Instrument Science Report STIS 2018-01

A Python Script for Aligning the STIS Echelle Blaze Function

Malinda Baer[1, 2], Charles R. Proffitt[2], & Sean A. Lockwood[2]

[1]*Ohio State University, Columbus, OH*
[2]*Space Telescope Science Institute, Baltimore, MD*

5 January 2018

ABSTRACT

Accurate flux calibration for the STIS echelle modes is heavily dependent on the proper alignment of the blaze function for each spectral order. However, due to changes in the instrument alignment over time and between exposures, the blaze function can shift in wavelength. This may result in flux calibration inconsistencies of up to 10%. We present the stisblazefix Python module as a tool for STIS users to correct their echelle spectra. The stisblazefix module assumes that the error in the blaze alignment is a linear function of spectral order, and finds the set of shifts that minimizes the flux inconsistencies in the overlap between spectral orders. We discuss the uses and limitations of this tool, and show that its use can provide significant improvements to the default pipeline flux calibration for many observations.

Table of Contents

Introduction

Changes in the alignment of the STIS echelle optics can cause the blaze function of the echelle orders to shift in wavelength (see Bowers & Lindler 2003). Failure to properly track and correct for these shifts can lead to flux inconsistencies of 10% or more in the overlapping regions of spectral orders, (e.g., Figure 1). The CALSTIS pipeline software includes provision to correct for such shifts by assuming that for a given CENWAVE the shift of the blaze function can be characterized as linear functions of the observation date and the wavelength offset measured from the WAVECAL lamp exposures.

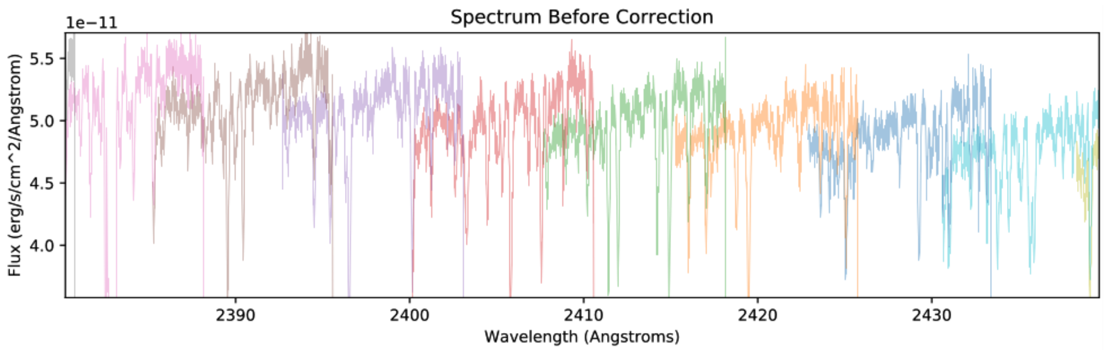

Figure 1: An example of the flux mismatch between echelle spectral orders that can occur if the blaze function is not properly aligned.

The derivation of such corrections for pre-SM4 STIS echelle data was discussed by Aloisi (2011) and Kim Quijano (2009). For any observation, the pipeline applies a predicted blaze function shift in pixels given by the equation:

```
BSHIFT = BSHIFT_VS_X × Δx + SHIFT_VS_Y × Δy + BSHIFT_VS_T × Δt + BSHIFT_OFFSET
```

where Δx, Δy, and Δt are given by

```
Δx = (REFWAVE – OBSW)/dispersion
Δy = OBSY – REFY
Δt = OBSDATE – REFMJD,
```

and the coefficients, `BSHIFT_VS_X`, `BSHIFT_VS_Y`, `BSHIFT_VS_T`, `BSHIFT_OFFSET`, `REFWAVE`, `REFY`, `and` `REFMJD` are obtained from the Photometric Conversion Table (PHT) reference file. The quantities OBSW, OBSY, and OBSDATE, are the calibrated wavelength of pixel 512 in the reference order, the average Y offset on the detector of pixel 512 in all orders, and the Modified Julian Date respectively of the observation being calibrated. Aloisi (2011) derived a set of coefficients for the PHT file that allowed the blaze function shift to be calculated as a function of Δx, Δy, and Δt for STIS echelle data taken prior to the failure of STIS in 2004.

After STIS was repaired in 2009 during *HST* Servicing Mission 4 (SM4), new baseline sensitivity curves were derived for all STIS echelle modes by Bostroem et al. 2012. At that time, the values of the blaze shift coefficients `BSHIFT_VS_X` and `BSHIFT_VS_Y` for each order were set to their pre-failure values; however, as there was yet no baseline to evaluate the post-SM4 time dependence of the blaze shift, `BSHIFT_VS_T` and `BSHIFT_OFFSET` were set to zero in the PHT reference file. This has subsequently resulted in an increasing misalignment of the blaze function over time.

Proper calibration of the time dependence of the blaze shifts for post-SM4 STIS echelle observations is still in progress, and to date updated coefficients that include the time dependent part of the blaze function shift have been delivered to the pipeline only for the E140H modes, (Monroe 2017). For the E230M, E230H, and E140M settings, the time dependent change of the blaze shift is not yet included in the calibration of post SM4 data, although delivery of the new coefficients is expected in the near future.

While delivery of these post-SM4 time dependent blaze shift coefficients will considerably ameliorate the blaze function misalignment and result in significant improvements to the flux calibration, they will not completely eliminate the problem for all STIS echelle observations. Limited calibration data for standard stars can result in inconsistent results for different settings, and changes in trends may take a number of years to measure and incorporate fully into the calibration reference files. Furthermore, it also appears that over short timescales there are modest alignment changes that are not well tracked by the pipeline algorithm. There are a number of pre-SM4 data sets where these effects cause residual flux inconsistencies of up to 5% despite the use of the full set of the Aloisi et al. (2011) coefficients, and similar problems will likely persist for some post-SM4 data as well.

Because of these residual calibration uncertainties, it is very useful to have a tool that can empirically find the shifts that best align the blaze functions for any given observation. This ISR presents the *stisblazefix* Python module, a tool for correcting individual STIS echelle spectra. This tool allows a significant improvement of the flux calibration for those echelle modes which either have not yet had an updated blaze function correction implemented into the pipeline, or for which the actual blaze shift deviates significantly from the calibrated trends. The purpose of this ISR is to explain how and when to use the *stisblazefix* tool to correct STIS echelle spectra.

Approach

The CALSTIS pipeline calculates the extracted fluxes by dividing the observed net count rate at each pixel by a sensitivity function. The baseline sensitivity function for each echelle order is separately tabulated in the PHOTAB reference file. It is first aligned in wavelength using the zero-points measured from the wavelength calibration exposure and the offset tabulated in the APDESTAB reference file for the selected aperture. The sensitivity curve is then shifted in wavelength based on the predicted offset of the blaze function as described in Aloisi (2011) and Kim Quijano (2009). The final sensitivity curve is also modified by the tabulated aperture throughput curve, (APERTAB reference file), and the encircled energy curve (PCTAB reference file), both of which are functions of wavelength and aperture, as well as by the time dependent sensitivity (TDSTAB) correction.

The purpose of the *stisblazefix* package is to empirically improve the alignment of the blaze function for cases where the CALSTIS prediction leaves significant errors in the flux calibration. This is done by finding the set of shifts for the sensitivity curves of individual orders that make the calibrated flux in the wavelength overlap between adjacent spectral orders most consistent. While similar concepts for correcting the blaze alignment have been suggested previously and utilized by individual authors, (e.g., Bowers & Lindler 2003; Ayres 2015), our goal is to provide a general tool that can be easily applied to any STIS echelle data set and allow the user to quickly evaluate the effect of the correction.

To do this a quantitative metric for the inconsistency needs to be defined. For each wavelength range where two echelle orders i and i+1 overlap, we defined the "Flux overlap residual" as $R_i = 1 - \sum F_i \,/\, \sum F_{i+1}$, where F is the calibrated flux and the sums are taken over the same set of wavelengths on each of the two overlapping orders. We also calculate the formal error in this flux overlap residual by propagating the flux errors produced by the pipeline extraction.

To ensure that the sums are taken at the same wavelengths, for each overlapping wavelength region, the flux and error vectors are interpolated from the higher numbered order onto the wavelength grid of the lower numbered order, while the data quality flags from the higher order are combined with those of the pixel at the nearest wavelength in the lower order. While this interpolation introduces some smoothing, it allows the sums to be taken at exactly the same wavelength points in each overlapping region. Any pixels flagged as "bad" in the combined data quality vector are excluded from the sums. For this purpose, bad pixels are defined as those with any bit values of 4 (detector problem), 8 (data masked), 32 (large blemish), 512 (calibration defect), or 2048 (bad background). The first and last five pixels at the edge of each order are also excluded from the residual sum to guard against any edge effects that are not properly flagged in the data quality vector.

Rather than attempting to recreate the detailed CALSTIS procedure for defining the sensitivity curve of each order, we instead recovered the final applied sensitivity curve from the tabulated "net" and "flux" vectors in the extracted spectra in the x1d files produced by CALSTIS. The calibrated flux vector was produced by dividing the net flux vector by the sensitivity curve; therefore, the original sensitivity curve is simply equal to the net flux divided by the calibrated flux. Locations where the tabulated flux is zero and the sensitivity cannot be directly recovered are filled in by interpolation.

An important assumption in both the CALSTIS code and our tool is that the shift of the blaze function can by approximated by shifting the entire sensitivity curve for that order. In principle, it would be better to separate out the echelle blaze function itself from other contributions to the sensitivity curve that depend primarily on the wavelength or the location on the detector, but this would require significant reanalysis of the STIS echelle flux calibration. We instead assumed that other contributions to the shape of the sensitivity curve vary only slowly over a single order.

Once the pipeline applied sensitivity curve has been recovered, it is straightforward to shift the sensitivity curve for each spectral order by any specified number of pixels and recalculate the new flux values, errors, and flux overlap residuals for that set of shifts. Our goal is to find the set of blaze function shifts that minimize the error weighted flux overlap residuals.

Since we do not make any a priori assumption about the flux distribution of the source, we can only estimate the shifts for the n spectral orders in the extracted spectrum by using the consistency of the flux residuals in at most $n-1$ overlapping wavelength regions. To do this we assumed that the needed shifts for any given observation vary linearly as a function of spectral order so that there are only two free parameters, a, the offset in pixels of the lowest order and, b, the change in offset between orders. The minimization then is done by the LMFIT module (see https://lmfit.github.io/lmfit-py/), an open source non-linear least squares minimizer using the Levenberg-Marquardt method. Given a starting guess to the blaze function shifts, *stisblazefix* iterates until it finds the linear relation for the blaze shift as a function of spectral order that minimizes the set of weighted flux overlap residuals. An example of the resulting corrections is illustrated in Figure 2.

For some modes there are a few orders for which the behavior of the flux residuals as a function of blaze function offset deviates significantly from the trends found for most of the other orders. This is most common near the extremes of the E230M and E230H gratings. To avoid having deviant points bias the overall solution, overlap regions for which the residuals differ significantly, (more than 3.5 σ for any order or more than 2 σ for the first 5 and last 5 overlap regions), from the residual trend measured for the majority of orders are assigned an artificially large error value. That is, we make it easier to reject edge orders that show inconsistent overlap ratios.

Any adjacent orders for which there are no valid points in the overlap region are assigned a flux residual value of zero, but with the error again set to a large value so they have insignificant weight in the final fit. However, the fitted blaze shift solution is applied to all orders, even where the lack of an overlap means that some orders did not contribute to the fit. This most commonly occurs with the E140M, as there is no overlap in wavelength for several orders at the long wavelength end.

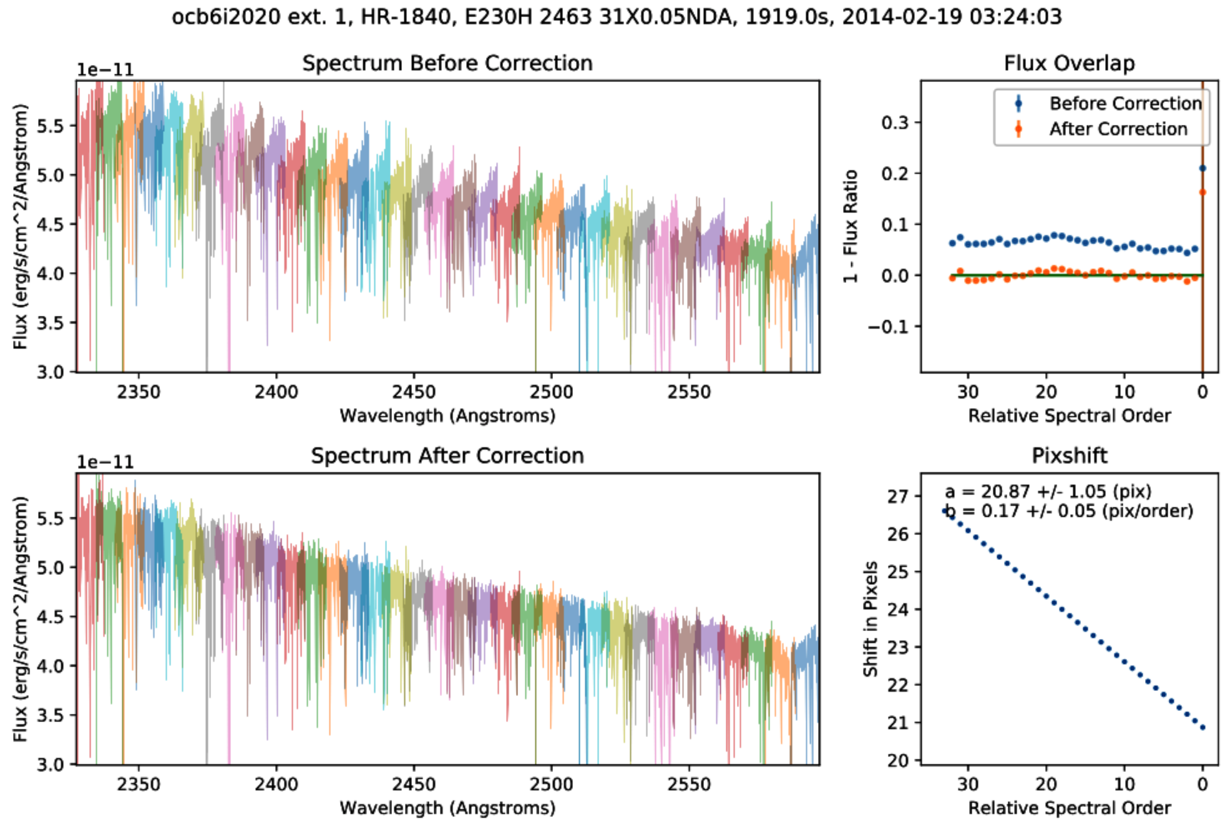

Figure 2: An example of diagnostic plot produced by the fluxfix function of the stisblazefix module. In the before and after flux plots on the left, alternating colors are used to show the flux for each spectral order, with darker regions showing the overlap between orders. The plot on the upper right shows the residuals before and after the correction is applied, while the plot in the lower right shows the shift applied to each order, along with the values for the offset and slope of the fit. In the plots on the right, we label the orders relative to the lowest echelle order included. In this case, the order assigned the index zero actually corresponds to the 298th order of the E230H echelle. Remember that higher order numbers correspond to shorter wavelengths. The adopted solution is very effective at removing the sawtooth pattern seen in the original calibration, however the flux overlap between the longest two wavelength spectral orders deviates significantly from the trend seen for the others. The observation name, extension (subexposure) number, target name, grating, CENWAVE, aperture, exposure time, and the date and time of the observation are listed at the top of the figure.

Installation

Documentation and installation instructions for the *stisblazefix* package can be found at https://stisblazefix.readthedocs.io/. These installation instructions are intended for those making use of the Conda utility which is provided by Continuum Analytics (see https://www.anaconda.com) or the miniconda utility which is provided by Anaconda Inc., (see https://conda.io/miniconda.html); however, assuming the required supporting packages are installed, the *stisblazefix* Python module should work with other installation methods. Instructions for manual installation are also given on the documentation page.

The *stisblazefix* module was primarily developed using Python 3.6.1, but was also successfully tested using Python 2.7.13. Numpy version 1.13 or later is required; issues were encountered when using Numpy 1.12.

The current implementation also uses the lmfit package, (https://lmfit.github.io/lmfit-py/), which is not available as part of either the default Conda channel or the AstroConda channel, but which is available on the conda-forge channel.

Users should first activate the Conda environment in which they wish to install this package. Remember that it is usually not advisable to install packages into the root conda environment. Users should then do

```
conda install -c conda-forge lmfit
conda install -c sean-lockwood stisblazefix
```

If the Conda environment being used already includes one of the AstroConda channel's recommended software stacks (see http://astroconda.readthedocs.io), other needed modules should already be available. If the user wishes to install *stisblazefix* without also installing the full AstroConda software stack in the same environment, it should be possible as long as compatible versions of Astropy and Matplotlib are available and installed.

It is anticipated that a future version of the *stisblazefix* module will eventually be added to the default AstroConda distribution, and this report will be updated at that time.

Usage

The *stisblazefix* correction is applied to calibrated STIS x1d FITS files using the *fluxfix* function in the *stisblazefix* module. This is called in a Python interpreter or from a Python script in the following manner:

```
import stisblazefix as sbf
filelist = ['firstfile_x1d.fits', 'secondfile_x1d.fits']
sbf.fluxfix(filelist, 'plots.pdf')
```

The *fluxfix* function takes two required arguments, with additional keyword arguments. The first argument is a list of the names of the FITS files to be corrected, and the second is the name of the PDF file where *fluxfix* saves the diagnostic plots. A few optional keyword arguments are described below.

Once the best shift values are determined, *fluxfix* generates new FITS files with corrected flux and error vectors. The corrected FITS files substitute "`x1f`" in the file name in place of the original "`x1d`". The routine also produces diagnostic plots comparing the spectra before and after correction, the flux residuals before and after correction, and the final shift applied in pixels as a function of spectral order, (see Figure 2 for an example). For input x1d files that contain multiple extensions with separate exposures, the *fluxfix* task finds separate solutions for each exposure, produces separate plots for each, and appropriately updates each extension of the "`x1f`" output file.

The keyword argument `guess`, allows the user to supply an alternate pair of starting values for the iteration to determine the offset and the slope of the blaze shift. Setting the keyword `iterate=False`, skips the minimization procedure, and forces the pair of values specified in `guess` to be used instead. Finally, setting the keyword `nxplot` to a value greater than 1 inserts additional full pages before and after flux comparison plots into the pdf file, each showing about `1/nxplot` of the full range with some modest overlap between figures. This may be useful for noisy spectra or observations of sharp-lined stars where it is difficult to judge from the single summary plot the effect that the correction has had on a particular piece of the spectrum.

While it is anticipated that most users will consider the corrected FITS files and the diagnostic plots as the primary output of this routine, the *fluxfix* function also returns a list of Python dictionaries containing the following quantities for each exposure processed.

```
{'pixshift': pixshift,   # array of pixel shift values applied to each order
 'acof': acof,           # shift of sensitivity curve of lowest order
 'bcof': bcof,           # change in offset per order
 'acoferr': acoferr,     # formal error in the value for acof
 'bcoferr': bcoferr,     # formal error in the value for bcof
 'oldresids': oldresids, # vector of residual differences in original data
 'oldresiderr': oldresiderr, # error vector for oldresids
 'newresids': newresids, # vector of residual differences in corrected data
 'newresiderr': newresiderr,  # error vector for newresids
 'filename': filename, #input file name
  'extno': i}   # extension number of input file for this exposure
```

Applications and Limitations

The *stisblazefix* module is most immediately useful for echelle data taken using modes that still lack an up-to-date blaze function correction in the standard pipeline. However, even for modes that already include the time dependence of the blaze correction, it will be useful in cases where extremely precise alignment is required or for individual observations that deviate significantly from the established trends. Running the tool on some recently calibrated post-SM4 E140H data reveals that there are some examples where the tool can make a noticeable improvement (Figure 3), even after the application of the updated Monroe et al. (2017) pipeline calibration that includes the time dependence of the blaze shift for the E140H settings. In this case the plot of the flux overlap residuals shows ~5 to 10% discrepancies that can be removed by tool. However, this improvement in the spectrum can be difficult to visualize at the scale of the default before and after plots presented in Figure 3. In such cases it can be useful to use the *fluxfix* module’s `nxplot` option to add additional plots showing smaller wavelength intervals (Figure 4).

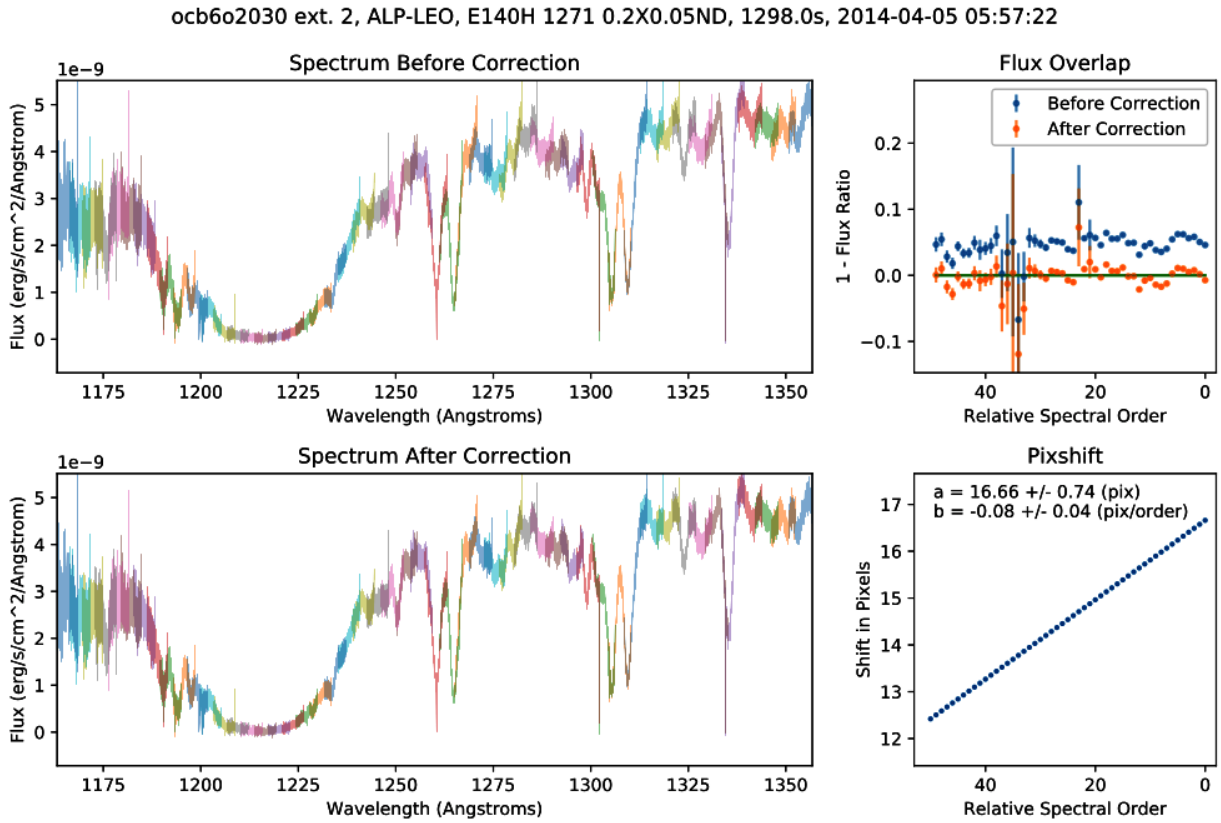

Figure 3: Although this E140H observation was recalibrated with the recently delivered reference file that included the post-SM4 time-dependence of the blaze shift, use of the stisblazefix tool can still produce a modest improvement (see also Figure 4) in the relative flux calibration between orders.

The assumption that the blaze shift can be approximated by a shift of the entire sensitivity curve for each spectral order begins to break down in some cases. This is especially noticeable for the shortest and longest wavelength orders in some E230M observations, where the sensitivity curve in the corners of the NUV MAMA detector appears to be affected by some vignetting that does not shift along with the blaze function. In this case these discrepant orders are rejected as outliers and assigned large error bars for fitting. While the final solution found noticeably improves the consistency for most orders, it actually degrades it for these edge orders (e.g., Figure 5).

In some cases, the applied shift appears to overcorrect the very shortest wavelengths in a spectral order, leading to a spike in flux on the very edge of the order. Similarly, for many E140M observations, the shifts that best align the overlap regions between the spectral orders leave a noticeable shape to the flux in the longest wavelength orders (Figure **6**). Both of these effects could be explained if either the shape of the blaze function has changed since the original calibration, or if the blaze shifted significantly in wavelength during the course of the observation itself.

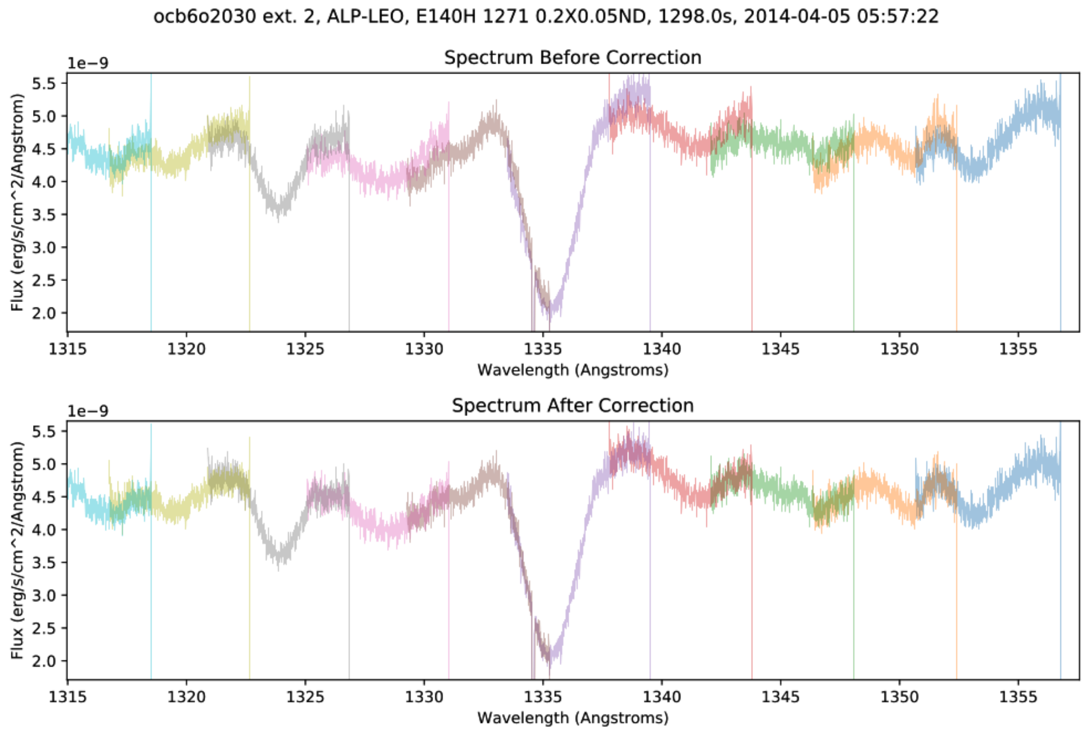

Figure 4: To produce this figure, `nxplot` was set equal to 5 to produce additional plots, each comparing about 1/5 of the overall spectral range that was shown in Figure 3. Only one of these ranges is shown here. The expanded scale makes it easier to see the improvements in the order-to-order flux consistency produced by the tool. Note that these detailed plots include all flux values in the x1d files, including those with dq flags that mark the individual points as questionable.

As long as there is significant continuum flux covering a number of the overlap regions, *stisblazefix* should work well even on relatively low S/N data. Tests on time-tag data sliced into successively smaller exposures showed good consistency between results for the full exposure and individual shorter and lower S/N time intervals. However, for observations where scattered light or detector background dominates the continuum flux in the overlap regions, it may not be possible to obtain reliable results. Attempts to run the algorithm on blank echelle exposures where the shutter had failed to open resulted in very large spurious shifts. Sources dominated by a few strong emission lines may be overly sensitive to the details of how the emission lines sample the overlap regions.

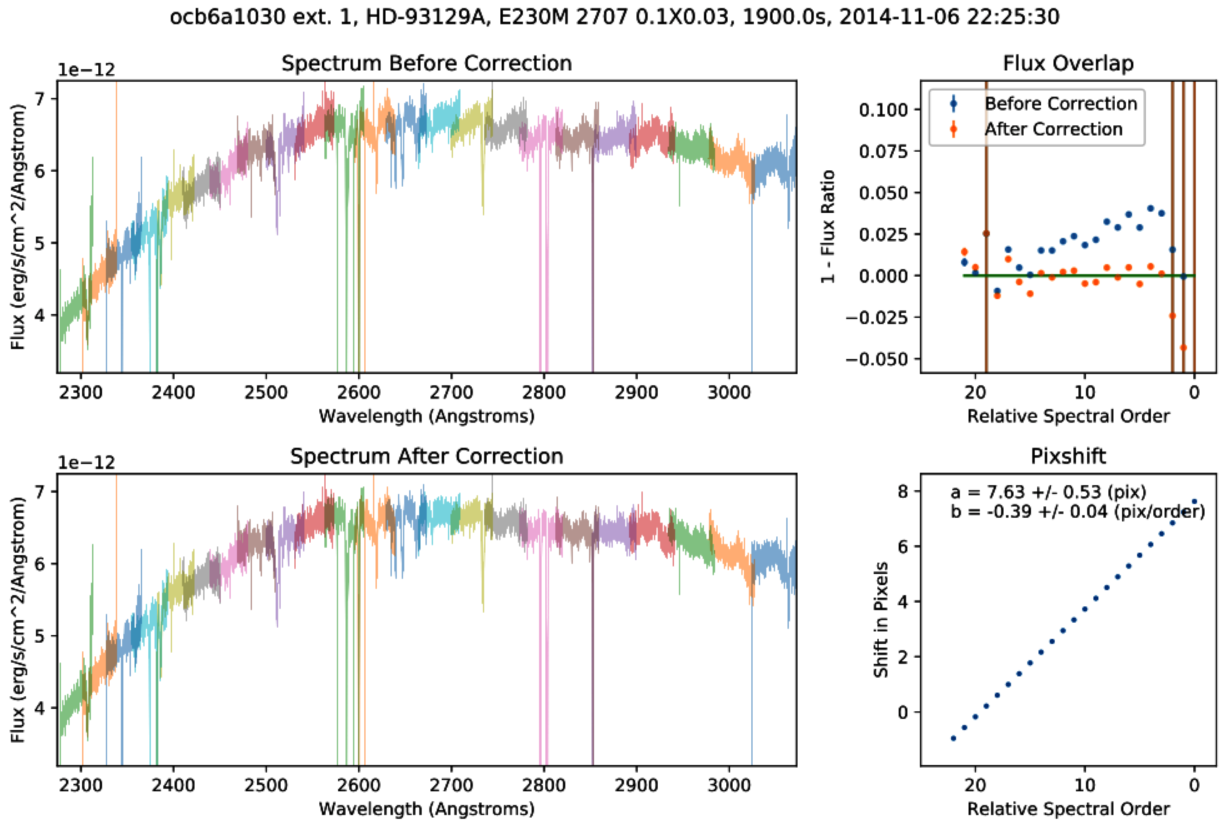

Figure 5: In this example, the correction results in significant improvement in the middle of the spectrum, where the inconsistencies between the spectral orders are mostly eliminated, but for the three longest wavelength overlapping regions, the order mismatch increases.

For these reasons, when *stisblazefix* is used to correct a spectrum, it is important for the users to review the diagnostic plots and make sure the program is doing a good job of correcting the data. While it is rare that *stisblazefix* will make results worse overall, there may be certain wavelength ranges where the correction introduces localized artifacts.

While the *stisblazefix* module is a useful tool for correcting a wide variety of echelle data, it is still subject to some limitations. Users should decide whether it is necessary for their data, and examine the results to ascertain whether it offers a useful improvement.

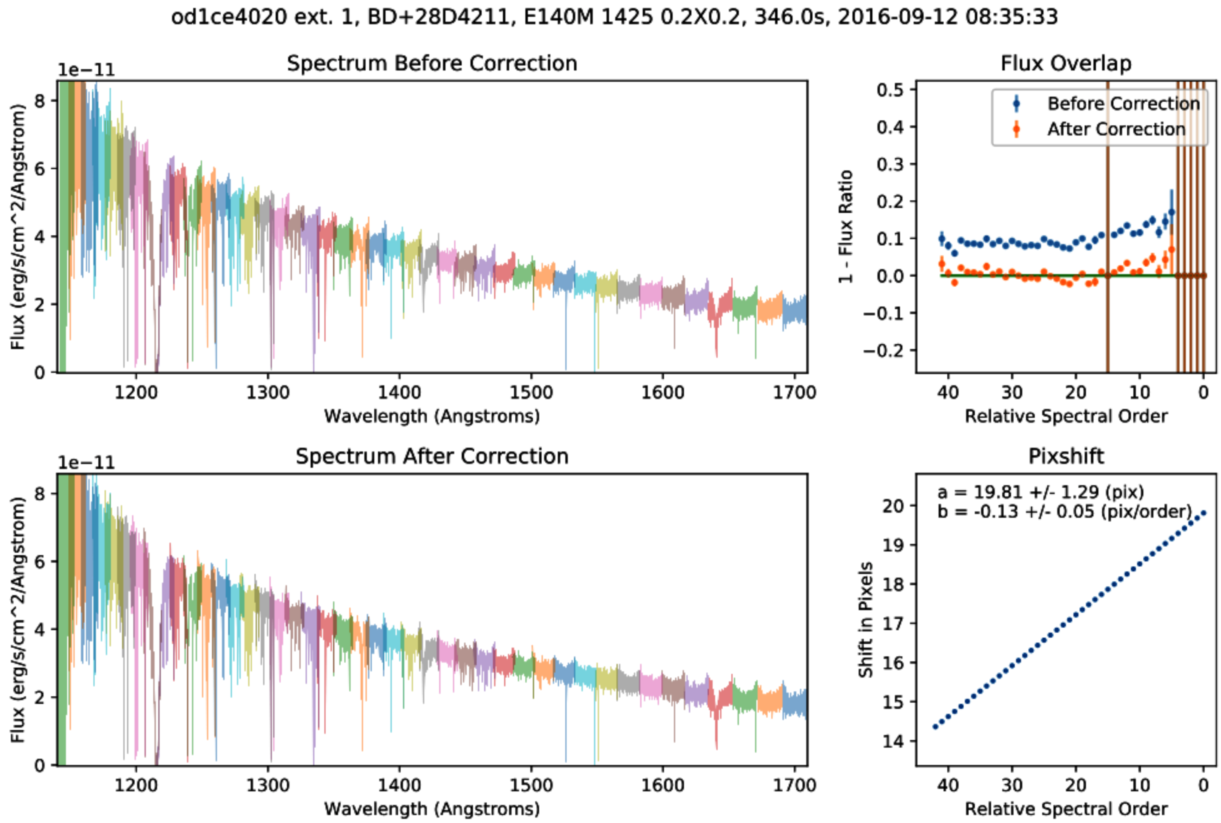

Figure 6: In this E140M example, application of the correction significantly improves the flux consistency at most wavelengths, but flux spikes appear at the short wavelength end of some orders, and for some of the longest wavelength orders, there is noticeable curvature in the calibrated flux that appears to correlate with the blaze function. In this case, the shapes of some of the blaze functions appear to have become broader than when the initial calibration was done. Since there is little or no wavelength overlap of the longest wavelength orders, the residual values for these orders are set to zero with large errors assigned so they do not constrain the fit.

Acknowledgements

The authors would like to thank Tala Monroe for useful discussions on the STIS echelle blaze function alignment. We would especially like to thank the Space Telescope Science Institute Space Astronomy Summer Program (SASP) for its support of MB during the summer of 2017, which made this project possible. Development of an earlier prototype of this algorithm, written by CP in IDL rather than Python, had been supported by GO grant HST-GO-14194.002-A.

Support for Program number 14194 was provided by NASA through a grant from the Space Telescope Science Institute, which is operated by the Association of Universities for Research in Astronomy, Incorporated, under NASA contract NAS5-26555.

Change History for STIS ISR 2018-01

Version 1: 22 September 2017 – Original Document

References

Aloisi, A. 2011, "Time Dependence of the Echelle Modes Sensitivity before the STIS Repair", Instrument Science Report STIS 2011-04, Baltimore, MD: STScI

Ayres, T. R. 2015, "HST STIS Advanced Spectral Library Project"

Boestrom, K. A., Aloisi, A., Bohlin, R., Hodge, P., & Proffitt, C. 2012, "Post-SM4 Sensitivity Calibration of the STIS Echelle Modes", Instrument Science Report STIS 2012-01, Baltimore, MD: STScI

Bowers, C. W., & Lindler, D. 2003, "STIS Echelle Blaze Shift Correction" in *The 2002 HST Calibration Workshop: Hubble after the Installation of the ACS and the NICMOS Cooling System*, Proceedings of a Workshop held at the Space Telescope Science Institute, Baltimore, Maryland, October 17 and 18, 2002. Edited by Santiago Arribas, Anton Koekemoer, and Brad Whitmore. Baltimore, MD: Space Telescope Science Institute, 2003, p.127

Kim Quijano, J. 2009, "The making of the STIS photometric throughput tables", Technical Instrument Report STIS 2009-03, Baltimore, MD: STScI

Monroe, T. R. 2017, "Echelle Blaze Function/FUV PHOTTAB Update", in STScI Analysis Newsletter (STIS), August 2017