



STScI | SPACE TELESCOPE
SCIENCE INSTITUTE

Instrument Science Report WFC3 2017-01

A more generalized coordinate transformation approach for grisms

Nor Pirzkal, R. Ryan

January 5, 2017

ABSTRACT

Current HST configuration files for the NICMOS, ACS and WFC3 grisms have been based on the aXe configuration file methodology. These files describe how spectra are physically dispersed for individual orders when using a particular detector and instrument. With the addition of direct filter wedge offsets in the grism calibration of the WFC3 G102 and G141 grisms and with new instruments on the horizon, such as JWST NIRCAM, the current aXe method to describe the grism dispersion needs to be generalized. Here we introduce an updated version of the grism dispersion parametrization.

1. Introduction

The goal of the grism configuration file is to allow a user to go back and forth between what are essentially two coordinate systems: an image+wavelength coordinate system and a dispersed spectrum coordinate system. Fundamentally, this is analogous to the WCS transformation functions, `pixtosky` and `skytopix`, which allow a user to go back and forth from units on the sky to detector pixels, allowing for corrections such as geometric distortions

and other effects. Users are isolated from the details of transformations from and to and are instead provided with a set of functions or modules to perform the required transformation. Similarly to the WCS coordinate transformation problem, when dealing with dispersed spectra, we are interested in knowing where light that would otherwise be on a pixel at coordinates x,y on the detector will be dispersed to in the dispersed frame. Since the dispersion is wavelength dependent, the wavelength of the light is one of the input parameter to compute this. Users can be fully isolated from how this operation is performed as long as an appropriate function f or module to implement the transformation

$$x', y' = f(x, y, \lambda) \tag{1}$$

is provided. In the Equation above, (x,y) refers to the pixel coordinate in the incident image and (x',y') to the pixel coordinates on the dispersed image, while λ is the wavelength of the incident light.

2. Current aXe format

The aXe configuration file format to describe the dispersion of spectra was developed back in 2000 to be used with the ACS grisms and prisms. All of these instruments dispersed the light in the x-direction (in the case of the ACS/HRC grism, with a large angle). The actual parametrization of this dispersion was done using two-dimensional polynomials which allowed for field dependence of the function $f(x, y, \lambda)$. A set of two dimensional polynomials were introduced to do the job of the function f described above. These polynomials, given a pixel at coordinates (x,y) on the detector allowed the user to compute the wavelength of a region of the dispersed spectrum that was a given distance along the trace. At the time, the path distance along the trace was chosen as one of the driving variable of these polynomials. This allowed for some flexibility when spectra were strongly non-linear (as in the case of the WFC3 UVIS grism for example).

The polynomial transformations are described in detail in WFC3-ISR-2016-15 from Pirzkal et al. The whole physical dispersion of a grism are described using only polynomials: $\Delta y(x, y, \Delta x)$ and $\lambda(x, y, \Delta s)$. The first one, $\Delta y(x, y, \Delta x)$, allows to compute the offset in the y-direction (i.e. Δy) of the trace as a function of an offset in the x-direction (Δx). The second polynomial, $\lambda(x, y, \Delta s)$, allows a user to compute the wavelength of the light incident on a pixel (λ), at a distance of Δs along the trace. Having defined where the trace is located using the $\Delta y(x, y, \Delta x)$ polynomial, one can compute the wavelength after computing

the pathlength (Δs) along the trace, which is computed using:

$$\Delta s(x, y, \Delta x) = f(\Delta x) = \int_0^{\Delta x} \sqrt{1 + \left(\frac{dy}{dx}\right)^2} dx \quad (2)$$

where $\frac{dy}{dx}$ is the first derivative, or slope, of the trace, in pixel coordinates, in the dispersed image.

The closed form of these equations is shown in Equations 3 and 8 in WFC3-ISR-2016-15.

The aXe software takes care of computing the path length for users.

The specific manner in which the dispersion is described in the aXe configuration files stems from the fact that these were designed for *extracting* spectra.

Unfortunately, there are several serious drawbacks and limitations to the way dispersers are parametrized:

- The description of the trace, $\Delta y(x, y, \Delta x)$, implicitly assumes that the trace is non degenerate in the x-direction and that Δy can be written as a functional form of Δx . A spectrum dispersed vertically on the detector can therefore not be described this way. This parametrization is therefore not general enough to describe both horizontally and vertically dispersed grisms.
- Parametrizing the wavelength calibration as a function of path length along the trace requires code to perform the integration shown in Equation 2 . This can be a computationally expensive integral.
- The effect of specific filter wedge (the variation of the filter width in the optical path that results in small x- and y- offsets in the image plane), as the ones that exist for the WFC3 IR filters, are not easily handled in the aXe configuration methodology and different 2D polynomials need to be re-derived for each possible direct filter offset.
- For the purpose of generating simulated spectra, it is actually the reverse relation $\lambda(x, y, \Delta s)$ that is required (i.e. $\Delta s(x, y, \lambda)$). That is, a way to compute how far light at a given wavelength will be dispersed. The integral to compute the pathlength Δs , shown in Equation 2 can be a very computationally expensive function to invert.

3. Solution

Modern instruments such as JWST/NIRCAM as well as modern WFC3 simulation code will benefit from a generalization of the way the grism dispersions are described. Here, we describe how this can be done by expanding the current aXe configuration file in two crucial ways: First, the description of the grism dispersions should not be assumed to be a function of the x-coordinate but rather more generalized one (which we will call t) which can be predefined to span the range $(0, 1)$. How the coordinate t is related, or not, to physical quantities such as x , y and λ can be determined on an instrument/disperser basis and is something that remains transparent to the user. In the original aXe configuration format, t was chosen to be Δs the pathlength along the trace, with the caveats we described above.

The physical dispersion and wavelength calibration of the trace of a single spectral order can be fully described using three functions:

$$\hat{x} = x' - x = f_x(x, y; t) \tag{3a}$$

$$\hat{y} = y' - y = f_y(x, y; t) \tag{3b}$$

$$\lambda = f_\lambda(x, y; t) \tag{3c}$$

Using these function, a user or extraction software can trivially computed the location and wavelengths along the trace for a source that is located at detector coordinates of (x, y) . However, and in addition to the Equations 3, we also require the reverse functions to be also defined:

$$t = f_x^{-1}(x, y; \hat{x}) \tag{4a}$$

$$t = f_y^{-1}(x, y; \hat{y}) \tag{4b}$$

$$t = f_\lambda^{-1}(x, y; \lambda) \tag{4c}$$

Note that in addition to the reverse functions, the *numerical inverse* functions could be provided. This would allow for code that relies on going back and forth between the imaging and the dispersed reference frames repetitively without loss of precision. In cases of simple polynomials, the *inverse* and reverse functions would be identical. Specific applications will drive the use of either the reverse or *inverse* functions.

While the use of a arbitrary parameter t might seem unnecessary, it avoids assuming that the spectra can be described well using λ alone, as is the case with some prisms for example. In simple cases, t could simply be equal to the wavelength. Alternatively, it could

be attractive to defined t to be a normalized wavelength range with values ranging from 0 to 1. In the latter case, the wavelength of a spectral order can trivially be computed using $\lambda_{min} = f_\lambda(x, y; 0)$ and $\lambda_{max} = f_\lambda(x, y; 1)$. In the end, t is chosen as a matter of convenience and convention. This set of 6 functions allow users and code to relate any of the x, y, λ, \hat{x} , and \hat{y} variable to one another in an efficient manner. In the description above, we made no assumption as to how the functions f_x, f_y , and f_λ are implemented.

4. Implication for calibration

The functions f_x, f_y , and f_λ (Eq. 3) can be determined in a manner that is similar to past ACS and WFC3 calibration efforts. As long as calibration or observational data exist to measure both the physical location of the trace and the position of known emission lines, one can uniquely determine these relations. The aXe configuration files have relied on a two-dimensional description of the field dependence of these functions but that is not technically required. Other approaches, such as interpolated lookup tables are also possible (although solutions near the edges of detector might be better extrapolated using a smooth two-dimensional fit). The details of how Equations 3 are implemented will not have an impact on users who will simply be provided with code to compute the Equations 3.

The reverse functions functions f_x^{-1}, f_y^{-1} , and f_λ^{-1} (Eq. 4) can be trivially derived from Equations 3 if the latter are expressed as simple polynomials of order less than 3. However, a better approach is to actually calibrate and implement Equations 4 independently of Equations 3 using the original data. In another word, the equivalent of fitting y versus x and then doing a separate fit of x versus y rather than trying to invert the fit of y versus x .

This ensures that any errors in the calibration of Equations 3 are not further propagated into Equations 4.

5. Example Workflow and use

Basic examples of how to perform the transformation from and to the (x, y) to the (x', y') coordinates are illustrated in Figure 1. A basic implementation of this approach had been coded in Python and is available at <https://github.com/npirzkal/GRISMCONF>. The functions f_x, f_y , and f_λ are named DISPX, DISPY, and DISPL. The inverse functions f_x^{-1}, f_y^{-1} , and f_λ^{-1} are named INVDISPX, INVDISPY, and INVDISPL. Currently, the GRISMCONF module implements these functions as well as a few helper functions and disperser specific parameters. The reader should refer to the GRISMCONF documentation for more informa-

tion. Two dimensional field-dependent polynomial up to orders (2,3), and their derivative are currently implemented, i.e.:

$$P_{2,3}(x, y, t) = a_{0,0} + x \times a_{0,1} + y \times a_{0,2} + x^2 \times a_{0,3} + x \times y \times a_{0,4} + y^2 \times a_{0,5} + x^3 \times a_{0,6} + x^2 \times y \times a_{0,7} + x \times y^2 \times a_{0,8} + y^3 \times a_{0,9} + t \times (a_{1,0} + x \times a_{1,1} + y \times a_{1,2} + x^2 \times a_{1,3} + x \times y \times a_{1,4} + y^2 \times a_{1,5} + x^3 \times a_{1,6} + x^2 \times y \times a_{1,7} + x \times y^2 \times a_{1,8} + y^3 \times a_{1,9}) + t^2 \times (a_{2,0} + x \times a_{2,1} + y \times a_{2,2} + x^2 \times a_{2,3} + x \times y \times a_{2,4} + y^2 \times a_{2,5} + x^3 \times a_{2,6} + x^2 \times y \times a_{2,7} + x \times y^2 \times a_{2,8} + y^3 \times a_{2,9})$$

Analytical inverse of the 2D polynomials up to order (1,3) are included, providing an efficient implementation of field dependent linear dispersion.

5.1. Extracting

The extraction process is illustrated on the right panel of Figure 1. The wavelength of a pixel in the dispersed image can be derived by first computing δx or δy , given knowledge of where the source is located in the field (x,y). t can be computed using either f_x^{-1}, f_y^{-1} and λ is computed using f_λ .

Using the GRISMCONF module, can be done using:

```
import grismconf

C = grismconf.Config("G102.conf")

dx = xp - x0
t = C.INVDISPX("A", x0, y0, dx)
wavelength = C.DISPL("A", x0, y0, t)
```

where xp are the x-coordinates of pixels on the dispersed image, x0,y0 are the pixel coordinates of the source in the direct image.

5.2. Dispersing

The reverse process, one where we want to generate a spectrum for a source (instead of extracting it) is illustrated in the Left panel of Figure 1. In this case, we want to determine where light emitted from a pixel (x,y) in the direct image, and at the wavelength λ , would end up in the disperse image. We first compute the value of t for the wavelength λ using f_λ^{-1} . We then use δy and δx using the function f_x, f_y to determine the pixel location (x',y')

where the light will be dispersed.

Using the GRISMCONF module, can be done using:

```
import grismconf

C = grismconf.Config("G102.conf")
t = C.DISPL("A", x0, y0, wavelength)
dx = C.DISPX("A", x0, y0, t)
dy = C.DISPY("A", x0, y0, t)

xp = x0 + dx
yp = y0 + dy
```

where x_0, y_0 are the coordinates of the source in the direct image, $wavelength$ is the desired wavelength value to compute the dispersion for, and x_p, y_p are the coordinate of the appropriate pixel in the dispersed image.

In both of the above examples, $wavelength$, x_p , y_p can be single values or vectors, allowing for the computation of a full spectra trace using a single function call.

6. Scope

The parametrization we described concentrated on a single spectrum. That is, a specific spectral order obtained using a specific dispersive element on a specific detector of a specific instrument and, finally, telescope. These functions are essentially the basic quanta to describe the dispersed trace and its wavelength calibration. The actual configuration file or system will need to allow for multiple orders (beams in the aXe world), detectors (e.g. ACS, NIRCAM), and so forth. Other important values such as the derivative of the trace dispersion or wavelength dispersion can be provided as part of the overall module. Depending on the specifics of the calibration these can be pre-calibrated or computed on the fly by the calibration module. We also did not include a discussion of other physical calibrations such as sensitivity or P and L flat-fielding, which should be allowed. Some of the latter might be beam specific, some might be detector specific and so a clear hierarchical definition will be necessary to keep things organized.

7. Acknowledgments

We would like to thank the SSWG (Slitless Spectroscopy Working Group) and Stefano Casertano specifically for inputs to this document and Knox S. Long for his review of this document.

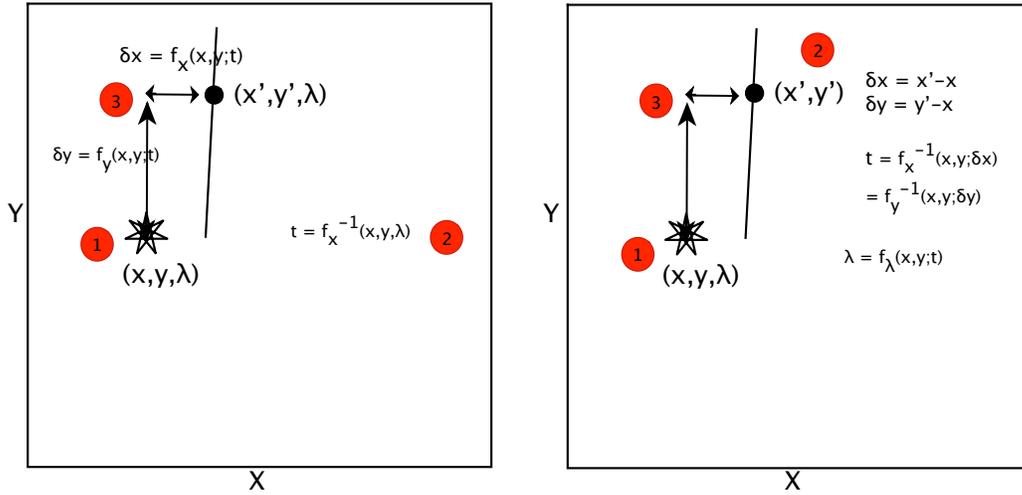


Fig. 1.—: Illustration of implementing a more generalized transformation from (x, y, λ) space (direct image) to dispersed (x', y') space (grism image). As long as functions to describe the dispersion do not assume either x- or y- direction dispersions, the processes of simulating or extracting spectra can be the same no matter what direction spectra are actually dispersed in.