No of pixels in y
	`DET CHIP1 ID`	Chip identifier

6.2 IMG frames

The IMG mode performs direct imaging applying different filters.

- **Screen flat field:**
  DO category: `SCREEN_FLAT_IMG`
  Processed by: `fors_img_screen_flat`

Classification keywords:	Association keywords:	Note:
`DPR CATG = CALIB`	*None: unused in processing*	
`DPR TYPE = FLAT,LAMP`		
`DPR TECH = IMAGE`		

- **Sky flat field:**
  DO category: `SKY_FLAT_IMG`
  Processed by: `fors_img_sky_flat`

Classification keywords:	Association keywords:	Note:
`DPR CATG = CALIB`	`INS FILT1 NAME`	Filter used
`DPR TYPE = FLAT,SKY`	`DET READ CLOCK`	Readout mode
`DPR TECH = IMAGE`	`DET WIN1 BINX`	x-binning
	`DET WIN1 BINY`	y-binning
	`DET OUTPUTS`	No of outputs
	`DET WIN1 STRX`	Window start in x
	`DET WIN1 STRY`	Window start in y
	`DET WIN1 NX`	No of pixels in x
	`DET WIN1 NY`	No of pixels in y
	`DET CHIP1 ID`	Chip identifier

- **Scientific observation:**
  DO category: `SCIENCE_IMG`
  Processed by: `fors_img_science`

Classification keywords:	Association keywords:	Note:
`DPR CATG = SCIENCE`	`INS FILT1 NAME`	Filter used
`DPR TECH = IMAGE`	`DET READ CLOCK`	Readout mode
	`INS COLL NAME`	Collimator name
	`DET WIN1 BINX`	x-binning
	`DET WIN1 BINY`	y-binning
	`DET OUTPUTS`	No of outputs
	`DET WIN1 STRX`	Window start in x
	`DET WIN1 STRY`	Window start in y
	`DET WIN1 NX`	No of pixels in x
	`DET WIN1 NY`	No of pixels in y
	`DET CHIP1 ID`	Chip identifier

- **Standard star field:**
  DO category: `STANDARD_IMG`
  Processed by: `fors_zeropoint`

Classification keywords:	Association keywords:	Note:
`DPR CATG = CALIB`	`INS FILT1 NAME`	Filter used
`DPR TECH = IMAGE`	`DET READ CLOCK`	Readout mode
`DPR TYPE = STD`	`INS COLL NAME`	Collimator name
	`DET WIN1 BINX`	x-binning
	`DET WIN1 BINY`	y-binning
	`DET OUTPUTS`	No of outputs
	`DET WIN1 STRX`	Window start in x
	`DET WIN1 STRY`	Window start in y
	`DET WIN1 NX`	No of pixels in x
	`DET WIN1 NY`	No of pixels in y
	`DET CHIP1 ID`	Chip identifier

6.3 MXU frames (FORS2 only)

The MXU mode performs multi-object spectroscopy using a mask exchange unit.

- **Screen flat field:**

  DO category: `SCREEN_FLAT_MXU`
  Processed by: `fors_calib`

  Classification keywords:
  `DPR CATG = CALIB`
  `DPR TYPE = FLAT,LAMP`
  `DPR TECH = MXU`

Association keywords:	Note:
`INS COLL NAME`	Collimator used
`INS GRIS1 NAME`	Grism used
`INS MASK ID`	MXU mask ID
`INS FILT1 NAME`	Filter used
`DET READ CLOCK`	Readout mode
`DET WIN1 BINX`	x-binning
`DET WIN1 BINY`	y-binning
`DET OUTPUTS`	No of outputs
`DET WIN1 STRX`	Window start in x
`DET WIN1 STRY`	Window start in y
`DET WIN1 NX`	No of pixels in x
`DET WIN1 NY`	No of pixels in y
`DET CHIP1 ID`	Chip identifier

- **Arc lamp spectrum:**

  DO category: `LAMP_MXU`
  Processed by: `fors_calib`

  Classification keywords:
  `DPR CATG = CALIB`
  `DPR TYPE = WAVE,LAMP`
  `DPR TECH = MXU`

Association keywords:	Note:
`INS COLL NAME`	Collimator used
`INS GRIS1 NAME`	Grism used
`INS MASK ID`	MXU mask ID
`INS FILT1 NAME`	Filter used
`DET READ CLOCK`	Readout mode
`DET WIN1 BINX`	x-binning
`DET WIN1 BINY`	y-binning
`DET OUTPUTS`	No of outputs
`DET WIN1 STRX`	Window start in x
`DET WIN1 STRY`	Window start in y
`DET WIN1 NX`	No of pixels in x
`DET WIN1 NY`	No of pixels in y
`DET CHIP1 ID`	Chip identifier

- **Scientific observation:**

  DO category: `SCIENCE_MXU`
  Processed by: `fors_science, fors_extract`

  Classification keywords:
  `DPR CATG = SCIENCE`

Association keywords:	Note:
`INS COLL NAME`	Collimator used

DPR TECH = MXU	INS GRIS1 NAME	Grism used
	INS MASK ID	MXU mask ID
	INS FILT1 NAME	Filter used
	DET READ CLOCK	Readout mode
	DET WIN1 BINX	x-binning
	DET WIN1 BINY	y-binning
	DET OUTPUTS	No of outputs
	DET WIN1 STRX	Window start in x
	DET WIN1 STRY	Window start in y
	DET WIN1 NX	No of pixels in x
	DET WIN1 NY	No of pixels in y
	DET CHIP1 ID	Chip identifier

6.4 MOS frames

The MOS mode performs multi-object spectroscopy using a set of movable slitlets.

- **Screen flat field:**

  DO category: SCREEN_FLAT_MOS
  Processed by: fors_calib

Classification keywords:	Association keywords:	Note:
DPR CATG = CALIB	INS COLL NAME	Collimator used
DPR TYPE = FLAT,LAMP	INS GRIS1 NAME	Grism used
DPR TECH = MOS	INS MOS CHECKSUM	MOS slit position checksum
	INS FILT1 NAME	Filter used
	DET READ CLOCK	Readout mode
	DET WIN1 BINX	x-binning
	DET WIN1 BINY	y-binning
	DET OUTPUTS	No of outputs
	DET WIN1 STRX	Window start in x
	DET WIN1 STRY	Window start in y
	DET WIN1 NX	No of pixels in x
	DET WIN1 NY	No of pixels in y
	DET CHIP1 ID	Chip identifier

- **Arc lamp spectrum:**

  DO category: LAMP_MOS
  Processed by: fors_calib

Classification keywords:	Association keywords:	Note:
DPR CATG = CALIB	INS COLL NAME	Collimator used
DPR TYPE = WAVE,LAMP	INS GRIS1 NAME	Grism used
DPR TECH = MOS	INS MOS CHECKSUM//	MOS slitposition checksum
	INS FILT1 NAME	Filter used
	DET READ CLOCK	Readout mode
	DET WIN1 BINX	x-binning

DET WIN1 BINY	y-binning
DET OUTPUTS	No of outputs
DET WIN1 STRX	Window start in x
DET WIN1 STRY	Window start in y
DET WIN1 NX	No of pixels in x
DET WIN1 NY	No of pixels in y
DET CHIP1 ID	Chip identifier

- **Standard star spectrum:**

  DO category: STANDARD_MOS
  Processed by: fors_science, fors_extract

Classification keywords:	Association keywords:	Note:
DPR CATG = CALIB	INS COLL NAME	Collimator used
DPR TYPE = STD	INS GRIS1 NAME	Grism used
DPR TECH = MOS	INS MOS CHECKSUM	MOS slit position checksum
	INS FILT1 NAME	Filter used
	DET READ CLOCK	Readout mode
	DET WIN1 BINX	x-binning
	DET WIN1 BINY	y-binning
	DET OUTPUTS	No of outputs
	DET WIN1 STRX	Window start in x
	DET WIN1 STRY	Window start in y
	DET WIN1 NX	No of pixels in x
	DET WIN1 NY	No of pixels in y
	DET CHIP1 ID	Chip identifier

- **Scientific observation:**

  DO category: SCIENCE_MOS
  Processed by: fors_science, fors_extract

Classification keywords:	Association keywords:	Note:
DPR CATG = SCIENCE	INS COLL NAME	Collimator used
DPR TECH = MOS	INS GRIS1 NAME	Grism used
	INS MOS CHECKSUM	MOS slit position checksum
	INS FILT1 NAME	Filter used
	DET READ CLOCK	Readout mode
	DET WIN1 BINX	x-binning
	DET WIN1 BINY	y-binning
	DET OUTPUTS	No of outputs
	DET WIN1 STRX	Window start in x
	DET WIN1 STRY	Window start in y
	DET WIN1 NX	No of pixels in x
	DET WIN1 NY	No of pixels in y
	DET CHIP1 ID	Chip identifier

6.5 LSS frames

The LSS mode is used to perform long-slit spectroscopy.

- **Screen flat field:**
  DO category: `SCREEN_FLAT_LSS`
  Processed by: `fors_calib`

Classification keywords:	Association keywords:	Note:
`DPR CATG = CALIB`	`INS COLL NAME`	Collimator used
`DPR TYPE = FLAT,LAMP`	`INS GRIS1 NAME`	Grism used
`DPR TECH = SPECTRUM`	`INS SLIT NAME`	Slit used
	`INS FILT1 NAME`	Filter used
	`DET READ CLOCK`	Readout mode
	`DET WIN1 BINX`	x-binning
	`DET WIN1 BINY`	y-binning
	`DET OUTPUTS`	No of outputs
	`DET WIN1 STRX`	Window start in x
	`DET WIN1 STRY`	Window start in y
	`DET WIN1 NX`	No of pixels in x
	`DET WIN1 NY`	No of pixels in y
	`DET CHIP1 ID`	Chip identifier

- **Arc lamp spectrum:**
  DO category: `LAMP_LSS`
  Processed by: `fors_calib`

Classification keywords:	Association keywords:	Note:
`DPR CATG = CALIB`	`INS COLL NAME`	Collimator used
`DPR TYPE = WAVE,LAMP`	`INS GRIS1 NAME`	Grism used
`DPR TECH = SPECTRUM`	`INS SLIT NAME`	Slit used
	`INS FILT1 NAME`	Filter used
	`DET READ CLOCK`	Readout mode
	`DET WIN1 BINX`	x-binning
	`DET WIN1 BINY`	y-binning
	`DET OUTPUTS`	No of outputs
	`DET WIN1 STRX`	Window start in x
	`DET WIN1 STRY`	Window start in y
	`DET WIN1 NX`	No of pixels in x
	`DET WIN1 NY`	No of pixels in y
	`DET CHIP1 ID`	Chip identifier

- **Frame for flat field lamp monitoring:**
  DO category: `FLUX_FLAT_LSS`
  Processed by: `fors_sumflux`

Classification keywords:	Association keywords:	Note:
`DPR CATG = CALIB`	`INS COLL NAME`	Collimator used

`DPR TYPE = FLAT,LAMP`	`INS GRIS1 NAME`	Grism used
`DPR TECH = INS-THROUGH`	`INS SLIT NAME`	Slit used
	`INS FILT1 NAME`	Filter used
	`DET READ CLOCK`	Readout mode
	`DET WIN1 BINX`	x-binning
	`DET WIN1 BINY`	y-binning
	`DET OUTPUTS`	No of outputs
	`DET WIN1 STRX`	Window start in x
	`DET WIN1 STRY`	Window start in y
	`DET WIN1 NX`	No of pixels in x
	`DET WIN1 NY`	No of pixels in y
	`DET CHIP1 ID`	Chip identifier

- **Frame for arc lamp monitoring:**

  DO category: `FLUX_ARC_LSS`
  Processed by: `fors_sumflux`

Classification keywords:	Association keywords:	Note:
`DPR CATG = CALIB`	`INS COLL NAME`	Collimator used
`DPR TYPE = WAVE,LAMP`	`INS GRIS1 NAME`	Grism used
`DPR TECH = INS-THROUGH`	`INS SLIT NAME`	Slit used
	`INS FILT1 NAME`	Filter used
	`DET READ CLOCK`	Readout mode
	`DET WIN1 BINX`	x-binning
	`DET WIN1 BINY`	y-binning
	`DET OUTPUTS`	No of outputs
	`DET WIN1 STRX`	Window start in x
	`DET WIN1 STRY`	Window start in y
	`DET WIN1 NX`	No of pixels in x
	`DET WIN1 NY`	No of pixels in y
	`DET CHIP1 ID`	Chip identifier

- **Standard star spectrum:**

  DO category: `STANDARD_LSS`
  Processed by: `fors_science, fors_extract`

Classification keywords:	Association keywords:	Note:
`DPR CATG = CALIB`	`INS COLL NAME`	Collimator used
`DPR TYPE = STD`	`INS GRIS1 NAME`	Grism used
`DPR TECH = SPECTRUM`	`INS SLIT NAME`	Slit used
	`INS FILT1 NAME`	Filter used
	`DET READ CLOCK`	Readout mode
	`DET WIN1 BINX`	x-binning
	`DET WIN1 BINY`	y-binning
	`DET OUTPUTS`	No of outputs
	`DET WIN1 STRX`	Window start in x
	`DET WIN1 STRY`	Window start in y
	`DET WIN1 NX`	No of pixels in x
	`DET WIN1 NY`	No of pixels in y

DET CHIP1 ID	Chip identifier

- **Scientific observation:**

  DO category: SCIENCE_LSS
  Processed by: fors_science, fors_extract

Classification keywords:	Association keywords:	Note:
DPR CATG = SCIENCE	INS COLL NAME	Collimator used
DPR TECH = SPECTRUM	INS GRIS1 NAME	Grism used
	INS SLIT NAME	Slit used
	INS FILT1 NAME	Filter used
	DET READ CLOCK	Readout mode
	DET WIN1 BINX	x-binning
	DET WIN1 BINY	y-binning
	DET OUTPUTS	No of outputs
	DET WIN1 STRX	Window start in x
	DET WIN1 STRY	Window start in y
	DET WIN1 NX	No of pixels in x
	DET WIN1 NY	No of pixels in y
	DET CHIP1 ID	Chip identifier

7 Static Calibration Data

In the following all the FORS static calibration tables related to direct imaging and spectroscopic modes are listed. The indicated *DO category*, written to the FITS header keyword PRO.CATG, is a label assigned to any data type after it has been classified. This label is then used to identify the frames listed in the *set-of-frames* (see Section 4.1.2, page 20).

7.1 Photometric table

DO category: `PHOT_TABLE`

This table lists parameters related to each standard filter in use in the FORS1 and FORS2 instruments. There is one such table for each CCD in the mosaic of each instrument. Currently available are:

fors1_Marlene_phot.fits
fors1_Norma_phot.fits
fors1_TK_phot.fits
fors2_1453_phot.fits
fors2_1456_phot.fits

Each table contains the following columns:

filter:	Name of filter
ext:	Atmospheric extinction coefficient (mag/airmass)
dext:	Error on atmospheric extinction coefficient (mag/airmass)
col:	Color correction term
dcol:	Error on color term
zpoint:	Clear night zeropoint
dzpoint:	Error on clear night zeropoint

To clarify the terminology used here: more than one filter may correspond to each of the standard bands, (U,B,V,G,R,I). For instance, in the FORS1 instrument the following filters are available:

Filter	*Band*
U_BESS	U
u_HIGH	U
B_BESS	B
b_HIGH	B
g_HIGH	G
V_BESS	V
v_HIGH	V
R_BESS	R
I_BESS	I

For each filter, beyond the corresponding atmospheric extinction and expected zeropoint, a color term is given in the column **col** of the photometric table. The color term is a correction factor which is applied to the color index of a given star, in order to compensate for the differences between the FORS filters and the standard UBVRI bands using the linear approximation:

$$M^{\star}_{s,b} = M_{s,b} - \Gamma_f \cdot C_{s,b}$$

where $M_{s,b}$ is the catalogue magnitude in the band b of a star s, $M^{\star}_{s,b}$ the color corrected magnitude, Γ_f the linear color correction term for the filter f, and $C_{s,b}$ the color index associated to the band b for the star s. A color index is conventionally associated to each band b as in the following table:

U	U-B
B	B-V
V	B-V
G	B-V
R	V-R
I	V-R

7.2 Photometric standard stars catalogue

DO category: `FLX_STD_IMG`

This table is a list of photometric standard stars parameters. Currently two catalogues are included in the static calibrations coming with the pipeline: selected UBVRI photometric standard stars from Landolt [21], and all entries from the Stetson's photometric standard stars catalogue [23] [24], respectively in the files landolt_std_UBVRI.tfits and stetson_std_BVRI.fits

The Landolt table lists U,B,V,R,I magnitudes and B-V, U-B, V-R, R-I, V-I for each star. Only the errors of the color index and the V band magnitude are provided. The pipeline assumes the V band error for all magnitudes, and will use the color index error when dealing directly with color index. In any case, the catalogue magnitude error is generally not a dominant contribution to the total error on the zeropoint computed by the pipeline.

The Stetson table lists only B,V,R,I magnitudes (no U magnitudes), and independent errors are provided for each measurement. Color indexes are not provided directly.

The pipeline recipes use whatever catalogue is specified in input, either Landolt's or Stetson's, or even both. If both catalogues are specified, they are merged before being used. In the merged catalogue, if two stars are within 5 arcseconds they are considered identical and the one with the largest magnitude error is removed.[8]

When a catalogue is loaded, a G magnitude is added to the (merged) catalogue applying the relation

$$G = V + 0.56(B - V) - 0.12$$

as in Fukugita (1996) [22]. In addition to that, a linear color correction is applied to the listed magnitudes to adapt them to the filter used in the standard star field observation (see previous Section). The error propagation

[8] It should be noted that using simultaneously two different catalogues carries a significant risk to produce inconsistent results, as the magnitudes are not derived with the same method in both catalogues, and as a consequence common stars can differ by up to 0.2 magnitudes. The use of two catalogues is therefore strongly discouraged.

is performed according to which uncorrelated errors are available. In order to do this, the generic form of the correction

$$M^{\star} = M - \Gamma \cdot C$$

(see previous Section) is expressed in terms of directly measured quantities. In the Landolt catalogue case it will be:

$$\begin{aligned}
U &= v + (b-v) + (u-b) - \Gamma \cdot (u-b) \\
B &= v + (b-v) - \Gamma \cdot (b-v) \\
G &= v + (0.56-\Gamma) \cdot (b-v) - 0.12 \ \ \text{(Fukugita, 1996)} \\
V &= v - \Gamma \cdot (b-v) \\
R &= v - (v-r) - \Gamma \cdot (v-r) \\
I &= v - (v-i) - \Gamma \cdot (v-r)
\end{aligned}$$

which leads to the variances

$$\begin{aligned}
\Delta U^2 &= \Delta v^2 + \Delta(b-v)^2 + (1-\Gamma)^2 \Delta(u-b)^2 + (u-b)^2 \Delta\Gamma^2 \\
\Delta B^2 &= \Delta v^2 + (1-\Gamma)^2 \Delta(b-v)^2 + (b-v)^2 \Delta\Gamma^2 \\
\Delta G^2 &= \Delta v^2 + (0.56-\Gamma)^2 \Delta(b-v)^2 + (b-v)^2 \Delta\Gamma^2 \\
\Delta V^2 &= \Delta v^2 + \Gamma^2 \Delta(b-v)^2 + (b-v)^2 \Delta\Gamma^2 \\
\Delta R^2 &= \Delta v^2 + (1-\Gamma)^2 \Delta(v-r)^2 + (v-r)^2 \Delta\Gamma^2 \\
\Delta I^2 &= \Delta v^2 + \Delta(v-i)^2 + \Gamma^2 \Delta(v-r)^2 + (v-r)^2 \Delta\Gamma^2
\end{aligned}$$

Similarly, in the Stetson catalogue case it will be:

$$\begin{aligned}
U &= (1-\Gamma) \cdot u + \Gamma \cdot b \\
B &= (1-\Gamma) \cdot b + \Gamma \cdot v \\
G &= v + (0.56-\Gamma) \cdot (b-v) - 0.12 \ \ \text{(Fukugita, 1996)} \\
V &= (1+\Gamma) \cdot v - \Gamma \cdot b \\
R &= (1+\Gamma) \cdot r - \Gamma \cdot v \\
I &= i + \Gamma \cdot r - \Gamma \cdot v
\end{aligned}$$

which leads to the variances

$$\begin{aligned}
\Delta U^2 &= (1-\Gamma)^2 \Delta u^2 + (b-u)^2 \Delta\Gamma^2 + \Gamma^2 \Delta b^2 \\
\Delta B^2 &= (1-\Gamma)^2 \Delta b^2 + (v-b)^2 \Delta\Gamma^2 + \Gamma^2 \Delta v^2 \\
\Delta G^2 &= (0.44+\Gamma)^2 \Delta v^2 + (v-b)^2 \Delta\Gamma^2 + (0.56-\Gamma) \Delta b^2 \\
\Delta V^2 &= (1+\Gamma)^2 \Delta v^2 + (v-b)^2 \Delta\Gamma^2 + \Gamma^2 \Delta b^2 \\
\Delta R^2 &= (1+\Gamma)^2 \Delta r^2 + (r-v)^2 \Delta\Gamma^2 + \Gamma^2 \Delta v^2 \\
\Delta I^2 &= \Delta i^2 + \Gamma^2 \Delta r^2 + (r-v)^2 \Delta\Gamma^2 + \Gamma^2 \Delta v^2
\end{aligned}$$

7.3 Grism table

DO category: `GRISM_TABLE`

This table defines a subset of recipe configuration parameters controlling the way spectra are extracted for any particular grism. The table consists of a single row of values labeled with the parameters names. A standard grism table is provided for each FORS1 and FORS2 grism: this table is named following the convention:

<instrument name>_GRS_<grism name>_<grism ID>_<filter name>_<filter ID>.fits

where in case no filter is used the *filter name* is set to "free" and the *filter ID* is set to "00". In case a grism table can be used with all the available filters, the *filter name* and *filter ID* fields are replaced by the suffix *all*. For instance, the standard grism table for FORS2 grism 300V and filter GG435 is named

FORS2_GRS_300V_20_GG435_81.fits

while the grism table for FORS1 grism 600B with no filter is named

FORS1_GRS_600B_12_free_00.fits

The table

FORS2_GRS_300V_20_all.fits

is available for all standard filters usable with the FORS2 300V grism.

A new set of grism tables has been provided for supporting FORS1 data obtained after the blue CCD mosaic upgrade (April 2007). These tables can be identified both by the content of the DET.CHIP1.ID keywords, and by their names, where a "_B_" is added: for instance,

FORS1_B_GRS_600B_12_free_00.fits

is the grism table for the upgraded FORS1 mosaic used with grism 600B and no filter.

If a grism table is used, it will modify the recipe parameters with its new values, with the exception of those which are explicitly given on the command line. Without a grism table, the input recipe parameters values will just be read from the command line, or from an *esorex* configuration file if present, or from their generic default values (that are rarely meaningful). The configuration parameters included in the grism table are the following:

--dispersion	rough expected spectral dispersion
--peakdetection	threshold for preliminary peak detection
--wdegree	polynomial degree for wavelength calibration
--cdegree	polynomial degree for spatial curvature
--startwavelength	start wavelength for spectral extraction
--endwavelength	end wavelength for spectral extraction

A complete description of these parameters is given in Section 9.8.3, page 86.

7.4 Arc lamp lines catalogue

DO category: `MASTER_LINECAT`

This table contains a set of reference wavelengths (in Ångstrom) for the arc lamp used. The only requirement for this table is to contain a column listing such wavelengths, whose name may be specified using the *fors_recipe* configuration parameter *--wcolumn* (see Section 9.8.3, page 86). A standard line catalogue is also provided for each FORS1 and FORS2 grism: this table is named following the convention:

<instrument name>_ACAT_<grism name>_<grism ID>_<filter name>_<filter ID>.fits

where in case no filter is used the *filter name* is set to "free" and the *filter ID* is set to "00". In case a line catalogue can be used with all the available filters, the *filter name* and *filter ID* sequence is replaced by the suffix *all*. For instance, the arc lamp line catalogue for FORS2 grism 300V and filter GG435 is named

FORS2_ACAT_300V_20_GG435_81.fits

while the line catalogue for FORS1 grism 600B with no filter is named

FORS1_ACAT_600B_12_free_00.fits

The table

FORS2_ACAT_300V_20_all.fits

is available for all standard filters usable with the FORS2 300V grism.

In practice, however, the correct (standard) line catalogue can be associated to a given arc lamp frame using the FITS keyword ESO INS GRIS1 NAME, that is written both to the line catalogue and to the raw input frames headers.

7.5 Global distortion table

DO category: `GLOBAL_DISTORTION_TABLE`

Table containing the modeling of the coefficients of the local distortion models listed in the DISP_COEFF_MXU and the CURV_COEFF_MXU tables (see page 77). This table is just used for allowing the on-line processing of scientific data with recipe *fors_extract*, when appropriate (day) calibrations are not yet available.

Conventionally this table consists of 6 columns and 10 rows. Each row corresponds to the modeling of one coefficient of the original polynomial coefficients belonging to the local distortion solutions (presumably obtained with a calibration mask), performed by fitting a bivariate polynomial:

$$c_r = \sum_{i=0}^{2} \sum_{j=0}^{2-i} a_{ij} x^i y^j$$

where r is the table row number (counted from 0) and c_r is a polynomial coefficient of a local solution. For $r = 0$ and $r > 6$ (x, y) are positions on the telescope focal plane (e.g., on a mask), otherwise they are positions on the CCD. The first 6 table rows are a global description of the dispersion solution up to the fifth polynomial degree; these rows are followed by a row where just the first element is assigned the value of the central wavelength used for the given dispersion solution. The remaining 3 rows are a global description of the spatial curvature up to the second polynomial degree. The local dispersion solutions could be obtained with:

$$x = \sum_{r=0}^{5} c_r (\lambda - \lambda_o)^r$$

where x is the x CCD pixel position and λ_o is the central wavelength of the grism used. The local spatial curvature solutions could be obtained with:

$$y = \sum_{r=7}^{9} c_r x^{(r-7)}$$

where y is the y CCD pixel position and x is obtained with the previous formula.

The global distortion table columns are labeled a00, a01, a02, a10, a11, a20, indicating the coefficients of the fitting bivariate polynomials.

8 Data Reduction

Besides the usual data reduction steps, such as master bias subtraction, flat fielding, source detection and extraction, etc., two major aspects characterise the FORS instrument pipeline:

1. Many of the FORS pipeline recipes are based on pattern-matching techniques, in the attempt to make the algorithms more robust and instrument independent.

2. The pipeline recipes related to imaging data reduction support a complete propagation of photonic noise and CCD read out noise to all of its products. In the spectral data reduction part errors are propagated less rigorously, being estimated on the final products neglecting the errors introduced by both flat fielding and master bias subtraction.

A more detailed description of the applied algorithms is given in Section 10, page 105. Here just an overview of the data reduction cascade is provided.

8.1 Imaging data reduction overview

Seven recipes are available for the reduction of FORS imaging data:

fors_bias, to compute a master bias frame from a set of raw bias exposures. This recipe is shared with the spectroscopic data reduction.

fors_dark, to compute a master dark frame from a set of raw dark exposures. Dark levels are so low that the master dark calibration is not applied to the data. This recipe is just used for instrument quality control.

fors_img_sky_flat, to compute a master sky flat frame from a master bias frame and a set of raw twilight sky exposures.

fors_img_screen_flat, to compute a master screen flat frame from a master bias frame and a set of raw dome flat lamp exposures. The screen flat field is not well suited for flat field correction, and it is not applied to the data. This recipe is just used for instrument quality control.

fors_img_science, to reduce a scientific exposure.

fors_zeropoint, to estimate a zeropoint and an atmospheric extinction from a standard star field exposure.

fors_photometry, to determine the systematic errors of a master sky flat field frame, on the basis of several (dithered) standard star field exposures corrected with that flat field frame. The corrected flat field may allow for a more accurate photometric correction. Moreover, this recipe enables the determination of physical parameters related to photometry, such as color term filter correction, atmospheric extinction, and (true) instrument zeropoint.

8.1.1 Required input data

The input data to the recipe *fors_bias* is just a sequence of raw bias exposures. The product will be a bias master calibration frame to be used for the bias correction in the next reduction steps.

The input data to the recipe *fors_dark* is, besides the bias master calibration, a sequence of raw bias exposures. The product will be a dark master calibration frame which is currently used just for quality control of the instrument.

The input to the recipe *fors_img_screen_flat* is, besides the bias master calibration, a sequence of raw flat field lamp exposures. The product will be a flat field master calibration frame which is currently used just for quality control of the instrument.

The input to the recipe *fors_img_sky_flat* is, besides the bias master calibration, a sequence of raw twilight sky exposures. The product will be a flat field master calibration frame to be used for the flat field correction in the next reduction steps.

The input to the recipe *fors_img_science* is, besides the master calibration frames produced earlier, one scientific exposure. The reduced scientific frame, the list of detected sources, and a background map will be produced.

The input to the recipe *fors_zeropoint* is, besides the master calibration frames produced earlier, one photometric standard star field exposure, one or two photometric standard star catalogue (see Section 7.2, page 45), and a photometric table carrying appropriate atmospheric absorbtion coefficients, color terms and expected zeropoints appropriate for the instrument setting (see Section 7.1, page 44). The reduced exposure, the list of detected and identified sources, and a computed zeropoint will be produced.

The recipe *fors_photometry* can be used to process source lists produced in several runs of the recipe *fors_zeropoint*. This recipe is basically fitting instrumental and physical properties of the stars to extract information about the instrument response and the atmosphere. In particular this recipe may be used to determine the atmospheric extinction, the filter-dependent color correction, the instrument zeropoint, and the systematic deviation from "flatness" of a flat field.

8.1.2 Reduction cascade

The possible data reduction paths which can be followed using the imaging data reduction recipes are shown in image 8.1.1. The recipes *fors_img_screen_flat* and *fors_dark* are not shown, because they are not part of the (typical) data reduction flows.

8.2 Spectral data reduction overview

The FORS spectroscopic pipeline is based on a set of 5 stand-alone recipes. Only 3 of them are involved in the off-line data reduction cascade: the remaining recipes are just meant for on-line data reduction and instrument monitoring. The available recipes are the following:

fors_bias, to compute a master bias frame from a set of raw bias exposures. This recipe is shared with the imaging pipeline.

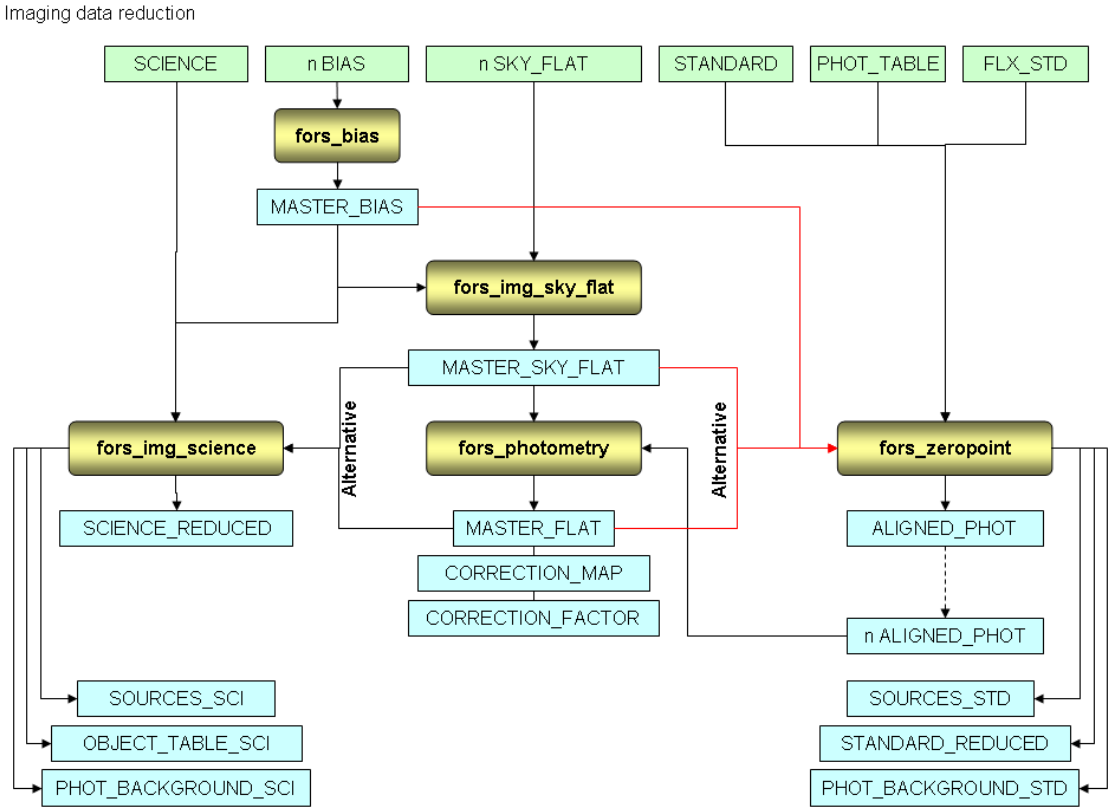

Figure 8.1.1: *This is the data reduction flow using the recipes for imaging data reduction. The rounded yellow boxes represent recipes handling input and intermediate products. The boxes indicating input and product data are labeled with their DO categories, stripped of their suffix _IMG (when present). Different line colors are used not to confuse crossing lines.*

fors_calib, to evaluate the spectral extraction mask on the basis of flat and arc calibration lamp exposures, and to create a normalised flat field frame.

fors_science, to apply the extraction mask and the normalised flat field to the scientific exposures.

fors_extract, identical to recipe *fors_science*, but used for the on-line data reduction based on fixed global optical and spectral distortion models.

fors_sumflux, to monitor the flux of the calibration lamps.

The data reduction recipes (namely, all but *fors_sumflux*) can handle LSS, MOS and MXU instrument modes, both for FORS1 and FORS2. Development work is still ongoing, as these recipes do not yet support important tasks such as the combination of a sequence of scientific exposures, or the determination of instrument response curves.

The *fors_calib* and the *fors_science* recipes basic functionality is accessible through 17 more low-level recipes: see Section 8.2.2, page 54 for more details.

8.2.1 Required input data

The input data to the recipe *fors_calib* are:

- a sequence of raw flat exposures,
- one raw arc lamp exposure,
- one master bias frame, or a sequence of raw bias exposures, and,
- one arc lamp reference lines catalogue.

where it is expected that flat and arc lamp exposures have been taken quasi-simultaneously, to guarantee that they are affected by the same instrument distortions.

The products of recipe *fors_calib* depend on the nature of the input data (MOS/MXU, LSS or LSS-like, number of available slits, etc.):

- wavelength calibration local models,
- wavelength calibration residuals,
- spatial curvature local models,
- residuals of flat field spectra tracing,
- optical and spectral global distortion model,
- map of wavelengths for each CCD pixel,
- map of spatial coordinate along a slit for each CCD pixel,
- extracted arc lamp spectra,
- location of slits on CCD, and on extracted arc lamp spectra image,
- master flat field,
- normalised master flat field, and,
- spectral resolution table.

The input data to the recipe *fors_science* depend on the nature of the data to reduce (MOS/MXU, LSS or LSS-like, number of available slits, etc.), and not all of them are always required (see Section 9 for more details). At best the following input might be required:

- one scientific exposure, or alternatively one spectrophotometric standard star exposure,
- one master bias frame,

- location of slits on CCD,
- spatial curvature local models, and,
- wavelength calibration local models.

All inputs but the first one are typically produced by *fors_calib*.

The products of recipe *fors_science* depend on what is requested: for instance, an upgrade of the wavelength calibration is only provided in case its alignment to a set of available sky lines is requested. In general the following products are created:

- upgraded wavelength calibration local models,
- upgraded map of wavelengths for each CCD pixel,
- upgraded slit locations on CCD and on rectified image,
- sky lines offsets against expected positions,
- sky subtracted scientific spectra on CCD,
- model sky spectra mapped on CCD,
- extracted slit spectra,
- extracted and sky subtracted slit spectra,
- location of detected objects on extracted slit spectra,
- optimally extracted scientific objects,
- sky spectra corresponding to extracted objects, and,
- error spectra of extracted objects.

8.2.2 Reduction cascade

The monolithic recipes *fors_calib* and *fors_science* are composed by 17 low-level recipes, each carrying out a specific task, in the fashion shown in the workflows on page 55, 56, and 57. Such recipes are directly available, and they can be used for a step-by-step spectroscopic data reduction, enabling the writing of scripts where the basic functionality of the pipeline can be accessed and possibly integrated by alternative data processing algorithms provided by the user. The low-level recipes are the following:

fors_trace_flat, to determine the spatial curvature model.

fors_resample, to remap spatially rectified spectra at a constant wavelength step.

fors_detect_objects, to detect objects in slit spectra.

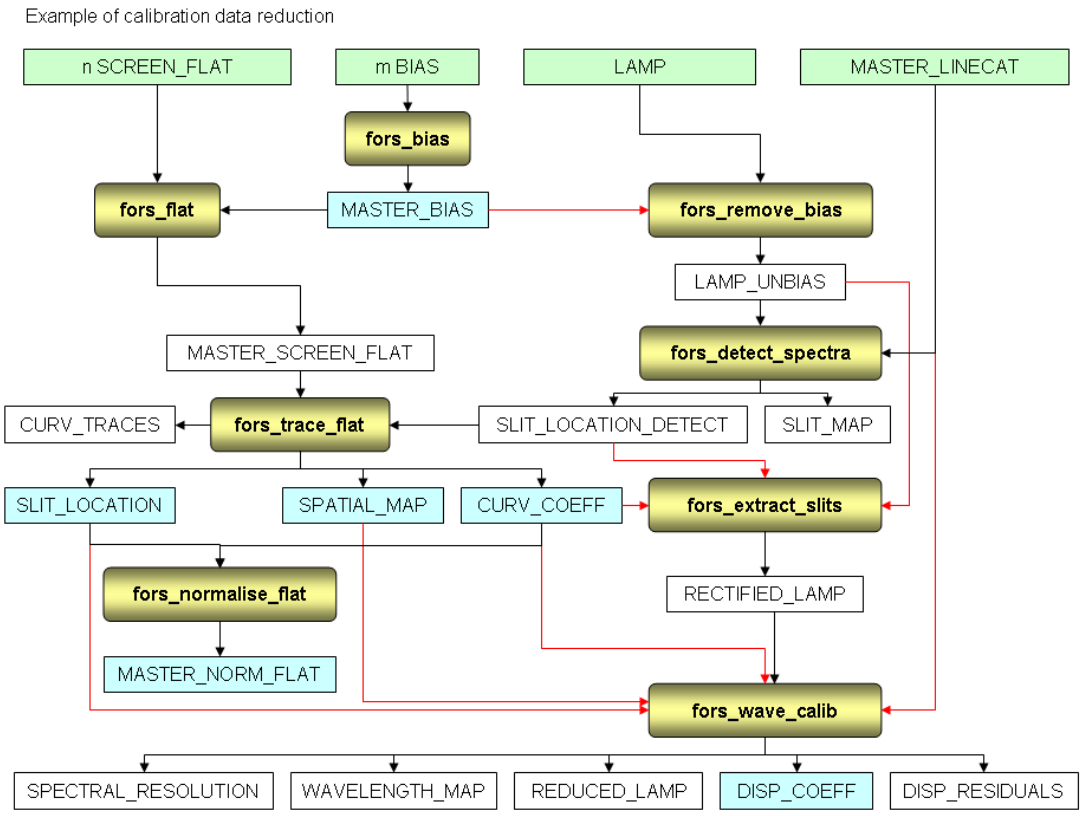

Figure 8.2.1: *This is a possible data reduction flow applied by recipe fors_calib on MOS/MXU data. The rounded yellow boxes represent low-level recipes handling input and intermediate products. The boxes indicating input and product data are labeled with their DO categories, stripped of their variable suffixes (_MOS, _MXU). Note that not all the categories are written to disk by the monolithic fors_calib recipe. The data corresponding to white boxes are not required in further processing of the scientific exposures by recipe fors_science. Different line colors are used not to confuse crossing lines.*

fors_extract_objects, to extract objects in slit spectra.

fors_wave_calib, to derive the dispersion relation from a spatially rectified arc lamp frame.

fors_detect_spectra, to detect MOS/MXU spectra on the CCD.

fors_extract_slits, for spatial rectification of a spectral exposure.

fors_wave_calib_lss, to derive the dispersion relation from a long-slit arc lamp frame.

fors_align_sky, to upgrade the wavelength solution using the sky lines.

fors_flat, to sum the input flat field frames and remove bias.

fors_subtract_sky_lss, to subtract sky from calibrated long slit exposure.

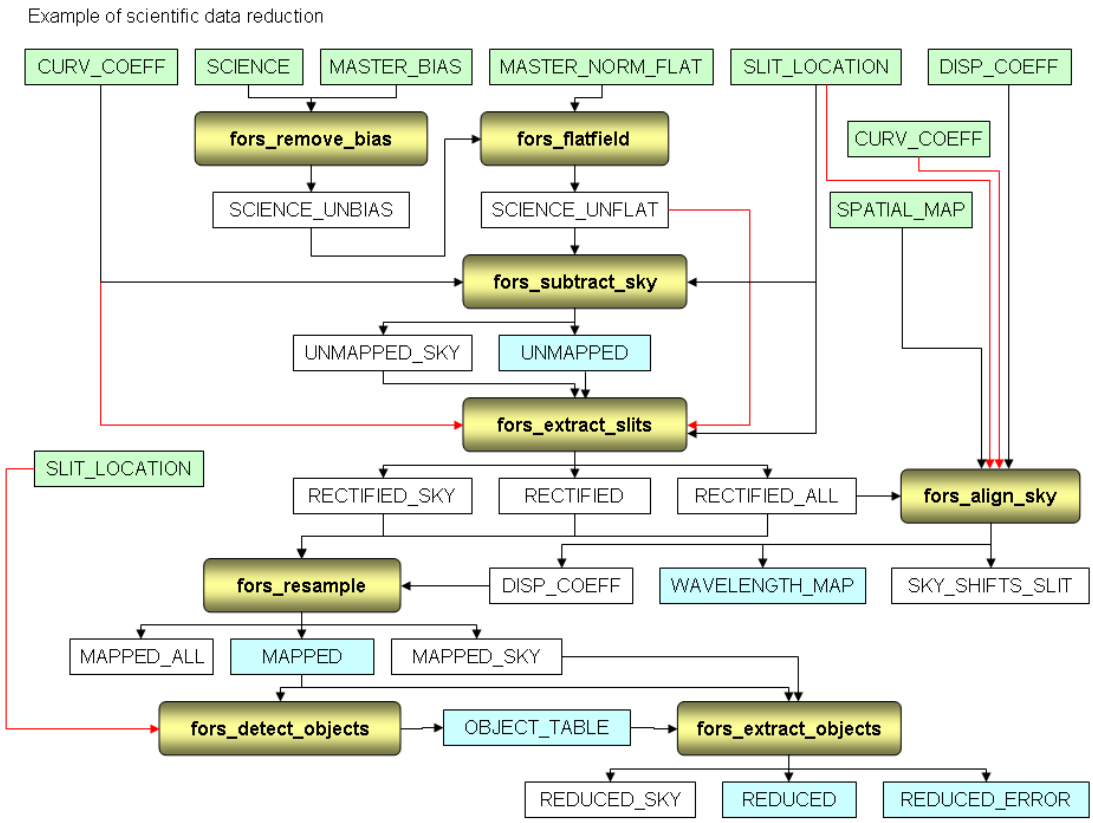

Figure 8.2.2: *This is a possible data reduction flow applied by recipe fors_science on MOS/MXU data. The rounded yellow boxes represent low-level recipes handling input and intermediate products. The boxes indicating input and product data are labeled with their DO categories, stripped of their variable suffixes (_MOS, _MXU, _STD, _SCI, etc.). Note that not all the categories are written to disk by the monolithic fors_science recipe. The data corresponding to white boxes are intermediate products, useful for checking the quality of the results. Different line colors are used not to confuse crossing lines.*

fors_align_sky_lss, to upgrade the wavelength solution using sky lines.

fors_flatfield, for flat field correction of the input frame.

fors_remove_bias, to subtract the bias from the input frame.

fors_subtract_sky, to subtract the sky from scientific spectra.

fors_normalise_flat, to normalise the master flat spectrum.

fors_config, to create alternative grism tables.

These recipes are not documented in detail in this manual, as the description of *fors_calib* and *fors_science* already includes all the necessary details. The on-line help associated to the low-level recipes should be sufficient

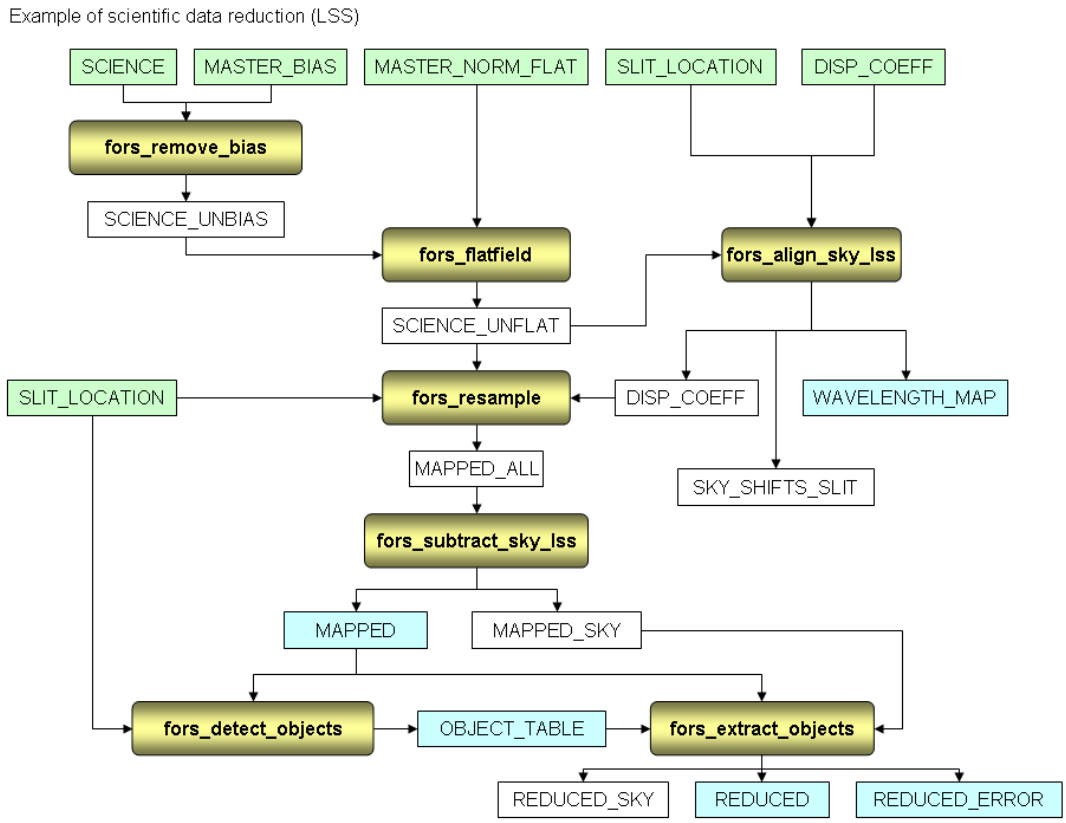

Figure 8.2.3: *This is a possible data reduction flow applied by recipe fors_science on LSS or LSS-like data. The rounded yellow boxes represent low-level recipes handling input and intermediate products. The boxes indicating input and product data are labeled with their DO categories, stripped of their variable suffixes (_LSS, _STD, _SCI, etc.). Note that not all the categories are written to disk by the monolithic fors_science recipe. The data corresponding to white boxes are intermediate products, useful for checking the quality of the results.*

to enable their usage. It is however strongly recommended to use these recipes only in case of overwhelming problems with the high-level spectroscopic recipes, and only after the applied data reduction algorithms are well understood.

8.2.3 VIMOS recipes

The algorithms which are applied in the reduction of FORS1 and FORS2 spectroscopic data are fundamentally instrument-independent. As a byproduct of the preparation of the FORS spectroscopic pipeline, a couple of recipes for the reduction of VIMOS MOS data are also provided:

vimos_calib, to evaluate the spectral extraction mask on the basis of flat and arc calibration lamp exposures, and to create a normalised flat field frame.

vimos_science, to apply the extraction mask and the normalised flat field to the scientific exposures.

These recipes are the same as the *fors_calib* and *fors_science* recipes. The only difference is in the input/output frames tags: the DO categories associated to the data files are the same as the ones of corresponding files handled and created by the VIMOS pipeline, which can be downloaded from [16]. Generally the accuracy of the calibration and the quality of the scientific reduction are significantly better with the new recipes: however, data with spectral multiplexing (typically low-resolution data, i.e., obtained with LR grisms) are not yet supported. Line catalogues and grism tables for VIMOS high-resolution spectral observations are available in the calibration tables included in the FORS pipeline release.

These recipes should just be considered as experimental, and they are offered without any warranty: any feedback on their usage is welcome, as it will be useful for the future improvement of the VIMOS MOS pipeline.

9 Pipeline Recipes Interfaces

In this Section a detailed description of the FORS pipeline recipes interfaces is given, with a complete specification of the recipes usage, their input, output, and configuration parameters. For a overview of the available pipeline recipes, please see Section 8, page 50.

9.1 fors_bias

The FORS pipeline recipe *fors_bias* is used to create a bias master calibration frame from a set of raw bias frames. The overscan regions, if present, are removed from the result.

9.1.1 Input files

BIAS: *required* set of raw, unprocessed bias frames.

9.1.2 Output files

MASTER_BIAS: Bias master calibration frame.

Configuration parameters directly affecting this product are: *--stack_method*.

9.1.3 Configuration parameters

The configuration parameters setting determines the way the *fors_bias* recipe will process the input frames.

--stack_method: Frames combination method. (**average** = simple average of all input frames, **median** = median stacking of all input frames, **minmax** = stacking frames with minmax rejection, **ksigma** = average frames with k-sigma clipping). *Default*: **minmax**

This parameter defines the way the frames will be stacked. The k-sigma clipping method is not yet implemented.

If *--stack_method* is set to *minmax*, the following parameters become relevant:

--minrejection: Number of lowest values to be rejected. *Default*: 1

For each pixel position, the number of lowest pixel values specified here are rejected before computing the mean of the remaining pixel values.

--maxrejection: Number of highest values to be rejected. *Default*: 1

For each pixel position, the number of highest pixel values specified here are rejected before computing the mean of the remaining pixel values. The sum of the number of highest and lowest rejected pixels should be less than the number of input frames.

If *--stack_method* is set to *ksigma*, the following parameters become relevant:

--*klow:* Number of sigmas for lower values rejection. *Default*: 3.0

For each pixel position, a robust determination of the standard deviation from the *median* pixel value is made. All pixel values with a negative residual greater than the specified number of sigmas are rejected, the other values are averaged.

--*khigh:* Number of sigmas for upper values rejection. *Default*: 3.0

For each pixel position, a robust determination of the standard deviation from the *median* pixel value is made. All pixel values with a positive residual greater than the specified number of sigmas are rejected, the other values are averaged.

--*kiter:* Maximum number of iterations. *Default*: 999

Maximum number of iterations of the rejection process. The iteration stops as soon as no outliers are detected, or when reaching the maximum number of iterations. At each iteration the median value and the standard deviation are recomputed, and a new k-sigma rejection is applied.

The ksigma *stacking method is not yet implemented.*

9.1.4 Quality control parameters

Currently the following QC parameters, used by PSO and DFO, are evaluated by the *fors_bias* recipe.

QC BIAS LEVEL: Bias level. *Units*: ADU

Median value of all pixels of the first raw bias in the input set-of-frames.

QC RON: Readout noise. *Units*: ADU

The difference between the first and the second input raw biases is computed. The standard deviation of the pixel values of this difference, divided by $\sqrt{2}$, is taken as the RON.

QC BIAS FPN: Bias fixed pattern noise. *Units*: ADU

The difference between the first two input raw bias frames, the second one shifted by 10×10 pixels in the increasing X- and Y-directions, is computed (where possible). The variance of the difference frame includes contributions of both fixed pattern and read out noise. The standard deviation of the difference is computed and divided by $\sqrt{2}$. The readout noise contribution QC RON is then quadratically subtracted from the result.

QC BIAS STRUCT: Bias structure. *Units*: ADU

The standard deviation of the pixel values of the first input raw bias is computed. This is the combination of structure, fixed pattern, and readout noise. The readout noise QC RON and the fixed pattern QC BIAS FPN contributions are then quadratically subtracted.

QC MBIAS LEVEL: Master bias level. *Units*: ADU

The median value of all pixels of the product master bias.

QC MBIAS RONEXP: Expected master bias readout noise. *Units*: ADU

Propagation of the measured RON (QC RON) according to the number of raw bias frames, and supplied stacking method.

QC MBIAS NOISE: Master bias noise. *Units*: ADU

The standard deviation of the pixel values from the master bias median level is determined, excluding from the computation all values that differ from it more than $3\times$ QC MBIAS RONEXP. This is done in the attempt to minimise deviations not caused by RON.

QC MBIAS NRATIO: Master bias ration observed/expected noise. *Units*: *None*

Ratio between QC MBIAS NOISE and QC MBIAS RONEXP. This ratio is expected to be 1, but it may be less than 1 because of RON overestimation due to incoherent pickup noise, or more than 1 in case of coherent pickup noise that would remain unchanged on the result master bias.

QC MBIAS STRUCT: Structure of master bias. *Units*: ADU

Standard deviation of all master bias pixels values minus the value of QC MBIAS NOISE (geometrically subtracted).

9.2 fors_dark

The FORS pipeline recipe *fors_dark* is used to create a dark master calibration frame from a set of raw dark frames. After bias subtraction, the input frames are combined applying the specified stack method. The combined frame is finally normalised to the unit time. The overscan regions, if present, are removed from the result.

Dark master calibration frames are generally not used in further data reduction steps: dark current in the FORS detectors is negligible, and trying to subtract it would just add noise to the data. Dark frames are just produced for monitoring the instrument.

9.2.1 Input files

DARK: *required* set of raw, unprocessed dark frames.

MASTER_BIAS: *required* master bias frame. Just one should be given.

9.2.2 Output files

MASTER_DARK: Dark master calibration frame.

Configuration parameters directly affecting this product are: *--stack_method*.

9.2.3 Configuration parameters

The configuration parameters setting determines the way the *fors_dark* recipe will process the input frames.

--stack_method: Frames combination method. *Default*: **median**

See explanation in recipe *fors_bias* configuration parameters (Section 9.1.3, page 59).

9.3 fors_img_screen_flat

The FORS pipeline recipe *fors_img_screen_flat* is used to create a screen flat field master calibration frame from a set of raw screen flat field frames. After bias subtraction, the input frames are combined applying the specified stack method. The overscan regions, if present, are removed from the result. The combined frame is then normalised dividing it by its large scale illumination trend. The large scale trend is obtained by applying a median filter with a large kernel, or by polynomial fitting.

9.3.1 Input files

SCREEN_FLAT_IMG: *required* set of raw, unprocessed screen flat field frames.

MASTER_BIAS: *required* master bias frame. Just one should be given.

9.3.2 Output files

MASTER_SCREEN_FLAT_IMG: Screen flat field master calibration frame.

Configuration parameters directly affecting this product are: *--stack_method*, *--xradius*, *--yradius*, *--degree*, and *--sampling*.

9.3.3 Configuration parameters

The configuration parameters setting determines the way the *fors_img_screen_flat* recipe will process the input frames.

--stack_method: Frames combination method. *Default*: **average**

See explanation in recipe *fors_bias* configuration parameters (Section 9.1.3, page 59).

--xradius: Median filter x radius (unbinned pixels). *Default*: 50 pixel

See the *--yradius* parameter.

--yradius: Median filter y radius (unbinned pixels). *Default*: 50 pixel

These parameters define the size of the running box used for smoothing the flat field for determining the large scale trend to remove. These parameters are ignored if the *--degree* parameter is greater than zero.

--degree: Degree of bivariate fitting polynomial. *Default*: -1

If this parameter is greater or equal 0 a polynomial with the specified degree will be fitted to the illuminated part of the CCD for determining the flat field large scale trend to remove.

--sampling: Sampling interval for fitting. *Default*: 100

If the parameter *--degree* is greater than 0, a polynomial will be fitted to the illuminated part of the CCD, sampling pixel values at the specified step.

9.3.4 Quality control parameters

Currently the following QC parameters, used by PSO and DFO, are evaluated by the *fors_img_screen_flat* recipe.

QC OVEREXPO: Percentage of overexposed pixels. *Units*: *none*

A saturated pixel is defined as a pixel having value either 65535 or 0 ADUs. In its practical implementation, a saturated pixel is identified by having a value either greater than 65534 or less than 1 ADUs. Saturated pixels are counted in the original raw flat field exposures before any processing (including bias subtraction), and excluding the overscan regions. The total number of saturated pixels is divided by the total number of pixels examined in all input raw frames, and the result is multiplied by 100.

QC FLAT EFF: Flat field lamp efficiency. *Units*: ADU/s

The efficiency is computed as the median of all pixels of the first input raw frame (bias subtracted), divided by its exposure time.

QC FLAT PHN: Photon noise in master screen flat field. *Units*: ADU

The photon noise is estimated in the following way: the difference frame between the first two input raw flat field frames is determined for their central 100x100 region. The standard deviation of all pixels of this difference frame is computed, and then divided by $\sqrt{2N}$ where N is the total number of raw flat field frames contributing to the master flat field. The RON contribution is considered negligible.

QC FLAT FPN: Fixed-pattern noise. *Units*: ADU

The FPN is estimated in the following way: The difference of two 100x100 sub-frames of the same master flat field image is determined. The first subframe is extracted at the frame center, while the second is extracted from a position shifted by 10 pixels in the increasing X- and Y-directions. The standard deviation of the difference frame, divided by $\sqrt{2}$, minus the photon noise QC FLAT PHN (geometrically subtracted) yields the fixed-pattern noise. The RON is considered negligible in the computation of FPN.

QC FLAT FPN REL: Relative fixed-pattern noise. *Units*: *none*

This quantity is the fixed pattern noise (QC FLAT FPN) divided by the median illumination level of the flat field exposure.

QC FLAT CONAD: Conversion factor from ADU to electrons. *Units*: e^-/ADU

If the exposure time of the first two input raw screen flat fields in the input set of frames is the same (within 4%), the absolute difference frame is computed. At each pixel the difference frame is divided by 2 and by the signal of the first frame. This frame estimates the gain (in ADU/e^-) at each pixel, and the conversion factor is computed as the reciprocal mean of this frame.

QC FLAT CONADERR: Error on conversion factor from ADU to electrons. *Units*: e^-/ADU

This quantity is computed as the standard deviation of the gain frame, also used for the computation of QC FLAT CONAD, multiplied by QC FLAT CONAD squared.

9.4 fors_img_sky_flat

The FORS pipeline recipe *fors_img_sky_flat* is used to create a twilight master calibration frame from a set of raw twilight sky exposures. In order to eliminate the contributions of field stars on the jittered sequence of flat fields, the frame combination method must be based on a rejection algorithm (rather than on a simple average). The overscan regions, if present, are removed from the result.

9.4.1 Input files

SKY_FLAT_IMG: *required* set of raw, unprocessed sky flat field frames.

MASTER_BIAS: *required* master bias frame. Just one should be given.

9.4.2 Output files

MASTER_SKY_FLAT_IMG: Sky flat field master calibration frame.

Configuration parameters directly affecting this product are: *--stack_method*.

9.4.3 Configuration parameters

The configuration parameters setting determines the way the *fors_img_sky_flat* recipe will process the input frames.

--stack_method: Frames combination method. *Default*: **median**

See explanation in recipe *fors_bias* configuration parameters (Section 9.1.3, page 59).

9.5 fors_img_science

The FORS pipeline recipe *fors_img_science* is used to reduce a direct imaging scientific exposure (including the pre-imaging performed for the preparation of spectroscopic observations). The bias master calibration is subtracted. The unbiased signal is then divided by the normalised sky flat field, and the overscan regions, if present, are removed from the result. The calibrated image is finally sent to a source detection and extraction application (SExtractor 2.5.0 [18]). The recipe *fors_img_science* is also used to reduce preimaging exposures (made for the preparation of spectroscopic observations).

9.5.1 Input files

SCIENCE_IMG: *required* direct imaging scientific exposure.

MASTER_BIAS: *required* bias master calibration frame.

MASTER_SKY_FLAT_IMG: *required* sky flat field master calibration frame (normalised or not).

9.5.2 Output files

SCIENCE_REDUCED_IMG: Reduced science image.

Configuration parameters directly affecting this product are: *--cr_remove*.

PHOT_BACKGROUND_SCI_IMG: Background map produced by SExtractor [18].

Configuration parameters directly affecting this product are: *--sex_radius*.

SOURCES_SCI_IMG: List of detected sources, SExtractor's cut (uncensored). The content of this table exclusively depends on the SExtractor configuration parameters setting. Please refer to [17] and [19] for details.

Configuration parameters directly affecting this product are: *--sex_config*, *--sex_mag*, *--sex_magerr*.

OBJECT_TABLE_SCI_IMG: Detected sources and source properties used by the pipeline. This table is the result of a selection applied to the list of detected sources produced by SExtractor: objects with either status FLAGS different from zero, or negative SExtractor FWHM_IMAGE, or no computed magnitude, are excluded from the object table. In addition to that, only some of the sources parameters are carried over to the object table:

x:	x CCD position (baricenter) of detected object. Identical to SExtractor parameter X_IMAGE.
y:	y CCD position (baricenter) of detected object. Identical to SExtractor parameter Y_IMAGE.
fwhm:	Source width at half maximum, from gaussian fit. Identical to SExtractor parameter FWHM_IMAGE.
a:	Semi-major axis of object flux distribution, defined as max RMS of the object profile along any direction. Identical to SExtractor parameter A_IMAGE.
b:	Semi-minor axis of object flux distribution, defined as min RMS of the object profile along any direction. Identical to SExtractor parameter B_IMAGE.
theta:	Position angle between semi-major axis and the horizontal axis, counted counter-clockwise, in the domain $[-\pi/2, \pi/2[$. Identical to SExtractor parameter THETA_IMAGE.
ell:	Derived as $1 - a/b$.
instr_mag:	Instrumental magnitude, corresponding to the chosen SExtractor magnitude. As a default, identical to SExtractor parameter MAG_APER.
dinstr_mag:	Error on instrumental magnitude. As a default, identical to SExtractor parameter MAGERR_APER.
class_star:	Stellarity index, 1 = star, 0 = galaxy. Identical to SExtractor parameter CLASS_STAR.

Configuration parameters directly affecting this product are: *--sex_config*, *--sex_mag*, *--sex_magerr*.

9.5.3 Configuration parameters

The configuration parameters setting determines the way the *fors_img_science* recipe will process the input frames.

--*extract_method:* Source extraction method. (**sex** = Use SExtractor, **test** = Test recipe *fors_img_science*). *Default*: **sex**

The "test" method is just used to check that the recipe works properly on internal data, independently on external source detection and extraction programs such SExtractor. This is not interesting for the general user.

--*sex_exe:* Path to SExtractor executable. *Default*: *installation_path*/bin/sex

This parameter can be changed in order to use other SExtractor installations, different from the one installed with the pipeline. In principle the pipeline should work also with previous (and likely future) SExtractor releases. This parameter could also be set to point to a user defined program, for example a script which calls SExtractor and then does some postprocessing of SExtractor's output before control is returned to the pipeline. The only requirement is that the specified command must behave like the SExtractor executable in terms of command line parameters and output tables used by the pipeline.

--*sex_config:* Path to SExtractor configuration file. *Default*: *installation_path*/share/fors/config/fors.sex

The SExtractor configuration file pointed by this parameter can be edited and modified, to control the way SExtractor operates. The SExtractor output parameters file, fors.param, can also be found in the same directory: it is a list of the source parameters SExtractor would compute (and that would be written to the SOURCES_SCI_IMG table, see previous Section). No entries should be removed from this file, because the pipeline recipes rely on them: just new ones may be added. The parameter --*sex_config* can also be changed in order to use SExtractor configuration files different from the one installed with the pipeline.

--*sex_mag:* SExtractor magnitude used by recipe. *Default*: MAG_APER

SExtractor computes instrumental magnitudes in different ways. Each computation method corresponds to one entry in the SExtractor output parameters file, fors.param (see previous recipe parameter, --*sex_config*): MAG_BEST, MAG_AUTO, MAG_ISO, MAG_ISOCOR, MAG_BEST, are possible choices (see the SExtractor User's Guide [17] for details). The default is a simple aperture magnitude, MAG_APER, where the aperture diameter (in pixel) is defined by the SExtractor configuration parameter PHOT_APERTURES. To modify the monitored magnitude, ensure that the corresponding entry is present in the SExtractor output parameters file, fors.param.

--*sex_magerr:* SExtractor error on computed magnitudes. *Default*: MAGERR_APER

The default is the error on the simple aperture magnitude, MAG_APER.

--*sex_radius:* Median filter radius for background map computation. *Default*: 64 (unbinned pixels)

This parameter overwrites the value of the parameter BACK_SIZE in the specified SExtractor configuration file (see recipe parameter --*sex_config*).

--*cr_remove:* Remove cosmic ray events from exposure. *Default*: false

Cosmic rays are detected and removed from image (by interpolation of surrounding pixels).

This parameter is not active yet.

9.5.4 Quality control parameters

Currently the following QC parameters, used by PSO and DFO, are evaluated by the *fors_science* recipe.

QC SKYAVG: Mean of sky background. *Units*: mag/arcsec^2

The background determination is part of the source extraction algorithm (provided by SExtractor). This QC parameter is defined as $M = -2.5 \log_{10}(F/A)$, where F is the mean of the background map, and A the area of one image pixel in arcsec^2. Note that this is an instrumental magnitude, where neither color correction nor zeropoint is applied.

QC SKYMED: Median of sky background. *Units*: mag/arcsec^2

The background determination is part of the source extraction algorithm (provided by SExtractor). This QC parameter is defined as $M = -2.5 \log_{10}(F/A)$, where F is the median of the background map, and A the area of one image pixel in arcsec^2. Note that this is an instrumental magnitude, where neither color correction nor zeropoint is applied.

QC SKYRMS: Standard deviation of sky background. *Units*: mag/arcsec^2

Standard deviation of the smoothed background level from its median value. The standard deviation is determined in term of flux (ADU/s) and then propagated to instrumental magnitude units. More specifically, if ΔF is the variation of flux, the corresponding variation in magnitudes ΔM is the differential of $M = -2.5 \log_{10}(F/A)$,

$$\Delta M = -2.5 \frac{\Delta F}{F} \log_{10} e = 1.086 \frac{\Delta F}{F}$$

The background determination is part of the source extraction algorithm (provided by SExtractor).

QC IMGQU: Image quality of scientific exposure. *Units*: arcsec

Median FWHM (as computed by SExtractor) of stars detected on a calibrated scientific exposure. An object is considered a star if the corresponding SExtractor stellarity index is greater than 0.7 and the FWHM is larger than 1 pixel. *This is currently not working correctly*.

QC IMGQUERR: Uncertainty of image quality. *Units*: arcsec

Population standard deviation of the FWHM values obtained for each detected source from their median value (see parameter QC IMGQU).

QC STELLAVG: Mean stellarity index. *Units*: *None*

Mean SExtractor stellarity index of all objects used for QC IMGQU computation.

QC IMGQUELL: Mean star ellipticity. *Units*: *None*

Mean ellipticity of all objects used for QC IMGQU computation.

QC IMGQUELLERR: Standard deviation of star ellipticities. *Units*: *None*

Computed as 0.6745 times the median absolute deviation with respect to the median ellipticity of all objects used for QC IMGQU computation.

9.6 fors_zeropoint

The FORS pipeline recipe *fors_zeropoint* is used to estimate from one imaging exposure on a photometric standard stars field the magnitude zeropoint assuming a nominal atmospheric extinction, as well as the atmospheric extinction assuming a nominal zeropoint.

The bias master calibration is subtracted from the raw exposure. The unbiased signal is then divided by the normalised sky flat field produced by the recipe *fors_img_sky_flat*, and the overscan regions, if present, are removed from the result. The calibrated image is then sent to a source detection and extraction application (SExtractor 2.5.0 [18]). The detected sources are compared to a catalogue of standard stars for identification. The recipe *fors_zeropoint* uses whatever catalogue is specified in input, either Landolt's or Stetson's, or even both. If both catalogues are specified, they are merged before being used. In the merged catalogue, if two stars are within 5 arcseconds they are considered identical and the one with the largest magnitude error is removed.[9] The comparison of observed *vs* catalogue positions is made, whenever possible, applying point-pattern-matching techniques (described in detail in Section 10.4.6, page 120). If pattern matching either fails or is not applicable (e.g., too few standard stars are present in the field-of-view), then the stars identification will be entirely based on the sky-to-CCD transformation specified in the input image FITS header (and if this transformation is incorrect the recipe will definitely fail). Finally, the difference between the catalogue magnitude (corrected for the transmission curve difference between the used filter and the catalogue filter) and the instrumental magnitude (based on electron counts, *not* ADUs, and corrected to airmass zero), is optimally averaged on all the identified standard stars. The derived quantity is conventionally referred to as the *frame zeropoint* Z.

In this Section and in the following (page 73), please always refer to table 9.6.1 for the meaning of the symbols used for the quantities involved in the photometric calibration.

Symbol	Definition	Unit
M	Catalogue magnitude	mag
C	Catalogue color	mag
Γ	Linear color correction term	
$M^\star$	Color corrected catalogue magnitude ($M - \Gamma \cdot C$)	mag
m	Observed instrumental magnitude	mag
A	Airmass	airmass
E	Atmospheric extinction coefficient	mag/airmass
Z	Zeropoint	mag
g	Gain	ADU/e^-
t	Exposure time	s

Table 9.6.1: *Photometry related symbols.*

[9] It should be noted that using simultaneously two different catalogues carries a significant risk to produce inconsistent results, as the magnitudes are not derived with the same method in both catalogues, and as a consequence common stars can differ by up to 0.2 magnitudes. The use of two catalogues is therefore strongly discouraged.

The extinction coefficient corresponding to the estimated zeropoint Z is computed as

$$E = E_o + \frac{(Z_o - Z)}{A}$$

The assumptions on a clear night atmospheric extinction E_o and instrument zeropoint Z_o are listed for each FORS filter in the photometric table (see Section 7.1, page 44).

The zeropoint and the atmospheric extinction computed by this recipe have the sole purpose of monitoring the instrument+telescope system and the quality of the atmosphere. With only one exposure it is impossible to obtain actual determinations of either the atmospheric extinction coefficient (mag/airmass) or the instrument zeropoint.[10] In order to evaluate them both at least two different exposures of standard star fields (not necessarily of the same field), obtained at (very) different airmasses, would be needed. This task is performed by the offline pipeline recipe *fors_photometry* (see Section 9.7, page 73).

9.6.1 Input files

STANDARD_IMG: *required* photometric standard stars field exposure.

MASTER_BIAS: *required* bias master calibration frame.

MASTER_SKY_FLAT_IMG: *required* sky flat field master calibration frame (normalised or not).

FLX_STD_IMG: *required* photometric standard stars catalogue (see Section 7.2, page 45).

PHOT_TABLE: *required* photometric table (see Section 7.1, page 44).

9.6.2 Output files

STANDARD_REDUCED_IMG: Reduced standard stars field image.

Configuration parameters directly affecting this product are: *none*.

PHOT_BACKGROUND_STD_IMG: Background map produced by SExtractor.

Configuration parameters directly affecting this product are: *--sex_radius*.

SOURCES_STD_IMG: List of detected sources, SExtractor's cut (uncensored). The content of this table exclusively depends on the SExtractor configuration parameters setting. Please refer to [17] and [19] for details.

[10] The problem is that there are two unknowns, E and Z, in the equation

$$Z = M - m + EA$$

(for a given star at a given filter, and ignoring for simplicity the color correction term). This means that Z can be evaluated by making an assumption on E, or the other way around. If an assumption is made on E, a Z will be derived which will include all absorption effects, from atmosphere, mirror, instrument optics, etc. If Z is fixed, the derived E will include as well the same mixed absorption effects. Clearly the zeropoint Z and the atmospheric extinction E computed by the recipe *fors_zeropoint* are related by

$$(Z - Z_o) = -A(E - E_o)$$

therefore computing them both does not add information, but it rather offers two different views to instrument+atmosphere monitoring.

Configuration parameters directly affecting this product are: *--sex_config*, *--sex_mag*, *--sex_magerr*.

ALIGNED_PHOT: Detected sources and source properties selected by the pipeline, with zeropoint estimate for each object identified as a standard star. This table is an expansion of the OBJECT_TABLE_SCI_IMG table produced by the recipe *fors_img_science* (see Section 9.5.2, page 65). To the columns contained in the OBJECT_TABLE_SCI_IMG, the following columns are added:

instr_cmag: Instrumental magnitude, corrected to detector gain = 1 (electrons), unit time (flux), and airmass zero (assuming the clear night atmospheric extinction E_o reported in the input PHOT_TABLE). This quantity is computed as

$$m + 2.5 \log g + 2.5 \log t - E_o A$$

dinstr_cmag: Error on corrected instrumental magnitude.
object: Name of identified standard star (from the input catalogue).
ra: Right Ascension of identified standard star (from the input catalogue).
dec: Declination of identified standard star (from the input catalogue).
cat_mag: Catalogue standard star magnitude (M).
dcat_mag: Error on catalogue magnitude (ΔM).
color: Color index of standard star (C), chosen as in Section 7.1, page 44.
mag: Color corrected catalogue magnitude ($M^\star$).
dmag: Error on color corrected catalogue magnitude ($\Delta M^\star$).
use_cat: Use (1) or do not use (0) the flagged standard star in recipe *fors_photometry*. The recipe *fors_zeropoint* will always set this flag to 1, but the user has the possibility to exclude it from further processing.
shift_x: Horizontal offset (in pixels) of identified standard star from expected position computed using the sky-to-CCD transformation of the input image.
shift_y: Vertical offset (in pixels) of identified standard star from expected position computed using the sky-to-CCD transformation of the input image.
zeropoint: Star zeropoint (Z), computed as **mag - instr_cmag**, that is

$$Z = M^\star - m - 2.5 \log g - 2.5 \log t + E_o A$$

dzeropoint: Error on zeropoint (ΔZ).
weight: This number is related to the importance given to each computed star zeropoint in the computation of the optimally averaged zeropoint. Some of the weights might be negative, because the errors on individual zeropoints are correlated (if the covariance matrix is not diagonal the optimal weights are no longer the inverse of the variance, and may happen to be negative: see Sections 10.2.1 and 10.2.2 for a mathematical treatment of this problem).

Configuration parameters directly affecting this product are: *--sex_config*, *--sex_mag*, *--sex_magerr*.

9.6.3 Configuration parameters

The configuration parameters setting determines the way the *fors_zeropoint* recipe will process the input frames.

--*extract_method:* Source extraction method. (**sex** = Use SExtractor, **test** = Test recipe *fors_img_science*). *Default*: **sex**

--*sex_exe:* Path to SExtractor executable. *Default*: *installation_path*/bin/sex

--*sex_config:* Path to SExtractor configuration file. *Default*: *installation_path*/share/fors/config/fors.sex

--*sex_mag:* SExtractor magnitude used by recipe. *Default*: MAG_APER

--*sex_magerr:* SExtractor error on computed magnitudes. *Default*: MAGERR_APER

--*sex_radius:* Median filter radius for background map computation. *Default*: 64 (unbinned pixels)

The above configuration parameters are in common with the recipe *fors_img_science*. They are explained in Section 9.5.3, page 66. Further parameters of recipe *fors_zeropoint* are:

--*ncat:* Number of catalogue stars to use in pattern matching. *Default*: 10

A limited number of catalogue stars (ideally, just 3) can be used to define a pattern to be searched within the position distribution of the detected objects. The sky-to-CCD transformation associated to the imaging exposure is used to select from the input catalogue the stars that are expected to be found in the field-of-view. The parameter --*ncat* specifies how many of these stars should be used to compose a pattern. The stars are selected starting from the brightest ones.

--*nsource:* Number of detected sources for each entry in pattern. *Default*: 3

The point-pattern made of selected catalogue stars (see parameter --*ncat*) must be searched among the detected objects. Typically there are hundreds of detected objects available, and to perform all possible comparisons would require a long execution time (with no real improvement of the final result). Therefore just a subset of detected object (starting from the brightest ones) needs to be selected for pattern comparison. The parameter --*nsource* controls how many detected objects should be selected for matching. The number of selected objects is given by

$$N \quad = \quad nsource \cdot ncat$$

As a default, *nsource* $= 3$ and *ncat* $= 10$, so 30 detected objects would be used in the pattern recognition process.

Once the pattern is recognised, an improved transformation from catalogue to CCD positions can be determined, and all the standard stars in the field of view can be safely identified. In other words, setting the parameters --*nsource* and --*ncat* does not limit the actual number of standard stars that can be finally identified.

--*kappa:* Number of sigmas in triangular pattern scale and angle rejection. *Default*: 5.0 sigma

Pattern matching is the result of several triangles comparisons (see details in Section 10.4.6, page 120). When similar triangles are found, the corresponding rotation angle and scale ratio can be determined. In the determination of the mean transformation, angles and scales that deviate more than the specified number of sigmas are excluded and the process iterated till covergence is reached.

--*search:* Start search radius applied to find standard stars in field of view. *Default*: 20 pixel

No matter whether pattern matching was successful or not, the catalogue sky positions are converted into CCD expected positions for each standard star. If pattern matching was successful, the sky-to-CCD transformation would be corrected by mean offset, angle, and scale, before being applied. The *search* parameter indicates what the search radius for a given star around its expected CCD position.

--*maxsearch:* Max search radius applied to find standard stars in field of view. *Default*: 20 pixel

If no sources are found applying the initial search radius, the search radius is increased by steps of 5 pixels till the maximum radius specified by this parameter.

--*magcutE:* Expected max systematic error. *Default*: 1.0 mag

This threshold is applied in zeropoint computation. Any individual (star) zeropoint which deviates from the best estimate by more than the specified threshold is excluded, and the zeropoint estimation is iterated. The sense of this parameter is that any deviation higher than specified should be interpreted as an object misidentification (since it couldn't possibly be explained by a systematic error in the determination of the color term).

--*magcutk:* Number of sigmas in individual zeropoint rejection. *Default*: 5.0 sigma

This threshold is applied in zeropoint computation, after the *magcutE* screening, and it deals with purely statistical errors. Any individual (star) zeropoint which deviates from the best estimate by more than the specified number of sigmas is excluded, and the zeropoint estimation is iterated. The sense of this parameter is that any deviation higher than specified should be interpreted as zeropoint miscalculation due to unexpected effects (such as imperfections of the detector), as it could hardly be explained by random error.

9.6.4 Quality control parameters

Currently the following QC parameters, used by PSO and DFO, are evaluated by the *fors_zeropoint* recipe.

QC ZPOINT: Frame zeropoint. *Units*: mag

See subsections of 9.6 above, and Section 10.2 on page 106.

QC ZPOINTRMS: Uncertainty of frame zeropoint. *Units*: mag

This is the expected random statistical error propagated from both observed and catalogue quantities (see Section 10.2, page 106).

QC ZPOINT NSTARS: Number of stars used for zeropoint computation. *Units*: *None*

QC EXTCOEFF: Atmospheric extinction coefficient. *Units*: mag/airmass

Atmospheric extinction coefficient E required to explain the difference between $Z =$ QC ZPOINT and the nominal expected zeropoint Z_o given as input to the recipe, that is $E = (Z_o - Z)/A + E_o$, where A is the airmass, and E_o is the atmospheric extinction coefficient that was assumed for the computation of Z (see begin of this Section, page 68).

QC EXTCOEFFERR: Error on atmospheric extinction coefficient. *Units*: mag/airmass

Error propagation on the computation of QC EXTCOEFF.

9.7 fors_photometry

This recipe is used for the processing of several ALIGNED_PHOT tables produced by the recipe *fors_zeropoint*, enabling the estimation of quantities such as the best atmospheric extinction and linear color correction terms related to a filter. In particular it enables accurate photometry by determining, from ALIGNED_PHOT tables derived from a set of dithered (rotated, translated) exposures of a photometric standard star field, a correction map relative to a given flat field frame (the same used in the *fors_zeropoint* processing). This method is described in [25].

The general model for the observed instrumental magnitude m_{ij} of star i on exposure j is

$$m_{ij} = M_i - \Gamma \cdot C_i + p(A_j, C_i) + EA_j - Z_j - 2.5 \log g_j - 2.5 \log t_j + f(x_{ij}, y_{ij})$$

where the symbols are consistent with table 9.6.1 (page 68), and x and y are detector coordinates in pixel. All quantities are referred to the i-th star in the j-th exposure as indicated. Non-linear dependencies are accounted for by the functions $p()$ and $f()$.

In the *fors_photometry* implementation $p()$ is a d-order polynomial with zero and first order terms set to zero, that is:

$$p(A, C) = \sum_{r=2}^{d} \sum_{k=0}^{r} p_{k,r-k} C^k A^{r-k}$$

The lower order terms are redundant, as p_{oo} would be the zeropoint, p_{o1} the atmospheric extinction E, and p_{1o} the linear color correction term Γ, which are already explicit in the general equation.

Also $f()$ is a polynomial modeling of the deviations which may be present in the used sky flat field.[11] In this polynomial only the constant term is set to zero:

$$f(x, y) = \sum_{r=1}^{d} \sum_{k=0}^{r} f_{k,r-k} x^k y^{r-k}$$

The constant term is redundant, as f_{oo} would be degenerate with M_i.

Indicating in the general equation $M_i - \Gamma \cdot C_i = M_i^\star$ (the color corrected catalogue magnitude, as reported in table 9.6.1, page 68), in the equation

$$m_{ij} = M_i^\star + p(A_j, C_i) + EA_j - Z_j - 2.5 \log g_j - 2.5 \log t_j + f(x_{ij}, y_{ij})$$

the following parameters can be assumed fixed or free:[12]

$M_i^\star$ It can be fixed or free for each individual star. For the i-th star, if the corresponding entry in the USE_CAT column of the input ALIGNED_PHOT table is set to 1, M_i and C_i are fixed to the catalogue values ($M_i^\star$ is fixed); otherwise the color corrected magnitude, $M_i^\star$, is determined ($M_i^\star$ is free). Since only relative magnitudes can be measured, the magnitude of at least one star

[11] This implies that the input ALIGNED_PHOT tables must all be produced by recipe *fors_zeropoint* runs where the same flat field correction was applied.

[12] Fixed parameters are input to the model (from either measurement or by assuming nominal values). The free parameters of the model are determined by the linear fit.

must be assumed; this defines the overall scale of the measured magnitudes (see below).

Γ The color term can be a free parameter if two or more M_i and C_i are provided (i.e., $M_i^\star$ is fixed). Physically, this corresponds to determining the color term by comparing the instrumental magnitudes of a star with known color to its physical magnitude; however, the magnitude of one more star must be provided in order to define the overall zeropoint.

Z_j Either the zeropoint for each exposure or a global zeropoint Z can be fitted.

E The atmospheric extinction coefficient can be determined, if two or more ALIGNED_PHOT tables at different airmass are provided.

f, p The number of free parameters are determined by the user specified degrees of the polynomial. For example, if both degrees are set to zero, the sums would be empty and there will be no polynomial coefficients to fit. It is not possible for the user to provide as input certain (fixed) values of the polynomial coefficients. To achieve the same effect, a corrected master flat field can be used in *fors_zeropoint* to compute the ALIGNED_PHOT tables.

Note that the degeneracy between constant terms is handled here by setting $f_{oo} = 0$ and $M_o^\star = M_o$, unlike in [25] where $M_o^\star = Z_o = 0$ is set. Note also that some combinations (such as fitting Z_j for each input ALIGNED_PHOT table along with E) do not make sense and are not allowed. The errors of g_j and t_j are not propagated because these are usually insignificant in the overall error budget, and not readily available.

The method described in [25] Appendix A corresponds to the following configuration:

$M_i^\star$ free
Γ fixed
Z_j all free
E fixed
f non-zero order, d
p zero order (no free parameters)

The method described in [25] Appendix B corresponds to

$M_i^\star$ fixed
Γ free
Z_j one free parameter, Z, for all frames
E free
f non-zero order, d
p second order (one free parameter)

For a more detailed description of the determination of the model's free parameters, see the Section 10.3, page 112.

The *fors_photometry* implementation identifies common stars in different frames by their celestial coordinates (RA, Dec). Therefore only photometric standard stars, for which RA and Dec are available from the catalogue, can be used. In principle all of the available sources could be used, if the astrometric information from the identified stars would be applied to align the coordinate systems of all exposures. This is not yet implemented.

9.7.1 Input files

ALIGNED_PHOT: *required* set of one or more tables of detected sources, as produced by the recipe *fors_zeropoint*.

MASTER_SKY_FLAT_IMG: *required* sky flat field master calibration frame as produced by the recipe *fors_img_sky_flat*. It should be the same frame used in all the runs of *fors_zeropoint* which produced the input ALIGNED_PHOT tables. The recipe will print warning messages if the flat fields used to compute any ALIGNED_PHOT table is different from this frame.

9.7.2 Output files

CORRECTION_MAP: Flat field correction map (magnitude), the evaluation of f at each pixel.

Configuration parameters directly affecting this product are: *all*.

CORRECTION_FACTOR: Flat field correction map (flux). The input flat field must be multiplied by this map in order to produce the corrected flat field.

Configuration parameters directly affecting this product are: *all*.

MASTER_FLAT_IMG: Corrected flat field.

Configuration parameters directly affecting this product are: *all*.

9.7.3 Configuration parameters

The configuration parameters setting determines the way the *fors_photometry* recipe will process the input files.

--*fitz:* Fit a zeropoint for each input table (Z_j). *Default*: true

If set to false, a common zeropoint Z is derived for all input tables.

--*fitm:* Fit star magnitudes ($M_i^\star$). *Default*: true

If set to true, all magnitudes are fitted. If set to false, all magnitudes are fixed, unless the corresponding entry in the USE_CAT column of the input ALIGNED_PHOT table is set to 0 (meaning "do not use this catalogue magnitude"). In either case, there is no effect on the magnitude of the first standard star in the first input table, which remains frozen by definition to its catalogue value (even if USE_CAT is set to 0).

--*fitx:* Fit atmospheric extinction coefficient (E). *Default*: false

If set to true, it would be appropriate to set --*fitz* to "false". If all input tables include only sources from the same airmass, the recipe would probably fail.

--*fitc:* Fit linear color correction term (Γ) *Default*: false

If set to true, the linear color correction term is fit.

--*degreef1:* Correction map polynomial degree *Default*: 2

--degreef2: Correction map polynomial degree *Default*: -1

Degree and type of correction polynomial: if both degrees are positive numbers the matrix of polynomial coefficients is rectangular (*degreef1* is related to x and *degreef2* to y directions), while if *degreef2* is negative the coefficients matrix is triangular of size *degreef1*.

--degreep: Polynomial degree of non linear color correction. *Default*: 0

9.7.4 Quality control parameters

Currently the following QC parameters, used by DFO, are evaluated by the *fors_photometry* recipe. See subsections of 9.7 above, and Section 10.3 on page 112 for more details on how they are computed.

QC INSTRUMENT ZEROPOINT: Instrument zeropoint. *Units*: mag

This is the *true* instrument zeropoint (different from the one computed by recipe *fors_zeropoint*, see Section 9.6, page 68). This quantity is computed only if *-fitz*=false.

QC INSTRUMENT ZEROPOINT ERROR: Error on instrument zeropoint. *Units*: mag

This is the expected random statistical error propagated from both observed and catalogue quantities.

QC ATMOSPHERIC EXTINCTION: Atmospheric extinction coefficient. *Units*: mag/airmass

This is the *true* (mean) atmospheric extinction (different from the one computed by recipe *fors_zeropoint*, see Section 9.6, page 68). This quantity is computed only if *-fitx*=true.

QC ATMOSPHERIC EXTINCTION ERROR: Error on atmospheric extinction. *Units*: mag/airmass

This is the expected random statistical error propagated from both observed and catalogue quantities.

QC COLOR CORRECTION: Linear color correction term. *Units*: *none*

This quantity is computed only if *-fitc*=true.

QC COLOR CORRECTION ERROR: Error color correction term. *Units*: *none*

This is the expected random statistical error propagated from both observed and catalogue quantities.

9.8 fors_calib

This recipe identifies reference lines on LSS, MOS and MXU arc lamp exposures, and traces (if available) the spectral edges on the associated flat field exposures. With this information the spectral extraction mask to be applied in the scientific data reduction is determined. From the input flat field exposures a normalised flat field frame is also derived.

The recipe *fors_calib* can process both FORS1 and FORS2 frames. The input arc lamp and flat field exposures are assumed to be obtained quasi-simultaneously, so that they would be described by exactly the same optical and spectral distortions.

In the following sections the MXU acronym in the products names can also be read MOS, or LSS, unless indicated otherwise.

9.8.1 Input files

In alphabetical order:

GRISM_TABLE: *optional* grism table. This table defines a subset of recipe configuration parameters controlling the way spectra are extracted for any particular grism. A set of standard grism tables is provided with the pipeline (see Section 7, page 47 for details).

LAMP_MXU: *required* raw arc lamp spectrum exposure. Just one frame should be specified.

MASTER_BIAS: *required* master bias frame. Just one should be given. Alternatively, it is possible to specify a set of raw, unprocessed bias frames (DO category: BIAS): in this case the input BIAS frames are used for the creation of an internal median MASTER_BIAS frame, that is finally added to the recipe products for further use by other recipes.

MASTER_LINECAT: *required* line catalogue. It must contain the reference wavelengths (in Ångstrom) for the arc lamp used. The only requirement for this table is to contain a column listing such wavelengths, whose name may be specified using the configuration parameter *--wcolumn* (see Section 9.8.3, page 86). A standard line catalogue is also provided with the pipeline for each FORS1 and FORS2 grism (see Section 7 for details).

SCREEN_FLAT_MXU: *required* raw spectral screen flat exposure. If more than one is provided, the input frames are averaged into one.

9.8.2 Output files

Not all output frames listed here are always produced. Some of them are created only on request (see Section 9.8.3, page 86), and some other are never created in case of LSS or LSS-like data.[13] Here is the list of all the possible output frames, in alphabetical order, together with a list of related configuration parameters.[14] Note that in case of calibrations associated to a MOS observation with all slits aligned, the product categories will contain also the acronym LONG before the instrument mode tag: for instance, DELTA_IMAGE_MOS will become DELTA_IMAGE_LONG_MOS.

CURV_COEFF_MXU: table containing the coefficients of the spatial curvature fitting polynomials. This table is not produced in case of LSS or LSS-like data. The table columns are the following:

slit_id:	Slit identification number (see the SLIT_LOCATION_MXU entry for a definition of the *slit_id*). Each identification appears twice, in consecutive rows: the top row refers to the top flat field spectrum edge, the bottom row to its bottom edge.
c0, **c1**, **c2**, ...:	Curvature coefficients, depending on the degree of the fitting polynomial.

Configuration parameters directly affecting this product are *--cdegree* and *--cmode*.

Configuration parameters having significant impact are *--startwavelength* and *--endwavelength*.

[13] LSS-like data are obtained in the MOS instrument mode when all the slits are aligned; this kind of data are processed as a single long slit spectrum.

[14] See Section 9.8.3, page 86, for a complete description of the recipe configuration parameters.

CURV_TRACES_MXU: table containing the y CCD positions of the detected spectral edges at different x CCD positions. This table is not produced in case of LSS or LSS-like data. The table columns are the following:

x:	x CCD positions.
t<slit_id>:	y CCD positions of the flat spectrum top edge from slit *slit_id* (for the definition of *slit_id* see the SLIT_LOCATION_MXU entry).
b<slit_id>:	y CCD positions of the flat spectrum bottom edge from slit *slit_id*.
t<slit_id>**_mod**:	Modeling of the flat spectrum top edge from slit *slit_id*.
b<slit_id>**_mod**:	Modeling of the flat spectrum bottom edge from slit *slit_id*.
t<slit_id>**_res**:	Residuals of curvature fit of the flat spectrum top edge from slit *slit_id*.
b<slit_id>**_res**:	Residuals of curvature fit of the flat spectrum bottom edge from slit *slit_id*.

Typical tracing residuals are not greater than 0.3 pixels (see Figure 9.8.6, page 89).

The traces of some edges may be missing because tracing is not always possible between spectra that are very close to each other. This does not prevent the final extraction of all the spectra, if a global spatial curvature model is applied by setting the configuration parameter *--cmode* > 0: but residuals cannot be evaluated in this case.

Note that in case of confusion between nearby spectra, where the exact position of the transition line between one spectrum and the other can be ambiguous, the position of the edge ideally traced by the global curvature model might not exactly correspond to the true (and not observable) spectral edge. It should be understood, however, that the aim of the computed model is primarily to eliminate the spatial curvature, and that this can be obtained without knowing the absolute positions of the traces. In summary, observing extracted spectra[15] that include signal from other spectra and/or extending beyond their true spatial extension, does not imply that the spatial curvature was not properly removed. As a matter of fact nearby spectra *do* sometimes contaminate each other physically, by actually mixing their signals (case of crossing edges). In case of doubt, the extracted spectra should be carefully examined and compared with the corresponding original spectra found in the CCD exposure, in order to set the configuration parameter *--cmode* as appropriately as possible.

The only real solution to this problem would be to design masks where spectra are always well separated from each other (a buffer zone of 3 or 4 pixels would be sufficient).[16]

Configuration parameters directly affecting this product are *--cdegree* and *--cmode*.

Configuration parameters having significant impact are *--startwavelength* and *--endwavelength*.

DELTA_IMAGE_MXU: deviation from the linear term of the fitting wavelength calibration polynomials. This image is used together with the DISP_RESIDUALS_TABLE_MXU to allow some quality control of the obtained solutions (see Figure 9.8.1).

Configuration parameters directly affecting this product are *--startwavelength* and *--endwavelength*.

Configuration parameters having significant impact are *--dispersion*, *--peakdetection*, *--wradius*, and *--wdegree*.

[15] See entry REDUCED_LAMP_MXU in this Section, or entries MAPPED_SCI_MXU and MAPPED_ALL_SCI_MXU on page 96.

[16] It may be pointed out that this problem would "easily" be solved by applying an accurate physical model of the instrument. This however would be possible only under the assumption of a perfectly stable instrument, a dream that – together with the availability of the accurate physical model – remains too often unfulfilled.

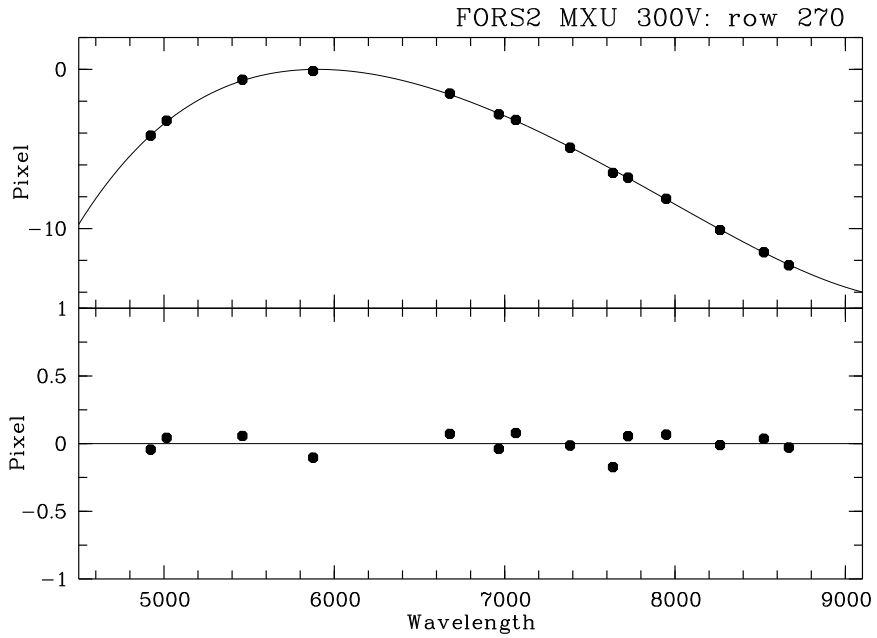

Figure 9.8.1: *Top panel: deviation of the identified peaks from the linear term of the 270th fitting polynomial (column d270 of the DISP_RESIDUALS_TABLE_MXU). The solid line is the polynomial model with the linear term subtracted, drawn from row 270 of the DELTA_IMAGE_MXU product. Bottom panel: fit residuals of the identified peaks (identical to the residuals recorded at row 270 of the DISP_RESIDUALS_MXU image).*

DISP_COEFF_MXU: table containing the wavelength calibration polynomial coefficients. This table contains as many rows as in the REDUCED_LAMP_MXU image, ordered in the same way. The table columns are the following:

c0, **c1**, **c2**, ...: Model coefficients, depending on the degree of the fitting polynomial.
nlines: Number of identified reference lines used in the fit.
error: Model mean accuracy computed from the observed fit residuals, keeping into account the number of model free parameters and the number of available reference lines:

$$\sigma = \sigma_{res}\sqrt{\frac{(n+1)}{N}}$$

where σ_{res} is the standard deviation of the residuals, n the polynomial degree, and N the total number of reference lines used in the fit. This evaluation of the model accuracy makes sense only in absence of systematic trends in the residuals shown in the DISP_RESIDUALS_MXU image. Typical values of the model accuracy range between 0.05 and 0.1 pixels.

Configuration parameters directly affecting this product are *--wdegree* and *--wmode*.

Configuration parameters having significant impact are *--dispersion*, *--peakdetection*, *--wradius*, *--wreject*, *--startwavelength* and *--endwavelength*.

DISP_RESIDUALS_MXU: residuals of each wavelength calibration fit (in pixels). This image is only created if the *--check* configuration parameter is set. The residuals of the derived wavelength calibration with respect to the measured pixel positions of the reference arc lamp lines are collected in this image, with x pixels corresponding to the original CCD pixels, and y pixels corresponding to the REDUCED_LAMP_MXU pixels (i.e., to the rectified spatial coordinate, see figure 9.8.2). Typical observed residuals should be around 0.2 pixels.[17] Note that all residuals are shown, including those from lines that were excluded from the polynomial fit, i.e., residuals larger than the threshold specified with the configuration parameter *--wreject* (see Section 9.8.3, page 86).

Configuration parameters directly affecting this product are *--startwavelength* and *--endwavelength*.

Configuration parameters having significant impact are *--dispersion*, *--peakdetection*, *--wradius*, and *--wdegree*.

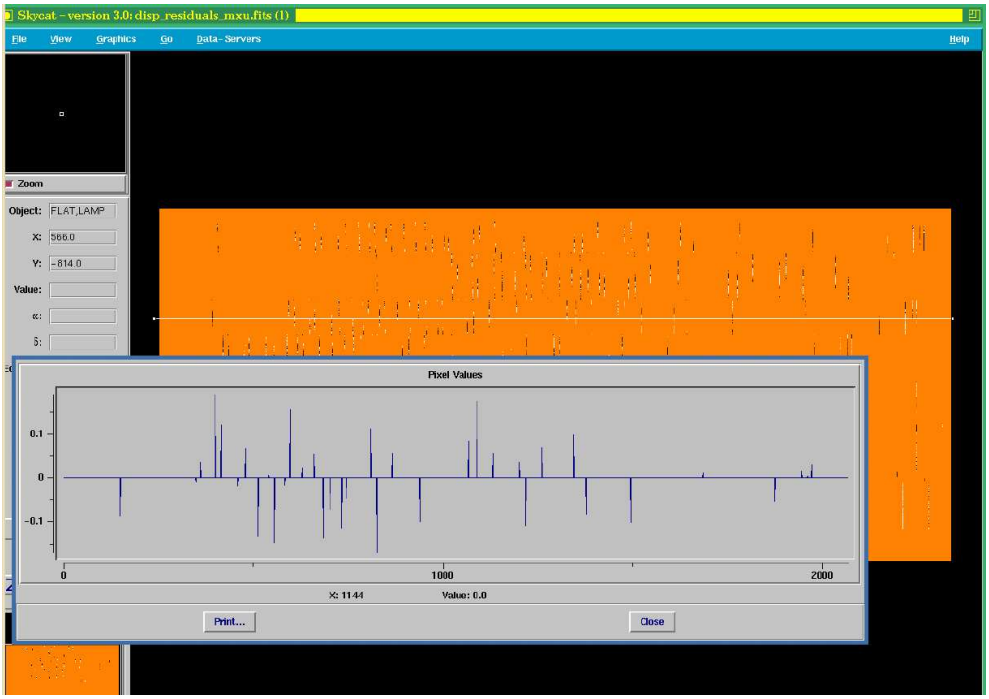

Figure 9.8.2: *RESIDUAL_MAP_MXU from a FORS2 MXU 600RI arc lamp calibration. In the foreground is a plot of the residuals from one image row.*

DISP_RESIDUALS_TABLE_MXU: table containing different kinds of residuals of a sample of wavelength calibration fits. Note that all residuals are shown, including those from lines that were excluded from the polynomial fit, i.e., residuals larger than the threshold specified with the configuration parameter *--wreject* (see Section 9.8.3, page 86). Just one every 10 of the polynomial fits listed in the DISP_COEFF_MXU ta-

[17] This is the accuracy of a single peak position measurement.

ble are examined. For an overview of all the polynomial fits residuals see the DISP_RESIDUALS_MXU image.

The residuals table columns are the following:

wavelength: Wavelengths of the reference lines (see entry MASTER_LINECAT).
r<*row*>: Fit residuals of the identified peaks (in CCD pixel). *row* is the number of the examined row of the DISP_COEFF_MXU table.
d<*row*>: Deviation of the identified peaks from the linear term of the fitting polynomial (in CCD pixel). This can be compared with the corresponding row of the DELTA_IMAGE_MXU product (see Figure 9.8.1).
p<*row*>: x pixel position of reference lines on CCD.

Configuration parameters directly affecting this product are *--startwavelength* and *--endwavelength*.

Configuration parameters having significant impact are *--dispersion*, *--peakdetection*, *--wradius*, and *--wdegree*.

GLOBAL_DISTORTION_TABLE: table containing the modeling of the coefficients of the local distortion models listed in the DISP_COEFF_MXU and the CURV_COEFF_MXU tables. It is produced only if the configuration parameter *--slit_ident* is set, and at least 12 spectra are found on the CCD. This table is currently used for quality control, and to support the on-line quick-look scientific data reduction. See Section 7 for more details.

MAPPED_NORM_FLAT_MXU: rectified and wavelength calibrated normalised screen flat field image (see entry MASTER_NORM_FLAT_MXU in this Section). This is the result of applying the extraction mask derived from the flat field and arc lamp exposures to the normalised flat field frame itself. This image is typically used for instrument health monitoring. Its size is identical to the size of the REDUCED_LAMP_MXU image (see corresponding entry in this Section).

Configuration parameters directly affecting this product are *--startwavelength* and *--endwavelength*.

Configuration parameters having significant impact are *--dispersion*, *--peakdetection*, *--wradius*, *--wdegree*, and *--wmode*.

MAPPED_SCREEN_FLAT_MXU: rectified and wavelength calibrated master screen flat field image (see entry MASTER_SCREEN_FLAT_MXU in this Section). This is the result of applying the extraction mask derived from the flat field and arc lamp exposures to the master flat field frame itself. This image is typically used for instrument health monitoring. Its size is identical to the size of the REDUCED_LAMP_MXU image (see corresponding entry in this Section).

Configuration parameters directly affecting this product are *--startwavelength* and *--endwavelength*.

Configuration parameters having significant impact are *--dispersion*, *--peakdetection*, *--wradius*, *--wdegree*, and *--wmode*.

MASTER_BIAS: master bias.

MASTER_NORM_FLAT_MXU: normalised flat field image, derived dividing the master screen flat by its smoothed version (see the smoothing configuration parameters description in Section 9.8.3, page 86).

Comparing this image with the MASTER_SCREEN_FLAT_MXU may give an immediate feeling of the goodness of the computed curvature model used for the extraction of the normalised spectra.

Configuration parameters directly affecting this product are *--sdegree*, *--ddegree*, *--sradius*, *--dradius*, *--startwavelength* and *--endwavelength*.

Configuration parameters having significant impact are *--cdegree* and *--cmode*.

MASTER_SCREEN_FLAT_MXU: combined flat field image. It is the sum of all the input screen flat fields.

REDUCED_LAMP_MXU: rectified and wavelength calibrated arc lamp image (see Figure 9.8.3). This is the

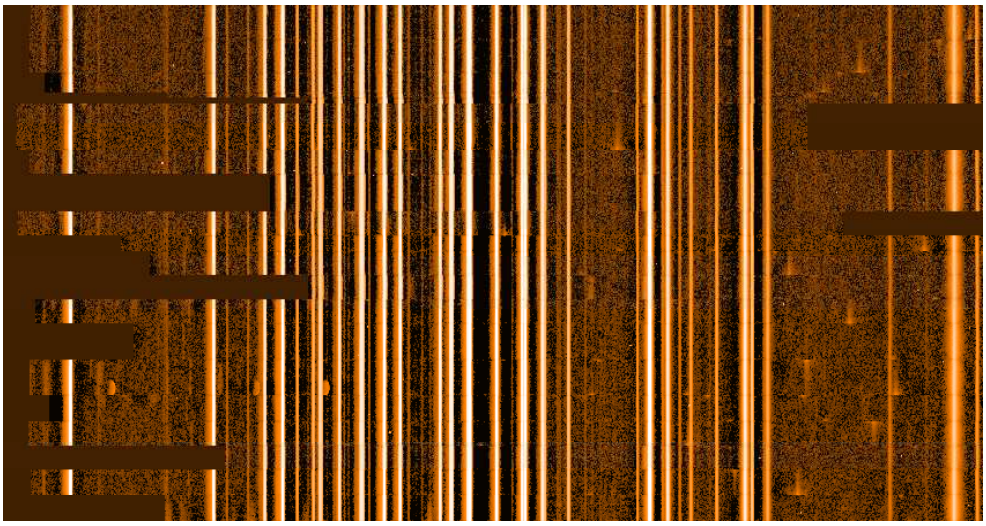

Figure 9.8.3: *REDUCED_LAMP_MXU from a FORS2 MXU 300I arc lamp exposure.*

result of applying the extraction mask derived from the flat field and arc lamp exposures to the input arc lamp exposure itself. This image is just useful to get an immediate feeling of the goodness of the computed extraction mask. Note that this image is also bias and background subtracted. Its x size depends on the spectral extraction range $(\lambda_{min}, \lambda_{max})$ and on the value used for the dispersion in wavelength units per pixel, D, defined by the configuration parameter *--dispersion* (see Section 9.9.3, page 99):

$$N_x = floor\Big(\frac{\lambda_{max} - \lambda_{min}}{D}\Big)$$

The y size of this image matches the y size of the exposed part of the CDD in the case of LSS os LSS-like data. In the case of multi-spectra observations the y size of this image is equal to the total number of spatially rectified pixels: each slit spectrum is extracted between the traces of its top and bottom edges (see products CURV_TRACES_MXU and CURV_COEFF_MXU), and spatially remapped into a constant number of pixels at each x CCD coordinate. The number of rectified pixels for the i-th slit spectrum is computed as

$$N_i = ceil(t_i - b_i) + 1$$

where t_i and b_i are the y CCD coordinates of the i-th slit spectrum edges at the position of the grism central wavelength.[18] N_i is increased by 1 to ensure a slight oversampling of the original signal. The total

[18]They correspond to the coefficients $c0$ of the CURV_COEFF_MXU table, or to $ytop$ and $ybottom$ in the SLIT_LOCATION_MXU table.

y size of the image is then given by

$$N_y = \sum_{i=0}^{n} N_i$$

where n is the number of extracted slit spectra. In case of MOS and MXU data the slit spectra are ordered from top to bottom as they appear on the CCD, and their positions are listed in the SLIT_LOCATION_MXU table. The wavelength of each image pixel can be computed using the CRPIX1, CRVAL1 and CDELT1 FITS keywords:

$$\lambda = CDELT1 \cdot (x - CRPIX1) + CRVAL1$$

where x is the pixel number counted from left starting from 1.

Note that resampling the original spectrum at a constant wavelength step introduces distortions of the signal, that depend on the original signal pixelisation on the CCD. This is a side-effect of interpolation, that tends to systematically overestimate and underestimate the interpolated value according to the position of the interpolation point with respect to the original CCD pixels.[19] This is especially evident in the case of LSS (REDUCED_LAMP_LSS) or LSS-like data (REDUCED_LAMP_MOS with all slits at the same offset): even if the reference lines of the resampled spectra will appear perfectly straight on the rectified image, the signal level along an image column corresponding to an arc lamp reference line will appear to follow a wavy pattern. This reminds that the resampling of scientific data is not always acceptable (depending on the scientific aim of a specific observation program). A detailed analysis of the scientific signal should be based on the unrebinned data matched with the corresponding wavelength map – see entry WAVELENGTH_MAP_MXU).

Configuration parameters directly affecting this product are *--startwavelength* and *--endwavelength*.

Configuration parameters having significant impact are *--dispersion*, *--peakdetection*, *--wradius*, *--wdegree*, and *--wmode*.

SLIT_LOCATION_MXU: slit positions, both on the CCD and on the rectified image of the arc lamp exposure (REDUCED_LAMP_MXU). The slits are listed from top to bottom, according to their y position on the CCD, and they are identified by a *slit_id* number. The *slit_id* is derived from the FITS header of the input data: in the case of MOS data the slits parameters are written to FITS keywords named

ESO INS MOS<slit_id> <parameter>

For instance, the width in millimeters of the third MOS slit in header is written to the FITS keyword:

ESO INS MOS3 POS

The slit identification number is the <*slit_id*> used in the naming convention for these keywords. The *slit_id* in the above example is 3. Note that in the FORS1 MOS case the *slit_id* is identical to the slit sequence number in the top–bottom ordering of the spectra on the CCD.

In the case of MXU data the slits parameters are written in FITS keywords named

ESO INS MOS1<slit_id> <parameter>

[19] No matter what interpolation method or kernel is chosen, this will always happen, unless the signal to resample is very well known in advance (which makes the interpolation pointless anyway): this would allow a perfect resampling of arc lamp spectra, for instance, but would not be applicable to scientific spectra.

For instance, the position angle of the 42nd MXU slit in header is written to the FITS keyword:

ESO INS MOS142 POSANG

The *slit_id* in this case is 42. Note that in the MXU case the *slit_id* is unrelated to the top–bottom ordering of the spectra on the CCD (rather referring to the way the mask was manufactured).

The slits location table columns are the following:

slit_id: Slit identification number.
xtop: x CCD position of central wavelength from top end of slit.
ytop: y CCD position of central wavelength from top end of slit.
xbottom: x CCD position of central wavelength from bottom end of slit.
ybottom: y CCD position of central wavelength from bottom end of slit.
position: First row of REDUCED_LAMP_MXU image containing the rectified slit spectrum bottom row. Image rows are counted from bottom, starting from 0.
length: Number of rows in REDUCED_LAMP_MXU image including the slit spectrum.

If the slit identification task is not run (see configuration parameter *--slit_ident*, Section 9.9.3, page 99), or if the slit identification task fails (e.g., in the case of just two slits) the *slit_id* is set to the slit sequence number in the top–bottom ordering of the spectra on the CCD: but in order to avoid confusion with tags assigned to identified slits, a negative integer is used in this case instead of a positive one.

SLIT_MAP_MXU: map of central wavelength on the CCD. This image is only created if the *--check* configuration parameter is set, and only in case the data are not LSS or LSS-like. It has the same size of the WAVELENGTH_MAP_MXU image, from which it is derived. This product can be seen as an image of the mask cast on the CCD (see step 5 in Section 10, page 114): the slits images on the CCD are compared with their positions on the mask, to derive the optical distortion model (see steps 6 and 7, always in Section 10).

Configuration parameters that may have some impact on this product are *--wdegree*, *--wmode*, *--dispersion*, *--peakdetection*, *--wradius*, and *--wreject*.

SPATIAL_MAP_MXU: map of spatial positions on the CCD. This image is not produced for LSS or LSS-like data. It has the same size of the CCD, where each pixel has the value of its distance (in CCD pixels) from the top edge of the spectrum it belongs to (see Figure 9.8.4). In case of confusion between nearby spectra, the spatial coordinate would just reflect the spatial curvature, and not the absolute spatial coordinate along the slit: see the note to the CURV_TRACES_MXU entry in this Section for more details.

Configuration parameters directly affecting this product are *--cdegree* and *--cmode*.

Configuration parameters having significant impact are *--startwavelength* and *--endwavelength*.

SPECTRA_DETECTION_MXU: result of the preliminary wavelength calibration applied to the arc lamp exposure. This image is only created if the *--check* configuration parameter is set, and only in case the data are not LSS or LSS-like.[20] The preliminary wavelength calibration is performed with the purpose of detecting and locating MXU and MOS spectra on the CCD (see step 2 in Section 10, page 113). In case of problems found in the recipe products, this image may be examined. All spectra should look aligned in

[20] In case of LSS or LSS-like data the preliminary wavelength calibration is actually identical to the final one.

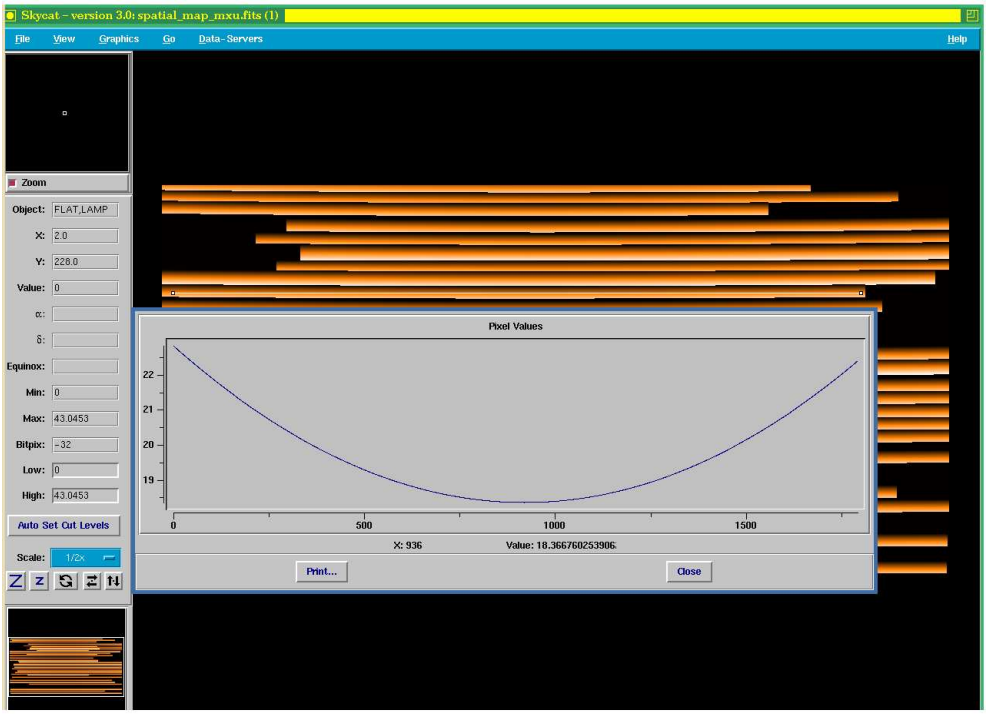

Figure 9.8.4: *SPATIAL_MAP_MXU from a FORS2 600RI flat field tracing, modeled with a 2nd degree polynomial. In the foreground is a plot of the distances from the top spectral edge of all pixels from one CCD row.*

wavelength, in particular around the central wavelength, that is the position used for constructing the slit map (SLIT_MAP_MXU). Gaps in the solution within a spectrum may appear, but if not overwhelming they have generally no consequences for the data reduction, because they are filled up consistently while creating the slit map. The x size of this image equals the x size of the REDUCED_LAMP_MXU image, while its y size matches the y size of the CCD (no spatial rectification performed).

Configuration parameters directly affecting this product are *--dispersion*, *--peakdetection*, and *--wdegree*. Configuration parameters having significant impact are *--startwavelength* and *--endwavelength*.

SPECTRAL_RESOLUTION_MXU: Mean spectral resolution for each reference arc lamp line. The table columns are the following:

wavelength:	Wavelength of reference line.
fwhm:	Mean FWHM of reference line.
fwhm_rms:	Standard deviation of all measured FWHM from all the CCD rows including the line.
resolution:	Mean spectral resolution, measured as the line *wavelength*, divided by its FWHM.
resolution_rms:	Standard deviation of all the measured spectral resolutions from all the CCD rows containing the line.

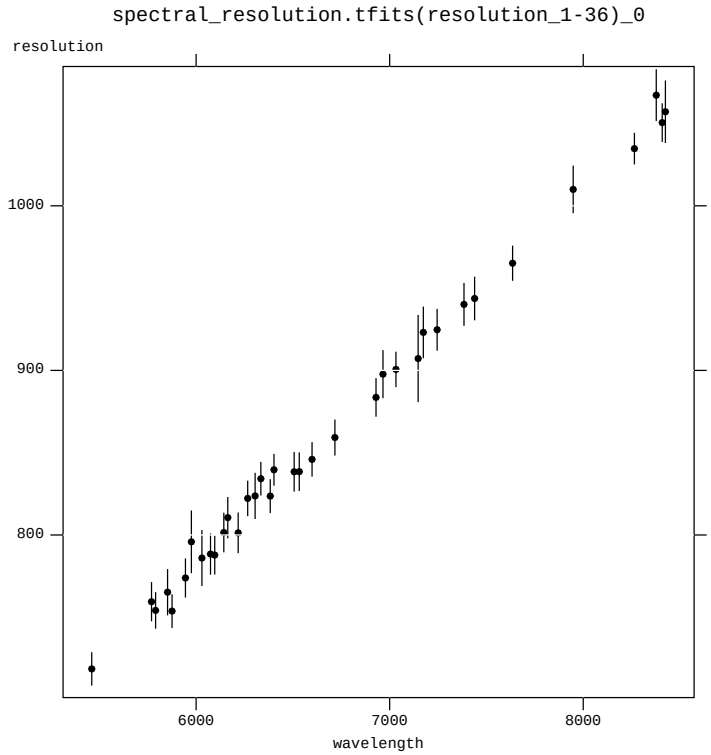

Figure 9.8.5: *Resolution vs. wavelength in a SPECTRAL_RESOLUTION_MXU table derived from a FORS2 MXU 600RI arc lamp exposure.*

WAVELENGTH_MAP_MXU: map of wavelengths on the CCD. This image has the same size of the CCD, where each pixel has the value of the wavelength at its center, if available.

Configuration parameters directly affecting this product are *--startwavelength* and *--endwavelength*.

Configuration parameters having significant impact are *--dispersion*, *--peakdetection*, *--wradius*, and *--wdegree*.

9.8.3 Configuration parameters

The configuration parameters setting determines the way the *fors_calib* recipe will process the input data, and to some extent the product files that will be created. The parameters are conveniently divided into four main sections: wavelength calibration, spatial curvature calibration, flat field normalisation, and quality control.

Wavelength calibration

--dispersion: Expected spectral dispersion. *Default*: 0.0 Å/pixel

This parameter is mandatory (using the default 0.0 would generate an error message). This is a rough value of the expected spectral dispersion, used by the pattern-recognition algorithm described in Section 10.4.2,

page 116. The dispersion values listed in the FORS1+2 User Manual [13] are good, but in exceptional cases they might be tuned for recovering possible failures of the data reduction procedure, or to improve the quality of unsatisfactory results. In general, however, the spectral detection algorithm is very robust to modifications of this parameter: as a typical example, with FORS2 300I grism data, for which the tabulated mean dispersion value is 1.62 Å/pixel, optimal results (at constant quality) are obtained within the interval 1.40–1.75 Å/pixel.[21]

Optimal values for this parameter, depending on the applied grism, are included in the GRISM_TABLE (see previous Section, page 47). Note that the *--dispersion* value must refer to the real CCD pixel size: the given value of the dispersion is internally multiplied by the rebin factor, to match the real pixel size of the input data.[22] In this way the value of the parameter *--dispersion* is made independent from the CCD readout mode.

The value of the *--dispersion* parameter, multiplied by the rebin factor, is defining also the constant wavelength step at which the rectified arc lamp frame and other similar products are resampled (see the REDUCED_LAMP_MXU entry, page 82).

--peakdetection: Initial peak detection threshold. *Default*: 0.0 ADU

This parameter is mandatory (using the default 0.0 would generate an error message). This is a threshold value used in the preliminary peak detection task (see Section 10.4.1, page 114): the reference lines candidates are selected from peaks having a maximum value *above the background* higher than this threshold. Weaker entries of the input line catalogue are recovered later on, after the preliminary wavelength calibration is obtained, if the parameter *--wradius* is set to a value greater than zero. It is however crucial that most of the reference lines are already detected at the earliest stage, if the pattern-recognition is meant to give the best possible results. A threshold value of 250 ADU is suitable in most cases, but sometimes the recovery of fainter reference lines may require to lower the threshold almost down to noise level.[23] Optimal values for this parameter, depending on the applied grism, are included in the GRISM_TABLE (see previous Section, page 47).

--startwavelength: Start wavelength in spectral extraction. *Default*: 0.0 Ångstrom

See the *--endwavelength* parameter.

--endwavelength: End wavelength in spectral extraction. *Default*: 0.0 Ångstrom

This parameter, together with the *--startwavelength* parameter, defines the wavelength interval where calibration is attempted: this interval may not be entirely contained in the CCD for all spectra. Default values of the extraction interval, depending on the applied grism, are included in the GRISM_TABLE (see previous Section, page 47). If both *--startwavelength* and *--endwavelength* are left to 0.0, the extraction interval is computed automatically as the interval between the first and the last identified arc lamp reference lines, extrapolated by 10% at its blue and red ends (see Section 10.4.3, page 119).

[21] This is not true for all kinds of data: for instance, in the case of the FORS2 600B grism data only dispersion values around 0.65–0.66 Å/pixel can provide good results. This tolerance mostly depends on the dispersion law of the applied grism, and on the available arc lamp lines.

[22] The rebin factor along the dispersion direction is written to the FITS header keyword ESO DET WIN1 BINX.

[23] Lowering this threshold below a 3-σ noise level would completely destroy the observed pattern. In such extreme cases a preliminary smoothing of the input arc lamp exposure for reducing the random noise may help.

--*wdegree:* Degree of wavelength calibration polynomial. *Default*: 0

This parameter is mandatory (using the default 0 would generate an error message). The degree used for the wavelength calibration polynomial should be the lowest that would provide non-systematic residuals to the solution (see the DISP_RESIDUALS_MXU entry, page 80). Optimal values of the extraction interval, depending on the applied grism, are included in the GRISM_TABLE (see previous Section, page 47).

Note that the --*wdegree* parameter should be more correctly intended as the *maximum* applicable polynomial order: the polynomial is really adapted to the number of identified arc lamp lines used in the fit. This is necessary, because spectra from slits with very high offsets on the telescope focal plane may not be entirely contained in the CCD, and several arc lamp reference lines might be unavailable for calibration. Such spectra would not be properly calibrated if a polynomial with too many free parameters were used. As a rule, a polynomial with the specified --*wdegree* is only used if the number of identified lines is at least twice the number of free parameters: if this were not the case, the applied polynomial order would be

$$n = floor\left(\frac{N}{2}\right) - 1$$

where N is the number of identified reference lines. Accordingly, no solution is computed if less than 4 reference lines are identified.

--*wradius:* Search radius, if iterating pattern-matching with first-guess method. *Default*: 4 pixel

If this parameter is greater than zero, the peak identification is iterated using the pattern-matching solution as a first-guess model: the wavelengths listed in the input line catalogue are transformed to CCD pixel positions using the model, and a peak is searched within the specified search radius.[24] Alternatively, setting --*wradius* = 0 means to accept the pattern-matching solution without further processing. Iterating the solution makes the wavelength calibration more robust, and increasing the search radius may help sometimes to recover from a bad result. It may happen however that the pattern-matching solution is more accurate than the one based on the iteration: this is because in the pattern-matching task peaks are identified by their being part of a pattern, while with a first-guess model each peak is identified by its vicinity to its expected position: the latter approach may lead to occasional misidentifications, and may be more negatively affected by contamination and lines blending (see also Section 10.4.1, page 114).

--*wreject:* Rejection threshold in dispersion relation fit (pixel). *Default*: 0.7 pixel

The wavelength calibration polynomial fit is iterated excluding any reference line position displaying a residual greater than the specified threshold.

--*wmode:* Interpolation mode of wavelength solution (0 = no interpolation, 1 = fill gaps, 2 = global model). *Default*: 2

This parameter only affects the processing of LSS and LSS-like data. Given the wide availability of similar information on a long slit spectrum, it is conceivable an improvement of the quality of the wavelength calibration by modeling the global trend of the local solutions obtained from each CCD row. If --*wmode* = 1 the global model is applied just to fill possible gaps in the solution, maintaining the result of the local calibrations where they are available. If --*wmode* = 2 the global model solution is used for replacing also the available local solutions. No interpolation is applied to the data if --*wmode* = 0.

[24] If a search radius greater than zero is specified, but the reference lines widths are even greater, the search radius is automatically set to the actual lines widths.

--wcolumn: Name of line catalogue table column with wavelengths. *Default*: WLEN

This is the name of the MASTER_LINECAT table column where the arc lamp reference wavelengths are listed. The default is the name of the relevant column in the standard line catalogue (see also the MASTER_LINECAT entry on page 77). This would allow the usage of any FITS table, supplied by the user, containing a list of wavelengths to be processed by the pattern-recognition task. The only requirement is that the listed wavelengths are given in Ångstrom, and that they are sorted from blue to red.

Spatial curvature calibration

--cdegree: Degree of spatial curvature polynomial. *Default*: 0

This parameter is mandatory (using the default 0 would generate an error message). In general a 2nd degree polynomial gives good results. Optimal values depending on the applied grism are included in the GRISM_TABLE (see previous Section, page 47).

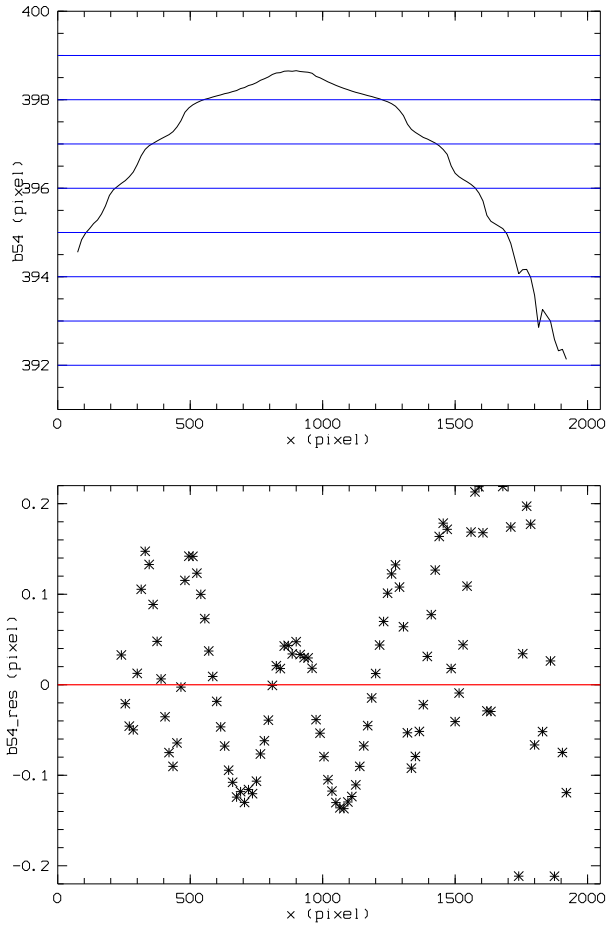

Figure 9.8.6: *Systematic residuals of curvature model (from a FORS2 MXU 600B flat field exposure).*

Systematic residuals, oscillating from positive to negative offsets of about 0.2–0.3 pixels, are frequently observed, and are confirmed also by other data reduction systems (see Figure 9.8.6). The systematic

residuals are due to the changing pixelisation of the spectral edges on the CCD, and therefore they should not be considered physical. A low degree polynomial fit appropriately circumvents this effect by cutting through such oscillations. Trying to fit such residuals with higher degree polynomials would lead to unstable and unrealistic solutions.

--*cmode:* Interpolation mode of curvature solution (0 = no interpolation, 1 = fill gaps, 2 = global model). *Default*: 1

This parameter does not affect the processing of LSS or LSS-like data. Using a global description of the spatial curvature helps to extract also those spectra whose edges cannot be traced because of confusion with nearby spectra. If --*cmode* = 0 the only recovery strategy consists in supplying a missing trace by replicating the trace of the opposite edge (opportunely shifted). This is however not very accurate, and it is not even applicable if a tracing is missing for both edges of a slit spectrum.[25] By setting --*cmode* = 1 a global trend of the curvature coefficients would be determined, allowing to derive a curvature model also for the spectral edges that are lacking a direct tracing. Setting --*cmode* = 2 would recompute the curvature model also for the spectra where a local solution is available: this is generally not advisable, because a local solution is generally more accurate than the one derivable from the global solution.

--*slit_ident:* Attempt slit identification. *Default*: TRUE

This parameter does not affect the processing of LSS or LSS-like data. Setting this parameter activates the 2D pattern-recognition task linking the slits positions on the mask with those on the CCD (see Section 10.4.6, page 120). In principle, the only outcome would be the identification of the detected spectra, i.e., their association to the slits on the mask, that is not required for a complete processing of the data: spectra would be extracted anyway, even if lacking a proper identification.[26] However, as shown in Section 10.4.6, the 2D pattern-recognition is also used to define an optical distortion model that helps to improve the accuracy of the preliminary spectra detection, and in some case even to allow the recovery of spectra that were lost to the spectral identification task. This is why the slit identification should always be requested: the only reason why the parameter --*slit_ident* was defined is to offer to possibility to switch the 2D pattern-matching task off in case this affected negatively the data reduction process.[27] Note also that excluding the slit identification would also allow to reduce data from instruments different from FORS1 and FORS2.[28]

Flat field normalisation

--*sdegree:* Degree of flat field fitting polynomial along spatial direction. *Default*: 4

This parameter only affects the processing of LSS and LSS-like data. If the configuration parameter --*sdegree* is set to a non-negative value, the master flat field normalisation is performed by modeling its large scale trend with a polynomial fitted along the spatial direction (and not along the dispersion direction, as

[25] In this case the spectrum would not be extracted.

[26] In fact, the 2D pattern-recognition task would fail in case less than three spectra were detected on the CCD, and also in case the spectra were regularly spaced, as it happens with some calibration masks: but in neither situation spectra identification represents a practical issue.

[27] This was never seen to happen, but one never knows...

[28] The only reason why the self-calibrating recipes described here are not readily usable for any MOS instrument is that the way the slit characteristics are listed in the data FITS headers is not standardised. It should also be pointed out that these recipes are not designed to handle data with spectral multiplexing, as those from low-resolution observations made with VIMOS.

for the case of shorter slits), for each CCD column. If --*sdegree* < 0 the illumination trend is obtained instead by median filtering with a running box of sizes --*dradius* and --*sradius*.

--*ddegree:* Degree of flat field fitting polynomial along dispersion direction. *Default*: -1

This parameter does not affect the processing of LSS or LSS-like data. If --*ddegree* is set to a non-negative value, the master flat field normalisation is performed by modeling its large scale trend with a polynomial fitted along the dispersion direction. The flat field spectra are spatially rectified applying the curvature model before the fit is performed, and the smoothed result is mapped back to the CCD frame before being used for normalising the master flat field. If --*ddegree* < 0 the illumination trend is obtained instead by median filtering the spatially rectified spectra with a running box of sizes --*dradius* and --*sradius*. Polynomial modeling should be preferred whenever appropriate, i.e., in all those cases where the flat field illumination does not vary with the wavelength in a too complex way, and the fit residuals would not display systematic displacements from the flat field illumination trend.

--*dradius:* Smooth box radius for flat field along dispersion direction. *Default*: 10 pixel

See the --*sradius* parameter.

--*sradius:* Smooth box radius for flat field along spatial direction. *Default*: 10 pixel

This parameter, together with --*dradius*, affects the processing of LSS and LSS-like data only in case --*sdegree* is negative, and the processing of multi-spectra data only in case --*ddegree* is negative. --*sradius* and --*dradius* are the sizes (in pixel) along the spatial and dispersion directions of the running box applied for smoothing the master flat field before its final normalisation.

Quality control

--*qc:* Compute QC1 parameters. *Default*: TRUE

Setting this parameter will trigger the Quality Control parameters computation. This is just relevant for the on–line pipeline run automatically on Paranal, and for the off-line reduction of Service Mode observations performed by DFO. See Section 9.8.4 for a list of the computed parameters.

--*check:* Create intermediate products. *Default*: FALSE

Setting this parameter will write to disk some intermediate products of the data reduction procedure. This may be useful for debug purposes, in case of unsatisfactory results or failures of the *fors_calib* recipe. Currently the created intermediate products are SPECTRA_DETECTION_MXU and SLIT_MAP_MXU (see page 84).

9.8.4 Quality control parameters

Currently the following QC parameters, used by PSO and DFO, are evaluated by the *fors_calib* recipe. Note that, unless indicated otherwise, the acronym LSS in the parameters names can also be read MXU and MOS.

QC WAVE ACCURACY: Mean wavelength calibration accuracy. *Units*: pixel

This is the mean value of the **error** column in table DISP_COEFF_LSS (see the corresponding entry in this Section 9.8.2, page 77).

QC WAVE ACCURACY ERROR: Error of mean wavelength calibration accuracy. *Units*: pixel

This is the standard deviation of the **error** column in table DISP_COEFF_LSS (see the corresponding entry in this Section 9.8.2, page 77).

QC LSS RESOLUTION: Mean spectral resolution of all identified arc lamp lines. *Units*: *none*

From a wavelength calibrated arc lamp exposure, the mean spectral resolution for each line in catalogue is evaluated as the ratio between its wavelength and its FWHM. The results are written to the SPECTRAL_RESOLUTION_LSS table (see page 85), and the mean resolution of all lines is written to the keyword ESO QC LSS RESOLUTION of its FITS header.

QC LSS RESOLUTION RMS: Scatter of all computed spectral resolutions. *Units*: *none*

Population RMS of all values contributing to the mean spectral resolution (QC LSS RESOLUTION). This value is written to the keyword ESO QC LSS RESOLUTION RMS of the SPECTRAL_RESOLUTION_LSS table (see above).

QC LSS RESOLUTION NLINES: Number of lines used for computing the mean resolution. *Units*: *none*

Number of reference arc lamp lines used in the computation of the mean resolution. This value is written to the keyword ESO QC LSS RESOLUTION NLINES of the SPECTRAL_RESOLUTION_LSS table (see above).

QC LSS RESOLUTION NWAVE: Number of wavelengths used for computing the mean resolution. *Units*: *none*

Number of distinct arc lamp lines wavelengths used in the computation of the mean resolution. This value is written to the keyword ESO QC LSS RESOLUTION NWAVE of the SPECTRAL_RESOLUTION_LSS table (see above).

QC LSS RESOLUTION MEANRMS: Mean RMS of spectral resolution at each wavelength. *Units*: *none*

Mean RMS of in the determination of the spectral resolution at each wavelength.

QC LSS CENTRAL WAVELENGTH: Wavelength at CCD center. *Units*: Ångstrom

After the wavelength calibration is obtained from an LSS or LSS-like arc lamp exposure, the wavelength corresponding to the central CCD pixel is calculated. This parameter is not computed for MOS/MXU observations, but just for LSS observations made with the 1 arcsec slit, and for MOS observations having all slitlets at offset 0.0 (in the MOS case the parameter is named QC MOS CENTRAL WAVELENGTH). The central wavelength is written to the keyword ESO QC LSS CENTRAL WAVELENGTH of the wavelength map FITS header (see entry WAVELENGTH_MAP_MXU, page 86).

9.9 fors_science

This recipe is used for reducing FORS1 and FORS2 LSS, MOS and MXU scientific spectra applying the extraction mask and the normalised flat field created by the recipe *fors_calib*. The slit spectra are bias subtracted, flat fielded if requested, and remapped eliminating the optical distortions. The input wavelength calibration can optionally be adjusted to a number of reference sky lines. Finally, objects are searched and extracted from all the slit spectra.

In the following sections the MXU acronym in the products names can also be read MOS, or LSS, unless indicated otherwise. In the same way the word SCIENCE may be alternatively read STANDARD, as the spectroscopic standard stars exposures are reduced as scientific exposures. In case of standard star observations the SCI acronym in the products names should also be read STD.[29]

9.9.1 Input files

In alphabetical order:

CURV_COEFF_MXU: *required* table with spatial curvature coefficients, however *not required* for LSS and LSS-like observations.[30] This table is produced by the *fors_calib* recipe (see page 77).

DISP_COEFF_MXU: *required* table with wavelength solution coefficients. This table is produced by the *fors_calib* recipe (see page 79).

GRISM_TABLE: *optional* grism table. See Section 9.8.1, page 47.

MASTER_BIAS: *required* master bias frame. Just one should be given.

MASTER_NORM_FLAT_MXU: *optional* normalised flat field. This frame is produced by the *fors_calib* recipe (see page 81), and it must be provided only if the flat field correction is requested (see configuration parameter *--flatfield*, Section 9.9.3, page 99).

MASTER_SKYLINECAT: *optional* sky lines catalogue. It must contain the reference wavelengths (in Ångstrom) of the sky lines used for adjusting the input wavelength solution to the observed scientific spectra. The only requirement for this table is to contain a column listing such wavelengths, whose name should be specified using the configuration parameter *--wcolumn* (see Section 9.9.3, page 99). If the alignment of the wavelength solution to the sky lines is requested, but a MASTER_SKYLINECAT is not specified in input, an internal sky line catalogue is used instead (see Table 9.9.1).

SCIENCE_MXU: *required* scientific exposure. Just one frame should be specified.

[29] In this way it may happen that up to 9 different category names may be assigned to a product having exactly the same content in exactly the same format. For instance, the table carrying the dispersion coefficients referring to the spatially rectified spectra, may be assigned the following category names:

1. DISP_COEFF_MXU for MXU data reduced by *fors_calib*
2. DISP_COEFF_MOS for MOS data reduced by *fors_calib*
3. DISP_COEFF_LSS for LSS data reduced by *fors_calib*
4. DISP_COEFF_SCI_MXU for MXU scientific data reduced by *fors_science*
5. DISP_COEFF_SCI_MOS for MOS scientific data reduced by *fors_science*
6. DISP_COEFF_SCI_LSS for LSS scientific data reduced by *fors_science*
7. DISP_COEFF_STD_MXU for MXU standard star data reduced by *fors_science*
8. DISP_COEFF_STD_MOS for MOS standard star data reduced by *fors_science* and,
9. DISP_COEFF_STD_LSS for LSS standard star data reduced by *fors_science*.

[30] Currently no spatial curvature correction is applied to LSS data, this will be fixed in the next releases.

SLIT_LOCATION_MXU: *required* table of slits positions. This table is produced by the *fors_calib* recipe (see page 83).

Wavelength	Low resolution	Wavelength	Low resolution
5577.338	√	7329.148	
5889.953		7340.885	
5895.923		7358.659	
5915.301		7571.746	√
5932.862		7750.640	
5953.420		7759.996	
6257.961		7794.112	
6287.434		7808.467	
6300.304	√	7821.503	
6306.869		7841.266	
6363.780		7913.708	
6498.729		7949.204	
6533.044		7964.650	√
6553.617		7993.332	√
6841.945		8014.059	
6863.955	√	8310.719	
6870.994		8344.602	
6889.288		8382.392	
6900.833		8399.170	
6912.623		8415.231	
6923.220		8430.174	
6939.521		8452.250	
6969.930		8493.389	
7003.858		8791.186	
7244.907		8827.096	
7276.405		8885.850	
7284.439		8903.114	
7316.282		8943.395	
		8988.366	

Table 9.9.1: *Default sky lines wavelengths used by the recipe* fors_science. *The marked lines are those used on data from low resolution grisms, i.e., all grisms with 300 gr/mm or less.*

9.9.2 Output files

Not all output frames listed here are always produced. Some of them are created only on request, and some other are never created in case of LSS or LSS-like data.[31] Here is the list of all the possible output frames, in alphabetical order, together with a list of related configuration parameters:[32]

DISP_COEFF_SCI_MXU: This adjustment of the input DISP_COEFF_MXU table is only created in case the alignment of the wavelength solution to the sky lines is requested (see the configuration parameter *--skyalign*, Section 9.9.3, page 99). For a description of this product see the DISP_COEFF_MXU entry on page 79. In the DISP_COEFF_SCI_MXU table the *error* column content is computed by (quadratically) summing the errors of the input wavelength solution with the errors of the sky alignment fit. Similarly, in the *nlines* column the number of sky lines used for the alignment replaces the number of reference arc lamp lines on which the input calibration was based.

Configuration parameters directly affecting this product are *--skyalign*, *--startwavelength* and *--endwavelength*.

GLOBAL_SKY_SPECTRUM_MXU: table with supersampled sky spectrum, created only if the global sky subtraction is requested (see configuration parameter *--skyglobal*, Section 9.9.3, page 99). Each wavelength bin is half the resampling step, multiplied by the CCD readout rebin factor (see the configuration parameter *--dispersion*, Section 9.9.3, page 99).

The spectra contained in the input scientific exposure (see the SCIENCE_MXU entry on page 93) are assumed to contain altogether at least 50% of their pixels on the sky. Moreover, all the *scientific* slits are assumed to have the same width.[33] The wavelength map derived from the input DISP_COEFF_MXU table (possibly adjusted by the sky lines alignment task) is used to map all the spectral signal in the CCD into a grid of wavelength bins. The sky spectrum is computed as the median level of all the pixel values of all the CCD spectra in each wavelength bin. The median of the contributing wavelengths (which are not uniformely distributed within the bin) is also assigned to each bin. Empty bins are computed by linear interpolation between the nearest valid bins, and in this case a bin is assigned its central wavelength.

The global sky table includes the following columns:

wavelength:	Bin wavelength.
sky:	Median signal level for each bin.
npoints:	Number of points contributing to each bin.

Configuration parameters directly affecting this product are *--skyglobal*, *--time_normalise*, *--startwavelength* and *--endwavelength*.

MAPPED_ALL_SCI_MXU: image with rectified and wavelength calibrated slit spectra. Its x size depends on the spectral extraction range $(\lambda_{min}, \lambda_{max})$ and on the specified resampling step in wavelength units

[31] LSS-like data are obtained in MOS or MXU instrument modes with all the slits aligned; this kind of data are processed as a single long slit spectrum.

[32] See Section 9.9.3, page 99, for a complete description of the recipe configuration parameters.

[33] If this were not the case, the global sky model quality would be poorer, and only the slits with a median slit width would be properly corrected. This may be fixed by applying a local sky subtraction following the global one, but this would eliminate the advantages of using a global sky model.

per pixel, D, defined by the configuration parameter *--dispersion* (see Section 9.9.3, page 99):

$$N_x = floor\Big(\frac{\lambda_{max} - \lambda_{min}}{D}\Big)$$

The y size is determined in the same way as for the REDUCED_LAMP_MXU frame (see page 82).

Note that resampling the original spectrum at a constant wavelength step introduces distortions of the signal, that depend on the original signal pixelisation on the CCD, and it introduces noise correlation. See the final note to the REDUCED_LAMP_MXU entry on page 82.

Configuration parameters directly affecting this product are *--dispersion*, *--flux*, *--flatfield*, *--time_normalise*, *--skyalign*, *--startwavelength* and *--endwavelength*.

MAPPED_SCI_MXU: image with rectified, wavelength calibrated and sky subtracted slit spectra. This image matches in size the MAPPED_ALL_SCI_MXU image, and is produced only if any kind of sky subtraction (global and/or local) is requested.

Configuration parameters directly affecting this product are *--dispersion*, *--flux*, *--cosmics*, *--flatfield*, *--time_normalise*, *--skyalign*, *--startwavelength* and *--endwavelength*.

Configuration parameters having significant impact are *--skymedian*, *--skylocal*, and *--skyglobal*.

MAPPED_SKY_SCI_MXU: image with rectified and wavelength calibrated slit sky spectra. This image contains the modeled sky that was subtracted from the scientific data, either before or after the scientific spectra rectification (or even both, if the configuration parameters *--skyglobal* and *--skymedian* were both set: the contribution of the global sky model is included in this image even if the global sky subtraction is really applied to the data before their rectification). The sky model component subtracted before the rectification of the scientific spectra can be viewed separately in the GLOBAL_SKY_SPECTRUM_MXU and the UNMAPPED_SKY_SCI_MXU products.

The MAPPED_SKY_SCI_MXU also includes the identified cosmic ray signal in case the cosmic rays removal is requested (see configuration parameter *--cosmics*, Section 9.9.3, page 99).

Configuration parameters directly affecting this product are *--skymedian*, *--skyglobal*, *--skylocal*, *--cosmics*, *--time_normalise*, *--startwavelength* and *--endwavelength*.

Configuration parameters having significant impact are *--skyalign*, *--flux*, and *--flatfield*.

OBJECT_TABLE_SCI_MXU: This table is an expansion of the input SLIT_LOCATION_MXU table (see page 83), where the positions and the extraction spatial intervals of the detected objects are also included.[34] This table is produced only if any kind of sky subtraction (global and/or local) is requested, otherwise no object detection or extraction is attempted. The slits location table columns are the following:

slit_id:	Slit identification number.
xtop:	x CCD position of central wavelength from top end of slit.
ytop:	y CCD position of central wavelength from top end of slit.
xbottom:	x CCD position of central wavelength from bottom end of slit.
ybottom:	y CCD position of central wavelength from bottom end of slit.

[34] A SLIT_LOCATION_LSS table is not defined for LSS or LSS-like data, but an OBJECT_TABLE_SCI_MXU containing one slit with its objects is produced anyway.

position:	First row of the rectified images (MAPPED_ALL_SCI_MXU and MAPPED_SCI_MXU) containing the rectified slit spectrum. Image rows are counted from bottom, starting from 0.
length:	Number of rows in rectified images including the slit spectrum.
object_1, object_2, ...:	Detected objects positions in the rectified images.
start_1, start_2, ...:	Start position of the extraction interval for each object.
end_1, end_2, ...:	End position of the extraction interval for each object.
row_1, row_2, ...:	Row number of the REDUCED_SCI_MXU image containing the extracted object spectrum. Image rows are counted from bottom, starting from 0.

Configuration parameters directly affecting this product are *--slit_margin*, *--ext_radius*, *--cont_radius*.

Configuration parameters that may have significant impact are *--startwavelength* and *--endwavelength*.

REDUCED_ERROR_SCI_MXU: image with errors (one sigma level) corresponding to the extracted objects spectra. This image matches the REDUCED_SCI_MXU image.

Configuration parameters directly affecting this product are *--dispersion*, *--ext_mode*, *--time_normalise*, *--startwavelength* and *--endwavelength*.

Configuration parameters having significant impact are *--slit_margin*, *--ext_radius*, *--cont_radius*, and *--flux*.

REDUCED_SCI_MXU: image with extracted objects spectra. This image has the same x size of the image with the extracted slit spectra, MAPPED_SCI_MXU, and as many rows as the detected and extracted object spectra. Extracted spectra are written to the image rows listed in the OBJECT_TABLE_SCI_MXU table.

Configuration parameters directly affecting this product are *--dispersion*, *--ext_mode*, *--time_normalise*, *--startwavelength* and *--endwavelength*.

Configuration parameters having significant impact are *--slit_margin*, *--ext_radius*, *--cont_radius*, *--skyalign*, *--flux*, *--flatfield*, *--skyglobal*, *--skylocal*, *--skymedian*, and *--cosmics*.

REDUCED_SKY_SCI_MXU: image with sky corresponding to the extracted objects spectra. The sky is extracted in the same way as the objects, e.g., if optimal weights were applied to the object extraction, the same weights are applied to the sky extraction. This image matches the REDUCED_SCI_MXU image.

Configuration parameters directly affecting this product are *--dispersion*, *--ext_mode*, *--time_normalise*, *--startwavelength* and *--endwavelength*.

Configuration parameters having significant impact are *--slit_margin*, *--ext_radius*, *--cont_radius*, *--skyalign*, *--flux*, *--flatfield*, *--skylocal*, *--skyglobal* and *--skymedian*.

SKY_SHIFTS_LONG_SCI_MXU: table containing the observed sky lines offsets that were used for adjusting the input wavelength solution. This table is only produced if the sky lines alignment is requested (see configuration parameter *--skyalign*, Section 9.9.3, page 99), and is specific to LSS and LSS-like data (for multi-spectra observations the SKY_SHIFTS_SLIT_SCI_MXU table is produced instead).

This table has as many rows as the MAPPED_ALL_SCI_MXU image, and it contains two columns for each of the examined sky lines:

off_5577: Observed skyline offset (in pixel) at each CCD row. The integer truncation of each sky line wavelength is used for composing the corresponding column name (in this example the column name refers to the 5577.338 O I emission).
fit_5577: Modeling of the *off_5577* offsets.

This table may be very useful for judging what would be the most appropriate modeling of the observed offsets, and to what extent the input wavelength calibration really needs to be adjusted.

Configuration parameters directly affecting this product are *--startwavelength* and *--endwavelength*.

SKY_SHIFTS_SLIT_SCI_MXU: table containing the observed sky lines offsets that were used for adjusting the input wavelength solution. This table is only produced if the sky lines alignment is requested (see configuration parameter *--skyalign*, Section 9.9.3, page 99), and is specific to multi-spectra observations (for LSS and LSS-like observations the SKY_SHIFTS_LONG_SCI_MXU table is produced instead). This table has one row for each of the sky lines used for the alignment, and one column for each slit where sky lines could be detected.[35]

The included columns are the following:

wave: Sky line wavelength.
offset<slit_id>: Observed offsets for the slit spectrum with identification *slit_id*.

This table may be very useful for judging what would be the most appropriate modeling of the observed offsets, and to what extent the input wavelength calibration really needs to be adjusted.

Configuration parameters directly affecting this product are *--startwavelength* and *--endwavelength*.

UNMAPPED_SCI_MXU: image with the sky subtracted scientific spectra on the CCD frame, created only if the global or the local sky subtraction is requested (see the configuration parameters *--skyglobal* and *--skylocal*, Section 9.9.3, page 99).[36] This image is derived subtracting the UNMAPPED_SKY_SCI_MXU from the bias subtracted and flat fielded scientific frame.

Configuration parameters directly affecting this product are *--skyglobal*, *--skylocal*, *--skyalign*, *--time_normalise*, *--startwavelength* and *--endwavelength*.

UNMAPPED_SKY_SCI_MXU: this image has the same size of the CCD, and is created if either the global or the local sky subtraction is requested.

If *--skyglobal* is set (see the configuration parameter *--skyglobal*, Section 9.9.3, page 99), this image contains the global sky model mapped on the CCD frame, derived from the supersampled sky spectrum contained in the GLOBAL_SKY_SPECTRUM_MXU table. Each one of its pixels is assigned a value obtained by linear interpolation of the two wavelengths of the supersampled spectrum that are closest to its wavelength.

If *--skylocal* is set (see the configuration parameter *--skylocal*, Section 9.9.3, page 99), this image contains the sky model obtained by interpolating the sky signal trend along the spatial direction, directly on the CCD frame.

[35] In general the sky lines detection fails for reference slits, that are typically filled up by very bright objects.

[36] In the case of LSS or LSS-like data this image is only created with the *--skyglobal* option, because the *--skylocal* option is not available.

The global sky subtraction consists of subtracting this image from the original bias subtracted and flat field corrected scientific exposure.

Configuration parameters directly affecting this product are *--skyglobal*, *--skyalign*, *--time_normalise*, *--startwavelength* and *--endwavelength*.

WAVELENGTH_MAP_SCI_MXU: This upgraded version of the wavelength map is only produced in case the adjustment of the wavelength solution to the sky lines is requested (see the configuration parameter *--skyalign*, Section 9.9.3, page 99). For a description of this product see the WAVELENGTH_MAP_MXU entry on page 86. Note that the coordinate system of the WAVELENGTH_MAP frames will generally differ, because they are derived from different input data: the coordinate system of WAVELENGTH_MAP_MXU is inherited from the arc lamp frame header, while the coordinate system of WAVELENGTH_MAP_SCI_MXU is inherited from the scientific frame header.

Configuration parameters directly affecting this product are *--skyalign*, *--startwavelength* and *--endwavelength*.

9.9.3 Configuration parameters

The configuration parameters setting determines the way the *fors_science* recipe will process the input data, and to some extent the product files that will be created. The parameters are conveniently divided into six main sections: wavelength calibration, spatial curvature calibration, flat field correction, sky subtraction, objects detection and extraction, and flux calibration.

Wavelength calibration

--skyalign: Polynomial order for sky lines alignment. *Default*: 0

The input wavelength calibration can be adjusted to the observed positions of a set of sky lines, whose wavelengths are listed in an input catalogue. The observed sky lines offsets from their expected positions (see entries SKY_SHIFTS_LONG_SCI_MXU and SKY_SHIFTS_SLIT_SCI_MXU, page 97) are fitted by polynomials that are then added to the input wavelength calibration polynomials (see DISP_COEFF_MXU entry on page 79). A *--skyalign* = 0 would just determine a median offset from all the observed sky lines, while *--skyalign* = 1 would try to fit a slope (rarely useful, but sometimes sky lines offsets display a significant dependency on the wavelength, due to a variation of the mean spectral dispersion with respect to the day calibrations). Polynomials with order greater than 2 generate a friendly error message. Setting *--skyalign* < 0 disables any sky line alignment, accepting the input wavelength calibration as-is.

Note that the *--skyalign* parameter should be more correctly intended as the *maximum* applicable polynomial order: the polynomial is really adapted to the number of identified sky lines used in the fit. As a rule, a polynomial with the specified order is only used if the number of identified sky lines is at least greater than the number of free parameters: if this were not the case, the applied polynomial order would be

$$n = N - 1$$

where N is the number of identified sky lines. Consistently, for $n = 0$ a median offset would be computed.

--wcolumn: Name of sky line catalogue table column with wavelengths. *Default*: WLEN

This is the name of the MASTER_SKYLINECAT table column where the arc lamp reference wavelengths are listed. This would allow the usage of any FITS table, supplied by the user, containing a list of sky lines wavelengths to be used by the sky lines alignment task. The only requirement is that the listed wavelengths are given in Ångstrom, and that they are sorted from blue to red. Note that it is not necessary to provide a sky lines catalogue to make the sky alignment work: see entry MASTER_SKYLINECAT on page 93 for more details.

Spatial curvature calibration

Currently the input curvature model is not aligned to the observed scientific slit spectra. This will be implemented in the next release: in the meantime offsets up to 1 pixel may be observed in the tracing of scientific spectra.

Flat field correction

--flatfield: Apply flat field correction. *Default*: TRUE

Setting this parameter makes mandatory to specify a normalised flat field frame (see page 81, entry MASTER_NORM_FLAT_MXU). The flat field correction consists in dividing the bias subtracted input scientific frame by the normalised flat field frame.

Sky subtraction

--skylocal: Subtract sky spectrum from CCD scientific data. *Default*: TRUE

The local sky subtraction consists on modeling the sky trend for each row of spatial pixels for each spectrum on the CCD.[37] The advantage of this method is that the signal is not resampled before the sky is subtracted, reducing in this way the problems related to small-scale interpolation.

A MAPPED_SKY_SCI_MXU (page 96) is produced in this case. Note that global and median sky subtractions cannot be used if the local sky subtraction is applied.

Note however that this method does not apply well to curved or tilted slits: in such case, the *--skymedian* option should be preferred. It is for this reason that, in case of LSS or LSS-like data *--skylocal* option is not implemented, and it's an alias to the *--skymedian* option.

--skymedian: Subtract sky spectrum from rectified scientific data. *Default*: FALSE

The median sky subtraction consists on subtracting a median value of the sky for each wavelength pixel of each rectified slit spectrum.[38]

In general the subtraction of a rectified sky spectrum from rectified data does not give the best results, and in almost all cases the local sky subtraction (see parameter *--skylocal*) should be preferred.

A MAPPED_SKY_SCI_MXU (page 96) is produced in this case. Note that global and median sky subtractions are not mutually exclusive.

[37] This is an iterative process: initially the sky trend is estimated with a robust linear fitting, then outliers (e.g., objects) are rejected, and according to the slith length the sky is trended using a 2nd degree polynomial.

[38] This is an iterative process: initially the sky is estimated as the median value of all the pixels at the same wavelength, then this first estimation of the sky is subtracted, and the objects are detected; finally the median level is evaluated only on pixels outside the object detection spatial interval.

--*skyglobal:* Subtract global sky spectrum from CCD. *Default*: FALSE

In general the subtraction of a global sky spectrum does not give the best results, because the spectral resolution may vary significantly with the position on the CCD. However this operation may turn out to be useful in case either a local or a median sky subtraction would actually destroy spectra from extended objects that fill all, or almost all, the extension of a slit. See entries UNMAPPED_SKY_SCI_MXU and GLOBAL_SKY_SPECTRUM_MXU on page 98 for more details.

--*cosmics:* Eliminate cosmic rays hits. *Default*: FALSE

If this parameter is set, then either the local or the global sky subtraction must be requested (see parameters --*skylocal* and --*skyglobal*). Cosmic rays cleaning is almost always superfluous, and should be viewed as mere cosmetics applied to the extracted slit spectra (see entry MAPPED_SCI_MXU, page 96). Cosmic ray hits are removed anyway by the optimal extraction procedure of the detected objects.

Objects detection and extraction

--*dispersion:* Resampling step for rectified and wavelength calibrated spectra. *Default*: 0.0 Å/pixel

This parameter is mandatory (using the default 0.0 would generate an error message). The default value for this parameter, depending on the applied grism, is included in the GRISM_TABLE (see page 47): this value is the same that was used by the pattern-matching task for the reference lines identification, which is very close to the mean spectral dispersion (see Section 9.8.3, page 86). It is however possible to specify here any resampling step, if it is found more appropriate: in some extreme cases to resample the signal at a higher resolution may be essential to prevent information loss. To undersample the signal, on the other side, is never advisable (and it makes ineffective the flux conservation correction – see the *flux correction* section ahead).

The products that are directly affected by the --*dispersion* parameter are the following:

- GLOBAL_SKY_SPECTRUM_MXU,
- MAPPED_ALL_SCI_MXU,
- MAPPED_SCI_MXU,
- MAPPED_SKY_SCI_MXU,
- REDUCED_ERROR_SCI_MXU,
- REDUCED_SCI_MXU, and,
- REDUCED_SKY_SCI_MXU.

Note that the --*dispersion* value must refer to the real CCD pixel size: the given value of the resampling step is internally multiplied by the rebin factor, to match the real pixel size of the input data.[39] In this way the value of the resampling step is made independent from the CCD reading mode, guaranteeing that the same supersampling factor is always applied.

--*startwavelength:* Start wavelength in spectral extraction. *Default*: 0.0 Ångstrom

See the --*endwavelength* parameter.

[39]The rebin factor along the dispersion direction is written to the FITS header keyword ESO DET WIN1 BINX.

--*endwavelength:* End wavelength in spectral extraction. *Default*: 0.0 Ångstrom

This parameter, together with the --*startwavelength* parameter, defines the wavelength interval to be extracted. Optimal values of the extraction interval, depending on the applied grism, are included in the GRISM_TABLE (see page 47). Both --*startwavelength* and --*endwavelength* must be otherwise specified (leaving them to 0.0 would generate an error message). It is generally not advisable to specify an extraction interval that is wider than the calibrated interval.

--*slit_margin:* Spectrum edge pixels to exclude from object search. *Default*: 3 (rebinned) pixel

The object detection task will reject objects that are detected too close to the edges of a slit spectrum. There might be different reasons for this, such as objects would be truncated, too close to a confusion region, etc.

--*ext_radius:* Maximum extraction radius for detected objects. *Default*: 6 (rebinned) pixel

The default value is generally good when dealing with point-like objects, but it should be adapted to the size of more extended objects when necessary. Large values of the extraction radius would not harm the extraction quality if an optimal extraction algorithm is applied, but may have devastating effects on the results of a simple aperture extraction. The applied extraction interval is reduced in case nearby objects are detected: an intermediate position between two objects, computed according to the objects luminosity ratio, is never passed.

--*cont_radius:* Contamination radius. *Default*: 0 pixel

This parameter may help to prevent the extraction of contaminated objects. The contamination radius is the minimum distance at which two point-like objects of equal luminosity are assumed not to contaminate each other. For two objects having different luminosities the reciprocal contamination distances depend on their luminosity ratio. Indicating with L_o the peak value of one object integrated spatial profile and with L the peak value of a nearby object, the quantity

$$S = C \cdot \left(\frac{L}{L_o}\right)$$

is computed, where C is the specified contamination radius. If the distance between the two objects is less than S, the examined object is flagged as contaminated and is not extracted. This empirical formula has the effect of assigning a larger contamination radius to relatively brighter objects with respect to dimmer ones.

--*ext_mode:* Object extraction method. *Default*: 1

Only two methods are currently available for spectral extraction: --*ext_mode* = 0 corresponds to simple aperture extraction, while --*ext_mode* = 1 applies Horne's optimal extraction [20].

Flux calibration

Currently only factors for flux conservation in the rebin operations is offered. No relative flux calibration based on a spectroscopic standard star observation is implemented yet.

--*flux:* Apply flux conservation factors. *Default*: TRUE

The flux conservation factors applied to the rebinned and extracted slit and object spectra is the ratio between the resampling step and the original signal sampling, along both the spatial and the dispersion directions.[40]

--time_normalise: Apply exposure time normalisation to relevant products. *Default*: TRUE

The following products are afftected by this parameter:

- GLOBAL_SKY_SPECTRUM_MXU,
- MAPPED_ALL_SCI_MXU,
- MAPPED_SCI_MXU,
- MAPPED_SKY_SCI_MXU,
- REDUCED_ERROR_SCI_MXU,
- REDUCED_SCI_MXU, and,
- REDUCED_SKY_SCI_MXU.
- UNMAPPED_SCI_MXU, and,
- UNMAPPED_SKY_SCI_MXU.

9.10 fors_extract

This recipe is used for reducing FORS1 and FORS2 LSS, MOS and MXU scientific spectra. It is identical to the recipe *fors_science*, with the only difference that a GLOBAL_DISTORTION_TABLE (see page 49) is required in input instead of the DISP_COEFF_MXU, CURV_COEFF_MXU, and SLIT_LOCATION_MXU tables (see pages 77, 79, and 83). This recipe is necessary for on-line data reduction on Paranal, where the extraction mask computed with the recipe *fors_calib* may not be immediately available: in this case a general spectral distortion description must be used to generate the appropriate standard extraction mask for any slits or slitlets configuration, and for any available instrument mode (MOS, MXU, LSS). The results are often less accurate than those obtained with the *fors_science* recipe, but *fors_extract* may turn out to be occasionally very useful for recovering possible failures of the standard data reduction procedure.

9.11 fors_sumflux

This recipe is used to monitor any lamp flux on the CCD. The input raw image should be either a FLUX_ARC_LSS or a FLUX_FLAT_LSS frame. After the background subtraction the total signal is integrated and divided by the exposure time and by the total number of CCD original pixels (keeping into account a possible rebinned readout). In the case of FORS2 frames the background is the median level evaluated from the available overscan regions. In the case of FORS1 data before the blue upgrade [10], where overscan regions are missing, the background is evaluated as the median level of the first 200 CCD columns for flat field data, while for arc lamp data a background map evaluated from the regions without spectral lines is computed and subtracted. The background subtracted frame is written to output in all cases, and the QC parameters QC LAMP FLUX and QC LAMP FLUXERR are computed.

[40] In order to have a view of the flux conservation correction, it is possible to run the *fors_science* recipe twice, first setting *--flux* = FALSE and then setting *--flux* = TRUE, finally computing the ratio between the corresponding MAPPED_ALL_SCI_MXU product images.

9.11.1 Input files

FLUX_FLAT_LSS: exposure for flat field lamp monitoring.

FLUX_ARC_LSS: exposure for arc lamp monitoring.

9.11.2 Output files

FLUX_ARC_LSS: frame including the background subtracted integration region.

9.11.3 Configuration parameters

--*xlow:* X coordinate of lower left corner of integration region *Default*: 0 pixel

--*ylow:* Y coordinate of lower left corner of integration region *Default*: 0 pixel

--*xhigh:* X coordinate of upper right corner of integration region *Default*: 0 pixel

--*yhigh:* Y coordinate of upper right corner of integration region *Default*: 0 pixel

If the default is used (i.e., all the configuration parameters are left to zero) the whole CCD is integrated.

10 Algorithms

The data reduction procedures applied by the pipeline recipes currently in use (see Section 8) are described here in some detail.

10.1 Propagation of photonic and readout noise

The imaging data reduction of the FORS pipeline is characterised by a complete propagation of photonic noise and CCD readout noise to all of its products. In particular, all the image products consist of a FITS file containing two images: the pixel data values are included in the FITS primary array, while the statistical error at each pixel is included in the FITS first extension. Errors are propagated using the error propagation formula in the approximation of gaussian statistics (see also Section 10.2.1, page 106 about this topic). For example, if D is the value of a pixel (x, y) in a raw frame, and B the corresponding value of the bias master calibration frame, the resulting bias subtracted value will be

$$S = D - B$$

The corresponding error, keeping into account the contribution of the readout noise r (read from the header keyword ESO DET OUT1 RON and converted to ADU), is computed as

$$\Delta S = \sqrt{r^2 + (\Delta B)^2 + \frac{D - B}{g}}$$

where g is the gain in e^-/ADU and ΔB is the (propagated) error of the master bias value. With regard to different techniques of image stacking, in the case of the average, the minmax rejection, and the k-sigma clipping methods the stacked frame error is computed as

$$\Delta S = \frac{\sqrt{\sum_i (\Delta S_i)^2}}{N}$$

where N is the number of non-rejected values. In the case of median stacking the error is computed as

$$\Delta S = f(N) \frac{\sqrt{\sum_i (\Delta S_i)^2}}{N}$$

where $f(N)$ is 1 for $N \leq 2$, and it converges towards $\sqrt{\pi/2}$ for $N \to \infty$.

In the case of the spectral data reduction (performed with the recipes *fors_calib* and *fors_science*), errors are just estimated on the final products of the *fors_science* recipe. This implies that the errors introduced by both flat fielding and master bias subtraction are neglected. The applied formula is

$$\Delta S = \sqrt{r^2 + \frac{(D - B)/F}{g}}$$

where F is the bias subtracted and normalised master sky flat field. The variance of the optimally extracted object spectra is determined as in [20], i.e. as the inverse of the sum of the optimal weights.

10.2 Frame zeropoint computation

It may appear puzzling that in the aligned photometric table (ALIGNED_PHOT product, see page 70), which reports the zeropoint estimates for each detected standard star, some of the optimal weights (listed in column **weight**) may be negative. How is it possible that an optimally weighted average, as applied for the estimation of the frame zeropoint, would include negative weights?

In order to answer this question, a brief recap of error propagation may be in order.

10.2.1 Error propagation in a nutshell

If a quantity y is a function of n independent variables x_i

$$y = y(x_1, x_2, ..., x_n)$$

the variance of y, $(\Delta y)^2$, can be derived from the variances of the independent variables applying the usual formula for error propagation

$$(\Delta y)^2 = \sum_{i=1}^{n} \left(\frac{\partial y}{\partial x_i}\right)^2 (\Delta x_i)^2$$

However, this is valid as long as the variables x_i are independent from each other: if the variable x_i correlates with the variable x_j, then their covariance

$$C_{ij} = \mathrm{cov}(x_i, x_j)$$

will be different from zero (by definition). In this case not just the variances but also the covariances of the independent variables x_i must be propagated. Variances and covariances are collected in the $n \times n$ covariance matrix $\mathbf{C}$. The variances lie on the diagonal of the matrix, since

$$C_{ii} = \mathrm{cov}(x_i, x_i) = \mathrm{var}(x_i) = (\Delta x_i)^2$$

Moreover $\mathbf{C}$ is a symmetric matrix, because

$$C_{ij} = \mathrm{cov}(x_i, x_j) = \mathrm{cov}(x_j, x_i) = C_{ji}$$

With non-diagonal elements different from zero, the full expression for the error propagation is

$$(\Delta y)^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\partial y}{\partial x_i} \frac{\partial y}{\partial x_j} C_{ij}$$

which turns into the usual formula in case the covariance matrix $\mathbf{C}$ were diagonal.

To further generalise, let us now suppose to have m quantities y_k ($1 \leq k \leq m$), depending on n values x_i ($1 \leq i \leq n$). In this case propagating errors means to transform the covariance matrix $\mathbf{C}$, related to the x_i, into the covariance matrix $\mathbf{G}$ related to the y_k. This is obtained by applying the most general form of the error propagation,

$$G_{kl} = \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\partial y_k}{\partial x_i} \frac{\partial y_l}{\partial x_j} C_{ij}$$

where

$$G_{kl} = \mathrm{cov}(y_k, y_l)$$

and

$$G_{kk} = \mathrm{var}(y_k) = (\Delta y_k)^2$$

for the diagonal terms. In matrix notation, if **J** is the jacobian of the transformation from the x's to the y's

$$J_{ij} = \frac{\partial y_i}{\partial x_j}$$

the error propagation formula is a transformation of the covariance matrix:

$$\mathbf{G} = \mathbf{J}\mathbf{C}\mathbf{J}^{\mathbf{T}}$$

The covariance matrix **G** is the one to apply for further error propagation into new quantities computed in terms of y_k's. In summary, error propagation is nothing but the covariance matrix propagation.[41]

Note that in the special case **C** were diagonal, **G** might still contain non-diagonal elements. The formula to apply would become

$$G_{kl} = \sum_{i=1}^{n} \frac{\partial y_k}{\partial x_i} \frac{\partial y_l}{\partial x_i} (\Delta x_i)^2$$

If both y_k and y_l (with $k \neq l$) are a function of x_i, then then both derivatives $\frac{\partial y_k}{\partial x_i}$ and $\frac{\partial y_l}{\partial x_i}$ would be different from zero, and therefore the covariance G_{kl} would be different from zero too. This is indeed what is expected: if both y_k and y_l share one or more common terms, they would turn out to be correlated.

This is exactly the case in the problem of zeropoint computation, where the zeropoints derived for each photometric standard star are indeed based on a number of shared quantities, such as the assumed atmospheric extinction coefficient E and the linear color correction term Γ, which are considered valid for all stars (see table 9.6.1, page 68, for a list of the quantities used in the zeropoint determination).

An example may be useful to clarify how to propagate errors of correlated quantities.

Let us consider the transformation

$$\begin{aligned} F_1 &= m a_1 \\ F_2 &= m a_2 \end{aligned}$$

In this case the quantities F_1 and F_2 depend (as a whole) on the quantities a_1, a_2, and m. Since m is shared by the two equations, F_1 and F_2 will be correlated. Assuming that a_1, a_2, and m are independent quantities, it is easy to determine the variance of F_1 and F_2 applying the usual

$$\begin{aligned} (\Delta F_1)^2 &= \Big(\frac{\partial F_1}{\partial a_1}\Big)(\Delta a_1)^2 + \Big(\frac{\partial F_1}{\partial m}\Big)(\Delta m)^2 \\ (\Delta F_2)^2 &= \Big(\frac{\partial F_2}{\partial a_2}\Big)(\Delta a_2)^2 + \Big(\frac{\partial F_2}{\partial m}\Big)(\Delta m)^2 \end{aligned}$$

[41] This statement is valid under the assumption that the first order derivatives of the transformation are all that is needed for propagating errors. The error propagation formula is derived by Taylor expansion of the transformation, where higher order terms are neglected.

that is

$$\begin{aligned}(\Delta F_1)^2 &= m^2(\Delta a_1)^2 + a_1^2(\Delta m)^2 \\ (\Delta F_2)^2 &= m^2(\Delta a_2)^2 + a_2^2(\Delta m)^2\end{aligned}$$

However, if the quantities F_1 and F_2 are used in a further transformation – for instance, to derive a quantity F corresponding to their sum

$$F = F_1 + F_2$$

the knowledge of the errors ΔF_1 and ΔF_2 is not enough to determine the error ΔF: also the covariances are needed, because the quantities F_1 and F_2 are correlated.

It is therefore necessary to transform the covariance matrix $\mathbf{C}$ of the quantities a_1, a_2, and m into the covariance matrix of the quantities F_1 and F_2. As shown, this is done by applying the $\mathbf{G} = \mathbf{JCJ^T}$, where $\mathbf{J}$ is the jacobian of the transformation from (a_1, a_2, m) to (F_1, F_2). In this specific case, since a_1, a_2, and m are independent, their covariance matrix is diagonal and includes only variances:

$$\mathbf{C} = \begin{pmatrix} (\Delta a_1)^2 & 0 & 0 \\ 0 & (\Delta a_2)^2 & 0 \\ 0 & 0 & (\Delta m)^2 \end{pmatrix}$$

The jacobian of the transformation is given by

$$\mathbf{J} = \begin{pmatrix} \frac{\partial F_1}{\partial a_1} & \frac{\partial F_1}{\partial a_2} & \frac{\partial F_1}{\partial m} \\ \frac{\partial F_2}{\partial a_1} & \frac{\partial F_2}{\partial a_2} & \frac{\partial F_2}{\partial m} \end{pmatrix} = \begin{pmatrix} m & 0 & a_1 \\ 0 & m & a_2 \end{pmatrix}$$

Hence the covariance matrix of (F_1, F_2) is

$$\mathbf{G} = \mathbf{JCJ^T} = \begin{pmatrix} m^2(\Delta a_1)^2 + a_1^2(\Delta m)^2 & a_1 a_2 (\Delta m)^2 \\ a_1 a_2 (\Delta m)^2 & m^2(\Delta a_2)^2 + a_2^2(\Delta m)^2 \end{pmatrix} = \begin{pmatrix} (\Delta F_1)^2 & a_1 a_2 (\Delta m)^2 \\ a_1 a_2 (\Delta m)^2 & (\Delta F_2)^2 \end{pmatrix}$$

Its diagonal terms are, as expected, the variances already found above applying the usual error propagation formula. The matrix $\mathbf{G}$ can now be used to derive the error on any quantity based on (F_1, F_2). For instance, if such quantity is $F = F_1 + F_2$, then the generic

$$(\Delta F)^2 = \sum_{i=1}^{2} \sum_{j=1}^{2} \frac{\partial F}{\partial F_i} \frac{\partial F}{\partial F_j} G_{ij}$$

becomes

$$(\Delta F)^2 = (\Delta F_1)^2 + (\Delta F_2)^2 + 2 a_1 a_2 (\Delta m)^2$$

If F_1 and F_2 were statistically independent the variance of F would simply be the sum of the variances of F_1 and F_2: however in this case the statistical correlation leads to a greater variance, by the additional term $2 a_1 a_2 (\Delta m)^2$. Only in case m were errorless ($\Delta m = 0$) there would be no correlation.

Naturally, this result would have also been obtained by directly applying the usual error propagation formula to $F = m a_1 + m a_2$. In simple cases such this one there is no need to get all the way to propagating covariance matrices, but when dealing with a larger or unknown number of variables it is safer and more convenient to use the covariance matrix and matrix algebra in error propagation.

10.2.2 Optimally weighted average

In order to estimate the frame zeropoint, a weighted average of the single zeropoints (computed for each identified standard star) is performed. The question is: what are the weights that would make for an *optimal* linear estimator? In other words, what is the best linear estimator that would minimise the variance of the estimated quantity? Typically the answer would be: each contributing quantity should be weighted by the inverse of its variance. However, this is only true for uncorrelated quantities. The individual zeropoints are correlated quantities, because they all depend on the same estimate of the atmospheric extinction, of the color correction term, and of the airmass.

It is very simple to determine the optimal weights leading to the best estimate, y, from the correlated quantities x_i: it is sufficient to apply the error propagation formula to the general expression of the weighted average

$$y = \frac{\sum_i \omega_i x_i}{\sum_i \omega_i}$$

As seen in the previous Section, the variance of the average is given by

$$\begin{aligned}(\Delta y)^2 &= \sum_i \sum_j \frac{\partial y}{\partial x_i}\frac{\partial y}{\partial x_j} C_{ij} \\ &= \sum_i \sum_j \frac{w_i}{\sum_k \omega_k}\frac{w_j}{\sum_k \omega_k} C_{ij} \\ &= \frac{1}{(\sum_k \omega_k)^2} \sum_i \sum_j \omega_i \omega_j C_{ij}\end{aligned}$$

Minimum variance is reached where all its partial derivatives are zero:

$$\frac{\partial(\Delta y)^2}{\partial \omega_l} = \frac{2}{(\sum_k \omega_k)^2} \sum_i \omega_i C_{li} - \frac{2}{(\sum_k \omega_k)^3} \sum_i \sum_j \omega_i \omega_j C_{ij} = 0$$

that is

$$\sum_i \omega_i C_{li} = \frac{\sum_i \sum_j \omega_i \omega_j C_{ij}}{\sum_i \omega_i}$$

Setting $A_j = \sum_i \omega_i C_{ij}$ this relation becomes

$$A_l = \frac{\sum_i \omega_i A_i}{\sum_i \omega_i}$$

This shows that A_l is independent on l. The constancy of A_l is the condition to be fulfilled by optimal weights. Since only the relative weights are significant, A_l can be set to any arbitrary constant, and for simplicity it is set to 1:

$$\sum_i \omega_i C_{ij} = 1$$

Solving this system of linear equations is of course only possible if the covariance matrix $\mathbf{C}$ is invertible (which is generally the case). It can be noted that, in case the averaged quantities were statistically independent, $\mathbf{C}$ would be diagonal and the linear system would be promptly solved:

$$\omega_j = \frac{1}{C_{jj}} = \frac{1}{(\Delta x_j)^2}$$

which is the well known inverse-of-the-variance rule. The generic linear system can be written in matrix notation

$$\mathbf{Cw} = \mathbf{1}$$

where $\mathbf{w}$ is the vector of the optimal weights, and $\mathbf{1}$ a vector whose elements are all 1s. The solution is

$$\mathbf{w} = \mathbf{C}^{-1}\mathbf{1}$$

that is, the j-th weight is given by the sum along the j-th row (or column, since $\mathbf{C}^{-1}$ is symmetric) of the inverse of the covariance matrix

$$\omega_j = \sum_i C_{ij}^{-1}$$

This result can also be derived from first principles (without assuming any particular functional form of the estimator) by considering n correlated normally distributed random variables $\mathbf{x} = (x_1, ..., x_n)$ with the same expectation value y (as in the case of individual star zeropoints), and covariance matrix $\mathbf{C}$. The maximum likelihood principle can be used to determine which value of y makes the observed $\mathbf{x}$ most likely. The likelihood function is just the combined probability density function

$$L(y) = \frac{1}{(2\pi)^{n/2}|\mathbf{C}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{y})^T\mathbf{C}^{-1}(\mathbf{x}-\mathbf{y})\right)$$

where $\mathbf{y}$ is the constant vector $(y, ..., y)$. Maximizing L is equivalent to minimizing

$$\chi^2(y) = -2\log L = \text{const} + (\mathbf{x}-\mathbf{y})^T\mathbf{C}^{-1}(\mathbf{x}-\mathbf{y}) = \text{const} + \sum_{ij}(x_i - y)C_{ij}^{-1}(x_j - y)$$

which has derivative

$$\frac{d\chi^2}{dy} = \sum_{ij} 1 \times C_{ij}^{-1}(x_j - y) + \sum_{ij}(x_i - y)C_{ij}^{-1} \times 1$$

By relabelling summation indices ij and noting that $\mathbf{C}^{-1}$ is symmetric, we see that the two sums are equal, so the optimal estimate is given by

$$0 = \sum_{ij} C_{ij}^{-1}(x_i - y) = \sum_{ij} C_{ij}^{-1}x_i - \sum_{ij} C_{ij}^{-1}y,$$

or

$$y = \frac{\sum_{ij} C_{ij}^{-1} x_i}{\sum_{ij} C_{ij}^{-1}}$$

which is a weighted average with weights

$$\omega_i = \sum_j C_{ij}^{-1}$$

as before. It can be easily shown, by replacing this solution in the general expression of the variance $(\Delta y)^2$, that the variance of the optimal average is simply the inverse of the sum of all weights, which is also the sum of all the elements of the matrix $\mathbf{C}^{-1}$:

$$(\Delta y)^2 = \frac{1}{\sum_i \omega_i} = \frac{1}{\sum_{ij} C_{ij}^{-1}}$$

Hence it appears that the sum of all the optimal weights must always be positive. However, within this constraint, nothing forbids that *some* of the weights could be negative. But would this happen with real data?

The answer is yes. A covariance matrix is always a positive definite matrix: when it is diagonalised, its diagonal would just be made of variances which are by definition positive. This proves that the covariance matrix has only positive *eigenvalues*, and therefore must be positive definite. This just means that the error ellipsoid has always positive main axes, which is no big news. The inverse is also true: any positive definite matrix is also a possible covariance matrix (i.e., any ellipsoid represents a realistic uncertainty region). Within such conditions, it can be shown that negative weights are still a possibility: for instance, the following 2×2 covariance matrix

$$\mathbf{C} = \begin{pmatrix} 1.20 & 1.00 \\ 1.00 & 0.90 \end{pmatrix}$$

has eigenvalues 0.0388 and 2.061, so it is positive definite. The inverse is

$$\mathbf{C}^{-1} = \begin{pmatrix} 11.25 & -12.50 \\ -12.50 & 15.00 \end{pmatrix}$$

The optimal weights are therefore $\omega_1 = -1.25$ and $\omega_2 = 2.50$. Even though their sum is positive, one of the weights is negative.

What is the physical meaning of negative weights? Typically a weight is perceived as a measure of the "importance" of a quantity in the computation of a weighted average: a high weight means that a contribution counts a lot, while a zero weight means that a contribution does not count at all. A negative weight implies that a contribution counts, but negatively. This means that the optimal average of two values *may not lie between the values themselves*. What is happening in practice is that, when the variances of the contributing quantities are not enough to explain the observed variation, then the scatter can only be explained by the covariances.

To better clarify this, let us consider the case of the frame mean zeropoint computation: after applying the color term and the atmospheric extinction corrections to the instrumental magnitudes, the individual zeropoints derived for each standard star should ideally be all identical (if they were not expected to be identical, it would not make sense to average them). However, because of statistical errors, the single zeropoints will differ from each other. The optimal average represents the best estimator of the true zeropoint, because it tries to minimize the variance of the estimate. There are two ways to do that, and both are (implicitly) applied by the optimal estimator:

1. assign smaller weights to zeropoints with greater statistical uncertainty, and
2. minimise the scatter due to possible systematic errors.

Indeed, since the single zeropoints are based on the same color term and atmospheric extinction coefficients, the random error of such common quantities acts like a systematic error affecting all the computed zeropoints. This systematic error can be reduced by finding values for the color term and the atmospheric extinction coefficients which would reduce the scatter of the averaged zeropoints. This is what is implicitly offset by a negative weight: when the zeropoints of two different stars differ more than it could be likely expected from their variances, it is probably because the color correction was biased by an inaccurate color correction term. Modifying appropriately the color term would modify the zeropoints of *both* stars - which is how the best estimate of the mean zeropoint may happen to lie outside the interval between the contributing values.

10.3 The fors_photometry recipe

As seen in Section 9.7, page 73, the general model for the observed instrumental magnitude m_{ij} of star i on exposure j is

$$M_i - \Gamma \cdot C_i + p(A_j, C_i) + EA_j - Z_j - 2.5 \log g_j - 2.5 \log t_j + f(x_{ij}, y_{ij}) = m_{ij}$$

where the symbols are consistent with table 9.6.1 (page 68), x and y are coordinates on the detector, and non-linear dependencies are accounted for by the polynomials $p()$ and $f()$. All quantities are referred to the i-th star in the j-th exposure as indicated.

In practice, the equations are solved by moving all assumed (known) terms to the right hand side, and leaving the terms containing free (unknown) parameters on the left hand side. In matrix notation this equation system can be written (as in [25])

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

where $\mathbf{x}$ is the vector of unknown parameters, $\mathbf{A}$ the matrix of their coefficients, and $\mathbf{b}$ the vector of measurements.

For example, in the simple model where all Z_j are constrained to be equal to the same (unknown) value Z_o and all other parameters are assumed (catalogue magnitudes, E, Γ), and where the non-linear dependencies $p()$ and $f()$ are ignored, the equations would become:

$$Z_o = M_i - \Gamma \cdot C_i - m_{ij} + EA_j - 2.5 \log g_j - 2.5 \log t_j$$

The measurements (on the right hand side) are typically based on common quantities: for instance, the i-th star may be observed in more than one exposure, and therefore its catalogue magnitude and color would be shared among different right hand sides. For this reason the covariance matrix $\mathbf{C}$ of $\mathbf{b}$ is generally not diagonal.

In order to solve this equation system, the recipe *fors_photometry* computes first the covariance matrix of the right hand side. By the principle of maximum likelihood, and assuming gaussian errors, the χ-squared of the problem is

$$\chi^2 = (\mathbf{b} - \mathbf{A}\mathbf{x})^{\mathbf{T}} \mathbf{C}^{-1} (\mathbf{b} - \mathbf{A}\mathbf{x})$$

From this the minimum χ-squared solution for $\mathbf{x}$ is derived by setting $\mathrm{d}\chi^2/\mathrm{d}\mathbf{x} = 0$:

$$\mathbf{A}^{\mathbf{T}} \mathbf{C}^{-1} \mathbf{A}\mathbf{x} = \mathbf{A}^{\mathbf{T}} \mathbf{C}^{-1} \mathbf{b}$$

The minimum χ-squared solution is therefore

$$\mathbf{x} = (\mathbf{A^T C^{-1} A})^{-1} \mathbf{A^T C^{-1} b}$$

and is obtained by Cholesky-decomposition of $\mathbf{A^T C^{-1} A}$ [42]. The overall execution time is $O(n^3)$ (where n is the number of equations) and dominated by the computation of $\mathbf{C}^{-1}$. The errors on the solution are derived by propagating the errors of $\mathbf{b}$ to $\mathbf{x}$. $(\mathbf{A^T C^{-1} A})^{-1}$ happens to be the covariance matrix of $\mathbf{x}$, $\mathbf{C}_x$: in the transformation from $\mathbf{b}$ to $\mathbf{x}$

$$\mathbf{J} = (\mathbf{A^T C^{-1} A})^{-1} \mathbf{A^T C^{-1}}$$

is necessarily the jacobian of the transformation. Computing the covariance matrix of $\mathbf{x}$ from the covariance matrix of $\mathbf{b}$ using the error propagation formula (see Section 10.2.1, page 106) leads to:

$$\begin{aligned} \mathbf{C}_x &= \mathbf{J C J^T} \\ &= (\mathbf{A^T C^{-1} A})^{-1} \mathbf{A^T C^{-1} C} ((\mathbf{A^T C^{-1} A})^{-1} \mathbf{A^T C^{-1}})^{\mathbf{T}} \\ &= (\mathbf{A^T C^{-1} A})^{-1} \end{aligned}$$

Note that, in the special case where $\mathbf{C}$ is diagonal, the solutions for $\mathbf{x}$ and $\mathbf{C}_x$ reduce to the solutions for the general linear weighted least squares problem [26].

Naturally, it may happen that the equation system remains underdetermined — trying to fit the atmospheric extinction coefficient E when all exposures are obtained at the same airmass would lead to no (unique) solution. In this case the *fors_photometry* recipe will stop with a message of the kind:

```
[ ERROR ] fors_photometry: Pivot 1 of 2 is non-positive: -1.42336e-10
```

10.4 Overview of the spectral self-calibration procedure

A more detailed description of the instrument-independent tasks involved in the complete self-calibration procedure is given in the next sections. Here is just provided an overview, which is useful for setting the individual tasks in their appropriate context.

1. Retrieve from the reference arc lamp line catalogue the line pattern to be searched on arc lamp exposures.
2. After bias and background subtraction, examine the arc lamp exposure one row at a time. For each CCD row:
   (a) Run the 1D peak-detection task, to produce a list of reference arc lamp lines candidates.
   (b) Run the 1D pattern-recognition task, to select from the list of candidates a list of identified peaks. Not all the arc lamp lines are expected to be always identified, because the spectra are presumably distorted, and some CCD rows may cross a spectrum just partially, or even miss it entirely (see Figure 10.4.1).
3. Apply a preliminary wavelength calibration to each CCD row, within the specified wavelength range.
4. Choose the central wavelength of the grism as a reference wavelength.

[42] In [25] SVD is used

5. Find the CCD position of each connected region of CCD pixels containing the reference wavelength.
6. Run the 2D pattern-recognition task, to match the physical positions of the slits on the focal plane with the positions found on the CCD for the reference wavelength.
7. If requested, and if there are enough slits, fit a transformation between slits positions and CCD positions, and upgrade the list of reference positions on the CCD.
8. Trace the edges of each flat field spectrum, starting from the found positions of the reference wavelength.
9. Fit the traces with a low-degree polynomial. If requested, and if there are enough slits, fit also a global model of the obtained coefficients.
10. Extract the arc lamp spectra following the determined spatial curvature (interpolating fluxes along the spatial direction). For each row of each arc lamp spectrum:
    (a) Run the 1D peak-detection task on the extracted spectra, to produce a list of reference arc lamp lines candidates from the *whole* spectral range.
    (b) Run the 1D pattern-recognition task, using the pattern from the line catalogue, to select from the list of candidates a list of identified peaks.
    (c) Fit a relation between the positions of the identified peaks *vs* the corresponding wavelengths.[43]
11. If requested, and if there are enough slits, fit also a global model of the obtained coefficients, in order to improve the local solutions.

At this point the spectral extraction mask is completely determined, and for each spectrum a specific coordinate system is defined, where to a CCD pixel correspond a wavelength and a position on the telescope focal plane. If the instrument were stable, it would be possible to extract the scientific spectra applying directly this extraction mask. In general, however, the extraction mask obtained from the day calibration exposures should be aligned to the scientific spectra before being applied.

10.4.1 1D peak-detection

Many sophisticated methods are available for detecting peaks and determining their positions along a one-dimensional signal. Any one of them is in principle suitable for the 1D peak-detection task of an automatic MOS data reduction pipeline.

The most important thing to note, however, is that on a calibration approach based on pattern-recognition the strongest requirement is that *the searched pattern must be present in the data*.[44]

In a traditional approach, peaks are initially rejected by the peak-detection task (depending on their statistical significance), and finally by the model fitting task (if they are found to be outliers). But in case a pattern-recognition algorithm is applied, the significance of a peak should be primarily judged by its being part of the expected pattern.

For this reason virtually any flux excess – no matter how significant – should be flagged as a peak candidate.

[43] This is the local wavelength calibration.

[44] Or at least long uninterruped portions of it.

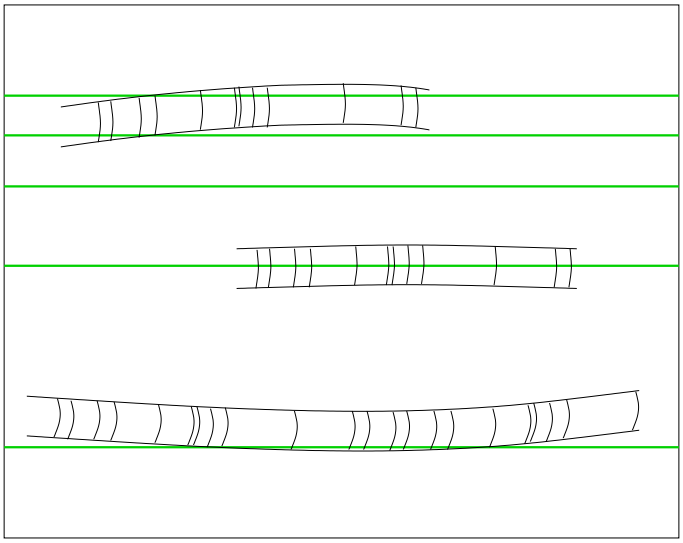

Figure 10.4.1: *CCD rows may not cut the whole range of the raw arc lamp spectra, because the spectra are not read along their curvature. However, even incomplete portions of the searched pattern can be identified by the pattern-matching algorithm.*

In the specific case of arc lamp spectra, the emission lines are very well exposed, and the S/N ratio of the lines to detect is almost always very high. This makes possible to apply a very simple 1D peak-detection method, based on the following two statements:

Any local maximum identifies a peak: in other words, a peak is identified by any pixel that is preceded and is followed by one pixel with a lower value (see Figure 10.4.2a).

A peak position is determined by parabolic interpolation of the three found pixel values: if a local maximum is found, the central pixel and its two neighbours are interpolated by a parabola. The position of the parabola's vertex is taken as the position of the peak (see Figure 10.4.2b). A peak position is then improved by applying more accurate methods: but if such methods fail, for instance finding positions that are significantly different from the parabolic ones, the original peak position is kept.

Even if obvious background noise fluctuations are excluded from the list of found peaks (e.g., by requiring that the values of the local maxima are greater than a given threshold), it is clear that with this method any contamination, hot pixel, cosmic ray, etc., would be reported as a "peak". This fulfills the critical requirement for the 1D pattern-recognition task reported above (see also Section 10.4.2, page 116).

The position x of a peak is given by

$$x = x_o + R$$

where x_o is the (integer) position of the pixel corresponding to a local maximum, and R the offset corresponding to the position of the maximum obtained by parabolic interpolation:

$$R = \frac{1}{2}\left(\frac{v_1 - v_{-1}}{2v_o - v_1 - v_{-1}}\right)$$

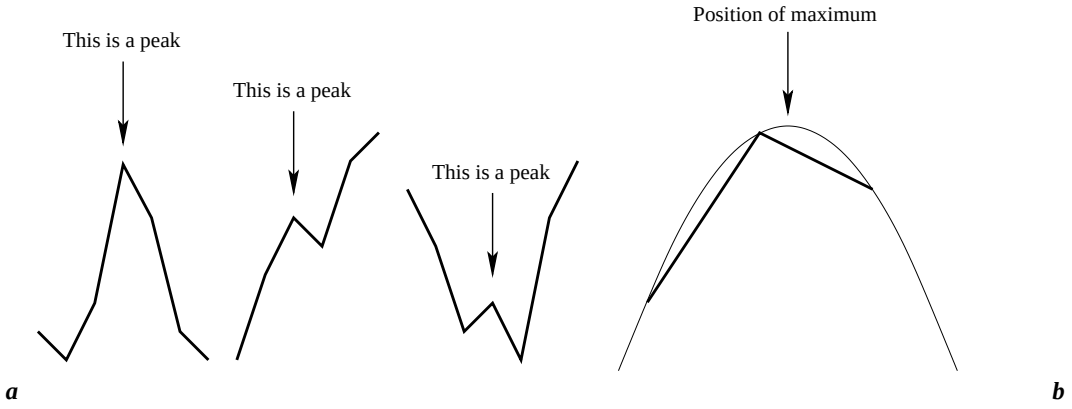

Figure 10.4.2: *a) Any local maximum identifies a peak. b) A peak position is determined by parabolic interpolation of three pixel values about the local maximum.*

where v_{-1}, v_o, and v_1 are the values of the pixels $x_o - 1$, x_o, and $x_o + 1$, always fulfilling $v_{-1} \leq v_o$ and $v_1 < v_o$, or $v_{-1} < v_o$ and $v_1 \leq v_o$ (see Figure 10.4.3).

The quantity R never diverges, and does not depend on the background level (assuming that the background level is the same for the three pixels).[45]

10.4.2 1D pattern-recognition

A simple method for 1D pattern-recognition has been developed in the attempt to increase the robustness of the wavelength calibration, despite possible mechanical instabilities of the instrument.

In order to work, this method just requires a rough expectation value of the spectral dispersion (in Å/pixel), and a line catalogue. The line catalogue should just include lines that are expected somewhere in the CCD exposure of the calibration lamp.[46]

The line-pattern would be searched in the list of CCD positions of arc lamp lines candidates produced by the 1D peak-detection task (see Section 10.4.1, page 114). Typically, the arc lamp lines candidates will include light contaminations, hot pixels, and other unwanted signal, but only in extreme cases this prevents the pattern-recognition algorithm from identifying all the reference lines. The pattern is detected even in the case the spectra contained more arc lamp lines than actually listed in the input line catalogue. In particular, this method is not deceived by spectral multiplexing, even in case of significant spectral overlap: all spectra are identified as separate instances of the same pattern.

This method is based on the assumption that the relation between wavelengths and CCD positions is with good approximation *locally* linear.[47]

[45] In the case of very wide slits, the emission lines profiles display a flat top that would prevent the direct application of this method. This is resolved by the preliminary application of a box filter as wide as the lines widths.

[46] The line catalogue represents the pattern that should be searched on the CCD, and adding extra lines would destroy this pattern. Note, however, that a catalogue including extra lines at its blue and/or red ends is still allowed.

[47] This is generally true for modern spectrographs, but if this were not the case the detected peaks positions may be preliminary transformed to roughly approach linearity, before being processed and identified by the pattern-matching task described here.

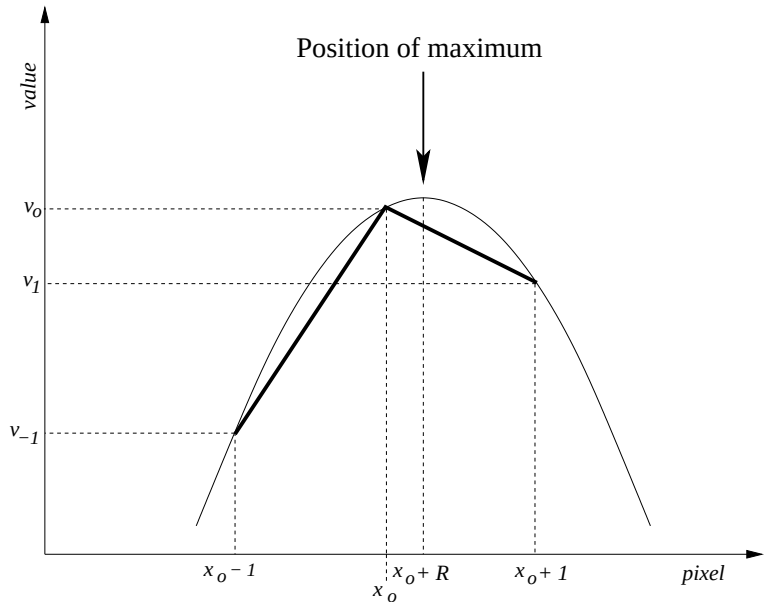

Figure 10.4.3: *Peak position estimate.*

The ratio between consecutive intervals in wavelength and in pixel is invariant to linear transformations, and therefore this quantity can be used in the recognition of *local* portions of the searched pattern. All the examined sub-patterns will overlap, leading to the final identification of the whole pattern.

Let be:

d: a rough value of the expected spectral dispersion (Å/pixel).

Δd: a tolerance value on the expected dispersion, large enough to ensure that, at all wavelengths, the real spectral dispersion will be included in the interval from $d - \Delta d$ to $d + \Delta d$.

W: the number of wavelengths in the input line catalogue.

N: the number of detected peaks.[48]

λ_i: the i-th wavelength of the input line catalogue, with $1 \leq i \leq W$.

p_j: the position of the j-th peak, with $1 \leq j \leq N$.

All the arc lamp wavelengths λ_i are taken one by one, excluding the first and the last wavelengths ($i = 1$ and $i = W$). The ratio R_i of the wavelength difference with the preceding and the following wavelength is computed:

$$R_i = \frac{\lambda_{i+1} - \lambda_i}{\lambda_i - \lambda_{i-1}}$$

The same ratio is now searched in the list of peak positions: for each i, all the peak positions p_j are checked, excluding the first and the last one, taking care however to exclude from the computation any interval that would

[48]Note that, as said above, it is typically $N > W$, or even $N >> W$.

be incompatible with the expected spectral dispersion. This is done in the following way: for each considered p_j, the following *forward* search interval p_{min} to p_{max} is defined (see Figure 10.4.4):

$$p_{min} = p_j + \frac{\lambda_{i+1} - \lambda_i}{d + \Delta d}$$

$$p_{max} = p_j + \frac{\lambda_{i+1} - \lambda_i}{d - \Delta d}$$

A *backward* search interval is similarly defined:

$$p_{min} = p_j - \frac{\lambda_i - \lambda_{i-1}}{d - \Delta d}$$

$$p_{max} = p_j - \frac{\lambda_i - \lambda_{i-1}}{d + \Delta d}$$

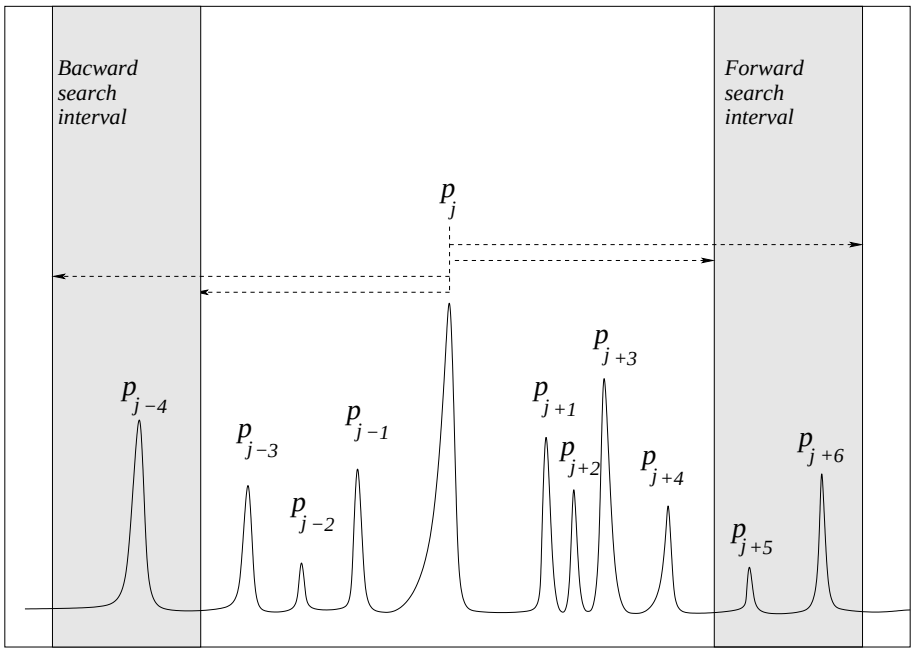

Figure 10.4.4: *Given a peak p_j, a forward and a backward search intervals compatible with an expected value of the spectral dispersion are defined. In this picture, peak positions p_{j-4}, p_{j+5} and p_{j+6} are used in the computation of the distance ratios to be compared with the wavelength interval ratio R_i. This process is repeated for each catalogue wavelength and for each peak position, accumulating scores that will allow the final peaks identification.*

Any peak position included either in the forward or in the backward search intervals is used for computing a ratio of distances from the position p_j (analogous to R_i). Each time a computed ratio equals R_i,[49] the three peak

[49]Within a given tolerance: this tolerance should be large enough to account for any deviation of the real wavelength calibration from the local linear approximation. A preposterously large value of 5% is used successfully with all the VIMOS and FORS instrument modes.

positions used for the computation (one is p_j, another is one from the backward search interval, and another is one from the forward search interval) are assigned respectively the wavelengths λ_{i-1}, λ_i, and λ_{i+1}. This assignment is not final: the same wavelength may even be assigned to different peaks, and the same peak may be assigned to different wavelengths. Each time a wavelength is assigned to a peak, a counter is increased, to keep a complete record of the assignments of wavelengths to peaks. Some wavelength assignments might be mistaken, and therefore not confirmed by successive comparisons.[50] The peaks that at the end of the analysis display a high score with respect to a given λ are considered identified, while ambiguous scores are rejected. The identified peaks are submitted to specialised sorting tasks that order them into separate self-consistent sequences (to take care of possible spectral multiplexing). This completes the peak identification process.

This procedure is surprisingly fast, and has been tested successfully with VIMOS spectroscopic data obtained with all the available grisms on all the instrument quadrants, both in MOS and IFU modes (i.e., using 48 independent instrument configurations), as well as all the FORS1 and FORS2 grisms in the LSS, MOS, and MXU intrument modes. All the arc lamp lines listed in the line catalogue are correctly identified without relying on a pre-existing instrument distortion modeling.

10.4.3 Determination of the spectral range

The spectral extraction range is specified by the user. A default range for each instrument configuration is given in the system configuration files (GRISM_TABLE, see page 47).

10.4.4 Choice of a reference wavelength

The reference wavelength is just an arbitrarily chosen origin for spectral coordinates (both in wavelength and in CCD pixels), used in the definition of the wavelength calibration and of the spatial curvature models. Typically, a reference wavelength may be chosen at the center of the extracted spectral range.

However, if different spectral ranges are specified for the same grism, or if the spectral range is computed automatically (see previous Section), a different reference wavelength might be computed for different data reduction sessions. Because of a need for consistency, for example in instrument health monitoring, the grism central wavelength specified in the FITS headers of the data to be processed is always used.

10.4.5 Position of the reference wavelength on the CCD

One of the products of the 1D pattern-recognition task run on the extracted CCD rows (see this Section, page 114) is a boolean image, where all the pixels including the reference wavelength are flagged. After applying morphological operators for reducing the impact of occasional gaps in the data, the baricenter of all the connected regions of flagged pixels is computed. The computed coordinates on the CCD should correspond to the positions of the slit centers on the mask plane. The match between the two sets is determined by the 2D pattern-recognition task (see next Section).

[50] Note that each peak is examined more than once, as the loop on wavelengths proceeds, since it may be included in forward and backward search intervals of other peaks.

10.4.6 2D pattern-recognition

The 2D pattern-recognition method applied here is based on a point-matching algorithm, and it is used for matching positions on the telescope focal plane (mask) with positions on the instrument focal plane (CCD). It will then be possible to determine the transformation between the two coordinate systems, and to match each spectrum with its slit.[51]

Straightforward invariants to translation, rotation, rescaling, and reflection, are distance ratios and angles. In the method described here, distance ratios are preferred, and the reflection-invariance is dropped for reducing the risk of false matches.

For each of the two sets of points – the *pattern* set P, and the *observed* set D – all the possible triangles are constructed. The sides of each triangle are read clockwise,[52] and their lengths L_1, L_2, and L_3 are conventionally listed starting from the longest side (if two sides are equal, the first of the consecutive equal sides is taken). An ordered pair, (α, β), can be associated to each triangle, with

$$\alpha = \frac{L_2}{L_1} \qquad\qquad \beta = \frac{L_3}{L_1}$$

Such quantities are used to match similar triangles from both sets. The matches are made by associating nearby points on the $\alpha - \beta$ plane (see Figure 10.4.5). To each triangle are also assigned the coordinates of their vertices, and the applied normalisation factor L_1.

Initially, only safe matches are selected, corresponding to (α, β) bins containing just *one* triangle from each of the two input sets.[53] Such matches are used to get a first estimate of the scale factor, that is taken as the median of all the scale factors derived from the pairs of matching triangles,

$$S = \frac{(L_1)_P}{(L_1)_D}$$

At this point the complete list of triangles is revisited, eliminating all the matches that are incompatible with the found scale factor.[54] Finally, a rotation angle is computed for each matching pair, and incompatibilities with the median rotation angle are eliminated as well.

From the surviving triangles a list of matching points can be drawn and the geometrical transformation between the two sets can be determined. With the fitted transformation, points that were possibily lost to the matching procedure may be recovered, and a better transformation obtained from the extended sample.

It should be noted that this procedure, like the human brain, fails for regular grids of points: in fact in this case there would be no bin in the $\alpha - \beta$ plane containing just one triangle pair. Regular grids of points are typical of MOS calibration masks, but such masks always contain at least one asymmetric point, misaligned with the rest of the grid. This single point is sufficient to create a great number of unique triangles, making this procedure work.

[51] The 2D pattern-recognition is not applied if less than three spectra are detected on the CCD: in such cases, just local solutions would be used. Incidentally, a mask containing just one or two slits can hardly be considered a MOS mask.

[52] Imposing a reading order to the triangle sides eliminates the reflection invariance of the computed quantities.

[53] A preliminary test on set P would ensure that the pattern is not ambiguous, i.e., that isolated points on the $\alpha - \beta$ plane exists.

[54] In practice, a third dimension is added to the $\alpha - \beta$ plane, corresponding to the absolute size of the triangles in one of the two input sets.

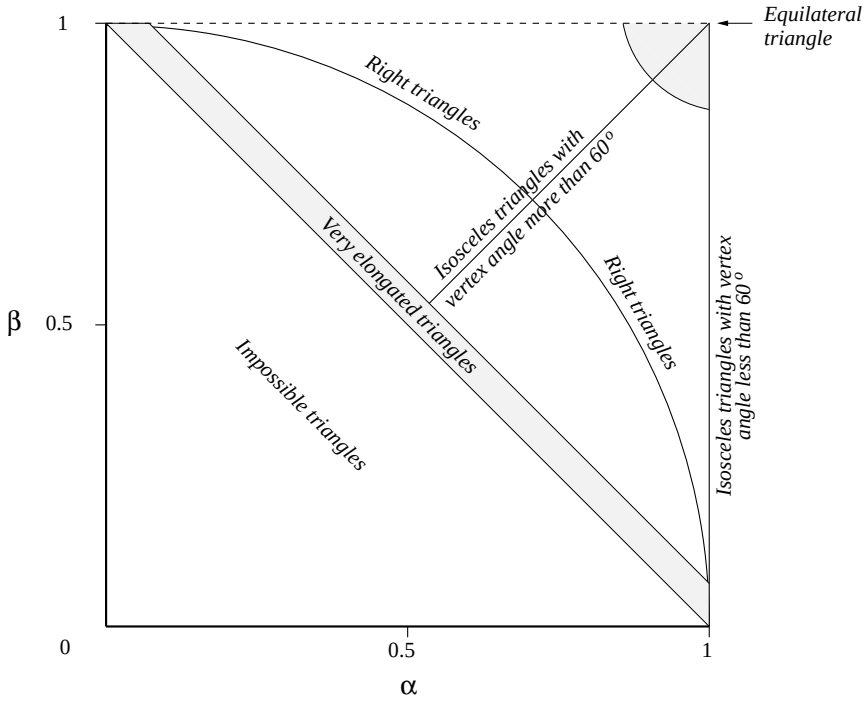

Figure 10.4.5: *The $\alpha - \beta$ plane. The long shaded region indicates very elongated triangles (including the cases of aligned points), while the region about the equilateral triangle includes ambiguous cases that would not lead to a safe identification of points: the triangles contained in those regions are therefore excluded from the analysis (unless they turn out to be the only triangles available). Note that the coordinates are cyclical: the line $\beta = 1$ (dashed) would include the same triangles described by the line $\alpha = 1$.*

A possible drawback of this method lies on the exploding number of triangles at the increase of the points in the pattern. The number of possible triangles that can be drawn from a distribution of n points is given by

$$N = \binom{n}{3} = \frac{n(n-1)(n-2)}{6}$$

A VIMOS mask may contain up to 200 slits, meaning more than three million triangles to be handled. Even if elongated and ambiguous triangles are excluded from the analysis, they still need to be computed, and the time complexity of this algorithm remains $O(n^3)$. For this reason a simplified version of this algorithm has been used in the FORS pipeline recipes implementation, where not all possible triangles are considered, but just those triangles defined by nearby slits.

The 2D pattern-recognition algorithm is also applied in the identification of standard stars, in the *fors_zeropoint* recipe: in this case the *observed* set, D, contains the positions (in pixel) of detected stars on a standard star field exposure. Typically hundreds of star positions would be available, but for efficiency reasons only the brightest ones are used for a preliminary identification. The input *observed* positions are therefore opportunely ordered from the brightest to the dimmest star positions. The *pattern* set, P, is derived from the input standard star catalogue. From the WCS defined in the FITS header of the image (which might be inaccurate, sometimes shifted by several pixels if the guide star was misidentified), a subset of stars expected in the field-of-view can be selected from the input catalogue: such stars will be stored in the *pattern*, ordered as well by their brightness,

and with their RA and Dec coordinates converted into CCD coordinates using again the WCS defined in the FITS header of the image. For an efficient pattern-recognition, the number of constructed triangles is drastically reduced by selecting just the 10 brightest catalogue stars, and searching their pattern among the 30 brightest observed stars. From this preliminary identification of a limited number of stars a first rough transformation from *pattern* to *observed* can be determined, and then applied to all the stars in *pattern* to complete the identification of all the standard stars in the observed set.

This 2D pattern-recognition method is also successfully applied in the correction of the WCS in the WFI pipeline, and as a possible recovery method for echelle instrument instabilities in the X-Shooter pipeline.

10.4.7 Optical distortion model determination

The optical distortion model may be (optionally) obtained by fitting a polynomial transformation to the matching points on the mask and on the CCD planes, as found by the 2D pattern-recognition task (see previous Section). The used polyomial model is described in Section 7.5.

Once the optical distortion model is determined, it is applied to the positions of the slits on the mask plane, improving the accuracy of their computed positions on the CCD.

No optical distortion model can really be defined if there are too few spectra on the CCD: in that case, just a local position of the reference wavelength is used for each individual spectrum, and the slits are left unidentified.

Note that slit identification is not essential to the data reduction, and it is hardly a requirement when very few slits are in use.

10.4.8 Tracing slit spectra edges

The spatial curvature is determined by tracing the slit spectra – typically from flat field and scientific exposures. Flat field spectra are ideal for this operation, because the signal is continuous and with high S/N ratio; on the other hand, it is generally necessary to trace also the scientific spectra, to compensate for possible instrument instabilities. Scientific spectra are generally traceable, because the exposure times are typically long enough to produce a very bright sky spectrum. In case the sky emission is not traceable, then the curvature model derived from the flat field exposures must be used.[55] Currently the pipeline does not support tracing of the scientific spectra.

Tracing spectral edges is not a simple task, because the slit spectra are not always so well detached and isolated from each other, and edges from different spectra may overlap. The only possibility is to try to determine a global trend of the spatial curvature based on the well traceable edges, in order to obtain the curvature also where it cannot be directly measured (see next Section).

10.4.9 Spatial curvature model determination

A local spatial curvature model is derived by fitting a low degree polynomial to the traces of one spectral edge. If enough spectra are available, the local curvature model may be superseded by a global description obtained by

[55] Tracing bright point-like object spectra is not a solution, as they are not distorted just by optics, but by atmospheric refraction too.

modeling the coefficients of the local models of all spectra. The used polyomial model is described in Section 7.5.

10.4.10 Extraction of slit spectra

The extraction of slit spectra consists in reading the spectra following their curvature. The extracted spectra are not wavelength calibrated. This extraction method is only applied to arc lamp or sky spectra before using them for determining the (local) wavelength calibration applying the 1D peak detection and pattern-recognition methods described in Sections 10.4.1 and 10.4.2.

All the spectra are read along the spatial direction (i.e., along the CCD columns), and each column is remapped to a new image where the spatial curvature is eliminated. In other words, the x coordinate of the rectified image is still the x coordinate of the CCD.

10.4.11 Alignment of the extraction mask to the scientific spectra

A variation of the instrument flexures between the calibration and the scientific exposures would invalidate the extraction mask derived from the calibrations. Also the removal and the insertion of the slit mask may slightly change the absolute positions of the slits on the telescope focal plane with respect to calibrations. Similarly, the grism alignment may also vary.

Flexures, non-reproducible mask positions, grism rotation, temperature changes, and other unpredictable effects, have a complex impact on each of the extraction mask components - i.e., the optical distortion model, the curvature model, and the wavelength calibration.

A practical approach to this problem might be to use the sky slit spectra for deriving a second extraction mask, following exactly the same procedure described in the previous sections – where a catalogue of sky-lines would be used instead of a catalogue of arc lamp lines. The sky-based extraction mask would be statistically less accurate than the one based on flat fields and arc lamps, but it could still be used to determine a best alignment of the high-quality extraction mask to the scientific observation.

First, the optical distortion models would need to be compared. The transformation matrix between one model and the other can be easily determined, because all CCD positions are already associated to the appropriate mask slit as a by-product of the data processing.

With a similar procedure, the modifications to be applied to spatial curvature and spectral dispersion models can also be derived.

Currently the FORS pipeline just derives a correction for the wavelength calibration.

A Abbreviations and acronyms

CPL	Common Pipeline Library
DFO	Data Flow Operations department
DFS	Data Flow System
DO	Data Organiser
DRS	Data Reduction System
ESO	European Southern Observatory
FITS	Flexible Image Transport System
FOV	Field Of View
FPN	Fixed Patter Noise
GUI	Graphical User Interface
IWS	Instrument WorkStation
MOS	Multi Object Spectroscopy
PAF	VLT PArameter File
PSO	Paranal Science Operations
PWS	Pipeline WorkStation
QC	Quality Control
RON	Read Out Noise
SDD	Software Development Division
SOF	Set Of Frames
UT	Unit Telescope
VLT	Very Large Telescope
WCS	World Coordinate System

B Troubleshooting Guide

See Sections 9.8.3 and 9.9.3. See also [11].