Observatoire de Genève
51, Ch. des Maillettes, CH-1290 Sauverny
Phone: +41 22 755 26 11 * Fax: +41 22 755 39 83

GIRAFFE

Detailed Design for BLDR Software

Doc. No. VLT-SPE-OGL-13730-0040
Issue 0.92
17 August, 2000

Working Draft post FDR
compiled on July 6, 2001 at 09:39

	Name	Date	Signature
Prepared	S. Baratchart, A. Blecha, V. Cayatte, L. Chemin, P. North, F. Royer, G. Simond	17 August, 2000	
Approved	A. Blecha	19 July 2000	
Released	A. Blecha	19 July 2000	

Change Record

Issue/Rev.	Date	Section/Page affected	Reason/Remarks
0.9	23 June, 2000	All	
0.91	19 July, 2000	Almost all	Includes RIXs answers as given for the FDR
0.92	17 August, 2000	Recipes, Data strucutres, DIC	1-st draft of post FDR working document

Table of Contents

1 Introduction

1.1 Organization of this document, level of treatment and completeness, how to read the document

Section 1 gives the reference information and some introductory and working remarks. **Note that this section only slightly differs from the Introduction in Functional Specifications (AD21) approved at the PDR and could therefore be skipped by an expert reader**. Section 2 is a brief description of the BLDRS design of the *environment* the data and the software structure. Sections 3 and sections 4, 5, **??** give the description of BLDRS functions of all classes and of recipes used during the standard observation and calibration runs. These are the most elaborated descriptions though still largely implementation independent. The section **??** gives the high-level general features of the design. In the section 6 all internal data representation and format are given. Note that this is not an exhaustive list of all data formats that are supported since an open-ended toolbox will provide the conversion tools from/to user data formats. Structurally this section (6) should be considered as an entity describing the interfaces. The section 9 outlines the foreseen implementation and deal with subject related to the development itself.

Since the document is also heavily used for the internal discussions of the GIRAFFE and FLAMES consortia, the authors of sections are identified within the text. If several names are given, the first name is to be contacted in priority. If a name is given at the subsection or paragraph, it applies only to this specific part of the text.

1.2 Purpose

The present document (DDD-BLDRS) specifies the functions provided by the Base-line Data Reduction Software (BLDRS), part of the Data Reduction Software (DRS), for the Giraffe Spectrograph which is a part of the VLT Multi Object Fibre Facility (called FLAMES) at UT2 telescope.

It provides the basis for the development of the BLDRS.

1.3 Scope

The FLAMES facility is made up of the fibre unit pick-up feeding the high resolution (10000-40000) spectrographs. The fibres may be fed by the light of individual objects (MEDUSA mode - 132 fibres with one fibre per object and 5 simultaneous calibration fibres), one single resolved object (ARGUS - 300 image elements on one object, 15 sky fibres and 5 simultaneous calibration fibres) or 15 small resolved objects (IFU - 20 image elements, 1 sky per object + 5 simultaneous calibration fibres). In MEDUSA mode the spectrograph could be Giraffe or UVES (8 objects - no simultaneous calibration fibres). The present specifications do not cover the UVES DRS.

The BLDRS should account for all instrumental effects and produce the flux and wavelength calibrated spectra in all standard observing modes.

The BLDRS will be complemented (not covered by this document) by the Ancillary Data Analysis Software (ADAS) which will provide the Data Analysis tools specific to the instrument such as High-precision Radial velocity package and image reconstruction in a given passband (ARGUS and IFU modes). However, at least crude image reconstruction will be provided by the BLDRS and by the Quick-Look Software for Argus and IFU modes.

The BLDRS will be used in two different ways.

1. The *on-line* DRS

   In the VLT DFS pipeline as a set of SW modules (to be integrated to the DFS under the responsibility of ESO) and called from MIDAS as external executable commands with appropriate set of parameters.

2. The *off-line* DRS

   The same SW as in *on-line* DRS is used but under different conditions; mainly:

   - High-level of interaction allowed
   - Preservation of the intermediate data possible
   - Extended set of commands available

In both cases the functions are identical. We assume that the DFS pipeline will not offer any interactivity and therefore some functions will not be used there.

Unlike the UVES coupling to the fibre positionner that carries light to an existing spectrograph not initially designed to that purpose, the GIRAFFE spectrograph is fully designed as a fibre-coupled instrument and integrates some original features. The most outstanding is the simultaneous calibration using the Th-Ar and Ne lamps. Though the primary motivation to commit permanently 5 spectra on each exposure to that purpose was the requirement of high accuracy of the wavelength solution, the by-product of this situation - the presence, in any mode, of five optimally exposed emission spectra that are close-to-uniformly distributed over the detector - is a substantial asset. At least 20 usable lines/spectrum are available in each spectral band. The number of lines and the scrambling effect of the fibres insure that the PSF variation both in form and position could be efficiently modeled for every single exposure. This is an exceptional opportunity to monitor most of instrumental parameters and greatly influences the design of the data reduction software since it may heavily rely on a very accurate localization.

1.4 Applicable and Reference Documents

1.4.1 Explanatory remarks

Through all GIRAFFE DRS documentation we keep the same numbering scheme.Since some documents referenced in previous GIRAFFE documents are not used in the present document the AD and RD numbering may be discontinuous.

1.4.2 Applicable Documents

The Applicable Documents give the information describing the work to be carried out, the available tools and the formal conditions of the work.

AD1 `INS-SPE-ESO-13730-1657,1.0` – GIRAFFE Technical Specifications
AD2 `VLT-SPE-ESO-10000-0004,1.0` – VLT Specification Environmental Conditions
AD3 `VLT-TRE-ESO-13740-1681,1.0 06/10/98` – GIRAFFE SPECTROGRAPH Optical Final Design
AD4 `VLT-PLA-ESO-13700-1788,1.0` – FLAMES IOCP - Instrument Operation and Calibration Plan
AD5 `INS-PLA-AUS-13721-0056 1.0 29/09/99` – FLAMES Fibre Positionner OpCalPlan
AD6 `GEN-SPE-ESO-19400-0794,1.1 25/11/97` – ESO DICD - Data Interface Control Document
AD7 `ARC-SPE-ESO-00000-0001,1.4 10/03/94` – ESO Archive - Data Interface Requirements
AD8 `VLT-SPE-ESO-19600-1233,0.3 15/10/96` – VLT DFS Specifications for Pipeline and Quality Control
AD9 `VLT-PLA-ESO-19000-1183,1.0 09/12/96` – VLT DFS Operations Model for VLT/VLTI Instrumentation
AD10 `VLT-SPE-OGL-13730-0020,2.0 02/10/99` – GIRAFFE User's Requirements for BLDR Software
AD11 `VLT-MAN-ESO-17240-0866,2.5 28/07/97` – Real Time Display - User Manual
AD12 ESO MIDAS November 1998 – Data Reduction Manual
AD13 `VLT-SPE-ESO-19000-1618,1.0 21/04/99` – Data Flow for VLT Instruments Requirements Specification
AD14 `VLT-SPE-ESO-19000-0749,1.0 ????????` – VLT On-line Data Flow Requirements Specification
AD15 `VLT-MAN-ESO-19500-1619 1.0 02/26/99` – Data Flow Pipeline and Quality Control - User's Manual
AD16 `VLT-MAN-ESO-19400-1785,1.0 23/02/99` – OLAS User's Guide
AD17 `VLT-SPE-ESO-19600-1217,1.0 ????????` – Data Flow System Quality Control Equipment Model API
AD18 `VLT-MAN-ESO-19500-1771,1.2 18/06/99` – FORS Pipeline & Quality Control - User's Manual
AD19 `VLT-MAN-ESO-19500-1772,1.0 25/02/99` – ISAAC Pipeline & Quality Control - User's Manual
AD20 `VLT-PLA-ESO-10000-0441,1.0 05/12/97` – Science Operation Plan
AD21 `VLT-PLA-ESO-13730-0030,1.1 09/11/99` – GIRAFFE Functional Specification for BLDRS Software

1.4.3 Reference Documents

The Reference Documents are *bibliographic references* in the classical sense of term. They supply the information complementary to the text.

RD1 `VLT-PLA-ESO-00000-0006,2.0 21/05/92` – VLT Software Management Plan
RD2 `VLT-PRO-ESO-10000-0228,1.0 10/03/93` – VLT Software Programming Standard
RD3 2dF User Manual, J. Bailay, K. Glazebrook, May 29, 1998
RD4 `VLT-TRE-VIRG-14616-0058,1.0 16/06/98` – VIMOS Final Design Review
RD5 Australis Concept Study Report, The Australis Consortium), K. Taylor, M. Colless et al., 27 Sept. 1996
RD6 `INS-TRE-OP-137400-0009,1.0 expected for 28/10/1999` – FLAMES simulator Design and Architecture Report, V. Cayatte
RD7 Horne K., 1986, PASP 98, 609 Optimal extraction
RD8 `GIR-TRE-OPM-XXXXX,1.0` – Qualité d'image en bord de ferrule, L. Jocou (OP), 27 July 1999
RD9 Sky PSF modeling – Viton & Milliard (LAS, Marseille), 1999, private communication
RD10 Baranne A., Queloz D., Mayor M., Adrianzyk G., Knispel G., Kohler D., Lacroix D., Meunier J.-P., Rimbaud G., Vin A., 1996, A&AS 119, 373 ELODIE
RD11 `GIR-OPT-OPM-137030-10,1.0 28 April 2000`, V. Cayatte, A. Blecha, P. North, Vignetting effect for the FLAMES instrument

1.5 Abbreviations, Acronyms and Glossary

ADAS	GIRAFFE Ancillary Data Analysis Software - Data Analysis tools specific to GIRAFFE instrument
ADU	CCD Analog-to-digital Unit - Units used to quantify the CCD signal intensity
ARGUS	GIRAFFE Integral Field Spectroscopy mode
AUSTRALIS	Australian project for fibre coupled spectrograph
BLDR	GIRAFFE Base Line Data Reduction
BLDRS	GIRAFFE Base Line Data Reduction Software
CRH	Cosmic Ray Hit (general term)
DAL	Data Access Layer - part of the GIRAFFE DRS
DFS	Data Flow Software - VLT software controlling the on-line data processing
DICB	Data Interface Control Board - VLT software
DMD	Data Management Division of ESO
DRS	Data Reduction Software (general term)
FF	Flat-Field (general term)
FITS	Flexible Image Transport System - file format for images and tables (general term)
FLAMES	VLT Multi Object Fibre Facility - UT2-VLT Instrument; GIRAFFE is a part of FLAMES
FORS	Focal Reducer Spectrograph - UT1-VLT instrument
FOV	Field of View (general term) used here in sense of Nasmyth FOV
FP	Fibre positionner - part of FLAMES instrument
FS-BLDR	Functional Specifications for GIRAFFE Base Line Data Reduction
FSFF	*Full-Slit* Flat Field - FF taken with entrance slit uniformly illuminated
GIRAFFE	Spectrograph - this instrument, part of FLAMES
GUI	Graphical User Interface - set of command windows controlling the DRS (general term)
IFU	Multi-Integral Field Unit mode of GIRAFFE (15 IFUs of 20 image element + sky each)
IOCP	Instrument Operation and Calibration Plan (general term) but referenced here for FLAMES
ISAAC	Infrared Spectrometer And Array Camera - UT1-VLT instrument
KW	Keyword (usually FITS KW) - part of the FITS format
LSQ	Least Square Fit (general term)
MEDUSA	Multi-Object Spectroscopy mode of GIRAFFE (one fibre per object)
MIDAS	Munich Image Data Analysis System - Standard ESO Image Processing System
MOS	VLT Multi-object slit spectrograph (general term)
NFF	*Narrow* Flat Field - FF taken with slit illuminated through fibres
OGL	Observatoire de Genève et l'Université de Lausanne
OP	Observatoire de Paris
PAF	Parameter File Format - VLT DFS standard format
PC	Process Control - part of the BLDRS, set of tools that control the BLDRS execution
PE	Processing Engine - core set of functions of BLDRS
PHFF	Photometric Flat Field - FF taken with CCD illuminated directly (without passing through spectrograph)
PSF	Point Spread Function - image resulting from monochromatic point-like illumination of the entrance-slit

QC0	Quality Control Level 0 - results form the comparison of current parameters with nominal values
QC1	Quality Control Level 1 - results form the comparison of trends within the Calibration Database with expected fluctuations
QLS	GIRAFFE Quick-Look Software - simplified reduction with purpose to asses the data quality
RON	Read-Out Noise - CCD noise measured on 0 time exposure or on the image overscan
RTD	Real-Time Display - VLT tool for display and graphical handling of images
RR	Reduction Recipe - DFS standard *scripts* linking the DRS function into a procedure executable by the DFS
SEWC	Separate Wavelength Calibration - Calibration made with all fibres illuminated by calibration lamp
SIWC	Simultaneous Wavelength Calibration - Calibration made with 5 dedicated fibres illuminated by calibration lamp
SL	Scattered Light - parasite light with unknown spectral distribution falling on the detector
SLM	Scattered Light Model - continuous model built from discontinuous SL measurements
SNR	Signal to Noise Ratio - (general term)
SOS	Flames Super Observing Software - SW controlling the FLAMES through requests to sub-systems
SPU	Spectrograph unit - GIRAFFE spectrograph
SW	SoftWare
TBC	To Be Confirmed
TBD	To Be Defined
TBU	To Be Updated by ...
UR	GIRAFFE User's Requirements - generic terms for UR-BLDRS, UR-ADAS
UT1-4	VLT Unit Telescope 1 to 4
UVES	U-V Echelle Spectrograph - Third UT2-VLT instrument coupled to Fibre Positionner of FLAMES (8 fibres)
VCS	VLT Control Software
VI(R)MOS	Visual (Infrared) Multi Object Spectrograph - UT3-VLT multi-slit deep low-resolution spectrograph due for 1 Quarter 2000 and 3 Quarter 2001 (IR part)
WLC	Wavelength Calibration - (general term)

1.6 Stylistic conventions

The following styles are used in **Mathematical description** section for:

functions parameters

`lowercaseTeletype` with each word without the first capitalized '`functionFirstParam`', for example: `darkThreshold`.
Corresponding LaTeX code:
`\texttt{darkThreshold}` or in math mode: `\mathtt{darkThreshold}`,

table names

`lowercaseTeletype` with each word without the first capitalized '`exampleTableName`', for example: `lineSkyMast`.
Corresponding LaTeX code:
`\texttt{lineSkyMast}` or in math mode: `\mathtt{lineSkyMast}`,

`table columns`

ALL_UPPERCASE_ITALIC prefixed with `'tableName:'` in the text, '`exTableName` :*FIRST_COLUMN*' for example: `lineSkyMast` :*WLEN*.
Corresponding LaTeX code:
`$\mathtt{lineSkyMast:}WLEN$` or in math mode: `\mathtt{lineSkyMast:}WLEN`,

`frame names`

lowercaseItalic with each word without the first capitalized '*exampleFrameName*', for example: *dbSubImage*.
Corresponding LaTeX code:
`$dbSubImage$` or in math mode: `dbSubImage`,

`frame keywords`

`ALL_UPPERCASE_TELETYPE` prefixed with *'frameName.'* in the text, '*exFrameName*.`FITS_KEYWORD`' for example: *dbSubImage*.`EXPTIME`.
Corresponding LaTeX code:
`$dbSubImage.\mathtt{EXPTIME}$` or in math mode: `dbSubImage.\mathtt{EXPTIME}`,

WREM: TB completed for function name

2 Overview

2.1 Introduction

The Data Reduction Software for GIRAFFE spectrograph relies on the specific GIRAFFE feature - the presence of five simultaneous calibration spectra - and the high instrument stability. The localization is adjusted for all exposures and several options are offered for the extraction. The sky subtraction is made via sky intensity distribution and extracted PSF modeling.

The number of software modules is kept low through the use of same modules for multi-objet (MEDUSA) and integral-field spectroscopy (ARGUS and IFU) during most of reduction steps. This is achieved through a careful parameterization of the reduction functions. Associated to the processed image, the numerical mask of *bad pixels* is initialized and progressively updated through the reduction process. The analysis software supports *flagged* pixels all through the analysis. The possibility to replace flagged-pixel at any stage of the data reduction is implemented.

The developed software relies only on UNIX environment and utilities available under UNIX public domain software and standard VLT tools. The link between the DRS and any specific data analysis system, if required, will be through FITS and ASCII data structures only.

2.2 Software Environment

2.2.1 Inputs

On the input, the BLDRS receives the following data:

- The raw images from the VLT acquisition pipeline.

  The acquisition pipeline is a part of the VLT DFS and is running under the control of the FLAMES Super Observing Software (SOS). The BLDRS communicates with the SOS through data only. All the necessary *registration data* (Objects ID, positions, magnitudes) are supplied via FITS keywords on raw images.

- The calibration data.

  The necessary calibration data and standard control parameters are extracted from the instrument database. Two sorts of calibration data are used:

  – The current calibrations.

    Current calibration files are the data obtained from the calibration pipeline. Most recent, standard or average data may be used. An example of these are the flat-fields, localization or wavelength solutions.

  – The calibration constants.

    These are the data which are not obtained from the normal calibration pipeline. They are not strictly speaking constant, but could only be changed during the maintenance run through a special calibration process or imported by an ad-hoc command. The calibration constants are set before any standard Data Reduction could proceed. An example of such constants are the table of the spectrograph optical parameters, the reference absolute spectrum of the calibration lamps, the tables of the spectral lines for the wavelength calibration and initial parameters for the processing algorithms.

- The control parameters.

  In order to make the automated processing possible and efficient, all options and controls are fully parameterized. A subset of comprehensible and widely accepted *classical* main parameters is put in evidence separately in order to facilitate the handling by technically inexpert user (with necessary astronomical background). An example is the level of the sigma-clipping, the method of the rebinning or the accepted level of the scattered light contamination.

There is no difference, except for the raw images and the control parameters access, between the use of the BLDRS within the VLT or MIDAS pipeline.

2.2.2 Outputs

The BLDRS produces the following data:

- Flux and wavelength calibrated spectra with optionally error and number of pixels in each bin

- Standard set of numerical values that characterize the results and are used for the long term performance monitoring and quality control
- Set of report/history files and graphics

Optionally all necessary calibration and control data can be exported for further off-line use.

2.2.3 Online mode

All the BLDRS modules will be used into the VLT DFS framework and launched via the Reduction Block scheduler. The DFS interface is defined in [AD8,AD9] and it is made of a set of ASCII files specifying:

1. the reduction rules definitions: `giraffe.dpr`,
2. how reduction rules apply to incoming frames: `giraffe.ehr`,
3. which basic data are required: `giraffe.bdd`,
4. what are the data products: `giraffe.pdd`,
5. what are the recipe description and parameters, for example: `gir_gen_mbias.rrd`,
6. the recipe implementation, for example: `gir_gen_mbias.rri`,
7. the recipe procedure itself, for example: `gir_gen_mbias.prg` for the MIDAS procedure and `gir_gen_mbias.sh` if it is a Unix shell script).

The DFS Data Organizer ensures that all necessary data, science and calibration frames, parameters and configuration files, as defined in the DFS interface file, are available for the reduction recipes and thus to the BLDRS modules.

Though the integration of the BLDRS modules into the DFS is not a part of the BLDRS workpackage, the *Recipe generation tools* (see section 4) will provide the way to automatically generate all necessary files listed above since there is almost one-to-one correspondence between the internal BLDRS structure and the VLT DFS.

2.2.4 Off-line mode

In this mode the BLDRS modules consisting of stand-alone programs will be chained by Unix shell scripts. These scripts will implement the same processing that the standard reduction recipes available in the VLT DFS.

For the off-line reduction process, all the necessary data products (observation and calibration frames, observation parameters, instrument data configurations, conditions and log files) will be extracted from the VLT OLAS or VLT science archive and will form the data-set delivered to the astronomer.
For maintenance or tests at the telescope, the required data will be available from disk or from VLT OLAS.

Though the large number of observing modes (high and medium resolutions in all spectral ranges in MEDUSA, IFU and ARGUS modes and for all fibre assemblies) will require a large number of calibration files, the organization of the data-set should provide information on how data products are related to each other. This can be achieved using a standard directory structure, file naming convention and/or data catalogs.

2.2.5 Software External Components

The following external components will be used by the GIRAFFE BLDRS:

Unix shell

CFITSIO Software library to handle FITS formatted files

Eclipse image processing library to handle FITS datacubes

PGPLOT 2-D/3-D graphics plotting library

perl Practical Extraction and Report Language

2.3 Quality Control

Virtually all high level functions return parameters that are used for the quality control. They are described in more details under *Architecture* (sections 3 and 4). There are therefore not specific QC Recipes (at least not within the DRS). Such recipes, if necessary, will be addressed in the Calibration Plan.

The quality control level 0 (QC0) is a part of the normal data reduction processing. This is done for all processed frames including the science frames in the DFS pipeline.

With the exception of the Flat Field (FF) calibration exposures, any GIRAFFE frame includes 5 optimally exposed spectra used for the simultaneous wavelength calibration. This is the opportunity for a full check in the context of the quality control level 1 (QC1) of the actual instrument performances compared to the nominal ones:

1. read-out noise and CCD offset,
2. number of cosmic hits and detector defects,
3. proportion of the scattered light,
4. spectra localization stability,
5. instrument throughput (integrated light per spectrum/calibration ,illumination)
6. wavelength solution stability and spectral resolution;

The detecting of trends is not a part of the DRS.

2.4 General features

2.4.1 Flagged Pixels nad associated masks

Associated to the processed image, the numerical mask of *bad pixels* is initialized and progressively updated through the reduction process. The analysis software supports *flagged* pixels all through the analysis. The possibility to replace flagged-pixel at any stage of the data reduction is implemented.

The mask is implemented as a 32 bit integer image, each bit-plane representing a dedicated logical mask associated to a specific function. It is used together with one or several `badPixFlag` which specify what plane has to be applied in each circumstance.

The actual assignation of `badPixFlags` to specific functions is dynamically configurable and flags are given to functions as parameters remaining therefore only symbolically adressed within this document.

2.4.2 Bad or unused fibres

Fibers and consequently the data associated to them through the slit geometry may have following status:

- Valid fibres (whatever may be the light source)
- Unused fibres, which may be either in parking position (no sky light) or left for sake of time efficiency at the current place on the plate.
- Dammaged fibres

The dynamic information on the fibre status is implicitely given by the binary `ozpoz` table through column `ozpoz` :*BUTTON*. Only valid fibres are listed here and it is complemented by the static `slitGeometry` table which lists all fibres for a given slit with `slitGeometry` :*BUTTON* used as a common reference.

During the preprocessing phase, data associated to all fibres recieve the same tratement whatever is their actual status.

Starting by localisation, only the spectra associated to valid fibres are processed. Extracted spectra of invalid fibres are set to predefined value (0 or null TBD) so as to made the data structures of the same dimension and compatible with the standard calibration data. A KW FIBRESKOi with list of invalid spectra is created during the extraction.

2.5 Standards

2.5.1 BLDRS data

All data needed or produced by the BLDRS modules will be in one of the two following format:

- FITS images and tables (ESO compliant)
- flat ASCII tabulator delimited tables (internal standard)

2.5.2 BLDRS modules

The BLDRS Software will be developed using C,C++ programming languages following the VLT programming standards guidelines. User interface will be developed in Tcl/Tk. All necessary scripts will be developed using a standard Unix shell (Bourne shell,C shell or Korn shell) or perl.

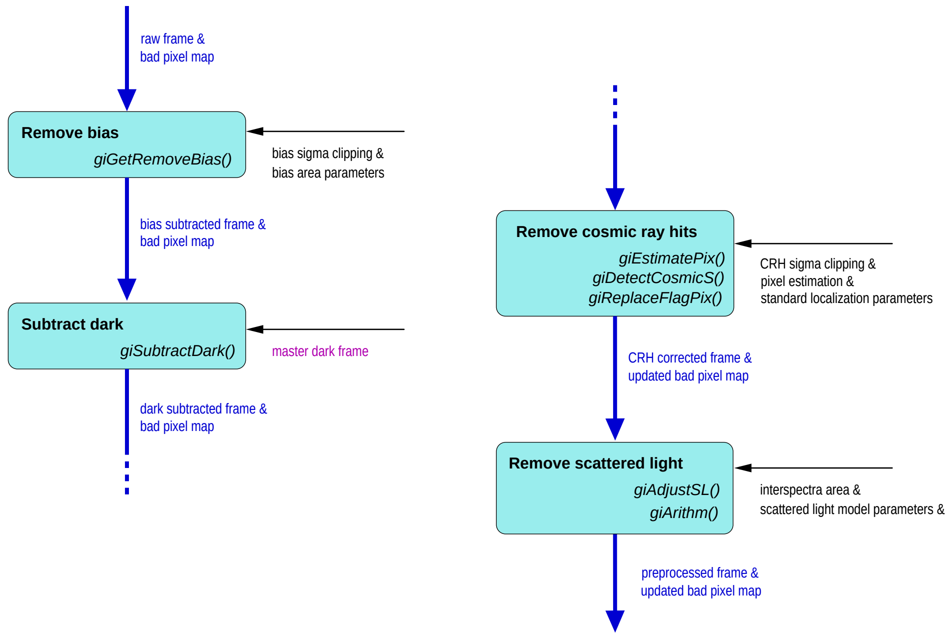
3 Architecture-Functions

3.1 Introduction

The GIRAFFE BLDRS is divided into the different modules grouped as follow: Modules could be both C-programs or Shell scripts (calling also programs) ...

3.1.1 Pre–processing modules

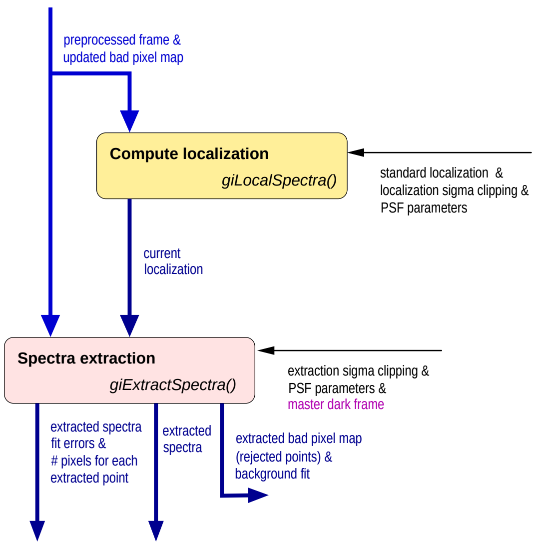
- **Remove Bias:** Calls `giGetRemoveBias()` function which computes and subtracts the bias value using CCD pre-scan and overscan areas of a raw frame and trimmed out these areas. This is the first step to get a pre–processed frame.
- **Subtract Dark:** Calls `giSubtractDark()` function which subtract the scaled (with exposure time and CCD gain) Master Dark Frame from the previously obtained pre–processed "bias–removed" frame.
- **Remove Cosmic Ray Hits:** Calls `giEstimatePix()`, `giDetectCosmicS()` and `giReplaceFlagPix()` functions which estimate (median estimation), flag and replace CRH identified pixels. If a list of frames or expositions is provided the function `giDetectCosmicM()` is used instead
- **Remove Scattered Light:** Calls `giAdjustSL()` which adjusts a model of scattered light to the inter–spectra regions. Optionally a photometric Flat–Field calibration frame can be used to correct the previously obtained pre–processed frame before the modelization.



3.1.2 Extraction modules

- **Compute localization:** Calls `giLocalSpectra()` function which determines the current centroid and width of each spectral bin for each spectrum of the pre–processed frame using an initial localization.

- **Spectra extraction**
  Calls `giExtractSpectra()` function which performs, using the current localization, the extraction of spectra of the pre–processed frame .

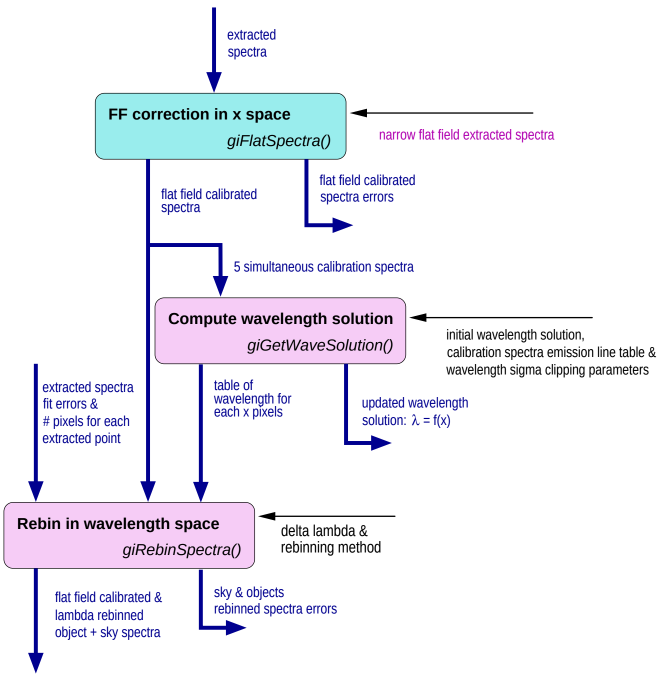


3.1.3 Flat–Field and wavelength calibration modules

- **Flat–Field correction in x space:** Calls `giFlatSpectra()` function which corrects the extracted spectra for spectrograph and detector signature using a Narrow Flat–Field calibration extracted spectra, but they remain modulated by the spectrum of the Flat–Field calibration lamp.

- **Compute wavelength solution:** Calls `giGetWaveSolution()` function which computes the current wavelength solution using a calibration exposure (SEWC) or the 5 simultaneous calibration (SIWC) extracted spectra .

- **Rebinning in wavelength space:** Calls `giRebinSpectra()` function which carry out the rebinning of extracted spectra using the current wavelength solution, so that the resulting rebinned spectra is equally spaced in λ. This is the standard most processed output of the GIRAFFE BLDRS.

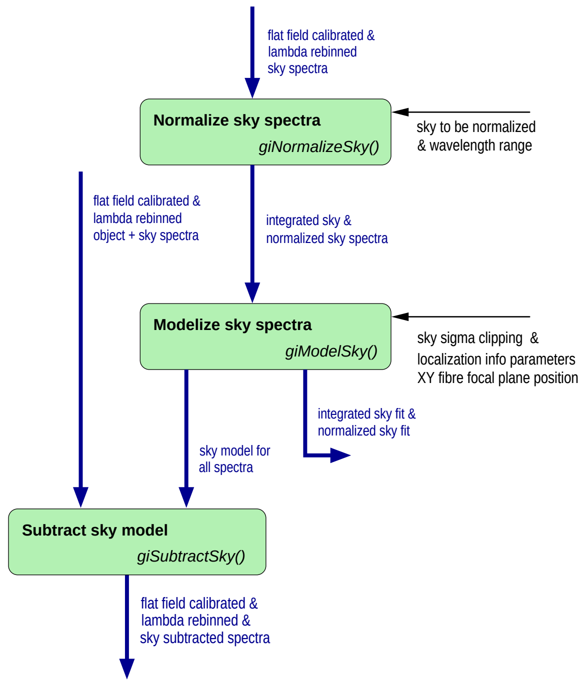


3.1.4 Sky normalization, modelization and subtraction modules

- **Normalize sky spectra:** Uses `giNormalizeSky()` function which computes normalized sky spectra, their integrated intensity is unity, for flat–fielded and λ–rebinned extracted spectra of sky fibres.

- **Modelize sky spectra:** Uses `giModelSky()` function which computes a model of the sky over the whole FOV and the whole CCD.

- **Sky subtraction:** Uses `giSubtractSky()` function which subtracts the previously modelized sky from all the flat–fielded and λ–rebinned extracted spectra.



3.2 General remarques on the presentation of Functions

The functions operate on data in physical units. It is assumed, that the image elements and henceforth the extracted spectra are in photoelectron/element. The conversion is made during the first operation in the DRS which is normaly the bias determination and subtraction. In all modules, before using any image, an appropriate check is made and the conversion of ADUs to electrons is carried out if necessary. This step is common to all functions and is not given in the Function descriptions and the necessary associated KWs, the GAIN and UNIT are not shown.

- We favor the type **float** to **real** or **double**. Only if well justified double precision is proposed (unless the KW is taken from existing dictionary).
- We keep trace of our description and comments to ESO standard KWs. The following syntax is adopted:
  - The comment (if any) from Alias file comes first prefixed by **AL:**
  - The DIC ESO comment is prefixed by **DIC** and is appended to Alias comment if both are present.

Each BLDRS function is described as follows:

Name , **Section** , **Purpose** and internal running number **nu**

Function parameters list of function control parameters initialized and passed by the calling recipe.

Input frames list of all input frames needed by the function with his name, type, format and description. The **Relevant FITS keywords** from the primary header used in the functions are listed:

- **Name** is the GIRAFFE BLDRS alias for the existing ESO HIERACH FITS keyword
- **DIC** is the name of the VLT DICB dictionary where the keyword is defined, '`DRS`' stands for GIRAFFE BLDRS own keyword and '`MISSING`' for keyword supposed to be provided but not found in any VLT dictionary
- **Type** , **Unit** and **Description** have the standard VLT dictionary meaning

Output frames list of all input frames produced by the function. Only new or updated FITS keywords are listed, all other FITS keywords for the output frames comes from the corresponding input frames. All functions parameters with their values used to produce the output frames are copied in the frame FITS header.

Other inputs/outputs list of all other inputs/outputs needed or produced by the function, this is essentially data tables. These ones are listed with the names of the columns, their type and description needed or updated by the function. The full table description with its FITS header and complete list of columns can be found in the 'Data Description' section.

General description describing the function itself

Mathematical description describing the mathematical algorithm used

Working Remarks used to internal purpose - will not appear in the final document.

3.3 Arithmetic operations on Images

Name: **giArithmetics** Section: **Preprocessing** nu: **10**
Author(s)/Date(s): **SB/20000625** Status: **To be validated by all**

Working remarks:
TBDeveloped by Gilles Simond. AB 20010123: Who is the author of the present version ? - name, date, status

3.3.1 Purpose

Compute basic pixel-to-pixel operations between images.

3.3.2 Function parameters

All input parameters are in one *string* expression

Name	Type	Default	Description
aExpression	string	undef	arithmetic expression which operates on input frames
computeKW	logical	undef	parameter indicating whether or not the statistical KW are to be updated

3.3.3 Input frames

Name: *inFrame1* **Type:** PREPIMG **Format:** xyPrep
Description: any 0-3D image
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
NX	DRS	integer	none	Size of image in X direction (lambda)
NY	DRS	integer	none	Size of image in Y direction (slit)
STARTX	CCDDCS	integer	none	DIC: First window pixel in X direction within the detector physical system.
STARTY	CCDDCS	integer	none	DIC: First window pixel in Y direction within the detector physical system.
CCDID	CCDDCS	string	none	DIC: Detector system identification Format: <CCD Id> - <ACE Id>

Name: *inFrame2* **Type:** PREPIMG **Format:** xyPrep
Description: any 0-3D image
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
NX	DRS	integer	none	Size of image in X direction (lambda)
NY	DRS	integer	none	Size of image in Y direction (slit)
STARTX	CCDDCS	integer	none	DIC: First window pixel in X direction within the detector physical system.
STARTY	CCDDCS	integer	none	DIC: First window pixel in Y direction within the detector physical system.
CCDID	CCDDCS	string	none	DIC: Detector system identification Format: <CCD Id> - <ACE Id>

Name: *badPixMap1* **Type:** BADPIX **Format:** xyPrep
Description: Current *inFrame*1 bad pixel map
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
CCDID	CCDDCS	string	none	DIC: Detector system identification Format: <CCD Id> - <ACE Id>

Name: *badPixMap2* **Type:** `BADPIX` **Format:** `xyPrep`
Description: Current *inFrame*2 bad pixel map
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`CCDID`	CCDDCS	`string`	none	DIC: Detector system identification Format: <CCD Id> - <ACE Id>

3.3.4 Other inputs

none

3.3.5 Output frames

Name: *outFrame* **Type:** `PREPIMG` **Format:** `xyPrep`
Description: any 0-3D image
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`GIRFTYPE`	DRS	`string`	none	type of frame data
`NX`	DRS	`integer`	none	Size of image in X direction (lambda)
`NY`	DRS	`integer`	none	Size of image in Y direction (slit)
`STARTX`	CCDDCS	`integer`	none	DIC: First window pixel in X direction within the detector physical system.
`STARTY`	CCDDCS	`integer`	none	DIC: First window pixel in Y direction within the detector physical system.
`DATAMIN`	PR_FITS	`double`	none *e-	DIC: Minimum pixel value across the frame excludes special values, i.e. BLANK). AL: Minimu signal of valid pixels
`DATAMAX`	PR_FITS	`double`	none *e-	DIC: Maximum pixel value across the frame (excludes special values, i.e BLANK). AL: Maximum signal of valid pixels
`DATAMEAN`	PRO	`double`	none *e-	DIC: Mean of the pixel values in the frame. All pixels are taken into account. AL: Average signal of valid pixels
`DATAMEDI`	PRO	`double`	none *e-	DIC: Exact median of the pixel values in the frame. All pixels are taken into account. AL: Median signal of valid pixels
`DATASIG`	PRO	`double`	none *e-	DIC: Population standard deviation (sigma) of the pixel values in the frame. All pixels are taken into account. AL: Sigma of the signal of valid pixels

Name: *badPixMap* **Type:** `BADPIX` **Format:** `xyPrep`
Description: Updated current *outFrame* bad pixel map
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`CCDID`	CCDDCS	`string`	none	DIC: Detector system identification Format: <CCD Id> - <ACE Id>

3.3.6 Other outputs

none

3.3.7 General description

Basic arithmetical and logical operations are computed over 0-3D images and/or masks. Mixed dimensions are accepted under condition of the size match. In such a case the low-dimensioned operand is used repeatedly. When needed, the bad pixels mask of the resulting image is taken as the union of both input bad pixels masks.

3.3.8 Mathematical description

This function computes operations (unary and binary) over 0-3D images. A complex operation is first parsed and separated into interweaving basic operations. Operators precedence is given below in decreasing order of priority:

- exponentiation
- logarithm
- multiplication and division (same priority level)
- addition and subtraction (same prioriry level)

Parentheses '(' and ')' can be used to enforce operator priority

The following binary arithmetic operations (pixel-to-pixel operations): addition, subtraction, multiplication, and division are available on two frames;

$$outFrame = inFrame1 + inFrame2$$

$$outFrame = inFrame1 - inFrame2$$

$$outFrame = inFrame1/inFrame2$$

$$outFrame = inFrame1 * inFrame2$$

Since mixed dimensions can be used (mixing 3D datacubes, frames, vectors and numeric values under condition of size match), these binary operations cover unary operations with a numeric constant. Unary operations exponentiation and logarithm (pixel-to-pixel) are also available

$$outFrame = (inFrame)^p$$

$$outFrame = \log(inFrame)$$

The bad pixels mask of the resulting image is taken as the union of both input bad pixels masks

$$badPixMap = badPixMap1 \cup badPixMap2 \cup errPixMap$$

Where $errPixMap$ is the mask of pixels where the current operator returns an error. Possible errors are division by zero, overflow exception, underflow exception ...
In the case of unary operations the resulting bad pixels mask is:

$$badPixMap = badPixMap1 \cup errPixMap$$

Depending on the logical parameter <`computeKW`> , statistical keywords are computed over the resulting frame:

- `DATAMIN` : the minimum intensity value
- `DATAMAX` : the maximum intensity value
- `DATAMEAN` : the mean intensity value
- `DATAMEDI` : the median intensity value
- `DATASIG` : the standard deviation of intensity values

The parameter <`computeKW`> will allow to switch off the computation of the statistical Keywords in cases where it is not required (intermediate operations). When the computation is done (`computeKW` = `TRUE`), and if the statistical keyword exists for both input data, it is possible for some of them to only update the resulting one using input keywords and not compute it over the whole frame:

- for linear combinations of frames, resulting `DATAMEAN`, `DATAMEDI`, `DATASIG` are combined in the same way, from input Keywords
- for unary operations (+, −, *, /) with a constant, resulting `DATAMEAN`, `DATAMEDI`, `DATASIG`, `DATAMIN` and `DATAMAX` are updated using this relation (`DATAMIN` and `DATAMAX` are permutated if multiplied by a negative constant).

3.4 Get bias value over bias test area

Name: **giGetRemoveBias** Section: **Preprocessing** nu: **20**
Author(s)/Date(s): **SB/20000607,AB** Status: **To be validated by all**

Working remarks:
TBV Corrected after review by SB blecha Thu 09/14/00 14:00:39. Corrected blecha Tue 09/12/00 17:40:45 after Internal rewiev Wed 09/06/00.

3.4.1 Purpose

The bias and the read-out noise are obtained as the average value and sigma over the overscan areas. The bias value is subtracted and the overscan lanes are trimmed out.

3.4.2 Function parameters

Control parameters for bias sigma clipping.

Name	Type	Default	Description
bClipSigma	float	2.5	multiple of σ (σ-clipping)
bClipNiter	integer	2	number of iterations (σ-clipping)
bClipMfrac	float	0.9	min fraction of points accepted/total (σ-clipping)
bMethod	string	ZMASTER + UNIFORM	Method used, one of: Uniform,Plane,Master,Zmaster

3.4.3 Input frames

Name: *rawFrame* **Type:** RAWIMG **Format:** xyRaw
Description: raw frame with bias prescans and overscans included
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
NX	DRS	integer	none	Size of image in X direction (lambda)
NY	DRS	integer	none	Size of image in Y direction (slit)
STARTX	CCDDCS	integer	none	DIC: First window pixel in X direction within the detector physical system.
STARTY	CCDDCS	integer	none	DIC: First window pixel in Y direction within the detector physical system.
PRESCX	CCDDCS	integer	none	DIC: Number of pixels in X read before real data arrives.
PRESCY	CCDDCS	integer	none	DIC: Number of pixels in Y read before real data arrives.
OVSCX	CCDDCS	integer	none	DIC: Number of pixels in X read as overscan.
OVSCY	CCDDCS	integer	none	DIC: Number of pixels in Y read as overscan.
RON	DRS	float	e-	Expected CCD readout noise
CCDID	CCDDCS	string	none	DIC: Detector system identification Format: <CCD Id> - <ACE Id>
CONAD	CCDDCS	double	e-/ADU	DIC: The conversion factor which translates ADU to photonic electrons.
EXPTIME	UVES_ICS	double	sec	Total exposure time for a single exposure or for the individual exposure within a multiple frame

Name: *biasMast* **Type:** `BIASIMG` **Format:** `xyPrep`
Description: master bias
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
NX	DRS	integer	none	Size of image in X direction (lambda)
NY	DRS	integer	none	Size of image in Y direction (slit)
STARTX	CCDDCS	integer	none	DIC: First window pixel in X direction within the detector physical system.
STARTY	CCDDCS	integer	none	DIC: First window pixel in Y direction within the detector physical system.
PRESCX	CCDDCS	integer	none	DIC: Number of pixels in X read before real data arrives.
PRESCY	CCDDCS	integer	none	DIC: Number of pixels in Y read before real data arrives.
OVSCX	CCDDCS	integer	none	DIC: Number of pixels in X read as overscan.
OVSCY	CCDDCS	integer	none	DIC: Number of pixels in Y read as overscan.
RON	DRS	float	e-	Expected CCD readout noise
CCDID	CCDDCS	string	none	DIC: Detector system identification Format: <CCD Id> - <ACE Id>
DATAMEAN	PRO	double	none *e-	DIC: Mean of the pixel values in the frame. All pixels are taken into account. AL: Average signal of valid pixels
DATAMOD	DRS	float	e-	Mode of the signal of valid pixels
EXPTIME	UVES_ICS	double	sec	Total exposure time for a single exposure or for the individual exposure within a multiple frame
BIASVALUE	DRS	float	none	Computed average bias level in pre/over-scan

Name: *badPixMast* **Type:** `BADPIX` **Format:** `xyPrep`
Description: Optional bias bad pixel map with active flag 'bBadPixFlag'
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
CCDID	CCDDCS	string	none	DIC: Detector system identification Format: <CCD Id> - <ACE Id>

3.4.4 Other inputs

Name	Type	Description
biasLimitsMast	girBiasBoxTable	Definition of rectangular areas on CCD overscan and prescan used for bias computation
Relevant FITS keywords:	GIRTTYPE CCDID	
Relevant table columns:		
BIAS_X1	integer	X-position of the upper-left corner
BIAS_Y1	integer	Y-position of the upper-left corner
BIAS_X2	integer	X-position of the lower-right corner
BIAS_Y2	integer	Y-position of the lower-right corner

3.4.5 Output frames

Name: *bSubImage* **Type:** `BRMIMG` **Format:** `xyPrep`
Description: bias subtracted image converted in electrons with overscan lane trimmed out
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`GIRFTYPE`	DRS	`string`	none	type of frame data
`BCLIPSIGMA`	DRS	`float`	none	multiple of sigma (sigma-clipping)
`BCLIPNITER`	DRS	`integer`	none	number of iterations (sigma-clipping)
`BCLIPMFRAC`	DRS	`float`	none	min fraction of points accepted/total (sigma-clipping)
`BIASVALUE`	DRS	`float`	none	Computed average bias level in pre/over-scan
`BIASSIGMA`	DRS	`float`	none	Computed bias sigma in pre/over-scan
`BIASPLANE`	DRS	`float`	none	3 coefficients of the x,y plane describing the fitted bias=B0+Bx*X+By*Y; X,Y origine is at the first active pixel.

3.4.6 Other outputs

none

3.4.7 General description

This is the first processing of any Giraffe raw image. It is the only function operating on images larger than the CCD active area (pre/overscans present) and using ADU units.

The Bias Areas in the prescan and overscan lanes in both, x and y, directions are defined through the `biasLimitsMast` table. A 2D plane is fitted on the data in these Bias Areas and the linear coefficients describing the plane are saved as quality indicators in *bSubImage*.`BIASPLANE`.

The control parameter <`bMethod`> is used to differentiate various level of the processing: the single valued average subtraction, the plane subtraction and the *biasMast* subtraction without or with zero adjustment.

In any case the pre/overscans are removed prior the bias subtraction (the *biasMast* not having necessarily the same size of pre/overscans as *rawFrame*) and the *rawFrame* is converted from ADU to e^- using the KW *rawFrame*.`CONAD`.

When *badPixMast* is supplied, the bit 'bBadPixFlag' is used to design flagged pixels.

3.4.8 Mathematical description

At least 2 areas (line prescan and overscan) are necessary, but it is desirable to use also the image (column) pre/overscans.

A bi-linear form $Bmodel = B_0 + B_x.X + B_y.Y$ is fitted with sigma-clipping. A coordinate system with [0,0] on the first active pixel [prescanX+1,prescanY+1] is used. In each iteration *iter* controlled by input parameters <`bClipSigma`> , <`bClipNiter`> , and <`bClipMfrac`> the plane is fitted and the `biasSigma` over valid points is computed as follows

$$biasSigma_{iter} = \left(1/bNpAccept \sum_{i=1}^{bNpAccept} (rawFrame(x_{j_i}, y_{j_i}) - Bmodel(x_{j_i}, y_{j_i}))^2 \right)^{0.5}$$

Only valid pixels $[x_{j_i}, y_{j_i}]$ satisfying condition

$$\mid rawFrame(x_{j_i}, y_{j_i}) - Bmodel(x_{j_i}, y_{j_i}) \mid < \texttt{bClipSigma} \times biasSigma_{iter-1}$$

are included. The sigma-clipping is repeated till no new points are rejected or the <`bClipNiter`> iteration is reached or when too many points are rejected. ($\frac{bNpAccept}{bNpTot} >$ `bClipMfrac`). The average bias of the *rawFrame* (used and saved further as the KW *bSubImage.BIASVALUE*) is computed as $BIASVALUE = Bmodel(X_{center}, Y_{center})$ where X_{center}, Y_{center} are coordinates of the center of the CCD active image as given in 1. Note that the formula 1 refers to KWs of the *rawFrame*.

$$X_{center} = (NX - (PRESCX + OVERSCX + 1)) \times 0.5$$
$$Y_{center} = (NY - (PRESCY + OVERSCY + 1)) \times 0.5 \quad (1)$$

Pre/overscans lanes are removed from $rawFrame$.
The control parameter <`bMethod`> is used as follows:

- <`bMethod`> ='UNIFORM'

  The average bias $bSubImage.BIASVALUE$ is subtracted.

  $bSubImage(x, y) = rawFrame(x, y) - bSubImage.BIASVALUE$

- <`bMethod`> ='PLANE'

  The fitted plane rather than a single value is subtracted pixel-by-pixel.

  $bSubImage(x, y) = rawFrame(x, y) - Bmodel(x, y)$

- <`bMethod`> ='MASTER'

  The biasMast image is subtracted without modification. The actual processing depends on the presence/absence of the $badPixMast$.

  - $badPixMast$ is absent
    $bSubImage(x, y) = rawFrame(x, y) - biasMast(x, y)$
  - $badPixMast$ is given
    the bit 'bBadPixFlag' is extracted from $badPixMast$ to form logical mask $bBadPixMask$ which is applied as follows:

$$\begin{aligned} bSubImage(x, y) &= (rawFrame(x, y) - biasMast(x, y)) * bBadPixMask(x, y) \cup \\ &\quad (rawFrame(x, y) - Bmodel(x, y)) * (1 - bBadPixMask(x, y)) \end{aligned}$$

    Note that in this and 'ZMASTER' method, the method to obtain the bias model may be specified, the default being the method 'UNIFORM'.

- <`bMethod`> ='ZMASTER'

  A possible drift of average bias is compensated using the pre/overscans reference value. We consider the $biasMast$ as an additive correction and therefore the adjustment is made by the shift of the $biasMast$ and not by the scaling (multiplication).

  Prior to any processing, the biasMast is corrected for the drift:

  $biasMast = biasMast(x, y) - biasMast.\texttt{BIASVALUE} + bSubImage.\texttt{BIASVALUE}$

  As for method 'MASTER', the subsequent processing depends on the presence/absence of the $badPixMast$

  - $badPixMast$ is absent
    $bSubImage(x, y) = rawFrame(x, y) - biasMast(x, y)$
  - $badPixMast$ is given
    extract $bBadPixMask$

$$\begin{aligned} bSubImage(x, y) &= (rawFrame(x, y) - biasMast(x, y)) * bBadPixMask(x, y) \cup \\ &\quad (rawFrame(x, y) - Bmodel(x, y)) * (1 - bBadPixMask(x, y)) \end{aligned}$$

The *rawFrame* is converted from ADU to e^- and KW TBD is set to indicate units (the *biasMast*, if used is already in e^-).
$bSubImage(x, y) = bSubImage(x, y) \times rawFrame.\texttt{CONAD}$

COMMENTS

If the *badPixMast* is not supplied, the bias subtraction is done on all pixels.

The subtraction of the *biasMast* is made on the already trimmed area (the *biasMast* not having necessarily same size of pre/overscans as *rawFrame*).

Note that KWs (PRE/OVR)SC(X/Y) are left unchanged on the output image though the prescan/overscan lanes are trimmed out.

blecha Thu 09/14/00 12:03:58

3.5 Subtract Dark

Name: **giSubtractDark** Section: **Preprocessing** nu: **30**
Author(s)/Date(s): **SB/20000616,LC,AB** Status: **To be validated by GS**

> ***Working remarks:***
> *TBV Corrected blecha Tue 09/12/00 17:40:45 after Internal rewiev Wed 09/06/00*
> *Added corresponding running python "pseudo"-code (gsimond 09/15/00)*

3.5.1 Purpose

The scaled (with exposure time) current standard mean dark is subtracted from the current image.

3.5.2 Function parameters

Name	Type	Default	Description
`darkThreshold`	`float`	`1`	dark threshold value in e/pixel
`darkMethod`	`string`	`UNIFORM`	one of 'UNIFORM' 'MASTER'

3.5.3 Input frames

Name: *bSubImage* **Type:** `BRMIMG` **Format:** `xyPrep`
Description: any bias subtracted GIRAFFE frame
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`GIRFTYPE`	DRS	`string`	none	type of frame data
`NX`	DRS	`integer`	none	Size of image in X direction (lambda)
`NY`	DRS	`integer`	none	Size of image in Y direction (slit)
`STARTX`	CCDDCS	`integer`	none	DIC: First window pixel in X direction within the detector physical system.
`STARTY`	CCDDCS	`integer`	none	DIC: First window pixel in Y direction within the detector physical system.
`EXPTIME`	UVES_ICS	`double`	sec	Total exposure time for a single exposure or for the individual exposure within a multiple frame
`CCDID`	CCDDCS	`string`	none	DIC: Detector system identification Format: <CCD Id> - <ACE Id>

Name: *darkMast* **Type:** `DARKIMG` **Format:** `xyPrep`
Description: dark frame in electron/second units (Master Dark Frame)
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`GIRFTYPE`	DRS	`string`	none	type of frame data
`NX`	DRS	`integer`	none	Size of image in X direction (lambda)
`NY`	DRS	`integer`	none	Size of image in Y direction (slit)
`STARTX`	CCDDCS	`integer`	none	DIC: First window pixel in X direction within the detector physical system.
`STARTY`	CCDDCS	`integer`	none	DIC: First window pixel in Y direction within the detector physical system.
`DATAMOD`	DRS	`float`	e-	Mode of the signal of valid pixels
`DATAMAX`	PR_FITS	`double`	none *e-	DIC: Maximum pixel value across the frame (excludes special values, i.e BLANK). AL: Maximum signal of valid pixels
`CCDID`	CCDDCS	`string`	none	DIC: Detector system identification Format: <CCD Id> - <ACE Id>
`EXPTIME`	UVES_ICS	`double`	sec	Total exposure time for a single exposure or for the individual exposure within a multiple frame

Name: *badPixMast* **Type:** `BADPIX` **Format:** `xyPrep`
Description: Optional dark bad pixel map with active flag 'dBadPixFlag'
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`CCDID`	CCDDCS	`string`	none	DIC: Detector system identification Format: <CCD Id> - <ACE Id>

3.5.4 Other inputs

none

3.5.5 Output frames

bad pixel map = badPixMaster unchanged

Name: *dbSubImage* **Type:** `BDRMIMG` **Format:** `xyPrep`
Description: dark and bias subtracted frame
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`GIRFTYPE`	DRS	`string`	none	type of frame data
`DATAMOD`	DRS	`float`	e-	Mode of the signal of valid pixels
`DARKEXPECT`	DRS	`float`	e-	The value of the expected dark current as computed by subtractDark from masterDark and exposure time
`DARKVALUE`	DRS	`float`	e-	The value of the actualy subtracted dark current. It is set to 0 if no subtraction is made (treshold not reached).
`DARKMETHOD`	DRS	`string`	none	One of 'UNIFORM' (we subtract a single number) and 'MASTER' (we subtract a scaled masterDark).
`DARKTHRESH`	DRS	`float`	e-	If the average expected dark is below this value, no dark subtraction is made.

3.5.6 Other outputs

none

3.5.7 General description

The bias subtracted frame is corrected for dark current. Based on the maximum value of expected dark current and the threshold given as input parameter <`darkThreshold`> , the decision whether the dark current will be subtracted is taken. The <`darkMethod`> indicates whether a constant value (this method does not introduce additional random noise) or a masterDark should be used. The first 'UNIFORM' method is intended typically for short exposures (to prevent any average bias due to the unsubtracted dark) while 'MASTER' will be appropriate for long exposures. The *badPixMast* is associated to *darkMast* and flag 'dBadPixFlag' is used to identify pixels where the dark current is far above the average. The *badPixMast* is used together with the <`darkThreshold`> input parameter to enable the local use of *darkMast*.

3.5.8 Mathematical description

The dark current subtraction proceeds as follows:

1. Get expected values

   The maximum, `darkMaxExpected`, and the mode, `darkModExpected`, of the expected dark are computed using the KWs *darkMast*.`DATAMAX` and *darkMast*.`DATAMOD` and the exposure times *bSubImage*.`EXPTIME`, *darkMast*.`EXPTIME` of the processed frame and of the master dark:

   $darkMaxExpected = darkMast.\texttt{DATAMAX}\frac{bSubImage.\texttt{EXPTIME}}{darkMast.\texttt{EXPTIME}}$

   $darkModExpected = darkMast.\texttt{DATAMOD}\frac{bSubImage.\texttt{EXPTIME}}{darkMast.\texttt{EXPTIME}}$

2. Subtract dark - method 'UNIFORM'

   We subtract the constant, uniform dark equal to `darkModExpected`. If the `darkThreshold` $> darkMaxExpected$ no dark subtraction is made.

3. Subtract dark - method 'MASTER'

   We subtract *darkMast* scaled by the exposure time of the processed frame (*bSubImage*.`EXPTIME`) on selected pixels and `darkModExpected` constant values on unselected pixels. The pixel selection is governed by the presence/absence of the *badPixMast* and the values of <`darkThreshold`> . A selection mask is either extracted from *badPixMast* or build using the <`darkThreshold`> value.

   Two preliminary steps are necessary:

   - Scaling of *darkMast* according to the exposure times:
     $scaledDark(x, y) = darkMast(x, y)\frac{bSubImage.\texttt{EXPTIME}}{darkMast.\texttt{EXPTIME}}$
   - Building of the internal selection mask `dMastPixMask`
     (a) *badPixMast* is absent and <`darkThreshold`> =0
     All points are selected for the pixel-pixel subtraction.
     $dMastPixMask = 1$
     Note that the implantation will not actually build (unnecessarily) the mask in this case since the general formula 2 will simplify.
     (b) *badPixMast* is absent and <`darkThreshold`> >0
     Pixs with expected dark above the <`darkThreshold`> are selected.
     $dMastPixMask(x, y) = scaledDark(x, y) > \texttt{darkThreshold}$
     (c) *badPixMast* is present
     The bit 'dBadPixFlag' is extracted from *badPixMast* to form logical mask and inverted to form `dMastPixMask` so as the bits flagged in the *badPixMast* are set to 1 in the `dMastPixMask`.

   Eventually, the $scaledDark(x, y)$ image and darkModExpect constant are merged according the formula

$$\begin{aligned} scaledDark(x, y) &= scaledDark(x, y) * dMastPixMask(x, y) \cup \\ &\quad darkModExpected * (1 - dMastPixMask(x, y)) \end{aligned} \tag{2}$$

   and subtracted from input frame

   $dbSubImage = bSubImage(x, y) - scaledDark(x, y)$

4. update *dbSubImage* keywords:

   The *dbSubImage*.`DATAMOD` is set to the `darkModExpected`, *dbSubImage*.`DARKMETHOD` is set to <`darkMethod`> with 'BADMASK' appended if *badPixMast* was used and all dark-associated KW (*dbSubImage*.`DARKMETHOD`, *dbSubImage*.`DARKTHRESH`) are written to the output image *dbSubImage*.

Note that in both methods and all cases, all pixels of *bSubImage* are affected and badPixMask remains unchanged and therefore there is no output associate to the *badPixMast* input.

3.6 Estimate pixels values in an image

Name: **giEstimatePix** Section: **Preprocessing** nu: **40**
Author(s)/Date(s): **FR/20000608** Status: **validated**

Working remarks:
Ready for pseudo-code (26 Sep 2000)

3.6.1 Purpose

Apply a median estimation to the input frame over the closest neighboring good pixels.

3.6.2 Function parameters

Estimation control parameters

Name	Type	Default	Description
`badPixMfrac`	`float`	`0.3`	min fraction of neighboring bad pixels accepted/total
`maxRadius`	`float`	`2.`	Maximum radius of the neighboring area (pixels)
`metric`	`integer`	`00`	XY metrics indicating how the averaging area spreads 11-X,Y; 10-X only 01-Y only 00-LAMP KW driven

3.6.3 Input frames

This function could take any `'xyPrep'` or `'nxExt'` formatted frame as input

Name: *dbSubImage* **Type:** `PREPIMG` **Format:** `xyPrep`
Description: any preprocessed GIRAFFE frame
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`GIRFTYPE`	DRS	`string`	none	type of frame data
`NX`	DRS	`integer`	none	Size of image in X direction (lambda)
`NY`	DRS	`integer`	none	Size of image in Y direction (slit)
`STARTX`	CCDDCS	`integer`	none	DIC: First window pixel in X direction within the detector physical system.
`STARTY`	CCDDCS	`integer`	none	DIC: First window pixel in Y direction within the detector physical system.
`SLIT`	FGIR_ICS	`integer`	none	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies
`GRATING`	FGIR_ICS	`string`	none	DIC: ESO name for grating unit.
`WLEN0`	FGIR_ICS	`double`	nm	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength
`CCDID`	CCDDCS	`string`	none	DIC: Detector system identification Format: <CCD Id> - <ACE Id>
`LAMP`	DRS	`string`	none	Name of the W Calibration source

Name: *locYMast* **Type:** `LOCY` **Format:** `nxExt`
Description: Master localization giving the spectrum center along y at each x for the current setup
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
EXTNX	DRS	integer	none	Number of pixel per spectrum
EXTNS	DRS	integer	none	Number of extracted spectra on the rebinned S2dFrame
SLIT	FGIR_ICS	integer	none	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies
GRATING	FGIR_ICS	string	none	DIC: ESO name for grating unit.
WLEN0	FGIR_ICS	double	nm	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength

Name: *locWyMast* **Type:** `LOCWY` **Format:** `nxExt`
Description: Master width of the NFF spectra (measured along the y axis) for the current setup
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
EXTNX	DRS	integer	none	Number of pixel per spectrum
EXTNS	DRS	integer	none	Number of extracted spectra on the rebinned S2dFrame
SLIT	FGIR_ICS	integer	none	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies
GRATING	FGIR_ICS	string	none	DIC: ESO name for grating unit.
WLEN0	FGIR_ICS	double	nm	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength

Name: *badPixMast* **Type:** `BADPIX` **Format:** `xyPrep`
Description: Current *dbSubImage* bad pixel map
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
CCDID	CCDDCS	string	none	DIC: Detector system identification Format: <CCD Id> - <ACE Id>

3.6.4 Other inputs

none

3.6.5 Output frames

Name: *estimateImage* **Type:** `ESTIMG` **Format:** `xyPrep`
Description: estimated image
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
BADPIXMFRAC	DRS	float	none	min fraction of neighboring bad pixels accepted/total
MAXRADIUS	DRS	float	pixel	maximum radius of the neighboring area
METRIC	DRS	integer	none	XY metrics indicating how the averaging area spreads 11-X,Y; 10-X only 01-Y only 00-LAMP KW driven

Name: *estimateSigmaImage* **Type:** `ESTSIGM` **Format:** `xyPrep`
Description: estimate of the standard error of each pixel
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
BADPIXMFRAC	DRS	float	none	min fraction of neighboring bad pixels accepted/total
MAXRADIUS	DRS	float	pixel	maximum radius of the neighboring area
METRIC	DRS	integer	none	XY metrics indicating how the averaging area spreads 11-X,Y; 10-X only 01-Y only 00-LAMP KW driven

Name: *estimateNpixels* **Type:** `ESTNPIX` **Format:** `xyPrep`
Description: number of points used in the average
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
BADPIXMFRAC	DRS	float	none	min fraction of neighboring bad pixels accepted/total
MAXRADIUS	DRS	float	pixel	maximum radius of the neighboring area
METRIC	DRS	integer	none	XY metrics indicating how the averaging area spreads 11-X,Y; 10-X only 01-Y only 00-LAMP KW driven

Name: *estBadPixMap* **Type:** `BADPIX` **Format:** `xyPrep`
Description: *estimateImage* bad pixel map (expected same as input bad pixel map)
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
CCDID	CCDDCS	string	none	DIC: Detector system identification Format: <CCD Id> - <ACE Id>
BADPIXMFRAC	DRS	float	none	min fraction of neighboring bad pixels accepted/total
MAXRADIUS	DRS	float	pixel	maximum radius of the neighboring area
METRIC	DRS	integer	none	XY metrics indicating how the averaging area spreads 11-X,Y; 10-X only 01-Y only 00-LAMP KW driven

3.6.6 Other outputs

none

3.6.7 General description

In order to get rid of bad pixels, each pixel of the whole image can be estimated using its neighbors. This operation is carried out over the spectra and inter-spectra regions.

The averaging area is defined by <`metric`> . The user can choose a symmetric bidimensional averaging area (`metric` = 11), used for example with NFF spectra, or a monodimensional area (`metric` = 01, along x-axis; `metric` = 10, along y-axis). The choice can be automatically set (`metric` = 00) according to the type of observation (*dbSubImage*.`LAMP`).

The size of the averaging area is set to include enough good pixels ($\frac{NbadPix}{NpointsTotal}$ < `badPixMfrac`). The area for estimating spectrum pixel remains embraced by the localization mask, without encroaching the inter-spectra region, and vice versa for inter-spectra pixel. The localization mask is defined by $locYMast$ and $locWyMast$ frames which depend on the values of $dbSubImage$.`SLIT`, $dbSubImage$.`GRATING` and $dbSubImage$.`WLENO` keywords of the input frame.

Pixels of *estimateImage* are flagged as bad if <`maxRadius`> and <`badPixMfrac`> have been reached.

3.6.8 Mathematical description

Each pixel of *estimateImage* (respectively *estimateSigmaImage*) is the median (respectively the standard error) of the pixel values (excluding bad pixels) of the input *dbSubImage* computed over the averaging area.

$$estimateImage(x, y) = \overline{dbSubImage(x, y)}_{\text{averaging area}}$$

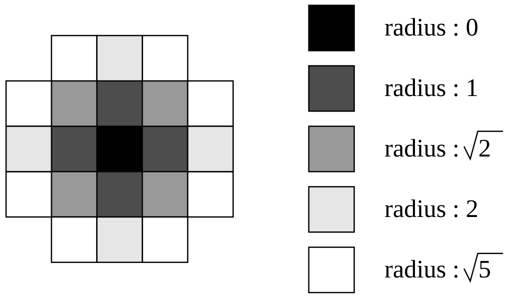


Figure 1: Increase of the averaging area

The default averaging area has a 1-pixel-radius, so that the 2 (or 4) closest pixels are used in the estimate depending whether the estimate is done in one (or two) dimensions. If the proportion of bad pixels in this averaging area is greater than <`badPixMfrac`> , the radius of the averaging area is incremented in a way to be defined (1-pixel increment, or following the series [1,$\sqrt{2}$,2,$\sqrt{5}$,2$\sqrt{2}$,3,$\sqrt{10}$ ...][1], see Fig. 1 for illustration) and the process is iterated until <`maxRadius`> is reached or the ratio of bad pixels over total pixels is low enough ($\frac{NbadPix}{NpointsTotal}$ <<`badPixMfrac`>).

$estimateNpixels(x, y)$ contains the corresponding radius of the averaging area used to compute the median and the standard error.

The purpose of the function is to estimate spectra-pixels using only pixels in the localization mask, and estimate interspectra-pixels using only pixels where no objet light is collected. The median is then computed on the pixels embraced by the limits of the localization mask. Those which overlap the limit contribute proportionnaly to the embraced fraction of pixel. It is equivalent to compute a weighted mean, where the weight $\alpha(x, y)$ is equal to unity when the corresponding pixel (x, y) lies entirely in the considered area (spectrum, or inter-spectra) and otherwise:

- if $(y + \frac{1}{2} > LocYMast(x, n) + LocWyMast(x, n))$ then $\alpha(x, y) = (LocYMast(x, n) + LocWyMast(x, n)) - (y - \frac{1}{2})$
- if $(y - \frac{1}{2} < LocYMast(x, n) - LocWyMast(x, n))$ then $\alpha(x, y) = (y - \frac{1}{2}) - (LocYMast(x, n) - LocWyMast(x, n))$

Portion of the mask limit intersecting a pixel is regarded as being linear, parallel to dispersion axis.

The output bad pixel mask is initially identical to the input one. The pixels that cannot be estimated (not enough good pixels in the maximum averaging area) are flagged in the output bad pixel mask with the bit 'estBadPixelFlag'.

3.6.9 Pseudo code

[1]This series is composed of the square root of integers that are the sum of two squares.

3.7 Detect Cosmic ray hits in a single image

Name: **giDetectCosmicS** Section: **Preprocessing** nu: **50**
Author(s)/Date(s): **VC/20000718** Status: **To be validated by all**

Working remarks:
Submitted to Internal rewiev Wed 09/06/00. Concerning the quality assesement, it is expected that there will be more CRH detection per unit exposure time on wavelength calibrations that on FFs (higher shot noise generated by the signal). This should be calibrated using the darks where the detection is the most efficient (lowest noise possible). Last uppdate blecha Mon 09/25/00 17:58:44

3.7.1 Purpose

The detection of CRH is made after pixels values estimation in a frame and is used to upgrade the bad pixel map.

3.7.2 Function parameters

Name	Type	Default	Description
crhsSigMul	float	4.5	multiple of σ used as threshold for local excess determination
crhsAlertVal	float	10.0	number of cosmic ray hits used as threshold for an alert
crhsCrhImg	logical	FALSE	if set we compute the image of CRH photons (crhsCount)
crhBadPixFlag	integer	TBD	flag used to indicate CRH hits
badPixFlags	integer	eBadPixFlag	flags to be used with associated *estimateImage*

3.7.3 Input frames

CRH detection for a single frame after pixels values estimation for this frame

Name: *dbSubImage* **Type:** BDRMIMG **Format:** xyPrep
Description: bias-and-dark-subtracted frame
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
NX	DRS	integer	none	Size of image in X direction (lambda)
NY	DRS	integer	none	Size of image in Y direction (slit)
STARTX	CCDDCS	integer	none	DIC: First window pixel in X direction within the detector physical system.
STARTY	CCDDCS	integer	none	DIC: First window pixel in Y direction within the detector physical system.
EXPTIME	UVES_ICS	double	sec	Total exposure time for a single exposure or for the individual exposure within a multiple frame

Name: *estimateImage* **Type:** ESTIMG **Format:** xyPrep
Description: computed estimated image of *dbSubImage* frame
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
NX	DRS	integer	none	Size of image in X direction (lambda)
NY	DRS	integer	none	Size of image in Y direction (slit)
STARTX	CCDDCS	integer	none	DIC: First window pixel in X direction within the detector physical system.
STARTY	CCDDCS	integer	none	DIC: First window pixel in Y direction within the detector physical system.

Name: *estimateSigmaImage* **Type:** `ESTSIGM` **Format:** `xyPrep`
Description: computed estimated sigma image of *dbSubImage* frame
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`GIRFTYPE`	DRS	`string`	none	type of frame data
`NX`	DRS	`integer`	none	Size of image in X direction (lambda)
`NY`	DRS	`integer`	none	Size of image in Y direction (slit)
`STARTX`	CCDDCS	`integer`	none	DIC: First window pixel in X direction within the detector physical system.
`STARTY`	CCDDCS	`integer`	none	DIC: First window pixel in Y direction within the detector physical system.

Name: *estBadPixMap* **Type:** `BADPIX` **Format:** `xyPrep`
Description: bad pixel map associated to *dbSubImage* and *estimateImage*
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`CCDID`	CCDDCS	`string`	none	DIC: Detector system identification Format: <CCD Id> - <ACE Id>

3.7.4 Other inputs

none

3.7.5 Output frames

Name: *crhsCount* **Type:** `CRHIMG` **Format:** `xyPrep`
Description: optional frame with the value of photons attributed to CRH for each pixel
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`GIRFTYPE`	DRS	`string`	none	type of frame data
`CSIGMAMULT`	DRS	`float`	none	multiple of sigma

Name: *crhBadPixMap* **Type:** `BADPIX` **Format:** `xyPrep`
Description: updated current bad pixel map associated to *dSubImage*
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`GIRFTYPE`	DRS	`string`	none	type of frame data
`CCDID`	CCDDCS	`string`	none	DIC: Detector system identification Format: <CCD Id> - <ACE Id>
`CSIGMAMULT`	DRS	`float`	none	multiple of sigma

3.7.6 Other outputs

none

3.7.7 General description

Detection of CRHs is made on the input image using pixel values, the local estimate supplied via *estimateImage* and the bad pixels mask extracted from the associated *estBadPixMap* (pixels flagged by the <`badPixFlags`>). The standard use will be the CRHs removal from a single image when multiple images are not available. It will be carried out in 3 steps:

1. Generation of the estimated image
2. CRHs detection (this function)
3. CRHs replacement

3.7.8 Mathematical description

The CRHs will be localized using the data of the current image *dbSubImage* and based on local estimates of pixel *estimateImage* value and standard error *estimateSigmaImage*. The *estBadPixMap* is used together with <`badPixFlags`> to extract the *badPixMask* which governs the identification of pixels excluded from the processing.

$badPixMask(x, y) = estBadPixMap(x, y) \cap (\texttt{badPixFlag1} \cup \texttt{badPixFlag2} \cup ...)$

The following operations are done on valid pixels ($badPixMask(x, y) = 0$).

- The excess count $crhsCount(x, y) = dbSubImage(x, y) - estimateImage(x, y)$ is attributed to CRHs if $crhsCount(x, y) > \texttt{crhsSigMul} \times estimateSigmaImage(x, y)$. Otherwise the $crhsCount(x, y)$ is set to 0.
- The bit corresponding to the bad pixels flag <`crhBadPixFlag`> is set in the output *crhBadPixMap*

  $crhBadPixMap(x, y) = (crhsCount(x, y) > 0) \cap \texttt{crhBadPixFlag}$

  and the result is *ored* with *estBadPixMap* to produce the output *crhBadPixMap*

  $crhBadPixMap(x, y) = estPixMap(x, y) \cup (crhsCount(x, y) > 0)$
- Optionally (if `crhsCrhImg` = `TRUE`) the image of the count of CRH photons $crhsCount(x, y)$ is given as output.
- **Quality Assessment**:
  The $crhsCount(x, y)$ frame is used to compute the total number of photons/second due to CRHs:

  $$crhsPhot = \left(\sum_{x,y} crhsCount(x, y) \right) / dbSubImage.\texttt{EXPTIME}$$

  If the total number `crhsPhot` is larger than the CRH alert value <`crhsAlertVal`> , a warning is due.

3.7.9 Pseudo code

3.8 Detect Cosmic ray hits in several frames

Name: **giDetectCosmicM** Section: **Preprocessing** nu: **60**
Author(s)/Date(s): **VC/20000507** Status: **To be validated by GS**

Working remarks:
Submitted to Internal rewiev Wed 09/06/00. The median instead of average will be used. Analysis has to be done in more details how we implement the median over few images (VC). AB 20010126: Verifier les image cmSubImage et mask crhmBadPixMap refrence dans la description math.; s'agit-il des donnees internes a la fonction ou des resultats (coherence avec les sortie declaree).

3.8.1 Purpose

The detection of CRH is obtained via the sigma-clipped (TBD mean or median) average of all input images.

3.8.2 Function parameters

Control parameters for σ-clipping average

Name	Type	Default	Description
`cmClipSigma`	`float`	`2.5`	multiple of σ (σ-clipping)
`cmClipNiter`	`integer`	`2`	number of iterations (σ-clipping)
`cmClipMfrac`	`float`	`0.7`	min fraction of points accepted/total (σ-clipping)
`cmMethod`	`integer`	`1`	1-noise from data, 2-noise from model
`cmSigMul`	`float`	`4.5`	multiple of σ used as threshold for local excess determination
`crhmAlertVal`	`float`	`10.0`	number of cosmic ray hits used as threshold for an alert
`badPixFlags`	`integer`	`TBD`	flags to be used with input images(s)
`cmCrhImg`	`logical`	`False`	flag, if set we produce image of the CRH count

3.8.3 Input frames

CRH detection for a set of multiple frames taken in similar conditions and with the same exposure time.

Name: *dbSubImage* **Type:** `BDRMIMG` **Format:** `xyPrep`
Description: *Nimage* similar bias-dark-subtracted GIRAFFE frames
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`GIRFTYPE`	DRS	`string`	none	type of frame data
`NX`	DRS	`integer`	none	Size of image in X direction (lambda)
`NY`	DRS	`integer`	none	Size of image in Y direction (slit)
`STARTX`	CCDDCS	`integer`	none	DIC: First window pixel in X direction within the detector physical system.
`STARTY`	CCDDCS	`integer`	none	DIC: First window pixel in Y direction within the detector physical system.
`EXPTIME`	UVES_ICS	`double`	sec	Total exposure time for a single exposure or for the individual exposure within a multiple frame
`SLIT`	FGIR_ICS	`integer`	none	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies
`GRATING`	FGIR_ICS	`string`	none	DIC: ESO name for grating unit.
`WLEN0`	FGIR_ICS	`double`	nm	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength

Name: *curBadPixMap* **Type:** `BADPIX` **Format:** `xyPrep`
Description: one or *Nimage* current mask(s) of bad pixels corresponding to the Nimage similar input frames; if only one mask is given (default) it will be used for all images.
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`CCDID`	CCDDCS	`string`	none	DIC: Detector system identification Format: <CCD Id> - <ACE Id>

3.8.4 Other inputs

none

3.8.5 Output frames

Name: *averageImage* **Type:** `ESTIMG` **Format:** `xyPrep`
Description: sigma-clipped (TBD mean or median) average image of the *Nimage* input frames
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`GIRFTYPE`	DRS	`string`	none	type of frame data
`CCLIPSIGMA`	DRS	`float`	none	multiple of sigma (sigma-clipping)
`CCLIPNITER`	DRS	`integer`	none	number of iterations (sigma-clipping)
`CCLIPMFRAC`	DRS	`float`	none	min fraction of points accepted/total (sigma-clipping)
`CSIGMAMULT`	DRS	`float`	none	multiple of sigma
`EXPTIME`	UVES_ICS	`double`	sec	Total exposure time for a single exposure or for the individual exposure within a multiple frame

Name: *sigmaImage* **Type:** `ESTSIGM` **Format:** `xyPrep`
Description: standard error image of the *Nimage* input frames
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`GIRFTYPE`	DRS	`string`	none	type of frame data
`CCLIPSIGMA`	DRS	`float`	none	multiple of sigma (sigma-clipping)
`CCLIPNITER`	DRS	`integer`	none	number of iterations (sigma-clipping)
`CCLIPMFRAC`	DRS	`float`	none	min fraction of points accepted/total (sigma-clipping)
`CSIGMAMULT`	DRS	`float`	none	multiple of sigma

Name: *crhmCount* **Type:** `CRHIMG` **Format:** `xyPrep`
Description: optional frame with the value of photons attributed to the total count of CRH for each pixel.
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`GIRFTYPE`	DRS	`string`	none	type of frame data
`CCLIPSIGMA`	DRS	`float`	none	multiple of sigma (sigma-clipping)
`CCLIPNITER`	DRS	`integer`	none	number of iterations (sigma-clipping)
`CCLIPMFRAC`	DRS	`float`	none	min fraction of points accepted/total (sigma-clipping)
`CSIGMAMULT`	DRS	`float`	none	multiple of sigma

Name: *crhBadPixMap* **Type:** `BADPIX` **Format:** `xyPrep`
Description: bad pixel map of the *averageImage*
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
CCDID	CCDDCS	string	none	DIC: Detector system identification Format: <CCD Id> - <ACE Id>
CCLIPSIGMA	DRS	float	none	multiple of sigma (sigma-clipping)
CCLIPNITER	DRS	integer	none	number of iterations (sigma-clipping)
CCLIPMFRAC	DRS	float	none	min fraction of points accepted/total (sigma-clipping)
CSIGMAMULT	DRS	float	none	multiple of sigma

3.8.6 Other outputs

none

3.8.7 General description

This function performs the CRH detection for a set of multiple frames taken in similar conditions (same `SLIT`, `GRATING` and `WLENO`). The detection of CRH is made simultaneously with the computation of the combined final image *averageImage* which is the output image and represents the most complete cleaned image available under the specified conditions of the computation.
First an averaged image is computed pixel by pixel from the complete set and with pixel value rejection. Two options are offered: the first one is a classical sigma clipping method and applies to frames with the same exposure time only because no scaling factor is used, while the second one operates with a variance computation using the averaged value at each iteration of the sigma clipping process. In this second option scaling by the exposure time can be adopted or not. The averaging method is either the median or the mean. Depending the number of frames (smaller than 10 or not) and the validity of the variance modelization, the median or the mean and one of the two options has to be chosen. The method which could apply with a fairly good result in a lot of cases (particularly with a small number of frames) is the use of median and the second option.
Then each individual frames of *dbSubImage* is scaled to the averaged image *averageImage* using the valid points only. The excess count is determined as $individualCrhmCount(x, y) = scaledDbSubImage(x, y) - averageImage(x, y)$. If the excess count exceeds a threshold, it is considered as a CRH.

3.8.8 Mathematical description

The CRHs will be detected using the data of $Nimage$ frames *dbSubImage* and their corresponding current masks of bad pixels *curBadPixMap* (a bad pixel mask 'MASTER' or only one bad pixel mask can be used for all the frames). The extraction of the $Nimage$ *badPixMask* is done with the same list of flags $<\texttt{badPixFlags}>$ for all the frames. Then for a frame j:

$$badPixMask_j(x, y) = curBadPixMap_j(x, y) \cap (\texttt{badPixFlag1} \cup \texttt{badPixFlag2} \cup ...)$$

In the following discussion the selection of pixel values will be done only if the pixel is valid in the bad pixel mask ($badPixMask(x, y) = 0$).

For each pixel (x, y) an iterative process is applied. First the preliminar estimate value $averageImage(x, y)$ is computed from the total number $Nimage$ of frames as well as the square root of the variance $sigmaImage(x, y)$ associated to it, taking only valid pixels.

- **if** $<\texttt{cmAverMeth} = \texttt{MEDIAN}>$, the estimate value is the mean of the two central values when the number of selected frames is even, and the central value when this number is odd.

- **if** $<\texttt{cmAverMeth} = \texttt{MEAN}>$, the value computed is the arithmetic mean of the selected points.

Then a rejection criterium is used iteratively and the pixel value for a frame j will be kept if:
$badPixMask_j(x, y) = 0$ and
$\mid dbSubImage_j(x, y) - averageImage(x, y) \mid < \texttt{cmClipSigma} \times sigmaImage(x, y)$
At each step new values for $averageImage(x, y)$ and $sigmaImage(x, y)$ are calculated. The sigma-clipping is

repeated till no pixel value is rejected or the < cmClipNiter > iterations have been applied or when two many frames have been rejected:

$$\frac{cmNpAccept}{Nimage} < \texttt{cmClipMfrac}$$

The number of retained values should never be smaller than 2. If after rejection the number of retained values is below the limit, the value with the smallest residual in absolute value is added back (in case of two values have the same residuals, they are kept). This process is repeated till the number of retained values reaches or exceeds the limit.
The detailed computations depend on the chosen method:

1. **if** < cmMethod > =**1**, then the estimate image *averageImage* is computed by attributing to each pixel the estimate value corresponding to selected frames and $sigmaImage(x, y)$ is the standard error computed for this sample:

$$sigmaImage(x, y)^2 = \left(\sum_{select.frames} (dbSubImage_j(x, y) - averageImage(x, y))^2 \right) / \; (select.frames.Number \, - \, 1)$$

2. **if** < cmMethod > =**2**, the same calculation is done for the estimate image but the variance is computed with a noise modelization.

   - **if** < cmScale = FALSE > , the data are not scaled and the variance is given by :

$$var(x, y) = averageImage(x, y) + dbSubImage.\texttt{RON}^2 + dbSubImage.\texttt{DARKVALUE} \times dbSubImage.\texttt{EXPTIME}$$

   - **if** < cmScale = TRUE > , the data are scaled by the exposure time:

$$cmSubImage_j(x, y) \; = \; dbSubImage_j(x, y) \, / \, dbSubImage_j.\texttt{EXPTIME}$$

   and then $averageImage(x, y)$ is computed from the selected values of $cmSubImage_j(x, y)$. Finally a variance is computed for each frame j:

$$var(x, y)_j = \frac{averageImage(x, y)}{dbSubImage_j.\texttt{EXPTIME}} + \left(\frac{dbSubImage_j.\texttt{RON}}{dbSubImage_j.\texttt{EXPTIME}} \right)^2 + \frac{dbSubImage_j.\texttt{DARKVALUE}}{dbSubImage_j.\texttt{EXPTIME}}$$

   The rejection criterium is now: $badPixMask_j(x, y) = 0$ and
   $\mid cmSubImage_j(x, y) - averageImage(x, y) \mid < \texttt{cmClipSigma} \times sigmaImage_j(x, y)$

At the end an estimate image *averageImage* and a standard error image *sigmaImage* are computed depending if a scaling has been performed:

1. if < cmScale = FALSE > this is done from the *dbSubImage* with only the retained pixels and the averaging method MEAN or MEDIAN. The standard error $sigmaImage(x, y)$ is the standard deviation of $dbSubImage_j(x, y)$ from its estimate value $averageImage(x, y)$ also for the retained pixels.
   In this case the exposure time of all the frames is assumed to be the same so the keyword EXPTIME of the output frame *averageImage* will be put equal to the one of the first frame in the set ($dbSubImage_1.\texttt{EXPTIME}$).

2. if < cmScale = TRUE > the estimate image is computed from the *cmSubImage* set with only the retained pixels and the averaging method MEAN or MEDIAN. The standard error $sigmaImage(x, y)$ is also the standard deviation of $cmSubImage_j(x, y)$ from its estimate value $averageImage(x, y)$ with the retained pixels.
   But those two images are scaled to the mean exposure time for the set calculated as:

$$averageImage.\texttt{EXPTIME} = \sum_{frames} dbSubImage_j.\texttt{EXPTIME} \, / \, Nimage$$

   And we have:

$$averageImage(x, y) = averageImage(x, y) \times averageImage.\texttt{EXPTIME}$$

$$sigmaImage(x, y) = sigmaImage(x, y) \times averageImage.\texttt{EXPTIME}$$

   The keyword EXPTIME of the output frame *averageImage* will be put equal to $averageImage.\texttt{EXPTIME}$.

Then an intensity scaling is applied to each individual image $dbSubImage_j$ in order to normalize each frame to the equivalent exposure time calculated for $averageImage$. For that a mean pixel intensity is computed with the valid pixels of each image; and the scaling factor is obtained by dividing the mean pixel intensity of $averageImage$ by the mean pixel intensity of image j.
If $<$ `cmScale` = `FALSE` $>$ the sigma-clipping process is repeated after substituting $scaledDbSubImage_j$ to $dbSubImage_j$. At the end a new intensity scaling is done for each image $dbSubImage_j$ but with the final set of valid pixels.

The scaled excess count represents the excess count which should occur for the equivalent exposure time of $averageImage$ and is determined as:

$$individualCrhmCount_j(x, y) = scaledDbSubImage_j(x, y) - averageImage(x, y)$$

In order to get of CRH for each individual frame, their bad pixel mask will be updated.
The following operations are done on valid pixels ($badPixMask_j = 0$).

- If $individualCrhmCount_j(x, y) < sigmaImage(x, y) \times$ `cmSigMul`, then the $individualCrhmCount_j(x, y)$ is set to 0. Otherwise the local excess is considered as a CRH.
- The bit corresponding to the bad pixels flag $<$ `crhBadPixFlag` $>$ is set in the output $crhBadPixMap_j$

  $$crhBadPixMapi_j(x, y) = (individualCrhmCount_j(x, y) > 0) \cap \texttt{crhBadPixFlag}$$

  and the result is *stored* with $curBadPixMap_j$ to produce the output $crhmBadPixMap$

  $$crhmBadPixMap_j(x, y) = curPixMap_j(x, y) \cup (individualCrhmCount_j(x, y) > 0)$$

The CRH bad pixels mask of the $averageImage$ is the product of all detected CRH:

$$crhmBadPixMap(x, y) = \prod_{n_{\text{image}}} ((individualCrhmCount_j(x, y) > 0) \cup curBadPixMap_j(x, y))$$

Optionally (if $<$ `cmCrhImg` $>$ =true) an image of the total count of CRH electrons is given as sum of all $individualCrhmCount(x, y)$.

Quality Assessment:
The $individualCrhmCount(x, y)$ frame is used to compute a frame with the total number of electrons/s due to CRH for each $dbSubImage_j$:

$$crhmPhot_j = \left(\sum_{x,y} individualCrhmCount(x, y)_j \right) / averageImage.\texttt{EXPTIME}$$

If the total number $\texttt{crhmPhot}_j$ for a given $dbSubImage_j$ is larger than the CRH alert value $<$ `crhmAlertVal` $>$, a warning message is printed out.

3.8.9 Pseudo code

3.9 Replace bad pixels

Name: **giReplaceFlagPix** Section: **Preprocessing** nu: **70**
Author(s)/Date(s): **FR/20000609** Status: **validated**

Working remarks:
Ready for pseudo-code (26 Sep 2000)

3.9.1 Purpose

Replace the bad pixels by their estimate in an image (could be the same image).

3.9.2 Function parameters

Name	Type	Default	Description
`badPixFlag`	`integer`	`"ALL"`	type of bad pixels to be replaced in the input image

3.9.3 Input frames

This function could take any `'xyPrep'` or `'nxExt'` formatted frame as input

Name: *dbSubImage* **Type:** `PREPIMG` **Format:** `xyPrep`
Description: any preprocessed GIRAFFE frame
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`GIRFTYPE`	DRS	`string`	none	type of frame data
`NX`	DRS	`integer`	none	Size of image in X direction (lambda)
`NY`	DRS	`integer`	none	Size of image in Y direction (slit)
`STARTX`	CCDDCS	`integer`	none	DIC: First window pixel in X direction within the detector physical system.
`STARTY`	CCDDCS	`integer`	none	DIC: First window pixel in Y direction within the detector physical system.
`CCDID`	CCDDCS	`string`	none	DIC: Detector system identification Format: <CCD Id> - <ACE Id>

Name: *estimateImage* **Type:** `ESTIMG` **Format:** `xyPrep`
Description: computed estimated image of *prepImage* frame
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`GIRFTYPE`	DRS	`string`	none	type of frame data
`NX`	DRS	`integer`	none	Size of image in X direction (lambda)
`NY`	DRS	`integer`	none	Size of image in Y direction (slit)
`STARTX`	CCDDCS	`integer`	none	DIC: First window pixel in X direction within the detector physical system.
`STARTY`	CCDDCS	`integer`	none	DIC: First window pixel in Y direction within the detector physical system.

Name: *crhBadPixMap* **Type:** `BADPIX` **Format:** `xyPrep`
Description: current *dbSubImage* bad pixel map
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`CCDID`	CCDDCS	`string`	none	DIC: Detector system identification Format: <CCD Id> - <ACE Id>

Name: *estBadPixMap* **Type:** `BADPIX` **Format:** `xyPrep`
Description: *estimateImage* bad pixel map
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`CCDID`	CCDDCS	`string`	none	DIC: Detector system identification Format: <CCD Id> - <ACE Id>

3.9.4 Other inputs

none

3.9.5 Output frames

Name: *repPixImage* **Type:** `REPIMG` **Format:** `xyPrep`
Description: clean image (bad pixels replaced)
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`GIRFTYPE`	DRS	`string`	none	type of frame data
`NX`	DRS	`integer`	none	Size of image in X direction (lambda)
`NY`	DRS	`integer`	none	Size of image in Y direction (slit)
`STARTX`	CCDDCS	`integer`	none	DIC: First window pixel in X direction within the detector physical system.
`STARTY`	CCDDCS	`integer`	none	DIC: First window pixel in Y direction within the detector physical system.

Name: *repBadPixMap* **Type:** `BADPIX` **Format:** `xyPrep`
Description: *repPixImage* bad pixel map = *estimateImage* bad pixel map
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`GIRFTYPE`	DRS	`string`	none	type of frame data
`CCDID`	CCDDCS	`string`	none	DIC: Detector system identification Format: <CCD Id> - <ACE Id>

3.9.6 Other outputs

none

3.9.7 General description

Pixels that are flagged in the input image, are replaced by their estimated value taken from *estimateImage*. This operation is implemented at various stages of the data reduction.
The bad pixels map of *repPixImage* is the update of *badPixMap* where replaced bad pixels are flagged.

3.9.8 Mathematical description

dbSubImage and *estimateImage* are merged according to *badPixMap* and <`badPixFlag`> :

$$\begin{aligned} repPixImage(x,y) &= estimateImage(x,y) \quad \text{if } \texttt{badPixFlag} \in crhBadPixMap(x,y) \\ &= dbSubImage(x,y) \quad \text{otherwise} \end{aligned}$$

The bad pixel mask of *repPixImage* contains the input bad-pixel mask, and the replaced pixels are moreover flagged with the 'repBadPixelFlag'.

3.9.9 Pseudo code

```
for\ i={\tt STARTX},{\tt NX} \{
     for\ j={\tt STARTY},{\tt NY} \{
         if\ ((badPixelMap(i,j) = badPixFlag) or (badPixFlag = 3))
             repPixImage(i,j) = estimateImage(i,j)
         else
             repPixImage(i,j) = prepImage(i,j)
     \}
\}
```

3.10 Adjust the scattered light

Name: **giAdjustSL** Section: **Preprocessing** nu: **80**
Author(s)/Date(s): **FR/20000608** Status: **To be validated by all**

Working remarks:
Review remarks included. To be validated. (18 Sep 2000)

3.10.1 Purpose

Adjust a model of scattered light to the inter-spectra region.

3.10.2 Function parameters

Modelization control parameters

Name	Type	Default	Description
slPhffCor	logical	FALSE	option for correcting the input frame from the photometric flat field
slModel	string	"POLYNOM"	option indicating if the model is a polynomial ("POLYNOM") or a polynomial fraction ("POLYFRAC")
slPolyDeg1X	integer	5	degree of the numerator polynomial of the model
slPolyDeg2X	integer	undef	degree of the denominator polynomial of the model
slPolyDeg1Y	integer	5	degree of the numerator polynomial of the model
slPolyDeg2Y	integer	undef	degree of the denominator polynomial of the model
contamW	integer	4	extra width parameter (in pixels) to cut out the part of inter-spectra region used for SL determination

3.10.3 Input frames

Name: *repPixImage* **Type:** REPIMG **Format:** xyPrep
Description: any cleaned, bias-and-dark subtracted image
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
NX	DRS	integer	none	Size of image in X direction (lambda)
NY	DRS	integer	none	Size of image in Y direction (slit)
STARTX	CCDDCS	integer	none	DIC: First window pixel in X direction within the detector physical system.
STARTY	CCDDCS	integer	none	DIC: First window pixel in Y direction within the detector physical system.
EXPTIME	UVES_ICS	double	sec	Total exposure time for a single exposure or for the individual exposure within a multiple frame
SLIT	FGIR_ICS	integer	none	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies
GRATING	FGIR_ICS	string	none	DIC: ESO name for grating unit.
WLEN0	FGIR_ICS	double	nm	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength
CCDID	CCDDCS	string	none	DIC: Detector system identification Format: <CCD Id> - <ACE Id>

Name: *phffImageMast* **Type:** PHFFIMG **Format:** xyPrep
Description: Master photometric flat-field calibration image for the current setup. Its use is optional
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
NX	DRS	integer	none	Size of image in X direction (lambda)
NY	DRS	integer	none	Size of image in Y direction (slit)
STARTX	CCDDCS	integer	none	DIC: First window pixel in X direction within the detector physical system.
STARTY	CCDDCS	integer	none	DIC: First window pixel in Y direction within the detector physical system.
EXPTIME	UVES_ICS	double	sec	Total exposure time for a single exposure or for the individual exposure within a multiple frame
SLIT	FGIR_ICS	integer	none	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies
GRATING	FGIR_ICS	string	none	DIC: ESO name for grating unit.
WLEN0	FGIR_ICS	double	nm	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength

Name: *locYMast* **Type:** LOCY **Format:** nxExt
Description: standard localization giving the spectrum center along y at each x for the current setup
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
EXTNX	DRS	integer	none	Number of pixel per spectrum
EXTNS	DRS	integer	none	Number of extracted spectra on the rebinned S2dFrame
SLIT	FGIR_ICS	integer	none	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies
GRATING	FGIR_ICS	string	none	DIC: ESO name for grating unit.
WLEN0	FGIR_ICS	double	nm	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength

Name: *locWyMast* **Type:** LOCWY **Format:** nxExt
Description: standard width of the NFF spectra (measured along the y axis) for the current setup
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
EXTNX	DRS	integer	none	Number of pixel per spectrum
EXTNS	DRS	integer	none	Number of extracted spectra on the rebinned S2dFrame
SLIT	FGIR_ICS	integer	none	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies
GRATING	FGIR_ICS	string	none	DIC: ESO name for grating unit.
WLEN0	FGIR_ICS	double	nm	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength

Name: *repBadPixMap* **Type:** `BADPIX` **Format:** `xyPrep`
Description: Current *repPixImage* bad pixel map
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`CCDID`	CCDDCS	`string`	none	DIC: Detector system identification Format: <CCD Id> - <ACE Id>

3.10.4 Other inputs

none

3.10.5 Output frames

Name: *slImage* **Type:** `SLIMG` **Format:** `xyPrep`
Description: image of scattered light
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`GIRFTYPE`	DRS	`string`	none	type of frame data
`SLPHFFCOR`	DRS	`logical`	none	option for correcting input frame frome photometric flat-field
`SLMODEL`	DRS	`string`	none	option indicating the model of scattered light: 'POLYNOM' or 'POLYFRAC'
`SLPOLYDEG1`	DRS	`integer`	none	degree of the numerator polynom of the SL model
`SLPOLYDEG2`	DRS	`integer`	none	degree of the denominator polynom of the SL model

3.10.6 Other outputs

none

3.10.7 General description

The inter-spectra pixels of the bias and dark subtracted image are used to model the contribution of scattered light on the whole image.
The model of scattered light (defined by <`slModel`>) is fitted on the inter-spectra regions of *repPixImage*. The inter-spectra regions are determined by the localization mask using $locYMast$ and $locWyMast$ which are chosen according to the keywords *repPixImage*.`SLIT`, *repPixImage*.`GRATING` and *repPixImage*.`WLENO` of the input frame. A width parameter <`contamW`> (additive to $locWyMast$) allows to shrink the interspectra region used in the estimation of the scattered light and then avoid the contamination by object light. This parameter will eliminate almost all the interspectra regions in an Argus/IFU frame, except the broader interslitlets regions.
As an option (depending on <`slPhffCor`>), the bias and dark subtracted image can be corrected by a photometric flat-field frame ($phffImageMast$) before modeling the scattered light.

3.10.8 Mathematical description

The model of scattered light, depending on <`slModel`> , is either a polynomial $\sum_{i,j=0}^{\texttt{slPolyDeg1X,slPolyDeg1Y}} c_{ij}T_i(x')T_j(y)$

or a polynomial fraction $\dfrac{\sum_{i,j=0}^{\texttt{slPolyDeg1X,slPolyDeg1Y}} a_{ij}T_i(x')T_j(y)}{\sum_{m,n=0}^{\texttt{slPolyDeg2X,slPolyDeg2Y}} b_{mn}T_m(x)T_n(y)}$.

Degrees of numerator and denominator are respectively defined by <`slPolyDeg1X`> , <`slPolyDeg1Y`> and <`slPolyDeg2X`> , <`slPolyDeg2Y`> (<`slPolyDeg2X`> and <`slPolyDeg2Y`> are zero in case of a polynomial model). Models are both represented using Chebyshev polynomials T_i defined on the normalized coordinates x' and y'.

A LSQ fit of the model is carried out on the inter-spectra region $repPixImage(x_{\rm is}, y_{\rm is})$ (optionally divided by $phffImageMast(x_{\rm is}, y_{\rm is})$), where $x_{\rm is}$ and $y_{\rm is}$ stand for coordinates of inter-spectra pixels ($locYMast(x_{\rm is}, n) + locWyMast(x_{\rm is}, n) + contamW < y_{\rm is} < locYMast(x_{\rm is}, n+1) - locWyMast(x_{\rm is}, n+1) - contamW$).
Then the full image of the scattered light *slImage* is constructed from the coefficients of the model with, optionally, the transform back to the unflatfielded image. There is no bad pixel mask for the scattered light image which is supposed defined (positive and lower than the linearity limit) for every pixels.

3.10.9 Pseudo code

3.11 Localization of spectra

Name: **giLocalSpectra** Section: **Extraction** nu: **90**
Author(s)/Date(s): **FR/19990902, AB/19991001** Status: **To be validated by all**

Working remarks:
Text is still to be detailed.

3.11.1 Purpose

Determine the centroid and the width of each spectral bin of each spectrum using the preliminary localization. A new current localization is defined through the LSQ fit of the optical model to positions and widths found.

3.11.2 Function parameters

Name	Type	Default	Description
`lClipSigma`	`float`	`4.5`	multiple of σ (σ-clipping)
`lClipNiter`	`integer`	`10`	number of iterations (σ-clipping)
`lClipMfrac`	`float`	`0.2`	min fraction of points accepted/total (σ-clipping)
`noiseMult`	`double`	`10`	noise detector multiple
`lPolyDeg`	`integer`	`5`	degree of the polynomial fit of the localization mask
`lComplete`	`integer`	`undef`	flag indicating if the localization is carried out over all spectra (0: according to the type of frames, 1: YES, 2: NO)
`lAddWy`	`double`	`0.5`	additional width to widen the fitted localization mask

3.11.3 Input frames

Name: *dbSubImage* **Type:** `BDRMIMG` **Format:** `xyPrep`
Description: any bias-dark-subtracted image
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`GIRFTYPE`	DRS	`string`	none	type of frame data
`NX`	DRS	`integer`	none	Size of image in X direction (lambda)
`NY`	DRS	`integer`	none	Size of image in Y direction (slit)
`STARTX`	CCDDCS	`integer`	none	DIC: First window pixel in X direction within the detector physical system.
`STARTY`	CCDDCS	`integer`	none	DIC: First window pixel in Y direction within the detector physical system.
`SLIT`	FGIR_ICS	`integer`	none	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies
`GRATING`	FGIR_ICS	`string`	none	DIC: ESO name for grating unit.
`WLEN0`	FGIR_ICS	`double`	nm	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength
`LINLIMIT`	DRS	`double`	none	linearity limit of the CCD in e-/pixel
`CCDID`	CCDDCS	`string`	none	DIC: Detector system identification Format: <CCD Id> - <ACE Id>
`RON`	DRS	`float`	e-	Expected CCD readout noise

Name: *locYMast* **Type:** LOCY **Format:** nxExt
Description: standard localization giving the spectrum center along y at each x for the current setup
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
EXTNX	DRS	integer	none	Number of pixel per spectrum
EXTNS	DRS	integer	none	Number of extracted spectra on the rebinned S2dFrame
SLIT	FGIR_ICS	integer	none	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies
GRATING	FGIR_ICS	string	none	DIC: ESO name for grating unit.
WLEN0	FGIR_ICS	double	nm	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength

Name: *locWyMast* **Type:** LOCWY **Format:** nxExt
Description: standard width of the NFF spectra (measured along the y axis) for the current setup
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
EXTNX	DRS	integer	none	Number of pixel per spectrum
EXTNS	DRS	integer	none	Number of extracted spectra on the rebinned S2dFrame
SLIT	FGIR_ICS	integer	none	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies
GRATING	FGIR_ICS	string	none	DIC: ESO name for grating unit.
WLEN0	FGIR_ICS	double	nm	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength

Name: *badPixMap* **Type:** BADPIX **Format:** xyPrep
Description: Current *dbSubImage* bad pixel map
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
CCDID	CCDDCS	string	none	DIC: Detector system identification Format: <CCD Id> - <ACE Id>

3.11.4 Other inputs

For standard localization the Narrow Flat Field frame is used.

Name	Type	Description
psfParamsMast	girPsfTable	Initial n-coefficient describing the shape of the PSF for any position $[x,y]$, SLIT dependent
Relevant FITS keywords:	GIRTTYPE SLIT GRATING WLEN0	
Relevant table columns:		
NSPEC	integer	running number of the spectrum
CWYIJKM	float	coefficients of 2-D polynomials describing the PSF(y) variation
ozpoz	girOzpozTable	Positionner Binary Table; this is in fact not an input, but part of the inframe1
Relevant FITS keywords:	none (attached to frame header)	
Relevant table columns:		
OBJECT_TYPE	character	object type: P=programme object, S=sky, G=guide star, X=simulation, F=flux refrence star (request OGL/OP)
SPECT_FIBRE	integer	fibre number on the spectrograph slit

3.11.5 Output frames

Name: *locY* **Type:** `LOCY` **Format:** `nxExt`
Description: current localization giving the spectrum center along y at each x
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`GIRFTYPE`	DRS	`string`	none	type of frame data
`LCLIPSIGMA`	DRS	`float`	none	multiple of sigma (sigma-clipping)
`LCLIPNITER`	DRS	`integer`	none	number of iterations (sigma-clipping)
`LCLIPMFRAC`	DRS	`float`	none	min fraction of points accepted/total (sigma-clipping)
`NOISEMULT`	DRS	`float`	none	noise detector multiple
`LPOLYDEG`	DRS	`integer`	none	degree of the polynomial fit
`EXTNX`	DRS	`integer`	none	Number of pixel per spectrum
`EXTNS`	DRS	`integer`	none	Number of extracted spectra on the rebinned S2dFrame

Name: *locWy* **Type:** `LOCWY` **Format:** `nxExt`
Description: current width of the NFF spectra (measured along the y axis)
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`GIRFTYPE`	DRS	`string`	none	type of frame data
`LCLIPSIGMA`	DRS	`float`	none	multiple of sigma (sigma-clipping)
`LCLIPNITER`	DRS	`integer`	none	number of iterations (sigma-clipping)
`LCLIPMFRAC`	DRS	`float`	none	min fraction of points accepted/total (sigma-clipping)
`NOISEMULT`	DRS	`float`	none	noise detector multiple
`LPOLYDEG`	DRS	`integer`	none	degree of the polynomial fit
`EXTNX`	DRS	`integer`	none	Number of pixel per spectrum
`EXTNS`	DRS	`integer`	none	Number of extracted spectra on the rebinned S2dFrame

3.11.6 Other outputs

Name	Type	Description
`psfParams`	`girPsfTable`	Current n-coefficient describing the shape of the PSF for any position [x,y]
Relevant FITS keywords:	`GIRTTYPE LCLIPSIGMA LCLIPNITER LCLIPMFRAC NOISEMULT LPOLYDEG`	
Relevant table columns:		
`NSPEC`	integer	running number of the spectrum
`CWYIJKM`	float	coefficients of 2-D polynomials describing the PSF(y) variation

3.11.7 General description

Though the standard application of the localization function will be with NFF images, the wavelength calibration frames or even the scientific images could be used in order to verify the current localization. Thus, localization is used either on NFF calibration frames (complete localization) or on scientific frames (only the 5 SIWC spectra), according to the parameter <`lComplete`> .

If the input frames *locYMast* and *locWyMast* are lacking when processing the complete localization using NFF calibration frame, preliminary values are computed as follows. Using the parameter <`noiseMult`> , pixels whose intensity is lower than the threshold value `noiseMult` $\times$ *dbSubImage*.`RON` are set to 0. For each x, each set of consecutive non-zero pixels gives the external limits of the border-pixels. If $y_{\max}$ and $y_{\min}$ are the respective borders of the set for the spectrum n:

$$upperborder(x, n) = y_{\max} + \frac{1}{2}$$

$$lowerborder(x,n) = y_{\min} - \frac{1}{2}$$

During this processing, saturated pixels (whose intensity is greater than the linearity limit *dbSubImage*.LINLIMIT)

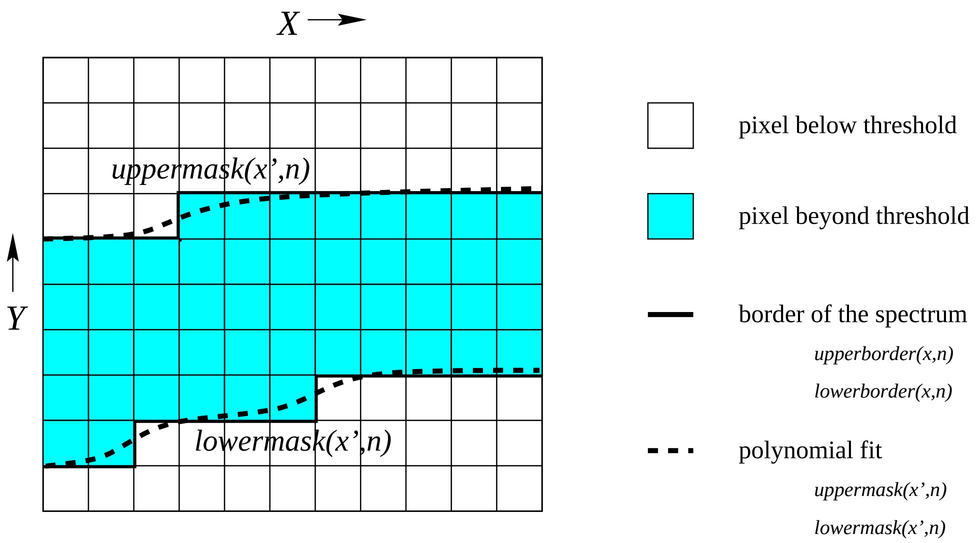


Figure 2: Preliminary localization for the determination of $locYMast$ and $locWyMast$

are not taken into account. In order to obtain a smooth localization, *upperborder* and *lowerborder* are fitted with polynomial expressions (resp. $uppermask(x,n)$ and $lowermask(x,n)$), of degree <lPolyDeg> . The LSQ fit is carried out using Chebyshev polynomials so that *uppermask* and *lowermask* are written as :

$$uppermask(x',n) = \sum_{i=0}^{\texttt{lPolyDeg}} p_i T_i(x')$$

$$lowermask(x',n) = \sum_{i=0}^{\texttt{lPolyDeg}} q_i T_i(x')$$

where T_i are Chebyshev polynomials and x' is the normalized x-coordinates ($\in [-1,+1]$).
The frames $locYMast(x,n)$ is set to $\frac{uppermask(x',n)+lowermask(x',n)}{2}$. In order to embrace the maximum flux, an extra-width represented by the parameter <lAddWy> is added to the half-width of the fitted mask and $locWyMast(x,n)$ is set to and $\frac{uppermask(x',n)-lowermask(x',n)}{2} + \texttt{lAddWy}$.
Preliminary values of $locYMast$ and $locWyMast$ could also be derived from the optical model of GIRAFFE.

3.11.8 Mathematical description

The localization proceeds as follows:

- Using the initial localization $locYMast(x,n)$, get the y position of the centroid $locCenter(x,n)$ in each x element of each spectrum n. For given x and n, limits of the localization mask are:

$$y_{\min} = \lfloor locYMast(x,n) - locWyMast(x,n) \rfloor$$

$$y_{\max} = \lceil locYMast(x,n) + locWyMast(x,n) \rceil$$

  The centroid is computed as follows:

$$locCenter(x,n) = \frac{\sum_{y=y_{\min}}^{y_{\max}} y \times dbSubImage(x,y) \times mask(x,y)}{\sum_{y=y_{\min}}^{y_{\max}} dbSubImage(x,y) \times mask(x,y)}$$

  where $mask(x,y)$ is equal to 0 for bad pixels, saturated pixels (whose intensity is greater than the linearity limit *dbSubImage*.LINLIMIT) and pixels below the threshold `noiseMult` $\times$ *dbSubImage*.RON.

- If the shape of the PSF is known and modeled, `psfParams` can be fitted to the observed profile to get simultaneously the center $y0$ and the width of the PSF. $y0$ is used as a new value of the centroid $locCenter(x, n)$ (see Fig. 3). Then, depending of the shape of the PSF, the half-width of the localization mask, $locWidth(x, n)$, is derived from the width parameter of the PSF.

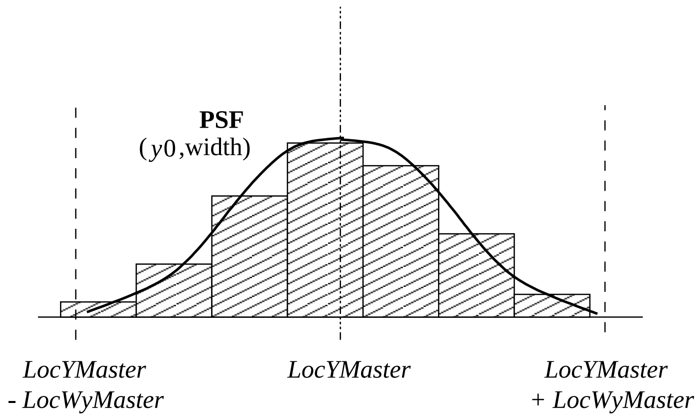


Figure 3: Fit of the PSF

- In order to insure the continuity of the localization solution, the difference between current centroid localization $locCenter(x, n)$ and previous localization $locYMast(x, n)$ is fitted using sigma clipping (if no specification is available on the shape of the localization, a polynomial fit of degree <`lPolyDeg`> will be carried out). If the difference is significant, the current localization $locY(x, n)$ is set to $locYMast(x, n)$ plus the fit of the difference. In a similar way the width $locWy(x, n)$ is set to the addition of $locWidth(x, n)$ and the fit of $locWyMast(x, n) - locWidth(x, n)$.

- The use of the knowledge of the PSF is made only for the standard localization with NFF frames. The localization on the 5 SIWC spectra will be carried out by simply computing the position of the centroid. The reduction recipe has the control over the decision whether this new localization becomes the new initial localization. The localization can be used iteratively.

3.11.9 Pseudo code

3.12 Spectra Extraction

Name: **giExtractSpectra** Section: **Extraction** nu: **100**
Author(s)/Date(s): **PN/20000615** Status: **To be validated by GS**

Working remarks:
Submitted to Internal rewiev Wed 09/06/00 including remarks that follow.

- **Disentangling of spectra:***In the option 1 (simple extraction by pixel addition), one could introduce an option allowing to disentangle the spectra, i.e. compensate for the contamination occuring at the detector level (especially in the IFU/ARGUS mode). The method would be the same as that adopted by the Italian group for the UVES instrument: after the extraction of the spectra, the observed intensity of a given spectrum at a given x is written as a sum of this spectrum and of adjacent spectra, weighted by the relative contributions expected from the known PSF profile. One gets a linear system of equations, the solution of which gives the uncontaminated intensities.*

  However, as pointed out by AB, curing for contamination at the CCD level does not solve the general contamination problem, far from it, since a significant amount of mutual contamination occurs anyway at the level of the focal plane in the IFU/ARGUS mode. Indeed, the microlenses are 0.52 arcsec across, which means that even with a 0.5 arcsec seeing, severe contamination of the adjacent fibres will occur. At the level of the focal surface, at least two adjacent fibres will give two adjacent spectra on the CCD; however, there will be other adjacent fibres whose corresponding spectra will not fall near the spectrum considered. Therefore, the partial cure suggested here might prove worse than the disease in some cases; nevertheless, disentangling may be offered as an option in the case of simple extraction, this is yet TBD.

  An argument in favour of the disentangling option is the following: in the case of narrow lines (especially emission lines, but strong absorption lines are also relevant), for instance calibration ThAr spectra, disentangling is mandatory even when the adjacent spectra on the CCD are identical, simply if they are slightly shifted relative to one another along the dispersion direction. This shift will exist in any case, even though it remains to be seen how significant it will be. Even if it is a small fraction of the PSF width, it may be large enough to bias RV estimates, typically in the case of rotation curves of galaxies measured in the IFU/ARGUS mode.

- **Default parameter values:** *Ideally, one should have a different default value for the MEDUSA mode than for the IFU/ARGUS mode, for some parameters at least. In practice a single default value is given, but one has to keep in mind that, at the recipe level, it may have to be modified according to the observing mode.*

- **Apparent inconsistency for extPsfWid:** *The transmitted parameters are* <extPsfWid1> *,* <extPsfWid2> *and* <extPsfWid3> *because one assumes that more than one width parameter might be required to model the PSF. All three parameters are referred to in the text as "*<extPsfWid> *i" (with i=1,2,3), hence the apparent inconsistency.*

3.12.1 Purpose

The extraction implements three options: summation along a virtual slit, classical Horne's and PSF fitting methods. The local error and the residual background are also computed.

3.12.2 Function parameters

control parameters for extraction PSF and σ-clipping.

Name	Type	Default	Description
extMethod	integer	3	Extraction method: 1=simple summation; 2=Horne's method; 3=LSQ fit
extWidthMult	float	1.	half-length of the extraction virtual slit, expressed in units of PSF width
extClipSigma	float	4.5	multiple of σ (σ-clipping)
extClipNiter	integer	10	number of iterations (σ-clipping)
extClipMfrac	float	0.2	min fraction of points accepted/total (σ-clipping)
extPsfPos	double	0.3	PSF position tolerance for CRH detection in Horne's extraction
extPsfWid1	double	0.03	PSF width1 tolerance for CRH detection in Horne's extraction
extPsfWid2	double	0.03	PSF width2 tolerance for CRH detection in Horne's extraction
extPsfWid3	double	0.03	PSF width3 tolerance for CRH detection in Horne's extraction
extPsfBkgd	double	0.04	PSF background tolerance for CRH detection in Horne's extraction
extPsfAmpl	double	0.10	PSF amplitude tolerance for CRH detection in Horne's extraction, or for LSQ fit extraction
extSmooBg	logical	true	flag for smoothing the background in option extMethod=3
extBkgXdeg	integer	5	polynomial degree of the background model for extMethod=3 and extSmooBg=TRUE (smoothing in the x direction)
extBkgYdeg	integer	15	polynomial degree of the background model (y direction)

3.12.3 Input frames

Name: *dbSubImage* **Type:** BDRMIMG **Format:** xyPrep
Description: any bias-dark-subtracted frame: extMethod=1 needs GIRFTYPE='REPIMG'
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
NX	DRS	integer	none	Size of image in X direction (lambda)
NY	DRS	integer	none	Size of image in Y direction (slit)
STARTX	CCDDCS	integer	none	DIC: First window pixel in X direction within the detector physical system.
STARTY	CCDDCS	integer	none	DIC: First window pixel in Y direction within the detector physical system.
RON	DRS	float	e-	Expected CCD readout noise
DARKVALUE	DRS	float	e-	The value of the actualy subtracted dark current. It is set to 0 if no subtraction is made (treshold not reached).
EXPTIME	UVES_ICS	double	sec	Total exposure time for a single exposure or for the individual exposure within a multiple frame
SLIT	FGIR_ICS	integer	none	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies
GRATING	FGIR_ICS	string	none	DIC: ESO name for grating unit.
WLEN0	FGIR_ICS	double	nm	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength

Name: *slImage* **Type:** `SLIMG` **Format:** `xyPrep`
Description: computed frame of fitted scattered light for *dbSubImage* (if available)
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`GIRFTYPE`	DRS	`string`	none	type of frame data
`NX`	DRS	`integer`	none	Size of image in X direction (lambda)
`NY`	DRS	`integer`	none	Size of image in Y direction (slit)
`STARTX`	CCDDCS	`integer`	none	DIC: First window pixel in X direction within the detector physical system.
`STARTY`	CCDDCS	`integer`	none	DIC: First window pixel in Y direction within the detector physical system.
`SLIT`	FGIR_ICS	`integer`	none	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies
`GRATING`	FGIR_ICS	`string`	none	DIC: ESO name for grating unit.
`WLEN0`	FGIR_ICS	`double`	nm	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength

Name: *locY* **Type:** `LOCY` **Format:** `nxExt`
Description: current localization giving the spectrum center along y at each x for the current setup.
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`GIRFTYPE`	DRS	`string`	none	type of frame data
`EXTNX`	DRS	`integer`	none	Number of pixel per spectrum
`EXTNS`	DRS	`integer`	none	Number of extracted spectra on the rebinned S2dFrame
`SLIT`	FGIR_ICS	`integer`	none	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies
`GRATING`	FGIR_ICS	`string`	none	DIC: ESO name for grating unit.
`WLEN0`	FGIR_ICS	`double`	nm	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength

Name: *locWy* **Type:** `LOCWY` **Format:** `nxExt`
Description: current width of the NFF spectra (measured along the y axis) for the current setup.
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`GIRFTYPE`	DRS	`string`	none	type of frame data
`EXTNX`	DRS	`integer`	none	Number of pixel per spectrum
`EXTNS`	DRS	`integer`	none	Number of extracted spectra on the rebinned S2dFrame
`SLIT`	FGIR_ICS	`integer`	none	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies
`GRATING`	FGIR_ICS	`string`	none	DIC: ESO name for grating unit.
`WLEN0`	FGIR_ICS	`double`	nm	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength

Name: *curBadPixMap* **Type:** `BADPIX` **Format:** `xyPrep`
Description: Current *dbSubImage* bad pixel map
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`CCDID`	CCDDCS	`string`	none	DIC: Detector system identification Format: <CCD Id> - <ACE Id>

3.12.4 Other inputs

Name	Type	Description
`psfParams`	`girPsfTable`	Current table of n-coefficient describing the shape of the PSF for any position [x,y]
Relevant FITS keywords:	`GIRTTYPE SLIT GRATING WLEN0`	
Relevant table columns:		
`NSPEC`	integer	running number of the spectrum
`CWYIJKM`	float	coefficients of 2-D polynomials describing the PSF(y) variation

3.12.5 Output frames

Name: *exSp* **Type:** `EXTSP` **Format:** `nxExt`
Description: frame of extracted spectra
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`GIRFTYPE`	DRS	`string`	none	type of frame data
`EXTNX`	DRS	`integer`	none	Number of pixel per spectrum
`EXTNS`	DRS	`integer`	none	Number of extracted spectra on the rebinned S2dFrame
`ECLIPSIGMA`	DRS	`float`	none	multiple of sigma (sigma-clipping)
`ECLIPNITER`	DRS	`integer`	none	number of iterations (sigma-clipping)
`ECLIPMFRAC`	DRS	`float`	none	min fraction of points accepted/total (sigma-clipping)
`EXTWMULT`	DRS	`double`	none	number of PSF width parameters defining the half-length of the extraction virtual slit
`EXTMETHOD`	DRS	`integer`	none	extraction method: 1=simple summation; 2=Horne's method; 3=LSQ fit
`EXTBKGYDEG`	DRS	`integer`	none	polyomial degree of the background model (y direction)
`EXTPSFPOS`	DRS	`float`	none	PSF position tolerance for CRH det. in method 2
`EXTPSFWID1`	DRS	`float`	none	PSF width1 tolerance for CRH det. in method 2
`EXTPSFWID2`	DRS	`float`	none	PSF width2 tolerance for CRH det. in method 2
`EXTPSFWID3`	DRS	`float`	none	PSF width3 tolerance for CRH det. in method 2
`EXTPSFBKGD`	DRS	`float`	none	PSF backgrnd tolerance for CRH det. in method 2
`EXTPSFAMPL`	DRS	`float`	none	PSF amplitude tolerance for CRH det. in method 2

Name: *exSpErr* **Type:** `EXTERRS` **Format:** `nxExt`
Description: frame of extracted spectra errors
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
EXTNX	DRS	integer	none	Number of pixel per spectrum
EXTNS	DRS	integer	none	Number of extracted spectra on the rebinned S2dFrame
ECLIPSIGMA	DRS	float	none	multiple of sigma (sigma-clipping)
ECLIPNITER	DRS	integer	none	number of iterations (sigma-clipping)
ECLIPMFRAC	DRS	float	none	min fraction of points accepted/total (sigma-clipping)
EXTWMULT	DRS	double	none	number of PSF width parameters defining the half-length of the extraction virtual slit
EXTMETHOD	DRS	integer	none	extraction method: 1=simple summation; 2=Horne's method; 3=LSQ fit
EXTBKGYDEG	DRS	integer	none	polyomial degree of the background model (y direction)
EXTPSFPOS	DRS	float	none	PSF position tolerance for CRH det. in method 2
EXTPSFWID1	DRS	float	none	PSF width1 tolerance for CRH det. in method 2
EXTPSFWID2	DRS	float	none	PSF width2 tolerance for CRH det. in method 2
EXTPSFWID3	DRS	float	none	PSF width3 tolerance for CRH det. in method 2
EXTPSFBKGD	DRS	float	none	PSF backgrnd tolerance for CRH det. in method 2
EXTPSFAMPL	DRS	float	none	PSF amplitude tolerance for CRH det. in method 2

Name: *exSpNpixels* **Type:** `EXTNPIX` **Format:** `nxExt`
Description: frame of pixel counts in extracted spectra
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
EXTNX	DRS	integer	none	Number of pixel per spectrum
EXTNS	DRS	integer	none	Number of extracted spectra on the rebinned S2dFrame
ECLIPSIGMA	DRS	float	none	multiple of sigma (sigma-clipping)
ECLIPNITER	DRS	integer	none	number of iterations (sigma-clipping)
ECLIPMFRAC	DRS	float	none	min fraction of points accepted/total (sigma-clipping)
EXTWMULT	DRS	double	none	number of PSF width parameters defining the half-length of the extraction virtual slit
EXTMETHOD	DRS	integer	none	extraction method: 1=simple summation; 2=Horne's method; 3=LSQ fit
EXTBKGYDEG	DRS	integer	none	polyomial degree of the background model (y direction)
EXTPSFPOS	DRS	float	none	PSF position tolerance for CRH det. in method 2
EXTPSFWID1	DRS	float	none	PSF width1 tolerance for CRH det. in method 2
EXTPSFWID2	DRS	float	none	PSF width2 tolerance for CRH det. in method 2
EXTPSFWID3	DRS	float	none	PSF width3 tolerance for CRH det. in method 2
EXTPSFBKGD	DRS	float	none	PSF backgrnd tolerance for CRH det. in method 2
EXTPSFAMPL	DRS	float	none	PSF amplitude tolerance for CRH det. in method 2

Name: *exBadPixMap* **Type:** `EXTBPM` **Format:** `nxExt`
Description: Current mask of bad pixels on extracted spectra
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`GIRFTYPE`	DRS	`string`	none	type of frame data
`EXTNX`	DRS	`integer`	none	Number of pixel per spectrum
`EXTNS`	DRS	`integer`	none	Number of extracted spectra on the rebinned S2dFrame
`ECLIPSIGMA`	DRS	`float`	none	multiple of sigma (sigma-clipping)
`ECLIPNITER`	DRS	`integer`	none	number of iterations (sigma-clipping)
`ECLIPMFRAC`	DRS	`float`	none	min fraction of points accepted/total (sigma-clipping)
`EXTWMULT`	DRS	`double`	none	number of PSF width parameters defining the half-length of the extraction virtual slit
`EXTMETHOD`	DRS	`integer`	none	extraction method: 1=simple summation; 2=Horne's method; 3=LSQ fit
`EXTBKGYDEG`	DRS	`integer`	none	polyomial degree of the background model (y direction)
`EXTPSFPOS`	DRS	`float`	none	PSF position tolerance for CRH det. in method 2
`EXTPSFWID1`	DRS	`float`	none	PSF width1 tolerance for CRH det. in method 2
`EXTPSFWID2`	DRS	`float`	none	PSF width2 tolerance for CRH det. in method 2
`EXTPSFWID3`	DRS	`float`	none	PSF width3 tolerance for CRH det. in method 2
`EXTPSFBKGD`	DRS	`float`	none	PSF backgrnd tolerance for CRH det. in method 2
`EXTPSFAMPL`	DRS	`float`	none	PSF amplitude tolerance for CRH det. in method 2

Name: *exBack* **Type:** `EXTBKG` **Format:** `xPolyY`
Description: frame of K polynomial coefficients for Nx independent background models (if extMethod=3)
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`GIRFTYPE`	DRS	`string`	none	type of frame data
`EXTNX`	DRS	`integer`	none	Number of pixel per spectrum
`EXTBKGCOEF`	DRS	`integer`	none	K polynomial coefficients for NX independant background models
`ECLIPSIGMA`	DRS	`float`	none	multiple of sigma (sigma-clipping)
`ECLIPNITER`	DRS	`integer`	none	number of iterations (sigma-clipping)
`ECLIPMFRAC`	DRS	`float`	none	min fraction of points accepted/total (sigma-clipping)
`EXTWMULT`	DRS	`double`	none	number of PSF width parameters defining the half-length of the extraction virtual slit
`EXTMETHOD`	DRS	`integer`	none	extraction method: 1=simple summation; 2=Horne's method; 3=LSQ fit
`EXTBKGYDEG`	DRS	`integer`	none	polyomial degree of the background model (y direction)
`EXTPSFPOS`	DRS	`float`	none	PSF position tolerance for CRH det. in method 2
`EXTPSFWID1`	DRS	`float`	none	PSF width1 tolerance for CRH det. in method 2
`EXTPSFWID2`	DRS	`float`	none	PSF width2 tolerance for CRH det. in method 2
`EXTPSFWID3`	DRS	`float`	none	PSF width3 tolerance for CRH det. in method 2
`EXTPSFBKGD`	DRS	`float`	none	PSF backgrnd tolerance for CRH det. in method 2
`EXTPSFAMPL`	DRS	`float`	none	PSF amplitude tolerance for CRH det. in method 2

Name: *cleanImage* **Type:** ESTIMG **Format:** xyPrep
Description: cleaned GIRAFFE frame with CRH replaced by interpolated values if extMethod=3
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
NX	DRS	integer	none	Size of image in X direction (lambda)
NY	DRS	integer	none	Size of image in Y direction (slit)
STARTX	CCDDCS	integer	none	DIC: First window pixel in X direction within the detector physical system.
STARTY	CCDDCS	integer	none	DIC: First window pixel in Y direction within the detector physical system.
ECLIPSIGMA	DRS	float	none	multiple of sigma (sigma-clipping)
ECLIPNITER	DRS	integer	none	number of iterations (sigma-clipping)
ECLIPMFRAC	DRS	float	none	min fraction of points accepted/total (sigma-clipping)
EXTWMULT	DRS	double	none	number of PSF width parameters defining the half-length of the extraction virtual slit
EXTMETHOD	DRS	integer	none	extraction method: 1=simple summation; 2=Horne's method; 3=LSQ fit
EXTBKGYDEG	DRS	integer	none	polyomial degree of the background model (y direction)
EXTPSFPOS	DRS	float	none	PSF position tolerance for CRH det. in method 2
EXTPSFWID1	DRS	float	none	PSF width1 tolerance for CRH det. in method 2
EXTPSFWID2	DRS	float	none	PSF width2 tolerance for CRH det. in method 2
EXTPSFWID3	DRS	float	none	PSF width3 tolerance for CRH det. in method 2
EXTPSFBKGD	DRS	float	none	PSF backgrnd tolerance for CRH det. in method 2
EXTPSFAMPL	DRS	float	none	PSF amplitude tolerance for CRH det. in method 2

Name: *slImage* **Type:** SLIMG **Format:** xyPrep
Description: updated frame of fitted scattered light (if extMethod=3)
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
NX	DRS	integer	none	Size of image in X direction (lambda)
NY	DRS	integer	none	Size of image in Y direction (slit)
STARTX	CCDDCS	integer	none	DIC: First window pixel in X direction within the detector physical system.
STARTY	CCDDCS	integer	none	DIC: First window pixel in Y direction within the detector physical system.
ECLIPSIGMA	DRS	float	none	multiple of sigma (sigma-clipping)
ECLIPNITER	DRS	integer	none	number of iterations (sigma-clipping)
ECLIPMFRAC	DRS	float	none	min fraction of points accepted/total (sigma-clipping)
EXTWMULT	DRS	double	none	number of PSF width parameters defining the half-length of the extraction virtual slit
EXTMETHOD	DRS	integer	none	extraction method: 1=simple summation; 2=Horne's method; 3=LSQ fit
EXTBKGYDEG	DRS	integer	none	polyomial degree of the background model (y direction)
EXTPSFPOS	DRS	float	none	PSF position tolerance for CRH det. in method 2
EXTPSFWID1	DRS	float	none	PSF width1 tolerance for CRH det. in method 2
EXTPSFWID2	DRS	float	none	PSF width2 tolerance for CRH det. in method 2
EXTPSFWID3	DRS	float	none	PSF width3 tolerance for CRH det. in method 2
EXTPSFBKGD	DRS	float	none	PSF backgrnd tolerance for CRH det. in method 2
EXTPSFAMPL	DRS	float	none	PSF amplitude tolerance for CRH det. in method 2

Name: *clbadPixMap* **Type:** `BADPIX` **Format:** `xyPrep`
Description: Updated current bad pixel map (associated to the *cleanImage* frame)
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
CCDID	CCDDCS	string	none	DIC: Detector system identification Format: <CCD Id> - <ACE Id>
ECLIPSIGMA	DRS	float	none	multiple of sigma (sigma-clipping)
ECLIPNITER	DRS	integer	none	number of iterations (sigma-clipping)
ECLIPMFRAC	DRS	float	none	min fraction of points accepted/total (sigma-clipping)
EXTWMULT	DRS	double	none	number of PSF width parameters defining the half-length of the extraction virtual slit
EXTMETHOD	DRS	integer	none	extraction method: 1=simple summation; 2=Horne's method; 3=LSQ fit
EXTBKGYDEG	DRS	integer	none	polyomial degree of the background model (y direction)
EXTPSFPOS	DRS	float	none	PSF position tolerance for CRH det. in method 2
EXTPSFWID1	DRS	float	none	PSF width1 tolerance for CRH det. in method 2
EXTPSFWID2	DRS	float	none	PSF width2 tolerance for CRH det. in method 2
EXTPSFWID3	DRS	float	none	PSF width3 tolerance for CRH det. in method 2
EXTPSFBKGD	DRS	float	none	PSF backgrnd tolerance for CRH det. in method 2
EXTPSFAMPL	DRS	float	none	PSF amplitude tolerance for CRH det. in method 2

3.12.6 Other outputs

none

3.12.7 General description

This function applies to any preprocessed science or calibration frame, obtained either in the MEDUSA or ARGUS/IFU mode. Three options are offered: the first and the second ones can be applied in the MEDUSA mode because contamination should be negligible in this mode, while the third one should be adopted for the ARGUS/IFU mode because it works even in cases of severe mutual contamination of the spectra. Nevertheless, the user can decide otherwise.

1. Simple addition of pixel values along a virtual slit; it is assumed that the SL was previously removed.
2. If the crosstalk is very low (expected case in MEDUSA mode) or the adjacent spectra are of similar intensity (close to uniform object in ARGUS/IFU mode) a pure Horne's method could be used. Note, that in that case the SL should be removed previously:

$$exSp(x_c, n) = \frac{\sum_{i \in slit} W_H(x_c, y_i) \times dbSubImage(x_c, y_i)/PSF(psfParams(x_c, y_c, npar), y_i - y_c))}{\sum_{i \in slit} W_H(x_c, y_i)}$$

   and

$$W_H(x_c, y_i) = PSF(psfParams(x_c, y_c, npar), y_i - y_c))^2/vardbSubImage(x_c, y_i)$$

   The elimination of CRH is carried out in the way proposed by Baranne et al. (RD10).

3. Each spectral bin (a slice in y-direction of the preprocessed image) is described as a linear combination of the PSF multiplying the spectral elements $exSp(x_c, n)$ which we want to extract (note that the y_c and the width of the PSF are supplied through the $locY(x_c, n)$ and $locWy(x_c, n)$ parameters not shown explicitly in the formula below):

$$modelexSp(x_c, y) = bg(x_c, y) + \sum_{n=1}^{\mathtt{EXTNS}} exSp(x_c, n) \times PSF(psfParams(x_c, y_c, npar), y - y_c)) \quad (3)$$

The extraction is carried out for each x-bin through the LSQ fit of preprocessed data $dbSubImage(x_c, y)$ to the equation 4. Only the amplitudes $exSp(x_c, n)$ multiplying the fixed PSF and the global background are free parameters. Each pixel $dbSubImage(x_c, y)$ is weighted, for the LSQ fit, by the inverse of its variance

$$vardbSubImage(x_c, y) = (dbSubImage(x_c, y) + dbSubImage.\mathtt{RON}^2 + dbSubImage.\mathtt{DARKVALUE} \times dbSubImage.\mathtt{EXPTIME})$$

and

$$weight(x_c, y) = \frac{1}{vardbSubImage(x_c, y)}$$

so that the quantity to be minimized is

$$\chi^2 = \sum_{i=\mathtt{STARTY}}^{\mathtt{STARTY+NY}} weight(x_c, y_i) \times (dbSubImage(x_c, y_i) - modelexSp(x_c, y_i))^2$$

The fitted model is compared to the data and *sigma-clipping* is implemented based on the local sigma expected from the model. This is the optimal way to remove the CRH. A mask of rejected points $curBadPixMap$ is updated in this process.

Equation 4 is a generalization of Horne's optimal extraction to the case where the spectra are not strictly independent due to the cross-talk. In addition, the background has to be determined simultaneously. We assume here that it is a slowly varying function of y:

$$bg(x_c, y) = \sum_{k=0}^{\mathtt{extBkgYdeg}} a_k(x_c) y^k$$

so that equation 4 is a linear system of equations.

The $exSpError(x_c, n)$ is set to the error of the $exSp(x_c, n)$ estimated from the fit and $exSpNpixels(x_c, n)$ to the number of pixels used for each extracted point. Note that the $exSpNpixels(x_c, n)$ plays the role of the bad pixel mask on the extracted spectra.

This is a general method well adapted to any situation (including the case where the inter-spectra contamination is severe).

3.12.8 Mathematical description

1. **if** `extMethod` $= 1$**, then:** Apply a simple summation along a virtual slit. The slit length is defined by

$$\begin{aligned} lengSlit(x, n) &= \min(2 \times \mathtt{extWidthMult} \times locWy(x, n), (locY(x, n+1) - locY(x, n)), \\ &\quad (locY(x, n) - locY(x, n-1)) \qquad \forall x, n \end{aligned}$$

since it cannot be longer than the distance between two successive spectra. In this formula, $locY(x, 0) = \mathtt{STARTY} - 1.0$ and $locY(x, \mathtt{EXTNS} + 1) = \mathtt{NY}$. (The slit width is 1 pixel, the extraction being done for each pixel in the x direction). The value of $lengslit(x, n)$ is stored into the keyword $exSp$.`EXTWMULT`.

The extracted spectrum will then be:

$$exSp(x_c, n) = \sum_{i \in slit} prop_i \times dbSubImage(x_c, y_i)$$

where $prop_i$ is the fraction of the pixel to extract (one has $prop_i = 1.0$ for all pixels but those two lying at the slit extremities, for which $0.0 < prop_i < 1.0$). *It is assumed that CRH detection has already been made*, since no additional CRH detection is made during this extraction process. *Provided the FF spectra are extracted in exactly the same way, it is not necessary that the bad pixels are interpolated*; in other word the bad pixels are simply dropped in the summation. This is true even when the scientific frame has been corrected for high spatial frequencies *before* extraction, because in such a case the extracted spectra are further divided by the high frequency corrected and extracted NFF ones: if the NFF spectra are also extracted dropping bad pixels, the corrections is valid.

This option also assumes that the scattered light has been already estimated and subtracted, using e.g. inter-subslit regions.

The extracted spectrum error will be the quadratic sum of the pixel flux errors:

$$exSpErr(x_c, n) = \left(\sum_{i \in slit} prop_i^2 \times vardbSubImage(x_c, y_i) \right)^{1/2}$$

where $vardbSubImage$ is given by

$$vardbSubImage(x_c, y) = \\ (dbSubImage(x_c, y) + dbSubImage.\texttt{RON}^2 + dbSubImage.\texttt{DARKVALUE} \times dbSubImage.\texttt{EXPTIME})$$

Note that the interpolated pixels are *not*included in this estimate, if, as stated above, the NFF spectra are extracted in the same way. The resulting estimated error will be larger if one or more bad pixels are dropped in the summation, which is the most natural way to account for bad pixels here.

The frame $exSpNpixels$ of pixel counts in extracted spectra does not make much sense here; the pixel number will be set to the fractional number of pixels in the virtual slit (including the interpolated pixels), which varies with the position on the CCD, since the PSF width also varies.

The frame $exBack$ of K polynomial coefficients for `NX` independent background models remains undefined in this option.

The map $exBadPixMap$ is computed as the ratio of the number of bad pixels present within the slit (according to $curBadPixMap$), over the total number of pixels in the slit (these numbers may be fractional); therefore, the numbers stored in $exBadPixMap$ are real and range between 0 and 1.

Deblending of extracted spectra: The simple extraction scheme described above would work well in the MEDUSA mode but not in the IFU and ARGUS modes because of the mutual contamination of the spectra. However, in the latter case it is still possible to recover the real intensities by solving a linear set of equations, provided the shape of the PSF and the spectra positions are well known. This option is described in the function *giDeblendSpectra*().

2. **if** `extMethod` $= 2$**, then:** Apply Horne's method (RD7) independently to each spectrum:

$$exSp(x_c, n) = \\ \frac{\sum_{i \in slit} curBadPixMap(x_c, y_i) \times W_H(x_c, y_i) \times dbSubImage(x_c, y_i) / PSF(psfParams(x_c, y_c, npar), y_i - y_c))}{\sum_{i \in slit} curBadPixMap(x_c, y_i) \times W_H(x_c, y_i)}$$

with

$$W_H(x_c, y_i) = PSF(psfParams(x_c, y_c, npar), y_i - y_c)^2 / var(x_c, y_i)$$

where var is *not* equal to $vardbSubImage$, but is the *expected* variance frame. This choice is made because in cases of low SNR, $vardbSubImage$ may be too far from the real variance image since it is based on highly uncertain counts. On the contrary, var is computed on the basis of the whole extracted flux (see below) whose SNR is significantly better.

The CRH are detected in the way proposed by Baranne et al. (RD10) during the extraction, in the following process:

(a) Preliminary extraction by summing the pixel fluxes along a virtual slit (as in the 1st option above, but excluding bad pixels), allowing a first estimate of the variance:

$$exSp(x_c, n) = \sum_{i \in slit} prop_i \times curBadPixMap(x_c, y_i) \times dbSubImage(x_c, y_i)$$

If no bad pixel is present, the PSF (which is normalized so that its integral is unity) multiplied by this extracted flux directly gives the expected spectrum profile. However, in the presence of bad pixels, the flux extracted in this way is smaller than the true one, and an appropriate scaling must be done (in the extreme case where all pixels are bad, one quits the loop):

(b) Computation of the expected profile at the relevant pixel y positions:

$$exprofile(x_c, y) = exSp(x_c, n) \times \frac{prop \times PSF(psfParams(x_c, y_c, npar), y - y_c)}{\sum_{i \in slit} pprop_i \times curBadPixMap(x_c, y_i) \times PSF(psfParams(x_c, y_c, npar), y_i - y_c)}$$

(c) Computation of the expected variance

$$var(x_c, y) = exprofile(x_c, y) + dbSubImage.\texttt{RON}^2 + dbSubImage.\texttt{DARKVALUE} \times dbSubImage.\texttt{EXPTIME} + slImage(x_c, y)$$

(d) Spectrum extraction using Horne's optimal method (see equation above)

(e) Search for CRH using both Horne's criterion and the "X_{max}" criterion that takes into account any error in profile modeling (notice that CRH are searched for only in pixels outside the current bad pixel map $curBadPixelMap(x_c, y)$):

$$(prop \times dbSubImage(x_c, y) - exprofile(x_c, y))^2 > \texttt{extClipSigma}^2 \times var(x_c, y)$$

$$\frac{prop \times dbSubImage(x_c, y) - exprofile(x_c, y)}{prop \times dbSubImage(x_c, y) + exprofile(x_c, y)} > X_{max,k}, \forall k = 1, ...3 + npar$$

where

$$X_{max,k} = \frac{PSF(par_k + \varepsilon_k) - PSF(par_k)}{PSF(par_k + \varepsilon_k) + PSF(par_k)}$$

for all ε_k but the last one ε_{ampl} (see below), for which

$$X_{max,ampl} = \frac{\varepsilon_{ampl}}{2}$$

all other parameters of the PSF being equal. The relevant parameters are the position <`extPsfPos`> , the width ε_{width_i} = `extPsfWid`i (possibly represented by $npar > 1$ parameters, depending on the complexity of the PSF profile), the background ε_{back} = `extPsfBkgd` and the amplitude ε_{ampl} = `extPsfAmpl` multiplying the PSF (the PSF itself is normalized to a unit integral). The ε_k values, which represent small uncertainties on the parameters mentioned[2], will have to be optimized during the commissioning phase.

(f) When a CRH is detected at the position (x_c, y), the bad pixel map $exBadPixMap$ is updated. Only one CRH pixel can be rejected per iteration: if other pixels satisfy the above criteria, only the most discrepant one is rejected. Then another iteration is performed by going to point 2a again, until no more CRH is detected.

(g) The bad pixels (due to cosmetics or CRH) are interpolated by attributing to them the fluxes

$$exSp(x_c, n) \times PSF(psfParams(x_c, y_c, npar), y - y_c)$$

and the result is stored in the new frame $cleanImage$.

(h) The expected variance of each profile point is updated:

$$var(x_c, y) = exSp(x_c, y) + dbSubImage.\texttt{RON}^2 + dbSubImage.\texttt{DARKVALUE} \times dbSubImage.\texttt{EXPTIME} + slImage(x_c, y)$$

where $slImage$ is added only if the scattered light is significant and has been subtracted.

[2]The uncertainty on the position is expressed in pixels, but the others are expressed as relative values: ε_{width_i} is expressed as a fraction of the width parameter, while ε_{back} and ε_{ampl} are both expressed as fractions of the amplitude. Note that ε_{ampl} takes into account relative variations of pixel sensitivity, which are not corrected at this level.

(i) The error on the extracted flux is computed:

$$exSpErr(x_c, n) = \left(\sum_{i \in slit} PSF(psfParams(x_c, y_c, npar), y_i - y_c)^2 / var(x_c, y_i) \right)^{-1/2}$$

The map $exBadPixMap$ is computed as the ratio of the total number of bad pixels found within the slit (before and during extraction), over the total number of pixels in the slit.

3. **if** `extMethod` $= 3$**, then:** Each spectral bin is modeled as a linear combination of the PSF multiplying the spectral elements $exSp(x_c, n)$ which we want to extract:

$$modelexSp(x_c, y) = bg(x_c, y) + \sum_{n=1}^{\mathtt{EXTNS}} exSp(x_c, n) \times PSF(psfParams(x_c, y_c, npar), y - y_c)) \quad (4)$$

with

$$bg(x_c, y') = \sum_{k=0}^{\mathtt{extBkgYdeg}} a_k(x_c) T_k(y') \quad (5)$$

where T_k are Chebyshev polynomials of degree k and $y' = \frac{2}{\mathtt{NY}-1}(y - STARTY - \frac{\mathtt{NY}-1}{2})$, so that equation 4 is a linear system of equations. The extraction is carried out for each x-bin through the LSQ fit of preprocessed data $dbSubImage(x_c, y)$ to the equation 4. Only the amplitudes $exSp(x_c, n)$ multiplying the fixed PSF and the global background are free parameters. The PSF profile is normalized so that its integral is unity; in this way, the adjusted amplitudes $exSp(x_c, n)$ may be directly interpreted as spectral intensities.

- In the very first iteration, each pixel $dbSubImage(x_c, y)$ is weighted, for the LSQ fit, by the inverse of its measured variance

$$weight(x_c, y) = \frac{1}{vardbSubImage(x_c, y)}$$

$$vardbSubImage(x_c, y) = (dbSubImage(x_c, y) + dbSubImage.\mathtt{RON}^2 + dbSubImage.\mathtt{DARKVALUE} \times dbSubImage.\mathtt{EXPTIME})$$

(only an average value of the dark is used; it does not seem justified to use the whole $darkImageMast$) so that the quantity to be minimized is

$$\chi^2 = \sum_{i=\mathtt{STARTY}}^{\mathtt{STARTY}+\mathtt{NY}-1} weight(x_c, y_i) \times (dbSubImage(x_c, y_i) - modelexSp(x_c, y_i))^2$$

In the estimate of the weight, it is assumed that the scattered light has *not* been subtracted yet, but is entirely modeled by Equation 5.

In terms of matrices, the problem can be expressed after defining the $(\mathtt{EXTNS} + \mathtt{extBkgYdeg} + 1) \times \mathtt{NY}$ matrix `A` containing the PSF functions and and Chebyshev polynomials, the `S` column vector containing the `NY` observed intensities and the `I` column vector containing the `EXTNS+extBkgYdeg+1` spectral intensities and coefficients of the background which result from the LSQ fit. Explicitly, the matrices are the following:

$$\mathtt{A} = \begin{bmatrix} PSF_{1,1} & .. & PSF_{1,EXTNS} & 1 & T_1(-1) & .. & T_{\mathtt{extBkgXdeg}}(-1) \\ . & .. & . & . & . & .. & . \\ PSF_{j,1} & .. & PSF_{j,EXTNS} & 1 & T_1(\frac{2}{\mathtt{NY}'}(j-1-\frac{\mathtt{NY}'}{2}) & .. & T_{\mathtt{extBkgXdeg}}(\frac{2}{\mathtt{NY}'}(j-1-\frac{\mathtt{NY}'}{2}) \\ . & .. & . & . & . & .. & . \\ PSF_{NY,1} & .. & PSF_{NY,EXTNS} & 1 & T_1(1) & .. & T_{\mathtt{extBkgXdeg}}(1) \end{bmatrix}$$

where

$$PSF_{j,l} = PSF(psfParams(x_c, y_{cl}, npar), STARTY + j - 1 - y_{cl})$$

The weights are put in a diagonal matrix `W`:

$$\begin{aligned} W_{jj} = weight(x_c, STARTY + j - 1) &= \frac{1}{vardbSubImage(x_c, STARTY + j - 1)} \quad \forall\ good\ pixels \quad (6) \\ &= 0 \quad \forall\ \text{bad}\ pixels \end{aligned}$$

But $vardbSubImage(x_c, j-1+STARTY)$ is replaced by $varmodelxSp(x_c, STARTY+j-1)$ as soon as the first iteration has been carried out.
There are two column vectors, one for the observed signal:

$$\mathtt{S} = \left[\ dbSubImage(x_c, STARTY + j - 1) \qquad\qquad j = 1, ...\mathtt{NY}\ \right]$$

and the other for the intensity to be extracted, including the coefficients of the background:

$$\mathtt{I} = \begin{bmatrix} exSp(x_c, 1) \\ . \\ exSp(x_c, EXTNS) \\ a_0 \\ . \\ a_{\mathtt{extBkgXdeg}} \end{bmatrix}$$

Writing $(S - AI)^t$ the transpose of $(S - AI)$, χ^2 can be expressed as:

$$\chi^2 = (S - AI)^t W (S - AI)$$

And χ^2 is minimum if:

$$I = (A^t W A)^{-1} A^t W S$$

The covariance matrix of the fitted parameters will be [RD12][3]:

$$cov(I) = (A^t W A)^{-1}$$

the variances themselves being the diagonal elements.

- In a second iteration, the weight is computed from $modelexSp$, which should give a better estimate:

$$weight(x_c, y) = \frac{1}{varmodelexSp(x_c, y)}$$

with

$$varmodelexSp(x_c, y) = \\ modelexSp(x_c, y) + dbSubImage.\mathtt{RON}^2 + dbSubImage.\mathtt{DARKVALUE} \times dbSubImage.\mathtt{EXPTIME}$$

Any pixel with a difference satisfying both conditions

$$dbSubImage(x_c, y_i) - modelexSp(x_c, y_i) > \mathtt{extClipSigma} \times (varmodelexSp(x_c, y_i))^{1/2}$$

$$\frac{dbSubImage(x_c, y_i) - modelexSp(x_c, y_i)}{dbSubImage(x_c, y_i) + modelexSp(x_c, y_i)} > X_{max}(\varepsilon_{ampl}) = \frac{\varepsilon_{ampl}}{2}$$

is rejected, and the mask of rejected points $curBadPixMap$ is updated. The second condition takes into account the pixel-to-pixel sensitivity variations through the ε_{ampl} parameter, which is a fraction of the signal.

- The LSQ fit is iterated until no additional pixel is rejected, or until the number of iterations < `extClipNiter` > is reached, or until the fraction of accepted pixels becomes lower than < `extClipMfrac` > .
- If `extSmooBg` = *true*, the background obtained from `NX` independent LSQ fits along the y axis is smoothed by `NY` LSQ fits, this time along the x axis:

$$bglis(x', y) = \sum_{l=0}^{\mathtt{extBkgXdeg}} b_l(y) T_l(x')$$

[3] Press et al. 1992, Numerical Recipes, 2nd edition, Cambridge University Press, p. 667

where T_l are Chebyshev polynomials of degree l and $x' = \frac{2}{\mathtt{NX}-1}(x - STARTX - \frac{\mathtt{NX}-1}{2})$.
Then, the smoothed background is subtracted

$$dbSubImage(x, y) = dbSubImage(x, y) - bglis(x, y)$$

and the extraction reiterated, but *without* the $bg(x_c, y)$ term in Equation 4.

The $exSpErr(x_c, n)$ is set to the error of the $exSp(x_c, n)$ estimated from the fit; however, the number of pixels $exSpNpixels(x_c, n)$ is not very relevant here, since the spectra are not considered separately but are extracted all at once: there is no well-defined integration range. Similarly, there is no simple way to define the map of rejected bad pixels $exBadPixMap$. The relevant information will be entirely included in $exSpErr(x_c, n)$, which will be large for "bad" pixels of extracted spectra.

The fitted background is then identified to the scattered light:

$$slImage(x, y) = bg(x, y) \quad \forall x, y$$

3.12.9 Pseudo code

bidon

3.13 Deblending of spectra extracted in option 1 or 2 of giExtractSpectra()

Name: **giDeblendSpectra** Section: **Extraction** nu: **105**
Author(s)/Date(s): **PN/20001106** Status: **To be validated by GS**

Working remarks:
Submitted to internal review Mon 06/11/00.

3.13.1 Purpose

3.13.2 Function parameters

none

3.13.3 Input frames

Name: *exSp* **Type:** EXTSP **Format:** nxExt
Description: extracted spectra
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
EXTNX	DRS	integer	none	Number of pixel per spectrum
EXTNS	DRS	integer	none	Number of extracted spectra on the rebinned S2dFrame
SLIT	FGIR_ICS	integer	none	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies
GRATING	FGIR_ICS	string	none	DIC: ESO name for grating unit.
WLEN0	FGIR_ICS	double	nm	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength
EXTWMULT	DRS	double	none	number of PSF width parameters defining the half-length of the extraction virtual slit

Name: *exSpErr* **Type:** EXTERRS **Format:** nxExt
Description: corresponding extracted spectra errors
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
EXTNX	DRS	integer	none	Number of pixel per spectrum
EXTNS	DRS	integer	none	Number of extracted spectra on the rebinned S2dFrame
SLIT	FGIR_ICS	integer	none	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies
GRATING	FGIR_ICS	string	none	DIC: ESO name for grating unit.
WLEN0	FGIR_ICS	double	nm	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength
EXTWMULT	DRS	double	none	number of PSF width parameters defining the half-length of the extraction virtual slit

Name: *locY* **Type:** `LOCY` **Format:** `nxExt`
Description: current localization giving the spectrum center along y at each x for the current setup.
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
EXTNX	DRS	integer	none	Number of pixel per spectrum
EXTNS	DRS	integer	none	Number of extracted spectra on the rebinned S2dFrame
SLIT	FGIR_ICS	integer	none	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies
GRATING	FGIR_ICS	string	none	DIC: ESO name for grating unit.
WLEN0	FGIR_ICS	double	nm	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength

Name: *locWy* **Type:** `LOCWY` **Format:** `nxExt`
Description: current width of the NFF spectra (measured along the y axis) for the current setup.
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
EXTNX	DRS	integer	none	Number of pixel per spectrum
EXTNS	DRS	integer	none	Number of extracted spectra on the rebinned S2dFrame
SLIT	FGIR_ICS	integer	none	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies
GRATING	FGIR_ICS	string	none	DIC: ESO name for grating unit.
WLEN0	FGIR_ICS	double	nm	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength

3.13.4 Other inputs

Name	Type	Description
psfParams	girPsfTable	Current table of n-coefficient describing the shape of the PSF for any position [x,y]
Relevant FITS keywords:	GIRTTYPE SLIT GRATING WLEN0	
Relevant table columns:		
NSPEC	integer	running number of the spectrum
CWYIJKM	float	coefficients of 2-D polynomials describing the PSF(y) variation

3.13.5 Output frames

Name: *exDebSp* **Type:** `EXTSP` **Format:** `nxExt`
Description: frame of deblended spectra (that were extracted using option 1 or 2 of giExtractSpectra)
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`GIRFTYPE`	DRS	`string`	none	type of frame data
`EXTNX`	DRS	`integer`	none	Number of pixel per spectrum
`EXTNS`	DRS	`integer`	none	Number of extracted spectra on the rebinned S2dFrame
`EXTWMULT`	DRS	`double`	none	number of PSF width parameters defining the half-length of the extraction virtual slit
`EXTMETHOD`	DRS	`integer`	none	extraction method: 1=simple summation; 2=Horne's method; 3=LSQ fit

Name: *exDebSpErr* **Type:** `EXTERRS` **Format:** `nxExt`
Description: frame of extracted and deblended spectra errors
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`GIRFTYPE`	DRS	`string`	none	type of frame data
`EXTNX`	DRS	`integer`	none	Number of pixel per spectrum
`EXTNS`	DRS	`integer`	none	Number of extracted spectra on the rebinned S2dFrame
`EXTWMULT`	DRS	`double`	none	number of PSF width parameters defining the half-length of the extraction virtual slit
`EXTMETHOD`	DRS	`integer`	none	extraction method: 1=simple summation; 2=Horne's method; 3=LSQ fit

3.13.6 Other outputs

none

3.13.7 General description

This function recovers the true spectral intensities of mutually contaminated spectra, after they have been extracted according to the option `extMethod` = 1 (i.e. simple pixel addition along a virtual slit). It cannot be applied to spectra extracted with `extMethod` = 2 (Horne's optimal extraction neglecting contamination) because in this method, each pixel is weighted in a different way and taking this into account would imply much complication.
This is done through the solution of a linear system of equations, whose coefficients are computed from the supposedly known PSF (in the y direction).
Ideally, this function should be applied only in the case where the high spatial frequency corrections have been applied *before* the spectra extraction, using the normalized FSFF image (see the $giNormalizeFSFF()$ function). Otherwise, the FF correction can only be approximate.

3.13.8 Mathematical description

This function recovers the real intensities by solving a linear set of equations, provided the shape of the PSF and the spectra positions are well known.
The extracted intensities are written as a linear superposition of the nearby spectra (these may be only both adjacent ones or include yet more distant ones, this remains TBD):

$$exSp(x_c, n) = \sum_{i=1}^{exSp.\mathtt{EXTNS}} a_{ni}\, exDebSp(x_c, i) \tag{7}$$

Or, in matrix notation:

$$S = A \cdot I \tag{8}$$

where the vector S contains the intensities of the extracted spectra at x_c

$$\mathtt{S} = \left[\; exSp(x_c, n) \qquad n = 1, ...exSp.\mathtt{EXTNS} \; \right]$$

the vector I contains the intensities of the deblended extracted spectra at x_c

$$\mathtt{I} = \left[\; exDebSp(x_c, n) \qquad n = 1, ...exSp.\mathtt{EXTNS} \; \right]$$

and the square matrix A contains the a_{ij} coefficients, which have to be computed from integrations of the PSF (see below). The solution to this system of equation is

$$I = A^{-1} \cdot S \tag{9}$$

Since a given spectrum will be essentially contaminated by both adjacent ones, and maybe (in cases of a PSF with very extended wings) by the second adjacent ones, the matrix A will be "band diagonal" and easier to invert (see e.g. Press et al. 1992, Numerical Recipes in FORTRAN, 2nd edition, pp. 43-47).
The A matrix has to be computed before, and this is done by using the coefficients of the PSF perpendicular to the dispersion stored in the `Psfparams` table:

$$a_{mn}(x_c) = \int_{locY(x_c,m)-0.5\cdot lengslit(x_c,m)}^{locY(x_c,m)+0.5\cdot lengslit(x_c,m)} PSF(x_c, y, n)dy \tag{10}$$

with

$$PSF(x_c, y, n) = f(width_1(x_c, n), .., width_l(x_c, n), .., width_{npar}(x_c, n)) \tag{11}$$

where f is a function yet TBD (e.g. a sum of gaussians) and

$$width_l(x'_c, n) = \sum_{i=0}^{I} \sum_{j=0}^{J} c_{ijkl} \, T_i(x'_c) T_j(y'(n)) \tag{12}$$

where the primed coordinates are normalized to the interval [-1,1] (e.g. $y' = \frac{2}{\mathtt{NY}-1}(y - STARTY - \frac{\mathtt{NY}-1}{2})$, T_i is the Chebyshev polynomial of degree i, I and J are the maximal degrees in the x and y directions respectively (implicitly given in the table `Psfparams` through the number of coefficients c_{ijkl} with same i and same j respectively) and c_{ijkl} being the coefficients given in the `Psfparams` table; k enumerates the subslit and is univocally defined by the spectrum number n. The length of the slit $lengslit(x_c, m)$ is computed again from the $exSp$.`EXTWMULT` keyword and from the frames $locY$ and $locWy$, in exactly the same way as in the function $giExtractSpectra()$ (with <`extWidthMult`> replaced by $exSp$.`EXTWMULT`).
The errors on the extracted fluxes can be computed in the same way as the fluxes themselves, since the variances of the observed extracted fluxes can be written as linear combinations of the variances of the true (deblended) fluxes:

$$varexSp(x_c, n) = \sum_{i=1}^{exSp.\mathtt{EXTNS}} a_{ni}^2 \, varexDebSp(x_c, i) \tag{13}$$

Or, in matrix notation:

$$VARS = A2 \cdot VARI \tag{14}$$

where the vector $VARS$ contains the variances of the extracted spectra at x_c

$$\mathtt{VARS} = \left[\; exSpErr(x_c, n)^2 \qquad n = 1, ...exSp.\mathtt{EXTNS} \; \right]$$

the vector $VARI$ contains the variances of the deblended extracted spectra at x_c

$$\mathtt{VARI} = \left[\; exDebSpErr(x_c, n)^2 \qquad n = 1, ...exSp.\mathtt{EXTNS} \; \right]$$

and the square matrix $A2$ contains the squared coefficients a_{ij}^2 of the matrix A. The solution to this system is

$$VARI = A2^{-1} \cdot VARS \tag{15}$$

and should be always positive. Finally, the error on the deblended fluxes is simply computed by taking the square root of the elements of the $VARI$ vector:

$$\begin{aligned} exDebSpErr(x_c, n) &= \sqrt{vari(x_c, n)} && \text{if } vari(x_c, n) > 0 \\ &= 0 && \text{if } vari(x_c, n) \leq 0 \end{aligned}$$

A warning must be issued if any variance value turns out to be negative or zero.

3.14 Global FF correction

Name: **giFlatSpectra** Section: **Flux calibration** nu: **110**
Author(s)/Date(s): **PN/20000615** Status: **To be validated by GS**

Working remarks:
Submitted to Internal review Wed 09/06/00.

3.14.1 Purpose

The extracted spectra are corrected for spectrograph and detector response (including high spatial frequencies), but they remain modulated by the spectrum of the FF calibration lamp.

3.14.2 Function parameters

none

3.14.3 Input frames

Name: *exSp* **Type:** `EXTSP` **Format:** `nxExt`
Description: extracted spectra
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`GIRFTYPE`	DRS	`string`	none	type of frame data
`EXTNX`	DRS	`integer`	none	Number of pixel per spectrum
`EXTNS`	DRS	`integer`	none	Number of extracted spectra on the rebinned S2dFrame
`SLIT`	FGIR_ICS	`integer`	none	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies
`GRATING`	FGIR_ICS	`string`	none	DIC: ESO name for grating unit.
`WLEN0`	FGIR_ICS	`double`	nm	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength

Name: *exSpErr* **Type:** `EXTERRS` **Format:** `nxExt`
Description: corresponding extracted spectra errors
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`GIRFTYPE`	DRS	`string`	none	type of frame data
`EXTNX`	DRS	`integer`	none	Number of pixel per spectrum
`EXTNS`	DRS	`integer`	none	Number of extracted spectra on the rebinned S2dFrame
`SLIT`	FGIR_ICS	`integer`	none	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies
`GRATING`	FGIR_ICS	`string`	none	DIC: ESO name for grating unit.
`WLEN0`	FGIR_ICS	`double`	nm	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength

Name: *exNffMast* **Type:** `EXTSP` **Format:** `nxExt`
Description: extracted NFF spectra (Master NFF); It is assumed without bad pixels. This frame is set-up dependent and the selection is made on KW's SLIT:GRATING:WLEN0
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
EXTNX	DRS	integer	none	Number of pixel per spectrum
EXTNS	DRS	integer	none	Number of extracted spectra on the rebinned S2dFrame
SLIT	FGIR_ICS	integer	none	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies
GRATING	FGIR_ICS	string	none	DIC: ESO name for grating unit.
WLEN0	FGIR_ICS	double	nm	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength
REFSRCMULT	MISSING	float	normed	Factor which multiply the reference emissivity of the lamp to take into the account the setup-dependent fraction of the light injected into the fiber.

Name: *exNffErrMast* **Type:** `EXTERRS` **Format:** `nxExt`
Description: corresponding extracted NFF spectra errors (Error of Master NFF); same condition as for the NFF applies
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
EXTNX	DRS	integer	none	Number of pixel per spectrum
EXTNS	DRS	integer	none	Number of extracted spectra on the rebinned S2dFrame
SLIT	FGIR_ICS	integer	none	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies
GRATING	FGIR_ICS	string	none	DIC: ESO name for grating unit.
WLEN0	FGIR_ICS	double	nm	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength

3.14.4 Other inputs

none

3.14.5 Output frames

Name: *ffSp* **Type:** `EXTSP` **Format:** `nxExt`
Description: flat-fielded spectra modulated (divided) by the reference spectrum
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
EXTNX	DRS	integer	none	Number of pixel per spectrum
EXTNS	DRS	integer	none	Number of extracted spectra on the rebinned S2dFrame
FFNORM	DRS	float	none	Multiplicative factor used to normalize the FF extracted spectra. To go back to the original signal: FForig=normedFF*FFNORM

Name: *ffSpErr* **Type:** `EXTERRS` **Format:** `nxExt`
Description: corresponding flat-fielded spectra errors
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
EXTNX	DRS	integer	none	Number of pixel per spectrum
EXTNS	DRS	integer	none	Number of extracted spectra on the rebinned S2dFrame

3.14.6 Other outputs

none

3.14.7 General description

This function applies to both science and sky spectra, *before* sky subtraction and before rebinning to the wavelength space. It corrects for both low and high spatial frequencies of instrumental efficiency, i.e. for blaze function, relative individual fibre transmission (as far as it does not depend on the fibre's position), overall CCD quantum efficiency and pixel-to-pixel sensitivity variations.
The same function applies to every mode of GIRAFFE since in this context, no fundamental difference exist between them, as soon as extraction was done. The only difference is in the number of spectra.
Because the high spatial frequencies can distort, in principle, the wavelength solution, the 5 calibration spectra in the science frames or all spectra of a SEWC should pass through this function before the wavelength solution is computed or updated.
Both scientific spectra and calibration Narrow Flat-Fields have been preprocessed in the same way, therefore both are sets of uncalibrated spectra. The optical path is *almost* the same (with the exception of the focal ratio of the illuminating beam) and we expect that even the high-frequency corrections (Pixel-to-pixel CCD response and fringing) will be accurate enough. The scientific and NFF extracted spectra could be written respectively:

$$exSp(x,n) = flxSp(x,n) \times tFibre(x,n) \times tSpectro(x,n) \times QE_n(x,n)$$

$$exNFF(x,n) = flxNff(x) \times tFibre(x,n) \times tSpectro(x,n) \times QE_n(x,n)$$

The $flxSp(x,n)$ are fluxes at the fibre entrance of the spectral element that fall on the CCD pixel x of the n-th spectrum. The $flxNff(x)$ is considered as unique (independent of n), which assumes the uniform illumination of all fibres during the calibration exposure.
The reference flux $flxNff(x)$ comes from the laboratory calibrations of the lamp and is sampled in the λ-space; it is not used here, but only in the function $giSubtractSky()$.
The ratio of the two equations above gives the Flat-field corrected input spectra $ffSp(x,n)$ modulated by the spectrum of the calibration source. The $ffSpError(x,n)$ is obtained in a similar way. In order to get intensities which remain close to the original ones, the NFF spectra are divided by a normalization factor. But the information on the fibre transmission must be kept, so this factor must be the same for all spectra. The best way to define it is first to take the average value of each NFF spectrum; then, the median of these average values is chosen as the normalization factor, so that dead fibres will have almost no influence on its value. NFF spectra divided by this factor will have average values close to one, except those corresponding to bad fibres, whose average values will be substantially smaller than one.

$$ffSp(x,n) = \frac{flxSp(x,n)}{flxNff(x)} \times \texttt{FFNORM} = \frac{extractSp(x,n)}{extractNff(x,n)} \times \texttt{FFNORM}$$

Note that the fibre transmissions $tFibre(x,n)$ and spectrograph response $tSpectro(x,n)$ are canceled out at this step and will not therefore be used explicitly anywhere in the reduction software. Also the *normalization* to the λ-sampled reference spectrum $flxNff$ will not be removed here and left to the wavelength re-sampled spectra (if required). However, the effects of telescope vignetting and of the Nasmyth corrector are not corrected for here; they are in the function $giVignCorr()$.

3.14.8 Mathematical description

The normalization factor is first defined:

$$ffSp.\texttt{FFNORM} = median(\overline{exNffMast(n)}), \quad n = 1, ...\texttt{EXTNS}$$

where

$$\overline{exNffMast(n)} = \frac{1}{\texttt{EXTNX}} \sum_{i=1}^{\texttt{EXTNX}} exNffMast(x_i, n)$$

These average values are stored in the table `MeanNff` for possible further use by the *giFiberTrans*() function. The flat-fielded spectra are obtained by dividing the extracted science spectra by the extracted and normalized NFF ones:

$$ffSp(x,n) = \frac{exSp(x,n)}{exNffMast(x,n)} \times ffSp.\texttt{FFNORM}$$

The flat-fielded spectra errors are obtained through the usual propagation formula:

$$ffSpError(x,n) = ffSp(x,n) \times \left(\left(\frac{exSpErr(x,n)}{exSp(x,n)} \right)^2 + \left(\frac{exNffErrMast(x,n)}{exNffMast(x,n)} \right)^2 \right)^{1/2}$$

In practice, the latter formula is only relevant to high SNR spectra.
The algorithm will then be the following:

1. For each pixel (in the x direction) of each spectrum, compute $ffSp(x,n)$
2. For each pixel for which $\frac{exNffErrMast(x,n)/exNffMast(x,n)}{exSpErr(x,n)/exSp(x,n)} > 0.1418$ (in which case the relative contribution of $exNffErrMast$ is 1% only[4]), compute the error following the formula

   $$ffSpError(x,n) = ffSp(x,n) \times \left(\left(\frac{exSpErr(x,n)}{exSp(x,n)} \right)^2 + \left(\frac{exNffErrMast(x,n)}{exNffMast(x,n)} \right)^2 \right)^{1/2}$$

   while for the other pixels, compute the error through the formula

   $$ffSpError(x,n) = \frac{exSpErr(x,n)}{exNffMast(x,n)} \times ffSp.\texttt{FFNORM}$$

3.14.9 Pseudo code

```
do\ n=1,{\tt EXTNS}
     do\ x=1,{\tt EXTNX}
         ffSp(x,n) = extractSp(x,n)/extractNff(x,n)
         if\ ((extractNffError(x,n)/extractNff(x,n))/(extractError(x,n)/extractSp(x,n)) > 0.1418)\ then
             ffSpError(x,n) = ffSp(x,n) *
                     sqrt((extractError(x,n)/extractSp(x,n))**2 + (extractNffError(x,n)/extractNff(x,n))*
         else
             ffSpError(x,n) = extractError(x,n)/extractNff(x,n)
         endif
     enddo
enddo
```

[4]If the condition is written $b/a > 0.1418$, then we have in the limiting case $(b/a)^2 = 0.020107$ and $\sqrt{a^2+b^2} = a\sqrt{1+(b/a)^2} = 1.010a$, hence the 1% contribution

3.15 Normalization of the FSFF calibration image

Name: **giNormalizeFSFF** Section: **Preprocessing** nu: **115**
Author(s)/Date(s): **PN/20000823** Status: **To be validated by GS**

> ***Working remarks:***
> *Submitted to Internal review Wed 09/06/00.*

3.15.1 Purpose

Normalizes the FSFF image to a unit average value, so that only the high spatial frequencies remain

3.15.2 Function parameters

Polynomial degrees to model the low spatial frequencies of the FSFF

Name	Type	Default	Description
`fsffXDeg`	`integer`	`undef`	polynomial degree along the x axis for FSFF modeling
`fsffYDeg`	`integer`	`undef`	polynomial degree along the y axis for FSFF modeling

3.15.3 Input frames

Name: *ffFullSlit* **Type:** `BDRMIMG` **Format:** `xyPrep`
Description: preprocessed FSFF calibration image
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`GIRFTYPE`	DRS	`string`	none	type of frame data
`NX`	DRS	`integer`	none	Size of image in X direction (lambda)
`NY`	DRS	`integer`	none	Size of image in Y direction (slit)
`STARTX`	CCDDCS	`integer`	none	DIC: First window pixel in X direction within the detector physical system.
`STARTY`	CCDDCS	`integer`	none	DIC: First window pixel in Y direction within the detector physical system.
`SLIT`	FGIR_ICS	`integer`	none	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies
`GRATING`	FGIR_ICS	`string`	none	DIC: ESO name for grating unit.
`WLEN0`	FGIR_ICS	`double`	nm	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength

3.15.4 Other inputs

none

3.15.5 Output frames

Name: *ffHighFreq* **Type:** `FSFFIMG` **Format:** `xyPrep`
Description: normalized FSFF image
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`GIRFTYPE`	DRS	`string`	none	type of frame data

3.15.6 Other outputs

none

3.15.7 General description

This function is used when the user chooses to correct for high spatial frequencies of the pixel sensitivity (or for fringes) *before* the spectra extraction rather than after it. This solution applies whenever suspicion arises that the scientific spectra are not localized at exactly the same places as the NFF ones.
In order to do so, a pre-processed FSFF image must be available. This function then fits the low frequency variations of this FSFF image and divides the original image by the resulting fit. The result is an image whose pixel values are all close to one. The scientific image has to be divided by this image, as well as the NFF image which will be used afterwards for low frequency corrections through the $giFlatSpectra()$ function.

3.15.8 Mathematical description

It is assumed that the input frame has been cleaned from the bad pixels (CRH detected by `giDetectCosmicM`, or by `giDetectCosmicS` if only one image is available, and all bad pixels – cosmetics and CRH – excluded using `giEstimatePix` and `giReplaceFlagPix`) before entering this function.
The following steps are carried out successively:

1. The low frequency variations of the FF intensity are modeled by 2D Chebyshev polynomials, after normalization of the CCD pixel coordinates to the interval [-1,+1]:

$$fflowfreq(x', y') = \sum_{i=0}^{\mathtt{fsffXDeg}} \sum_{j=0}^{\mathtt{fsffYDeg}} clf_{i,j} T_i(x') T_j(y') \tag{16}$$

   where $x' = \frac{2}{\mathtt{NX}-1}(x - STARTX - \frac{\mathtt{NX}-1}{2})$, $y' = \frac{2}{\mathtt{NY}-1}(y - STARTY - \frac{\mathtt{NY}-1}{2})$, and T_i are Chebyshev polynomials of degree i. An LSQ fit is done of the $fflowfreq(x', y')$ function.

2. The FSFF image is divided by the model to give the normalized FFSF image:

$$ffHighFreq(x, y) = ffFullSlit(x, y)/fflowfreq(x, y) \tag{17}$$

   where the coordinates x', y' have been transformed back to the original x, y pixel coordinates.

For the sake of simplicity, we assume that the error introduced by the normalized FSFF is negligible for all practical purposes. Indeed, if the FSFF is an average of e.g. ten well exposed images, one may expect its SNR to be around 1000, while the SNR of individual scientific exposures will never exceed about 300 per pixel. So, in the worst case, the contribution of the normalized FSFF to the variance of a scientific image divided by it will remain below 10% (9% for SNR=300 in the initial science image).

3.15.9 Pseudo code

3.16 Measurement of the relative fiber transmissions

Name: **giFiberTrans** Section: **Flux calibration** nu: **117**
Author(s)/Date(s): **PN/20010117** Status: **To be validated by GS**

Working remarks:
Submitted to Internal review Wed 17/01/01.

3.16.1 Purpose

Estimates the relative fiber transmissions for the integrated light in a given set-up

3.16.2 Function parameters

Method to be used (requiring FSFF or not)

Name	Type	Default	Description
fibMethod	integer	1	Method to compute relative transmissions: 1=uses extracted FSFF spectra; 2=uses the { VignSpectro} table of spectrograph vignetting

3.16.3 Input frames

Name: *exNffMast* **Type:** EXTSP **Format:** nxExt
Description: extracted NFF spectra (Master NFF); It is assumed without bad pixels. This frame is set-up dependent and the selection is made on KW's SLIT:GRATING:WLEN0
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
EXTNX	DRS	integer	none	Number of pixel per spectrum
EXTNS	DRS	integer	none	Number of extracted spectra on the rebinned S2dFrame
SLIT	FGIR_ICS	integer	none	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies
GRATING	FGIR_ICS	string	none	DIC: ESO name for grating unit.
WLEN0	FGIR_ICS	double	nm	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength
REFSRCMULT	MISSING	float	normed	Factor which multiply the reference emissivity of the lamp to take into the account the setup-dependent fraction of the light injected into the fiber.

Name: *exFullSlit* **Type:** EXTSP **Format:** nxExt

Description: extracted FSFF spectra (with <{> extMethod}=1); It is assumed without bad pixels. This frame is set-up dependent and the selection is made on KW's SLIT:GRATING:WLEN0

Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
EXTNX	DRS	integer	none	Number of pixel per spectrum
EXTNS	DRS	integer	none	Number of extracted spectra on the rebinned S2dFrame
SLIT	FGIR_ICS	integer	none	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies
GRATING	FGIR_ICS	string	none	DIC: ESO name for grating unit.
WLEN0	FGIR_ICS	double	nm	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength
REFSRCMULT	MISSING	float	normed	Factor which multiply the reference emissivity of the lamp to take into the account the setup-dependent fraction of the light injected into the fiber.

3.16.4 Other inputs

Name	Type	Description
MeanNff	girMeanSpTable	current table of the wavelength averaged intensities of the NFF spectra
Relevant FITS keywords:	GIRTTYPE SLIT GRATING WLEN0	
Relevant table columns:		
NSPEC	integer	Running number of the spectrum
MEANSP	float	spectrum intensity averaged over the x direction
VignSpectro	girVignSpectroTable	vignetting effect of the spectrograph for each spectrum, integrated along the dispersion direction (one number lower or equal to 1, 1 for no vignetting)
Relevant FITS keywords:	GIRTTYPE SLIT GRATING WLEN0	
Relevant table columns:		
NSPEC	integer	Runnibg nuber of the spectrum
TRANSPEC	float	(relative) spectrograph transmission integrated on the x-direction

3.16.5 Output frames

none

3.16.6 Other outputs

Name	Type	Description
FiberTrans	girFiberTransTable	table of the relative fiber transmissions (normalized to the fiber with the maximum transmission)
Relevant FITS keywords:	GIRTTYPE SLIT GRATING WLEN0	
Relevant table columns:		
NSPEC	integer	Running numberof the spectrum
FIBTRAN	float	fiber transmission

3.16.7 General description

This function gives the relative overall transmission of the fibers in a given set-up (wavelength range). The purpose is not to get the detailed wavelength dependent transmission, but only the λ averaged transmission over

the whole wavelength range of the given set-up.
Notice that the calibration exposures are affected neither by vignetting effects due to the telescope, nor by transmission and pseudo-vignetting of the Nasmyth corrector, since the light does not go through these optical components. This simplifies the computation of the fiber transmissions.
Although the most direct way to measure the relative transmission would be, at first sight, to rely on the $\overline{exNffMast(n)}$ average intensities computed in the *giFlatSpectra*() function, such a solution is not straightforward because vignetting effects of the spectrograph will be mixed with the fiber transmission. In principle, one could disentangle the two effects (fiber transmission on the one hand, spectrograph vignetting and possibly large-scale sensitivity inhomogeneities of the CCD on the other hand) by using an FSFF frame taken with the same set-up. Indeed the FSFF image is sensitive to the spectrograph transmission and CCD efficiency exactly in the same way as the NFF one, except that the latter depends, in addition, on the fiber transmission. However, good cancellation of the spectrograph effects assumes a perfectly homogeneous slit illumination, while there is no explicit technical specifications as yet about it.
The only way to estimate the fiber transmission without resorting to any FSFF exposure but only to NFF one is to correct for modeled vignetting effects, which are hopefully realistic enough. Large scale CCD inhomogeneities can only be neglected in this option.
Two options are offered:

- The extracted NFF spectra are divided by FSFF spectra extracted according to `extMethod` = 1. Each resulting spectrum is then summed over all wavelengths, and the maximum sum is adopted as the normalizing value. Each sum is then divided by the maximum one, so that all relative fiber transmissions so obtained will be lower or equal to one.
- The mean values of the spectra, $\overline{exNffMast(n)}$, are taken from the *giFlatSpectra*() function[5], corrected for theoretical spectrograph vignetting effects and divided by their maximum value as above to yield relative transmission.

3.16.8 Mathematical description

Two options are offered (TBC), according to the parameter <`fibMethod`> :

1. **If `fibMethod` = 1, then:** the relative transmissions are computed according to the following scheme:

$$relIntens(n) = \frac{\sum_{i=1}^{EXTNX} exNffMast(x_i, n)}{\sum_{i=1}^{EXTNX} exFSFFMast(x_i, n)} \tag{18}$$

   where the $exFSFFMast(x, n)$ spectra have been extracted (after standard preprocessing of the FSFF image) through the function *giExtractSpectra*(), with the option `extMethod` = 1 and with `extWidthMult` = 1.0. The latter point ensures a homogeneous definition of the relative fiber transmissions: if the slit length used for extraction of the FSFF spectra is not the same for all measurements of the fiber transmission, slight differences might occur. Naturally, the FSFF image must have been taken with the same set-up as for the NFF image.

   Assuming that the slit illumination was perfectly homogeneous during the FSFF exposure, the relative fiber transmissions can be defined as:

$$\texttt{FiberTrans} : FIBTRAN(n) = \frac{relIntens(n)}{max(relIntens(n))} \tag{19}$$

   where $max(relIntens(n))$ is the maximum *relIntens* value, used as the normalization constant. In this way, all relative intensities will be smaller than or equal to one.

2. **If `fibMethod` = 2, then:** the relative transmissions are computed from quantities provided by the *giFlatSpectra*() function:

$$relIntens(n) = \frac{\texttt{MeanNff} : MEANSP(n)}{\texttt{VignSpectro} : TRANSPEC} \tag{20}$$

   where the `MeanNff` table comes from the *giFlatSpectra*() function and is the average of the NFF spectra taken over the x direction (it is identical with the $\overline{exNffMast(n)}$ defined in the *giFlatSpectra*() function). The `VignSpectro` table gives the modeled vignetting effect of the spectrograph integrated along each

[5]Note that this function will *always* be called in the normal reduction pipeline, even when the pixel-to-pixel sensitivity variations are corrected for *before* extraction

spectrum: there is one number per spectrum, expressed as the ratio of light intensity (integrated on all wavelengths of the relevant set-up) falling on the detector by the light intensity entering the spectrograph from the corresponding fiber. The normalization is then done in the same way as in the first option:

$$\texttt{FiberTrans} : FIBTRAN(n) = \frac{relIntens(n)}{max(relIntens(n))} \tag{21}$$

The results are stored in the `FiberTrans` table. No error estimates are made because the results will be essentially used for instrument monitoring purpose, and the systematic errors, which are unknown, are more likely to be predominent than the random ones.

3.16.9 Pseudo code

3.17 Wavelength Solution

Name: **giGetWaveSolution** Section: **Wavelength calibration** nu: **120**
Author(s)/Date(s): **PN/20000615** Status: **To be validated by GS**

Working remarks:
Submitted to Internal review Wed 09/06/00 including remarks that follow.

- *Point 6(b): When fitting the λ residuals, should we take into account the defocalization effect linked with the subslit positions? Does this defocalization has any effect on the wavelength solution? Maybe this issue could be tackled with using the simulator?*

  The answer by AB (on 29.08.2000) is that it is better to assume that the defocalization within subslits does indeed have a non-negligible effect on the wavelength solution.

- *Points 5(b) and 7: Does it really make sense to define as many images $fitWxl$ as there are width parameters to describe the PSF along the dispersion direction, or should we define only one (assuming that e.g. two width parameters are not independent so that some effective width make sense)?*

  The answer by AB and FR (on 29.08.2000) is that it is better to keep these informations. The problem comes at the level of giModelSky, where some PSF width (along the wavelength axis) has to be used as the independent variable when modeling the sky variation across the detector: a unique width must be defined there in a way TBD

3.17.1 Purpose

The wavelength solution is obtained from a separate (all fibres exposed to calibration lamp - SEWC) or simultaneous (5 fibres exposed to calibration lamp - SIWC) exposure

3.17.2 Function parameters

control parameters for line detection and for σ-clipping

Name	**Type**	**Default**	**Description**
`wsOptSol`	`logical`	`TRUE`	flag for fitting the optical parameters (wsOptSol=TRUE) or only the residuals relative to the standard optical model (wsOptSol=FALSE)
`wsLineType`	`logical`	`TRUE`	flag for use of ThArNe lines (wslineType=TRUE) or of telluric emission lines (lineType=FALSE)
`wsWindow`	`double`	`10.`	width in pixels of the window for line detection and fit
`wslineThres`	`float`	`10.`	multiple of σ above which a pixel value is considered to betray the presence of a calibration (emission) line
`wsClipSigma`	`float`	`3.0`	multiple of σ (σ-clipping)
`wsClipNiter`	`integer`	`10`	number of iterations (σ-clipping)
`wsClipMfrac`	`float`	`0.6`	min fraction of accepted/total lines (σ-clipping)
`widClipSigma`	`float`	`3.0`	multiple of σ above which clipping occurs (width in x)
`widClipNiter`	`integer`	`10`	number of σ iterations (width in x)
`widClipMfrac`	`float`	`0.7`	min fraction of accepted/total lines (width in x)
`wsXwidXDeg`	`integer`	`3`	polynomial degree along the x axis for x-width modeling
`wsXwidYDeg`	`integer`	`2`	polynomial degree along the y axis for x-width modeling
`wsWlenXDeg`	`integer`	`3`	polynomial degree along the x axis for wavelength solution
`wsWlenYDeg`	`integer`	`2`	polynomial degree along the y axis for wavelength solution

3.17.3 Input frames

Name: *exSp* **Type:** `EXTSP` **Format:** `nxExt`

Description: extracted spectra (either SEWC - full λ calibration or science exposure with SIWC; the distinction between SEWC and SIWC is made using the value of the KW DCATG=SCIENCE/CALIB and DTYPE=OTHER/LAMP)

Relevant FITS keywords:

Name	DIC	Type	Unit	Description
DCATG	DPR	string	none	DIC: Category according to section 4.7 of GEN-SPE-ESO-19400-0794. (science, calib, test, simulation, other)
DTYPE	DPR	string	none	DIC: Type of observation according to section 4.7 of GEN-SPE-ESO-19400-0794.
GIRFTYPE	DRS	string	none	type of frame data
NX	DRS	integer	none	Size of image in X direction (lambda)
NY	DRS	integer	none	Size of image in Y direction (slit)
STARTX	CCDDCS	integer	none	DIC: First window pixel in X direction within the detector physical system.
STARTY	CCDDCS	integer	none	DIC: First window pixel in Y direction within the detector physical system.
EXTNX	DRS	integer	none	Number of pixel per spectrum
EXTNS	DRS	integer	none	Number of extracted spectra on the rebinned S2dFrame
SLIT	FGIR_ICS	integer	none	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies
GRATING	FGIR_ICS	string	none	DIC: ESO name for grating unit.
GRATRES	FGIR_ICS	double	none	DIC: ENC = OFFST+ RESOL* RAD2DEG* arcsin(WLEN* GRV/(2* cos(ROT)))
GSPACE	FGIR_ICS	double	none	DIC: Number of grooves per nanometer ENC = OFFST+ RESOL* RAD2DEG* arcsin(WLEN* GRV/(2* cos(ROT)))
GRATORD	DRS	integer	none	Order of the grating in the current setup
GRATROT	FGIR_ICS	double *float	deg	DIC: Rotation angle of grating. The reference frame is given in the corresponding instrument specification. ENC = OFFST+ RESOL* RAD2DEG* arcsin(WLEN* GRV/(2* cos(ROT))) AL: Rotation angle of grating
FOCLEN	DRS	double	m	focal length of the collimator
CAMGFACT	DRS	float	none	magnification factor of the camera
WLEN0	FGIR_ICS	double	nm	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength
WLENMIN	DRS	float	nm	Shortest wavelength in the current setup
WLENMAX	DRS	float	nm	Longest wavelength in the current setup
LAMP	DRS	string	none	Name of the W Calibration source
LAMPID	FP_LP	undef	none	
SATLEVEL	DRS	integer	e-	CCD saturation level

Name: *exSpErr* **Type:** `EXTERRS` **Format:** `nxExt`

Description: corresponding extracted spectra errors

Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
EXTNX	DRS	integer	none	Number of pixel per spectrum
EXTNS	DRS	integer	none	Number of extracted spectra on the rebinned S2dFrame

Name: *locY* **Type:** `LOCY` **Format:** `nxExt`
Description: current localization giving the spectrum center along y at each x for the current setup.
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
EXTNX	DRS	integer	none	Number of pixel per spectrum
EXTNS	DRS	integer	none	Number of extracted spectra on the rebinned S2dFrame
SLIT	FGIR_ICS	integer	none	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies
GRATING	FGIR_ICS	string	none	DIC: ESO name for grating unit.
WLEN0	FGIR_ICS	double	nm	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength

3.17.4 Other inputs

The calibration spectra used here may be either flat-fielded ones (in the normal situations) or raw extracted spectra; the choice is made at the recipe level.

Name	Type	Description
`linesWlampMast`	`girLineTable`	table of $Nlines$ line wavelengths of the calibration spectrum, depends on SETUP
Relevant FITS keywords:	`GIRTTYPE LAMP LAMPID`	
Relevant table columns:		
WLEN	float	Wavelength of the line
FLUX	float	Expected central flux/[pixel.sec]
`linesSkyMast`	`girLineTable`	table of $Nlines$ line wavelengths of the telluric emission lines
Relevant FITS keywords:	`GIRTTYPE SLIT GRATING WLEN0`	
Relevant table columns:		
WLEN	float	Wavelength of the line
FLUX	float	Expected central flux/[pixel.sec]
`waveSolutionMast`	`girWavCoefTable`	table containing the standard $NparWv$ parameters of the analytical model of the wavelength solution for all spectra
Relevant FITS keywords:	`GIRTTYPE SLIT GRATING WLEN0`	
Relevant table columns:		
NSPEC	integer	Running number of the spectrum
XF	float	Fiber position in x on the slit
YF	float	Fiber position in y on the slit
CIJ	double	Coefficients of the wavelength solution (for the residuals relative to the optical model)
`psfXwidthMast`	`girPsfXwidTable`	standard $NparPSFw$ parameters of the adjusted analytical model of the PSF width in the x (λ) direction (in pixels)
Relevant FITS keywords:	`GIRTTYPE SLIT GRATING WLEN0`	
Relevant table columns:		
NSPEC	integer	running number of the spectrum
CWIJKM	double	coefficient of 2-D polynomials describing the x-width variation
`slitGeometryMast`	`girSlitGeoTable`	prameters used to compute the initial optical model
Relevant FITS keywords:	`GIRTTYPE SLIT`	
Relevant table columns:		
NSPEC	integer	running number of the spectrum/fibre
NSUBSLIT	integer	running number of the subslit
XF	float	Fiber position in x on the slit
YF	float	Fiber position in y on the slit

3.17.5 Output frames

there are npar frames named fitWx1, fitWx2,... fitWxnpar

Name: *fitWxl* **Type:** `FITCOEF` **Format:** `nxExt`

Description: frame of smooth fitted width parameters along the x axis

Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`GIRFTYPE`	DRS	`string`	none	type of frame data
`WSNS`	DRS	`integer`	pixel	X spectrum
`WSCOEFi`	DRS	`float`	none	PSF X-width coef
`WSLINETYPE`	DRS	`logical`	none	use of ThArNe or telluric lines for Wavelength Solution
`WSWINDOW`	DRS	`float`	pixel	width in pixel for line detection and fit
`WSCLIPMFRAC`	DRS	`float`	none	min fraction of points accepted/total (sigma-clipping)
`WSCLIPSIGMA`	DRS	`float`	none	multiple of sigma (sigma-clipping)
`WSCLIPNITER`	DRS	`integer`	none	number of iterations (sigma-clipping)
`WSXWIDXDEG`	DRS	`integer`	none	polynomial degree along the x direction for x-width modeling
`WSXWIDYDEG`	DRS	`integer`	none	polynomial degree along the y direction for x-width modeling
`WSWLENXDEG`	DRS	`integer`	none	polynomial degree along the x direction for wavelength solution
`WSWLENYDEG`	DRS	`integer`	none	polynomial degree along the y direction for wavelength solution

3.17.6 Other outputs

Name	Type	Description
waveSolution	girWavCoefTable	$NparWv$ parameters of the adjusted analytical model of the wavelength solution
Relevant FITS keywords:	GIRTTYPE WSCOEFi WSLINETYPE WSWINDOW WSCLIPMFRAC WSCLIPSIGMA WSCLIPNITER WSXWIDXDEG WSXWIDYDEG WSWLENXDEG WSWLENYDEG	
Relevant table columns:		
NSPEC	integer	Running number of the spectrum
XF	float	Fiber position in x on the slit
YF	float	Fiber position in y on the slit
CIJ	double	Coefficients of the wavelength solution (for the residuals relative to the optical model)
cleanlineTableMW	girLineTable	table of line wavelengths of the calibration spectrum, for the lines retained in the calibration process (must be a subset of lineTableModeWave)
Relevant FITS keywords:	GIRTTYPE WSCOEFi WSLINETYPE WSWINDOW WSCLIPMFRAC WSCLIPSIGMA WSCLIPNITER WSXWIDXDEG WSXWIDYDEG WSWLENXDEG WSWLENYDEG	
Relevant table columns:		
WLEN	float	Wavelength of the line
FLUX	float	Expected central flux/[pixel.sec]
cleanlineTableMWSky	girLineTable	table of line wavelengths of the sky spectrum, for the lines retained in the calibration process (must be a subset of lineTableMWSky)
Relevant FITS keywords:	GIRTTYPE WSCOEFi WSLINETYPE WSWINDOW WSCLIPMFRAC WSCLIPSIGMA WSCLIPNITER WSXWIDXDEG WSXWIDYDEG WSWLENXDEG WSWLENYDEG	
Relevant table columns:		
WLEN	float	Wavelength of the line
FLUX	float	Expected central flux/[pixel.sec]
psfXwidth	girPsfXwidTable	$NparPSFw$ parameters of the adjusted analytical model of the PSF width in the x direction (in pixels)
Relevant FITS keywords:	GIRTTYPE WSCOEFi WSLINETYPE WSWINDOW WSCLIPMFRAC WSCLIPSIGMA WSCLIPNITER WSXWIDXDEG WSXWIDYDEG WSWLENXDEG WSWLENYDEG	
Relevant table columns:		
NSPEC	integer	running number of the spectrum
CWIJKM	double	coefficient of 2-D polynomials describing the x-width variation

3.17.7 General description

The same processing is used for SEWC and SIWC. The FITS KWs of the input extracted spectra $extractSp(Ns, Nx)$ give the necessary information to distinguish between the two.

1. An initial solution, if no valid current solution exists, is obtained from the optical design.
2. The analytical model of the dispersion relation is used

   The `waveSolution` table is a set of parameters describing the analytical model representing the x position (in pixels) as a function of λ and n.

$x_{opt}(\lambda, n) =$

$$\frac{G\,f_{\rm coll}\left(\cos\theta\left(-\frac{\lambda\,m}{a}+\frac{xf(n)\,\cos\theta}{\mathcal{D}}+\frac{f_{\rm coll}\,\sin\theta}{\mathcal{D}}\right)+\sin\theta\sqrt{1-\frac{yf(n)^2}{\mathcal{D}^2}-\left(-\frac{\lambda\,m}{a}+\frac{xf(n)\,\cos\theta}{\mathcal{D}}+\frac{f_{\rm coll}\,\sin\theta}{\mathcal{D}}\right)^2}\right)}{-\left(\sin\theta\left(-\frac{\lambda\,m}{a}+\frac{xf(n)\,\cos\theta}{\mathcal{D}}+\frac{f_{\rm coll}\,\sin\theta}{\mathcal{D}}\right)\right)+\cos\theta\sqrt{1-\frac{yf(n)^2}{\mathcal{D}^2}-\left(-\frac{\lambda\,m}{a}+\frac{xf(n)\,\cos\theta}{\mathcal{D}}+\frac{f_{\rm coll}\,\sin\theta}{\mathcal{D}}\right)^2}}$$

where $f_{\rm coll}$ (FOCLEN) and G (CAMGFACT) are respectively the focal length of the collimator and the magnification factor of the camera. The $xf(n)$ and $yf(n)$ give the reference position of the fibre n on the entrance slit, while a (GSPACE) is the groove spacing, θ (GRATROT) is the grating angle and m (GRATORD) is the diffraction order; x is the pixel coordinate. All necessary optical parameters are provided via FITS KWs as given between parenthesis in the text above.

3. A list of lines (based on the FEROS line list, in the case of instrumental calibration) is associated to each source of the calibration spectra (thorium/argon/neon source, sky). We expect that after preliminary set-up, this list will be *cleaned* in order to avoid unnecessary systematic rejections during the step 4.

4. A line profile is fitted to each line and the polynomial expansion is adjusted to the differences between the found x positions and those obtained from the current model. The input model is updated. Though the full degree of the polynomial will always be used, the list of fitted coefficients will depend on the calibration spectra used (see below). The sigma clipping is implemented to eliminate a few lines which do not fit the model.

5. Optionally (technical set-up) the control of elimination/iteration will include access to all optical parameters in order to *tune* the process. We expect, that routinely the Wavelength Solution will be adjusted fully automatically.

3.17.8 Mathematical description

The following steps are written assuming the use of standard calibrations based on Th-Ar-Ne spectra (`wsLineType` = `TRUE`). However, the possibility is offered to use the telluric emission lines if they are numerous enough, through the <`wsLineType`> flag; the changes – if any – implied by this possibility are mentioned at the end of each point.

1. **Definition of the initial solution:**

   Read the `waveSolutionMast` table, which contains the coefficients of the standard wavelength solution; these coefficients are either recent ones (e.g. obtained from the five SIWC spectra of the previous science frame), or standard average ones. In both cases, a first hint of the wavelength solution is computed through the equation of the optical model:

$x_{opt}(\lambda, n) =$

$$\frac{G f_{\rm coll}\left(\cos\theta\left(-\frac{\lambda m}{a}+\frac{xf(n)\cos\theta}{\mathcal{D}}+\frac{f_{\rm coll}\sin\theta}{\mathcal{D}}\right)+\sin\theta\sqrt{1-\frac{yf(n)^2}{\mathcal{D}^2}-\left(-\frac{\lambda m}{a}+\frac{xf(n)\cos\theta}{\mathcal{D}}+\frac{f_{\rm coll}\sin\theta}{\mathcal{D}}\right)^2}\right)}{-\left(\sin\theta\left(-\frac{\lambda m}{a}+\frac{xf(n)\cos\theta}{\mathcal{D}}+\frac{f_{\rm coll}\sin\theta}{\mathcal{D}}\right)\right)+\cos\theta\sqrt{1-\frac{yf(n)^2}{\mathcal{D}^2}-\left(-\frac{\lambda m}{a}+\frac{xf(n)\cos\theta}{\mathcal{D}}+\frac{f_{\rm coll}\sin\theta}{\mathcal{D}}\right)^2}} \tag{22}$$

   where $\mathcal{D}^2 = xf^2 + yf^2 + f_{\rm coll}^2$ and where $f_{\rm coll} = exSp.$`FOCLEN`, $G = exSp.$`CAMGFACT`, $\theta = exSp.$`GRATROT`, $a = exSp.$`GSPACE` and $m = exSp.$`GRATORD` are the same $\forall n$, while $xf(n)$, $yf(n)$ are different for each spectrum([6]). The setup-dependent values of $f_{\rm coll}$, G, θ, a and m are given by FITS keywords on the input spectra $exSp$ as indicated between parenthesis in the text above. The values of xf and yf come from the `waveSolutionMast` table. $xf =$ `waveSolutionMast` $:XF$ and $yf =$ `waveSolutionMast` $:YF$

2. **Mask:** For each calibration spectrum, a mask is computed from the `linesWlampMast` table. The central x position of each line to be considered (i.e., *a priori*, each line of `linesWlampMast`) is computed from the preliminary wavelength solution above, and the pixels to be examined are defined as belonging to a range centered on this computed position and having a width <`wsWindow`> .

   If `wsLineType` = `FALSE`, the mask is computed from the `linesSkyMast` .

3. **Line search:** In each range, the pixels are sorted by order of intensities. The x_{max} position of the pixel with the largest intensity I_{max} is retained as a first guess of the position of the gaussian to be fitted. The average of the two lowest pixel values $fback$ is retained as a first guess of the background. The inverse contrast r is computed as the ratio of the background error (which is computed in the course of the extraction process) and of the largest intensity, and the line is rejected if $r >$ `wsLineThres`$^{-1}$.

   If $I_{max} \geq exSp.$`SATLEVEL`, the line is rejected and the list of rejected lines is updated.

[6]This equation is derived from a crude optical model; it will be replaced by a more sophisticated one as soon as a more realistic model is available

4. **Line fitting:** A PSF function TBD is fitted to each retained line, using x_{max}, $fback$, $\Delta x = x_{opt}(\lambda + 0.5\, exSp.\texttt{WLEN0}/exSp.\texttt{GRATRES}) - x_{opt}(\lambda - 0.5\, exSp.\texttt{WLEN0}/exSp.\texttt{GRATRES})$ and I_{max} as first guesses for the position, background, FWHM width[7] and amplitude respectively. The formal error $\sigma_w(l)$ (with $l = 1, ...npar$) on the width is provided by the fit. ($npar$ width parameters may be needed to adequately describe the PSF, in which case the ratios between them will be assumed constant for the first guess, so that a first guess value on one parameter will be sufficient). Likewise, the formal error σ_p on the line position is provided by the fit and will be used to compute the weights in the fit of the wavelength solution.

5. **Polynomial fit of the PSF width variation:** Because of the mechanical disposition of the subslits, it is impossible to model the PSF width with a single 2-D function; at best, we can only fit Nss 2-D functions, Nss being the number of subslits, and this will only be possible with the full SEWC calibration. In the SIWC case, five independent 1-D functions will have to be adjusted.

   (a) **LSQ fit of the PSF width, case of SEWC:** The widths will be LSQ fitted with Chebyshev polynomials, after normalization of the CCD pixel coordinates to the interval [-1,+1]:

$$widthx(x', n_k, l) = \sum_{i=0}^{I} \sum_{j=0}^{J} cw_{i,j,k,l} T_i(x') T_j(y'(n_k)), \quad l = 1, ...npar \tag{23}$$

   where the subslit number $k = 1, ...13$ in the MEDUSA mode and $k = 1, ...15$ in the IFU/ARGUS modes, $x'_k = \frac{2}{\texttt{NX}-1}(x_k - STARTX - \frac{\texttt{NX}-1}{2})$, $I \equiv$ `wsXwidXDeg`, $J \equiv$ `wsXwidYDeg`, and T_i are Chebyshev polynomials of degree i. Therefore, there will be $13 \times npar$ such fits for the MEDUSA mode and $15 \times npar$ for the ARGUS/IFU modes. The information on the y coordinate cannot be replaced by the spectrum number n because even within a given subslit, the function $y(n)$ is not always continuous. The y coordinate has to be normalized separately for each subslit k, so that $[y_{min}(n_k^{min}), y_{max}(n_k^{max})] \rightarrow [-1, +1]$, where n_k^{min} and n_k^{max} are the minimum and maximum spectrum numbers within the subslit k:

$$y'_k = \frac{2}{locY_{max}(n_k^{max}) - locY_{min}(n_k^{min})} \left(locY(n_k) - \frac{locY_{min}(n_k^{min}) + locY_{max}(n_k^{max})}{2} \right) \tag{24}$$

   The LSQ fit is done by weighting the measured widths with the inverse of the variance:

$$weightw(x, n_k, l) = 1/\sigma_w(x, n_k, l)^2$$

   where $\sigma_w(x, n_k, l)$ was estimated for each line through the LSQ fit of the profile.

   The lines whose widths differ by more than `widClipSigma` $\times\, \sigma_w(l = 1)$ from the fitted function are discarded (where σ_w was estimated through the fit of the line profile), the list of rejected lines is updated and the fit is re-iterated until no more rejection occurs (a test on only the first width parameter is considered sufficient) or until the number of iterations reaches `widClipNiter` or until the fraction of accepted lines becomes lower than `widClipMfrac`. The number `widClipNiter` of iterations and the number of rejected lines `widClipMfrac` are updated.

   This process generates an estimate of the PSF width as a function of position on the CCD, but it is expressed in pixels while an estimate in wavelength is needed for sky subtraction. For that, one needs a linear transformation using the wavelength solution (see point 7).

   (b) **LSQ fit of the PSF width, case of SIWC:** Since there are only 5 spectra, each one belonging to another subslit, one can only fit the coefficients describing the variation of the width along the x axis, and consider the other ones as reliable. More explicitly, Equ. 23 can be written

$$widthx(x', n_k, l) = \sum_{i=0}^{I} cw_{i,0,k,l} T_i(x') + \sum_{i=0}^{I} \sum_{j=1}^{J} cw_{i,j,k,l} T_i(x') T_j(y'(n_k)), \quad l = 1, ...npar \tag{25}$$

   and only the coefficients $cw_{i,0,k,l}$ in the first sum will be adjusted. Those in the second sum will come from a previous SEWC calibration and will remain fixed in the LSQ fit. The same weights will be used in the LSQ fit as in the previous case.

[7] for a gaussian, one would rather consider the quantity $\sigma = FWHM/2.355$

If `wsLineType` = `FALSE` (calibration using telluric lines), the situation is similar to that of SIWC, because the number of sky spectra per subslit will be too small in general to allow any reliable fit of the y dependence of the PSF width. In the MEDUSA mode, the number of sky spectra is not defined *a priori* and it can be very large, but in general it will hardly be larger than 1 or 2 per subslit. In the ARGUS/IFU modes, there is one sky fibre per subslit.

Finally, the fitted coefficients are stored in the table `psfXwidth` and the width parameters for every x pixel and every spectrum are stored in $npar$ images $fitWxl$ ($l = 1, ...npar$). For instance, if the PSF can be represented by a single gaussian of width $\sigma(x, n)$, there will be only one image in the frame $fitWxl$, where the value of each pixel is set to $\sigma(x, n)$. If the PSF has to be modeled with more than one parameter and is e.g. a superposition of two gaussians of widths σ_1, σ_2 and of amplitudes A_1,A_2, there will be at least three images whose pixel values will represent respectively σ_1, σ_2 and A_2/A_1. But it may be yet other quantities if the PSF is modeled by another type of function.

6. **Fit of the coefficients of the wavelength solution (SEWC and SIWC):** The fit is done in two stages, the first of which being carried out only with SEWC spectra, either when one wants to fit the wavelength solution from scratch or update it using a SEWC calibration:

   (a) If `wsOptSol` = *true*, an iterative non-linear LSQ fit is done on Equation 22 above. The 4 parameters common to all spectra are set to their specification values or set-up keywords as first guesses, (a fifth parameter, the diffraction order m, does not need any adjustment). The two parameters which are specific to each spectrum are also set to their specifications values ($xf(n)$, $yf(n)$). In the LSQ fit, the measured line positions are weighted by the inverse of their variance estimated previously, during the line fitting:

   $$weightp = 1/\sigma_p(x, n_k)^2$$

   The fitted coefficients are stored in 4 KW which are updated ($exSp$.`FOCLEN`, $exSp$.`CAMGFACT`, $exSp$.`GRATROT`, $exSp$.`GSPACE`), and in two columns of the `WaveSolution` table (`WaveSolution` :XF, `WaveSolution` :YF).

   (b) The residuals $x - x_{opt}(\lambda_{\rm lab}, n)$ are computed and modelized with Chebyshev polynomials:

   - **SEWC spectra:** It is assumed *a priori* that defocalization of extreme fibers of a given subslit may have a non-negligible effect on the wavelength solution, so that only the first approximation of it – obtained through the optical model – is expressed as a single formula. Unfortunately, the fit to the residuals from this first guess must be done independently for each subslit, so the final wavelength solution can only be expressed as a set of either 13 or 15 formulae, whether we are in the MEDUSA or ARGUS/IFU mode respectively.

     $$x(\lambda', n_k) - x_{opt}(\lambda', n_k) = \sum_{i=0}^{\texttt{wsWlenXDeg}} \sum_{j=0}^{\texttt{wsWlenYDeg}} c_{i,j,k} T_i(\lambda') T_j(y'(n_k))$$

     where λ' and y' are normalized quantities as in 23, while $k = 1, ...13$ in the MEDUSA mode and $k = 1, ...15$ in the IFU/ARGUS mode. As in the case of the fit to the optical model, this fit involves the same weights $weightp = 1/\sigma_p^2$. The lines farther away than `wsClipSigma` $\times\ \sigma_p$ from the fitted solution are discarded (σ_p is the error on the line position estimated from the fit of the profile), the list of discarded lines is updated and the fit iterated until no more outlier is detected, or until the number of iterations or the fraction of rejected lines is reached. The number of iterations <`wsClipNiter`> is updated as well as the number of rejected lines <`wsClipMfrac`> .

     An update of the current wavelength solution using SEWC spectra will always be done relative to the last fit of the optical model, so that the $c_{i,j,k}$ coefficients above are determined anew, rather than by fitting the difference between the measured positions and those given by the current solution. In this way the $c_{i,j,k}$ coefficients have always the same physical meaning, while otherwise x_{opt} would be replaced by $x_{currentsol.}$ and the coefficients would mean something different. Of course, an even more satisfactory update would include a new determination of the optical model. Note that the y variable is used here rather than the spectrum number n, because the $y(n)$ function is not continuous (gaps between the subslits). This is why $locY$ is needed.

   - **SIWC spectra:** In case of SIWC spectra, all coefficients describing the y contribution to $x(\lambda', n)$ will be considered constant in the LSQ fit, since only 5 spectra are available, less than one per subslit. The fixed coefficients are those of the current solution, which was obtained from a SEWC calibration, and new residuals are computed relative to the current optical solution, by adjusting only the $c_{i,0,k}$ coefficients of the x contribution to the wavelength solution, still using the weights $weightp$:

$$x(\lambda', n_k) - x_{opt}(\lambda', n_k) = \sum_{i=0}^{wsWlenXDeg} c_{i,0,k} T_i(x') + \sum_{i=0}^{\texttt{wsWlenXDeg}} \sum_{j=1}^{\texttt{wsWlenYDeg}} c_{i,j,k} T_i(\lambda') T_j(y'(n_k))$$

Finally, the fitted $c_{i,0,k}$ coefficients are stored in additional columns of the `WaveSolution` table (`WaveSolution` :CIJ).

7. **PSF x-width expressed in wavelength:** The width expressed in wavelength can be written as the width in pixels multiplied by the derivative of the wavelength solution:

$$widthl(\lambda, n, l) = \frac{d\lambda}{dx} \times widthx(x, n, l), \qquad l = 1, npar$$

In practice, $\frac{d\lambda}{dx}$ will be computed numerically from the wavelength solution while $widthx(x, n, l)$ will be taken from the frames $fitWxl$ (with $l = 1, ...npar$), but only in the function $giRebinSpectra()$.

3.17.9 Pseudo code

a faire...

3.18 Rebinning in wavelength space

Name: **giRebinSpectra** Section: **Wavelength calibration** nu: **130**
Author(s)/Date(s): **FR/20000609** Status:

Working remarks:
Review remarks included. To be validated (18 Sep 2000)

3.18.1 Purpose

Carry out rebinning of extracted spectra so that the result is equally spaced in λ

3.18.2 Function parameters

Name	Type	Default	Description
binWStep	double	undef	wavelength step used in rebinning (= 0, if rebinning is carried out in logarithmic wavelength space)
binWRange	integer	undef	flag indicating how is derived the spectral range that is rebinned
binWStepLog	double	undef	logarithmic wavelength step used in rebinning (= 0, if rebinning is carried out in wavelength space)

3.18.3 Input frames

Name: *ffSp* **Type:** EXTSP **Format:** nxExt
Description: flat-fielded spectra modulo (divided by) LAMP reference spectrum
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
EXTNX	DRS	integer	none	Number of pixel per spectrum
EXTNS	DRS	integer	none	Number of extracted spectra on the rebinned S2dFrame
SLIT	FGIR_ICS	integer	none	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies
GRATING	FGIR_ICS	string	none	DIC: ESO name for grating unit.
WLEN0	FGIR_ICS	double	nm	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength
BINWNX	DRS	integer	none	number of spectral elements on the rebinned S2dFrame
WLENMIN	DRS	float	nm	Shortest wavelength in the current setup
WLENMAX	DRS	float	nm	Longest wavelength in the current setup
FOCLEN	DRS	double	m	focal length of the collimator
CAMGFACT	DRS	float	none	magnification factor of the camera
GRATROT	FGIR_ICS	double *float	deg	DIC: Rotation angle of grating. The reference frame is given in the corresponding instrument specification. ENC = OFFST+ RESOL* RAD2DEG* arcsin(WLEN* GRV/(2* cos(ROT))) AL: Rotation angle of grating

Name: *ffSpErr* **Type:** EXTERRS **Format:** nxExt
Description: corresponding frame of Flat-field corrected extracted spectra errors
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
EXTNX	DRS	integer	none	Number of pixel per spectrum
EXTNS	DRS	integer	none	Number of extracted spectra on the rebinned S2dFrame

Name: *fitWxl* **Type:** FITCOEF **Format:** nxExt
Description: frame of smooth fitted width parameters along the x axis
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
WSNS	DRS	integer	pixel	X spectrum
WSCOEFi	DRS	float	none	PSF X-width coef
WSLINETYPE	DRS	logical	none	use of ThArNe or telluric lines for Wavelength Solution
WSWINDOW	DRS	float	pixel	width in pixel for line detection and fit

3.18.4 Other inputs

Name	Type	Description
waveSolution	girWavCoefTable	parameters of the adjusted analytical model of the wavelength solution
Relevant FITS keywords:	GIRTTYPE SLIT GRATING WLENO FOCLEN CAMGFACT GRATROT	
Relevant table columns:		
NSPEC	integer	Running number of the spectrum
XF	float	Fiber position in x on the slit
YF	float	Fiber position in y on the slit
CIJ	double	Coefficients of the wavelength solution (for the residuals relative to the optical model)

3.18.5 Output frames

Name: *wlFfSp* **Type:** BINSP **Format:** nlWbin
Description: frame of rebinned spectra
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
BINWNX	DRS	integer	none	number of spectral elements on the rebinned S2dFrame
BINWNY	DRS	integer	none	number of spectra on the rebinned S2dFrame
BINWSTEP	DRS	double	nm	wavelength step of the linear rebinned spectra
BINWL0	DRS	float	nm	wavelength of the centre of the first spectral elements
BINWSTEPLOG	DRS	double	none	wavelength step of the logarithmic rebinned spectra; the spectral element N has a central wavelength=w0+10**((N-1)*step)

Name: *wlFfSpErr* **Type:** BINERRS **Format:** nlWbin
Description: frame of rebinned spectra errors
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
BINWNX	DRS	integer	none	number of spectral elements on the rebinned S2dFrame
BINWNY	DRS	integer	none	number of spectra on the rebinned S2dFrame
BINWSTEP	DRS	double	nm	wavelength step of the linear rebinned spectra
BINWL0	DRS	float	nm	wavelength of the centre of the first spectral elements
BINWSTEPLOG	DRS	double	none	wavelength step of the logarithmic rebinned spectra; the spectral element N has a central wavelength=w0+10**((N-1)*step)

Name: *psflwidth* **Type:** `BINSP` **Format:** `nlWbin`
Description: Width parameter for 1-D PSF along the dispersion direction (x)
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`GIRFTYPE`	DRS	`string`	none	type of frame data
`BINWNX`	DRS	`integer`	none	number of spectral elements on the rebinned S2dFrame
`BINWNY`	DRS	`integer`	none	number of spectra on the rebinned S2dFrame
`BINWSTEP`	DRS	`double`	nm	wavelength step of the linear rebinned spectra
`BINWLO`	DRS	`float`	nm	wavelength of the centre of the first spectral elements
`BINWSTEPLOG`	DRS	`double`	none	wavelength step of the logarithmic rebinned spectra; the spectral element N has a central wavelength=w0+10**((N-1)*step)

3.18.6 Other outputs

none

3.18.7 General description

The rebinning is done to obtain the $wlFfSp(Nlambda, Ns)$ spectra. The extracted spectra $ffSp(Nx, Ns)$ and associated information (error, number of points used) are used.
In order to get pixel positions x as a function of λ, the expression of the wavelength solution, via coefficients of the `waveSolution` table, is used.
For homogenization's sake, the spectral domain where spectra are rebinned is taken to be the same for each spectrum of the frame, according to <`binWRange`> :

- `binWRange` $= 0$: the spectral range is chosen as defined by the keywords $ffSp$.`WLENMIN` and $ffSp$.`WLENMAX` corresponding to the observed mode,
- `binWRange` $= 1$: the spectral range is taken as the common interval $[\lambda_1, \lambda_2]$ to all spectra of the frame. The wavelength limits $\lambda_{\min}(n)$ $\lambda_{\max}(n)$ are computed by fitting spline on series of points $(\lambda_i, x(\lambda_i, n))$ around the defined limits of the bandpass, for each spectrum. Then $\lambda_1 = \max_{n\in[1,Ns]} \left\lceil \frac{\lambda_{\min}(n) - ffSp.\texttt{WLENMIN}}{\texttt{binWStep}} \right\rceil \times \texttt{binWStep} + ffSp.\texttt{WLENMIN}$ and $\lambda_2 = \min_{n\in[1,Ns]} \left\lfloor \frac{\lambda_{\max}(n) - ffSp.\texttt{WLENMAX}}{\texttt{binWStep}} \right\rfloor \times \texttt{binWStep} + ffSp.\texttt{WLENMAX}$. The limits are shifted to wedge the general wavelength scale (defined by $ffSp$.`WLENMIN` and $ffSp$.`WLENMAX`).

If the parameter <`binWStep`> is set to 0, then the rebinning is carried out in the $\log\lambda$ space using the step parameter <`binWStepLog`> expressed in logarithm of a wavelength in nm. Limits of the spectral domain are chosen as above but expressed in logarithm of wavelength (λ in nm) and using <`binWStepLog`> instead of <`binWStep`> .

3.18.8 Mathematical description

The new sampling in wavelength space is defined by the chosen extrema and the spacing <`binWStep`> .

- `binWRange` $= 0$: $ffSp$.`BINWLO` $= ffSp$.`WLENMIN` and $ffSp$.`BINWNX` $= \left\lceil \frac{ffSp.\texttt{WLENMAX} - ffSp.\texttt{WLENMIN}}{\texttt{binWStep}} \right\rceil + 1$
- `binWRange` $= 1$: $ffSp$.`BINWLO` $= \lambda_1$ and $ffSp$.`BINWNX` $= \left\lceil \frac{\lambda_2 - \lambda_1}{\texttt{binWStep}} \right\rceil + 1$

The corresponding positions in pixel-space are derived from the analytical model of the wavelength solution stored in `waveSolution` and for each point of the new sampling, the suitable part of the spectrum is integrated to derive the associated flux (see Fig. 4):

$$wlFfSp(\lambda_j, n) = \varepsilon_i \times ffSp(x_i, n) + \varepsilon_{i+1} \times ffSp(x_i + 1, n)$$

where ε_i is the fraction of points of the pixel x_i encroached by the pixel in wavelength-space λ_j. The errors of the rebinned spectra are computed from the errors of the input spectra and their associated number of points. Knowing that the rebinned spectra are oversampled, the error of a pixel is the quadratic sum of the errors of the

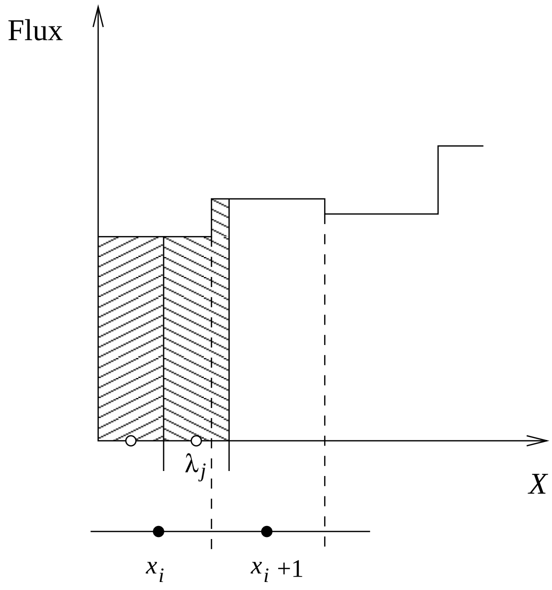


Figure 4: Rebinning in wavelength space

(maximum) 2 input pixels it encroaches, weighted by the corresponding fractions of points of each input pixels falling in the output pixel:

$$wlFfSpError(\lambda_j, n) = \left(\varepsilon_i^2 \times ffSpError(x_i, n)^2 + \varepsilon_{i+1}^2 \times ffSpError(x_i + 1, n)^2\right)^{1/2}$$

When rebinning in logarithmic space (`binWStep` = 0), the spectra are processed in the same way and the corresponding positions in pixel-space of the rebinned pixels in $\log \lambda$ are also derived from the table `waveSolution` .

PSF x-width expressed in wavelength: the frame *psflwidth* expressing the width of the PSF in the direction of the dispersion for each wavelength bin in constructed using the formula:

$$widthl(\lambda, n, l) = \frac{widthx(x(\lambda), n, l)}{\frac{dx}{d\lambda}(\lambda)} \qquad l = 1, npar$$

where $widthx(x(\lambda), n, l)$ is interpolated between the $fitWxl(x, n)$ for the corresponding l , $\frac{dx}{d\lambda}(\lambda)$ is the numerically computed derivative in λ of the wavelength solution stored in `waveSolution` . The frame *psflwidth* is a linear combination (TBD, depending on the shape of the PSF) of the *npar widthl*.

3.18.9 Pseudo code

3.19 Vignetting correction

Name: **giVignCorr** Section: **Flux calibration** nu: **140**
Author(s)/Date(s): **PN/20000713** Status: **To be validated by GS**

Working remarks:
Submitted to Internal rewiev Wed 09/06/00. The answer to our report sent to ESO is pending. Coarse estimate will be given by DRS. The ESO should provide the model and Luca Pasquini precise the calibration plan.

3.19.1 Purpose

The extracted spectra are corrected for vignetting effects linked to both the VLT M2 and M3 mirrors, and to the corrector chromatic aberration

3.19.2 Function parameters

none

3.19.3 Input frames

Name: *wlFfSp* **Type:** BINSP **Format:** nlWbin
Description: frame of rebinned spectra
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
BINWNX	DRS	integer	none	number of spectral elements on the rebinned S2dFrame
BINWNY	DRS	integer	none	number of spectra on the rebinned S2dFrame
BINWSTEP	DRS	double	nm	wavelength step of the linear rebinned spectra
BINWLO	DRS	float	nm	wavelength of the centre of the first spectral elements
DCATG	DPR	string	none	DIC: Category according to section 4.7 of GEN-SPE-ESO-19400-0794. (science, calib, test, simulation, other)
LAMP	DRS	string	none	Name of the W Calibration source
LAMPID	FP_LP	undef	none	
SLIT	FGIR_ICS	integer	none	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies
GRATING	FGIR_ICS	string	none	DIC: ESO name for grating unit.
WLEN0	FGIR_ICS	double	nm	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength

Name: *wlFfSpErr* **Type:** BINERRS **Format:** nlWbin
Description: frame of rebinned spectra errors
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
BINWNX	DRS	integer	none	number of spectral elements on the rebinned S2dFrame
BINWNY	DRS	integer	none	number of spectra on the rebinned S2dFrame
BINWSTEP	DRS	double	nm	wavelength step of the linear rebinned spectra
BINWLO	DRS	float	nm	wavelength of the centre of the first spectral elements

3.19.4 Other inputs

Name	Type	Description
vignCorrMast	girVignCorrTable	Vignetting correction, table selected according to DCAT and we check that it was made for the same lamp
Relevant FITS keywords:	DCATG LAMP LAMPID	
Relevant table columns:		
LAMBDA	float	wavelength
R	float	distance in the Nasmyth focal plane from optical axis
VIGNMEDUSA	float	fraction of the light compared to what we would have at R=0 position
VIGNIFU	float	fraction of the light compared to what we would have at R=0 position

3.19.5 Output frames

Name: *wlFfVcSp* **Type:** BINSP **Format:** nlWbin
Description: frame of rebinned spectra corrected for vignetting
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
BINWNX	DRS	integer	none	number of spectral elements on the rebinned S2dFrame
BINWNY	DRS	integer	none	number of spectra on the rebinned S2dFrame
BINWSTEP	DRS	double	nm	wavelength step of the linear rebinned spectra
BINWLO	DRS	float	nm	wavelength of the centre of the first spectral elements

Name: *wlFfVcSpErr* **Type:** BINERRS **Format:** nlWbin
Description: frame of rebinned spectra errors corrected for vignetting
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
BINWNX	DRS	integer	none	number of spectral elements on the rebinned S2dFrame
BINWNY	DRS	integer	none	number of spectra on the rebinned S2dFrame
BINWSTEP	DRS	double	nm	wavelength step of the linear rebinned spectra
BINWLO	DRS	float	nm	wavelength of the centre of the first spectral elements

3.19.6 Other outputs

none

3.19.7 General description

The extracted science spectra are corrected for the vignetting effect due to the M2 and M3 mirrors in the VLT, and to another vignetting effect occurring at the level of the fibre entrance, due to aberrations introduced by the Nasmyth corrector. The first effect is not chromatic but the second one is. The preliminary detailed analysis is given in RD11.

These effects become significant for objects several arcminutes away from the optical axis, so they need to be corrected only in the MEDUSA and IFU modes, not in the ARGUS mode.

Both effects should vary only as a function of R, the radial distance from the FoV centre. They can amount up to 10 percent near the edge of the FoV. They are modelized by an analytic transmission function of R yet TBD.

3.19.8 Mathematical description

The extracted, rebinned and flatfielded science spectra are corrected for both vignetting effects through the formula

$$lFfVcSp(\lambda, n) = \frac{wlFfSp(\lambda, n)}{\texttt{vignCorrMast}:VIGNMEDUSA(R, LAMBDA)}$$

for the MEDUSA mode, and through the formula

$$lFfVcSp(\lambda, n) = \frac{wlFfSp(\lambda, n)}{\texttt{vignCorrMast}:VIGNIFU(R, LAMBDA)}$$

for the IFU mode, where the table `vignCorrMast` gives the ratio of the light transmitted for an object lying at distance R from the telescope's optical axis, and of the light transmitted for the same object on the optical axis, including the effect of the Nasmyth corrector. The table `vignCorrMast` combines two effects:

- The achromatic vignetting due to the mirrors M2 and M3 of the VLT, which is relevant only to the sciences spectra
- The pseudo-vignetting due to the aberrations of the Nasmyth corrector, i.e. the fraction of the light (i.e. of the telescope pupil imaged by the microlens) entering the fibre, and which is relevant to both science spectra and calibration spectra taken with the Nasmyth screen. On the optical axis, this fraction is unity in the IFU mode, but slightly smaller than one in the MEDUSA mode because the fibre diameter is smaller than the pupil diameter; off axis, the pupil displacement due to the Nasmyth corrector may further decrease this fraction (this depends on the way in which the VLT vignetting occurs). Note that the transmission of the corrector proper (without the vignetting effects due to the imperfect telecentricity correction) is not corrected for here, but only in the `giFluxConv` function.

Although a preliminary study (RD11) was done, the exact analytical form giving `vignCorrMast` $:VIGNMEDUSA(R, LAMBD$ and `vignCorrMast` $:VIGNIFU(R, LAMBDA)$ is yet TBD. The contribution of the corrector to the whole "vignetting" effect depends in fact from that of the VLT: in particular, the former seems to vanish in the IFU mode because of the pupil truncation linked with the VLT vignetting. The whole effect might probably be represented by a polynomial function of R and λ, but for the time being we have chosen to interpolate in a table.
The errors will be amplified by the same factor:

$$wlFfVcSpError(\lambda, n) = \frac{wlFfSpError(\lambda, n)}{\texttt{vignCorrMast}:VIGNMEDUSA(R, LAMBDA)}$$

or

$$wlFfVcSpError(\lambda, n) = \frac{wlFfSpError(\lambda, n)}{\texttt{vignCorrMast}:VIGNIFU(R, LAMBDA)}$$

This correction should, in general, **not** be applied to calibration spectra, since the light does not go through the telescope nor the corrector. The only exception is calibration spectra taken with the Nasmyth screen: in this case, the light goes through the corrector **only**, and the specific table `vignCorrMast` is different. The choice of the right `vignCorrMast` table is made through the keyword $wlFfSp$.`DCATG`.

3.19.9 Pseudo code

a faire...

3.20 Separation of the sky spectra into two independent components

Name: **giSplitSky** Section: **Sky subtraction** nu: **145**
Author(s)/Date(s): **PN/20000906** Status: **To be validated by GS**

Working remarks:
Added to allow to tackle a 2-component sky

3.20.1 Purpose

The sky spectra are separated into two components

3.20.2 Function parameters

control parameter to define the minimum width of the telluric emission lines for a proper interpolation

Name	Type	Default	Description
ssWidMul	float	6.0	minimum full width of sky emission lines in terms of the PSF width parameter in wavelength (multiple of sigma in case of a gaussian PSF)

3.20.3 Input frames

The input frame contains all spectra corrected for instrumental response and rebinned in wavelength space.

Name: *wlFfSp* **Type:** BINSP **Format:** nlWbin
Description: any rebinned spectra frame after correction for instrumental response and rebinning in the wavelength space.
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
BINWNX	DRS	integer	none	number of spectral elements on the rebinned S2dFrame
BINWNY	DRS	integer	none	number of spectra on the rebinned S2dFrame
BINWSTEP	DRS	double	nm	wavelength step of the linear rebinned spectra
BINWLO	DRS	float	nm	wavelength of the centre of the first spectral elements
SLIT	FGIR_ICS	integer	none	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies
GRATING	FGIR_ICS	string	none	DIC: ESO name for grating unit.
WLEN0	FGIR_ICS	double	nm	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength
EXPTIME	UVES_ICS	double	sec	Total exposure time for a single exposure or for the individual exposure within a multiple frame

Name: *wlFfSpErr* **Type:** BINERRS **Format:** nlWbin
Description: frame of rebinned spectra errors
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
BINWNX	DRS	integer	none	number of spectral elements on the rebinned S2dFrame
BINWNY	DRS	integer	none	number of spectra on the rebinned S2dFrame
BINWSTEP	DRS	double	nm	wavelength step of the linear rebinned spectra
BINWLO	DRS	float	nm	wavelength of the centre of the first spectral elements

Name: *psflwidth* **Type:** `BINSP` **Format:** `nlWbin`
Description: width parameter for 1-D PSF along the dispersion direction (x)
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
BINWNX	DRS	integer	none	number of spectral elements on the rebinned S2dFrame
BINWNY	DRS	integer	none	number of spectra on the rebinned S2dFrame
BINWSTEP	DRS	double	nm	wavelength step of the linear rebinned spectra
BINWL0	DRS	float	nm	wavelength of the centre of the first spectral elements

3.20.4 Other inputs

Name	Type	Description
ozpoz	girOzpozTable	Positionner Binary Table; this is in fact not an input, but part of the inframe1
Relevant FITS keywords:	none (attached to frame header)	
Relevant table columns:		
OBJECT_TYPE	character	object type: P=programme object, S=sky, G=guide star, X=simulation, F=flux refrence star (request OGL/OP)
SPECT_FIBRE	integer	fibre number on the spectrograph slit
numSky	girSkyTable	optional (off-line mode only) single column table with the $Nsky$ fibre running numbers of spectra to be normalized.
Relevant FITS keywords:	GIRTTYPE SLIT GRATING WLEN0	
Relevant table columns:		
NSPEC	integer	Running number of the spectrum
linesSkyMast	girLineTable	table of $Nlines$ line wavelengths of the telluric emission lines
Relevant FITS keywords:	GIRTTYPE SLIT GRATING WLEN0	
Relevant table columns:		
WLEN	float	Wavelength of the line
FLUX	float	Expected central flux/[pixel.sec]

3.20.5 Output frames

This function produces, in the red, two s2d images contained in one frame, each image containing all spectra, but only one component of the sky (continuum or emission lines)

Name: *wlFfSkyComp* **Type:** `BINSP` **Format:** `nlWbin`
Description: frame of all rebinned spectra, containing two images for the two sky components; each image of this frame contains one component only in the sky spectra
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
BINWNX	DRS	integer	none	number of spectral elements on the rebinned S2dFrame
BINWNY	DRS	integer	none	number of spectra on the rebinned S2dFrame
BINWSTEP	DRS	double	nm	wavelength step of the linear rebinned spectra
BINWL0	DRS	float	nm	wavelength of the centre of the first spectral elements
SLIT	FGIR_ICS	integer	none	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies
GRATING	FGIR_ICS	string	none	DIC: ESO name for grating unit.
WLEN0	FGIR_ICS	double	nm	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength
SKCOMP	DRS	string	none	sky component: one of 'EMISSION' or 'CONTINUUM'

Name: *wlFfSkyCompErr* **Type:** `BINSP` **Format:** `nlWbin`
Description: frame of all rebinned spectra errors, containing two images for the two sky components; each image of this frame contains one component only in the sky errors
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
BINWNX	DRS	integer	none	number of spectral elements on the rebinned S2dFrame
BINWNY	DRS	integer	none	number of spectra on the rebinned S2dFrame
BINWSTEP	DRS	double	nm	wavelength step of the linear rebinned spectra
BINWL0	DRS	float	nm	wavelength of the centre of the first spectral elements
SLIT	FGIR_ICS	integer	none	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies
GRATING	FGIR_ICS	string	none	DIC: ESO name for grating unit.
WLEN0	FGIR_ICS	double	nm	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength
SKCOMP	DRS	string	none	sky component: one of 'EMISSION' or 'CONTINUUM'

3.20.6 Other outputs

none

3.20.7 General description

This function splits each sky spectrum into two components:

1. A "continuum" component, which should be understood as that part of the sky unrelated with telluric emission lines (zodiacal light or Gegenschein, scattered moonlight...). In practice, it is the observed sky spectrum from which the emission lines have been removed by linear interpolation. The term "continuum" is adopted by contrast to "emission lines", but it is misleading in that this component may well include sharp absorption lines if it is due to scattering of moonlight in the earth's atmosphere.

2. An "emission lines" component containing only the telluric emission lines.

It is usefull only in the yellow, red and near IR regions, where telluric lines lie; this function will not be used below 520 nm. A few lines are present in the range 520–526 nm, then in the range 554–567 nm (where the very strong [O I] 5577Å line lies), but they become ubiquitous beyond 585 nm.
Each sky component will then be normalized using the $giNormalizeSky()$ function and independently modeled with the $giModelSky()$ function. Then the two models will have to be added together before being subtracted from the object spectra.

3.20.8 Mathematical description

It is assumed that the KW $wlFfSp$.WLENO is tested at the recipe level, in order to decide whether this function is called or not. The decision is based only on the expected presence or absence of telluric emission lines and is, in this respect, an *a priori* decision. The question of whether these lines have a significant amplitude to be considered in practice or not is addressed later, in the context of sky subtraction.
The `linesSkyMast` table is first checked to test for the existence of any telluric emission line. If no such line is found in the interval

$$[wlFfSp.\texttt{BINWLO}, wlFfSp.\texttt{BINWLO} + wlFfSp.\texttt{BINWNX} \cdot wlFfSp.\texttt{BINWSTEP}]$$

, then a warning is issued and one goes out of the function.
The object type and fiber number allowing to identify the sky spectra are given either by the `ozpoz` table (which gives the initially defined skies) or by the `numSky` table (which gives an *a posteriori* corrected sample of uncontaminated skies), as in the function $giNormalizeSky()$.
The positions λ_t of the telluric emission lines are given by the `LinesSkyMast` table. The widths w_t of the same lines have to be defined in a different way for each line, because the wings of very bright lines (like [O I] 5577Å) will have a significant flux at e.g. $3\,\sigma$ from the line center, while those of dim lines will not. The criterium will be that the flux due to the line is negligible at the positions $\lambda_t \pm w_t/2$, i.e. not larger than the average estimated error of the sky at $(10 \pm 2)\,psflwidth$ from the line center (this is an arbitrary criterion, it may be changed if necessary); in addition, the width is required to be larger than an *a priori* arbitrary limit:

$$\begin{aligned} w_t(\lambda_t, n) &\geq psflwidth(\lambda_t, n) \left[8 \ln \left(\frac{wlFfSp.\texttt{EXPTIME} \times \texttt{lineSkyMast}:FLUX(\lambda_t)}{wlFfSpErr(\lambda_t \pm 10\,psflwidth(\lambda_t, n), n)} \right) \right]^{1/2} \\ \text{and} \quad w_t(\lambda_t, n) &\geq ssWidMul \times psflwidth(\lambda_t, n) \end{aligned}$$

where $psflwidth$ is the width of the line given by the `giRebinSpectra` function (under the form of an image). This equation holds for a gaussian shape of the PSF and may be modified when a more realistic PSF shape will be available. The second condition, which must be satisfied as well, ensures that w_t cannot be lower than a given threshold; this precaution is motivated by the possibility that a faint line may be close to a strong one, where the flux error would be much larger than the continuum, leading to a too small value of w_t according to the first criterion alone. The average error of the sky quoted above is estimated by taking an average of the errors on the two intervals $[\lambda_t - 12\,psflwidth, \lambda_t - 8\,psflwidth]$ and $[\lambda_t + 8psflwidth, \lambda_t + 12psflwidth]$. The interval is dropped if a sky line falls in it; if lines fall in both, an interval defined for the preceding line is adopted.

- The "continuum" spectrum is obtained by linearly interpolating an average flux under each emission line. The average flux on each side of an emission line is defined as follows:

$$aveWlFfSp\left(\lambda_{ti} - \frac{w_{ti}}{2}, n\right) = \frac{\sum_{jmmini}^{jmmaxi} prop_j \cdot wlFfSp(\lambda_j, n)}{\sum_{jmmini}^{jmmaxi} prop_j} \tag{26}$$

$$aveWlFfSp\left(\lambda_{ti} + \frac{w_{ti}}{2}, n\right) = \frac{\sum_{jpmini}^{jpmaxi} prop_j \cdot wlFfSp(\lambda_j, n)}{\sum_{jpmini}^{jpmaxi} prop_j} \tag{27}$$

  where the sum limits are defined by

$$jmmini = ANINT\left(\frac{\lambda_{ti} - \frac{w_{ti}}{2} - psflwidth - wlFfSp.\texttt{BINWLO}}{wlFfSp.\texttt{BINWSTEP}}\right) + 1 \tag{28}$$

$$jmmaxi = ANINT\left(\frac{\lambda_{ti} - \frac{w_{ti}}{2} + psflwidth - wlFfSp.\texttt{BINWLO}}{wlFfSp.\texttt{BINWSTEP}}\right) + 1 \tag{29}$$

$$jpmini = ANINT\left(\frac{\lambda_{ti} + \frac{w_{ti}}{2} - psflwidth - wlFfSp.\texttt{BINWLO}}{wlFfSp.\texttt{BINWSTEP}}\right) + 1 \tag{30}$$

$$jpmaxi = ANINT\left(\frac{\lambda_{ti} + \frac{w_{ti}}{2} + psflwidth - wlFfSp.\texttt{BINWLO}}{wlFfSp.\texttt{BINWSTEP}}\right) + 1 \tag{31}$$

the function $ANINT()$ taking the nearest integer as in FORTRAN. $prop_j$ is the fraction of a wavelength bin included in the interval $[\lambda_{ti} - \frac{w_{ti}}{2} - psflwidth, \lambda_{ti} - \frac{w_{ti}}{2} + psflwidth]$ and should be ≤ 1 and ≥ 0:

$$prop_j = 1 \qquad \forall j \in [jmmini+1, jmmaxi-1], \quad [jpmini+1, jpmaxi-1] \tag{32}$$

$$prop_{jmmini} = \frac{ABS\left(\lambda_{ti} - \frac{w_{ti}}{2} - psflwidth - (\lambda_{jmmini} + 0.5 \cdot wlFfSp.\texttt{BINWSTEP})\right)}{wlFfSp.\texttt{BINWSTEP}} \tag{33}$$

$$prop_{jmmaxi} = \frac{ABS\left(\lambda_{ti} - \frac{w_{ti}}{2} + psflwidth - (\lambda_{jmmaxi} - 0.5 \cdot wlFfSp.\texttt{BINWSTEP})\right)}{wlFfSp.\texttt{BINWSTEP}} \tag{34}$$

$$prop_{jpmini} = \frac{ABS\left(\lambda_{ti} + \frac{w_{ti}}{2} - psflwidth - (\lambda_{jpmini} + 0.5 \cdot wlFfSp.\texttt{BINWSTEP})\right)}{wlFfSp.\texttt{BINWSTEP}} \tag{35}$$

$$prop_{jpmaxi} = \frac{ABS\left(\lambda_{ti} + \frac{w_{ti}}{2} + psflwidth - (\lambda_{jpmaxi} - 0.5 \cdot wlFfSp.\texttt{BINWSTEP})\right)}{wlFfSp.\texttt{BINWSTEP}} \tag{36}$$

Here $ABS()$ takes the absolute value. Finally, we have:

$$contSky(\lambda, n) \tag{37}$$

$$= wlFfSp(\lambda, n) \qquad \text{if} \quad \lambda \notin \left[\lambda_{ti} - \frac{w_{ti}}{2}, \lambda_{ti} + \frac{w_{ti}}{2}\right] \quad \forall i = 1, ...N_{lines} \tag{38}$$

$$= \frac{1}{2}\left(aveWlFfSp\left(\lambda_{ti} + \frac{w_{ti}}{2}, n\right) + aveWlFfSp\left(\lambda_{ti} - \frac{w_{ti}}{2}, n\right)\right) + \frac{\Delta wlFfSp}{w_{ti}}(\lambda - \lambda_{ti}) \tag{39}$$

$$\text{if} \quad \lambda \in \left[\lambda_{ti} - \frac{w_{ti}}{2}, \lambda_{ti} + \frac{w_{ti}}{2}\right] \quad \forall i = 1, ...N_{lines} \tag{40}$$

where $\Delta wlFfSp = aveWlFfSp(\lambda_{ti} + \frac{w_{ti}}{2}, n) - aveWlFfSp(\lambda_{ti} - \frac{w_{ti}}{2}, n)$ and where N_{lines} is the number of telluric emission lines in the observed wavelength range (it can be known by scanning the `LinesSkyMast` table). In the case of blends, i.e. if $\lambda_{ti} + w_{ti}/2 > \lambda_{ti+1} - w_{ti+1}$, then the limit $\lambda_{ti} + w_{ti}/2$ is replaced by the next limit, $\lambda_{ti+1} + w_{ti+1}/2$. $contSky$ is then stored into the first $wlFfSkyComp$ image.

The error on the "continuum" spectrum is simply interpolated between the errors at the limits of the interpolated segments. We prefer this solution to the one using the usual propagation formula, because according to the latter the error would be minimal at the center of the segment and would converge to the same single value (corresponding to the average of the errors of both points) at both ends of it. Here, the error varies linearly from one end of the segment to the other and takes the oberved value at each end of it, which appears more natural.

$$contSkyErr(\lambda, n) \tag{41}$$

$$= wlFfSpErr(\lambda, n) \qquad \text{if} \quad \lambda \notin \left[\lambda_{ti} - \frac{w_{ti}}{2}, \lambda_{ti} + \frac{w_{ti}}{2}\right] \quad \forall i = 1, ...N_{lines} \tag{42}$$

$$= \frac{1}{2}\left(wlFfSpErr\left(\lambda_{ti} + \frac{w_{ti}}{2}, n\right) + wlFfSpErr\left(\lambda_{ti} - \frac{w_{ti}}{2}, n\right)\right) + \frac{\Delta wlFfSpErr}{w_{ti}}(\lambda - \lambda_{ti}) \tag{43}$$

$$\text{if} \quad \lambda \in \left[\lambda_{ti} - \frac{w_{ti}}{2}, \lambda_{ti} + \frac{w_{ti}}{2}\right] \quad \forall i = 1, ...N_{lines} \tag{44}$$

where $\Delta wlFfSpErr = wlFfSpErr(\lambda_{ti} + \frac{w_{ti}}{2}, n) - wlFfSpErr(\lambda_{ti} - \frac{w_{ti}}{2}, n)$. $contSkyErr$ is then stored into the first $wlFfSkyCompErr$ image.

- The emission line spectrum, then, is simply the difference between the observed and "continuum" ones:

$$emiSky(\lambda, n) = wlFfSp(\lambda, n) - contSky(\lambda, n) \tag{45}$$

$emiSky$ is then stored into the second $wlFfSkyComp$ image. The error is considered as zero outside the emission lines, and takes its original, observed value in the emission lines:

$$emiSkyErr(\lambda, n) = 0 \qquad \text{if} \quad \lambda \notin \left[\lambda_{ti} - \frac{w_{ti}}{2}, \lambda_{ti} + \frac{w_{ti}}{2}\right] \quad \forall i = 1, ...N_{lines} \tag{46}$$

$$= wlFfSpErr(\lambda, n) \qquad \text{if} \quad \lambda \in \left[\lambda_{ti} - \frac{w_{ti}}{2}, \lambda_{ti} + \frac{w_{ti}}{2}\right] \quad \forall i = 1, ...N_{lines} \tag{47}$$

$emiSkyErr$ is then stored into the second $wlFfSkyCompErr$ image.

There are two output images in each of the two output frames, one for each sky component. The sky component is identified by putting respectively `CONTINUUM` or `EMISSION` into the KW $wlFfSkyComp$.`SKCOMP` of both $wlFfSkyComp$ images, and also into the KW $wlFfSkyCompErr$.`SKCOMP` of both $wlFfSkyCompErr$ images. The $wlFfSkyComp$ frame contains the spectra themselves while the $wlFfSkyCompErr$ frame contains the errors.

3.20.9 Pseudo code

to be done

3.21 Normalize the Sky spectra

Name: **giNormalizeSky**	Section: **Sky subtraction**	nu: **150**
Author(s)/Date(s): **VC/20000612**		Status: **To be validated by SB**

Working remarks:
Submitted to Internal rewiev Wed 09/06/00. Will be used fro both emission ad continuum subtraction (PN is doing the analysis).

3.21.1 Purpose

The total flux normalization is done for sky spectra

3.21.2 Function parameters

parameters to define the wavelength range where the total flux will be normalized.

Name	Type	Default	Description
`skWlenMin`	`float`	`undef`	minimum of the wavelength range
`skWlenMax`	`float`	`undef`	maximum of the wavelength range

3.21.3 Input frames

The input frame contains all spectra corrected for instrumental response and rebinned in wavelength space.

Name: *wlFfSp* **Type:** `BINSP` **Format:** `nlWbin`
Description: any rebinned spectra frame after correction for instrumental response and rebinning in the wavelength space.
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`GIRFTYPE`	DRS	`string`	none	type of frame data
`BINWNX`	DRS	`integer`	none	number of spectral elements on the rebinned S2dFrame
`BINWNY`	DRS	`integer`	none	number of spectra on the rebinned S2dFrame
`BINWSTEP`	DRS	`double`	nm	wavelength step of the linear rebinned spectra
`BINWLO`	DRS	`float`	nm	wavelength of the centre of the first spectral elements
`SLIT`	FGIR_ICS	`integer`	none	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies
`GRATING`	FGIR_ICS	`string`	none	DIC: ESO name for grating unit.
`WLEN0`	FGIR_ICS	`double`	nm	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength
`SKCOMP`	DRS	`string`	none	sky component: one of 'EMISSION' or 'CONTINUUM'

Name: *wlFfSpErr* **Type:** `BINSP` **Format:** `nlWbin`
Description: frame of the errors on the rebinned spectra
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
BINWNX	DRS	integer	none	number of spectral elements on the rebinned S2dFrame
BINWNY	DRS	integer	none	number of spectra on the rebinned S2dFrame
BINWSTEP	DRS	double	nm	wavelength step of the linear rebinned spectra
BINWLO	DRS	float	nm	wavelength of the centre of the first spectral elements
SLIT	FGIR_ICS	integer	none	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies
GRATING	FGIR_ICS	string	none	DIC: ESO name for grating unit.
WLEN0	FGIR_ICS	double	nm	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength
SKCOMP	DRS	string	none	sky component: one of 'EMISSION' or 'CONTINUUM'

3.21.4 Other inputs

Name	Type	Description
ozpoz	girOzpozTable	Positionner Binary Table; this is in fact not an input, but part of the inframe1
Relevant FITS keywords:	none (attached to frame header)	
Relevant table columns:		
OBJECT_TYPE	character	object type: P=programme object, S=sky, G=guide star, X=simulation, F=flux refrence star (request OGL/OP)
SPECT_FIBRE	integer	fibre number on the spectrograph slit
numSky	girSkyTable	single column table with the $Nsky$ fibre running numbers of spectra to be normalized.
Relevant FITS keywords:	GIRTTYPE SLIT GRATING WLEN0	
Relevant table columns:		
NSPEC	integer	Running number of the spectrum

3.21.5 Output frames

Name: *normSky* **Type:** `BINSP` **Format:** `nlWbin`
Description: $Nsky$ normalized sky spectra
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
BINWNX	DRS	integer	none	number of spectral elements on the rebinned S2dFrame
BINWNY	DRS	integer	none	number of spectra on the rebinned S2dFrame
BINWSTEP	DRS	double	nm	wavelength step of the linear rebinned spectra
BINWLO	DRS	float	nm	wavelength of the centre of the first spectral elements
SKWLENMIN	DRS	double	nm	SKY normalization: minimum wavelength
SKWLENMAX	DRS	double	nm	SKY normalization: maximum wavelength
SKCOMP	DRS	string	none	sky component: one of 'EMISSION' or 'CONTINUUM'

Name: *normSkyErr* **Type:** `BINSP` **Format:** `nlWbin`
Description: errors on the $Nsky$ normalized sky spectra
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
BINWNX	DRS	integer	none	number of spectral elements on the rebinned S2dFrame
BINWNY	DRS	integer	none	number of spectra on the rebinned S2dFrame
BINWSTEP	DRS	double	nm	wavelength step of the linear rebinned spectra
BINWL0	DRS	float	nm	wavelength of the centre of the first spectral elements
SKWLENMIN	DRS	double	nm	SKY normalization: minimum wavelength
SKWLENMAX	DRS	double	nm	SKY normalization: maximum wavelength
SKCOMP	DRS	string	none	sky component: one of 'EMISSION' or 'CONTINUUM'

3.21.6 Other outputs

Name	Type	Description
intSky	girIntSkyTable	integrated light sky parameter table: fibre running number $NSPEC$, integrated light for sky spectrum $NSPEC$ and corresponding error
Relevant FITS keywords:	GIRTTYPE SLIT GRATING WLEN0 SKWLENMIN SKWLENMAX SKCOMP	
Relevant table columns:		
NSPEC	integer	Running number of the spectrum
INTFLX	float	Integrated flux for sky spectrum
INTFLXERR	float	Error on the integrated flux

3.21.7 General description

The sky spectra are normalized to their total flux in the specified λ-range and a normalization factor is given. This function is called only once for blue wavelength ranges, where no telluric emission line exist. However, when working in the yellow or the red, i.e. in ranges spoiled by emission lines, the sky spectra are split into two components by the *giSplitSky*() function; therefore, in such cases each sky component has to be normalized separately. This is done by calling this function twice successively. The recipe will have to take into account that there will be two distinct *normSky* images and two distinct `intSky` tables, each for another sky component.

3.21.8 Mathematical description

The `ozpoz` table will give the object type and the fibre number on the spectrograph slit for all the spectra. A selection of the objects identified in this table as sky object type allows to build a table with the fibre number for the spectra identified as sky spectra during the preparation of the observations.
Another `numSky` table can be given, containing the fibre number retained for real sky spectra (this option allows the user to select the sky spectra whatever the fibre assignation during the P2PP phase - i.e. to remove some sky spectrum polluted by an object or to add a spectrum for which no object is attempted or no signal is observed). The total flux is normalized in the wavelength range given by the user (<`skWlenMin`> , <`skWlenMax`>). If one of those values is not furnished or is outside the wavelength range of the spectra, the wavelength of the first pixel is taken for <`skWlenMin`> and the wavelength of the last pixel for <`skWlenMax`> . The integrated light is computed for each flat-fielded and rebinned sky spectrum with a number n

$$\texttt{intSky}:INTFLX(n) = \sum_{\lambda=\texttt{skWlenMin}}^{\texttt{skWlenMax}} wlFfSp(\lambda, n)$$

The error on the integrated light is then:

$$\texttt{intSky}:INTFLXERR(n) = \left(\sum_{\lambda=\texttt{skWlenMin}}^{\texttt{skWlenMax}} wlFfSpErr(\lambda, n)^2 \right)^{1/2}$$

Each rebinned sky spectrum is normalized so that its integrated intensity is unity:

$$normSky(\lambda, n) = \frac{wlFfSp(\lambda, n)}{\texttt{intSky}:INTFLX(n)}$$

and the error on $normSky$ is simply the scaled error on $wlFfSp$, because the relative error on the integrated intensity remains negligible:

$$normSkyErr(\lambda, n) = \frac{wlFfSpErr(\lambda, n)}{\texttt{intSky}:INTFLX(n)}$$

This step will allow to interpolate the sky spectra using the width of the local 1-D PSF in the dispersion direction for each spectral element of the x axis, in order to remove the differential effects of a PSF which varies along the spectra.

3.21.9 Pseudo code

3.22 Model the Sky spectrum

Name: **giModelSky** Section: **Sky subtraction** nu: **160**
Author(s)/Date(s): **VC/20000612** Status: **To be validated by SB**

Working remarks:
Submitted to Internal rewiev Wed 09/06/00. Will be used fro both emission ad continuum subtraction (PN is doing the analysis) see following remarks.
Modifications brought on 08.11.2000:

- *The polynomial degrees I and J are adjusted automatically, but they cannot exceed maximum values transmitted as parameters.*
- *Rectangular coordinates X,Y rather than polar coordinates are adopted (because a sky due e.g. to scattered moonlight will have a more or less 1D gradient, not a cylindrical symmetry)*
- *The maximum degree K for the polynomial fit of fitNormSky is specified by a parameter. It is automatically adjusted (lowered) if the corresponding coefficient turns out to be insignificant.*
- *The possibility of a two-component sky is foreseen, though this function tackles only one component of it at a time. Therefore, it must be called twice in cases of a two-component sky. In any case, there will be only one component below about 5200 Angstroms.*
- *The errors in the sky spectra are used to define weights in the LSQ fits.*

(PN)

3.22.1 Purpose

The spatial sky light distribution is optionally modelized before a model of sky spectrum is build depending on the PSF width in the dispersion direction

3.22.2 Function parameters

σ-clipping parameters for the spatial distribution modeling of sky

Name	Type	Default	Description
`skModel`	`logical`	`TRUE`	Flag for no spatial distribution modeling (skModel=FALSE) or spatial distribution modeling (skModel=TRUE)
`skClipSigmamax`	`float`	`2.5`	multiple of σ for the upper σ-clipping limit
`skClipSigmamin`	`double`	`2.5`	multiple of σ for the lower σ-clipping limit
`skClipNiter`	`integer`	`1`	number of iterations (σ-clipping)
`skClipMfrac`	`float`	`0.9`	min fraction of points accepted/total (σ-clipping)
`skXdegmax`	`integer`	`2`	max polynomial degree for fitting sky spatial distribution in X
`skYdegmax`	`integer`	`2`	max polynomial degree for fitting sky spatial distribution in Y
`skWdegmax`	`integer`	`2`	max polynomial degree for fitting sky intensity versus PSF width
`skWClipSigma`	`float`	`3`	multiple of σ for the σ-clipping limit in the fit of the sky intensity versus PSF width
`skWClipNiter`	`integer`	`1`	number of iterations (σ-clipping of intensity vs PSF width)
`skWClipMfrac`	`float`	`0.9`	min fraction of points accepted/total (σ-clipping of intensity vs PSF width)
`skMeanSigma`	`float`	`3.0`	multiple of σ for the quality test (mean of the residuals)
`skMom2Sigma`	`float`	`4.5`	multiple of σ for the quality test (2nd moment of the residuals)
`skMom3Sigma`	`float`	`12.0`	multiple of σ for the quality test (3d moment of the residuals)
`skMom4Sigma`	`float`	`15.0`	multiple of σ for the quality test (3d moment of the residuals)

3.22.3 Input frames

The input frame contains the Nsky normalized sky spectra

Name: *normSky* **Type:** BINSP **Format:** nlWbin
Description: Normalized $Nsky$ sky spectra in the wavelength space
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
BINWNX	DRS	integer	none	number of spectral elements on the rebinned S2dFrame
BINWNY	DRS	integer	none	number of spectra on the rebinned S2dFrame
BINWSTEP	DRS	double	nm	wavelength step of the linear rebinned spectra
BINWLO	DRS	float	nm	wavelength of the centre of the first spectral elements
SLIT	FGIR_ICS	integer	none	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies
GRATING	FGIR_ICS	string	none	DIC: ESO name for grating unit.
WLEN0	FGIR_ICS	double	nm	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength
SKCOMP	DRS	string	none	sky component: one of 'EMISSION' or 'CONTINUUM'

Name: *normSkyErr* **Type:** BINSP **Format:** nlWbin
Description: errors on the $Nsky$ normalized sky spectra
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
BINWNX	DRS	integer	none	number of spectral elements on the rebinned S2dFrame
BINWNY	DRS	integer	none	number of spectra on the rebinned S2dFrame
BINWSTEP	DRS	double	nm	wavelength step of the linear rebinned spectra
BINWLO	DRS	float	nm	wavelength of the centre of the first spectral elements
SKWLENMIN	DRS	double	nm	SKY normalization: minimum wavelength
SKWLENMAX	DRS	double	nm	SKY normalization: maximum wavelength
SKCOMP	DRS	string	none	sky component: one of 'EMISSION' or 'CONTINUUM'

Name: *psflwidth* **Type:** BINSP **Format:** nlWbin
Description: Width parameter for 1-D PSF along the dispersion direction (x)
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
BINWNX	DRS	integer	none	number of spectral elements on the rebinned S2dFrame
BINWNY	DRS	integer	none	number of spectra on the rebinned S2dFrame
BINWSTEP	DRS	double	nm	wavelength step of the linear rebinned spectra
BINWLO	DRS	float	nm	wavelength of the centre of the first spectral elements

3.22.4 Other inputs

Name	Type	Description
intSky	girIntSkyTable	sky integrated over the observed spectral range
Relevant FITS keywords:	GIRTTYPE SLIT GRATING WLENO SKCOMP	
Relevant table columns:		
NSPEC	integer	Running number of the spectrum
INTFLX	float	Integrated flux for sky spectrum
INTFLXERR	float	Error on the integrated flux
ozpoz	girOzpozTable	Positionner Binary Table; this is in fact not an input, but part of the inframe1
Relevant FITS keywords:	none (attached to frame header)	
Relevant table columns:		
R	integer	R position of the fibre
THETA	real	theta position of the fibre
SPECT_FIBRE	integer	fibre number on the spectrograph slit

3.22.5 Output frames

Name: *skyModel* **Type:** BINSP **Format:** nlWbin

Description: frame with modeled sky spectra corresponding to each spectrum of the rebinned spectra frame

Relevant FITS keywords:

Name	DIC	Type	Unit	Description
SKCLIPSIGMA	DRS	float	none	multiple of sigma (sigma-clipping)
SKCLIPNITER	DRS	integer	none	number of iterations (sigma-clipping)
SKCLIPMFRAC	DRS	float	none	min fraction of points accepted/total (sigma-clipping)
SKMODEL	DRS	integer	none	SKY modelling: 0=no spatial distribution, 1 or more=spatial distribution
SKINTXY	DRS	float	none	Coefficients of (x,y) DEGXIS, DEGYIS degree normet-to-one polynomial describing the spatial variation of sky intensity. The coefficient are sorted in order x,y as c1+c2*x+c3*y+c4*x*y+c5*x*y2+c6*x2*y+c7*x3+ ...
SKINTDEGX	DRS	integer	none	X - degree of the polynomial describing the spatial variation of sky intensity
SKINTDEGY	DRS	integer	none	X - degree of the polynomial describing the spatial variation of sky intensity
SKCOMP	DRS	string	none	sky component: one of 'EMISSION' or 'CONTINUUM'

Name: *skyModelErr* **Type:** `BINSP` **Format:** `nlWbin`
Description: Estimated errors on skyModel (at each λ for each spectrum)
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
SKCLIPSIGMA	DRS	float	none	multiple of sigma (sigma-clipping)
SKCLIPNITER	DRS	integer	none	number of iterations (sigma-clipping)
SKCLIPMFRAC	DRS	float	none	min fraction of points accepted/total (sigma-clipping)
SKMODEL	DRS	integer	none	SKY modelling: 0=no spatial distribution, 1 or more=spatial distribution
SKINTXY	DRS	float	none	Coefficients of (x,y) DEGXIS, DEGYIS degree normet-to-one polynomial describing the spatial variation of sky intensity. The coefficient are sorted in order x,y as c1+c2*x+c3*y+c4*x*y+c5*x*y2+c6*x2*y+c7*x3+ ...
SKINTDEGX	DRS	integer	none	X - degree of the polynomial describing the spatial variation of sky intensity
SKINTDEGY	DRS	integer	none	X - degree of the polynomial describing the spatial variation of sky intensity
SKCOMP	DRS	string	none	sky component: one of 'EMISSION' or 'CONTINUUM'

3.22.6 Other outputs

Name	Type	Description
sNorm	girSkyCoefTable	Coefficients matrix $(Nlambda, K)$ of the polynomial expression for the fit of the sky intensity versus the PSF width for each spectral bin
Relevant FITS keywords:	GIRTTYPE SLIT GRATING WLEN0 SKCLIPSIGMA SKCLIPNITER SKCLIPMFRAC SKMODEL SKINTXY SKINTDEGX SKINTDEGY FATHER SKCOMP	
Relevant table columns:		
SNORMK	float	coefficients of 1-D polynomial of degree K in PSF width
qualSkyMod	girSkyMomTable	table with mean, 2nd, 3rd and 4th moments calculated from the difference between sky spectra and the modeled ones for each spectral bin
Relevant FITS keywords:	none	
Relevant table columns:		
LAMBDA	float	wavelength
RMEAN	float	mean
RMOM2	float	2nd moment
RMOM3	float	3rd moment
RMOM4	float	4th moment

3.22.7 General description

The sky spectrum is modeled optionally over the field of view and then over the whole CCD applying the modified Viton-Milliard method (RD9).
This function may be called either only once, when the sky contains no emission telluric lines, or twice, when telluric emission lines are present and the sky is splitted into two components. Each of these components is treated independently through a corresponding call to this function (after a call

3.22.8 Mathematical description

1. **Build optionally a model of the spatial distribution of the sky integrated light (SKY-INT)**

   This option is offered if `skModel` = 1.
   If the sky is smoothly varying over the FoV, a model depending on the position in the FoV is fitted to the

integrated sky intensities obtained, `intSky` :$INTFLX(n)$, using by default a 2-D polynomial form:

$$fitIntSky(X',Y') = \sum_{i,j=0}^{I,J} sInt(i,j)\,T_i(X')\,T_j(Y')$$

where T_i and T_j are Chebyshev polynomials of degree i and j, and $X' = \frac{2}{\texttt{XMAX}-\texttt{XMIN}}(X - \frac{\texttt{XMIN}+\texttt{XMAX}}{2})$, $Y' = \frac{2}{\texttt{YMAX}-\texttt{YMIN}}(Y - \frac{\texttt{YMIN}+\texttt{YMAX}}{2})$ so that $-1 < X' < +1$ and $-1 < Y' < +1$. $I \leq$ `skXdegmax` and $J \leq$ `skYdegmax` are the polynomial degrees adjusted automatically depending on the number of sky fibres and significance of coefficients (see below), and $sInt(i,j)$ are the fitted coefficients. The model is sigma-clipped according to $<$`skClipSigmamax`$>$,$<$`skClipSigmamin`$>$,$<$`skClipNiter`$>$, $<$`skClipMfrac`$>$ parameters (a different rejection limit can be used for points lying above or below the fitted intensity in order to reject sky spectra polluted by an object). If some spectra are rejected by the sigma-clipping treatment, they are not kept for the further modelization.

The LSQ fit above is a weighted one, the weights being

$$weightintSky(n) = 1/\texttt{intSky}\,{:}INTFLXERR(n)^2$$

In addition, special care is brought to the choice of the polynomial degrees through an iterative procedure.

(a) the procedure verifies that the number of coefficients to be fitted according to the values of $<$`skXdegmax`$>$ and $<$`skYdegmax`$>$ is strictly smaller than the number $normSky$.`BINWNY` of measured skies: `skXdegmax` $\cdot$ `skYdegmax` $< normSky$.`BINWNY`.
If the number of coefficients is too large, one adopts `skXdegmax` $=$ `skXdegmax` $- 1$ and `skYdegmax` $=$ `skYdegmax` $- 1$, and one verifies again.

(b) All possible fits are done with all values of I and J, from a weighted average ($I = J = 0$) to the case $I =$ `skXdegmax`, $J =$ `skYdegmax`

(c) For each fit, the residuals are computed and their standard deviation σ_{res} are intercompared. The fit with the smallest σ_{res} (within 5%) and with the smallest I and the smallest J (which corresponds to the smallest number of coefficients) is adopted; in other words, among a few fits leading to σ_{res} values which are similar within 5%, priority is given to that with the smallest value of I and the smallest value of J.

(d) The σ_{res} value of the adopted fit, which may be called an "external" σ, is compared with the expected average standard deviation (an "internal" σ):

$$exp\sigma_{res} = \frac{1}{\sum_{n=1}^{normSky.\texttt{BINWNY}} weightintSky(n)}$$

(e) Sigma clipping is applied as mentioned above. If the remaining number of points $NremainSky$ is smaller than the minimum allowed value, i.e. if

$$\frac{NremainSky}{normSky.\texttt{BINWNY}} < \texttt{skClipMfrac}$$

then the rejected data with the smallest residuals are reintroduced until the above fraction reaches or exceeds $<$`skClipMfrac`$>$. Finally, the whole procedure is repeated until there is no more rejection, or the number of iterations reaches $<$`SkClipNiter`$>$, or the fraction $<$`skClipMfrac`$>$ is reached.

This fitted model provides a continuous description of the sky intensities all over the field.
The coordinates (X,Y) of the fibre in the FoV should be computed from the polar coordinates (R,$THETA$) given in the `ozpoz` table:

$$X(n) = \texttt{ozpoz}\,{:}R(n) \times \cos(\texttt{ozpoz}\,{:}THETA(n))$$

$$Y(n) = \texttt{ozpoz}\,{:}R(n) \times \sin(\texttt{ozpoz}\,{:}THETA(n))$$

2. **Build a model of normalized sky spectra for all spectra**

In order to match the PSF of all spectra to be sky-subtracted, a modified Viton-Milliard method, taking into account a discontinuous PSF variation, is implemented.

Since the PSF does not vary smoothly on the y axis (local defocusing due to the sub-slit mechanical assembly discontinuities), the Viton-Milliard method cannot be applied without modification.

In the modified method for each λ-bin a 1-D polynomial using the width of the PSF as independent variable is fitted to the normalized sky spectra $normSky(\lambda, n)$

$$fitNormSky(\lambda, n) = \sum_{k=1}^{K} sNorm(\lambda, k)\, T_k(psflwidth'(\lambda, n)) \tag{48}$$

where $sNorm(\lambda, k)$ are the fitted coefficients, T_k are Chebyshev polynomials of degree k and $psflwidth'(\lambda, n)$ is the width of the PSF, normalized in such a way that all its values span the range $[-1, +1]$. Notice that another normalization has to be done for each λ value, since the range of $psflwidth$ is slightly different from one wavelength to another. Then, the sky to be subtracted from any extracted spectrum n can be estimated on the basis of the PSF width corresponding to this position. Notice that only the $NremainSky$ sky spectra are used in this fit, if option `skModel` = 1 has been performed.

In this LSQ fit, weights are taken into account:

$$weightnormSky(\lambda, n) = 1/normSkyErr(\lambda, n)^2 \tag{49}$$

(a) The polynomial degree K is set to a minimum value of 1. Note that <`skWdegmax`> should probably be larger for the emission line component of the sky than for the continuum component, since the emission lines are expected to provide the largest variations.

(b) The fit is performed with the adopted value of K.

(c) One sets $K = K + 1$ and the fit is performed again, provided that $K \leq$ `skWdegmax`. If the coefficient of the largest degree polynomial $sNorm(\lambda, K) < 1.5\sigma_{sNorm(\lambda,K)}$ (where $\sigma_{sNorm(\lambda,K)}$ is the error on the coefficient estimated from the fit), then one adopts the preceding fit with $K = K - 1$. If not, the last fit is adopted.

(d) Sigma clipping is applied, rejecting points farther away than `skWClipSigma`$\times normSkyErr(\lambda, n)$ from the fitted curve but reintroducing the least bad ones as soon as the fraction of good points is below <`skWClipMfrac`> .

(e) The fit is reiterated by going to 2b.

The iterations are stopped as soon as no more point is rejected, or the ratio of accepted to initial number of points is just above or equal to <`skWClipMfrac`> , or the number of iterations reaches <`skWClipNiter`> . The final value of K is defined in the process, and the final number of data used is stored in the internal vector $NremainNormSky(\lambda)$. The running numbers of these data are kept in $nremNorSky(\lambda)$.

The above scheme should be valid as such for the continuum component of the sky, which is always positive. However, additional precautions have to be taken for the emission line component, because in this case parts of the spectra are set to zero and therefore their variance is null and their weight undefined. The limits have to be defined very carefully, in the following way (which is valid for both emission and continuum components):

(a) For each λ, define the limits $Wsky_{min}$, $Wsky_{max}$ from the frame $psflwidth$ and taking into account the spectra numbers given in the `intSky` table.

(b) Compare the limits obtained – which hold for the sky spectra only – with those defined from the whole set of spectra, W_{min} and W_{max}. One has necessarily $(Wsky_{max} - Wsky_{min}) \leq (W_{max} - W_{min})$, so that in many cases some degree of extrapolation will have to be practiced to model the sky spectra. Issue a warning whenever $(Wsky_{max} - Wsky_{min}) < 0.7(W_{max} - W_{min})$.

(c) Check whether or not zero intensities with zero errors exist (this should only be the case for the emission line component!).

- **If not, then:**
  Define the new variable $psflwidth'(\lambda, n)$ as

  $$psflwidth'(\lambda, n) = \frac{2}{Wsky_{max} - Wsky_{min}} \cdot \left(psflwidth - \frac{Wsky_{max} + Wsky_{min}}{2}\right) \tag{50}$$

  and make a classical weighted LSQ fit according to equations 48 and 49.

- **If yes, then** sort the normalized sky intensities $normSky$ by increasing order, as well as their errors $normSkyErr$.
  - If all of them are in this case, the model is set to zero (this is the case between the emission lines, in the emission component) in the full $[W_{min}, W_{max}]$ interval.
  - In the wings of emission lines, zero intensities may appear for small $Wsky$ values while positive values (even some negative ones, in the far wings) will appear at larger $Wsky$ values. In this case, set the model to zero on the whole domain $[Wsky_{min}, Wsky0_{max}]$ covered by consecutive null values; $Wsky0_{max}$ is the largest $Wsky$ value for which the sky intensity is zero. In the remaining part of the interval, the new variable $psflwidth'(\lambda, n)$ is defined as

$$psflwidth'(\lambda, n) = \frac{2}{Wsky_{max} - Wsky0_{max}} \cdot \left(psflwidth - \frac{Wsky0_{max} + Wsky_{max}}{2}\right) \quad (51)$$

  and a LSQ solution of the equation 48 is computed, constrained by the condition that the function must be (very nearly) zero at $psflwidth = Wsky0_{max}$ or $psflwidth'(\lambda, n) = -1$. This condition will be imposed by weighting, as described by Lawson & Hanson (1974, Solving Least Squares Problems, Prentice-Hall, Chapter 22, pp. 148-157). An additional condition should require that the fitted function is positive over the whole relevant range; however, not imposing this condition from the beginning yields the opportunity of a quality test.

(d) Perform the iterative adjustment of the K polynomial degree and sigma clipping, as described above.

(e) Verify that the fitted function is positive over the whole pertinent interval; if it is not, then issue a warning and perform the fit again with the condition of positivity according to the method of Lawson & Hanson Chapter 23 (continuum sky component) or with both conditions of nul value at $psflwidth'(\lambda, n) = -1$ and of positivity, with the method of Lawson & Hanson Chapter 23, Section 6, p. 168 (emission line component). Finally, the number of data used is stored in $NremainNormSky(\lambda)$ and the running numbers of these data are kept in $nremNorSky(\lambda)$, as in 2.

3. **Transform the model back to original intensities (de-normalize)**

The amplitude of the PSF-corrected sky is then transformed back to the original intensity: if the sky is smoothly varying over the FoV, then

$$skyModel(\lambda, n) = fitNormSky(\lambda, n) \times fitIntSky(X, Y)$$

and the error on the modeled sky is defined in the following way, neglecting the error on the integrated sky spectra:

$$skyModelErr(\lambda, n) = \frac{1}{\sqrt{NremainNormSky(\lambda)}} \times \overline{normSkyErr(\lambda)} \times fitIntSky(X, Y)$$

with

$$\overline{normSkyErr(\lambda)} \approx \frac{1}{NremainNormSky(\lambda)} \sum_{i=1}^{NremainNormSky(\lambda)} normSkyErr(\lambda, nremNorSky(\lambda, i))$$

If the sky is strongly heterogeneous over the FoV, the spatial sky modeling (`skModel` $= 0$) is skipped and

$$skyModel(\lambda, n) = fitNormSky(\lambda, n) \times \texttt{intSky}:INTFLX(m)$$

where m is the number of the closest sky-fibre neighboring the n-th. Therefore, m is such that

$$(X(m) - X(n))^2 + (Y(m) - Y(n))^2$$

is minimum. The error on the modeled sky is now defined as:

$$skyModelErr(\lambda, n) = normSkyErr(\lambda, m) \times \texttt{intSky}:INTFLX(m)$$

Quality assessment

The measured sky spectra are compared to the modeled ones. Several numerical indicators (rms and higher moments for each spectral bin; TBC the value of the highest moment to be given, for the moment 4) will be stored in a table $qualSkyMod(\lambda, i^{th}\ moment)$.

To perform this test, one has to compute back the original sky intensities, $obsSky(\lambda, n)$. The residual intensities can then be computed, as well as the moments:

$$obsSky(\lambda, n) = \texttt{intSky}:INTFLX(n) \cdot normSky(\lambda, n)$$

$$res(\lambda, n) = obsSky(\lambda, n) - skyModel(\lambda, n)$$

$$\texttt{qualSkyMod}:RMEAN = \frac{1}{Nsky} \sum_{i=1}^{Nsky} res(\lambda, i)$$

$$\texttt{qualSkyMod}:RMOM2 = \frac{1}{Nsky-1} \sum_{i=1}^{Nsky} res(\lambda, i)^2$$

$$\texttt{qualSkyMod}:RMOM3 = \frac{1}{Nsky-1} \sum_{i=1}^{Nsky} res(\lambda, i)^3$$

$$\texttt{qualSkyMod}:RMOM3 = \frac{1}{Nsky-1} \sum_{i=1}^{Nsky} res(\lambda, i)^4$$

where $Nsky = normSky.\texttt{BINWNY}$.
The values of these moments should remain within the following boundaries (TBC), otherwise a warning is issued:

$$|\texttt{qualSkyMod}:RMEAN(\lambda)| < skMeanSigma \times \sigma(\overline{Sky}) \qquad (52)$$

with

$$\sigma(\overline{Sky}) = \frac{1}{Nsky} \left(\sum_{i=1}^{Nsky} \texttt{intSky}:INTFLX(i) \times normSkyErr(\lambda, i) \right)^{1/2}$$

$$\left| \left(\frac{\texttt{qualSkyMod}:RMOM2}{\overline{var(Sky)}} - 1 \right) \cdot \sqrt{Nsky} \right| < skMom2Sigma \qquad (53)$$

with

$$\overline{var(Sky)} = \frac{1}{Nsky} \sum_{i=1}^{Nsky} \left(\texttt{intSky}:INTFLX(i) \times normSkyErr(\lambda, i) \right)^2$$

$$\left| \frac{\texttt{qualSkyMod}:RMOM3}{\overline{var(Sky)}^{3/2}} \cdot \sqrt{Nsky} \right| < skMom3Sigma \qquad (54)$$

$$\left| \left(\frac{\texttt{qualSkyMod}:RMOM4}{\overline{var(Sky)}^{2}} - 3 \right) \cdot \sqrt{Nsky} \right| < skMom4Sigma \qquad (55)$$

3.22.9 Pseudo code

3.23 Subtract the Sky spectrum

Name: **giSubtractSky** Section: **Sky subtraction** nu: **170**
Author(s)/Date(s): **VC/20000612** Status: **To be validated by SB**

Working remarks:
The description TBcompleted and developed (PN is doing the analysis) see following remarks.

- *3.20.8: $diffAmpli(i)$ and $varAmpli(i)$ should be defined more clearly, as well as k. The whole description lacks clarity.*
- *Two-component sky: We have to decide how to implement this. It depends on whether we modify `giModelSky()` or not: if yes, a single call to `giModelSky()` will produce the final combined sky model (addition of both components); if no, then this function will have to be modified in order to combine the two models given as two successive outputs of `giModelSky()`. However, options 2 and 3 will be difficult to handle and should be reworked, because they should apply to each of the two sky components instead of a single one (in the red only, however).*
- *(28.11.2000) There should be no major problem in leaving this function more or less as it is, and calling it twice successively, once for the continuum component and a second time for the emission line component (when necessary).*

(PN)

3.23.1 Purpose

The modelized sky spectra are subtracted in the wavelength space.

3.23.2 Function parameters

Name	Type	Default	Description
`skSubMethod`	`integer`	`1`	1=direct subtraction of sky model, 2=amplitude adjustment for sky model intensity with cross-correlation method, 3=amplitude adjustment for sky model intensity with error minimization method
`skVarAmpl`	`float`	`0.1`	Initial increment of the variation of the sky amplitude for skSubMethod=2 or 3
`skFracAmpl`	`float`	`0.1`	Maximum fraction of the relative difference in the sky amplitude, for skSubMethod=2 or 3, stopping iterations as soon as it is reached
`skConv`	`float`	`1`	Convergence parameter defining skVarAmpl at each iteration

3.23.3 Input frames

The first input frame contains all spectra corrected for instrumental response and rebinned in wavelength space and the second one contains the Nsky normalized sky spectra.

Name: *wlFfSp* **Type:** `BINSP` **Format:** `nlWbin`
Description: frame of extracted spectra after correction for instrumental response and rebinning in the wavelength space
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
BINWNX	DRS	integer	none	number of spectral elements on the rebinned S2dFrame
BINWNY	DRS	integer	none	number of spectra on the rebinned S2dFrame
BINWSTEP	DRS	double	nm	wavelength step of the linear rebinned spectra
BINWLO	DRS	float	nm	wavelength of the centre of the first spectral elements
SKCOMP	DRS	string	none	sky component: one of 'EMISSION' or 'CONTINUUM'
GRATRES	FGIR_ICS	double	none	DIC: ENC = OFFST+ RESOL* RAD2DEG* arcsin(WLEN* GRV/(2* cos(ROT)))

Name: *wlFfSpErr* **Type:** `BINSP` **Format:** `nlWbin`
Description: frame of extracted spectra errors
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
BINWNX	DRS	integer	none	number of spectral elements on the rebinned S2dFrame
BINWNY	DRS	integer	none	number of spectra on the rebinned S2dFrame
BINWSTEP	DRS	double	nm	wavelength step of the linear rebinned spectra
BINWLO	DRS	float	nm	wavelength of the centre of the first spectral elements
SKCOMP	DRS	string	none	sky component: one of 'EMISSION' or 'CONTINUUM'

Name: *skyModel* **Type:** `BINSP` **Format:** `nlWbin`
Description: frame with modeled sky spectra corresponding to each spectrum of the rebinned spectra frame
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
BINWNX	DRS	integer	none	number of spectral elements on the rebinned S2dFrame
BINWNY	DRS	integer	none	number of spectra on the rebinned S2dFrame
BINWSTEP	DRS	double	nm	wavelength step of the linear rebinned spectra
BINWLO	DRS	float	nm	wavelength of the centre of the first spectral elements
SKCOMP	DRS	string	none	sky component: one of 'EMISSION' or 'CONTINUUM'

Name: *skyModelErr* **Type:** `BINSP` **Format:** `nlWbin`
Description: frame with modeled sky spectra errors
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
BINWNX	DRS	integer	none	number of spectral elements on the rebinned S2dFrame
BINWNY	DRS	integer	none	number of spectra on the rebinned S2dFrame
BINWSTEP	DRS	double	nm	wavelength step of the linear rebinned spectra
BINWLO	DRS	float	nm	wavelength of the centre of the first spectral elements
SKCOMP	DRS	string	none	sky component: one of 'EMISSION' or 'CONTINUUM'

3.23.4 Other inputs

none

3.23.5 Output frames

Name: *skySubSp* **Type:** `BINSP` **Format:** `nlWbin`
Description: frame with sky-subtracted spectra
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`SKSUBMETHOD`	DRS	`integer`	none	1=direct subtraction of sky model, 2=amplitude adjustement for sky model intensity with cross-correlation method, 3=amplitude adjustement for sky model intensity with error minimization method
`SKSUB`	DRS	`string`	none	subtracted sky component: one of 'WHOLE','EMISSION', 'CONTINUUM' or 'EMISSION+CONTINUUM'

Name: *skySubSpErr* **Type:** `BINSP` **Format:** `nlWbin`
Description: frame with sky-subtracted spectra errors
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`SKSUBMETHOD`	DRS	`integer`	none	1=direct subtraction of sky model, 2=amplitude adjustement for sky model intensity with cross-correlation method, 3=amplitude adjustement for sky model intensity with error minimization method
`SKSUB`	DRS	`string`	none	subtracted sky component: one of 'WHOLE','EMISSION', 'CONTINUUM' or 'EMISSION+CONTINUUM'

3.23.6 Other outputs

none

3.23.7 General description

The sky model is subtracted from each object spectrum. Three options will be offered: the first one is a direct subtraction of the sky model and in case where the user would have a more precise subtraction of the telluric emission lines component, the control of the sky subtraction will be done by adjusting the amplitude of the sky model intensity via two methods. The amplitude adjustment is done iteratively for each spectrum with two methods:

- a cross-correlation between the sky subtracted spectrum with the modeled sky or with a mask containing the telluric emission lines will give the amplitude of the correlation function for a zero shift.
- the RMS scatter of the sky subtracted spectrum in the spectral ranges where the telluric lines are spread is computed.

3.23.8 Mathematical description

1. If `skSubMethod` = 1, the model is subtracted directly:

$$skySubSp(\lambda, n) = wlFfSp(\lambda, n) - skyModel(\lambda, n)$$

   Note, that this is done on all spectra, including the measured sky spectra. The keyword *skySubSp*.`SKSUB` is set to the following value:

   - If *wlFfSp*.`SKCOMP` is undefined (not existent), then *skySubSp*.`SKSUB` $=' WHOLE'$ because it means that the image has not passed through the *giSplitSky*() function, so that the sky has not been separated into two components.

- If $wlFfSp$.SKCOMP is undefined but $skyModel$.SKCOMP $=' EMISSION'$ or $skyModel$.SKCOMP $=' CONTINUUM'$, then $skySubSp$.SKSUB $= skyModel$.SKCOMP because it means that this is the first call of the *giSubtractSky*() function, for the first of the two sky components.
- If $wlFfSp$.SKCOMP is defined but $skyModel$.SKCOMP $\neq wlFfSp$.SKCOMP, then $skySubSp$.SKSUB $=' CONTINUUM{+}EMISSION'$, because in this case the input frame has already passed once through the *giSubtractSky*() function.

The errors on the sky subtracted spectra become:

$$skySubSpErr(\lambda, n) = \left(wlFfSpErr(\lambda, n)^2 + skyModelErr(\lambda, n)^2\right)^{1/2}$$

2. If skSubMethod = 2, an iterative procedure is applied to refine the sky subtraction, $skyModel$ being considered only as a first approximation. The iteration proceeds as follows:

   (a) For each iteration i, three sky subtracted spectra are evaluated with $j = -1, 0, 1$:

   $$skySubSp_j(\lambda, n) = wlFfSp(\lambda, n) - (1 + diffAmpl(i) + j \times skVarAmpl(i)) \times skyModel(\lambda, n)$$

   where $diffAmpl$ is a correction to the amplitude of $skyModel$ and <skVarAmpl> is the step adopted for this correction, to explore a larger and a smaller correction. Initially one has $diffAmpl = 0$ and normally one should always have $diffAmpl(i) \ll 1 \quad \forall i$, if $skyModel$ was built successfully. The initial value of <skVarAmpl> is arbitrary and given through a parameter.

   (b) The 3 CCFs are computed, between each of the 3 sky subtracted spectra and the sky model spectrum of the relevant sky component. The CCF could optionally be computed with the binary mask of the telluric emission lines (TBC). The correlation is computed in the wavelength domain, not in the velocity domain since we are not interested in any shift but only in the amplitude of the CCF. The CCF is computed in the following way: Formally, the CCF can be written

   $$C(\epsilon) = \frac{R(\epsilon)}{R(\epsilon = \infty)}$$

   with

   $$R(\epsilon) = \int_{-\infty}^{+\infty} F(v) \cdot T(v - \epsilon)\, dv$$

   where F is the spectrum to be examined and T the template spectrum, here the model sky. In practice, the integral is replaced by a sum:

   $$R(\epsilon, n) = \frac{\Delta\lambda}{\Delta\lambda - abs(\epsilon)} \cdot \sum_{i=1}^{wlFfSp.\mathtt{BINWNX}} skySubSp_j(\lambda_i, n) \cdot skyModel(\lambda_i - \epsilon, n)$$

   where

   $$\lambda_i = wlFfSp.\mathtt{BINWLO} + (i-1) \cdot wlFfSp.\mathtt{BINWSTEP}$$
   $$\Delta\lambda = (wlFfSp.\mathtt{BINWNX} - 1) \cdot wlFfSp.\mathtt{BINWSTEP}$$

   and where the fraction before the sum cancels the slow variation of the CCF due to the shrinking of the wavelength range common to both spectra. The shift ϵ and its limits must now be defined:

   $$\epsilon = l \cdot wlFfSp.\mathtt{BINWSTEP}, \qquad l = -l_{max}, ..., 0, ... + l_{max}$$

   with

   $$l_{max} = nint\left(\frac{2 \cdot FWHM}{wlFfSp.\mathtt{BINWSTEP}}\right) = nint\left(\frac{2 \cdot wlFfSp.\mathtt{WLEN0}}{wlFfSp.\mathtt{BINWSTEP} \cdot wlFfSp.\mathtt{GRATRES}}\right)$$

   so that the correlation range is $\pm\epsilon_{max} \sim \pm 2 \cdot FWHM$ (the $nint()$ function takes the nearest integer, as in the FORTRAN definition).

   The normalization of the CCF has to be carefully defined because in the case of the $wlFfSp$.SKCOMP $=' EMISSION'$ sky component, there is no continuum left, so the CCF is in principle zero far from the center and it is not possible to normalize it by $R(\epsilon = \infty)$. Therefore we make the following choice:

- If $skyModel.\texttt{SKCOMP} =' EMISSION'$, then

$$C(\epsilon) = \frac{R(\epsilon)}{R(0)}$$

- If $skyModel.\texttt{SKCOMP} =' CONTINUUM'$, then

$$C(\epsilon) = \frac{2 \cdot R(\epsilon)}{R(-\epsilon_{max}) + R(+\epsilon_{max})}$$

- If $skyModel.\texttt{SKCOMP} =' UNDEFINED'$, then

$$C(\epsilon) = \frac{R(\epsilon)}{R(0)}$$

The case $skyModel.\texttt{SKCOMP} =' UNDEFINED'$ occurs when the $giSplitSky()$ function has not been called, either because there is only one component physically (blue wavelength range) or because the user judges that both components can be modeled as a single entity.

(c) The 3 values obtained give the amplitude of the CCF at the 3 points $diffAmpl - \texttt{skVarAmpl}$, $diffAmpl$ and $diffAmpl + \texttt{skVarAmpl}$. These values may be positive or negative: for instance, for the emission line sky component, the CCF amplitude will be positive when the subtracted sky is too small (thus leaving positive emission line signatures) while it will be negative when the subtracted sky is too large. The same holds for the "continuum" sky component, if it includes absorption lines (typically in the case of scattered moonlight).

The CCF amplitudes are linearly interpolated to find the zero of the function giving the amplitude of the CCF peak. This computed zero value is then defined as $diffAmpl(i + 1)$. Then, a new iteration is performed if $diffAmpl(i + 1) - diffAmpl(i)$ is larger than a given fraction $<\texttt{skFracAmpl}>$ of $diffAmpl(i)$. For the iteration $i + 1$ the value of $varAmpl(i + 1)$ is set equal to

$$(diffAmpl(i + 1) - diffAmpl(i))/\texttt{skConv}$$

where $<\texttt{skConv}>$ is a convergence parameter chosen such that $\texttt{skConv} \sim 1$ or > 1.

The above procedure is repeated for each spectrum.

Finally each sky subtracted spectrum is computed for the m-th iteration:

$$skySubSp(\lambda, n) = wlFfSp(\lambda, n) - (1 + diffAmpl(m)) \times skyModel(\lambda, n)$$

and the keyword $skySubSp.\texttt{SKSUB}$ takes the value defined under the option $\texttt{skSubMethod} = 1$. Here, the errors on the sky subtracted spectra are:

$$skySubSpErr(\lambda, n) = \left(wlFfSpErr(\lambda, n)^2 + (1 + diffAmpl(m))^2 \times skyModelErr(\lambda, n)^2\right)^{1/2}$$

3. If $\texttt{skSubMethod} = 3$,

The method is similar to the previous one, except that the parameter to minimize is the RMS scatter of the sky subtracted spectrum in selected spectral ranges and a quadratic interpolation will be performed to find the zero of the function which gives the RMS scatter at each iteration.

This option can only be used for the emission line component of the sky. Otherwise, the choice of the wavelength ranges in which the RMS is defined becomes difficult, since in principle all strong lines of the solar spectrum could be used; this may lead to some confusion when solar type stars are observed. Therefore, a warning is issued if $skyModel.\texttt{SKCOMP} \neq' EMISSION'$

Here the iteration proceeds as follows:

- Same as 2a
- Three RMS values are computed at the 3 points $diffAmpl - \texttt{skVarAmpl}$, $diffAmpl$ and $diffAmpl + \texttt{skVarAmpl}$ in the following way:

$$RMS_j = \left(\sum_{k=1}^{N_{lines}} \frac{1}{MAX(k) - MIN(k)} \sum_{i=MIN(k)}^{MAX(k)} \left(skySubSp_j(\lambda_i, n) - \overline{skySubSp_j}(k, n)\right)^2 \right)^{1/2}$$

with $j = -1, 0, +1$ and where k numbers the wavelength ranges where telluric emission lines occur (these ranges are defined as corresponding to non-zero values of $skyModel$),while i numbers the wavelengths within each range (but i is not redefined for each range: it remains defined in a unique way for the whole spectrum, varying from 1 to $wlFfSp$.BINWNX). $\overline{skySubSp_j}(k, n)$ is the average value of the jth sky subtracted spectrum in the kth wavelength range.

- A parabola is fitted to the 3 values of the RMS amplitude and its minimum is computed and identified to $diffAmpl(i + 1)$. The same convergence criteria are applied as in the case `skSubMethod` = 2, though with a different value of <`skConv`> (maybe `skConv` < 1, TBD).

This procedure is repeated for each spectrum, and the final sky subtraction is performed as in 2. The keyword $skySubSp$.SKSUB is also defined as in 1. The errors are defined as in 2.

One should notice that the two last options could only be used in the yellow-red, where emission lines are strong enough.

Quality assessment:
For the first option, the RMS scatter of the sky subtracted spectra for sky objects can be computed on the total wavelength range. The cross-correlation with solar spectrum template could be used when justified.
For the two other options the quality of the sky subtraction can be estimated for each spectrum by giving the value of $diffAmpl(m) - diffAmpl(m - 1)$.

3.23.9 Pseudo code

3.24 Conversion to physical flux

Name: **giFluxConv** Section: **Flux calibration** nu: **180**
Author(s)/Date(s): **VC/20000620** Status: **To be validated by all**

Working remarks:
More work TBdone by VC. Not highest priority. Follow remarks by PN.

- *3.20.8. point 1: I do not understand well the sentence "Then an equivalent global transfer function... with a constant intensity of 1"; it seems to me that the figure "1" is wrong.*
- *Point 2: what do you mean exactly by "a special observational procedure to assure a perfect centering of the standard star..."? a detailed description of this procedure would be welcome!*
- *Is the stellar image at the Nasmyth focus really "a simple gaussian"? It seems to me that André pointed out some time that it is precisely not the case, but that the wings of the real PSF are more extended than those of a gaussian. Why not adopt immediately this more realistic shape? (this is a question for André).*
- *MEUSA processing: I do not understand the last equation; has the airmass term disappeared because one assumes that the airmass differences are negligible within a given FoV?; and why does the $collEff$ term still appear? one would expect that it is the same all over the FoV, hence for any star as well as for the standard one, so that this term would cancel out. Same question for IFU processing.*

(PN)

3.24.1 Purpose

The spectra intensities are converted from electrons to physical flux of the object.

3.24.2 Function parameters

Name	Type	Default	Description
`flconvstand`	`logical`	`undef`	Flag for flux conversion using a spectral transfer function obtained with spectrophotometric standard(s) (flconvstand=True) or a model of transmission of atmosphere and telescope (flconvstand=false)

3.24.3 Input frames

Name: *skySubSp* **Type:** `BINSP` **Format:** `nlWbin`
Description: any frame containing all the sky subtracted spectra
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`GIRFTYPE`	DRS	`string`	none	type of frame data
`BINWNX`	DRS	`integer`	none	number of spectral elements on the rebinned S2dFrame
`BINWNY`	DRS	`integer`	none	number of spectra on the rebinned S2dFrame
`BINWSTEP`	DRS	`double`	nm	wavelength step of the linear rebinned spectra
`BINWLO`	DRS	`float`	nm	wavelength of the centre of the first spectral elements
`SLIT`	FGIR_ICS	`integer`	none	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies
`GRATING`	FGIR_ICS	`string`	none	DIC: ESO name for grating unit.
`WLEN0`	FGIR_ICS	`double`	nm	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength
`AIRMASS`	PR_FITS	`double`	none	DIC: Average airmass for the optical axis during the exposure computed for the time while the shutter is open as described in GEN-SPE-ESO-00000-0794.
`ATMPRES`	ASM	`undef`	none	
`ATMRHUM`	ASM	`undef`	none	
`ATMTEMP`	ASM	`undef`	none	
`EXPTIME`	UVES_ICS	`double`	sec	Total exposure time for a single exposure or for the individual exposure within a multiple frame
`LAMP`	DRS	`string`	none	Name of the W Calibration source
`LAMPID`	FP_LP	`undef`	none	
`REFSRC`	MISSING	`string`	none	Type of reference source (LAMP, STAR, SKY, OTHER); same as LAMP KW if type is LAMP
`REFSRCID`	MISSING	`string`	none	Unique ID of reference source (same as LAMPID if type is LAMP)
`REFSRCOM`	MISSING	`string`	none	Comment on the reference source
`REFSRCMULT`	MISSING	`float`	normed	Factor which multiply the reference emissivity of the lamp to take into the account the setup-dependent fraction of the light injected into the fiber.
`TELESCOPE`	TCS	`string`	none	DIC: cmm version number of the integration module (tcsBUILD/ntt), which defines the version of all TCS modules.

Name: *exNffMast* **Type:** EXTSP **Format:** nxExt
Description: extracted NFF spectra
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
GIRFTYPE	DRS	string	none	type of frame data
EXTNX	DRS	integer	none	Number of pixel per spectrum
EXTNS	DRS	integer	none	Number of extracted spectra on the rebinned S2dFrame
SLIT	FGIR_ICS	integer	none	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies
GRATING	FGIR_ICS	string	none	DIC: ESO name for grating unit.
WLEN0	FGIR_ICS	double	nm	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength
EXPTIME	UVES_ICS	double	sec	Total exposure time for a single exposure or for the individual exposure within a multiple frame
LAMP	DRS	string	none	Name of the W Calibration source
LAMPID	FP_LP	undef	none	
REFSRC	MISSING	string	none	Type of reference source (LAMP, STAR, SKY, OTHER); same as LAMP KW if type is LAMP
REFSRCID	MISSING	string	none	Unique ID of reference source (same as LAMPID if type is LAMP)
REFSRCOM	MISSING	string	none	Comment on the reference source
REFSRCMULT	MISSING	float	normed	Factor which multiply the reference emissivity of the lamp to take into the account the setup-dependent fraction of the light injected into the fiber.
TELESCOPE	TCS	string	none	DIC: cmm version number of the integration module (tcsBUILD/ntt), which defines the version of all TCS modules.

3.24.4 Other inputs

Name	Type	Description
atmExtMast	girAtmExtTable	Atmospheric extinction for the atmospheric parameters specified by columns AMODPRES, AMODRHUM, AMODTEMP
Relevant FITS keywords:	GIRTTYPE	
Relevant table columns:		
LAMBDA	float	wavelength
K	float	atmospheric extinction coefficient for wavelength LAMBDA
refFluxMast	girFluxcalTable	Spectral emissivity of the reference source
Relevant FITS keywords:	GIRTTYPE REFSRC REFSRCID	
Relevant table columns:		
LAMBDA	float	wavelength
FLUX	float	spectral transfer function for wavelength LAMBDA
transVltMast	girTransVltTable	telescope + Nasmyth corrector transmission model
Relevant FITS keywords:	GIRTTYPE TELESCOPE	
Relevant table columns:		
LAMBDA	float	wavelength
TRANSVLTN	float	transmission VLT+Nasmyth corrector for wavelength LAMBDA
TRANSADC	float	transmission of the ADC (ARGUS only)

3.24.5 Output frames

Name: *wlFlxSp* **Type:** `BINSP` **Format:** `nlWbin`
Description: frame with all flux calibrated spectra
Relevant FITS keywords:

Name	DIC	Type	Unit	Description
`GIRFTYPE`	DRS	`string`	none	type of frame data
`BINWNX`	DRS	`integer`	none	number of spectral elements on the rebinned S2dFrame
`BINWNY`	DRS	`integer`	none	number of spectra on the rebinned S2dFrame
`FLCONVSTAND`	DRS	`logical`	none	Frflag for flux conversion

3.24.6 Other outputs

none

3.24.7 General description

Since the flatfielded spectra are modulated by a normalized spectrum of the FF calibration lamp, the conversion to physical flux of the intensity value given in any frame of sky subtracted spectra will use a special spectral transfer function rather than a classical response curve. Three method are offered. Two are based on the known emissivity of a reference source, in the first case the continous lamp used for the NFF and in the second case a spectrophotometric standard star observed simultaneously. The third method will use a transfer function provided by other observations of spectrophotometric standard star(s).
Only the first two methods can be used within the DFS pipeline. The model of the telescope and atmosphere transmission is used when the flat-field lamp is the reference source, while the measured spectral transfer function including the atmosphere and the telescope is used if simultaneous observation of a spectrophotometric standard star is available.
The third method, presently limited to the off-line processing, leaves the determination of the adequate transfer function to the responsibility of the observer. This determination could be done by using the function with the second method applied to an observation of a spectrophotometric standard star. One output of this function is the global transfer curve including the atmosphere, the telescope, the Nasmyth corrector, the fibre system, the spectrograph and the detector.
Note that in IFU and ARGUS modes, a reasonable accuracy could be expected on all sources, while in the MEDUSA mode only the surface brightness spectrophotometry where the geometrical loss factor due to the unknown decentering is negligible will reach a reproducible accuracy.

3.24.8 Mathematical description

- if **flconvsour** = **True** then a spectral reference source is chosen among two kinds:

  1. the flat-field lamp
     This is the case when the KW *skySubSp*.`REFSRC` of the sky subtracted spectra frame is FFLAMP. The KW *skySubSp*.`REFSRCID` must contain the identification of the lamp which was used for the flat-fielding. The reference emissivity of the lamp is read from the table `refFluxMast` which matches the *skySubSp*.`REFSRCID` KW. The `refFluxMast:` FLX(λ) are energies per unit area and per Angstrom given by the flat-field lamp at the microlens or ARGUS optical system entrance during 1 sec of time.
     The emissivity curve `refFluxMast:` FLX(λ) is rebinned to the required λ step BINWSTEP in the wavelength domain from BINWL0 to BINWL0+(BINWNX-1)*BINSTEP, which gives $lampEmiss(\lambda)$. The emissivity is multiplied by the REFSRCMULT value to take into account the illumination efficiency at the fibre entrance, the normalisation factor applied to the flat-field frame during the FF Master frame treatment and the exposure time of this FF frame. The REFSRCMULT value is setup-dependent and is taken from the flat-field frame (normaly $exNffMast$) during the FF correction. For the $exNffMast$ frame, the fluxes $flxNff(\lambda)$ collected by a fibre are deduced from the lamp emissivity, the normalisation factor $ffSp$.`FFNORM`, the exposure time of the frame $exNffMast$.`EXPTIME` (the value of the keyword EXPTIME for the NFF frame), an optical scale factor contained in the keyword OPTSCALE (always equal to unity except for the ARGUS mode for which two different scales are allowed onto the microlens array: the first scale has a factor equal to 1 and the second one a factor equal to 0.3/0.52) and the microlens area $miLensSurf$:

$$flxNff(\lambda) = REFSRCMULT \times lampEmiss(\lambda)$$

then:

$$REFSRCMULT = \frac{exNffMast.\mathtt{EXPTIME} \times OPTSCALE \times miLensSurf}{ffSp.\mathtt{FFNORM}}$$

The FF correction assumes a uniform illumination of all fibres by the flat-field lamp (this is obtained only for Positioner flat-field calibration) then no additional effect depending on the fibre position (like VLT and corrector vignetting) is taken into account.

The flat field calibration lamp then plays the role of the standard star but the telescope (including Nasmyth corrector and in the case of the ARGUS mode the ARGUS camera transmission) and atmosphere transmissions should be modeled since the reference source is located at the microlens or ARGUS optical system entrance. Note that the accuracy of this method is limited by the stability of the lamp emissivity and the accuracy of the REFSRCMULT factor, but does not suffer from the loss factor in MEDUSA mode (beside the obvious uncertainty on the actual fraction of the measured source which enters the fibre).

The tables of the atmospheric extinction coefficient (`atmExtMast`) and the telescope+corrector+optical system transmission (`transVltMast`) are read and resampled in the same way as the emissivity. The atmospheric transmission is $e^{-\kappa(\lambda)\times M(z)}$ where $\kappa(\lambda)$ is the extinction coefficient and $M(z)$ the airmass of the field center averaged during the skySubSp frame exposure time (z is the zenithal distance).

Then an equivalent global transfer function is computed, corresponding to a standard source located at unit airmass and giving a flatfield spectrum with a constant intensity of 1. if the source is observed during an exposure time of 1 second:

$$T(\lambda) = \frac{flxNff(\lambda)}{transVlt(\lambda) \times \mathrm{e}^{-\kappa(\lambda)}} = \frac{lampEmiss(\lambda) \times REFSRCMULT}{transVlt(\lambda) \times \mathrm{e}^{-\kappa(\lambda)}}$$

The conversion to physical flux for any sky subtracted spectrum with a fibre running number n is done by multiplying the flatfielded intensities by $T(\lambda)$ after a normalization by the exposure time $skySubSp$.`EXPTIME` and a correction for differential atmospheric extinction between the averaged airmass $M(z)$ of the skySubSp frame and an airmass of 1. Note that the flat-fielded spectrum has already been corrected for vignetting induced by the M2 and M3 mirrors of the VLT and by the Nasmyth corrector.

$$wlFlxSp(\lambda, n) = \frac{skySubSp(\lambda, n) \times T(\lambda) \times \mathrm{e}^{\kappa(\lambda)M(z)}}{EXPTIME \times \mathrm{e}^{\kappa(\lambda)}}$$

It is possible to cancel out some factor in the expression of the physical flux:

$$wlFlxSp(\lambda, n) = \frac{skySubSp(\lambda, n) \times REFSRCMULT \times lampEmiss(\lambda) \times \mathrm{e}^{\kappa(\lambda)M(z)}}{EXPTIME \times transVlt(\lambda)}$$

2. a spectrophotometric standard star observed simultaneously

   This is the case when the KW $skySubSp$.`REFSRC`=STAR. The processing is partly different for the MEDUSA, ARGUS and IFU modes. Since the geometrical loss factor due to the decentering is not exactly known for a star observed in the MEDUSA mode, the spectral transfer function deduced from a standard star can in principle be computed only for the ARGUS and IFU modes.

   Note however that if the first method with the flatfield lamp has to be cross-calibrated even for the MEDUSA mode, a special observational procedure to assure a perfect centering of the standard star in the fibre is always possible and could allow the use of the second method. Then the processing is described first for the MEDUSA mode assuming a perfect centering of the standard star.

   MEDUSA processing The running number $nref$ for the standard star reference spectrum is found through ozpos.OBJECT_TYPE column which is set to 'F' and the ozpos.NAME column which is set to 'REFSRCID'.

   The collection efficiency at the microlens entrance is computed with the full width at half maximum of the PSF obtained at the Nasmyth focus and by assuming that this PSF is well approximated by a simple gaussian. A generic keyword ATMFWHM will give the temporal mean of site seeing value; if this parameter is not recorded at the Nasmyth focus level, it can be multiplied by a correction factor to give the FWHM of the PSF through the atmosphere, telescope and Nasmyth corrector. The collection efficiency is then :

$$collEff = \frac{2.355^2}{2\pi FWHM^2} \times \int_{miLensSurf} e^{-2.355^2 \times \frac{x^2+y^2}{2FWHM^2}}$$

The standard star emissivity $starEmiss(\lambda)$ is read from the table `refFlux` which matches the REFSRCID and is processed in the same way as for the flat-field lamp.
The spectral transfer function is computed as the quotient of the star emissivity corrected by the collection efficiency to the flatfielded reference spectrum $nref$ normalized by its exposure time EXPTIME and corrected for differential atmospheric extinction:

$$T(\lambda) = \frac{starEmiss(\lambda) \times collEff \times EXPTIME \times \mathrm{e}^{\kappa(\lambda)}}{skySubSp(\lambda, nref) \times \mathrm{e}^{\kappa(\lambda)M(z)}}$$

The formula giving the conversion to physical flux for any sky subtracted spectrum n of the same frame leads to a final result:

$$wlFlxSp(\lambda, n) = \frac{skySubSp(\lambda, n)}{\mathrm{e}^{\kappa(\lambda)}} \times \frac{T(\lambda) \times \mathrm{e}^{\kappa(\lambda)M(z)}}{EXPTIME} = \frac{skySubSp(\lambda, n) \times starEmiss(\lambda) \times collEff}{skySubSp(\lambda, nref)}$$

IFU processing: For the IFU mode the 20 running numbers for the fibres which are contained in the IFU bundle which was positioned at the R and THETA positions given for the sphectrophotometric standard star will be found also through the ozpos.OBJECT_TYPE column which should be set to 'F'.
The collection efficiency for the stellar flux is assumed to be close to 1, because all 20 spectra of the relevant IFU are added to compute the flux, and the pointing is considered accurate enough that the star can be considered as properly centred.
The last steps of the processing are similar to the MEDUSA one (with $skySubSp(\lambda, nref)$ replaced by the sum of the 20 spectra of the IFU centred on the standard star), except that the physical flux will be normalized to 1 arcsec square aperture by using the value of the microlens area **miLensSurf**. More explicitly:

$$wlFlxSp(\lambda, n) = \frac{skySubSp(\lambda, n) \times \frac{1}{0.52^2} \times starEmiss(\lambda) \times collEff}{\sum_{i=nref1}^{nref20} skySubSp(\lambda, i)}$$

where 0.52 is the side of an IFU microlens in arcsec.

ARGUS processing: The situation is the same as for IFU, except that the fibres collecting the flux of the standard star have to be selected. Thanks to the pointing accuracy of the VLT, one may consider that it will be enough to add the signals of all fibres lying within a given radius (TBD, but typically 1.5 arcsec) of the nominal position of the standard star (coordinates taken from the ozpoz table).

- if **flconvsour** = **False** then an external spectral transfer function derived from spectrophotometric standard(s) is used.

  All the observed flat-fielded spectra are multiplied by the spectral transfer function RABS after rebinning to the wavelength sampling of the *skySubSp* frame. The correction for differential atmospheric extinction is taken into account after the atmospheric extinction coefficient $\kappa(\lambda)$ was rebinned as well:

$$wlFlxSp(\lambda, n) = \frac{skySubSp(\lambda, n) \times RABS(\lambda) \times \mathrm{e}^{\kappa(\lambda)M(z)}}{EXPTIME \times \mathrm{e}^{\kappa(\lambda)}}$$

3.24.9 Pseudo code

4 Architecture-Recipes

4.1 Introductory remarks

Recipes are structures describing a complete Data Reduction of one or several input frame(s) of a given type. A recipe defines in what order and with what parameters the modules are called in the execution pipeline as well as where all the necessary parameters and data are coming from. The recipes are implemented as shell scripts calling the DRS Modules and insuring the housekeeping and transmission of all intermediate data and parameters.
Following elements are used to design BLDRS recipes:

- Recipe name

  This is a full name of the recipe

- Recipe purpose

  Is a short description of the recipe in general terms.

- Recipe control parameters (RCP)

  In order to make the automated processing possible and efficient, all options and controls of all functions are fully parameterized. The set of recipe control parameters is generated automatically using the function definitions. The flow control parameters, controlling the execution of the recipe itself such as verbosity, level of interactivity and definition of break-points are not given for individual recipes.

  All control parameters are defaulted in an initial parameter files. Two level of default may exist for a particular recipe. The compulsory general-level default which defines all control parameters for all existing functions, and recipe-level defaults whose defaults (if any) supersede the general default for a particular recipe.

- Input Frames and data

  Same description as for function (see 3) applies. For brevity the *other data* are referenced here as *data*.

- Output Frames and data

  These are data written to files. Same formalism as for input data applies.

- Recipe detailed description

  This is a complete verbal description what the recipe is doing. In general it includes the following elements.

  - A brief description of the context of the use (assumed conditions, category, ...)
  - A brief description of the output
  - Discussion of the possible errors and failures and the measurement of the quality

- Development notes

4.2 Recipe generation

Since all informations necessary for the execution of each module are transmitted via FITS keyword written in the header of the input data-frames, the basic chaining of modules making up the recipe is straightforward and does not require any specific tool. In practice, however, a high-level of the data-flow checking and verification is not only desirable, but probably a necessary pre-requisite to efficient recipe implementation. This is one of the reasons why we implemented the function definition as a data-base. The recipe generation made using this data-base made all possible formal verification when parsing the list of functions to be chained in a recipe, mainly:

- Verification of input data existence, compatibility and completeness for each module
- Verification of FITS KW existence, compatibility and completeness for all data-frame
- Verification of input/output match between modules
- Verification of control parameters
- Produces the list of all input to the recipe

- Produces the list of all output to the recipe (first guess of what the data-product could be)
- Produces the list of all internal output/input (outputs of modules used only for the internal purpose of the recipe by other modules)

4.2.1 Evaluation of data size

Up to five parameters gives the information about the data volume:

N	Number of frames (this usually 1)
Med	Size in MEDUSA mode used within a single recipe
TMed	Total size that of data of fully calibrated instrument (should be available in the Archive). This number differs from *Med* for calibration data only.
Arg	Size in ARGUS and IFU mode used within a single recipe
TArg	Total size that of data of fully calibrated instrument (should be available in the Archive). This number differs from *Med* for calibration data only.

Totals (in Mbytes) for datasets are given for each parameter. Note that when same data are used several times (mostly on input), the total given is smaller that the arithmetic sum of the column and the volume is rounded to 1MB so that some small files appear to have size "0". Following datasets are described:

1. Input Data Set

   This reflects the volume of the data extracted from the Archive prior to run the recipe

2. Internal Input/Output Data Set

   Added to the volume under item 1, this gives the lowest estimate of the necessary scratch space to run the recipe

3. Output Data Set

   This is the volume to be saved in the Archive

In the present design we consider these data-base related tools as internal to the development team and therefore not a part of deliverables and not subject to any VLT compliant constraint. It is still evolving and the present description is by no means complete. Note that though the estimation of data volume is done for frames only it is representative since the size of tables is in general much smaller.
In following section, only few standard recipes are given. Most of other recipes will be written once the system (including the recipe generation) is fully validated. Also, only frame analysis is shown.

5 Data Reduction Recipes

5.1 GiStand

History:	Generated by blecha at 20000926.103851 Updated automaticaly by blecha at 20000926.111845 Updated automaticaly by blecha at 20000926.113701 Updated automaticaly by blecha at 20010103.154438
Description:	This is a complete standard processing, including the build of the Scatered Light Model and Sky subtraction The FFing is done on extracted spectra
Compiled at/by:	20010103.1544 blecha

5.1.1 GiStand - Calls to Functions

Function	Purpose
giGetRemoveBias	The bias and the read-out noise are obtained as the average value and sigma over the overscan areas. The bias value is subtracted and the overscan lanes are trimmed out.
giSubtractDark	The scaled (with exposure time) current standard mean dark is subtracted from the current image.

Function	Purpose
giEstimatePix	Apply a median estimation to the input frame over the closest neighboring good pixels. This step is necessary for CRH removal
giDetectCosmicS	The detection of CRH is made after pixels values estimation in a frame and is used to upgrade the bad pixel map.
giReplaceFlagPix	Replace the bad pixels by their estimate in an image (could be the same image).
giAdjustSL	Adjust a model of scattered light to the inter-spectra region.
giLocalSpectra	Determine the centroid and the width of each spectral bin of each spectrum using the preliminary localization. A new current localization is defined through the LSQ fit of the optical model to positions and widths found.
giExtractSpectra	The extraction implements three options: summation along a virtual slit, classical Horne's and PSF fitting methods. The local error and the residual background are also computed.
giFlatSpectra	The extracted spectra are corrected for spectrograph and detector response (including high spatial frequencies), but they remain modulated by the spectrum of the FF calibration lamp.
giGetWaveSolution	The wavelength solution is obtained from a separate (all fibres exposed to calibration lamp - SEWC) or simultaneous (5 fibres exposed to calibration lamp - SIWC) exposure
giRebinSpectra	Carry out rebinning of extracted spectra so that the result is equally spaced in λ
giSplitSky	The sky spectra are separated into two components
giNormalizeSky	The total flux normalization is done for sky spectra
giModelSky	The spatial sky light distribution is optionally modelized before a model of sky spectrum is build depending on the PSF width in the dispersion direction
giSubtractSky	The modelized sky spectra are subtracted in the wavelength space.

5.1.2 GiStand - Control parameters

All input parameters

Function	Parameter	Value	Type	Description
giGetRemoveBias	bClipSigma	2.5	float	multiple of σ (σ-clipping)
	bClipNiter	2	integer	number of iterations (σ-clipping)
	bClipMfrac	0.9	float	min fraction of points accepted/total (σ-clipping)
	bMethod	ZMASTER + UNIFORM	string	Method used, one of: Uniform,Plane,Master,Zmaster
giSubtractDark	darkThreshold	1	float	dark threshold value in e/pixel
	darkMethod	UNIFORM	string	one of 'UNIFORM' 'MASTER'
giEstimatePix	badPixMfrac	0.3	float	min fraction of neighboring bad pixels accepted/total
	maxRadius	2.	float	Maximum radius of the neighboring area (pixels)
	metric	00	integer	XY metrics indicating how the averaging area spreads 11-X,Y; 10-X only 01-Y only 00-LAMP KW driven
giDetectCosmicS	crhsSigMul	4.5	float	multiple of σ used as threshold for local excess determination
	crhsAlertVal	10.0	float	number of cosmic ray hits used as threshold for an alert
	crhsCrhImg	FALSE	logical	if set we compute the image of CRH photons (crhsCount)
	crhBadPixFlag	TBD	integer	flag used to indicate CRH hits
	badPixFlags	eBadPixFlag	integer	flags to be used with associated *estimateImage*
giReplaceFlagPix	badPixFlag	"ALL"	integer	type of bad pixels to be replaced in the input image

All input parameters

Function	Parameter	Value	Type	Description
giAdjustSL	slPhffCor	FALSE	logical	option for correcting the input frame from the photometric flat field
	slModel	"POLYNOM"	string	option indicating if the model is a polynomial ("POLYNOM") or a polynomial fraction ("POLYFRAC")
	slPolyDeg1X	5	integer	degree of the numerator polynomial of the model
	slPolyDeg2X	0	integer	degree of the denominator polynomial of the model
	slPolyDeg1Y	5	integer	degree of the numerator polynomial of the model
	slPolyDeg2Y	0	integer	degree of the denominator polynomial of the model
	contamW	4	integer	extra width parameter (in pixels) to cut out the part of inter-spectra region used for SL determination
giLocalSpectra	lClipSigma	4.5	float	multiple of σ (σ-clipping)
	lClipNiter	10	integer	number of iterations (σ-clipping)
	lClipMfrac	0.2	float	min fraction of points accepted/total (σ-clipping)
	noiseMult	10	double	noise detector multiple
	lPolyDeg	5	integer	degree of the polynomial fit of the localization mask
	lComplete	0	integer	flag indicating if the localization is carried out over all spectra (0: according to the type of frames, 1: YES, 2: NO)
	lAddWy	0.5	double	additional width to widen the fitted localization mask
giExtractSpectra	extMethod	3	integer	Extraction method: 1=simple summation; 2=Horne's method; 3=LSQ fit
	extPsfBkgd	0.04	double	PSF background tolerance for CRH detection in Horne's extraction
	extPsfAmpl	0.10	double	PSF amplitude tolerance for CRH detection in Horne's extraction, or for LSQ fit extraction
	extSmooBg	true	logical	flag for smoothing the background in option extMethod=3
	extBkgXdeg	5	integer	polynomial degree of the background model for extMethod=3 and extSmooBg=TRUE (smoothing in the x direction)
	extBkgYdeg	15	integer	polynomial degree of the background model (y direction)
	extWidthMult	1.	float	half-length of the extraction virtual slit, expressed in units of PSF width
	extClipSigma	4.5	float	multiple of σ (σ-clipping)
	extClipNiter	10	integer	number of iterations (σ-clipping)
	extClipMfrac	0.2	float	min fraction of points accepted/total (σ-clipping)
	extPsfPos	0.3	double	PSF position tolerance for CRH detection in Horne's extraction
	extPsfWid1	0.03	double	PSF width1 tolerance for CRH detection in Horne's extraction
	extPsfWid2	0.03	double	PSF width2 tolerance for CRH detection in Horne's extraction
	extPsfWid3	0.03	double	PSF width3 tolerance for CRH detection in Horne's extraction

All input parameters

Function	Parameter	Value	Type	Description
giGetWaveSolution	wsOptSol	TRUE	logical	flag for fitting the optical parameters (wsOptSol=TRUE) or only the residuals relative to the standard optical model (wsOptSol=FALSE)
	widClipMfrac	0.7	float	min fraction of accepted/total lines (width in x)
	wsXwidXDeg	3	integer	polynomial degree along the x axis for x-width modeling
	wsXwidYDeg	2	integer	polynomial degree along the y axis for x-width modeling
	wsWlenXDeg	3	integer	polynomial degree along the x axis for wavelength solution
	wsWlenYDeg	2	integer	polynomial degree along the y axis for wavelength solution
	wsLineType	TRUE	logical	flag for use of ThArNe lines (wslineType=TRUE) or of telluric emission lines (lineType=FALSE)
	wsWindow	10.	double	width in pixels of the window for line detection and fit
	wslineThres	10.	float	multiple of σ above which a pixel value is considered to betray the presence of a calibration (emission) line
	wsClipSigma	3.0	float	multiple of σ (σ-clipping)
	wsClipNiter	10	integer	number of iterations (σ-clipping)
	wsClipMfrac	0.6	float	min fraction of accepted/total lines (σ-clipping)
	widClipSigma	3.0	float	multiple of σ above which clipping occurs (width in x)
	widClipNiter	10	integer	number of σ iterations (width in x)
giRebinSpectra	binWStep		double	wavelength step used in rebinning (= 0, if rebinning is carried out in logarithmic wavelength space)
	binWRange		integer	flag indicating how is derived the spectral range that is rebinned
	binWStepLog		double	logarithmic wavelength step used in rebinning (= 0, if rebinning is carried out in wavelength space)
giSplitSky	ssWidMul	6.0	float	minimum full width of sky emission lines in terms of the PSF width parameter in wavelength (multiple of sigma in case of a gaussian PSF)
giNormalizeSky	skWlenMin		float	minimum of the wavelength range
	skWlenMax		float	maximum of the wavelength range
giModelSky	skModel	TRUE	logical	Flag for no spatial distribution modeling (skModel=FALSE) or spatial distribution modeling (skModel=TRUE)
	skWClipNiter	1	integer	number of iterations (σ-clipping of intensity vs PSF width)
	skWClipMfrac	0.9	float	min fraction of points accepted/total (σ-clipping of intensity vs PSF width)
	skClipSigmamax	2.5	float	multiple of σ for the upper σ-clipping limit
	skClipSigmamin	2.5	double	multiple of σ for the lower σ-clipping limit
	skClipNiter	1	integer	number of iterations (σ-clipping)
	skClipMfrac	0.9	float	min fraction of points accepted/total (σ-clipping)

All input parameters

Function	Parameter	Value	Type	Description
	skXdegmax	2	integer	max polynomial degree for fitting sky spatial distribution in X
	skYdegmax	2	integer	max polynomial degree for fitting sky spatial distribution in Y
	skWdegmax	2	integer	max polynomial degree for fitting sky intensity versus PSF width
	skWClipSigma	3	float	multiple of σ for the σ-clipping limit in the fit of the sky intensity versus PSF width
giSubtractSky	skSubMethod	1	integer	1=direct subtraction of sky model, 2=amplitude adjustment for sky model intensity with cross-correlation method, 3=amplitude adjustment for sky model intensity with error minimization method
	skVarAmpl	0.1	float	Initial increment of the variation of the sky amplitude for skSubMethod=2 or 3
	skFracAmpl	0.1	float	Maximum fraction of the relative difference in the sky amplitude, for skSubMethod=2 or 3, stopping iterations as soon as it is reached
	skConv	1	float	Convergence parameter defining skVarAmpl at each iteration

5.1.3 GiStand - Data Flow Summary

Frames - Data Flow

Function	In	Framename	Format	Type	Source	On
giGetRemoveBias	I1	rawFrame	xyRaw	RAWIMG		
	I2	biasMast	xyPrep	BIASIMG		
	I3	badPixMast	xyPrep	BADPIX		
	O1	bSubImage	xyPrep	BRMIMG		
giSubtractDark	I1	bSubImage	xyPrep	BRMIMG	giGetRemoveBias	O1
	I2	darkMast	xyPrep	DARKIMG		
	I3	badPixMast	xyPrep	BADPIX		
	O1	dbSubImage	xyPrep	BDRMIMG		
giEstimatePix	I1	dbSubImage	xyPrep	PREPIMG	giSubtractDark	O1
	I2	locYMast	nxExt	LOCY		
	I3	locWyMast	nxExt	LOCWY		
	I4	badPixMap	xyPrep	BADPIX		
	O1	estimateImage	xyPrep	ESTIMG		
	O2	estimateSigmaImage	xyPrep	ESTSIGM		
	O3	estimateNpixels	xyPrep	ESTNPIX		
	O4	estBadPixMap	xyPrep	BADPIX		
giDetectCosmicS	I1	dbSubImage	xyPrep	BDRMIMG	giSubtractDark	O1
	I2	estimateImage	xyPrep	ESTIMG	giEstimatePix	O1
	I3	estimateSigmaImage	xyPrep	ESTSIGM	giEstimatePix	O2
	I4	estBadPixMap	xyPrep	BADPIX	giEstimatePix	O4
	O1	crhsCount	xyPrep	CRHIMG		
	O2	crhBadPixMap	xyPrep	BADPIX		
giReplaceFlagPix	I1	dbSubImage	xyPrep	PREPIMG	giSubtractDark	O1
	I2	estimateImage	xyPrep	ESTIMG	giEstimatePix	O1
	I3	badPixMap	xyPrep	BADPIX		
	I4	estBadPixMap	xyPrep	BADPIX	giEstimatePix	O4
	O1	repPixImage	xyPrep	REPIMG		
	O2	repBadPixMap	xyPrep	BADPIX		
giAdjustSL	I1	repPixImage	xyPrep	REPIMG	giReplaceFlagPix	O1

Frames - Data Flow

Function	In	Framename	Format	Type	Source	On
	I2	phffImageMast	xyPrep	PHFFIMG		
	I3	locYMast	nxExt	LOCY		
	I4	locWyMast	nxExt	LOCWY		
	I5	repBadPixMap	xyPrep	BADPIX	giReplaceFlagPix	O2
	O1	slImage	xyPrep	SLIMG		
giLocalSpectra	I1	dbSubImage	xyPrep	BDRMIMG	giSubtractDark	O1
	I2	locYMast	nxExt	LOCY		
	I3	locWyMast	nxExt	LOCWY		
	I4	badPixMap	xyPrep	BADPIX		
	O1	locY	nxExt	LOCY		
	O2	locWy	nxExt	LOCWY		
giExtractSpectra	I1	dbSubImage	xyPrep	BDRMIMG	giSubtractDark	O1
	I2	slImage	xyPrep	SLIMG	giAdjustSL	O1
	I3	locY	nxExt	LOCY	giLocalSpectra	O1
	I4	locWy	nxExt	LOCWY	giLocalSpectra	O2
	I5	curBadPixMap	xyPrep	BADPIX		
	O1	exSp	nxExt	EXTSP		
	O2	exSpErr	nxExt	EXTERRS		
	O3	exSpNpixels	nxExt	EXTNPIX		
	O4	exBadPixMap	nxExt	EXTBPM		
	O5	exBack	xPolyY	EXTBKG		
	O6	cleanImage	xyPrep	ESTIMG		
	O10	clbadPixMap	xyPrep	BADPIX		
giFlatSpectra	I1	exSp	nxExt	EXTSP	giExtractSpectra	O1
	I2	exSpErr	nxExt	EXTERRS	giExtractSpectra	O2
	I3	exNffMast	nxExt	EXTSP		
	I4	exNffErrMast	nxExt	EXTERRS		
	O1	ffSp	nxExt	EXTSP		
	O2	ffSpErr	nxExt	EXTERRS		
giGetWaveSolution	I1	exSp	nxExt	EXTSP	giExtractSpectra	O1
	I2	exSpErr	nxExt	EXTERRS	giExtractSpectra	O2
	I3	locY	nxExt	LOCY	giLocalSpectra	O1
	O1	fitWxl	nxExt	FITCOEF		
giRebinSpectra	I1	ffSp	nxExt	EXTSP	giFlatSpectra	O1
	I2	ffSpErr	nxExt	EXTERRS	giFlatSpectra	O2
	I3	fitWxl	nxExt	FITCOEF	giGetWaveSolution	O1
	O1	wlFfSp	nlWbin	BINSP		
	O2	wlFfSpErr	nlWbin	BINERRS		
	O3	psflwidth	nlWbin	BINSP		
giSplitSky	I1	wlFfSp	nlWbin	BINSP	giRebinSpectra	O1
	I2	wlFfSpErr	nlWbin	BINERRS	giRebinSpectra	O2
	I3	psflwidth	nlWbin	BINSP	giRebinSpectra	O3
	O1	wlFfSkyComp	nlWbin	BINSP		
	O2	wlFfSkyCompErr	nlWbin	BINSP		
giNormalizeSky	I1	wlFfSp	nlWbin	BINSP	giRebinSpectra	O1
	I2	wlFfSpErr	nlWbin	BINSP	giRebinSpectra	O2
	O1	normSky	nlWbin	BINSP		
	O2	normSkyErr	nlWbin	BINSP		
giModelSky	I1	normSky	nlWbin	BINSP	giNormalizeSky	O1
	I2	normSkyErr	nlWbin	BINSP	giNormalizeSky	O2
	I3	psflwidth	nlWbin	BINSP	giRebinSpectra	O3
	O1	skyModel	nlWbin	BINSP		
	O2	skyModelErr	nlWbin	BINSP		
giSubtractSky	I1	wlFfSp	nlWbin	BINSP	giRebinSpectra	O1
	I2	wlFfSpErr	nlWbin	BINSP	giRebinSpectra	O2
	I3	skyModel	nlWbin	BINSP	giModelSky	O1

Frames - Data Flow

Function	In	Framename	Format	Type	Source	On
	I4	skyModelErr	nlWbin	BINSP	giModelSky	O2
	O1	skySubSp	nlWbin	BINSP		
	O2	skySubSpErr	nlWbin	BINSP		

Other Data - Data Flow

Function	In	Dataname	Type	Source	On
giGetRemoveBias	I1	biasLimitsMast	BiasBoxTable		
giLocalSpectra	I1	psfParamsMast	PsfTable		
	I2	ozpoz	OzpozTable		
	O1	psfParams	PsfTable		
giExtractSpectra	I1	psfParams	PsfTable	giLocalSpectra	O1
giGetWaveSolution	I1	linesWlampMast	LineTable		
	I2	linesSkyMast	LineTable		
	I3	waveSolutionMast	WavCoefTable		
	I4	psfXwidthMast	PsfXwidTable		
	I5	slitGeometryMast	SlitGeoTable		
	O1	waveSolution	WavCoefTable		
	O2	cleanlineTableMW	LineTable		
	O3	cleanlineTableMWSky	LineTable		
	O4	psfXwidth	PsfXwidTable		
giRebinSpectra	I1	waveSolution	WavCoefTable	giGetWaveSolution	O1
giSplitSky	I1	ozpoz	OzpozTable		
	I2	numSky	SkyTable		
	I3	linesSkyMast	LineTable		
giNormalizeSky	I1	ozpoz	OzpozTable		
	I2	numSky	SkyTable		
	O1	intSky	IntSkyTable		
giModelSky	I1	intSky	IntSkyTable	giNormalizeSky	O1
	I2	ozpoz	OzpozTable		
	O1	sNorm	SkyCoefTable		
	O2	qualSkyMod	SkyMomTable		

5.1.4 GiStand - Required Input Data Set

Frames - Input Data Set (size in MB)

Function	In	Framename	Frm	Type	N	Med	Arg	TMed	TArg
giGetRemoveBias	I1	rawFrame	xyRaw	RAWIMG	1	37	37	37	37
	I2	biasMast	xyPrep	BIASIMG	1	33	33	33	33
	I3	badPixMast	xyPrep	BADPIX	1	33	33	33	33
giSubtractDark	I2	darkMast	xyPrep	DARKIMG	1	33	33	33	33
giAdjustSL	I2	phffImageMast	xyPrep	PHFFIMG	1	33	33	33	33
	I3	locYMast	nxExt	LOCY	1	2	5	2	5
	I4	locWyMast	nxExt	LOCWY	1	2	5	2	5
giLocalSpectra	I4	badPixMap	xyPrep	BADPIX	1	33	33	33	33
giExtractSpectra	I5	curBadPixMap	xyPrep	BADPIX	1	33	33	33	33
giFlatSpectra	I3	exNffMast	nxExt	EXTSP	1	2	5	2	5
	I4	exNffErrMast	nxExt	EXTERRS	1	2	5	2	5
Total						243	255	243	255

Frames - Input Data Set Description

Function	Framename	Framedescription
giGetRemoveBias	rawFrame	raw frame with bias prescans and overscans included
	biasMast	master bias
	badPixMast	Optional bias bad pixel map with active flag 'bBadPixFlag'
giSubtractDark	darkMast	dark frame in electron/second units (Master Dark Frame)
giAdjustSL	phffImageMast	Master photometric flat-field calibration image for the current setup. Its use is optional
	locYMast	standard localization giving the spectrum center along y at each x for the current setup
	locWyMast	standard width of the NFF spectra (measured along the y axis) for the current setup
giLocalSpectra	badPixMap	Current *dbSubImage* bad pixel map
giExtractSpectra	curBadPixMap	Current *dbSubImage* bad pixel map
giFlatSpectra	exNffMast	extracted NFF spectra (Master NFF); It is assumed without bad pixels. This frame is set-up dependent and the selection is made on KW's SLIT:GRATING:WLEN0
	exNffErrMast	corresponding extracted NFF spectra errors (Error of Master NFF); same condition as for the NFF applies

Other Data - Input Data Set (size in MB)

Function	In	Dataname	Type	Med	Arg	TMed	TArg
giGetRemoveBias	I1	biasLimitsMast	BiasBoxTable	0	0	0	0
giLocalSpectra	I1	psfParamsMast	PsfTable	0.1	0.25	6.35	22.27
giGetWaveSolution	I1	linesWlampMast	LineTable	0	0	0.05	0.05
	I3	waveSolutionMast	WavCoefTable	0.01	0.02	0.63	2.22
	I4	psfXwidthMast	PsfXwidTable	0.05	0.12	3.17	11.13
	I5	slitGeometryMast	SlitGeoTable	0	0	0.01	0.06
giSplitSky	I2	numSky	SkyTable	0	0	0	0
	I3	linesSkyMast	LineTable	0	0	0.05	0.05
giNormalizeSky	I1	ozpoz	OzpozTable	0.02	0.02	0.02	0.02
Total				0.18	0.41	10.28	35.8

Other data - Input Set Description

Function	Dataname	Datadescription
giGetRemoveBias	biasLimitsMast	Definition of rectangular areas on CCD overscan and prescan used for bias computation
giLocalSpectra	psfParamsMast	Initial n-coefficient describing the shape of the PSF for any position [x,y], SLIT dependent
giGetWaveSolution	linesWlampMast	table of $Nlines$ line wavelengths of the calibration spectrum, depends on SETUP
	waveSolutionMast	table containing the standard $NparWv$ parameters of the analytical model of the wavelength solution for all spectra
	psfXwidthMast	standard $NparPSFw$ parameters of the adjusted analytical model of the PSF width in the x (λ) direction (in pixels)
	slitGeometryMast	prameters used to compute the initial optical model
giSplitSky	numSky	optional (off-line mode only) single column table with the $Nsky$ fibre running numbers of spectra to be normalized.
	linesSkyMast	table of $Nlines$ line wavelengths of the telluric emission lines
giNormalizeSky	ozpoz	Positionner Binary Table; this is in fact not an input, but part of the inframe1

5.1.5 GiStand - Internal Input/Output Data

Frames - Internal Input/Output Data Set

Function	In	Framename	Frm	Type	Source	On	N	Med	Arg
giSubtractDark	I1	bSubImage	xyPrep	BRMIMG	giGetRemoveBias	O1	1	33	33
giEstimatePix	I1	dbSubImage	xyPrep	PREPIMG	giSubtractDark	O1	1	33	33
giDetectCosmicS	I1	dbSubImage	xyPrep	BDRMIMG	giSubtractDark	O1	1	33	33
	I2	estimateImage	xyPrep	ESTIMG	giEstimatePix	O1	1	33	33
	I3	estimateSigmaImage	xyPrep	ESTSIGM	giEstimatePix	O2	1	33	33
	I4	estBadPixMap	xyPrep	BADPIX	giEstimatePix	O4	1	33	33
giReplaceFlagPix	I1	dbSubImage	xyPrep	PREPIMG	giSubtractDark	O1	1	33	33
	I2	estimateImage	xyPrep	ESTIMG	giEstimatePix	O1	1	33	33
	I4	estBadPixMap	xyPrep	BADPIX	giEstimatePix	O4	1	33	33
giAdjustSL	I1	repPixImage	xyPrep	REPIMG	giReplaceFlagPix	O1	1	33	33
	I5	repBadPixMap	xyPrep	BADPIX	giReplaceFlagPix	O2	1	33	33
giLocalSpectra	I1	dbSubImage	xyPrep	BDRMIMG	giSubtractDark	O1	1	33	33
giExtractSpectra	I1	dbSubImage	xyPrep	BDRMIMG	giSubtractDark	O1	1	33	33
	I2	slImage	xyPrep	SLIMG	giAdjustSL	O1	1	33	33
	I3	locY	nxExt	LOCY	giLocalSpectra	O1	1	2	5
	I4	locWy	nxExt	LOCWY	giLocalSpectra	O2	1	2	5
giFlatSpectra	I1	exSp	nxExt	EXTSP	giExtractSpectra	O1	1	2	5
	I2	exSpErr	nxExt	EXTERRS	giExtractSpectra	O2	1	2	5
giGetWaveSolution	I1	exSp	nxExt	EXTSP	giExtractSpectra	O1	1	2	5
	I2	exSpErr	nxExt	EXTERRS	giExtractSpectra	O2	1	2	5
	I3	locY	nxExt	LOCY	giLocalSpectra	O1	1	2	5
giRebinSpectra	I1	ffSp	nxExt	EXTSP	giFlatSpectra	O1	1	2	5
	I2	ffSpErr	nxExt	EXTERRS	giFlatSpectra	O2	1	2	5
	I3	fitWxl	nxExt	FITCOEF	giGetWaveSolution	O1	1	2	5
giSplitSky	I1	wlFfSp	nlWbin	BINSP	giRebinSpectra	O1	1	5	12
	I2	wlFfSpErr	nlWbin	BINERRS	giRebinSpectra	O2	1	5	12
	I3	psflwidth	nlWbin	BINSP	giRebinSpectra	O3	1	5	12
giNormalizeSky	I1	wlFfSp	nlWbin	BINSP	giRebinSpectra	O1	1	5	12
	I2	wlFfSpErr	nlWbin	BINSP	giRebinSpectra	O2	1	5	12
giModelSky	I1	normSky	nlWbin	BINSP	giNormalizeSky	O1	1	5	12
	I2	normSkyErr	nlWbin	BINSP	giNormalizeSky	O2	1	5	12
	I3	psflwidth	nlWbin	BINSP	giRebinSpectra	O3	1	5	12
giSubtractSky	I1	wlFfSp	nlWbin	BINSP	giRebinSpectra	O1	1	5	12
	I2	wlFfSpErr	nlWbin	BINSP	giRebinSpectra	O2	1	5	12
	I3	skyModel	nlWbin	BINSP	giModelSky	O1	1	5	12
	I4	skyModelErr	nlWbin	BINSP	giModelSky	O2	1	5	12
Total							22	313	383

Other Data - Internal Input/Output Data Set

Function	In	Dataname	Type	Source	On	Med	Arg
giExtractSpectra	I1	psfParams	PsfTable	giLocalSpectra	O1	0.1	0.25
giRebinSpectra	I1	waveSolution	WavCoefTable	giGetWaveSolution	O1	0.01	0.02
giModelSky	I1	intSky	IntSkyTable	giNormalizeSky	O1	0	0
Total						0.11	0.27

5.1.6 GiStand - Net Output (Possible Output Data Set)

Frames - all Outputs (Possible Output Data Set)

Function	On	Framename	Format	Type	Dest
giGetRemoveBias	O1	bSubImage	xyPrep	BRMIMG	Garbage(def)
giSubtractDark	O1	dbSubImage	xyPrep	BDRMIMG	Garbage(def)
giEstimatePix	O1	estimateImage	xyPrep	ESTIMG	Garbage(def)
	O2	estimateSigmaImage	xyPrep	ESTSIGM	Garbage(def)
	O3	estimateNpixels	xyPrep	ESTNPIX	Data Product
	O4	estBadPixMap	xyPrep	BADPIX	Garbage(def)
giDetectCosmicS	O1	crhsCount	xyPrep	CRHIMG	Data Product
	O2	crhBadPixMap	xyPrep	BADPIX	Data Product
giReplaceFlagPix	O1	repPixImage	xyPrep	REPIMG	Garbage(def)
	O2	repBadPixMap	xyPrep	BADPIX	Garbage(def)
giAdjustSL	O1	slImage	xyPrep	SLIMG	Garbage(def)
giLocalSpectra	O1	locY	nxExt	LOCY	Garbage(def)
	O2	locWy	nxExt	LOCWY	Garbage(def)
giExtractSpectra	O1	exSp	nxExt	EXTSP	Garbage(def)
	O2	exSpErr	nxExt	EXTERRS	Garbage(def)
	O3	exSpNpixels	nxExt	EXTNPIX	Data Product
	O4	exBadPixMap	nxExt	EXTBPM	Data Product
	O5	exBack	xPolyY	EXTBKG	Data Product
	O6	cleanImage	xyPrep	ESTIMG	Data Product
	O10	clbadPixMap	xyPrep	BADPIX	Data Product
giFlatSpectra	O1	ffSp	nxExt	EXTSP	Garbage(def)
	O2	ffSpErr	nxExt	EXTERRS	Garbage(def)
giGetWaveSolution	O1	fitWxl	nxExt	FITCOEF	Garbage(def)
giRebinSpectra	O1	wlFfSp	nlWbin	BINSP	Garbage(def)
	O2	wlFfSpErr	nlWbin	BINERRS	Garbage(def)
	O3	psflwidth	nlWbin	BINSP	Garbage(def)
giSplitSky	O1	wlFfSkyComp	nlWbin	BINSP	Data Product
	O2	wlFfSkyCompErr	nlWbin	BINSP	Data Product
giNormalizeSky	O1	normSky	nlWbin	BINSP	Garbage(def)
	O2	normSkyErr	nlWbin	BINSP	Garbage(def)
giModelSky	O1	skyModel	nlWbin	BINSP	Garbage(def)
	O2	skyModelErr	nlWbin	BINSP	Garbage(def)
giSubtractSky	O1	skySubSp	nlWbin	BINSP	Data Product
	O2	skySubSpErr	nlWbin	BINSP	Data Product

Other Data - all Outputs (Possible Output Data Set)

Function	On	Dataname	Type	Dest
giLocalSpectra	O1	psfParams	PsfTable	Garbage(def)
giGetWaveSolution	O1	waveSolution	WavCoefTable	Garbage(def)
	O2	cleanlineTableMW	LineTable	Data Product
	O3	cleanlineTableMWSky	LineTable	Data Product
	O4	psfXwidth	PsfXwidTable	Data Product
giNormalizeSky	O1	intSky	IntSkyTable	Garbage(def)
giModelSky	O1	sNorm	SkyCoefTable	Data Product
	O2	qualSkyMod	SkyMomTable	Data Product

5.1.7 GiStand - Output Data Set

Frames - Data Product

Function	On	Framename	Frm	Type	N	Med	Arg	TMed	TArg
giEstimatePix	O3	estimateNpixels	xyPrep	ESTNPIX	1	33	33	33	33

Frames - Data Product

Function	On	Framename	Frm	Type	N	Med	Arg	TMed	TArg
giDetectCosmicS	O1	crhsCount	xyPrep	CRHIMG	1	33	33	33	33
	O2	crhBadPixMap	xyPrep	BADPIX	1	33	33	33	33
giExtractSpectra	O3	exSpNpixels	nxExt	EXTNPIX	1	2	5	2	5
	O4	exBadPixMap	nxExt	EXTBPM	1	2	5	2	5
	O5	exBack	xPolyY	EXTBKG	1	0	0	0	0
	O6	cleanImage	xyPrep	ESTIMG	1	33	33	33	33
	O10	clbadPixMap	xyPrep	BADPIX	1	33	33	33	33
giSplitSky	O1	wlFfSkyComp	nlWbin	BINSP	1	5	12	5	12
	O2	wlFfSkyCompErr	nlWbin	BINSP	1	5	12	5	12
giSubtractSky	O1	skySubSp	nlWbin	BINSP	1	5	12	5	12
	O2	skySubSpErr	nlWbin	BINSP	1	5	12	5	12
Total [MB]					12	189	223	189	223

Frames - Data Product Description

Function	Framename	Framedescription
giEstimatePix	estimateNpixels	number of points used in the average
giDetectCosmicS	crhsCount	optional frame with the value of photons attributed to CRH for each pixel
	crhBadPixMap	updated current bad pixel map associated to *dSubImage*
giExtractSpectra	exSpNpixels	frame of pixel counts in extracted spectra
	exBadPixMap	Current mask of bad pixels on extracted spectra
	exBack	frame of K polynomial coefficients for Nx independent background models (if extMethod=3)
	cleanImage	cleaned GIRAFFE frame with CRH replaced by interpolated values if extMethod=3
	clbadPixMap	Updated current bad pixel map (associated to the *cleanImage* frame)
giSplitSky	wlFfSkyComp	frame of all rebinned spectra, containing two images for the two sky components; each image of this frame contains one component only in the sky spectra
	wlFfSkyCompErr	frame of all rebinned spectra errors, containing two images for the two sky components; each image of this frame contains one component only in the sky errors
giSubtractSky	skySubSp	frame with sky-subtracted spectra
	skySubSpErr	frame with sky-subtracted spectra errors

Other Data - Data Product

Function	On	Dataname	Type	Med	Arg	TMed	TArg
giGetWaveSolution	O2	cleanlineTableMW	LineTable	0	0	0.05	0.05
	O3	cleanlineTableMWSky	LineTable	0	0	0.05	0.05
	O4	psfXwidth	PsfXwidTable	0.05	0.12	3.17	11.13
giModelSky	O1	sNorm	SkyCoefTable	0.02	0.06	1.58	5.56
	O2	qualSkyMod	SkyMomTable	0.2	0.2	11.6	17.4
Total [MB]				0.27	0.38	16.45	34.19

Other data - Data Product Description

Function	Dataname	Datadescription
giGetWaveSolution	cleanlineTableMW	table of line wavelengths of the calibration spectrum, for the lines retained in the calibration process (must be a subset of lineTableModeWave)

Other data - Data Product Description

Function	Dataname	Datadescription
	cleanlineTableMWSky	table of line wavelengths of the sky spectrum, for the lines retained in the calibration process (must be a subset of lineTableMWSky)
	psfXwidth	$NparPSFw$ parameters of the adjusted analytical model of the PSF width in the x direction (in pixels)
giModelSky	sNorm	Coefficients matrix $(Nlambda, K)$ of the polynomial expression for the fit of the sky intensity versus the PSF width for each spectral bin
	qualSkyMod	table with mean, 2nd, 3rd and 4th moments calculated from the difference between sky spectra and the modeled ones for each spectral bin

5.1.8 GiStand - KW tracking

Frames - Input Data KWs

Function	On	Type	Framename	Kw	Source
giGetRemoveBias	I1	RAWIMG	rawFrame	CCDID	rawFrame
				CONAD	UNKNOWN
				EXPTIME	rawFrame
				GIRFTYPE	rawFrame
				NX	rawFrame
				NY	rawFrame
				OVSCX	rawFrame
				OVSCY	rawFrame
				PRESCX	rawFrame
				PRESCY	rawFrame
				RON	rawFrame
				STARTX	rawFrame
				STARTY	rawFrame
	I2	BIASIMG	biasMast	BIASVALUE	UNKNOWN
				CCDID	biasMast
				DATAMEAN	UNKNOWN
				DATAMOD	UNKNOWN
				EXPTIME	biasMast
				GIRFTYPE	biasMast
				NX	biasMast
				NY	biasMast
				OVSCX	biasMast
				OVSCY	biasMast
				PRESCX	biasMast
				PRESCY	biasMast
				RON	biasMast
				STARTX	biasMast
				STARTY	biasMast
	I3	BADPIX	badPixMast	CCDID	badPixMast
giSubtractDark	I1	BRMIMG	bSubImage	CCDID	rawFrame
				EXPTIME	rawFrame
				GIRFTYPE	rawFrame
				NX	rawFrame
				NY	rawFrame
				STARTX	rawFrame
				STARTY	rawFrame
	I2	DARKIMG	darkMast	CCDID	darkMast
				DATAMAX	UNKNOWN

Frames - Input Data KWs

Function	On	Type	Framename	Kw	Source
				DATAMOD	UNKNOWN
				EXPTIME	darkMast
				GIRFTYPE	darkMast
				NX	darkMast
				NY	darkMast
				STARTX	darkMast
				STARTY	darkMast
	I3	BADPIX	badPixMast	CCDID	badPixMast
giEstimatePix	I1	PREPIMG	dbSubImage	CCDID	rawFrame
				GIRFTYPE	rawFrame
				GRATING	rawFrame
				LAMP	rawFrame
				NX	rawFrame
				NY	rawFrame
				SLIT	rawFrame
				STARTX	rawFrame
				STARTY	rawFrame
				WLEN0	rawFrame
	I2	LOCY	locYMast	EXTNS	locYMast
				EXTNX	locYMast
				GIRFTYPE	locYMast
				GRATING	locYMast
				SLIT	locYMast
				WLEN0	locYMast
	I3	LOCWY	locWyMast	EXTNS	locWyMast
				EXTNX	locWyMast
				GIRFTYPE	locWyMast
				GRATING	locWyMast
				SLIT	locWyMast
				WLEN0	locWyMast
	I4	BADPIX	badPixMap	CCDID	badPixMap
giDetectCosmicS	I1	BDRMIMG	dbSubImage	EXPTIME	rawFrame
				GIRFTYPE	rawFrame
				NX	rawFrame
				NY	rawFrame
				STARTX	rawFrame
				STARTY	rawFrame
	I2	ESTIMG	estimateImage	GIRFTYPE	rawFrame
				NX	rawFrame
				NY	rawFrame
				STARTX	rawFrame
				STARTY	rawFrame
	I3	ESTSIGM	estimateSigmaImage	GIRFTYPE	rawFrame
				NX	rawFrame
				NY	rawFrame
				STARTX	rawFrame
				STARTY	rawFrame
	I4	BADPIX	estBadPixMap	CCDID	rawFrame
giReplaceFlagPix	I1	PREPIMG	dbSubImage	CCDID	rawFrame
				GIRFTYPE	rawFrame
				NX	rawFrame
				NY	rawFrame
				STARTX	rawFrame
				STARTY	rawFrame
	I2	ESTIMG	estimateImage	GIRFTYPE	rawFrame
				NX	rawFrame

Frames - Input Data KWs

Function	On	Type	Framename	Kw	Source
				NY	rawFrame
				STARTX	rawFrame
				STARTY	rawFrame
	I3	BADPIX	badPixMap	CCDID	badPixMap
	I4	BADPIX	estBadPixMap	CCDID	rawFrame
giAdjustSL	I1	REPIMG	repPixImage	CCDID	rawFrame
				EXPTIME	rawFrame
				GIRFTYPE	rawFrame
				GRATING	rawFrame
				NX	rawFrame
				NY	rawFrame
				SLIT	rawFrame
				STARTX	rawFrame
				STARTY	rawFrame
				WLEN0	rawFrame
	I2	PHFFIMG	phffImageMast	EXPTIME	phffImageMast
				GIRFTYPE	phffImageMast
				GRATING	phffImageMast
				NX	phffImageMast
				NY	phffImageMast
				SLIT	phffImageMast
				STARTX	phffImageMast
				STARTY	phffImageMast
				WLEN0	phffImageMast
	I3	LOCY	locYMast	EXTNS	locYMast
				EXTNX	locYMast
				GIRFTYPE	locYMast
				GRATING	locYMast
				SLIT	locYMast
				WLEN0	locYMast
	I4	LOCWY	locWyMast	EXTNS	locWyMast
				EXTNX	locWyMast
				GIRFTYPE	locWyMast
				GRATING	locWyMast
				SLIT	locWyMast
				WLEN0	locWyMast
	I5	BADPIX	repBadPixMap	CCDID	rawFrame
giLocalSpectra	I1	BDRMIMG	dbSubImage	CCDID	rawFrame
				GIRFTYPE	rawFrame
				GRATING	rawFrame
				LINLIMIT	rawFrame
				NX	rawFrame
				NY	rawFrame
				RON	rawFrame
				SLIT	rawFrame
				STARTX	rawFrame
				STARTY	rawFrame
				WLEN0	rawFrame
	I2	LOCY	locYMast	EXTNS	locYMast
				EXTNX	locYMast
				GIRFTYPE	locYMast
				GRATING	locYMast
				SLIT	locYMast
				WLEN0	locYMast
	I3	LOCWY	locWyMast	EXTNS	locWyMast
				EXTNX	locWyMast

Frames - Input Data KWs

Function	On	Type	Framename	Kw	Source
				GIRFTYPE	locWyMast
				GRATING	locWyMast
				SLIT	locWyMast
				WLEN0	locWyMast
	I4	BADPIX	badPixMap	CCDID	badPixMap
giExtractSpectra	I1	BDRMIMG	dbSubImage	DARKVALUE	dbSubImage
				EXPTIME	rawFrame
				GIRFTYPE	rawFrame
				GRATING	rawFrame
				NX	rawFrame
				NY	rawFrame
				RON	rawFrame
				SLIT	rawFrame
				STARTX	rawFrame
				STARTY	rawFrame
				WLEN0	rawFrame
	I2	SLIMG	slImage	GIRFTYPE	rawFrame
				GRATING	rawFrame
				NX	rawFrame
				NY	rawFrame
				SLIT	rawFrame
				STARTX	rawFrame
				STARTY	rawFrame
				WLEN0	rawFrame
	I3	LOCY	locY	EXTNS	locYMast
				EXTNX	locYMast
				GIRFTYPE	rawFrame
				GRATING	rawFrame
				SLIT	rawFrame
				WLEN0	rawFrame
	I4	LOCWY	locWy	EXTNS	locYMast
				EXTNX	locYMast
				GIRFTYPE	rawFrame
				GRATING	rawFrame
				SLIT	rawFrame
				WLEN0	rawFrame
	I5	BADPIX	curBadPixMap	CCDID	curBadPixMap
giFlatSpectra	I1	EXTSP	exSp	EXTNS	locYMast
				EXTNX	locYMast
				GIRFTYPE	rawFrame
				GRATING	rawFrame
				SLIT	rawFrame
				WLEN0	rawFrame
	I2	EXTERRS	exSpErr	EXTNS	locYMast
				EXTNX	locYMast
				GIRFTYPE	rawFrame
				GRATING	rawFrame
				SLIT	rawFrame
				WLEN0	rawFrame
	I3	EXTSP	exNffMast	EXTNS	exNffMast
				EXTNX	exNffMast
				GIRFTYPE	exNffMast
				GRATING	exNffMast
				REFSRCMULT	UNKNOWN
				SLIT	exNffMast
				WLEN0	exNffMast

Frames - Input Data KWs

Function	On	Type	Framename	Kw	Source
	I4	EXTERRS	exNffErrMast	EXTNS	exNffErrMast
				EXTNX	exNffErrMast
				GIRFTYPE	exNffErrMast
				GRATING	exNffErrMast
				SLIT	exNffErrMast
				WLEN0	exNffErrMast
giGetWaveSolution	I1	EXTSP	exSp	CAMGFACT	UNKNOWN
				DCATG	rawFrame
				DTYPE	rawFrame
				EXTNS	locYMast
				EXTNX	locYMast
				FOCLEN	rawFrame
				GIRFTYPE	rawFrame
				GRATING	rawFrame
				GRATORD	rawFrame
				GRATRES	rawFrame
				GRATROT	rawFrame
				GSPACE	rawFrame
				LAMP	rawFrame
				LAMPID	rawFrame
				NX	rawFrame
				NY	rawFrame
				SATLEVEL	rawFrame
				SLIT	rawFrame
				STARTX	rawFrame
				STARTY	rawFrame
				WLEN0	rawFrame
				WLENMAX	rawFrame
				WLENMIN	rawFrame
	I2	EXTERRS	exSpErr	EXTNS	locYMast
				EXTNX	locYMast
				GIRFTYPE	rawFrame
	I3	LOCY	locY	EXTNS	locYMast
				EXTNX	locYMast
				GIRFTYPE	rawFrame
				GRATING	rawFrame
				SLIT	rawFrame
				WLEN0	rawFrame
giRebinSpectra	I1	EXTSP	ffSp	BINWNX	UNKNOWN
				CAMGFACT	UNKNOWN
				EXTNS	locYMast
				EXTNX	locYMast
				FOCLEN	rawFrame
				GIRFTYPE	rawFrame
				GRATING	rawFrame
				GRATROT	rawFrame
				SLIT	rawFrame
				WLEN0	rawFrame
				WLENMAX	rawFrame
				WLENMIN	rawFrame
	I2	EXTERRS	ffSpErr	EXTNS	locYMast
				EXTNX	locYMast
				GIRFTYPE	rawFrame
	I3	FITCOEF	fitWxl	GIRFTYPE	rawFrame
				WSCOEFi	fitWxl
				WSLINETYPE	fitWxl

Frames - Input Data KWs

Function	On	Type	Framename	Kw	Source
				WSNS	fitWxl
				WSWINDOW	fitWxl
giSplitSky	I1	BINSP	wlFfSp	?	GIRFTYPE
				?	UNKNOWN
				BINWL0	wlFfSp
				BINWNX	wlFfSp
				BINWNY	wlFfSp
				BINWSTEP	wlFfSp
				EXPTIME	rawFrame
				GRATING	rawFrame
				SLIT	rawFrame
				WLEN0	rawFrame
	I2	BINERRS	wlFfSpErr	BINWL0	wlFfSp
				BINWNX	wlFfSp
				BINWNY	wlFfSp
				BINWSTEP	wlFfSp
				GIRFTYPE	rawFrame
	I3	BINSP	psflwidth	BINWL0	wlFfSp
				BINWNX	wlFfSp
				BINWNY	wlFfSp
				BINWSTEP	wlFfSp
				GIRFTYPE	rawFrame
giNormalizeSky	I1	BINSP	wlFfSp	BINWL0	wlFfSp
				BINWNX	wlFfSp
				BINWNY	wlFfSp
				BINWSTEP	wlFfSp
				GIRFTYPE	rawFrame
				GRATING	rawFrame
				SKCOMP	UNKNOWN
				SLIT	rawFrame
				WLEN0	rawFrame
	I2	BINSP	wlFfSpErr	BINWL0	wlFfSp
				BINWNX	wlFfSp
				BINWNY	wlFfSp
				BINWSTEP	wlFfSp
				GIRFTYPE	rawFrame
				GRATING	rawFrame
				SKCOMP	UNKNOWN
				SLIT	rawFrame
				WLEN0	rawFrame
giModelSky	I1	BINSP	normSky	BINWL0	wlFfSp
				BINWNX	wlFfSp
				BINWNY	wlFfSp
				BINWSTEP	wlFfSp
				GIRFTYPE	rawFrame
				GRATING	rawFrame
				SKCOMP	wlFfSkyComp
				SLIT	rawFrame
				WLEN0	rawFrame
	I2	BINSP	normSkyErr	BINWL0	wlFfSp
				BINWNX	wlFfSp
				BINWNY	wlFfSp
				BINWSTEP	wlFfSp
				GIRFTYPE	rawFrame
				SKCOMP	wlFfSkyComp
				SKWLENMAX	UNKNOWN

Frames - Input Data KWs

Function	On	Type	Framename	Kw	Source
				SKWLENMIN	normSky
	I3	BINSP	psflwidth	BINWL0	wlFfSp
				BINWNX	wlFfSp
				BINWNY	wlFfSp
				BINWSTEP	wlFfSp
				GIRFTYPE	rawFrame
giSubtractSky	I1	BINSP	wlFfSp	BINWL0	wlFfSp
				BINWNX	wlFfSp
				BINWNY	wlFfSp
				BINWSTEP	wlFfSp
				GIRFTYPE	rawFrame
				GRATRES	rawFrame
				SKCOMP	wlFfSkyComp
	I2	BINSP	wlFfSpErr	BINWL0	wlFfSp
				BINWNX	wlFfSp
				BINWNY	wlFfSp
				BINWSTEP	wlFfSp
				GIRFTYPE	rawFrame
				SKCOMP	wlFfSkyComp
	I3	BINSP	skyModel	BINWL0	wlFfSp
				BINWNX	wlFfSp
				BINWNY	wlFfSp
				BINWSTEP	wlFfSp
				GIRFTYPE	rawFrame
				SKCOMP	wlFfSkyComp
	I4	BINSP	skyModelErr	BINWL0	wlFfSp
				BINWNX	wlFfSp
				BINWNY	wlFfSp
				BINWSTEP	wlFfSp
				GIRFTYPE	rawFrame
				SKCOMP	wlFfSkyComp

Frames - KWs from uknown source

Kw	Allreferences
	wlFfSp.giSplitSky
BIASVALUE	biasMast.giGetRemoveBias
BINWNX	ffSp.giRebinSpectra
CAMGFACT	exSp.giGetWaveSolution ffSp.giRebinSpectra
CONAD	rawFrame.giGetRemoveBias
DATAMAX	darkMast.giSubtractDark
DATAMEAN	biasMast.giGetRemoveBias
DATAMOD	biasMast.giGetRemoveBias darkMast.giSubtractDark
REFSRCMULT	exNffMast.giFlatSpectra
SKCOMP	wlFfSp.giNormalizeSky wlFfSpErr.giNormalizeSky
SKWLENMAX	normSkyErr.giModelSky

The KWs referenced in an input frame produced within the recipe during the preceeding processing are listed as externals. All such frames are listed fro each KW. This table is a developper help. He has to decide about the KW origine. Note that KWs on input frames which are NOT produced within the recipe are not checked (obviously the KWs must be present when frames are comming from the archive)

Other Data - Input Data Set KWs

Function	In	Dataname	Type	KWs
giGetRemoveBias	I1	biasLimitsMast	BiasBoxTable	CCDID (GIRTTYPE CCDID)
giLocalSpectra	I1	psfParamsMast	PsfTable	SLIT GRATING WLEN0 (GIRTTYPE SLIT GRATING WLEN0)
giGetWaveSolution	I1	linesWlampMast	LineTable	LAMP LAMPID SLIT GRATING WLEN0 (GIRTTYPE LAMP LAMPID)
giGetWaveSolution	I3	waveSolutionMast	WavCoefTable	SLIT GRATING WLEN0 (GIRTTYPE SLIT GRATING WLEN0)
giGetWaveSolution	I4	psfXwidthMast	PsfXwidTable	SLIT GRATING WLEN0 (GIRTTYPE SLIT GRATING WLEN0)
giGetWaveSolution	I5	slitGeometryMast	SlitGeoTable	SLIT (GIRTTYPE SLIT)
giSplitSky	I2	numSky	SkyTable	(GIRTTYPE SLIT GRATING WLEN0)
giSplitSky	I3	linesSkyMast	LineTable	LAMP LAMPID SLIT GRATING WLEN0 (GIRTTYPE SLIT GRATING WLEN0)
giNormalizeSky	I1	ozpoz	OzpozTable	(none (attached to frame header))

KWs required to data selection as specified in giTable.lst preceeds KWs required by the functions given between ().
Complete set of KWs required for the selection of input tables is : CCDID GRATING LAMP LAMPID SLIT WLEN0

6 Data Description

6.1 Description of Frames

Frames used and produced during the reduction steps have different formats and types.
Each *format* defines a physical structure which has, for a given observing mode, always the same number of columns, rows and planes (if it is 3-D structure). Note that the number of images within a frame is not defined by the format. This is controlled by the standard FITS KWs and is supported only at the level of raw and preprocessed frames.
Each *type* correspond to a specific stage of the processing and identify in a unique way the meaning of data pixels.

Frame formats (table /obs/ccd3/VLT/GIRAFFE/MAY99/doc/dd/TAB/giFrameFormats.r)

Format	D	Nz	Nxm	Nym	Mt	Nxa	Nya	At	Nxi	Nyi	It	Description
xyRaw	2	1	4200	2248	37.7	4200	2248	37.7	4200	2248	37.7	image with prescan and overscan
xyPrep	2	1	4096	2048	33.5	4096	2048	33.5	4096	2048	33.5	image without prescan and overscan
xPolyY	2	1	4096	10	0.1	4096	10	0.1	4096	10	0.1	coefficients of Y polynomial for each X
nxExt	2	1	4096	137	2.2	4096	320	5.2	4096	320	5.2	image of any variable in [spectra running number,X] space
nlWbin	2	1	10000	137	5.4	10000	320	12.8	10000	320	12.8	image of any variable in [spectra running number,lamda] space
xyl3D	3	10000	0	0	0	20	15	12	4	5	0.8	image of any variable in [X,Y,lamda] space

KWs associated to Frame formats

Format	InputKWs
xyRaw	GIRFTYPE NX NY STARTX STARTY PRESCX PRESCY OVSCX OVSCY RON CCDID AIRMASS ATMPRES ATMRHUM ATMTEMP DCATG DTYPE EXPTIME FOCLEN GCCDSCALE GRATING GRATORD GRATRES GRATROT GSPACE LAMP LAMPID LINLIMIT REFSRCID REFSRCOM SATLEVEL SLIT TELESCOPE TELWLEN WLEN0 WLENMAX WLENMIN
xyPrep	GIRFTYPE NX NY STARTX STARTY PRESCX PRESCY OVSCX OVSCY RON CCDID AIRMASS ATMPRES ATMRHUM ATMTEMP DCATG DTYPE EXPTIME FOCLEN GCCDSCALE GRATING GRATORD GRATRES GRATROT GSPACE LAMP LAMPID LINLIMIT REFSRCID REFSRCOM SATLEVEL SLIT TELESCOPE TELWLEN WLEN0 WLENMAX WLENMIN
xPolyY	
nxExt	GIRFTYPE NX NY STARTX STARTY PRESCX PRESCY OVSCX OVSCY RON CCDID AIRMASS ATMPRES ATMRHUM ATMTEMP DCATG DTYPE EXPTIME FOCLEN GCCDSCALE GRATING GRATORD GRATRES GRATROT GSPACE LAMP LAMPID LINLIMIT REFSRCID REFSRCOM SATLEVEL SLIT TELESCOPE TELWLEN WLEN0 WLENMAX WLENMIN EXTNX EXTNS
nlWbin	GIRFTYPE NX NY STARTX STARTY PRESCX PRESCY OVSCX OVSCY RON CCDID AIRMASS ATMPRES ATMRHUM ATMTEMP DCATG DTYPE EXPTIME FOCLEN GCCDSCALE GRATING GRATORD GRATRES GRATROT GSPACE LAMP LAMPID LINLIMIT REFSRCID REFSRCOM SATLEVEL SLIT TELESCOPE TELWLEN WLEN0 WLENMAX WLENMIN EXTNX EXTNS BINWNX BINWNY BINWSTEP BINWL0
xyl3D	TBD

Frame format table - Field description

Fieldname	Description
Format	format name
InputKWs	This is a list of all KWs in the header when frame is on input from archive or on the output as data product. It does not include KWs which may be added on during the processing.
SelectionKWs	list of KWs used to select the frame
D	number of dimensions
Nz	number of planes for all modes (always 1 in Medusa mode)
Nxm	number of rows in Medusa mode
Nym	number of columns in Medusa mode
Mt	size [MB] in Medusa mode
Nxa	number of rows in Argus mode
Nya	number of columns in Argus mode
At	size [MB] in Argus mode
Nxi	number of rows in Ifu mode
Nyi	number of columns in Ifu mode
It	size [MB] in IFU mode
Description	full designation of the format

Frame types (table /obs/ccd3/VLT/GIRAFFE/MAY99/doc/dd/TAB/giFrameTypes.r)

Type	Format	Description	SelectionKWs
RAWIMG	xyRaw	raw CCD	
BADPIX	xyPrep	bad pixel mask	CCDID
BDRMIMG	xyPrep	bias and dark subtracted	
BIASIMG	xyPrep	master bias	CCDID
BRMIMG	xyPrep	bias subtracted	
CRHIMG	xyPrep	Cosmic Ray Hit counts for each pixel	
DARKIMG	xyPrep	Master dark	CCDID

Frame types (table /obs/ccd3/VLT/GIRAFFE/MAY99/doc/dd/TAB/giFrameTypes.r)

Type	Format	Description	SelectionKWs
ESTIMG	xyPrep	estimate image (median, average or extraction)	
ESTNPIX	xyPrep	number of points used int the estimate image	
ESTSIGM	xyPrep	the standard error for each pixel after estimation	
FSFF	xyPrep	Full slit flat-field preprocessed image (usually calibration lamp)	
FSFFIMG	xyPrep	Normalized Full slit flat-field preprocessed image (usually calibration lamp)	
PHFFIMG	xyPrep	Photometric FF calibration	
PREPIMG	xyPrep	one of preprocessed image in any stage	
REPIMG	xyPrep	flagged pixels replaced by their estimation	
SLIMG	xyPrep	scattered light model	
EXTBKG	xPolyY	K polynomial coefficients for each x independant background model	
EXTBPM	nxExt	bap pixel map computed during extraction	
EXTERRS	nxExt	extracted spectra errors	
EXTNPIX	nxExt	extracted spectra pixel counts for each x bin	
EXTSP	nxExt	extracted spectra	
FITCOEF	nxExt		
LOCWY	nxExt	Y-width of localisation area for all spectra	SLIT GRATING WLEN0
LOCY	nxExt	Y-coordinate of localisation area center	SLIT GRATING WLEN0
BINERRS	nlWbin	λ rebinned extracted spectra errors	
BINSP	nlWbin	λ rebinned extracted spectra	

6.2 Description of Tables

Table summary (table /obs/ccd3/VLT/GIRAFFE/MAY99/doc/dd/TAB/giTables.r)

Name	Type	Nxm	Nym	Mt	Nxa	Nya	At	Title
girAtmExt	AtmExt	10000	20	0.8	10000	20	2.4	Atmospheric extinction coefficient
girBiasBox	BiasBox	10	5	0	10	5	0	Limits of the CCD areas for bias determination
girCcdConst	CcdConst	10	5	0	10	5	0	CCD Constants
girFiberTrans	FiberTrans	10	5	0	10	5	0	TBD
girFlat	Flat	10000	2	0	10000	2	0.2	Flux spectrum of the FF calibration lamp or other refrence source
girFluxcal	Fluxcal	10000	2	2.3	10000	2	2.3	Spectral transfer function for the current setup
girIntSky	IntSky	137	1	0	320	1	0	Sky integrated over the spectral range
girLine	Line	100	5	0.2	100	5	0.6	Catalogue of emission lines used for Wavelength Calibration
girMeanSp	MeanSp	10	5	0	10	5	0	TBD
girOzpoz	Ozpoz	137	50	0	15	50	0	Positioner binary table
girPsf	Psf	137	100	6.3	320	100	22.2	Parameters describing the shape and width of the transverse (Y-direction) PSF.
girPsfXwid	PsfXwid	137	50	3.1	320	50	11.1	Width of the PSF in the dispersion direction
girSky	Sky	137	1	0	320	1	0	Running numbers of sky spectra to be used for the sky modeling
girSkyCoef	SkyCoef	10000	5	0.2	10000	5	0.2	Polynomial model of the normalized sky
girSkyMom	SkyMom	10000	5	11.6	10000	5	17.4	Sky quality - 1-4th moments of the residual sky (after sky subtraction)

Table summary (table /obs/ccd3/VLT/GIRAFFE/MAY99/doc/dd/TAB/giTables.r)

Name	**Type**	**Nxm**	**Nym**	**Mt**	**Nxa**	**Nya**	**At**	**Title**
girSlitGeo	SlitGeo	137	6	0	320	6	0	Geometry of slits - position of fibers
girSpectroConst	SpectroConst	145	10	0	145	10	0	Spectrograph Setup Constants
girTransVlt	TransVlt	10000	2	0	10000	2	0	Telescope and Nasmyth corrector transmission
girVignCorr	VignCorr	580	4	0	580	4	0	Nasmyth vigneting (screen and telescope)
girVignSpectro	VignSpectro	10	5	0	10	5	0	Spectrograph Internal Vignetting
girWavCoef	WavCoef	137	20	0.6	320	20	2.2	Coefficients of the Polynomial Wavelength Solution

Table types - Field description

Fieldname	**Description**
Name	table name
Type	table type (compulsory syntaxe: girXxx..Table)
Nxm	Number of rows in MEDUSA mode
Nym	Number of collumns in MEDUSA mode
Mt	size [MB] in Medusa mode
Nxa	Number of rows in ARGUS mode (default - same as for MEDUSA)
Nya	Number of collumns in ARGUS mode (default - same as for MEDUSA)
At	size [MB] in Argus mode
Title	Title of the table

7 Detailed Description of Tables

7.1 General remarks

The following section is generated from the development database. It insures that every FITS KW is uniquely defined in and traceable to the Data Dictionary (DIC) and that all data-structures used within the function design are defined in a consistent way. Since not a reliable up-to-date DIC database for developers is available for the time being we reference DIC through an alias look-up table. The FITS KW described in *Table Header* subsections are referenced by their aliases and traced to the existing KW in ESO DICs (column *dicname*). We reference provisionally the DIC named *DRS* whenever we need a new KW created and maintained by the DRS and DIC named *MISSING* or FPMISS for *Fiber Positionner* whenever we failed to find a KW which we expect to receive from FLAMES OS.
The descriptions and comments are as complete as possible and similar formalism as for DIC presentation is used:

- If description/comments are present in both Alias definition and DIC definition files, the texts are prefixed *AL:* and *DIC:* respectively. If specific comment is given in table description it is appended with prefix *TAB:*.
- If description/comments are present only in Alias (usually new KW), there is no prefix.
- If description/comments are present only in DIC (usually ESO KW), there is a prefix *DIC:*
- The description is shortened to 160 characters (followed by + sign). The full description is given in section 7.24.
- If there is a conflict between DIC and Alias table concerning the *type* or *unit* both definition are printed separated by .

7.2 girAtmExt - Atmospheric extinction coefficient

Purpose: Provide model of typical atmospheric spectral extinction (full spectral coverage).
Reference Function: giFluxConv

girAtmExt: description

This is a unique set of static data to be introduced by the calibration team, using the data available from Paranal. Atmospheric extinction coefficient table with two columns is given. The first column contains the wavelength and the second one the atmospheric extinction coefficient $k(\lambda)$. The columns AMODPRES, AMODRHUM, AMODTEMP are used to use the model which is the closest match to the current situation (KWs ATMPRES, ATMRHUM, ATMTEMP on in the input frame). This rather redundant way to provide sevral models is motivate by the VLT DFS limitation (only exact match between KWs could be used for table selection)

girAtmExt: working status

running number of the table:	2
table name:	girAtmExt
author:	Cayatte
date:	20000623
status:	draft/template submitted to FDR
table source:	**archive**
source comment:	Supplied by VLT

girAtmExt: archiving technical informations

Total size is negligible (below 1MB)

Column	Description	Type	Unit	Comment

girAtmExt: Columns

Column	Description	Type	Unit	Comment
LAMBDA	wavelength	float	nm	
K	atmospheric extinction coefficient for wavelength LAMBDA	float	none	
AMODPRES	atmospheric pressure of the model	integer	mbar	
AMODRHUM	relative humidity of the model	float		Number between 0-1
AMODTEMP	absolute temperature of the model	float	deg K	

girAtmExt: Header

Alias	DIC	Description	Type	Unit	Comment
TITLE	DRS	Title of data structure	string	none	Standard table KW
PURPOSE	DRS	Purpose of data structure	string	none	Standard table KW
AUTHOR	DRS	Source of data (responsible or processing recipe)	string	none	Standard table KW
DATE	CCDDCS	DIC: UT date when this file was written (ISO 8601).	string	none	DIC: UT date when this file was written
STATUS	DRS	Comment on last update	string	none	Standard table KW
GIRTTYPE	DRS	TAB: type of data type of table data	string	none	TAB: 'ATMEXTAB'

7.3 girBiasBox - Limits of the CCD areas for bias determination

Purpose: Provide CCD coordinates of the corners of the box-shaped bias areas
Reference Function: giGetRemoveBias

girBiasBox: description

This is a uniq configuration table which defines frame portions used to compute the bias. The KW CCDID must math with KW on input frame.

girBiasBox: working status

running number of the table:	3
table name:	girBiasBox
author:	Simond
date:	20000621
status:	draft/template submitted to FDR
table source:	**archive**
source comment:	Created during the CCD commissioning based on the data (several Biases and Darks) supp

girBiasBox: archiving technical informations

Total size is negligible (below 1MB)

girBiasBox: Columns

Column	Description	Type	Unit	Comment
BIAS_X1	X-position of the upper-left corner	integer	pixel	
BIAS_Y1	Y-position of the upper-left corner	integer	pixel	
BIAS_X2	X-position of the lower-right corner	integer	pixel	
BIAS_Y2	Y-position of the lower-right corner	integer	pixel	
COMMENT	Any comment	string		

girBiasBox: Header

Alias	DIC	Description	Type	Unit	Comment
TITLE	DRS	Title of data structure	string	none	Standard table KW
PURPOSE	DRS	Purpose of data structure	string	none	Standard table KW
AUTHOR	DRS	Source of data (responsible or processing recipe)	string	none	Standard table KW
DATE	CCDDCS	DIC: UT date when this file was written (ISO 8601).	string	none	DIC: UT date when this file was written
STATUS	DRS	Comment on last update	string	none	Standard table KW
GIRTTYPE	DRS	type of table data	string	none	TAB: 'BIASBOXTAB'
CCDID	CCDDCS	DIC: Detector system identification Format: <CCD Id> - <ACE Id>	string	none	DIC: Detector system Id

7.4 girCcdConst - CCD Constants

Purpose: Provide the setup dependent constants
Reference Function:

girCcdConst: description

This, unique look-up table gives all informations specific to the CCD unit unavailable from the OS. It is therefore MANDATORY that this table is updated if there is a modification in the CCD performances. The constants given here are not critical to the performance of the DRS.
The CCDID and CCDPIXTIME are used to select the line defining the actual values of DRS KWs starting in Column 3. This is done in the specific DRS function (yet TBD) which creates all missing KWs and assign values according to this table.
Any unforseen, CCD-dependent constant, will be added in this table as a new column.

girCcdConst: working status

running number of the table:	21
table name:	girCcdConst
author:	Blecha
date:	20001109
status:	1-st introduction according the progress Meeting
table source:	**From CCD comissioning; 1 line per setup**
source comment:	Emperical formula for the RON could be used

girCcdConst: archiving technical informations

Total size is negligible (below 1MB)

girCcdConst: Columns

Column	Description	Type	Unit	Comment
CCDID	Detector system Id	string	none	Input - Lookup value of the KW CCDID
CCDPIXTIME	Time to read pixel	string	usec	Input - Lookup value; actual KW must be converted in string before the row selection
LINLIMIT	linearity limit of the CCD	float	e	Output
RON	Expected CCD readout noise	float	e	Output
SATLEV	CCD saturation leve	float	e	Output

girCcdConst: Header

Alias	DIC	Description	Type	Unit	Comment
TITLE	DRS	Title of data structure	string	none	Standard table KW
PURPOSE	DRS	Purpose of data structure	string	none	Standard table KW
AUTHOR	DRS	Source of data (responsible or processing recipe)	string	none	Standard table KW
DATE	CCDDCS	DIC: UT date when this file was written (ISO 8601).	string	none	DIC: UT date when this file was written
STATUS	DRS	Comment on last update	string	none	Standard table KW
GIRTTYPE	DRS	TAB: type of data type of table data	string	none	TAB: 'CCDCONST'

7.5 girFiberTrans - TBD

Purpose:
Reference Function: giFiberTrans

girFiberTrans: description

This table is the result of the *giFiberTrans*() function and contains the relative fiber transmissions. These involve the whole flux integrated over the dispersion direction for each spectrum, and they are normalized by the best transmission, so that they are all $leq1.0$.

girFiberTrans: working status

running number of the table:	24
table name:	girFiberTrans
author:	North
date:	20010121
status:	1-st introduction TBC with PN
table source:	
source comment:	

girFiberTrans: archiving technical informations

Total size is negligible (below 1MB)

girFiberTrans: Columns

Column	Description	Type	Unit	Comment
NSPEC	Running numberof the spectrum	integer		
FIBTRAN	fiber transmission	float		

girFiberTrans: Header

Alias	DIC	Description	Type	Unit	Comment
TITLE	DRS	Title of data structure	string	none	Standard table KW
PURPOSE	DRS	Purpose of data structure	string	none	Standard table KW
AUTHOR	DRS	Source of data (responsible or processing recipe)	string	none	Standard table KW
DATE	CCDDCS	DIC: UT date when this file was written (ISO 8601).	string	none	DIC: UT date when this file was written
STATUS	DRS	Comment on last update	string	none	Standard table KW
GIRTTYPE	DRS	type of table data	string	none	

Alias	DIC	Description	Type	Unit	Comment
SLIT	FGIR_ICS	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies	integer	none	DIC: Position (hs). AL: What is the range, [0-4] ? I ma not happy with that.
GRATING	FGIR_ICS	DIC: ESO name for grating unit.	string	none	DIC: Grating common name (hls). AL: Caution: For FGIR_ICS KWs there is a dicrepancy in INS used by FGIR_ICS and INS1 required by VLT standard
WLEN0	FGIR_ICS	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength	double	nm	DIC: Grating central wavelength [nm] (hs).

7.6 girFlat - Flux spectrum of the FF calibration lamp or other refrence source

Purpose: Provide spectral emissivity of the reference source (full spectra coverage).
Reference Function: giFluxConv

girFlat: description

This table gives the reference physical flux of the source (Spectrophotometric standard star or FF lamp). The first column contains the wavelength and the second one the flux per spectral bin. A uniq spectral sampling (TBC) is used. The selection of the table is made on the KWs LAMPID, REFSRC, REFSRCID.

girFlat: working status

running number of the table:	4
table name:	girFlat
author:	Cayatte
date:	20000623
status:	draft/template submitted to FDR
table source:	**archive**
source comment:	From laboratory calibration for LAMP, user provided for standard stars

girFlat: archiving technical informations

Total size is negligible (below 1MB)

girFlat: Columns

Column	Description	Type	Unit	Comment
LAMBDA	wavelength	float	nm	referenced to center of the bin
FLXNFF	Flat field lamp flux per electron for wavelength LAMBDA	float	see *)	*) $erg.s^{-1}.cm^{-2}.Å^{-1}.e^{-}$

girFlat: Header

Alias	DIC	Description	Type	Unit	Comment
TITLE	DRS	Title of data structure	string	none	Standard table KW
PURPOSE	DRS	Purpose of data structure	string	none	Standard table KW

Alias	DIC	Description	Type	Unit	Comment
AUTHOR	DRS	Source of data (responsible or processing recipe)	string	none	Standard table KW
DATE	CCDDCS	DIC: UT date when this file was written (ISO 8601).	string	none	DIC: UT date when this file was written
STATUS	DRS	Comment on last update	string	none	Standard table KW
GIRTTYPE	DRS	type of table data	string	none	TAB: 'FLXNFFTAB'
LAMP	DRS	Name of the W Calibration source	string	none	One of ThAr,Ne,ThArNe or 'SKY' (telluric lines)
LAMPID	FP_LP		undef	none	
REFSRC	MISSING	Type of reference source (LAMP, STAR, SKY, OTHER); same as LAMP KW if type is LAMP	string	none	
REFSRCID	MISSING	Unique ID of reference source (same as LAMPID if type is LAMP)	string	none	
SLIT	FGIR_ICS	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies	integer	none	DIC: Position (hs). AL: What is the range, [0-4] ? I ma not happy with that.
GRATING	FGIR_ICS	DIC: ESO name for grating unit.	string	none	DIC: Grating common name (hls). AL: Caution: For FGIR_ICS KWs there is a dicrepancy in INS used by FGIR_ICS and INS1 required by VLT standard
WLEN0	FGIR_ICS	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength	double	nm	DIC: Grating central wavelength [nm] (hs).
FLCONVSTAND	DRS	Frflag for flux conversion	logical	none	Better name TBF

7.7 girFluxcal - Spectral transfer function for the current setup

Purpose: Provides user-established spectral transfer function for spectrophotomeric flux calibration (off-line only).
Reference Function: giFluxConv

girFluxcal: description

Spectral transfer function for one or several standard stars. The first column contains the wavelength and the second one the value of flux per spectral bin and per electron. This table has to be computed for a given observing mode.

girFluxcal: working status and source informations

running number of the table:	15
table name:	girFluxcal
author:	Cayatte
date:	20000623
status:	draft/template submitted to FDR
table source:	**user's input**
source comment:	Only off-line mode. Is produced by a separate observation.

girFluxcal: archiving technical informations

comment concerning the size:		unique
total number of tables in MEDUSA mode:	29	
size per table	0.1	MBytes
total size of tables in MEDUSA mode:	**2.9**	MBytes
total number of tables in ARGUS/IFU mode:	0	
size per table	0.0	MBytes
total size of tables in ARGUS/IFU mode:	**0.0**	MBytes
Total for all modes	**2.9**	MBytes

girFluxcal: Columns

Column	**Description**	**Type**	**Unit**	**Comment**
LAMBDA	wavelength	float	nm	
FLUX	spectral transfer function for wavelength LAMBDA	float	see *)	*) erg.s^{-1}.cm^{-2}.$Å^{-1}$.e^{-} Changed column name from RABS to FLX (blecha Wed 12/06/00 11:51:42)

girFluxcal: Header

Alias	**DIC**	**Description**	**Type**	**Unit**	**Comment**
TITLE	DRS	Title of data structure	string	none	Standard table KW
PURPOSE	DRS	Purpose of data structure	string	none	Standard table KW
AUTHOR	DRS	Source of data (responsible or processing recipe)	string	none	Standard table KW
DATE	CCDDCS	DIC: UT date when this file was written (ISO 8601).	string	none	DIC: UT date when this file was written
STATUS	DRS	Comment on last update	string	none	Standard table KW
GIRTTYPE	DRS	type of table data	string	none	TAB: 'SPTRANSTAB'
STARNAME	DRS	ID of standard star(s)	logical	none	Better name TBF
SLIT	FGIR_ICS	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies	integer	none	DIC: Position (hs). AL: What is the range, [0-4] ? I ma not happy with that.
GRATING	FGIR_ICS	DIC: ESO name for grating unit.	string	none	DIC: Grating common name (hls). AL: Caution: For FGIR_ICS KWs there is a dicrepancy in INS used by FGIR_ICS and INS1 required by VLT standard
WLEN0	FGIR_ICS	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength	double	nm	DIC: Grating central wavelength [nm] (hs).
FLCONVSTAND	DRS	Frflag for flux conversion	logical	none	Better name TBF
REFSRC	MISSING	Type of reference source (LAMP, STAR, SKY, OTHER); same as LAMP KW if type is LAMP	string	none	
REFSRCID	MISSING	Unique ID of reference source (same as LAMPID if type is LAMP)	string	none	

7.8 girIntSky - Sky integrated over the spectral range

Purpose: Provide the flux of sky spectra integrated over the current bandwidth.
Reference Function: giNormalizeSky

girIntSky: description

This is a unique, internal *working table* containing the *nspec* values of the integrated flux of *nspec* sky spectra of the current frame. The corresponding estimated errors are given too. The header contains the information concerning the instrument setup and associated frame (KW FATHER).

girIntSky: working status

running number of the table:	9
table name:	girIntSky
author:	Cayatte
date:	20000622
status:	draft/template submitted to FDR
table source:	**processing**
source comment:	Intermediate table

girIntSky: archiving technical informations

Total size is negligible (below 1MB)

girIntSky: Columns

Column	Description	Type	Unit	Comment
NSPEC	Running number of the spectrum	integer	none	
INTFLX	Integrated flux for sky spectrum	float	$e^{-}\times$nm	
INTFLXERR	Error on the integrated flux	float	$e^{-}\times$nm	

girIntSky: Header

Alias	DIC	Description	Type	Unit	Comment
TITLE	DRS	Title of data structure	string	none	Standard table KW
PURPOSE	DRS	Purpose of data structure	string	none	Standard table KW
AUTHOR	DRS	Source of data (responsible or processing recipe)	string	none	Standard table KW
DATE	CCDDCS	DIC: UT date when this file was written (ISO 8601).	string	none	DIC: UT date when this file was written
STATUS	DRS	Comment on last update	string	none	Standard table KW
GIRTTYPE	DRS	type of table data	string	none	TAB: 'INTSKYTAB'
SLIT	FGIR_ICS	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies	integer	none	DIC: Position (hs). AL: What is the range, [0-4] ? I ma not happy with that.
GRATING	FGIR_ICS	DIC: ESO name for grating unit.	string	none	DIC: Grating common name (hls). AL: Caution: For FGIR_ICS KWs there is a dicrepancy in INS used by FGIR_ICS and INS1 required by VLT standard

Alias	DIC	Description	Type	Unit	Comment
WLEN0	FGIR_ICS	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength	double	nm	DIC: Grating central wavelength [nm] (hs).
FATHER	DRS	ID of father frame; used to associate an auxilliary data to original frame	string	none	
SKWLENMIN	DRS	SKY normalization: minimum wavelength	double	nm	
SKWLENMAX	DRS	SKY normalization: maximum wavelength	double	nm	

7.9 girLine - Catalogue of emission lines used for Wavelength Calibration

Purpose: Provide emission line position and flux for wavelength calibration
Reference Function: giGetWaveSolution

girLine: description

This table, one for each setup category (see below), contains the laboratory calibration of a Wavelength Calibration source, which could be Calibration lamp or SKY. This is a static table, though some initial adjustement will have to be made concerning the final set of used lines. The selection is made on the exact match of KWs SLIT, LAMP, GRATING and WLEN0. Same table is used for various instances of the LAMP of the same type (but LAMPID different) and for both SLITs in MEDUSA mode, possibly for all SLITSs whatever is the mode.
The accurate wavelengths are given while the flux accuracy and the complete interpretation of given fluxes is yet TBD. In order to maintain the calibration stability, the line selection should remain static through the life-cycle of the GIRAFFE. Only lines with $flux > 0$ will be used.

girLine: working status

running number of the table:	5
table name:	girLine
author:	Blecha
date:	20000605
status:	draft/template submitted to FDR
table source:	**archive**
source comment:	Laboratory calibration provided by GIRAFFE DRS team.

girLine: archiving technical informations

Total size is negligible (below 1MB)

girLine: Columns

Column	Description	Type	Unit	Comment
WLEN	Wavelength of the line	float	nm	Central wavelength of the line
NAME	Name of line	string		If any
FLUX	Expected central flux/[pixel.sec]	float	e^-	Mode and Resolution dependent (See table description)
COMMENT	Any comment	string		Mainly to explain any manual line-selection

girLine: Header

Alias	DIC	Description	Type	Unit	Comment

Alias	DIC	Description	Type	Unit	Comment
TITLE	DRS	Title of data structure	string	none	Standard table KW
PURPOSE	DRS	Purpose of data structure	string	none	Standard table KW
AUTHOR	DRS	Source of data (responsible or processing recipe)	string	none	Standard table KW
DATE	CCDDCS	DIC: UT date when this file was written (ISO 8601).	string	none	DIC: UT date when this file was written
STATUS	DRS	Comment on last update	string	none	Standard table KW
GIRTTYPE	DRS	type of table data	string	none	TAB: 'LINETAB'
LAMP	DRS	Name of the W Calibration source	string	none	One of ThAr,Ne,ThArNe or 'SKY' (telluric lines)
LAMPID	FP_LP		undef	none	
SLIT	FGIR_ICS	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies	integer	none	DIC: Position (hs). AL: What is the range, [0-4] ? I ma not happy with that.
GRATING	FGIR_ICS	DIC: ESO name for grating unit.	string	none	DIC: Grating common name (hls). AL: Caution: For FGIR_ICS KWs there is a dicrepancy in INS used by FGIR_ICS and INS1 required by VLT standard
WLEN0	FGIR_ICS	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength	double	nm	DIC: Grating central wavelength [nm] (hs).
WSCOEFi	DRS	PSF X-width coef	float	none	
WSLINETYPE	DRS	use of ThArNe or telluric lines for Wavelength Solution	logical	none	
WSWINDOW	DRS	width in pixel for line detection and fit	float	pixel	
WSCLIPMFRAC	DRS	min fraction of points accepted/total (sigma-clipping)	float	none	
WSCLIPSIGMA	DRS	multiple of sigma (sigma-clipping)	float	none	
WSCLIPNITER	DRS	number of iterations (sigma-clipping)	integer	none	
WSXWIDXDEG	DRS	polynomial degree along the x direction for x-width modeling	integer	none	
WSXWIDYDEG	DRS	polynomial degree along the y direction for x-width modeling	integer	none	
WSWLENXDEG	DRS	polynomial degree along the x direction for wavelength solution	integer	none	
WSWLENYDEG	DRS	polynomial degree along the y direction for wavelength solution	integer	none	

7.10 girMeanSp - TBD

Purpose:
Reference Function: giFiberTrans

girMeanSp: description

This table gives the average flux of each spectrum. The average is simply the arithmetic average of the pixel values of the extracted spectrum over the whole wavelength range, for a given set-up of the instrument.

girMeanSp: working status

running number of the table:	23
table name:	girMeanSp
author:	North
date:	20010121
status:	1-st introduction TBC with PN
table source:	
source comment:	

girMeanSp: archiving technical informations

Total size is negligible (below 1MB)

girMeanSp: Columns

Column	Description	Type	Unit	Comment
NSPEC	Running number of the spectrum	integer		
MEANSP	spectrum intensity averaged over the x direction	float	e^-	

girMeanSp: Header

Alias	DIC	Description	Type	Unit	Comment
TITLE	DRS	Title of data structure	string	none	Standard table KW
PURPOSE	DRS	Purpose of data structure	string	none	Standard table KW
AUTHOR	DRS	Source of data (responsible or processing recipe)	string	none	Standard table KW
DATE	CCDDCS	DIC: UT date when this file was written (ISO 8601).	string	none	DIC: UT date when this file was written
STATUS	DRS	Comment on last update	string	none	Standard table KW
GIRTTYPE	DRS	type of table data	string	none	
SLIT	FGIR_ICS	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies	integer	none	DIC: Position (hs). AL: What is the range, [0-4] ? I ma not happy with that.
GRATING	FGIR_ICS	DIC: ESO name for grating unit.	string	none	DIC: Grating common name (hls). AL: Caution: For FGIR_ICS KWs there is a dicrepancy in INS used by FGIR_ICS and INS1 required by VLT standard
WLEN0	FGIR_ICS	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength	double	nm	DIC: Grating central wavelength [nm] (hs).

7.11 girOzpoz - Positioner binary table

Purpose: Provides fibre assignation and configuration
Reference Function:

girOzpoz: description

This is the FITS binary table of the positioner, part of the input *rawFrame*, describing fibre configuration see "FLAMES FIBRE POSITIONER, Instrument Control Software Design Description" INS-SPE-AUS-13721-0042 for more details.

girOzpoz: working status

running number of the table:	11
table name:	girOzpoz
author:	Simond
date:	20000613
status:	draft/template submitted to FDR
table source:	**archive**
source comment:	Part of the rawImage

girOzpoz: archiving technical informations

Total size is negligible (below 1MB)

girOzpoz: Columns

Column	Description	Type	Unit	Comment
OBJECT	Name of the object	string	none	
RA	J200 mean right ascension	double	radians	
DEC	J200 mean declinaison	double	radians	
R	R position of the fibre	integer	microns	
THETA	theta position of the fibre	real	radians	
R_ERROR	error in R	integer	microns	
THETA_ERROR	error in theta	real	radians	
OBJECT_TYPE	object type: P=programme object, S=sky, G=guide star, X=simulation, F=flux refrence star (request OGL/OP)	character	none	OBJECT_TYPE=X refers to the ability to feed simulated starlight into purpose-built fibres, this can only be used while the positioner is above the plate
FIBRE_TYPE	fibre type: F=FACB, I=IFU, T=test, U=UVES, M=Medusa	character	none	FIBRE_TYPE=T refers to fibres allocated to internal positioner test fibres
BUTTON	positioner button number	integer	none	
MAGNITUDE	object's magnitude	real	none	
PID	program ID associated with the object	integer	none	
COMMENT	comment associated with the object	string	none	
PRIORITY	object's allocated priority	integer	none	
SPECT_FIBRE	fibre number on the spectrograph slit	integer	none	

girOzpoz: Header

Alias	DIC	Description	Type	Unit	Comment
TITLE	DRS	Title of data structure	string	none	Standard table KW
PURPOSE	DRS	Purpose of data structure	string	none	Standard table KW
AUTHOR	DRS	Source of data (responsible or processing recipe)	string	none	Standard table KW

Alias	DIC	Description	Type	Unit	Comment
DATE	CCDDCS	DIC: UT date when this file was written (ISO 8601).	string	none	DIC: UT date when this file was written
STATUS	DRS	Comment on last update	string	none	Standard table KW
GIRTTYPE	DRS	type of table data	string	none	TAB: 'OZPOZTAB'
CENRA	FPMISS	field centre right ascension	real	radians	input value of P2PP
CENDEC	FPMISS	field centre declination	real	radians	input value of P2PP
CENEQNX	FPMISS	equinox of the above (FK5 Julian)	real	none	input value of P2PP
APPRA	FPMISS	apparent right ascension	real	radians	value calculated by P2PP
APPDEC	FPMISS	apparent declination	real	radians	value calculated by P2PP
APPEPOCH	FPMISS	epoch of the above	real	none	value calculated by P2PP
CONFMJD	FPMISS	MJD for which the field was configured	real	none	
ACTMJD	PR_FITS	DIC: Modified Julian Day of the start of the exposure. The MJD is related to the Julian Day (JD) via the formula: MJD = JD-2400000.5 The comment includes a civil representatio+	double *real	days	DIC: Obs start
UNALLOBJ	FPMISS	number of unallocated program objects	integer	none	
UNALLFACB	FPMISS	number of unallocated guide stars	integer	none	
UNALLSKY	FPMISS	number of unallocated sky positions	integer	none	
ALLOCOBJ	FPMISS	number of allocated program objects	integer	none	
ALLOCFACB	FPMISS	number of allocated guide stars	integer	none	
ALLOCSKY	FPMISS	number of allocated sky positions	integer	none	
LABEL	FPMISS	name of the field	string	none	
FILENAME	FPMISS	name of the configuration file used	string	none	

7.12 girPsf - Parameters describing the shape and width of the transverse (Y-direction) PSF.

Purpose:
Reference Function: giLocalSpectra

girPsf: description

This table contains the coefficients of (x,y) polynomials describing the variation of the width, and in general the shape, of the PSF along the direction perpendicular to the dispersion (y - in the pixel space), as a function of position on the CCD. Since, beside the width parameter which has to be understood as a simplest attempt to parameterize the PSF shape, there are also up to *npar* parameters describing the y-shape of the spectra, the table has also *npar* sets of columns describing the polynomials fitted to each shape parameter. Each set consists in $i \times j$ coefficients, where i=0,... DEGXW and j=0,... DEGYW (polynomial degrees). Each line relates to one fiber (one spectrum). We assume that the degrees of the polynomials will be freezed during the commissioning period.
Only the polynomials describing the PSF of fibers belonging to the same subslit could be fitted in a continuous function since there are physical discontinuities between subslits. Therefore, there will be, e.g. for the MEDUSA mode, 13 blocks of 9 or 13 lines with identical coefficients in each block. Although this way of storing information is redundant, it is probably simpler than keeping only one line per block.
This table will be produced/updated, for each setup, during each FF Calibration run during the localisation step. The table is setup-and-slit-dependent and will be selected, when used on the input for localisation or extraction,

by the match on KWs SLIT, GRATING and WLEN0.

girPsf: working status and source informations

running number of the table:	7
table name:	girPsf
author:	North
date:	20000623
status:	draft/template submitted to FDR
table source:	**archive**
source comment:	The first instance is produced during the COMMISSIONING (0-version in Europe with GIRAFFE only), then routinely: Created if not available, updated during processing

girPsf: archiving technical informations

comment concerning the size:		Setup dependent
total number of tables in MEDUSA mode:	58	
size per table	0.1	MBytes
total size of tables in MEDUSA mode:	**5.8**	MBytes
total number of tables in ARGUS/IFU mode:	87	
size per table	0.3	MBytes
total size of tables in ARGUS/IFU mode:	**26.1**	MBytes
Total for all modes	**31.9**	MBytes

girPsf: Columns

Column	**Description**	**Type**	**Unit**	**Comment**
NSPEC	running number of the spectrum	integer		
NSUBSLIT	running number of the subslit	integer		
CWYIJKM	coefficients of 2-D polynomials describing the PSF(y) variation	float	pixel	there are m=1,...$npar$ sets of CWIJK coefficients, i, j designating the polynomial degrees and k designating the subslit running number

girPsf: Header

Alias	**DIC**	**Description**	**Type**	**Unit**	**Comment**
TITLE	DRS	Title of data structure	string	none	Standard table KW
PURPOSE	DRS	Purpose of data structure	string	none	Standard table KW
AUTHOR	DRS	Source of data (responsible or processing recipe)	string	none	Standard table KW
DATE	CCDDCS	DIC: UT date when this file was written (ISO 8601).	string	none	DIC: UT date when this file was written
STATUS	DRS	Comment on last update	string	none	Standard table KW
GIRTTYPE	DRS	type of table data	string	none	TAB: 'PSFPRMTAB'
SLIT	FGIR_ICS	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies	integer	none	DIC: Position (hs). AL: What is the range, [0-4] ? I ma not happy with that.

Alias	DIC	Description	Type	Unit	Comment
GRATING	FGIR_ICS	DIC: ESO name for grating unit.	string	none	DIC: Grating common name (hls). AL: Caution: For FGIR_ICS KWs there is a dicrepancy in INS used by FGIR_ICS and INS1 required by VLT standard
WLEN0	FGIR_ICS	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength	double	nm	DIC: Grating central wavelength [nm] (hs).
LCLIPSIGMA	DRS	multiple of sigma (sigma-clipping)	float	none	
LCLIPNITER	DRS	number of iterations (sigma-clipping)	integer	none	
LCLIPMFRAC	DRS	min fraction of points accepted/total (sigma-clipping)	float	none	
NOISEMULT	DRS	noise detector multiple	float	none	
LPOLYDEG	DRS	degree of the polynomial fit	integer	none	

7.13 girPsfXwid - Width of the PSF in the dispersion direction

Purpose: parameterisation the PSF along the x directions (x-space)

Reference Function: giGetWaveSolution

girPsfXwid: description

This table contains the coefficients of (x,y) polynomials describing the variation of the width, and in general the shape, of the PSF along the direction of the dispersion (x - in the pixel space), as a function of the position on the CCD. The same remarks as for *girPsfTable* aplies: since, beside the width parameter, there are also up to *npar* parameters describing the x-shape of the spectra, the table has also *npar* sets of columns describing the polynomials fitted to each shape parameter. Each set consists in $i \times j$ coefficients, where i=0,... DEGXW and j=0,... DEGYW (polynomial degrees). Each line relates to one fiber (one spectrum).

This table will be produced/updated, for each setup, during each SEWC Calibration run during the GetWaveSolution step. The table is setup-and-slit-dependent and will be selected, when used on the input for GetWaveSolution or ModelSky, by the match on KWs SLIT, GRATING and WLEN0. Note that unlike the similar table *girPsfTable* the width polynomial of the input master table will be updated for all science exposures. This should maintain contunuously the optimal PSF model, crutial to the sky subtraction.

girPsfXwid: working status and source informations

running number of the table:	12
table name:	girPsfXwid
author:	North
date:	20000621
status:	draft/template submitted to FDR
table source:	**archive**
source comment:	The first instance is produced during the COMMISSIONING (0-version in Europe with GIRAFFE only), then routinely: Created if not available, updated during processing

girPsfXwid: archiving technical informations

comment concerning the size:		number of collumns is approximate, it depends on the degree of the polynomial
total number of tables in MEDUSA mode:	58	
size per table	0.1	MBytes
total size of tables in MEDUSA mode:	**5.8**	MBytes
total number of tables in ARGUS/IFU mode:	87	
size per table	0.1	MBytes
total size of tables in ARGUS/IFU mode:	**8.7**	MBytes
Total for all modes	**14.5**	MBytes

girPsfXwid: Columns

Column	**Description**	**Type**	**Unit**	**Comment**
NSPEC	running number of the spectrum	integer		
NSUBSLIT	running number of the subslit	integer		
CWIJKM	coefficient of 2-D polynomials describing the x-width variation	double	pixel	there are m=1,...*npar* sets of CWIJK coefficients, i, j designating the polynomial degrees and k designating the subslit running number

girPsfXwid: Header

Alias	**DIC**	**Description**	**Type**	**Unit**	**Comment**
TITLE	DRS	Title of data structure	string	none	Standard table KW
PURPOSE	DRS	Purpose of data structure	string	none	Standard table KW
AUTHOR	DRS	Source of data (responsible or processing recipe)	string	none	Standard table KW
DATE	CCDDCS	DIC: UT date when this file was written (ISO 8601).	string	none	DIC: UT date when this file was written
STATUS	DRS	Comment on last update	string	none	Standard table KW
GIRTTYPE	DRS	type of table data	string	none	TAB: 'PSFXWIDTAB'
SLIT	FGIR_ICS	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies	integer	none	DIC: Position (hs). AL: What is the range, [0-4] ? I ma not happy with that.
GRATING	FGIR_ICS	DIC: ESO name for grating unit.	string	none	DIC: Grating common name (hls). AL: Caution: For FGIR_ICS KWs there is a dicrepancy in INS used by FGIR_ICS and INS1 required by VLT standard
WLEN0	FGIR_ICS	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength	double	nm	DIC: Grating central wavelength [nm] (hs).
FOCLEN	DRS	focal length of the collimator	double	m	
CCDSCALE	-				
SIGMA	-				

Alias	DIC	Description	Type	Unit	Comment
GRATROT	FGIR_ICS	DIC: Rotation angle of grating. The reference frame is given in the corresponding instrument specification. ENC = OFFST+ RESOL* RAD2DEG* arcsin(WLEN* GRV/(2* cos(ROT))) AL: Ro+	double *float	deg	DIC: Grating rot angle [deg] (hl).
WSCOEFi	DRS	PSF X-width coef	float	none	
WSLINETYPE	DRS	use of ThArNe or telluric lines for Wavelength Solution	logical	none	
WSWINDOW	DRS	width in pixel for line detection and fit	float	pixel	
WSCLIPMFRAC	DRS	min fraction of points accepted/total (sigma-clipping)	float	none	
WSCLIPSIGMA	DRS	multiple of sigma (sigma-clipping)	float	none	
WSCLIPNITER	DRS	number of iterations (sigma-clipping)	integer	none	
WSXWIDXDEG	DRS	polynomial degree along the x direction for x-width modeling	integer	none	
WSXWIDYDEG	DRS	polynomial degree along the y direction for x-width modeling	integer	none	
WSWLENXDEG	DRS	polynomial degree along the x direction for wavelength solution	integer	none	
WSWLENYDEG	DRS	polynomial degree along the y direction for wavelength solution	integer	none	

7.14 girSky - Running numbers of sky spectra to be used for the sky modeling

Purpose:
Reference Function: giNormalizeSky

girSky: description

This is an input, observation-dependent table identifying the sky spectra to be used for sky modeling. It is used only during the off-line processing. In the DFS pipeline all sky spectra as given by *ozpos* assignation table are taken. The header contains the information concerning the instrument setup and associated frame (KW FATHER).

girSky: working status

running number of the table:	10
table name:	girSky
author:	Cayatte
date:	20000623
status:	draft/template submitted to FDR
table source:	**user's input**
source comment:	Only in off-line mode

girSky: archiving technical informations

Total size is negligible (below 1MB)

girSky: Columns

Column	Description	Type	Unit	Comment
NSPEC	Running number of the spectrum	integer		

girSky: Header

Alias	DIC	Description	Type	Unit	Comment
TITLE	DRS	Title of data structure	string	none	Standard table KW
PURPOSE	DRS	Purpose of data structure	string	none	Standard table KW
AUTHOR	DRS	Source of data (responsible or processing recipe)	string	none	Standard table KW
DATE	CCDDCS	DIC: UT date when this file was written (ISO 8601).	string	none	DIC: UT date when this file was written
STATUS	DRS	Comment on last update	string	none	Standard table KW
GIRTTYPE	DRS	type of table data	string	none	TAB: 'NUMSKYTAB'
SLIT	FGIR_ICS	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies	integer	none	DIC: Position (hs). AL: What is the range, [0-4] ? I ma not happy with that.
GRATING	FGIR_ICS	DIC: ESO name for grating unit.	string	none	DIC: Grating common name (hls). AL: Caution: For FGIR_ICS KWs there is a dicrepancy in INS used by FGIR_ICS and INS1 required by VLT standard
WLEN0	FGIR_ICS	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength	double	nm	DIC: Grating central wavelength [nm] (hs).
FATHER	DRS	ID of father frame; used to associate an auxilliary data to original frame	string	none	
SKWLENMIN	DRS	SKY normalization: minimum wavelength	double	nm	
SKWLENMAX	DRS	SKY normalization: maximum wavelength	double	nm	

7.15 girSkyCoef - Polynomial model of the normalized sky

Purpose: Combined with *psfLwidth* table provides a complete parametric model of the sky for the current exposure
Reference Function: giModelSky

girSkyCoef: description

This table contains the coefficients of the polynomial describing the normalized (extracted) sky spectra. The sky is described, for each λ-bin, by a 1-D, K-degree polynomial using the λ-width of the standard PSF as the independent variable. Hence each row correspond to the λ-bin and each column to the coefficient of the polynomial.

This is an internal, object-dependent table produced and used by the pipeline. It is associated to the specific extracted frame and to the table giving the λ-width of the sky spectra.

girSkyCoef: working status

running number of the table:	14
table name:	girSkyCoef
author:	Cayatte
date:	20000623
status:	draft/template submitted to FDR
table source:	**processing**
source comment:	Produced by giModelSky; this table is a processing product. No initial solution exists.

girSkyCoef: archiving technical informations

Total size is negligible (below 1MB)

girSkyCoef: Columns

Column	Description	Type	Unit	Comment
LAMBDA	wavelength ogf the spectral bin	float	nm	
SNORMK	coefficients of 1-D polynomial of degree K in PSF width	float		Combined with $psfLwidth$ table it gives a complete description of the SKY model

girSkyCoef: Header

Alias	DIC	Description	Type	Unit	Comment
TITLE	DRS	Title of data structure	string	none	Standard table KW
PURPOSE	DRS	Purpose of data structure	string	none	Standard table KW
AUTHOR	DRS	Source of data (responsible or processing recipe)	string	none	Standard table KW
DATE	CCDDCS	DIC: UT date when this file was written (ISO 8601).	string	none	DIC: UT date when this file was written
STATUS	DRS	Comment on last update	string	none	Standard table KW
GIRTTYPE	DRS	type of table data	string	none	TAB: 'SKYNORMTAB'
SLIT	FGIR_ICS	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies	integer	none	DIC: Position (hs). AL: What is the range, [0-4] ? I ma not happy with that.
GRATING	FGIR_ICS	DIC: ESO name for grating unit.	string	none	DIC: Grating common name (hls). AL: Caution: For FGIR_ICS KWs there is a dicrepancy in INS used by FGIR_ICS and INS1 required by VLT standard
WLEN0	FGIR_ICS	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength	double	nm	DIC: Grating central wavelength [nm] (hs).
FATHER	DRS	ID of father frame; used to associate an auxilliary data to original frame	string	none	

Alias	DIC	Description	Type	Unit	Comment
SKMODEL	DRS	SKY modelling: 0=no spatial distribution, 1 or more=spatial distribution	integer	none	
SKCLIPNITER	DRS	number of iterations (sigma-clipping)	integer	none	
SKCLIPSIGMA	DRS	multiple of sigma (sigma-clipping)	float	none	
SKCLIPMFRAC	DRS	min fraction of points accepted/total (sigma-clipping)	float	none	
SKINTXY	DRS	Coefficients of (x,y) DEGXIS, DEGYIS degree normet-to-one polynomial describing the spatial variation of sky intensity. The coefficient are sorted in order x,y as c1+c2*x+c3*y+	float	none	replace table sInt
SKINTDEGX	DRS	X - degree of the polynomial describing the spatial variation of sky intensity	integer	none	replace table sInt
SKINTDEGY	DRS	X - degree of the polynomial describing the spatial variation of sky intensity	integer	none	replace table sInt

7.16 girSkyMom - Sky quality - 1-4th moments of the residual sky (after sky subtraction)

Purpose: table: wavelength, mean, 2nd, 3rd and 4th moments
Reference Function: giModelSky

girSkyMom: description

none

girSkyMom: working status and source informations

running number of the table:	17
table name:	girSkyMom
author:	Cayatte
date:	20000623
status:	draft/template submitted to FDR
table source:	**processing**
source comment:	Output table; this table is a processing product. No initial solution exists.

girSkyMom: archiving technical informations

comment concerning the size:		
total number of tables in MEDUSA mode:	58	
size per table	0.2	MBytes
total size of tables in MEDUSA mode:	**11.6**	MBytes
total number of tables in ARGUS/IFU mode:	87	
size per table	0.2	MBytes
total size of tables in ARGUS/IFU mode:	**17.4**	MBytes
Total for all modes	**29.0**	MBytes

girSkyMom: Columns

Column	Description	Type	Unit	Comment
LAMBDA	wavelength	float	nm	
RMEAN	mean	float		average residual of the sky spectra after model subtraction
RMOM2	2nd moment	float		average moments of residuals
RMOM3	3rd moment	float		
RMOM4	4th moment	float		

girSkyMom: Header

Alias	DIC	Description	Type	Unit	Comment
TITLE	DRS	Title of data structure	string	none	Standard table KW
PURPOSE	DRS	Purpose of data structure	string	none	Standard table KW
AUTHOR	DRS	Source of data (responsible or processing recipe)	string	none	Standard table KW
DATE	CCDDCS	DIC: UT date when this file was written (ISO 8601).	string	none	DIC: UT date when this file was written
STATUS	DRS	Comment on last update	string	none	Standard table KW
GIRTTYPE	DRS	type of table data	string	none	TAB: 'SKYMOMTAB'
SLIT	FGIR_ICS	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies	integer	none	DIC: Position (hs). AL: What is the range, [0-4] ? I ma not happy with that.
GRATING	FGIR_ICS	DIC: ESO name for grating unit.	string	none	DIC: Grating common name (hls). AL: Caution: For FGIR_ICS KWs there is a dicrepancy in INS used by FGIR_ICS and INS1 required by VLT standard
WLEN0	FGIR_ICS	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength	double	nm	DIC: Grating central wavelength [nm] (hs).
FATHER	DRS	ID of father frame; used to associate an auxilliary data to original frame	string	none	
SKMODEL	DRS	SKY modelling: 0=no spatial distribution, 1 or more=spatial distribution	integer	none	
SKCLIPNITER	DRS	number of iterations (sigma-clipping)	integer	none	
SKCLIPSIGMA	DRS	multiple of sigma (sigma-clipping)	float	none	
SKCLIPMFRAC	DRS	min fraction of points accepted/total (sigma-clipping)	float	none	

7.17 girSlitGeo - Geometry of slits - position of fibers

Purpose: Provide the prameters to the initial optical model used for localisation and PSF modeling

Reference Function: giGetWaveSolution

girSlitGeo: description

This table gives the description of the slit geometry. The table selection is made by the match on KW SLIT. It is necessary to establish the wavelength calibration. This is a static table provided initially by the mechanical (collumns NS, NSUBSLIT, XF, YF and ZF) and optical (ZFDEFOCUS) designs with possible adjustement during the commissionning. The column BUTTON is the only link between the physical geometry of the slit and the positionner. The BUTTON value gives access key to the informations contained in the `ozpoz` table concerning.

girSlitGeo: working status

running number of the table:	18
table name:	girSlitGeo
author:	Blecha
date:	20000710
status:	draft/template submitted to FDR
table source:	**archive**
source comment:	From GIRAFFE design.

girSlitGeo: archiving technical informations

Total size is negligible (below 1MB)

girSlitGeo: Columns

Column	Description	Type	Unit	Comment
NSPEC	running number of the spectrum/fibre	integer		
BUTTON	running number of the positionner button associated to the fibre	integer		This is the only link between the physical geometry of the slit and the positionner. The BUTTON value gives access key to the informations contained in the `ozpoz` tabl+
STATUS	numerical code giving the permanent fibre status (1 for valid fibre, 0 unused)	integer		Other codes could be used in the future
COMMENT	any comment concerning this fibre	string		
NSUBSLIT	running number of the subslit	integer		
XF	Fiber position in x on the slit	float	m	Reference point TBD
YF	Fiber position in y on the slit	float	m	Reference point TBD
ZF	Fiber position in z on the slit	float	m	Reference point is focal surface at [XF,YF]
ZFDEFOCUS	Fiber distance from the focal plane	float	m	+ toward the spectrograph
XIND	Fiber position in x in focal plane image	integer	pixel	Reference point TBD
YIND	Fiber position in y in focal plane image	integer	pixel	Reference point TBD

girSlitGeo: Header

Alias	DIC	Description	Type	Unit	Comment
TITLE	DRS	Title of data structure	string	none	Standard table KW
PURPOSE	DRS	Purpose of data structure	string	none	Standard table KW
AUTHOR	DRS	Source of data (responsible or processing recipe)	string	none	Standard table KW
DATE	CCDDCS	DIC: UT date when this file was written (ISO 8601).	string	none	DIC: UT date when this file was written
STATUS	DRS	Comment on last update	string	none	Standard table KW

Alias	DIC	Description	Type	Unit	Comment
GIRTTYPE	DRS	type of table data	string	none	TAB: 'SLTGEOMTAB'
SLIT	FGIR_ICS	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies	integer	none	DIC: Position (hs). AL: What is the range, [0-4] ? I ma not happy with that.
SLITSIM	-	TAB: function is software simulated			

7.18 girSpectroConst - Spectrograph Setup Constants

Purpose: Provide the setup dependent constants
Reference Function: TBD

girSpectroConst: description

This, unique look-up table gives all informations specific to the spectrograph setup and necessary to the data reduction. Most of data are very likely redundant to those used by the OS for the actual instrument setup and it this therefore MANDATORY that this table is updated if there is modification in the available instrument setups.
The KWs SLIT, GRATING and WLEN0 are used to select the line defining the actual values of DRS KWs starting in Column 4. This is done in the specific DRS function which creates all missing KWs and assign values according to this table
Any unforseen setup-dependent constant will be added in this table as a new column.

girSpectroConst: working status

running number of the table:	20
table name:	girSpectroConst
author:	Blecha
date:	20001109
status:	1-st introduction according to the progress Meeting
table source:	**From Spectrograph optical design; 1 line per setup**
source comment:	From Spectrograph optical design (see description)

girSpectroConst: archiving technical informations

Total size is negligible (below 1MB)

girSpectroConst: Columns

Column	Description	Type	Unit	Comment
SLIT	Index of the position of the slit selection carriage	integer	none	Input - Lookup value of the KW SLIT
GRATING	Grating common name	string	none	Input - Lookup value of s the KW GRATING
WLEN0	Grating central wavelength	double	nm	Input - Lookup value of the KW WLEN0
FOCLEN	focal length of the collimator	double	m	Output
CAMGFACT	magnification factor of the camera	float	none	Output
CSPACE	groove spacing	float	m	Output
GRATORD	Order of the grating	integer	none	Output
WLENMIN	Shortest usable wavelength in the current setup	float	nm	Output
WLENMAX	Longest usable wavelength in the current setup	float	nm	Output

girSpectroConst: Header

Alias	DIC	Description	Type	Unit	Comment
TITLE	DRS	Title of data structure	string	none	Standard table KW
PURPOSE	DRS	Purpose of data structure	string	none	Standard table KW
AUTHOR	DRS	Source of data (responsible or processing recipe)	string	none	Standard table KW
DATE	CCDDCS	DIC: UT date when this file was written (ISO 8601).	string	none	DIC: UT date when this file was written
STATUS	DRS	Comment on last update	string	none	Standard table KW
GIRTTYPE	DRS	TAB: type of data type of table data	string	none	TAB: 'SPECTROCONST'

7.19 girTransVlt - Telescope and Nasmyth corrector transmission

Purpose: Telescope + Nasmyth corrector on-axis spectral transmission
Reference Function: giFluxConv

girTransVlt: description

On axis optical transmissions ares given for the assembly telescope + Nasmyth corrector + the Atmospheric Dispersion Corrector - ADC (the last one for the ARGUS mode only) . This transmission does not include possible variations within the field of view which is given through the table girVignCorrTbale describing the radial vignetting effects. Data are to be supplied by the optical group.

girTransVlt: working status

running number of the table:	16
table name:	girTransVlt
author:	Cayatte
date:	20000623
status:	draft/template submitted to FDR
table source:	**archive**
source comment:	From VLT maintenance

girTransVlt: archiving technical informations

Total size is negligible (below 1MB)

girTransVlt: Columns

Column	Description	Type	Unit	Comment
LAMBDA	wavelength	float	nm	
TRANSVLTN	transmission VLT+Nasmyth corrector for wavelength LAMBDA	float	none	
TRANSADC	transmission of the ADC (ARGUS only)	float	none	

girTransVlt: Header

Alias	DIC	Description	Type	Unit	Comment
TITLE	DRS	Title of data structure	string	none	Standard table KW
PURPOSE	DRS	Purpose of data structure	string	none	Standard table KW
AUTHOR	DRS	Source of data (responsible or processing recipe)	string	none	Standard table KW
DATE	CCDDCS	DIC: UT date when this file was written (ISO 8601).	string	none	DIC: UT date when this file was written
STATUS	DRS	Comment on last update	string	none	Standard table KW

Alias	DIC	Description	Type	Unit	Comment
GIRTTYPE	DRS	type of table data	string	none	TAB: 'VLTTRANSTAB'
SLIT	FGIR_ICS	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies	integer	none	DIC: Position (hs). AL: What is the range, [0-4] ? I ma not happy with that.
GRATING	FGIR_ICS	DIC: ESO name for grating unit.	string	none	DIC: Grating common name (hls). AL: Caution: For FGIR_ICS KWs there is a dicrepancy in INS used by FGIR_ICS and INS1 required by VLT standard
WLEN0	FGIR_ICS	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength	double	nm	DIC: Grating central wavelength [nm] (hs).
TELESCOPE	TCS	DIC: cmm version number of the integration module (tcsBUILD/ntt), which defines the version of all TCS modules.	string	none	DIC: TCS version number

7.20 girVignCorr - Nasmyth vigneting (screen and telescope)

Purpose:
Reference Function: giGetWaveSolution

girVignCorr: description

This unique table gives the ratio of the flux entering a fibre on the optical axis (R=0) to that we will have at the position R. Two different columns VIGNMEDUSA and VIGNIFU correspond to MEDUSA and IFU mode (it is assumed that no correction for ARGUS is necessary). Wavelength is sampled so that at least central wavelength of all possible setups are represented and the radial position so that the variation between two adjacent R-shells is smaller than 1 percent. To correct for vignetting the raw signal has to be divided by the VIGNMEDUSA or VIGNIFU factor.
Two types of instances are available, selected according the KWs DCATG="NASMYTH CALIBRATION" or SCIENCE and also LAMP and LAMPID in the first case.
In case of SCIENCE exposure two effects are included: The achromatic vignetting effect of the M2 and M3 mirrors of the VLT, and the chromatic "vignetting" effect induced by the aberrations of the Nasmyth corrector.
In case of NASMYTH CALIBRATION, only the chromatic "vignetting" effect induced by the aberrations of the Nasmyth corrector is included.

girVignCorr: working status

running number of the table:	19
table name:	girVignCorr
author:	Blecha
date:	20000710
status:	draft/template submitted to FDR
table source:	**archive**
source comment:	From GIRAFFE optical design (study pending).

girVignCorr: archiving technical informations

Total size is negligible (below 1MB)

girVignCorr: Columns

Column	Description	Type	Unit	Comment
LAMBDA	wavelength	float	nm	Central wavelength for which the vignetting was computed
R	distance in the Nasmyth focal plane from optical axis	float	m	
VIGNMEDUSA	fraction of the light compared to what we would have at R=0 position	float	none	MEDUSA mode, the recorded signal si TB divided by this number to correct vignetting effect
VIGNIFU	fraction of the light compared to what we would have at R=0 position	float	none	IFU mode, the recorded signal si TB divided by this number to correct vignetting effect

girVignCorr: Header

Alias	DIC	Description	Type	Unit	Comment
TITLE	DRS	Title of data structure	string	none	Standard table KW
PURPOSE	DRS	Purpose of data structure	string	none	Standard table KW
AUTHOR	DRS	Source of data (responsible or processing recipe)	string	none	Standard table KW
DATE	CCDDCS	DIC: UT date when this file was written (ISO 8601).	string	none	DIC: UT date when this file was written
STATUS	DRS	Comment on last update	string	none	Standard table KW
GIRTTYPE	DRS	type of table data	string	none	TAB: 'VIGNETTAB'
SLIT	FGIR_ICS	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies	integer	none	DIC: Position (hs). AL: What is the range, [0-4] ? I ma not happy with that.
GRATING	FGIR_ICS	DIC: ESO name for grating unit.	string	none	DIC: Grating common name (hls). AL: Caution: For FGIR_ICS KWs there is a dicrepancy in INS used by FGIR_ICS and INS1 required by VLT standard
DCATG	DPR	DIC: Category according to section 4.7 of GEN-SPE-ESO-19400-0794. (science, calib, test, simulation, other)	string	none	DIC: Observation category AL: Used also by DFS to select recipe
LAMP	DRS	Name of the W Calibration source	string	none	One of ThAr,Ne,ThArNe or 'SKY' (telluric lines)
LAMPID	FP_LP		undef	none	

7.21 girVignSpectro - Spectrograph Internal Vignetting

Purpose:
Reference Function: giFiberTrans

girVignSpectro: description

This table contains the modeled values of the spectrograph relative transmission (assumed to be due essentially to vignetting) between the different spectra. For each spectrum, the table gives a single number which is the ratio of the wavelength integrated light reaching the CCD to the light incident on the spectrograph from the fiber exit on the slit. There are as many such tables as instrumental set-ups. This table is used by the *giFiberTrans*() function to compute the relative transmissions of the fibers.

girVignSpectro: working status

running number of the table:	22
table name:	girVignSpectro
author:	North
date:	20010121
status:	1-st introduction TBC with PN
table source:	**TBD**
source comment:	

girVignSpectro: archiving technical informations

Total size is negligible (below 1MB)

girVignSpectro: Columns

Column	Description	Type	Unit	Comment
NSPEC	Runnibg nuber of the spectrum	integer		
TRANSPEC	(relative) spectrograph transmission integrated on the x-direction	float		

girVignSpectro: Header

Alias	DIC	Description	Type	Unit	Comment
TITLE	DRS	Title of data structure	string	none	Standard table KW
PURPOSE	DRS	Purpose of data structure	string	none	Standard table KW
AUTHOR	DRS	Source of data (responsible or processing recipe)	string	none	Standard table KW
DATE	CCDDCS	DIC: UT date when this file was written (ISO 8601).	string	none	DIC: UT date when this file was written
STATUS	DRS	Comment on last update	string	none	Standard table KW
GIRTTYPE	DRS	type of table data	string	none	
SLIT	FGIR_ICS	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies	integer	none	DIC: Position (hs). AL: What is the range, [0-4] ? I ma not happy with that.
GRATING	FGIR_ICS	DIC: ESO name for grating unit.	string	none	DIC: Grating common name (hls). AL: Caution: For FGIR_ICS KWs there is a dicrepancy in INS used by FGIR_ICS and INS1 required by VLT standard
WLEN0	FGIR_ICS	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength	double	nm	DIC: Grating central wavelength [nm] (hs).

7.22 girWavCoef - Coefficients of the Polynomial Wavelength Solution

Purpose: Provide and save wavelengt solution

Reference Function: giGetWaveSolution

girWavCoef: description

This table contains the coefficients of the wavelength solution for each GIRAFFE and slit setups. The table selection is made by the match on KWs SLIT, GRATING and WLEN0. There are two kinds of solutions: those linked with the optical model of the spectrograph computed from KWs FOCLEN, CAMGFACT, GRATROT and GSPACE. Note that only the KWs SLIT, WLEN0 and GRATING are supplied by OS while all other spectrograph constant are obtained via the girSpectroConstTable; there may be additional parametres that are purely empirical and take into account any departures from the optical model, due to second-order optical effects and/or CCD deformation. The CIJ coefficients given in this table refer only to the empirical fit to the residuals relative to the optical model; the XF and YF coordinates are used in the optical solution, which also uses (or adjusts) the KWs mentioned above, which are not repeated in this table.
This table is used to keep through the header KW's last standard solution of the wavelength equation (true AB -> FR ?).

girWavCoef: working status

running number of the table:	6
table name:	girWavCoef
author:	Blecha
date:	20000605
status:	draft/template submitted to FDR
table source:	**archive, updated during processing**
source comment:	The first instance is produced during the COMMISSIONING (0-version in Europe with GIRAI

girWavCoef: archiving technical informations

Total size is negligible (below 1MB)

girWavCoef: Columns

Column	Description	Type	Unit	Comment
NSPEC	Running number of the spectrum	integer	none	
XF	Fiber position in x on the slit	float	m	used in the optical solution only
YF	Fiber position in y on the slit	float	m	used in the optical solution only
CIJ	Coefficients of the wavelength solution (for the residuals relative to the optical model)	double	nm	same value for a given subslit (residuals)

girWavCoef: Header

Alias	DIC	Description	Type	Unit	Comment
TITLE	DRS	Title of data structure	string	none	Standard table KW
PURPOSE	DRS	Purpose of data structure	string	none	Standard table KW
AUTHOR	DRS	Source of data (responsible or processing recipe)	string	none	Standard table KW
DATE	CCDDCS	DIC: UT date when this file was written (ISO 8601).	string	none	DIC: UT date when this file was written
STATUS	DRS	Comment on last update	string	none	Standard table KW
GIRTTYPE	DRS	type of table data	string	none	TAB: 'WSOLTAB'
SLIT	FGIR_ICS	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies	integer	none	DIC: Position (hs). AL: What is the range, [0-4] ? I ma not happy with that.

Alias	DIC	Description	Type	Unit	Comment
GRATING	FGIR_ICS	DIC: ESO name for grating unit.	string	none	DIC: Grating common name (hls). AL: Caution: For FGIR_ICS KWs there is a dicrepancy in INS used by FGIR_ICS and INS1 required by VLT standard
WLEN0	FGIR_ICS	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength	double	nm	DIC: Grating central wavelength [nm] (hs).
FOCLEN	DRS	focal length of the collimator	double	m	
CAMGFACT	DRS	magnification factor of the camera	float	none	Name changed from GCCD-SCALE to CAMGFAC
GRATROT	FGIR_ICS	DIC: Rotation angle of grating. The reference frame is given in the corresponding instrument specification. ENC = OFFST+ RESOL* RAD2DEG* arcsin(WLEN* GRV/(2* cos(ROT))) AL: Ro+	double *float	deg	DIC: Grating rot angle [deg] (hl).
GSPACE	FGIR_ICS	DIC: Number of grooves per nanometer ENC = OFFST+ RESOL* RAD2DEG* arcsin(WLEN* GRV/(2* cos(ROT)))	double	none	DIC: Grating grooves/nm (hl).
WSCOEFi	DRS	PSF X-width coef	float	none	
WSLINETYPE	DRS	use of ThArNe or telluric lines for Wavelength Solution	logical	none	
WSWINDOW	DRS	width in pixel for line detection and fit	float	pixel	
WSCLIPMFRAC	DRS	min fraction of points accepted/total (sigma-clipping)	float	none	
WSCLIPSIGMA	DRS	multiple of sigma (sigma-clipping)	float	none	
WSCLIPNITER	DRS	number of iterations (sigma-clipping)	integer	none	
WSXWIDXDEG	DRS	polynomial degree along the x direction for x-width modeling	integer	none	
WSXWIDYDEG	DRS	polynomial degree along the y direction for x-width modeling	integer	none	
WSWLENXDEG	DRS	polynomial degree along the x direction for wavelength solution	integer	none	
WSWLENYDEG	DRS	polynomial degree along the y direction for wavelength solution	integer	none	

7.23 DIC alias table

Alias	Paramname	DIC	Type	Format	Unit
ACTMJD	MJD-OBS	PR_FITS	double *real	%.8f	days
AIRMASS	AIRMASS	PR_FITS	double	%.5f	none
ALLOCFACB	undef	FPMISS	integer		none
ALLOCOBJ	undef	FPMISS	integer		none
ALLOCSKY	undef	FPMISS	integer		none
APPDEC	undef	FPMISS	real		radians
APPEPOCH	undef	FPMISS	real		none
APPRA	undef	FPMISS	real		radians
ARGSCALE	INS ARGS SCALE	FP_LP	string	%10s	none
ATMPRES	GEN AMBI PRES*	ASM	undef		none
ATMRHUM	GEN AMBI RHUMi*	ASM	undef		none
ATMTEMP	GEN AMBI TEMPi*	ASM	undef		none
AUTHOR	PRO DATA AUTHOR	DRS	string		none
BADPIXMFRAC	PRO EST BADPIX MFRAC	DRS	float		none
BCLIPMFRAC	PRO BIAS SIGMA MFRAC	DRS	float		none
BCLIPNITER	PRO BIAS SIGMA NITER	DRS	integer		none
BCLIPSIGMA	PRO BIAS SIGMA MULT	DRS	float		none
BIASPLANE	PRO BIAS PLANEn	DRS	float		none
BIASSIGMA	PRO BIAS SIGMA	DRS	float		none
BIASVALUE	PRO BIAS VALUE	DRS	float		none
BINWL0	PRO WLEN REBIN WL0	DRS	float		nm
BINWNX	PRO WLEN REBIN NX	DRS	integer		none
BINWNY	PRO WLEN REBIN NY	DRS	integer		none
BINWSTEP	PRO WLEN REBIN WSTEP	DRS	double		nm
BINWSTEPLOG	PRO WLEN REBIN WSTEPLOG	DRS	double		none
CAMGFACT	DRS CAM GFACT	DRS	float		none
CCDID	DET ID	CCDDCS	string	%30s	none
CCDPIXTIME	DET READ PIXTIME	CCDDCS *DRS	double	%.1f	us
CCLIPMFRAC	PRO CRH SIGMA MFRAC	DRS	float		none
CCLIPNITER	PRO CRH SIGMA NITER	DRS	integer		none
CCLIPSIGMA	PRO CRH SIGMA MULT	DRS	float		none
CENDEC	undef	FPMISS	real		radians
CENEQNX	undef	FPMISS	real		none
CENRA	undef	FPMISS	real		radians
CONAD	DET OUTi CONAD	CCDDCS	double	%.2f	e-/ADU
CONFMJD	undef	FPMISS	real		none
CSIGMAMULT	PRO CRH SIGMA MULTX	DRS	float		none
DARKEXPECT	PRO DARK EXPECT	DRS	float		e-
DARKMETHOD	PRO DARK METHOD	DRS	string		none
DARKTHRESH	PRO DARK THRESH	DRS	float		e-
DARKVALUE	PRO DARK VALUE	DRS	float		e-
DATAMAX	DATAMAX	PR_FITS	double	%.7f	none *e-
DATAMEAN	PRO DATAMEAN	PRO	double	%e	none *e-
DATAMEDI	PRO DATAMEDI	PRO	double	%e	none *e-
DATAMIN	DATAMIN	PR_FITS	double	%.7f	none *e-
DATAMOD	PRO DATAMOD	DRS	float		e-
DATASIG	PRO DATASIG	PRO	double	%e	none *e-
DATE	DATE	CCDDCS	string	%30s	none
DCATG	DPR CATG	DPR	string	%15s	none
DTYPE	DPR TYPE	DPR	string	%15s	none
ECLIPMFRAC	PRO EXT SIGMA MFRAC	DRS	float		none
ECLIPNITER	PRO EXT SIGMA NITER	DRS	integer		none
ECLIPSIGMA	PRO EXT SIGMA MULT	DRS	float		none
EXPTIME	INS DETi UIT	UVES_ICS	double	%.3f	sec
EXTBKGCOEF	PRO EXT BACKGD COEF	DRS	integer		none

Alias	Paramname	DIC	Type	Format	Unit
EXTBKGYDEG	PRO EXT BACKGD YDEG	DRS	integer		none
EXTMETHOD	PRO EXT METHOD	DRS	integer		none
EXTNS	PRO EXT NY	DRS	integer		none
EXTNX	PRO EXT NX	DRS	integer		none
EXTPSFAMPL	PRO EXT PSF AMPL	DRS	float		none
EXTPSFBKGD	PRO EXT PSF BKGD	DRS	float		none
EXTPSFPOS	PRO EXT PSF POS	DRS	float		none
EXTPSFWID1	PRO EXT PSF WID1	DRS	float		none
EXTPSFWID2	PRO EXT PSF WID2	DRS	float		none
EXTPSFWID3	PRO EXT PSF WID3	DRS	float		none
EXTWMULT	PRO EXT PSFPN	DRS	double		none
FATHER	PRO DATA FATHER	DRS	string		none
FFNORM	PRO FF NORM	DRS	float		none
FIBRESKOi	INS FIBRESi KO	MISSING	integer		none
FILENAME	undef	FPMISS	string		none
FLCONVSTAND	PRO DATA FLCVS	DRS	logical		none
FOCLEN	DRS COL FOCAL	DRS	double		m
FRAME	PRO DATA SKYFRAME	DRS	string		none
GIRFTYPE	PRO FRAME TYPE	DRS	string		none
GIRTTYPE	PRO TABLE TYPE	DRS	string		none
GRATING	INS GRAT NAMEi	FGIR_ICS	string	%30s	none
GRATORD	DRS GRAT ORDER	DRS	integer		none
GRATRES	INS GRAT RESOL	FGIR_ICS	double	%.1f	none
GRATROT	INS GRAT ROT	FGIR_ICS	double *float	%.3f	deg
GSPACE	INS GRAT GRVi	FGIR_ICS	double	%.7f	none
LABEL	undef	FPMISS	string		none
LAMP	PRO DATA LSOURCE	DRS	string		none
LAMPDENS	INS DENS VALUE	MISSING	float		percent
LAMPID	INS LAMPi ID*	FP_LP	undef		none
LCLIPMFRAC	PRO LOC SIGMA MFRAC	DRS	float		none
LCLIPNITER	PRO LOC SIGMA NITER	DRS	integer		none
LCLIPSIGMA	PRO LOC SIGMA MULT	DRS	float		none
LINLIMIT	DRS CCD LINLIM	DRS	double		none
LPOLYDEG	PRO LOC POLYDEG	DRS	integer		none
MAXRADIUS	PRO EST AREA MAX	DRS	float		pixel
METRIC	PRO EST METRIC	DRS	integer		none
NOISEMULT	PRO LOC NOISE MULT	DRS	float		none
NX	PRO DATA NX	DRS	integer		none
NY	PRO DATA NY	DRS	integer		none
OVSCX	DET OUTi OVSCX	CCDDCS	integer	%d	none
OVSCY	DET OUTi OVSCY	CCDDCS	integer	%d	none
PRESCX	DET OUTi PRSCX	CCDDCS	integer	%d	none
PRESCY	DET OUTi PRSCY	CCDDCS	integer	%d	none
PURPOSE	PRO DATA PURPOSE	DRS	string		none
REFSRC	GEN OBJ REF	MISSING	string		none
REFSRCID	GEN OBJ REFID	MISSING	string		none
REFSRCMULT	GEN OBJ REFMULT	MISSING	float		normed
REFSRCOM	GEN OBJ REFCOM	MISSING	string		none
RON	DRS CCD RON	DRS	float		e-
SATLEVEL	DRS CCD SATLEV	DRS	integer		e-
SKCLIPMFRAC	PRO SKY SIGMA MFRAC	DRS	float		none
SKCLIPNITER	PRO SKY SIGMA NITER	DRS	integer		none
SKCLIPSIGMA	PRO SKY SIGMA MULT	DRS	float		none
SKCOMP	PRO SKY COMPONENT	DRS	string		none
SKINTDEGX	PRO SKY INTDEGX	DRS	integer		none
SKINTDEGY	PRO SKY INTDEGY	DRS	integer		none
SKINTXY	PRO SKY INTXYi	DRS	float		none

Alias	Paramname	DIC	Type	Format	Unit
SKMODEL	PRO SKY MODEL	DRS	integer		none
SKSUB	PRO SKY SUBTRACTED	DRS	string		none
SKSUBMETHOD	PRO SKY SUB METHOD	DRS	integer		none
SKWLENMAX	PRO SKY WLEN MAX	DRS	double		nm
SKWLENMIN	PRO SKY WLEN MIN	DRS	double		nm
SLIT	INS SLITS NO	FGIR_ICS	integer	%d	none
SLMODEL	PRO SCL MODEL	DRS	string		none
SLPHFFCOR	PRO SCL PHFFCOR	DRS	logical		none
SLPOLYDEG1	PRO SCL POLYDEG1	DRS	integer		none
SLPOLYDEG2	PRO SCL POLYDEG2	DRS	integer		none
STARNAME	PRO DATA STAR	DRS	logical		none
STARTX	DETi WINi STRX	CCDDCS	integer	%d	none
STARTY	DETi WINi STRY	CCDDCS	integer	%d	none
STATUS	PRO DATA STATUS	DRS	string		none
TELESCOPE	TEL ID	TCS	string	%30s	none
TITLE	PRO DATA TITLE	DRS	string		none
UNALLFACB	undef	FPMISS	integer		none
UNALLOBJ	undef	FPMISS	integer		none
UNALLSKY	undef	FPMISS	integer		none
WLEN0	INS GRAT WLEN	FGIR_ICS	double	%.1f	nm
WLENMAX	DRS GRAT WLORDMAX	DRS	float		nm
WLENMIN	DRS GRAT WLORDMIN	DRS	float		nm
WSCLIPMFRAC	PRO WSOL SIGMA MFRAC	DRS	float		none
WSCLIPNITER	PRO WSOL SIGMA NITER	DRS	integer		none
WSCLIPSIGMA	PRO WSOL SIGMA MULT	DRS	float		none
WSCOEFi	PRO WSOL NX	DRS	float		none
WSLINETYPE	PRO WSOL LINETYPE	DRS	logical		none
WSNS	PRO WSOL NY	DRS	integer		pixel
WSWINDOW	PRO WSOL WINDOW	DRS	float		pixel
WSWLENXDEG	PRO WSOL WLEN XDEG	DRS	integer		none
WSWLENYDEG	PRO WSOL WLEN YDEG	DRS	integer		none
WSXWIDXDEG	PRO WSOL XWIDTH XDEG	DRS	integer		none
WSXWIDYDEG	PRO WSOL XWIDTH YDEG	DRS	integer		none

7.24 Complete description of data dictionnary

Alias	ACTMJD
Dicname	PR_FITS
Paramname	MJD-OBS
Type	double *real
Format	%.8f
Unit	days
Function	
Author	
Comment	DIC: Obs start
Paramdesc	DIC: Modified Julian Day of the start of the exposure. The MJD is related to the Julian Day (JD) via the formula: MJD = JD-2400000.5 The comment includes a civil representation of the date and time. 8 decimals are required for a precision of one millisecond, 5 decimals for a precision of one second. AL: MJD at which the field was observed
Context	FITS
Dicstatus	Released
Dicrev	1.0
I-Reffunctions	
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	girOzpoz
Nref	1

Alias	AIRMASS
Dicname	PR_FITS
Paramname	AIRMASS
Type	double
Format	%.5f
Unit	none
Function	giFluxConv
Author	simond
Comment	DIC: Averaged air mass
Paramdesc	DIC: Average airmass for the optical axis during the exposure computed for the time while the shutter is open as described in GEN-SPE-ESO-00000-0794.
Context	FITS
Dicstatus	Released
Dicrev	1.0
I-Reffunctions	giFluxConv.skySubSp
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	
Nref	1

Alias	ALLOCFACB
Dicname	FPMISS
Paramname	undef
Type	integer
Format	
Unit	none
Function	
Author	
Comment	
Paramdesc	number of allocated guide stars
Context	
Dicstatus	Missing (TBD by FLAMES consortium)
Dicrev	
I-Reffunctions	
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	girOzpoz
Nref	1

Alias	ALLOCOBJ
Dicname	FPMISS
Paramname	undef
Type	integer
Format	
Unit	none
Function	
Author	
Comment	
Paramdesc	number of allocated program objects
Context	
Dicstatus	Missing (TBD by FLAMES consortium)
Dicrev	
I-Reffunctions	
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	girOzpoz
Nref	1

Alias	ALLOCSKY
Dicname	FPMISS
Paramname	undef
Type	integer
Format	
Unit	none
Function	
Author	
Comment	
Paramdesc	number of allocated sky positions
Context	
Dicstatus	Missing (TBD by FLAMES consortium)
Dicrev	
I-Reffunctions	
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	girOzpoz
Nref	1

Alias	APPDEC
Dicname	FPMISS
Paramname	undef
Type	real
Format	
Unit	radians
Function	
Author	
Comment	value calculated by P2PP
Paramdesc	apparent declination
Context	
Dicstatus	Missing (TBD by FLAMES consortium)
Dicrev	
I-Reffunctions	
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	girOzpoz
Nref	1

Alias	APPEPOCH
Dicname	FPMISS
Paramname	undef
Type	real
Format	
Unit	none
Function	
Author	
Comment	value calculated by P2PP
Paramdesc	epoch of the above
Context	
Dicstatus	Missing (TBD by FLAMES consortium)
Dicrev	
I-Reffunctions	
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	girOzpoz
Nref	1

Alias	APPRA
Dicname	FPMISS
Paramname	undef
Type	real
Format	
Unit	radians
Function	
Author	
Comment	value calculated by P2PP
Paramdesc	apparent right ascension
Context	
Dicstatus	Missing (TBD by FLAMES consortium)
Dicrev	
I-Reffunctions	
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	girOzpoz
Nref	1

Alias	ARGSCALE
Dicname	FP_LP
Paramname	INS ARGS SCALE
Type	string
Format	%10s
Unit	none
Function	
Author	blecha
Comment	DIC: Argus scale AL: TBadded in the header ozpoz
Paramdesc	DIC: There are two possible scale selections "1:1" "1:1.67"
Context	Instrument
Dicstatus	Submitted
Dicrev	0.1
I-Reffunctions	
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	
Nref	0

Alias	ATMPRES
Dicname	ASM
Paramname	GEN AMBI PRES*
Type	undef
Format	
Unit	none
Function	giFluxCon
Author	blecha
Comment	To be matched with model pressure
Paramdesc	
Context	
Dicstatus	
Dicrev	
I-Reffunctions	giFluxConv.skySubSp
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	
Nref	1

Alias	ATMRHUM
Dicname	ASM
Paramname	GEN AMBI RHUMi*
Type	undef
Format	
Unit	none
Function	giFluxCon
Author	blecha
Comment	To be matched with model rhum
Paramdesc	
Context	
Dicstatus	
Dicrev	
I-Reffunctions	giFluxConv.skySubSp
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	
Nref	1

Alias	ATMTEMP
Dicname	ASM
Paramname	GEN AMBI TEMPi*
Type	undef
Format	
Unit	none
Function	giFluxCon
Author	blecha
Comment	To be matched with model temperature
Paramdesc	
Context	
Dicstatus	
Dicrev	
I-Reffunctions	giFluxConv.skySubSp
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	
Nref	1

Alias	AUTHOR
Dicname	DRS
Paramname	PRO DATA AUTHOR
Type	string
Format	
Unit	none
Function	
Author	blecha
Comment	Standard table KW
Paramdesc	Source of data (responsible or processing recipe)
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	giTableStandard
Nref	1

Alias	BADPIXMFRAC
Dicname	DRS
Paramname	PRO EST BADPIX MFRAC
Type	float
Format	
Unit	none
Function	giEstimatePix
Author	royer
Comment	
Paramdesc	min fraction of neighboring bad pixels accepted/total
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giEstimatePix.estBadPixMap giEstimatePix.estimateImage giEstimatePix.estimateNpixels giEstimatePix.estimateSigmaImage
Aiprefs	
Aoprefs	
Reftables	
Nref	4

Alias	BCLIPMFRAC
Dicname	DRS
Paramname	PRO BIAS SIGMA MFRAC
Type	float
Format	
Unit	none
Function	giGetRemoveBias
Author	simond
Comment	
Paramdesc	min fraction of points accepted/total (sigma-clipping)
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giGetRemoveBias.bSubImage
Aiprefs	
Aoprefs	
Reftables	
Nref	1

Alias	BCLIPNITER
Dicname	DRS
Paramname	PRO BIAS SIGMA NITER
Type	integer
Format	
Unit	none
Function	giGetRemoveBias
Author	simond
Comment	
Paramdesc	number of iterations (sigma-clipping)
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giGetRemoveBias.bSubImage
Aiprefs	
Aoprefs	
Reftables	
Nref	1

Alias	BCLIPSIGMA
Dicname	DRS
Paramname	PRO BIAS SIGMA MULT
Type	float
Format	
Unit	none
Function	giGetRemoveBias
Author	simond
Comment	
Paramdesc	multiple of sigma (sigma-clipping)
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giGetRemoveBias.bSubImage
Aiprefs	
Aoprefs	
Reftables	
Nref	1

Alias	BIASPLANE
Dicname	DRS
Paramname	PRO BIAS PLANEn
Type	float
Format	
Unit	none
Function	giGetRemoveBias
Author	blecha
Comment	
Paramdesc	3 coefficients of the x,y plane describing the fitted bias=B0+Bx*X+By*Y; X,Y origine is at the first active pixel.
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giGetRemoveBias.bSubImage
Aiprefs	
Aoprefs	
Reftables	
Nref	1

Alias	BIASSIGMA
Dicname	DRS
Paramname	PRO BIAS SIGMA
Type	float
Format	
Unit	none
Function	giGetRemoveBias
Author	simond
Comment	
Paramdesc	Computed bias sigma in pre/over-scan
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giGetRemoveBias.bSubImage
Aiprefs	
Aoprefs	
Reftables	
Nref	1

Alias	BIASVALUE
Dicname	DRS
Paramname	PRO BIAS VALUE
Type	float
Format	
Unit	none
Function	giGetRemoveBias
Author	simond
Comment	
Paramdesc	Computed average bias level in pre/over-scan
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	giGetRemoveBias.biasMast
O-Reffunctions	giGetRemoveBias.bSubImage
Aiprefs	
Aoprefs	
Reftables	
Nref	2

Alias	BINWL0
Dicname	DRS
Paramname	PRO WLEN REBIN WL0
Type	float
Format	
Unit	nm
Function	giRebinSpectra
Author	blecha
Comment	
Paramdesc	wavelength of the centre of the first spectral elements
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	giFluxConv.skySubSp giModelSky.normSky giModelSky.normSkyErr giModelSky.psflwidth giNormalizeSky.wlFfSp giNormalizeSky.wlFfSpErr giSplitSky.psflwidth giSplitSky.wlFfSp giSplitSky.wlFfSpErr giSubtractSky.skyModel giSubtractSky.skyModelErr giSubtract-Sky.wlFfSp giSubtractSky.wlFfSpErr giVignCorr.wlFfSp giVignCorr.wlFfSpErr
O-Reffunctions	giNormalizeSky.normSky giNormalizeSky.normSkyErr giRebinSpectra.psflwidth giRebinSpectra.wlFfSp giRebinSpectra.wlFfSpErr giSplitSky.wlFfSkyComp giSplit-Sky.wlFfSkyCompErr giVignCorr.wlFfVcSp giVignCorr.wlFfVcSpErr
Aiprefs	
Aoprefs	
Reftables	
Nref	24

Alias	BINWNX
Dicname	DRS
Paramname	PRO WLEN REBIN NX
Type	integer
Format	
Unit	none
Function	giRebinSpectra
Author	blecha
Comment	
Paramdesc	number of spectral elements on the rebinned S2dFrame
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	giFluxConv.skySubSp giModelSky.normSky giModelSky.normSkyErr giModelSky.psflwidth giNormalizeSky.wlFfSp giNormalizeSky.wlFfSpErr giRebinSpectra.ffSp giSplitSky.psflwidth giSplitSky.wlFfSp giSplitSky.wlFfSpErr giSubtractSky.skyModel giSubtractSky.skyModelErr giSubtractSky.wlFfSp giSubtractSky.wlFfSpErr giVignCorr.wlFfSp giVignCorr.wlFfSpErr
O-Reffunctions	giFluxConv.wlFlxSp giNormalizeSky.normSky giNormalizeSky.normSkyErr giRebinSpectra.psflwidth giRebinSpectra.wlFfSp giRebinSpectra.wlFfSpErr giSplitSky.wlFfSkyComp giSplitSky.wlFfSkyCompErr giVignCorr.wlFfVcSp giVignCorr.wlFfVcSpErr
Aiprefs	
Aoprefs	
Reftables	
Nref	26

Alias	BINWNY
Dicname	DRS
Paramname	PRO WLEN REBIN NY
Type	integer
Format	
Unit	none
Function	giRebinSpectra
Author	royer
Comment	
Paramdesc	number of spectra on the rebinned S2dFrame
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	giFluxConv.skySubSp giModelSky.normSky giModelSky.normSkyErr giModelSky.psflwidth giNormalizeSky.wlFfSp giNormalizeSky.wlFfSpErr giSplitSky.psflwidth giSplitSky.wlFfSp giSplitSky.wlFfSpErr giSubtractSky.skyModel giSubtractSky.skyModelErr giSubtractSky.wlFfSp giSubtractSky.wlFfSpErr giVignCorr.wlFfSp giVignCorr.wlFfSpErr
O-Reffunctions	giFluxConv.wlFlxSp giNormalizeSky.normSky giNormalizeSky.normSkyErr giRebinSpectra.psflwidth giRebinSpectra.wlFfSp giRebinSpectra.wlFfSpErr giSplitSky.wlFfSkyComp giSplitSky.wlFfSkyCompErr giVignCorr.wlFfVcSp giVignCorr.wlFfVcSpErr
Aiprefs	
Aoprefs	
Reftables	
Nref	25

Alias	BINWSTEP
Dicname	DRS
Paramname	PRO WLEN REBIN WSTEP
Type	double
Format	
Unit	nm
Function	giRebinSpectra
Author	blecha
Comment	set to 0 for loarithmic rebinning
Paramdesc	wavelength step of the linear rebinned spectra
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	giFluxConv.skySubSp giModelSky.normSky giModelSky.normSkyErr giModelSky.psflwidth giNormalizeSky.wlFfSp giNormalizeSky.wlFfSpErr giSplitSky.psflwidth giSplitSky.wlFfSp giSplitSky.wlFfSpErr giSubtractSky.skyModel giSubtractSky.skyModelErr giSubtract-Sky.wlFfSp giSubtractSky.wlFfSpErr giVignCorr.wlFfSp giVignCorr.wlFfSpErr
O-Reffunctions	giNormalizeSky.normSky giNormalizeSky.normSkyErr giRebinSpectra.psflwidth giRebinSpectra.wlFfSp giRebinSpectra.wlFfSpErr giSplitSky.wlFfSkyComp giSplit-Sky.wlFfSkyCompErr giVignCorr.wlFfVcSp giVignCorr.wlFfVcSpErr
Aiprefs	
Aoprefs	
Reftables	
Nref	24

Alias	BINWSTEPLOG
Dicname	DRS
Paramname	PRO WLEN REBIN WSTEPLOG
Type	double
Format	
Unit	none
Function	giRebinSpectra
Author	blecha
Comment	set to 0 for linear rebinning
Paramdesc	wavelength step of the logarithmic rebinned spectra; the spectral element N has a central wavelength=w0+10**((N-1)*step)
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giRebinSpectra.psflwidth giRebinSpectra.wlFfSp giRebinSpectra.wlFfSpErr
Aiprefs	
Aoprefs	
Reftables	
Nref	3

Alias	CAMGFACT
Dicname	DRS
Paramname	DRS CAM GFACT
Type	float
Format	
Unit	none
Function	giGetWaveSolution
Author	blecha
Comment	Name changed from GCCDSCALE to CAMGFAC
Paramdesc	magnification factor of the camera
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	giGetWaveSolution.exSp giRebinSpectra.ffSp
O-Reffunctions	
Aiprefs	giRebinSpectra.waveSolution
Aoprefs	
Reftables	girWavCoef
Nref	3

Alias	CCDID
Dicname	CCDDCS
Paramname	DET ID
Type	string
Format	%30s
Unit	none
Function	
Author	blecha
Comment	DIC: Detector system Id
Paramdesc	DIC: Detector system identification Format: <CCD Id> - <ACE Id>
Context	Detector
Dicstatus	Released
Dicrev	3.6
I-Reffunctions	giAdjustSL.repBadPixMap giAdjustSL.repPixImage giArithmetics.badPixMap1 giArithmetics.badPixMap2 giArithmetics.inFrame1 giArithmetics.inFrame2 giDetectCosmicM.curBadPixMap giDetectCosmicS.estBadPixMap giEstimatePix.badPixMast giEstimatePix.dbSubImage giExtractSpectra.curBadPixMap giGetRemoveBias.badPixMast giGetRemoveBias.biasMast giGetRemoveBias.rawFrame giLocalSpectra.badPixMap giLocalSpectra.dbSubImage giReplaceFlagPix.crhBadPixMap giReplaceFlagPix.dbSubImage giReplaceFlagPix.estBadPixMap giSubtractDark.bSubImage giSubtractDark.badPixMast giSubtractDark.darkMast
O-Reffunctions	giArithmetics.badPixMap giDetectCosmicM.crhBadPixMap giDetectCosmicS.crhBadPixMap giEstimatePix.estBadPixMap giExtractSpectra.clbadPixMap giReplaceFlagPix.repBadPixMap
Aiprefs	giGetRemoveBias.biasLimitsMast
Aoprefs	
Reftables	girBiasBox
Nref	29

Alias	CCDPIXTIME
Dicname	CCDDCS *DRS
Paramname	DET READ PIXTIME
Type	double
Format	%.1f
Unit	us
Function	
Author	blecha
Comment	DIC: Pixel digitalization time [us] AL: used to estimate RON
Paramdesc	DIC: Time needed to digitize one pixel.
Context	Detector
Dicstatus	Released
Dicrev	3.6
I-Reffunctions	
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	
Nref	0

Alias	CCLIPMFRAC
Dicname	DRS
Paramname	PRO CRH SIGMA MFRAC
Type	float
Format	
Unit	none
Function	giDetectCosmicS:giDetectCosmicM
Author	simond
Comment	
Paramdesc	min fraction of points accepted/total (sigma-clipping)
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giDetectCosmicM.averageImage giDetectCosmicM.crhBadPixMap giDetectCosmicM.crhmCount giDetectCosmicM.sigmaImage
Aiprefs	
Aoprefs	
Reftables	
Nref	4

Alias	CCLIPNITER
Dicname	DRS
Paramname	PRO CRH SIGMA NITER
Type	integer
Format	
Unit	none
Function	giDetectCosmicS:giDetectCosmicM
Author	simond
Comment	
Paramdesc	number of iterations (sigma-clipping)
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giDetectCosmicM.averageImage giDetectCosmicM.crhBadPixMap giDetectCosmicM.crhmCount giDetectCosmicM.sigmaImage
Aiprefs	
Aoprefs	
Reftables	
Nref	4

Alias	CCLIPSIGMA
Dicname	DRS
Paramname	PRO CRH SIGMA MULT
Type	float
Format	
Unit	none
Function	giDetectCosmicS:giDetectCosmicM
Author	simond
Comment	
Paramdesc	multiple of sigma (sigma-clipping)
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giDetectCosmicM.averageImage giDetectCosmicM.crhBadPixMap giDetectCosmicM.crhmCount giDetectCosmicM.sigmaImage
Aiprefs	
Aoprefs	
Reftables	
Nref	4

Alias	CENDEC
Dicname	FPMISS
Paramname	undef
Type	real
Format	
Unit	radians
Function	
Author	
Comment	input value of P2PP
Paramdesc	field centre declination
Context	
Dicstatus	Missing (TBD by FLAMES consortium)
Dicrev	
I-Reffunctions	
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	girOzpoz
Nref	1

Alias	CENEQNX
Dicname	FPMISS
Paramname	undef
Type	real
Format	
Unit	none
Function	
Author	
Comment	input value of P2PP
Paramdesc	equinox of the above (FK5 Julian)
Context	
Dicstatus	Missing (TBD by FLAMES consortium)
Dicrev	
I-Reffunctions	
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	girOzpoz
Nref	1

Alias	CENRA
Dicname	FPMISS
Paramname	undef
Type	real
Format	
Unit	radians
Function	
Author	
Comment	input value of P2PP
Paramdesc	field centre right ascension
Context	
Dicstatus	Missing (TBD by FLAMES consortium)
Dicrev	
I-Reffunctions	
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	girOzpoz
Nref	1

Alias	CONAD
Dicname	CCDDCS
Paramname	DET OUTi CONAD
Type	double
Format	%.2f
Unit	e-/ADU
Function	
Author	simond
Comment	DIC: Conversion from ADUs to electrons
Paramdesc	DIC: The conversion factor which translates ADU to photonic electrons.
Context	Detector
Dicstatus	Released
Dicrev	3.6
I-Reffunctions	giGetRemoveBias.rawFrame
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	
Nref	1

Alias	CONFMJD
Dicname	FPMISS
Paramname	undef
Type	real
Format	
Unit	none
Function	
Author	
Comment	
Paramdesc	MJD for which the field was configured
Context	
Dicstatus	Missing (TBD by FLAMES consortium)
Dicrev	
I-Reffunctions	
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	girOzpoz
Nref	1

Alias	CSIGMAMULT
Dicname	DRS
Paramname	PRO CRH SIGMA MULTX
Type	float
Format	
Unit	none
Function	giDetectCosmicS:giDetectCosmicM
Author	blecha
Comment	The KW name TBFinalised after the clarification of use in giDetectCosmicM (AB8/9/2000)
Paramdesc	multiple of sigma
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giDetectCosmicM.averageImage giDetectCosmicM.crhBadPixMap giDetectCosmicM.crhmCount giDetectCosmicM.sigmaImage giDetectCosmicS.crhBadPixMap giDetectCosmicS.crhsCount
Aiprefs	
Aoprefs	
Reftables	
Nref	6

Alias	DARKEXPECT
Dicname	DRS
Paramname	PRO DARK EXPECT
Type	float
Format	
Unit	e-
Function	giSubtractDark
Author	blecha
Comment	
Paramdesc	The value of the expected dark current as computed by subtractDark from masterDark and exposure time
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giSubtractDark.dbSubImage
Aiprefs	
Aoprefs	
Reftables	
Nref	1

Alias	DARKMETHOD
Dicname	DRS
Paramname	PRO DARK METHOD
Type	string
Format	
Unit	none
Function	giSubtractDark
Author	blecha
Comment	
Paramdesc	One of 'UNIFORM' (we subtract a single number) and 'MASTER' (we subtract a scaled masterDark).
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giSubtractDark.dbSubImage
Aiprefs	
Aoprefs	
Reftables	
Nref	1

Alias	DARKTHRESH
Dicname	DRS
Paramname	PRO DARK THRESH
Type	float
Format	
Unit	e-
Function	giSubtractDark
Author	blecha
Comment	
Paramdesc	If the average expected dark is below this value, no dark subtraction is made.
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giSubtractDark.dbSubImage
Aiprefs	
Aoprefs	
Reftables	
Nref	1

Alias	DARKVALUE
Dicname	DRS
Paramname	PRO DARK VALUE
Type	float
Format	
Unit	e-
Function	giSubtractDark
Author	simond
Comment	
Paramdesc	The value of the actualy subtracted dark current. It is set to 0 if no subtraction is made (treshold not reached).
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	giExtractSpectra.dbSubImage
O-Reffunctions	giSubtractDark.dbSubImage
Aiprefs	
Aoprefs	
Reftables	
Nref	2

Alias	DATAMAX
Dicname	PR_FITS
Paramname	DATAMAX
Type	double
Format	%.7f
Unit	none *e-
Function	giSubtractDark
Author	blecha
Comment	DIC: Maximal pixel value AL: TB divided by EXPTIME if we want the flux
Paramdesc	DIC: Maximum pixel value across the frame (excludes special values, i.e BLANK). AL: Maximum signal of valid pixels
Context	FITS
Dicstatus	Released
Dicrev	1.0
I-Reffunctions	giSubtractDark.darkMast
O-Reffunctions	giArithmetics.outFrame
Aiprefs	
Aoprefs	
Reftables	
Nref	2

Alias	DATAMEAN
Dicname	PRO
Paramname	PRO DATAMEAN
Type	double
Format	%e
Unit	none *e-
Function	giSubtractDark
Author	blecha
Comment	DIC: Mean of pixel values AL: TB divided by EXPTIME if we want the flux
Paramdesc	DIC: Mean of the pixel values in the frame. All pixels are taken into account. AL: Average signal of valid pixels
Context	process
Dicstatus	submitted
Dicrev	1.5
I-Reffunctions	giGetRemoveBias.biasMast
O-Reffunctions	giArithmetics.outFrame
Aiprefs	
Aoprefs	
Reftables	
Nref	2

Alias	DATAMEDI
Dicname	PRO
Paramname	PRO DATAMEDI
Type	double
Format	%e
Unit	none *e-
Function	giSubtractDark
Author	blecha
Comment	DIC: Median of pixel values AL: TB divided by EXPTIME if we want the flux
Paramdesc	DIC: Exact median of the pixel values in the frame. All pixels are taken into account. AL: Median signal of valid pixels
Context	process
Dicstatus	submitted
Dicrev	1.5
I-Reffunctions	
O-Reffunctions	giArithmetics.outFrame
Aiprefs	
Aoprefs	
Reftables	
Nref	1

Alias	DATAMIN
Dicname	PR_FITS
Paramname	DATAMIN
Type	double
Format	%.7f
Unit	none *e-
Function	giSubtractDark
Author	blecha
Comment	DIC: Minimal pixel value
Paramdesc	DIC: Minimum pixel value across the frame excludes special values, i.e. BLANK). AL: Minimu signal of valid pixels
Context	FITS
Dicstatus	Released
Dicrev	1.0
I-Reffunctions	
O-Reffunctions	giArithmetics.outFrame
Aiprefs	
Aoprefs	
Reftables	
Nref	1

Alias	DATAMOD
Dicname	DRS
Paramname	PRO DATAMOD
Type	float
Format	
Unit	e-
Function	giSubtractDark
Author	blecha
Comment	TB divided by EXPTIME if we want the flux
Paramdesc	Mode of the signal of valid pixels
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	giGetRemoveBias.biasMast giSubtractDark.darkMast
O-Reffunctions	giSubtractDark.dbSubImage
Aiprefs	
Aoprefs	
Reftables	
Nref	3

Alias	DATASIG
Dicname	PRO
Paramname	PRO DATASIG
Type	double
Format	%e
Unit	none *e-
Function	giSubtractDark
Author	blecha
Comment	DIC: Standard deviation of pixel values
Paramdesc	DIC: Population standard deviation (sigma) of the pixel values in the frame. All pixels are taken into account. AL: Sigma of the signal of valid pixels
Context	process
Dicstatus	submitted
Dicrev	1.5
I-Reffunctions	
O-Reffunctions	giArithmetics.outFrame
Aiprefs	
Aoprefs	
Reftables	
Nref	1

Alias	DATE
Dicname	CCDDCS
Paramname	DATE
Type	string
Format	%30s
Unit	none
Function	
Author	
Comment	DIC: UT date when this file was written
Paramdesc	DIC: UT date when this file was written (ISO 8601).
Context	FITS
Dicstatus	Released
Dicrev	3.6
I-Reffunctions	
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	giTableStandard
Nref	1

Alias	DCATG
Dicname	DPR
Paramname	DPR CATG
Type	string
Format	%15s
Unit	none
Function	giGetWaveSolution
Author	blecha
Comment	DIC: Observation category AL: Used also by DFS to select recipe
Paramdesc	DIC: Category according to section 4.7 of GEN-SPE-ESO-19400-0794. (science, calib, test, simulation, other)
Context	DPR
Dicstatus	Released
Dicrev	1.3
I-Reffunctions	giGetWaveSolution.exSp giVignCorr.wlFfSp
O-Reffunctions	
Aiprefs	giVignCorr.vignCorrMast
Aoprefs	
Reftables	girVignCorr
Nref	3

Alias	DTYPE
Dicname	DPR
Paramname	DPR TYPE
Type	string
Format	%15s
Unit	none
Function	giGetWaveSolution
Author	blecha
Comment	DIC: Observation type AL: Used also by DFS to select recipe
Paramdesc	DIC: Type of observation according to section 4.7 of GEN-SPE-ESO-19400-0794.
Context	DPR
Dicstatus	Released
Dicrev	1.3
I-Reffunctions	giGetWaveSolution.exSp
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	
Nref	1

Alias	ECLIPMFRAC
Dicname	DRS
Paramname	PRO EXT SIGMA MFRAC
Type	float
Format	
Unit	none
Function	giExtractSpectra
Author	simond
Comment	
Paramdesc	min fraction of points accepted/total (sigma-clipping)
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giExtractSpectra.clbadPixMap giExtractSpectra.cleanImage giExtractSpectra.exBack giExtractSpectra.exBadPixMap giExtractSpectra.exSp giExtractSpectra.exSpErr giExtractSpectra.exSpNpixels giExtractSpectra.slImage
Aiprefs	
Aoprefs	
Reftables	
Nref	8

Alias	ECLIPNITER
Dicname	DRS
Paramname	PRO EXT SIGMA NITER
Type	integer
Format	
Unit	none
Function	giExtractSpectra
Author	simond
Comment	
Paramdesc	number of iterations (sigma-clipping)
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giExtractSpectra.clbadPixMap giExtractSpectra.cleanImage giExtractSpectra.exBack giExtractSpectra.exBadPixMap giExtractSpectra.exSp giExtractSpectra.exSpErr giExtractSpectra.exSpNpixels giExtractSpectra.slImage
Aiprefs	
Aoprefs	
Reftables	
Nref	8

Alias	ECLIPSIGMA
Dicname	DRS
Paramname	PRO EXT SIGMA MULT
Type	float
Format	
Unit	none
Function	giExtractSpectra
Author	simond
Comment	
Paramdesc	multiple of sigma (sigma-clipping)
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giExtractSpectra.clbadPixMap giExtractSpectra.cleanImage giExtractSpectra.exBack giExtractSpectra.exBadPixMap giExtractSpectra.exSp giExtractSpectra.exSpErr giExtractSpectra.exSpNpixels giExtractSpectra.slImage
Aiprefs	
Aoprefs	
Reftables	
Nref	8

Alias	EXPTIME
Dicname	UVES_ICS
Paramname	INS DETi UIT
Type	double
Format	%.3f
Unit	sec
Function	
Author	blecha
Comment	DIC: User defined Integration time AL: TBD what KW(S) appropriate
Paramdesc	Total exposure time for a single exposure or for the individual exposure within a multiple frame
Context	Instrument
Dicstatus	Approved
Dicrev	1.39
I-Reffunctions	giAdjustSL.phffImageMast giAdjustSL.repPixImage giDetectCosmicM.dbSubImage giDetectCosmicS.dbSubImage giExtractSpectra.dbSubImage giFluxConv.exNffMast giFluxConv.skySubSp giGetRemoveBias.biasMast giGetRemoveBias.rawFrame giSplitSky.wlFfSp giSubtractDark.bSubImage giSubtractDark.darkMast
O-Reffunctions	giDetectCosmicM.averageImage
Aiprefs	
Aoprefs	
Reftables	
Nref	13

Alias	EXTBKGCOEF
Dicname	DRS
Paramname	PRO EXT BACKGD COEF
Type	integer
Format	
Unit	none
Function	giExtractSpectra
Author	simond
Comment	
Paramdesc	K polynomial coefficients for NX independant background models
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giExtractSpectra.exBack
Aiprefs	
Aoprefs	
Reftables	
Nref	1

Alias	EXTBKGYDEG
Dicname	DRS
Paramname	PRO EXT BACKGD YDEG
Type	integer
Format	
Unit	none
Function	giExtractSpectra
Author	simond
Comment	
Paramdesc	polyomial degree of the background model (y direction)
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giExtractSpectra.clbadPixMap giExtractSpectra.cleanImage giExtractSpectra.exBack giExtractSpectra.exBadPixMap giExtractSpectra.exSp giExtractSpectra.exSpErr giExtractSpectra.exSpNpixels giExtractSpectra.slImage
Aiprefs	
Aoprefs	
Reftables	
Nref	8

Alias	EXTMETHOD
Dicname	DRS
Paramname	PRO EXT METHOD
Type	integer
Format	
Unit	none
Function	giExtractSpectra
Author	simond
Comment	
Paramdesc	extraction method: 1=simple summation; 2=Horne's method; 3=LSQ fit
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giDeblendSpectra.exDebSp giDeblendSpectra.exDebSpErr giExtractSpectra.clbadPixMap giExtractSpectra.cleanImage giExtractSpectra.exBack giExtractSpectra.exBadPixMap giExtractSpectra.exSp giExtractSpectra.exSpErr giExtractSpectra.exSpNpixels giExtractSpectra.slImage
Aiprefs	
Aoprefs	
Reftables	
Nref	10

Alias	EXTNS
Dicname	DRS
Paramname	PRO EXT NY
Type	integer
Format	
Unit	none
Function	giExtractSpectra
Author	simond
Comment	
Paramdesc	Number of extracted spectra on the rebinned S2dFrame
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	giAdjustSL.locWyMast giAdjustSL.locYMast giDeblendSpectra.exSp giDeblendSpectra.exSpErr giDeblendSpectra.locWy giDeblendSpectra.locY giEstimatePix.locWyMast giEstimatePix.locYMast giExtractSpectra.locWy giExtractSpectra.locY giFiberTrans.exFullSlit giFiberTrans.exNffMast giFlatSpectra.exNffErrMast giFlatSpectra.exNffMast giFlatSpectra.exSp giFlatSpectra.exSpErr giFluxConv.exNffMast giGetWaveSolution.exSp giGetWaveSolution.exSpErr giGetWaveSolution.locY giLocalSpectra.locWyMast giLocalSpectra.locYMast giRebinSpectra.ffSp giRebinSpectra.ffSpErr
O-Reffunctions	giDeblendSpectra.exDebSp giDeblendSpectra.exDebSpErr giExtractSpectra.exBadPixMap giExtractSpectra.exSp giExtractSpectra.exSpErr giExtractSpectra.exSpNpixels giFlatSpectra.ffSp giFlatSpectra.ffSpErr giLocalSpectra.locWy giLocalSpectra.locY
Aiprefs	
Aoprefs	
Reftables	
Nref	34

Alias	EXTNX
Dicname	DRS
Paramname	PRO EXT NX
Type	integer
Format	
Unit	none
Function	giExtractSpectra
Author	simond
Comment	
Paramdesc	Number of pixel per spectrum
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	giAdjustSL.locWyMast giAdjustSL.locYMast giDeblendSpectra.exSp giDeblendSpectra.exSpErr giDeblendSpectra.locWy giDeblendSpectra.locY giEstimatePix.locWyMast giEstimatePix.locYMast giExtractSpectra.locWy giExtractSpectra.locY giFiberTrans.exFullSlit giFiberTrans.exNffMast giFlatSpectra.exNffErrMast giFlatSpectra.exNffMast giFlatSpectra.exSp giFlatSpectra.exSpErr giFluxConv.exNffMast giGetWaveSolution.exSp giGetWaveSolution.exSpErr giGetWaveSolution.locY giLocalSpectra.locWyMast giLocalSpectra.locYMast giRebinSpectra.ffSp giRebinSpectra.ffSpErr
O-Reffunctions	giDeblendSpectra.exDebSp giDeblendSpectra.exDebSpErr giExtractSpectra.exBack giExtractSpectra.exBadPixMap giExtractSpectra.exSp giExtractSpectra.exSpErr giExtractSpectra.exSpNpixels giFlatSpectra.ffSp giFlatSpectra.ffSpErr giLocalSpectra.locWy giLocalSpectra.locY
Aiprefs	
Aoprefs	
Reftables	
Nref	35

Alias	EXTPSFAMPL
Dicname	DRS
Paramname	PRO EXT PSF AMPL
Type	float
Format	
Unit	none
Function	giExtractSpectra
Author	simond
Comment	
Paramdesc	PSF amplitude tolerance for CRH det. in method 2
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giExtractSpectra.clbadPixMap giExtractSpectra.cleanImage giExtractSpectra.exBack giExtractSpectra.exBadPixMap giExtractSpectra.exSp giExtractSpectra.exSpErr giExtractSpectra.exSpNpixels giExtractSpectra.slImage
Aiprefs	
Aoprefs	
Reftables	
Nref	8

Alias	EXTPSFBKGD
Dicname	DRS
Paramname	PRO EXT PSF BKGD
Type	float
Format	
Unit	none
Function	giExtractSpectra
Author	simond
Comment	
Paramdesc	PSF backgrnd tolerance for CRH det. in method 2
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giExtractSpectra.clbadPixMap giExtractSpectra.cleanImage giExtractSpectra.exBack giExtractSpectra.exBadPixMap giExtractSpectra.exSp giExtractSpectra.exSpErr giExtractSpectra.exSpNpixels giExtractSpectra.slImage
Aiprefs	
Aoprefs	
Reftables	
Nref	8

Alias	EXTPSFPOS
Dicname	DRS
Paramname	PRO EXT PSF POS
Type	float
Format	
Unit	none
Function	giExtractSpectra
Author	simond
Comment	
Paramdesc	PSF position tolerance for CRH det. in method 2
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giExtractSpectra.clbadPixMap giExtractSpectra.cleanImage giExtractSpectra.exBack giExtractSpectra.exBadPixMap giExtractSpectra.exSp giExtractSpectra.exSpErr giExtractSpectra.exSpNpixels giExtractSpectra.slImage
Aiprefs	
Aoprefs	
Reftables	
Nref	8

Alias	EXTPSFWID1
Dicname	DRS
Paramname	PRO EXT PSF WID1
Type	float
Format	
Unit	none
Function	giExtractSpectra
Author	simond
Comment	
Paramdesc	PSF width1 tolerance for CRH det. in method 2
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giExtractSpectra.clbadPixMap giExtractSpectra.cleanImage giExtractSpectra.exBack giExtractSpectra.exBadPixMap giExtractSpectra.exSp giExtractSpectra.exSpErr giExtractSpectra.exSpNpixels giExtractSpectra.slImage
Aiprefs	
Aoprefs	
Reftables	
Nref	8

Alias	EXTPSFWID2
Dicname	DRS
Paramname	PRO EXT PSF WID2
Type	float
Format	
Unit	none
Function	giExtractSpectra
Author	simond
Comment	
Paramdesc	PSF width2 tolerance for CRH det. in method 2
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giExtractSpectra.clbadPixMap giExtractSpectra.cleanImage giExtractSpectra.exBack giExtractSpectra.exBadPixMap giExtractSpectra.exSp giExtractSpectra.exSpErr giExtractSpectra.exSpNpixels giExtractSpectra.slImage
Aiprefs	
Aoprefs	
Reftables	
Nref	8

Alias	EXTPSFWID3
Dicname	DRS
Paramname	PRO EXT PSF WID3
Type	float
Format	
Unit	none
Function	giExtractSpectra
Author	simond
Comment	
Paramdesc	PSF width3 tolerance for CRH det. in method 2
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giExtractSpectra.clbadPixMap giExtractSpectra.cleanImage giExtractSpectra.exBack giExtractSpectra.exBadPixMap giExtractSpectra.exSp giExtractSpectra.exSpErr giExtractSpectra.exSpNpixels giExtractSpectra.slImage
Aiprefs	
Aoprefs	
Reftables	
Nref	8

Alias	EXTWMULT
Dicname	DRS
Paramname	PRO EXT PSFPN
Type	double
Format	
Unit	none
Function	giExtractSpectra
Author	simond
Comment	
Paramdesc	number of PSF width parameters defining the half-length of the extraction virtual slit
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	giDeblendSpectra.exSp giDeblendSpectra.exSpErr
O-Reffunctions	giDeblendSpectra.exDebSp giDeblendSpectra.exDebSpErr giExtractSpectra.clbadPixMap giExtractSpectra.cleanImage giExtractSpectra.exBack giExtractSpectra.exBadPixMap giExtractSpectra.exSp giExtractSpectra.exSpErr giExtractSpectra.exSpNpixels giExtractSpectra.slImage
Aiprefs	
Aoprefs	
Reftables	
Nref	12

Alias	FATHER
Dicname	DRS
Paramname	PRO DATA FATHER
Type	string
Format	
Unit	none
Function	
Author	blecha
Comment	
Paramdesc	ID of father frame; used to associate an auxilliary data to original frame
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	
Aiprefs	
Aoprefs	giModelSky.sNorm
Reftables	girIntSky girSkyCoef girSky girSkyMom
Nref	4

Alias	FFNORM
Dicname	DRS
Paramname	PRO FF NORM
Type	float
Format	
Unit	none
Function	giFlatSpectra
Author	north
Comment	
Paramdesc	Multiplicative factor used to normalize the FF extracted spectra. To go back to the original signal: FForig=normedFF*FFNORM
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giFlatSpectra.ffSp
Aiprefs	
Aoprefs	
Reftables	
Nref	1

Alias	FIBRESKOi
Dicname	MISSING
Paramname	INS FIBRESi KO
Type	integer
Format	
Unit	none
Function	giLocalSpectra
Author	blecha
Comment	If not provided by SOS (KW DIC FGIR_ICS) could come from slitGeometry table and ozpoz (TBD) (in the last case the KW belongs to DRS).
Paramdesc	Running number of the invalid spectra/fibres
Context	
Dicstatus	Missing (TBD by FLAMES consortium)
Dicrev	
I-Reffunctions	
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	
Nref	0

Alias	FILENAME
Dicname	FPMISS
Paramname	undef
Type	string
Format	
Unit	none
Function	
Author	
Comment	
Paramdesc	name of the configuration file used
Context	
Dicstatus	Missing (TBD by FLAMES consortium)
Dicrev	
I-Reffunctions	
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	girOzpoz
Nref	1

Alias	FLCONVSTAND
Dicname	DRS
Paramname	PRO DATA FLCVS
Type	logical
Format	
Unit	none
Function	giFluxConv
Author	blecha
Comment	Better name TBF
Paramdesc	Frflag for flux conversion
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giFluxConv.wlFlxSp
Aiprefs	
Aoprefs	
Reftables	girFlat girFluxcal
Nref	3

Alias	FOCLEN
Dicname	DRS
Paramname	DRS COL FOCAL
Type	double
Format	
Unit	m
Function	giWaveSolution
Author	blecha
Comment	
Paramdesc	focal length of the collimator
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	giGetWaveSolution.exSp giRebinSpectra.ffSp
O-Reffunctions	
Aiprefs	giRebinSpectra.waveSolution
Aoprefs	
Reftables	girPsfXwid girWavCoef
Nref	4

Alias	FRAME
Dicname	DRS
Paramname	PRO DATA SKYFRAME
Type	string
Format	
Unit	none
Function	giModelSky
Author	blecha
Comment	Better name TBF
Paramdesc	Frame name with sky spectra associated to the table
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	
Nref	0

Alias	GIRFTYPE
Dicname	DRS
Paramname	PRO FRAME TYPE
Type	string
Format	
Unit	none
Function	all
Author	simond
Comment	
Paramdesc	type of frame data
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	giAdjustSL.locWyMast giAdjustSL.locYMast giAdjustSL.phffImageMast giAdjustSL.repPixImage giArithmetics.inFrame1 giArithmetics.inFrame2 giDeblendSpectra.exSp giDeblendSpectra.exSpErr giDeblendSpectra.locWy giDeblendSpectra.locY giDetectCosmicM.dbSubImage giDetectCosmicS.dbSubImage giDetectCosmicS.estimateImage giDetectCosmicS.estimateSigmaImage giEstimatePix.dbSubImage giEstimatePix.locWyMast giEstimatePix.locYMast giExtractSpectra.dbSubImage giExtractSpectra.locWy giExtractSpectra.locY giExtractSpectra.slImage giFiberTrans.exFullSlit giFiberTrans.exNffMast giFlatSpectra.exNffErrMast giFlatSpectra.exNffMast giFlatSpectra.exSp giFlatSpectra.exSpErr giFluxConv.exNffMast giFluxConv.skySubSp giGetRemoveBias.biasMast giGetRemoveBias.rawFrame giGetWaveSolution.exSp giGetWaveSolution.exSpErr giGetWaveSolution.locY giLocalSpectra.dbSubImage giLocalSpectra.locWyMast giLocalSpectra.locYMast giModelSky.normSky giModelSky.normSkyErr giModelSky.psflwidth giNormalizeFSFF.ffFullSlit giNormalizeSky.wlFfSp giNormalizeSky.wlFfSpErr giRebinSpectra.ffSp giRebinSpectra.ffSpErr giRebinSpectra.fitWxl giReplaceFlagPix.dbSubImage giReplaceFlagPix.estimateImage giSplitSky.psflwidth giSplitSky.wlFfSpErr giSubtractDark.bSubImage giSubtractDark.darkMast giSubtractSky.skyModel giSubtractSky.skyModelErr giSubtractSky.wlFfSp giSubtractSky.wlFfSpErr giVignCorr.wlFfSp giVignCorr.wlFfSpErr
O-Reffunctions	giAdjustSL.slImage giArithmetics.outFrame giDeblendSpectra.exDebSp giDeblendSpectra.exDebSpErr giDetectCosmicM.averageImage giDetectCosmicM.crhBadPixMap giDetectCosmicM.crhmCount giDetectCosmicM.sigmaImage giDetectCosmicS.crhBadPixMap giDetectCosmicS.crhsCount giEstimatePix.estBadPixMap giEstimatePix.estimateImage giEstimatePix.estimateNpixels giEstimatePix.estimateSigmaImage giExtractSpectra.clbadPixMap giExtractSpectra.cleanImage giExtractSpectra.exBack giExtractSpectra.exBadPixMap giExtractSpectra.exSp giExtractSpectra.exSpErr giExtractSpectra.exSpNpixels giExtractSpectra.slImage giFlatSpectra.ffSp giFlatSpectra.ffSpErr giFluxConv.wlFlxSp giGetRemoveBias.bSubImage giGetWaveSolution.fitWxl giLocalSpectra.locWy giLocalSpectra.locY giNormalizeFSFF.ffHighFreq giNormalizeSky.normSky giNormalizeSky.normSkyErr giRebinSpectra.psflwidth giRebinSpectra.wlFfSp giRebinSpectra.wlFfSpErr giReplaceFlagPix.repBadPixMap giReplaceFlagPix.repPixImage giSplitSky.wlFfSkyCompErr giSubtractDark.dbSubImage giVignCorr.wlFfVcSp giVignCorr.wlFfVcSpErr
Aiprefs	
Aoprefs	
Reftables	
Nref	99

Alias	GIRTTYPE
Dicname	DRS
Paramname	PRO TABLE TYPE
Type	string
Format	
Unit	none
Function	all
Author	simond
Comment	
Paramdesc	type of table data
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	
Aiprefs	giDeblendSpectra.psfParams giExtractSpectra.psfParams giFiberTrans.MeanNff giFiberTrans.VignSpectro giFluxConv.atmExtMast giFluxConv.refFluxMast giFluxConv.transVltMast giGetRemoveBias.biasLimitsMast giGetWaveSolution.linesSkyMast giGetWaveSolution.linesWlampMast giGetWaveSolution.psfXwidthMast giGetWaveSolution.slitGeometryMast giGetWaveSolution.waveSolutionMast giLocalSpectra.psfParamsMast giModelSky.intSky giNormalizeSky.numSky giRebinSpectra.waveSolution giSplitSky.linesSkyMast giSplitSky.numSky
Aoprefs	giGetWaveSolution.cleanlineTableMW giGetWaveSolution.cleanlineTableMWSky giGetWaveSolution.psfXwidth giGetWaveSolution.waveSolution giLocalSpectra.psfParams giModelSky.sNorm giNormalizeSky.intSky
Reftables	girAtmExt girBiasBox girCcdConst girFiberTrans girFlat girFluxcal girGrating girIntSky girLine girMeanSp girOzpoz girPsf girPsfXwid girSkyCoef girSky girSkyMom girSlitGeo girSpectroConst girTransVlt girVignCorr girVignSpectro girWavCoef refFlux
Nref	23

Alias	GRATING
Dicname	FGIR_ICS
Paramname	INS GRAT NAMEi
Type	string
Format	%30s
Unit	none
Function	giLocalSpectra
Author	simond
Comment	DIC: Grating common name (hls). AL: Caution: For FGIR_ICS KWs there is a dicrepancy in INS used by FGIR_ICS and INS1 required by VLT standard
Paramdesc	DIC: ESO name for grating unit.
Context	Instrument
Dicstatus	Development
Dicrev	*Revision* : 1.0
I-Reffunctions	giAdjustSL.locWyMast giAdjustSL.locYMast giAdjustSL.phffImageMast giAdjustSL.repPixImage giDeblendSpectra.exSp giDeblendSpectra.exSpErr giDeblendSpectra.locWy giDeblendSpectra.locY giDetectCosmicM.dbSubImage giEstimatePix.dbSubImage giEstimatePix.locWyMast giEstimatePix.locYMast giExtractSpectra.dbSubImage giExtractSpectra.locWy giExtractSpectra.locY giExtractSpectra.slImage giFiberTrans.exFullSlit giFiberTrans.exNffMast giFlatSpectra.exNffErrMast giFlatSpectra.exNffMast giFlatSpectra.exSp giFlatSpectra.exSpErr giFluxConv.exNffMast giFluxConv.skySubSp giGetWaveSolution.exSp giGetWaveSolution.locY giLocalSpectra.dbSubImage giLocalSpectra.locWyMast giLocalSpectra.locYMast giModelSky.normSky giNormalizeFSFF.ffFullSlit giNormalizeSky.wlFfSp giNormalizeSky.wlFfSpErr giRebinSpectra.ffSp giSplitSky.wlFfSp giVignCorr.wlFfSp
O-Reffunctions	giSplitSky.wlFfSkyComp giSplitSky.wlFfSkyCompErr
Aiprefs	giDeblendSpectra.psfParams giExtractSpectra.psfParams giFiberTrans.MeanNff giFiberTrans.VignSpectro giGetWaveSolution.linesSkyMast giGetWaveSolution.psfXwidthMast giGetWaveSolution.waveSolutionMast giLocalSpectra.psfParamsMast giModelSky.intSky giNormalizeSky.numSky giRebinSpectra.waveSolution giSplitSky.linesSkyMast giSplitSky.numSky
Aoprefs	giModelSky.sNorm giNormalizeSky.intSky
Reftables	girFiberTrans girFlat girFluxcal girGrating girIntSky girLine girMeanSp girPsf girPsfXwid girSkyCoef girSky girSkyMom girTransVlt girVignCorr girVignSpectro girWavCoef
Nref	54

Alias	GRATORD
Dicname	DRS
Paramname	DRS GRAT ORDER
Type	integer
Format	
Unit	none
Function	giGetWaveSolution
Author	blecha
Comment	
Paramdesc	Order of the grating in the current setup
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	giGetWaveSolution.exSp
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	
Nref	1

Alias	GRATRES
Dicname	FGIR_ICS
Paramname	INS GRAT RESOL
Type	double
Format	%.1f
Unit	none
Function	giGetWaveSolution
Author	blecha
Comment	DIC: Resolution in encoder steps (h). AL: Verify that this is relevant to current setup (angle, order)
Paramdesc	DIC: ENC = OFFST+ RESOL* RAD2DEG* arcsin(WLEN* GRV/(2* cos(ROT)))
Context	Instrument
Dicstatus	Development
Dicrev	*Revision* : 1.0
I-Reffunctions	giGetWaveSolution.exSp giSubtractSky.wlFfSp
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	
Nref	2

Alias	GRATROT
Dicname	FGIR_ICS
Paramname	INS GRAT ROT
Type	double *float
Format	%.3f
Unit	deg
Function	giGetWaveSolution
Author	simond
Comment	DIC: Grating rot angle [deg] (hl).
Paramdesc	DIC: Rotation angle of grating. The reference frame is given in the corresponding instrument specification. ENC = OFFST+ RESOL* RAD2DEG* arcsin(WLEN* GRV/(2* cos(ROT))) AL: Rotation angle of grating
Context	Instrument
Dicstatus	Development
Dicrev	*Revision* : 1.0
I-Reffunctions	giGetWaveSolution.exSp giRebinSpectra.ffSp
O-Reffunctions	
Aiprefs	giRebinSpectra.waveSolution
Aoprefs	
Reftables	girPsfXwid girWavCoef
Nref	4

Alias	GSPACE
Dicname	FGIR_ICS
Paramname	INS GRAT GRVi
Type	double
Format	%.7f
Unit	none
Function	giGetWaveSolution
Author	blecha
Comment	DIC: Grating grooves/nm (hl).
Paramdesc	DIC: Number of grooves per nanometer ENC = OFFST+ RESOL* RAD2DEG* arcsin(WLEN* GRV/(2* cos(ROT)))
Context	Instrument
Dicstatus	Development
Dicrev	*Revision* : 1.0
I-Reffunctions	giGetWaveSolution.exSp
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	girWavCoef
Nref	2

Alias	LABEL
Dicname	FPMISS
Paramname	undef
Type	string
Format	
Unit	none
Function	
Author	
Comment	
Paramdesc	name of the field
Context	
Dicstatus	Missing (TBD by FLAMES consortium)
Dicrev	
I-Reffunctions	
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	girOzpoz
Nref	1

Alias	LAMP
Dicname	DRS
Paramname	PRO DATA LSOURCE
Type	string
Format	
Unit	none
Function	
Author	blecha
Comment	One of ThAr,Ne,ThArNe or 'SKY' (telluric lines)
Paramdesc	Name of the W Calibration source
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	giEstimatePix.dbSubImage giFluxConv.exNffMast giFluxConv.skySubSp giGetWaveSolution.exSp giVignCorr.wlFfSp
O-Reffunctions	
Aiprefs	giGetWaveSolution.linesWlampMast giVignCorr.vignCorrMast
Aoprefs	
Reftables	girFlat girLine girVignCorr
Nref	8

Alias	LAMPDENS
Dicname	MISSING
Paramname	INS DENS VALUE
Type	float
Format	
Unit	percent
Function	
Author	blecha
Comment	It should be added to FGIR_ICS
Paramdesc	Setup-dependent factor which multiply the lamp flux to take into account the density filtre of the lamp.
Context	
Dicstatus	Missing (TBD by FLAMES consortium)
Dicrev	
I-Reffunctions	
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	
Nref	0

Alias	LAMPID
Dicname	FP_LP
Paramname	INS LAMPi ID*
Type	undef
Format	
Unit	none
Function	
Author	blecha
Comment	
Paramdesc	
Context	
Dicstatus	
Dicrev	
I-Reffunctions	giFluxConv.exNffMast giFluxConv.skySubSp giGetWaveSolution.exSp giVignCorr.wlFfSp
O-Reffunctions	
Aiprefs	giGetWaveSolution.linesWlampMast giVignCorr.vignCorrMast
Aoprefs	
Reftables	girFlat girLine girVignCorr
Nref	7

Alias	LCLIPMFRAC
Dicname	DRS
Paramname	PRO LOC SIGMA MFRAC
Type	float
Format	
Unit	none
Function	giLocalSpectra
Author	royer
Comment	
Paramdesc	min fraction of points accepted/total (sigma-clipping)
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giLocalSpectra.locWy giLocalSpectra.locY
Aiprefs	
Aoprefs	giLocalSpectra.psfParams
Reftables	girPsf
Nref	3

Alias	LCLIPNITER
Dicname	DRS
Paramname	PRO LOC SIGMA NITER
Type	integer
Format	
Unit	none
Function	giLocalSpectra
Author	royer
Comment	
Paramdesc	number of iterations (sigma-clipping)
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giLocalSpectra.locWy giLocalSpectra.locY
Aiprefs	
Aoprefs	giLocalSpectra.psfParams
Reftables	girPsf
Nref	3

Alias	LCLIPSIGMA
Dicname	DRS
Paramname	PRO LOC SIGMA MULT
Type	float
Format	
Unit	none
Function	giLocalSpectra
Author	royer
Comment	
Paramdesc	multiple of sigma (sigma-clipping)
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giLocalSpectra.locWy giLocalSpectra.locY
Aiprefs	
Aoprefs	giLocalSpectra.psfParams
Reftables	girPsf
Nref	3

Alias	LINLIMIT
Dicname	DRS
Paramname	DRS CCD LINLIM
Type	double
Format	
Unit	none
Function	giLocalSpectra
Author	royer
Comment	
Paramdesc	linearity limit of the CCD in e-/pixel
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	giLocalSpectra.dbSubImage
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	
Nref	1

Alias	LPOLYDEG
Dicname	DRS
Paramname	PRO LOC POLYDEG
Type	integer
Format	
Unit	none
Function	giLocalSpectra
Author	royer
Comment	
Paramdesc	degree of the polynomial fit
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giLocalSpectra.locWy giLocalSpectra.locY
Aiprefs	
Aoprefs	giLocalSpectra.psfParams
Reftables	girPsf
Nref	3

Alias	MAXRADIUS
Dicname	DRS
Paramname	PRO EST AREA MAX
Type	float
Format	
Unit	pixel
Function	giEstimatePix
Author	royer
Comment	
Paramdesc	maximum radius of the neighboring area
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giEstimatePix.estBadPixMap giEstimatePix.estimateImage giEstimatePix.estimateNpixels giEstimatePix.estimateSigmaImage
Aiprefs	
Aoprefs	
Reftables	
Nref	4

Alias	METRIC
Dicname	DRS
Paramname	PRO EST METRIC
Type	integer
Format	
Unit	none
Function	giEstimatePix
Author	royer
Comment	
Paramdesc	XY metrics indicating how the averaging area spreads 11-X,Y; 10-X only 01-Y only 00-LAMP KW driven
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giEstimatePix.estBadPixMap giEstimatePix.estimateImage giEstimatePix.estimateNpixels giEstimatePix.estimateSigmaImage
Aiprefs	
Aoprefs	
Reftables	
Nref	4

Alias	NOISEMULT
Dicname	DRS
Paramname	PRO LOC NOISE MULT
Type	float
Format	
Unit	none
Function	giLocalSpectra
Author	royer
Comment	
Paramdesc	noise detector multiple
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giLocalSpectra.locWy giLocalSpectra.locY
Aiprefs	
Aoprefs	giLocalSpectra.psfParams
Reftables	girPsf
Nref	3

Alias	NX
Dicname	DRS
Paramname	PRO DATA NX
Type	integer
Format	
Unit	none
Function	all
Author	blecha
Comment	
Paramdesc	Size of image in X direction (lambda)
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	giAdjustSL.phffImageMast giAdjustSL.repPixImage giArithmetics.inFrame1 giArithmetics.inFrame2 giDetectCosmicM.dbSubImage giDetectCosmicS.dbSubImage giDetectCosmicS.estimateImage giDetectCosmicS.estimateSigmaImage giEstimatePix.dbSubImage giExtractSpectra.dbSubImage giExtractSpectra.slImage giGetRemoveBias.biasMast giGetRemoveBias.rawFrame giGetWaveSolution.exSp giLocalSpectra.dbSubImage giNormalizeFSFF.ffFullSlit giReplaceFlagPix.dbSubImage giReplaceFlagPix.estimateImage giSubtractDark.bSubImage giSubtractDark.darkMast
O-Reffunctions	giArithmetics.outFrame giExtractSpectra.cleanImage giExtractSpectra.slImage giReplaceFlagPix.repPixImage
Aiprefs	
Aoprefs	
Reftables	
Nref	24

Alias	NY
Dicname	DRS
Paramname	PRO DATA NY
Type	integer
Format	
Unit	none
Function	all
Author	blecha
Comment	
Paramdesc	Size of image in Y direction (slit)
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	giAdjustSL.phffImageMast giAdjustSL.repPixImage giArithmetics.inFrame1 giArithmetics.inFrame2 giDetectCosmicM.dbSubImage giDetectCosmicS.dbSubImage giDetectCosmicS.estimateImage giDetectCosmicS.estimateSigmaImage giEstimatePix.dbSubImage giExtractSpectra.dbSubImage giExtractSpectra.slImage giGetRemoveBias.biasMast giGetRemoveBias.rawFrame giGetWaveSolution.exSp giLocalSpectra.dbSubImage giNormalizeFSFF.ffFullSlit giReplaceFlagPix.dbSubImage giReplaceFlagPix.estimateImage giSubtractDark.bSubImage giSubtractDark.darkMast
O-Reffunctions	giArithmetics.outFrame giExtractSpectra.cleanImage giExtractSpectra.slImage giReplaceFlagPix.repPixImage
Aiprefs	
Aoprefs	
Reftables	
Nref	24

Alias	OVSCX
Dicname	CCDDCS
Paramname	DET OUTi OVSCX
Type	integer
Format	%d
Unit	none
Function	giGetRemoveBias
Author	blecha
Comment	DIC: Overscan region in X
Paramdesc	DIC: Number of pixels in X read as overscan.
Context	Detector
Dicstatus	Released
Dicrev	3.6
I-Reffunctions	giGetRemoveBias.biasMast giGetRemoveBias.rawFrame
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	
Nref	2

Alias	OVSCY
Dicname	CCDDCS
Paramname	DET OUTi OVSCY
Type	integer
Format	%d
Unit	none
Function	giGetRemoveBias
Author	blecha
Comment	DIC: Overscan region in Y
Paramdesc	DIC: Number of pixels in Y read as overscan.
Context	Detector
Dicstatus	Released
Dicrev	3.6
I-Reffunctions	giGetRemoveBias.biasMast giGetRemoveBias.rawFrame
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	
Nref	2

Alias	PRESCX
Dicname	CCDDCS
Paramname	DET OUTi PRSCX
Type	integer
Format	%d
Unit	none
Function	
Author	blecha
Comment	DIC: Prescan region in X
Paramdesc	DIC: Number of pixels in X read before real data arrives.
Context	Detector
Dicstatus	Released
Dicrev	3.6
I-Reffunctions	giGetRemoveBias.biasMast giGetRemoveBias.rawFrame
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	
Nref	2

Alias	PRESCY
Dicname	CCDDCS
Paramname	DET OUTi PRSCY
Type	integer
Format	%d
Unit	none
Function	
Author	blecha
Comment	DIC: Prescan region in Y
Paramdesc	DIC: Number of pixels in Y read before real data arrives.
Context	Detector
Dicstatus	Released
Dicrev	3.6
I-Reffunctions	giGetRemoveBias.biasMast giGetRemoveBias.rawFrame
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	
Nref	2

Alias	PURPOSE
Dicname	DRS
Paramname	PRO DATA PURPOSE
Type	string
Format	
Unit	none
Function	
Author	blecha
Comment	Standard table KW
Paramdesc	Purpose of data structure
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	giTableStandard
Nref	1

Alias	REFSRC
Dicname	MISSING
Paramname	GEN OBJ REF
Type	string
Format	
Unit	none
Function	giFluxConv
Author	blecha
Comment	
Paramdesc	Type of reference source (LAMP, STAR, SKY, OTHER); same as LAMP KW if type is LAMP
Context	
Dicstatus	Missing (TBD by FLAMES consortium)
Dicrev	
I-Reffunctions	giFluxConv.exNffMast giFluxConv.skySubSp
O-Reffunctions	
Aiprefs	giFluxConv.refFluxMast
Aoprefs	
Reftables	girFlat girFluxcal refFlux
Nref	5

Alias	REFSRCID
Dicname	MISSING
Paramname	GEN OBJ REFID
Type	string
Format	
Unit	none
Function	giFluxConv
Author	blecha
Comment	
Paramdesc	Unique ID of reference source (same as LAMPID if type is LAMP)
Context	
Dicstatus	Missing (TBD by FLAMES consortium)
Dicrev	
I-Reffunctions	giFluxConv.exNffMast giFluxConv.skySubSp
O-Reffunctions	
Aiprefs	giFluxConv.refFluxMast
Aoprefs	
Reftables	girFlat girFluxcal refFlux
Nref	5

Alias	REFSRCMULT
Dicname	MISSING
Paramname	GEN OBJ REFMULT
Type	float
Format	
Unit	normed
Function	giFluxConv
Author	blecha
Comment	
Paramdesc	Factor which multiply the reference emissivity of the lamp to take into the account the setup-dependent fraction of the light injected into the fiber.
Context	
Dicstatus	Missing (TBD by FLAMES consortium)
Dicrev	
I-Reffunctions	giFiberTrans.exFullSlit giFiberTrans.exNffMast giFlatSpectra.exNffMast giFluxConv.exNffMast giFluxConv.skySubSp
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	
Nref	5

Alias	REFSRCOM
Dicname	MISSING
Paramname	GEN OBJ REFCOM
Type	string
Format	
Unit	none
Function	giFluxConv
Author	blecha
Comment	
Paramdesc	Comment on the reference source
Context	
Dicstatus	Missing (TBD by FLAMES consortium)
Dicrev	
I-Reffunctions	giFluxConv.exNffMast giFluxConv.skySubSp
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	
Nref	2

Alias	RON
Dicname	DRS
Paramname	DRS CCD RON
Type	float
Format	
Unit	e-
Function	
Author	blecha
Comment	
Paramdesc	Expected CCD readout noise
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	giExtractSpectra.dbSubImage giGetRemoveBias.biasMast giGetRemoveBias.rawFrame giLocalSpectra.dbSubImage
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	
Nref	4

Alias	SATLEVEL
Dicname	DRS
Paramname	DRS CCD SATLEV
Type	integer
Format	
Unit	e-
Function	giGetWaveSolution
Author	simond
Comment	Should include effect of both ADC conversion and the CCD physical saturation
Paramdesc	CCD saturation level
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	giGetWaveSolution.exSp
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	
Nref	1

Alias	SKCLIPMFRAC
Dicname	DRS
Paramname	PRO SKY SIGMA MFRAC
Type	float
Format	
Unit	none
Function	giModelSky
Author	cayatte
Comment	
Paramdesc	min fraction of points accepted/total (sigma-clipping)
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giModelSky.skyModel giModelSky.skyModelErr
Aiprefs	
Aoprefs	giModelSky.sNorm
Reftables	girSkyCoef girSkyMom
Nref	4

Alias	SKCLIPNITER
Dicname	DRS
Paramname	PRO SKY SIGMA NITER
Type	integer
Format	
Unit	none
Function	giModelSky
Author	cayatte
Comment	
Paramdesc	number of iterations (sigma-clipping)
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giModelSky.skyModel giModelSky.skyModelErr
Aiprefs	
Aoprefs	giModelSky.sNorm
Reftables	girSkyCoef girSkyMom
Nref	4

Alias	SKCLIPSIGMA
Dicname	DRS
Paramname	PRO SKY SIGMA MULT
Type	float
Format	
Unit	none
Function	giModelSky
Author	cayatte
Comment	
Paramdesc	multiple of sigma (sigma-clipping)
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giModelSky.skyModel giModelSky.skyModelErr
Aiprefs	
Aoprefs	giModelSky.sNorm
Reftables	girSkyCoef girSkyMom
Nref	4

Alias	SKCOMP
Dicname	DRS
Paramname	PRO SKY COMPONENT
Type	string
Format	
Unit	none
Function	giSplitSky
Author	north
Comment	
Paramdesc	sky component: one of 'EMISSION' or 'CONTINUUM'
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	giModelSky.normSky giModelSky.normSkyErr giNormalizeSky.wlFfSp giNormalizeSky.wlFfSpErr giSubtractSky.skyModel giSubtractSky.skyModelErr giSubtractSky.wlFfSp giSubtractSky.wlFfSpErr
O-Reffunctions	giModelSky.skyModel giModelSky.skyModelErr giNormalizeSky.normSky giNormalizeSky.normSkyErr giSplitSky.wlFfSkyComp giSplitSky.wlFfSkyCompErr
Aiprefs	giModelSky.intSky
Aoprefs	giModelSky.sNorm giNormalizeSky.intSky
Reftables	
Nref	14

Alias	SKINTDEGX
Dicname	DRS
Paramname	PRO SKY INTDEGX
Type	integer
Format	
Unit	none
Function	giModelSky
Author	blecha
Comment	replace table sInt
Paramdesc	X - degree of the polynomial describing the spatial variation of sky intensity
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giModelSky.skyModel giModelSky.skyModelErr
Aiprefs	
Aoprefs	giModelSky.sNorm
Reftables	girSkyCoef
Nref	3

Alias	SKINTDEGY
Dicname	DRS
Paramname	PRO SKY INTDEGY
Type	integer
Format	
Unit	none
Function	giModelSky
Author	blecha
Comment	replace table sInt
Paramdesc	X - degree of the polynomial describing the spatial variation of sky intensity
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giModelSky.skyModel giModelSky.skyModelErr
Aiprefs	
Aoprefs	giModelSky.sNorm
Reftables	girSkyCoef
Nref	3

Alias	SKINTXY
Dicname	DRS
Paramname	PRO SKY INTXYi
Type	float
Format	
Unit	none
Function	giModelSky
Author	blecha
Comment	replace table sInt
Paramdesc	Coefficients of (x,y) DEGXIS, DEGYIS degree normet-to-one polynomial describing the spatial variation of sky intensity. The coefficient are sorted in order x,y as c1+c2*x+c3*y+c4*x*y+c5*x*y2+c6*x2*y+c7*x3+ ...
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giModelSky.skyModel giModelSky.skyModelErr
Aiprefs	
Aoprefs	giModelSky.sNorm
Reftables	girSkyCoef
Nref	3

Alias	SKMODEL
Dicname	DRS
Paramname	PRO SKY MODEL
Type	integer
Format	
Unit	none
Function	giModelSky
Author	cayatte
Comment	
Paramdesc	SKY modelling: 0=no spatial distribution, 1 or more=spatial distribution
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giModelSky.skyModel giModelSky.skyModelErr
Aiprefs	
Aoprefs	giModelSky.sNorm
Reftables	girSkyCoef girSkyMom
Nref	4

Alias	SKSUB
Dicname	DRS
Paramname	PRO SKY SUBTRACTED
Type	string
Format	
Unit	none
Function	giSubtractSky
Author	north
Comment	
Paramdesc	subtracted sky component: one of 'WHOLE','EMISSION', 'CONTINUUM' or 'EMISSION+CONTINUUM'
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giSubtractSky.skySubSp giSubtractSky.skySubSpErr
Aiprefs	
Aoprefs	
Reftables	
Nref	2

Alias	SKSUBMETHOD
Dicname	DRS
Paramname	PRO SKY SUB METHOD
Type	integer
Format	
Unit	none
Function	giSubtractSky
Author	cayatte
Comment	
Paramdesc	1=direct subtraction of sky model, 2=amplitude adjustement for sky model intensity with cross-correlation method, 3=amplitude adjustement for sky model intensity with error minimization method
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giSubtractSky.skySubSp giSubtractSky.skySubSpErr
Aiprefs	
Aoprefs	
Reftables	
Nref	2

Alias	SKWLENMAX
Dicname	DRS
Paramname	PRO SKY WLEN MAX
Type	double
Format	
Unit	nm
Function	giNormalizeSky
Author	cayatte
Comment	
Paramdesc	SKY normalization: maximum wavelength
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	giModelSky.normSkyErr
O-Reffunctions	giNormalizeSky.normSky giNormalizeSky.normSkyErr
Aiprefs	
Aoprefs	giNormalizeSky.intSky
Reftables	girIntSky girSky
Nref	5

Alias	SKWLENMIN
Dicname	DRS
Paramname	PRO SKY WLEN MIN
Type	double
Format	
Unit	nm
Function	giNormalizeSky
Author	cayatte
Comment	
Paramdesc	SKY normalization: minimum wavelength
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	giModelSky.normSkyErr
O-Reffunctions	giNormalizeSky.normSky giNormalizeSky.normSkyErr
Aiprefs	
Aoprefs	giNormalizeSky.intSky
Reftables	girIntSky girSky
Nref	5

Alias	SLIT
Dicname	FGIR_ICS
Paramname	INS SLITS NO
Type	integer
Format	%d
Unit	none
Function	
Author	blecha
Comment	DIC: Position (hs). AL: What is the range, [0-4] ? I ma not happy with that.
Paramdesc	DIC: Index of the position of the slit selection carriage AL: It identify in a unique way (MODE and PLATE) the MEDUSA, ARGUS, IFU slit assemblies
Context	Instrument
Dicstatus	Development
Dicrev	*Revision* : 1.0
I-Reffunctions	giAdjustSL.locWyMast giAdjustSL.locYMast giAdjustSL.phffImageMast giAdjustSL.repPixImage giDeblendSpectra.exSp giDeblendSpectra.exSpErr giDeblendSpectra.locWy giDeblendSpectra.locY giDetectCosmicM.dbSubImage giEstimatePix.dbSubImage giEstimatePix.locWyMast giEstimatePix.locYMast giExtractSpectra.dbSubImage giExtractSpectra.locWy giExtractSpectra.locY giExtractSpectra.slImage giFiberTrans.exFullSlit giFiberTrans.exNffMast giFlatSpectra.exNffErrMast giFlatSpectra.exNffMast giFlatSpectra.exSp giFlatSpectra.exSpErr giFluxConv.exNffMast giFluxConv.skySubSp giGetWaveSolution.exSp giGetWaveSolution.locY giLocalSpectra.dbSubImage giLocalSpectra.locWyMast giLocalSpectra.locYMast giModelSky.normSky giNormalizeFSFF.ffFullSlit giNormalizeSky.wlFfSp giNormalizeSky.wlFfSpErr giRebinSpectra.ffSp giSplitSky.wlFfSp giVignCorr.wlFfSp
O-Reffunctions	giSplitSky.wlFfSkyComp giSplitSky.wlFfSkyCompErr
Aiprefs	giDeblendSpectra.psfParams giExtractSpectra.psfParams giFiberTrans.MeanNff giFiberTrans.VignSpectro giGetWaveSolution.linesSkyMast giGetWaveSolution.psfXwidthMast giGetWaveSolution.slitGeometryMast giGetWaveSolution.waveSolutionMast giLocalSpectra.psfParamsMast giModelSky.intSky giNormalizeSky.numSky giRebinSpectra.waveSolution giSplitSky.linesSkyMast giSplitSky.numSky
Aoprefs	giModelSky.sNorm giNormalizeSky.intSky
Reftables	girFiberTrans girFlat girFluxcal girIntSky girLine girMeanSp girPsf girPsfXwid girSkyCoef girSky girSkyMom girSlitGeo girTransVlt girVignCorr girVignSpectro girWavCoef
Nref	54

Alias	SLMODEL
Dicname	DRS
Paramname	PRO SCL MODEL
Type	string
Format	
Unit	none
Function	giAdjustSL
Author	royer
Comment	
Paramdesc	option indicating the model of scattered light: 'POLYNOM' or 'POLYFRAC'
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giAdjustSL.slImage
Aiprefs	
Aoprefs	
Reftables	
Nref	1

Alias	SLPHFFCOR
Dicname	DRS
Paramname	PRO SCL PHFFCOR
Type	logical
Format	
Unit	none
Function	giAdjustSL
Author	royer
Comment	
Paramdesc	option for correcting input frame frome photometric flat-field
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giAdjustSL.slImage
Aiprefs	
Aoprefs	
Reftables	
Nref	1

Alias	SLPOLYDEG1
Dicname	DRS
Paramname	PRO SCL POLYDEG1
Type	integer
Format	
Unit	none
Function	giAdjustSL
Author	royer
Comment	
Paramdesc	degree of the numerator polynom of the SL model
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giAdjustSL.slImage
Aiprefs	
Aoprefs	
Reftables	
Nref	1

Alias	SLPOLYDEG2
Dicname	DRS
Paramname	PRO SCL POLYDEG2
Type	integer
Format	
Unit	none
Function	giAdjustSL
Author	royer
Comment	
Paramdesc	degree of the denominator polynom of the SL model
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giAdjustSL.slImage
Aiprefs	
Aoprefs	
Reftables	
Nref	1

Alias	STARNAME
Dicname	DRS
Paramname	PRO DATA STAR
Type	logical
Format	
Unit	none
Function	giFluxConv
Author	blecha
Comment	Better name TBF
Paramdesc	ID of standard star(s)
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	girFluxcal
Nref	1

Alias	STARTX
Dicname	CCDDCS
Paramname	DETi WINi STRX
Type	integer
Format	%d
Unit	none
Function	
Author	blecha
Comment	DIC: Lower left pixel in X AL: (usually 1, but <0 if we have image with PRESCAN)
Paramdesc	DIC: First window pixel in X direction within the detector physical system.
Context	Detector
Dicstatus	Released
Dicrev	3.6
I-Reffunctions	giAdjustSL.phffImageMast giAdjustSL.repPixImage giArithmetics.inFrame1 giArithmetics.inFrame2 giDetectCosmicM.dbSubImage giDetectCosmicS.dbSubImage giDetectCosmicS.estimateImage giDetectCosmicS.estimateSigmaImage giEstimatePix.dbSubImage giExtractSpectra.dbSubImage giExtractSpectra.slImage giGetRemoveBias.biasMast giGetRemoveBias.rawFrame giGetWaveSolution.exSp giLocalSpectra.dbSubImage giNormalizeFSFF.ffFullSlit giReplaceFlagPix.dbSubImage giReplaceFlagPix.estimateImage giSubtractDark.bSubImage giSubtractDark.darkMast
O-Reffunctions	giArithmetics.outFrame giExtractSpectra.cleanImage giExtractSpectra.slImage giReplaceFlagPix.repPixImage
Aiprefs	
Aoprefs	
Reftables	
Nref	24

Alias	STARTY
Dicname	CCDDCS
Paramname	DETi WINi STRY
Type	integer
Format	%d
Unit	none
Function	
Author	blecha
Comment	DIC: Lower left pixel in Y AL: (usually 1, but <0 if we have image with PRESCAN)
Paramdesc	DIC: First window pixel in Y direction within the detector physical system.
Context	Detector
Dicstatus	Released
Dicrev	3.6
I-Reffunctions	giAdjustSL.phffImageMast giAdjustSL.repPixImage giArithmetics.inFrame1 giArithmetics.inFrame2 giDetectCosmicM.dbSubImage giDetectCosmicS.dbSubImage giDetectCosmicS.estimateImage giDetectCosmicS.estimateSigmaImage giEstimatePix.dbSubImage giExtractSpectra.dbSubImage giExtractSpectra.slImage giGetRemoveBias.biasMast giGetRemoveBias.rawFrame giGetWaveSolution.exSp giLocalSpectra.dbSubImage giNormalizeFSFF.ffFullSlit giReplaceFlagPix.dbSubImage giReplaceFlagPix.estimateImage giSubtractDark.bSubImage giSubtractDark.darkMast
O-Reffunctions	giArithmetics.outFrame giExtractSpectra.cleanImage giExtractSpectra.slImage giReplaceFlagPix.repPixImage
Aiprefs	
Aoprefs	
Reftables	
Nref	24

Alias	STATUS
Dicname	DRS
Paramname	PRO DATA STATUS
Type	string
Format	
Unit	none
Function	
Author	blecha
Comment	Standard table KW
Paramdesc	Comment on last update
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	giTableStandard
Nref	1

Alias	TELESCOPE
Dicname	TCS
Paramname	TEL ID
Type	string
Format	%30s
Unit	none
Function	
Author	blecha
Comment	DIC: TCS version number
Paramdesc	DIC: cmm version number of the integration module (tcsBUILD/ntt), which defines the version of all TCS modules.
Context	Telescope
Dicstatus	Released
Dicrev	1.80
I-Reffunctions	giFluxConv.exNffMast giFluxConv.skySubSp
O-Reffunctions	
Aiprefs	giFluxConv.transVltMast
Aoprefs	
Reftables	girTransVlt
Nref	3

Alias	TITLE
Dicname	DRS
Paramname	PRO DATA TITLE
Type	string
Format	
Unit	none
Function	
Author	blecha
Comment	Standard table KW
Paramdesc	Title of data structure
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	giTableStandard
Nref	1

Alias	UNALLFACB
Dicname	FPMISS
Paramname	undef
Type	integer
Format	
Unit	none
Function	
Author	
Comment	
Paramdesc	number of unallocated guide stars
Context	
Dicstatus	Missing (TBD by FLAMES consortium)
Dicrev	
I-Reffunctions	
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	girOzpoz
Nref	1

Alias	UNALLOBJ
Dicname	FPMISS
Paramname	undef
Type	integer
Format	
Unit	none
Function	
Author	
Comment	
Paramdesc	number of unallocated program objects
Context	
Dicstatus	Missing (TBD by FLAMES consortium)
Dicrev	
I-Reffunctions	
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	girOzpoz
Nref	1

Alias	UNALLSKY
Dicname	FPMISS
Paramname	undef
Type	integer
Format	
Unit	none
Function	
Author	
Comment	
Paramdesc	number of unallocated sky positions
Context	
Dicstatus	Missing (TBD by FLAMES consortium)
Dicrev	
I-Reffunctions	
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	girOzpoz
Nref	1

Alias	WLEN0
Dicname	FGIR_ICS
Paramname	INS GRAT WLEN
Type	double
Format	%.1f
Unit	nm
Function	giLocalSpectra
Author	simond
Comment	DIC: Grating central wavelength [nm] (hs).
Paramdesc	DIC: Wavelength that the grating transmits along the optical axis of the instrument. AL: grating central wavelength
Context	Instrument
Dicstatus	Development
Dicrev	*Revision* : 1.0
I-Reffunctions	giAdjustSL.locWyMast giAdjustSL.locYMast giAdjustSL.phffImageMast giAdjustSL.repPixImage giDeblendSpectra.exSp giDeblendSpectra.exSpErr giDeblendSpectra.locWy giDeblendSpectra.locY giDetectCosmicM.dbSubImage giEstimatePix.dbSubImage giEstimatePix.locWyMast giEstimatePix.locYMast giExtractSpectra.dbSubImage giExtractSpectra.locWy giExtractSpectra.locY giExtractSpectra.slImage giFiberTrans.exFullSlit giFiberTrans.exNffMast giFlatSpectra.exNffErrMast giFlatSpectra.exNffMast giFlatSpectra.exSp giFlatSpectra.exSpErr giFluxConv.exNffMast giFluxConv.skySubSp giGetWaveSolution.exSp giGetWaveSolution.locY giLocalSpectra.dbSubImage giLocalSpectra.locWyMast giLocalSpectra.locYMast giModelSky.normSky giNormalizeFSFF.ffFullSlit giNormalizeSky.wlFfSp giNormalizeSky.wlFfSpErr giRebinSpectra.ffSp giSplitSky.wlFfSp giVignCorr.wlFfSp
O-Reffunctions	giSplitSky.wlFfSkyComp giSplitSky.wlFfSkyCompErr
Aiprefs	giDeblendSpectra.psfParams giExtractSpectra.psfParams giFiberTrans.MeanNff giFiberTrans.VignSpectro giGetWaveSolution.linesSkyMast giGetWaveSolution.psfXwidthMast giGetWaveSolution.waveSolutionMast giLocalSpectra.psfParamsMast giModelSky.intSky giNormalizeSky.numSky giRebinSpectra.waveSolution giSplitSky.linesSkyMast giSplitSky.numSky
Aoprefs	giModelSky.sNorm giNormalizeSky.intSky
Reftables	girFiberTrans girFlat girFluxcal girIntSky girLine girMeanSp girPsf girPsfXwid girSkyCoef girSky girSkyMom girTransVlt girVignSpectro girWavCoef
Nref	52

Alias	WLENMAX
Dicname	DRS
Paramname	DRS GRAT WLORDMAX
Type	float
Format	
Unit	nm
Function	giGetWaveSolution
Author	blecha
Comment	
Paramdesc	Longest wavelength in the current setup
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	giGetWaveSolution.exSp giRebinSpectra.ffSp
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	
Nref	2

Alias	WLENMIN
Dicname	DRS
Paramname	DRS GRAT WLORDMIN
Type	float
Format	
Unit	nm
Function	giGetWaveSolution
Author	blecha
Comment	
Paramdesc	Shortest wavelength in the current setup
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	giGetWaveSolution.exSp giRebinSpectra.ffSp
O-Reffunctions	
Aiprefs	
Aoprefs	
Reftables	
Nref	2

Alias	WSCLIPMFRAC
Dicname	DRS
Paramname	PRO WSOL SIGMA MFRAC
Type	float
Format	
Unit	none
Function	giGetWaveSolution
Author	simond
Comment	
Paramdesc	min fraction of points accepted/total (sigma-clipping)
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giGetWaveSolution.fitWxl
Aiprefs	
Aoprefs	giGetWaveSolution.cleanlineTableMW giGetWaveSolution.cleanlineTableMWSky giGetWaveSolution.psfXwidth giGetWaveSolution.waveSolution
Reftables	girLine girPsfXwid girWavCoef
Nref	4

Alias	WSCLIPNITER
Dicname	DRS
Paramname	PRO WSOL SIGMA NITER
Type	integer
Format	
Unit	none
Function	giGetWaveSolution
Author	simond
Comment	
Paramdesc	number of iterations (sigma-clipping)
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giGetWaveSolution.fitWxl
Aiprefs	
Aoprefs	giGetWaveSolution.cleanlineTableMW giGetWaveSolution.cleanlineTableMWSky giGetWaveSolution.psfXwidth giGetWaveSolution.waveSolution
Reftables	girLine girPsfXwid girWavCoef
Nref	4

Alias	WSCLIPSIGMA
Dicname	DRS
Paramname	PRO WSOL SIGMA MULT
Type	float
Format	
Unit	none
Function	giGetWaveSolution
Author	simond
Comment	
Paramdesc	multiple of sigma (sigma-clipping)
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giGetWaveSolution.fitWxl
Aiprefs	
Aoprefs	giGetWaveSolution.cleanlineTableMW giGetWaveSolution.cleanlineTableMWSky giGetWaveSolution.psfXwidth giGetWaveSolution.waveSolution
Reftables	girLine girPsfXwid girWavCoef
Nref	4

Alias	WSCOEFi
Dicname	DRS
Paramname	PRO WSOL NX
Type	float
Format	
Unit	none
Function	giGetWaveSolution
Author	simond
Comment	
Paramdesc	PSF X-width coef
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	giRebinSpectra.fitWxl
O-Reffunctions	giGetWaveSolution.fitWxl
Aiprefs	
Aoprefs	giGetWaveSolution.cleanlineTableMW giGetWaveSolution.cleanlineTableMWSky giGetWaveSolution.psfXwidth giGetWaveSolution.waveSolution
Reftables	girLine girPsfXwid girWavCoef
Nref	5

Alias	WSLINETYPE
Dicname	DRS
Paramname	PRO WSOL LINETYPE
Type	logical
Format	
Unit	none
Function	giGetWaveSolution
Author	simond
Comment	
Paramdesc	use of ThArNe or telluric lines for Wavelength Solution
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	giRebinSpectra.fitWxl
O-Reffunctions	giGetWaveSolution.fitWxl
Aiprefs	
Aoprefs	giGetWaveSolution.cleanlineTableMW giGetWaveSolution.cleanlineTableMWSky giGetWaveSolution.psfXwidth giGetWaveSolution.waveSolution
Reftables	girLine girPsfXwid girWavCoef
Nref	5

Alias	WSNS
Dicname	DRS
Paramname	PRO WSOL NY
Type	integer
Format	
Unit	pixel
Function	giGetWaveSolution
Author	simond
Comment	
Paramdesc	X spectrum
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	giRebinSpectra.fitWxl
O-Reffunctions	giGetWaveSolution.fitWxl
Aiprefs	
Aoprefs	
Reftables	
Nref	2

Alias	WSWINDOW
Dicname	DRS
Paramname	PRO WSOL WINDOW
Type	float
Format	
Unit	pixel
Function	giGetWaveSolution
Author	simond
Comment	
Paramdesc	width in pixel for line detection and fit
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	giRebinSpectra.fitWxl
O-Reffunctions	giGetWaveSolution.fitWxl
Aiprefs	
Aoprefs	giGetWaveSolution.cleanlineTableMW giGetWaveSolution.cleanlineTableMWSky giGetWaveSolution.psfXwidth giGetWaveSolution.waveSolution
Reftables	girLine girPsfXwid girWavCoef
Nref	5

Alias	WSWLENXDEG
Dicname	DRS
Paramname	PRO WSOL WLEN XDEG
Type	integer
Format	
Unit	none
Function	giGetWaveSolution
Author	simond
Comment	
Paramdesc	polynomial degree along the x direction for wavelength solution
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giGetWaveSolution.fitWxl
Aiprefs	
Aoprefs	giGetWaveSolution.cleanlineTableMW giGetWaveSolution.cleanlineTableMWSky giGetWaveSolution.psfXwidth giGetWaveSolution.waveSolution
Reftables	girLine girPsfXwid girWavCoef
Nref	4

Alias	WSWLENYDEG
Dicname	DRS
Paramname	PRO WSOL WLEN YDEG
Type	integer
Format	
Unit	none
Function	giGetWaveSolution
Author	simond
Comment	
Paramdesc	polynomial degree along the y direction for wavelength solution
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giGetWaveSolution.fitWxl
Aiprefs	
Aoprefs	giGetWaveSolution.cleanlineTableMW giGetWaveSolution.cleanlineTableMWSky giGetWaveSolution.psfXwidth giGetWaveSolution.waveSolution
Reftables	girLine girPsfXwid girWavCoef
Nref	4

Alias	WSXWIDXDEG
Dicname	DRS
Paramname	PRO WSOL XWIDTH XDEG
Type	integer
Format	
Unit	none
Function	giGetWaveSolution
Author	simond
Comment	
Paramdesc	polynomial degree along the x direction for x-width modeling
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giGetWaveSolution.fitWxl
Aiprefs	
Aoprefs	giGetWaveSolution.cleanlineTableMW giGetWaveSolution.cleanlineTableMWSky giGetWaveSolution.psfXwidth giGetWaveSolution.waveSolution
Reftables	girLine girPsfXwid girWavCoef
Nref	4

Alias	WSXWIDYDEG
Dicname	DRS
Paramname	PRO WSOL XWIDTH YDEG
Type	integer
Format	
Unit	none
Function	giGetWaveSolution
Author	simond
Comment	
Paramdesc	polynomial degree along the y direction for x-width modeling
Context	
Dicstatus	NEW (Proposal June 2000)
Dicrev	
I-Reffunctions	
O-Reffunctions	giGetWaveSolution.fitWxl
Aiprefs	
Aoprefs	giGetWaveSolution.cleanlineTableMW giGetWaveSolution.cleanlineTableMWSky giGetWaveSolution.psfXwidth giGetWaveSolution.waveSolution
Reftables	girLine girPsfXwid girWavCoef
Nref	4

8 Function Traceability Matrix

Item (spec.)	Compliance	Reference
Cosmic rays and bad pixels correction	YES	3.6, 3.7 ,3.8 ,3.9
Image statistics	YES	
Image truncation	YES	3.4 (overscan zones)
Image combination	YES	low level functional specifications of 3.8
Optimal spectrum extraction	YES	3.12
Sky subtraction	YES	sky normalization 3.21 sky modeling 3.22 sky subtraction 3.23
Optimal weighted sum of spectra with different S/N	NO	This overlaps with ADAS. The specification is being postponed till the ADAS release
λ calibration based on calibration frames or on sky spectra	YES	3.17
λ calibration based on a model of dispersion law vs. slit coordinates	YES	(mixed with preceding item) 3.17
Flux calibration	PARTIAL	conversion to physical flux 3.24
Multi-fibre image reconstruction	NO	This overlaps with ADAS. The specification is being postponed till the ADAS release

9 Development And Test Factors

9.1 Development Requirements and constraints

The most challenging aspect of the DRS development is the necessity to fulfill two apparently contradictory requirements:

1. Build a software package little dependent on a very specific VLT environment.

   This requirement is dictated by the laboratory tests to be carried out under SW/HW conditions far from VLT environment, by the not-yet-mature state of the current VLT DFS and the astronomer's need to re-process or further process data produced by DFS in a non-VLT environment.

2. Develop and test the SW in a most realistic possible VLT environment.

   This requirement merely reflects the primary DRS goal which is to supply the VLT DFS with working recipes as soon as the FLAMES is assembled for the commissioning at Paranal.

The above considerations leads the development team to following principles of development:

1. We keep a strict VLT compliance at the function level (SW modules). Here the VLT environment is clearly defined. Though the key element - the Data Interface Dictionary - is unfortunately not available at the appropriately updated level, the associated data-structures are stable (FITS), well tested and fitted with the existing public domain interface software of good quality.
2. The chaining of modules is VLT compliant in the sense that the basic mechanisms is built in data-structures itself and all informations necessary for the execution of a module are transmitted via FITS keyword of input data-frames.
3. The control of the execution of the module chain (recipe) is VLT independent. The necessary tasks will be accomplished by a shell-scripts which could be run within or outside the VLT environment. At the DRS level we assume that all initial calibration and control data are available (through the *Data Organizer* in DFS) and we only check that this is the case. For development and test purpose only a test data-base and data-selection selection tools will be built.

9.2 Implementation Phases

9.2.1 Prototyping

The prototyping is not a necessary step for all elements of the SW to be developed. Only the aspects which we are uncertain of should be subject to a prototype development. The DRS prototypes aim at two goals:

- Develop and test the SW modules functionalities and the overall integrity of the SW structure.

  Under the standard VLT environment we expect to prototype only the SW structure. The main concern here is the coherence and integrity of data-structures.

- Test, validate and possibly develop the algorithms.

  We expect to pursue the algorithms validation under MATLAB environment. Algorithms, once accepted will, be *ported* to C code.

9.2.2 Beta Version

The validation procedure will use of followings elements:

- A representative set of simulated input frames for selected modes.
  - FF frames
  - Wavelength Calibration Frames
  - Science Frame
- A representative set of auxiliary data (normally retrieved from Archive by Data Organizer).
- One or several validation recipes calling all functions and recipe-associated data description.

Functions Index

G

Data Tables Index

G

Inputs/Outputs Index

Frame types/FITS Keywords Index

__oOo__