



STScI | SPACE TELESCOPE
SCIENCE INSTITUTE

Linear Reconstruction of Grism Spectroscopy

I. Simulation and Extraction Examples

R. E. Ryan, S. Casertano, & N. Pirzkal

October 2, 2018

ABSTRACT

*We present simulations and extraction of slitless spectroscopy from WFC3/IR G102 and G141 aimed at mitigating the effect of overlapping spectral traces using **LINEAR**. The IDL scripts provided here generate the data used for various figures presented by Ryan, Casertano, & Pirzkal (2018). We also discuss the basic file formats for **LINEAR** and mention a few common problems that may plague users. We hope this to be a “living document” that evolves as the code matures or new instruments (such as those for JWST) come on line.*

1. Introduction

Slitless spectroscopy uses a transmissive and dispersive optical element to produce a complete spectroscopic view of an astronomical scene, free from the selection biases often present in other forms of multiplexed spectroscopy (such as slit masks). Traditional methods for extracting and analyzing slitless spectroscopy rely on the concept of “contamination,” which describes the flux associated with unrelated, neighboring sources. Recently, we presented a new algorithm for using observations at many orientations to determine the optimal, one-dimensional spectrum (Ryan, Casertano, & Pirzkal 2018) that obviates the need for contamination corrections. In that work, we developed the formalism, tested the method with simulated data, explored observations with Wide-Field Camera 3 (WFC3), and mused about possible future applications, limitations, and improvements. However that publication is rather spartan on programmatic details and/or usage of our codes, and so we intend for this Report to bridge that gap.

We refer the reader to Ryan, Casertano, & Pirzkal (2018) for a detailed description of the algorithm, however we will simply highlight a few concepts that are relevant. `LINEAR` uses a direct image and segmentation map of the astronomical scene as a “finding chart” that defines the spectral extraction apertures. The code reprojects each pixel belonging to each object on to each grism image at each wavelength of interest, and computes various weighting factors (pixel overlap areas, wavelength-dependent flat field, pixel-area map, etc.). These weights are summed for each source, while preserving the unique wavelength and grism-image pixel coordinate¹. We refer to the collection of weights (for a given dispersion order) as an *object-dispersion table* (ODT). These weights are formed into a matrix (W) that acts as a linear operator to transform the incident spectra (in physical f_λ units) into a pixelated dispersed image (in units of $e^- s^{-1}$). Now the process of spectral extraction can be written as a least-squares problem:

$$\arg \min_f (\|W f - \mathcal{G}\|_2^2 + \ell \|W\|_F^2 \|f\|_2^2), \quad (1)$$

where \mathcal{G} are the grism images, f are the set of spectra, $\|W\|_F^2$ is the Frobenius norm² of W , and ℓ is a regularization parameter (discussed more below). As we noted in Ryan, Casertano, & Pirzkal (2018), we deviate from usual presentations of regularized least-squares by including the Frobenius norm simply so that the regularization term (ℓ) is dimensionless.

We provide functionality to mask pixels associated with higher-order dispersions with an *object-mask table* (OMT) that we generate with a similar procedure to the ODTs, except we treat each object as a convex hull, which dramatically decreases the computational expense. Since the convex hull is generally larger than the segmentation region for the typical source, this approach is quite conservative. In Figure 1, we compare the differences between treating each source as a convex hull (thick black line) or the collection of pixels (each source is shown by a different color).

The software will write these tables to disk, following customized data format, so that they can be efficiently read. With these tables in hand, there are many calculations that can be readily performed:

extraction: this was the primary goal of the effort, and we expect the primary interest;

simulation: with a model spectrum of each source, it is possible to simulate a grism image;

¹This summation over source pixels is required only to limit the data volume and memory usage. In future work, this step could be relaxed with either larger computing resources or smaller-scale datasets.

²The Frobenius norm is defined as $\|W\|_F^2 = \sum_{i,j} w_{i,j}^2$.

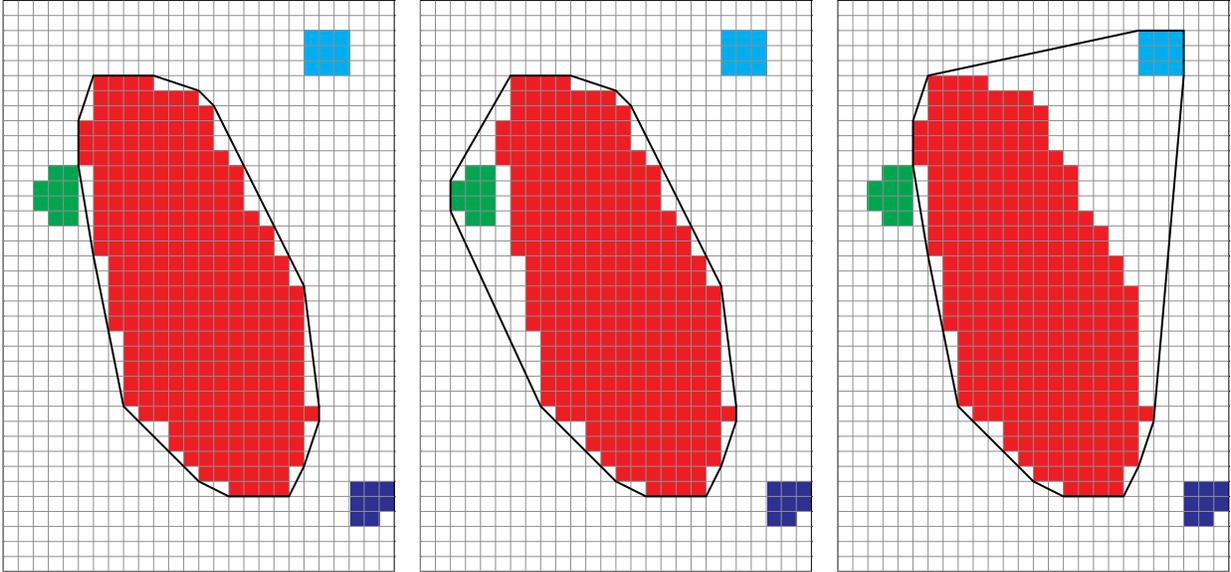


Fig. 1.— Illustration of convex hull. In the left panel, we show the simple conversion of a set of pixels (each source is shown by a different color) to a convex hull (thick black line) to outline a source. Clearly, the convex hull contains more area than its parent set of pixels, hence we feel that using a convex hull for the source masking is rather conservative. However, this object is nearly convex, and so middle panel illustrates the convex hull procedure for a significantly concave object that we create by merging two sources (green and red sources). In this case, the method includes many pixels unrelated to either source would be included. This is of course exacerbated for a *pixel islands*, such as in the last panel, where we group to wholly separated sources (cyan and red sources).

masks: create regions files for `ds9` to identify regions where the spectral traces from different sources overlap (*i.e.* the contamination);

cutouts: excise stamps for a given source and beam; and

regions: produce a `ds9`-like regions file indicating the locations of spectral traces for given object and beams.

This list is not meant to be exhaustive, but rather illustrative of the value of the ODT philosophy.

In this Report, we describe a few example usages of the code, some of which were presented by Ryan, Casertano, & Pirzkal (2018). This report is organized as: Section 2 describes the installation and basic functionality of `LINEAR`, Section 3 presents several examples of the code, Section 4 highlights a few common failures we have identified,

Section 5 provides a brief summary, and Section 6 catalogs the revisions. In this report (and indeed `LINEAR` assumes as well), all magnitudes are quoted in the AB system (Oke & Gun 1983).

2. `LINEAR`: A Novel Algorithm for Grism Spectroscopy

2.1. Installation

`LINEAR` is predominately written in IDL, and so the installation is generally straightforward. The User is expected to download the source code, which can be stored in a single directory. Assuming the IDL environment is properly set up, then installing `LINEAR` is little more than setting a few environment variables that point the the IDL and C files. We recommend the following settings:

```
setenv LINEAR /Users/rryan/LINEAR/  
setenv LINEAR_SRC /Users/rryan/Dropbox/GRISM_STUFF/IDL_PRO/LINEAR/src/  
setenv LINEAR_CONFIG ${LINEAR}config  
setenv LINEAR_FILTER ${LINEAR}filters  
setenv IDL_PATH +${LINEAR_SRC}:${IDL_PATH}
```

in the `.cshrc` file. Finally, the log-in file must be loaded by either `source .cshrc` or by restarting the terminal. We expect other shells may have similar settings. In this way, the software resides in a directory, but can be run from any number of directories and/or other versions can be executed by simply altering the relevant variable(s).

There are a few components that are written in C threaded with OpenMP, however the main IDL codes will automatically compile and link to the C libraries. In the event the C cannot be successfully linked, `LINEAR` will default to IDL implementations — at the expense of computational performance. We expect to include additional components in the configuration file (discussed below) to control the compilation of the C code: forcing to the IDL only or controlling settings with the C-compiler or OpenMP.

2.2. Basic Operations

In Table 1, we illustrate the configuration file that drives the execution of `LINEAR`, and give a brief description of each keyword. We also indicate the default variables and whether

they are mandatory³. The main wrapper procedure (`linear.pro`) can accept an arbitrary number of additional keywords, and so one can control each of these parameters from the calling sequence. For example,

```
IDL> linear, 'linear.config', OUTPUT='test.fits'
```

will override the setting for `OUTPUT` that is supplied in the file `linear.config` with `test.fits`. There are several files that control the input/output of `LINEAR`, here are representative examples:

`OBSFILE`: describes the observed direct images

```
# An example  OBSFILE
# 1: full path to image
# 2: full path to transmission curve
# 3: AB magnitude zeropoint
b.fits b_filter.dat  25.3
v.fits v_filter.dat  25.8
```

`SEDLST`: describes the model SEDs for each object (for image simulation)

```
# An example  SEDLST
# 1: source ID number as indicated in the segmentation map
# 2: full path to the SED text file
# 3: redshift to apply to the SED
# 4: AB magnitude to normalize the SED (thru first filter in OBSFILE)
1   sed1.dat  0.2  25.4
3   sed10.dat 5.6  28.4
10  a0v.dat   0.0  15.4
```

³Some of the variables are only mandatory in certain circumstances. For example, it is not necessary to set variables that control the extraction if the mode is to simulate.

WCSLST: describes the model WCS (for image simulation)

```
# An example WCSLST
# 1: image root name (for HST is the IPPPSS00T)
# 2: CRVAL1 (in degrees)
# 3: CRVAL2 (in degrees)
# 4: ORIENTAT (angle East of North)
imageone 53.22331455 -27.3300304 0.0
two      183.22331455 +87.33004 10.0
1        3.225          0.0300304 190.0
```

IMGLST: describes the grism images for extraction

```
# An example IMGLST
# 1: image file name
imageoneflt.fits.gz
1flt.fits.gz
hello_word1flt.fits
```

The SED file is a two column file of wavelength (in angstroms) and flux density (f_λ units). Similarly, the filter file is a two column file of wavelength (in angstroms) and the transmission (arbitrary units).

2.3. Dependencies

We have taken care to package the IDL/C programs that are used by **LINEAR**. But for reference, these are described in Table 2.

3. Examples

In the following subsections, we provide our basic IDL scripts used to generate the data for Ryan, Casertano, & Pirzkal (2018). We color code the components for clarity (bold black: IDL routines written as part of **LINEAR**, red: comments, purple: reserved IDL keywords, blue: procedure/function name, gray: native IDL routines, green: optional keywords, brown: string variables, and orange: routines from the IDL AstroLib (Landsman 1993)).

3.1. Systematics in the Astrometry (Simulation/Extraction)

Grism exposures generally suffer from the same pointing uncertainties that plague classic direct imaging (**example**), which must be remedied if sensible results are expected. To address this, most grism programs take a relatively short direct image⁴ (~ 100 s; referred to as *pre-imaging*), which provides a sampling of bright point sources. These pre-imaging catalogs are used to adjust the relative astrometry of the grism images, and/or tie to a larger reference dataset. This procedure naturally raises the question: “*What is the systematic uncertainty introduced by inaccurately registered grism exposures?*” We addressed this concern in Ryan, Casertano, & Pirzkal (2018), and provide the basic script using LINEAR to simulate the magnitude of this uncertainty in Listing 1.

```
1 pro one_astro, sig, NITER=niter
2
3 ;set number of iterations
4 if n_elements(niter) eq 0 then niter=100L
5
6 fmt='(I05)' ;formatting character
7 damp=1.0 ;set damping
8
9 ;make grism iamges
10 linear, 'linear.config', mode='simulate', remake=1b
11
12 ;read images and unzip them --- preserves old copies
13 readcol, 'wcs.lst', root, ra, dec, ang, format='a,d,d,f', /silent, count=nimg
14 new=root+'_flt_new.fits' ;new filename (unzipped)
15 new_gz=new+'.gz' ;new filename (zipped)
16 old=root+'_flt.fits.gz' ;original filename (zipped)
17 out=strtrim(damp,2)+''.dat' ;output file of spectra moments
18
19 ;print the filenames to a file, these are files to extract
20 fltlist='flt_extract.lst' ;list of FLTs to extract
21 forprint, new_gz, textout=fltlist, /nocomment
22
23 for i=0, niter-1 do begin ;loop over iterations
24   for j=0, nimg-1 do begin ;process each image
25     ;copy and unpack the file
26     file_copy, old[j], new_gz[j], /overwrite
27     file_gunzip, new_gz[j], /delete
28
29     ;read the header, and tweak the astrometry
30     h=headfits(new[j], ext=1)
31     getrot, h, rot, cdelt
32     pix=abs(cdelt) ;pixscl
33     dcrval=randomn(seed, 2, /double)*sig*pix ;deg
34     crval=sxpar(h, 'CRVAL*')+dcrval ;deg
35     sxaddpar, h, 'CRVAL1', crval[0] ;update the header
36     sxaddpar, h, 'CRVAL2', crval[1] ;update the header
37
38     modfits, new[j], 0, h, exten_no=1
```

⁴Although Bohlin, Deustua, & Pirzkal (2015) present an alternative method based on the properties of the 0th order dispersions, we do not consider such options.

```

39     file_gzip,new[j],/delete
40     endfor
41
42     ;create filename for output, and run linear
43     out='RESULTS/'+string(i,format=fmt)+'.fits'
44     linear,'linear.config',mode='extract',fltlist=fltlist,$
45         output=out,damping=damp,remake=0b
46
47     ;clean up space and read the results
48     file_delete,new_gz
49     t=mrdfits(out,1,/silent)
50     if i eq 0 then begin
51         nlam=n_elements(t)
52         flam=fltarr(nlam,niter)
53         lam=t.lam
54     endif
55     flam[:,i]=t.flam
56 endfor
57
58 ;compute moments over iteration number
59 std=stddev(flam,dimension=2)
60 ave=mean(flam,dimension=2)
61
62 ;print results to a file
63 openw,lun,out,/get_lun
64 printf,lun,'# damp = '+strtrim(damp,2)
65 printf,lun,'# niter = '+strtrim(niter,2)
66 printf,lun,'# sig = '+strtrim(sig,2)
67 for i=0,nlam-1 do printf,lun,lam[i],ave[i],std[i]
68 free_lun,lun
69 end
70
71 pro astro
72     sigma=1. ;RMS of the pointing (in native pixels)
73     one_astro,sigma
74 end

```

Listing 1 Astrometric test script (astro.pro).

3.2. Systematics in the Sky Background Subtraction (Simulation/Extraction)

The classic method for subtracting sky background in spectroscopy of using regions adjacent to the trace generally does not suffice for grism spectroscopy as the traces from nearby sources pollute the background estimate. Therefore the subtraction is better performed using a *master-sky image* (e.g. Pirzkal *et al.* 2004), which may be broken into multiple spectral components (such as geocoronal He I, zodiacal, and scattered light emission; Brammer *et al.* 2014). Applying these background estimates requires special care, particularly if one of the spectral components varies with time (as is the case for the He I; Brammer, Ryan, & Pirzkal 2015). These procedures generally reduce the typical background from $\sim 5 \text{ e}^- \text{ s}^{-1}$ to $\sim 0.1 \text{ e}^- \text{ s}^{-1}$, and with a column-averaged correction we

expect further refinement to $\sim 0.01 \text{ e}^- \text{ s}^{-1}$. However this residual is still brighter than what WFC3/IR is generally capable of achieving, which suggests it is important to consider ways of further reducing this systematic and/or assessing its effect on the one-dimensional spectra.

In Ryan, Casertano, & Pirzkal (2018), we simulated the effect of this residual by estimating the systematic uncertainty on the extracted spectra. In Listing 2, we show the script to use LINEAR to establish this systematic uncertainty.

```

1  pro one_back, sig, NITER=niter
2
3      ;get number of iterations
4  if n_elements(niter) eq 0 then niter=1001
5  outdir='RESULTS/'           ;output directory
6  fmt='(I05)'                 ;formatting character
7
8      ;ensure the directory exists
9  if ~file_test(outdir) then file_mkdir, outdir
10
11     ;default damping
12     damp=1.
13
14     ;run LINEAR to simulate the background-less images
15     linear, 'linear.config', mode='simulate', remake=1b
16
17     ;read the astrometry
18     readcol, 'wcs.lst', root, ra, dec, ang, f='a,d,d,f', /silent, count=nimg
19     g=where(strmid(root,0,1) ne '#', nimg)
20     root=root[g]
21     ra=ra[g]
22     dec=dec[g]
23     ang=ang[g]
24
25     ;new output file names
26     new=root+'_flt_new.fits'
27     new_gz=new+'.gz'
28     old=root+'_flt.fits.gz'
29
30     ;make a new file list for extraction
31     forprint, new_gz, textout='flt_extract.lst', /nocomment
32
33     for i=0, niter-1 do begin           ;process for each iteration
34         for j=0, nimg-1 do begin       ;process each image
35             ;get and unzip the original file
36             file_copy, old[j], new_gz[j], /overwrite
37             file_gunzip, new_gz[j], /delete
38
39             ;read the new image
40             img=readfits(new[j], hdr, extension=1, /silent)
41
42             ;create a random background image
43             sz=size(img, /dimension)
44             back=randomn(seed)*sig
45             img+=back                 ;add it back
46
47             ;update the file
48             modfits, new[j], img, hdr, exten_no=1

```

```

49     file_gzip,new[j],/delete
50     endfor
51
52     ;file name for output
53     out='RESULTS/'+string(i,format=fmt)+'.fits'
54
55     ;extract the simulated images (w/ background tweak)
56     linear,'linear.config',mode='extract',fltlst='flt_extract.lst',$
57         output=out,damping=damp,remake=1b
58
59     ;clean up temporary files
60     file_delete,new_gz
61
62     ;read these results to store in data structure
63     t=mrdfits(out,1,/silent)
64     if i eq 0 then begin
65         nlam=n_elements(t)
66         flam=fltarr(nlam,niter)
67         lam=t.lam
68     endif
69     flam[* ,i]=t.flam           ;keep the results
70 endfor
71
72 ;compute moments of the flux
73 std=stddev(flam,dimension=2)
74 ave=mean(flam,dimension=2)
75
76 ;output filename
77 outfile=string(sig,format='(F5.3)')+ '_'+strtrim(damp,2)+'.dat'
78
79 ;output the results
80 openw,lun,outfile,/get_lun
81 printf,lun,'# damp = '+strtrim(damp,2)
82 printf,lun,'# niter = '+strtrim(niter,2)
83 printf,lun,'# sig = '+strtrim(sig,2)
84 for i=0,nlam-1 do printf,lun,lam[i],ave[i],std[i]
85 free_lun,lun           ;close the file
86 end
87
88 pro back
89     sigma=1.0           ;RMS of background tweakx
90     one_back,sigma,NITER=100L
91 end

```

Listing 2 Background-subtraction test script (back.pro).

3.3. Optimizing the Damping with G141 (Extraction)

As suggested in Section 1, the fidelity of the reconstructed spectra subtly depends on the regularization (or damping) parameter ℓ , in Equation 1. Therefore it is crucial to establish a method for estimating the ideal value for ℓ and/or validating the choices made. Hansen (1992) proposed a heuristic argument for optimizing the damping term as a balance between fitting the data (the first term in Equation 1) and the regularization

or smoothness (the second term in Equation 1). In this scheme, a plot of $\|Wf - \mathcal{G}\|_2^2$ versus $\|f\|_2^2$ that is parameterized by ℓ generally takes the shape of a capital L (so this plot is often referred to as the “L-curve”). The point of maximum curvature (which is the vertex of the L) represents a compromise between the two terms. To illustrate the L-curve analysis, Ryan, Casertano, & Pirzkal (2018) extracted the one-dimensional spectra of the Hubble Ultra-Deep Field (HUDF; Beckwith *et al.* 2006).

The HUDF data were collected by programs 12099 and 12177 in three distinct orients for 36 exposures⁵ with a combined exposure time of 43.3 ks. We correct the grism-image astrometry with standard techniques and remove the two-component, time-varying sky background following Pirzkal & Ryan (2017). We define extraction apertures using Source Extractor (SExtractor; Bertin & Arnouts 1996) on the stacked F140W image, which was referenced to the original HUDF imaging with ACS (Beckwith *et al.* 2006). With the detector parameters and the astrophysical scene, we generated the ODTs, then iterated through various damping choices (see Listing 3) and recorded the various terms of Equation 1 (which are saved in the primary header-data unit) and corresponding one-dimensional spectra. In Figure 2, we reproduce Figure 8 of Ryan, Casertano, & Pirzkal (2018) that shows the resulting L-curve as well as the optimized damping value (ℓ_{opt} — as derived from the inset figure).

As a final note, we employed a “brute force” search for ℓ_{opt} for simplicity, however Cultrera & Callegaro (2016) present an iterative scheme that accelerates this process. We do not provide utility for such approaches in the current codebase.

```

1 pro optdamp,configfile
2   if size(configfile,/type) ne 7 then configfile='linear.config'
3
4   psep=path_sep()
5
6   ;directory for the output
7   dir='optimized3'+psep
8   if ~file_test(dir,/directory) then file_mkdir,dir ;make the dir
9
10  ;read the configuration file from LINEAR
11  conf=config_read(configfile, _EXTRA=_extra)
12
13  ;initialize the LINEAR code base. LINEAR is object-oriented
14  code=obj_new('linear')
15
16  ;load the grism config
17  pamname=(tag_exist(conf, 'pamname')?conf.pamname:'none')
18  ffname=(tag_exist(conf, 'ffname')?conf.ffname:'none')
19  code->load_grismconf, conf.axeconf, /silent, $
20                                     FNAME=ffname, PAMNAME=pamname

```

⁵We omit several exposures with anomalous sky background estimates (see Ryan, Casertano, & Pirzkal 2018 for more discussion).

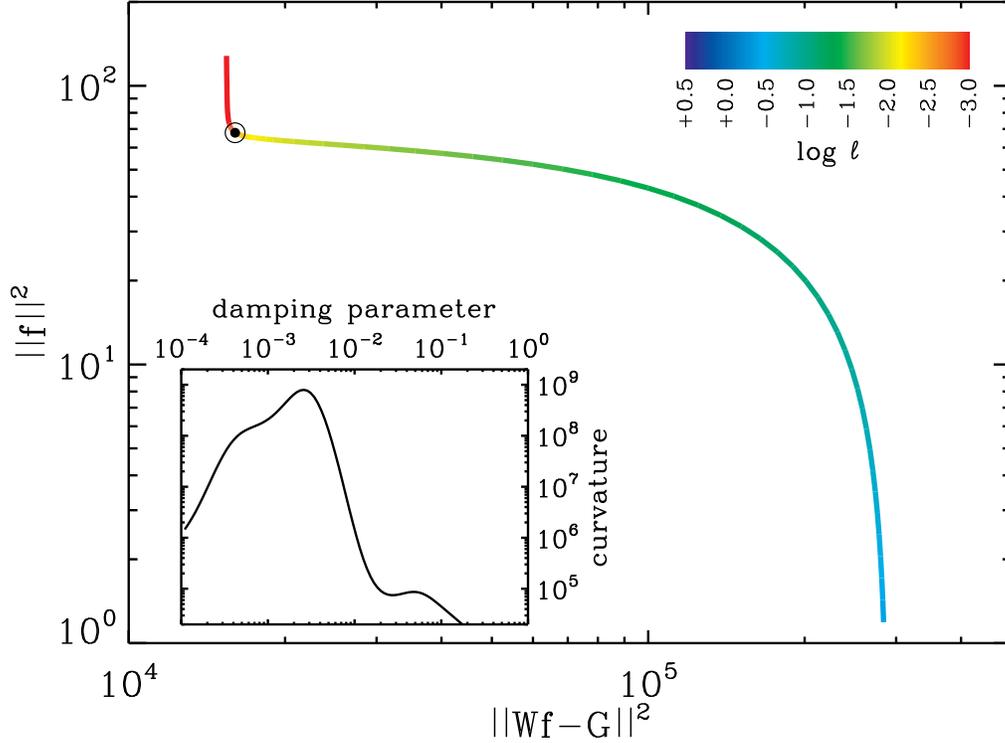


Fig. 2.— Optimization of the L-curve taken from Ryan, Casertano, & Pirzkal (2018). Here the plot of $\|WF - \mathcal{G}\|^2$ and $\|F\|^2$, parameterized by ℓ (shown in rainbow), provides a means of identifying where there is a trade-off between minimizing the fluctuations in the classic χ^2 term and resulting spectra, respectively. The optimal value is taken as the point that maximizes the curvature, and can be readily identified in the inset figure. The data shown here were derived from the G141 spectroscopy of the HUDF observations (see Ryan, Casertano, & Pirzkal 2018 for more details).

```

21
22 ;load the grism images
23 code->load_images,conf.imglst,/flush
24
25 ;load the sources
26 maglimit=tag_exist(conf,'maglimit'?conf.maglimit:0
27 code->load_sources,conf.segmap,conf.obsfile,odtdir=conf.tabledir,$
28             MAGLIMIT=maglimit
29
30
31 ;make a directory
32 if strcmp(conf.tabledir,strlen(conf.tabledir)-1,1) ne psep then $
33   conf.tabledir+=psep
34 if ~file_test(conf.tabledir,/directory) then $
35   file_mkdir,conf.tabledir
36
37 ;create an array of damping values

```

```

38 n=100.
39 logdamp0=-4.
40 logdamp1=0.0
41 logdamp=findgen(n)*(logdamp1-logdamp0)/(n-1)+logdamp0
42 logdamp=reverse(logdamp)
43
44 ;load/build the W-matrix
45 code->load_matrix,conf.extbeam,odtmdir=conf.tabledir,$
46             LAMBO=conf.lamb0,LAMB1=conf.lamb1,DLAMB=conf.dlamb
47
48 ;variables for getting the optimal damping
49 outfile=strarr(n)
50 normr=dblarr(n)
51 normx=dblarr(n)
52 for i=0,n-1 do begin ;loop through damping values
53     ;temporary output file
54     logd=strcompress(string(logdamp[i],format='(F+8.5)'),/remove_all)
55     outfile[i]=dir+logd+'.fits'
56
57     ;extract this object
58     code->extract,outfile[i],conf.extbeam,/silent,/nouupdate,$
59             damping=10.^logdamp[i],odtmdir=conf.tabledir,$
60             LAMBO=conf.lamb0,LAMB1=conf.lamb1,DLAMB=conf.dlamb
61
62     ;extract the values
63     h=headfits(outfile[i],extension=0)
64     normr[i]=sxparr(h,'R1NORM')
65     normx[i]=sxparr(h,'XNORM')
66 endfor
67
68 ;recall... LINEAR is object-oriented and in IDL, one should
69 ;explicitly destroy objects (saves on memory leaks)
70 obj_destroy,code
71
72 ;make a diagnostic plot. the point of max curvature (the vertex
73 ;of the "L") is the Hansen (1992) optimal value
74 plot,normr,normx,$
75     /xlog,xtitle='!3!!!6Wf-G!3!!!6!U2!N', $
76     /ylog,ytitle='!3!!!6f!3!!!6!U2!N'
77 end

```

Listing 3 HUDF extraction script and damping optimization (optdamp.pro).

4. Common Failures

In principle, LINEAR defines the output spectral grid for each source individually, which has several advantages, such as extracting very faint sources in large wavelength bins to improve the signal-to-noise). On the other hand, one should be cautious in reducing the bandwidth to an extreme level. Since we have framed the spectral modeling as an inverse problem, the presence of noise leads to highly unstable solutions for samplings smaller than the native dispersion of the instrument. To illustrate this effect, we simulate a point-source

with an L5V spectrum⁶ that has F160W=10 AB mag. We assume a pair of 350 s exposures with the G141 grism with standard WFC3/IR dark, readnoise, and pixelation. We add zodiacal background estimated from the WFC3 exposure-time calculator and include Poisson noise to each exposure, then extract this pair of simulated images with two different bandwidths: 10 and 50 Å. In Figure 3, we show the two different extraction options, but additionally show the 10 Å extraction binned *post facto* to 50 Å. Since the native dispersion of the G141 grism is $\sim 45 \text{ \AA pix}^{-1}$, there is no unique information available when extracting at a sampling much narrower bandwidth. Extracting at these narrow bandwidths will over estimate the noise in the spectra, which is only exacerbated when signal-to-noise is decreased.

5. Summary

We have presented three basic scripts that highlight the usage of our grism package: **LINEAR**. We are preparing **LINEAR** for release on GitHub, however until then, we encourage interested users to email us for pre-release versions. We anticipate altering various components and/or adding additional methods to facilitate additional functionality (such as automation for the damping optimization).

6. Revisions

None.

References

- Beckwith *et al.* 2006, AJ, 132, 1729
- Bertin, E. & Arnouts, S. 1996, A&AS, 117, 393
- Bohlin, R. C., Deustua, S. E., & Pirzkal, N. 2015, WFC3-ISR, 10
- Brammer, G., Pirzkal, N., McCullough, P., & MacKenty, J. 2014, WFC3-ISR, 3
- Brammer, G., Ryan, R. E., & Pirzkal, N. 2015, WFC3-ISR, 17
- Cultrera, A. & Callegaro, L. 2016, arXiv: 1608.04571

⁶Obtained from the SpeX/Prism Library maintained by Adam Burgasser at <http://pono.ucsd.edu/adam/browndwarfs/spexprism/>.

Hansen, P. C., 1992, SIAMR, 34, 561
Markwardt, C. B. 2009, APSC, 411, 251
Landsman, W. B. 1993, ASPC, 52, 246
Pirzkal, N., *et al.* 2004, ApJS, 154, 501
Pirzkal, N. & Ryan, R. 2017, WFC3-ISR, 05
Ryan, R. E., Casertano, S., & Pirzkal, N. 2018, PASP, 130, 034501
Smith, J. D., *et al.* 2007, PASP, 119, 1133

Table 1: CONFIGURATION KEYWORDS FOR LINEAR

keyword	default	mandatory	description
Keywords for Input Images			
SEGMAP	seg.fits	✓	segmentation map to define extraction apertures
OBSFILE	obs.lst		file describing direct images
Controls for Running			
MODE	extract	✓	mode for running LINEAR
TABLEDIR	TABLEs/		directory to store ODTs and OMTs
REMAKE	0	✓	flag to remake the tables
ZIP	1		flag to gzip the tables
WCSTYPE	imglst		switch specifying where the WCS is specified
Variables for Simulation			
SEDLST	seeds.lst	✓	file that points to the model SEDs
WCSLST	wcs.lst		file containing WCS keywords
SIMBEAMS	A		flag to specify which beam(s) to simulate
Variables for Extraction			
IMGLST	fits.lst	✓	file to specify the grism images to extract
DAMP	0.0		unscaled damping parameter
OUTPUT	out.fits	✓	output fits table containing extracted spectra
LAMBO	7500	✓	starting wavelength for extraction (Å)
LAMB1	11000	✓	ending wavelength for extraction (Å)
DLAMB	25.0	✓	bin width of the sampling (Å)
MAGLIMIT	26		the faintest (aperture) magnitude to extract
MCMCUNC	0		do MCMC simulation for improved uncertainty estimates
EXTBEAM	A	✓	flag to specify which beam(s) to extract
MSKBEAM	none		flag to specify which beam(s) to mask
Calibration Files			
AXECONF		✓	aXe-based configuration file
FOVNAME		✓	fits file specifying properties of the detector
CHIPNAME	IR	✓	name of the chip to extract
CPU Settings			
POLLTIME	1.0		frequency to poll the multithreading utility
NCPU	0		maximum number of CPUs to use (≤ 0 is optimal)

Table 2: DEPENDENCIES FOR LINEAR

Name	Reference	Language	Description
<code>AstroLib</code>	Landsman (1993)	IDL	primary codes for <code>fits</code> -file handling, astrometry, and basic utilities
<code>PolyClip</code>	Smith et al. (2007)	IDL/C	basic polygon clipping routines
<code>mpfit</code>	Markwardt (2008)	IDL	Levenberg-Marquardt least-squares solver

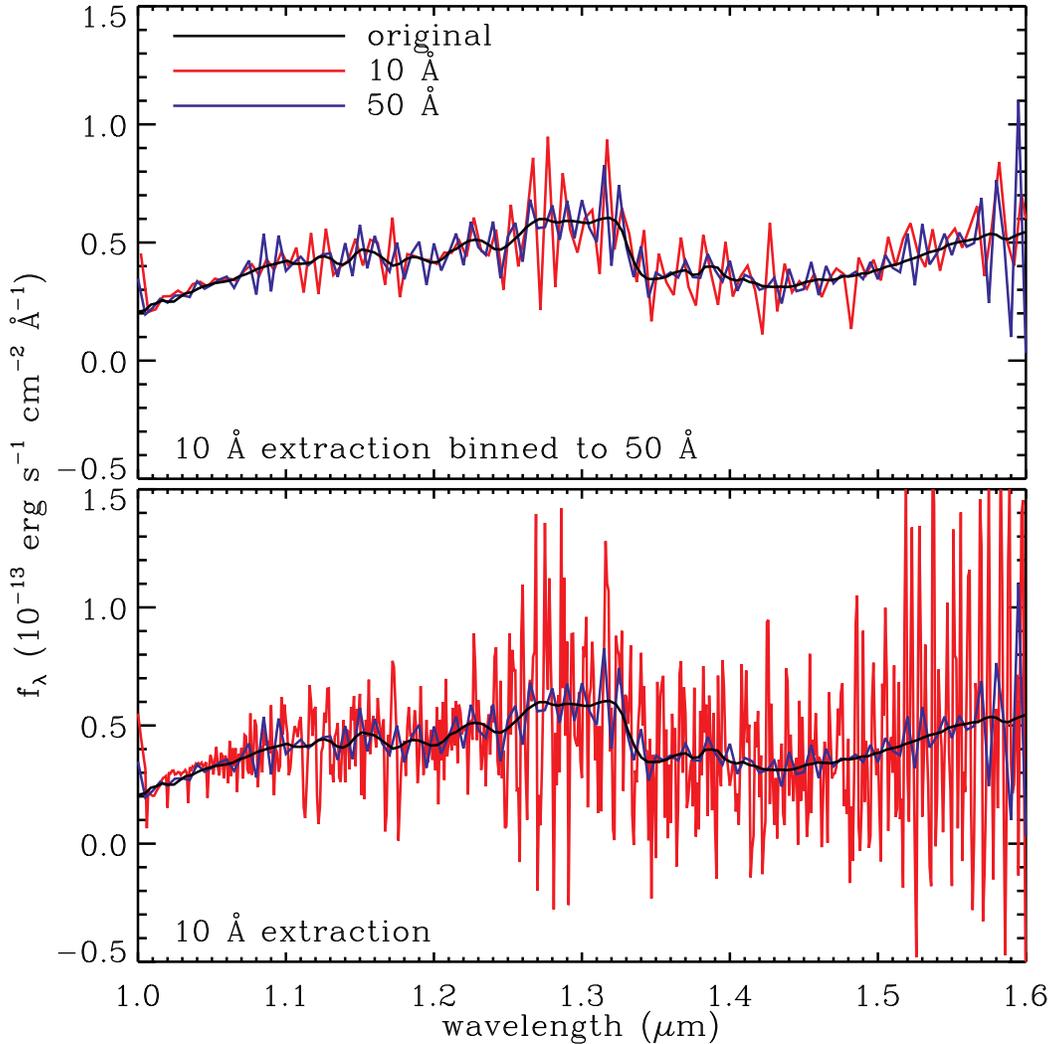


Fig. 3.— Extracted oversampling problems. In both panels, we show the original L5V spectrum (black line) used to simulate the point source. The blue line represents the the 50 Å extraction, and is the same in both panels. The red line for the bottom panel shows the 10 Å extraction, which exhibits much high-frequency noise. Some of this noise arises from fixing the signal-to-noise in the simulations, but reducing the bandwidth, which necessarily degrades the signal-to-noise per spectral element. However, when we bin this degraded signal to the same resolution as the blue line (50 Å), we still find an enhanced noise component (top panel). We believe this arises by trying to solve an inverse problem in the presence of noise, particularly when dramatically oversampling the wavelength. This effect is only amplified when the signal-to-noise is decreased. Therefore it is important to keep the extracted bandwidth from becoming much smaller than the native scale of the grism element.