L2LENS USER'S GUIDE

May 2, 2014

This simple User's Guide describes all steps to successfully process the FRODOSpec observations of Q0957+561 (q0957) and the calibration star Feige 34 (feige34) on 2011 March 1. We assume that /home/user/l2lens is the path to the directory (folder) `l2lens`. The `l2lens` folder houses the Python scripts of the L2LENS software, some auxiliary files and the subfolder `110301`. All relevant data files of q0957, feige34, and the W and Xe lamps are located in `110301`. There is also a sub-subfolder `database` containing additional auxiliary files. The initial `l2lens` folder and a README file (explaining its contents; see also here below for a description of the main files and its usage) are freely available at this website: **download the single compressed file l2lens.zip**. We also assume that IRAF and Python are properly installed. The Python modules PyRAF, PyFITS, NumPy, SciPy and Matplotlib are needed to run L2LENS scripts.

Fibre spectrum extraction (IRAF V2.16 + Python 2.7)

For convenience, the pre-processing pipeline (L1) outputs for the blue arm of FRODOSpec are renamed with shorter labels. This is done in `110301`, where the data files are found, using the IRAF command lines:

```
cl> imcopy b_w_20110301_2_1_1_1 bw
cl> imcopy b_e_20110301_7_1_1_2.fits[0] be1
cl> imcopy b_a_20110301_8_1_1_1 ba1
cl> imcopy b_e_20110301_9_1_1_2.fits[0] be2
cl> imcopy b_a_20110301_10_1_1_1 ba2
```

Similar FITS files rw, re1, ra1, re2 and ra2 are produced for the red arm. Here, *w, *e1, *a1, *e2 and *a2 refer to W lamp, q0957, Xe lamp for q0957, feige34 and Xe lamp for feige34, respectively.

The next steps are:

1. *Removing cosmic-ray events*
   Based on the spectroscopic version of the L.A.Cosmic algorithm. This is put into the `extern` directory of IRAF (/iraf/iraf/extern), and then

   ```
   cl> task lacos_spec = /iraf/iraf/extern/
   lacos_spec.cl
   cl> stsdas
   cl> lacos_spec be1 be1cr be1m.pl gain=2.134 readn=3.85
   cl> lacos_spec re1 re1cr re1m.pl gain=2.350 readn=4.44
   ```

   A similar procedure is followed for cleaning the feige34 frames be2 and re2. After subtraction of cosmic-ray events, the main FITS files of q0957 and feige34 are be1cr, re1cr, be2cr and re2cr

2. *Finding and tracing fibre positions along the dispersion axis*
   Based on the IRAF/SPECRED package. The command lines are

   ```
   cl> noao
   cl> imred
   cl> specred
   cl> apall bw nfind=144 resize- lower=-2 upper=2
   background- minsep=5 maxsep=10 width=7 weights-
   clean- t_func="legendre" t_step=50 t_niter=1
   t_order=3 t_sample="500:2500"
   cl> apall rw nfind=144 t_order=6 t_sample="1450:3450"
   ```

   The FITS files bw.ms and rw.ms contain the blue-arm and red-arm W-lamp (continuum emission) spectrum for each fibre

3. *Removing scattered-light backgrounds*
   Based on the IRAF/SPECRED package

   ```
   cl> apscatter be1cr be1sc ref=bw
   buffer=0 apscat1.order=5 apscat2.order=5
   apscat2.sample="5:4096" inter-
   ```

A similar procedure is followed for cleaning the q0957 frame in the red arm, i.e., re1cr → re1sc. Now the main FITS files of q0957 are be1sc and re1sc

4. *Fibre flux extraction*
   Based on the IRAF/SPECRED package

   ```
   cl> apall be1sc ref=bw out=be1.ms trace- recen-
   intera-
   cl> apall ba1 ref=bw trace- recen- intera-
   cl> apall be2cr ref=bw out=be2.ms trace- recen-
   intera-
   cl> apall ba2 ref=bw trace- recen- intera-
   ```

   The FITS file be1.ms (ba1.ms) contains the q0957 (Xe-lamp) spectrum in the blue arm for each fibre, while the file be2.ms (ba2.ms) includes the blue-arm feige34 (Xe-lamp) spectra. Using similar commands, it is also possible to extract specta in the red arm

5. *Wavelength calibration (dispersion solution)*
   Based on the Python script reident.py

   ```
   ./reident.py
   ```

   This program uses the idba0.ms and idra0.ms files (approximated solutions) in the `database` sub-subfolder. The lists of Xe emission lines are frodo_blue.dat (blue arm) and frodo_red.dat (red arm). Both lists are available in `l2lens`. It is necessary to run the Python script four times: `name = 'ba1','ba2','ra1'` and `'ra2'` in reident.py

6. *Throughput correction*
   Based on the Python script norm.py

   ```
   ./norm.py
   ```

   Run this script twice: `arm = 'b'` and `'r'` in norm.py. The normalized spectrum for each fibre can be found in the FITS files be1nr.ms (q0957/blue

arm), re1nr.ms (q0957/red arm), be2nr.ms (feige34/blue arm) and re2nr.ms (feige34/red arm)

7. *Spectral rebinning (dispersion correction)*
   Based on the Python script disp_cor.py

   ```
   ./disp_cor.py
   ```

   Run the script twice: `arm = 'b'` and `'r'` in disp_cor.py. This gives the final raw spectrum for each fibre: be1dc.ms (q0957/blue arm), re1dc.ms (q0957/red arm), be2dc.ms (feige34/blue arm) and re2dc.ms (feige34/red arm)

8. *Making spectral data cubes*
   Each data cube gives the 2D flux in the 12×12 fibre array at each wavelength pixel. We use the script rss_cube.py

   ```
   ./rss_cube.py
   ```

   Run the script four times: `inname = 'be1','be2','re1'` and `'re2'` in rss_cube.py. This produces the data cubes (FITS files) b1 (q0957/blue arm), r1 (q0957/red arm), b2 (feige34/blue arm) and r2 (feige34/red arm)

Flux-calibrated spectra of the lens system (Python 2.7)

1. *Feige 34: photometry on slices along the spectral axis*
   Based on the Python script m2free.py. Both spectral data cubes (b2 and r2) are split into 40 slices along the spectral axis. Each slice is fitted to a seven-parameter model, where the free parameters are: the centroid of the star, the FWHM along major and minor axis, the position angle of the major axis, the uncalibrated flux of the star, and the background sky level. We use the command line `./m2free.py directory x0 y0`, where `directory` refers to the subfolder containing the two data cubes, and `x0 y0` is an approximate stellar centroid, i.e.,

   ```
   ./m2free.py 110301 6 3
   ```

The b2.free and r2.free outputs show 40 solutions each (one per slice)

2. *Feige 34: polynomial fits to position-structure parameters*
   Based on the Python script m2fit.py. The five position-structure parameters are fitted to smooth polynomial functions of wavelength

   ```
   ./m2fit.py 110301
   ```

   The b2.fit and r2.fit outputs show 2001 solutions each (one per dispersion pixel)

3. *Feige 34: photometry on monochromatic frames*
   Based on the Python script m2fix.py. Each monochromatic frame is fitted to a two-parameter model. The five position-structure parameters are evaluated through smooth polynomial functions (b2.fit and r2.fit), leaving only the uncalibrated flux and the sky level as free parameters

   ```
   ./m2fix.py 110301
   ```

   The b2.fix and r2.fix outputs show 2001 (stellar flux, sky level) pairs each. These files also include the standard spectral response of FRODOSpec, which is based on the true spectrum of Feige 34 (f34a.oke in `l2lens`) and the standard atmospheric extinction curve at the Roque de los Muchachos Observatory (lam_extin.dat in `l2lens`). The optional script m2graph.py (use the command line `./m2graph.py 110301`) also allows the user to check b2.fix and r2.fix

4. *Lens system: photometry on slices along the spectral axis*
   Based on the Python script m1free.py. Both spectral data cubes (b1 and r1) are split into 40 slices along the spectral axis. Each slice is fitted to a ten-parameter model, where the free parameters are: the centroid of Q0957+561A, the FWHM along major and minor axis, the position angle of the major axis, the orientation of the frame, the uncalibrated total fluxes of the two quasar images (Q0957+561A and Q0957+561B) and the lensing galaxy, and the sky level. We use the command line `./m1free.py directory x0 y0`, where `directory` refers to the subfolder containing the two data cubes, and `x0 y0` is an approximate centroid of Q0957+561A, i.e.,

```
./m1free.py 110301 8 6.5
```

The b1.free and r1.free outputs show 40 solutions each (one per slice)

5. *Lens system: polynomial fits to position-structure parameters and estimation of orientations*
   Based on the Python script m1fit.py. The five position-structure parameters are fitted to smooth polynomial functions of wavelength, excluding the spectral edges. For each spectral arm, it is also obtained the median orientation for all wavelength slices

   ```
   ./m1fit.py 110301
   ```

   The b1.fit and r1.fit outputs show 2001 solutions each (one per dispersion pixel). Apart from position-structure data, the outputs contain the orientations of each arm

6. *Lens system: photometry on monochromatic frames*
   Based on the Python script m1fix.py. Each monochromatic frame is fitted to a four-parameter model. The five position-structure data and the frame orientation are taken from the b1.fit and r1.fit files. Thus, only the uncalibrated fluxes and the sky level are free parameters

   ```
   ./m1fix.py 110301
   ```

   The b1.fix and r1.fix outputs include 2001 (flux_A, flux_B, flux_G, sky level) vectors each

7. *Final spectra of the two quasar images*
   Based on the Python script m1graph.py

   ```
   ./m1graph.py
   ```

   The b.dat and r.dat files contain the flux-calibrated spectra of the two quasar images and the lens galaxy. The m1graph.py script also produces figures showing spectra