Astronomical Data Analysis Software and Systems X
ASP Conference Series, Vol. 238, 2001
F. R. Harnden Jr., F. A. Primini, and H. E. Payne, eds.

SLIM: A Program to Simulate the ACS Spectroscopic Modes

N. Pirzkal, A. Pasquali, R. N. Hook, J. R. Walsh, R. A. E. Fosbury, W. Freudling, R. Albrecht

ST-ECF, Karl-Schwarzschild Str-2, Garching bei Muenchen D-85748, Germany

Abstract. SLIM is a program developed at ST-ECF to simulate direct and slitless spectroscopic images. It is written in Python to keep development time short and features all of the spectral elements available in the Advanced Camera for Surveys (ACS) which is to be installed on-board the *Hubble Space Telescope*. SLIM produces slitless spectroscopy simulations which include the effects of geometric distortions, fringing, and the field-dependence of the spectral dispersion. We illustrate the use of SLIM with simulations of the HDF-N with the G800L grism using both the Wide Field Channel and with the High Resolution Channel.

1. A Slitless Spectroscopy Simulator

A slitless spectroscopy simulator was required to create realistic data of the upcoming Advanced Camera for Surveys (ACS) slitless spectroscopic modes. ACS is scheduled to be launched in 2001, and simulated data are needed to test future slitless spectroscopy extraction software. Our basic requirement was to have a simulator that could, based on our current knowledge of the instrument, generate both geometrically and photometrically realistic images. While this simulator would in principle be used mainly to generate ACS images, an effort was made to keep it general enough to be used to simulate other instruments. This ensured that the simulator would remain usable to generate ACS simulations even if the performances and/or the characteristics of the instrument were to change in the future.

2. Why Python?

Creating such a simulator can be facilitated greatly if one chooses a language which can easily handle complex data types such as arrays, and interpolated functions. In addition to these, Python has the advantage of being a free, well documented, and well supported language, which can be used on almost all known combinations of hardware and operating systems. While a scripting language, Python has also been shown to be very flexible and appropriate for both small and large projects. One thing that made Python attractive for this project was the existence of many Python extension packages which provide useful high-level data types. For example, we have made extensive use of the

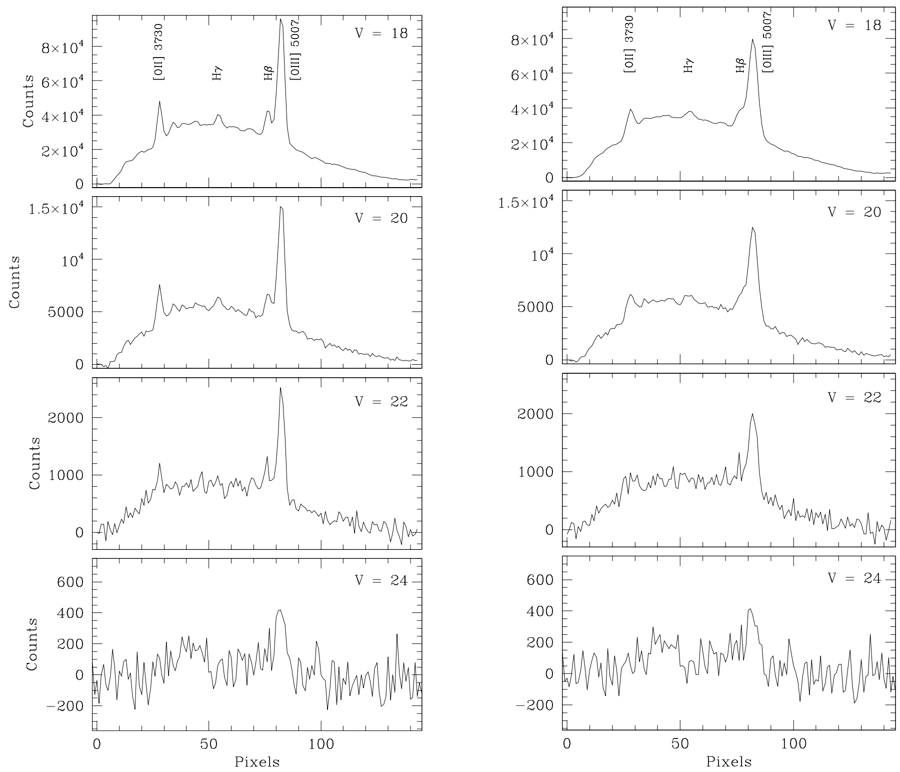

Figure 1. Simulated and extracted spectra of starburst galaxies at z=0.7 (1000 s SLIM simulations). The object is an ellipse of size 2×5 pixels, with PA of 90° (left) and 45° (right) with respect to the dispersion direction. The change in the final spectroscopic resolution is clearly visible.

Python Numeric package, which allowed us to store images as sets of arbitrarily large arrays. Furthermore, input spectra and throughput curves were loaded as Interpolated functions using the Python Scientific module. This allowed us to treat them as if they were continuous functions, and not discrete datasets, allowing us to avoid all the problems associated with non-uniform sampling of the input spectra and throughput curves. The speed of an interpreted scripting language such as Python is somewhat lower than that of a low level C program. This drawback is, however, offset by the lowered development time that is achieved using a modern, feature rich, object oriented language which requires no compilation phase. Furthermore, in reality one can reach close to C performance with extensive use of Python's high level data types and functions, which operate at C language speed. For example, using Numeric array types, one can add two 1000×1000 arrays with a simple Python declaration like $\mathrm{c} = \mathrm{a} + \mathrm{b}$.The actual work of adding up all the elements of the two arrays is done in the underlying, optimized C Numeric module, and performance becomes very close to that achieved by some stand-alone C code.

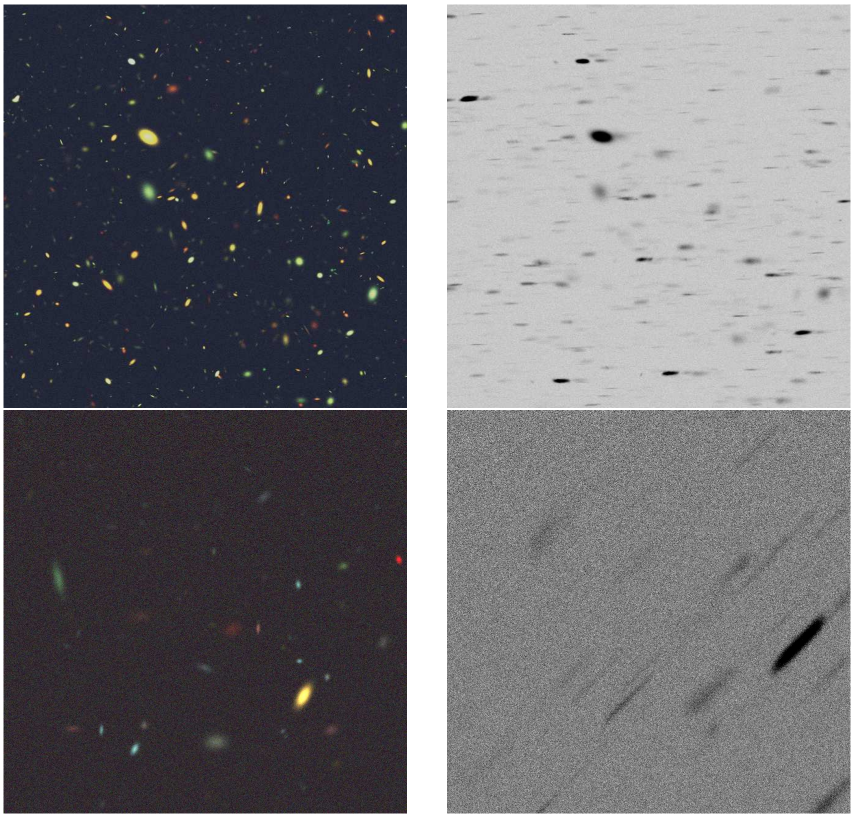

Figure 2. Upper-left: Color composite (F435W, F606W, F814W) using SLIM simulations of ACS/WFC images of an HDF-N like field and (upper-right) the same field when simulated as seen with the WFC grism GL800. The true HDF-N distribution of objects was preserved near the center of this field. Integration time is 1000 seconds in each filter. The field of view is $202'' \times 202''$. Lower panels: SLIM simulations of the same HDF-N like field in 10 hour HRC direct and grism observations. The field of view is $27'' \times 27''$.

3. SLIM Interface

The program interface was kept as simple as possible. Every aspect of the simulator is controlled though the use of three simple text files containing the various simulation parameters. The first one contains a list of input objects (positions, spectral types, red-shifts, and reference magnitudes). The second one contains general simulation parameters such as mirror size, gain, pixel size, and a list of throughput files. Finally, the last configuration file contains a field-dependent polynomial description of the polynomial dispersion relation to apply to the input spectra.

4. SLIM Main Features

The main feature of SLIM are (a) configuration using simple text files, (b) creating slitless simulations by dispersing a 1-D input spectrum and, at each wavelength, convolving it with a PSF of the appropriate shape and size, ¡c) an arbitrary number of throughput files can be used to simulate any combination of mirrors, windows, filters, dispersive elements, and detectors, (d) photometric zero points are computed on-the-fly using the input list of filters and throughputs and a spectrum of Vega, (e) arbitrary n^{th} order polynomial descriptions of the dispersion relations can be used, allowing both near-linear (grism) and highly non-linear (prism) simulations to be generated, and field dependence to be accounted for, (f) no requirement for any a-priori sampling of the input spectrum or throughput files, and (g) producing ACS simulations that agree photometrically within 1 percent with the ACS ETC.

5. Examples

Several examples of spectra which were extracted from simulated ACS WFC grism images are shown in Figure 1. Figure 2 contains simulations of the HDF-N as seen in a three color direct image composite and with the GL800 grism when using the WFC. The same field as seen with the High Resolution Camera (HRC) is also shown. While providing a smaller field of view, the HRC has a higher resolution than WFC in both the direct and grism modes.

6. Conclusion

The use of Python allowed us to efficiently produce a slitless spectroscopic simulator which has been successfully used to produce geometrically and photometrically correct images. More information about SLIM can be found at `http://www.stecf.org/software/`.