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 1.0

Date 2006-10-10

Prepared:	C. Izzo	2006-10-10	
	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

This page was intentionally left blank

Contents

1 Introduction

1.1 Purpose

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

This manual is a complete description of the spectroscopic data reduction recipes reflecting the status of the FORS pipeline as of 09.09.2006 (version 3.0.5).

Currently, only the spectroscopic data reduction recipes are described in this manual. FORS1 polarimetric and FORS2 echelle data reduction are not yet supported.

1.2 Acknowledgements

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

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

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

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

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

1.3 Scope

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

Updated versions of the present document may be found on [1]. For general information about the current instrument pipelines status we remind the user of [2]. Quality control information are at [3].

Additional information on the Common Pipeline Library (CPL), *Esorex* and *Gasgano* can be found at [4], [5], and [13]. A description of the instrument is in [6]. The FORS instrument user manuals can be found in [7], while the calibration plan is described in [14].

1.4 Reference documents

[1]	Pipelines Web Page	http://www.eso.org/pipelines
[2]	Current pipeline status	http://www.eso.org/observing/dfo/quality/pipeline-status.html
[3]	ESO-Data Flow Operation home page	http://www.eso.org/observing/dfo/quality/
[4]	CPL home page	http://www.eso.org/cpl
[5]	ESOREX home page	http://www.eso.org/cpl/esorex.html
[6]	FORS home page	http://www.eso.org/instruments/fors/
[7]	FORS1+2 User Manual	VLT-MAN-ESO-13100-1543 http://www.eso.org/instruments/fors/doc/

1.5 Applicable documents

[9]	VLT Data Flow System Specifications for Pipeline and Quality Control	VLT-SPE-ESO-19600-1233
[10]	DFS Pipeline & Quality Control – User Manual	VLT-MAN-ESO-19500-1619
[11]	ESO DICB – Data Interface Control Document	GEN-SPE-ESO-19400-0794
[12]	Common Pipeline Library User Manual	VLT-MAN-ESO-19500-2720
[13]	Gasgano User's Manual	VLT-PRO-ESO-19000-1932
[14]	FORS Calibration Plan	VLT-PLA-ESO-14700-2617
[15]	FORS Pipeline and Quality Control User's Manual	VLT-MAN-ESO-19500-1771

2 Overview

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

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

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

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

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

ESO offers two front-end applications for launching pipeline recipes, *Gasgano* [13] and *Esorex* [5], both included in the pipeline distribution (see Appendix A, page 79). These applications can also be downloaded separately from http://www.eso.org/gasgano and http://www.eso.org/cpl/esorex.html. An illustrated introduction to Gasgano is provided in the "Quick Start" Section of this manual (see page 13).

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. In Section 5 we advice the user about known data reduction problems providing also possible solutions.

An overview of the data reduction, what are the input data, and the recipes involved in the calibration cascade is provided in Section 8.

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

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

In Appendix A the installation of the FORS pipeline recipes is described and in Appendix B 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.

Only a brief description of the instruments is given here. For more details please refer to:

http://www.eso.org/instruments/fors/overview.html
http://www.eso.org/instruments/fors/inst/

3.1 Instruments overview

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

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

FORS2 is offered at UT1 with a detector consisting of a mosaic of two 2k x 4k MIT CCD (15 μm pixels). Compared to the FORS1 2k x 2k Site detector, the FORS2 mosaic provides greatly improved red sensitivity (> 7500 Å), better cosmetic quality, and faster readout. However, the blue violet response (< 4000 Å) is slightly reduced. 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.

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

- FORS2 has no polarimetric capability.
- FORS2 offers medium resolution echelle spectroscopy with 44" slit length.
- FORS2 offers an extended set of high-throughput volume phased holographic grisms.
- 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.

4.1 FORS pipeline recipes

The current FORS spectroscopic pipeline is based on a set of 4 stand-alone recipes. Only two 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_calib, to evaluate the spectral extraction mask on the basis of flat and arc calibration lamp exposures, and to create a normalised flat field frame.

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

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

fors_sumflux, to monitor the flux of the calibration lamps.

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

4.2 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*, from the automatic data management tools available at Paranal, or from the graphical *Gasgano* tool.

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.

4.2.1 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 4.2.1 (next page), a view on a set of FORS1 MOS 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 classification, the instrument setup id (which indicates the band), the instrument pre-optic (which indicates the camera setting), the template exposure number and the number of exposures in the template, and the value of the DPR.TYPE.

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 4.2.2, a MOS arc lamp exposure with a sequence of flat field exposures, a master bias frame, and a catalog of reference lines, are all selected and sent to the *fors_calib* recipe. This will open a *Gasgano* recipe execution window (see Figure 4.2.3), 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* [13] for a more complete description of the *Gasgano* interface.

Figure 4.2.1: *The Gasgano main window.*

Figure 4.2.2: *Selecting files to be processed by a FORS pipeline recipe.*

Figure 4.2.3: *The Gasgano recipe execution window.*

4.2.2 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* doesn't 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 27). The user should also take care of defining the input set-of-frames and the appropriate configuration parameters for each recipe run:

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

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

```
FORS1.2006-05-10    T12:58:27.122.fits SCREEN_FLAT_MOS
FORS1.2006-05-10    T12:59:45.326.fits SCREEN_FLAT_MOS
FORS1.2006-05-10    T13:00:20.930.fits SCREEN_FLAT_MOS
FORS1.2006-05-10    T13:01:17.711.fits SCREEN_FLAT_MOS
FORS1.2006-05-10    T13:02:14.559.fits SCREEN_FLAT_MOS
FORS1.2006-05-10    T13:03:37.926.fits LAMP_MOS
../../cal/FORS1_    MBIAS.fits MASTER_BIAS
../../cal/FORS1_    ACAT_300I_11_OG590_72.fits MASTER_LINECAT
```

This file contains for each input frame its full path 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 4.2.1, page 14).

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

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

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

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

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

[1] The set-of-frames corresponds to the *Input Frames* panel of the *Gasgano* recipe execution window (see Figure 4.2.3, page 15).

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

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

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

```
esorex --recipes
```

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

```
esorex --params recipe_name
```

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

```
esorex --help recipe_name
```

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

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

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

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

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

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

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

[2]The *Esorex* recipe configuration file corresponds to the *Parameters* panel of the *Gasgano* recipe execution window (see Figure 4.2.3, page 15).

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 we specify 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 recipe configuration file different from the default one can be specified on the command line:

```
esorex --recipe-config=my_alternative_recipe_config
```

More than one configuration file may be maintained for the same recipe but, in order to be used, a configuration file not located under `$HOME/.esorex`, or having a name different from the recipe name, should be explicitly specified when launching a recipe.

A description of the recipe parameters are provided in Sections 9 and 10.

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 `http://www.eso.org/cpl/esorex.html`.

4.3 Example of MOS/MXU data reduction using Esorex

The processing of both MXU and MOS FORS1/2 data is identical: the only difference lays in the suffix, either _MXU or _MOS, assigned to the DO categories of input and output files. It should also be noted that a MOS observation might be performed setting the same offet to all the slitlets: in this specific case, the aligned slitlets are perfectly equivalent to a single long-slit, that 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.[3]

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

It is assumed that the following data are available:

One scientific exposure:

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

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

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

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

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

Five bias exposures:

```
FORS2.2004-09-27    T08:00:27.821.fits          BIAS
FORS2.2004-09-27    T08:01:05.604.fits          BIAS
FORS2.2004-09-27    T08:01:44.091.fits          BIAS
FORS2.2004-09-27    T08:02:22.070.fits          BIAS
FORS2.2004-09-27    T08:03:01.042.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` for images, and `.tfits` for tables. Moreover, it is assumed that all the handled files (inputs and products) are located in the current directory. The only exception is represented by the standard calibration tables (*e.g.*, line catalogues), that here are assumed to be located under `/cal/fors2/fmos`.

In order to process the calibration exposures available for the scientific observation, the recipe *fors_calib* is used (see Section 9.1, page 41). The input SOF may be defined as follows:

File: *cal.sof*

```
FORS2.2004-09-27    T08:00:27.821.fits          BIAS
FORS2.2004-09-27    T08:01:05.604.fits          BIAS
FORS2.2004-09-27    T08:01:44.091.fits          BIAS
FORS2.2004-09-27    T08:02:22.070.fits          BIAS
FORS2.2004-09-27    T08:03:01.042.fits          BIAS
FORS2.2004-09-27    T18:59:03.641.fits          SCREEN_FLAT_MXU    U
FORS2.2004-09-27    T19:00:07.828.fits          SCREEN_FLAT_MXU    U
```

```
FORS2.2004-09-27    T19:01:14.252.fits                SCREEN_FLAT_MX   U
FORS2.2004-09-27    T19:13:03.631.fits                LAMP_MXU
/cal/fors2/mos/F    ORS2_ACAT_300I_21_OG590_32.fits MASTER_LINECAT
/cal/fors2/mos/F    ORS2_GRS_300I_21_OG590_32.fits  GRISM_TABLE
```

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

File: *cal.sof*

```
master_bias.fits                                      MASTER_BIAS
FORS2.2004-09-27    T18:59:03.641.fits                SCREEN_FLAT_MX   U
FORS2.2004-09-27    T19:00:07.828.fits                SCREEN_FLAT_MX   U
FORS2.2004-09-27    T19:01:14.252.fits                SCREEN_FLAT_MX   U
FORS2.2004-09-27    T19:13:03.631.fits                LAMP_MXU
/cal/fors2/mos/F    ORS2_ACAT_300I_21_OG590_32.fits MASTER_LINECAT
/cal/fors2/mos/F    ORS2_GRS_300I_21_OG590_32.fits  GRISM_TABLE
```

The MASTER_LINECAT and the GRISM_TABLE are static calibration tables that are available in the calibration directories delivered with the pipeline recipes. The file FORS2_ACAT_300I_21_OG590_32.fits is the default 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 sent to the shell prompt:

```
esorex  fors_calib  cal.sof
```

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

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

master_norm_flat_mxu.fits: normalised flat field image.

slit_location_mxu.tfits: slit positions on the CCD.

curv_coeff_mxu.tfits: coefficients of the spectral curvature fitting polynomials.

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

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

disp_residuals_mxu.fits: residuals for each wavelength calibration fit.

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

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

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

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

wavelength_map_mxu.fits: map of wavelengths on the CCD.

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

slit_map_mxu.fits: map of the grism central wavelength.

spectra_resolution_mxu.tfits: 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 9.2, page 56). The following set-of-frames file may be created:

File: *sci.sof*

```
FORS2.2004-09-27    T02:39:11.479.fits                SCIENCE_MXU
master_bias.fits                                      MASTER_BIAS
master_norm_flat    _mxu.fits                         MASTER_NORM_FL    AT_MXU
disp_coeff_mxu.t    fits                              DISP_COEFF_MXU
curv_coeff_mxu.t    fits                              CURV_COEFF_MXU
slit_location_mx    u.tfits                           SLIT_LOCATION_    MXU
/cal/fors2/mos/F    ORS2_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.tfits: 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.tfits: 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.tfits: wavelength calibration polynomials coefficients after alignment of the solutions to the position of the sky lines, only created if the sky line alignment was requested.

Currently there is no support for a spectro-photometric calibration. Standard star spectra are reduced in a similar fashion, applying the long-slit data reduction strategy[4] that is described in the next Section.

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 spectral curvature solution: for this reason the spectral curvature related products are not created.[5].

In the following example a FORS2 LSS observation of a standard star is processed.[6]

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

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

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

It is assumed that the following data are available:

One standard star exposure:

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

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

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

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

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

Five bias exposures:

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

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

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

In order to process the calibration exposures available for the standard star observation, the recipe *fors_calib* is used (see Section 9.1, page 41). The input SOF may be defined as follows:

File: *cal.sof*

```
FORS2.2004-09-27    T08:00:27.821.fits              BIAS
FORS2.2004-09-27    T08:01:05.604.fits              BIAS
FORS2.2004-09-27    T08:01:44.091.fits              BIAS
FORS2.2004-09-27    T08:02:22.070.fits              BIAS
FORS2.2004-09-27    T08:03:01.042.fits              BIAS
FORS2.2004-09-27    T19:22:22.308.fits              SCREEN_FLAT_LS    S
FORS2.2004-09-27    T19:23:14.722.fits              SCREEN_FLAT_LS    S
FORS2.2004-09-27    T19:24:52.651.fits              SCREEN_FLAT_LS    S
FORS2.2004-09-27    T19:33:44.097.fits              LAMP_LSS
/cal/fors2/mos/F    ORS2_ACAT_300I_21_OG590_32.fits MASTER_LINECAT
/cal/fors2/mos/F    ORS2_GRS_300I_21_OG590_32.fits  GRISM_TABLE
```

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

File: *cal.sof*

```
master_bias.fits                                    MASTER_BIAS
FORS2.2004-09-27   T19:22:22.308.fits               SCREEN_FLAT_LS   S
FORS2.2004-09-27   T19:23:14.722.fits               SCREEN_FLAT_LS   S
FORS2.2004-09-27   T19:24:52.651.fits               SCREEN_FLAT_LS   S
FORS2.2004-09-27   T19:33:44.097.fits               LAMP_LSS
/cal/fors2/mos/F   ORS2_ACAT_300I_21_OG590_32.fits   MASTER_LINECAT
/cal/fors2/mos/F   ORS2_GRS_300I_21_OG590_32.fits    GRISM_TABLE
```

The MASTER_LINECAT and the GRISM_TABLE are static calibration tables that are available in the calibration directories delivered with the pipeline recipes. The file FORS2_ACAT_300I_21_OG590_32.fits is the default 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 sent to the shell prompt:

```
esorex  fors_calib  cal.sof
```

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

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

master_norm_flat_lss.fits: normalised flat field image.

slit_location_lss.tfits: slit positions on the CCD.

disp_coeff_lss.tfits: 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.tfits: 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.tfits: 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 9.2, page 56). The following set-of-frames file may be created:

File: *sci.sof*

```
FORS2.2004-09-27    T03:12:12.006.fits                  STANDARD_LSS
master_bias.fits                                        MASTER_BIAS
master_norm_flat    _lss.fits                           MASTER_NORM_FL    AT_LSS
disp_coeff_lss.t    fits                                DISP_COEFF_LSS
slit_location_ls    s.tfits                             SLIT_LOCATION_    LSS
/cal/fors2/mos/F    ORS2_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_lss.fits: image with wavelength calibrated slit spectra.

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

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

reduced_std_lss.fits: image with extracted objects spectra.

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

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

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

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

5 Known Problems

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

The automatic sky subtraction is not guaranteed to always work well, especially in the case of LSS or LSS-like data: the presence of extended objects on slit would make 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).

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

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

6 Instrument Data Description

FORS data can be separated into *raw* frames and *product* frames. Raw frames are the unprocessed output of the FORS instrument observations, while product frames are either the result of the FORS pipeline processing (as reduced frames, master calibration frames, etc.), or are outsourced (as standard stars catalogs, lists of grism characteristics, etc.).

Any raw or product 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* [13], that 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 a different set of header keywords.

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

A product 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 product data frame alone wouldn't trigger the launch of any recipe.

In the following all raw and product FORS *spectroscopic* data frames are listed, together with the keywords used for their classification and correct association. The indicated *DO category* is a label assigned 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 4.2.2, page 16).

Raw frames can be distinguished in *general* frames, *MXU* frames, *MOS* frames, and *LSS* frames. Their intended use is implicitly defined by the assigned recipe. The *direct imaging* frames are here ignored, as they are part of the old (MIDAS-based) pipeline. They are described in the previous FORS Pipeline and Quality Control User's Manual [15]. In future releases of the new (CPL-based) pipeline the direct imaging reduction recipes will also be implemented.

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_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: *none*

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

- **Arc lamp spectrum:**

  DO category: LAMP_MXU
  Processed by: fors_calib

Classification keywords:	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

6.3 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

6.4 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 = IMS-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 = IMS-THROUGH	INS SLIT NAME	Slit used
	INS FILT1 NAME	Filter used
	DET READ CLOCK	Readout mode
	DET WIN1 BINX	x-binning
	DET WIN1 BINY	y-binning
	DET OUTPUTS	No of outputs
	DET WIN1 STRX	Window start in x
	DET WIN1 STRY	Window start in y
	DET WIN1 NX	No of pixels in x

DET WIN1 NY		No of pixels in y
DET CHIP1 ID		Chip identifier

- **Standard star spectrum:**

  DO category: STANDARD_LSS
  Processed by: fors_science, fors_extract

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

- **Scientific observation:**

  DO category: SCIENCE_LSS
  Processed by: fors_science, fors_extract

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

7 Static Calibration Data

In the following all the FORS static calibration tables related to spectroscopy 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, which is then used to identify the frames listed in the *set-of-frames* (see Section 4.2.2, page 16).

7.1 Grism table

DO category: GRISM_TABLE

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

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

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

FORS2_GRS_300V_20_GG435_81.fits

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

FORS1_GRS_600B_12_free_00.fits

The table

FORS2_GRS_300V_20_all.fits

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

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 spectral curvature
--startwavelength	start wavelength for spectral extraction
--endwavelength	end wavelength for spectral extraction

A complete description of these parameters is given in Section 9.1.3, page 51.

7.2 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 listing such wavelengths, whose name may be specified using the *fors_recipe* configuration parameter *--wcolumn* (see Section 9.1.3, page 51). 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

The table

FORS2_ACAT_300V_20_all.fits

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

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

7.3 Global distortion table

DO category: GLOBAL_DISTORTION_TABLE

Table containing the modeling of the coefficients of the local distortion models listed in the DISP_COEFF_MXU and the CURV_COEFF_MXU tables (see page 42. This table is just used for allowing the on-line processing of scientific data with recipe *fors_extract*, when appropriate (day) calibrations are not yet available.

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

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

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 spectral 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 spectral curvature solutions could be obtained with:

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

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

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

8 Data Reduction

The FORS spectroscopic pipeline recipes are based on the self-calibrating procedures defined and implemented in the MOSES (Multi-Object Spectroscopy Empirical Self-calibration) library.

A more detailed description of the instrument-independent tasks involved in the complete self-calibration procedure is given in Section 10, page 68. Here we just provide an overview of the data reduction cascade.

8.1 Data reduction overview

Two do-it-all recipes are made available for supporting the reduction of FORS spectroscopic data: the first, *fors_calib*, is used for the definition of the extraction mask based on flat field and arc lamp exposures, and is used also for the creation of the normalised flat field; the second, *fors_science*, is used for applying the extraction mask and the normalised flat field to the scientific and the standard star exposures.

8.2 Required input data

The input data to the recipe *fors_calib* are:

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

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

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

- wavelength calibration local models,
- wavelength calibration residuals,
- spectral curvature local models,
- residuals of flat field spectra tracing,
- optical and spectral global distortion model,
- map of wavelengths for each CCD pixel,
- map of spatial coordinate along a slit for each CCD pixel,

- extracted arc lamp spectra,
- location of slits on CCD, and on extracted arc lamp spectra image,
- master flat field,
- normalised master flat field, and,
- spectral resolution table.

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

- one scientific exposure, or alternatively one spectrophotometric standard star exposure,
- one master bias frame,
- location of slits on CCD,
- spectral curvature local models, and,
- wavelength calibration local models.

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

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

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

8.3 Reduction cascade

The monolithic recipes mentioned here may be viewed as composed by many smaller sub-recipes, each carrying out a specific task: an example of tasks and data flow is given in figures 8.3.1 and 8.3.2.

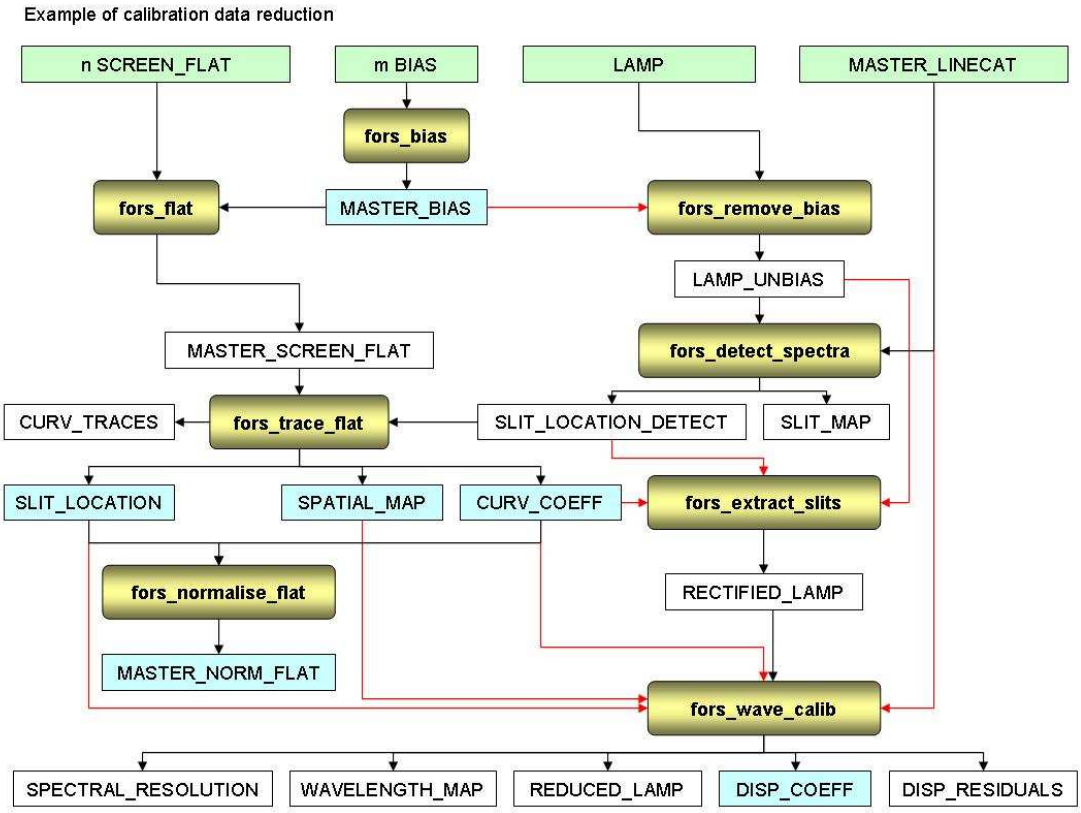

Figure 8.3.1: *This is a possible data reduction flow applied by recipe fors_calib on MOS/MXU data. The rounded yellow boxes represent sub-tasks handling input and intermediate products. The boxes indicating input and product data are labeled with their DO categories, stripped of their variable suffixes (_MOS, _MXU). Note that not all the categories are written to disk. The data corresponding to white boxes are not required in further processing of the scientific exposures by recipe fors_science.*

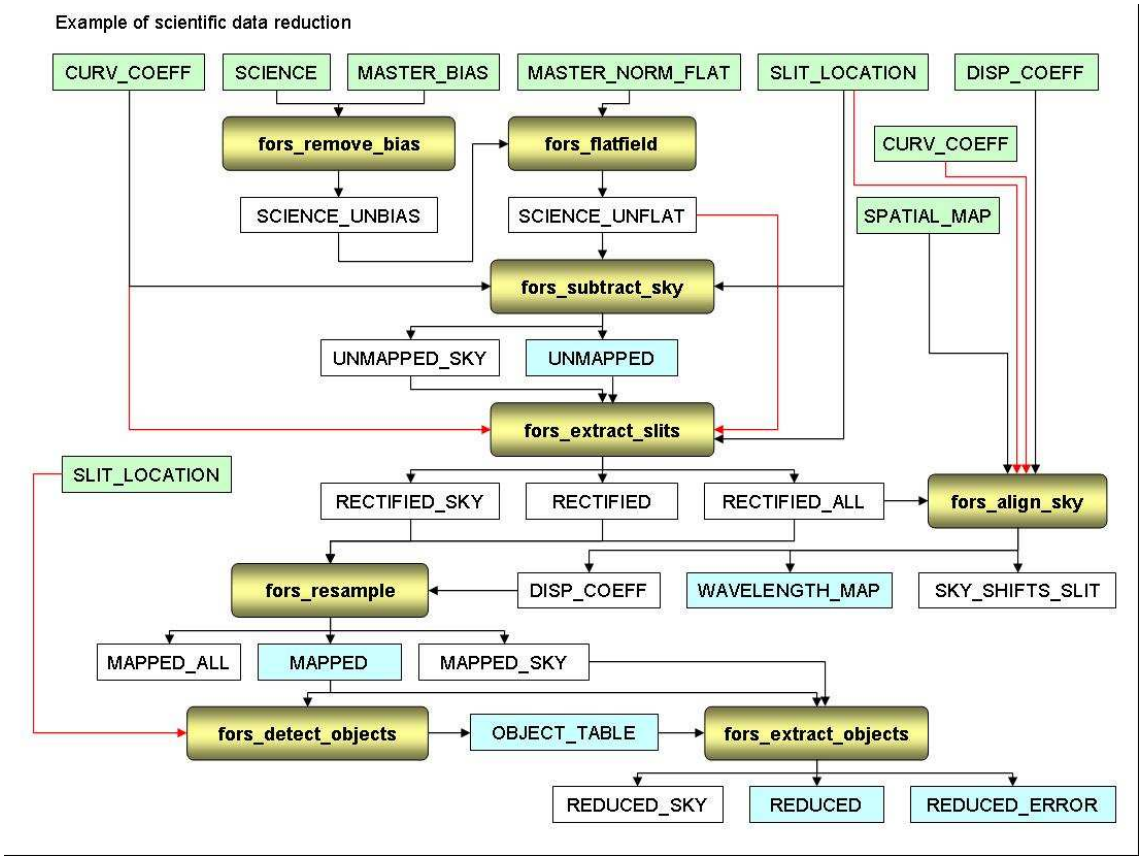

Figure 8.3.2: *This is a possible data reduction flow applied by recipe fors_science on MOS/MXU data. The rounded yellow boxes represent sub-tasks handling input and intermediate products. The boxes indicating input and product data are labeled with their DO categories, stripped of their variable suffixes (_MOS, _MXU, _STD, _SCI, etc.). Note that not all the categories are written to disk. The data corresponding to white boxes are intermediate products, useful for checking the quality of the results.*

9 Pipeline Recipes Interfaces

In this Section a detailed description of the FORS pipeline recipes interfaces is given, with a complete specification of the recipes usage, their input, output, and configuration parameters.

9.1 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.

9.1.1 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 for details).

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

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

MASTER_LINECAT: *required* line 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 listing such wavelengths, whose name may be specified using the configuration parameter *--wcolumn* (see Section 9.1.3, page 51). 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 averaged into one.

9.1.2 Output files

Not all output frames listed here are always produced. Some of them are created only on request (see Section 9.1.3, page 51), and some other are never created in case of LSS or LSS-like data.[7] Here is the list of all the possible output frames, in alphabetical order, together with a list of related configuration parameters.[8] 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_MXU will become DELTA_IMAGE_LONG_MOS.

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

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

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

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

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

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

Typical tracing residuals are not greater than 0.3 pixels (see Figure 9.1.6, page 54).

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

[7] 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.

[8] See Section 9.1.3, page 51, for a complete description of the recipe configuration parameters.

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 spectral curvature, and that this can be obtained without knowing the absolute positions of the traces. In summary, observing extracted spectra[9] that include signal from other spectra and/or extending beyond their true spatial extension, doesn't imply that the spectral curvature was not properly removed. As a matter of fact nearby spectra *do* sometimes contaminate each others 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 much appropriately as possible.

The only real solution to this problem would be to design masks where spectra are always well separated from each others (a buffer zone of 3 or 4 pixels would be sufficient).[10]

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

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

DELTA_IMAGE_MXU: deviation from the linear term of the fitting wavelength calibration polynomials. This image is used together with the DISP_RESIDUALS_TABLE_MXU to allow some quality control of the obtained solutions (see Figure 9.1.1).

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

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

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

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

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

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

[9] See entry REDUCED_LAMP_MXU in this Section, or entries MAPPED_SCI_MXU and MAPPED_ALL_SCI_MXU on page 59.

[10] 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.

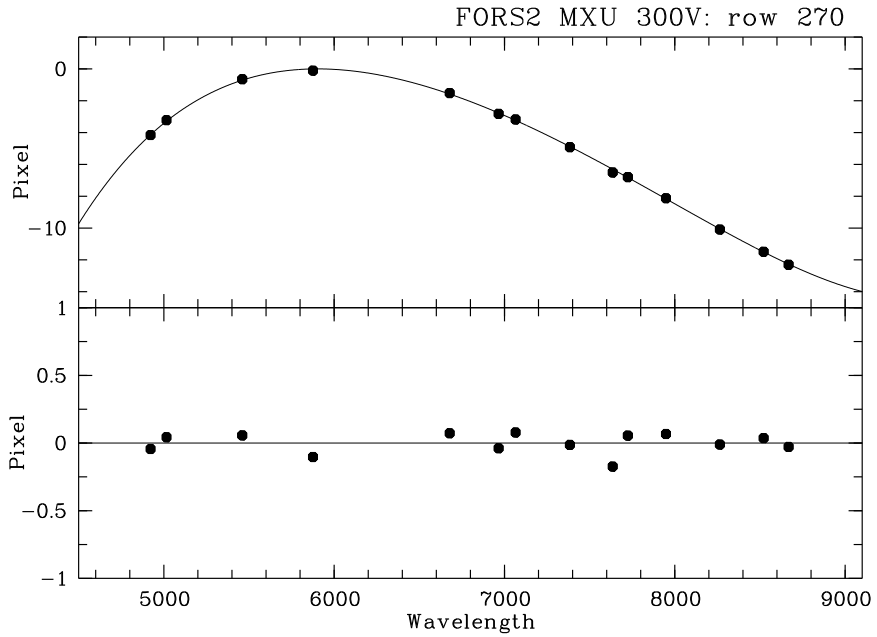

Figure 9.1.1: *Top panel: deviation of the identified peaks from the linear term of the 270th fitting polynomial (column d270 of the DISP_RESIDUALS_TABLE_MXU). The solid line is the polynomial model with the linear term subtracted, drawn from row 270 of the DELTA_IMAGE_MXU product. Bottom panel: fit residuals of the identified peaks (identical to the residuals recorded at row 270 of the DISP_RESIDUALS_MXU image).*

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

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

DISP_RESIDUALS_MXU: residuals of each wavelength calibration fit (in pixels). This image is just created if the *--check* configuration parameter is set. The residuals of the derived wavelength calibration with respect to the measured pixel positions of the reference arc lamp lines are collected in this image, with x pixels corresponding to the original CCD pixels, and y pixels corresponding to the REDUCED_LAMP_MXU pixels (*i.e.* to the rectified spatial coordinate, see figure 9.1.2). Typical observed residuals should be around 0.2 pixels.[11] 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 9.1.3, page 51).

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

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

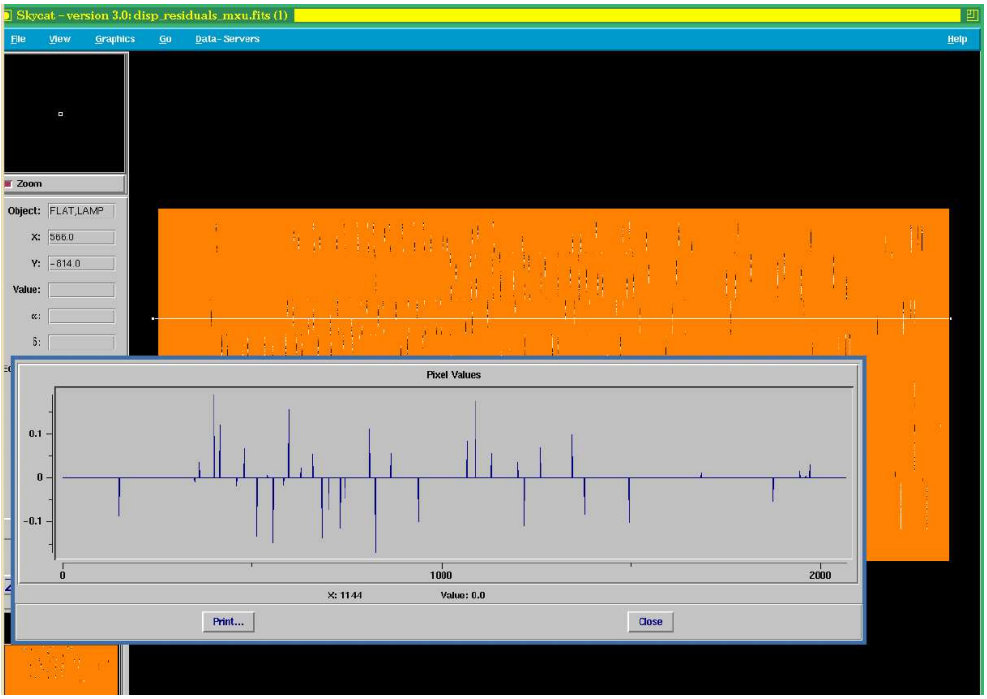

Figure 9.1.2: *RESIDUAL_MAP_MXU from a FORS2 MXU 600RI arc lamp calibration. In the foreground is a plot of the residuals from one image row.*

polynomial fit, *i.e.* residuals larger than the threshold specified with the configuration parameter *--wreject* (see Section 9.1.3, page 51). Just one every 10 of the polynomial fits listed in 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 9.1.1).
p<*row*>: x pixel position of reference lines on CCD.

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

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

GLOBAL_DISTORTION_TABLE: table containing the modeling of the coefficients of the local distortion models listed in the DISP_COEFF_MXU and the CURV_COEFF_MXU tables. It is produced only if the configuration parameter *--slit_ident* is set, and at least 12 spectra are found on the CCD. This table

is currently used for quality control, and to support the on-line quick-look scientific data reduction. See Section 7 for more details.

MASTER_BIAS: master bias. This frame is only produced if a set of raw BIAS frames is given in input instead of a MASTER_BIAS (see previous Section, page 41).

MASTER_NORM_FLAT_MXU: normalised flat field image, derived dividing the master screen flat by its smoothed version (see the smoothing configuration parameters description in Section 9.1.3, page 51). Comparing this image with the MASTER_SCREEN_FLAT_MXU may give an immediate feeling of the goodness of the computed curvature model used for the extraction of the normalised spectra.

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

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

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

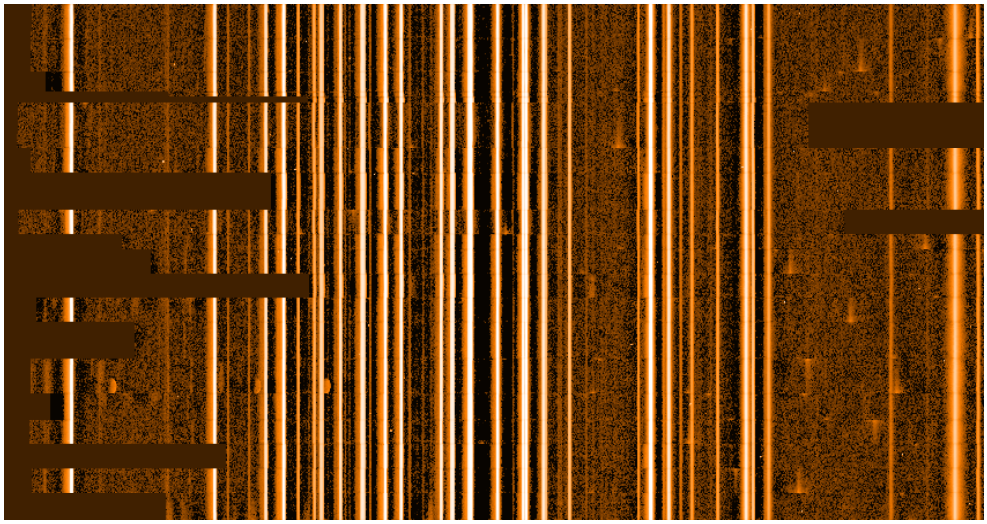

Figure 9.1.3: *REDUCED_LAMP_MXU from a FORS2 MXU 300I arc lamp exposure.*

REDUCED_LAMP_MXU: rectified and wavelength calibrated arc lamp image (see Figure 9.1.3). 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 and background subtracted. Its x size depends on the spectral extraction range $(\lambda_{min}, \lambda_{max})$ and on the value used for the dispersion in wavelength units per pixel, D, defined by the configuration parameter *--dispersion* (see Section 9.2.3, page 62):

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

The y size of this image matches the y size of the exposed part of the CDD in the case of LSS os LSS-like data. In the case of multi-spectra observations the y size of this image is equal to the total number of spatially rectified pixels: each slit spectrum is extracted between the traces of its top and bottom

edges (see products CURV_TRACES_MXU and CURV_COEFF_MXU), and spatially remapped into a constant number of pixels at each x CCD coordinate. The number of rectified pixels for the i-th slit spectrum is computed as

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

where t_i and b_i are the y CCD coordinates of the i-th slit spectrum edges at the position of the grism central wavelength.[12] 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 the interpolation point with respect to the original CCD pixels.[13] This is especially evident in the case of LSS (REDUCED_LAMP_LSS) or LSS-like data (REDUCED_LAMP_MOS with all slits at the same offset): even if the reference lines of the resampled spectra will appear perfectly straight on the rectified image, the signal level along an image column corresponding to an arc lamp reference line will appear to follow a wavy pattern. This reminds that the resampling of scientific data is not always acceptable (depending on the scientific aim of a specific observation program). A detailed analysis of the scientific signal should be based on the unrebinned data matched with the corresponding wavelength map – see entry WAVELENGTH_MAP_MXU).

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

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

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

ESO INS MOS<slit_id> <parameter>

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

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

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

ESO INS MOS3 POS

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

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

ESO INS MOS1<slit_id> <parameter>

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

ESO INS MOS142 POSANG

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

The slits location table columns are the following:

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

If the slit identification task is not run (see configuration parameter *--slit_ident*, Section 9.2.3, page 62), or if the slit identification task fails (*e.g.*, in the case of just two slits) the *slit_id* is set to the slit sequence number in the top–bottom ordering of the spectra on the CCD: but in order to avoid confusion with tags assigned to identified slits, a negative integer is used in this case instead of a positive one.

SLIT_MAP_MXU: map of central wavelength on the CCD. This image is just created if the *--check* configuration parameter is set, and only in case the data are not LSS or LSS-like. It has the same size of the WAVELENGTH_MAP_MXU image, from which it is derived. This product can be seen as an image of the mask cast on the CCD (see step 5 in Section 2, page 68): 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 2).

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

SPATIAL_MAP_MXU: map of spatial positions on the CCD. This image is not produced for LSS or LSS-like data. It has the same size of the CCD, where to each pixel is assigned the value of its distance (in CCD pixels) from the top edge of the spectrum it belongs to (see Figure 9.1.4). In case of confusion between nearby spectra, the spatial coordinate would just reflect the spectral 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*.

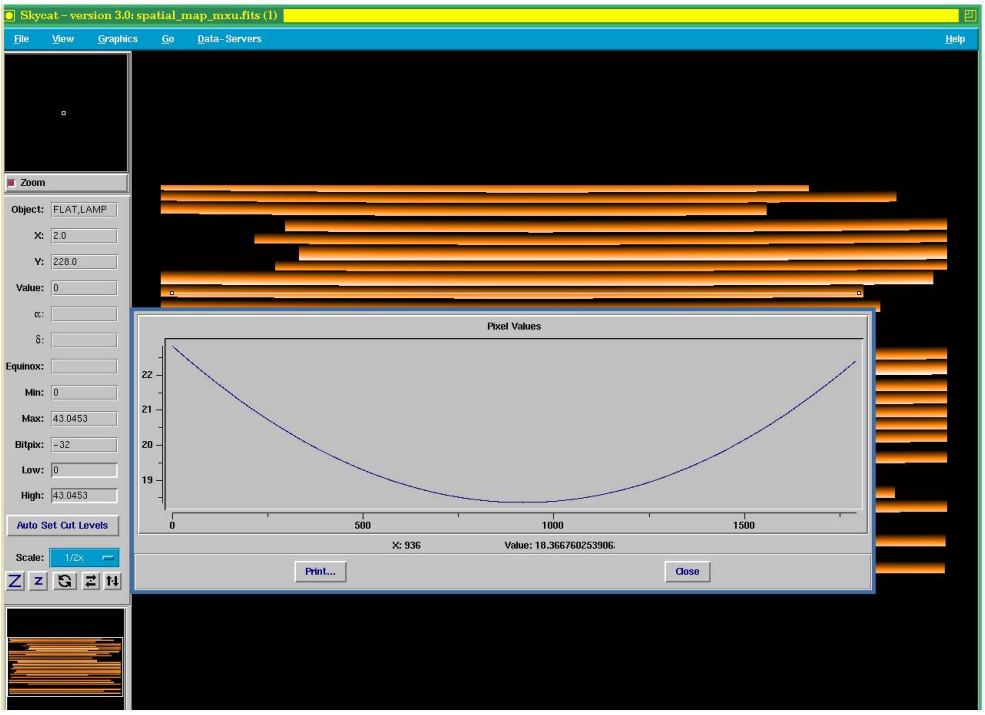

Figure 9.1.4: *SPATIAL_MAP_MXU from a FORS2 600RI flat field tracing, modeled with a 2nd degree polynomial. In the foreground is a plot of the distances from the top spectral edge of all pixels from one CCD row.*

SPECTRA_DETECTION_MXU: result of the preliminary wavelength calibration applied to the arc lamp exposure. This image is just created if the *--check* configuration parameter is set, and only in case the data are not LSS or LSS-like.[14] 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 2, page 68). 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).

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

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

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

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

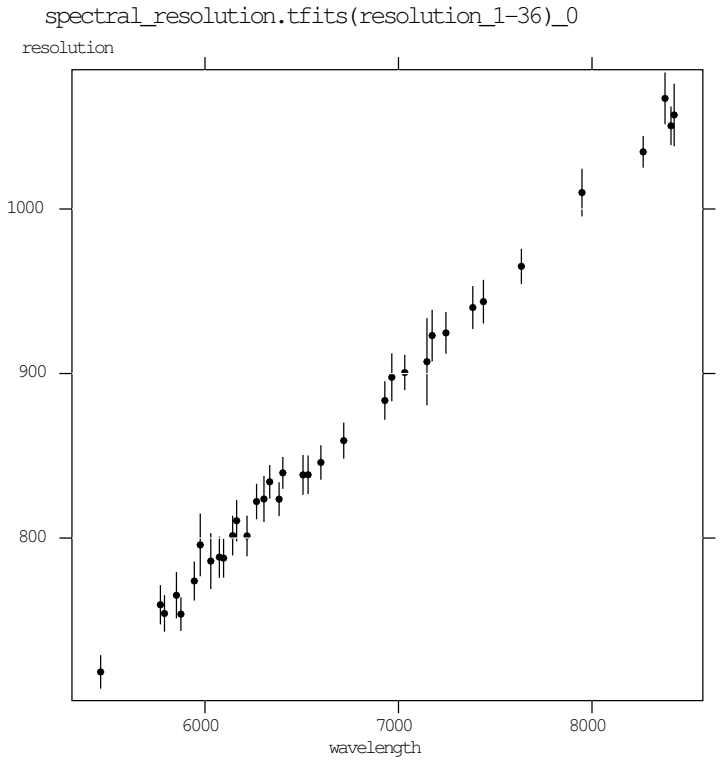

Figure 9.1.5: *Resolution vs. wavelength in a SPECTRAL_RESOLUTION_MXU table derived from a FORS2 MXU 600RI arc lamp exposure.*

WAVELENGTH_MAP_MXU: map of wavelengths on the CCD. This image has the same size of the CCD, where to each pixel is assigned 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*.

9.1.3 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, spectral 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 10.3, page 71. The dispersion values listed in the FORS1+2 User Manual [7] 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.[15]

Optimal values for this parameter, depending on the applied grism, are included in the GRISM_TABLE (see previous Section, page 41). 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.[16] 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 46).

--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 10.2, page 69): 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.[17] Optimal values for this parameter, depending on the applied grism, are included in the GRISM_TABLE (see previous Section, page 41).

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

[16] The rebin factor along the dispersion direction is written to the FITS header keyword ESO DET WIN1 BINX. The FORS1 chip is generally read with a rebin factor 1, while the FORS2 chips are read with a rebin factor 2.

[17] 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.

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

See the *--endwavelength* parameter.

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

This parameter, together with the *--startwavelength* parameter, defines the wavelength interval where calibration is attempted: this interval may not be entirely contained in the CCD for all spectra. Default values of the extraction interval, depending on the applied grism, are included in the GRISM_TABLE (see previous Section, page 41). If both *--startwavelength* and *--endwavelength* are left to 0.0, the extraction interval is computed automatically as the interval between the first and the last identified arc lamp reference lines, extrapolated by 10% at its blue and red ends (see Section 10.4, page 74).

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

This parameter is mandatory (using the default 0 would generate an error message). The degree used for the wavelength calibration polynomial should be the lowest that would provide non-systematic residuals to the solution (see the DISP_RESIDUALS_MXU entry, page 44). Optimal values of the extraction interval, depending on the applied grism, are included in the GRISM_TABLE (see previous Section, page 41).

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

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

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

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

If this parameter is greater than zero, the peak identification is iterated using the pattern-matching solution as a first-guess model: the wavelengths listed in the input line catalog are transformed to CCD pixel positions using the model, and a peak is searched within the specified search radius.[18] 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 10.2, page 69).

[18] 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.

--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 global model is applied to the data if *--wmode* = 0.

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

This is the name of the MASTER_LINECAT table column where the arc lamp reference wavelengths are listed. The default is the name of the relevant column in the standard line catalog (see also the MASTER_LINECAT entry on page 41). This would allow the usage of any FITS table, supplied by the user, containing a list of wavelengths to be processed by the pattern-recognition task. The only requirement is that the listed wavelengths are given in Ångstrom, and that they are sorted from blue to red.

Spectral curvature calibration

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

This parameter is mandatory (using the default 0 would generate an error message). In general a 2nd degree polynomial gives good results. Optimal values depending on the applied grism are included in the GRISM_TABLE (see previous Section, page 41).

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 9.1.6). The systematic residuals are due to the changing pixelisation of the spectral edges on the CCD, and therefore they should not be considered physical. A low degree polynomial fit appropriately circumvents this effect by cutting through such oscillations. Trying to fit such residuals with higher degree polynomials would lead to unstable and unreliable 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 spectral 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 isn't even applicable if a tracing is missing for both edges of a slit spectrum.[19] By setting *--cmode* = 1 a global trend of the curvature coefficients would be determined, allowing to derive a curvature model also for the spectral edges that are lacking a direct tracing. Setting *--cmode* = 2 would recompute the curvature

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

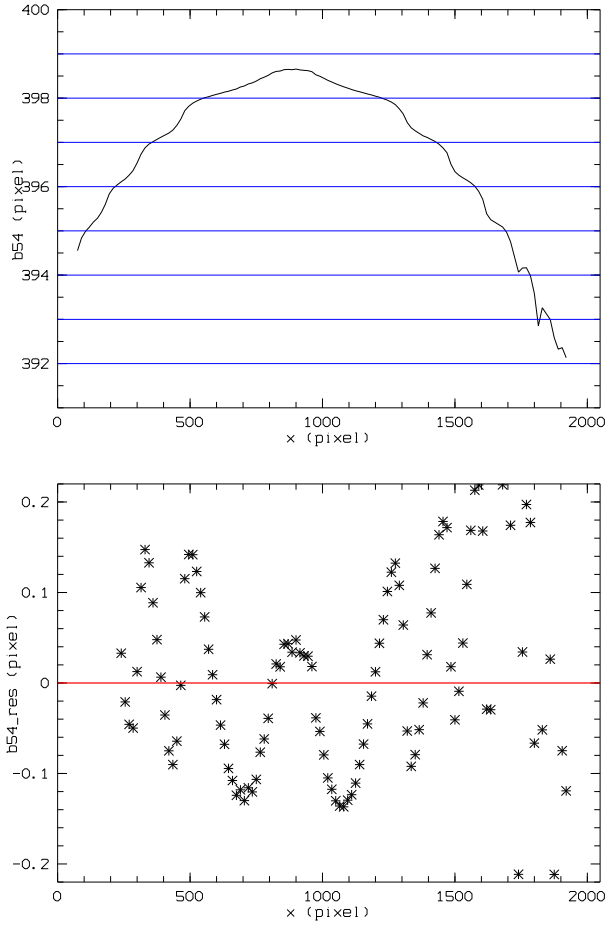

Figure 9.1.6: *Systematic residuals of curvature model (from a FORS2 MXU 600B flat field exposure).*

model also for the spectra where a local solution is available: this is generally not advisable, because a local solution is generally more accurate than the one derivable from the global solution.

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

This parameter does not affect the processing of LSS or LSS-like data. Setting this parameter activates the 2D pattern-recognition task linking the slits positions on the mask with those on the CCD (see Section 10.7, page 74). 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.[20] However, as shown in Section 10.7, 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:

[20]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.

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.[21] Note also that excluding the slit identification would also allow to reduce data from instruments different from FORS1 and FORS2.[22]

Flat field normalisation

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

This parameter only affects the processing of LSS and LSS-like data. If the configuration parameter *--sdegree* is set to a non-negative value, the master flat field normalisation is performed by modeling its large scale trend with a polynomial fitted along the spatial direction (and not along the dispersion direction, as for the case of shorter slits), for each CCD column. If *--sdegree* < 0 the illumination trend is obtained instead by median filtering with a running box of sizes *--dradius* and *--sradius*.

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

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

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

See the *--sradius* parameter.

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

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

Quality control

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

Setting this parameter will trigger the Quality Control parameters computation. This is just relevant for the on–line pipeline run automatically on Paranal, and for the off-line reduction of Service Mode observations performed by DFO. See Section 9.1.4 for a list of the computed parameters.

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

[22] 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.

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

Setting this parameter will write to disk some intermediate products of the data reduction procedure. This may be useful for debug purposes, in case of unsatisfactory results or failures of the *fors_calib* recipe. Currently the created intermediate products are SPECTRA_DETECTION_MXU and SLIT_MAP_MXU (see page 48).

9.1.4 Quality control parameters

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

QC 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_MXU table (see page 50), 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_MXU 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 LINES of the SPECTRAL_RESOLUTION_MXU table (see above).

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 MXU observations, but just for LSS observations made with the 1 arcsec slit, and for MOS observations having all slitlets at offset 0.0 (in the MOS case the parameter is named QC MOS CENTRAL WAVELENGTH). The central wavelength is written to the keyword ESO QC LSS CENTRAL WAVELENGTH of the wavelength map FITS header (see entry WAVELENGTH_MAP_MXU, page 50).

9.2 fors_science

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

In the following sections the MXU acronym in the products names can also be read MOS, or LSS, unless indicated otherwise. In the same way the word SCIENCE may be alternatively read STANDARD, as the spectroscopic standard stars exposures are reduced as scientific exposures. In case of standard star observations the SCI acronym in the products names should also be read STD.[23]

9.2.1 Input files

In alphabetical order:

CURV_COEFF_MXU: *required* table with spectral curvature coefficients, however *not required* for LSS and LSS-like observations.[24] This table is produced by the *fors_calib* recipe (see page 42).

DISP_COEFF_MXU: *required* table with wavelength solution coefficients. This table is produced by the *fors_calib* recipe (see page 43).

GRISM_TABLE: *optional* grism table. See Section 9.1.1, page 41.

MASTER_BIAS: *required* master bias frame. Just one should be given. This image may be produced by the *fors_calib* recipe (see page 41).

MASTER_NORM_FLAT_MXU: *optional* normalised flat field. This frame is produced by the *fors_calib* recipe (see page 46), and it must be provided only if the flat field correction is requested (see configuration parameter *--flatfield*, Section 9.2.3, page 62).

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 listing such wavelengths, whose name should be specified using the configuration parameter *--wcolumn* (see Section 9.2.3, page 62). If the alignment of the wavelength solution to the sky lines is requested, but a MASTER_SKYLINECAT is not specified in input, an internal sky line catalog is used instead.

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

[23] In this way it may happen that up to 9 different category names may be assigned to a product having exactly the same content in exactly the same format. For instance, the table carrying the dispersion coefficients referring to the spatially rectified spectra, may be assigned the following category names:

1. DISP_COEFF_MXU for MXU data reduced by *fors_calib*
2. DISP_COEFF_MOS for MOS data reduced by *fors_calib*
3. DISP_COEFF_LSS for LSS data reduced by *fors_calib*
4. DISP_COEFF_SCI_MXU for MXU scientific data reduced by *fors_science*
5. DISP_COEFF_SCI_MOS for MOS scientific data reduced by *fors_science*
6. DISP_COEFF_SCI_LSS for LSS scientific data reduced by *fors_science*
7. DISP_COEFF_STD_MXU for MXU standard star data reduced by *fors_science*
8. DISP_COEFF_STD_MOS for MOS standard star data reduced by *fors_science* and,
9. DISP_COEFF_STD_LSS for LSS standard star data reduced by *fors_science*.

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

SLIT_LOCATION_MXU: *required* table of slits positions. This table is produced by the *fors_calib* recipe (see page 47).

9.2.2 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.[25] Here is the list of all the possible output frames, in alphabetical order, together with a list of related configuration parameters:[26]

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 9.2.3, page 62). For a description of this product see the DISP_COEFF_MXU entry on page 43. In the DISP_COEFF_SCI_MXU table the *error* column content is computed by (quadratically) summing the errors of the input wavelength solution with the errors of the sky alignment fit. Similarly, in the *nlines* column the number of sky lines used for the alignment replaces the number of reference arc lamp lines on which the input calibration was based.

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

GLOBAL_SKY_SPECTRUM_MXU: table with supersampled sky spectrum, created only if the global sky subtraction is requested (see configuration parameter *--skyglobal*, Section 9.2.3, page 62). Each wavelength bin is half the resampling step, multiplied by the CCD readout rebin factor (see the configuration parameter *--dispersion*, Section 9.2.3, page 62).

The spectra contained in the input scientific exposure (see the SCIENCE_MXU entry on page 57) 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.[27] 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. To each bin is also assigned the median of the contributing wavelengths (which are not uniformely distributed within the bin). Empty bins are computed by linear interpolation between the nearest valid bins, and in this case to each bin is assigned its central wavelength.

The global sky table includes the following columns:

wavelength: Bin wavelength.
sky: Median signal level for each bin.
npoints: Number of points contributing to each bin.

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

[25] 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.

[26] See Section 9.2.3, page 62, for a complete description of the recipe configuration parameters.

[27] 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.

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 9.2.3, page 62):

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

The y size is determined in the same way as for the REDUCED_LAMP_MXU frame (see page 46).

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 46.

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

MAPPED_SCI_MXU: image with rectified, wavelength calibrated and sky subtracted slit spectra. This image matches in size the MAPPED_ALL_SCI_MXU image, and is produced only if any kind of sky subtraction (global and/or local) is requested.

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

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

MAPPED_SKY_SCI_MXU: image with rectified and wavelength calibrated slit sky spectra. This image contains the modeled sky that was subtracted from the scientific data, either before or after the scientific spectra rectification (or even both, if the configuration parameters *--skyglobal* and *--skymedian* were both set: the contribution of the global sky model is included in this image even if the global sky subtraction is really applied to the data before their rectification). The sky model component subtracted before the rectification of the scientific spectra can be viewed separately in the GLOBAL_SKY_SPECTRUM_MXU and the UNMAPPED_SKY_SCI_MXU products.

The MAPPED_SKY_SCI_MXU also includes the identified cosmic ray signal in case the cosmic rays removal is requested (see configuration parameter *--cosmics*, Section 9.2.3, page 62).

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

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

OBJECT_TABLE_SCI_MXU: This table is an expansion of the input SLIT_LOCATION_MXU table (see page 47), where the positions and the extraction spatial intervals of the detected objects are also included.[28] This table is produced only if any kind of sky subtraction (global and/or local) is requested, otherwise no object detection or extraction is attempted. The slits location table columns are the following:

slit_id:	Slit identification number.
xtop:	x CCD position of central wavelength from top end of slit.
ytop:	y CCD position of central wavelength from top end of slit.

[28] 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.

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.
end_1, end_2, ...:	End position of the extraction interval for each object.
row_1, row_2, ...:	Row number of the REDUCED_SCI_MXU image containing the extracted object spectrum. Image rows are counted from bottom, starting from 0.

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

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

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

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

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

REDUCED_SCI_MXU: image with extracted objects spectra. This image has the same x size of the image with the extracted slit spectra, MAPPED_SCI_MXU, and as many rows as the detected and extracted object spectra. Extracted spectra are written to the image rows listed in the OBJECT_TABLE_SCI_MXU table.

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

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

REDUCED_SKY_SCI_MXU: image with sky corresponding to the extracted objects spectra. The sky is extracted in the same way as the objects, *e.g.*, if optimal weights were applied to the object extraction, the same weights are applied to the sky extraction. This image matches the REDUCED_SCI_MXU image.

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

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

SKY_SHIFTS_LONG_SCI_MXU: table containing the observed sky lines offsets that were used for adjusting the input wavelength solution. This table is only produced if the sky lines alignment is requested (see configuration parameter *--skyalign*, Section 9.2.3, page 62), and is specific to LSS and LSS-like data (for multi-spectra observations the SKY_SHIFTS_SLIT_SCI_MXU table is produced instead).

This table has as many rows as the MAPPED_ALL_SCI_MXU image, and it contains two columns for each of the examined sky lines:

off_5577: Observed skyline offset (in pixel) at each CCD row. The integer truncation of each sky line wavelength is used for composing the corresponding column name (in this example the column name refers to the 5577.338 O I emission).
fit_5577: Modeling of the *off_5577* offsets.

This table may be very useful for judging what would be the most appropriate modeling of the observed offsets, and to what extent the input wavelength calibration really needs to be adjusted.

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

SKY_SHIFTS_SLIT_SCI_MXU: table containing the observed sky lines offsets that were used for adjusting the input wavelength solution. This table is only produced if the sky lines alignment is requested (see configuration parameter *--skyalign*, Section 9.2.3, page 62), and is specific to multi-spectra observations (for LSS and LSS-like observations the SKY_SHIFTS_LONG_SCI_MXU table is produced instead). This table has one row for each of the sky lines used for the alignment, and one column for each slit where sky lines could be detected.[29]

The included columns are the following:

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

This table may be very useful for judging what would be the most appropriate modeling of the observed offsets, and to what extent the input wavelength calibration really needs to be adjusted.

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

UNMAPPED_SCI_MXU: image with the sky subtracted scientific spectra on the CCD frame, created only if the global or the local sky subtraction is requested (see the configuration parameters *--skyglobal* and *--skylocal*, Section 9.2.3, page 62).[30] This image is derived subtracting the UNMAPPED_SKY_SCI_MXU from the bias subtracted and flat fielded scientific frame.

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

UNMAPPED_SKY_SCI_MXU: this image has the same size of the CCD, and is created if either the global or the local sky subtraction is requested.

If *--skyglobal* is set (see the configuration parameter *--skyglobal*, Section 9.2.3, page 62), 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

[29] In general the sky lines detection fails for reference slits, that are typically filled up by very bright objects.
[30] In the case of LSS or LSS-like data this image is just created with the *--skyglobal* option, because the *--skylocal* option is not available.

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 9.2.3, page 62), 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*, *--time_normalise*, *--startwavelength* and *--endwavelength*.

WAVELENGTH_MAP_SCI_MXU: This upgraded version of the wavelength map is only produced in case the adjustment of the wavelength solution to the sky lines is requested (see the configuration parameter *--skyalign*, Section 9.2.3, page 62). For a description of this product see the WAVELENGTH_MAP_MXU entry on page 50. Note that the WCS of the WAVELENGTH_MAP frames will generally differ, because they are derived from different input data: the WCS of WAVELENGTH_MAP_MXU is inherited from the arc lamp frame header, while the WCS of WAVELENGTH_MAP_SCI_MXU is inherited from the scientific frame header.

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

9.2.3 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, spectral curvature calibration, flat field correction, sky subtraction, objects detection and extraction, and flux calibration.

Wavelength calibration

--skyalign: Polynomial order for sky lines alignment. *Default*: 0

The input wavelength calibration can be adjusted to the observed positions of a set of sky lines, whose wavelengths are listed in an input catalog. The observed sky lines offsets from their expected positions (see entries SKY_SHIFTS_LONG_SCI_MXU and SKY_SHIFTS_SLIT_SCI_MXU, page 60) are fitted by polynomials that are then added to the input wavelength calibration polynomials (see DISP_COEFF_MXU entry on page 43). A *--skyalign* = 0 would just determine a median offset from all the observed sky lines, while *--skyalign* = 1 would try to fit a slope (rarely useful, but sometimes sky lines offsets display a significant dependency on the wavelength, due to a variation of the mean spectral dispersion with respect to the day calibrations). Polynomials with order greater than 2 generate a friendly error message. Setting *--skyalign* < 0 disables any sky line alignment, accepting the input wavelength calibration as-is.

Note that the *--skyalign* parameter should be more correctly intended as the *maximum* applicable polynomial order: the polynomial is really adapted to the number of identified sky lines used in the fit. As a rule, a polynomial with the specified order is only used if the number of identified sky lines is at least greater

than the number of free parameters: if this were not the case, the applied polynomial order would be

$$n = N - 1$$

where N is the number of identified sky lines. Consistently, for $n = 0$ a median offset would be computed.

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

This is the name of the MASTER_SKYLINECAT table column where the arc lamp reference wavelengths are listed. This would allow the usage of any FITS table, supplied by the user, containing a list of sky lines wavelengths to be used by the sky lines alignment task. The only requirement is that the listed wavelengths are given in Ångstrom, and that they are sorted from blue to red. Note that it is not necessary to provide a sky lines catalog to make the sky alignment work: see entry MASTER_SKYLINECAT on page 57 for more details.

Spectral 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 46, entry MASTER_NORM_FLAT_MXU). The flat field correction consists in dividing the bias subtracted input scientific frame by the normalised flat field frame.

Sky subtraction

--skylocal: Subtract sky spectrum from CCD scientific data. *Default*: TRUE

The local sky subtraction consists on modeling the sky trend for each row of spatial pixels for each spectrum on the CCD.[31] The advantage of this method is that the signal is not resampled before the sky is subtracted, reducing in this way the problems related to small-scale interpolation.

A MAPPED_SKY_SCI_MXU (page 59) 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 doesn't apply well to curved or tilted slits: in such case, the *--skymedian* option should be preferred. In case of LSS or LSS-like data *--skylocal* option is identical to the *--skymedian* option.

--skymedian: Subtract sky spectrum from rectified scientific data. *Default*: FALSE

[31] This is an iterative process: initially the sky trend is estimated with a robust linear fitting, then outliers (*e.g.*, objects) are rejected, and according to the slith length the sky is trended using a low degree polynomial.

The median sky subtraction consists on subtracting a median value of the sky for each wavelength pixel of each rectified slit spectrum.[32]

In general the subtraction of a rectified sky spectrum from rectified data doesn't 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 59) 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 doesn't 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 61 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*). Cosmics 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 59). Cosmics ray hits are removed anyway by the optimal extraction procedure of the detected objects.

Objects detection and extraction

--dispersion: Resampling step for rectified and wavelength calibrated spectra. *Default*: 0.0 Å/pixel

This parameter is mandatory (using the default 0.0 would generate an error message). The default value for this parameter, depending on the applied grism, is included in the GRISM_TABLE (see page 41): this value is the same that was used by the pattern-matching task for the reference lines identification, which is very close to the mean spectral dispersion (see Section 9.1.3, page 51). It is however possible to specify here any resampling step, if it is found more appropriate: in some extreme cases to resample the signal at a higher resolution may be essential to prevent information loss. To undersample the signal, on the other side, is never advisable (and it makes ineffective the flux conservation correction – see the *flux correction* section ahead).

The products that are directly affected by the *--dispersion* parameter are the following:

- GLOBAL_SKY_SPECTRUM_MXU,
- MAPPED_ALL_SCI_MXU,
- MAPPED_SCI_MXU,
- MAPPED_SKY_SCI_MXU,
- REDUCED_ERROR_SCI_MXU,
- REDUCED_SCI_MXU, and,

[32] 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.

- REDUCED_SKY_SCI_MXU.

Note that the *--dispersion* value must refer to the real CCD pixel size: the given value of the resampling step is internally multiplied by the rebin factor, to match the real pixel size of the input data.[33] In this way the value of the resampling step is made independent from the CCD reading mode, guaranteeing that the same supersampling factor is always applied.

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

See the *--endwavelength* parameter.

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

This parameter, together with the *--startwavelength* parameter, defines the wavelength interval to be extracted. Optimal values of the extraction interval, depending on the applied grism, are included in the GRISM_TABLE (see page 41). Both *--startwavelength* and *--endwavelength* must be otherwise specified (leaving them to 0.0 would generate an error message). It is generally not advisable to specify an extraction interval that is wider than the calibrated interval.

--slit_margin: Spectrum edge pixels to exclude from object search. *Default*: 3 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, *e.g.* objects would be truncated, too close to a confusion region, etc.

--ext_radius: Maximum extraction radius for detected objects. *Default*: 6 pixel

The default value is generally good when dealing with point-like objects, but it should be adapted to the size of more extended objects when necessary. Large values of the extraction radius would not harm the extraction quality if an optimal extraction algorithm is applied, but may have devastating effects on the results of a simple aperture extraction. The applied extraction interval is reduced in case nearby objects are detected: an intermediate position between two objects, computed according to the objects luminosity ratio, is never passed.

--cont_radius: Contamination radius. *Default*: 0 pixel

This parameter may help to prevent the extraction of contaminated objects. The contamination radius is the minimum distance at which two point-like objects of equal luminosity are assumed not to contaminate each other. For two objects having different luminosities the reciprocal contamination distances depend on their luminosity ratio. Indicating with L_o the peak value of one object integrated spatial profile and with L the peak value of a nearby object, the quantity

$$S = C \cdot \left(\frac{L}{L_o}\right)$$

is computed, where C is the specified contamination radius. If the distance between the two objects is less than S, the examined object is flagged as contaminated and is not extracted. This empirical formula has the effect of assigning a larger contamination radius to relatively brighter objects with respect to dimmer ones.

[33] The rebin factor along the dispersion direction is written to the FITS header keyword ESO DET WIN1 BINX. The FORS1 chip is generally read with a rebin factor 1, while the FORS2 chips are read with a rebin factor 2.

--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 (Horne, K., 1986, PASP, 98, 609).

Flux calibration

Currently only factors for flux conservation in the rebin operations is offered. No relative flux calibration based on a spectroscopic standard star observation is implemented yet.

--flux: Apply flux conservation factors. *Default*: TRUE

The flux conservation factors applied to the rebinned and extracted slit and object spectra is the ratio between the resampling step and the original signal sampling, along both the spatial and the dispersion directions.[34]

--time_normalise: Apply exposure time normalisation to relevant products. *Default*: TRUE

The following products are afftected by this parameter:

- GLOBAL_SKY_SPECTRUM_MXU,
- MAPPED_ALL_SCI_MXU,
- MAPPED_SCI_MXU,
- MAPPED_SKY_SCI_MXU,
- REDUCED_ERROR_SCI_MXU,
- REDUCED_SCI_MXU, and,
- REDUCED_SKY_SCI_MXU.
- UNMAPPED_SCI_MXU, and,
- UNMAPPED_SKY_SCI_MXU.

9.3 fors_extract

This recipe is used for reducing FORS1 and FORS2 LSS, MOS and MXU scientific spectra. It is identical to the recipe *fors_science*, with the only difference that a GLOBAL_DISTORTION_TABLE (see page 35) is required in input instead of the DISP_COEFF_MXU, CURV_COEFF_MXU, and SLIT_LOCATION_MXU tables (see pages 42, 43, and 47). 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.

[34] In order to have a view of the flux conservation correction, it is possible to run the *fors_science* recipe twice, first setting *--flux* = FALSE and then setting *--flux* = TRUE, finally computing the ratio between the corresponding MAPPED_ALL_SCI_MXU product images.

9.4 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 read-out). In the case of FORS2 frames the background is the median level evaluated from the available overscan regions. In the case of FORS1 data, 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.

9.4.1 Input files

FLUX_FLAT_LSS: exposure for flat field lamp monitoring.

FLUX_ARC_LSS: exposure for arc lamp monitoring.

9.4.2 Output files

FLUX_ARC_LSS: frame including the background subtracted integration region.

9.4.3 Configuration parameters

--xlow: X coordinate of lower left corner of integration region *Default*: 0 pixel

--ylow: Y coordinate of lower left corner of integration region *Default*: 0 pixel

--xhigh: X coordinate of upper right corner of integration region *Default*: 0 pixel

--yhigh: Y coordinate of upper right corner of integration region *Default*: 0 pixel

If the default is used (*i.e.*, all the configuration parameters are left to zero) the whole CCD is integrated.

10 Algorithms

The data reduction procedures applied by the pipeline recipes currently in use (see Section 4.1) are described here in some detail. This section is almost entirely drawn from the MOSES library manual, that is still in preparation.

10.1 Overview of the self-calibration procedure

A more detailed description of the instrument-independent tasks involved in the complete self-calibration procedure is given in the next sections. Here we just provide an overview, that is useful for setting 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 bias and background subtraction, examine the arc lamp exposure one row at a time. For each CCD row:
   (a) Run the 1D peak-detection task, to produce a list of reference arc lamp lines candidates.
   (b) Run the 1D pattern-recognition task, to select from the list of candidates a list of identified peaks. Not all the arc lamp lines are expected to be always identified, because the spectra are presumably distorted, and some CCD rows may cross a spectrum just partially, or even miss it entirely (see Figure 10.1.1).
3. Apply a preliminary wavelength calibration to each CCD row (within the wavelength range of the used line catalog, if not specified otherwise).
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.
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 spectral 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.[35]

11. If requested, and if there are enough slits, fit also a global model of the obtained coefficients, in order to improve the local solutions.

At this point the spectral extraction mask is completely determined, and for each spectrum a specific coordinate system is defined, where to a CCD pixel correspond a wavelength and a position on the telescope focal plane. If the instrument were stable, it would be possible to extract the scientific spectra applying directly this extraction mask. In general, however, the extraction mask obtained from the day calibration exposures should be aligned to the scientific spectra before being applied.

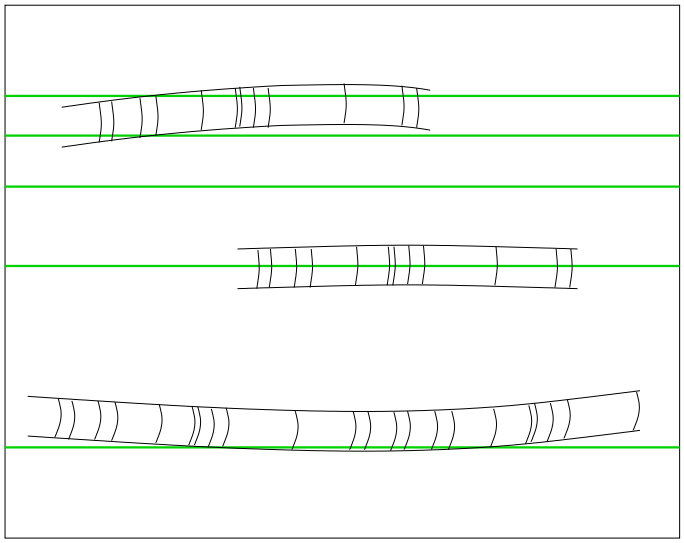

Figure 10.1.1: *CCD rows may not cut the whole range of the raw arc lamp spectra, because the spectra are not read along their curvature. However, even incomplete portions of the searched pattern can be identified by the pattern-matching algorithm.*

10.2 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.*[36]

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-

[35] This is the local wavelength calibration.

[36] Or at least long uninterruped portions of it.

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 10.2.1a).

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 10.2.1b). A peak position may be 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.

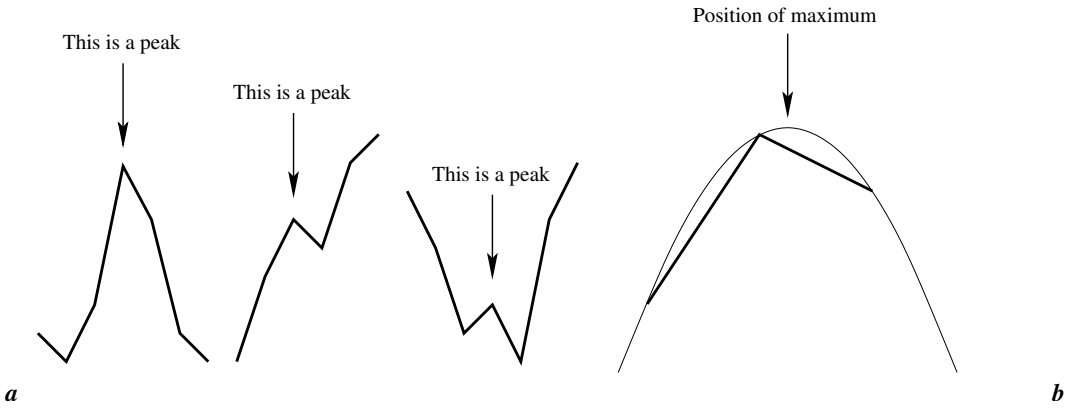

Figure 10.2.1: *a) Any local maximum identifies a peak. b) A peak position is determined by parabolic interpolation of three pixel values about the local maximum.*

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 10.3, page 71).

The position x of a peak is given by

$$x = x_o + R$$

where x_o is the (integer) position of the pixel corresponding to a local maximum, and R the offset corresponding to the position of the maximum obtained by parabolic interpolation:

$$R = \frac{1}{2}\left(\frac{v_1 - v_{-1}}{2v_o - v_1 - v_{-1}}\right)$$

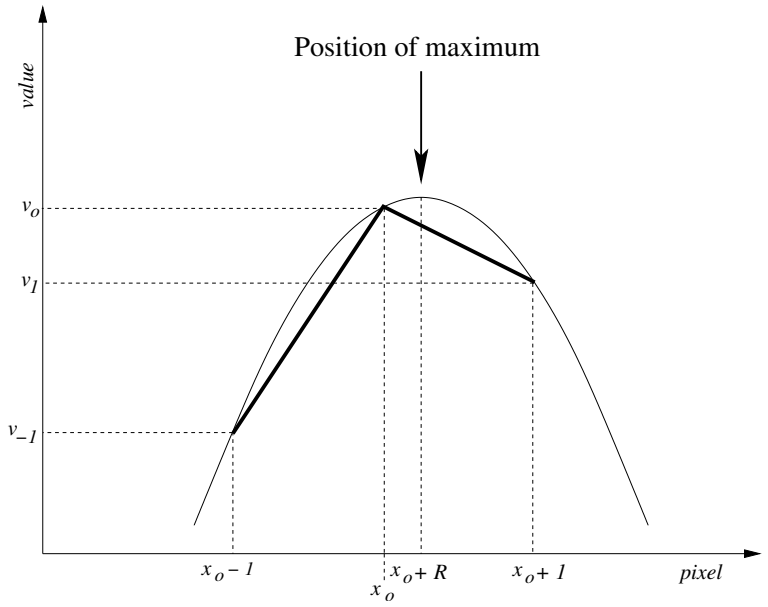

Figure 10.2.2: *Peak position estimate.*

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 10.2.2).

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).[37]

The drawback of this method is that the obtained peak positions do not reach the level of accuracy that can be obtained with other methods:[38] even in the case of well isolated peaks, a standard error of about 0.3 pixels is typical. This is why it is always advisable to improve, whenever possible, the accuracy of a peak position.

10.3 1D pattern-recognition

A simple method for 1D pattern-recognition has been developed in the attempt to increase the robustness of the wavelength calibration in the VIMOS pipeline, despite the occasional 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.[39]

The line-pattern would be searched in the list of CCD positions of arc lamp lines candidates produced by the

[37] 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.

[38] Multi-gaussian fitting and computation of the line baricenter are theoretically capable of very accurate positioning of isolated peaks (up to one hundredth of a pixel, or even more).

[39] 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.

1D peak-detection task (see Section 10.2, page 69). 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.[40]

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.[41]

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 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 10.3.1):

$$p_{min} = p_j + \frac{\lambda_{i+1} - \lambda_i}{d + \Delta d}$$

$$p_{max} = p_j + \frac{\lambda_{i+1} - \lambda_i}{d - \Delta d}$$

A *backward* search interval is similarly defined:

$$p_{min} = p_j - \frac{\lambda_i - \lambda_{i-1}}{d - \Delta d}$$

[40] 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.

[41] Note that, as said above, it is typically $N > W$, or even $N >> W$.

$$p_{max} = p_j - \frac{\lambda_i - \lambda_{i-1}}{d + \Delta d}$$

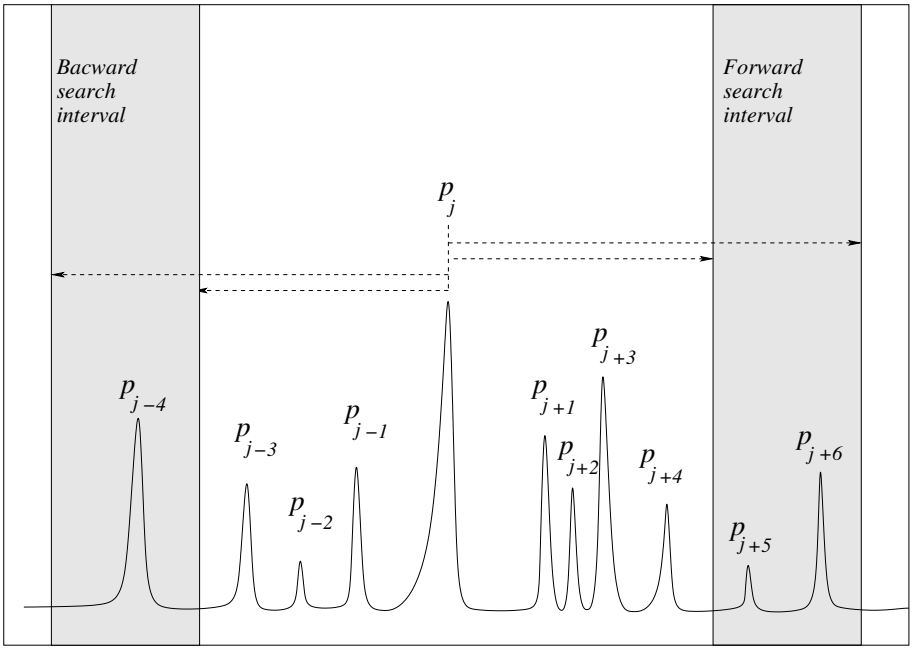

Figure 10.3.1: *Given a peak p_j, a forward and a backward search intervals compatible with an expected value of the spectral dispersion are defined. In this picture, peak positions p_{j-4}, p_{j+5} and p_{j+6} are used in the computation of the distance ratios to be compared with the wavelength interval 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,[42] the three peak 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 accidental, and therefore not confirmed by successive comparisons.[43] 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

[42] 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.

[43] 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.

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 catalogs are correctly identified without relying on a pre-existing instrument distortion modeling.

10.4 Determination of the spectral range

An automatic definition of the spectral range may be obtained from the lowest and the highest wavelengths reported by the 1D pattern-recognition task (see Section 10.3, page 71). This interval might be extended beyond those wavelengths, depending on the number of usable arc lamp lines contained in the interval, and by the degree of the polynomial used for the wavelength calibration.

An automatic computation of the spectral range is not advisable, however, because it doesn't guarantee that the same spectral range is determined for different data reduction sessions. This may have a negative impact on the instrument health monitoring (see next Section). It would therefore be appropriate to specify a fixed spectral extraction range in the system configuration files.

10.5 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 spectral curvature models. Typically, the 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, *e.g.* in instrument health monitoring, it would then be advisable to specify the preferred reference wavelength in the DRS configuration files, or always use the grism central wavelength specified in the FITS headers of the data to be processed.

10.6 Position of the reference wavelength on the CCD

The main product of the 1D pattern-recognition task run on the extracted CCD rows (see Section 2, page 11) is a boolean image, where all the pixels including the reference wavelength are flagged. After applying morphological operators for reducing the impact of occasional gaps in the data, the baricenter of all the connected regions of flagged pixels is computed. The computed coordinates on the CCD should correspond to the positions of the slit centers on the mask plane. The match between the two sets is determined by the 2D pattern-recognition task (see next Section).

10.7 2D pattern-recognition

The 2D pattern-recognition method applied here is based on a point-matching algorithm, and it is used for matching positions on the telescope focal plane (mask) with positions on the instrument focal plane (CCD). It

will then be possible to determine the transformation between the two coordinate systems, and to match each spectrum with its slit.[44]

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,[45] 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 10.7.1). 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.[46] 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,

$$S = \frac{(L_1)_P}{(L_1)_D}$$

Alternatively, an expected scale factor may be specified in the DRS configuration files.

At this point the complete list of triangles is revisited, eliminating all the matches that are incompatible with the found scale factor.[47] Finally, a rotation angle is computed for each matching pair, and incompatibilities with the median rotation angle are eliminated as well.

From the surviving triangles a list of matching points can be drawn and the geometrical transformation between the two sets can be determined. With the fitted transformation, points that were possibily lost to the matching procedure may be recovered, and a better transformation obtained from the extended sample.

It should be noted that this procedure, like the human brain, fails for regular grids of points: in fact in this case there would be no bin in the $\alpha - \beta$ plane containing just one triangle pair. Regular grids of points are typical of MOS calibration masks, but such masks always contain at least one asymmetric point, misaligned with the rest of the grid. This single point would be sufficient to create a great number of unique triangles, making this procedure work.

A possible drawback of this method lays 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

$$N = \binom{n}{3} = \frac{n(n-1)(n-2)}{6}$$

[44] 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.

[45] Imposing a reading order to the triangle sides eliminates the reflection invariance of the computed quantities.

[46] 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.

[47] 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.

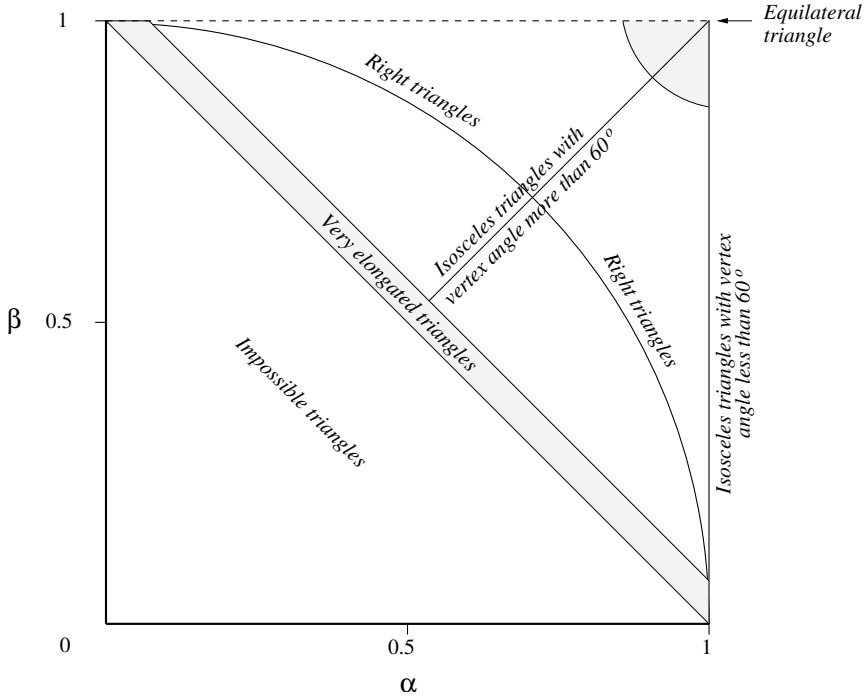

Figure 10.7.1: *The $\alpha - \beta$ plane. The long shaded region indicates very elongated triangles (including the cases of aligned points), while the region about the equilateral triangle includes ambiguous cases that would not lead to a safe identification of points: the triangles contained in those regions are therefore excluded from the analysis (unless they turn out to be the only triangles available). Note that the coordinates are cyclical: the line $\beta = 1$ (dashed) would include the same triangles described by the line $\alpha = 1$.*

A 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.

10.8 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 7.3.

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 not identified.

Note that slit identification is not essential to the data reduction, and it is hardly a requirement when very few

slits are in use.

10.9 Tracing slit spectra edges

The spectral curvature is determined by tracing the slit spectra – typically from flat field and scientific exposures. Flat field spectra are ideal for this operation, because the signal is continuous and with high S/N ratio; on the other hand, it is generally necessary to trace also the scientific spectra, to compensate for possible instrument instabilities. Scientific spectra are generally traceable, because the exposure times are typically long enough to produce a very bright sky spectrum. In case the sky emission is not traceable, then the curvature model derived from the flat field exposures must be used.[48]

Tracing spectral edges is not a simple task, because the slit spectra are not always so well detached and isolated from each other, and edges from different spectra may overlap. The only possibility is to try to determine a global trend of the spectral curvature based on the well traceable edges, in order to obtain the curvature also where it cannot be directly measured (see next Section).

10.10 Spectral curvature model determination

A local spectral 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 7.3.

10.11 Extraction of slit spectra

The extraction of slit spectra consists in reading the spectra following their curvature. The extracted spectra are not wavelength calibrated. This extraction method is only applied to arc lamp or sky spectra before using them for determining the (local) wavelength calibration applying the 1D peak detection and pattern-recognition methods described in Sections 10.2 and 10.3.

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 spectral curvature is eliminated. In other words, the x coordinate of the rectified image is still the x coordinate of the CCD.

10.12 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.

[48] Tracing bright point-like object spectra is not a solution, as they are not distorted just by optics, but by atmospheric refraction too.

Flexures, non-reproducible mask positions, grism rotation, temperature changes, and other unpredictable effects, have a complex impact on each of the extraction mask components - *i.e.*, the optical distortion model, the curvature model, and the wavelength calibration.

A practical approach to this problem might be to use the sky slit spectra for deriving a second extraction mask, following exactly the same procedure described in the previous sections – where a 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 spectral curvature and spectral dispersion models can also be derived.[49]

[49] Analogous alignment methods are already applied in the VIMOS IFU pipeline.

A Installation

This chapter gives generic instructions on how to obtain, build and install the FORS spectroscopic pipeline. A pipeline for imaging data reduction is not yet prepared for public distribution.

Even if this chapter is kept as up-to-date as much as possible, it may not be fully applicable to a particular release. This might especially happen for patch releases. One is therefore advised to read the installation instructions delivered with the FORS pipeline distribution kit. These release-specific instructions can be found in the file `README` located in the top-level directory of the unpacked FORS pipeline source tree. The supported platforms are listed in Section A.1. It is recommended reading through Section A.2.2 before starting the installation.

A bundled version of the FORS spectroscopic pipeline with all the required tools and an installer script is available from `http://www.eso.org/pipelines/`, for users who are not familiar with the installation of software packages.

A.1 Supported platforms

The utilisation of the GNU build tools should allow to build and install the FORS pipeline on a variety of UNIX platforms, but it has only been verified on the VLT target platforms:

- Linux (glibc 2.1 or later),
- Sun Solaris 2.8 or later,

using the GNU C compiler (version 3.2 or newer).

A.2 Building the FORS spectroscopic pipeline

This section shows how to obtain, build and install the FORS spectroscopic pipeline from the official source distribution.

A.2.1 Requirements

To compile and install the FORS spectroscopic pipeline one needs:

- the GNU C compiler (version 3.2 or later),
- the GNU `gzip` data compression program,
- a version of the `tar` file-archiving program, and,
- the GNU `make` utility.

An installation of the Common Pipeline library (CPL) must also be available on the system. Currently the CPL version 2.1.1 or newer is required. The CPL distribution can be obtained from http://www.eso.org/cpl.

Please note that CPL itself depends on an existing qfits installation. The qfits sources are available from the CPL download page or directly from the qfits homepage at http://www.eso.org/projects/aot/qfits. In conjunction with CPL 2.1.1 qfits 5.3.1 must be used, while CPL 3.0 is matched by qfits 6.0.0.

In order to run the FORS spectroscopic pipeline recipes a front-end application is also required. Currently there are two such applications available, a command-line tool called *Esorex* and the Java based data file organizer, *Gasgano*, which provides an intuitive graphical user interface (see Section 4.2, page 13). At least one of them must be installed. The *Esorex* and *Gasgano* packages are available at

http://www.eso.org/cpl/esorex.html
http://www.eso.org/gasgano

respectively.

For installation instructions of any of the additional packages mentioned before please refer to the documentation of these packages.

A.2.2 Compiling and installing the FORS spectroscopic pipeline

The FORS spectroscopic pipeline distribution kit 1.0 contains:

fors-manual-1.0.pdf	The FORS pipeline manual
install_pipeline	Install script
qfits-6.0.0.tar.gz	QFITS 6.0.0
cpl-3.0.0.tar.gz	CPL 3.0.0
esorex-3.6.0.tar.gz	esorex 3.6.0
gasgano-2.2.3-Linux.tar.gz	GASGANO 2.2.3 for Linux
gasgano-2.2.3-SunOS.tar.gz	GASGANO 2.2.3 for SunOS
fors-3.0.5.tar.gz	FORS 3.0.5
fors-calib-3.0.5.tar.gz	FORS calibration files 3.0.5

Here is a description of the installation procedure:

1. Change directory to where you want to retrieve the FORS spectroscopic pipeline recipes 3.0.5 package. It can be any directory of your choice but not:

   $HOME/gasgano
   $HOME/.esorex

2. Download from the ESO ftp server, http://www.eso.org/pipelines/, the latest release of the FORS spectroscopic pipeline distribution.

3. Verify the checksum value of the tar file with the cksum command.

4. Unpack using the following command:

   ```
   tar -xvf fors-kit-1.0.tar
   ```

   Note that the size of the installed software (including *Gasgano*) together with the static calibration data is about 5Mb.

5. Install: after moving to the top installation directory,

   ```
   cd fors-kit-1.2
   ```

   it is possible to perform a simple installation using the available installer script (*recommended*):

   ```
   ./install_pipeline
   ```

   (beware: the execution may take a few minutes on Linux and several minutes on SunOS).

   By default the script will install the FORS recipes, *Gasgano*, *Esorex*, all the necessary libraries, and the static calibration tables, into a directory tree rooted at `$HOME`. A different path may be specified as soon as the script is run.

   The only exception to all this is the *Gasgano* tool, that will always be installed under the directory `$HOME/gasgano`. Note that the installer will move an existing `$HOME/gasgano` directory to `$HOME/gasgano.old` before the new *Gasgano* version is installed.

   Important: the installation script would ensure that any existing *Gasgano* and *Esorex* setup would be inherited into the newly installed configuration files (avoiding in this way any conflict with other installed instrument pipelines).

Alternatively, it is possible to perform a manual installation (*experienced users only*): the `README` file located in the top installation directory contains more detailed information about a step-by-step installation.

B Abbreviations and acronyms

ANSI	American National Standards Institute
ASCII	American Standard Code for Information Interchange
CalibDB	Calibration Database
CPL	Common Pipeline Library
DFO	Data Flow Operations department
DFS	Data Flow System department
DMD	Data Management and Operations Division
DRS	Data Reduction System
ESO	European Southern Observatory
ESOREX	ESO-Recipe Execution tool
FITS	Flexible Image Transport System
GUI	Graphical User Interface
MOSES	Multi-Object Spectroscopy Empirical Self-calibration
OB	Observation Block
PSO	Paranal Science Operations
QC	Quality Control
RON	Read Out Noise
SOF	Set Of Frames
UT	Unit Telescope
VLT	Very Large Telescope

C Troubleshooting Guide

See Sections 9.1.3 and 9.2.3.