Grizli Q&A

General questions

1. Is grizli using the same grismconf as Nor's code (e.g. does it import https://github.com/npirzkal/GRISMCONF)?
    a. Ans: It is using an earlier version, still based on the aXe approach (detailed in the 2014 TIR)
2. How hard might it be to segregate all FITS and FITS-WCS dependencies into separate I/O routines (as a step towards moving over to ASDF + gWCS)?
    a. Ans: FITS dependencies are probably pretty widespread in the code
3. Is EAZY a strict dependency?
    a. Ans: This is probably less intricately woven into the code than FITS
4. Looks like it's using pysynphot instead of synphot_refactor. Have you looked at updating?
    a. Ans: Yes; just using this for computing fluxes in direct images, so may not be a big deal to update

Model based extraction

1. How does grizli specify parameters to vary?
2. What is the merit-function for the fit?
    a. Ans: chisq
3. How are the data weighted ?
    a. Ans: Standard lsq. But can add in additional down weighting for contamination
4. What optimizer is used?
5. How are uncertainties calculated?
    a. Ans: detector noise model (straightforward because we are in FLT frames)
    b. Drawback – can be slow if there are many exposures (e.g. FIGS)
6. Which modules/functions/classes are doing the fitting?
    a. Most of the work is one in multifit

Ans:

- Turns input 1D spectrum & flux into 2D dispersed using object morphology
- LSQ fitting scaling the models (template library)
- multifit is doing the chisq minimization
- Current approach – generate static contamination model (first starting with flat spectrum, then polynomials)
- Start with brightest object, subtract fainter ones, fit polynomial, then iterate to fainter objects
- For now, stop there for contamination – could imagine going to the next step of iterating all the fitted spectra
- Possible to simultaneously fit in cutouts with multiple objects

GrismDisperser

- In model.GrismDisperser, do compute_model and compute_model_psf correspond to pixel- and object-based dispersion, respectively?
    - Ans: the compute_model_psf is a module to use an "effective psf" to estimate the flux in each pixel in the dispersed image

model.py

- Seems to have a lot of hard-coded instrument specifics. Is that common elsewhere in the code?
    - Ans: probably mostly confined to grismconf and model.py
    - maybe also some in multifit

Approach for JWST – just rotated 90 degrees the cross-dispersed images.