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 5.16

Date 15th Mar 2023

Prepared:	C. Izzo, L. de Bilbao, J. M. Larsen	15th Mar 2023	
	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, 7.6, 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
3.0	07/07/2009	All	Documenting new polarimetric pipeline
3.1	09/09/2009	1, 7, 8, 9, A, B	Integrating input from reviewers of 3.0
4.0	06/06/2010	1, 4.3, 4.4, 7.6-7.8, 9.8, 9.9, 9.12, 9.13,	Documenting flux calibration, and
4.1	02/02/2011	All	Correct units for flux calibrated data
4.2	02/02/2011	All	Minor changes. Added magsyserr parameter
4.3	24/05/2012	All	Changes for version 4.9.11
4.4	15/10/2012	All	Changes for version 4.9.18
4.5	01/03/2013	All	Changes for version 4.9.22
4.6	01/04/2013	All	Changes for version 4.9.23
4.7	01/11/2013	All	Changes for version 4.11.12
4.8	15/05/2014	All	Changes for version 4.12.12
4.9	01/10/2014	All	Changes for version 5.0.0
5.0	20/11/2014	All	Changes for version 5.0.2
5.1	20/04/2015	All	Changes for version 5.0.11
5.2	24/06/2015	All	Changes for version 5.0.14
5.3	15/04/2016	All	Changes for version 5.3.2
5.4	02/11/2016	All	Changes for version 5.3.11
5.5	09/03/2017	All	Changes for version 5.3.23
5.6	15/01/2018	All	Changes for version 5.3.30
5.7	26/02/2018	All	Changes for version 5.3.32
5.8	01/08/2018	All	Changes for version 5.3.34
5.9	13/08/2018	All	Changes for version 5.3.35
5.10	29/08/2018	All	Changes for version 5.3.36
5.11	25/02/2020	All	Changes for version 5.5.0
5.12	02/06/2020	All	Changes for version 5.5.3
5.13	20/05/2021	All	Changes for version 5.5.7
5.14	11/02/2022	All	Changes for version 5.6.1
5.15	03/08/2022	All	Changes for version 5.6.3
5.16	03/08/2022	9	Changes for version 5.6.4

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 environments, for the ESO *Data Management Operations* (DMO), 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 enable a more personalised processing of the data from the instrument.

This manual is a complete description of the imaging and spectroscopic data reduction recipes reflecting the status of the FORS pipeline as of 15th Mar 2023(version 5.6.4).

1.2 Acknowledgements

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

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

Wolfram Freudling (ESO-DMO) provided the idea and the expertise for the implementation of the photometric illumination correction.

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

In relation to the polarimetric package of the pipeline, Stefano Bagnulo (Armagh Observatory, Northern Ireland, UK) and Ferdinando Patat (ESO-DMO) provided the necessary experience to this project.

Emanuela Pompei, Kieran O'Brien, Emmanuel Jehin, and Gianni Marconi (ESO La Silla 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.

Updated versions of the present document may be found on [22]. For the work of data reduction, see also [17]. For general information about the current instrument pipelines status see [3]. Quality control information are at [6].

Additional information on the Common Pipeline Library (CPL), *Esorex* and *Gasgano* can be found at [1], [2], [8], [20], and [21]. A description of the instrument is in [12]. The FORS user manuals [10], [18], and [19], can be found in http://www.eso.org/instruments/fors/doc/.

[1] *Common Pipeline Library User Manual*. VLT-MAN-ESO-19500-2720. 13

[2] *CPL home page*. http://www.eso.org/cpl/. 13

[3] *Current pipeline status*. http://www.eso.org/observing/dfo/quality/pipeline-status.html. 13

[4] *Deliverables Specification*. VLT-SPE-ESO-19000-1618 (2.0).

[5] *DFS Pipeline & Quality Control – User Manual*. VLT-MAN-ESO-19500-1619.

[6] *ESO-Data Flow Operation home page*. http://www.eso.org/observing/dfo/quality/. 13

[7] *ESO DICB – Data Interface Control Document*. GEN-SPE-ESO-19400-0794 (3.0).

[8] *ESOREX home page*. http://www.eso.org/cpl/esorex.html. 13, 16, 20, 25, 27, 45

[9] *ESOTK pipeline manual*. https://ftp.eso.org/pub/dfs/pipelines/instruments/esotk/esotk-pipeline-manual-0.9.5.pdf. 180

[10] *FIMS User Manual*. VLT-MAN-ESO-13100-2308. 13

[11] *FORS Calibration Plan*. VLT-PLA-ESO-14700-2617. 62

[12] *FORS instrument home page*. http://www.eso.org/sci/facilities/paranal/instruments/fors/. 13

[13] *FORS Pipeline and Quality Control User's Manual*. VLT-MAN-ESO-19500-1771.

[14] *FORS1 and FORS2 Instruments Description*. http://www.eso.org/sci/facilities/paranal/instruments/fors/inst/. 14, 18

[15] *FORS1 and FORS2 Overview*. http://www.eso.org/sci/facilities/paranal/instruments/fors/overview.html. 18

[16] *FORS1 blue upgrade*. [14]/FactSheet/fors1_E2V_upgrade.html. 18, 130

[17] *FORS1+2 Data Reduction Cookbook*. VLT-MAN-ESO-13100-4030. 13, 16, 184

[18] *FORS1+2 Template User Manual*. VLT-MAN-ESO-13100-2309. 13

[19] *FORS1+2 User Manual*. VLT-MAN-ESO-13100-1543. 13, 108, 136, 175, 188

[20] *GASGANO home page*. http://www.eso.org/gasgano. 13, 16, 20, 21, 45

[21] *Gasgano User's Manual*. VLT-PRO-ESO-19000-1932. 13, 16, 21

[22] *Pipelines Web Page*. http://www.eso.org/pipelines. 13

[23] *VLT Data Flow System Specifications for Pipeline and Quality Control*. VLT-SPE-ESO-19600-1233.

[24] E. Bertin. *SExtractor v2.5*. http://astroa.physics.metu.edu.tr/MANUALS/sextractor/sextractor.pdf. 83, 85, 89

[25] E. Bertin, S. Arnouts. *SExtractor: Software for source extraction*. (1996), A&A Supp. 117, 393-404. 30, 83, 176

[26] E. Landi Degl'Innocenti, S. Bagnulo, L. Fossati. *Polarimetric standardization*. (2007), *The future of photometric, spectrophotometric and polarimetric standardization*, ASP Conference Series, Vol. 999. 136

[27] ESO/ASG, *http://www.eso.org/sci/observing/phase3/p3sdpstd.pdf*. *ESO Science Data Products Standard*, 5th edition, January 2013. GEN-SPE-ESO-33000-5335. 84, 124

[28] ESO/DOE, *https://ftp.eso.org/pub/dfs/pipelines/fors/fors-pmos-reflex-tutorial-1.8.pdf*. *Reflex PMOS FORS2 Tutorial*. VLT-SPE-XXX-XXX. 16

[29] ESO/DOE, *https://ftp.eso.org/pub/dfs/pipelines/fors/fors-spec-reflex-tutorial-1.26.pdf*. *Reflex SPEC FORS2 Tutorial*. VLT-SPE-XXX-XXX. 16

[30] ESO/DOE, *http://www.eso.org/sci/software/reflex//*. *Reflex User Manual*. VLT-SPE-XXX-XXX. 16

[31] ESO/SDD/DFI, *https://ftp.eso.org/pub/dfs/pipelines/fors/fors-img-reflex-tutorial-1.6.pdf*. *Reflex IMG FORS2 Tutorial*. VLT-SPE-XXX-XXX. 16

[32] F. Patat, M. Romaniello. *Error Analysis for Dual-Beam Optical Linear Polarimetry*. (2006), PASP 118:146-161. 136

[33] B. W. Holwerda. *Source Extractor for Dummies*. http://astroa.physics.metu.edu.tr/MANUALS/sextractor/Guide2source_extractor.pdf. 83, 89

[34] K. Horne. *An Optimal Extraction Algorithm for CCD Spectroscopy*. (1986), PASP 98, 609. 127, 156, 180

[35] L. Fossati, S. Bagnulo, E. Mason, E. Landi Degl'Innocenti. *Standard Stars for Linear Polarization Observed with FORS1*. (2007), *The future of photometric, spectrophotometric and polarimetric standardization*, ASP Conference Series, Vol. 364. 63

[36] A. U. Landolt. *$UBVRI$ Photometric Standard Stars in the Magnitude Range $11.5 < V < 16.0$ Around the Celestial Equator*. (1992), AJ 104, 340. 56

[37] M. Fukugita et al. *The Sloan Digital Sky Survey photometric system*. (1996), AJ 111, 1748. 57

[38] S. Bagnulo, M. Landolfi, J. D. Landstreet, E. Landi Degl'Innocenti, L. Fossati, M. Sterzik. *Stellar spectropolarimetry with retarder waveplate and beam splitter devices*. (2009), PASP 121, 993. 136, 138, 140, 141, 175

[39] P. B. Stetson. *Homogeneous Photometry for Star Clusters and Resolved Galaxies. II. Photometric Standard Stars*. (2000), PASP 112, 925-931. 56

[40] P. B. Stetson. *Stetson Photometric Standard Fields*. http://www3.cadc-ccda.hia-iha.nrc-cnrc.gc.ca/community/STETSON/standards/, August 2007. 56

[41] 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. 89, 92, 93, 94, 160, 161

[42] W. H. Press et al. *Numerical Recipes in C*. Second Edition (1992). 161

2 Overview

In collaboration with instrument consortia, the Science Operation Software Department (SOSD) of the Directorate of Engineering 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 can be 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 three front-end applications for launching pipeline recipes, *Gasgano* [21] *Esorex* [8] and *Reflex* [30] These applications can be downloaded separately from the ESO web pages (see [20], [8], [30]). An illustrated introduction to Gasgano is provided in the "Quick Start" Section of this manual (see page 20). The FORS pipeline comes with three Reflex workflows:

- Spectroscopy workflow. This is used for the LSS, MOS and MXU modes. This workflow has been extensively reviewed.
- Imaging workflow. This should be used for imaging data. It is a basic worflow provided for convinience.
- PMOS worfklow. This workflow can be used with polarimetry spectroscopy data. It is a basic worflow provided for convinience.

In order to use each of these workflows, please refer to the corresponding Reflex tutorial that exist, ([29], [31], [28]).

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 Troubleshooting Guide in the Appendix (page 184), and the pipeline related sections in the FORS1+2 Data Reduction Cookbook [17], which go even deeper into that.

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

There are now recipes (Sections 9.15, 9.16 and 9.17) that can perform the removal of telluric absorption from spectroscopic observations. These may be applied to 1D spectra produced by the data reduction recipes, either in LSS, MXU, or MOS configuration.

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

In Appendix C 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 was used since April 7, 2007 until the instrument was retired in March 31, 2009.

Only a brief description of the instruments is given here. For more details please refer to [14] and [15].

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 worked 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 the 2k x 2k Site detector with 24 μm pixels used til April 2007. The corresponding field sizes were 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. 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 [16].

A new CCD mosaic with blue optimised E2V detectors for FORS1 was 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.

FORS2 is offered with a detector consisting of a mosaic of two 2k x 4k MIT CCD (15 μm pixels). The FORS2 MIT mosaic provides greatly improved red sensitivity. The geometrical properties are the same as for the new FORS1 E2V CCD mosaic. Starting from April 2009 (P83) only FORS2 is operational. It has the polarisation mode previously offered for FORS1 and the E2V detector mosaic will only be offered in Visitor Mode.

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

- FORS2 had no polarimetric capability till April 2009, inheriting it from FORS1 afterwards.
- FORS2 offers an extended set of high-throughput volume phased holographic grisms (FORS1 offered 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 65.

Please note that the Reflex workflow has a step by step tutorial of its own. It can be downloaded from *http://www.eso.org/pipelines*

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* [8], from the automatic data management tools available at Paranal, or from the graphical *Gasgano* tool [20].

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 (page 22), 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 4.2, an MXU arc lamp exposure with a sequence of flat field exposures, bias frames, a catalog 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 4.3), 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* [21] for a more complete description of the *Gasgano* interface. See also [20].

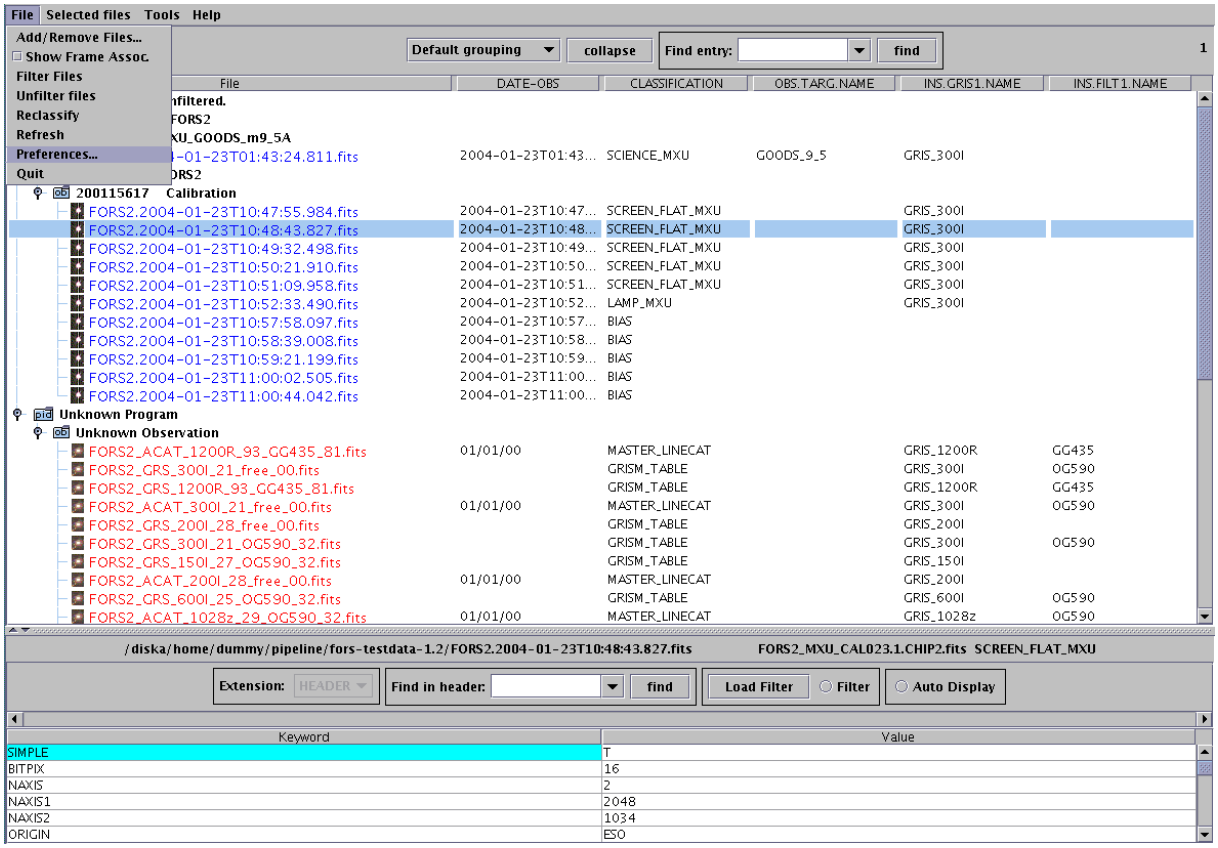

Figure 4.1: *The Gasgano main window.*

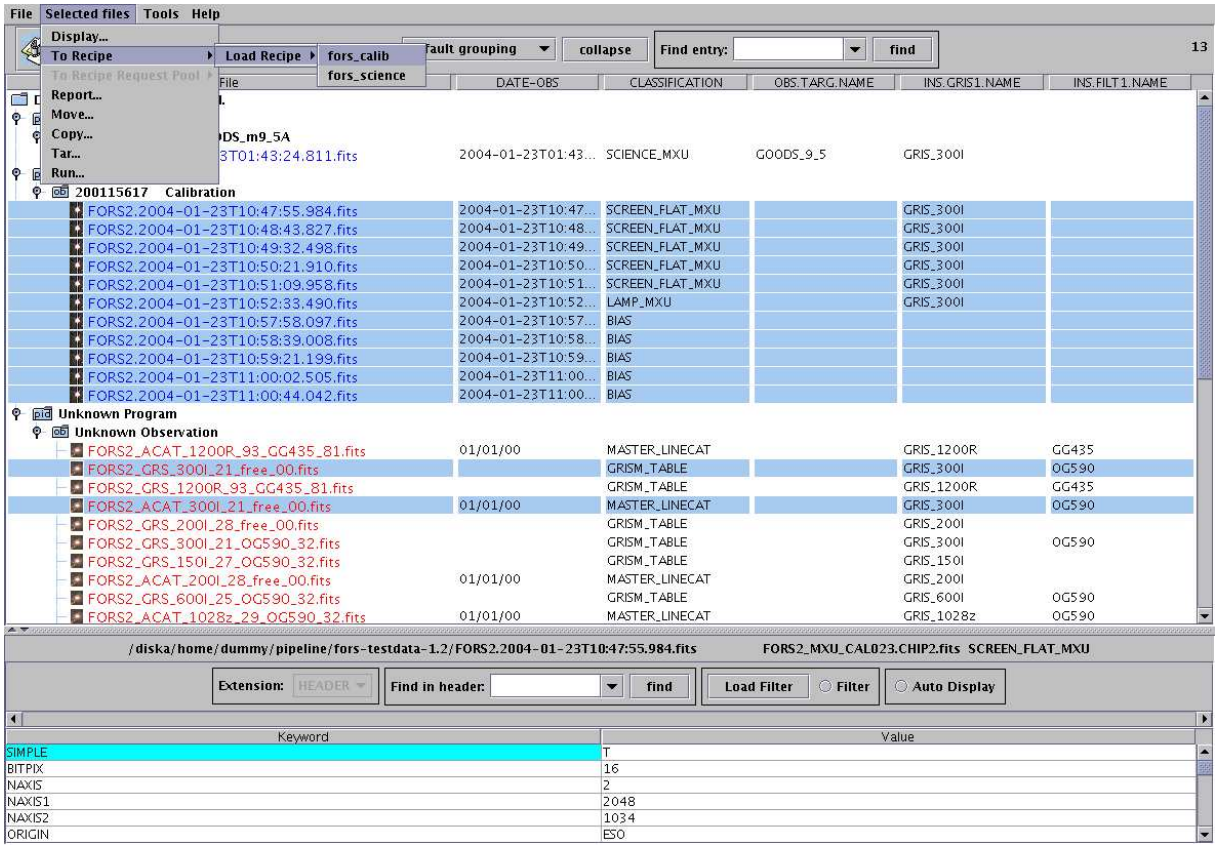

Figure 4.2: *Selecting files to be processed by a FORS pipeline recipe.*

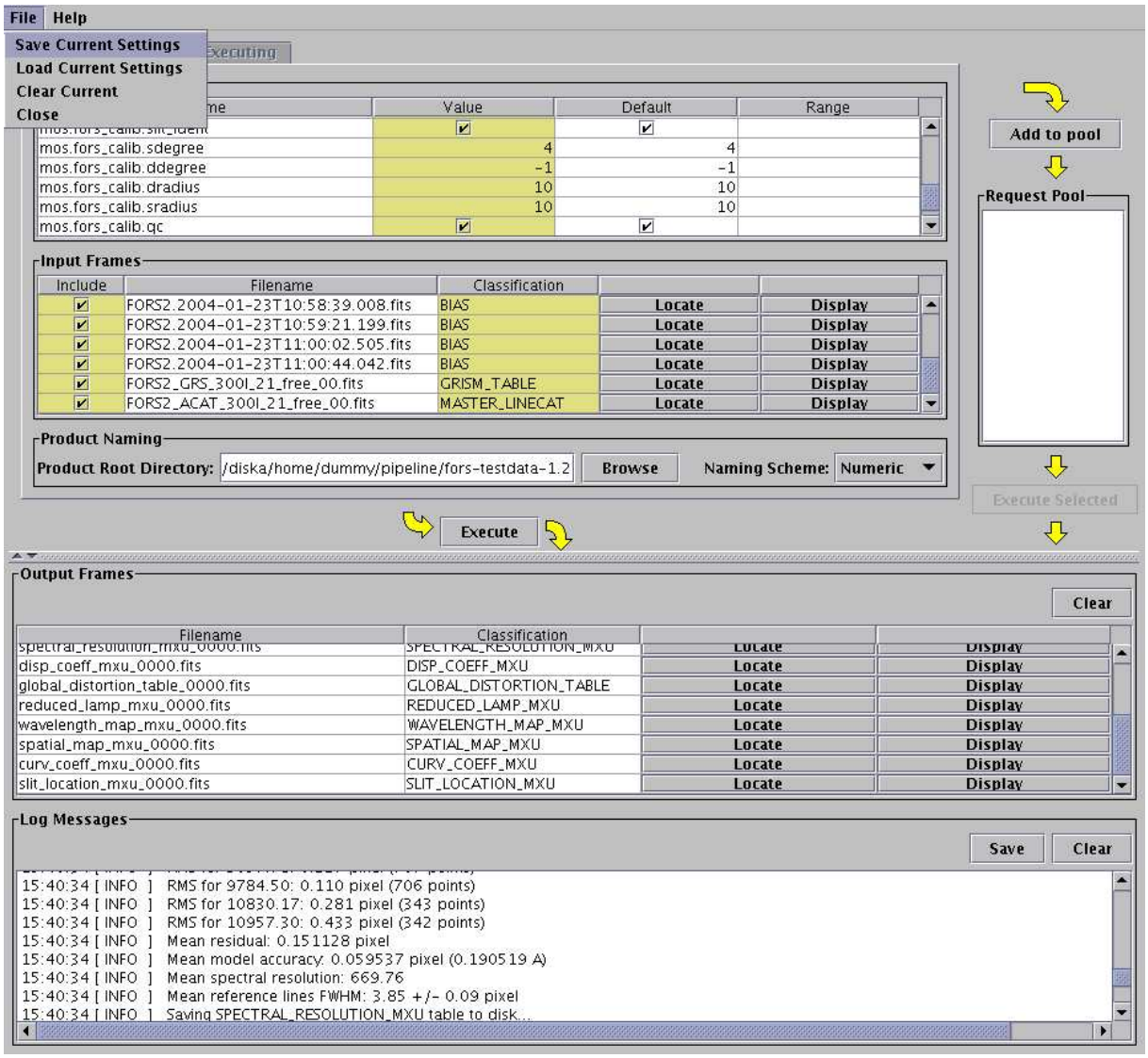

Figure 4.3: *The Gasgano recipe execution window.*

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 45). 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), which 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 20).

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 [8]) 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 4.3, page 24).

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.
fors.fors_calib.slit_ident=TRUE
```

[2]The *Esorex* recipe configuration file corresponds to the *Parameters* panel of the *Gasgano* recipe execution window (see Figure 4.3, page 24).

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 76.

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 [8].

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 catalogs), which are here assumed to be located in `/cal/fors1`.[4]

In order to produce a master bias calibration, the recipe *fors_bias* should be used (see Section 9.1, page 76). 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.

[4]The user decides where the calibration tables should be copied at installation time.

A twilight sky flat master calibration is produced using the recipe *fors_img_sky_flat* (see Section 9.5, page 81). 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 9.6, page 83), 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 [25]).

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 9.7, page 88), 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 catalog (FLX_STD_IMG) must be specified.

Currently the Landolt and the Stetson catalogs 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 catalogs carries a significant risk to produce inconsistent results, as the magnitudes are not derived with the same method in both catalogs, and as a consequence common stars can differ by up to 0.2 magnitudes. The use of two catalogs 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 [25]).

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 offset 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.[5] Note that this is an ideal session, where nothing goes wrong: for checking the quality of the results and troubleshooting, please refer to Appendix A.

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

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

the current directory. The only exception is represented by the standard calibration tables (e.g., line catalogs), which are here assumed to be located in `/cal/fors2`.[6]

The first step is to create a master bias with recipe *fors_bias* (described in the imaging data reduction Section 4.2). The input SOF will be:

File: *bias.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
```

The following command line will produce as master_bias.fits frame:

```
esorex fors_bias bias.sof
```

In order to process the calibration exposures available for the scientific observation, the recipe *fors_calib* is used (see Section 9.9, page 97). The input SOF may be defined as follows:

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_BIAS file is the one created in the previous step. The MASTER_LINECAT and the GRISM_TABLE are static calibration tables which are available in the calibration directories delivered with the pipeline recipes. The file FORS2_ACAT_300I_21_OG590_32.fits is the default catalog of reference arc lamp lines for grism 300I and filter OG590 of the FORS2 instrument. This catalog 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:

[6]The user decides where the calibration tables should be copied at installation time.

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, produced only if the recipe configuration --check is set.

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 6 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, produced only if the recipe configuration --check is set.

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

mapped_screen_flat_mxu.fits: flat field with distortion removed.

mapped_norm_flat_mxu.fits: normalised flat field with distortion removed.

Now the scientific frame can be processed, and for this the recipe *fors_science* is used (see Section 9.10, page 116). 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.

Support for a spectro-photometric calibration is also available, but in this case an atmospheric extinction table (see Section 7.8, page 62) and a spectral response curve for the present instrument configuration (see SPECPHOT_TABLE entry, page 123) must also be specified in input. Spectral response curves can be produced by applying the *fors_science* recipe to standard star exposures. Standard star spectra are reduced applying the long-slit data reduction strategy[7] which is described in the next Section.

Note that a MASTER_SPECPHOT_TABLE may be specified instead of a SPECPHOT_TABLE.

[7] 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.[8].

In the following example a FORS2 LSS observation of a standard star is processed.[9] Note that this is an ideal session, where nothing goes wrong: for checking the quality of the results and troubleshooting, please refer to Appendix A.

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).

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

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

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 catalogs), which are here assumed to be located in `/cal/fors2`.[10]

In order to produce a master bias calibration, the recipe *fors_bias* should be used (see Section 9.1, page 76). The input SOF may be defined as follows:

File: *bias.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
```

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
```

In order to process the calibration exposures available for the standard star observation, the recipe *fors_calib* is used (see Section 9.9, page 97). The input SOF may be defined as follows:

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_BIAS frame is the one just created. The MASTER_LINECAT and the GRISM_TABLE are static calibration tables which are available in the calibration directories delivered with the pipeline recipes. The file FORS2_ACAT_300I_21_OG590_32.fits is the default catalog of reference arc lamp lines for grism 300I and filter OG590 of the FORS2 instrument. This catalog 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:

[10] The user decides where the calibration tables should be copied at installation time.

```
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 9.10, page 116). 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
/cal/fors2/extinct_table.fits               EXTINCT_TABLE
/cal/fors2/ltt7379.fits                     STD_FLUX_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. An atmospheric extinction table and a standard star flux table (see Sections 7.8, page 62, and 7.9, page 62) are specified here, leading to the production of spectral efficiency and response curves for the present instrument configuration (see SPECPHOT_TABLE entry, page 123). The response curve can be used by recipe *fors_science* to produce also flux calibrated extracted spectra.

With the following command:

```
esorex fors_science sci.sof
```

the following products are created on disk:

specphot_table.fits: table with efficiency and response curves.

mapped_all_std_lss.fits: image with wavelength calibrated slit spectra.

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

mapped_flux_std_lss.fits: image with flux calibrated, 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_flux_std_lss.fits: image with flux calibrated 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.

reduced_flux_error_std_lss.fits: image with the statistical errors corresponding to the flux calibrated extracted objects spectra.

sky_shifts_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.

4.5 Example of PMOS data reduction using Esorex

The processing of spectro-polarimetric data is identical for FORS1 and FORS2.

In the following, a typical FORS1 PMOS data reduction procedure is described.[11] Note that this is an ideal session, where nothing goes wrong: for checking the quality of the results and troubleshooting, please refer to Appendix A.

It is assumed that the following data are available:

Eight scientific exposures, meant for linear polarization measurement, obtained at eight different angles ($k \cdot 22.50°$, with $k = 0, 1, 2, ...7$) of the half-wave retarder waveplate:

```
FORS1.2008-11-16T04:53:24.340.fits          SCIENCE_PMOS
FORS1.2008-11-16T04:55:10.841.fits          SCIENCE_PMOS
```

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

```
FORS1.2008-11-16T04:56:56.851.fits          SCIENCE_PMOS
FORS1.2008-11-16T04:58:42.861.fits          SCIENCE_PMOS
FORS1.2008-11-16T05:00:28.932.fits          SCIENCE_PMOS
FORS1.2008-11-16T05:02:14.862.fits          SCIENCE_PMOS
FORS1.2008-11-16T05:03:59.863.fits          SCIENCE_PMOS
FORS1.2008-11-16T05:05:44.844.fits          SCIENCE_PMOS
```

(other configurations would require 4 or 16 input frames for linear polarization measurements, and 2 or 4 input frames for circular polarization measurements).

Five flat field exposures obtained with the same slit configuration used for the scientific exposure:

```
FORS1.2008-11-16T09:06:08.980.fits          SCREEN_FLAT_PMOS
FORS1.2008-11-16T09:07:08.086.fits          SCREEN_FLAT_PMOS
FORS1.2008-11-16T09:08:07.071.fits          SCREEN_FLAT_PMOS
FORS1.2008-11-16T09:09:06.077.fits          SCREEN_FLAT_PMOS
FORS1.2008-11-16T09:10:05.114.fits          SCREEN_FLAT_PMOS
```

One or more arc lamp exposures[12] obtained with the same slit configuration used for the scientific exposure:

```
FORS1.2008-11-16T09:12:02.965.fits          LAMP_PMOS
FORS1.2008-11-16T09:19:43.041.fits          LAMP_PMOS
FORS1.2008-11-16T09:23:31.894.fits          LAMP_PMOS
FORS1.2008-11-16T09:27:05.935.fits          LAMP_PMOS
FORS1.2008-11-16T09:30:38.446.fits          LAMP_PMOS
FORS1.2008-11-16T09:34:24.578.fits          LAMP_PMOS
FORS1.2008-11-16T09:38:26.063.fits          LAMP_PMOS
FORS1.2008-11-16T09:42:43.218.fits          LAMP_PMOS
```

It is assumed here that a master bias frame has been already produced, either 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):

```
master_bias.fits                            MASTER_BIAS
```

All the listed data are meant to be obtained from the same FORS1 chip, with the same grism and filter in use. In this example it is assumed that the grism is 600B with no filter (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

[12] In this example the arc lamp exposures are eight, one for each angle of the retarder waveplate: this is typically required for quality control purposes. However, the recipe would process any number of input arc lamp frames, no matter at what angles of the retarder plate, and would produce an equal number of independent wavelength calibrations.

the current directory. The only exception is represented by the standard calibration tables (e.g., line catalogs), which are here assumed to be located in `/cal/fors1`.[13]

In order to process the calibration exposures available for the scientific observation, the recipe *fors_pmos_calib* is used (see Section 9.13, page 131). The input SOF may be defined as follows:

File: *cal.sof*

```
master_bias.fits                                MASTER_BIAS
FORS1.2008-11-16T09:06:08.980.fits              SCREEN_FLAT_PMOS
FORS1.2008-11-16T09:07:08.086.fits              SCREEN_FLAT_PMOS
FORS1.2008-11-16T09:08:07.071.fits              SCREEN_FLAT_PMOS
FORS1.2008-11-16T09:09:06.077.fits              SCREEN_FLAT_PMOS
FORS1.2008-11-16T09:10:05.114.fits              SCREEN_FLAT_PMOS
FORS1.2008-11-16T09:12:02.965.fits              LAMP_PMOS
FORS1.2008-11-16T09:19:43.041.fits              LAMP_PMOS
FORS1.2008-11-16T09:23:31.894.fits              LAMP_PMOS
FORS1.2008-11-16T09:27:05.935.fits              LAMP_PMOS
FORS1.2008-11-16T09:30:38.446.fits              LAMP_PMOS
FORS1.2008-11-16T09:34:24.578.fits              LAMP_PMOS
FORS1.2008-11-16T09:38:26.063.fits              LAMP_PMOS
FORS1.2008-11-16T09:42:43.218.fits              LAMP_PMOS
/cal/fors1/FORS1_ACAT_600B_12_free_00.fits      MASTER_LINECAT
/cal/fors1/FORS1_B_GRS_600B_12_free_00.fits     GRISM_TABLE
/cal/fors1/FORS1_B_DIST_600B_12_free_00_2.fits MASTER_DISTORTION_TABLE
```

The MASTER_LINECAT, the GRISM_TABLE, and the MASTER_DISTORTION_TABLE, are static calibration tables which are available in the calibration directories delivered with the pipeline recipes. The file FORS1_ACAT_600B_12_free_00.fits is the default catalog of reference arc lamp lines for grism 600B of the FORS1 instrument. This catalog may be replaced with alternative ones provided by the user, if found appropriate.

The FORS1_B_GRS_600B_12_free_00.fits table contains the default *fors_pmos_calib* recipe configuration parameters for grism 600B 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 FORS1_B_DIST_600B_12_free_00_2.fits table contains the coefficients of an empirical distortion model[14]

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

```
esorex fors_pmos_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_norm_flat_pmos.fits: normalised flat field image.

[13] The user decides where the calibration tables should be copied at installation time.

[14] This is computed by the *fors_calib* recipe, see Section 4.3, page 31.

slit_location_pmos.fits: slit positions on the CCD.

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

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

Products for checking the quality of the result are:

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

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

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

disp_residuals_pmos.fits: residuals for each wavelength calibration fit, produced only if the recipe configuration --check is set.

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

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

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

wavelength_map_pmos.fits: map of wavelengths on the CCD.

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

slit_map_pmos.fits: map of the grism central wavelength, produced only if the recipe configuration --check is set.

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

mapped_screen_flat_pmos.fits: flat field with distortion removed.

mapped_norm_flat_pmos.fits: normalised flat field with distortion removed.

Now the scientific frames (one for each observed angle) can be processed with the recipe *fors_pmos_science* (see Section 9.14, page 136). The following set-of-frames file may be created:

File: *sci.sof*

```
FORS1.2008-11-16T04:53:24.340.fits          SCIENCE_PMOS
FORS1.2008-11-16T04:55:10.841.fits          SCIENCE_PMOS
FORS1.2008-11-16T04:56:56.851.fits          SCIENCE_PMOS
FORS1.2008-11-16T04:58:42.861.fits          SCIENCE_PMOS
FORS1.2008-11-16T05:00:28.932.fits          SCIENCE_PMOS
FORS1.2008-11-16T05:02:14.862.fits          SCIENCE_PMOS
FORS1.2008-11-16T05:03:59.863.fits          SCIENCE_PMOS
FORS1.2008-11-16T05:05:44.844.fits          SCIENCE_PMOS
master_bias.fits                            MASTER_BIAS
curv_coeff_pmos.fits                        CURV_COEFF_PMOS
```

```
master_norm_flat_pmos.fits                    MASTER_NORM_FLAT_PMOS
slit_location_pmos.fits                       SLIT_LOCATION_PMOS
disp_coeff_pmos.fits                          DISP_COEFF_PMOS
/cal/fors1/FORS1_B_GRS_600B_12_free_00.fits GRISM_TABLE
/cal/fors1/eps_theta_2000-01-25.fits          RETARDER_WAVEPLATE_CHROMATISM
```

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_pmos_science sci.sof
```

the following products are created on disk (under the indicated conditions):

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

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

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

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

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

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

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

disp_coeff_sci_pmos.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.

reduced_sci_pmos.fits: image with extracted objects spectra.

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

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

reduced_q_sci_pmos.fits: image with normalised Q Stokes parameter, only produced in case of linear polarimetry observations.

reduced_u_sci_pmos.fits: image with normalised U Stokes parameter, only produced in case of linear polarimetry observations.

reduced_v_sci_pmos.fits: image with normalised V Stokes parameter, only produced in case of circular polarimetry observations.

reduced_l_sci_pmos.fits: image with linear polarization as a function of wavelength, only produced in case of linear polarimetry observations.

reduced_angle_sci_pmos.fits: image with angle of linear polarization (degrees) as a function of wavelength, only produced in case of linear polarimetry observations.

reduced_error_q_sci_pmos.fits: error on computed Q.

reduced_error_u_sci_pmos.fits: error on computed U.

reduced_error_v_sci_pmos.fits: error on computed V.

reduced_error_l_sci_pmos.fits: error on computed linear polarisation.

reduced_error_angle_sci_pmos.fits: error on computed linear polarisation angle.

reduced_nul_q_sci_pmos.fits: Q null parameter, only computed if at least 8 scientific frames are specified in input.

reduced_nul_u_sci_pmos.fits: U null parameter, only computed if at least 8 scientific frames are specified in input.

reduced_nul_v_sci_pmos.fits: V null parameter, only computed if at least 4 scientific frames are specified in input.

Standard star exposures (STANDARD_PMOS) are reduced in the same way.

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.

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 catalog (adding more lines, if available). See Appendix A for more details.

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").

Imaging polarimetry (IPOL) is 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* [20], but not by *Esorex* [8]. 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 catalogs, 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.

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 25).

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 55.

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`	`DET CHIP1 ID`	Chip identifier
`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

- **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:	Association keywords:	Note:
`DPR CATG = CALIB`	`INS COLL NAME`	Collimator used
`DPR TYPE = FLAT,LAMP`	`INS GRIS1 NAME`	Grism used
`DPR TECH = MXU`	`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:	Association keywords:	Note:
`DPR CATG = CALIB`	`INS COLL NAME`	Collimator used
`DPR TYPE = WAVE,LAMP`	`INS GRIS1 NAME`	Grism used
`DPR TECH = MXU`	`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:	Association keywords:	Note:
`DPR CATG = SCIENCE`	`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

6.6 PMOS frames

The PMOS mode performs multi-object spectroscopic polarimetry using a set of movable slitlets.

- **Polarized screen flat field:**

  DO category: SCREEN_FLAT_PMOS
  Processed by: fors_pmos_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 = POLARIMETRY	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

- **Polarized arc lamp spectrum:**

  DO category: LAMP_PMOS
  Processed by: fors_pmos_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 = POLARIMETRY	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

- **Polarimetric standard star spectrum:**

  DO category: STANDARD_PMOS
  Processed by: fors_pmos_science, fors_pmos_extract

Classification keywords:	Association keywords:	Note:
DPR CATG = CALIB	INS COLL NAME	Collimator used
DPR TYPE = STD	INS GRIS1 NAME	Grism used
DPR TECH = POLARIMETRY	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_PMOS
  Processed by: fors_pmos_science, fors_pmos_extract

Classification keywords:	Association keywords:	Note:
DPR CATG = SCIENCE	INS COLL NAME	Collimator used
DPR TECH = POLARIMETRY	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

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 25).

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:	Photometric night zeropoint
dzpoint:	Error on photometric night zeropoint

To clarify the terminology used here: more than one filter may correspond to each of the standard bands, (U,B,V,R,I). In the FORS1/2 instruments the following filters are or have been available:

Filter	*Band*
U_BESS	U
U_SPEC	U
u_HIGH	U
B_BESS	B
b_HIGH	B
g_HIGH	V
V_BESS	V
v_HIGH	V
R_BESS	R
R_SPEC	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 catalog 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
R	V-R
I	V-R

7.2 Photometric standard stars catalog

DO category: `FLX_STD_IMG`

This table is a list of photometric standard stars parameters. Currently two catalogs are included in the static calibrations supplied with the pipeline: selected UBVRI photometric standard stars from Landolt [36], and all entries from the Stetson's photometric standard stars catalog [39] [40], respectively in the files landolt_std_UBVRI.tfits and stetson_std_BVRI.fits

The stetson_std_BVRI.fits table includes the following columns:

OBJECT:	Name of standard star
RA:	Right Ascension (degrees)
DEC:	Declination (degrees)
VAR:	Estimated intrinsic variability in magnitude RMS (mag)
B:	Magnitude (mag)
ERR_B:	Magnitude error (mag)
NP_B:	Number of observations obtained on photometric nights
N_B:	Total number of observations

where B can also be V, R, and I.

The landolt_std_UBVRI.tfits table includes the following columns:

OBJECT:	Name of standard star
RA:	Right Ascension (degrees)
DEC:	Declination (degrees)
EPOCH:	Epoch
RA_S:	Right Ascension in HH MM SS.SSS format

DEC_S:	Declination in DD MM SS.SSS format
N:	Number of observations
M:	Number of nights
U:	U magnitude (mag)
B:	B magnitude (mag)
V:	V magnitude (mag)
R:	R magnitude (mag)
I:	I magnitude (mag)
B_V:	B-V color (mag)
U_B:	U-B color (mag)
V_R:	V-R color (mag)
R_I:	R-I color (mag)
V_I:	V-I color (mag)
ERR_V:	Error on V magnitude (mag)
ERR_B_V:	Error on B-V color (mag)
ERR_U_B:	Error on U-B color (mag)
ERR_V_R:	Error on V-R color (mag)
ERR_R_I:	Error on R-I color (mag)
ERR_V_I:	Error on V-I color (mag)

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 indexes 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 using directly a color index. In any case, the catalog 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 catalog is specified in input, either Landolt's or Stetson's, or even both. If both catalogs are specified, they are merged before being used. In the merged catalog, if two stars are within 5 arcseconds they are considered identical and the one with the largest magnitude error is removed.[15]

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

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

as in Fukugita (1996) [37][16]. 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 is performed according to which uncorrelated errors are available. In order to do this, the generic form of the correction

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

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

[16]Admittedly, this is an unnecessary complication: treating the g Sloan filter in the context of the standard V band would have had the pipeline to compute a Γ around 0.56 and a zeropoint off by about 0.12 mag – reintroducing implicitely Fukugita's relation. For the moment, however, in the following treatment a reference to some "G" band defined as above will still be made. This just means that the pipeline computes the deviations from Fukugita's relation for the FORS g_HIGH filter with respect to an ideal G band.

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

$$\begin{aligned}
U^\star &= V + (B-V) + (U-B) - \Gamma \cdot (U-B) \\
B^\star &= V + (B-V) - \Gamma \cdot (B-V) \\
G^\star &= V + (0.56-\Gamma) \cdot (B-V) - 0.12 \ \ \text{(Fukugita, 1996)} \\
V^\star &= V - \Gamma \cdot (B-V) \\
R^\star &= V - (V-R) - \Gamma \cdot (V-R) \\
I^\star &= V - (V-I) - \Gamma \cdot (V-R)
\end{aligned}$$

which leads to the variances

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

Similarly, in the Stetson catalog case it will be:

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

which leads to the variances

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

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

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
--resp_use_flat_sed	whether to use the flat sed normalisation for the response
--resp_fit_degree	degree to use in the polynomial fit of the response
--resp_fit_nknots	number of nknots to use in the spline fit of the response
--shift_y	offset along the y-axis introduced by the grism, in mm, *optional*

A complete description of these parameters is given in sections 9.9.3, page 108 and 9.10, page 116.

7.4 Arc lamp lines catalog

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 with name "WLEN" listing such wavelengths and another column with name "CHEMICAL_ION" which contains the name of the chemical element (like Ar, Ne, etc...)

A standard line catalog 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 catalog 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 catalog for FORS2 grism 300V and filter GG435 is named

FORS2_ACAT_300V_20_GG435_81.fits

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

FORS1_ACAT_600B_12_free_00.fits

In practice, however, the correct (standard) line catalog can be associated to a given arc lamp frame using the FITS keywords ESO INS GRIS1 ID, ESO INS GRIS1 NAME and ESO INS OPTI7 NAME, found both in the line catalog and in the raw input frames headers.

The `MASTER_LINECAT` files delivered with the pipeline have their wavelengths referenced to the air.

7.5 Detector illuminated region table

DO category: `DETECTOR_ILLUMINATED_REGION`

This table specifies which regions of the detector are in fact illuminated when light enters the telescope pupil. For different combination of detectors and collimators this can range from the whole detector to a subset of the detector. Note that the table specifies a rectangular region, which encloses the real illuminated region, which might be smaller and irregular.

The table contains the following columns

det_chip1_id:	Detector identification
ins_coll_name:	The name of the collimator
illuminated_region_llx:	X coordinate of lower left corner of illuminated rectangle
illuminated_region_lly:	Y coordinate of lower left corner of illuminated rectangle
illuminated_region_urx:	X coordinate of upper right corner of illuminated rectangle
illuminated_region_ury:	Y coordinate of upper right corner of illuminated rectangle

The pipeline matches the chip identification and collimator name of the science observation with the corresponding entry in the table.

Note that the coordinates of the pixels start counting at one and that they refer to unbinned pixels. If binning has been used, the pipeline makes the appropiate conversions.

7.6 Master distortion table

DO category: `MASTER_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 98). This table is used for enabling the on-line processing of scientific data with the recipes *fors_extract* and *fors_pmos_extract*, when appropriate (day) calibrations are not yet available. It is also used by the recipe *fors_pmos_calib*, in order to guarantee the correct identification of the matching ordinary and extraordinary beam pairs in the calibration frames.[17]

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.

The global distortion table is produced by the *fors_calib* recipe with the tag GLOBAL_DISTORTION_TABLE (see Section 9.9, page 97). This tag is changed into MASTER_DISTORTION_TABLE when a high quality

[17] Point pattern matching is generally inapplicable to slits identification in this case, because a polarimetric observation is typically carried out with all slitlets at the same offset.

global distortion table is selected and inserted in the static calibrations directory, where these files are also named according to the following convention:

<instrument name>_DIST_<grism name>_<grism ID>_<*some irrelevant information*>_<chip index>.fits

A global distortion table doesn't depend on the filter in use: only the grism and the chip matter (this is why in the template above the filter related part is marked as "irrelevant" - the reason why it's still there is purely historical). For instance, the (master) global distortion table for the FORS2 grism 300I+21 and related to chip 2 is named

FORS2_DIST_300I_21_OG590_32_2.fits

(the part "OG590_32_" is irrelevant).

In practice, the correct global distortion table can be associated to a given scientific frame using the FITS keywords ESO INS GRIS1 NAME and ESO INS CHIP1 ID, found both in the table and in the raw input frames headers.

7.7 Global distortion table

The global distortion table can used to correct the trace distortion for LSS and PMOS data. The format of the table is exactly like the one described for MASTER_DISTORTION_TABLE. The table can be created by the user with recipe *fors_calib* using a MXU mask with well distributed slits across the full field of view.

The global distortion table is used by the *fors_science* recipe when LSS data is input to correct the spatial distortion. No other information is used from the global distortion table.

7.8 Atmospheric extinction table

Currently the atmospheric extinction table valid for Paranal, as extended by Moehler et al. (2014, A&A 568, A9), is made available in the calibration directory, in a file named `extinct_table.fits`.

This table includes the following columns:

wave: Wavelength (Å)
extinction: Atmospheric extinction (mag/airmass)

7.9 Standard star flux table

A set of standard star flux tables, corresponding to the 40 spectro-photometric standard stars which are included in the FORS calibration plan [11], is available in the calibration directory.

A standard star flux table includes the following columns:

WAVE:	Wavelength (Å)
FLUX:	Flux (10^{-16} erg cm^{-2} s^{-1} Å^{-1})
BIN:	Bin width (Å)
STLLR_ABSORP:	Flag to specify if a bin is affected by stellar absorption

In some tables the bin width is not available, and it is set to zero.

The names of the available standard star flux tables, and the name of the standard stars as reported in the FITS header keyword `ESO OBS TARG NAME`, are listed in Table 7.1.

7.10 Telluric contamination table

The file named `fors2_telluric_regions.fits` contains a list of wavelength intervals that might be affected by atmospheric telluric contamination. Since each grism might be affected in a different way, there is a column per grism with the proper flag.

In order to use this information, the *fors_science* recipe has to have this table as an input and parameter *resp_ignore_mode* must include the string *telluric*.

7.11 Spectro-polarimetric standard star table

A table named `pol_sta.fits`, listing the measured linear polarisation from a number of standard stars, is available in the calibration directory. The listed values are read from [35] (giving preference to the PMOS measurements with respect to the IPOL ones).

A spectro-polarimetric standard star table includes the following columns:

name:	Name of standard star
RA:	Right Ascension (degrees)
DEC:	Declination (degrees)
polarised:	1 if the star is polarised, 0 if not
Bp:	Total linear polarisation in band B (%)
Bdp:	Error on total linear polarisation in band B (%)
Ba:	Linear polarisation angle in band B (degrees)
Bda:	Error on linear polarisation angle in band B (degrees)

The band B can be any of the U, B, V, R, I standard photometric bands.

7.12 Earth Observation Parameter table

A table named `fors_eop_param.fits` is available in the calibration directory. This table contains the Earth Orientation Parameter as a function of time (`MJD-OBS`). An updated version of this table may be downloaded by using the recipe *fors_eop*.

File name	Target name	Catalog
bd25d4655.tfits	BD+25d4655	Oke (1990)
bd28d4211.tfits	BD+28d4211	Oke (1990)
bd33d2642.tfits	BD+33d2642	Oke (1990)
bpm16274.fits	BPM16274	Hamuy et al. (1992, 1994)
cd32d9927.tfits	CD-32-9927	Hamuy et al. (1992, 1994)
eg21.tfits	EG-21	Hamuy et al. (1992, 1994)
eg274.tfits	EG-274	Hamuy et al. (1992, 1994)
feige110.tfits	Feige-110	Hamuy et al. (1992, 1994)
feige56.tfits	Feige-56	Hamuy et al. (1992, 1994)
feige66.tfits	Feige-66	Oke (1990)
feige67.tfits	Feige-67	Oke (1990)
g138_31.fits	G138-31	Oke (1990)
g158_100.tfits	G-158-100	Oke (1990)
g60_54.fits	G60-54	Oke (1990)
g93_48.tfits	G-93-48	Oke (unpublished)
gd108.tfits	GD-108	Oke (1990)
gd248.fits	GD248	Oke (1990)
gd50.tfits	GD-50	Oke (1990)
gd71.fits	GD71	Bohlin et al. (1995)
hd49798.fits	HD49798	Bohlin et al. (1995)
hilt600.tfits	Hiltner-600	Hamuy et al. (1992, 1994)
hz2.tfits	Hz-2	Oke (unpublished)
hz21.fits	HZ21	Bohlin et al. (1995)
hz4.fits	HZ4	Bohlin et al. (1995)
hz44.tfits	Hz-44	Oke (1990)
lb227.fits	LB227	Bohlin et al. (1995)
lds749b.tfits	LDS-749b	Oke (1990)
ltt1020.tfits	LTT-1020	Hamuy et al. (1992, 1994)
ltt1788.tfits	LTT-1788	Hamuy et al. (1992, 1994)
ltt2415.tfits	LTT-2415	Hamuy et al. (1992, 1994)
ltt3218.fits	LTT3218	Hamuy et al. (1992, 1994)
ltt377.tfits	LTT-377	Hamuy et al. (1992, 1994)
ltt3864.tfits	LTT-3864	Hamuy et al. (1992, 1994)
ltt4816.tfits	LTT-4816	Hamuy et al. (1992, 1994)
ltt6248.tfits	LTT-6248	Hamuy et al. (1992, 1994)
ltt7379.tfits	LTT-7379	Hamuy et al. (1992, 1994)
ltt7987.tfits	LTT-7987	Hamuy et al. (1992, 1994)
ltt9239.tfits	LTT-9239	Hamuy et al. (1992, 1994)
ltt9491.tfits	LTT-9491	Hamuy et al. (1992, 1994)
ngc7293.tfits	NGC-7293	Oke (1990)

Table 7.1: *Spectro-photometric standard stars in the FORS Calibration Plan. Full references are: Oke, 1990, AJ 99, 1621; Hamuy et al., 1992, PASP 104, 533; Hamuy et al., 1994, PASP 106, 566; Bohlin et al., 1995, AJ 110, 1316. See also* `http://www.eso.org/sci/observing/tools/standards/spectra/`

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 readout 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. See Section 10.1.3, page 155, for more about the way errors are propagated.

A more detailed description of the applied algorithms is given in Section 10, page 152. 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 and spectropolarimetric 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 catalogs (see Section 7.2, page 56), and a photometric table carrying appropriate atmospheric absorbtion coefficients, color terms and expected zeropoints appropriate for the instrument setting (see Section 7.1, page 55). The reduced exposure, the list of detected and identified sources, a background map, 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. 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.

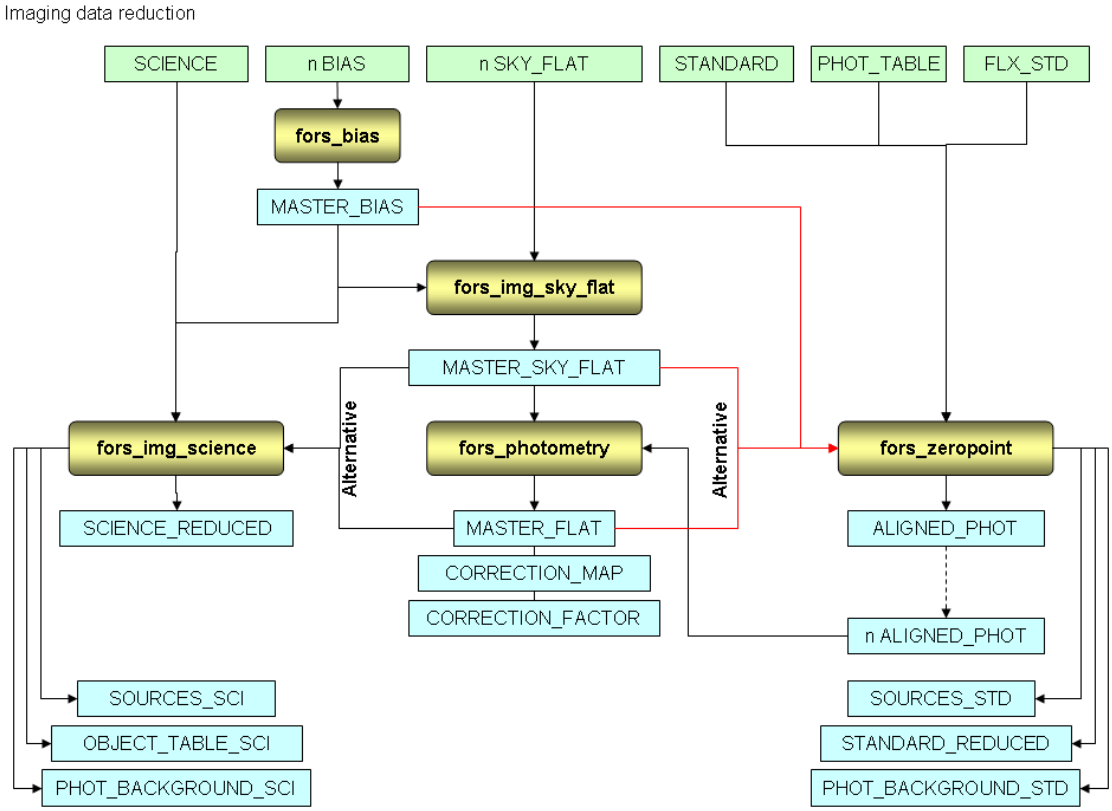

Figure 8.1: *This is the data reduction flow using the recipes for imaging data reduction. The rounded yellow boxes represent recipes handling input and intermediate products. The boxes indicating input and product data are labeled with their DO categories, stripped of their suffix _IMG (when present). 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.

The *fors_calib* and the *fors_science* recipes basic functionality is accessible through 17 more low-level recipes: see Section 8.2.2, page 70 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, and,
- one arc lamp reference lines catalog.

where it is expected that flat and arc lamp exposures have been taken quasi-simultaneously, i.e., without changes in the instrument mechanical configuration (moving masks, changing flexures, etc.), to guarantee that they are affected by the same instrument distortions.

Additionally, the following ancillary data is needed:

- a grism table
- a table with arc lamp reference wavelengths.

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 (i.e., rebinned at constant wavelength step),
- location of slits on CCD, and on extracted arc lamp spectra image,
- master flat field,
- normalised master flat field, and,
- spectral resolution table.
- the flat field spectral energy distribution for each of the slits.

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.
- a response curve (only if flux calibration is needed)
- the flat spectral energy distribution

All inputs but the first one and the response curve are typically produced by *fors_calib*. The response curve is created by calling *fors_science* with a standard star.

Additionally, the following ancillary data might be an input:

- a global distortion table (only LSS)
- a standard star reference flux (only if standard star)
- a extinction table (only if standard star)
- a telluric contamination table (only if standard star)

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:

- 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,
- error spectra of extracted objects
- sky spectra corresponding to extracted objects, and,
- flux calibrated data.

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 pages 70, 71, and 72. 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.

This recipes are experimental and therefore unsupported.

The low-level recipes are the following:

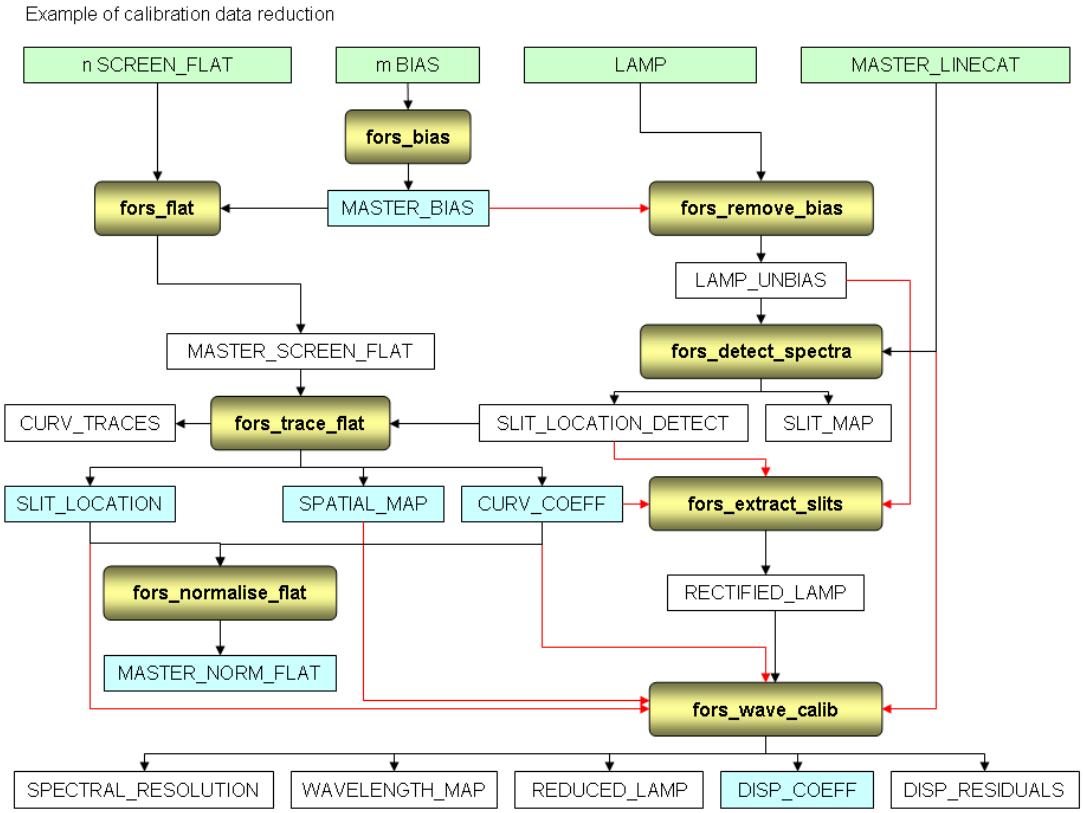

Figure 8.2: *This is a possible data reduction flow applied by recipe fors_calib on MOS/MXU data. The rounded yellow boxes represent low-level recipes handling input and intermediate products. The boxes indicating input and product data are labeled with their DO categories, stripped of their variable 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_trace_flat, to determine the spatial curvature model.

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

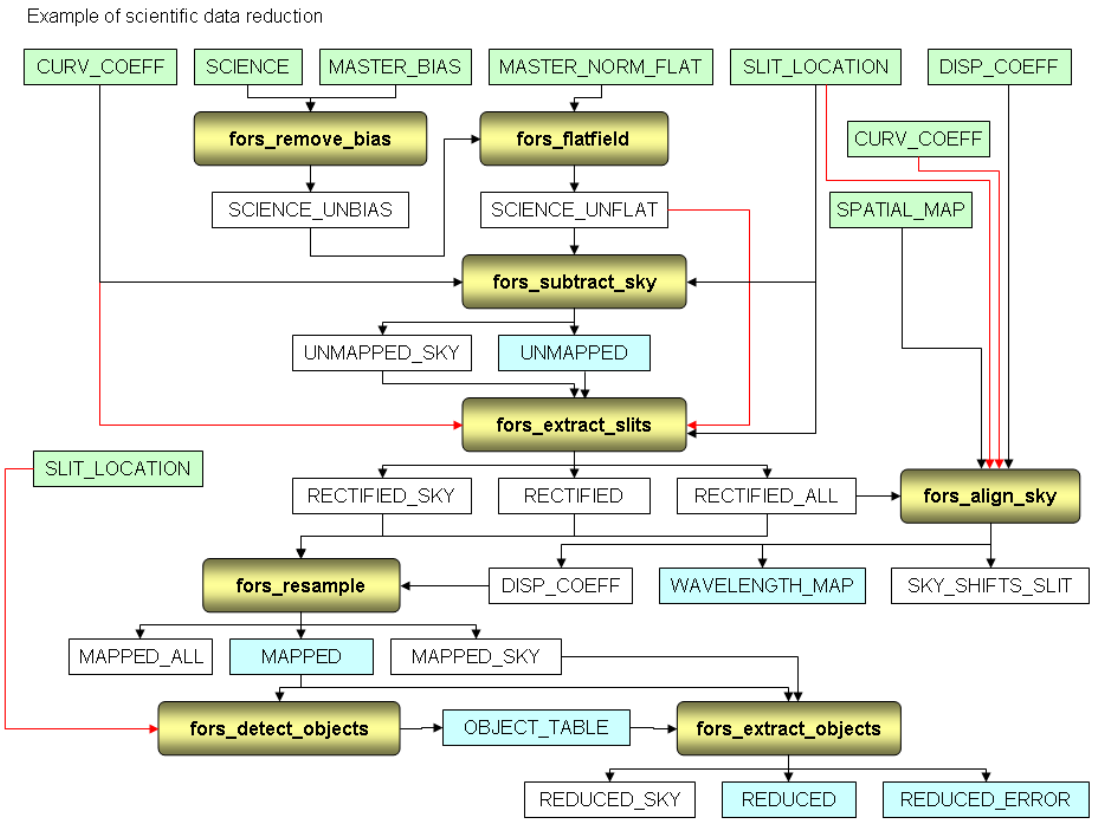

Figure 8.3: *This is a possible data reduction flow applied by recipe fors_science on MOS/MXU data. The rounded yellow boxes represent low-level recipes handling input and intermediate products. The boxes indicating input and product data are labeled with their DO categories, stripped of their variable 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_detect_objects, to detect objects in slit spectra.

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_spec_mflat, to compute the master flat from flat field frames.

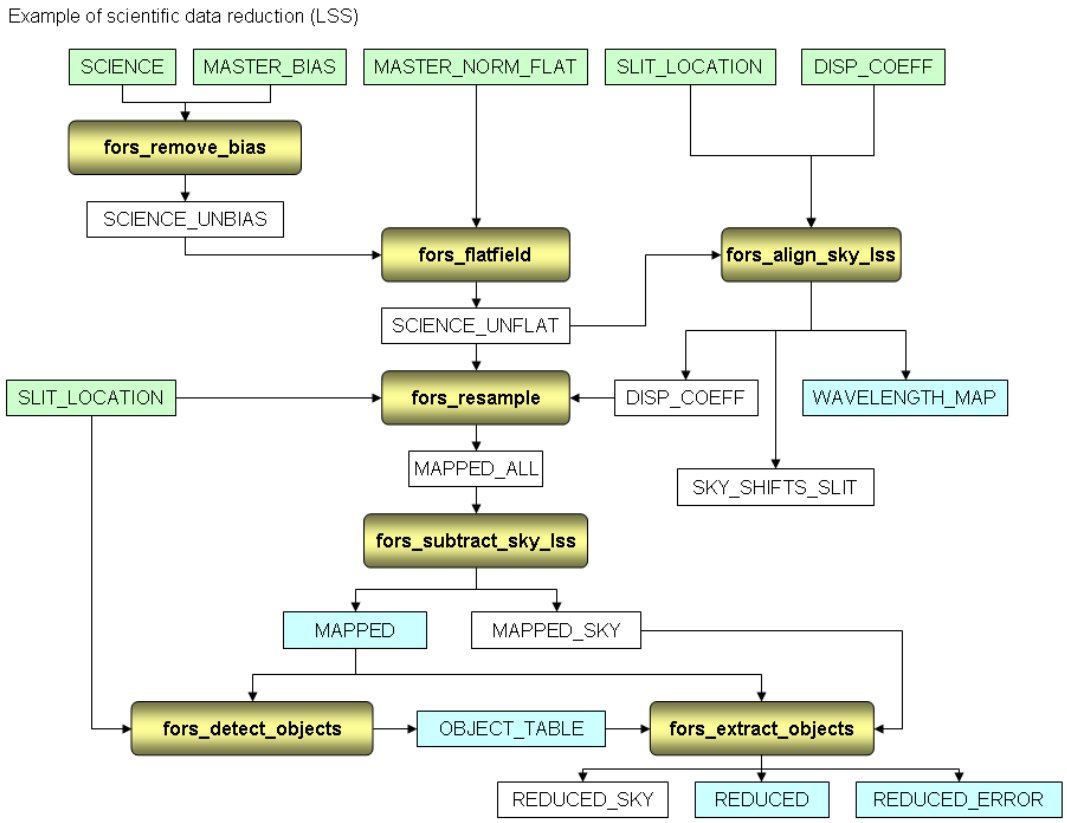

Figure 8.4: *This is a possible data reduction flow applied by recipe fors_science on LSS or LSS-like data. The rounded yellow boxes represent low-level recipes handling input and intermediate products. The boxes indicating input and product data are labeled with their DO categories, stripped of their 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.*

fors_normalise_flat, to normalise a flat.

fors_subtract_sky_lss, to subtract sky from calibrated long slit exposure.

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 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.

8.3 Spectro-polarimetric data reduction overview

The FORS spectro-polarimetric pipeline is based on a set of 4 stand-alone recipes. Only 3 of them are involved in the off-line data reduction cascade: the remaining recipe is 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 and the spectroscopic pipeline.

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

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

The data reduction recipes can handle spectro-polarimetric data both from FORS1 and FORS2.

8.3.1 Required input data

The input data to the recipe *fors_pmos_calib* are:

- a sequence of raw flat exposures,
- a sequence of raw arc lamp exposures,
- one master bias frame, or a sequence of raw bias exposures,
- one arc lamp reference lines catalog, and,
- one master distortion table.

where it is expected that flat and arc lamp exposures have been taken quasi-simultaneously, i.e., without changes in the instrument mechanical configuration (moving masks, changing flexures, etc.), to guarantee that they are affected by the same instrument distortions.

The products of recipe *fors_pmos_calib* are:

- wavelength calibration local models,

- wavelength calibration residuals,
- spatial curvature local models,
- residuals of flat field spectra tracing,
- maps 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.

At least the following input data to the recipe *fors_pmos_science* are required:

- a set of scientific (or standard star) exposures, at different angles of the polariser. Currently, the science recipe supports only one science image per polarisation angle, since no combination of exposures with the same angle is applied. The supported set of angles are:
  - For circular polarimetry:
    - 2 angles: -45.0, 45.0
    - 4 angles: -45.0, 45.0, 135.0, 225.0
  - For linear polarimetry:
    - 4 angles: 0.0, 22.5, 45.0, 67.5,
    - 8 angles: 0.0, 22.5, 45.0, 67.5, 90.0, 112.5, 135.0, 157.5
    - 16 angles:0.0, 22.5, 45.0, 67.5, 90.0, 112.5, 135.0, 157.5, 180.0, 202.5, 225.0, 247.5, 270.0, 292.5, 315.0, 337.5
- 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_pmos_calib*.

The products of recipe *fors_pmos_science* depend on the nature of the input, and on what is requested: for instance, observations related to linear polarisation will not generate the same output of observations related to circular polarisation. Some products, such as the null Stokes parameters, are only created when the number of input scientific frames is sufficient. In general the following products are created:

- upgraded wavelength calibration local models,
- upgraded map of wavelengths for each CCD pixel,
- upgraded slit spectra 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,
- error spectra of extracted objects,
- Normalized Stokes parameters (U/I, Q/I, V/I), absolute linear polarisation and polarisation angle (depending on input),
- null Stokes parameters (if possible), and,
- error on extracted polarimetric signal.

9 Pipeline Recipe Interfaces

In this Section a detailed description of the FORS pipeline recipe interfaces is given, with a complete specification of the recipe usage, input, output, and configuration parameters. For an overview of the available pipeline recipes, please see Section 8, page 65.

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, If the over/prescan regions are present, the overscan correction is applied and the regions trimmed from the result.

9.1.1 Input files

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

9.1.2 Output files

MASTER_BIAS: Master bias calibration frame.

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

9.1.3 Configuration parameters

The following configuration parameters determine how the *fors_bias* recipe will process the input frames.

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

This parameter defines the way the frames will be stacked.

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:

--ksigma: Lower and upper rejection threshold in units of sigma. *Default*: "-3.0,3.0"

This parameter is a string containing two comma-separated values. For each pixel position, a robust determination of the standard deviation from the *median* pixel value is made. All pixel values with a negative or positive residual greater than the specified numbers 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.

9.1.4 Quality control parameters

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

QC DET CHIP NUM: Chip number. *Units*: *none*

QC INS_COLL_NAME: Collimator name. *Units*: *none*

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 ratio 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).

QC PATTERN: RMS of master bias averaged along y-axis.

QC PATTERN BIN: Bin size needed to have flux of 1D master bias in 256 PIX bins.

9.2 fors_dark

The FORS pipeline recipe *fors_dark* is used to create a master dark 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 1 second. The overscan regions, if present, are trimmed from the result.

Master dark 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: Master dark calibration frame.

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

9.2.3 Configuration parameters

The following parameters determine how 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 76).

9.3 fors_eop

This recipe downloads the latest version of the Earth Orientation Parameter and DUT[18] (Difference to Universal Time: UT1-UTC) from IERS. The IERS Earth Orientation Parameters (EOP) describe the irregularities of the earth's rotation. It stores the parameter in a fits-file that is used as input for the computation of the barycentric correction.

9.3.1 Input files

None.

9.3.2 Output files

EOP_PARAM: File containing the Earth Orientation Parameter

9.3.3 Configuration parameters

--eop_host: Host to retrieve the EOP from. *Default*: **https://datacenter.iers.org**

--eop_urlpath: URL path of the EOP file to retrieve. *Default*: **/products/eop/rapid/standard/finals2000A.data**

--eop_usertag: TAG provided by the user. *Default*: **None**

--eop_instrument: Product header keyword value for INSTRUME. *Default*: **FORS2**

9.4 fors_img_screen_flat

The FORS pipeline recipe *fors_img_screen_flat* is used to create a master screen flat field 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 trimmed 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.

9.4.1 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.

[18]The time correction equal to the difference between Universal Time (UT1), which is defined by Earth's rotation, and Coordinated Universal Time (UTC), which is defined by a network of precision atomic clocks.

9.4.2 Output files

MASTER_SCREEN_FLAT_IMG: Master screen flat field calibration frame.

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

9.4.3 Configuration parameters

The following parameters determine how 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 76).

--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 than or equal to 0, then 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 pixel

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

9.4.4 Quality control parameters

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

QC DET CHIP NUM: Chip number. *Units*: *none*

QC INS_COLL_NAME: Collimator name. *Units*: *none*

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

A saturated pixel is defined as a pixel having value either 65535 or 0 ADUs in a raw frame. 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.

QC MASTER MEDIAN: Median value of master screen flat. *Units*: ADU

QC MASTER SIGMA: Sigma of master screen flat. *Units*: ADU

QC RAW MEDIAN: Median value of first raw screen flat. *Units*: ADU

QC RAW SIGMA: Sigma value of first raw screen flat. *Units*: ADU

9.5 fors_img_sky_flat

The FORS pipeline recipe *fors_img_sky_flat* is used to create a master twilight 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)[19]. The overscan regions, if present, are trimmed from the result.

[19]Note that images are normalised to the same illumination level before applying the rejection algoritm.

9.5.1 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.

9.5.2 Output files

MASTER_SKY_FLAT_IMG: Sky flat field master calibration frame.

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

9.5.3 Configuration parameters

The following parameters determine how 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 76).

9.5.4 Quality control parameters

Currently the following QC parameters, used by PSO and DPD, are evaluated by the *fors_img_sky_flat* recipe.

QC DET CHIP NUM: Chip number. *Units*: *none*

QC INS_COLL_NAME: Collimator name. *Units*: *none*

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

QC SKYFLAT FLUX MIN: Median level of dimmest input raw sky flat. *Units*: ADU

QC SKYFLAT FLUX MAX: Median level of brightest input raw sky flat. *Units*: ADU

QC MASTER MEDIAN: Median value of master sky flat. *Units*: ADU

QC MASTER SIGMA: Sigma of master sky flat. *Units*: ADU

QC RAW MEDIAN: Median value of first raw sky flat. *Units*: ADU

QC RAW SIGMA: Sigma value of first raw sky flat. *Units*: ADU

9.6 fors_img_science

The FORS pipeline recipe *fors_img_science* is used to reduce a direct imaging scientific exposure. The master bias calibration is subtracted. The debiased signal is then divided by the normalised sky flat field, and the overscan regions, if present, are trimmed from the result. The calibrated image is finally sent to a source detection and extraction application (SExtractor, see 10.8).[20] As a last step, and if the input files contain a DETECTOR_ILLUMINATED_REGION file (see 7.5), the recipe will trimm the image to the rectangular area of the detector that actually receives light from the sky.

9.6.1 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).

PHOT_COEFF_TABLE: observed extinction coefficients from *fors_photometry*, see Section 9.8.

EXTINCTION_PER_NIGHT: extinction per night, from *fors_photometry*, see Section 9.8.

STATIC_PHOT_COEFF_TABLE: *required for IDP*, filters static coefficients. Static table containing zeropoints, extinction coefficients and offsets between Vega and AB magnitude for all filters. The pipeline should use zeropoints and extinction coefficients for that night. If the results from *fors_photometry* are not available the values in this table are used.

DETECTOR_ILLUMINATED_REGION: *optional*, specifies the regions of the detector that get actually illuminated.

9.6.2 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 [25].

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 [24] and [33] for details.

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

[20] SExtractor is included in the pipeline software package, and is installed automatically.

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*.

SCIENCE_REDUCED_IMG_IDP: IDP compliant reduced image. See [27].

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

SCIENCE_REDUCED_WEIGHT_IDP: Ancillary weight map for the IDP compliant reduced image. See [27].

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

SCIENCE_REDUCED_ERR_IDP: Error on the IDP compliant reduced image. See [27].

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

9.6.3 Configuration parameters

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

--sex_exe: Path to SExtractor executable. *Default*: *installation_path*/lib/fors-*version*/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/esopipes/fors-*version*/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, are possible choices (see the SExtractor User's Guide [24] for details). The default is a simple aperture magnitude, MAG_APER, where the aperture diameter (pix) 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*).

--magsyserr: Systematic error in magnitude added in quadrature to the uncertainty of each instrumental magnitude measurement. *Default*: 0.01 mag

This parameter has been implemented to allow the user to place a reasonable limit on the reported photometric uncertainties by the recipe, especially for the brightest stars/objects in an image. It is known that without performing exquisitely detailed modelling there is a fundamental limit of the order of 1% on the absolute photometric accuracy that may be achieved with high S/N photometric measurements. However, the CCD noise model adopted by the FORS pipeline, which is perfectly valid (including photon and readout noise and similar noise from the calibrations), may report unrealistic uncertainties as precise

as 1 mmag for the brightest objects in an image. By setting this parameter to a positive value, the reported photometric uncertainties will be forced to be at least as great as this value, with the uncertainties reported for the brightest objects in the image being affected most.

--idp_generate: If true IDP-compliant science products are generated. *Default*: false

--idp_weights_threshold: The parameter is used only when IDP products are generated. The value of a pixel in SCIENCE_REDUCED_WEIGHT_IDP is set to 1 if the corresponding position in the MASTER_SKY_FLAT_IMG is greater than the threshold, 0 is set otherwise. *Default*: 0.5

9.6.4 Quality control parameters

Currently the following QC parameters, used by PSO and DPD, 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 a colour correction nor a zeropoint correction 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 a colour correction nor a zeropoint correction 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 terms of flux (ADU/s) and then propagated to instrumental magnitude units. More specifically, if ΔF is the variation of flux, then 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.

9.7 fors_zeropoint

The FORS pipeline recipe *fors_zeropoint* is used to estimate the atmospheric extinction from one imaging exposure on a photometric standard stars field.

Symbol	Definition	Unit
M	Catalog magnitude	mag
C	Catalog color	mag
Γ	Linear color correction term	
$M^\star$	Color corrected catalog magnitude ($M - \Gamma \cdot C$)	mag
g	Detector gain	ADU/e^-
t	Detector exposure time	s
D	Observed instrumental magnitude ($-2.5 \log$ of counts)	mag
m	Corrected instrumental magnitude ($D + 2.5 \log g + 2.5 \log t$)	mag
A	Airmass	airmass
E	Atmospheric extinction coefficient	mag/airmass
Z	Zeropoint	mag

Table 9.1: Photometry related symbols used in this document. An index i may be added to indicate that a quantity refers to a specific star (for instance, M_i is the catalog magnitude of star i), while an index j would refer to a specific frame (for instance, E_j is the atmospheric extinction related to frame j). In the text the instrumental magnitude D is practically never used, and the exposure time and gain = 1 corrected magnitude m is used instead.

The bias master calibration is subtracted from the raw exposure, and the median residual in the overscan regions is subtracted too. The unbiased signal is then divided by the normalised sky flat field produced by the recipe *fors_img_sky_flat*. The calibrated image is sent to a source detection and extraction application (SExtractor 2.5.0, see 10.8).[21] The detected sources are compared to a catalog of standard stars for identification. The recipe *fors_zeropoint* uses whatever catalog is specified in input, either Landolt's or Stetson's, or even both. If both catalogs are specified, they are merged before being used. In the merged catalog, if two stars are within 5 arcseconds they are considered identical and the one with the largest magnitude error is removed.[22] The comparison of observed *vs* catalog positions is made, in order to determine the offset between the image and the catalogue WCS (see Section 10.2.1, page 156 for details). Finally, the atmospheric extinction E is estimated by optimally averaging[23] the extinctions E_i, computed from each detected catalog star using the known values for the instrument zeropoint Z and the color filter correction Γ, applying the following formula:

$$E_i = \frac{Z + m_i - (M_i - \Gamma \cdot C_i)}{A}$$

[21] SExtractor is included in the pipeline software package, and is installed automatically.

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

[23] Described in detail in Sect. 10.2.2, p. 157

(where higher order terms are neglected).[24] In order to directly evaluate both E and Z 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 9.8, page 92).

9.7.1 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 catalog (see Section 7.2, page 56).

PHOT_TABLE: *required* photometric table (see Section 7.1, page 55).

9.7.2 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 [24] and [33] for details.

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

[24] In the actual recipe implementation, the differences

$$Z_i = (M_i - \Gamma \cdot C_i) - (m_i - E_o A)$$

between the color corrected catalog magnitude and the corresponding instrumental magnitude corrected to airmass zero assuming an ideal atmospheric extinction coefficient E_o, are optimally averaged (see Section 10.2.2). The derived quantity is conventionally referred to as the *frame zeropoint* Z_{frame}. This way to proceed is quite misleading, because it conceals the fact that the zeropoint Z is a known constant of the instrument, while the variable term to determine is E, and not really Z. Such implementation is there for historical reasons: traditionally the frame zeropoint Z_{frame} is the Quality Control parameter used in the monitoring of the atmosphere quality. Now this has been adapted to a more physical view (see [41]): there was however no need to modify the old pipeline implementation, as the optimal estimate for the extinction coefficient E is related to the optimal estimate of the frame zeropoint Z by the

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

The assumptions on a photometric night atmospheric extinction E_o and instrument zeropoint Z are listed for each FORS filter in the photometric table (see Section 7.1, page 55).

table produced by the recipe *fors_img_science* (see Section 9.6.2, page 83). 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 a photometric night atmospheric extinction E_o reported in the input PHOT_TABLE). This quantity is computed as

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

dinstr_cmag: Error on corrected instrumental magnitude.
object: Name of identified standard star (from the input catalog).
ra: Right Ascension of identified standard star (from the input catalog).
dec: Declination of identified standard star (from the input catalog).
mag: Color corrected catalog magnitude ($M^\star$).
dmag: Error on color corrected catalog magnitude ($\Delta M^\star$).
cat_mag: Catalog standard star magnitude (M).
dcat_mag: Error on catalog magnitude (ΔM).
color: Color index of standard star (C), chosen as in Section 7.1, page 55.
dcolor: Error on color index (ΔC).
cov_catm_col: Covariance of catalog magnitude and color (needed for error propagation).
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 any star 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

$$Z = M^\star - m + E_o A$$

dzeropoint: Error on star 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 some may happen to be negative: see Sections 10.1.1 and 10.2.2 for a mathematical treatment of this problem).

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

OFFSET_HISTOGRAM: Histogram of candidate pixel offsets between the image and catalogue WCS.

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

9.7.3 Configuration parameters

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

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

--sex_config: Path to SExtractor configuration file. *Default*: *installation_path*/share/esopipes/fors-*version*/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 9.6.3, page 85. Further parameters of recipe *fors_zeropoint* are:

--maxoffset: Maximum acceptable offset between the image and catalogue WCS (pixels). *Default*: 150.0 pixel

--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.

--magsyserr: Systematic error in magnitude added in quadrature to the uncertainty of each instrumental magnitude measurement. *Default*: 0.01 mag

This parameter has been implemented to allow the user to place a reasonable limit on the reported photometric uncertainties by the recipe, especially for the brightest stars/objects in an image. It is known that without performing exquisitely detailed modelling there is a fundamental limit of the order of 1% on the absolute photometric accuracy that may be achieved with high S/N photometric measurements. However, the CCD noise model adopted by the FORS pipeline, which is perfectly valid (including photon and readout noise and similar noise from the calibrations), may report unrealistic uncertainties as precise as 1 mmag for the brightest objects in an image. By setting this parameter to a positive value, the reported photometric uncertainties will be forced to be at least as great as this value, with the uncertainties reported for the brightest objects in the image being affected most.

9.7.4 Quality control parameters

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

QC ZPOINT: Frame zeropoint (Z_{frame}). *Units*: mag

See subsections of 9.7 above, and Section 10.2 on page 156.

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

This is the expected random statistical error propagated from both observed and catalog quantities (see Section 10.2, page 156).

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_{frame} = QC ZPOINT and the nominal expected zeropoint Z given as input to the recipe, that is $E = E_o + (Z - Z_{frame})/A$, where A is the airmass, and E_o is the atmospheric extinction coefficient that was assumed for the computation of Z_{frame} (see begin of this Section, page 88).

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

Error propagation on the computation of QC EXTCOEFF.

9.8 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 [41].

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

$$m_{ij} = M_i - \Gamma \cdot C_i + p(A_j, C_i) + E_j A_j - Z + f(x_{ij}, y_{ij})$$

where the symbols are consistent with table 9.1 (page 88), 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.[25] 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.

Substituting $M_i - \Gamma \cdot C_i = M_i^\star$ the general equation becomes

$$m_{ij} = M_i^\star + p(A_j, C_i) + E_j A_j - Z + f(x_{ij}, y_{ij})$$

The *fors_photometry* recipe can solve this system of equations by leaving fixed or fitting any of the parameters detailed below:

$M_i^\star$ 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 catalog values (i.e., $M_i^\star$ 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 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., if two or more $M_i^\star$ are 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 The instrument zeropoint can be determined, if it is not attempted to fit at the same time an atmospheric extinction for each available frame.

E_j The atmospheric extinction coefficient can be determined for each frame if the instrument zeropoint Z is fixed. It is also possible to fit a single E valid for all frames, or one E for each group of frames belonging to the same night: in such cases, Z can be a free parameter.

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 [41] where $M_o^\star = Z = 0$ is set. Note also that some combinations (such as fitting E_j for each input ALIGNED_PHOT table along with Z) do not make sense and are not allowed. The errors of g_j and t_j, used in

[25] 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.

the computation of m_{ij} (see table 9.1, page 88), are not propagated because they are usually insignificant in the overall error budget, and not readily available.

The method described in [41] Appendix A corresponds to the following configuration:

$M_i^\star$	free
Γ	fixed
Z	fixed
E_j	all free, but same value for same nights
f	non-zero order, d
p	zero order (no free parameters)

The method described in [41] Appendix B corresponds to

$M_i^\star$	fixed
Γ	free
Z	free
E	one free parameter, valid for all frames
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 160.

The *fors_photometry* implementation identifies common stars in different frames by their celestial coordinates (RA, Dec). Therefore both standard and non-standard stars can be used in the model fitting. However, if fitting f is not requested, only standard stars will be actually processed by the recipe.

9.8.1 Input files

PHOT_TABLE: *required* photometric table, containing atmospheric extinction coefficients for each filter (see Section 7.1, page 55).

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*. In case the fitting of f is requested, it should be the same frame used in *all* the runs of *fors_zeropoint* which produced the input ALIGNED_PHOT tables.

9.8.2 Output files

PHOT_COEFF_TABLE: A one row table with (potentially) the same columns as the PHOT_TABLE (see Section 7.1, 55):

filter:	Name of filter
ext:	Created only if *one* extinction is fitted
dext:	Created only if *one* extinction is fitted
col:	Created only if the color term is fitted
dcol:	Created only if the color term is fitted
zpoint:	Created only if the instrument zeropoint is fitted
dzpoint:	Created only if the instrument zeropoint is fitted

If neither one of E, Γ and Z are fitted, this table is not created.

EXTINCTION_PER_NIGHT: List of atmospheric extinction coefficients per night. This table is only produced if an individual extinction for each night in the input set of frames is fitted, and it has the following structure:

filename:	Name of input PHOT_TABLE
ext:	Atmospheric extinction (mag/airmass)
dext:	Error on atmospheric extinction (mag/airmass)
mjd-night:	Night (unique) identifier (integer MJD)

EXTINCTION_PER_FRAME: List of atmospheric extinction coefficients per input frame. This table is only produced if an individual extinction for each one of the input frames is fitted, and it has the following structure:

filename:	Name of input PHOT_TABLE
ext:	Atmospheric extinction (mag/airmass)
dext:	Error on atmospheric extinction (mag/airmass)
mjd-obs:	MJD of observation

CORRECTION_MAP: Flat field correction map (magnitude), the evaluation of f at each pixel. This is only produced if the correction f is requested.

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. This is only produced if the correction f is requested.

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

MASTER_FLAT_IMG: Corrected flat field. This is only produced if the correction f is requested.

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

9.8.3 Configuration parameters

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

--fitz: Fit the instrument zeropoint (Z). *Default*: true

If set to false, the zeropoint Z read from the input PHOT_TABLE is assumed.

--fit_all_mag: Fit star magnitudes ($M_i^\star$). *Default*: false

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 catalog 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 catalog value (even if USE_CAT is set to 0).

--fite: Fit atmospheric extinction coefficient (E). (**no** = don't fit any, **one** = fit one extinction for all, **pernight** = fit an extinction for each night, **perframe** = fit an extinction for each input frame). *Default*: **pernight**

If set to **no**, the atmospheric extinction E_o read from the input PHOT_TABLE is assumed. The option **one** would fit just one mean atmospheric extinction for all frames. Choosing **pernight** would fit an independent atmospheric extinction for each night found in the input dataset. When choosing **perframe**, it would be appropriate to set *--fitz* to "false", or the fit will likely fail (this holds also in the **pernight** case, if only one frame per night is available). Trying to fit any atmospheric extinction when all input tables include only sources from the same airmass, would also lead to a failure.

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

If set to true, the linear color correction term is fit. If set to false, the linear color correction term read from the input PHOT_TABLE is assumed.

--degreef1: Correction map polynomial degree *Default*: 0

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

Degree and type of the f 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*. *degreef1* = 0 disables the fitting of the f correction, and in this case only standard stars are involved in the fitting procedure.

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

Degree of the p polynomial accounting for non-linear dependencies from color and airmass. *degreep* = 0 disables the fitting of the p correction.

--mjd_obs_target: EXTINCTION_PER_NIGHT and PHOT_COEFF will inherit MJD-OBS from the raw frame having the closes MJD-OBS to the parameter. If the parameter is negative, the first raw frame is used instead. *Default*: -1.0

9.8.4 Quality control parameters

Currently the following QC parameters, used by DPD, are evaluated by the *fors_photometry* recipe. See subsections of 9.8 above, and Section 10.3 on page 160 for more details on how they are computed.

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

This is the *true* instrument zeropoint Z (different from the one computed by recipe *fors_zeropoint*, see Section 9.7, page 88). This quantity is computed only if *-fitz*=true.

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

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

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

This is the (mean) atmospheric extinction, valid for all input frames. This quantity is computed only if *-fite* = one.

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

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

9.9 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.

9.9.1 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 7.3, page 59 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.

MASTER_LINECAT: *required* line catalog. It must contain the reference wavelengths (in Ångstrom) for the arc lamp used. The only requirement for this table is to contain a column with name "WLEN" listing such wavelengths and another column with name "CHEMICAL_ION" which contains the name of the

chemical element (like Ar, Ne, etc...) The recipe will use only those lines for which the chemical element was actually present in the lamp (by looking at keywords ESO INS LAMPi. A standard line catalog 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 stacked into one.

9.9.2 Output files

Not all output frames listed here are always produced. Some are never created in case of LSS or LSS-like data.[26] Here is the list of all the possible output frames, in alphabetical order, together with a list of related configuration parameters.[27] 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.

ARC_RECTIFIED_MXU: The spatial rectified arc to which the positions in DETECTED_LINES_MXU product refer.

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*, *--endwavelength* and *--wdegree*.

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*.

[26] 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.

[27] See Section 9.9.3, page 108, for a complete description of the recipe configuration parameters.

Typical tracing residuals are not greater than 0.3 pixels (see Figure 9.7, page 112).

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[28] 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).[29]

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

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

DELTA_IMAGE_MXU: deviation from the linear term of the wavelength calibration polynomials.

This image is used together with the DISP_RESIDUALS_TABLE_MXU to enable quality control of the obtained solutions (see Figure 9.1).

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

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

DETECTED_LINES_MXU: table containing each individual line that has been detected in the 2D spectra. It also contains the lines that have been identified afterwards for the wavelength calibration. In this case, the assigned wavelength is also present, as well as the residual. If the line has been identified in further iterations of the wavelength calibration, it is also shown.

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.

[28] See entry REDUCED_LAMP_MXU in this Section, or entries MAPPED_SCI_MXU and MAPPED_ALL_SCI_MXU on page 120.

[29] 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.

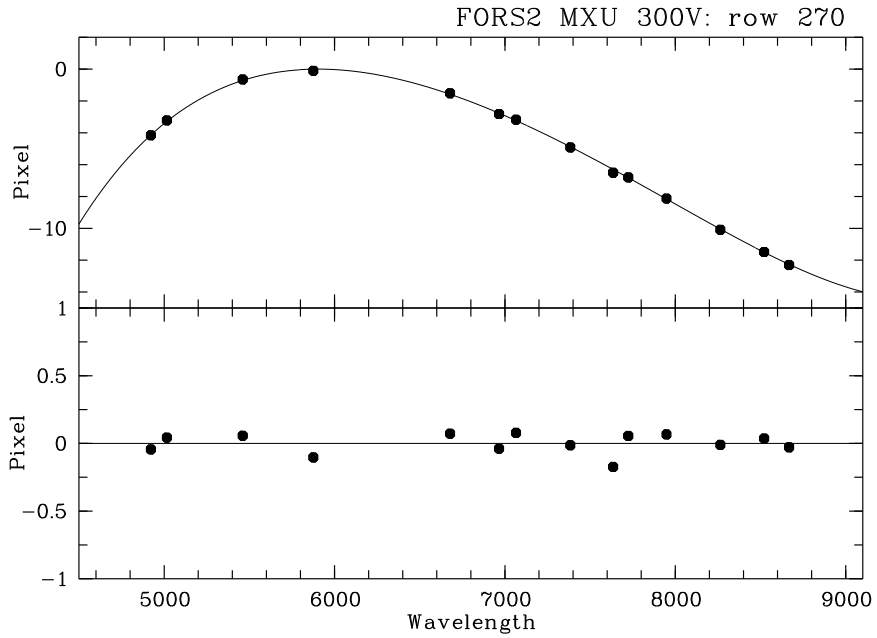

Figure 9.1: *Top panel: deviation of the identified peaks from the linear term of the 270th fitting polynomial (column d270 of the DISP_RESIDUALS_TABLE_MXU). The solid line is the polynomial model with the linear term subtracted, drawn from row 270 of the DELTA_IMAGE_MXU product. Bottom panel: fit residuals of the identified peaks (identical to the residuals recorded at row 270 of the DISP_RESIDUALS_MXU image).*

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, and only under the assumption that data are not overfitted (i.e., the degree of the fitting polynomial is not higher than necessary, or practically speaking is the lowest capable of eliminating sistematic trends in the residuals). Typical values of the model accuracy range between 0.05 and 0.1 pixels as confirmed by simulations (see Figure 9.2).

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

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

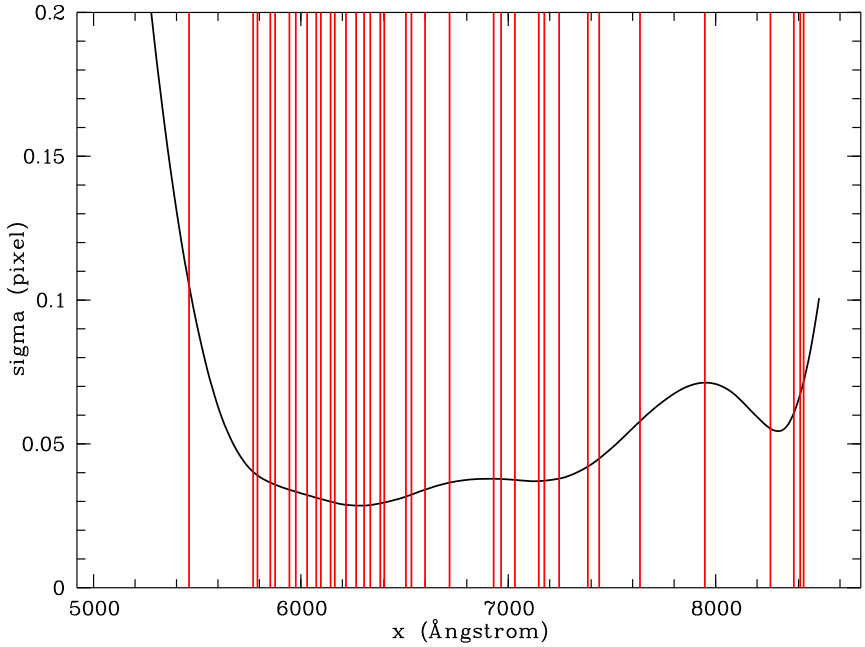

Figure 9.2: *Montecarlo simulation of wavelength calibration model accuracy (from FORS2/MXU 600RI calibration lamp). The curve describes the changing accuracy (in pixel), given at 1-σ confidence level. The vertical lines indicate the positions of the used reference arc lamp lines. It is possible to see that, as expected, the calibration is more accurate where the density of lines is higher. The accuracy of the model degrades rapidly at the blue and red ends of the spectrum, where uncertainties due to overfitting and extrapolation are greater.*

DISP_RESIDUALS_MXU: residuals of each wavelength calibration fit (in pixels). 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 9.3). Typical observed residuals should be around 0.2 pixels.[30] 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 9.9.3, page 108).

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

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

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*. Please, also note that the table contains rows for the wavelengths excluded via *--ignore_lines*. For these rows residuals and deviations are not calculated. For more information on the aforementioned parameters, refer to section 9.9.3, page 108. Just one every 10 of the polynomial fits listed in

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

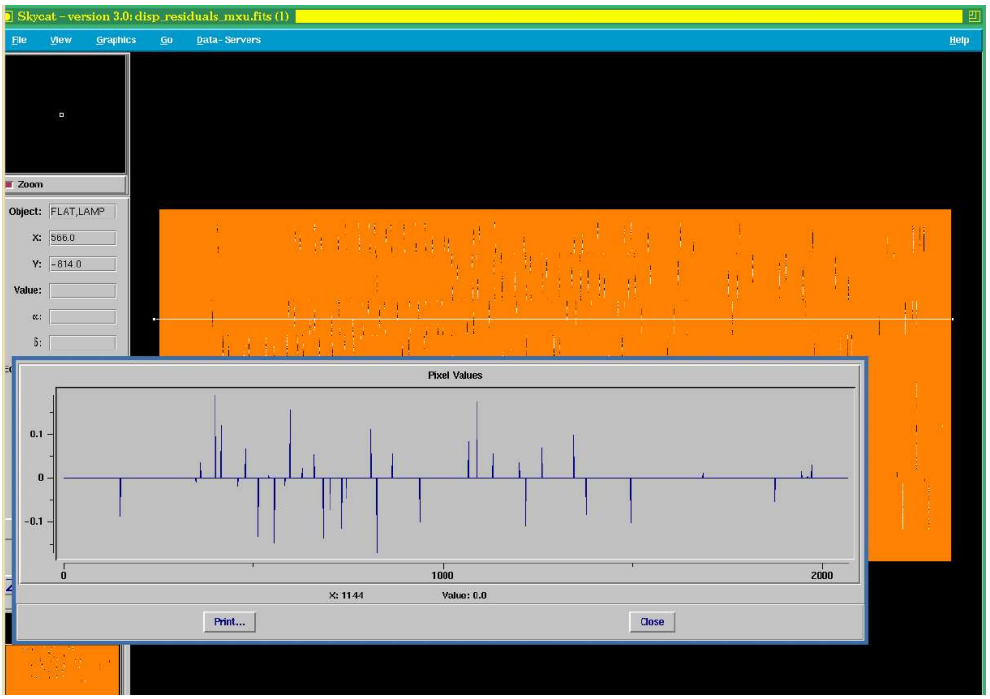

Figure 9.3: *DISP_RESIDUAL_MXU from a FORS2 MXU 600RI arc lamp calibration. In the foreground is a plot of the residuals from one image row.*

the DISP_COEFF_MXU table 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 9.1).
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*.

FLAT_SED_MXU: image containing the spectral energy distribution of the flat. Each image row corresponds to the SED of one slit. The slit order follows that of the SLIT_LOCATION_MXU table. The SED profile is wavelength calibrated using the polynomial of the central row of the illuminated region.

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 6 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. In some cases where the slit identification could not be performed (for instance, only one slit is illuminated), then it is advisable to set this parameter to FALSE. See Section 7 for more details.

This product can be used to correct LSS and PMOS spatial distortion.

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*, *--wmode* and *--wmosmode*.

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*, *--wmode* and *--wmosmode*.

MASTER_BIAS: master bias.

MASTER_NORM_FLAT_MXU: normalised flat field image, derived dividing the master screen flat by its smoothed/fitted version (see the smoothing/fitting configuration parameters description in Section 9.9.3, page 108). 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 *--s_degree*, *--sradius*, *--d_nknots*, *--dradius*, *--dradius_aver*, *--startwavelength* and *--endwavelength*.

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

MASTER_SCREEN_FLAT_MXU: combined flat field image.

REDUCED_LAMP_MXU: rectified and wavelength calibrated arc lamp image (see Figure 9.4). This is the 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 corrected. 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 9.10.3, page 124):

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

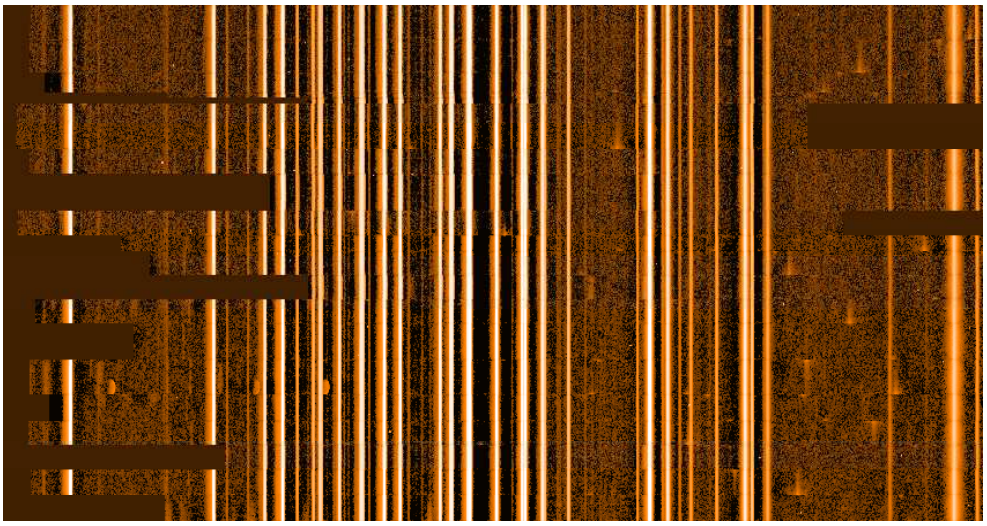

Figure 9.4: *REDUCED_LAMP_MXU from a FORS2 MXU 300I arc lamp exposure.*

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.[31] N_i is increased by 1 to ensure a slight oversampling of the original signal. The total 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

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

the interpolation point with respect to the original CCD pixels.[32] 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 makes apparent 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*, *--wmode* and *--wmosmode*.

As the product is rectified, also the quality of the slit tracing (and thus the parameters *--cdegree* and *--cmode*) may affect it.

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 position 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 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>

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:

[32] 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 would make the interpolation pointless anyway: this would allow a perfect resampling of arc lamp spectra, for instance, but would not be applicable to scientific spectra.

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 9.10.3, page 124), 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. The length calculation takes into account whether the slit falls partially outside the CCD. In this case the slit length from the header is used, therefore the determined slit length is inconsistent with the observed one.

SLIT_MAP_MXU: map of the reference wavelength on the CCD. The reference wavelength is usually the undeviated wavelength and is taken from keyword ESO INS GRIS1 WLEN. This image is only created if 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 164): 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*, *--wmosmode*, *--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 9.5). 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 data are not LSS or LSS-like.[33] 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 164). In case of problems found in the recipe products, this image may be examined. All spectra should look aligned in 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).

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

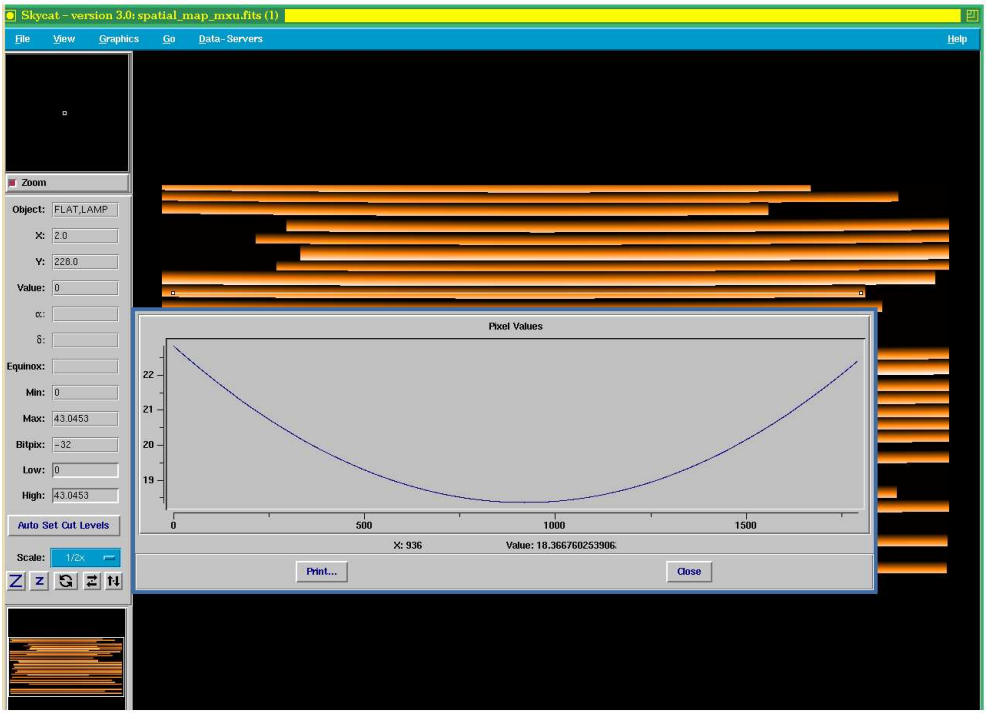

Figure 9.5: *SPATIAL_MAP_MXU from a FORS2 600RI flat field tracing, modeled with a 2nd degree polynomial. In the foreground is a plot of the distances from the top spectral edge of all pixels from one CCD row.*

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 including the line.

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*.

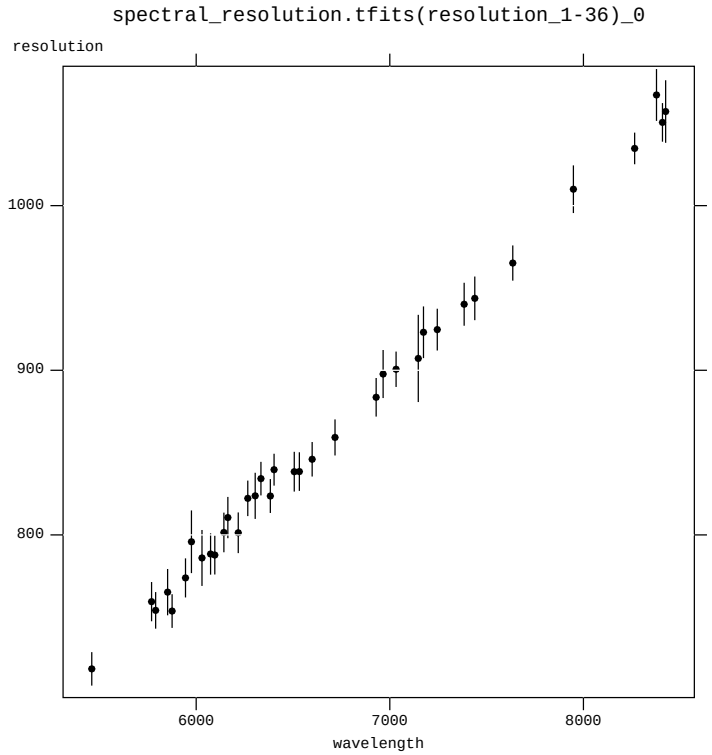

Figure 9.6: *Resolution vs. wavelength in a SPECTRAL_RESOLUTION_MXU table derived from a FORS2 MXU 600RI arc lamp exposure.*

9.9.3 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 10.6.3, page 168. The dispersion values listed in the FORS1+2 User Manual [19] 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. *--dispersion* may need to be changed for very blue or very red slits.

Optimal values for this parameter, depending on the applied grism, are included in the GRISM_TABLE (see Section 7.3, page 59). 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.[34] 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 103).

--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 10.6.2, page 167): 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 catalog 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.[35] Optimal values for this parameter, depending on the applied grism, are included in the GRISM_TABLE (see Section 7.3, page 59).

--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 Section 7.3, page 59). 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 10.6.4, page 171).

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

This parameter is mandatory (using the default 0 would generate an error message). The maximum allowed value for *--wdegree* is 5. 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 100).

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.

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

[35] 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.

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 catalog are transformed to CCD pixel positions using the model, and a peak is searched within the specified search radius.[36] 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 10.6.2, page 167).

Regardless of the value of wradius, the minimumn width that will be used to recompute the line position is 5 pixels.

--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.

--wmosmode: Interpolation mode of wavelength solution (0 = no interpolation, 1 = local (slit) solution, 2 = global model). *Default*: 0

This parameter only affects the processing of MOS and MXU data. It is conceivable an improvement of the quality of the wavelength calibration by modeling the trend of the solutions within each slit, or even

[36] 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.

globally (as a function of the position of the slits on the focal plane). If *--wmosmode* = 1 the solutions within each slit are replaced by their best linear fit, while if *--wmosmode* = 2 a bivariate, second order global solution is fitted to the available local solutions and then replaces them. No interpolation is applied to the data if *--wmosmode* = 0, and this is the default.

--ignore_lines: Catalog lines nearest to wavelengths in this list will be ignored for wavelength calibration. *Default*: empty

This parameter contains a string with a comma separated list of lines to be ignore from the reference catalogue line. In fact it is not needed to specify the exact wavelength present in the catalogue, instead, the closest lines in the catalogue to each of the values of this parameter will be ignored for the wavelength calibration. For instance, a value of *4300,5400* will ignore lines 4358.343 and 5460.742 from the standard catalog.

--used_linesets: Linesets in the line catalog to use. *Default*: standard

The [LINE_CATALOG] catalog used to compute the wavelength calibration contains a column [LINE_SET] which defines whether the line belongs to the *standard* set or to the *extended* set. Using the standard set will provide good results in most of the cases, but in some cases the user can also specify to use additionally the extended set, which might contain weaker lines or doublets.

The parameter syntax is a comma separated list of the sets to use, for instance *standard,extended*

Spatial curvature calibration

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

This parameter is mandatory (using the default 0 would generate an error message). The maximum allowed value for *--cdegree* is 5. In general a 2nd degree polynomial gives good results. Optimal values depending on the applied grism are included in the GRISM_TABLE (see Section 7.3, page 59).

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 9.7). 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.[37] By setting *--cmode* = 1 a

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

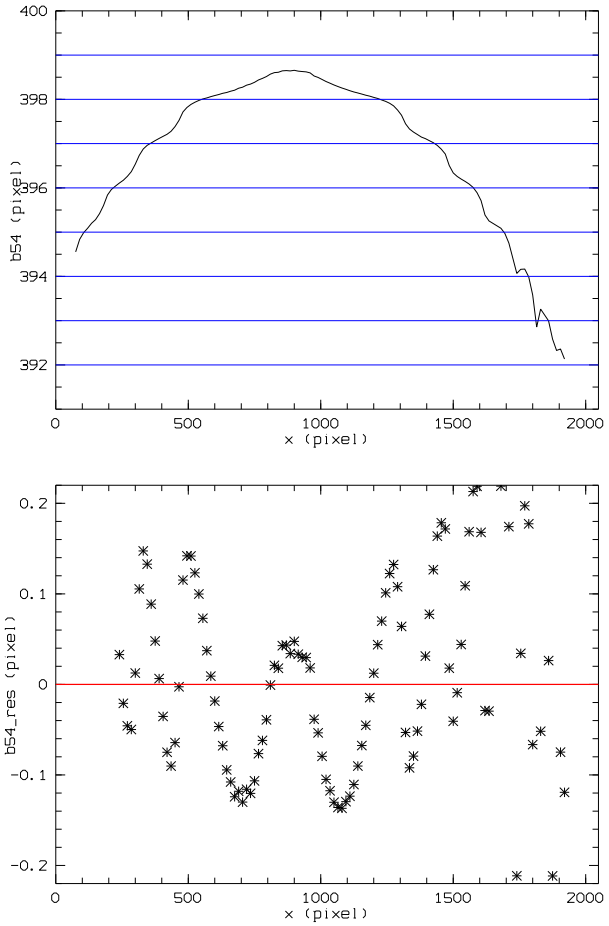

Figure 9.7: *Systematic residuals of curvature model (from a FORS2 MXU 600B flat field exposure).*

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*: FALSE

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 10.6.7, page 172). 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.[38] However, as shown in

[38]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.

Section 10.6.7, 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. Note also that excluding the slit identification would also allow to reduce data from instruments different from FORS1 and FORS2.[39] If the slit identification fails (e.g. if the number of CCD slits differs from the number of mask slits or if just two slits are provided) the parameter *--slit_ident* is forced to FALSE and the reduction proceeds without slit identification. The change of parameters is also recorded in the products header and a warning detailing the reason for the failure is printed in the logs.

Flat field normalisation

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

This parameter affects the way the raw flat field frames are stacked to produce the master flat field frame which will then be normalised.

--ksigma: Low and high threshold in ksigma method. [-3.0,3.0]

--kiter: Max number of iterations in ksigma method. [999]

These parameters are only active if *--stack_method*=ksigma. See more details in the description given in Section 9.1.3, page 76.

--s_degree: Degree of polynomial used for fitting along spatial direction. *Default*: -1

If the configuration parameter *--s_degree* 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 for each slit. See 10.10.2 for details of the normalisation algorithm.

--d_nknots: Number of knots used for the spline fitting along dispersion direction. *Default*: -1

If *--d_nknots* is set to a non-negative value, the master flat field normalisation is performed by modeling its large scale trend with a cubic spline polynomial fitted along the dispersion direction. The flat field spectra are spatially rectified applying the curvature model, then any smoothing specified by the use of the *--dradius* or *--dradius_aver* parameters is done before the fit is performed (only for non LSS nor LSS-like data). The result is mapped back to the CCD frame before being used for normalising the master flat field.

If *--d_nknots* < 0 the spline fitting is not performed and only the smoothing steps are done.

[39] 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.

--fit_threshold: Threshold percentage for flat spline fitting and polynomial fitting with respect to the maximum. *Default*: 0.01

The input pixels used for the cubic spline fiting and the polynomial fitting are first filtered to reject values below *--fit_threshold* times the maximum value in the spectrum of that row.

--dradius: Radius for median smoothing of flat field along dispersion direction. *Default*: 10 pixel

If it is not negative, a median smoothing with half width *--dradius* is applied along the dispersion direction. This occurs before any smoothing specified by the use of the *--dradius_aver* parameter.

--dradius_aver: Radius for average smoothing of flat field along dispersion direction. *Default*: -1

If it is not negative, an average smoothing with half width *--dradius_aver* is applied along the dispersion direction. This occurs after any smoothing specified by the use of the *--dradius* parameter.

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

If it is not negative, a median smoothing with half width *--sradius* is applied along the spatial direction.

Setting *--sradius=-1* and *--s_degree=-1* will leave the spatial illumination gradient in the normalised master flat field, which is helpful for LSS data.

Saturation

--nonlinear_level: Level above which the detector is not linear *Default*: 60000

This sets the number of counts above which the detector has a non-linear behaviour. Pixels are marked and ignored from the master flat combination.

--max_nonlinear_ratio: Maximum allowed ratio of non-linear pixels per slit. *Default*: 0.2

If during the master flat combination there is a slit in a flat which contains more non-linear pixels that the specified value then the whole slit is marked as bad for that flat, and is ignored from the flat combination.

9.9.4 Quality control parameters

Currently the following QC parameters, used by PSO and DPD, 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 DET CHIP.NUM: Chip number. *Units*: *none*

QC INS_COLL_NAME: Collimator name. *Units*: *none*

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 9.9.2, page 98).

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 9.9.2, page 98).

QC WAVE_MIN: Minimum wavelength. *Units*: Ångstrom

QC WAVE_MAX: Maximum wavelength. *Units*: Ångstrom

QC X_MIN: Pixel corresponding to minimum wavelength. *Units*: pixel

QC X_MAX: Pixel corresponding to maximum wavelength. *Units*: pixel

QC N_LIN: Number of lines used. *Units*: *none*

QC TRACE MAX CURVATURE: Max curvature in spectral tracing. *units*: (Y pixel/X pixel^2)

This is computed as the max absolute value of the column "c2" of the CURV_COEFF_MXU table.

QC TRACE MAX SLOPE: Max slope in spectral tracing (Y pixel/X pixel)

This is computed as the max absolute value of the column "c1" of the CURV_COEFF_MXU table.

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 catalog is evaluated as the ratio between its wavelength and its FWHM. The results are written to the SPECTRAL_RESOLUTION_LSS table (see page 107), 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 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 107).

QC NUM_SLIT: Number of slits (for MOS/MXU). *Units*: *none*

QC FLAT<jj> SLIT<ii> SAT COUNT: Number of non-linear/saturated pixels. *Units*: *None*

Non-linear or saturated pixels have either negative values or values greater than the *--nonlinear_level* parameter. The saturation count is calculated for each slit in each raw flat, where <jj> gives the position of the raw flat in the input list. For MOS or MXU, <ii> is the slit_id. For LSS data, <ii> is always 00.

QC FLAT<jj> SLIT<ii> SAT RATIO: Ratio of non-linear/saturated pixels to total pixels. *Units*: *None*

Non-linear or saturated pixels have either negative values or values greater than the *--nonlinear_level* parameter. The saturation ratio is calculated for each slit in each raw flat, where <jj> gives the position of the raw flat in the input list. For MOS or MXU, <ii> is the slit_id. For LSS data, <ii> is always 00.

QC FLAT<jj> SAT COUNT: Number of non-linear/saturated pixels. *Units*: *None*

Non-linear or saturated pixels have either negative values or values greater than the *--nonlinear_level* parameter. The saturation count is calculated across all slits in each raw flat, where <jj> gives the position of the raw flat in the input list.

QC FLAT SAT COUNT AVG: Average of QC FLAT<jj> SAT COUNT *Units*: *None*

QC N_SATURATED: Number of saturated pixels in the master product. *Units*: *None*

QC N_FLAT: Number of input raw flat frames used to produce the master. *Units*: *None*

QC SN_START: The first pixel in the central row of the master screen flat field with flux > 10000. *Units*: *pixel*

QC SN_END: The last pixel in the central row of the master screen flat field with flux > 10000. *Units*: *pixel*

QC SN_RANGE: The number of pixels in the central row of the master screen flat field with flux > 10000. *Units*: *none*

QC AVG_SN: Average signal-to-noise value in the central row of the master screen flat field with flux > 10000. *Units*: *none*

9.10 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*. In case of a standard star observation, efficiency and response curves are also calculated.

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. Also, if requested, flux calibration is performed.

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.

9.10.1 Input files

In alphabetical order:

CURV_COEFF_MXU: *required* table with spatial curvature coefficients, however *not required* for LSS and LSS-like observations.[40] This table is produced by the *fors_calib* recipe (see page 98).

DISP_COEFF_MXU: *required* table with wavelength solution coefficients. This table is produced by the *fors_calib* recipe (see page 99). The wavelength range and dispersion that will will be used for the science reduction is taken from this input table, specificalyy from keywords ESO PRO WLEN START, ESO PRO WLEN END, ESO PRO WLEN INC and ESO PRO WLEN CEN.

EOP_PARAM: *optional* table containing the Earth Orientation Parameter as a function of time (`MJD-OBS`). It must be provided when barycentric correction is requested (see configuration parameter *--apply_barycorr*, Section 9.10.3, page 124). The EOP_PARAM table is provided as a static calibration, or an updated version may be downloaded using the recipe *fors_eop*.

EXTINCT_TABLE: *optional* atmospheric extinction table. If a standard star exposure is specified in input, and efficiency and response curves are wanted, this table must also be specified in input. It must also be specified in case a spectro-photometric calibration will be applied to the extracted spectra (i.e., when a SPECPHOT_TABLE or MASTER_SPECPHOT_TABLE frame is present in the sof). See Section 7.8, page 62.

FLAT_SED_MXU: *optional* master flat spectral profile. If this is input, the extracted spectra will be divided by the master flat spectral profile corresponding to the spectra slit. For standard star observations this will change the final values of the response. It should be used for the science photometric correction if the response was computed with it.

GLOBAL_DISTORTION_TABLE: *optional* global distortion table. This input is used only for LSS data. If present, the spatial calibration of LSS will be perfomed.

GRISM_TABLE: *optional* grism table. See Section 7.3, page 59.

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 103), and it must be provided only if the flat field correction is requested (see configuration parameter *--flatfield*, Section 9.10.3, page 124).

[40] Currently no spatial curvature correction is applied to LSS data, this will be fixed in the next releases.

MASTER_SKYLINECAT: *optional* sky lines catalog. 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 with name "WLEN" listing such wavelengths. If the alignment of the wavelength solution to the sky lines is requested, and a MASTER_SKYLINECAT is not specified in input, an internal sky line catalog is used instead (see Table 9.2).

MASTER_SPECPHOT_TABLE: *optional* table with master response curve. If it is specified, a spectro-photometric calibration will be applied to the extracted spectra. See also Section 10.5, page 163, about how the photometric calibration is applied. Note that if both SPECPHOT_TABLE and MASTER_-SPECPHOT_TABLE are specified, then only MASTER_SPECPHOT_TABLE will be used.

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

SLIT_LOCATION_MXU: *required* table of slits positions. This table is produced by the *fors_calib* recipe (see page 105).

SPECPHOT_TABLE: *optional* table with efficiency and response curves. If it is specified, a spectro-photometric calibration will be applied to the extracted spectra This table is also a product of this recipe, and it is described in more detail in the next Section. See also Section 10.5, page 163, about how the photometric calibration is applied.

STD_FLUX_TABLE: *optional* standard star flux table. If a standard star exposure is specified in input, and efficiency and response curves are wanted, this table must also be specified in input. Its keyword ESO OBS TARG NAME must match the one in the input standard star exposure frame. See Section 7.9, page 62.

TELLURIC_CONTAMINATION: *optional* wavelength ranges affected by telluric contamination. This table can be used to mask the atmospheric telluric lines from the response fit.

9.10.2 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.[41] Here is the list of all the possible output frames, in alphabetical order, together with a list of related configuration parameters:[42]

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 9.10.3, page 124). For a description of this product see the DISP_COEFF_MXU entry on page 99. In the DISP_COEFF_SCI_MXU table the *error* column content is computed by

[41] 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.

[42] See Section 9.10.3, page 124, for a complete description of the recipe configuration parameters.

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 9.2: *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.*

(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.

GLOBAL_SKY_SPECTRUM_MXU: table with supersampled sky spectrum, created only if the global sky subtraction is requested (see configuration parameter *--skyglobal*, Section 9.10.3, page 124). Each wavelength bin is half the resampling step, multiplied by the CCD readout rebin factor (see the configuration parameter *--dispersion*, Section 9.10.3, page 124).

The spectra contained in the input scientific exposure (see the SCIENCE_MXU entry on page 117) 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.[43] 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*.

MAPPED_FLUX_SCI_MXU: photometrically calibrated scientific slit spectra. This image matches the non flux-calibrated MAPPED_SCI_MXU image, and it is produced only if the spectrophotometric calibration is applied (i. e. if a SPECPHOT_TABLE or MASTER_SPECPHOT_TABLE is present).

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 per pixel, D, defined by the configuration parameter *--dispersion* (see Section 9.10.3, page 124):

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

The y size is determined in the same way as for the REDUCED_LAMP_MXU frame (see page 103).

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 103.

Configuration parameters directly affecting this product are *--flatfield* and *--skyalign*.

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 is requested.

[43] 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.

Configuration parameters directly affecting this product are *--cosmics*, *--flatfield*, *--skyalign*.

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 9.10.3, page 124).

Configuration parameters directly affecting this product are *--skymedian*, *--skyglobal*, *--skylocal*, and *--cosmics*.

Configuration parameters having significant impact are *--skyalign* and *--flatfield*.

OBJECT_TABLE_SCI_MXU: This table is an expansion of the input SLIT_LOCATION_MXU table (see page 105), where the positions and the extraction spatial intervals of the detected objects are also included.[44] This table is produced only if any kind of sky subtraction is requested, otherwise no object detection or extraction is attempted. The object table columns are the following[45]:

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 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. This row is **not** included in the extracted spectrum.
end_1, end_2, ...:	End position of the extraction interval for each object. This row is included in the extracted spectrum.
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. The sources themselves are numbered from the top to the bottom of the detector. If more than one object per slit is detected, the object referenced

[44] 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.

[45] All pixel coordinates refer to actual pixels, i.e. for the default binning 2x2 one pixel refers to the original 2 along the spatial axis. Some data display tools (for instance skycat) instead keep the memory of the binning, so that the numbers in the table need to be multiplied by two to compare for instance the contents of the columns positions and length to the displayed image.

in row_1 will be in the row above object in row_2,
i.e., row_1 = row_2 + 1.

ra_1, dec_1, ra_2, dec_2 ... Sky coordinates of the extracted spectra.
See 10.12 for more details
about the coordinate computation.

Configuration parameters directly affecting this product are *--slit_margin*, *--ext_radius*, *--cont_radius*.

REDUCED_ERROR_SCI_MXU: image with errors (one sigma level) corresponding to the extracted objects spectra. This image matches the REDUCED_SCI_MXU image.

The configuration parameter *--ext_mode* affects directly this product.

Configuration parameters having significant impact are *--slit_margin*, *--ext_radius* and *--cont_radius*.

REDUCED_FLUX_ERROR_SCI_MXU: error on photometrically calibrated scientific spectra. This image matches the REDUCED_ERROR_SCI_MXU image, and it is produced only if the spectrophotometric calibration is applied (i. e. if a SPECPHOT_TABLE or MASTER_SPECPHOT_TABLE is present).

REDUCED_FLUX_SCI_MXU: photometrically calibrated scientific spectra. This image matches the REDUCED_SCI_MXU image, and it is produced only if the spectrophotometric calibration is applied (i. e. if a SPECPHOT_TABLE or MASTER_SPECPHOT_TABLE is present).

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 (columns row_XXX), which starts numbering from the top to the bottom of the detector.

The configuration parameter *--ext_mode* affects directly this product.

Configuration parameters having significant impact are *--slit_margin*, *--ext_radius*, *--cont_radius*, *--skyalign*, *--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.

The configuration parameter *--ext_mode* affects directly this product.

Configuration parameters having significant impact are *--slit_margin*, *--ext_radius*, *--cont_radius*, *--skyalign*, *--flatfield*, *--skylocal*, *--skyglobal* and *--skymedian*.

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 9.10.3, page 124). 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.[46]

The included columns are the following:

wave: Sky line wavelength.
offset<slit_id>: Observed offsets for the slit spectrum with identification *slit_id*.

[46] In general the sky lines detection fails for reference slits, that are typically filled up by very bright objects.

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.

SPECPHOT_TABLE: table with efficiency and response curves, produced only when input includes a standard star observation, an EXTINCT_TABLE, and the appropriate STD_FLUX_TABLE matching the observed star.

The SPECPHOT_TABLE has two table extensions. The first table extension contains wavelength bins that correspond to the input STD_FLUX_TABLE while the second extension contains wavelength bins that correspond to the observed spectrum.

The first table extension include the following columns:

WAVE:	Wavelength (Å)
STD_FLUX:	Standard star flux (10^{-16} erg cm^{-2} s^{-1} Å^{-1})
OBS_FLUX:	Observed flux (e^- s^{-1} Å^{-1})
RAW_EFFICIENCY:	Ratio between input and detected photons
EFFICIENCY:	Fit of **RAW_EFFICIENCY** by using polynomial or spline
RAW_RESPONSE:	Ratio between **std_flux** and **obs_flux**
RESPONSE:	Fit of **RAW_RESPONSE** by using polynomial or spline
USED_FIT:	Flag to signal whether this bin was used in the response fit
IS_EXTRAPOLATED:	Flag to signal whether this bin was calculated through extrapolation

If the response has been corrected by the flat SED then there are additional columns added:

OBS_FLUX_FFSED:	Observed flux (OBS_FLUX) divided by the flat SED
RAW_RESPONSE_FFSED:	Ratio between **std_flux** and **obs_flux_ffsed**
RESPONSE_FFSED:	Fit of **RAW_RESPONSE_FFSED** by using polynomial or spline

The second table extension include the following columns:

WAVE:	Wavelength (Å)
EFFICIENCY:	Fit of **RAW_EFFICIENCY** (from the in first table extension) by using polynomial or spline
RESPONSE:	Fit of **RAW_RESPONSE** (from the in first table extension) by using polynomial or spline

The way this table is produced is described in Section 10.4, page 162.

Configuration parameters directly affecting this product are:
--resp_fit_degree, *--resp_fit_nknots*, *--resp_ignore_mode*, *--resp_ignore_points*, *--resp_use_flat_sed*.

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 9.10.3, page 124).[47] 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*.

[47] 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.

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 9.10.3, page 124), 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 9.10.3, page 124), this image contains the sky model obtained by interpolating the sky signal trend along the spatial direction, directly on the CCD frame.

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*.

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 9.10.3, page 124). For a description of this product see the WAVELENGTH_MAP_MXU entry on page 107. 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.

The configuration parameter *--skyalign* affects directly this product.

REDUCED_IDP_SCI_MXU_i: IDP-compliant extracted spectra, see [27]. i is an index between 1 and the number of extracted spectra. A product is generated for each spectra. The products are generated only if *--generate_idp* is set to TRUE.

The configuration parameter *--generate_idp* affects directly this product.

9.10.3 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*: -1

The input wavelength calibration can be adjusted to the observed positions of a set of sky lines, whose wavelengths are listed in an input catalog. The observed sky lines offsets from their expected positions (see entry SKY_SHIFTS_SLIT_SCI_MXU, page 122) are fitted by polynomials that are then added to the input wavelength calibration polynomials (see DISP_COEFF_MXU entry on page 99). 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.

--apply_barycorr Apply barycentric correction. *Default*: FALSE

Setting this parameter to TRUE will cause the recipe to apply a barycentric correction to the output products. This requires that an EOP_PARAM input frame be provided.

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

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 103, 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 column of pixels for each spectrum on the CCD.[48] 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.

[48] 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 slit length the sky is trended using a 2nd degree polynomial.

A MAPPED_SKY_SCI_MXU (page 121) 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.[49]

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 121) is produced in this case. Note that global and median sky subtractions are not mutually exclusive.

--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 124 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*). Note that for LSS or LSS-like data *--skyglobal* must be requested. 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 120). Cosmic ray hits are removed anyway by the optimal extraction procedure of the detected objects.

Objects detection and extraction

--slit_margin: Spectrum edge pixels to exclude from object search. *Default*: 3 (irrespective of binning) 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*: 12 (unbinned) pixel

[49] 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.

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. Please note that the provided radius refers to the original, i.e. unbinned pixels and that the binning factor is applied to it.

--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 [34].

Flux calibration

--resp_fit_nknots: Number of knots of the response spline fitting. If -1, then no spline fitting is performed. If -2, then the value is read from the GRISM_TABLE (see Section 7.3).*Default*: -2

See Section 10.4, page 162, for the meaning of this parameter.

--resp_fit_degree: Degree of polynomial for the response polynomial fitting. If -1, then no polynomial fitting is performed. If -2, then the value is read from the GRISM_TABLE (see Section 7.3)*Default*: -2

Take into account that if both *--resp_fit_nknots* and *--resp_fit_degree* are greater than 0 the settings are incompatible and the pipeline will stop. See Section 10.4, page 162, for the meaning of this parameter.

--resp_ignore_mode: Types of lines/regions to ignore in response. Valid ones are 'stellar_absorption', 'telluric' and 'command_line' (from parameter resp_ignore_lines) *Default*: stellar_absorption,telluric,command_line

This parameter contains a comma separated list of types of lines or regions to ignore during response computation. If it includes *stellar_absorption* then the lines marked as stellar absorption in the standard star calibration table (STD_FLUX_TABLE) will be ignored. If it includes *telluric* then the lines and regions specified in the telluric contamination table (TELLURIC_CONTAMINATION) will be ignored. If it includes *command_line*, the lines and regions specified in parameter *resp_ignore_points* will be ignored.

--resp_ignore_points: Extra lines/regions to ignore in response *Default*:

This parameter contains a comma separated list of lines to ignore during response computation (if parameter *response_ignore_model* contains *command_line*). A range can also be specified like 4500.0-4600.0.

--resp_use_flat_sed: Flag to determine whether to apply flat sed correction *Default*: grism_table

Possible values are *true, false, grism_table*. If *true*, then the observed spectra will be divided by the flat sed before applying the photometric calibration. This is needed for the proper calibration of holographic grisms which show a position dependant response. If the observed targe tis a standard star, then the response will contain this correction and the science must also be corrected by the same effect.

If the value is *grism_table* then the option is read from the GRISM_TABLE table, which contains the column **RESP_USE_FLAT_SED**. This is because for some grism (specially holographic ones) this option is strongly recommended, while for others it is not needed.

--generate_idp: If set to TRUE FITS tables in the VO compliant "IDP" format are created, one FITS table per extracted spectrum. *Default*: TRUE

9.10.4 Quality control parameters

Currently the following QC parameters, used by PSO and DPD, are evaluated by the *fors_science* recipe in the case of standard star observations.

QC SPEC EFFICIENCYi: Efficiency at given wavelength. *Units*: e^-/photon

The efficiency curve is computed as the ratio between the standard star catalog flux converted to photons (per second per Ångstrom), and the observed standard star spectrum converted to electrons (per second per Ångstrom).

The details of the calculation and smoothing of the efficiency curve are given in Section 10.4, page 162. This smoothed efficiency curve is used to calculate discrete efficiency values at the wavelengths of the observed spectrum, and these are the values written to the second extension of the output SPECPHOT_TABLE, as produced by this recipe.

However, the QC parameters are always computed in fixed wavelength bins. The wavelengths of QC SPEC EFFICIENCYi LAMBDA start at 3700 Ångstrom with steps of 400 Ångstrom and a total of 15 steps. For each of these wavelength steps the QC SPEC EFFICIENCYi value is calculated from a linear interpolation from the table of discrete efficiency values.

QC SPEC EFFICIENCYi LAMBDA: Wavelength at which the efficiency was measured. *Units* Ångstrom

QC RESP FLAT_SED_CORR Set to true if the response contains the flat sed correction.

QC RESP FLAT_SED_NORM Flat sed correction normalisation factor.

Currently the following QC parameters, used by PSO and DPD, are evaluated by the *fors_science* recipe for all observations.

QC DET CHIP NUM: Chip number. *Units*: *none*

QC INS_COLL_NAME: Collimator name. *Units*: *none*

QC AIRM MEAN: Average airmass.

Average of TEL.AIRM.START and TEL.AIRM.END

QC IQ: Image quality. *Units*: *arcsec*

Value of the keyword TEL.IA.FWHM * (QC.AIRM.MEAN**0.6)

QC AMBI REQ: Requested maximum seeing. *Units*: *arcsec*

Value of the OBS AMBI FWHM keyword or, if keyword missing, then value of 1.0

QC N_SATURATED: Number of saturated pixels. *Units*: *None*

QC OBJECT_POSITION: Position of brightest object along the slit. *Units*: *pixel*

QC PROC SCHEME: Character string that denotes the phase-3 release.

Set to 'FORS2_R'.

QC AVG_ACCURACY: The QC AVG_ACCURACY value from the MASTER_NORM_FLAT.

QC SIG_ACCURACY: The QC SIG_ACCURACY value from the MASTER_NORM_FLAT.

QC N_SATURATED_FLAT: The QC N_SATURATED value from the MASTER_NORM_FLAT.

QC SAT COUNT: Number of non-linear/saturated pixels per object. *Units*: *None*

Saturated pixels are those where the pixel value is negative or is greater than the *--nonlinear_level* parameter. The saturation count is made within the object extraction region within the MAPPED_ALL_SCI_MXU image.

Each IDP product will contain this QC keyword. The keywords for all objects will appear in the MAPPED_ALL_SCI_MXU and MAPPED_SCI_MXU products, with the appropriate object number included in the keyword.

QC SAT RATIO: Ratio of non-linear/saturated pixels to total pixels per object. *Units*: *None*

Saturated pixels are those where the pixel value is negative or is greater than the *--nonlinear_level* parameter. The saturation count is made within the object extraction region within the MAPPED_ALL_SCI_MXU image.

Each IDP product will contain this QC keyword. The keywords for all objects will appear in the MAPPED_ALL_SCI_MXU and MAPPED_SCI_MXU products, with the appropriate object number included in the keyword.

9.11 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 MASTER_DISTORTION_TABLE (see page 61) is required in input instead of the DISP_COEFF_MXU, CURV_COEFF_MXU, and SLIT_LOCATION_MXU tables (see pages 98, 99, and 105). 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.

9.12 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 [16], 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.

9.12.1 Input files

FLUX_FLAT_LSS: exposure for flat field lamp monitoring.

FLUX_ARC_LSS: exposure for arc lamp monitoring.

9.12.2 Output files

FLUX_LAMP_LSS: frame including the background subtracted integration region.

9.12.3 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.

9.13 fors_pmos_calib

This recipe is, except for minor differences, identical to the *fors_calib* recipe (see Section 9.9, page 97). Its scope is processing flat field and arc lamp exposures associated to spectro-polarimetric scientific observations.

The recipe *fors_pmos_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.

9.13.1 Input files

In alphabetical order:

GRISM_TABLE: *optional* grism table. See *fors_calib* recipe, Section 9.9.1, page 97.

LAMP_PMOS: *required* raw arc lamp spectrum exposures. Differently from the *fors_calib* recipe, more than one arc lamp exposure can be specified here (typically one per retarder plate angle, even if this is not strictly required).

MASTER_BIAS: *required* master bias frame. See *fors_calib* recipe, Section 9.9.1, page 97.

MASTER_DISTORTION_TABLE: *required* description of the instrument optical and spectral distortions (see Section 7.6, page 61). This is necessary for the identification of the ordinary and extraordinary spectral beams.

MASTER_LINECAT: *required* line catalog. See *fors_calib* recipe, Section 9.9.1, page 97.

SCREEN_FLAT_PMOS: *required* raw spectral screen flat exposures. If more than one is provided, the input frames are stacked into one. Typically the flat field exposures are all obtained at zero retarder plate angle.

9.13.2 Output files

Not all output frames listed here are always produced. Some of them are created only on request (see Section 9.13.3, page 134). Here is a list of all the possible output frames, in alphabetical order, together with a list of related configuration parameters.[50]

CURV_COEFF_PMOS: table containing the coefficients of the spatial curvature fitting polynomials. The table columns are the following:

slit_id:	Slit identification number (see the SLIT_LOCATION_PMOS 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.

[50] See Section 9.13.3, page 134, for a complete description of the recipe configuration parameters.

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

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

CURV_TRACES_PMOS: table containing the y CCD positions of the detected spectral edges at different x CCD positions. 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*.

See entry CURV_TRACES_MXU in Section 9.9.2, page 98, for more details.

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

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

DELTA_IMAGE_PMOS: deviation from the linear term of the fitting wavelength calibration polynomials. This is a multi-extension FITS file, with one image extension for each input arc lamp exposure. See entry DELTA_IMAGE_MXU, Section 9.9.2, page 99 for more details.

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

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

DISP_COEFF_PMOS: tables containing the wavelength calibration polynomial coefficients. This is a multi-extension FITS file, with one table extension for each input arc lamp exposure. Each table contains as many rows as in the REDUCED_LAMP_PMOS images, ordered in the same way. The table columns are the same as described for entry DISP_COEFF_MXU, Section 9.9.2, page 99.

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

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

DISP_RESIDUALS_PMOS: residuals of each wavelength calibration fit (in pixels). These images are only created and inserted in different extensions of a FITS files if the *--check* configuration parameter is set. See entry DISP_RESIDUALS_MXU in Section 9.9.2, page 100.

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

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

DISP_RESIDUALS_TABLE_PMOS: tables containing different kinds of residuals of a sample of wavelength calibration fits. See entry DISP_RESIDUALS_TABLE_MXU, Section 9.9.2, page 101.

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

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

MAPPED_NORM_FLAT_PMOS: rectified and wavelength calibrated normalised screen flat field image (see entry MAPPED_NORM_FLAT_MXU in Section 9.9.2, page 103).

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

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

MAPPED_SCREEN_FLAT_PMOS: rectified and wavelength calibrated master screen flat field image (see entry MAPPED_SCREEN_FLAT_MXU in Section 9.9.2, page 103).

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

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

MASTER_BIAS: master bias.

MASTER_NORM_FLAT_PMOS: normalised flat field image, derived dividing the master screen flat by its smoothed version (see the smoothing configuration parameters description in Section 9.9.3, page 108). Comparing this image with the MASTER_SCREEN_FLAT_PMOS 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 *--s_degree*, *--sradius*, *--d_nknots*, *--dradius*, *--startwavelength* and *--endwavelength*.

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

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

REDUCED_LAMP_PMOS: rectified and wavelength calibrated arc lamp images, inserted in different extensions of an output FITS file. See entry REDUCED_LAMP_MXU, Section 9.9.2, page 103, for more details.

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

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

SLIT_LOCATION_PMOS: slit positions, both on the CCD and on the rectified images of the arc lamp exposures (REDUCED_LAMP_PMOS). See entry SLIT_LOCATION_MXU in Section 9.9.2, page 105 for more details.

SLIT_MAP_PMOS: map of central wavelength on the CCD. This image is only created if the *--check* configuration parameter is set. It has the same size of the WAVELENGTH_MAP_PMOS 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 164): 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*, *--wmosmode*, *--dispersion*, *--peakdetection*, *--wradius*, and *--wreject*.

SPATIAL_MAP_PMOS: map of spatial positions on the CCD. See entry SPATIAL_MAP_MXU in Section 9.9.2, page 98, for more details.

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

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

SPECTRA_DETECTION_PMOS: result of the preliminary wavelength calibration applied to the arc lamp exposure. This image is only created if the *--check* configuration parameter is set. See entry SPECTRA_DETECTION_MXU, Section 9.9.2, page 106, for details.

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

SPECTRAL_RESOLUTION_PMOS: Mean spectral resolution for each reference arc lamp line. This is a multi-extension FITS file, with one extension for each input arc lamp frame. For more details, see the entry SPECTRAL_RESOLUTION_MXU, Section 9.9.2, page 107.

WAVELENGTH_MAP_PMOS: map of wavelengths on the CCD. This is a multi-extension FITS file, with one extension for each input arc lamp frame. Each 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*.

9.13.3 Configuration parameters

The configuration parameters of the *fors_pmos_calib* recipe are exactly the same as those of the *fors_calib* recipe. See therefore Section 9.9.3, page 108, for a description of such parameters.

9.13.4 Quality control parameters

A similar set of QC parameters to those produced by recipe *fors_calib* are also produced by *fors_pmos_calib*. Refer to Section 9.9.4, page 114, for details. Note that, in the case of multi-extension product FITS files, a set of QC parameters is produced for each extension.

QC WAVE ACCURACY: See Section 9.9.4.

QC WAVE ACCURACY ERROR: See Section 9.9.4.

QC TRACE MAX CURVATURE: See Section 9.9.4.

QC TRACE MAX SLOPE: See Section 9.9.4.

QC PMOS RESOLUTION: See QC LSS RESOLUTION in Section 9.9.4.

QC PMOS RESOLUTION RMS: See QC LSS RESOLUTION RMS in Section 9.9.4.

QC PMOS RESOLUTION NLINES: See QC LSS RESOLUTION NLINES in Section 9.9.4.

QC PMOS RESOLUTION NWAVE: See QC LSS RESOLUTION NWAVE in Section 9.9.4.

QC PMOS RESOLUTION MEANRMS: See QC LSS RESOLUTION MEANRMS in Section 9.9.4.

QC PMOS CENTRAL WAVELENGTH: See QC LSS CENTRAL WAVELENGTH in Section 9.9.4.

QC RET_ANGLE: Retarder angle. *Units*: *degrees*

QC NFLAT: Number of raw flats used to create the master flat. *Units*: *None*

QC NUM_SLIT: Number of slits. *Units*: *None*

QC N_LIN_AVG: Average number of lines used over all slits. *Units*: *None*

QC AVG_ACCURACY<i>_<j>: Mean of model accuracy. *Units*: pixel

Calculated over all rows, where <i>_<j> (angle in degrees) = 0_0, 22_5, 45_0, 67_5, 90_0, 112_5, 135_0, 157_5, 180_0, 202_5, 225_0, 47_5, 270_0, 292_5, 315_0, 337_5.

QC SIG_ACCURACY<i>_<j>: Sigma of model accuracy. *Units*: pixel

Calculated over all rows, where <i>_<j> (angle in degrees) = 0_0, 22_5, 45_0, 67_5, 90_0, 112_5, 135_0, 157_5, 180_0, 202_5, 225_0, 47_5, 270_0, 292_5, 315_0, 337_5.

QC CENTRAL_WLEN<i>_<j>: Computed wavelength at the center of slitlet 10. *Units*: Ångstrom

Where <i>_<j> (angle in degrees) = 0_0, 22_5, 45_0, 67_5, 90_0, 112_5, 135_0, 157_5, 180_0, 202_5, 225_0, 47_5, 270_0, 292_5, 315_0, 337_5.

QC CENTRAL_RESOLUTION<i>_<j>: Mean spectral resolution. *Units*: *None*

Averaged over whole CCD, where <i>_<j> (angle in degrees) = 0_0, 22_5, 45_0, 67_5, 90_0, 112_5, 135_0, 157_5, 180_0, 202_5, 225_0, 47_5, 270_0, 292_5, 315_0, 337_5.

QC CENTRAL_RESOLUTION_MEANRMS<i>_<j>: Mean RMS of spectral resolution for each line. *Units*: *None*

Averaged over all lines, where <i>_<j> (angle in degrees) = 0_0, 22_5, 45_0, 67_5, 90_0, 112_5, 135_0, 157_5, 180_0, 202_5, 225_0, 47_5, 270_0, 292_5, 315_0, 337_5.

QC NWAVE<i>_<j>: Independent number of wavelengths used for resolution. *Units*: *None*

Where <i>_<j> (angle in degrees) = 0_0, 22_5, 45_0, 67_5, 90_0, 112_5, 135_0, 157_5, 180_0, 202_5, 225_0, 47_5, 270_0, 292_5, 315_0, 337_5.

QC FLAT SAT COUNT AVG: Average of the number of saturated pixels in the input raw flats. *Units*: *None*

QC N_SATURATED: Number of saturated pixels in the master product. *Units*: *None*

9.14 fors_pmos_science

This recipe is used for reducing FORS1 and FORS2 spectro-polarimetric scientific data, applying the extraction mask and the normalised flat field created by the recipe *fors_pmos_calib*. This recipe acts exactly like the recipe *fors_science* (see Section 9.10, page 116), but in addition it combines the extracted spectra corresponding to the same objects observed in the different light beams and at different angles of the retarder plate, determining the polarization signal for each object. This part of the recipe is based on [26], [32], and it strictly implements [38] (preferring the "difference method"[51] to the "ratio method"). It is advisable to read [38] for a complete understanding of how both FORS and *fors_pmos_science* works. See also Section 10.7 for an overview of the applied algorithms.

In this Section it is intended that there is no one-to-one relation between "extracted spectrum" and "object". For each object there will be several extracted spectra, i.e., two for each exposure (one for each light beam). For instance, if a spectro-polarimetric observation consisted of four exposures at four different angles of the retarder plate, each object would have 4 angles x 2 beams = 8 spectra. For clarity, in the following sections it will be consistently indicated with "object-spectrum" one of the spectra from an object, and with "object-source" the object itself. This nomenclature is intentionally cumbersome, so that there is no risk of missing it.

In the following sections the word SCIENCE may be alternatively read STANDARD, as the spectro-polarimetric standard star exposures are reduced as scientific exposures. In case of standard star observations the SCI acronym in the products names should also be read STD.

9.14.1 Input files

In alphabetical order:

CURV_COEFF_PMOS: *required* table with spatial curvature coefficients. This table is produced by the *fors_pmos_calib* recipe (see page 131).

DISP_COEFF_PMOS: *required* tables with wavelength solution coefficients. This table is produced by the *fors_pmos_calib* recipe (see page 132). Note that the solution of the first table extension, valid for the first angle of the retarder plate, is applied to all the scientific exposures (i.e., to all observed angles). This is acceptable, not only because the dependency of the wavelength solution on the retarder plate angle is well below the uncertainties, but also because it eliminates an important class of systematic errors in the extracted polarimetric signal, as described in [38].

This means that the recipe *fors_pmos_calib* is not really necessary for enabling the reduction of spectro-polarimetric scientific data: *fors_pmos_science* can also make use of the products of the recipe *fors_calib*.[52]

EOP_PARAM: *optional* table containing the Earth Orientation Parameter as a function of time (`MJD-OBS`). It must be provided when barycentric correction is requested (see configuration parameter *--apply_barycorr*, Section 9.10.3, page 124). The EOP_PARAM table is provided as a static calibration, or an updated version may be downloaded using the recipe *fors_eop*.

[51] A quick description of this method can also be found in [19].

[52] The recipe *fors_pmos_calib* was only created to enable the monitoring of the wavelength solution at different angles of the retarder plate.

GRISM_TABLE: *optional* grism table. See Section 7.3, page 59.

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

MASTER_NORM_FLAT_PMOS: *optional* normalised flat field. This frame is produced by the *fors_pmos_calib* recipe (see page 133), and it must be provided only if the flat field correction is requested (see configuration parameter *--flatfield*, Section 9.10.3, page 124).

MASTER_SKYLINECAT: *optional* sky lines catalog. See page 118.

RETARDER_WAVEPLATE_CHROMATISM: *optional* chromatism correction. If the *--chromatism* parameter is set, the tabulated color dependency of the direction of the *linear* polarization vector is kept into account by the pipeline (see figure 9.8).

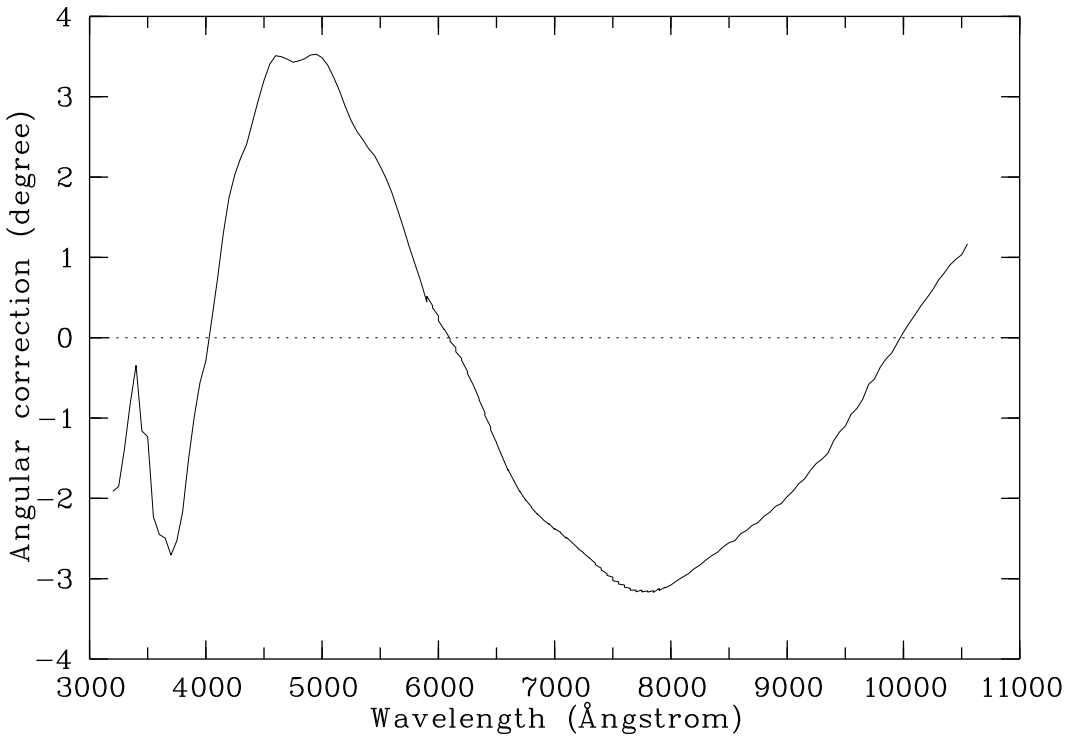

Figure 9.8: *Half-wave retarder plate chromatism. Currently this correction is assumed valid both for FORS1 and FORS2 data, since they are obtained with the same retarder waveplate.*

SCIENCE_PMOS: *required* scientific exposures. They must represent one complete polarimetric observation. A complete observation consists of a set of scientific exposures obtained at predefined angles of the retarder plate.

For circular polarimetry observations, at least two scientific exposures are required, with the 1/4 wave retarder plate set respectively at $-45°$ and $45°$; the set may be extended with two more scientific exposures, at $135°$ and $225°$: with four exposures it becomes possible for the recipe to also compute the so-called null-parameters (see [38]).

For linear polarimetry observations, at least four scientific exposures are required, with the 1/2 wave retarder plate set respectively at $0°$, $22.5°$, $45°$, and $67.5°$; the set can be extended with four more scientific exposures, at $90°$, $112.5°$, $135°$, and $157.5°$, and further extended up to 16 exposures, adding the angles $180°$, $202.5°$, $225°$, $247.5°$, $270°$, $292.5°$, $315°$, and $337.5°$. Only with a set of 8 and 16 scientific exposures the computation of the null parameters becomes possible [38].

SLIT_LOCATION_PMOS: *required* table of slits positions. This table is produced by the *fors_pmos_calib* recipe. See entry SLIT_LOCATION_MXU in Section 9.9.2, page 105 for more details.

STD_PMOS_TABLE: *required* table if a polarimetric standard star frame is specified in input, and the computation of QC parameters was requested. See Section 9.14.1, page 138 for a description of this table.

9.14.2 Output files

Not all output frames listed here are always produced. Some of them are created only on request, and some other are created according to the character and quantity of the input data.

Here is the list of all the possible output frames, in alphabetical order, together with a list of related configuration parameters:[53]

DISP_COEFF_SCI_PMOS: This adjustment of the input DISP_COEFF_PMOS table is only created in case the alignment of the wavelength solution to the sky lines is requested (see the configuration parameter *--skyalign*, Section 9.10.3, page 124). For a description of this product see the DISP_COEFF_MXU entry on page 99. In the DISP_COEFF_SCI_PMOS 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*.

MAPPED_ALL_SCI_PMOS: FITS file with as many extensions as input scientific exposures, including images with rectified and wavelength calibrated slit spectra from each input image. Their x size depends on the spectral extraction range $(\lambda_{min}, \lambda_{max})$ and on the specified resampling step in wavelength units per pixel, D, defined by the configuration parameter *--dispersion* (see Section 9.10.3, page 124):

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

Their y size is determined in the same way as for the REDUCED_LAMP_PMOS frame.

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 103.

[53] See Section 9.14.3, page 142, for a complete description of the recipe configuration parameters.

Configuration parameters directly affecting this product are *--dispersion*, *--flatfield*, *--skyalign*, *--startwavelength* and *--endwavelength*.

MAPPED_SCI_PMOS: FITS file with as many extensions as input scientific exposures, including images with rectified, wavelength calibrated and sky subtracted slit spectra from each input image. These images match in size the MAPPED_ALL_SCI_PMOS images, and are produced only if any kind of sky subtraction is requested.

Configuration parameters directly affecting this product are *--dispersion*, *--cosmics*, *--flatfield*, *--skyalign*, *--startwavelength* and *--endwavelength*.

Configuration parameters having significant impact are *--skymedian* and *--skylocal*.

MAPPED_SKY_SCI_PMOS: FITS file with as many extensions as input scientific exposures, including images with rectified and wavelength calibrated slit sky spectra. These images contain the modeled sky that was subtracted from the scientific data, either before or after the scientific spectra rectification. The sky model component subtracted before the rectification of the scientific spectra can be also viewed in the UNMAPPED_SKY_SCI_PMOS product.

The MAPPED_SKY_SCI_PMOS also includes the identified cosmic ray signal in case the cosmic rays removal is requested (see configuration parameter *--cosmics*, Section 9.10.3, page 124).

Configuration parameters directly affecting this product are *--skymedian*, *--skylocal*, *--cosmics*, *--startwavelength* and *--endwavelength*.

Configuration parameters having significant impact are *--skyalign*, and *--flatfield*.

OBJECT_TABLE_SCI_PMOS: FITS file with as many extensions as input scientific exposures, including tables which are an expansion of the input SLIT_LOCATION_PMOS table (see page 138), where the positions and the extraction spatial intervals of the detected object-spectra are also included. This file is only produced if any kind of sky subtraction is requested, otherwise no object-spectra detection or extraction is attempted. The object table columns are the following[54]:

slit_id:	Spectrum identification number. This column is uninteresting for slit identification, because the beam from each slit is split in two paths. Use column **pair_id** for slit identification.
pair_id:	Pair identification number. Each slit produces two slit spectra: the one on top is from the ordinary beam, the one on bottom is from the extraordinary beam.
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 the rectified images (MAPPED_ALL_SCI_PMOS and MAPPED_SCI_PMOS) 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.

[54] All pixel coordinates refer to actual pixels, i.e. for the default binning 2x2 one pixel refers to the original 2 along the spatial axis. Some data display tools (for instance skycat) instead keep the memory of the binning, so that the numbers in the table need to be multiplied by two to compare for instance the contents of the columns positions and length to the displayed image.

object_1, object_2, ...: Detected object-spectra positions in the rectified images.
start_1, start_2, ...: Start position of the extraction interval for each object-spectrum. This row is **not** included in the extracted spectrum.
end_1, end_2, ...: End position of the extraction interval for each object-spectrum. This row is included in the extracted spectrum.
row_1, row_2, ...: Row number of the REDUCED_SCI_PMOS image containing the extracted object-spectrum. Image rows are counted from bottom, starting from 0. The sources themselves are numbered from the top to the bottom of the detector. If more than one object per slit is detected, the object referenced in row_1 will be in the row above object in row_2, i.e., row_1 = row_2 + 1.

Configuration parameters directly affecting this product are *--slit_margin*, *--ext_radius*, *--cont_radius*.

Configuration parameters that may have significant impact are *--startwavelength* and *--endwavelength*.

OBJECT_TABLE_POL_SCI_PMOS: table with the same structure and same number of rows of a single OBJECT_TABLE_SCI_PMOS table extension. Just one extra set of coulms is added:

row_stokes_1, row_stokes_2, ...: Row number of the REDUCED_SCI_**X**_PMOS image containing the extracted object-source **X** polarisation parameter. Image rows are counted from bottom, starting from 0.

The content of the other columns is identical to the first table in the OBJECT_TABLE_SCI_PMOS file. This summary table may be preferred to that more extended (and complex) product, since the variation between different tables is generally negligible.

REDUCED_ERROR_SCI_PMOS: FITS file with as many extensions as input scientific exposures, including images with errors (one sigma level) corresponding to the extracted object-spectra. These images match the REDUCED_SCI_PMOS images.

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

Configuration parameters having significant impact are *--slit_margin*, *--ext_radius* and *--cont_radius*.

REDUCED_ERROR_X_SCI_PMOS: Image with errors (one sigma level) corresponding to the extracted polarisation signals from the object-sources in the corresponding REDUCED_X_SCI_PMOS images. X may be any of V (circular polarisation), Q, U (linear polarisation), L (total linear polarisation, i.e. the geometrical sum of the Q and U components), ANGLE (direction of the linear polarisation vector in degrees, according to the convention described in [38]), and finally I (flux in ADU/s). Note that V, Q, U and L are all normalized by I. REDUCED_ERROR_SCI_Q_PMOS, –U_PMOS, –L_PMOS, and –ANGLE_PMOS are produced when the observation was performed using the half-wave retarder plate, while REDUCED_ERROR_SCI_V_PMOS is produced when the observation was performed using the 1/4-wave retarder plate. REDUCE_ERROR_SCI_I_PMOS is always produced.

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

Configuration parameters having significant impact are *--slit_margin*, *--ext_radius* and *--cont_radius*.

REDUCED_NUL_X_SCI_PMOS: Image with "null" Stokes parameters corresponding to the extracted polarisation signals in the REDUCED_X_SCI_PMOS images. X may be any of V (circular polarisation), Q and U (linear polarisation). The "null" parameters provide an extra check on the quality of the extracted polarimetric signal (see [38] for details), and can only be computed under specific conditions. REDUCED_NUL_SCI_Q_PMOS and –U_PMOS are produced when the observation was performed using at least 8 positions of the half-wave retarder plate, while REDUCED_NUL_SCI_V_PMOS is produced when the observation was performed using at least 4 positions of the 1/4-wave retarder plate.

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

Configuration parameters having significant impact are *--slit_margin*, *--ext_radius* and *--cont_radius*.

REDUCED_SCI_PMOS: FITS file with as many extensions as input scientific exposures, including images with extracted object-spectra. These images have the same x size of the images with the extracted slit spectra, MAPPED_SCI_PMOS, and as many rows as the detected and extracted object-spectra. Extracted object-spectra are written to the image rows listed in the OBJECT_TABLE_SCI_PMOS table (columns row_XXX), which starts numbering from the top to the bottom of the detector.

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

Configuration parameters having significant impact are *--slit_margin*, *--ext_radius*, *--cont_radius*, *--skyalign*, *--flatfield*, *--skylocal*, *--skymedian*, and *--cosmics*.

REDUCED_X_SCI_PMOS: Image whose rows contain the extracted polarisation parameters for each object-source. These images have the same x size of the REDUCED_SCI_PMOS images, and half their y size, i.e., as many rows as the detected object-sources. X may be any of V (circular polarisation), Q, U (linear polarisation), L (total linear polarisation, i.e. the geometrical sum of the Q and U components), ANGLE (direction of the linear polarisation vector in degrees, according to the convention described in [38]), and finally I (flux in ADU/s). REDUCED_SCI_Q_PMOS, –U_PMOS, –L_PMOS, and –ANGLE_PMOS are produced when the observation was performed using the half-wave retarder plate, while REDUCED_SCI_V_PMOS is produced when the observation was performed using the 1/4-wave retarder plate. REDUCED_SCI_I_PMOS is always produced.

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

Configuration parameters having significant impact are *--slit_margin*, *--ext_radius*, *--cont_radius*, *--skyalign*, *--flatfield*, *--skylocal*, *--skymedian*, and *--cosmics*.

REDUCED_SKY_SCI_PMOS: FITS file with as many extensions as input scientific exposures, including 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. These images match the REDUCED_SCI_PMOS images.

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

Configuration parameters having significant impact are *--slit_margin*, *--ext_radius*, *--cont_radius*, *--skyalign*, *--flatfield*, *--skylocal* and *--skymedian*.

SKY_SHIFTS_SLIT_SCI_PMOS: FITS file with as many extensions as input scientific exposures, including tables containing the observed sky lines offsets that were used for adjusting the input wavelength solution. These tables are only produced if the sky lines alignment is requested (see configuration parameter *--skyalign*, Section 9.10.3, page 124). These tables have one row for each of the sky lines used for the alignment, and one column for each slit where sky lines could be detected.

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_PMOS: FITS file with as many extensions as input scientific exposures, including images 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 parameter *--skylocal*, Section 9.10.3, page 124). These images are derived subtracting the UNMAPPED_SKY_SCI_MXU images from the bias subtracted and flat fielded scientific frames.

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

UNMAPPED_SKY_SCI_PMOS: FITS file with as many extensions as input scientific exposures, including images which have the same size of the CCD, and created if the local sky subtraction is requested.

If *--skylocal* is set (see the configuration parameter *--skylocal*, Section 9.10.3, page 124), this image contains the sky model obtained by interpolating the sky signal trend along the spatial direction, directly on the CCD frame.

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

WAVELENGTH_MAP_SCI_PMOS: 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 9.10.3, page 124). For a description of this product see the WAVELENGTH_MAP_MXU entry on page 107. 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_PMOS is inherited from the arc lamp frame header, while the coordinate system of WAVELENGTH_MAP_SCI_PMOS is inherited from the scientific frame header.

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

9.14.3 Configuration parameters

The configuration parameters setting determines the way the *fors_pmos_science* recipe will process the input data, and to some extent the product files that will be created. The parameters are exactly the same as for the recipe *fors_science*, with the exception of *--skyglobal*, which is not supported. There are only four more parameters, which are described below:

--rebin: Rebin extracted spectra by an integer factor. *Default*: 1

This parameter will sum interval of N pixels, and derive the polarimetric signal from the rebinned frames. This should be preferred to the resampling performed by specifying a value for *--dispersion* greater than the one specified in *fors_pmos_calib*.

--chromatism: Apply chromatism correction to linear polarization angles. *Default*: true

If this parameter is set to "true", an input RETARDER_WAVEPLATE_CHROMATISM table must be specified in input.

--wollaston: Wollaston mounting. *Default*: true

This parameter is only active in case of FORS2 data (for FORS1 data it is frozen to "true"), and only for linear polarisation observations. The Wollaston (beam splitter) can be mounted at two different orientations, 180 degrees apart. The effect will be beam swapping: if this parameter is set to "true", the ordinary beam is on top; if "false", it is at bottom. Beam swapping causes a 90 degrees rotation of the linear polarisation vector.

9.14.4 Quality control parameters

Currently the following QC parameters, used by PSO and DPD, are evaluated by the *fors_pmos_science* recipe.

QC NULL X MEAN: Weighted average of "null" parameter associated to X Stokes parameter. *Units*: *none*

X may be any of V (circular polarisation), Q and U (linear polarisation). When the "null" images REDUCED_NUL_X_SCI_PMOS (page 141 can be produced, an average of these images, weighted on the intensity of the corresponding object-spectra (available from the REDUCED_SCI_PMOS images, see page 141) is computed and written to the corresponding QC parameter.

QC DET CHIP NUM: Chip number. *Units*: *none*

QC INS_COLL_NAME: Collimator name. *Units*: *none*

QC OBJECT_POSITION: Position of brightest object along the slit. *Units*: *pixel*

QC N_SATURATED: Number of saturated pixels. *Units*: *None*

QC N_SATURATED_FLAT: The QC N_SATURATED value from the MASTER_NORM_FLAT. *Units*: *None*

QC NUM_SLIT: Number of slits. *Units*: *None*

QC RET_ANGLE: Retarder angle. *Units*: *degrees*

QC AIRM MEAN: Average airmass.

Average of TEL.AIRM.START and TEL.AIRM.END

QC AVG_ACCURACY<i>_<j>: Mean of model accuracy. *Units*: pixel

Copied from the MASTER_NORM_FLAT.

QC SIG_ACCURACY<i>_<j>: Sigma of model accuracy. *Units*: pixel

Copied from the MASTER_NORM_FLAT.

The following parameters are only computed for standard star observations:

QC PMOS BAND: Photometric band where polarisation was measured. *Units*: *none*

This parameter indicates in what standard photometric band (U, B, V, R, I) the *fors_pmos_science* recipe measured the polarisation signal. The band is chosen as the closest to the center of the extracted standard star spectrum.

QC PMOS POLARISED: Polarisation is expected (1 = yes, 0 = no). *Units*: *none*

The standard star is not expected to be linearly polarised if this parameter is set to 0.

QC PMOS L OFFSET: Deviation from expected linear polarisation. *Units*: *none*

If QC.PMOS.POLARISED = 1, this parameter reports the linear polarisation *relative* offset, defined as

$$\frac{(L_{observed} - L_{expected})}{L_{expected}}$$

If QC.PMOS.POLARISED = 0, the linear polarisation *absolute* offset $(L_{observed} - L_{expected})$ is reported.

QC PMOS L OFFSETERR: Error on QC.PMOS.L.OFFSET. *Units*: *none*

This error is propagated from the measurement and the catalog errors.

QC PMOS ANGLE OFFSET: Polarisation angle offset *Units*: degree

Polarisation angle offset, defined as $(A_{observed} - A_{expected})$. This parameter is only computed and written if QC.PMOS.POLARISED = 1

QC PMOS ANGLE OFFSETERR: Error on polarisation angle offset *Units*: degree

This error is propagated from the measurement and the catalog errors. This parameter is only computed and written if QC.PMOS.POLARISED = 1

9.15 fors_molecfit_model

The recipe runs molecfit_model on the input spectrum. The accepted formats are defined by the following categories: REDUCED_IDP_SCI_LSS, REDUCED_IDP_SCI_MXU and REDUCED_IDP_SCI_MOS. If atmospheric profiles (GDAS and ATM_PROFILE_STANDARD) are given in the input SOF, then they will be used.

The products of *fors_molecfit_model* are used by *fors_molecfit_calctrans* to construct the full atmospheric transmission correction. The *fors_molecfit_correct* recipe can then apply the derived correction.

Please see the molecfit manual if you require more detailed information on molecfit parameters and usage.

Please be aware that the molecfit recipes are newly available in 2022 and unexpected behaviour may still be encountered while using these recipes. We encourage users to please report any unexpected behaviour by submitting a helpdesk ticket at *https://support.eso.org*.

9.15.1 Input files

In alphabetical order:

ATM_PROFILE_STANDARD: *optional* Fits table indicating the reference atmospheric profile. If not given, a default profile will be used.

GDAS: *optional* GDAS profile providing pressure [hPa], altitude [m], and temperature [K] to use. If not provided, the recipe will retrieve data from the ESO database.

KERNEL_LIBRARY: *optional* Library specifying the instrumental line spread function.

WAVE_EXCLUDE: *optional* Fits table specifying the wavelength boundaries of the regions to exclude in the fit. Ignored if the recipe parameter WAVE_EXCLUDE is not 'NULL'. If a region is outside the wavelength range of the input spectrum, it will be ignored.

WAVE_INCLUDE: *optional* Fits table specifying the wavelength boundaries of the regions to fit. Ignored if the recipe parameter WAVE_INCLUDE is not 'NULL'. If a region is outside the wavelength range of the input spectrum, it will be ignored.

MOLECULES: *optional* Fits table specifying the molecules to consider in the fit and their relative column densities. Ignored if all the recipe parameters LIST_MOLEC, FIT_MOLEC, and REL_COL are specified.

PIXEL_EXCLUDE: *optional* Fits table specifying the pixels boundaries of the regions to exclude in the fit. Ignored if the recipe parameter PIXEL_EXCLUDE is not 'NULL'. If a region is outside the wavelength range of the input spectrum, it will be ignored.

REDUCED_IDP_SCI_LSS: *required* The input spectrum to model. Choose one of these tags.

or REDUCED_IDP_SCI_MOS: Choose one of these tags.

or REDUCED_IDP_SCI_MXU: Choose one of these tags.

9.15.2 Output files

The possible output frames in alphabetical order are:

ATM_PARAMETERS: The atmospheric profile used, obtained as combination of the GDAS and ATM_PROFILE_STANDARD (either input or default).

ATM_PROFILE_COMBINED: The result of combining the GDAS, MIPAS and EMM data and represents the gridded atmospheric profile data to be used in the LNFL and LBLRTM third-party tools. Please refer to the molecfit manual for further information.

ATM_PROFILE_STANDARD: The MIPAS atmospheric composition data used with GDAS and EMM data to construct an atmospheric model. Please refer to the molecfit manual for further information.

BEST_FIT_MODEL: Table containing the best fit model. Please refer to the molecfit manual for further information.

BEST_FIT_PARAMETERS: Table containing the best fit parameters. Please refer to the molecfit manual for further information.

GDAS: Used GDAS profile (either copy of input, or computed from the ESO database).

GDAS_AFTER: The GDAS profile downloaded from the ESO database immediately after the MJD-OBS of the input spectrum. Produced only if GDAS was not provided in the input sof.

GDAS_BEFORE: The GDAS profile downloaded from the ESO database immediately before the MJD-OBS of the input spectrum. Produced only if GDAS was not provided in the input sof.

KERNEL_LIBRARY: Kernel library used in the fit (values are interpolated from the input library). Produced only if a KERNEL_LIBRARY was given as input.

MODEL_MOLECULES: Fits table containing the information on the fitted molecules.

MOLECFIT_DATA: The input spectrum saved as the molecfit internal fits binary table format.

9.15.3 Configuration parameters

The following configuration parameters are available for *fors_molecfit_model*. The parameters are only a subset of the typical molecfit_model parameters. Please refer to the molecfit manual for further information.

Important note on default behaviour: If *fors_molecfit_model* is run with the default parameters, such that insufficient information is provided to determine the wavelength ranges to fit, then the following parameters are overridden internally.[55] Not all wavelength ranges specified by *--WAVE_INCLUDE* need to overlap the input spectrum, but for each input spectrum at least one range must overlap for *fors_molecfit_model* to run.

--FIT_MOLEC: 1,1

--LIST_MOLEC: H2O,O2

--REL_COL: 1,1.06

--WAVE_INCLUDE: 0.61,0.64,0.68,0.71,0.711,0.74,0.75,0.78,0.81,0.84,0.91,0.95,0.96,0.98

The default parameters in alphabetical order are:

--COLUMN_DFLUX: Name of the column in the input that identifies the flux error. Note: this parameter is relevant only for inputs in binary table format. This option may also be set using the environment variable COLUMN_DFLUX. *Default:* ERR

--COLUMN_FLUX: Name of the column in the input that identifies the flux. Note: this parameter is relevant only for inputs in binary table format. This option may also be set using the environment variable COLUMN_FLUX. *Default:* FLUX

[55] Ordinarily these values would be the default for the following parameters, however 'NULL' is currently set as the default to facilitate the usage of the relevant input files in the sof.

--*COLUMN_LAMBDA:* Name of the column in the input that identifies the wavelength. Note: this parameter is relevant only for inputs in binary table format. This option may also be set using the environment variable COLUMN_LAMBDA. *Default:* WAVE

--*CONTINUUM_N:* Polynomial order for the continuum model fit to each wavelength region, presented as a comma delimited string. If a single value is given, then it is assumed to be valid for all the wavelength ranges. If set to 'NULL', then the values are taken from the column CONT_POLY_ORDER of the input table WAVE_INCLUDE. If this file is not given, then the default is used. This option may also be set using the environment variable CONTINUUM_N. *Default:* 1

--*DEFAULT_ERROR:* Default error relative to mean for the case that the error column is not provided. This option may also be set using the environment variable DEFAULT_ERROR. *Default:* 0.01

--*FIT_CONTINUUM:* Comma delimited string of flags (1=true, 0=false) for fitting the continuum in specific regions. The number of values must match the number of wavelength ranges to fit. If one single value is given, then it is assumed to be valid for all the wavelength ranges. If set to 'NULL', then the values are taken from the column CONT_POLY_ORDER of the input table WAVE_INCLUDE. If this file is not given, then the default is used. This option may also be set using the environment variable FIT_CONTINUUM. *Default:* 1

--*FIT_MOLEC:* List of flags (1 and 0), comma separated, that specifies if a molecule has to be fitted (flag=1) or computed (flag=0). If set to 'NULL', the values provided in MOLECULES. If not, the provided list overrides what is specified in the MOLECULES input. Note: in order to override MOLECULES, the following recipe parameters rel_col, fit_molec, and list_molec must be all specified and different from 'NULL'. This option may also be set using the environment variable FIT_MOLEC. *Default:* NULL

--*FIT_RES_BOX:* Flag that specifies if the instrumental line spread function is fitted by a Boxcar function. This option may also be set using the environment variable FIT_RES_BOX. *Default:* FALSE

--*FIT_RES_GAUSS:* Flag that specifies if the instrumental line spread function is fitted by a Gaussian function. This option may also be set using the environment variable FIT_RES_GAUSS. *Default:* TRUE

--*FIT_RES_LORENTZ:* Flag that specifies if the instrumental line spread function is fitted by a Lorentzian function. This option may also be set using the environment variable FIT_RES_LORENTZ. *Default:* FALSE

--*FIT_TELESCOPE_BACKGROUND:* Flag that indicates whether the telescope background should be fitted. This option may also be set using the environment variable FIT_TELESCOPE_BACKGROUND. *Default:* FALSE

--*FIT_WLC:* A list of flags (1 or 0) that specifies which ranges are to be included as a part of the wavelength correction fitting if wavelength fitting has been selected. If a single value is given, then it is assumed to be valid for all the wavelength ranges. This option may also be set using the environment variable FIT_WLC. *Default:* 1

--*FTOL:* Relative chi2 convergence criterion. This option may also be set using the environment variable FTOL. *Default:* 0.001

--*KERNFAC:* Size of the Gaussian/Lorentzian/Voigtian kernel, expressed in units of FWHM. This option may also be set using the environment variable KERNFAC. *Default:* 3.0

--*KERNMODE:* Flag that indicates whether to use a Voigt approximation instead of the Lorentz or Gauss functions to fit the instrumental line spread function. This option may also be set using the environment variable KERNMODE. *Default:* FALSE

--*LIST_MOLEC:* List of molecules to include in the fit (comma separated). If set to 'NULL', the values provided in MOLECULES. If not, the provided list overrides what is specified in the MOLECULES input. Note: in order to override MOLECULES, the recipe parameters REL_COL, FIT_MOLEC, and LIST_MOLEC must be all specified and different from 'NULL'. This option may also be set using the environment variable LIST_MOLEC. *Default:* NULL

--*LNFL_LINE_DB:* AER version in format aer_v_X.X. For example aer_v_3.8 or aer_v_3.6. This option may also be set using the environment variable LNFL_LINE_DB. *Default:* aer_v_3.8.1.2

--*PIXEL_EXCLUDE:* Comma separated list of values that define the boundaries of the pixel regions to exclude in the fit (low_1, upper_1, low_2, upper_2,...,low_n, upper_n), in pixel units. If set to 'NULL' the values specified in the fits table given by the tag PIXEL_EXCLUDE. If not, these values override those specified in PIXEL_EXCLUDE. This option may also be set using the environment variable PIXEL_EXCLUDE. *Default:* NULL

--*PWV:* Value in mm of the precipitable water vapour for the input water vapor profile. If set to a positive value, then the merged profile composed of ref_atm, GDAS, and local meteorological data will scaled to this value. If negative, then no scaling is done. This option may also be set using the environment variable PWV. *Default:* -1.0

--*REL_COL:* List of relative column densities of the molecules used (comma separated). If set to 'NULL', the values provided in MOLECULES. If not, the provided list overrides what is specified in the MOLECULES input. Note: in order to override MOLECULES, the following recipe parameters rel_col, fit_molec, and list_molec must be all specified and different from 'NULL'. This option may also be set using the environment variable REL_COL. *Default:* NULL

--*RES_BOX:* Initial value in pixels of the Boxcar function width that fits the instrumental line spread function (only used if –FIT_RES_BOX=TRUE). This option may also be set using the environment variable RES_BOX. *Default:* 1.0

--*RES_GAUSS:* Inital value in pixels of the FWHM of the Gaussian function that fits the instrumental line spread function (only used if –FIT_RES_GAUSS=TRUE). This option may also be set using the environment variable RES_GAUSS. *Default:* 1.0

--*RES_LORENTZ:* Initial value in pixels of the FWHM of the Lorentz function that fits the instrumental line spread function (only used if –FIT_RES_LORENTZ=TRUE). This option may also be set using the environment variable RES_LORENTZ. *Default:* 2.0

--*TELESCOPE_BACKGROUND_CONST:* Initial value for the telescope background fit. This option may also be set using the environment variable TELESCOPE_BACKGROUND_CONST. *Default:* 0.1

--*USE_INPUT_KERNEL:* If TRUE, use the kernel library if it is provided. This option may also be set using the environment variable USE_INPUT_KERNEL. *Default:* TRUE

--*VARKERN:* Flag indicating if the kernel is constant or varies linearly with wavelength (i.e. resolving power is constant). This option may also be set using the environment variable VARKERN. *Default:* TRUE

--*WAVE_EXCLUDE:* Comma separated list of values that define the boundaries of the wavelength regions to exclude in the fit (low_1, upper_1, low_2, upper_2,...,low_n, upper_n), in microns. If set to 'NULL' the values specified in the fits table given by the tag WAVE_EXCLUDE. If not, these values override those specified in WAVE_EXCLUDE. This option may also be set using the environment variable WAVE_EXCLUDE. *Default:* NULL

--*WAVE_INCLUDE:* Comma separated list of values that define the boundaries of the wavelength regions to fit (low_1, upper_1, low_2, upper_2,...,low_n, upper_n), in microns. If set to 'NULL' the values specified in the fits table given by the tag WAVE_INCLUDE. If not, these values override those specified in WAVE_INCLUDE. This option may also be set using the environment variable WAVE_INCLUDE. *Default:* NULL

--*WLC_CONST:* Initial term for refinement of the wavelength solution. This option may also be set using the environment variable WLC_CONST. *Default:* 0.0

--*WLC_N:* Degree of the polynomial used to refine the wavelength solution. This option may also be set using the environment variable WLC_N. *Default:* 1

--*XTOL:* Relative parameter convergence criterion. This option may also be set using the environment variable XTOL. *Default:* 0.001

9.16 fors_molecfit_calctrans

The recipe accepts as input an atmosperic model and instrument configuration as defined in the inputs produced by *fors_molecfit_model* and creates the full atmospheric transmission function correction suitable for the input spectrum. The accepted format of the input spectrum is a 1D binary fits table as one of REDUCED_IDP_SCI_LSS, REDUCED_IDP_SCI_MXU or REDUCED_IDP_SCI_MOS.

The products of *fors_molecfit_calctrans* are used by *fors_molecfit_correct* to apply the derived correction.

Please see the molecfit manual if you require more detailed information on molecfit parameters and usage.

Please be aware that the molecfit recipes are newly available in 2022 and unexpected behaviour may still be encountered while using these recipes. We encourage users to please report any unexpected behaviour by submitting a helpdesk ticket at *https://support.eso.org*.

9.16.1 Input files

In alphabetical order:

ATM_PARAMETERS: *required* Description of the atmospheric parameters as produced by *fors_molecfit_model*.

BEST_FIT_PARAMETERS: *required* Best fit parameters as produced by *fors_molecfit_model*.

KERNEL_LIBRARY: *optional* Kernel library used in *fors_molecfit_model*. If not provided, or if USE_INPUT_KERNEL=false, then the information in BEST_FIT_PARAMETERS will be used.

MODEL_MOLECULES: *required* Best fit parameters of the fitted molecules as produced by *fors_molecfit_model*.

REDUCED_IDP_SCI_LSS: *required* The input spectrum to compute the transmission correction for. Choose one of these tags.

or REDUCED_IDP_SCI_MOS: Choose one of these tags.

or REDUCED_IDP_SCI_MXU: Choose one of these tags.

9.16.2 Output files

The possible output frames in alphabetical order are:

CALCTRANS_KERNEL_LIBRARY: The kernel used by *fors_molecfit_calctrans*.

LBLRTM_RESULTS: Results of the LBLRTM function.

TELLURIC_CORR: Binary fits table containing the atmospheric transmission function. The columns contain 1 raw vector with names 'WAVE' and 'TRANSMISSION'.

9.16.3 Configuration parameters

A single option is provided for this recipe.

Please refer to the molecfit manual for further information.

--USE_INPUT_KERNEL: If TRUE, then the input KERNEL_LIBRARY given in the SOF is used. If FALSE, or if the KERNEL_LIBRARY is not given, then the information stored in BEST_FIT_PARAMETERS will be used to compute the line spread function. This option may also be set using the environment variable USE_INPUT_KERNEL. *Default:* True

9.17 fors_molecfit_correct

The recipe divides the input spectrum by an atmospheric transmission function to correct for telluric absorption features. The product has the same format and extensions as the input. The recipe currently accepts only 1D (IDP format) spectra. The science spectrum must also currently have the same wavelength scale and dispersion as the telluric correction determined by *fors_molecfit_calctrans*.

Please see the molecfit manual if you require more detailed information on molecfit parameters and usage.

Please be aware that the molecfit recipes are newly available in 2022 and unexpected behaviour may still be encountered while using these recipes. We encourage users to please report any unexpected behaviour by submitting a helpdesk ticket at *https://support.eso.org*.

9.17.1 Input files

In alphabetical order:

TELLURIC_CORR: *required* The telluric correction to be applied produced by *fors_molecfit_calctrans*.

REDUCED_IDP_SCI_LSS: *required* The 1D fits binary table input spectrum to be corrected. Choose one of these tags.

or REDUCED_IDP_SCI_MOS: Choose one of these tags.

or REDUCED_IDP_SCI_MXU: Choose one of these tags.

9.17.2 Output files

The possible output frames in alphabetical order are:

SCIENCE_TELLURIC_CORR_<input_pro_catg>: The input science frame corrected for telluric contamination. Please see the molecfit_correct section of the molecfit manual for more information on the output products.

SPECTRUM_TELLURIC_CORR_<input_pro_catg>: The telluric corrected spectrum as a single lambda, flux binary table. Please see the molecfit_correct section of the molecfit manual for more information on the output products.

9.17.3 Configuration parameters

The following options are provided for this recipe.

Please refer to the molecfit manual for further information.

--*COLUMN_DFLUX:* In the case of fits binary science input: name of the column in the input that identifies the flux errors. This option may also be set using the environment variable COLUMN_DFLUX. *Default:* ERR

--*COLUMN_FLUX:* In the case of fits binary science input: name of the column in the input that identifies the flux. This option may also be set using the environment variable COLUMN_FLUX. *Default:* FLUX

--*COLUMN_WAVE:* In the case of fits binary science input: name of the column in the input that identifies the wavelength. This option may also be set using the environment variable COLUMN_WAVE. *Default:* WAVE

--*THRESHOLD:* Use this value when the transmission function is lower than the specified threshold. This option may also be set using the environment variable THRESHOLD. *Default:* 0.01

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 Error propagation

10.1.1 Basics of error propagation

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 $\mathbf{J}$ 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:

$$\mathbf{G} = \mathbf{JCJ^T}$$

The covariance matrix $\mathbf{G}$ 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.[56]

Note that in the special case $\mathbf{C}$ were diagonal, $\mathbf{G}$ 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 9.1, page 88, 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

$$\begin{aligned} (\Delta F_1)^2 &= \left(\frac{\partial F_1}{\partial a_1}\right)(\Delta a_1)^2 + \left(\frac{\partial F_1}{\partial m}\right)(\Delta m)^2 \\ (\Delta F_2)^2 &= \left(\frac{\partial F_2}{\partial a_2}\right)(\Delta a_2)^2 + \left(\frac{\partial F_2}{\partial m}\right)(\Delta m)^2 \end{aligned}$$

[56] 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.

10.1.2 Error propagation of stacked frames

When several frames are combined, the final error of the stacked image depends on the method used to combine. The errors in the stacked frames are computed on a per pixel basis. Let's call σ_i the error of the individual frame i to stack. In the different cases the error in the combined frame will be:

- average.

$$\sigma^2_{\rm av} = \sum_{i=1}^{N} \frac{\sigma_i^2}{N}. \quad (1)$$

- weighted average

$$\frac{1}{\sigma^2_{\rm av}} = \sum_{i=1}^{N} \frac{1}{\sigma_i^2}. \quad (2)$$

- median

$$\sigma^2_{\rm med} = \frac{\pi}{2}\sigma^2_{\rm av} = \sum_{i=1}^{N} \frac{\pi\sigma_i^2}{2N}. \quad (3)$$

- minmax and kappa sigma clipping. The error is computed as in the average frame but with the remaining pixels after the rejection has been performed.

10.1.3 Propagation of photonic and readout noise in raw images

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 10.1.1, page 152 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 (taken from the bias overscan), 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

$$\Delta S = \sqrt{\frac{\sum_i (\Delta S_i)^2}{N}}$$

where N is the number of non-rejected values. In the case of median stacking the error is computed as

$$\Delta S = f(N)\sqrt{\frac{\sum_i (\Delta S_i)^2}{N}}$$

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 propagated through the full reduction chain using the mentioned error propagation formulas, including bias and flat. There are however two steps where errors are not taking into account: sky subtraction and response computation (and therefore also photometric correction).

The variance of the optimally extracted object spectra is determined by propagating the errors in the individual 2-D pixels with the weights provided by the Horne algorith ([34]). In the case of PMOS, such error is further propagated to the measured polarimetric parameters derived by the appropriate combination of extracted spectra.

10.2 Frame zeropoint computation

Here we describe how observed standard star positions are matched to the photometric catalogue provided (Sect. 10.2.1) and how the weights for the optimal determination of the frame zeropoint are derived (Sect. 10.2.2).

10.2.1 Robust shift determination

Standard star identification in a standard star image is implemented as the following algorithm in the recipe *fors_zeropoint*

1. Reads in the Stetson standard star catalogue and converts the standard star coordinates to image pixel coordinates using the WCS in the image header.
2. Extracts the subset of standard stars that are expected to lie in the field-of-view of the image under consideration. If there are no such standard stars, then a failure is declared.
3. Performs object detection and analysis in the image using `SExtractor`.
4. For each standard star in the subset of relevant standard stars, the algorithm calculates x and y pixel offsets to all detected objects in the image.
5. Constructs a two-dimensional histogram of the x and y pixel offsets calculated for all standard stars in step (4). The bin size is set to one square image pixel, and the histogram is constructed over the domain $-150 < dx < 150$ pix and $-150 < dy < 150$ pix, since we do not expect a systematic offset between the image WCS and the catalogue WCS of more than $\sim$50 pix.
6. Calculates the maximum value of the histogram. If the maximum value is zero because all the histogram bin values are zero, then a failure is declared.
7. Determines the number of peaks in the histogram that attain the maximum histogram value. We define such a maximal-peak as a set of spatially connected histogram bins that attain the maximum histogram value.

8. If there is more than one maximal-peak in the histogram, then the histogram is recalculated with a bin size of nine square image pixels (i.e. it is binned 3×3), and steps (6) and (7) are repeated. If there is still more than one maximal-peak in the histogram, then a failure is declared.

9. Calculates the centroid of the unique maximal-peak in the histogram using a 3×3 box centred on the maximal-peak. The centroid coordinates in the histogram are adopted as the estimate of the offset (pix) between the image WCS and the catalogue WCS.

10. Applies the offset derived in step (9) to the image pixel coordinates of the subset of relevant standard stars. Then, for each of these standard stars, the algorithm selects the closest detected object in the image with a centroid within 5 pix of the corrected standard star coordinates as the object corresponding to the standard star in question, with the possibility that no such detected object exists. If this results in no standard star matches with detected objects, then a failure is declared.

Any non-negligible ($> 0.1\ degree$) rotation between the image and catalogue WCS will most likely cause the shift determination algorithm to fail or to derive a spurious shift.

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. It may appear puzzling that some of the optimal weights (listed in column **weight** of the ALIGNED_PHOT table described in Sect. 9.7.2, p. 89) 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.

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

$$\omega_j = \sum_i C^{-1}_{ij}$$

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 9.8, page 92, the general model for the corrected instrumental magnitude m_{ij} of star i on exposure j is

$$M_i - \Gamma \cdot C_i + p(A_j, C_i) + E_j A_j - Z + f(x_{ij}, y_{ij}) = m_{ij}$$

where the symbols are consistent with table 9.1 (page 88), 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 [41])

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

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 E_j are constrained to be equal to the same (unknown) value E and all other parameters are assumed (catalog magnitudes, instrument zeropoint Z, linear color correction term Γ), and where the non-linear dependencies $p()$ and $f()$ are ignored, the equations would become:

$$E = \frac{1}{A_j}(m_{ij} - M_i + \Gamma \cdot C_i + Z)$$

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 catalog 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

$$\mathbf{x} = (\mathbf{A}^\mathbf{T}\mathbf{C}^{-1}\mathbf{A})^{-1}\mathbf{A}^\mathbf{T}\mathbf{C}^{-1}\mathbf{b}$$

and is obtained by Cholesky-decomposition of $\mathbf{A}^\mathbf{T}\mathbf{C}^{-1}\mathbf{A}$ [57]. 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 $\mathbf{x}$. $(\mathbf{A}^\mathbf{T}\mathbf{C}^{-1}\mathbf{A})^{-1}$ is the covariance matrix of $\mathbf{x}$, $\mathbf{C}_x$: in the transformation from $\mathbf{b}$ to $\mathbf{x}$

$$\mathbf{J} = (\mathbf{A}^\mathbf{T}\mathbf{C}^{-1}\mathbf{A})^{-1}\mathbf{A}^\mathbf{T}\mathbf{C}^{-1}$$

is clearly 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 10.1.1, page 152) leads to:

$$\begin{aligned}
\mathbf{C}_x &= \mathbf{J}\mathbf{C}\mathbf{J}^\mathbf{T} \\
&= (\mathbf{A}^\mathbf{T}\mathbf{C}^{-1}\mathbf{A})^{-1}\mathbf{A}^\mathbf{T}\mathbf{C}^{-1}\mathbf{C}((\mathbf{A}^\mathbf{T}\mathbf{C}^{-1}\mathbf{A})^{-1}\mathbf{A}^\mathbf{T}\mathbf{C}^{-1})^\mathbf{T} \\
&= (\mathbf{A}^\mathbf{T}\mathbf{C}^{-1}\mathbf{A})^{-1}
\end{aligned}$$

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 [42].

Naturally, it may happen that the equation system remains underdetermined — trying to fit an atmospheric extinction coefficient E_j for each single frame when also Z is left free would be one of such cases, because the terms $E_j A_j$ would degenerate with Z.

[57] In [41] singular value decomposition (SVD) is used

10.4 Determination of spectral response and efficiency curves

The efficiency of an instrument is defined as the ratio between detected photons and incoming photons, and can be derived by comparing the observed fluxes with the tabulated fluxes of known objects.

The spectral efficiency can be obtained with the recipe *fors_science* (see Section 9.10, page 116) applied to a standard star observation.

The extracted standard star spectrum, $X(\lambda)$ is converted into e^- s^{-1} Å^{-1}:

$$S(\lambda) = \frac{g\,X(\lambda)}{t\,\Delta\lambda}$$

where g is the gain factor in e$^-$ / ADU, t the exposure time in seconds, and $\Delta\lambda$ the constant wavelength step at which the spectrum was resampled after its calibration in wavelength.

The magnitude losses $\Delta m(\lambda)$ listed in the column `EXTINCTION` of the atmospheric extinction table (see Section 7.8, page 62) are turned into flux losses, and applied to the observed spectrum:

$$S_o(\lambda) = S(\lambda)\ 10^{0.4\ a\ \Delta m(\lambda)}$$

where $S_o(\lambda)$ is the spectrum at airmass zero and a is the airmass of the standard star observation. The values of the atmospheric extinction are linearly interpolated from the tabulated values for all the wavelength of the observed spectrum. At those wavelengths where no atmospheric extinction data are available, $S_o(\lambda)$ is set to zero.

The standard star catalog fluxes $C(\lambda)$, given in erg cm^{-2} s^{-1} Å^{-1} 10^{-16} in the STD_FLUX_TABLE (see Section 7.9, page 62), are converted into photons collected by the telescope using

$$F(\lambda) = C(\lambda)\ \frac{A_t}{h\nu}$$

Considering $A_t = 5.18\ 10^5$ cm^2 the VLT efficient area, and $hc = 1.98\ 10^{-8}$ erg Å, one derives

$$F(\lambda) = 2.6\ 10^{-3}\ C(\lambda)\ \lambda$$

(expressed in photons s^{-1} Å^{-1}, as the 10^{-16} factor is included in the numeric part).

The efficiency is finally computed as

$$E(\lambda) = \frac{S_o(\lambda)}{F(\lambda)}$$

(electrons per photon).

The efficiency $E(\lambda)$ is set to zero at those wavelengths where no standard star catalog fluxes are available, and is written to the column `RAW_EFFICIENCY` of the output spectro-photometric table (see Table 9.10.2, page 123).

The response curve used in the flux calibration of observed scientific spectra (see next Section) is obtained by:

$$R(\lambda) = \frac{C(\lambda)}{S_o(\lambda)}$$

$R(\lambda)$ is set to zero where no standard star catalog fluxes are available, and is written to the column `RAW_RESPONSE` of the output spectro-photometric table.

As a final step either a cubic spline fit or a polynomial fit performed, in order to derive a smoothed version of the curves. If parameter `resp_fit_nknots` is greater than 0, then a spline fitting is performed with the parameter specifiyng the number of knots. If parameter `resp_fit_degree` is greater than 0, a polynomial fit is performed with the parameter specifiyng the polynomial degree. The two options are mutually exclusive. The fit is performed on the response points that have not been excluded from parameters `resp_ignore_lines` and `resp_ignore_range`. Additionally, all points below 0.1% of the mean value of the response are ignored. If the number of remaining points is less than `nknots+2` in the case of spline fitting or less than `degree+1` in the case of polynomial fitting, the fitting algorithm parameters will be adjusted.

The fitted curves are written to the output spectro-photometric table, in columns `EFFICIENCY` and `RESPONSE`.

Some grisms, notably the holographic grisms, have different responses depending on the incident angle. That means that slits placed at different locations will show different behaviour. It has been demonstrated that this can be corrected by first dividing each spectrum by the average spectral energy distribution of the corresponding slit flat. The fors_science recipe supports this mode if the input frameset contains the FLAT_SED_MXU file. Both during response computation, and spectrophotometric calibration, the extracted spectrum will be divided by the corresponding flat field spectral energy distribution. Obviously, this will work only if this is done in both steps. The recipe will issue a warning if one tries to calibrate spectrophotometrically a spectrum corrected with a flat field sed with a response that does not include the correction or vice versa. Note that the [SPECPHOT_TABLE] product contains the keyword **QC RESP FLAT_SED_CORR** that is set to true if the response contains the flat sed correction. The flat sed correction normalisation factor is recorded in the keyword **QC RESP FLAT_SED_NORM**.

Due to the variations of the SED of the blue flat field lamp the flat field SED correction for GRIS_1200B was set to false in the GRISM_TABLE, because the errors due to the variation of the response with position are generally smaller than the errors caused by the flat field variability.

10.5 Spectrophotometric calibration

Scientific spectra extracted by the recipe *fors_science* can be calibrated in flux by specifying an atmospheric extinction table (see Section 7.8, page 62), and an appropriate spectro-photometric table (see SPECPHOT_TABLE, page 123). The spectro-photometric table can be produced by the recipe *fors_science* itself (see Section 9.10, page 116).

As explained in 10.4, the spectrophotometric calibration can also be previously corrected from the flat field Spectral Energy Distribution (SED). This is needed for grisms that show a position-dependant response, like the holographic ones. If the FLAT_SED_MXU file is part of the input frameset *and* was used to create the response curve, the extracted science spectra will be divided by the flat field sed during flux calibration. This correction should not be applied to non-holographic grisms, as it is not required and can cause problems in the case of unstable flat field lamps.

10.5.1 Master response curves

A MASTER_SPECPHOT_TABLE may be specified instead of the SPECPHOT_TABLE. These files were created to improve the robustness of automatic processing during which the quality of the individual response curves (and their fits) cannot be controlled. For most grisms MASTER_SPECPHOT_TABLEs exist and are currently applicable to data observed after January 1, 2015, with the MIT mosaic detector. In order to get a better spectrophotometric correction in the blue part the flat fields used for GRIS_1200B and GRIS_600B data entering the MASTER_SPECPHOT_TABLE were smoothed with `dradius_aver = 50`, so that the small-scale variations in the blue part are still present in the normalized flat fields and thus corrected in the observed spectra. The pipeline requires the same value of `dradius_aver = 50` for the flat fields used for the science data in order to use the MASTER_SPECPHOT_TABLE. Due to the variations of the SED of the blue flat field lamp the MASTER_SPECPHOT_TABLE for GRIS_1200B was determined for different slit positions, without applying the flat field SED correction. The GRISM_TABLE has been updated accordingly.

10.6 Overview of the spectral self-calibration procedure

Here we describe the instrument-independent tasks involved in the complete spectral self-calibration procedure (wavelength calibration and distortion correction), starting with an overview to set the individual tasks in their appropriate context.

1. Retrieve from the reference arc lamp line catalog the line pattern to be searched on arc lamp exposures.
2. After overscan and bias 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 10.1). Because the pattern recognition relies on relative line positions it is robust against shifts in the spectra introduced by different slit positions.
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.
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.

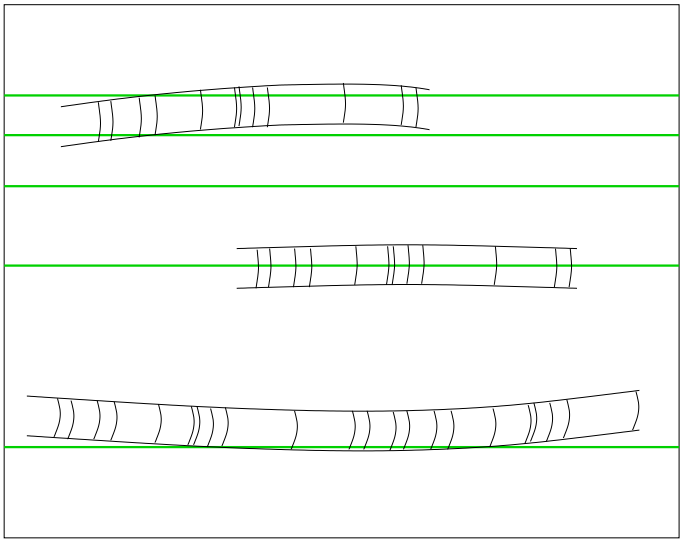

Figure 10.1: *CCD rows may not cut the whole range of the raw arc lamp spectra, because the spectra are not read along their curvature. However, even incomplete portions of the searched pattern can be identified by the pattern-matching algorithm.*

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 catalog, 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.[58]

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 location 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.

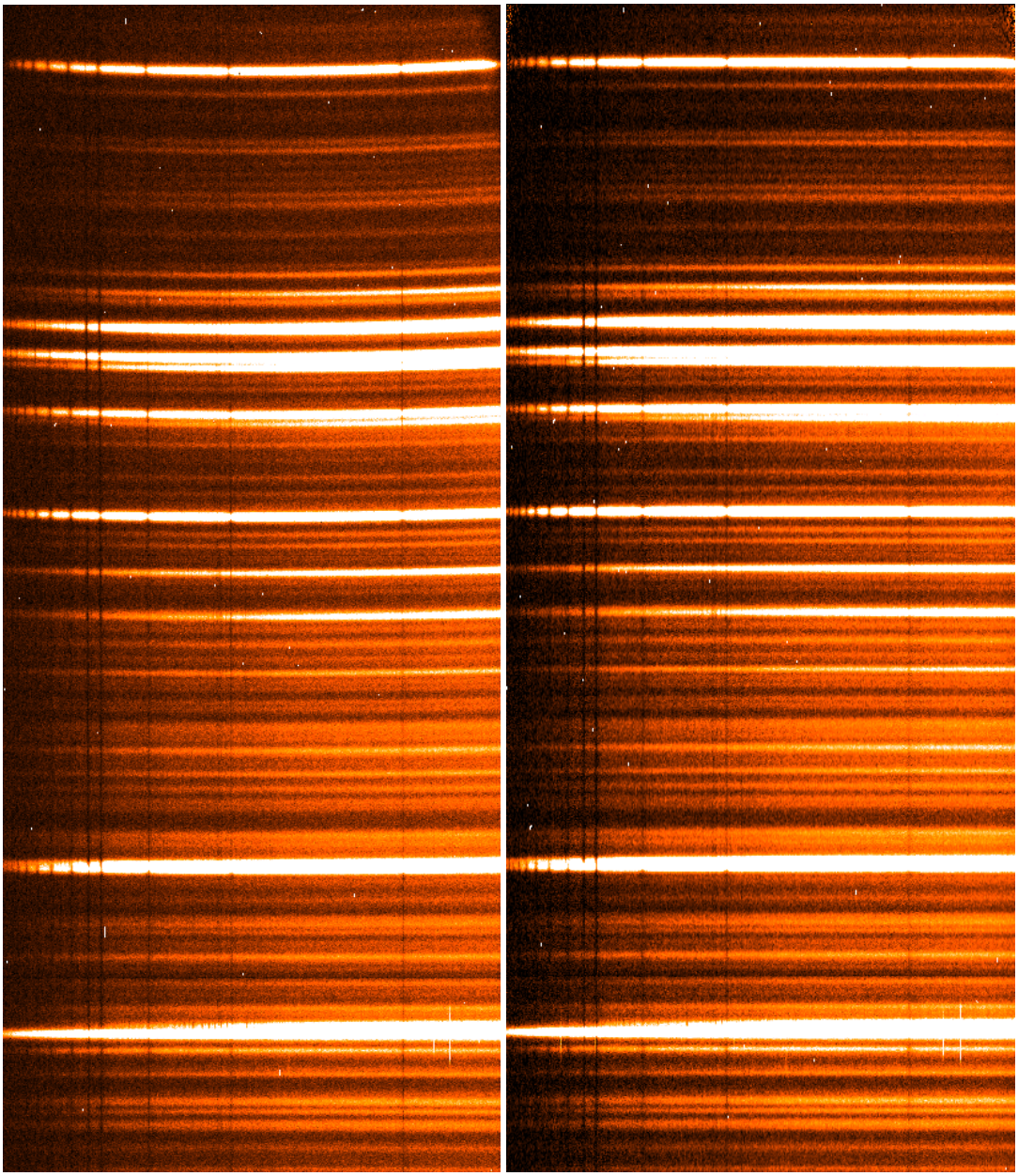

Figure 10.2: *Here we show LSS raw data (left) and the distortion corrected* `MAPPED` *data. After the distortion correction the previously curved spectra are now well aligned with horizontal wavelength axis.*

10.6.1 LSS distortion corrections and source extraction

For LSS or LSS-like spectra, the images can be corrected for spectral distortion using the GLOBAL_DISTORTION_TABLE as an input of the fors_science recipe.

[58] This is the local wavelength calibration.

The spectra is using the information of the first slit defined in the *SLIT_LOCATION_MODE* input of the *fors_science* recipe.

10.6.2 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*.[59]

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.

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 10.3a).

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 10.3b). 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 10.6.3, page 168).

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:

$$R = \frac{1}{2}\left(\frac{v_1 - v_{-1}}{2v_o - v_1 - v_{-1}}\right)$$

[59] Or at least long uninterruped portions of it.

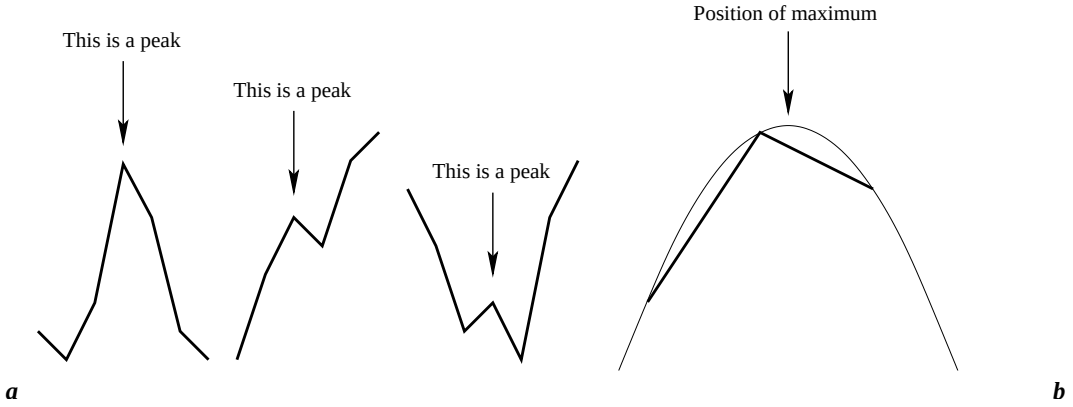

Figure 10.3: *a) Any local maximum identifies a peak. b) A peak position is determined by parabolic interpolation of three pixel values about the local maximum.*

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).

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).[60]

10.6.3 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 catalog. The line catalog should just include lines that are expected somewhere in the CCD exposure of the calibration lamp.[61]

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 10.6.2, page 167). 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 catalog. 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.[62]

[60] 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.

[61] The line catalog represents the pattern that should be searched on the CCD, and adding extra lines would destroy this pattern. Note, however, that a catalog including extra lines at its blue and/or red ends is still allowed.

[62] 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.

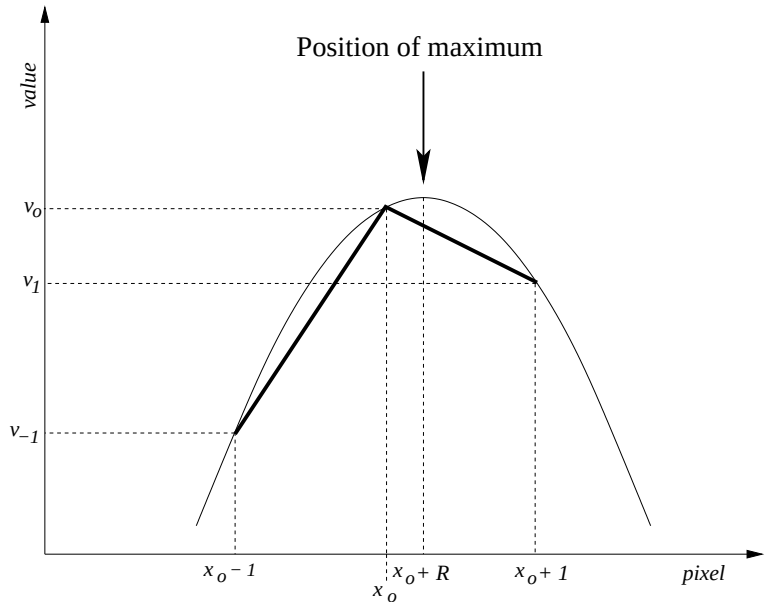

Figure 10.4: *Peak position estimate.*

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 catalog.

N: the number of detected peaks.[63]

λ_i: the i-th wavelength of the input line catalog, 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

[63] 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 10.5):

$$p_{min} = p_j + \frac{\lambda_{i+1} - \lambda_i}{d + \Delta d}$$

$$p_{max} = p_j + \frac{\lambda_{i+1} - \lambda_i}{d - \Delta d}$$

A *backward* search interval is similarly defined:

$$p_{min} = p_j - \frac{\lambda_i - \lambda_{i-1}}{d - \Delta d}$$

$$p_{max} = p_j - \frac{\lambda_i - \lambda_{i-1}}{d + \Delta d}$$

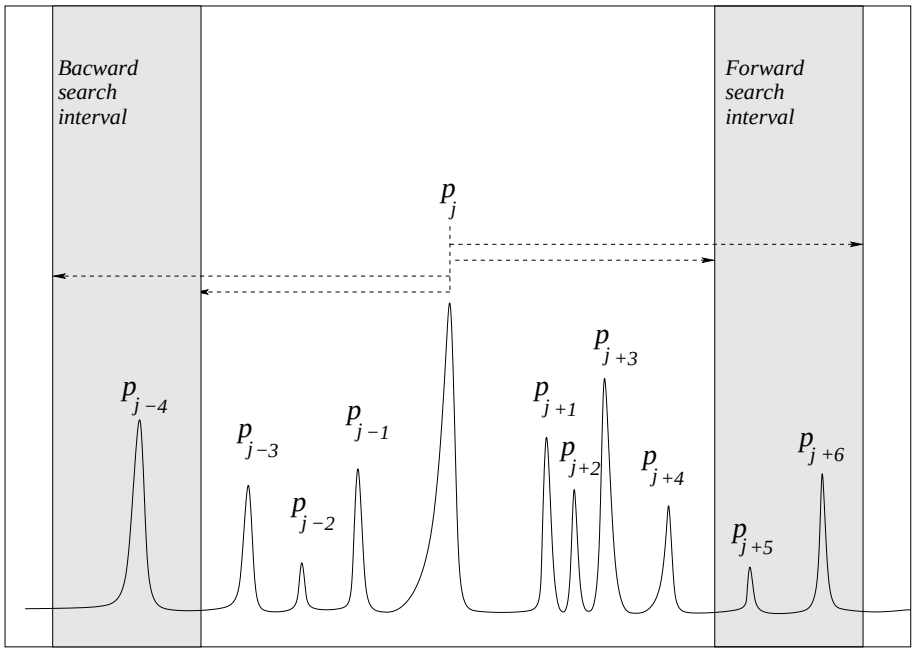

Figure 10.5: *Given a peak p_j, a forward and a backward search intervals compatible with an expected value of the spectral dispersion are defined. In this picture, peak positions p_{j-4}, p_{j+5} and p_{j+6} are used in the computation of the distance ratios to be compared with the wavelength interval ratio R_i. This process is repeated for each catalog 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,[64] the three peak

[64]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.[65] 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 catalog are correctly identified without relying on a pre-existing instrument distortion modeling.

10.6.4 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 59).

10.6.5 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.

The best choice is the undeviated wavelength, i.e., the wavelength that would fall at the same position of the CCD if the instrument is operated in imaging mode. For FORS data, the keyword ESO INS GRIS1 WLEN usually contains this value. Alternatively, 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.

10.6.6 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 164) 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

[65] 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.

positions of the slit centers on the mask plane in the case the reference wavelength is the undeviated wavelength. The match between the two sets is determined by the 2D pattern-recognition task (see next Section).

10.6.7 2D pattern recognition based on similar triangles

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.[66]

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,[67] 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 10.6). 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.[68] 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.[69] 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

[66] 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.

[67] Imposing a reading order to the triangle sides eliminates the reflection invariance of the computed quantities.

[68] 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.

[69] 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.

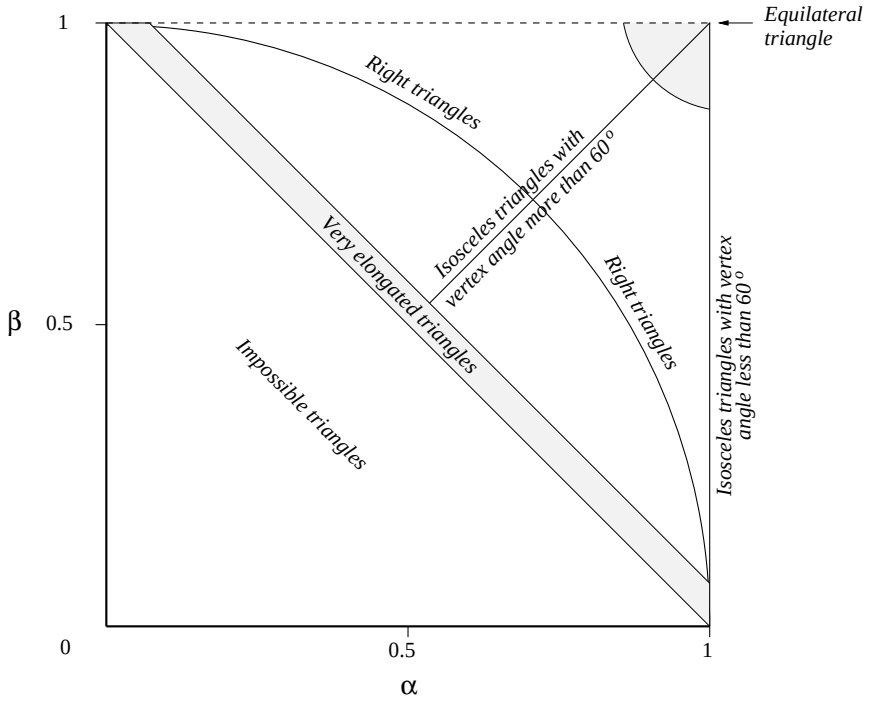

Figure 10.6: *The $\alpha-\beta$ plane. The long shaded region indicates very elongated triangles (including the cases of aligned points), while the region about the equilateral triangle includes ambiguous cases that would not lead to a safe identification of points: the triangles contained in those 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 $\alpha = 1$.*

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.

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 catalog. 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 catalog: 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 catalog 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.

10.6.8 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 7.6.

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.

10.6.9 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;

Tracing spectral edges is not a simple task, because the slit spectra are not always 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).

In principle it may be necessary to trace also the scientific spectra, to compensate for possible instrument instabilities like flexure. This is not a problem for the FORS instruments and therefore not supported by the pipeline.

10.6.10 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

modeling the coefficients of the local models of all spectra. The used polyomial model is described in Section 7.6.

10.6.11 Extraction of slit spectra

The extraction of slit spectra consists in reading the 2D 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 10.6.2 and 10.6.3.

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.

10.6.12 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.

Currently the FORS pipeline just derives a correction for the wavelength calibration using sky lines.

In principle one could use the sky slit spectra for deriving a second extraction mask, following exactly the same procedure described in the previous sections – where a catalog of sky-lines would be used instead of a catalog 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.

10.7 Overview of spectro-polarimetry

It is not the scope of this manual to describe the characteristics of the FORS spectro-polarimeter and the terminology involved when dealing with the reduction of spectro-polarimetric data. Please refer to the FORS Instrument Manual [19], and always keep at hand the "*Stellar spectropolarimetry with retarder waveplate and beam splitter devices*" [38].

The essential point with the FORS pipeline spectro-polarimetric recipes is that they are almost entirely based on the spectroscopic recipes *fors_calib* and *fors_science*, and work therefore in the same way.

The added functionality merely consists on solving the problem of identifying the ordinary from the extraordinary light beams slit spectra, and on performing a safe bookkeeping of all the object-spectra detected and extracted in the different light beams from images obtained at different angles of the retarder plate. The extracted spectra belonging to common object-sources are matched, and their signals appropriately combined to evaluate the Stokes parameters as a function of wavelength, removing the (first order) polarisation internal to the instrument optics. The standard error propagation is also applied.

10.8 Source extraction

The FORS pipeline uses the widely used SExtractor application ([25]) for the detection of sources in an image. SExtractor version 2.5.0 is included in the FORS pipeline.

The default parameters used to run SExtractor are defined in the file *installation_path*/share/esopipes/fors-*version*/config/fors.sex. The most relevant parameters and their default values are:

DETECT_MINAREA: Minimum number of pixels above threshold. *Default*: 5

DETECT_THRESH: Sigmas above the background. *Default*: 1.5

PHOT_APERTURES: MAG_APER aperture diameter in pixels. *Default*: 40. Note that the FORS pipeline does not re-scale the diameter in case of binning. If required, the user should modify this parameter accordingly

PHOT_AUTOPARAMS: MAG_AUTO parameters (Kron factor, minnimum radius). *Default*: 2.5, 3.5

BACK_SIZE: Background mesh size. *Default*: 64

The FORS pipeline rejects any source identified by SExtractor if any of the following conditions are met:

- The center of the star falls outside the image;
- The source has values that do not make any physical sense (e.g. FWHM is negative or zero);
- SExtractor has marked the source with a *flag* (see [25]) value different from 0.

10.9 Flux conservation

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.

10.10 Master flat creation

10.10.1 Spectroscopic flat combination

The flat combination is done on a per-slit basis. The following steps are applied to compute the combined flat, with an extra loop on each slit:

- Get a normal 2D average of all the individual flats.
- Divide each individual by the average flat in order to remove the pixel-to-pixel variation and retain deviations from average.
- Using the spatial distortion and wavelength calibration, the spectral energy distribution of each individual flat is computed by averaging along the spatial axis.
- Each original flat is divided by its spectral energy distribution. This will remove any variations in SED among flats. This happens due to instabilities in the SED of the flats lamps.
- Stack the individual flats obtained in the previous step with the desired method (median, kappa-sigma clipping, sum or mean).
- Multiply the stacked flat obtained in previous step by the average spectral energy distribution obtained from the first average.

10.10.2 Flat normalisation

Flat normalisation is done via two different methods: smoothing and spline/polynomial fitting. Both methods can be combined, but if smoothing is required, it is done first. The algorithms for both normalisation methods work on a slit per slit basis. Spline fitting can be applied to dispersion axis, while polynomial fitting only to the spatial axis.

Smoothing can be applied to both directions. In the dispersion direction both median and average smoothing can be applied. If they are both specified then the median smoothing is done first. In the spatial direction only median smoothing can be applied. The smoothing is applied via a 1-D filter on the collapsed slit profile. In the spatial direction there is a limitation on the size of the median kernel: if it is larger than half the length of the slit, the later will be used as the median kernel size.

Spline fitting normalisation depends on two parameters: the number of knots and a certain flux threshold. The algorithm applied is as follows:

- The master flat is rectified from spatial distortion.
- From the rectified master flat, each slit is collapsed in the opposite direction of the fitting.
- Smoothing is applied, if specified.
- A cubic bspline with the specified number of knots is fitted to the smoothed collapsed flat per slit. Only values higher than the specified threshold are used for the fit (note that the threshold is specified relative to the highest value in the collapsed slit).
- An image is created with the fitted spline covering the full slit. This image is de-rectified to go back to CCD pixels space. This is the fit image.
- The original master flat is divided by the smoothed image.

Polynomial fitting normalisation depends on two parameters: polynomial degree and a certain flux threshold. The algorithm is equivalent to spline fitting but applying a polynomial fit instead.

10.11 Object detection in spectroscopy

In the science frame, the objects to be extracted must be detected prior to any spectra extraction. The object detection is performed on the rectified image for MOS, MXU, and PMOS data. For LSS and LSS-like the rectification (spatial distortion correction) takes place only if GLOBAL_DISTORTION_TABLE is and input.

The algorithm applied is based on the relative peak intensity of each detected object. First of all, a collapsed image is created averaging along the dispersion axis. On this 1-d spatial profile, and on a slit per slit basis, peaks are identified. A peak is identified by a positive value that is preceded and followed by two lower positive values that decrease with distance. For non-rectified LSS data broader peaks might appear due to the remaining curvature. Each peak is compared with all the other peaks to determine if this peak is contaminated by any of the others. Indicating with L_o the peak value of the examined peak and with L the peak value of another peak, the quantity $S = C * (L/L_o)^2$ is computed, where C is the indicated contamination radius that represents the minimum distance at which two point-like object of equal luminosity can stay without contaminating each others. This parameter can be set through recipe parameter *con_radius*. If the distance between the two peaks (as detected in the 1-d spatial profile) is less than S, the examined peak is excluded from the list of detected objects.

With the final list of object positions the extraction intervals are determined in the following way: for each pair of consecutive peaks, the intermediate positions are determined with the inverse barycenter formula (defining the point of minimal reciprocal contamination): $B_i = (P_i * L_j + P_j * L_i)/(L_i + L_j)$ where P_i is the position of the i-th peak, L_i its peak value, and $j = i + 1$. This formula gives intermediate positions which are usually closer to the weaker peak. The extraction interval of an object goes from the lower intermediate position to the upper intermediate position. The position of the upper limit of the first object is set at the top border of the slit, excluding the number of pixels indicated by the *slit_margin* argument. Analogously, the position of the lower limit of the last object is set at the bottom border of the slit, excluding the same number of pixels.

10.12 Sky coordinates of extracted spectra

The sky coordinates of the extracted spectra in LSS, MXU and MOS data can be estimated using the WCS keywords provided by the telescope. The algorithm works as follows:

- The middle point of the extraction window of the object to be extracted is computed as the average of columns start_i and end_i (i being the number of the object) in the OBJECT_TABLE_SCI_<mode> (<mode> = LSS, MOS, or MXU) table. This spatial coordinate is in the coordinate system of the rectified image.
- At the middle point of the object, the reference wavelength position is identified (read from the header keyword HIRARCH ESO INS GRIS1 WLEN).
- Using the spatial distortion correction, the coordinates previously computed are transformed into non-rectified coordinate system.
- For grism which introduce an additional shift along the spatial axis (GRIS_600RI and GRIS_1400V) the nominal shift values are corrected. Also, the coordinates are compensated for the overscan in the original raw images. After this, the middle point of the object has been identified in the raw image.

- Using the raw image coordinates together with the WCS[70] from the telescope the sky coordinates can be computed.

It should be taken into account that this gives an approximation and several factors can introduce discrepancies:

- It is assumed that the reference wavelength is the undeviated wavelength of the grism.
- The telescope WCS should be accurate. This is usually the case, but discrepancies have been observed between the WCS of the raw image and the WCS of the through-slit images of up to 10 pixels. The reason for this is unkown.
- The raw image WCS doesn't take into account distortions introduced by the optics.

10.13 Cosmic Rays correction

The cosmic rays correction is responsible for detecting and correcting cosmic rays. The algorithm is applied to not-rectified spectral exposures from where the sky spectrum is already subtracted. The algorithm for detecting the cosmic rays works as follows:

- The noise due to detector readout and to the background signal level is estimated. This is done averaging the negative values of the input image I:

$$n_0 = -1.25 \cdot \frac{\sum_{I(x,y)<0} I(x,y)}{\sum_{I(x,y)<0} 1} \tag{4}$$

- The input image is smoothed with a 3×3 median filter, obtaining I_m.
- A pixel having coordinates (x, y) is selected as candidate cosmic ray if:

$$I(x,y) - I_m(x,y) \geq T \cdot n(x,y) \tag{5}$$

  where T is a threshold and $n(x, y)$ is the estimation of the noise in position (x, y):

$$n(x,y) = \sqrt{n_0^2 + \frac{I_m(x,y)}{G}} \tag{6}$$

  where G is the gain.

- The candidates are grouped together, iteratively growing a search box until no neighbor candidate can be added.
- For every group C (a group can contain one or more pixels) the pixel with the maximum value $f_{Max}^{(C)}$ is identified.

[70] Currently the keywords that describe the WCS in FORS FITS files are *CRVAL1, CRPIX1, CTYPE1, CRVAL2, CRPIX2, CTYPE2, CD1_1, CD1_2, CD2_1, CD2_2*

- The average $f_{3\times3}^{(C)}$ of the pixels belonging to a 3×3 window centered on $f_{Max}^{(C)}$ is calculated. Note that the average is calculated on the pixels belonging to the window, regardless of whether they belong or not to the group.
- If
$$f_{Max}^{(C)} > R \cdot f_{3\times3}^{(C)} \tag{7}$$
each pixel of the group C is considered a cosmic ray. R is a threshold.

For what concerns the correction step, a linear interpolation along the horizontal axis is applied in order substitute the values of the pixels affected by cosmic rays.

10.14 Spectra extraction

The extraction of spectra is performed on rectified images for the detected objects. For LSS images, unless a global distortion model has rectified the images, the remaining curvature will affect the extraction. However for Horne extraction, the spectra will be traced by the extraction algorithm.

There are two extraction methods: aperture extraction and Horne optimal extraction (see [34] for details).

The aperture extraction works as follows:

- A window is extracted from the rectified image of the size defined during object detection (see 10.11).
- The spectrum is collapsed across the rows of the object window. The collapsing is done with a sum.

The Horne extraction method works as follows:

1. A 3x3 median filtering is applied to the object window only if the number of rows of the object window is larger than 5.
2. A first extracted spectrum is computed as in aperture extraction.
3. A profile is computed as the 2-D spectrum divided by the current estimation of the extracted spectrum.
4. The profile is smoothed in spectral direction with a median kernel of 32 pixels.
5. For each pixel, a weight is computed as the profile value divided by the variance. The variance is derived from the detector noise model, using the photon noise from the source multiplied by the profile and adding the sky noise in quadrature.
6. The spectrum is extracted using a weighted sum. Steps from point 3 are repeated a predefined number of iterations (currrently 2). Due to their high variance, cosmic rays have low weights, therefore they are usually removed by this step.

The calculation and application of the barycentric correction is detailed in Section 7.5 and 7.6 of the ESOTK manual ([9]).

10.15 Overscan correction

The FORS1 and FORS2 detectors use a FIERA controller which can be read using several ports (usually 1 or 4). For each port, the controller can read a few pixels which have not been illuminated before any illuminated pixel (prescan) and/or continue reading the last pixel for a few extra reads (overscan).

These prescan and overscan regions contain the overall bias level for an image. It is usually recommended that for each image from the detector the prescan/overscan values are subtracted.

The FORS pipeline removes the prescan on a port by port basis. The overscan is ignored (according to detector engineers this is the best practice). The algorithm is as follows:

- For each port, the regions which contain the prescan, overscan and valid pixels are identified using the ESO DET keywords present in the header (DET OUTi PRSCX/Y, DET OUTi OVSCX/Y, DET OUTi NX/NY, DET OUTi X/Y, DET CHIP1 NX/NY, DET WIN1 BINX/BINY for i=1...4). Only one prescan region per port is supported (FIERA controller cannot be configured with more than one per port).
- Each pre/overscan region is a few pixels wide. We will call this the rows, which can be along the X or Y direction of the image, depending on the port configuration. The overscan runs along one of the port parallel readout directions (called here columns). The prescan is computed using a median filter along the whole prescan region.

  For some combinations of detectors and binning factors, it has been observed that a significant portion of the prescan rows are corrupted. In this case, only one row from the prescan has been kept. These are the exceptions:

  - Detector CCID20-14-5-3 with binning 1. The row 2068 is the only one kept for the prescan region.
  - Detector CCID20-14-5-3 with binning 2. The row 1034 is the only one kept for the prescan region.
  - Detector CCID20-14-5-6 in all cases. The row 1 is the only one kept for the prescan region.
  - Detector Norma III with binning 1. The row 2068 is the only one kept for the prescan region.
  - Detector Norma III with binning 2. The row 1034 is the only one kept for the prescan region.
  - Detector Marlene in all cases. The row 1 is the only one kept for the prescan region.
- The error in the computed value is determined as the readout noise of the detector divided by the number of pixels used for the mean computation. The readout noise of the detector is estimated as the square root of the robust variance (excluding outliers) of the master bias after the pre/overscan has been trimmed.
- For each row of the science/calibration pixels, the computed pre/overscan is subtracted and the error is propagated adding it in quadrature as usual.

11 Master bias creation

The master bias creation works following this algorithm:

- the prescan is subtracted from the individual bias frames as explained in 10.15.
- the readout noise for each individual frame is computed from the robust standard deviation in each of the detector ports.
- the master bias is combined using any of the available methods (mean, wmean, median, minmax, ksigma) and its error is propagated using the formulas given in 10.1.2
- the readout noise of each of the detector ports is stored in the master bias under keywords ESO QC DET OUTi RON.

12 Master bias subtraction

The bias subtraction for any of the raw files proccessed by the pipeline, the master bias subtraction is performed as follows:

- the readout noise for each of the detector channels is retrieved from keywords ESO QC DET OUTi RON found in the master bias.
- the detector noise model is computed as shown in 10.1.3.
- the pre/overscan is subtracted for each detector port individually. The error in the subtraction is propagated.
- the master bias is subtracted and the variance of the result is the raw variance computed in previous step added in quadrature to the variance of the master bias

A Troubleshooting Guide

In the following sections, a troubleshooting guideline for the FORS pipeline spectroscopic recipes is given. It is assumed here that some familiarity with these recipes was already acquired. These notes have been mostly adapted from the pipeline sections of [17].

See also Sections 9.9.3 and 9.10.3 of this Manual.

A.1 Checking the results of recipes fors_calib and fors_pmos_calib

Things can go wrong. In this Section a number of basic checks are suggested for ensuring that the *fors_calib* recipe worked properly. Troubleshooting is given separately, in the next Section, in order to avoid too many textual repetitions: it often happens, in fact, that different problems have the same solution. Three basic checks are described here: spectra localisation, wavelength calibration, and spectral resolution. It is advisable to perform such checks in the given order, because some results make only sense under the assumption that some previous tasks were performed appropriately. For instance, an apparently good wavelength calibration does not imply that the slit spectra were all properly traced.

A.1.1 Were all spectra detected and properly traced?

Compare (blink) the *master_norm_* and the *master_screen_flat_mxu.fits* images. The normalised flat field image can be used as a map showing where the spectra were found and how they were cut out from the CCD, while the master flat image shows where the spectra actually are. A quick visual inspection will immediately expose any badly traced, or even lost, spectrum. This kind of failure may not be so apparent in the *reduced_lamp_mxu.fits* image, which includes just what has been successfully extracted.

The *curv_traces_mxu.fits* table enables a closer look at the tracing accuracy. The tracings of the top and bottom edges of the spectrum from slit 10, for instance, are given in the table columns labeled "t10" and "b10", for each CCD pixel along the horizontal direction given in column "x". Each tracing may be compared with the fitted model: for instance, the modeling of the tracing "t10" is given in the table column "t10_mod", together with the fit residuals in column "t10_res", enabling the generation of plots like those shown in Figure A.1. In order to reduce the residuals, the degree of the fitting polynomial may be increased (using the configuration parameter *--cdegree*): it is however advisable to never use polynomials above the 2nd order, unless the residuals are really not acceptable. In Figure A.1 the residuals are less than 3 hundreds of a pixel, and this is acceptable even if they display a systematic trend that may be easily eliminated by fitting a 3rd degree polynomial. When systematic trends in the residuals are so small (with respect to the pixel size), they can no longer be considered "physical", but rather an effect of the pixelisation of the edge changing with the position along the CCD. See also Figure 9.7, page 112 of this Manual, and related Section.

A.1.2 Were all spectra properly calibrated in wavelength?

Check the *reduced_lamp_mxu.fits* image first. This image contains the arc lamp spectra from each slit with all the optical and spectral distortions removed. The spectral lines should all appear perfectly aligned and vertical.

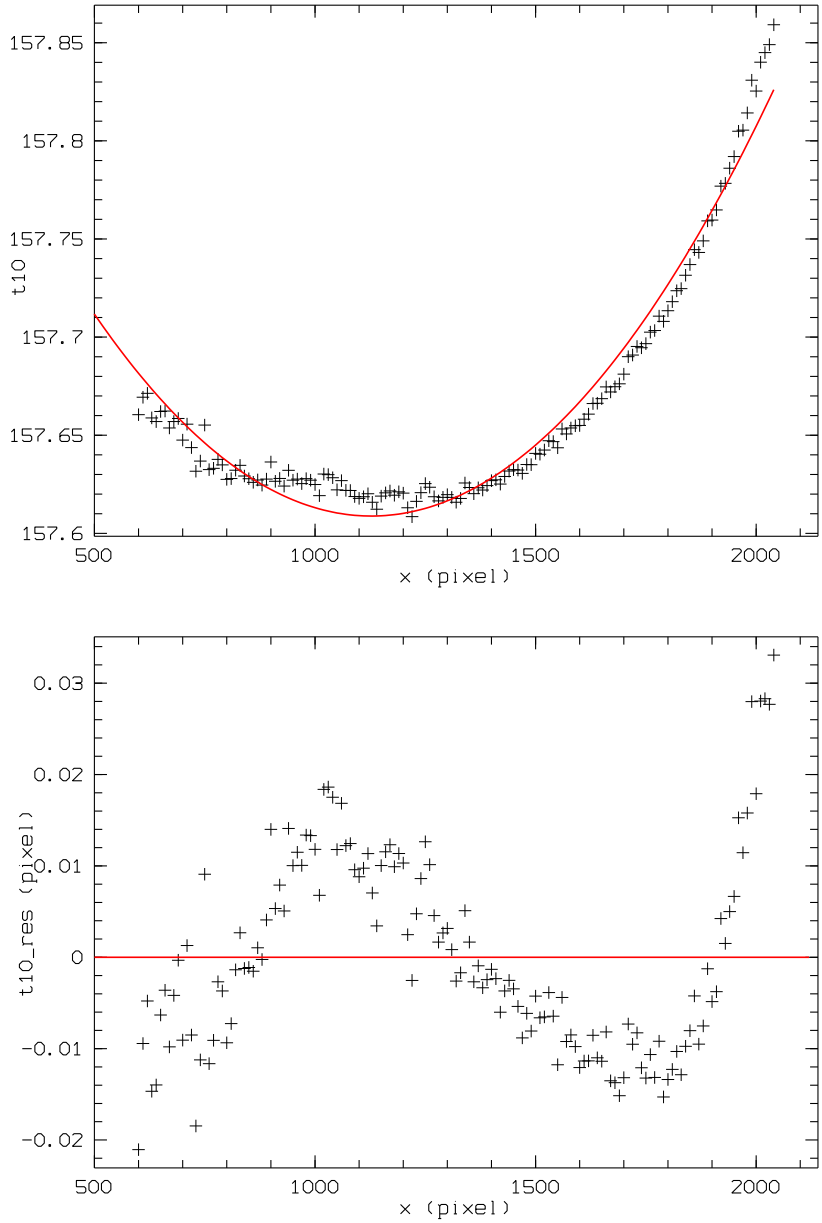

Figure A.1: *Tracing, modeling, and systematic residuals (in pixel) of one spectral edge tracing.*

Particular attention should be given to lines at the blue and red ends of each spectrum, where the polynomial fit is more sensitive to small variations of the signal. The calibrated slit spectra are vertically ordered as in the original CCD frame. The boundaries between individual slit spectra are generally easy to recognise: both because they are often dotted by the emission lines from nearby spectra on the original CCD frame, and because each slit spectrum may cover different wavelength intervals according to its position within the original CCD frame (see Figure 9.4, page 104). The position of each spectrum in the calibrated image is always reported in the table *slit_location_mxu.fits*, at the columns "position" and "length".

More detailed checks on the quality of the solution can be made by examining other pipeline products. The image *disp_residuals_mxu.fits* contains the residuals of the wavelength solution for each row of each slit spectrum. This image is mostly padded with zeroes, with the only exception of the pixels where a reference line was detected and identified: those pixels report the value of the corresponding residual (in pixel). This image will in general be viewed applying small cuts (typically between -0.2 and 0.2 pixels): systematic trends in the residuals, along the dispersion direction, would appear as sequences of all-positive (white) followed by all-negative (black) residuals, in a wavy fashion, that could also be viewed by simply plotting a profile at different image rows (see Figure 9.3, page 102). Systematic residuals in the wavelength calibration are in general not acceptable, and they may be eliminated by increasing the order of the fitting polynomial.

Another product that can be used for evaluating the quality of the fit is the *disp_residuals_table_mxu.fits* table. Here the residuals are reported in a tabulated form for each wavelength in the reference lines catalog, but just for one out of 10 rectified image rows (i.e., one out of 10 solutions). In conjunction with the *delta_image_mxu.fits* image, plots like the ones in Figure 9.1, page 100, can be produced.

Finally, the table *disp_coeff_mxu.fits* might be examined to check how many arc lamp lines were used (column "nlines") and what is the mean uncertainty of the fitted wavelength calibration solution (column "error"), for each row of each slit spectrum. The model mean uncertainty is given at a 1-σ level, and has a statistical meaning only if the fit residuals do not display any systematic trend and have a random (gaussian) distribution around zero. Typically this uncertainty will be of the order of 0.05 pixels, i.e., much smaller than the root-mean-squared residual of the fit, depending on the number of fitted points (a fit based on a large number of points is more accurate than a fit based on few points). It should be anyway kept in mind that the model uncertainty can be much larger than that (up to 1 pixel in the worst cases) at the blue and red ends of the fitted wavelength interval, as shown in Figure 9.2, page 101. This is because in the pipeline the wavelength solution is obtained by fitting a polynomial, rather than a physical model of the instrument behaviour.

In the case of LSS data, if the parameter *--wmode* is set to 2 (the default), the wavelength calibration can be much more accurate than that, even at the extremes of the spectral range. The errors reported in *disp_coeff_lss.fits* always refer to the single calibrations (each CCD row is calibrated independently), but if *--wmode* is set to 2 a global model is fitted to all the reference lines visible on the whole CCD, leading to a calibration accuracy of the order of 0.001 pixels (at least theoretically: systematic errors, e.g., due to physical irregularities of the long slit, are not included in this estimate).

A.1.3 Is the spectral resolution as expected?

The table *spectra_resolution_mxu.fits* reports on the mean spectral resolution, defined as $R = \lambda/\Delta\lambda$ (with $\Delta\lambda$ determined at half-maximum), which was measured for each reference arc lamp line (see Figure 9.6, page 108). The standard deviation from this mean is also given, together with the number of independent determinations of R in column "nlines".

A.2 Fixing pattern-recognition failures in fors_calib and fors_pmos_calib

In this Section and the following a set of possible solutions to almost any problem met with the *fors_calib* recipe is given. It is advisable to try them in the same order as they are listed here. It may be useful to go through this check list even in case the recipe seemed to work well: there might always be room for improvement.

In practice, almost any problem with the pipeline is caused by a failure of the pattern-recognition task. Pattern-recognition is applied to detect the slit spectra on the CCD, assuming that they all will include an illumination pattern similar to the pattern of wavelengths listed in the reference arc lamp line catalog.

In the case of either MOS or MXU data, for an immediate visualisation of how successful was the pattern-recognition check image *spectra_detection_mxu.fits* , a by-product of the pattern-recognition task, displaying a preliminary wavelength calibration of the CCD. This image has as many rows as the CCD: if at any CCD row the line catalog pattern is detected, the spectral signal is wavelength calibrated, resampled at a constant wavelength step, and written to the same row of the *spectra_detection_mxu.fits* image. If a row of this image is empty, it is either because the corresponding CCD row doesn't contain any spectrum, or because the pattern-recognition task failed for that row. The check image may simply be placed side by side with the original CCD exposure, in order to see if and how frequently a spectral signal was not recognised as such. A few failures (i.e., a few empty rows) are generally acceptable, as they are recovered by interpolation during the final wavelength calibration task. However, a high failure rate is probably the reason why a bad spectral localisation, or tracing, or final wavelength calibration, were possibly obtained.

In the case of LSS or LSS-like MOS data, for an immediate visualisation of how successful was the pattern-recognition just rerun the *fors_calib* recipe setting the *--wmode* parameter to 0. This will disable the computation of a global model of the wavelength calibration. The image *reduced_lamp_lss.fits* may look a bit noisier than the same image obtained with *--wmode* set to 2, and some of its rows may contain no signal. The *reduced_lamp_lss.fits* image covers an interval of CCD rows beginning with the first and ending with the last row where a local solutions was found. Within this interval, if at any CCD row the line catalog pattern is detected, the spectral signal is wavelength calibrated, resampled at a constant wavelength step, and written to the corresponding row of the *reduced_lamp_lss.fits* image. If a row of this image is empty it is because the pattern-recognition task failed for that row. A few failures (i.e., a few empty rows) are generally acceptable, as they are easily recovered by the global model interpolation. However, a high failure rate is probably the reason why a bad (global) wavelength calibration was possibly obtained.

What can make the pattern-recognition task fail? One or more of the following causes may be determined:

A.2.1 Some arc lamp reference lines are very faint

It is possible that the exposure time for the arc lamp frame is too short, or one of the lamps got too faint with age. If some of the reference lines listed in the catalog do not peak above a given threshold, they are not used by the pattern-matching task.

Solution: Specify a lower value for the *--peakdetection* parameter. Alternatively, if this gets too close to the noise level, remove the faint lines from the reference line catalog.

A.2.2 The reference lines in the arc lamp exposure are very broad

If very wide slits are used, the reference lines would become accordingly wider (and would display a box-like, flat-top profile). The calibration recipe can handle this in case of well isolated lines, but if nearby lines blend together it is impossible to safely determine their positions.

Solution: None. These spectra cannot be calibrated.

A.2.3 The spectral dispersion is not what expected

The actual mean spectral dispersion is significantly higher (or lower) than expected. The first-guess spectral dispersion is specified via the parameter *--dispersion*, and is tabulated for each grism in the FORS1+2 User Manual [19], or in the grism tables which are included in the distributed FORS pipeline package. In general the pattern-recognition algorithm is quite robust against changes of the spectral dispersion (up to 20% from expectation), but for some grisms (such as the 600B in FORS1 and FORS2) good results can only be obtained within a much narrower window of values of the first-guess. For this reason a small change of the spectral dispersion (perhaps caused by a large temperature variation) may cause the wavelength calibration to fail.

Solution: Try different values of the *--dispersion* parameter around the expected (default) value, and select the one producing the lowest failure rate of the pattern-recognition task.

A.2.4 There are spectra at very large offsets

The CCD may include spectra at such large x offsets that only part (red or blue) of their full wavelength range is really included in the CCD. If the line catalog contains too few reference lines in this region (say, less than 5), they might not be enough to define an unambiguous pattern to detect.

Solution: Add extra reference lines to the line catalog, for a more complete coverage of the bluest/reddest parts of the complete spectral range. If there are no extra lines to be used as a reference, the truncated spectra will then be definitely lost.

A.3 Fixing other possible failures in fors_calib and fors_pmos_calib

If the pattern-recognition seems to have worked properly, the reason of a *fors_calib* recipe failure can be found elsewhere:

A.3.1 The spectra are too tightly packed

If slits are too close to each others, there is a risk that (some of) the spectra would not be properly traced, or not traced at all, on the flat field frames. As a default, the *fors_calib* recipe tries to recover untraceable edges by interpolating a global curvature model based on other traceable edges (if they are available). Using this global description of the spectral curvature helps to extract also those spectra whose edges cannot be traced. In some cases however the recipe may find and accept a bad tracing as if it were good, producing a bad global curvature model, and therefore a bad spectral extraction.

Solution: Setting the parameter *--cmode* to zero will suppress the usage of the global curvature model. In this case the recovery strategy of lost spectral edges will consist in replicating the trace of the other available spectral edge (opportunely shifted) of the same slit spectrum. This may improve the results in some cases: however, if a tracing is missing for both edges of a slit spectrum, the spectrum will not be extracted.

A.3.2 The wavelength calibration residuals display systematic trends

Especially if the extracted spectral range is very large, the fitting polynomial may be incapable to replicate the physical relation between pixel and wavelength. In this case, any estimate of the statistical error (such as the fit uncertainties listed in *disp_coeff_mxu.fits*) will become meaningless.

Solution: Increase the degree of the fitting polynomial, using the parameter *--wdegree*. Beware that this may introduce overfitting, especially at the red and blue ends of the spectra (i.e., the polynomial is so poorly constrained in those regions where few points are available, that it also fits their position uncertainty, incorporating this noise into the solution: the corresponding residuals may therefore look very small, and yet the calibrated spectra will appear to be badly calibrated; an extreme case of overfitting is, for instance, fitting 4 points with a 3rd degree polynomial: the residuals will be exactly zero, and yet the obtained model will be highly inaccurate). For this reason, while applying this solution it may be also appropriate to set the parameter *--wmode* to 2 (for LSS or LSS-like data), or the parameter *--wmosmode* to 1 or 2 (for MOS or MXU data).

A.3.3 The calibrated spectra look "noisy" at their ends

This problem is symmetric to the previous one: the fit residuals may look very small, and yet the calibrated spectra will appear to be badly calibrated at their blue and red ends. This is the effect of model overfitting (see Figure 9.2).

Solution: Decrease the degree of the fitting polynomial, using the parameter *--wdegree*. Beware that this may introduce systematic fit residuals. Other option is to increase parameter *--wradius*

A.3.4 The calibrated LSS spectrum looks distorted

If the global wavelength calibration model is bad, it is either because some of the local solutions are bad, or because there are too few local solutions available for global modeling.

Solution: Rerun the recipe with the parameter *--wmode* set to 0, and change opportunely the recipe configuration (especially the parameter *--dispersion*), trying to maximise the number of obtained local solutions and to get more uniform results in the *reduced_lamp_lss.fits* image. Finally run the recipe with the new found configuration, but setting the parameter *--wmode* back to 2.

A.3.5 The flat field is not properly normalised

The master flat field is normalised by dividing it by a smoothed version of itself. For various reasons the result may be judged unsatisfactory.

Solution: If you are dealing with MOS/MXU data, change the smoothing box sizes using the parameters *--dradius*, *--dradius_aver*, and *--sradius*. Alternatively, instead of the default smoothing, a spline may be used to fit the large scale trend: the number of knots in the fitting spline is specified via the *--d_nknots* parameter. Note that the spline fitting is done after any smoothing.

Solution: If you are dealing with LSS/LSS-like data, change the *--s_degree* parameter, indicating the degree of the polynomial fitting the large scale illumination trend along the spatial direction. Alternatively, instead of the default polynomial fitting a median smoothing may be applied: *--s_degree* should be set to -1, so that the smoothing box sizes can be specified with parameter *--sradius*. Along the dispersion direction, the recommended normalisation is to use a median smoothing of 10 pixels (*--dradius*=10, *--dradius_aver*=-1, and *--d_nknots*=-1).

A.3.6 Valid reference lines are rejected

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

Solution: Increase the value of the *--wreject* parameter. Extreme care should be used here: a tolerant line identification may provide an apparently good fit, but if this is based on misidentified lines the calibration would include unknown systematic errors.

A.4 Checking the results of recipes fors_science and fors_pmos_science

In this Section a number of basic checks are suggested for ensuring that the recipe *fors_science* worked properly. Troubleshooting is given separately, in the next Section, in order to avoid too many textual repetitions: it often happens, in fact, that different problems have the same solution. Four basic checks are described here: wavelength calibration, sky subtraction, object detection, and object extraction. It is advisable to perform such checks in the given order, because some results make only sense under the assumption that some previous tasks were performed appropriately. For instance, an apparently good sky subtraction does not imply that the slit spectra were all properly wavelength calibrated.

A.4.1 Were all spectra properly wavelength calibrated?

The wavelength calibration based on calibration lamps, performed at day-time, may not be appropriate for an accurate calibration of the scientific spectra: systematic differences due to instrumental effects, such as flexures, may intervene in the meantime.

To overcome this, the day calibration may be upgraded by testing it against the observed positions of the sky lines in the scientific slit spectra. The alignment of the input distortion models to the true sky lines positions is controlled by the parameter *--skyalign*, that as a default is set to 0 (i.e., the sky lines correction will be a median offset).

The overall quality of the wavelength calibration (whether a sky line alignment was applied or not) can be examined in the *mapped_all_sci_mxu.fits* image. This image contains the scientific spectrum after resampling

at a constant wavelength step. The visible sky lines should all appear perfectly aligned and vertical.

In the case of MOS/MXU data the quality of the alignment may be quite poor. It is possible, however, that an alignment of the distortion models was unnecessary: if this were the case, it would be better to avoid it entirely (any extra manipulation increases the statistical uncertainties on the final product). In order to decide whether a sky alignment is necessary or not, the *sky_shifts_slit_sci_mxu.fits* table can be examined. This table has a column labeled "wave", listing the wavelengths of all the reference sky lines found within the extracted spectral interval, and a number of columns labeled "offset_*id*", listing the median offset in pixels for each sky line from its expected position, for the slit identified by "id" (see Figure A.2). *Beware*: the listed offsets are not the residuals of the final sky line alignment, but really the comparison of the sky line positions against expectations from the input distortion models. In case the sky line offsets are compatible with zero, the sky line alignment is really unnecessary, and the *fors_science* recipe may be run again setting the *--skyalign* parameter to -1 (i.e., the sky lines correction will be disabled). This is not strictly necessary, but it is often wise to keep data manipulation to a minimum. On the other hand, observing systematic offsets would confirm that an alignment of the distortion model to the true sky lines positions was in order, and there would be no need to reprocess the data. In case the offset appears to depend on the wavelength, it may be appropriate to set the parameter *--skyalign* to 1.

The overall quality of the wavelength calibration (whether a sky line alignment was applied or not) can be examined in the *mapped_all_sci_mxu.fits* image. This image contains the scientific spectra from each slit after removing the optical and spectral distortions. The visible sky lines should all appear perfectly aligned and vertical. The position of each spectrum in the calibrated image is listed in the table *object_table_sci_mxu.fits*, at the columns "position" and "length".

A further check on the quality of the solution can be made by examining the *disp_coeff_sci_mxu.fits* table. This table is only produced in case a sky line alignment was performed. Column "nlines" reports how many sky lines were used for the distortion model correction, while the "error" column reports the mean uncertainty of the new wavelength calibration solution for each slit spectrum row. The model uncertainty is given at a 1-σ level, and is computed as the quadratic mean of the input model accuracy and the sky line correction accuracy. Typically this uncertainty will be of the order of 0.1 pixel, i.e., much smaller than the root-mean-squared residual of the lamp calibration and of the sky line correction, depending on the number of fitted points. It should be anyway kept in mind that the model uncertainty can be much larger than that (up to 1 pixel in the worst cases) at the blue and red ends of the fitted wavelength interval, as shown in Figure 9.2. This is because in the pipeline the wavelength solution is obtained by fitting a polynomial, rather than a physical model of the instrument behaviour.

A.4.2 Is the sky background properly subtracted?

A quick check on sky subtraction can be made by examining the sky subtracted frames, *mapped_sci_mxu.fits* and *unmapped_sci_mxu.fits* (if available, depending on how the recipe was run). The spectra should have a generally smooth look, and will only appear to be noisier in those regions where bright sky lines were subtracted.

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

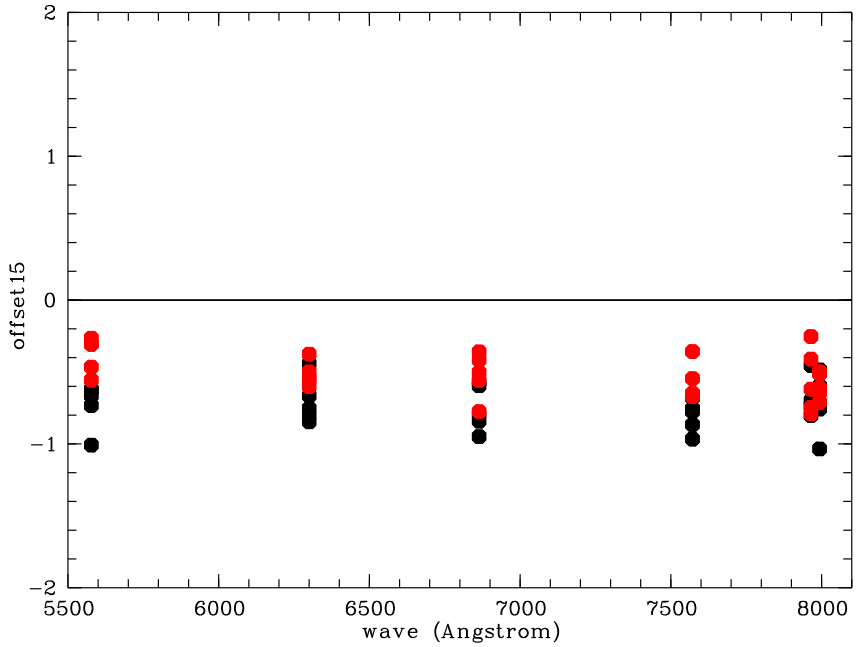

Figure A.2: *Systematic sky line offsets (in pixel) from day-calibration expectation, observed in a FORS2 MXU 600RI scientific exposure. The offsets from all slits are plotted, black and red points belonging to slits from the upper and lower part of the CCD. All offsets are systematically negative, and therefore a sky alignment of the wavelength solution is due. Offsets depending on the slit position on the CCD, as shown by the different distribution of black and red points in figure, indicate that the scientific exposure is not just translated (e.g., by flexures) but also rotated with respect to day calibrations.*

is probably as good as it can get.

Note that the subtracted sky can be viewed in the images *mapped_* and *unmapped_sky_sci_mxu.fits*. More useful is perhaps the image containing the extracted sky spectra, *reduced_sky_sci_mxu.fits*: such spectra are extracted applying to the modeled slit sky spectra exactly the same weights that were used in the object extraction.

A.4.3 Were all objects detected?

The answer to this question is almost always "no". The pipeline, after removing the instrument signature and the sky background from each slit spectrum, will run an object detection algorithm in order to find all the objects which need to be extracted. There will always be a detection threshold beyond which an object will not be significant enough for selection – no matter what detection algorithm is applied. Using more tolerant detection criteria would not eliminate this threshold effect, and may increase the number of false detections to the point of making the object detection task impractical.

The list of detected objects can be found in the *object_table_sci_mxu.fits* table.

A.4.4 Were all the detected objects properly extracted?

As a default the *fors_science* recipe would apply an optimal extraction algorithm to each detected object spectrum. This algorithm is only appropriate for point-like objects emitting over (almost) all the extracted wavelength interval, while it is not appropriate for extended objects, and it is ineffective for objects having a spectrum only consisting of emission lines with no continuum.

The statistical noise on the extracted object spectra should in principle decrease if the spectra are optimally extracted. In order to check the improvement of the noise level, it is easy to compare the *reduced_error_sci_mxu.fits* images obtained by running the recipe with and without optimal extraction. A 30% increase of the signal-to-noise ratio can be obtained for faint-objects (background-noise limited), while there would be little or no improvement for brighter sources. The photometric accuracy of the optimal extraction can be checked by simply computing the ratio (or the difference) of the *reduced_sci_mxu.fits* images obtained once with the standard and once with the optimal extraction: the result should be a flat image, displaying no trends or systematic deviations from 1 (or 0).

A.5 Fixing failures in fors_science and fors_pmos_science

In this Section a set of possible solutions to the most common problems with the *fors_science* recipe is given. It is advisable to try them in the same order as they are listed here. It may be useful to go through this check list even in case the recipe seemed to work well: there might always be room for improvement.

A.5.1 The wavelength calibration is bad

Aligning the wavelength calibration to the position of the observed sky lines may be inaccurate, especially if very few reference lines are used. If a sky line alignment is really required (see previous Section), then action needs to be taken to solve this problem.

Solution: If very few reference sky lines are used, supplying a sky line catalog including more lines (even if weak and/or blended) may help a lot.

Solution: If the wavelength calibration appears to be bad only at the blue and/or red ends of the spectra, go back to the *fors_calib* recipe to obtain a more stable wavelength calibration in those regions (e.g., either by adding new reference arc lamp lines, or by decreasing the fitting polynomial degree).

A.5.2 The sky alignment of the wavelength solution failed

In case a blue grism is used, or if a spectrum has a large offset toward the red, no sky lines may be visible within the observed spectral range.

Solution: None. It is however possible to modify the columns of coefficients in the input *disp_coeff_mxu.fits* table, if the correction can be evaluated in some other way. For instance, the solution can be shifted by adding a constant value (in pixel) to column "c0".

A.5.3 The sky subtraction failed for resolved sources

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

Solution: The default sky subtraction method (parameter *--skylocal*) performs very well for point-like sources where plenty of sky is directly observable within the slit. An alternative method is made available for extended objects (parameter *--skyglobal*). Setting *--skylocal* to *false* and *--skyglobal* to *true* would subtract from all spectra a supersampled model of the median sky spectrum observed in all slits. This method would perform optimally only in case the spectral resolution were the same all over the detector: in practice, this method is always less accurate than the "skylocal" method. But even if it is less accurate, this method remains the extended sources best friend. It is always possible to process the scientific exposures in both ways, one for processing point-like sources and the other for processing spatially resolved sources.

A.5.4 The sky subtraction failed for curved or tilted slits

Obvious residuals related to the sky subtraction are visible on the extracted slit spectra.

Solution: Change sky subtraction method: set *--skylocal* to *false* and *--skymedian to true. The difference between the two methods is that "skylocal" would subtract the sky before, and "skymedian" after the rectification of the spectral data. The second method performs very poorly in comparison to the first, but in the case of curved or slanted slits there is at the moment no other choice than using it.*

A.5.5 Cosmic rays are not removed

As a default the fors_science recipe does not remove cosmic rays hits, leaving them on the sky-subtracted slit spectra: if the optimal spectral extraction is applied, most of the cosmics are removed anyway from the extracted spectra. Optimal extraction is however not always applicable, especially in the case of resolved sources.

Solution: Set the --cosmics parameter to true. This will apply a cosmics removal algorithm to the sky subtracted spectra. The removed cosmic rays hits will be included in the (modeled) sky images, mapped_sky_sci_mxu.fits and unmapped_sky_sci_mxu.fits. Check results carefully for possible distortions.

A.5.6 The sampling of the remapped scientific spectra is poor

When the slit spectra are rectified and wavelength calibrated, they are remapped undistorted to images such as mapped_sky_sci_mxu.fits or mapped_sci_mxu.fits. This remapping may be judged to undersample the signal along the dispersion direction.

Solution: Change the value of the --dispersion parameter. This parameter doesn't need to be identical to the one used in the fors_calib recipe.

It should be noted, however, that making the sampling step smaller will not really increase the information contained in the remapped spectra. As a matter of fact, even maintaining a resampling step close to the original CCD pixel size, the remapped pixel values would still be obtained by interpolating the values from a number of

original pixels that are close to the interpolation point: nearby interpolation points would surely share common information, and this is what introduces correlated noise in the result. Decreasing the resampling step would just increase this effect. In general, working on remapped spectra means to accept that the spectral signal was heavily manipulated, and it is for this very reason that the fors_science recipe also produces reduced – but still unmapped – spectra, as in the unmapped_sci_mxu.fits image.

A.5.7 There are often spurious objects detected at the slit edges

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

Solution: Increase the value of the --slit_margin parameter.

A.5.8 Some "obvious" objects are not detected

Examining the mapped_sci_mxu.fits and unmapped_sci_mxu.fits images it may appear that some clearly visible object spectra are not detected (let alone extracted) by the recipe.

Solution: Setting --cosmics to true (cleaning cosmic rays hits) may help.

Solution: Try different set of values for the parameters --ext_radius and --cont_radius.

A.6 Reducing FORS2 Landolt standard stars taken before 2008

From 2008 on the Stetson catalogue has been used to get standard star fields. The fors_zeropoint recipe is therefore optimized for the Stetson catalogue. Before 2008, however, the Landolt catalogue has been used. It is possible to reduce such data by doing the following:

- *In the SOF substitute file fors2_stetson_2010Dec09.fits by fors2_landolt_std_UBVRI.fits*
- *Run the fors_zeropoint recipe with option --sex_config=/path/to/fors_landlot.sex. The file fors_landolt.sex is delivered with the pipeline and it is in the same location as the default file fors.sex used for the Stetson catalogue. A hint to get the path is to run esorex –help fors_zeropoint and look for the default value of parameter --sex_config*

B Software Installation

The FORS pipeline can be installed in different ways: via package repositories, via the `install_esoreflex` *script or manually installing the software tar files.*

The recommended way is to use the package repositories if your operating system is supported. The `macports` *repositories support OS X 10.11 to 10.14, while the* `rpm/yum` *repositories support Fedora 26 to 29, CentOS 7, Scientific Linux 7. For any other operating system it is recommended to use the* `install_esoreflex` *script. Please note that in case of major or minor (affecting the first two digit numbers) Reflex upgrades, the user should erase the* `$HOME/KeplerData`, `$HOME/.kepler` *directories if present, to prevent possible aborts (i.e. a hard crash) of the esoreflex process.*

B.1 Installing FORS pipeline via `macports`

This method is supported for the macOS operating system. It is assumed that macports (http://www.macports.org) and java JDK 1.8 are installed. The rest of the dependencies are automatically handled by macports itself. Please read the full documentation at http://www.eso.org/sci/software/pipelines/installation/macports.html.

B.2 Installing FORS pipeline via `rpm/yum/dnf`

This method is supported for Fedora 26 to 29, CentOS 7, Scientific Linux 7 operating systems. All software dependencies are automatically handled by `yum/dnf`*. Please read the full documentation at http://www.eso.org/sci/software/pipelines/installation/rpm.html.*

B.3 Installing FORS pipeline via `install_esoreflex`

This method allows to install the FORS pipeline, including the Reflex workflow and Reflex itself. It can also be used to install at the same time any other pipeline distributed by ESO. The software pre-requisites for `Reflex 2.11` *may be found at: http://www.eso.org/sci/software/pipelines/reflex_workflows. The user has to ensure that those pre-requisites are installed beforehand.*

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

1. *From any directory, download the installation script:*

```
wget https://ftp.eso.org/pub/dfs/reflex/install_esoreflex
```

2. *Make the installation script executable:*

```
chmod u+x install_esoreflex
```

3. *Execute the installation script:*

```
./install_esoreflex
```

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

4. *You will be asked whether you want to use your Internet connection. Unless you want to reuse already downloaded packages (only advanced users), use the default Yes.*

5. *You will be given a choice of pipelines (with the corresponding workflows) to install. Please specify the numbers for the pipelines you require (if you want to support others together with FORS), separated by a space, or type "A" for all pipelines.*

6. *For the pipelines to be installed you will be prompted for the demo data sets to be installed. Type "A" for all demo datasets. Take into account that if you are installing in a directory that already contains data, it won't be removed.*

7. *The script will also detect whether previous versions of the workflows or Reflex were installed and in this case you have the option to update links or remove obsolete cache directories. It is advised to use the defaults.*

8. *If some of the prerequisite binaries for* `Reflex` *are not under one of the paths indicated by the command,*

```
getconf PATH
```

then you will need to add the appropriate paths as a colon separated list to the `esoreflex.path` *parameter in the configuration file* `<install_dir>/etc/esoreflex.rc`*. This will usually be necessary when the FITS viewer (*`fv`*) is installed outside of* `/usr/bin`*. As an example, assume* `fv` *is installed into the directory* `/usr/local/fv5.4`*, the file* `esoreflex.rc` *should then have the line setting* `esoreflex.path` *look similar to the following:*

```
esoreflex.path=/usr/local/fv5.4
```

In the case of OS X `/Applications/fv.app/Contents/MacOS/` *is the typically installation directory. Thus, this should be similar to the following line instead:*

```
esoreflex.path=/opt/local/bin:/Applications/fv.app/Contents/MacOS
```

9. *To start* `Reflex`*, issue the command:*

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

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

C Abbreviations and acronyms

CPL	*Common Pipeline Library*
DFS	*Data Flow System*
DMO	*Data Management and Operation Division*
DO	*Data Organiser*
DPD	*Data Products Department*
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*
IPOL	*Imaging polarimetry*
IWS	*Instrument WorkStation*
MOS	*Multi Object Spectroscopy*
PAF	*VLT PArameter File*
PMOS	*Multi Object Spectropolarimetry*
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*