

PyCOS Tutorial

PyCOS is a spectral reduction pipeline for the Cosmic Origin spectrograph on the Hubble Space Telescope.

This document attempts to explain the work-flow for PyCOS

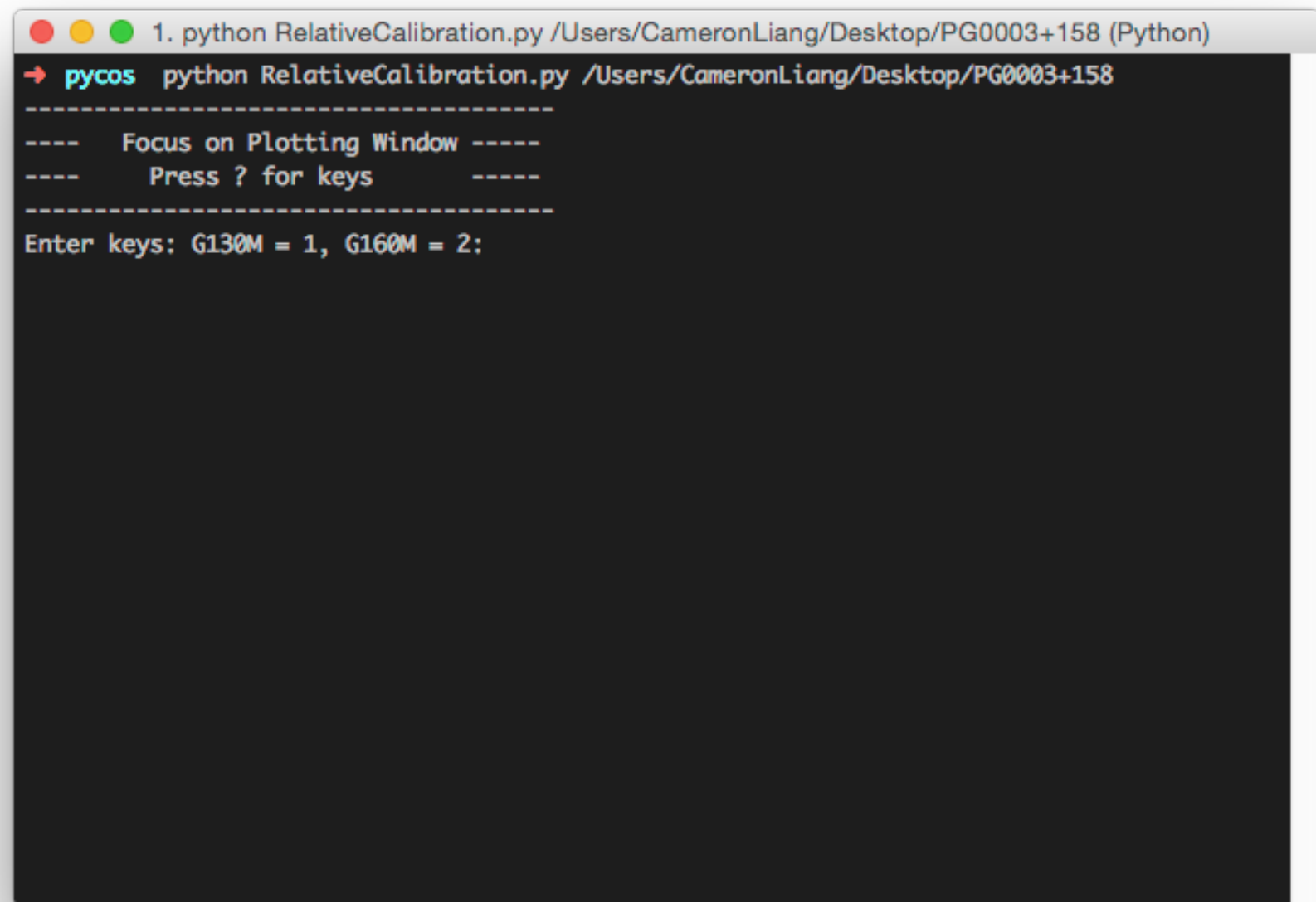
1. Process x1dsum fits files

- Download the raw data from HST archive. <https://archive.stsci.edu/> (PyCOS only requires intermediate data product with names “xxxxxx_x1dsum.fits. (See COS data handbook for details)
- Put all x1dsum.fits under a directory “raw”. I recommend .../object_name/raw/
 - e.g. /home/username/desktop/PKS0405-123/raw
- Run Process_x1dsum.py as followed:
 - Python Process_x1dsum.py full_path_to_so_name
- This will produce a collection of processed intermediate products output to .../object_name

2. Relative Calibration

- Run RelativeCalibration.py as followed:
 - `python RelativeCalibration.py full_path_to_object_directory`

Choose grating (G130M
or G160M) to align
between exposures

A terminal window with a dark background and light-colored text. The title bar at the top reads "1. python RelativeCalibration.py /Users/CameronLiang/Desktop/PG0003+158 (Python)". The first line of the command prompt shows a red arrow pointing to the word "pycos", followed by the command "python RelativeCalibration.py /Users/CameronLiang/Desktop/PG0003+158". Below this, there are several lines of text: "-----", "---- Focus on Plotting Window ----", "---- Press ? for keys ----", "-----", and "Enter keys: G130M = 1, G160M = 2:". The text is centered and uses a monospaced font.

```
1. python RelativeCalibration.py /Users/CameronLiang/Desktop/PG0003+158 (Python)
→ pycos python RelativeCalibration.py /Users/CameronLiang/Desktop/PG0003+158
-----
---- Focus on Plotting Window ----
---- Press ? for keys ----
-----
Enter keys: G130M = 1, G160M = 2:
```

2. Relative Calibration

Reference spectrum is automatically chosen and user to choose another spectrum to align.

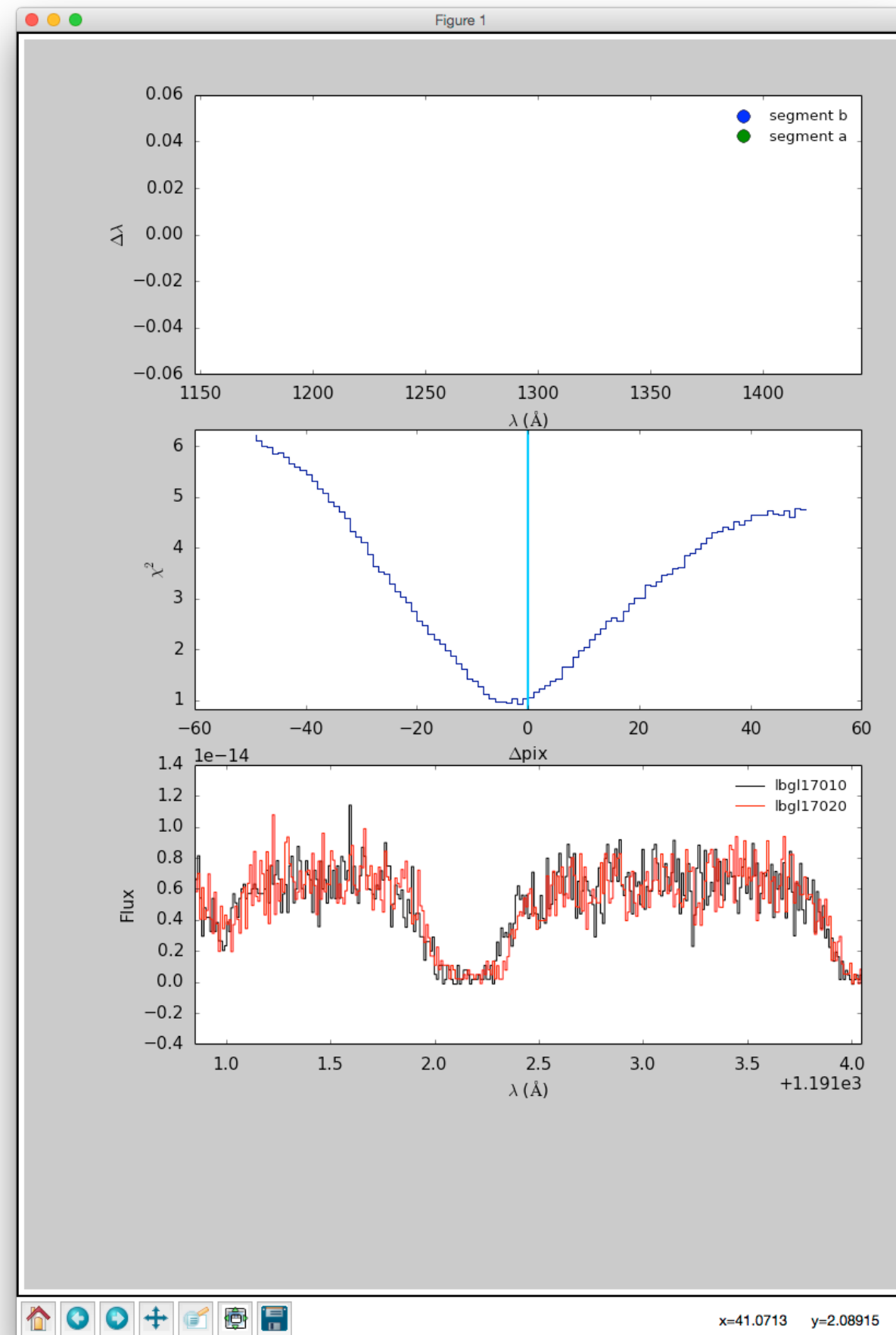
```
1. python RelativeCalibration.py /Users/CameronLiang/Desktop/PG0003+158 (Python)
→ pycos python RelativeCalibration.py /Users/CameronLiang/Desktop/PG0003+158
-----
---- Focus on Plotting Window ----
---- Press ? for keys ----
-----
Enter keys: G130M = 1, G160M = 2: 1
-----
Grating = G130M
-----
Files in /Users/CameronLiang/Desktop/PG0003+158/G130M_filelist:
1 = /Users/CameronLiang/Desktop/PG0003+158/lbgl17010.x1dsum
2 = /Users/CameronLiang/Desktop/PG0003+158/lbgl17020.x1dsum
3 = /Users/CameronLiang/Desktop/PG0003+158/lbgl17030.x1dsum
4 = /Users/CameronLiang/Desktop/PG0003+158/lbgl17040.x1dsum

reference file = 1
starting align file = 2_
```

2. Relative Calibration

A GUI interface will appear for user to align the spectrum (red) against the reference (black)

Focus on the plotting window, and press “?” on the keyboard - a list of keys mapping to some functions will print in terminal. Follow the key map to proceed for aligning, fitting, and writing a rectified spectrum.



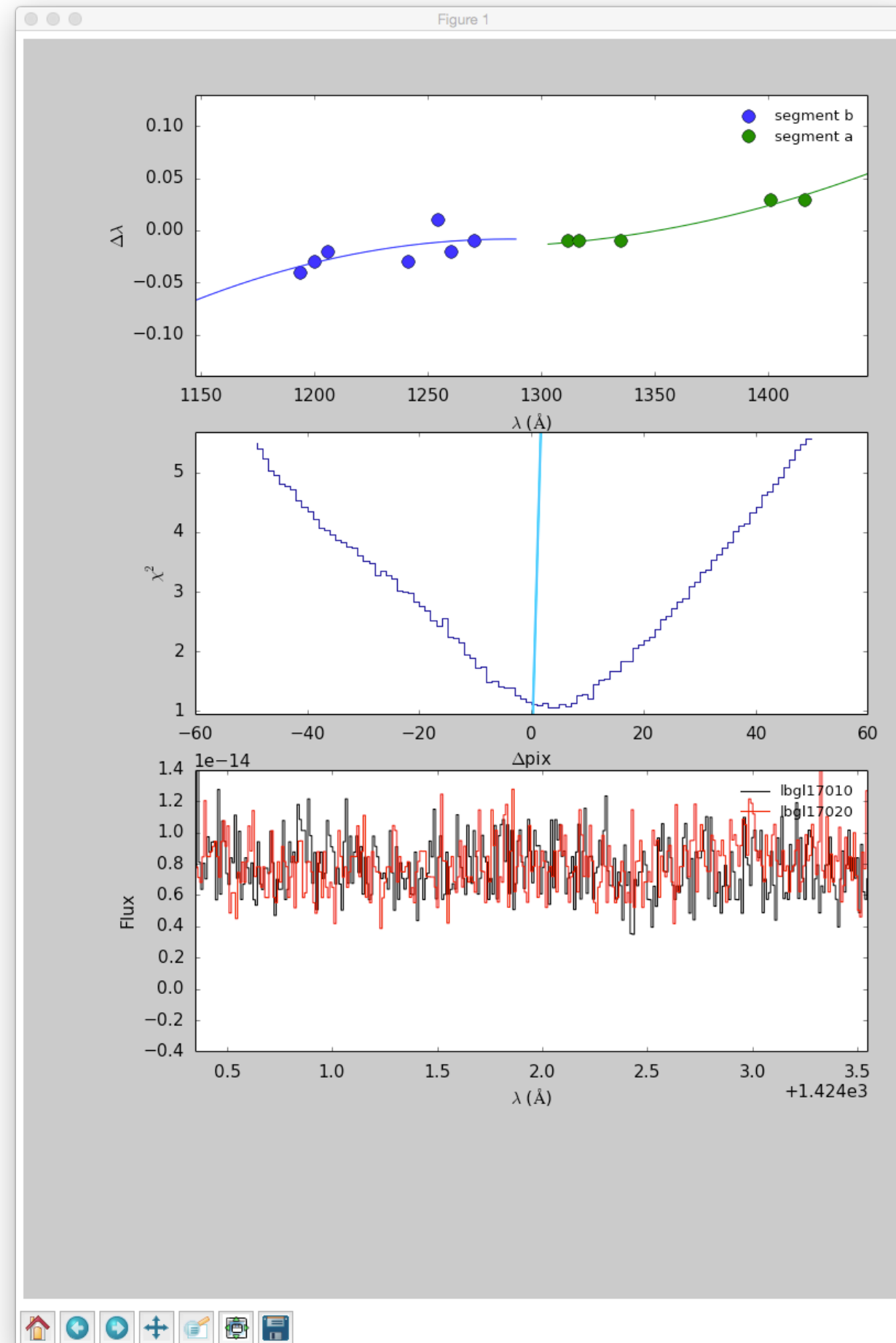
2. Relative Calibration

After a series of alignment between some absorption lines present in the data. There will be a list of points in the top panel. Users can delete points using according to the key map.

Segment a and b can be fitted independently with a polynomial of order n . We caution users not to unnecessarily high orders due to potential danger in extrapolation near both ends of the spectra.

After writing the new spectrum, users can exit the plotting window. The terminal will now ask whether to proceed to the next available spectra (if available). The process then repeats until no more spectra available.

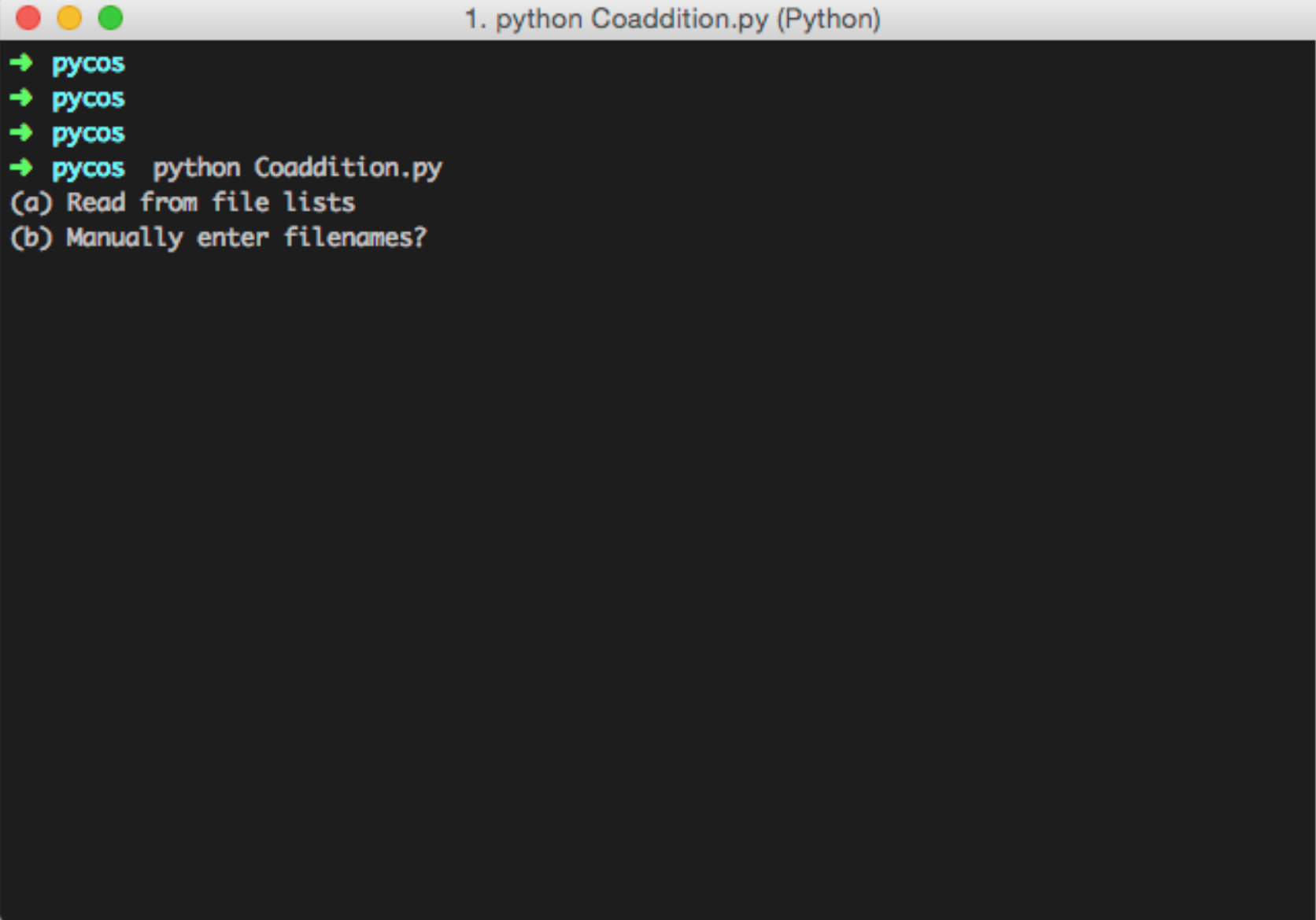
User can choose exit and come back to choose any spectra to align.



3. Co-addition

Run coaddition.py as below:
python Coaddition.py

First, we want to coadd
G130M spectra and G160M
spectra separately. Choose
the first option to read from a
file list.

A terminal window titled "1. python Coaddition.py (Python)" with a dark background and light-colored text. It shows a series of four green arrows pointing to the command "pycos", followed by the command "python Coaddition.py". The output of the script is displayed in a light color, showing two menu options: "(a) Read from file lists" and "(b) Manually enter filenames?".

```
1. python Coaddition.py (Python)
→ pycos
→ pycos
→ pycos
→ pycos python Coaddition.py
(a) Read from file lists
(b) Manually enter filenames?
```

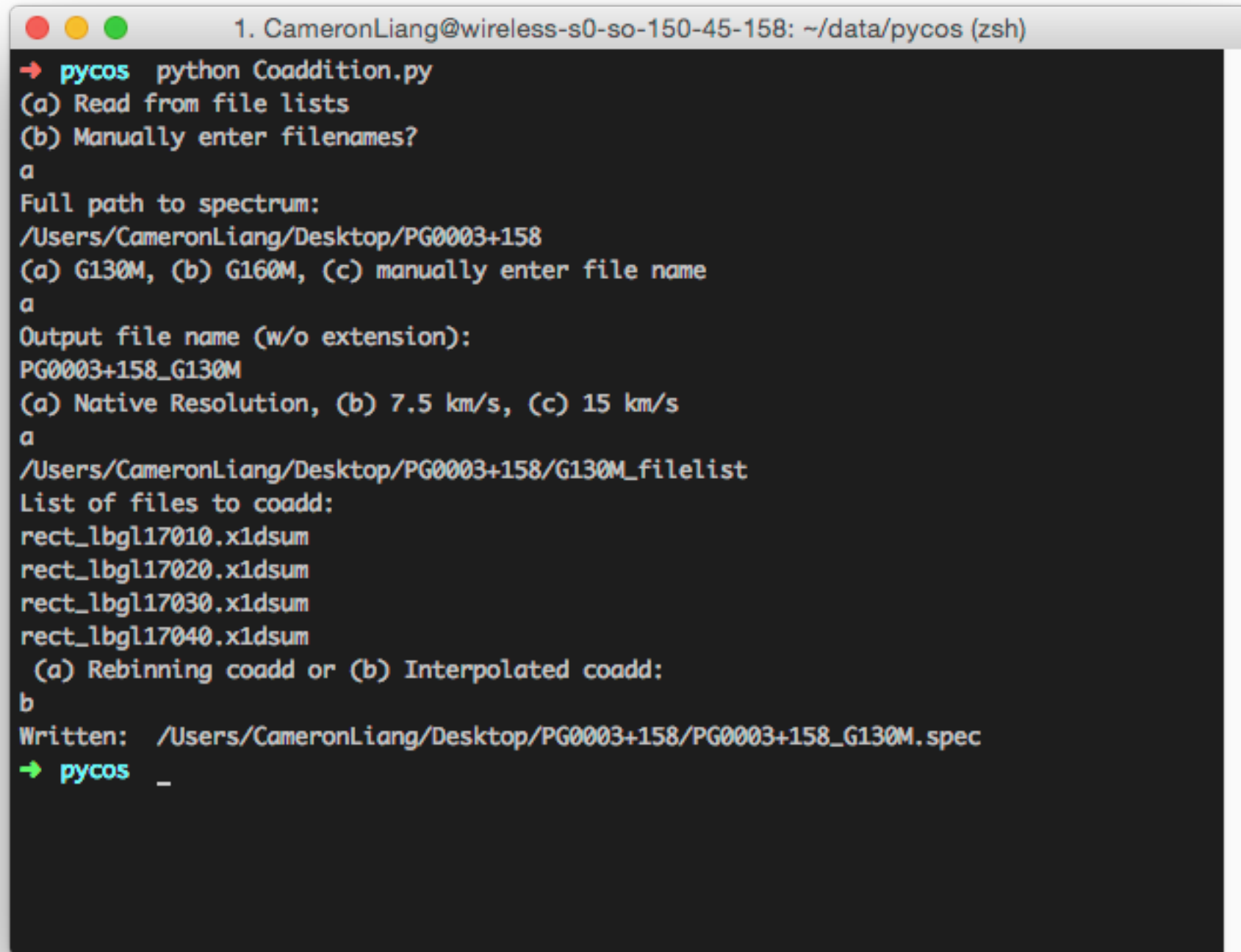
3. Co-addition

Run coaddition.py as below:
python Coaddition.py

First, we want to coadd
G130M spectra and G160M
spectra separately. Choose
the first option to read from a
file list (automatically
produced by
Process_x1dsum.py)

Then choose the resolution
of the co-added spectrum. I
recommend native
resolution, i.e., to use the
same wavelength grid as the
input.

Repeat the process for
G160M data if exists.

A terminal window titled "1. CameronLiang@wireless-s0-so-150-45-158: ~/data/pycos (zsh)" displays the execution of a Python script. The user enters "python Coaddition.py". The script prompts for input: "(a) Read from file lists" and "(b) Manually enter filenames?". The user selects 'a'. The script asks for the "Full path to spectrum:" and the user enters "/Users/CameronLiang/Desktop/PG0003+158". The script then offers "(a) G130M, (b) G160M, (c) manually enter file name". The user selects 'a'. The script prompts for the "Output file name (w/o extension):" and the user enters "PG0003+158_G130M". The script then offers "(a) Native Resolution, (b) 7.5 km/s, (c) 15 km/s". The user selects 'a'. The script displays the path "/Users/CameronLiang/Desktop/PG0003+158/G130M_filelist" and lists the files to be coadded: "rect_lbgl17010.x1dsum", "rect_lbgl17020.x1dsum", "rect_lbgl17030.x1dsum", and "rect_lbgl17040.x1dsum". The script then asks "(a) Rebinning coadd or (b) Interpolated coadd:". The user selects 'b'. The script finally writes the output to "/Users/CameronLiang/Desktop/PG0003+158/PG0003+158_G130M.spec". The prompt returns to "pycos _".

```
1. CameronLiang@wireless-s0-so-150-45-158: ~/data/pycos (zsh)
→ pycos python Coaddition.py
(a) Read from file lists
(b) Manually enter filenames?
a
Full path to spectrum:
/Users/CameronLiang/Desktop/PG0003+158
(a) G130M, (b) G160M, (c) manually enter file name
a
Output file name (w/o extension):
PG0003+158_G130M
(a) Native Resolution, (b) 7.5 km/s, (c) 15 km/s
a
/Users/CameronLiang/Desktop/PG0003+158/G130M_filelist
List of files to coadd:
rect_lbgl17010.x1dsum
rect_lbgl17020.x1dsum
rect_lbgl17030.x1dsum
rect_lbgl17040.x1dsum
(a) Rebinning coadd or (b) Interpolated coadd:
b
Written: /Users/CameronLiang/Desktop/PG0003+158/PG0003+158_G130M.spec
→ pycos _
```


4. Absolute Calibration

Run AbsoluteCalibration.py as below:

```
python AbsoluteCalibration.py
```

This module works by fitting gaussian to obtain the observed centroid of a given Milky-Way absorption lines to give a measure of absolute wavelength calibration. Then a polynomial fit is used to correct any elongation/compression of the wavelength array.

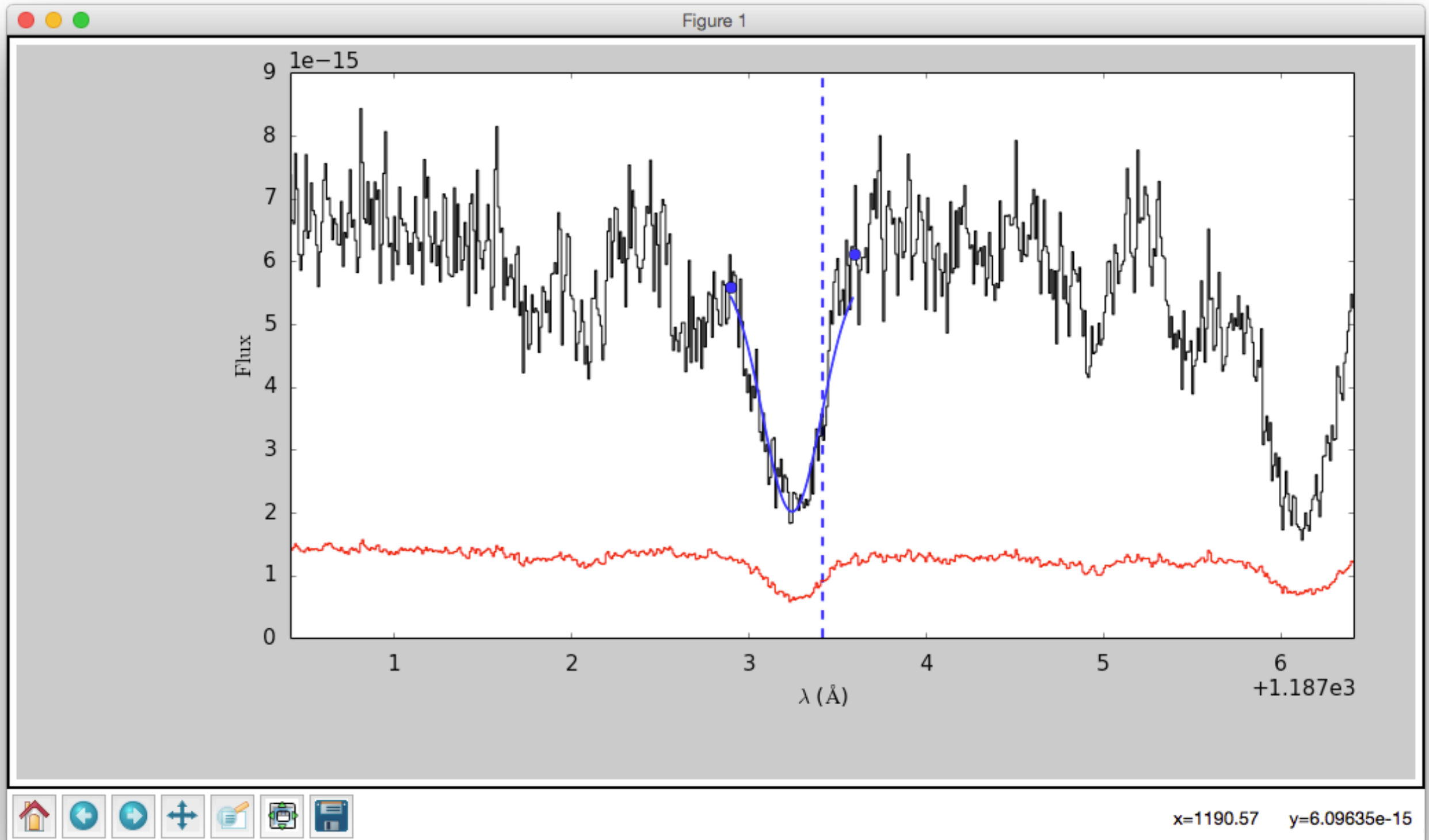
For the first time entering the program choose (f) for fitting a new line. Enter the name of the transition to fit.

```
1. python AbsoluteCalibration.py (Python)
→ pycos python AbsoluteCalibration.py
Full path to QSO spectrum directory:
/Users/CameronLiang/Desktop/PG0003+158
File name of spectrum:
PG0003+158_G130M.spec
(f)it new line, choose offset to (d)etele, (p)rint current offsets,(w)rite new spec, (e)xit program:
f
-----
High ions:
NeVIIIa 780.324 | NVb 1242.804 |
NeVIIIb 770.409 | SiIVa 1393.76018 |
OVIa 1031.9261 | SiIVb 1402.77291 |
OVIb 1037.6167 | CIVa 1548.2049 |
NVa 1238.821 | CIVb 1550.77845 |
-----
Low ions:
CIII 977.0201 | SiIIc 1259.519 |
FeIIa 1144.9379 | SiIIc 1260.4221 |
SiIIa 1190.4158 | OI 1302.1685 |
SiIIb 1193.2897 | SiIIId 1304.3702 |
NIa 1199.5496 | CII 1334.5323 |
NIb 1200.2233 | CII* 1335.7077 |
NIc 1200.7098 | SiIIe 1526.70698 |
SiIII 1206.5000 | FeIIb 1608.45085 |
SiIIa 1250.584 | AlII 1670.7886 |
SiIIb 1253.811 | NiII 1703.4119 |
-----
Warning: be careful for the choice of MW lines for the use of absolute calibration.
Wavelength range of data = [1132.698624 1459.776493]
i.e., do not use lines outside of the range.

Transition Name: SiIIa
_
```

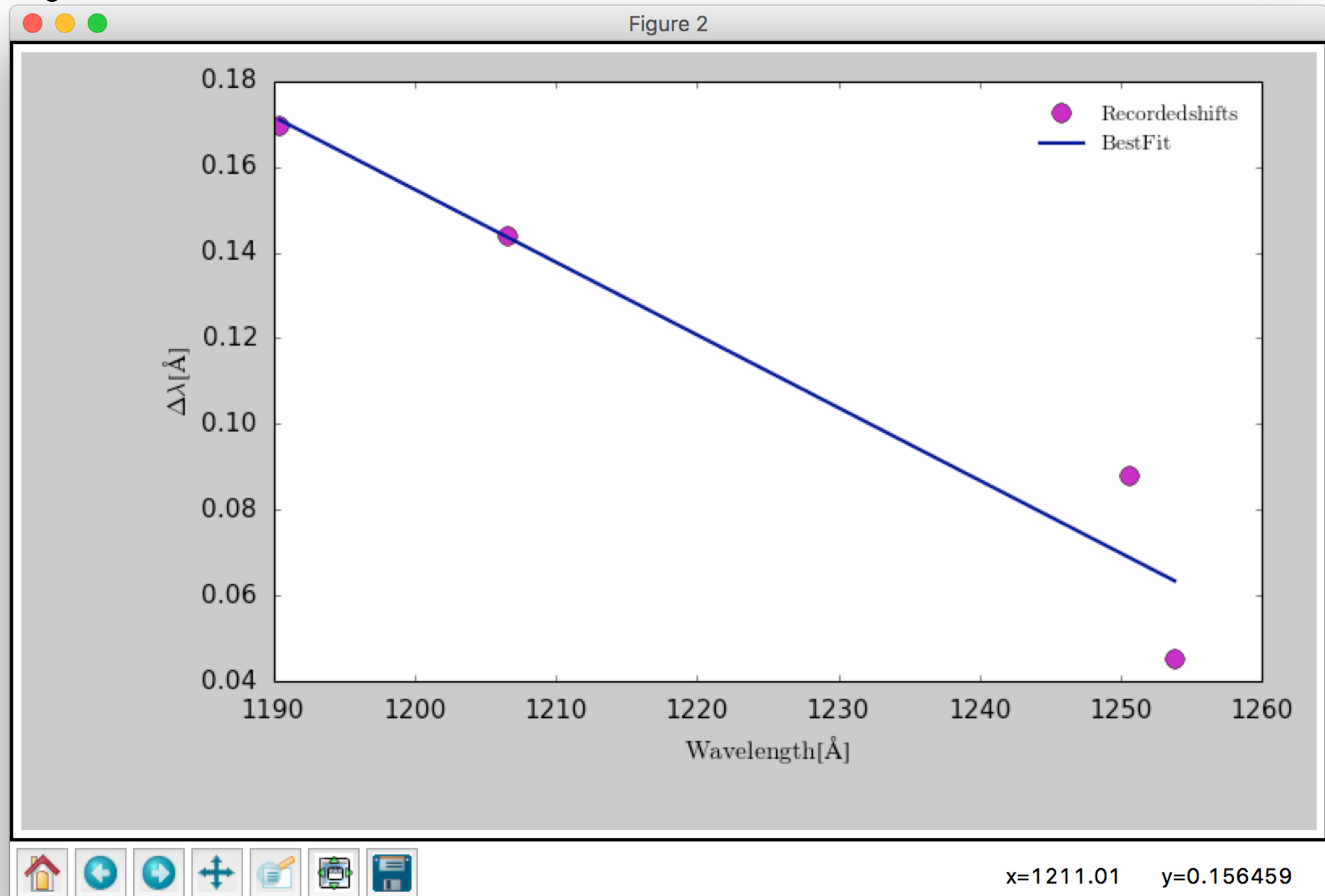
4. Absolute Calibration

A new window will appear. For a key map, press “?” when focusing on the plotting window. Press “a” to add two points then “shift+g” to fit a gaussian. One can repeat the gaussian fitting process by simply choosing different spectral region to add two points with “a” and then “shift+g” again. The fitted centroid is automatically recorded. After “shift+g”, user can close the plotting window.



4. Absolute Calibration

During and after fitting gaussian to Milky-Way lines, user can check if there are any outliers or the best polynomial fit for the current offsets recorded. Press “p” for printing current recorded gaussian fits, and choose the order of polynomial fit for the data (see example of linear fit below). User can delete any of the points. If user decides to write a new spectrum (after inspecting the fit), press “w” to write and choose the same polynomial order. Then repeat the process for the spectrum of the grating.



In principle, that's it. There will be two spectra (G130M and G160M grating). If one desire to combines the two, use Coaddition.py again to combine.