

# CALSTIS

## REFERENCE GUIDE



**Don Lindler**  
**August, 2003**

# CALSTIS Reference Guide

## CONTENTS

1.0 Introduction	1
2.0 Raw STIS Data Files	1
3.0 Using CALSTIS	3
4.0 CALSTIS Reduction Parameters	5
5.0 CALSTIS Reference Files	9
6.0 CALSTIS Processing Steps	10
7.0 Image Processing Steps	10
7.1 Conversion of MAMA Data to Lo-Res	11
7.2 Overscan Subtraction	11
7.3 Statistical Error Initialization	12
7.4 Cosmic Ray Removal	12
7.5 Doppler Smoothing Function	13
7.6 Bias Image Subtraction	13
7.7 Conversion to Count Rates	13
7.8 Dark Rate Image Subtraction	14
7.9 Flat Field Processing	14
7.10 Bad/Hot Pixel Processing	14
7.11 Image Processing Outputs	15
7.12 Camera Mode Geometric Distortion	16
8.0 Point Source Spectral Extraction	18
8.1 Point Source Spectral Location	18
8.2 Gross and Background Spectral Extraction	20
8.3 Background Processing and Subtraction	21
8.4 Wavelength Assignment	22
8.5 Conversion to Absolute Flux Units	23
8.6 Heliocentric Wavelength Correction	24
8.7 Point Source Spectral Extraction Outputs	25
8.8 Echelle Scattered Light Correction	26
9.0 Extended Source Spectral Extraction	28
9.1 Extended Source Image Rectification	28
9.2 Extended Source Flux Calibration	30
9.3 Non-linear Interpolation in Extended Source Extractions	32
9.4 Extended Source Extraction Outputs	33
10.0 CALSTIS Output Data Files	33
10.1 Output Flat Fielded Count Rate Image File.	34
10.2 Output Extended Source Extraction File	34
10.3 Extracted Point Source Spectral File	34
11.0 Supplying Input Data to CALSTIS	36
12.0 STIS Data Catalog, STISLOG	37

13.0 Wavelength Zero Point Offset Calibration	38
14.0 Introduction to Time-Tag Data Processing	40
14.1 Working with Very Large Time-Tag Data Sets	44
Appendix A Data Quality Flag Values	47
Appendix B Sample Sessions	48
B.1 Finding and Reading STIS Data	48
B.2 First Order Grating Point Source Spectral Extraction	50
B.3 G750M Extended Source Extraction of a Long Slit Observation	51
B.4 Long Slit Extended Source Extraction (M84) with a Constant Delta Radial Velocity	52
B.5 Calibrate an Echelle Spectrum Including a Scattered Light Correction	54
B.6 Calibrate and Merge a Two Readout Echelle Observation	55
Appendix C Recent changes to CALSTIS	56

## 1.0 Introduction

CALSTIS is an IDL procedure that performs the routine reduction of Space Telescope Imaging Spectrograph data. This manual describes how to

- ❖ search for data in the data catalog (Section 12)
- ❖ read the raw STIS data (Section 2)
- ❖ calibrate the data using CALSTIS (Sections 3 through 9)
- ❖ read the output data files produced by CALSTIS (Section 10)
- ❖ use the wavecalcs taken with the science data to correct for thermal motion and mode select mechanism non-repeatability (Section 13)

The current version of CALSTIS performs the basic image processing steps (Section 7), and can either perform a point source echelle or first order grating extraction (Section 8) or a first order grating extended source image rectification (Section 9). The current version of CALSTIS does not support echelle extended source extractions or prism spectral extractions.

## 2.0 Raw STIS Data Files

Raw STIS data files are stored in FITS files with names:

*rootname\_raw.fits* - uncalibrated images  
*rootname\_spt.fits* - engineering data for the raw images  
*rootname\_wav.fits* - uncalibrated wavelength calibration lamp images.  
*rootname\_wsp.fits* - engineering data for the wavelength calibration images.  
*rootname\_tag.fits* - MAMA time-tag data file  
*rootname\_trl.fits* - ASCII Table Containing ST Sci Processing History  
*rootname\_lrc.fits* - local rate check image  
*rootname\_lsp.fits* - engineering data for the local rate check image

Further information on the format of these files can be found in the STScI STIS Instrument Handbook and the in ST Sci Data Handbook.

The raw STIS data files are catalogued in the IDL data base STISLOG (Section 12). As received, each observation is assigned an ID number (the entry number in the STISLOG catalog). Data can be referenced when using the STIS IDL software either by the filename or by the ID number.

The routine STIS\_READ is used to read raw STIS data files. Its calling sequence is:

***STIS\_READ, file, header, data, udl, heng***

where

***file*** is the ID number or the raw image filename.

***header*** is the output FITS header.

*data* is the output image array.

*udl* is binary engineering data and can be typically ignored by the average user.

*heng* is the FITS header of the engineering (*\_spt*) file.

For example, to read observation 1000 use:

```
stis_read,1000, h, data
```

or equivalently,

```
stis_read,'o3xv03w3m_raw.fits', h, data
```

You can then proceed to examine the header and data. For example,

```
hprint, h                ;list header contents  
tvsc, data              ;display data
```

STIS\_READ has the following optional keyword inputs:

*/HIRES* to leave MAMA data in the Hi-Res mode. The default is to convert the data to Lo-Res mode.

*/NO\_OVERSCAN* to disable the CCD overscan bias removal.

*READOUT* = number of the readout to read. The default is to read the first readout (*readout=1*)

Multiple readout input data files can be read with STIS\_READ using the keyword parameter *READOUT*. This method of reading the data is fine when only one of the readouts is to be read. If however, you are going to process all the readouts, this method can be very inefficient, particularly if the input raw data is stored in compressed format. When you read each readout, the data is uncompressed into a temporary file, the readout read, the file closed, and the temporary file deleted. A better approach is to uncompress, open the file once, and then read all readouts before closing and deleting the temporary uncompressed file. This can be done using the routines STIS\_OPEN and STIS\_CLOSE. To open the file use:

```
STIS_OPEN, '<filename>', fcb, nreads=nreads
```

where *fcb* (file control block) is an output data structure which will be passed to the other routines. The optional keyword output, *nreads*, gives the number of readouts in the file. Now you can start reading the data:

```
for i=1,nreads do begin  
    STIS_READ, fcb, h, d, readout=i  
    ...  
end
```

when done with the file, close it with:

*STIS\_CLOSE, fcb*

### 3.0 Using CALSTIS

CALSTIS can be called from the IDL command line or from another IDL procedure with the following calling sequences:

1) Point Source Spectral Extraction

*CALSTIS, id, params, h, m, wave, flux, epsf, errf, gross, back, net, pos, image, eps, err*

2) Extended Source Spectral Extraction

*CALSTIS, id, params, h, m, wave, flux, epsf, errf, sens, x, y, unused, image, eps, err*

3) Imaging Mode

*CALSTIS, id, params, h, unused, unused, image, eps, err*

where:

*id* is the input filename or the ID number in the STISLOG data base (See section 2.0).

*params* is a user specified parameter string used to control the reduction steps (Section 4.0).

*h* is the output FITS header string array.

*m* is a vector of spectral order numbers for the echelle modes. For the first order grating modes, *m* has the value of 1.

*wave* is a vector of wavelengths for first order grating modes. For echelle modes, it contains a two dimensional array giving the wavelengths for each spectral order.

*flux* is an array of extracted flux values for point source spectra, a rectified image for extended source extractions, or a flat fielded image for imaging mode observations.

*epsf* is a data quality array for flux. It is the same size as *flux* and has values as defined in Appendix A.

*errf* is the propagated statistical error array for *flux*.

*gross* is the extracted (point source) gross spectrum in count rates.

*back* is the extracted (point source) background spectrum in count rates.

*net* is the net spectrum (*gross* minus the smoothed background spectrum).

*pos* is an array of spectral extraction positions for point source extractions. The center of the extraction slit in raw data image coordinates for flux(i,j) is at position x=i, y=pos(i,j).

*image* is a flat fielded count rate image.

*eps* is a data quality array for *image*.

*err* is the propagated statistical error array for *image*.

*sens* is the sensitivity vector for an extended source extraction.

*x* is the raw image x-positions for the center of each pixel in an extended source rectified image

*y* is the raw image *y*-positions for the center of each pixel in an extended source rectified image  
*unused* indicates that the output parameter position is not used for this type of extraction.

The only required input/output parameter that must be supplied to CALSTIS is *id*. If CALSTIS is called by:

*CALSTIS, id*

all **default processing** will be performed and the results will be placed in output FITS files.

CALSTIS also has the following optional IDL keyword input parameters:

*READOUT* specifies which readout is to be processed for files containing multiple readouts. Unless specified, CALSTIS will processing the first readout (READOUT=1). To process the second readout, for example, use:

*CALSTIS, id, ..., readout=2*

*QUICK* can be used to get a quick spectral extraction. Specifying QUICK causes the bias image removal, dark rate image subtraction, and flat fielding to be turned off. Its usage is:

*CALSTIS, id, ..., /quick*

*CR\_REJECT* is used to specify that multiple readouts in a CCD data file are to be combined with cosmic ray removal by the routine STIS\_CR. Use:

*CALSTIS, id, ..., /cr\_reject*

*AUTOWAVE* is used to control the zero point shift in the wavelengths using an associated wavecal observation. See section 13.1 for its usage.

*NO\_PLOT* can be used to turn off the plotting during auto-wavecal processing.

*CALSTIS, id, ..., /no\_plot*

Additional keyword inputs, *HEADER*, *DATA*, *EPSIN*, and *ERRIN*, allow you to explicitly supply the data to be processed instead of having CALSTIS read it from an input file. Their usage is described in section 11.

As CALSTIS executes, it will print a processing history to the screen. This same history is placed into the output FITS header. To turn off the screen printing, set !dump to 0 before executing CALSTIS:

*!dump = 0*

*calstis,id,....*

To turn the screen printing back on, set !dump to 1.

## 4.0 CALSTIS Reduction Parameters

The second parameter in the calling sequence of CALSTIS, PARAMS, is the user specified parameters to CALSTIS. These parameters control how CALSTIS reduces the data, including which reduction steps are performed and which calibration reference files are used. PARAMS can have any of the following forms:

VALUE

- 0 - use all default parameters
- 1 - interactively select the parameter values
- 'string' - parameter definition of the form:  
          'pname=value,pname=value,....'
- '<filename>' - name of a text file with the contents:  
          pname=value  
          pname=value  
          ....

Table 4.1 gives a listing of the parameter names and the default values. Further information on them can be found in the individual reduction sections which follow.

**TABLE 4.1**  
**CALSTIS Parameter Selection**

Parameter Name	Default Value	Description	Sections
outspec	DEF	Spectral file name (DEF, none, or a filename)	10
outimage	none	Output flat fielded file name (DEF, none, or a filename)	10
outgeom	none	Output geometrically corrected camera mode image file name (DEF, none, or a filename)	7.12
extended	0	Extended Source Extraction (0 - no, 1 - yes)	9
hires	0	MAMA Hires processing (0 - no, 1 - yes)	
minerrv	0	Minimum MAMA raw data variance	7.3
nospec	0	Skip Spectral Extraction or extended source image rectification (0 - no, 1 - yes)	
errors	1	Propagate the statistical errors (0-no, 1=yes)	
tau	0	MAMA global non-linearity time constant	7.7

Parameter Name	Default Value	Description	Sections
soffset	0	Spatial offset (pixels) from nominal spectral position to center of the search area used to locate a point source spectrum.	8.1
dist1	200 (G140L) 100 (all other modes)	Maximum distance from center of search area (in the spatial direction) to look for a point source spectrum.	8.1
gwidth	11.0 (MAMA) 7.0 (CCD)	Width of the region defining the point source gross spectrum (i.e. height of the extraction slit)	8.2
bwidth	11.0 (first order) 3.0 (echelle)	Width of the regions defining the point source background spectra (i.e. height of the upper and lower background extraction slits)	8.2
bdist	300	distance to the center of the background extraction slit from the center of the gross spectrum (pixels).	8.2
bdistL	0	bdist for upper background spectrum	8.2
bdistU	0	bdist for lower background spectrum	8.2
bupper	0	y position of the upper background spectrum	8.2
blower	0	y position of the lower background spectrum	8.2
trace	1	Display tracing of the location of the point source extraction (0 - no, 1-yes)	8.1
woffset	0	Wavelength offset in pixels. Typically determined using STIS_WAVECAL (section 13.0)	8.4
targoffs	0	Target offset from center of the slit (arcsec)	8.4
b_median	9 (CCD) 0 (MAMA)	Median filter width used for smoothing the background spectra	8.3
b_mean1	1 (first order) 31 (echelle)	Mean filter width used for smoothing the background spectra	8.3
b_mean2	1 (first order) 31 (echelle)	Mean filter width used for smoothing the background spectra	8.3
b_order	-1 (echelle) 3 (first order)	Order of the polynomial used to fit the background spectra (-1 for no fit).	8.3
rate	1	Perform count rate conversion (0 - no, 1 - yes)	7.7
geomcorr	0	Apply camera mode geometric correction	7.12
subback	1	Subtract the background spectrum in point source spectral processing (0 - no, 1-yes)	8.3
timecorr	1	Apply sensitivity time/temperature correction	8.5
ctecorr	1	Apply CCD CTE correction for point source extractions.	8.5

Parameter Name	Default Value	Description	Sections
helio	1	Heliocentric correction of the wavelengths (0 - no, 1-yes)	8.6
biasfile	DEF	Name of the CCD bias image file	5.0, 7.6
ccdpar	DEF	Name of the CCD parameter table	5.0, 7.3, 7.7
darkfile	DEF	Name of the detector dark image file	5.0, 7.8
hpxtab	DEF	Name of the hot pixel table	5.0, 7.10
bpxtab	DEF	Name of the bad pixel table	5.0, 7.10
hotthresh	0.2 (CCD) 5e-5 (MAMA)	Dark rate threshold for flagging CCD pixel as hot (counts/second)	7.10
warmthresh	0.05 (CCD)	Dark rate threshold for flagging CCD pixel as warm (counts/second)	7.10
hrepair	0	Repair threshold for automatic repair of hot pixels (counts/second)	7.10
pfltfile	DEF	Pixel to pixel flat field file name	5.0, 7.9
lfltfile	DEF	Low frequency variation flat field file name	5.0, 7.0
dfltfile	DEF	Delta flat field file name	5.0, 7.9
geomfile	DEF	Camera mode distortion file	7.12
exttab	DEF	Spectral extraction position table	5.0, 8.1, 9.1
dctab	DEF	Dispersion coefficient table	5.0, 8.4,
msmtab	DEF	Correction table for the dispersion coefficients to account for the echelle monthly MSM offsets	5.0, 8.4
iactab	DEF	Incidence angle table for correcting for non-concentric aperture locations	5.0, 8.4
angletab	DEF	Table of longslit angles	8.4
scaletab	DEF	Table of image plate scales	8.4
senstab	DEF	Sensitivity table	5.0, 8.5, 9.2
pcttab	DEF	Slit GWIDTH photometric correction table	8.5
apptab	DEF	Aperture throughput table	5.0, 8.5
apdtab	DEF	Aperture description table	5.0, 8.4
esdeltaw	0	Extended source extraction wavelength spacing (0 to use value of esdeltas)	9.1
esdeltas	1	Extended source extraction pixel size in the wavelength direction for the output extended source image (in input pixels); Ignored if esdelaw is non-zero	9.1
esdeltal	1	Extended source extraction pixel size in the spatial direction for the output extended source image (in input pixels)	9.1

<b>Parameter Name</b>	<b>Default Value</b>	<b>Description</b>	<b>Sections</b>
esnsout	1,064	Number of samples in the output extended source image	9.1
esnlout	1,064	Number of lines in the output extended source image	9.1
escentw	0	Central wavelength of the output extended source image (default is the wavelength at the center of the input image)	9.1
escentl	511.5	Central line position of the output extended source image	9.1
eslogl	0	Output wavelengths on an equal delta log(wavelength) scale (0-no, 1-yes)	9.1
surfaceb	0	Convert extended source extraction to surface brightness flux units	9.2
esgwidth	600	Extraction region width (extraction slit height) for adjusting extended source flux levels	9.2
esps	1	Flag to specify that the correction for the point source slit throughput should be applied in an extended source calibration.	9.2
essnwave	0	Wavelength of isolated line for selection of the sensitivity value for extended source extraction flux calibration	9.2
geomrot	0	Camera mode rotation angle	7.12
geomxsiz	0	Number of columns in output geometrically corrected camera mode image (if not specified it is selected by routine depending on rotation and magnification)	7.12
geomysiz	0	Number of rows in output geometrically corrected camera mode image	7.12
geomxc	511.5	X-Pixel location in the input image to be mapped to the center of the output image	7.12
geomyc	511.5	Y-Pixel location in the input image to be mapped to the center of the output image	7.12
geommag	1	Magnification factor for output camera mode image	7.12
geomxmag	0	X- magnification factor for output camera mode image	7.12
geomymag	0	Y- magnification factor for output camera mode image	7.12

CALSTIS adds all of these parameters to the FITS header which is included in the output data files. In the case of reference file names with a value of DEF, CALSTIS replaces the DEF with the actual reference file name selected.

## 5.0 CALSTIS Reference Files

The following calibration reference files can be specified in the user parameters:

- biasfile** - CCD bias image
- ccdpar** - CCD parameter table
- darkfile** - detector dark rate image
- hpxtab** - hot pixel table
- bpxtab** - bad pixel table
- pfltfile** - pixel to pixel flat field
- lfltfile** - low frequency flat field
- dfltfile** - delta flat field
- geomfile** - camera mode geometric distortion maps
- exttab** - extraction position table (spatial distortion table for extended source calibrations)
- dctab** - dispersion coefficient table
- msmtab** - echelle dispersion coefficient correction table for the MAMA monthly offsets.
- iactab** - incidence angle correction table
- angletab** - table of angles for the long slits
- scaletab** - table of plate scales for each grating mode
- senstab** - sensitivity table
- pcctab** - sensitivity correction versus extraction GWIDTH
- apttab** - point source aperture throughput table
- apdtab** - aperture description table

The user can specify a *filename*, *DEF*, or *NONE* for the parameter value giving a reference file. A value of *NONE* indicates the processing step should be skipped. A value of *DEF* indicates that the default reference file should be selected for the observation's date and observing configuration. The default reference files can be found in the text file, *calstis\_ref.txt*, tabulated by reference file type, observing configuration, and Modified Julian Date (giving the use after date). The file *calstis\_ref.txt* can be found in the software library in a directory specified by the environmental variable, SCAL. SCAL also shows the location of where the calibration reference files are stored. On the STIS team computer systems at GSFC, *calstis\_ref.txt* can be found in:

*/stis0/data/cal/calstis\_ref.txt*

To limit the size of *calstis\_ref.txt*, it may contain pointers to other text files in the same directory.

## 6.0 CALSTIS Processing Steps

CALSTIS performs the following image processing steps on all observations to generate a flat fielded count rate image (details can be found in section 7):

- Conversion of MAMA data to Lo-Res
- Overscan bias level subtraction (CCD only)
- Statistical error initialization
- Cosmic ray rejection (CCD only for multiple readout observations when the keyword input /CR\_REJECT is set)
- Generation of the Doppler smoothing function for smoothing the dark rate file, the flat fields, and the bad pixel table files (MAMA observations only with Doppler correction enabled)
- Bias image subtraction (CCD only)
- Conversion to count rates (including the MAMA global non-linearity correction)
- Subtraction of a dark rate image
- Division by a flat field image
- Bad/hot pixel processing

For camera mode images, the following additional step is available:

- Correction for geometric distortion (Section 7.12)

For a point source spectral extraction, the following additional steps are performed (details can be found in section 8).

- point source spectral location
- gross and background spectral extraction
- background processing and subtraction
- wavelength assignment
- conversion to absolute flux units
- heliocentric wavelength correction

For extended source extractions (section 9) as indicated by the user parameter, *EXTENDED=1*, the following steps are performed:

- wavelength assignment
- heliocentric wavelength correction
- 2-D image rectification

## 7.0 Image Processing Steps

Routine image processing generates a flat fielded image in units of surface brightness. For the MAMA detector, the units are counts/second/Lo-res pixel. For the CCD, the units are electrons/second/unbinned CCD pixel. These units can easily be converted to other units using

available keyword values in the output header. For example, Let *H* be the output header from CALSTIS and *IMAGE* be the flat fielded image. To convert MAMA data to counts/Lo-Res Pixel use:

$$IMAGE = IMAGE * s\text{par}(H, 'EXPTIME')$$

To convert CCD data to counts/original binned pixel size use:

$$\begin{aligned} b\text{inx} &= s\text{par}(H, 'BINAXIS1') > 1 \\ b\text{iny} &= s\text{par}(H, 'BINAXIS2') > 1 \\ g\text{ain} &= s\text{par}(H, 'CCDGAIN') \\ e\text{xptime} &= s\text{par}(H, 'EXPTIME') \\ IMAGE &= IMAGE * b\text{inx} * b\text{iny} * e\text{xptime} / g\text{ain} \end{aligned}$$

The following sections describe the image processing steps performed on all data.

## 7.1 Conversion of MAMA Data to Lo-Res

MAMA data are taken in Hi-Res mode, but typically reduced after conversion to Lo-Res. The conversion is done by binning the raw 2048 x 2048 Hi-Res array to a 1024 x 1024 array. Each Lo-Res pixel contains the total counts (not the average) of 4 Hi-Res pixels.

## 7.2 Overscan Subtraction

For all CCD observations, CALSTIS subtracts a bias level computed from the serial overscan. The image is first rotated to detector coordinates so that the trailing overscan is at the right side of the image and the parallel overscan is at the top of the image. A bias level is computed for each row of the image by taking the median data value in the trailing serial overscan. The following table gives the pixel range used for the various binning factors (keyword BINAXIS1) in the serial direction (Note: the first pixel in the row is pixel 0):

<u>Serial Bin Factor</u>	<u>Overscan pixels</u>
0	1043 to 1061
1	1043 to 1053
2	522 to 531
4	261 to 270

A serial bin factor of 0 indicates no binning in either direction. The median value of all overscan values (one value per image row) is computed. Values more than twice the median or less than half of the median are ignored. A linear fit versus row number is made to the remaining values and evaluated for each row of the image. The fitted overscan value for each row is then subtracted from all pixels within the row. The overscan regions are trimmed from the image and the image is then rotated back to the original data coordinates.

For full frame images (including binned ones), an overscan plane is created using both the serial and parallel overscan regions. The table below lists the pixel ranges used in the parallel direction:

<u>Parallel Bin Factor</u>	<u>Overscan pixels</u>
0	1024 to 1043
1	1024 to 1033
2	512 to 521
4	256 to 265

As with the serial overscan, a linear fit versus column number is made to the un-rejected overscan pixels. The overscan plane is computed using the serial overscan vector as the intercept for the last column in the plane. The parallel overscan vector is used to define the slope of each row. The overscan plane is subtracted from the image.

### 7.3 Statistical Error Initialization

CALSTIS assigns an error to each raw data value using the formula

$$Error = \sqrt{\frac{D}{Gain} + \left(\frac{\sigma_r}{Gain}\right)^2}$$

where,

*D* is the data value clipped to be greater than or equal to *MINERRV*.

*Gain* is the CCD gain factor

$\sigma_r$  is the CCD readout noise in electrons per second.

*MINERRV* is the minimum Poisson variance (default = 0)

The values of *Gain* and  $\sigma_r$  are extracted from the CCD parameter table controlled by user parameter CCDPAR. For MAMA observations, *Gain* = 1 and  $\sigma_r$  = 0.

The statistical initialization step also flags saturated CCD pixels with a data quality (epsilon) of 190. The saturation levels for the various CCD amplifier and gain settings are stored in the CCD parameter table, CCDPAR.

### 7.4 Cosmic Ray Removal

If the keyword parameter /CR\_REJECT is supplied in the calling sequence, CALSTIS will combine the multiple readouts of a CCD observation with cosmic ray removal using the routines STIS\_CR and CR\_REJECT. STIS\_CR formats the raw images into a data cube and a statistical

error cube. These are passed to CR\_REJECT which uses an iterative cosmic ray rejection algorithm.

## 7.5 Doppler Smoothing Function

When on-board Doppler compensation is used, each output data value can be accumulated from data taken at multiple detector pixel locations. In this case, it is necessary to “smooth” the input reference files.

- 1) The observation is divided into 1 second intervals from observation start to observation end.
- 2) The Doppler shift (in Hi-Res pixels) is computed for each 1 second interval by:

$$\text{Shift} = \text{Doppmag} \times \sin\left(\frac{2\pi T}{\text{Period}}\right)$$

where,

*Doppmag* is the Doppler magnitude in Hi-Res pixels (header keyword DOPPMAG)

*Period* is the orbital period (header keyword ORBITPER)

*T* is the time since the Doppler Zero time (header keyword DOPPZERO)

- 3) A smoothing vector *DS* of length  $\text{Doppmag} * 2 + 1$  is accumulated by adding 1 to  $\text{DS}(\text{Doppmag} + \text{Shift})$  for the shift computed for each one second interval. This is a histogram of the shifts.
- 4) *DS* is then normalized by the total number of 1 second intervals.

The Doppler smoothing function, *DS*, now contains the fractional amount of time that each shift was used and can be convolved with a Hi-Res calibration reference image to generate an appropriate Doppler smoothed reference image.

## 7.6 Bias Image Subtraction

A “super” bias image is subtracted from CCD observations as specified by the user parameter BIASFILE. The default bias image is selected by the values of the header keywords CCDAMP, CCDGAIN, CCDOFFST, BINAXIS1, and BINAXIS2. If the input image was a combination of readouts (cosmic ray removal), then the bias image is multiplied by the number of images combined (as stored in the header keyword NCOMBINE) before subtraction.

## 7.7 Conversion to Count Rates

The input image in counts is converted to count rates (counts/second/Lo-Res pixel for the MAMA detectors and electrons/second/pixel for the CCD) by dividing by the integration time, multiplying by the CCD Gain, and dividing by a binning factor. The binning factor has the value of BINAXIS1 times BINAXIS2 for the CCD, 1.0 for MAMA Lo-Res processing, and 0.25 for MAMA Hi-Res processing. The calibrated value of the CCD Gain is read from the CCD

parameter table, CCDPAR. Statistical error propagation is done by applying the same operations to the propagated error image.

MAMA observations are also corrected for a global non-linearity by multiplying each data value by  $e^{-\tau \cdot \text{true}}$ , where  $\tau$  is an input user parameter and  $\text{true}$  is the true global count rate. To turn off the MAMA global non-linearity correction, set the user parameter  $\tau$  to zero. Note, that the formula needs the true global count rate. Since we only know the observed count rate the true global count rate is computed iteratively from the observed global count rate by:

```

true = obs           ;start with the observed count rate
for i=1,10 do true = obs/exp(tau*true)

```

## 7.8 Dark Rate Image Subtraction

CALSTIS subtracts a dark rate image selected by the parameter *DARKFILE*. If the dark rate calibration reference image is for a different binning, it is rebinned to the same binning as the observation before subtraction. Normally, the reference files will be unbinned CCD dark rate and Lo-Res MAMA dark rate images. If Doppler compensation is enabled, the MAMA darks are expanded to 2048 by 2048 then smoothed by the Doppler smoothing function (section 7.5). If Lo-Res processing is being performed, the dark is then rebinned to Lo-Res before subtraction.

For data take after the switch to the redundant side 2 electronics (since July 2001) the CCD darks are adjusted for the CCD temperature. Side 2 does not have a functioning temperature sensor for the CCD, and so a constant current is applied to the thermoelectric cooler. This results in much larger temperature variations than were present during side 1 operations. CALSTIS uses the CCD housing temperature (OCCDHTAV) to estimate the CCD temperature. To account for the change in dark rate versus temperature, the darks are scaled by  $1+0.07 \cdot (\text{OCCDHTAV}-18.0)$ , where OCCDHTAV is the CCD housing temperature for the observation and the reference dark is adjusted for a CCD housing temperature of 18 degrees C. (see ST Sci Instrument Science Report STIS 2001-03, T. Brown, "Temperature Dependence of the STIS CCD Dark Rate During Side-2 Operations")

No changes to the propagated statistical errors are done during dark rate subtraction.

## 7.9 Flat Field Processing

The STIS flat fields consist of three components: a pixel-to-pixel flat field; a low frequency variation flat field; and a delta flat field. These three components are selected based on the values of the parameters: *PFLTFILE*, *LFLTFILE*, and *DFLTFILE*. CALSTIS combines the three components, optionally smoothes by the Doppler smoothing function, rebins the flats to the binning of the observation being processed, and divides the rebinned flat into the data and the associated statistical error array.

## 7.10 Bad/Hot Pixel Processing

The parameters ***BPXTAB*** and ***HPXTAB*** specify the bad pixel and hot pixel tables used by CALSTIS. ***BPXTAB*** contains columns ***AXIS1***, ***AXIS2***, and ***FLAG***, where ***AXIS1*** and ***AXIS2*** give the column and row of the bad pixel, and ***FLAG*** gives the data quality or epsilon value for the pixel. For MAMA observations, ***AXIS1*** and ***AXIS2*** are given in Lo-Res pixel coordinates. A given data quality flag is therefore propagated to 4 Hi-Res pixels during Hi-Res processing. CALSTIS flags each image pixel with the maximum ***FLAG*** value associated with the pixel (in the case of binned data, multiple detector pixels contribute a single data value). If the data quality value of the pixel is already greater than any of the flags in the bad pixel table, it is left as is. If Doppler compensation is enabled, bad pixel data quality flags are propagated to any data value in which the detector element contributed.

The hot pixel table contains columns, ***AXIS1***, ***AXIS2***, and ***RATE***. ***RATE*** gives the estimated dark rate for the pixel in counts or electrons per second. Two CALSTIS parameters, ***HOTTHRESH*** and ***WARMTHRESH*** are used to determine which pixels are flagged as HOT or WARM. Any pixels with a dark rate greater than the thresholds will be flagged with a quality value of 180 if HOT, and 150 if WARM.

CALSTIS also has the option to repair hot pixels using the values of the neighboring pixels. Care should be taken when interpreting results with the hot pixels repaired. The parameter, ***HREPAIR***, is used to specify hot pixel repair. If ***HREPAIR*** = 0.0, no hot pixel repair is performed.

If ***HREPAIR*** is greater than 0, all pixels with a dark rate in the hot pixel table greater than ***HREPAIR*** are replaced by the average of its 8 neighbors with data quality values less than 125. Repaired pixels are flagged with a data quality of 125. If all neighbors have data quality values greater than or equal to 125, the pixel value is set to 0 and the data quality is set to 252.

If ***HREPAIR*** is set to a value less than zero, The absolute value of ***HREPAIR*** is used as the threshold, but only the left- and right-hand neighbors (instead of all 8 neighbors) are averaged when repairing the value. ***HREPAIR*** < 0 is useful for repairing point source spectra.

Hint: For short exposures, many of the hot pixels have negligible effect on the data. Increase the values of ***HOTTHRESH*** and ***WARMTHRESH*** to avoid so many pixels being flagged as hot or warm. For example, you may want to set ***HOTTHRESH*** to the 10.0/(the exposure time) and ***WARMTHRESH*** to a value 3 times lower.

## 7.11 Image Processing Outputs

The calling sequence of CALSTIS with the outputs of the image processing steps shown in bold is:

*CALSTIS, id, params, h, m, wave, flux, epsf, errf, gross, back, net, pos, image, eps, err*

Where: *image* gives the flat fielded count rate image in units of counts or electrons/second/unbinned pixel; *eps* is the epsilon or data quality image containing coded values for each image pixel as described in appendix A; and *err* is a propagated statistical error image for *image*.

In the case where no spectral processing is performed (either a camera mode image or an image processed with parameter *nospec=1*), *flux, epsf, and errf* will have the same values as *image, eps, and err*.

The outputs of the image processing steps can also be automatically written to a FITS file using the **outimage** user parameter (see section 10).

## 7.12 Camera Mode Geometric Distortion

To turn on geometric distortion correction for camera mode images, set **GEOMCORR=1** (the default is **GEOMCORR=0**):

*calstis,id,'geomcorr=1',h,m,w,image,eps,err*

The output *image* will now be a geometrically corrected flat fielded count rate image. Note: for camera mode reductions, output parameters *m* and *w* are not used. The value of **GEOMCORR** has no effect on the processing of images not taken in a camera mode.

You can also have CALSTIS write the geometrically corrected image to a file by setting the parameter **OUTGEOM** to a file name or to DEF.

*calstis,id,'geomcorr=1,outgeom=myfile.fits'*

*or*

*calstis,id,'geomcorr=1,outgeom=def'*

If **OUTGEOM** is set to DEF, the output fits image file will have the name *gim\_<id>.fits*.

If you want both the flat fielded image before and after geometric correction, use:

*calstis,id,'geomcorr=1',h,m,w,gimage,eps,g,errg,g,b,net,pos,image,eps,err*

*gimage, epsg, and errg* will contain the geometrically corrected image and its associated data quality image and propagated statistical error image. *image, eps* and *err* will contain the flat fielded image without geometric correction. *m, w, g, b, net, and pos* are not used for camera mode images.

Note, that if *GEOMCORR* was not set to 1, *gimage* and *image* would both contain the same image (the flat fielded image without geometric correction).

Camera mode images can also be calibrated independently of CALSTIS by using the routine *CALSTIS\_GEOM*. Its calling sequence is:

*calstis\_geom, h, image, eps, err, image\_out, eps\_out, err\_out*

where *h* is the image header and *image*, *eps* and *err* are the input image, data quality, and errors. *image\_out*, *eps\_out*, and *err\_out* are the geometrically corrected results. For example:

*calstis, id, 'GEOMCORR=1', h, m, w, gimage, geps, gerr*

is equivalent to:

*calstis, id, 0, h, m, w, image, eps, err*  
*calstis\_geom, h, image, eps, err, gimage, geps, gerr*

To geometrically correct a raw image use:

*stis\_read, id, h, image*  
*calstis\_geom, h, image, 0, 0, gimage*

During the geometric correction of a camera mode image, you can optionally change plate scale, center pixel and rotation angle using the following parameters. This is useful for matching your data to the orientation and platescale of an existing image without having to perform another interpolation on the output from CALSTIS.

<i>geomrot</i>	camera mode rotation angle (default = 0.0)
<i>geomxsiz</i>	Number of rows in output camera mode image default = $1024 * \sqrt{2} * \text{geomxmag} * \max(\text{abs}([\cos(\text{geomrot}), \sin(\text{geomrot})]))$
<i>geomysiz</i>	Number of columns in output camera mode image default = $1024 * \sqrt{2} * \text{geomymag} * \max(\text{abs}([\cos(\text{geomrot}), \sin(\text{geomrot})]))$
<i>geomxc</i>	X Position of the center of the output image in the input image (default = 511.5)
<i>geomyc</i>	Y Position of the center of the output image in the input image (default = 511.5)
<i>geommag</i>	magnification factor for output image (default = 1)
<i>geomxmag</i>	x-direction magnification factor for output image (default = <i>geommag</i> )
<i>geomymag</i>	y-direction magnification factor for output image (default = <i>geommag</i> )

NOTES:

1. The magnification is applied after the rotation.
2. The header astrometry is correctly updated.
3. **geomxc** and **geomyc** are the center of a geometrically corrected raw image with no rotation or magnification. These parameters are useful for generating a fractional pixel shift.
4. The geometric correction, rotation, and magnification are all applied with only one bilinear interpolation of the image.

For Example, to geometrically correct a flat fielded count rate image with a 0.25 pixel shift in the x-direction:

```
calstis,id,'geomcorr=1,geomxc=511.75',h,m,w,image
```

To geometrically correct with a factor of 2 increased sampling:

```
calstis,id,'geomcorr=1,geommag=2',h,m,w,image
```

To geometrically correct an image with a 10 degree rotation, 1.01 magnification in the y-direction, and a 1024x1024 output image size:

```
par = 'geomcorr=1,geomrot=10.0,geomymag=1.01,geomxsiz=1024,geomysiz=1024'  
calstis,id,par,h,m,w,image
```

To geometrically correct with cubic interpolation:

```
calstis,id,'geomcorr=1',h,m,w,image,eps,err,sens,x,y,junk,fimage  
cimage = interpolate(fimage,x,y,/cubic,missing=0)
```

*fimage* contains the flat fielded image, *image* contains the geometrically corrected image using bilinear interpolation, and *cimage* contains the geometrically corrected image using cubic interpolation.

## 8.0 Point Source Spectral Extraction

### 8.1 Point Source Spectral Location

The first step in point source spectral extraction is to locate the center of the spectrum in the spatial direction. The reference file *exttab* contains tracings (spatial distortions) of the spectrum at various image lines (positions) for the first order gratings and tracings for each spectral order at the nominal position for echelle gratings. For the first order spectral modes, line position 512 is used for the nominal position. The default *exttab* is selected by the optical element and central wavelength of the observation. The spectral location is found as follows.

The nominal position is modified by an optional user supplied offset, parameter *soffset*. *Soffset* should be supplied in pixels (Lo-Res pixels for the MAMA even if a Hi-Res reduction is being performed). This offset is added to the nominal spectral location to define the center of the region to search for the spectrum. Figure 8.1 shows the shift of the nominal position (lower dotted line) to the new position (upper dotted line).

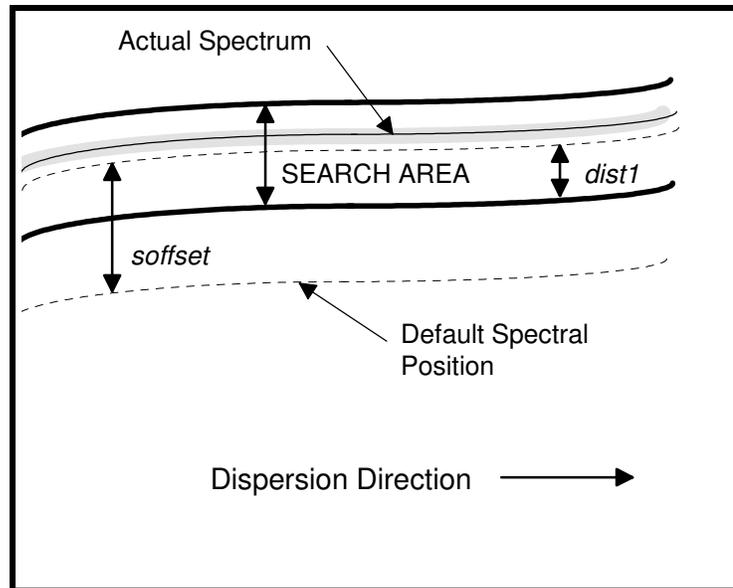


Figure 8.1  
Point Source Spectral Location

The spectral tracing at the center of the search area (upper dotted line in figure 8.1) is used as a spectral template. The region plus or minus the parameter *dist1* about the center of the search area is searched by cross correlation with this template. This region is between the two dark solid lines in Figure 8.1. At each line position in this region, the total counts along the tracing shifted in increments of 1 pixel in the spatial direction are summed. The maximum sum, computed to sub-pixels by quadratic refinement, defines the center of the point source spectrum (solid line at the center of the spectrum in Figure 8.1). For echelle spectra, the correlation is made using all orders simultaneously to obtain a single offset.

Once CALSTIS has completed the spectral location, it will display the image with an overlay so that the user can verify that the spectrum was properly located. This overlay will show the regions to be extracted as the gross and the background spectra. To turn off this display, set the user parameter *trace* to 0. For weak spectra, particularly when residual cosmic rays or hot pixels are present, spectral location may fail, and the trace display is a very useful verification tool.

In the event that the spectral location does fail, the user can recover using a number of strategies.

1. Limit the size of the search area by specifying a smaller value for *dist1*.
2. Supply an approximate offset using *soffset* to allow *dist1* to be decreased.
3. If all else fails, set *dist1* to 0 and specify the final center of the spectrum using *soffset*.

The following example shows the use of the point source location parameters with a user supplied offset of 75 and a small search area of 10 pixels.

```
calstis,id,'soffset=75,dist1=10',h,m,wave,flux
```

Note: For the E1 aperture positions (e.g. 52X0.1E1) used to minimize the effect of the CCD charge transfer inefficiency, the default value of SOFFSET is set to 390 (the approximate location of the spectrum near the top of the detector).

## 8.2 Gross and Background Spectral Extraction

Figure 8.2 illustrates the various parameters which control the point source extraction. For each spectral order, three extractions are made illustrated by the regions between the three sets of solid lines in figure 8.2. The gross spectral region has a width in the spatial direction of *gwidth* and is centered on the spectral order shown by the dotted line determined in section 8.1. The upper background spectral region has a width of *bwidth* and is centered *bdist* above the center of the gross spectral region. The lower background region also has a width of *bwidth* and is centered *bdist* below the center of the gross spectral region. For echelle data, *bdist* is not used. Instead, the upper and lower background spectra are centered exactly between the spectral orders. Thus, the upper background for one order is the lower background for the order immediately above it. *Gwidth*, *bwidth*, and *bdist* are always specified in units of unbinned CCD pixels or Lo-Res MAMA pixels. They are not restricted to be integers.

In some cases it may be desirable to have different distances to the upper and lower background regions, or to specify absolute background positions. If you want different distances to the upper and lower background position use parameters *bdistL* and *bdistU* to specify the lower and upper distances. *bdistL* and *bdistU* override the value of *bdist*. Alternately, you can specify the absolute y-position of the center of background regions by parameters *blower* and *bupper*. For example the following would extract the background from y positions 300 and 650:

```
calstis, id, 'blower=300,bupper=650',...
```

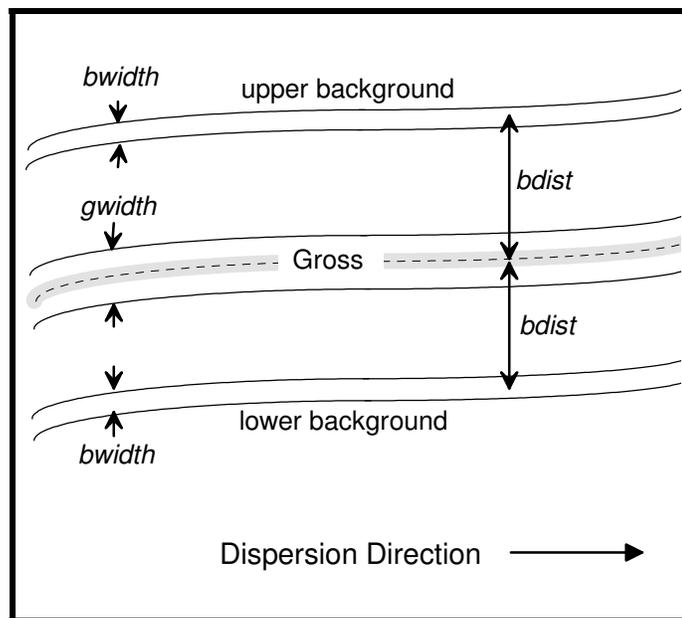


Figure 8.2  
Point Source Spectral Extraction Parameters

Setting values for *blower* and *bupper* will override any values set for *bdist*, *bdistU*, *bdistL*. Note: Background regions may have a slope or curve to them. *blower* and *bupper* will refer to the y-position at the center of the detector (i.e, x=512).

#### HINTS:

- 1) Setting either *blower* and *bupper* to a negative value can be used to set both background to be on the same side of the spectrum.
- 2) Setting a background region such that it is off of the detector (e.g. *bdistU* = 1025) can be used to force CALSTIS to use only an upper or lower background region and not both.

For each sample position in the dispersion direction, an extraction slit is constructed for the gross and two background spectra. This slit is illustrated in Figure 8.3. The count rates within the slit are summed. The upper and lower ends of the slit may cover only a portion of a pixel. In this case the count rate is weighted by the area of the pixel covered. As the sum of the count rates in the extraction slit is computed, the statistical errors are propagated using the same weights.

The output of this extraction is normalized by the binning factor in the spatial direction to give units of counts/second/pixel in the extraction slit.

The data quality value of the extracted data is set to the largest data quality value (the worst condition) of all pixels within the extraction slit. If a portion of the spectral order is outside of the image, the data quality values for that portion of the spectrum are set to 254.

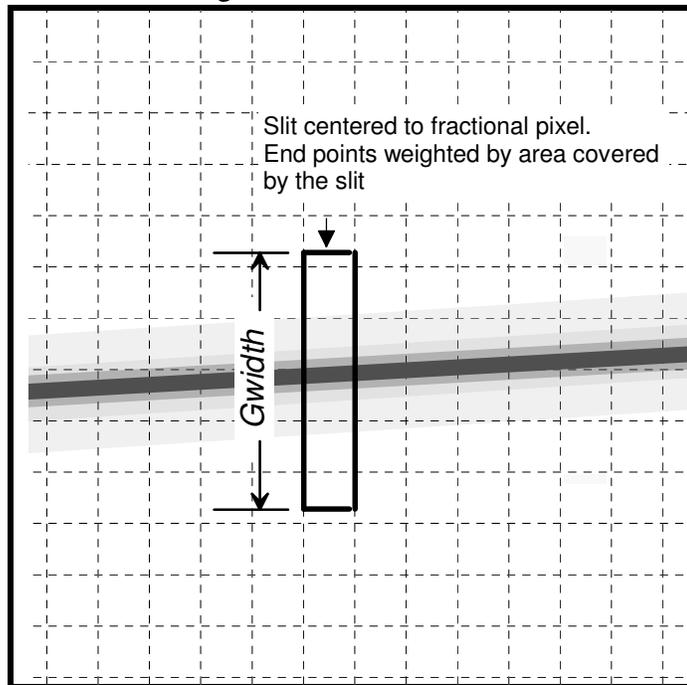


Fig. 8.3 Point Source Extraction Slit

### 8.3 Background Processing and Subtraction

Before the background is subtracted from the gross spectrum, the upper and lower backgrounds are averaged, smoothed, and then normalized to account for differences in the size of *gwidth* and *bwidth*. When averaging the upper and lower background, if they do not have the same data quality value, the better of the two values is used.

CALSTIS has a multi-step smoothing process for the background. It performs a median filter, followed by two mean filters, followed by a polynomial fit to the background. Any or all of the smoothing steps can be disabled. The median filter width is specified by the parameter ***b\_median***. A value of 0 or 1 indicates no median filtering. The widths of the mean filters are controlled by parameters ***b\_mean1*** and ***b\_mean2***. Again a value of 0 or 1 will disable the filtering. The final optional smoothing is a polynomial fit to the background of each spectral order. This is controlled by the parameter ***b\_order***. Setting ***b\_order*** to -1 disables the polynomial fitting. If ***b\_order*** is set to 0, all data points in the smoothed background are set to the average of the unsmoothed background values. When fitting a polynomial to the background, the regions of the airglow lines at 1215.6, 1302.1, 1304.8, and 1306.0 are ignored in the fitting process for modes G140L, G140M, and the PRISM. These regions are left unsmoothed in the fitted background vector. This allows proper subtraction of the airglow lines from the gross spectrum.

The background is normalized to the area of the extraction slit of the gross spectrum by multiplying it by  $gwidth/bwidth$  and then subtracting it from the gross spectrum to give the net spectrum. The errors in the background spectrum are considered to be negligible after smoothing. Only the areas in the unsmoothed airglow lines are propagated to the net spectrum. Errors for all other data points in the net spectrum are the same as the ones in the gross spectrum.

## 8.4 Wavelength Assignment

CALSTIS assigns wavelengths using the dispersion relation:

$$s = a_0 + a_1 m \lambda + a_2 m^2 \lambda^2 + a_3 m + a_4 \lambda + a_5 m^2 \lambda + a_6 m \lambda^2 + a_7 m^3 \lambda^3 + a_8 m^2 + a_9 \lambda^2$$

where:

- s* is the pixel location in the dispersion direction,
- m* is the spectral order (1 for the first order gratings),
- $\lambda$  is the wavelength, and
- a*<sub>0</sub>, *a*<sub>1</sub>, *a*<sub>2</sub>, ... are dispersion coefficients

The dispersion coefficients are read from the table specified by user parameter ***dctab***. The default dispersion coefficient table is selected based on optical element and central wavelength. Within each table, the dispersion coefficients are tabulated as a function of image position along the slit (image row or line number). If the desired image line number is not in the table, linear interpolation between the coefficients versus line or slit position is used.

The dispersion coefficients must be adjusted for the movements of the echelle format resulting from the monthly MSM offsets. For each central wavelength,  $\lambda_c$ , a coefficient, *a4cor*, is stored in the reference file (msmtab), along with information of the reference location of the spectral format. The dispersion coefficients are adjusted for the y-motion of the format,  $\Delta y$  (in lores pixels) from the reference position by:

$$a_4 = a_4 + a4cor * \Delta y$$

$$a_0 = a_0 + a4cor * \lambda_c * \Delta y$$

The dispersion coefficients must be corrected for any offset due to the slit non-concentricity. A table of relative slit positions, *apdtab*, and a table of slit angles for the long slits, *angletab*, are used to determine the offset,  $\delta$ , between the wavelength calibration *dctab* slit and the observation's slit in arcsecs.

$$\delta = \delta_0 \quad \text{short slits}$$

$$\delta = \delta_0 + P(y - y_c)(\tan(\theta_{obs}) - \tan(\theta_{dc})) \quad \text{long slits}$$

Where  $\delta_0$  is the slit offset in arcsec at the center of the format from table *apdtab*,  $P$  is the platescale in the dispersion direction (arcsec/pixel) from the table *scaletab*,  $y$  is the line position in the observation,  $y_c$  is the y-position for the center of the format adjusted for any monthly MSM offsets,  $\theta_{obs}$  is the slit angle for the observations slit, and  $\theta_{dc}$  is the slit angle for the wavecal observation's slit used to construct the dispersion coefficient file. If the target is not centered in the slit (user parameter *targoffs* is not zero),  $\delta$  is also adjusted by adding the value of *targoffs* to it.

The table, *iactab*, contains the coefficients  $C_i$  and  $D_i$  which are used to adjust the dispersion coefficients by the relations:

$$a_i = a_i + C_i \delta \quad \text{for each } a_i$$

$$a_1 = a_1 + D_1 \delta + D_2 \delta^2$$

Additionally  $a_0$  is adjusted for MSM non-repeatability and thermal motion by the auto-wavecal (Section 13) or by the user supplied offset, parameter *woffset*, given in pixels:

$$a_0 = a_0 + woffset$$

## 8.5 Conversion to Absolute Flux Units

For point source spectra, taken with the CCD, the first step in conversion to absolute flux units is to correct for changes in the CCD charge transfer efficiency. A description of the algorithm used for this correction can be found on the [www.stsci.edu](http://www.stsci.edu) web site in STIS ISR 2003-03, Ralph Bohlin and Paul Goudfrooij, "An Algorithm for Correcting CTE Loss in Spectrophotometry of Point Sources with the STIS CCD." The CTE correction can be turned off by setting the user parameter *ctecorr* to 0.

Point source spectra are converted to absolute flux units using the sensitivity table selected by the parameter *senstab*. To turn off conversion to absolute flux units use:

*calstis,id,'senstab=none',...*

For echelle observations, separate sensitivity curves are stored in *senstab* for each spectral order. These curves contain the echelle blaze function which shifts (in wavelength) with changes in the y position of the spectral format (monthly MSM offsets) and with time. Coefficients describing this shift are stored in *senstab*. If this blaze shift is not correctly removed, the resulting flux will show a discontinuity between spectral orders. If this discontinuity is significant, it may be possible to improve the calibration using the data itself, set the user parameter, *auto\_bs=1*, and calstis will attempt to determine the blaze shift to match the flux levels in the overlap regions between the spectral orders.

**Warning:** *The current sensitivity tables do not have accurate blaze calibrations for intermediate echelle central wavelengths. The tables also have no information for the intermediate central wavelengths to shift the blaze function with time and with y-position of the spectral format. Results for these central wavelengths may be improved using *auto\_bs=1*.*

If the target aperture is different than the aperture of the sensitivity table, *senstab*, CALSTIS corrects the sensitivity for the relative point source aperture throughput between the *senstab* aperture and your target aperture. The relative point source throughputs are stored in a table selected by *apptab*. It also adjusts the sensitivity for your specified GWIDTH using the adjustment tables in the *pcttab* table.

STIS sensitivities are adjusted for changes with time and temperature as described in ST ScI STIS Instrument Science Report 99-07, R. Bohlin, "*Changes in Sensitivity of the Low Dispersion Modes*". The current version of CALSTIS uses time and temperature corrections updated by R. Bohlin in August, 2003. To turn off these corrections in calstis, set user parameter, *timecorr*, to 0.

CALSTIS uses linear interpolation between wavelengths in both the sensitivity table and the relative aperture throughput table. It does not extrapolate either table. The flux values of data points outside of the wavelength range of the tables are set to 0 and their data quality values are set to 251.

## 8.6 Heliocentric Wavelength Correction

After the sensitivity correction is completed, CALSTIS will apply a heliocentric correction to the wavelengths provided that the user parameter *helio* is not set to 0. The Earth's velocity toward the target,  $v$ , is computed (in km/sec) and added to the output FITS header in keyword EARTHVEL. Wavelengths are then multiplied by the factor  $1+v/c$  where  $c$  is the speed of light. If *helio* is set to 0, CALSTIS will still populate the value of the keyword EARTHVEL but will leave the wavelengths uncorrected.

When computing sensitivity functions from observations of standard stars, *helio* should be set to zero:

*calstis,id,'helio=0,senstab=none',h,m,w,f*

The wavelengths, *ws*, of the standard star flux table should then be converted to the velocity frame of the observation by:

*ws = ws/(1+sxpar(h,'earthvel')/2.99792458D5)*

CALSTIS **does not** convert wavelengths to an AIR reference frame. If you need air wavelengths, run the routine *vactoir* on the output wavelengths of CALSTIS:

*vactoir,w*

## 8.7 Point Source Spectral Extraction Outputs

The calling sequence of CALSTIS with the outputs of the point source extraction processing shown in bold is:

*CALSTIS, id, params, **h, m, wave, flux, epsf, errf, gross, back, net, pos, image, eps, err***

*h* is the output FITS header and includes all the keywords present in the input data set, a set of keywords giving the values of all parameters specified by *params* updated to reflect the selected reference file names, history records describing the processing, and a few additional keyword values added by the processing steps of CALSTIS. This header can be examined with the IDL routine *hprint*:

*hprint,h*

Individual keyword values extracted for use in post-reduction analysis can be examined with *sxpar*:

*value = sxpar(h,'<keyword name>')*  
or  
*print,sxpar(h,'<keyword name>')*

*m* is the spectral order numbers. For a first order grating extraction, *m* will have the scalar value of 1. For echelle observations, *m* will be a vector of spectral order numbers for the spectral orders extracted.

*wave* contains the wavelengths. For first order grating mode point source extractions, *wave* will be a 1024 point vector (2048 point vector for Hi-Res extractions). For echelle mode extractions, *wave* will be a 2-dimensional array of size 1024 or 2048 by *N*, where *N* is the number of spectral

orders extracted. *wave(\*,i)* will have the wavelengths for spectral order *m(i)*. *wave* is a vector for first order grating extended source extractions. Assuming all processing steps are executed, *wave* will be heliocentric wavelengths in Angstroms.

*flux* is the calibrated flux vectors or calibrated image. For first order grating point source extractions, *flux* will be a vector with the same length as *wave*. For echelle point source extractions *flux* will be an array of the same size as *wave* where *flux(\*,i)* contains the calibrated spectrum for order *m(i)*. Assuming all processing steps are executed, the output units of *flux* for point source extractions is ergs/sec/cm<sup>2</sup>/Angstrom. If conversion to absolute flux is not performed, *flux* will be identical to *net*.

*epsf* is the data quality or epsilon value array for *flux*. It has the same size as *flux* and is coded with the values given in Appendix A.

*errf* is the propagated statistical error array for *flux* and has the same size and units as *flux*. It contains the statistical errors (as defined in section 7.3) propagated through the calibration process. It does not include statistical errors introduced from the statistical or systematic errors in calibration reference files.

*gross* gives the extracted gross spectrum as described in section 8.2. It has the same size and dimension as *wave* and *flux* and is in units of counts/second/pixel in the dispersion direction.

*back* is an *ns* by *N+1* array with the upper and lower background vectors, where *ns* is the number of elements in the spectra and *N* is the number of spectral orders. *back(\*,i)* and *back(\*,i+1)* give the lower and upper background spectra for *gross(\*,i)*. *back* is in the same units as *gross* but has **not** been normalized to the slit size of *gross* (section 8.2).

*net* is the *gross* minus the smoothed average of the upper and lower background spectra. It has the same size and units as *gross*. *net* is not used for extended source reductions. CALSTIS does not return the smoothed background. It can easily be obtained after running CALSTIS by using:

$$\mathit{smoothed\_background} = \mathit{gross} - \mathit{net}$$

*pos* is an array of the positions of the center of the extraction slit for the gross spectrum. It has the same size as *gross* and is defined such that the center of the extraction slit for *gross(i,j)* is at raw data pixel *i*, *pos(i,j)*. For example, if you want to overplot the positions of the spectral orders used for extraction, use:

```
tvsc1,alog10(image>0.1)
for i=0,N-1 do plots,findgen(1024),pos(*,i),/dev
```

## 8.8 Echelle Scattered Light Correction

CALSTIS uses the interorder flux to estimate the dark rate and scattered light in a spectral order. Although this is a good first approximation, it can result in either under or over subtraction of the

background depending on the flux distribution of the source. A routine, ECHELLE\_SCAT (which can be run after CALSTIS), constructs a scattered light model, removes it from the image, and re-extracts the spectrum. To run the routine, first run CALSTIS with parameter *outimage* set to DEF, followed by ECHELLE\_SCAT.

```
calstis, id, 'outimage=def,...', h, m, w, f,...
echelle_scat, id
```

*outimage=def* tells CALSTIS to write the flat fielded count rate image to a file with the name **im\_<id>.fits**. ECHELLE\_SCAT uses this file and the output extracted spectral file to remove compute a model image and a scattered light model image. The scattered light model is subtracted from the flat fielded count rate image (file **im\_<id>.fits**). A new spectrum is extracted from the difference image using the same extraction parameters specified in the original call to CALSTIS. The extracted spectral output is then written to a FITS binary table named **newspec\_<id>.fits**. This file has the same format as the original **spec<id>.fits** written by CALSTIS (see section 10.3).

#### IMPORTANT NOTE:

- ❖ *Echelle\_scat requires a lot memory. You should probably have at least 512 MB of memory on the machine to run it. You can use less but it may significantly slow the process down.*
- ❖ *Execution time is on the order of 10 to 15 minutes.*

Briefly, the method used by ECHELLE\_SCAT is as follows:

- 1) *Using the original net flux produced by CALSTIS, create an estimated source flux distribution by correcting each spectral order for the echelle blaze and merging the orders. Use this as a first guess of the flux distribution.*
- 2) *Perform the following iteration on the flux distribution (3 iterations are currently used)*
  - *Using the current flux distribution, construct a model light image using reference models for the echelle scatter function, telescope PSF, detector halo and grating isotropic scattering, the cross disperser scatter function, and detector ghosts.*
  - *Extract a net spectrum from the model using the same parameters specified in the original call to CALSTIS. Use this difference to adjust the current flux distribution and repeat the iteration.*
  - *If it is the last iteration, also construct an object model image (without scattered light) by using the core of the echelle scattered light function (light within 5 pixels of the center) and the 13 x13 pixel core of the combined telescope PSF, detector PSF,*

*isotropic grating scatter, and cross-disperser scatter. Subtract this from the model image to give a model scattered light image.*

*3) Subtract the model scattered light image from the original flat fielded count rate image. Extract the spectrum from this difference and flux calibrate using the same parameters specified in the original call to CALSTIS.*

If, in addition to constructing a scattered light corrected spectrum, you would like to see the final model images, use the keyword parameter */model\_file* in the call to ECHELLE\_SCAT.

*echelle\_scat,id,/model\_file*

ECHELLE\_SCAT will then write a FITS file, **model\_<id>.fits** containing two extensions with names **MODEL** and **SCAT**. To read these images from the file use:

```
fits_read, 'model_<id>.fits', model, header, extname='MODEL'
fits_read, 'model_<id>.fits', scat, header, extname='SCAT'
```

Where **<id>** is replaced with the observation id number or name.

## 9.0 Extended Source Spectral Extraction

CALSTIS extended source spectral extractions are done by including *extended=1* in the parameters:

```
calstis,id,'extended=1...', h, m, wave, flux, epsf, errf, sens,x,y, unused, image,eps,err
```

Extended source spectral extraction is used to create a two dimensional map with wavelength in one dimension and position along the slit in the other dimension. Extended source extraction is appropriate for both extended sources observed in a long slit or for a slitless spectra. Fig. 9.1 illustrates a spectrum taken in the long slit mode. Optical/detector distortions have been exaggerated. Pixels of constant wavelength and pixels of constant spatial position do not lie in a straight line due to optical and detector distortions.

### 9.1 Extended Source Image Rectification.

CALSTIS will generate a rectified image with wavelength running from left to right (horizontally) and spatial position (in the direction along the slit) running vertically. The rectification uses the dispersion coefficients (which vary with position along the slit) and the spectral tracings table, EXTTAB (which gives the spatial distortion at various positions along the slit), to compute the position of each pixel in the output rectified image in raw data coordinates (output parameters *x, y*). The value for each output pixel is then computed using bilinear interpolation in the flat fielded input image. Wavelengths are computed using the same

dispersion relation as described in section 8.4 and are converted to the heliocentric system as described in section 8.6.

The default size of the rectified image is 1064 x 1064. This has been increased from 1024 x 1024 so as not to lose any information at the edges of the image during rectification. In the default rectification, the output wavelengths are equally spaced using the average dispersion. The default spacing in the cross-dispersion direction is equal to one pixel in raw image coordinates.

A number of parameters allow the user to control the size of the output rectified image, the spacing between output pixels, and the center position of the output image in wavelength and spatial position along the slit.

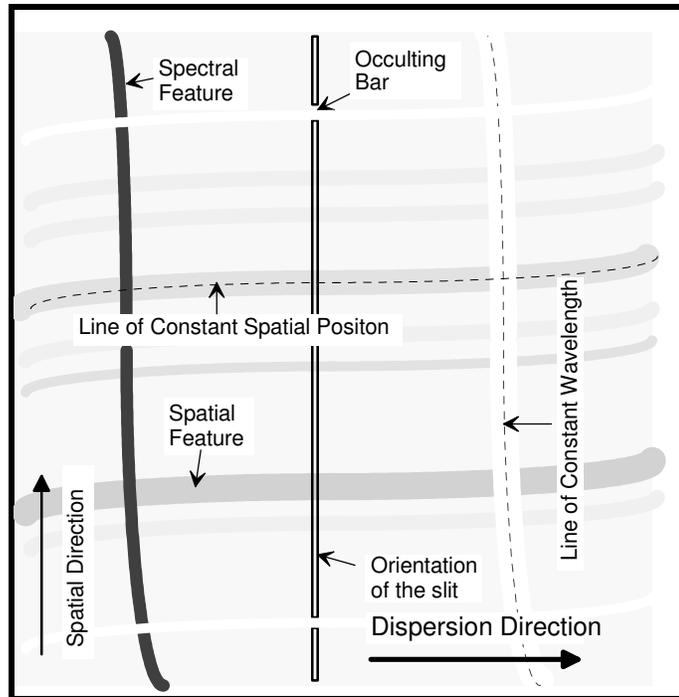


Figure 9.1  
Long Slit Extended Source Spectrum

The size of the output image is controlled by parameters *esnsout* and *esnlout*. For example, to make the output image 1024 x 1024, use:

```
calstis,id,'extended=1,esnsout=1024,esnlout=1024',...
```

The spacing between pixels can be controlled by the parameters: *esdeltas*, *esdeltal*, *esdeltaw*, and *eslogl*. *esdeltal* and *esdeltas* give the sample spacing of the output image in the cross-dispersion direction and the dispersion direction, respectively. They are specified in units of raw data pixels. They can be used to change the spacing of the rectified image such that the arcsec/output pixel is the same in both directions or to simply increase or decrease the sampling interval. *esdeltaw* (specified in Angstroms) can be used to specify the sampling spacing in the output image in the dispersion direction. If *esdeltaw* is specified, then *esdeltas* is ignored.

The center of the rectified image can be controlled by the parameters: *escentl* (spatial center in raw image coordinates); *escents* (center in the dispersion direction in raw data coordinates); and *escentw* (central wavelength). If *escentw* is specified, then *escents* is ignored. These parameters, along with the output image size parameters, can be used to extract multiple emission lines from the same or different observations on the same output image scale to allow easy comparison of the different spectral lines.

The output wavelengths of the extended source extraction are on a linear scale (constant delta wavelength between pixels). In some cases, it may be advantageous to place the wavelengths on a constant radial velocity scale (constant  $\log(\lambda)$ ). This is done by setting the parameter *eslogl* to 1. When *eslogl* is set to 1, the spacing between pixels is such that the  $\log_{10}$  of the wavelength is equal to the average difference of the  $\log_{10}$  of the maximum wavelength and  $\log_{10}$  of the minimum wavelength divided by 1024 times *esdeltas*. Alternatively, you can specify the spacing when *eslogl* is set to 1 by setting *esdeltaw* to the  $\log_{10}$  of your desired spacing.

## 9.2 Extended Source Flux Calibration

You have three choices for flux units from CALSTIS when run in extended source mode:

- 1) counts/sec/pixel (count rates)  
*CALSTIS,id,'extended=1,senstab=none',...*
- 2) ergs/cm<sup>2</sup>/sec/Ang/cross-dispersion detector pixel (standard CALSTIS point source flux units)  
*CALSTIS,id,'extended=1',...*
- 3) ergs/sec/cm<sup>2</sup>/Ang/Arcsec<sup>2</sup> (surface brightness units/ specific intensity)  
*CALSTIS,id,'extended=1,surfaceb=1'*

In default mode, CALSTIS will convert extended source extractions to absolute flux in units of ergs/cm<sup>2</sup>/sec/Ang/cross-dispersion detector pixel, where a detector pixel is a Lo-Res pixel for the MAMA detector. CALSTIS converts the rectified count rate image (counts/second/pixel) to flux units by dividing by the calibrated point source sensitivity function adjusted to an "infinite" extraction slit height (i.e. GWIDTH = 600) using table *apptab*.

To extract a point source-like spectrum from the rectified flux image (with units of ergs/cm<sup>2</sup>/sec/Ang), first find the image lines (rows) in the rectified image containing the object. Call them *L1* and *L2*. Integrate the lines:

```
CALSTIS,id,'extended=1', h, m, w, f, eps, err
ext_flux = total(f(*,L1:L2),2)
```

or if you used a non-default value for *esdeltal*:

```
ext_flux = total(f(*,L1:L2),2)*sxp(h,'esdeltal')
```

*edeltal* will be 1.0 for the default extraction. It contains the cross-dispersion spacing between rectified image pixels in raw detector pixel units.

This extraction suffers from a couple of possibly serious problems:

**PROBLEM 1)** No background has been subtracted. Errors in the dark subtraction or contamination by other bright sources can affect the flux levels. To combat this problem, estimate the background in nearby regions (in the y direction) that do not have objects present. For example:

```
back_flux = total(f(*,L3:L4),2)/(L4-L3+1) * (L2-L1+1)
  <at this point you may want to smooth back_flux>
ext_flux = (total(f(*,L1:L2),2)-back_flux)*sypar(h,'esdeltal')
```

**PROBLEM 2)** The measured flux for the object will depend on the extent in y-direction (the range of L1 and L2) of the extraction slit. In point source extraction mode, CALSTIS refers to this as the **GWIDTH**. The default extended source extraction assumes (GWIDTH=600) which is considered as a "infinite" gwidth for calibration purposes. To adjust sensitivity calibration for the width of the extraction region compute:

```
gwidth = (L2-L1+1)*sypar(h,'edeltal')
print,gwidth
```

Now, before extracting the spectrum, tell CALSTIS that you are going to be using this extraction gwidth by:

```
CALSTIS,id,'extended=1,esgwidth=<gwidth>', h, m, w, f, eps, err
```

where *<gwidth>* is the value you computed above.

If you are going to extract multiple objects with different *gwidths* you may not want to have to keep re-running CALSTIS. Alternately you can make the correction after running CALSTIS without specifying *esgwidth*. Get the sensitivity function for an infinite slit and for your extraction slit:

```
CALSTIS_SENS, h, msens, wsens1, sens1, gwidth=gwidth
linterp, wsens1, sens1, w, int_sens1 ;interpolate to observation
CALSTIS_SENS, h, msens, wsens2, sens2, gwidth=600
linterp, wsens2, sens2, w, int_sens2
```

Adjust the extracted flux to your extracted slit size:

```
ext_flux = ext_flux*int_sens2/int_sens1
```

Note: the *gwidth* correction becomes more and more inaccurate as *gwidth* decreases for values less than 5.

**PROBLEM 3)** The default for extended source calibration does not correct for the point source throughput of the slit. After all, in extended source calibrations, you do not necessarily

have point sources centered in the slit. This correction is important for apertures with widths less than or equal to 0.2 arcsec. To tell CALSTIS to perform this correction, set parameter *esps=1*:

*CALSTIS,id,'extended=1,esps=1...*

Note however, this correction will be inaccurate if the target is not centered in the slit or if the target has any significant spatial extent (a blob).

**PROBLEM 4)** During extended source calibrations, no CCD CTE correction is done. The CTE correction done by CALSTIS for point source extractions assume that the target is an unresolved point source (star) and that no other target is present in the slit. Since you are doing an extended source extraction, this is probably not the case and a valid CTE correction would be extremely difficult.

An extended source calibration keyword, *essnwave* (containing a wavelength for sensitivity calibration), can be used to calibrate the entire extended source rectified image with the same sensitivity value (the value at *essnwave*). Suppose the rectified image is a slitless spectrum of an emission line source. If portions of the source are offset spatially in the dispersion direction, they will appear in the rectified image at different wavelengths when in fact they are approximately the same wavelength. In this case, the sensitivity correction should be made using the actual wavelength of the source. For example, an emission line source at 2500 Angstroms could be calibrated by:

*CALSTIS,id,'extended=1,senstab=def,essnwave=2500.0',h,m,w,f*

### 9.3 Non-linear interpolation in Extended Source Extractions:

CALSTIS uses bi-linear interpolation to resample extended source extractions onto a linear wavelength (x) and spatial (y) grid. To use alternate interpolation methods you must do it yourself. However, CALSTIS does supply outputs to make this job easier for you. First run CALSTIS with your desired parameters and supply all output variables:

*CALSTIS,id,'extended=1',h,m,w,f,epsf,errf,sens,x,y,junk,image,eps,err*

The important output parameters are:

*image* - the flat fielded count rate image in raw data space  
*x* - a 2-D array of x locations of each output flux position in *image*  
*y* - a 2-D array of y locations of each output flux position in *image*  
*sens* - the sensitivity vector for converting count rates to flux units

For example, to duplicate the interpolation used by CALSTIS use:

```
f = interoplate(image,x,y,missing=0)
f = size(f) & ny = s(2)
for i=0,ny-1 do f(0,i) = f(*,i)/sens
```

Or to perform a cubic interpolation use:

```
f = interplate(image,x,y,missing=0,cubic=-0.5)
f = size(f) & ny = s(2)
for i=0,ny-1 do f(0,i) = f(*,i)/sens
```

## 9.4 Extended Source Extraction Outputs

The calling sequence for an extended source extraction is different than the point source extraction:

```
calstis,id,'extended=1..., h, m, wave, flux, epsf, errf, sens,x,y, unused, image,eps,err
```

The output parameters that were used for *gross*, *background*, and *net* spectra are now used for *sens*, *x*, and *y*.

For first order grating extended source extractions, *flux* will be a 2-D rectified image of size *esnsout* x *esnlout* where *esnsout* and *esnlout* are parameters with a default value of 1064. Wavelength runs from left to right in this image and spatial position (along the slit) runs from top to bottom. For an extended source extraction, *wave* will have *esnsout* values, such that *wave*(*i*) is the wavelength for the image column, *flux*(*i*,\*). *epsf* is a data quality image with the same size as *flux*. The coded values in *epsf* can be found in Appendix A. *errf* is the propagated statistical error image. It is important to note that since the values in *flux* were constructed using bilinear interpolation, the propagated statistical errors in *errf* can be correlated between neighboring pixels (i.e. have a non-zero covariance). *sens* contains a sensitivity vector of length *esnsout*. *sens*(*i*) contains the sensitivity used to calibrate column *flux*(*i*,\*). *x* and *y* are images with the same size as *flux*. *x*(*i*,*j*) and *y*(*i*,*j*) give the location in the raw image (to a fraction of a pixel) of where the value of *flux*(*i*,*j*) was derived. *unused* is an unused output parameter. *image*, *eps*, and *err* have the same results as for camera mode images and point source extractions. They contain the flat fielded count rate image along with its associated data quality and error image.

## 10.0 CALSTIS Output Data Files

In addition to returning the results in the calling sequence, CALSTIS can also write them into FITS data files. Two parameters control the FITS file output: *outimage* and *outspec*. *outimage* tells CALSTIS whether or not to write the flat fielded count rate image to a file. It currently has a default value of **NONE** (i.e. do not write this file). *outspec* controls whether or not CALSTIS will write the output of the point source spectral extraction or the extended source rectified image file. The default for *outspec* is **DEF** (i.e. write the file using the default file name). *outspec* and *outimage* can be set to a **file name**, **NONE**, or **DEF**. If set to a file name (e.g. *calstis*,*id*,*'outimage=myfile.fits'*,...), CALSTIS will write the results to the specified file, if set to **NONE**, CALSTIS will not write the file, and if set to **DEF**, CALSTIS will write the file using the default name. The default filenames are:

<u>Filetype</u>	<u>Filename</u>
Flat fielded count rate image file	im_<id>.fits
Extended source extraction file	espec_<id>.fits
Point source spectral extraction file	spec_<id>.fits

where <id> is the observation identification (first input to CALSTIS).

## 10.1 Output Flat Fielded Count Rate Image File

The output flat fielded count rate image is placed into a FITS file with three extensions. The first contains the flat fielded image, the second contains the data quality image with coded values described in Appendix A, and the third contains the propagated statistical error image.

This file can be read by IDL using:

```
fits_open, '<filename>', fcb
fits_read, fcb, image, h
fits_read, fcb, eps, h
fits_read, fcb, err, h
fits_close, fcb
```

Or alternatively, the extensions can be randomly accessed by:

```
fits_read, fcb, err, h, exten=3
or fits_read, '<filename>', err, h, exten=3
```

When the file name is supplied to *fits\_read*, the routine will open and close the file for you.

## 10.2 Output Extended Source Extraction File

The output rectified image from the extended source extraction has the same format for the first three extensions as the flat fielded count rate image except that the images are the rectified spectral images optionally converted to absolute flux. The file has a fourth extension containing the 1-D wavelength vector and a fifth extension containing the sensitivity vector used to convert from count rates to flux units.

## 10.3 Extracted Point Source Spectral File

The point source extractions are written into a FITS binary table. The table has one row per spectral order and has the following columns:

```
id - the observation ID number or input filename
mode_id - the mode ID (1.1, 1.2, 1.3, ...)
cenwave - the commanded central wavelength
order - the spectral order number for this row
m1 - the first spectral order extracted
```

*m2* - the last spectral order extracted  
*line* - the central line position (cross-dispersion direction) at the center of the spectrum  
*slitnum* - the number of the slit used  
*wavelength* - the wavelength vector  
*flux* - the flux vector  
*epsf* - data quality vector for flux  
*errf* - propagated statistical error vector for flux  
*gross* - the gross spectrum  
*blower* - the lower background spectrum  
*bupper* - the upper background spectrum  
*net* - the net spectrum (gross - smoothed background)  
*position* - the position vector of the center of the extraction slit. *position(i)* gives the position of the extraction slit for *flux(i)*.  
*coef* - the dispersion coefficients used for computing the wavelengths

Some of the columns have been included to allow an extracted point source spectral file to be used as an input dispersion coefficient table (**DCTAB**) or extraction trace table (**EXTTAB**) for subsequent calls to CALSTIS for this observation or for another. For example, to calibrate another observation using the exact same spectral position and dispersion coefficients used to extract observation 9999:

```
calstis,id,'exttab=spec_9999.fits,dctab=spec_9999.fits',....
```

The easiest way to read the FITS binary table into IDL is to use the function **MRDFITS**.

```
a = mrdfits(filename,1,header)
```

This will read the table stored in the supplied filename into an IDL structure, *a*. For an echelle mode, *a* will be a structure array. The optional output parameter, *header*, contains the FITS header. The following examples show how to extract information from the structure. Additional information can be found in the IDL user manuals.

Example 1: Plot the background normalized to the extraction slit of the gross and overplot the smoothed background for a first order grating point source extraction.

```

back = (a.blower + a.bupper)/2*sxpar(header,'gwidth')/sxpar(header,'bwidth')
sback = a.gross - a.net ;smoothed background
plot,a.wavelength,back,psym=1
oplot,a.wavelength,sback

```

Example 2: Put the results of an echelle extraction into 2-D arrays as they would have been returned on the calling sequence to CALSTIS.

```

m = a.order
wave = a.wavelength

```

```
flux = a.flux
epsf = a.epsf
errf = a.errf
```

Example 3: Plot the extracted spectrum for echelle spectral order 470.

```
index = where(a.order eq 470)
plot, a(index).wavelength, a(index).flux
```

## 11.0 Supplying Input Data to CALSTIS

In some cases, you may want to perform some pre-processing to the data before calling CALSTIS. To do this, CALSTIS has four optional keyword input variables, which allow you to specify the data to be calibrated. These variables are *data*, *header*, *errin*, and *epsin*. To use them, you are REQUIRED to specify both *data* and a valid *header*. *errin* and *epsin* are optional.

For example, read an observation, do some pre-processing and then call CALSTIS:

```
stis_read,id,hraw,raw ;read raw data

; Do some pre-processing (for example, fix some cosmic rays)

calstis,id,0,h,m,w,f,epsf,errf,data=raw,header=hraw
```

Even though you are passing the data to CALSTIS, an ID number or input file name must still be supplied. It is used only for identification purposes and for constructing output file names. The input observation is not read by CALSTIS.

Say you want to combine all readouts in a list of multiple CCD observations (stored in vector *list*) with cosmic ray removal. You can use:

```
stis_cr,list,header,data,err,eps
calstis,'aeiou', 0, data=data, header=header, epsin=eps, errin=err
```

Notice that a nonsense observation ID was used. Any ID could have been used to identify the data. Also note that since the cosmic ray rejection algorithm can affect the statistical errors calculation in data points where cosmic rays were rejected, an *errin* is supplied to CALSTIS. *epsin* is supplied because it has information on the number of cosmic ray values that were rejected for each pixel. The previous example could also have been done using:

```
stis_cr,list,outfile='myfile.fits'
calstis,'myfile.fits'
```

## 12.0 STIS Data Catalog, STISLOG

All STIS data are stored in an IDL data base catalog, STISLOG. To search, sort, or display portions of this catalog, a set of IDL routines (beginning with characters DB) can be used. These routines include:

***dbopen*** - open a catalog.  
***dbhelp*** - print help information.  
***dbfind*** - search the catalog.  
***dbsort*** - sort the catalog or selected entries from the catalog.  
***dbprint*** - print the catalog or selected portions of the catalog.  
***dbext*** - extract values from the catalog.

Additional documentation on these routines can be found in their help files or in the GHRS software users guide found at the GHRS IDT website:

[http://hrssun.gsfc.nasa.gov/swg-v2\\_1/online\\_swg/node29.html](http://hrssun.gsfc.nasa.gov/swg-v2_1/online_swg/node29.html)

The following examples will get you started:

```
dbhelp ;print list of available catalogs
dbhelp,1 ;print list of catalogs with short descriptions
dbopen,'stislog' ;open the stis catalog
dbhelp,1 ;print the items stored in the STIS catalog
list = dbfind('proposid=7605') ;find all observations for proposal 7605
sort_list = dbsort(list,'texpstr') ;sort list by exposure start time
;
; print some information about the observations
;
dbprint,sort_list,'entry,filename,targname,opt_elem,cenwave,texptime,sclamp'
;
; print the same results into a file, stislog.prt, for later printing
;
dbprint,sort_list,'entry,targname,opt_elem,cenwave,texptime,nreads,sclamp',textout=3
```

Note that the integer values in ***list*** and ***sort\_list*** are the same as the value of item ***entry***. This is the value that can be used in the call to CALSTIS as the first input parameter instead of supplying a filename. For example, the following two lines will calibrate the same observation.

```
calstis,'o42603060_raw.fits'
and
calstis,2600
```

Some additional search examples are:

1) Find all G140L wavelength calibration observations

```
list = dbfind('sclamp=[hitm,line],opt_elem=G140L')
```

2) Find all G750M observations with a central wavelength between (and including) 7500 and 10000:

```
list = dbfind('opt_elem=G750L,7500<cenwave<10000')
```

3) Average every FUV-MAMA dark:

```
list = dbfind('detector=fuv_mama,targname=dark')
dark = ftarr(1024,1024)
exptime = 0
.run
- for i=0,n_elements(list)-1 do begin
-   stis_read,list(i),h,d
-   dark = dark+d
-   exptime = exptime + sxpar(h,'exptime')
- end
- end
dark = dark/exptime ;convert to count/sec/pixel
tvsc1,alog10(dark>0.0000001)
```

## 13.0 Wavelength Zero Point Calibration

The dispersion relations used by CALSTIS were derived from engineering wavecals taken both before and after launch. These dispersion relations must be adjusted to account for thermal motion and MSM non-repeatability. A typical STIS observing sequence will have wavecals associated with the science observations. These can be used to generate a zero point adjustment to the library dispersion coefficients using the AUTOWAVE option to CALSTIS or using the routine STIS\_WOFFSET.

Use the AUTOWAVE keyword parameter to control zero point wavelength adjustments due to MSM non-repeatability and thermal/temporal image motion. The following five examples of using the AUTOWAVE keyword will give identical results.

*Observation 11555 will be processed using the associated wavecal 11556.*

- 1) *calstis,11555,0,h,m,wave,flux,/autowave*
- 2) *stis\_woffset,11556,results*  
*calstis,11555,0,h,m,wave,flux,autowave=results*
- 3) *calstis,11555,0,h,m,wave,flux,autowave=11556*

4) *calstis,11555,0,h,m,wave,flux,autowave=-0.74*

5) *calstis,11555,'woffset=-0.74',h,m,wave,flux*

In the first example (the default), CALSTIS will automatically find the associated wavecal from the header of the science observation and process it. The correct associated wavecal observation filename is found in the header keyword WAVECAL, populated by in the raw data file by the ST ScI. In this case observation 11556 was the associated WAVECAL.

In the second example, the routine **STIS\_WOFFSET** is used to compute the offset. The results are returned in a structure (*results*) where

*results.mjd* - Modified Julian Date(s) at the midpoint of the wavecal observation(s)

*results.offsets* - zero-point offset for each wavecal

*results.history* - string history array

In the third example, CALSTIS will automatically call **STIS\_WOFFSET** using wavecal observation 11556. If you have a wavecal that you want to use which was not automatically associated with the observation, use this method to supply the id number of the wavecal.

In the fourth and fifth example, the actual zeropoint offset is passed to CALSTIS where the offset was noted from a prior call to CALSTIS. NOTE: if you use the *autowave* keyword parameter, the value of *WOFFSET* parameter is ignored.

If multiple wavecal observations are found in the WAVECAL file, they are all processed. The zero point shift for the science observation is then computed using linear interpolation between the wavecal shifts using the time of the observation midpoint. If the science observation midpoint is outside of the range of midpoints for the wavecal observations, the offset of the closest wavecal is used (i.e. no extrapolation).

Calling **STIS\_WOFFSET** prior to CALSTIS has an advantage over letting CALSTIS make the call. If you need to vary the reduction parameters to CALSTIS, you do not need to waste time reprocessing the wavecal. Also if **STIS\_WOFFSET** fails to compute a correct offset, you do not waste time reducing the science observation with an incorrect offset.

In almost all cases, **STIS\_WOFFSET** will give the correct offset. It is however possible for the cross-correlation of the spectral features in the WAVECAL with the template spectrum to fail and find the maximum correlation at the wrong location. This can happen if the spectrum is offset by more than the maximum search distance or if echelle spectral orders are identified with incorrect spectral order numbers. To assist in identifying the rare cases where the automatic processing fails, **STIS\_WOFFSET** generates a plot of the results and also optionally returns the calibrated WAVECAL and template spectrum to the user.

In the first order grating modes, STIS\_WOFFSET will plot six regions of the spectrum centered on the brightest spectral lines with the template spectrum overplotted. The observed spectrum is plotted with a solid line connecting diamonds. The template (smoothed to approximately the resolution of the observed WAVECAL) is plotted with a dashed line connecting asterisks. If the plots do not match, STIS\_WOFFSET may have failed. To further investigate if the correlation has failed, you can examine the other optional outputs of STIS\_WOFFSET. The calling sequence is:

***stis\_woffset,id,results,h,m,wave,flux,wt,t,image***

where *h* is the image FITS header, *m* is a vector of spectral order numbers, *wave* is the calibrated wavelength array, *flux* is the observed spectrum, *wt*, *t* are the wavelengths and flux values of the template and *image* is the rectified cosmic ray filtered image in the first order grating modes for the last readout processed. If the routine was successful, *wt* and *t* should overplot on *wave* and *flux* without any offset. Note, however, line strengths may vary from the template and the observed spectrum. They may have different lamps and different current settings. Keyword parameter *MAXOFF* can be used to vary the maximum search distance for cross-correlating the spectra.

In echelle observations, an offset is computed for each spectral order. The returned offset is the median of these offsets. The offsets for each order are printed and also plotted. If most of the offsets are within a pixel of the median offset, the routine was successful. If not, you might try increasing the keyword parameter, *MAXOFF*, or verifying that the orders were correctly numbered with the proper spectral order numbers. Spectral format plots are available on the team web page:

**<http://hires.gsfc.nasa.gov/stis/stispage.html>**

under "Description" to assist you in determining the order numbers of a spectral calibration lamp observation. If the spectral orders are misidentified, STIS\_WOFFSET keyword parameters *offset* and *dist1* can be used to correct the situation.

## 14. Introduction to Time-Tag Data Processing

The following routines are available for analysis of STIS time-tag data.

**TTAG\_INFO** - routine to extract information about a time-tag data set  
**TIME\_TAG** - routine to read time-tag data sets  
**TTAG\_IMAGE** - routine to construct an image from the time-tag data  
**TTAG\_BIN** - routine to generate the count rate versus time for time interval  
**TTAG\_CENTROID** - routine to compute centroid of a feature versus time  
**TTAG\_MOVIE** - routine to construct a data cube (movie) for a selected image region of a time-tag dataset

To get a summary of a time-tag dataset use:

***TTAG\_INFO,id,out***

Where *id* is the observation ID (entry number in STISLOG) or the file name of the time\_tag data set (e.g. 'o3xl08bnq\_tag.fits'). The routine will print a summary and also place the information in the structure ***OUT***. Refer to the helpfile for TTAG\_INFO for the format of ***OUT***.

***WARNING*** - Do not try to read a very large time-tag data set entirely into memory at once. It may cause your computer to choke.  
If ***TTAG\_INFO*** reports that the data set has more than 10 to 20 million events, refer to "Working with very large time-tag data sets" found in section 14.1.

To read the time-tag data, use the routine:

***TIME\_TAG,id,h,time,x,y,xdop***

where *id* is the observation ID (entry number in STISLOG) or the filename (post-launch time-tag data files have names <rootname>\_tag.fits). *h* is the output FITS header, *time* is a vector of times for each event (in units of seconds since the exposure start), *x* is the vector of x-locations of the events (in Hi-Res coordinates from 0 to 2047), *y* is the y-locations of the events, and *xdop* is the Doppler corrected x positions for echelle spectral observations.

To construct an image from the time-tag data, use:

***TTAG\_IMAGE,h,x,y,image***

where *h,x* and *y* are input header, x and y locations (as output from TIME\_TAG). *image* is the output Hi-Res image. You can tell TTAG\_IMAGE to use only events occurring within a specified time interval using optional keyword parameters (*time*, *tmin*, and *tmax*). For example to generate an image from the events from 5 to 6 seconds after the exposure start, use:

***TTAG\_IMAGE,h,xdop,y,time=time,tmin=5.0,tmax=6.0***

where *time* is the output of TIME\_TAG.

To generate a plot of count rates versus time, use the routine:

***TTAG\_BIN,time,tout,rate***

where *time* is the time for each event (from TIME\_TAG) and *tout* and *rate* are vectors giving count rate (*rate*) versus time (*tout*). The default is to use events anywhere within the image and to integrate rate on 0.01 second intervals. Keyword parameter *interval* can be used to set a different time interval. Keyword *range*=[*xmin,xmax,ymin,ymax*] can be used to examine events in a subset of the image, and keyword */counts* can be used to get output counts/time interval instead of count rates. Keywords *tmin* and *tmax* can be used to limit the time period processed, For example:

```
TTAG_BIN,time, t, c, range=[1500,1510,800,810], interval=0.001, $
tmin = 50.0, tmax = 52.0, /counts
```

To find the centroid of a feature versus time, use TTAG\_CENTROID.

```
TTAG_CENTROID, time, x, y, tout, xcent, ycent, xrange=[xmin,xmax], $
yrange = [ymin, ymax]
```

where *time*, *x*, and *y* are the input times and positions of the events, and *tout*, *xcent*, and *ycent* are the output times, x-centroid, and y-centroid on one second intervals. *xrange* and *yrange* are keyword inputs giving the range of x/y coordinates to centroid. To change the time interval of the centroids use keyword *interval*. To only centroid a portion of the data (in time) use keyword parameters *tmin* and *tmax*.

Much of the archived data is compressed. It will significantly slow you down if every time you read an archived time-tag file, it gets uncompressed. To avoid this problem, use the */KEEP* keyword parameter in your call to TTAG\_INFO or TIME\_TAG. If the file was compressed, an uncompressed version will be left in your current directory. This version will be used for subsequent calls to TIME\_TAG or TTAG\_INFO.

**EXAMPLE SESSION 1:** Make a movie of a Lyman Alpha for a G140L observation of target JUP-AUR-OVALN and perform an extended source flux calibration for a selected time period of the image.

```
;
; get information about the observation and keep the uncompressed version
; if the file was compressed
;
id = 16164
ttag_info,id,out,/keep
;
; read the events
;
time_tag, id, h, time, x, y
;
```

```

; construct a Hi-Res image from the events
;
ttag_image, h, x, y, image
;
; use STISLOOK to find a region of interest
;
stislook, image
region = [92,491,1020,1419] ; [xmin,xmax,ymin,ymax]
;
; plot the total count rate versus time for the region averaged in 10
; second intervals
;
ttag_bin, time, tout, rate, x=x, y=y, range=region, interval=10
plot, tout, rate
;
; create a data cube (movie) of the region by integrating in 20 second
; intervals with data binned by 4 (2 x 2 lores pixels)
;
ttag_movie, time, x, y, cube, interval=20.0, range=region, binsize=4
;
; load cube into IDL movie widget which will also let you save it as an
; MPEG file
;
s = size(cube) & nx = s(1) & ny = s(2) & nframes = s(3)
xinteranimate, set=[nx,ny,nframes], /showload
for i=0,nframes-1 do xinteranimate, frame=i, $
image = bytscl(cube(*,*,i),min=0,max=5,top=!d.n_colors)
xinteranimate,/keep_pixmaps
;
; create an image and calibrate it for the period of 100 to 300 seconds since
; the observation start.
;
ttag_image, h, x, y, image, time=time, tmin=100, tmax = 300
sxaddpar,h,'exptime',200
sxaddpar,h,'integ',200
calstis,id,'extended=1',h,m,wave,flux,eps,err, header=h, data=image

```

## EXAMPLE SESSION 2: Examine Band 2 MAMA dark microbursts.

```

;
; read time tag data set

```

```

;
    time_tag, 2657, h, time, x, y
;
; create an image from the time tag data and display
;
    ttag_image, h, x, y, image
    tvsc1, rebin(float(image),512,512) < 1
;
; determine count rates in 0.01 second time bins and plot rate versus time
;
    ttag_bin, time, tout, rate, interval=0.01
    plot, tout, rate
;
; find the time bin with the peak rate and generate an expanded plot of
; counts versus fine time interval (125 microseconds) for a 0.04 second
; interval containing the peak
;
    print, max(rate,position)
    t = tout(position)           ;time of maximum count rate
    ttag_bin, time, t1, c1, tmin = t-0.02, tmax = t+0.02, $
            /counts, interval=0.000125
    plot, t1, c1, psym=-4
;
; It looks like a micro-burst; plot a scatter plot of the x and y coordinates
; of the events for the time period of the microburst.
; (i.e. where we got 4 counts/fine time interval)
;
    burst = where(c1 ge 4)
    start_burst = min(t1(burst)) - 0.000125
    stop_burst = max(t1(burst)) + 0.000125
    burst = where((time ge start_burst) and (time le stop_burst))
    plot, x(burst), y(burst), psym=4

```

## 14.1 Working with Very Large Time-Tag Data Sets.

Most of the time-tag data sets in the archive have only a few million or fewer number of events. Some, however, will have many millions of events. In these cases, unless your computer has a very large amount of memory, it will be very inefficient to read the entire time-tag data set into IDL. Use TTAG\_INFO to determine if this is the case.

To process the very large data sets, you will need to read the image in parts. The routine TIME\_TAG has two methods to allow you to read parts of the data set. The time-tag data in the raw data file is partitioned into separate groups or file extensions where each extension contains an on-board memory dump. Each group is a good size to read without getting into memory problems.

To read a single group use the **GROUP** keyword parameter when calling TIME\_TAG (**GROUP** runs from 1 to the number of groups). TTAG\_INFO will print the number of groups in the file and also return it in an output data structure. For example, to create an image from a large E140M time-tag data set, use:

```

ttag_info,2222,out,/keep
image = lonarr(2048,2048)
n = 0L
.run
for i=1,out.ngroups do begin                               ;START LOOP AT 1 not 0!
    time_tag,2222,h,time,x,y,xdop,group=i
    ttag_image,h,xdop,y,im
    n = n+n_elements(time)
    print,min(xdop),min(y),max(xdop),max(y)
    image = image + im
endfor
end

```

Note that in the example, the Doppler corrected x position was used for the echelle time-tag observation.

To create a movie of a selected region of a large time tag observation use:

```

ttag_info,id,out,h,/keep
cube = 0
.run
for i=1,out.ngroups do begin
    time_tag,id,h,time,x,y,group=i
    ttag_movie,time,x,y,cube,tmin=0,tmax=tmax,interval=100, $
        range=[300,400,500,600],binsize=2,/update
endfor
end

```

To create a vector of count rate versus time, use:

```

ttag_info,id,out,h,/keep
tmax = smpar(h,'exptime')
.run
for i=1,out.ngroups do begin
    time_tag,id,h,time,x,y,group=i
    ttag_bin,time,tout,r,tmin=0,tmax=tmax,interval=1
    if i eq 1 then rate=r else rate=rate+r
endfor
end

```

An alternate way of selecting which events to use is to specify the number of the first event and the number of events to read. For example:

***time\_tag,2222,h,time,x,y,x dop,first=4000000,number=2000000***

## Appendix A

### Data Quality Flag Values

A value of 0 indicates no anomalous condition. Flags are coded using the same basic principal used by the FOS and GHRS ST ScI calibration pipelines. Values from 1 to 99 are minor conditions which are reflected in the propagated statistical errors. Values from 100 to 199 are more serious conditions which are not properly reflected in the propagated statistical error array. Values from 200 to 255 represent data that is totally useless. Each data value is flagged with its most serious condition (largest data quality value).

Flag	Condition
255	Fill data (telemetry dropout)
254	Data point outside boundary of the image
252	Unrepairable hot pixel
251	For absolute flux vector (data outside tabulated sensitivity curve)
239	All pixels rejected during cosmic ray removal/ image combination
240	Dead detector element of bad CCD column
190	Saturated pixel
180	Hot pixel ( dark rate > HOTTHRESH counts/second)
175	Severe flat field blemish
174	Data corrupted by the band 1 repeller wire
155	Data near edge of detector
154	NUV MAMA Vignetted pixels
150	Warm pixel (CCD dark rate > 0.05 counts/second)
125	Hot pixel repaired by interpolation
65	5 or more pixels rejected during CR-SPLIT image combination
64	4 pixels rejected during CR-SPLIT image combination
63	3 pixels rejected during CR-SPLIT image combination
62	2 pixels rejected during CR-SPLIT image combination
61	1 pixel rejected during CR-SPLIT image combination
0	No anomalous condition noted

## Appendix B Sample Sessions

### B.1 Finding and Reading STIS Data

```

;
; List available catalogs
;
dbhelp

3cr                abell                ans                bruzual
cycle7_abs         cycle7_exp         donas              faust

                    . . .

stisgto           stislog           tdl                tdl_spec
td1_x             uit_bright        uvgalaxy          virgo
wdwarf            wdwarf_iue        yale_bs            yale_bs_rmks

;
; open stis catalog and list items stored for each observation
;

dbopen,'stislog'
dbhelp,1

----- STISLOG  STIS observation log
      ITEM                TYPE                DESCRIPTION
ENTRY                int*4                Entry or Record Number
ROOTNAME             char*10              rootname of the observation set
FILENAME             char*20
FILETYPE             char*3                file type RAW, TAB, or WAV
NREADS               int*4                number of image readouts in the file
TARGNAME             char*18              proposer's target name
RA_TARG              real*4                right ascension of the target (deg) (J2000)
DEC_TARG              real*4                declination of the target (deg) (J2000)
PROPOSID             int*4                PEP proposal identifier

                    . . .

;
; Find observations from proposal 7123
;

list = dbfind('proposid=7123')
      12 entries found in STISLOG

```

```

;
; print selected information about the observations
;
dbprint, list, $
'entry,filename,nreads,targname,mode_id,cenwave,aperture,sclamp'

```

## STIS observation log

```

-----
ENTRY          FILENAME          n          TARGNAME          mode   cen   wave   APERTURE   sc
read          read          _id          wave          lamp
-----
933 o3xl07a3m_raw.fits          3 LMC-SN1987A-OFF 3.6          0 F28X50LP   NONE
934 o3xl07010_raw.fits          2 LMC-SN1987A    4.2          6581 52X2    NONE
935 o3xl07010_wav.fits          1 WAVELINE       4.2          6581 52X0.1  LINE
936 o3xl07a8m_raw.fits          1 WAVEHITM       4.2          6581 52X0.1  HITM1
937 o3xl07020_raw.fits          2 LMC-SN1987A    3.2          4961 52X2    NONE
938 o3xl07020_wav.fits          1 WAVELINE       3.2          4961 52X0.1  LINE
939 o3xl07acm_raw.fits          1 WAVEHITM       3.2          4961 52X0.1  HITM1

```

...

```

;
; read one of them
;
stis_read, 934, h, d
; or
stis_read, 'o3xl07010_raw.fits', h, d ; does the same thing

hprint,h ; display header

window, xs=1024, ys=1024
tvsci, alog10(d>0.1) ; display image

stis_read, 934, h, d, readout=2 ; get second readout

stis_cr, 934, h, d ; cosmic ray rejection using
; both readouts

stislook,d ; interactively examine it

```

## B.2 First Order Grating Point Source Spectral Extraction

```

;
; calibrate observation 730 with cosmic ray rejection for the cr-split science observation
; using the associated wavecal, id=731, to adjust the wavelength zero point.
;
calstis, 730, 0, h, m, wave, flux, eps, err,/cr_reject,autowave=731

plot, wave, flux
;
; read calstis extracted spectrum file
;
a = mrdfits('spec_730.fits',1,h)
;
; scale upper and lower background to gross extraction slit
;
bupper = a.bupper / sxpar(h,'bwidth')*sxpar(h,'gwidth')
blower = a.blower / sxpar(h,'bwidth')*sxpar(h,'gwidth')
plot, a.wavelength, bupper, psym=2
oplot, a.wavelength, blower, psym=4
;
; overplot the smoothed background used by CALSTIS
;
oplot,a.wavelength, a.gross - a.net, thick=3

```

### B.3 G750M Extended Source Extraction of a Long Slit Observation.

```

;
; Perform extended source extraction on G750M observation 934 using wavecal 935 to adjust
; the wavelength zero point. Use STIS_CR to combine the cosmic ray splits with the results
; written into file myimage.fits.
;
stis_cr, 934, outfile='myimage.fits'
stis_woffset,935,results
calstis, 'myimage.fits', 'extended=1', h, m, wave, flux, $
autowave=results
;
; extract a single spectral line into a 40 x 100 image. Adjust the x-direction plate scale in the
; output image to match the y-direction plate scale. The plate scales in the raw images are
; x = 0.54 arcsec/pixel and y = 0.5 arcsec/pixel. Use an x sample spacing of 0.926 (ratio of y
; plate scale over the x plate scale) to match the y-sample spacing.
;
par = 'extended=1,esnsout=40,esnlout=100,' + $
'escentw=6594,esdeltas=0.926'
calstis, 'myimage.fits', par, h, m, w1, f1,autowave=results

```

## B.4 Long Slit Extended Source Extraction (M84) with a Constant Delta Radial Velocity

*; Data Preparation, Cosmic Ray and Hot Pixel removal*

```
stis_cr, 863, h, data,err,eps
imgclean, data,h
```

*; Extended source extraction on constant delta Log(Wavelength) Scale so that radial velocity shifts are equal at all wavelengths. we will provide the input data to CALSTIS as IDL variables instead of using the data in the raw data file*

```
calstis, 863, 'extended=1,eslogl=1', h, m, wave, flux, $
data=data, header=h, epsin=eps, errin=err
```

*; Create a template spectrum for cross correlation using the rest wavelengths of the spectral lines present in the observation. The line at rest wavelength 6583.4 is approximately twice as bright as the others.*

```
sig = 7.0/2.3548 ; seven pixel FWHM
gauss, wave, 6548.1, sig, 1.0, l1
gauss, wave, 6562.8, sig, 1.0, l2
gauss, wave, 6583.4, sig, 1.0, l3
gauss, wave, 6716.4, sig, 1.0, l4
gauss, wave, 6730.8, sig, 1.0, l5
template = l1 + l2 + 2*l3 + l4 + l5
```

*; cross correlate each spatial position with template*

```
offsets = fltarr(1000)
.run
- for i=450,600 do begin
-   cross_correlate, template, flux(*,i), offset, ishift=-40, $
-                                     width=100
-   offsets(i) = offset
- end
- end
```

*; convert to slit position and radial velocities and plot*

```
slitpos = (findgen(1000)-534.0) * 0.05  
rv = -offsets * (wave(500)-wave(499))/wave(500) * 2.9979e5  
plot, slitpos, rv, psym=-4, xrange=[-2.0,2.0], yrange=[700,1600], $  
min_value=700, max_value=1600, $  
  ytitle = 'Radial Velocity (km/sec)', $  
  xtitle = 'Position Along Slit (arcsec)'
```

## B.5 Calibrate an Echelle Spectrum Including a Scattered Light Correction

```
;  
; Calibrate an E140M observation of CPD-59D2603 in MAMA Hi-Res (2048x2048) mode  
; using the associated wavecal for zero point wavelength adjustment.  
;  
    calstis,12729,'outimage=def,hires=1',/autowave  
;  
; Perform echelle scattered light correction  
;  
    echelle_sc,12729  
;  
; read spectral output for both before and after scattered light correction  
;  
    before = mrdfits('spec_12729.fits', 1)  
    after = mrdfits('newspec_12729.fits', 1)  
;  
; compare one of the orders before and after the scattered light correction  
;  
    !grid = 1  
    plot, before(35).wavelength, before(35).flux<1e-12, nsum=11  
    oplot, after(35).wavelength, after(35).flux, nsum=11, thick=2
```

## B.6 Calibrate and Merge a Two Readout Echelle Observation

```

;
; Calibrate and scatter light correct the first readout. Rename the output spectral file so that it
; will not be over-written when the second readout is calibrated
;
    calstis,93224, 'outimage=def,hotthresh=0.001', readout=1
    echelle_scst, 93224
    spawn, 'cp newspec_93224.fits newspec_93224_1.fits'
;
; Calibrate the second readout
;
    calstis,93224, 'outimage=def,hotthresh=0.001', readout=2
    echelle_scst, 93224
    spawn, 'cp newspec_93224.fits newspec_93224_2.fits'
;
; read both spectral files and place into 2-D arrays
;
    read1 = mrdfits('newspec_93224_1.fits',1,h1)
    read2 = mrdfits('newspec_93224_2.fits',1,h2)
;
    wave = [[read1.wavelength],[read2.wavelength]]
    flux = [[read1.flux],[read2.flux]]
    err = [[read1.errf],[read2.errf]]
    eps = [[read1.epsf],[read2.epsf]]
;
; merge both readouts and all of the spectral orders
; (Use the errors for weighting. Smooth error vectors before computing weights)
;
    smooth_err = err
    for i=0,n_elements(err(0,*))-1 do smooth_err(0,i) = $
    smooth(err(*,i),11)
;
; Merge using data quality = 150 as screening limit
;
    hrs_merge, wave, flux, eps, err, wout, fout, epsout, errout, $
    badeps=150, weight = 1/(smooth_err^2)

```

## Appendix C

### Recent Changes to CALSTIS

Version	Date	Modifications
6.42	May 1999	Added <i>esps</i> parameter to allow point source slit throughput correction during extended source calibrations.
6.43	May 1999	Added MAMA hot pixel flagging
6.5	Sep 1999	Added monthly MSM offset correction to the echelle dispersion coefficients. (new reference file <i>msmtab</i> )
6.6	Jan 2000	Added the R. Bohlin time/temperature correction to the sensitivities
6.7	Oct 2000	Added prism point source spectral extraction support. Added <i>targoffs</i> parameter to allow correction of wavelengths for targets not centered in the slit.
6.8	Jan 2002	Added echelle blaze shift as a function of time and monthly MSM offset position and <i>auto_bs</i> option. Modified to make auto wavecal processing the default
	Feb 2002	Added scaling of CCD darks using the CCD housing temperature for side two operations.
6.9	Jun 2002	Added sensitivity vector to output extended source extraction files. Added <i>bupper</i> , <i>blower</i> , <i>bdistu</i> , and <i>bdistl</i> background parameters. Added MAMA global non-linearity correction.
7	Aug 2002	Added new background processing where airglow lines are left unsmoothed during the polynomial fit to the background.
7.1	May 2003	Added CCD CTE correction for point source extractions.
7.2	Aug 2003	Improved CTE correction coefficients and sensitivity time/temperature correction.