CARACAL: pipeline manual

Document:	CA-CS-DP-MA-001
Short title:	CARACAL: pipeline manual
Issue number:	1.02
Issue date:	24/June/2015
Nº of pages:	35

	Name	Centre	Date
Prepared:	M. Zechmeister	IAG	21/Apr/2015
Revised:	A. Reiners	IAG	06/May/2015
Approved:	L. Gesa	ICE	
Authorised:	A. Perez-Calpena	Fractal	

Document change record

Author	issue	date	pages	Description
M. Zechmeister	1.00	21-Apr-2015	All	Draft
M. Zechmeister	1.01	08-May-2015	All	Revised draft
M. Zechmeister	1.02	23-Jun-2015	All	Revised draft
M. Zechmeister	1.03	xx-Sep-2015	All	Revised draft

Contents

List of Figures

List of Tables

1. Overview

CARMENES is a planet search program at Calar Alto and consists of two spectrographs (Quirrenbach et al., 2012). Both echelle spectrographs have a resolving power of 82 000.

The CARMENES pipeline receives the raw spectra, extracts them and measures the radial velocity. This document describes the installation, usage and organisation of the pipeline. It consist of three parts:

- CSFS - **C**ARMENES **S**pectral **F**orward **S**imulator
- CARACAL - **CA**RMENES **R**eduction **A**nd **CAL**ibration software
- SERVAL - **S**p**E**ctrum **R**adial **V**elocity **A**na**L**yser (RV pipeline)

CSFS can be used to generate test data which have the format of CARMENES raw spectra (note: the order curvature deviates by up to 20 pixels). Different inputs (e.g. PHOENIX spectrum, line list, laser comb, Fabry-Perot) are traced with a simple physical model of the spectrograph.

CARACAL is the extraction software. The starting point for the development of CARACAL was the IDL package REDUCE of Piskunov & Valenti (2002) (which is the reason why CARACAL is implemented in IDL). However, many parts have been modified and a number of new scripts have been added.

SERVAL computes the RVs by least square fitting (similar as in Anglada-Escudé & Butler, 2012).

This document gives a general overview as well as important details about the software for requirements, interfaces, data organisation and usage.

2. Requirements, Hardware and Software Environment

2.1. General

The pipeline was developed in the environment of:

- Linux/OpenSuse
- bash

At Calar Alto the two computers banard and lalande are in normal operation dedicated for the data extraction of the visual and near-infrared spectrograph, respectively. The software described below is installed on both computers and generally identical (exceptions will be noted). On both computers the user account pipedev is forseen to maintain the pipeline. (Furthermore, there is a root/admin account. Guest accounts with restricted access might be possible.)

The computers barnard and lalande are located in the ICS rack in the new 3.5m computer room. They have no monitor or keyboard and can be accessed only remotely. To access the pipeline on both computer, the observers and telescope operators have to use the computers kapteyn and lacaille which are located in the control room in laboratory and are equipped with monitors.

Pipeline developers (pipedev) should be able to have remote access e.g. with

```
> ssh pipedev@barnard.caha.es
```

or vncviewer (which is more convenient to work with)

```
> vncviewer barnard.caha.es::5900
```

2.2. CSFS - Forward simulator

- csfs source code (maintained by C. Marvin, IAG, https://bitbucket.org/cmarvin/csfs/)

- GCC GNU Scientific Library (GSL)
- GNU Make
- ds9 (for viewing output FITS)
- Python 2.6 or 2.7
  - NumPy 1.6.1 or later
  - Scipy 0.12 or later
  - Matplotlib 1.1 or later
  - Astropy 0.3 (or Pyfits 3.0) or later

2.3. CARACAL - Extraction pipeline

- IDL v6.3
- IDL CARACAL package (distributed by M. Zechmeister)
- inotify
- xterm
- ds9 (`http://hea-www.harvard.edu/RD/ds9/site/Download.html`)
- gnuplot
- Tk/Tcl 8.5
- BarCor (`http://sirrah.troja.mff.cuni.cz/~mary/barcor.for`, already included in CARACAL)

2.4. SERVAL - RV pipeline

- SERVAL source code (distributed by M. Zechmeister)
- Python 2.7.2 including packages: numpy, scipy
- f2py
- gcc
- gnuplot
- bash
- BarCor (`http://sirrah.troja.mff.cuni.cz/~mary/barcor.for`, already included in CARACAL)

2.5. Miscellaneous

- git (to install CSFS)
- latex2html (for QC web plots)
- gps (gnuplot script to directly plot fits spectra)
- imhead (simple bash-awk fits header script)
- ech2dat (python script to convert fits echelle spectra to a ascii table)

3. CSFS - CARMENES spectrum forward simulator

The purpose of CSFS is to test the extraction and RV pipeline. Normal pipeline users will not need CSFS. For completeness some instructions are given here. CSFS is mainly written in Python and some time critical parts in c.

3.1. Installation

The source code including installation instruction is online available: https://bitbucket.org/cmarvin/csfs/. The code can be downloaded with

```
> git clone https://cmarvin@bitbucket.org/cmarvin/csfs.git ~/programs/csfs/csfs_v5.6.0
```

After download, CSFS can be installed by typing

```
> make
```

CSFS (v5.5.0) is currently installed on barnard[1] (~/programs/csfs/).

3.2. Usage

CSFS is used from the command line and has a number of options. Appendix A.4 shows the help output of

```
> python csfs -h
```

in detail.

CSFS outputs fits files whose format and header mimic the real format. Appendix A.6 shows a header created by csfs (plus a number of keywords added by CARACAL). Figure 3.1 a section of the CSFS output created by using the command

```
> python csfs.py N 2 -mc -a P -b X --noise --blaze -nr 1e8 --gap --tell CO2 H2O O2
```

and displayed using the mosaicimage option of ds9

```
> ds9 -mosaicimage iraf output/csfs.2015-04-16T11:34:15-nir-phofab.fits.gz
```

4. CARACAL - Extraction pipeline

4.1. Installation

The pipeline is packed in one tar-file (currently caracal_v0.04.tar in the home directory) which contains a howtodo.txt (see also Sect. A.5). Please follow the installing instruction there.

4.2. CARMENES workflow and data organisation

In normal mode at the telescope, the observer/telescope operator just has to execute the pipestart command (via command line or the xpipe GUI) which starts the pipeline trigger (see Sec. 4.3.2 for more details).

[1]During installation a problem occured because gsl was missing. gsl is currently installed as user in ~/programs/gsl-1.16. This required a modification in the Makefile:
@line: LIBS = -lgsl -lgslcblas append: -I/disk-a/pipedev/usr/include -L/disk-a/pipedev/usr/lib64
and also an environement variable must be set
export LD_LIBRARY_PATH=/disk-a/pipedev/usr/lib64

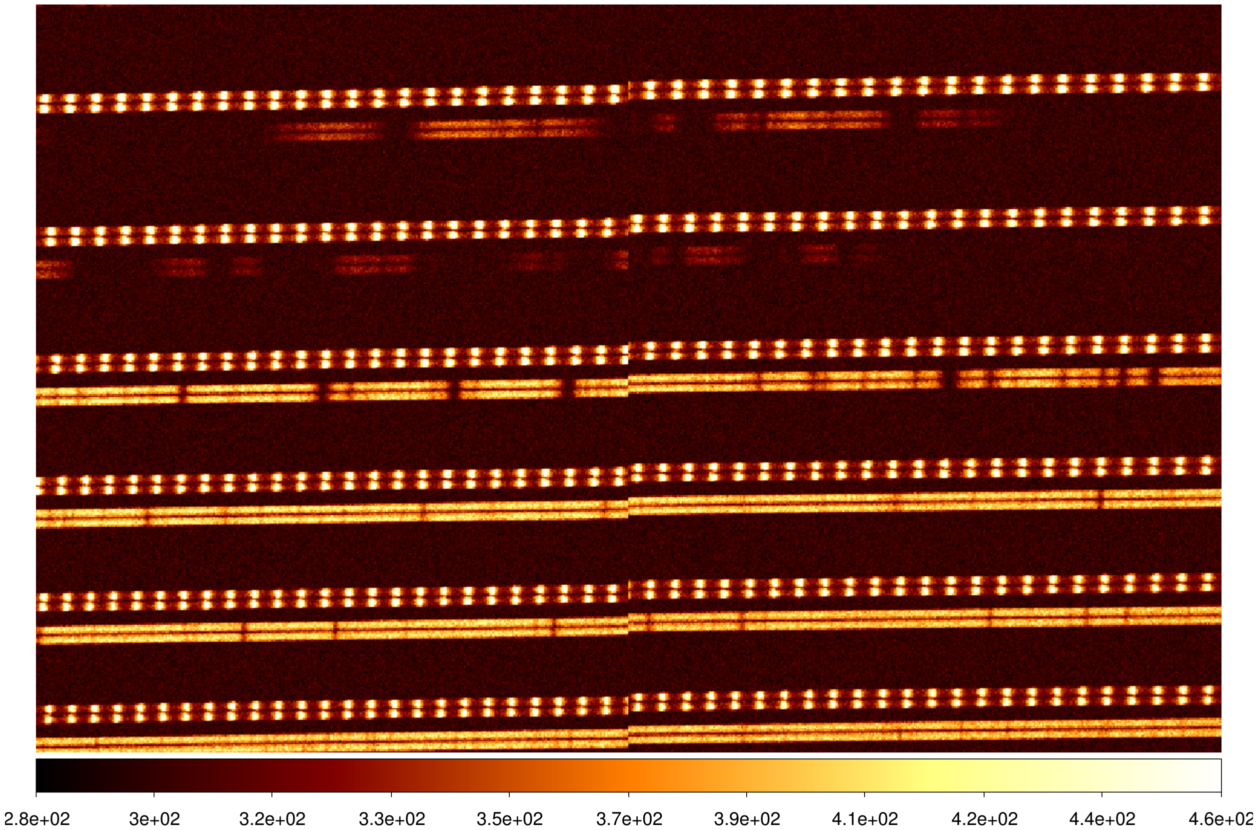

Figure 3.1.: Zoom-in to a CSFS NIR spectrum (fibre A: star, fibre B: FP). The jump in orders (center) is due to the gap between the two mosaic NIR detector. At the top the stellar spectrum has strong telluric absorption.

Table 4.1.: File and data organisation.

Files/data	location
raw frames	/disk-b/data/raw/yymmdd/
reduced science frame	/disk-b/data/red/yymmdd/
calibration products	/disk-b/data/prod/yymmdd/
database	/disk-b/data/db/yymmdd/
list of noticed raw files	rawframes.log
list of processed raw files	redframes.log
status and history of trigger	piperun.log
working directory	~/caracal/
this manual	~/caracal/caracal_manual.pdf
caracal GUI	~/bin/xpipe
source code	~/idl/caracal/

Table 4.2.: Overview of some important IDL scripts/files.

file	location	comment
caracal.pro	~/idl/caracal/	main script (file classification, task selection, trigger)
param_carm_vis.pro	~/caracal/src/	parameter file for extraction (file location)
modeinfo_carm.pro	~/idl/reduce/REDUCE	scans the header of the fits files
carm_vis_A_2D.sav	~/idl/reduce/WAVE_2D	start guess for wavelength solution
fitwls2D.pro	~/idl/caracal/wavecal	wavelength calibration algorithm

In trigger mode CARACAL waits for new raw files which are provided by the instrument control system (ICS) and should be located at /disk-b/data/raw/YYMMDD/. Details about the interface between CARACAL and ICS can be found in the Pipeline-ICS-CAHA Interface Document (CA-CS-IF-001).

The raw files are automatically processed and are not modified by CARACAL. All information needed for the data reduction must be passed via fitsheader keywords (e.g. calibration type).

The calibration and data reduction products are stored in separate directories and then ready for archiving. Table 4.1 summarises the data organisation.

In special cases, like development and re-processing, CARACAL can also be used at lower level. The CARACAL is started directly and individual task and options can be selected (Sect. 4.4). We call this mode reduce.

4.3. CARACAL - Extraction pipeline

4.3.1. Setup and file organisation

Each parameter file (param_carm_vis.pro and param_carm_nir.pro) contains several setup parameters, e.g.:

- pathnames for directories of raw frames, product files, reduced files, databases
- fits keyword used for classification
- extraction parameters (e.g. extraction width)
- quality control parameter (e.g. number of orders to identify)
- reference files (database filenames) to be used for quality control

4.3.2. Pipeline: start, trigger, end

Start: There are several possibilities to start the automatic pipeline. Automatic means that the pipeline runs without any human interaction. This mode is foreseen for the real-time data processing at the telescope. A trigger will be started and detects the arrival of new files, which are then processed by the pipeline. To start the pipeline in its automatic mode, one can type on the IDL command line

```
IDL> pipestart
```

On unix command line, one can also type

```
> pipestart
```

This will start an IDL session as well as the IDL routine pipestart above.

Table 4.3.: Overview of CARACAL tasks.

task	purpose	product suffixes
mstbias	master bias	*_mbias.fits
	linearity calibration	
mstflat	master flat	*_flat.fits
odef	order definition	*_ord_A.sav, *_ord_A.reg, *_bgmap.fits.gz
mflat	straylight subtracted master flat	*_mflat.fits.gz
nflat	normalised flat	*_nflat_vis.fits, *_blaze_A.fits
wave	extraction of wavelength calibration files	*_wave_A.fits
wavecal	wavelength calibration	
waverecal	wavelength re-calibration	*_A.fits, *_A.re2D.sav
obj	extraction of object spectra	*_x2d_A.fits, *_x2d_B.fits
calflux	flux calibration	*_eff_coeffs.sav, *_eff_A.fits
?	guess task	

Finally, also the button pipestart can be pressed in xpipe. This will start an xterm console (within xpipe; white panel in Fig. 4.3), an IDL session, and the trigger script.

It is also possible to run the reduction manually and without the trigger. This can be used e.g. for data re-processing. An example for a command in the manual mode within the IDL environment is

```
IDL> caracal, 'obj', nite='20130404', inst='carm_vis'
```

It allows the user to pass some more parameters and to select specific reduction tasks, hence providing more manual control. E.g. spectra could be re-processed with different calibration set. With some more limitation a re-processing is also possible with xpipe via the reduce button.

Trigger: When the Trigger is started, first, the directory is searched for unprocessed files. If there are unprocessed files, they are passed to the pipeline. The processing status (extracted/unprocessed/failed) of the files is logged (redframes.log). Then, using the unix command *inotify*, the trigger waits until it notifies the arrival of new files or the modification of the ~/caracal/piperun.log file. The later event terminates the trigger, otherwise the loop is repeated which again begins with searching for unprocessed files. The pipeline is started with the '?' task, meaning that the tasks will be guessed, based on the fits header entries. In case there are not yet enough frames to complete the corresponding task, like the creation a master flat, the pipeline does not process these frames, but returns to the trigger and waits for more frames in the next trigger loop. The directories which are monitored are inferred from the instrument parameter file.

Exit: pipeexit will append the current time to piperun.log and this file modification event leads to the termination of the trigger loop. pipeexit can be executed as unix command

```
> pipeexit
```

or by pressing the xpipe button pipeexit. (If the IDL session was started automatically, it terminates automatically on pipeexit.)

4.4. Data reduction and tasks

The data processing starts with the call of the IDL script caracal.pro. The first step is a classification of the raw frames. Each fits header is scanned and the corresponding tasks are identified from appropriate fits keywords (SOURCE, ...). Then the files are passed to the tasks described below and in Table 4.3. For visual real-time quality control (QC), the current processed raw spectra, products and extracted spectra are displayed in a ds9 window entitled QC.CARACAL.

For each task a quality check should be performed. If the check is passed, the calibration database is updated and the calibration product used for subsequent data processing. The calibration products are updated with fits-keywords (Table 4.4).

CARACAL deals with three file formats for input and output: fits files (*.fits, *.fits.gz), IDL files (*.sav) and ds9 region files (*.reg). The suffixes are appended to the basenames of the raw files and indicate the type of the output products e.g

- car-20150506T02h10m30s-cali-nir_wave_A.fits
- car-20150506T04h53m16s-gtoc-vis_x2d_A.fits

4.4.1. Master bias

Input: bias frames ($n_{\text{bias}} \geq 4$) , tag: SOURCE == 'OFF,OFF'

Description: Creation of a mean master bias from at least 4?? bias observations

Table 4.4.: Example of HIERARCH CARACAL fits cards.

fits keyword	value	fits comment
CARACAL MBIAS NIMAGES	int	number of images summed
CARACAL MBIAS NPIXFIX	int	pixels corrected for cosmic rays
CARACAL MBIAS BGNOISE	float	[e-/pix] background noise in mbias
CARACAL MBIAS RON	float	[e-/pix] background noise per image
C ARACAL MBIAS QC	int	passed|failed
CARACAL MFLAT NPIXFIX	int	pixels corrected for cosmic rays
CARACAL MFLAT QC	int	passed|failed
CARACAL WAVERECAL WAVESOLTYPE	int	
CARACAL WAVERECAL RMS	float	[m/s] scatter around solution
CARACAL WAVERECAL RMS PIX	float	[pix]
CARACAL WAVERECAL HCL NUSED	int	number of used lines
CARACAL WAVERECAL HCL NCLIP	int	number of clipped lines
CARACAL WAVERECAL HCL WIDTH	float	[pix] median line width
CARACAL WAVERECAL HCL RESOLUTION	float	median value
CARACAL WAVERECAL QC	int	passed|failed
CARACAL BIAS FILE	float	YYYY-MM-DD
CARACAL ODEF FILE	float	YYYY-MM-DD
CARACAL FLAT FILE	float	YYYY-MM-DD
CARACAL WAVE FILE	float	YYYY-MM-DD
E_BERV	float	[km/s] Barycentric correction
CARACAL FOX XWD	int	extraction width
CARACAL FOX KAPPA	int	kappa-sigma clipping threshold
CARACAL FOX SNR 0	float	SNR in order 64
CARACAL FOX CHI2RED 0	float	sqrt(chi2red) in order 64
CARACAL FOX VERSION	string	Author: M. Zechmeister
CARACAL USERHOST	string	
CARACAL VERSION	string	Author: M. Zechmeister
CARACAL DRIFT REF	string	Ref file used
CARACAL DRIFT RV	string	[m/s] Drift in cal fib
CARACAL DRIFT RVERR	string	[m/s] Estimated RV uncertainty

Output: Master Bias (car-$YYYYMMDD$Thhhmmmsss-cali-vis_bias.fits)
Print and display of the results
Measured read noise (fits header)

QC check: readout noise (RON) $<$ 10 e-/pix??
number of bad pixels $<$??

Note: Creation of bias master and bias correction is likely only needed for VIS. NIR data will be taken in multi-end-point (MEP) sampling mode which actually is intrinsicly a pixel-wise bias subtraction for each raw frame.

4.4.2. Linearity calibration

Only NIR. The non-linearity correction is done by GEIRS (Generic Infrared Detector Software). It is consists of a "NIR First Stage Pipeline" (`http://www.mpia-hd.mpg.de/~mathar/public/CARMENES-AIV04B-NIR-DCS-MAN02.pdf`) which pre-processes the intermediate frames to a single image. The correction uses laboratory calibration data taken at MPIA with the detectors fully illuminated and with sample up the ramp (SSR) mode.

4.4.3. Master flat

Input: $n_{\rm flat} \geq 11$ flat frames, tag: SOURCE == 'FLAT,FLAT'

Description: Creation of a mean master flat from at least 11?? flat observations

Output: Master flat (car-$YYYYMMDD$Thhhmmmsss-cali-vis_mflat.fits)
Mean signal-to-noise (fits header)

QC check: Error for too low or high flux levels

4.4.4. Order location

Input: $n_{\mathrm{odef}} = 1$ order definition frames, tag: SOURCE == ['FLAT,OFF' | 'OFF,FLAT'] && OBJ=='MFLAT'

Description: Definition of the order center and order width to be extracted

Output: Order location files (car-$YYYYMMDD$Thhhmmmsss-cali-vis_ord_A.sav) containing the polynomial coefficients of the order trace
Aperture map (car-$YYYYMMDD$Thhhmmmsss-cali-vis_ord_A.fits.gz)

QC check: number of identified orders $n_{\mathrm{ord}} = \{53 : \mathrm{vis\_AB}, 29 : \mathrm{nir\_AB}\}$

4.4.5. Normalised flat

Input: Flat frame tag: SOURCE == 'FLAT,FLAT'

Description: Creation of a normalised flat from the master flat (Note: this product is currently obsolete.)

Output: Normalised master flat (car-$YYYYMMDD$Thhhmmmsss-cali-vis_nflat.fits)
Blaze function (car-$YYYYMMDD$Thhhmmmsss-cali-vis_blaze.fits)
Mean signal-to-noise (fits header)

QC check:

4.4.6. Wave

Input: Wavelength calibration frames, tag: SOURCE == ['WAVE,NONE' | 'NONE,WAVE']

Description: Creation of a nightly wavelength solution for fibre A and B

Output: Linear extracted spectra (car-$YYYYMMDD$Thhhmmmsss-cali-vis_wave_A.fits)

QC check:

Note: The wavelength calibration frames are currently linearily extracted. Due to the image slicers and fibre shape+orientation, optimal extraction is not suited. Sharp features, like emission lines, have different cross-sections compared to the flat field.

4.4.7. Wavelength calibration

Note: Wavelength calibration is an interactive task which is not executed by the automatic pipeline (see wavelength re-calibration instead).

Description: Creation of a start guess wavelength solution for fibre A and B.

Wavelength calibration is based on the IDL REDUCE package.

4.4.8. Wavelength re-calibration

Input: Wavelength calibration frames, tag: SOURCE == ['WAVE,NONE' | 'NONE,WAVE']
Guess solution specifying the location of the calibration lines (car-$YYYYMMDD$Thhhmmmsss-cali-vis_wave_A_2D.sav)

Description: Creation of a nightly wavelength solution and wavemap for fibre A and B

Output: Fit parameters (polynomial coefficients) and statistics of the wavelength solution
A wavemap (wavelength for each extracted pixel)

QC check: Number of identified lines > ??
Number of rejected (saturated, off-ccd, clipped) lines < ??
RMS around the wavelength solution < ?? m/s
Median line width
Median spectral resolution

Wavelength re-calibration loads the guess solution. The most important variable is cs_lines ("calibration set lines") which is a structure containing information about line positions and wavelengths of selected calibration lines (Table 4.5). First, the positions x_k of the calibration lines λ_k are re-measured in the order o_k of the extracted spectrum. (Currently, Gaussian profiles are used to determine the line centers, but the real line spread function will have features of a top hat due to the slicer.) Then

Table 4.5.: The structure of the variable cs_lines.

field	type	unit	comment
WLC	double	Å	"computed" wavelength $\lambda(x_i, o_i)$
WLL	double	Å	laboratory wavelength λ_i
POSC	double	pix	computed line position $x(\lambda_i, o_i)$
POSM	double	pix	measured line position x_i
POSME	double	pix	error estimate of POSM $\sigma(x_i)$
POSMY	double	pix	vertical position y_i of the line
XFIRST	int	pix	start of line window
XLAST	int	pix	end of line window
APPROX	string		fit type ('G': gaussian)
WIDTH	double		fit parameter width
HEIGHT	double		fit parameter amplitude
ORDER	int		relative echelle order m
ABSO	int		absolute echelle order o_m
FLAG	int		line flag (0: ok, 1: saturated, 2: offccd, 4: clipped)

a wavelength solution $x(\lambda, o)$ is fitted to the measured position x_k with direct regression. A 2D polynomial is used as model $x(\lambda, o) = \mathrm{poly}(o\lambda, o)$. See Bauer et al. (2015, submit.) for more details about the concept for wavelength calibration.

In the parameter file, the regression type for the wavelength calibration can be specified:

- 0 – reverse regression
- 1 – direct regression
- 2 – direct regression with fixed discountinueties/gaps
- 3 – direct regression with discountinueties/gaps

The default is 2 for CARM_VIS and 3 for CARM_NIR.

The two NIR detectors, TOM (SCA2) and JERRY (SCA1), are "glued" together before the wavelength calibration. The gap is treated as a virtual pixel having a size of about $a = 140$ pixels (re-indexing $X_{\mathrm{TOM}} = 1...2048$, $X_{\mathrm{gap}} = 2049$, $X_{\mathrm{JERRY}} = 2050...4097$). The gap size a can be a free parameter (length changes might be noticable?). The intention of glueing with a virtual pixel is to have less parameters in the wavelength solution and to reduce the number of borders where the polynomials are only poorly constrained, i.e. to make the solution more robust.

Figure 4.1 gives an example of a QC plot from wavelength re-calibration.

The calibration shall be done with vacuum wavelength.

4.4.9. Science data extraction

Input: all raw science frames, tag: 'SCI'

Description: read last calibrations (master bias, master flat, order location, wavelength solution)
subtraction of master bias/dark
(linearity correction)
straylight modelling and subtraction
extraction (with master flat) from 2D to 1D
wavemap assignment
barycentric correction
drift measurement in reference fibre B
correction for instrument response and flat lamp spectrum
order merging

Output: Extracted spectra (car-$YYYYMMDD$T$hhmmsss$-gtoc-vis_x2d_A.fits). The fits file has the extentions:
[0]: propagated primary header
[1], [SPEC]: spectrum (flat relative flux values)
[2], [CONT]: continuum (smooth correction for flat spectrum)
[3], [SIG]: error estimates for SPEC
[4], [WAVE]: wavelength for each pixel [Å]
The fits files can be converted to ascii format with the (python) script ech2dat.

In science data extraction, the products of the calibration are applied to the science exposures. The calibration files are located (closest valid set before the exposure) and applied to the frames. The names of the calibration files are added to the fits header of the reduced files (content of calibration file header is not propagated).

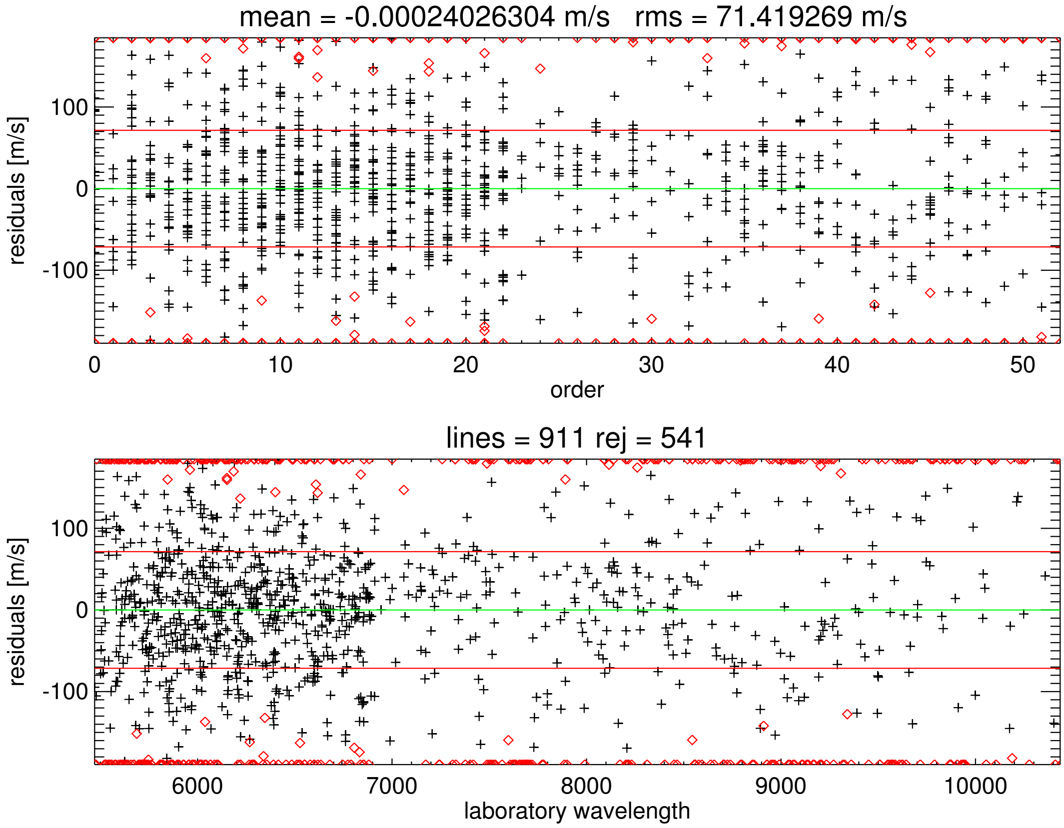

Figure 4.1.: A QC plot (postscript version) of wavelength re-calibration (waverecal) for VIS_A.

Straylight modelling: A two-dimensional B-spline is fitted to the region masked as background (complement of the aperture mask from the order tracing of both fibres). The fit smoothes the background globally (not order wise) and iteratively (up to 7 iterations) to reject e.g. cosmics. The fit is then evaluated over the full image (i.e. also in the aperture mask) and subtracted.

Extraction: Flat-relative optimal extraction is performed (Zechmeister et al., 2014). Figure 4.2 show an extracted and wavelength calibrated CSFS spectrum. The spectrum was converted to ascii using

```
> ech2dat csfs.2015-04-17T14:07:32-nir-phofab_A.fits
```

and plotted with the gnuplot command

```
gnuplot> plot 'csfs.2015-04-17T14:07:32-nir-phofab_A.dat' i 0::2 w l, '' i 1::2 w l lt 3
```

Barycentric correction: Currently, the BarCor code of Hrudková (2009) is applied which is precise to about 50 cm/s. It is forseen to replace it with code of Wright & Eastman (2014) with a precision statement of 1 cm/s written in IDL[1].

4.4.10. Flux calibration

Input: extracted science spectra, tag: SOURCE == ['FLUX,SKY']

Description: Comparison of extracted spectrum with a model to derive a wavelength depend efficiency (response) function and to remove the flat lamp spectrum

Output: Efficiency function $\eta(\lambda)$ (B-spline coefficients stored in car-$YYYYMMDD$T$hhh$$mm$m$sss$-cali-vis_eff_coeffs.sav)
Efficiency map for order and pixel $\eta_{i,o} = \eta(\lambda(i, o))$

Flat-relative extraction removes ideally the blaze, pixel-to-pixel sensitivity and fringing effects. I.e. extracted science spectra are corrected for all efficiency effects of the spectrograph. Yet the stellar spectrum is relative to the flat lamp spectrum and by observing another absolute standard (B star) it can be calibrated. This also needs a modelling/knowledge of the atmospheric extinction for the reference star (and later as well as for the science object) at the time of observation which is yet not implemented.

[1] For SERVAL and in general a Python version would be desirable.

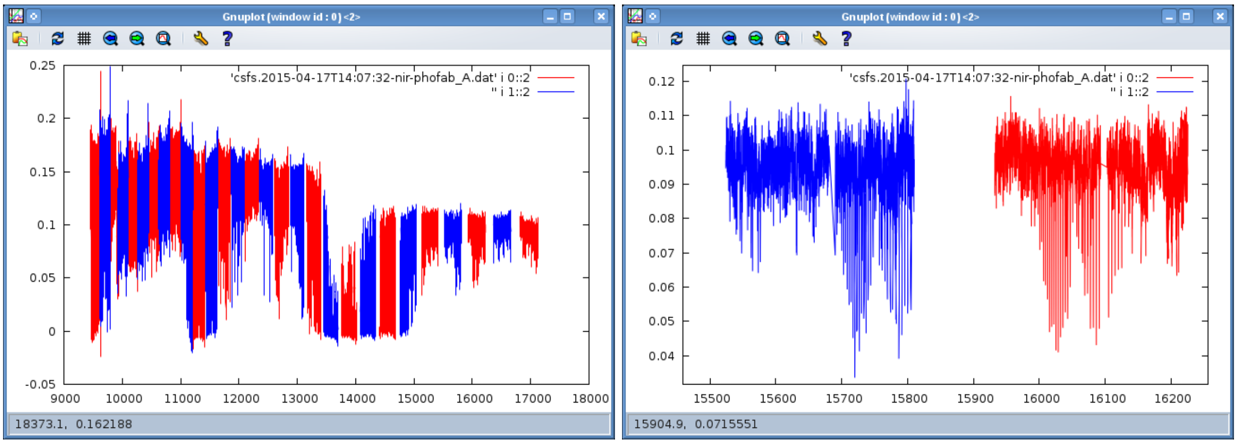

Figure 4.2.: Extracted CSFS NIR spectrum. Left: full view (odd and even orders are color-coded). Right: Zoom-in to the fourth and third last orders showing the order gap (missing overlap) and the mosaic detector gap.

When science spectra are processed, the efficiency map is restored and included as the CONT extension to the fits files (in REDUCE the CONT extension contains the flat spectrum/blaze function).

4.5. xpipe - pipeline frontend GUI

The frontend is a tool developed for convenience. It is a graphical user interface (GUI, Fig. 4.3). xpipe is programmed with Tcl/Tk and allows to:

- start and stop the pipeline
- browse raw frames and product files
- preview fits header
- preview raw spectra (ds9) and reduced spectra (ds9, gnuplot)
- setup path variable
- (forseen: setup extraction parameters)

When starting the CARACAL from xpipe, IDL will be called with the specified instrument setup. The graphical user interface (Fig. 4.3) consists of fields showing path names and progid/nite. Below, the files found in the directories are displayed like in a browser. Furthermore, there are the buttons:

'reduce' manual reduction

'pipestart' start the automatic trigger of the pipeline

'pipeexit' stops and exits the trigger

'update' manual re-read the directories (note: some automatic update is provided)

'fits header' displays the fits header of the marked file in a separate window

'exit' exits xpipe

The menubar is structured as follow:

▷ **Inst** selects the instruments
  ▷ **CARM_VIS**
  ▷ **CARM_NIR**

▷ **View** toggle the view of the list boxes
  ▷ **prod**
  ▷ **raw**
  ▷ **red**
  ▷ **all dirs**
  ▷ **path**

▷ **Preview**

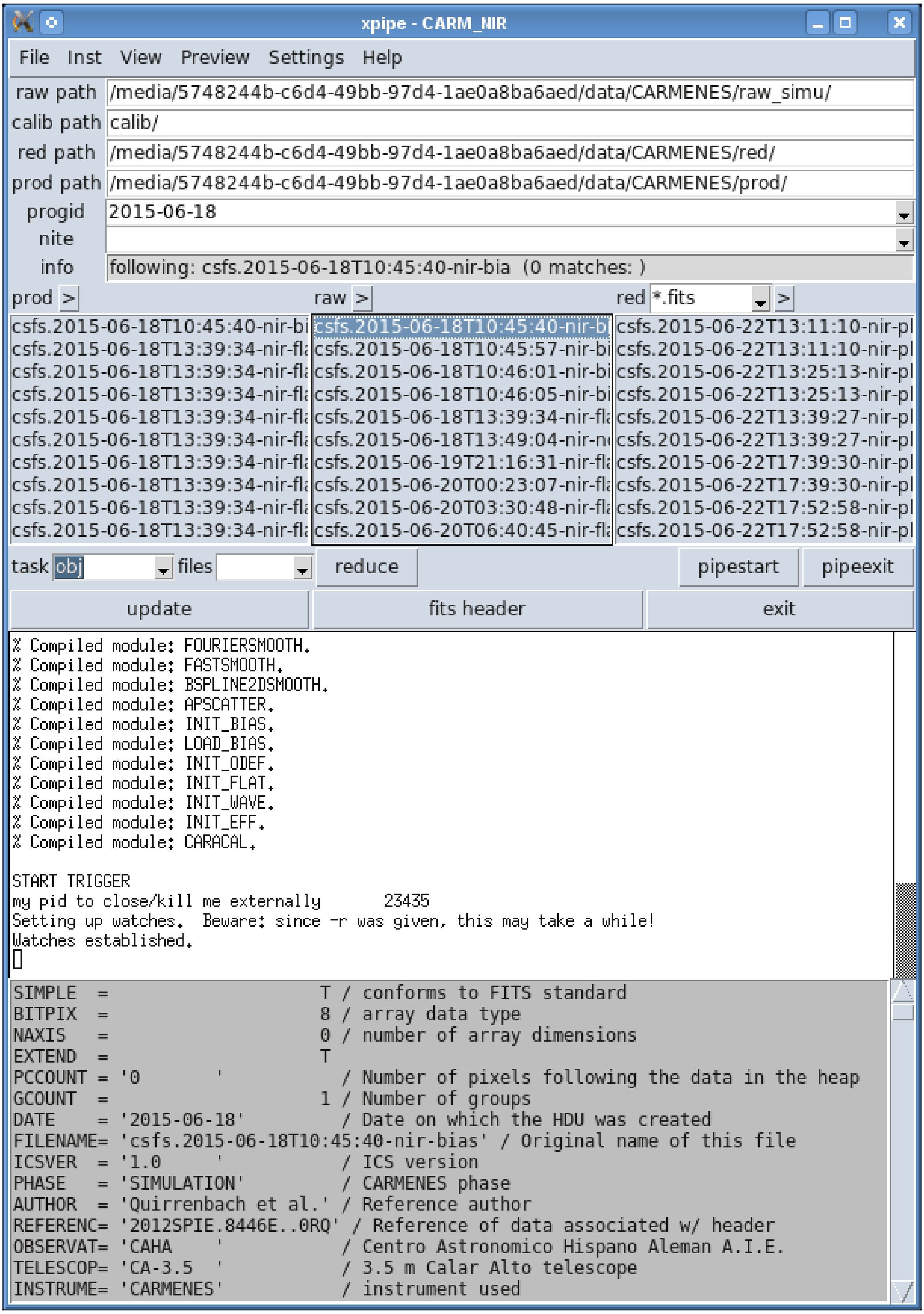

Figure 4.3.: xpipe - CARACAL pipeline frontend.

▷ **state** if checked, the processing status is color coded (black - unprocessed; blue - new frame; green - processed raw frames; red - failed (forseen)).
▷ **OBJECT** if checked, the OBJECT keyword of the fits header is displayed for the raw frames
▷ **header** if checked, the fits header is previewed in an embedded text field panel (bottom panel of Fig. 4.3)
▷ **ds9** if checked, then each single click displays the file with ds9
▷ **gps** if checked, then clicking on a reduced spectrum displays it with gnuplot (experimental)

▷ **Settings**
▷ **watch** if checked, the file listing is updated when new files are in the folder (experimental, to replace manual update)
▷ **follow** if checked, clicking on a raw frame finds the corresponding reduced spectra (blue selected files) and vice versa
▷ **verbose info** print the info text displayed in info field also to stdout
▷ **Preferences...**

▷ **Help**
▷ **Info**
▷ **CARACAL manual (pdf)** shows this manual with a pdf viewer
▷ **About**

A right mouse click on a file opens a context menu with the following items:

▷ **get filename** copies the absolute filename to clipboard (can be inserted with Shift-Ins/Ctrl-Shift-v)

▷ **ds9** display a fits file with ds9 (newframe)

▷ **gps** display a fits spectrum with the gnuplot

▷ **ech2dat** convert the echelle spectrum (fits file) to ascii (*.fits -> *.dat) using ech2dat.py

▷ **header** get the header of a fits file (separate window)

▷ **terminal** opens a new konsole (directory of selected file)

▷ **reduce** re-reduce a nite or a selected raw frame with the selected tasks

▷ **delete** remove file from disk (pipedev version only)

5. SERVAL - RV pipeline

5.1. General overview and statement

The first question for RV computations is: What to use as template? It should be understood that the answer depends on whether precision or accuracy is required. Of cause we want both (precise RVs for planet search, absolute RVs for equivalent width measurement, etc.), but it seems not possible to get both simultaneously. This situation is actually also reflected by the different algorithms available in literature, e.g. cross correlation function with binary mask (Pepe et al., 2002), least square fit with template matching (Anglada-Escudé & Butler, 2012), or in case of the iodine cell with additional modelling of the instrumental profile (Butler et al., 1996). Therefore SERVAL (Spectrum Radial Velocity AnaLyser) has a number input options and modes to target different situations and aims.

1. The main mode of SERVAL is high precision RV (PRV). The template is established from the observations itself by a "shift and co-add" algorithm. This template will be the best match to the observations (regarding spectral type, instrumental and rotational line broading). Thus the RV content of the spectrum is properly weighted promising highest precision (provided that the assumptions of constant instrumental and stellar line profile are met). However, this mode is fully differential (i.e. no absolute RV). Also it is not well suited for realtime pipeline at the telescope since it requires archive access to reduced spectra and individual templates. A number of observations should be available to obtain a well sampled, high signal-to-noise template. The PRV mode is default in SERVAL. It fully automatised (but not pipelined) and should be executed by the scientist for post-processing to obtain RVs with publication quality. The details of the SERVAL algorithm are described in Appendix A.8.

2. To obtain absolute stellar RVs, SERVAL has an option to specify a reference template. This mode is only poorly tested and under development. A library of the reference templates must be created which might be a PHOENIX spectrum (evt. concolved with instrumental profile) or observed spectra of some reference stars (at commissioning). For the stars to be observed, a desired reference templates (or spectral types) must be specified in advance (via fits keyword SpT). Due to the template mismatch a lower precision is expected. When implemented and pipelined, CARACAL will call automatically SERVAL to get RVs when the pipeline runs at the telescope, i.e. normal users do not have to execute this mode manually. Eventually, the method will be replaced/complemented by a CCF routine with binary masks to obtain a CCF which would be helpful for further analysis of the mean line shapes (bisector span (BIS), full width half maximum (FWHM), contrast, $v \sin i$).

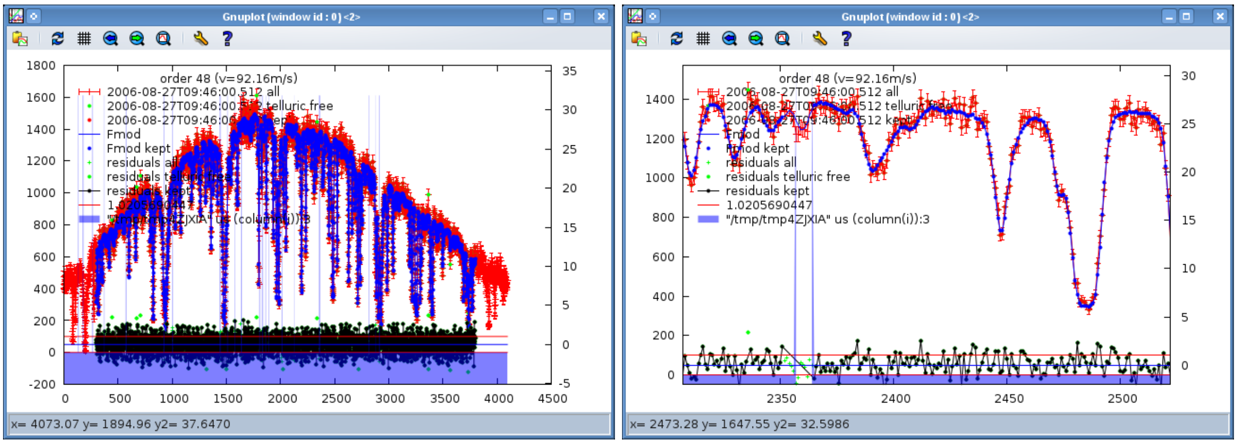

Figure 5.1.: Interactive plots provided by the SERVAL look option (real HARPS DRS spectrum). Data points are in red, model in blue (refering to left axis), error normalised residuals in black (refering to right axis) and telluric mask in transparent light blue.

3. Finally, there is a mode called drift. CARACAL will call SERVAL with this mode to compute the RV drift using the reference fibre B (ThNe, UNe, FP). As template the spectrum from the last valid wavelength calibration set will be used. This drift mode aims for high precision RV. Provided that the drift is small (small fraction of a pixel), a simplier algorithm than in the PRV mode can be applied. The drift is estimated by correlating (scaling) the gradient of the template to the difference between spectrum and template (Bouchy et al., 2001). The equations for the drift mode given in A.9. CARACAL stores the measured drift as a fits keyword.

5.2. Installation

SERVAL is packed in two tar-file (serval_vx.xx.tar containing the source code and serval_lib_2014-12-18.tar containing a library for Thorium and telluric masks and stellar PHOENIX templates) in the home directory.

SERVAL is located in ~/serval/ after unpacking. Two c files (cbspline.c and polyregression.c) are precomplied but can be recomplied (see file header). Additionally, the barycentric RV code BarCor must be installed (~/program/BarCor; installed already with the CARACAL package).

5.3. Usage

SERVAL is coded in Python and started from shell command line. The syntax is given in Appendix A.7. The main input for SERVAL is the directory name which contains all reduced spectra of the target (i.e. spectra must be sorted by target). An example for a call to SERVAL is

```
~/serval> serval gj699 /dir/to/spectra/of/gj699/ −coadd post3 −targ gj699 −snmin 10 −oset 40:
```

The working directory is ~/serval, while the main source file is located in a sub-directory (~/serval/src/serval.py). The name of the output directory (here gj699, same as the object name) and the path to the reduced spectra must be specified. The next optional arguments tell SERVAL to coadd with method post3 (coadding with cubic B-spline regression), to look for the target gj699 in an ascii table (star.cat) for stellar coordinates (for barycentric correction) and proper motion (for correction of secular acceleration), to use only spectra with a minimum signal-to-noise of 10 and to compute the RVs only from (relative) order 40 to the last order.

Appendix A.7 shows the help output with all available options. Noteworthy are the look and debug options. The look options visualised the least square modelling with gnuplot (for each or selected orders, see Fig. 5.1). With the debug option, the programmed is stopped (and not exited) at the point where the program crashed.

The next example shows how CARACAL (getdrift.pro) calls the drift mode of SERVAL via the -driftref option

```
IDL> spawn, 'cd ~/serval'; ./src/serval.py drift file_b −driftref file_ref −inst CARM_NIR −fib B'
```

5.4. Output

In the specified output directory (~/serval/gj699 in the example above) contains the output of SERVAL, most important the RV file *obj*.dat and the template template.fits. A summary of the files and the content is given in Table 5.1.

Table 5.1.: Overview of SERVAL file output.

filename	comment	format
berv.txt	barycentric radial velocity	bjd berv drsbjd drsberv drsdrift timeid tmmean exptime
log.*obj*.dat	log file	plain text file
obj.chi	reduced χ^2_{red} for each order	bjd rchi[0:*]
obj.col.dat	chromatic RV index (color dependency)	bjd col colerr
obj.dat	RVs (wrt. coadded template)	bjd RV RVerr
obj.dfwhm	differential FWHM	bjd dFWHM dFWHMerr
obj.halpha	line index (requires absolute RVs)	bjd Hα Hαerr haleft halefterr harigh harigherr CaI CaIerr CaH CaHerr
obj.info.txt	infomation from fits file	timeid bjd berv sn55 obj exptime SpT|ccf.mask flag
obj.rvo.dat	RVs of each order	bjd RV RVerr RV_median RV_median_err RVo[0:*]
obj.rvo.daterr	RV errors of each order	bjd RV RVerr RV_median RV_median_err RVoerr[0:*]
obj.post.dat	post processed RVs	bjd RV RVerr
obj.pre.dat	“pre” RVs	bjd RV RVerr
obj.sa.dat	RVs corrected for secular acceleration	sbjd (RV-sa) RVerr sa
obj.snr	signal-to-noise ratio for each orders	bjd snr[0:*]
template.fits	last template	fits image: [0] primary header (keywords for coadd method, used spectra and RVs, template SNR for each order) [1] flux [2] wavelength
template_post3.fits	template from coadd method post3	same as for template.fits

The residuals of the modelling for each file are stored as fits file in res/ (currently deactivated, yet no option available).

The file cmdhistory.txt contains a history of all previous calls and the used command line options.

Bibliography

Anglada-Escudé, G. & Butler, R. P. 2012, ApJS, 200, 15 1, 5.1, A.8.1

Bauer, F. F., Zechmeister, M., & Reiners, A. 2015, submit., A&A 4.4.8

Bouchy, F., Pepe, F., & Queloz, D. 2001, A&A, 374, 733 3

Butler, R. P., Marcy, G. W., Williams, E., et al. 1996, PASP, 108, 500 5.1

Hrudková, M. 2009, Phd thesis 4.4.9

Pepe, F., Mayor, M., Galland, F., et al. 2002, A&A, 388, 632 5.1

Piskunov, N. E. & Valenti, J. A. 2002, A&A, 385, 1095 1

Quirrenbach, A., Amado, P. J., Seifert, W., et al. 2012, in Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, Vol. 8446, Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series 1

Wright, J. T. & Eastman, J. D. 2014, PASP, 126, 838 4.4.9

Zechmeister, M., Anglada-Escudé, G., & Reiners, A. 2014, A&A, 561, A59 4.4.9

A. Appendix

A.1. gps - gnuplot spectrum

gps (~/bin/gps) is a gnuplot script to directly plot CARACAL fits spectra, i.e. there is no need to export the fits file to ascii. This provides a fast access, however wavelengths (present in an extension) cannot be shown[1]. gps can be started from command line

[1]The experimental shortcut '$' toggles to a wavelength scale approximated from a simple echelle grating equation.

Key	Task	Key	Task
+	zoom-in	Left	pan left
–	zoom-out	Alt-Left	pan left by one page
Alt-+	fast zoom-in	Ctrl-Left	horizontal zoom-in
Alt-–	fast zoom-out	Ctrl-Alt-Left	fast horizontal zoom-in
Alt-u	reset ranges (unzoom all)	Home	scroll to first data point
U (Shift-u)	set ranges to data min/max	z	zero yaxis

Table A.1.: A selection of additional key bindings.

```
> gps csfs.2015-05-15T08:52:37-vis-photha_A.fits
```

or used from xpipe (context menu, right mouse click on file).

gps loads the ~/zoom.gnu script, which allowing for additional interactive panning and zooming via the arrow keys and additional short cuts. Table A.1.

A.2. ech2dat

ech2dat (~/bin/ech2dat) is a python script to convert CARACAL echelle spectra to ascii format. It can be started from command line

```
> ech2dat csfs.2015-05-15T08:52:37-vis-photha_A.fits
```

The name of the ascii file is the same as fits file, but with suffix .dat (instead of .fits). In xpipe ech2dat can be executed via the right click context menu.

The dat file contains four columns: wavelength, flux, flux_err and continuum. Each order begins with a comment line starting with a '#'. Moremore, the orders are separated by two newlines.

Tip: The two newlines are interpreted as block separators in gnuplot. I.e. you can select orders using the index option, e.g.

```
gnuplot> pl 'car-20150818T15h01m10s_a-gtoc-nir_P_A.dat' index 10:13
```

The block number you can get with column number -2. The pixel number (zero-based indexing) with column 0. So to plot flux against pixel and color-coded the order number, one can use:

```
gnuplot> pl 'car-20150818T15h01m10s_a-gtoc-nir_P_A.dat' us 0:2:-2 w l palette
```

A.3. imhead

imhead (~/bin/imhead) is a short bash+awk script to print the header of fits files. It can be started from command line

```
> imhead csfs.2015-05-15T08:*_A.fits
```

imhead accepts wildcards. imhead is also used by xpipe.

Two noteworthy options of imhead are -l (long format) and -k to search for fits keywords. Option -h outputs a little help.

A.4. Help output of CSFS

```
    ::::::::::   main start time: Thu Jun 25 15:32:46 2015   ::::::::::

usage: CARMENES Spectrum Forward Simulator [-h] [-v] [-g] [-p] [--maxsnr [DN]]
                                              [-ns NS] [-mc] [-sm {cdf,rej}]
                                              [-nr [...]]
                                              [--interp-flag INTERP_FLAG] [-a]
                                              [-b] [-A [...]] [-B [...]] [-rva]
                                              [-rvb] [-wta] [-wtb] [--wtlist]
                                              [-dwt] [-dw NW] [--oset [...]]
                                              [--gap] [--blaze] [--noise [DN]]
                                              [--tell [CH4 CO2 H2O N2O O2 O3 [CH4 CO2 H2O N2O O2 O3 ...]]]
                                              [--crop CROP] [--no-compress]
```

```
                                [--subwindows-ext] [--single-ext]
                                [--out output/filename/filename.fits]
                                [--full-output]
                                [--path /path/to/output/] [--ds9]
                                [--ds9-mosaic] [--reset] [--solve]
                                [--debug] [--preview]
                                {N,V} slit [slit ...]

Forward simulates input spectra into a modelled CARMENES
spectrograph, outputting a 2D raw image.

 positional arguments:
  {N,V}                Spectral arm: {'N': NIR, 'V': VIS}.
  slit                 Slit function/PSF/cross-section:
                       0 : 0-D point source at absolute slit image center,
                       1 : 2-D half-moon sampled from an even grid,
                       2 : 2-D half-moon sampled from random uniform grid,
                       3 : 2-D half-moon sampled from a Gaussian distribution
                           Gaussian depends on the following number of arguments [a,b,c,d] of type <float>:
                             0 args -> centered at (0,0), with standard deviation (width) of (1,1);
                             1 args -> centered at (0,0), width is argument (a, a);
                             2 args -> centered at (a,a), width is (b,b);
                             3 args -> centered at (a,b), width is (c,c);
                             4 args -> centered at (a,b), width is (c,d);
                           (see examples).

optional arguments:
  -h, --help           show this help message and exit
  -v, --version        show program's version number and exit
  -g, --grid           Grid sampling of ray wavelengths and slit positions.
                       (default: False)
  -mc, --monte-carlo Monte Carlo (MC) sampling of ray wavelengths and slit
                       positions. (default: False)

Options work ONLY if '-g, --grid' sampling is selected.:
  Grid sampling arguments.

  -p, --perturb        Perturb slit locations for a quasi-random simulation.
                       This helps to avoid aliasing artifacts in the
                       extraction of the image. Only works with '-g, --grid'
                       option. (default: False)
  --maxsnr [DN]        Maximum Signal-to-Noise Ratio of simulated image
                       (before noise). Renormalizes image to this value. Only
                       works with '-g, --grid' option. (default: 200.0)
  -ns NS               Number of samples to create the slit function grid before
                       the cross-section filter.
                       For slit=1, ns is defined along the fiber diameter.
                       The effective number of slit samples will be ~0.8*ns.
                       ns should be >= 60.

                       For slit=2, ns is defined along the fiber radius.
                       The number of effective slit samples will be ~Pi*ns^2. ns should be >= 10,000.
                       Only works with '-g, --grid' option.
                        (default: 30)
  -dw NW               Sampling of wavelength grid [Angstrom]. This defines
                       the sampling resolution of functional fiber inputs {C,
                       F U, W, X}. With the '-g, --grid' option, this also
                        regrids the input wavelengths to the specified
                        resolution. (default: 0.001)

Monte Carlo sampling arguments.:
  Options work ONLY if '-mc, --monte-carlo' sampling is selected

  -sm {cdf,rej}, --mc-method {cdf,rej}
                       "R|Method for MC wavelength sampling: cdf: Full Monte
                       Carlo sampling. Cumulative Distribution Function (CDF)
                       for wavelength sampling and Rejection Sampling for
                        slit sampling rej : Full Monte Carlo sampling.
                       Rejection Sampling for both wavelengths and slit. .
                       Only works with '-mc, --monte-carlo' option. (default:
                       cdf)
  -nr [ ...], --nrays [ ...]
                       Approximate number of rays to simulate.
                        If 2 numbers are specified, then the first input goes to Fiber A,
                       and the second goes to Fiber B.
                       Due to order overlapping of wavelengths, this parameter is
                       only a lower-limit approximation.
                       Only works with '-mc, --monte-carlo' option.
                        (default: 100000000)
  --interp-flag INTERP_FLAG
                        Interpolation flag for CDF function.
                       0 : do not interpolate
                       3 : cubic interpolation. (default: False)

Fiber arguments:
  -a                   Fiber A input:
```

```
                        B : bias frame,
                        C : laser comb calibration spectrum,
                        F : ideal flatfield exposure (completely normalized continuum spectrum),
                        L : line list ,
                            note: input parameter also required , eg.
                            '-ainput </path/to/ linelist >.txt'
                        P : example PHOENIX model M dwarf spectrum,
                            T_eff=3000 K, [Fe/H]=0.0, log(g)=5.0 (Husser et al., 2013)
                        S : input object spectrum,
                            note: input parameters also required , (see '-ainput') eg.
                            '-ainput </path/to/spectrum.fits>' or
                            '-ainput </path/to/wavelengths.fits> </path/to/flux.fits>'
                        T : Thorium Argon calibration lamp line list (Lovis 2007; Kerber 2008),
                        U : Uranium Neon calibration lamp spectrum (Sarmiento & Reiners 2013),
                        W : wavemap (pixel values are wavelength instead of counts),
                        X : Fabry-Perot calibration spectrum,
                        0 : none
                         (default: 0)
  -b                    Fiber B input. Similar to Fiber A. See '-a'.
                         (default: 0)
  -A [ ...], --ainput [ ...]
                        Input file(s) for Fiber A.
                        For fiber argument 'L': the expected parameter is of the form
                            '</path/to/ linelist >.txt'.
                        The file must be an ASCII-text file, with no header or footer, and 2 columns:
                            wavelengths [angstroms] and flux [ arbitrary units ].

                        For fiber argument 'S': the expected parameters must be one of the following:
                            1) '</path/to/spectrum>.fits' or
                            2) '</path/to/wavelengths>.fits </path/to/flux>.fits'.
                        The wavelengths fits-file must be in units of angstroms,
                        and the flux fits-file can be of any units ( arbitrary ).
                         (default: None)
  -B [ ...], --binput [ ...]
                        Input file(s) for Fiber B. See '-ainput'.
                         (default: None)
  -rva                  Radial velocity shift of Fiber A in [m/s].
                         (default: 0.0)
  -rvb                  Radial velocity shift of Fiber B in [m/s].
                         (default: 0.0)

Wavelength trace arguments:
  WARNING: Not currently supported!

  -wta                  Create a wavelength region file (.reg) for Fiber A.
                        The region file is a text file which includes the
                        detector-positions of wavelengths. When opened in a
                        FITS viewer with its associated FITS image, certain
                        wavelengths are labeled at their respective positions.
                        (default: False)
  -wtb                  Create a wavelength region file (.reg) for Fiber B.
                        See '-wta'. (default: False)
  --wtlist              Input file of wavelengths for wavelength trace region
                         file. Must be of the form
                        '</path/to/wavelengthlist>.txt'. The file is an ASCII
                        txt file, with 1 column (wavelengths [Angstrom]).
                        (default: None)
  -dwt                  Wavelength sampling [Angstrom] for wavelength trace
                        region file. (default: 2.0)

Spectral arguments:
  --oset [ ...]         Specify a list of orders to use in the simulation. The
                         list can consist of either all integers, or can be 3
                         integers as follows: (min, max-1, increment). The
                         latter is similar to the python/numpy range/arange
                        function. Eg., '-oset 59 60 61 62 63' will simulate
                        orders 59-63, whereas '-oset 59 80 ' will only use
                        orders 59, 62, 65, 68, 71, 74, 77 (80 is not included
                        because of the Python style). (default: None)
  --gap                 Include gap between the two NIR detectors [2.54 mm].
                         If spectral arm is VIS, this will have no effect and
                         default to False. (default: False)
  --blaze               Include echelle blaze function in computation.
                         (default: False)
  --noise [DN]          Include noise. Currently shot noise, readout noise,
                        and a bias of 5.0 [DN] are included. A value for the
                        readout noise can also be given as an argument.
                        (default: False)
  --tell [CH4 CO2 H2O N2O O2 O3 [CH4 CO2 H2O N2O O2 O3 ...]]
                        Include telluric lines computed using LBLRTM (Line-By-
                        Line Radiative Transfer Model, based on Clough et al.
                        2005) Independent species can also be given as an
                        argument. (default: False)
  --crop CROP           Number of pixels to crop on border of CCD. These
                        cropped pixels will only be affected by readout noise
                         if noise is selected, otherwise they will be set to 0.
```

```
                        Must be an integer value. Only applies to NIR arm.
                        (default: 0)

Output arguments:
  --no-compress         Do not compress output FITS file to .gz format.
                        (default: False)
  --subwindows-ext      Include creation of SUBWINDOWS extension. Does not
                        apply to VIS arm. (default: False)
  --single-ext          Suppress creation of TOM and JERRY extensions,
                        outputting to a single Primary HDU. Only applies to
                        NIR arm. (default: False)
  --out output/filename/filename.fits
                        Specify a name for FITS file (and region file if it
                        applies). The default format is
                        '<date><time><a_type><b_type>.fits.gz'. Output is
                        saved to a folder of this name in 'output/' unless
                        another path is specifed. If --no-full-output is
                        passed, then only a fits image will save to 'output/',
                        with no folder created. (default: None)
  --full-output         Outputs auxiliary files (wavelengths.dat &
                        slit_cross_section.pdf (default: False)
  --path /path/to/output/
                        Specify the output path. (default: output/)
  --ds9                 Open FITS output file (and associated region file) in
                        SAO-DS9 after simulation finishes. (default: False)
  --ds9-mosaic          Open FITS output file (and associated region file) in
                        SAO-DS9 after simulation finishes with --mosaicimage
                        iraf option. This will merge all extensions into one.
                        (default: False)

Other arguments:
  --reset               Delete and re-solve all existing offset data,
                        wavelength data, and wavelength solutions (equivalent
                        to "rm DATA/*"). NOTE: Usage was mostly for debugging
                        purposes, and should be deprecated in favor of 'make
                        clean' in the shell. (default: False)
  --solve               For the current run, re-solve all existing offset
                        data, wavelength data, and wavelength solutions
                        instead of importing data. Mostly used for debugging
                        purposes. (default: False)
  --debug               Debug mode. Used for checking the optical model - not
                        for debugging the program. CURRENTLY UNDER
                        CONSTRUCTION. (default: False)
  --preview             Preview fiber illumination and exit. Use only with
                        '-g/--grid' option. (default: False)

Example Usage
=============

Running CSFS from the command line must follow these basic guidelines:
>>> python csfs.py <spectral-arm> <slit-function> --<sampling-method> -a <a-fiber-type> -b <b-fiber-type>

One sampling method must be specified. For extraction purposes,
we recommend using:
    --monte-carlo
(-sm cdf is optional as it is actually the default setting for
--monte-carlo sampling)

Examples
--------

>>> python csfs.py N 1 -a S -b C -ainput spectrum.fits --grid
Half-moon 2d image, uniformly sampled. Fiber A is a given input spectrum
comprised of "spectrum.fits". Fiber B is a laser comb spectrum. Wavelengths
and fiber samples are on a grid.

>>> python csfs.py V 2 -a S -ainput wavelength.fits flux.fits -ds9 -mc -sm rej
Half-moon 2d image, uniform-randomly sampled. The spectrum is comprised of
"wavelength.fits" and "flux.fits". The FITS image will be opened in SAO-DS9
upon completion. Wavelengths are sampled using the spectrum as a Probability
 Distribution Function, using Rejection Sampling to pick random samples.

>>> python csfs.py V 2 -a P -b 0 -noise -tell -blaze -mc -sm cdf
Include noise telluric lines. Also includes the echelle blaze function in the
simulation. Wavelengths are sampled using the spectrum as a Probability
 Distribution Function, using its Cumulative Distribution Function (CDF) to pick
random samples.

Gaussian Fiber Examples
-----------------------
>>> python csfs.py V 3 -g -a P
This creates a slit cross section which has a randomly sampled Gaussian
 distribution centered at (0,0) and a width of 1 in both the x and y direction.

>>> python csfs.py V 3 0.5 -g -a P -g
The "0.5" argument given after the 3 indicates the width (standard deviation)
```

of the Gaussian in both the x and y direction. It should be a positive, real number.

>>> python csfs.py V 3 −0.5 0.7 −g −a P −g
The first argument here is the offset location for (x,y) in terms of the unit circle. This means that the center (mean) of the Gaussian will be at (−0.5, −0.5). The second argument 0.7 is the width in both x and y directions.

>>> python csfs.py V 3 −0.5 0.2 0.7 −g −a P −g
Here, the center of the distribution is the first two arguments, so that the central location is (x,y)=(−0.5, 0.2). The width in both x− and y− directions is 0.7.

>>> python csfs.py V 3 −0.5 0.2 0.7 0.3 −g −a P −g
The last of the possibilities is with 4 following arguments. The first two define the central location (x,y)=(−0.5, 0.2) The third, 0.7, defines the width in the x−direction. The fourth, 0.3, defines the width in the y−direction.

>>> python csfs.py V 3 −g −a P −−preview
Instead of fully running the simulation, Images of the slit cross section will pop up to show the slit image, and the program will exit.
The '−−preview' option has been included to help visualize and test different arguments.

A.5. CARACAL installing instruction (from howtodo.txt)

```
#################################
##### CARACAL version v0.05 #######
#################################

# 2014−06−12 (changes for v0.02)
# trigger implemented
# db_read.pro separated from db_find.pro
# thar database included
# new wavelength solution (F. Bauer)


# 2015−02−12 (changes for v0.03)
# name and file re−organisation
#    folder idl/caracal added
#    rename reduce_carm to caracal
#    instmode carm to carm_vis
#    instmode carm_nir added
# wavelength solution (F. Bauer)
#    now mode: x(l,o)
# xpipe GUI included
# draft of CARACAL manual included
# new: cs_lines2reg.pro
# changed: saturated.pro (=> saturated.pro + lines2reg.pro)
# new: qcplot_wave
# changed: fitwls2D.pro


# 2015−05−19 (changes for v0.04)
# NIR gap added
#     wavesol tom and jerry added
#     hamdord workaround (if the order tracing is done with gap for jerry too few points the reddest orders have distance from the border: default )
# Trigger: mode watchdir added
# JD included in classification
# QC for the tasks revised
# naming scheme for calibration products now with time−id
# DB stores now absolute file IDs (instead of dir names)
# wavesol2reg.pro removed (replaced by cs_lines2reg)
# new: getdrift.pro


# 2015−06−23 (changes for v0.05)
# xpipe
# new: root version
# new: save preferences
# new: ds9 with mosaic mode for NIR (experimental)
# new: conditional context menu
# CARACAL
# new: gluecal option (NIR products now with extensions instead of separate files)
# new: mosaic mode (display NIR with ds9 mosaic mode)

# TODO list for CARACAL
# − flat: bias subtraction before coadding?
# − coadd more than 10 flat (too many units open problem)
# − check merge
# − classify in inst mode?
# − check gain propagation through code and headers
# − modeinfo: read NIR gain from primary header instead of ext−header
```

```
# - drift based on tag WAVE
# - JD tag as double
# - dark correction
# - non-linearity correction
# - FP task
# - check unnecessary vactoair-airtovac conversion (needed?)
# - SERVAL: no fixed working directory
# - barycentric correction with Wright & Eastman (2014)
# - perfect extraction


# explanation for some files
.ds9.ans            # some short cut for ds9
bin/ds9pan          # panning with arrow keys in ds9
bin/ds9zoom         # zooming with arrow keys in ds9
bin/gps             # gnuplot script to view directly the extracted fits files
bin/ech2dat         # converting the fits file to ascii
idl/idlstartup.pro  # IDL startup file (to be customised)


# Instructions for installing the pipeline

# backup your ~/idl/idlstartup.pro. It will be overwritten and must edited and backuped!!!
# unpack the pipeline in the home directory
cp ~/idl/idlstartup.pro ~/idl/idlstartup.pro_$(date +"%Y-%m-%dT%Hh%Mm%S")

mkdir ~/caracal
cd ~/caracal
wget http://www.astro.physik.uni-goettingen.de/~zechmeister/carmenes/caracal_v0.05.tar
tar -xf caracal_v0.05.tar -C ~/

# if reinstall
# check ds9 path in ~/bin/xds9

##################################
### include this to ~/.bashrc ###
##################################
 alias kate="kate --use"
 alias ds9='xds9'
 alias gp='gnuplot'

export IDL_STARTUP=~/idl/idlstartup.pro
export PATH=$PATH:$HOME/programs/fv5.3/xpabin/
export PATH=$PATH:$HOME/bin

_complete() { eval '_sub'$2'() { shopt -s nullglob; COMPREPLY=( ${COMP_WORDS[COMP_CWORD]}*.'$1' ); }'; shift; eval 'complete -o default -o
_complete "{fits,fits.gz}" ds9 xds9
_complete "{pdf,ps}" okular ok
_complete pdf acroread
_complete "{gnu,plt}" gnuplot gp
_complete "{ps,eps}" epstopdf

#############################################
### include this to ~/idl/idlstartup.pro ###
#############################################
; in .bashrc: export IDL_STARTUP=~/idl/idlstartup.pro
defsysv,'!TEXTUNIT',0      ; for goddard catalog database reading
defsysv,'!TEXTOUT',1       ; for terminal with "MORE"
defsysv,'!PRIV',0
set_plot,'x'

.run ~/idl/pro/addpath
.compile '~/idl/pro/setcolors.pro', stretch
.compile gaussfit

addpath, '~/idl/pro'
addpath, '~/idl/mpfit'
addpath, '+~/idl/astrolib/pro'

addpath, '~/idl/coyote/'
addpath, '+~/idl/bauer/fp_ext'
addpath, '+~/idl/zechmeister'
addpath, '+~/idl/caracal'
addpath, '+~/idl/reduce'

device, true_color=16
device, decomposed=0, retain=2

window,0,xsize=500,ysize=500 ;& window,1,xsize=500,ysize=500 & wset,0

arv=findgen(32) * (!pi*2/32.)
usersym, cos(arv), sin(arv),/fill       ; circle plot sym #8

!x.style=1
!y.style=1
!p.charsize=1.3
```

```
;! p.psym=10

;To use colors in plots use
 setcolors, /SYSTEM_VARIABLES

;#fl run_in_fl=1
IF ~keyword_set(run_in_fl) THEN BEGIN &$
device, cursor_standard=46 &$
DEFINE_KEY, 'HOME', /START_OF_LINE &$
DEFINE_KEY, 'END', /END_OF_LINE&$
endif

########################################

# REDUCE
# The original package is here: http://www.astro.uu.se/~piskunov/RESEARCH/REDUCE/reduce.tar.gz
# A number of scripts were modified. Therefore it comes the package comes with caracal_v0.05.tar.
# See folder ~/idl/reduce/*

# Install third party packages
# use bash
```

A.6. CARACAL fitsheader

```
SIMPLE =                    T /Primary Header created by MWRFITS v1.12
BITPIX =                    8 /
NAXIS  =                    0 /
EXTEND =                    T /
PCCOUNT = '0     '            / Number of pixels following the data in the heap
GCOUNT =                    1 / Number of groups
DATE   = '2015-05-15'         / Date on which the HDU was created
FILENAME= 'csfs.2015-05-15T08:52:37-vis-photha' / Original name of this file
ICSVER = '1.0     '           / ICS version
PHASE = 'SIMULATION'          / CARMENES phase
AUTHOR = 'Quirrenbach et al.' / Reference author
REFERENC= '2012SPIE.8446E..0RQ' / Reference of data associated w/ header
OBSERVAT= 'CAHA '             / Centro Astronomico Hispano Aleman A.I.E.
TELESCOP= 'CA-3.5 '           / 3.5 m Calar Alto telescope
INSTRUME= 'CARMENES'          / instrument used
TIMETYPE= 'Guaranteed'        / Guaranteed or open time
PROG-NUM= 'CARMENES'          / Calar Alto programme identification
PROG-PI = 'CARMENES'          / Programme principal investigator identification
OBSERVER= 'Service '          / Service or visiting observer name
OPERATOR= '' / Telescope operator
OBJECT = 'PHOE_lte03000-5.00-0.0' / Name for the object observed
ORIGIN = 'CARMENES'           / Organization responsible for creating FITS file
OBS-MODE= 'NIRmaster+VIS' / Observation mode
RA     =                  0.0 / 00:00:00.00 Requested RA (J2000) telescope coor
DEC    =                  0.0 / +00:00:00.00 Requested DEC (J2000) telescope co
EQUINOX =              2000.0 / standard FK5 (years)
RADECSYS= 'FK5 '              / coordinate reference frame
SPECTRNG= '520-1700 nm (VIS+NIR)' / Spectral range
SPECTRES= '82000 '            / Spectral resolution Delta lambda / lambda
APERTURE= '1.5  '             / Aperture of fibres on sky in arcsec
CATG   = 'SCI     '           / Observation category
TECH   = 'ECHELLE '           / Technical type
TYPE   = 'STAR,WAVE'          / Observation type in fibA and fibB
SOURCE = 'OBJ,ThAr'           / Observation source in fibA and fibB
EXPTIME =                 0.0 / Exposition time
RO_TIME =                 0.0 / Read-out time
SEQNUM =                    0 / Exposition number
EXPNUM =                    0 / Number of exposures
ARCVER = '' / ARC version
DATE-OBS= '2015-05-15T08:52:37' / date (yyyy-mm-dd) of observation
MJD-OBS = 57157.28653935203 / Modified Julian Date at start of observation
OBSEPOCH= '' / Observation epoch
EFFTIME = '2015-05-15T08:52:37' / Effective time
AIRMASS =                 0.0 / Airmass
AMSTART =                 0.0 / Airmass at start
AMEND =                   0.0 / Airmass end
HIERARCH CAHA GEN AMBI WIND SPEED = 0.0 / [m/s] Wind speed
HIERARCH CAHA GEN AMBI WIND DIR = 0.0 / [deg] Wind direction
HIERARCH CAHA GEN AMBI RHUM = 0.0 / [%] Relative humidity
HIERARCH CAHA GEN AMBI PRES = 0.0 / [1e2 Pa] Air pressure
HIERARCH CAHA GEN AMBI TEMP = 0.0 / [degC] Air temperature
HIERARCH CAHA TEL GEOELEV = 2168 / Elevation above sea [m]
HIERARCH CAHA TEL GEOLAT = 37.2236 / Geographic latitude [deg]
HIERARCH CAHA TEL GEOLON = -2.5463 / Geographic longitude [deg]
HIERARCH CAHA TEL DOME.AZ = 147.0 / Dome azumut [deg]
HIERARCH CAHA TEL DOME.EL_LOW_E = 70.0 / Dome door low limit
HIERARCH CAHA TEL DOME.EL_UPP_E = 93.0 / Dome door upper limit
HIERARCH CAHA TEL FOCU ID = 'CASS ' / Instrument focal position
```

HIERARCH CAHA TEL MIRR S1 COLLAREA = 2.94 / Effective collecting area of primary
HIERARCH CAHA TEL FOCU F_RATIO = 8.0 / Telescope F−ratio
HIERARCH CAHA TEL FOCU LEN = 17.61 / Telescope focal length [m]
HIERARCH CAHA TEL FOCU SCALE = 0.08260000000000001 / Telescope focus scale at in
HIERARCH CAHA TEL FOCU VALUE = 23.06 / Focus relative setting
HIERARCH CAHA TEL POS SET RA = 388065.0 / 07:11:11:0 RA preset [sec]
HIERARCH CAHA TEL POS SET DEC = 43.499167 / +43:29:57.0 DEC preset [sec]
HIERARCH CAHA TEL POS SET EQUINOX = 2000.0 / Equinox at present
HIERARCH CAHA TEL POS AZ_START = 0.0 / Telescope azimuth at start of observation
HIERARCH CAHA TEL POS EL_START = 0.0 / Telescope elevation at start of observati
HIERARCH CAHA TEL POS HA_START = 6.662358 / 00:26:39.0 HA [deg]
HIERARCH CAHA TEL SLATEL = '' / Telescope name known to SLALIB
HIERARCH CAHA INS FRONTEND PICKMIRR = '' / Pick−up mirror position
HIERARCH CAHA INS FRONTEND ADCANG1 = 0.0 / Rotating angle of first ADC prism [de
HIERARCH CAHA INS FRONTEND ADCANG2 = 0.0 / Rotating angle of second ADC prism [d
HIERARCH CAHA INS FRONTEND ACGFOCUS = 0.0 / Acquisition & guiding camera focus [
HIERARCH CAHA INS FRONTEND NIRMIRROR = 'A+B ' / Position of calibration pick
HIERARCH CAHA INS FRONTEND VISMIRROR = '' / Position of calibration pickup mirro
HIERARCH CAHA INS FRONTEND REFLECT = 'Beamsplitter' / Beamsplitter or Mirror in
HIERARCH CAHA INS ACQGUIDE GUIDING = '' / Acquisition and guide computer status
HIERARCH CAHA INS ACQGUIDE NIMAGES = 1 / Number of images taken by A&G system du
HIERARCH CAHA CRYO PRESS1 = 0.0 / Pressure in sensor 1 [mbar]
HIERARCH CAHA CRYO PRESS2 = 0.0 / Pressure in sensor 2 [mbar]
HIERARCH CAHA CRYO TEMPMON1 = 0.0 / Filter box [K]
HIERARCH CAHA CRYO TEMPMON2 = 0.0 / Motor [K]
HIERARCH CAHA CRYO TEMPMON3 = 0.0 / Optics [K]
HIERARCH CAHA CRYO TEMPMON4 = 0.0 / Fan out [K]
HIERARCH CAHA CRYO TEMPMON5 = 0.0 / Detector plate [K]
HIERARCH CAHA CRYO TEMPMON6 = 0.0 / Cold plate [K]
HIERARCH CAHA CRYO TEMPMON7 = 0.0 / Inner shield [K]
HIERARCH CAHA CRYO TEMPMON8 = 0.0 / Outer shield [K]
HIERARCH CAHA DET BIASEC = '[2049:2070,1:2048]' / Bias section. Only for VIS
HIERARCH CAHA DET TRIMSEC = '4:2045,4:2045' / Effective section for cutting the
HIERARCH CAHA DET CAMERA = '' / Camera name
HIERARCH CAHA DET PIXSCALE = 0.0 / arcsec/pixel
HIERARCH CAHA DET ENOISE = 0.0 / Electrons/read
HIERARCH CAHA DET ELECTRON = ''
HIERARCH CAHA DET FILTER = '' / Filter macro name of filter combinations
HIERARCH CAHA DET TPLNAME = '' / Macro/template name
HIERARCH CAHA DET TIMER0 = '' / milliseconds
HIERARCH CAHA DET TIMER1 = 0 / milliseconds
HIERARCH CAHA DET TIMER2 = 0 / microseconds
HIERARCH CAHA DET PTIME = 0 / Pixel−time (units)
HIERARCH CAHA DET READMODE = '' / Read cycle−type
HIERARCH CAHA DET IDLEMODE = '' / Idle to read transition
HIERARCH CAHA DET SAVEMODE = 0 / Cave cycle−type
HIERARCH CAHA DET CPAR1 = '' / Cycle type parameter
HIERARCH CAHA DET ITIME = 0.0 / (on chip) integration time [s]
HIERARCH CAHA DET HCOADDS = 0 / Number of hardware coadds
HIERARCH CAHA DET PCOADDS = 0 / Number of coadded plateus/periods
HIERARCH CAHA DET SCOADDS = 0 / # of software coadds
HIERARCH CAHA DET NCOADDS = 0 / Effective coadds (total)
HIERARCH CAHA DET EXPTIME = 0.0 / Total integration time [s]
HIERARCH CAHA DET FRAMENUM = 0
HIERARCH CAHA DET SKYFRAME = ''
HIERARCH CAHA DET SAVEAREA = ''
HIERARCH CAHA DET SOFTWARE = '' / Software version
HIERARCH CAHA DET CCDNAME = '' / Name of CCD detector
HIERARCH CAHA DET CCDPSIZ = 15.0 / Pixel size [mum]
HIERARCH CAHA DET CCDORI = 0.0 / CCD orientation
HIERARCH CAHA DET CCDSPEED = 0.0 / CCD readout speed
HIERARCH CAHA DET CCDBINX = 1.0 / Binning factor along X axis
HIERARCH CAHA DET CCDBINY = 1.0 / Binning factor along Y axis
HIERARCH CAHA DET CCDGAIN = 1.0 / CCD gain e−/ADU
HIERARCH CAHA DET CCDMEAN = 300 / Bias of CCD at selected Gain (DN)
HIERARCH CAHA DET CCDSAT = 0 / Saturation of CCD (DN)
HIERARCH CAHA DET CCDRON = 5.0 / Readout noise of CCD at selected gain (e−)
HIERARCH CAHA DET CCDTEMP = 0.0 / [K] CCD temperature
HIERARCH CAHA DET DATASEC = '' / Image portion of frame
HIERARCH CAHA DET CCDSEC = '' / Orientation of full frame
HIERARCH CAHA DET ORIGSECX = 4096 / Original size full frame nx
HIERARCH CAHA DET ORIGSECY = 4096 / Original size full frame ny
HIERARCH CAHA INS CALUNIT SOCKET NUM1 = '' / Daily hollow−cathode lamp
HIERARCH CAHA INS CALUNIT SOCKET NUM2 = '' / First master hollow−cathode lamp
HIERARCH CAHA INS CALUNIT SOCKET NUM3 = '' / Second master hollow−cathode lamp
HIERARCH CAHA INS CALUNIT SOCKET NUM4 = '' / Third master hollow−cathode lamp
HIERARCH CAHA INS CALUNIT SOCKET NUM5 = '' / First super−master hollow−cathode l
HIERARCH CAHA INS CALUNIT SOCKET NUM6 = '' / Second super−master hollow−cathode
HIERARCH CAHA INS CALUNIT SOCKET NUM7 = '' / Third super−master hollow−cathode l
HIERARCH CAHA INS CALUNIT SOCKET HALOGEN = '' / Flat−field halogen lamp
HIERARCH CAHA INS CALUNIT SOCKET CURRENT1 = 0.0 / Daily hollow−cathode lamp curr
HIERARCH CAHA INS CALUNIT SOCKET CURRENT2 = 0.0 / First master hollow−cathode la
HIERARCH CAHA INS CALUNIT SOCKET CURRENT3 = 0.0 / Second master hollow−cathode l
HIERARCH CAHA INS CALUNIT SOCKET CURRENT4 = 0.0 / Third master hollow−cathode la
HIERARCH CAHA INS CALUNIT SOCKET CURRENT5 = 0.0 / First super−master hollow−cath
HIERARCH CAHA INS CALUNIT SOCKET CURRENT6 = 0.0 / Second super−master hollow−cat

HIERARCH CAHA INS CALUNIT SOCKET CURRENT7 = 0.0 / Third super−master hollow−cath
HIERARCH CAHA INS CALUNIT SOCKET AGE1 = 0 / Daily hollow−cathode lamp age [mA h]
HIERARCH CAHA INS CALUNIT SOCKET AGE2 = 0 / First master hollow−cathode lamp age
HIERARCH CAHA INS CALUNIT SOCKET AGE3 = 0 / Second master hollow−cathode lamp ag
HIERARCH CAHA INS CALUNIT SOCKET AGE4 = 0 / Third master hollow−cathode lampage
HIERARCH CAHA INS CALUNIT SOCKET AGE5 = 0 / First super−master hollow−cathode la
HIERARCH CAHA INS CALUNIT SOCKET AGE6 = 0 / Second super−master hollow−cathode l
HIERARCH CAHA INS CALUNIT SOCKET AGE7 = 0 / Third super−master hollow−cathode la
HIERARCH CAHA INS CALUNIT SOCKET LAMPNUM1 = '' / Daily hollow−cathode lamp code
HIERARCH CAHA INS CALUNIT SOCKET LAMPNUM2 = '' / First master hollow−cathode lam
HIERARCH CAHA INS CALUNIT SOCKET LAMPNUM3 = '' / Second master hollow−cathode la
HIERARCH CAHA INS CALUNIT SOCKET LAMPNUM4 = '' / Third master hollow−cathode lam
HIERARCH CAHA INS CALUNIT SOCKET LAMPNUM5 = '' / First super−master hollow−catho
HIERARCH CAHA INS CALUNIT SOCKET LAMPNUM6 = '' / Second super−master hollow−cath
HIERARCH CAHA INS CALUNIT SOCKET LAMPNUM7 = '' / Third super−master hollow−catho
HIERARCH CAHA INS CALUNIT SOCKET LAMPNUM8 = '' / Flat−field halogen lamp code
HIERARCH CAHA INS CALUNIT OCTAGON = 1 / Socket number selected by the mirror at
HIERARCH CAHA INS CALUNIT WHEEL−A = '' / Position of filter wheel of A fibre
HIERARCH CAHA INS CALUNIT WHEEL−B = '' / Position of filter wheel of B fibre
HIERARCH CAHA INS CALUNIT NOTCHFIL = '' / Notch filter
HIERARCH CAHA INS CALUNIT 3RD−SHUT = '' / Third−input shutter
HIERARCH CAHA INS CALUNIT LGHTSNSR = '' / Light sensor output
HIERARCH CAHA INS CHAMBER TEMP1 = 0.0
HIERARCH CAHA INS CHAMBER TEMP2 = 0.0
HIERARCH CAHA INS CHAMBER POXYGEN1 = 0.0 / Partial pressure of O2 (Master) [unit
HIERARCH CAHA INS CHAMBER POXYGEN2 = 0.0 / Partial pressure of O2 (Slave) [units
HIERARCH CAHA INS EXPMETER DETECTOR = 'ON '
HIERARCH CAHA INS EXPMETER STATUS = 'Online ' / Connected to ICS
HIERARCH CAHA INS EXPMETER WHEEL = 'Open ' / Position of the expmeter filter
HIERARCH CAHA INS EXPMETER FILE = '' / ExpMeter ascii filename
HIERARCH CAHA INS FIBRE SHAKER = 'ON ' / Optical fibre shaker
HIERARCH CAHA INS FIBRE PDU = 0
HIERARCH CAHA INS TANK TEMP1 = 0.0 / [K]
HIERARCH CAHA INS TANK TEMP2 = 0.0 / [K]
HIERARCH CAHA INS TANK TEMP3 = 0.0 / [K]
HIERARCH CAHA INS TANK TEMP4 = 0.0 / [K]
HIERARCH CAHA INS TANK TEMP5 = 0.0 / [K]
HIERARCH CAHA INS TANK TEMP6 = 0.0 / [K]
HIERARCH CAHA INS TANK TEMP7 = 0.0 / [K]
HIERARCH CAHA INS TANK TEMP8 = 0.0 / [K]
HIERARCH CAHA INS TANK TEMP9 = 0.0 / [K]
HIERARCH CAHA INS TANK TEMP10 = 0.0 / [K]
HIERARCH CAHA INS TANK TEMP11 = 0.0 / [K]
HIERARCH CAHA INS TANK TEMP12 = 0.0 / [K]
HIERARCH CAHA INS TANK TEMP13 = 0.0 / [K]
HIERARCH CAHA INS TANK TEMP14 = 0.0 / [K]
HIERARCH CAHA INS TANK TEMP15 = 0.0 / [K]
HIERARCH CAHA INS TANK TEMP16 = 0.0 / [K]
HIERARCH CAHA INS TANK PRESS1 = 0.0 / [Pa]
HIERARCH CAHA INS TANK PRESS2 = 0.0 / [Pa]
HIERARCH CAHA INS TANK SORPPUMP STATUS = ''
HIERARCH CAHA INS TANK SORPPUMP TEMP1 = 0
HIERARCH CAHA INS TANK SORPPUMP TEMP2 = 0
HIERARCH CAHA TARG NAME = '' / Name of target
HIERARCH CAHA TARG RA = 0.0 / 00:00:00.000 (J2000)
HIERARCH CAHA TARG DEC = 0.0 / 00:00:00.00 (J2000)
HIERARCH CAHA TARG MURA = 0.0 / [mas/a]
HIERARCH CAHA TARG MUDEC = 0.0 / [mas/a]
HIERARCH CAHA TARG RV = 0 / [km/s] Radial velocity
HIERARCH CAHA TARG SPEC TYP = 'MV ' / Spectral Type
HIERARCH CAHA TARG MASK = '' / Spectral Mask
HIERARCH CAHA SIMU VERSION = '5.6.0 ' / Simulation version
HIERARCH CAHA SIMU SMP METH = 'mc ' / Sampling method
HIERARCH CAHA SIMU MC METH = 'cdf ' / Monte−Carlo sampling method
HIERARCH CAHA SIMU FIBA = 'P ' / Simulation type Fiber A
HIERARCH CAHA SIMU FIBA SRC = 'OBJ ' / Fiber A source
HIERARCH CAHA SIMU INA1 = 'phx_wavelengths.npy' / FibA infile1
HIERARCH CAHA SIMU INA2 = 'phx_intensities.npy' / FibA infile2
HIERARCH CAHA SIMU RVA = 0.0 / [m/s] Input radial velocity Fiber A
HIERARCH CAHA SIMU FIBB = 'T ' / Simulation type Fiber B
HIERARCH CAHA SIMU FIBB SRC = 'ThAr ' / Fiber B source
HIERARCH CAHA SIMU INB1 = 'L ' / FibB infile1
HIERARCH CAHA SIMU INB2 = '' / FibB infile2
HIERARCH CAHA SIMU RVB = 0.0 / [m/s] Input radial velocity Fiber B
HIERARCH CAHA SIMU SLITTYP = 'uniform ' / Slit function cross−section type
HIERARCH CAHA SIMU NRA IN = 1000000000 / Input number of rays fib A(at cmdline)
HIERARCH CAHA SIMU NRB IN = 1000000000 / Input number of rays fib B(at cmdline)
HIERARCH CAHA SIMU NRA59 = 69511261 / Input number of rays order 59 fib A
HIERARCH CAHA SIMU NRA60 = 66939891 / Input number of rays order 60 fib A
HIERARCH CAHA SIMU NRA61 = 62730925 / Input number of rays order 61 fib A
HIERARCH CAHA SIMU NRA62 = 68589302 / Input number of rays order 62 fib A
HIERARCH CAHA SIMU NRA63 = 76123834 / Input number of rays order 63 fib A
HIERARCH CAHA SIMU NRA64 = 74474686 / Input number of rays order 64 fib A
HIERARCH CAHA SIMU NRA65 = 68725350 / Input number of rays order 65 fib A
HIERARCH CAHA SIMU NRA66 = 61139683 / Input number of rays order 66 fib A
HIERARCH CAHA SIMU NRA67 = 62784150 / Input number of rays order 67 fib A

HIERARCH CAHA SIMU NRA68 = 54401297 / Input number of rays order 68 fib A
HIERARCH CAHA SIMU NRA69 = 52937922 / Input number of rays order 69 fib A
HIERARCH CAHA SIMU NRA70 = 52557366 / Input number of rays order 70 fib A
HIERARCH CAHA SIMU NRA71 = 45880477 / Input number of rays order 71 fib A
HIERARCH CAHA SIMU NRA72 = 39372692 / Input number of rays order 72 fib A
HIERARCH CAHA SIMU NRA73 = 45808831 / Input number of rays order 73 fib A
HIERARCH CAHA SIMU NRA74 = 48568619 / Input number of rays order 74 fib A
HIERARCH CAHA SIMU NRA75 = 44768560 / Input number of rays order 75 fib A
HIERARCH CAHA SIMU NRA76 = 41979868 / Input number of rays order 76 fib A
HIERARCH CAHA SIMU NRA77 = 34336304 / Input number of rays order 77 fib A
HIERARCH CAHA SIMU NRA78 = 27261655 / Input number of rays order 78 fib A
HIERARCH CAHA SIMU NRA79 = 21618086 / Input number of rays order 79 fib A
HIERARCH CAHA SIMU NRA80 = 25036274 / Input number of rays order 80 fib A
HIERARCH CAHA SIMU NRA81 = 35378767 / Input number of rays order 81 fib A
HIERARCH CAHA SIMU NRA82 = 34973706 / Input number of rays order 82 fib A
HIERARCH CAHA SIMU NRA83 = 29636708 / Input number of rays order 83 fib A
HIERARCH CAHA SIMU NRA84 = 19574803 / Input number of rays order 84 fib A
HIERARCH CAHA SIMU NRA85 = 11186925 / Input number of rays order 85 fib A
HIERARCH CAHA SIMU NRA86 = 11229668 / Input number of rays order 86 fib A
HIERARCH CAHA SIMU NRA87 = 14330026 / Input number of rays order 87 fib A
HIERARCH CAHA SIMU NRA88 = 10979817 / Input number of rays order 88 fib A
HIERARCH CAHA SIMU NRA89 = 7075857 / Input number of rays order 89 fib A
HIERARCH CAHA SIMU NRA90 = 5749936 / Input number of rays order 90 fib A
HIERARCH CAHA SIMU NRA91 = 5792982 / Input number of rays order 91 fib A
HIERARCH CAHA SIMU NRA92 = 8136435 / Input number of rays order 92 fib A
HIERARCH CAHA SIMU NRA93 = 10430056 / Input number of rays order 93 fib A
HIERARCH CAHA SIMU NRA94 = 9915026 / Input number of rays order 94 fib A
HIERARCH CAHA SIMU NRA95 = 7825187 / Input number of rays order 95 fib A
HIERARCH CAHA SIMU NRA96 = 6393540 / Input number of rays order 96 fib A
HIERARCH CAHA SIMU NRA97 = 5173780 / Input number of rays order 97 fib A
HIERARCH CAHA SIMU NRA98 = 3920590 / Input number of rays order 98 fib A
HIERARCH CAHA SIMU NRA99 = 5344392 / Input number of rays order 99 fib A
HIERARCH CAHA SIMU NRA100 = 7416545 / Input number of rays order 100 fib A
HIERARCH CAHA SIMU NRA101 = 6868984 / Input number of rays order 101 fib A
HIERARCH CAHA SIMU NRA102 = 4849592 / Input number of rays order 102 fib A
HIERARCH CAHA SIMU NRA103 = 3013742 / Input number of rays order 103 fib A
HIERARCH CAHA SIMU NRA104 = 3403531 / Input number of rays order 104 fib A
HIERARCH CAHA SIMU NRA105 = 4363256 / Input number of rays order 105 fib A
HIERARCH CAHA SIMU NRA106 = 5167639 / Input number of rays order 106 fib A
HIERARCH CAHA SIMU NRA107 = 4401588 / Input number of rays order 107 fib A
HIERARCH CAHA SIMU NRA108 = 3556735 / Input number of rays order 108 fib A
HIERARCH CAHA SIMU NRA109 = 3508321 / Input number of rays order 109 fib A
HIERARCH CAHA SIMU NRA110 = 3667323 / Input number of rays order 110 fib A
HIERARCH CAHA SIMU NRA111 = 3453627 / Input number of rays order 111 fib A
HIERARCH CAHA SIMU NRB59 = 1118626 / Input number of rays order 59 fib B
HIERARCH CAHA SIMU NRB60 = 1186585 / Input number of rays order 60 fib B
HIERARCH CAHA SIMU NRB61 = 2248826 / Input number of rays order 61 fib B
HIERARCH CAHA SIMU NRB62 = 7033246 / Input number of rays order 62 fib B
HIERARCH CAHA SIMU NRB63 = 44270883 / Input number of rays order 63 fib B
HIERARCH CAHA SIMU NRB64 = 10780045 / Input number of rays order 64 fib B
HIERARCH CAHA SIMU NRB65 = 15224112 / Input number of rays order 65 fib B
HIERARCH CAHA SIMU NRB66 = 30250930 / Input number of rays order 66 fib B
HIERARCH CAHA SIMU NRB67 = 49979160 / Input number of rays order 67 fib B
HIERARCH CAHA SIMU NRB68 = 1661612 / Input number of rays order 68 fib B
HIERARCH CAHA SIMU NRB69 = 1109715 / Input number of rays order 69 fib B
HIERARCH CAHA SIMU NRB70 = 4952539 / Input number of rays order 70 fib B
HIERARCH CAHA SIMU NRB71 = 5235652 / Input number of rays order 71 fib B
HIERARCH CAHA SIMU NRB72 = 78785356 / Input number of rays order 72 fib B
HIERARCH CAHA SIMU NRB73 = 43498091 / Input number of rays order 73 fib B
HIERARCH CAHA SIMU NRB74 = 36541546 / Input number of rays order 74 fib B
HIERARCH CAHA SIMU NRB75 = 103809571 / Input number of rays order 75 fib B
HIERARCH CAHA SIMU NRB76 = 190020424 / Input number of rays order 76 fib B
HIERARCH CAHA SIMU NRB77 = 122043689 / Input number of rays order 77 fib B
HIERARCH CAHA SIMU NRB78 = 2928200 / Input number of rays order 78 fib B
HIERARCH CAHA SIMU NRB79 = 13951821 / Input number of rays order 79 fib B
HIERARCH CAHA SIMU NRB80 = 20414686 / Input number of rays order 80 fib B
HIERARCH CAHA SIMU NRB81 = 50390785 / Input number of rays order 81 fib B
HIERARCH CAHA SIMU NRB82 = 51364228 / Input number of rays order 82 fib B
HIERARCH CAHA SIMU NRB83 = 4884631 / Input number of rays order 83 fib B
HIERARCH CAHA SIMU NRB84 = 12623615 / Input number of rays order 84 fib B
HIERARCH CAHA SIMU NRB85 = 14353965 / Input number of rays order 85 fib B
HIERARCH CAHA SIMU NRB86 = 37268671 / Input number of rays order 86 fib B
HIERARCH CAHA SIMU NRB87 = 62806364 / Input number of rays order 87 fib B
HIERARCH CAHA SIMU NRB88 = 71150913 / Input number of rays order 88 fib B
HIERARCH CAHA SIMU NRB89 = 35543326 / Input number of rays order 89 fib B
HIERARCH CAHA SIMU NRB90 = 44223330 / Input number of rays order 90 fib B
HIERARCH CAHA SIMU NRB91 = 61850243 / Input number of rays order 91 fib B
HIERARCH CAHA SIMU NRB92 = 54589610 / Input number of rays order 92 fib B
HIERARCH CAHA SIMU NRB93 = 23142239 / Input number of rays order 93 fib B
HIERARCH CAHA SIMU NRB94 = 17911908 / Input number of rays order 94 fib B
HIERARCH CAHA SIMU NRB95 = 31662629 / Input number of rays order 95 fib B
HIERARCH CAHA SIMU NRB96 = 33553515 / Input number of rays order 96 fib B
HIERARCH CAHA SIMU NRB97 = 27844852 / Input number of rays order 97 fib B
HIERARCH CAHA SIMU NRB98 = 29866642 / Input number of rays order 98 fib B
HIERARCH CAHA SIMU NRB99 = 33155186 / Input number of rays order 99 fib B
HIERARCH CAHA SIMU NRB100 = 34730999 / Input number of rays order 100 fib B

```
HIERARCH CAHA SIMU NRB101 = 32558155 / Input number of rays order 101 fib B
HIERARCH CAHA SIMU NRB102 = 28362316 / Input number of rays order 102 fib B
HIERARCH CAHA SIMU NRB103 = 23127885 / Input number of rays order 103 fib B
HIERARCH CAHA SIMU NRB104 = 16175334 / Input number of rays order 104 fib B
HIERARCH CAHA SIMU NRB105 = 17479985 / Input number of rays order 105 fib B
HIERARCH CAHA SIMU NRB106 = 15764538 / Input number of rays order 106 fib B
HIERARCH CAHA SIMU NRB107 = 13260614 / Input number of rays order 107 fib B
HIERARCH CAHA SIMU NRB108 = 14591335 / Input number of rays order 108 fib B
HIERARCH CAHA SIMU NRB109 = 17296763 / Input number of rays order 109 fib B
HIERARCH CAHA SIMU NRB110 = 17996731 / Input number of rays order 110 fib B
HIERARCH CAHA SIMU NRB111 = 16271774 / Input number of rays order 111 fib B
HIERARCH CAHA SIMU MEANNRA = 317.6785869259231 / Mean rays per raw pixel fiber A
HIERARCH CAHA SIMU MEANNRB = 3102.615394439101 / Mean rays per raw pixel fiber B
HIERARCH CAHA SIMU MINNRA = 1 / Min rays per raw pixel fiber A
HIERARCH CAHA SIMU MINNRB = 1 / Min rays per raw pixel fiber B
HIERARCH CAHA SIMU MAXNRA = 2301 / Max rays per raw pixel fiber A
HIERARCH CAHA SIMU MAXNRB = 1907936 / Max rays per raw pixel fiber B
HIERARCH CAHA SIMU TOTNRA = 817027392 / Total rays fiber A
HIERARCH CAHA SIMU TOTNRB = 777425442 / Total rays fiber B
HIERARCH CAHA SIMU DW = 0.001 / [nm] Input wavelength grid sampling
HIERARCH CAHA SIMU OMIN = 59 / Minimum echelle order
HIERARCH CAHA SIMU OMAX = 111 / Maximum echelle order
HIERARCH CAHA SIMU NIRGAP = F / [mm] Gap length of NIR detectors
HIERARCH CAHA SIMU BLAZE = T / Blaze function simulated
HIERARCH CAHA SIMU PN = T / Shot noise simulated
HIERARCH CAHA SIMU BIAS = 300 / [DN] Input bias
HIERARCH CAHA SIMU RON = 5.0 / [DN] Input readout noise
HIERARCH CAHA SIMU TELL CH4 = F / Telluric species included
HIERARCH CAHA SIMU TELL CO2 = F / Telluric species included
HIERARCH CAHA SIMU TELL H2O = F / Telluric species included
HIERARCH CAHA SIMU TELL N2O = F / Telluric species included
HIERARCH CAHA SIMU TELL O2 = F / Telluric species included
HIERARCH CAHA SIMU TELL O3 = F / Telluric species included
HIERARCH CAHA SIMU TIME FIBA = 681.49 / [s] Fiber A simulation time
HIERARCH CAHA SIMU TIME FIBB = 677.99 / [s] Fiber B simulation time
HIERARCH CARACAL BIAS FILE = 57155.6059999999997672 / csfs.2015-05-13T16:32:5
HIERARCH CARACAL ODEF FILE = 57156.7869999999966240 / csfs.2015-05-14T18:09:4
HIERARCH CARACAL FLAT FILE = 57155.8380000000033760 / csfs.2015-05-13T22:07:2
HIERARCH CARACAL WAVE FILE = 57156.7880000000004657 / csfs.2015-05-14T20:55:1
E_OBSLON=    2.54630000000 / [deg] Longitude of observatory
E_OBSLAT=    37.2236000000 / [deg] Latitude of observatory
E_OBSALT=             2168 / [m] Altitude of observatory
E_RA2000=          0.00000 / [hr] Right ascension of object for 2000 in hour
E_DE2000=          0.00000 / [deg] Declination of object for 2000
E_JD   =    57157.78653935 / Julian date - 2400000 at middle of exposure
E_HJD =     57157.78308484 / Heliocentric Julian date - 2400000 at middle of
E_HELCOR=    23.7701629522 / [km/s] Baricentric correction
E_BJD =     57157.86646140 / Barycentric Julian Day
E_BERV =     23.60721800 / [km/s] Barycentric correction
E_HVERS =          1.10000 / Header keyword version number
E_SPMODE= 'carm_vis'       / Instrument mode
E_PREFMO= 'carm_vis'       / Instrument mode
E_ORIENT=                0 / Reorientation flag for extraction
E_AMPL =                 1 / Number of amplifiers used for readout
E_XLO =                  0 / Lowest X index to extract (0 base)
E_YLO =                  0 / Lowest Y index to extract (0 base)
E_XHI =               4095 / Highest X index to extract (0 base)
E_YHI =               4095 / Highest Y index to extract (0 base)
E_GAIN =           1.00000 / Gain (electrons/ADU)
E_READN =          5.00000 / Read noise (electrons)
E_BACKG =          0.00000 / Total background dark and sky (electrons)
E_TIME =           0.00000 / [s] Exposure time
HIERARCH CARACAL FOX XWD = 14 / extraction width
HIERARCH CARACAL FOX KAPPA = 10 / kappa-sigma clipping threshold
HIERARCH CARACAL FOX SNR 0 = 17.86 / SNR in order 111
HIERARCH CARACAL FOX SNR 1 = 20.363 / SNR in order 110
HIERARCH CARACAL FOX SNR 2 = 17.756 / SNR in order 109
HIERARCH CARACAL FOX SNR 3 = 18.547 / SNR in order 108
HIERARCH CARACAL FOX SNR 4 = 21.504 / SNR in order 107
HIERARCH CARACAL FOX SNR 5 = 22.918 / SNR in order 106
HIERARCH CARACAL FOX SNR 6 = 23.28 / SNR in order 105
HIERARCH CARACAL FOX SNR 7 = 14.961 / SNR in order 104
HIERARCH CARACAL FOX SNR 8 = 16.671 / SNR in order 103
HIERARCH CARACAL FOX SNR 9 = 20.888 / SNR in order 102
HIERARCH CARACAL FOX SNR 10 = 26.07 / SNR in order 101
HIERARCH CARACAL FOX SNR 11 = 29.027 / SNR in order 100
HIERARCH CARACAL FOX SNR 12 = 22.199 / SNR in order 99
HIERARCH CARACAL FOX SNR 13 = 18.87 / SNR in order 98
HIERARCH CARACAL FOX SNR 14 = 23.581 / SNR in order 97
HIERARCH CARACAL FOX SNR 15 = 25.671 / SNR in order 96
HIERARCH CARACAL FOX SNR 16 = 30.281 / SNR in order 95
HIERARCH CARACAL FOX SNR 17 = 34.415 / SNR in order 94
HIERARCH CARACAL FOX SNR 18 = 35.336 / SNR in order 93
HIERARCH CARACAL FOX SNR 19 = 31.926 / SNR in order 92
HIERARCH CARACAL FOX SNR 20 = 26.888 / SNR in order 91
HIERARCH CARACAL FOX SNR 21 = 27.676 / SNR in order 90
```

```
HIERARCH CARACAL FOX SNR 22 = 29.659 / SNR in order 89
HIERARCH CARACAL FOX SNR 23 = 37.711 / SNR in order 88
HIERARCH CARACAL FOX SNR 24 = 45.307 / SNR in order 87
HIERARCH CARACAL FOX SNR 25 = 34.443 / SNR in order 86
HIERARCH CARACAL FOX SNR 26 = 35.621 / SNR in order 85
HIERARCH CARACAL FOX SNR 27 = 48.231 / SNR in order 84
HIERARCH CARACAL FOX SNR 28 = 60.202 / SNR in order 83
HIERARCH CARACAL FOX SNR 29 = 65.475 / SNR in order 82
HIERARCH CARACAL FOX SNR 30 = 68.359 / SNR in order 81
HIERARCH CARACAL FOX SNR 31 = 56.419 / SNR in order 80
HIERARCH CARACAL FOX SNR 32 = 54.477 / SNR in order 79
HIERARCH CARACAL FOX SNR 33 = 60.528 / SNR in order 78
HIERARCH CARACAL FOX SNR 34 = 68.294 / SNR in order 77
HIERARCH CARACAL FOX SNR 35 = 76.291 / SNR in order 76
HIERARCH CARACAL FOX SNR 36 = 80.408 / SNR in order 75
HIERARCH CARACAL FOX SNR 37 = 80.999 / SNR in order 74
HIERARCH CARACAL FOX SNR 38 = 80.058 / SNR in order 73
HIERARCH CARACAL FOX SNR 39 = 71.034 / SNR in order 72
HIERARCH CARACAL FOX SNR 40 = 80.572 / SNR in order 71
HIERARCH CARACAL FOX SNR 41 = 85.833 / SNR in order 70
HIERARCH CARACAL FOX SNR 42 = 87.625 / SNR in order 69
HIERARCH CARACAL FOX SNR 43 = 89.82 / SNR in order 68
HIERARCH CARACAL FOX SNR 44 = 98.056 / SNR in order 67
HIERARCH CARACAL FOX SNR 45 = 97.343 / SNR in order 66
HIERARCH CARACAL FOX SNR 46 = 99.147 / SNR in order 65
HIERARCH CARACAL FOX SNR 47 = 104.88 / SNR in order 64
HIERARCH CARACAL FOX SNR 48 = 106.97 / SNR in order 63
HIERARCH CARACAL FOX SNR 49 = 104.28 / SNR in order 62
HIERARCH CARACAL FOX SNR 50 = 99.496 / SNR in order 61
HIERARCH CARACAL FOX SNR 51 = 104.43 / SNR in order 60
HIERARCH CARACAL FOX SNR 52 = 107.5 / SNR in order 59
HIERARCH CARACAL FOX CHI2RED 0 = 3.3282 / sqrt(chi2red) in order 111
HIERARCH CARACAL FOX CHI2RED 1 = 3.2898 / sqrt(chi2red) in order 110
HIERARCH CARACAL FOX CHI2RED 2 = 3.24 / sqrt(chi2red) in order 109
HIERARCH CARACAL FOX CHI2RED 3 = 3.2021 / sqrt(chi2red) in order 108
HIERARCH CARACAL FOX CHI2RED 4 = 3.0759 / sqrt(chi2red) in order 107
HIERARCH CARACAL FOX CHI2RED 5 = 3.0224 / sqrt(chi2red) in order 106
HIERARCH CARACAL FOX CHI2RED 6 = 3.1534 / sqrt(chi2red) in order 105
HIERARCH CARACAL FOX CHI2RED 7 = 3.3394 / sqrt(chi2red) in order 104
HIERARCH CARACAL FOX CHI2RED 8 = 3.3859 / sqrt(chi2red) in order 103
HIERARCH CARACAL FOX CHI2RED 9 = 2.9993 / sqrt(chi2red) in order 102
HIERARCH CARACAL FOX CHI2RED 10 = 2.8981 / sqrt(chi2red) in order 101
HIERARCH CARACAL FOX CHI2RED 11 = 2.9305 / sqrt(chi2red) in order 100
HIERARCH CARACAL FOX CHI2RED 12 = 3.1068 / sqrt(chi2red) in order 99
HIERARCH CARACAL FOX CHI2RED 13 = 3.0658 / sqrt(chi2red) in order 98
HIERARCH CARACAL FOX CHI2RED 14 = 2.9115 / sqrt(chi2red) in order 97
HIERARCH CARACAL FOX CHI2RED 15 = 2.8509 / sqrt(chi2red) in order 96
HIERARCH CARACAL FOX CHI2RED 16 = 3.1526 / sqrt(chi2red) in order 95
HIERARCH CARACAL FOX CHI2RED 17 = 3.0573 / sqrt(chi2red) in order 94
HIERARCH CARACAL FOX CHI2RED 18 = 3.0032 / sqrt(chi2red) in order 93
HIERARCH CARACAL FOX CHI2RED 19 = 3.1043 / sqrt(chi2red) in order 92
HIERARCH CARACAL FOX CHI2RED 20 = 3.1796 / sqrt(chi2red) in order 91
HIERARCH CARACAL FOX CHI2RED 21 = 3.1318 / sqrt(chi2red) in order 90
HIERARCH CARACAL FOX CHI2RED 22 = 3.0126 / sqrt(chi2red) in order 89
HIERARCH CARACAL FOX CHI2RED 23 = 2.859 / sqrt(chi2red) in order 88
HIERARCH CARACAL FOX CHI2RED 24 = 2.8853 / sqrt(chi2red) in order 87
HIERARCH CARACAL FOX CHI2RED 25 = 3.1378 / sqrt(chi2red) in order 86
HIERARCH CARACAL FOX CHI2RED 26 = 2.837 / sqrt(chi2red) in order 85
HIERARCH CARACAL FOX CHI2RED 27 = 2.5602 / sqrt(chi2red) in order 84
HIERARCH CARACAL FOX CHI2RED 28 = 2.3613 / sqrt(chi2red) in order 83
HIERARCH CARACAL FOX CHI2RED 29 = 2.3428 / sqrt(chi2red) in order 82
HIERARCH CARACAL FOX CHI2RED 30 = 2.3435 / sqrt(chi2red) in order 81
HIERARCH CARACAL FOX CHI2RED 31 = 2.6143 / sqrt(chi2red) in order 80
HIERARCH CARACAL FOX CHI2RED 32 = 2.5879 / sqrt(chi2red) in order 79
HIERARCH CARACAL FOX CHI2RED 33 = 2.4875 / sqrt(chi2red) in order 78
HIERARCH CARACAL FOX CHI2RED 34 = 2.3626 / sqrt(chi2red) in order 77
HIERARCH CARACAL FOX CHI2RED 35 = 2.2944 / sqrt(chi2red) in order 76
HIERARCH CARACAL FOX CHI2RED 36 = 2.2908 / sqrt(chi2red) in order 75
HIERARCH CARACAL FOX CHI2RED 37 = 2.3136 / sqrt(chi2red) in order 74
HIERARCH CARACAL FOX CHI2RED 38 = 2.3637 / sqrt(chi2red) in order 73
HIERARCH CARACAL FOX CHI2RED 39 = 2.4393 / sqrt(chi2red) in order 72
HIERARCH CARACAL FOX CHI2RED 40 = 2.3389 / sqrt(chi2red) in order 71
HIERARCH CARACAL FOX CHI2RED 41 = 2.3038 / sqrt(chi2red) in order 70
HIERARCH CARACAL FOX CHI2RED 42 = 2.4264 / sqrt(chi2red) in order 69
HIERARCH CARACAL FOX CHI2RED 43 = 2.3926 / sqrt(chi2red) in order 68
HIERARCH CARACAL FOX CHI2RED 44 = 2.369 / sqrt(chi2red) in order 67
HIERARCH CARACAL FOX CHI2RED 45 = 2.3885 / sqrt(chi2red) in order 66
HIERARCH CARACAL FOX CHI2RED 46 = 2.3365 / sqrt(chi2red) in order 65
HIERARCH CARACAL FOX CHI2RED 47 = 2.3357 / sqrt(chi2red) in order 64
HIERARCH CARACAL FOX CHI2RED 48 = 2.3513 / sqrt(chi2red) in order 63
HIERARCH CARACAL FOX CHI2RED 49 = 2.3681 / sqrt(chi2red) in order 62
HIERARCH CARACAL FOX CHI2RED 50 = 2.3431 / sqrt(chi2red) in order 61
HIERARCH CARACAL FOX CHI2RED 51 = 2.4073 / sqrt(chi2red) in order 60
HIERARCH CARACAL FOX CHI2RED 52 = 2.3711 / sqrt(chi2red) in order 59
HIERARCH CARACAL FOX VERSION = 'v1.02 2013-11-19' / Author: M. Zechmeister
HIERARCH CARACAL USERHOST = 'zechmeister@astro43'
```

```
HIERARCH CARACAL VERSION = 'v0.04 2015-05-19' / Author: M. Zechmeister
ECHVERS =              1.00000 / version number of echelle data format
HIERARCH CARACAL DRIFT REF = 'too lang' / Ref file used
HIERARCH CARACAL DRIFT RV = -0.61543884 / [m/s] Drift in cal fib
HIERARCH CARACAL DRIFT RVERR = 0.54015754 / [m/s] Estimated RV uncertainty
END
```

A.7. Help output of SERVAL (RV pipeline)

```
usage: serval.py [-h] [-sa SA] [-targ TARG] [-targrade TARGRADE TARGRADE]
                 [-targpm TARGPM TARGPM] [-rvguess RVGUESS] [-atmmask ATMMASK]
                 [-ccf [CCF]] [-coadd {fly,post,post2,post3}] [-coset COSET]
                 [-co_excl CO_EXCL] [-driftref DRIFTREF] [-fib {,A,B,AB}]
                 [-inst {HARPS,HARPN,CARM_VIS,CARM_NIR,FEROS,FTS}]
                 [-iset ISET] [-kapsig KAPSIG] [-last] [-look [LOOK]]
                 [-nclip NCLIP] [-oset OSET] [-o_excl O_EXCL] [-outres]
                 [-pmin PMIN] [-pmax PMAX] [-pspline [PSPLLAM]] [-reana]
                 [-review [REVIEW]] [-rvwarn RVWARN] [-skippre] [-snmin SNMIN]
                 [-snmax SNMAX] [-template TEMPLATE] [-tnr [TNR]] [-verb]
                 [-vref VREF] [-vtfix] [-vrange [VRANGE [VRANGE ...]]] [-wfix]
                 [-debug [DEBUG]] [-bp [BP [BP ...]]] [-pdb] [-cprofile]
                 obj path

SERVAL - SpEctrum Radial Velocity AnaLyser (2015-06-10)

positional arguments:
  obj                   Object name
  path                  path to directory with reduced data fits/tar

optional arguments:
  -h, --help            show this help message and exit
  -sa SA                [m/s/yr] secular acceleration
  -targ TARG            target name looked up in star.cat
  -targrade TARGRADE TARGRADE
                        target coordinates: [ra|hh:mm:ss.sss de|de:mm:ss.sss]
  -targpm TARGPM TARGPM
                        target proper motion: pmra [mas/yr] pmde [mas/yr]
  -rvguess RVGUESS      [km/s] target rv guess (for ccf)
  -atmmask ATMMASK      telluric line mask ( for no masking) (default: auto)
  -ccf [CCF]            mode ccf [with files ]
  -coadd {fly,post,post2,post3}
                        coadd method (default: post3)
  -coset COSET          index for order in coadding (default: oset)
  -co_excl CO_EXCL      orders to exclude in coadding (default: o_excl)
  -driftref DRIFTREF    reference file for drift mode
  -fib {,A,B,AB}        fibre (currenly HARPS only)
  -inst {HARPS,HARPN,CARM_VIS,CARM_NIR,FEROS,FTS}
                        instrument (default: HARPS)
  -iset ISET            index for file subset (e.g. 1:10, ::5)
  -kapsig KAPSIG        kappa sigma clip value (default: 3.0)
  -last                 use last template
  -look [LOOK]          set of orders to view [-1 all]
  -nclip NCLIP          max. number of clipping iterations (default: 2)
  -oset OSET            index for order subset (e.g. 1:10, ::5)
  -o_excl O_EXCL        orders to exclude (e.g. 1,10,3)
  -outres               output the residuals
  -pmin PMIN            minimum pixel (default: 300)
  -pmax PMAX            maximum pixel (default: 3800)
  -pspline [PSPLLAM]    pspline as coadd filter [smooth value]
  -reana                flag reanalyse only
  -review [REVIEW]      level of review template
  -rvwarn RVWARN        [km/s] warning threshold in debug (default: 2.0)
  -skippre              skip pre RVs
  -snmin SNMIN          minimum S/N (considered as no bad and used in template
                        building) (default: 10)
  -snmax SNMAX          maximum S/N (considered as no bad and used in template
                        building) (default: 400)
  -template TEMPLATE template file
  -tnr [TNR]            template: None=create template, ''= highest S/N
                        spectrum 'filenr'= filenr
  -verb                 verbose
  -vref VREF            [km/s] reference RV of the template (for index
                        measures)
  -vtfix                fix RV in template creation
  -vrange [VRANGE [VRANGE ...]]
                        velocity grid (v_lo, v_hi,v_num) (default: (-5.5, 5.5,
                        33))
  -wfix                 fix wavelength solution
  -debug [DEBUG]        debug flag
  -bp [BP [BP ...]]     break points
  -pdb                  debug post_mortem
  -cprofile             profiling

usage example: ./src/serval.py test /dir/to/spectra/of/gj699/ -coadd post3
```

−targ gj699 −snmin 10 −oset 40:

A.8. SERVAL algorithm

A.8.1. Concept

The radial velocity computation aims for highest precision and is based on least square fitting. Anglada-Escudé & Butler (2012) demonstrated that this approach can provide more precise RVs than the CCF method. The reason is that the RV precision depends on both the signal-to-noise ratio (SNR) of the data and the match of model to the data. Aiming for 1 m/s precision requires typically SNR of ~ 100 and the model match becomes important. With a proper model of the observations the RV information is properly weighted. However, it should be also mentioned that in case the coadded template is still too noisy, e.g. too few and/or too noisy observations, using some external template might give better results.

Our very general concept is to simultaneously decompose all observations into a high signal-to-noise template $f(\lambda, b)$, the radial velocity shifts v_n and multiplicative background polynomials $p(x, a)$ to account for flux variation. Therefore our forward model for the observed flux is

$$y(\lambda, x) = p(x, a) f(\ln \lambda - v, b) \tag{A.1}$$

We denote some $n = 1 \dots N$ observations taken at time t_n each consisting of flux measurements y_i at pixel i with wavelengths λ_i and flux error estimate σ_i. For echelle spectrographs these measurements are done in several echelle orders o (so a more explicit indexing is $y_{n,o,i}$). The weighted sum of residuals is

$$\chi^2 = \sum_{n,o,i} \frac{[y_{n,o,i} - y(\lambda_{n,i}, v_n, a_{n,o}, b)]^2}{\sigma^2_{n,o,i}} = \sum_{n,o,i} w_{n,o,i} [y_{n,o,i} - p(x, a) f(\ln \lambda - v, b)]^2 \tag{A.2}$$

However, due to the large number of data ($N \times O \times K$) and parameters this simultaneous approach is less feasible in practice, so that the decomposition is done sequentially and iteratively. First, the highest signal-to-noise observation is taken as a reference to shift all observations into this reference frame and to coadd them to a high signal-to-noise template f. Then this new template is used to re-compute the radial velocity while the coefficient of the background polynomial are fitted simultaneously (a simplification would be to fit them in advance, but it may not result the best full fit). This one iteration is usually enough.

We assume that the spectrum is smooth and use (cubic) spline interpolation to evaluate the template f at any wavelength λ, which is needed for the forward modelling in Eq. (A.1). The knot values of the template are tabulated with a uniform grid $\ln \lambda_k$ and f_k (or B-spline coefficient b_k, resp.).

Logarithmic wavelengths are used. In this case the Doppler shift is additive (assuming small shift)

$$\frac{\Delta v}{c} = \frac{\Delta \lambda}{\lambda} = \Delta \ln \lambda \, .$$

A given a template $T(\ln \lambda)$, a spectral feature at λ_0 in T appears at λ_1 in the observation O, i.e.$O(\ln \lambda_1) = T(\ln \lambda_0) = T(\ln \lambda_1 - \frac{v}{c})$, so the most basic model is

$$O(\ln \lambda) = T(\ln \lambda - \frac{v}{c})$$

were v is the only one parameter.

A.8.2. Least square RV fitting

In the step when computing the RV, we optimise Eq. (A.2) wrt. the polynomial coefficients a and the RV shift v. The fit is linear wrt. to a, but the RV shift makes the fit non-linear. There are several algorithms, like e.g. Levenberg-Marquardt or bisection, to solve non-linear least square problems. We use, in analogy to the CCF method, a brute force stepping through the velocity parameter space v. At fixed v, the template $f_i = f(\lambda_i, v, b)$ can be evaluated and $\chi^2(v)$ can be obtained from a simple linear least fit for a

$$\chi^2 = \sum_i w_i [y_i - p(x_i, a) f_i]^2$$

We note that in the form $\chi^2 = \sum_i f_i^2 w_i [\frac{y_i}{f_i} - p(x_i, a)]^2$ and with the substitution $f_i^2 w_i \rightarrow w_i$ and $\frac{y_i}{f_i} \rightarrow y_i$, standard library routines for polynomial fitting can be applied. However, division with zero flux f_i should be avoided/checked.

The $\chi^2(v)$ function sampled at (v_k, χ^2_k) is exploited for its global minimum and the Taylor expansion of the $\chi^2(v)$ function at the minimum to second order is

$$\chi^2 = \chi^2_{\min} + (\chi^2)' \Delta v + \frac{1}{2} (\chi^2)'' (\Delta v)^2 = \chi^2_{\min} + \frac{1}{2} (\chi^2)'' (\Delta v)^2 \tag{A.3}$$

Since we use cubic interpolation for the template, the second derivative of the template (and of χ^2) is continuous. Then a parabolic interpolation around the minimum of the $\chi^2(v)$ function provides a refined estimate for v from the parabola minimum and an error estimate from the parabola curvature. The parabola minimum is located at the point (Sect. A.10)

$$v = v_k - (v_k - v_{k-1}) \frac{\chi^2_{k+1} - \chi^2_{k-1}}{\chi^2_{k-1} - 2\chi^2_k + \chi^2_{k+1}}$$

where $k = \arg\min_i \chi^2_i$.

An uncertainty estimate might be obtained from Eq. (A.3) with the $\Delta\chi^2 = \chi^2 - \chi^2_{\min} = 1$ criterion

$$(\Delta v)^2 = 2\frac{1}{(\chi^2)''} = 2\frac{(v_k - v_{k-1})^2}{\chi^2_{k-1} - 2\chi^2_k + \chi^2_{k+1}} \tag{A.4}$$

This value might be re-scaled with χ^2_{red}.

Finally, a statistical analysis of the fit is performed. The χ^2 and χ^2_{red} are computed. Outliers in the (normalised) residuals $r_i = y_i - y_{\text{mod},i}$ ($\chi_i = \frac{r_i}{\sigma_i}$) with

$$|r_i| > \kappa\sigma_i\sqrt{\chi^2_{\text{red}}}$$

$$|\chi_i| > \kappa\sqrt{\chi^2_{\text{red}}}$$

where $\kappa = 3...5$ is a clipping threshold, are flagged (e.g. by setting their weight to zero $w_i = 0$) and excluded in a repeated fitting. Also the average signal-to-noise ratio for the spectrum is estimated as

$$\text{SNR}_{n,o} = \frac{\sqrt{\frac{1}{N}\sum y_i}}{\sqrt{\frac{1}{N}\sum r_i^2}}$$

$$\text{SNR}_{n,o} = \sqrt{\frac{1}{N}\sum\left(\frac{y_i}{r_i}\right)^2}$$

Each order is fitted separately and finally a weighted mean radial velocity over all orders is computed

$$v = \frac{\left\langle \sigma_{v_o}^{-2} v_o \right\rangle}{\left\langle \sigma_{v_o}^{-2} \right\rangle}$$

with the error estimate

$$\sigma_v = \sqrt{\frac{N}{\left\langle \sigma_{v_o}^{-2} \right\rangle}}$$

A.8.3. Template creation

The "co-adding" is done with a cubic uniform B-spline fit to the "normalised" data. Given all $a_{n,o}$ and v_n for the N observations from Sect. A.8.2, we can write Eq. (A.2) in the form

$$\chi^2 = \sum_{n,i} p^2_{n,i} w_{n,i}\left[\frac{y_{n,i}}{p_{n,i}} - f(\ln\lambda_{n,i} + \frac{v_n}{c}, b)\right]^2 \tag{A.5}$$

Now the task is to find all the coefficients b_k of the spline with K knots, so that the residuals are minimal. We will have about one knot per pixel, i.e. the number of knots K is similar to the number of data points N per spectrum in the order. It can be inferred from Eq. (A.5) that the normalised data are $\frac{y_{n,i}}{p_{n,i}}$ and the corresponding normalised error estimates are $\frac{\sigma_{n,i}}{p_{n,i}}$.

Equation (A.5) is a linear least square problem due to the linearity of the B-spline. The advantage of this method is that 1) there is no need to interpolate the data, 2) the possibility to detect outliers and 3) a spline function as direct output (i.e. consistent with the input for the forward model).

However, caution is needed when there are large gaps in the data, i.e. there are regions where we have no information to say anything about the template (i.e. no constraints for b_k). Possible solutions are 1) splitting the spline fit at those points, 2) fitting penalised splines, 3) to only heavily downweight (and not to fully reject) the masked points (outliers, tellurics) which caused the gaps.

$$f(\ln\lambda) = \frac{y(\ln\lambda - v, x)}{p(x, a)}$$

$$\chi^2 = \sum\frac{[y_i - y(\ln\lambda_i + \frac{v}{c})]^2}{\sigma_i}$$

A.9. Drift algorithm

As template we choose the first observation y_0, e.g. the daily calibration frame. We also need the first derivative of the spectrum $\frac{\mathrm{d}f}{\mathrm{d}v}$, which might be taken from the mean of adjacent flux differences or from a the first derivative of a spline interpolation. Then model in the drift algorithm is

$$y \approx Af(\ln\lambda - \Delta v) \approx Af(\ln\lambda) - A\frac{\mathrm{d}f}{\mathrm{d}v}\Delta v$$

Now, given a new observations y_i we first

A.10. Parabola interpolation

We want to derive the interpolating parabola $y = ax^2 + bx + c$ through the three points (x_i, y_i), (x_{i-1}, y_{i-1}) and (x_{i+1}, y_{i+1}). The points should be equidistant, i.e. $x_{i+1} - x_i = \Delta x$ and $x_i - x_{i-1} = \Delta x$. Now we center points to (x_i, y_i) and the three transformed points are $(0, 0)$, $(-\Delta x, Y_{-1})$ and $(\Delta x, Y_{+1})$. Inserting into the parabola equation leads to a system of three equations

$$
\begin{aligned}
0 &= c \\
Y_{-1} &= a\Delta x^2 - b\Delta x \\
Y_1 &= a\Delta x^2 + b\Delta x
\end{aligned}
$$

The solution is

$$
\begin{aligned}
a &= \frac{Y_{-1} + Y_1}{2\Delta x^2} = \frac{y_{i-1} - 2y_i + y_{i+1}}{2\Delta x^2} \\
b &= \frac{Y_1 - Y_{-1}}{2\Delta x} = \frac{y_{i+1} - y_{i-1}}{2\Delta x} \\
c &= 0
\end{aligned}
$$

Reforming the parabola equation to $y = y_i + a(x - x_i + \frac{b}{2a})^2 - (\frac{b}{2a})^2 + c$, we see that the minimum of the parabola will be at

$$x_c = x_i - \frac{b}{2a} = x_i - \Delta x \frac{y_{i+1} - y_{i-1}}{y_{i-1} - 2y_i + y_{i+1}}$$

The second derivative at x_c is

$$y'' = 2a\,.$$