

The **aXe** SIMulation package **aXeSIM**

User manual version 1.4



M. Kümmel
and
J.R. Walsh, H. Kuntschner
July 1st 2010

Contents

1	Description	7
1.1	Disclaimer	8
1.2	User-provided input for the PyRAF module	8
1.3	The output of aXeSIM	9
1.4	Limitations of aXeSIM	10
1.5	General use of aXeSIM	10
1.6	The PyRAF module	11
2	Installing aXeSIM	13
2.1	Requirements	13
2.2	Installing from the aXeSIM source distribution	13
2.3	Compiling the C-code	15
2.4	aXeSIM Support	15
3	Illustrated aXeSIM examples	17
3.1	Example 1	17
3.2	Example 2	20
3.3	Example 3	23
3.4	Example 4	28
4	The aXeSIM tasks	33
4.1	High Level Tasks – Low Level Tasks	33
4.2	High Level Tasks	33
4.2.1	SIMDATA	33
4.3	Low Level Tasks	36
4.3.1	PREPIMAGES	37
4.3.2	PREPSPECTRA	37
4.3.3	SIMDIRIM	38
4.3.4	SIMDISPIM	40
5	File Formats	43
5.1	Image Template List	43
5.2	Spectrum Template List	43
5.3	Total Passband File	43

5.4	Template Spectrum	44
5.5	Template Image	44
5.6	Model Images	44
5.7	Model Spectra	44
5.8	Model Object Table	45
6	Environmental variables	47

Preamble

The Space Telescope - European Coordinating Facility (STECF) is responsible for supporting the slitless spectroscopic modes of the Hubble instruments ACS and WFC3 until late 2010 with `stdesk@eso.org` as the main contact address.

After that time, all support (simulation and extraction software, calibration and user support) will be provided by STScI Space Telescope Science Institute (STScI) with `help@stsci.edu` as the central contact.

The slitless spectroscopy URL's provided in this manual all point to www.stecf.org and below, and this web site will be kept up-to-date until late 2010.

Starting in 2011 the webpages for slitless spectroscopy will be hosted at: http://www.stsci.edu/resources/software_hardware/stdas/axe/. Pages at www.stecf.org will reflect this and then point to the corresponding location at www.stsci.edu.

Chapter 1

Description

As part of the ST-ECF support for the spectroscopic modes of Wide Field Camera 3 (WFC3), the slitless spectroscopy group of the ST-ECF has developed a dedicated simulation package applicable to WFC3. Whilst the package was initiated for exploitation of WFC3 slitless grism modes, it is equally applicable to other slitless spectroscopy modes of the Hubble Space Telescope (HST), such as ACS and NICMOS. An option of the simulator is to produce a direct image through a selected filter to match the slitless spectrum image.

The simulation package uses the same components as used by aXe [2] for the extraction of slitless spectra and thus aims at spectrophotometric integrity - useful for observation design but essential for the quantitative assessment of slitless data. The simulation package will help HST users during their Phase I and Phase II proposal preparation to:

- gain a 2D-impression of the layout of the target field and the likely problems to be encountered in slitless spectroscopy, e.g. crowding, spectral overlap, presence of bright spectra;
- check the exposure time to determine a given signal-to-noise in a 2D spectrum. The spectroscopic ETC's at the STScI provide more flexible determination of S/N for an individual 1D spectrum;
- by adjustment, determine the optimal pointing and roll angle for a specific target field, such as to minimize the contamination of spectra of interest;
- learn what to expect (in terms of spectral resolution and sensitivity) from slitless spectra, and thus choose the best instrument and configuration;
- and incidentally become familiar working with and reducing slitless spectroscopic data with the aXe package.

While the primary motivation to start the aXeSIM project was WFC3 with its three grisms, aXeSIM will also work for all the ACS grism and prism modes

(WFC/G800L, HRC/G800L, HRC/PR200L, SBC/PR130L, SBC/PR110). Including these slitless modes simply requires including the necessary configuration files specifying the instrument-specific aspects of the slitless spectra; these files are already available from the ST-ECF web (<http://www.stecf.org/instruments/ACSgrism/calibration>).

The simulation package also finds application to analysis of existing slitless spectroscopy datasets. The simulations can be extracted identically to slitless spectral data, allowing quantitative assessment of detected spectra and spectral features, such as emission lines. Tasks such as matching spectra against templates, convolved with the actual object size and at the grism spectral resolution, determining the detection limits for spectra of given types and measuring cross-contamination between spectra are readily achieved. More extensive studies such as assessing completeness limits of survey observations for a variety of object classes can also be performed.

aXeSIM is made available in two ways. It is distributed via the ST-ECF homepage as a PyRAF/IRAF module and is made available via a web interface. Though based on identical software, the web interface targets new, less experienced or occasional users and offers a subset of the options existing in **aXeSIM**.

In addition to the software we will also provide from our webpages the necessary configuration and sensitivity files to run **aXeSIM** at <http://www.stecf.org/instruments/WFC3grism>. The core part of these files (trace and dispersion description) will coincide with the configuration and calibration files necessary for the reduction of slitless data with the 'traditional' aXe extraction software (see [1], [2], [3] and [4]).

1.1 Disclaimer

aXeSIM was developed independently of the Exposure Time Calculators (ETC) that are provided by the Space Telescope Science Institute (STScI) for the various slitless spectroscopic instrument modes. While every effort has been made to ensure that the imaging simulator provides similar estimates of detected counts to the ETC, differences are bound to result. The STScI ETC's are to be taken as the official tool and should be used to determine exposure times for the purposes of critical time justification in Phase I and II. These ETC's can be found at: <http://etc.stsci.edu/webetc/index.jsp>.

1.2 User-provided input for the PyRAF module

The user will have to prepare only a single list, called `Model Object Table`, with the descriptions of the simulation objects as a minimal input. Then the simulations will be done by executing one PyRAF task `simdata`.

With this minimal input, the `Model Object Table` contains columns with

all the necessary shape and spectral information to completely characterize one object per row. The table format is based on the SExtractor ASCII table format, and the minimal input for the table contains the columns:

- NUMBER: the object identifier; int
- X_IMAGE: the x-position of the object [pix]; float
- Y_IMAGE: the y-position of the object [pix]; float
- A_IMAGE: the major axis rms [pix]; float
- B_IMAGE: the minor axis rms [pix]; float
- THETA_IMAGE: the position angle of the major axis [deg]; float
- MAG_?<number>*: the AB-magnitude at the wavelength; float <number> [nm]

In addition the user has the opportunity to perform more detailed simulations by e.g. using high resolution spectral templates at different redshifts and flux levels, image templates or total passband curves

For more detailed simulations, the object list needs additional columns such as:

- SPECTEMP: the index of the model spectrum to use; int
- MODIMAGE: the index of the image template to use; int
- Z: redshift of the object; float

The columns SPECTEMP and MODIMAGE refer to the spectral templates and image templates to be used in the simulation, respectively. As a first step, the template is translated to the redshift given in the column Z.

1.3 The output of aXeSIM

The output of the aXeSIM simulation task will always consist of a simulated 2D slitless dispersed image with the spectra of the simulated objects. For grism images, the simulations can include several dispersed orders.

If provided with a proper total passband of a filter, aXeSIM will also produce the direct image associated to the slitless image. The total passband curve contains the total system throughput (mirror + instrument + detector + ...) as a function of wavelength for the direct image to be simulated.

The set of direct image – slitless image will be identical to an ideal observed direct image – slitless image pair in standard ACS/WFC3 slitless observations.

This means that a positional offset, which might be applied in the observations by default (as it is the case for e.g. ACS data with HRC/PR200L) will also be present in the simulated images.

Sky background (provided by the user either as single value or background image file in $[e/s]$) and random noise (readout- and photon-noise from background and objects) can be added to the output images.

It is possible to close the loop by performing a simple aXe extraction at the end of the simulations in **aXeSIM**. Alternatively, the user can use SExtractor on the direct images and then perform a standard extraction using the 'original' aXe software. Making the spectral extractions on the simulated data is very important to, for example, check the detectability of spectral features as a function of the signal-to-noise ratio or to make completeness and reliability tests for scientific publications.

All final output of **aXeSIM** is written to the directory to which the environmental variable `AXE_OUTSIM_PATH` is pointing to (see Sect. 6).

1.4 Limitations of aXeSIM

There are a number of effects and features in real data that are currently **not** included in the **aXeSIM** simulations:

- non-linear detector effects such as saturation or blooming are not simulated;
- **aXeSIM** neither adds dark current nor takes dark current into account in the noise model;
- **aXeSIM** does not include a detector or large scale flatfield in the simulations;
- images simulated with **aXeSIM** do not have intra pixel sensitivity variations as images from real detectors (e.g. NICMOS) do;
- **aXeSIM** takes as input object positions and sizes only as pixel coordinates on the detector. No attempt is made to correct for instrument distortions from sky to detector.

Users can compensate some of these **aXeSIM** deficiencies, e.g. by adding the dark current to the sky background, or applying non-linear effects on **aXeSIM** images.

1.5 General use of aXeSIM

aXeSIM was developed for the slitless spectroscopic modes of the HST instruments (WFC3, ACS, NICMOS).

However, **aXeSIM** is not restricted to be used for HST and HST instruments **only**. The software can be used for any telescope and instrument which can be described with aXe configuration and calibration files and the parameters offered by **aXeSIM** (e.g. a sky which is described either in e/s , hence flat, or with a sky image in $[e/s]$).

To use **aXeSIM** for simulating data for a new telescope/instrument, you have to:

- create a new aXe configuration file describing the trace and dispersion solutions for all spectral orders (see Sect. 5.2 of the aXe manual);
- derive sensitivity files for all spectral orders;
- derive total passband files (see Sect. 5.3) for the direct image filters (for simulating direct images; see below);
- specify the collecting area of the telescope [cm^2] with the keyword **TELAREA** in the configuration file (for simulating direct images; see below).

The sensitivity files for the spectral orders must implicitly **include** the telescope size. For flux scaling spectral templates to a given AB-magnitude, **aXeSIM** needs only **relative** throughputs. In case that the total passbands for the filters are not available, “standard” passband files for e.g. the Johnson or Kron-Cousins photometric system can be used.

For simulations of direct images, however, the total passband of the filter and the collecting area of the telescope must be known. The **aXeSIM** default (if not specified in the configuration file) for the collecting area is the HST mirror size ($45238.93cm^2$, calculated from its diameter). For HST, the shading by the secondary mirror is included in the throughput of the mirrors, which is an input for computing the total passbands of the filters.

1.6 The PyRAF module

Similar to the aXe extraction package [2] (available in PyRAF under `stdas.slitless.axe`) the **aXeSIM** package for PyRAF has three layers:

- at the bottom layer, executables compiled from C-programs provide the fast processing speed necessary for computing the pixel values in the simulated images. Most of this code does already exist as part of the aXe extraction package (in ‘quantitative contamination’, see [5]). This guarantees the symmetry between the simulations and the extractions. Moreover this ‘legacy’ code has been used for some years and therefore has proven to be without major bugs or problems.
- a middle layer with python scripts connects the various C-executables and performs less computationally expensive preparatory work
- a very thin upper layer makes the python scripts accessible from within PyRAF.

This concept is currently followed in all software published in the `stsci.python` package [\[6\]](#).

Chapter 2

Installing aXeSIM

2.1 Requirements

The following are required to run aXeSIM:

- STSDAS 3.11 or later (http://www.stsci.edu/resources/software_hardware/stsdas)
- PyRAF 1.8 or later (http://www.stsci.edu/resources/software_hardware/pyraf)

2.2 Installing from the aXeSIM source distribution

After downloading the aXeSIM-1.4 tarball from (http://www.stecf.org/software/slitless_software/axesim), move it to the installation directory `/your/aXeSIM/path` and unpack it there with:

```
>gunzip taxe21_taxesim14_src.tar.gz
>tar -xvf taxe21_taxesim14_src.tar
```

aXeSIM consists of a part written in ANSI C and a second part written in Python (www.python.org).

For several operating systems, statically linked binaries are available at: http://www.stecf.org/software/slitless_software/axesim To install them, please download them and do the following:

```
>mv aXe-2.1.<arch>.bin.tar.gz /your/aXeSIM/path/taxe/iraf/bin/
>cd /your/aXeSIM/path/taxe/iraf/bin/
>gunzip aXe-2.1.<arch>.bin.tar.gz
>tar -xvf aXe-2.1.<arch>.bin.tar
```

In the (hopefully unlikely) case that these binaries do not work, or you work with a different operating system, you have to build the executables by yourself as described in Chapter 2.3.

On the Python side there exists a script to compile the code. Go to the Python directory and compile the Python code there with:

```
>cd /your/aXeSIM/path/taxe/iraf/
>python compileaXe.py .
```

If all went well, aXeSIM-1.4 is now ready. Since aXeSIM-1.4, the code basis of aXeSIM and the extraction package aXe [2] are united. As a consequence, the new version 2.1 of aXe is delivered and installed together with aXeSIM-1.4.

The new packages with their tasks must be declared in PyRAF. To do this, add the following lines near the end of your `login.cl` file, or in `loginuser.cl`:

```
reset  taxe21      = /your/aXe2.1/path/taxe/iraf/
task   taxe21.pkg  = taxe21$taxe21.cl
reset  helpdb      = (envget("helpdb") //" ,taxe21$lib/helpdb.mip")

reset  taxesim14   = /your/aXe2.1/path/taxe/iraf/
task   taxesim14.pkg = taxesim14$taxesim14.cl
reset  helpdb      = (envget("helpdb") //" ,taxesim14$lib/helpdb.mip")
```

The next time PyRAF is launched, the package aXeSIM and aXe are available. aXeSIM can then be loaded as any other package by simply typing its name:

```
--> taxesim14
```

```
The aXeSIM software package 1.4 was developed by the
Slitless Spectroscopy Group of the ST-ECF. Maintenance
is provided by the Space Telescope Science Institute.
Further information is available at:
http://www.stsci.edu/resources/software_hardware/stsdas/axe
```

```
Any questions regarding this software can be directed to:
help@stsci.edu
```

```
taxesim14/:
  tprepimages  tprepspectra  tsimdata  tsimdirim  tsimdispim
```

The message which appears during the loading of the package and the task overview indicate that everything went OK and that the tasks can be used from now on. The package can be used by more than one user. Other users only have to modify their `login.cl` or `loginuser.cl` as described above to access the aXeSIM package and the tasks within it (of course provided that they have access to the installation directory).

aXeSIM was successfully built and tested under Fedora 6. It should be no problem to install aXeSIM under other Unix or Unix-like operating systems such as HP-UX or MacOSX.

2.3 Compiling the C-code

To compile the C-code you need:

- a C compiler such as GNUC CC
- GNU Scientific Libraries (<http://www.gnu.org/software/gsl/>)
- WCStools 3.x libraries (<http://tdc-www.harvard.edu/software/wcstools/>)
- CFITSIO 3.x libraries (<http://heasarc.gsfc.nasa.gov/docs/software/fitsio/fitsio.html>)

For the C part, a configure script is included with aXeSIM. To configure and then compile the C tasks do the following:

```
>cd taxe/iraf/ccc
>./configure
```

For MacOSX, the option “-build=powerpc” must be added to the `configure` command for Power PC machines, and the option “-build=i386-pc-macosx” for Intel Macs. If some libraries used by aXeSIM are not installed in the usual places, online parameters must be used to tell the configure script where to find them. For example

```
./configure --with-cfitsio-prefix=/your_axelibs/cfitsio
            --with-gsl-prefix=/your_axelibs/gsl-1.0
            --with-wcstools-prefix=/your_axelibs/wcstools-3.0.4
```

specifies explicitly the location of the GSL and the CFITSIO library. Follow the instructions given by the configure script to solve problems. The configure script generates a Makefile which is used to compile the aXeSIM binaries. Type

```
>make
```

to execute the Makefile and create the binaries. The tasks must be installed in the `bin` sub-directory of the aXeSIM-iraf directory. Simply execute

```
>make install
```

to move the binaries to their proper location.

2.4 aXeSIM Support

The aXeSIM software package was developed by a dedicated group at the Space Telescope - European Coordinating Facility (STECF). STECF will be supporting the use of the aXe software and the slitless spectroscopic modes of ACS and WFC3 until late 2010. Until that time, you can request further help and information via email to stdesk@eso.org. After that time, support will be provided by STScI via email to help@stsci.edu.

Chapter 3

Illustrated aXeSIM examples

In this chapter we show some examples for aXeSIM simulations. In a step-by-step approach, we present the simulation input, the aXeSIM task executed and display the simulation results. In order to illustrate all possibilities of aXeSIM, we start with very simple examples and finish with rather complex simulations that expose all options in aXeSIM.

Working with aXeSIM frequently requires modifying or creating ASCII tables in the SEXtractor catalogue format. The python module `AstroAsciiData` (distributed at: <http://www.stecf.org/software/PYTHONtools/astroasciidata/>) is very convenient to do this interactively or within small python scripts.

3.1 Example 1

Simulation details

instrument:	WFC3/IR G102
# of objects:	1
object spectrum:	defined by one magnitude value
object brightness:	defined by one magnitude value
flux normalization:	not required
object shape	Gaussian
default extraction	YES

Directories and files before the simulation

```
example_1>ls -R
.:
CONF DATA OUTPUT OUTSIM

./CONF:
wfc3_abscal_IRg102_0th_sens.fits wfc3_abscal_IRg102_2nd_sens.fits
wfc3_abscal_IRg102_1st_sens.fits WFC3_IR.G102_TV2_sim.conf

./DATA:
example_1_MOT.dat

./OUTPUT:

./OUTSIM:
example_1>setenv AXE_IMAGE_PATH ./DATA
```

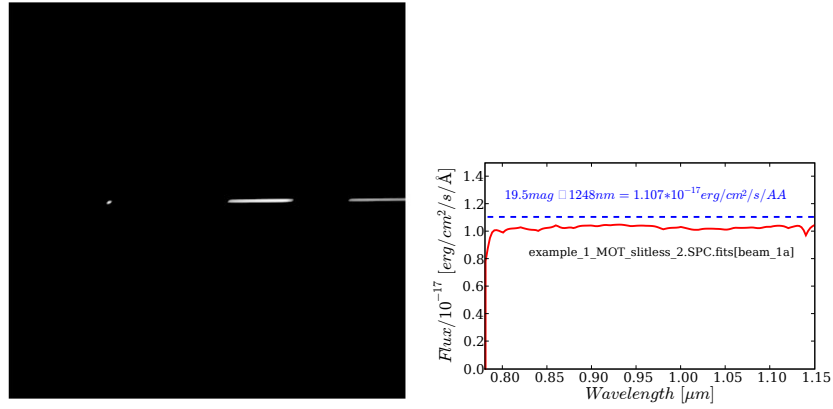


Figure 3.1: The results of the first simulation: the simulated slitless image in the left panel and the default-extracted spectrum (solid, red line) in comparison to the input “spectrum” (dashed, blue line) in the right panel. The extraction box did not include all the source flux.

```
example_1>setenv AXE_OUTPUT_PATH ./OUTPUT
example_1>setenv AXE_CONFIG_PATH ./CONF
example_1>setenv AXE_OUTSIM_PATH ./OUTSIM
```

The Model Object Table example_1.MOT.dat

```
# 1 NUMBER
# 2 X_IMAGE
# 3 Y_IMAGE
# 4 A_IMAGE
# 5 B_IMAGE
# 6 THETA_IMAGE
# 7 MAG_F1248W
1 512.0 512.0 2.0 1.0 30.5 19.5
```

The simulation command

```
--> simdata incat='example_1.MOT.dat' config='WFC3.IR.G102.TV2_sim.conf' extraction='YES'
```

Directories and files after the simulation

```
example_1>ls -R
.:
CONF DATA OUTPUT OUTSIM

./CONF:
wfc3_abscal_IRg102_0th_sens.fits wfc3_abscal_IRg102_2nd_sens.fits WFC3.IR.G102.TV2_sim.conf.simul
wfc3_abscal_IRg102_1st_sens.fits WFC3.IR.G102.TV2_sim.conf

./DATA:
example_1.MOT.dat

./OUTPUT:

./OUTSIM:
example_1.MOT_slitless_2.SPC.fits example_1.MOT_slitless_2.STP.fits example_1.MOT_slitless.fits
```

Spectral Extrapolation and Interpolation in aXeSIM simulations

When simulating dispersed or direct images, aXeSIM simply extrapolates the last input data points into the regions with no spectral information. Hence an object with a single AB-magnitude as “spectrum” is simulated as a flat-spectrum (in f_λ) source at the corresponding f_λ -value (see Fig.3.1), but also spectral templates are extrapolated as flat-spectrum sources. For interpolation within the defined spectral range, aXeSIM uses *linear interpolation* for spectral energy distribution defined by several magnitude values and for high resolution values.

The simulation results

As already mentioned in Sect. 1.3, aXeSIM collects all output in the directory AXE_OUTSIM_PATH, which is ./OUTSIM in this example. This result comprises in detail:

- the simulated slitless image `example_1.MOT_slitless.fits`
- the extracted 1D spectrum `example_1.MOT_slitless_2.SPC.fits`
- the 2D grism stamp image file `example_1.MOT_slitless_2.STP.fits`

Figure 3.1 displays the main results of the example, the simulated slitless image (left) and the extracted spectrum (right). The blue line marks the “input spectrum”, which is a flat spectrum at the value f_λ corresponding to the given AB-magnitude ($19.5mag_{AB}@1248nm = 1.107 * 10^{-17} erg/cm^2/s/\text{\AA}$).

Some annotations

- The dispersed image in Fig. 3.1 shows, besides the central first order, also the zeroth order (left) and second order (right) slitless spectrum.
- The extracted spectrum appears to differ in flux level from the input spectrum (solid and dashed lines in Fig. 3.1). The quantitative difference is explained by the finite extraction width (which corresponds to the aXe parameter $m_{fwhm} = 3.0$, see Chapt. 4.2.1 for an explanation of this parameter).
- An observant reader may have noticed that, after performing the example simulation, there is, next to the configuration file `WFC3_IR.G102.TV2_sim.conf`, a file with the similar name `WFC3_IR.G102.TV2_sim.conf.simul`. aXeSIM has created and used in the simulations this variant of the original configuration file. The new configuration file contains a keyword for the telescope area (necessary for generating direct images), which has the HST collecting area as default. Moreover the new configuration file **always** marks the extensions 1, 2 and 3 as science-, error- and dq-extension. As a consequence, aXeSIM simulates only images with one science extension, and for simulating ACS/WFC and WFC3/UVIS grism images the two chips must

be treated separately. Concerning the essential quantities, the descriptions of trace and dispersion of the various orders, there is **no** difference between the original and the new configuration file, and both can be used to extract spectra from the simulated slitless images (the original one may need to be corrected to address the correct image extension).

- A close inspection of the Model Object Table reveals that during the simulation some additional columns were appended. For internal reasons these added columns are necessary for running **axE**SIM. However these modifications to the Model Object Table **never** change essential simulation input given by the user, and repeating a simulation with the modified table will yield the **identical** simulated images.

3.2 Example 2

Simulation details

instrument:	WFC3/IR G102
# of objects:	1
object spectrum:	spectral energy distribution defined by several magnitude values
object brightness:	defined by several magnitude values
flux normalization:	not required
object shape	template image
default extraction	YES
direct image	YES, filter: wfc3_ir_f105w_tpass_m01.dat

Directories and files before the simulation

```

>ls -R
.:
CONF DATA OUTPUT OUTSIM SIMDATA template_images.lis

./CONF:
wfc3_abscal_IRg102_0th_sens.fits wfc3_abscal_IRg102_2nd_sens.fits
wfc3_abscal_IRg102_1st_sens.fits WFC3_IR.G102.TV2_sim.conf

./DATA:
example_2_MOT.dat

./OUTPUT:

./OUTSIM:

./SIMDATA:
spiral_1.fits wfc3_ir_f105w_tpass_m01.dat
WFC3_IR_1400nm_psf.fits WFC3_UVIS_350nm_psf.fits
WFC3_IR_950nm_psf.fits

>setenv AXE_IMAGE_PATH ./DATA
>setenv AXE_OUTPUT_PATH ./OUTPUT
>setenv AXE_CONFIG_PATH ./CONF
>setenv AXE_SIMDATA_PATH ./SIMDATA
>setenv AXE_OUTSIM_PATH ./OUTSIM

```

The Model Object Table example_2_MOT.dat

```

# 1 NUMBER
# 2 X_IMAGE
# 3 Y_IMAGE
# 4 A_IMAGE
# 5 B_IMAGE
# 6 THETA_IMAGE
# 7 MAG_F432W

```

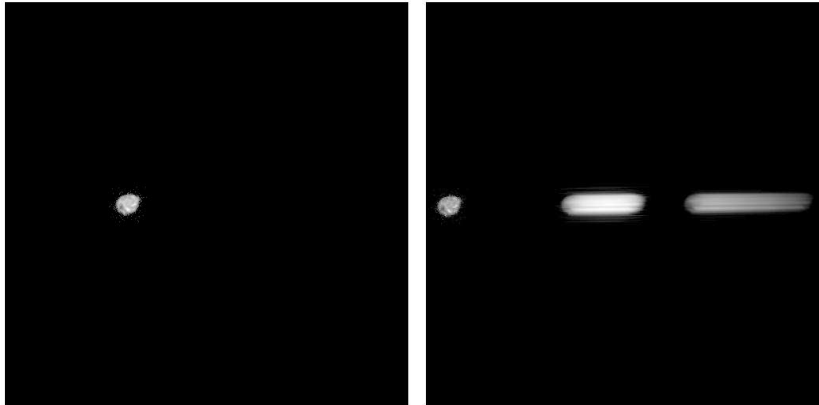


Figure 3.2: The results of the second simulation: the direct image (left) and the slitless image (right). Both images are displayed with a logarithmic lookup table to enhance the contrast.

```
# 8 MAG_F592W
# 9 MAG_F769W
# 10 MAG_F906W
# 11 MAG_F1123W
# 12 MAG_F1603W
# 13 MODIMAGE
269 312.0 512.0 8.0 8.0 90.0 22.984 22.163 21.890 21.807 22.027 21.788 4
```

The Image Template List `template_images.lis`

```
> more template_images.lis
WFC3_IR_1400nm_psf.fits
WFC3_IR_950nm_psf.fits
WFC3_UVIS_350nm_psf.fits
spiral_1.fits
```

The simulation command

```
--> simdata incat='example_2_MOT.dat' config='WFC3_IR.G102.TV2_sim.conf'
inlist_ima='template_images.lis' tpass_direct='wfc3_ir_f105w_tpass_m01.dat'
```

Directories and files after the simulation

```
> ls -R
.:
CONF DATA gaga.txt OUTPUT OUTSIM SIMDATA template_images.lis

./CONF:
wfc3_abcscal_IRg102_0th_sens.fits WFC3_IR.G102.TV2_sim.conf
wfc3_abcscal_IRg102_1st_sens.fits WFC3_IR.G102.TV2_sim.conf.simul
wfc3_abcscal_IRg102_2nd_sens.fits

./DATA:
example_2_MOT.dat template_images.fits

./OUTPUT:

./OUTSIM:
example_2_MOT_direct.fits          example_2_MOT_slitless.fits
example_2_MOT_slitless_2.SPC.fits  template_images.fits
example_2_MOT_slitless_2.STP.fits

./SIMDATA:
spiral_1.fits          wfc3_ir_f105w_tpass_m01.dat
WFC3_IR_1400nm_psf.fits wfc3_ir_f105w_tpass_m01.fits
WFC3_IR_950nm_psf.fits  WFC3_UVIS_350nm_psf.fits
```

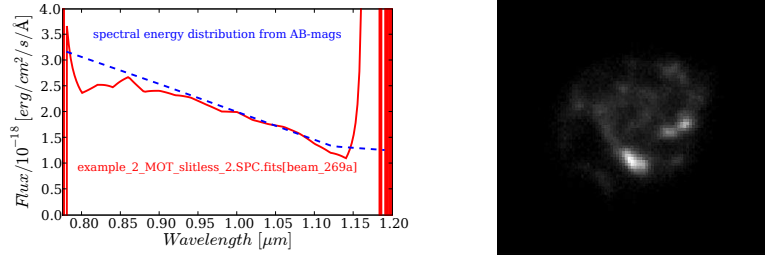


Figure 3.3: Left panel: Comparison between the default extracted spectrum (red, solid line) and the input spectrum (dashed, blue line), a spectral energy distribution defined by the AB-magnitude values. Right panel: The image template.

The simulation results

In the Model Object Table, the column `MODIMAGE` contains the value 4, thus pointing to the fourth entry in the Image Template List. This entry, named `spiral_1.fits`, is then used as the image template when simulating the direct and dispersed image. The spiral structure of the template image is clearly visible in Figure 3.2, which shows the direct image and the dispersed image on the left and right side, respectively.

In total, the results comprise:

- the simulated slitless image `example_2_MOT_slitless.fits`
- the simulated direct image `example_2_MOT_direct.fits`
- the extracted 1D spectrum `example_1_MOT_slitless_2.SPC.fits`
- the 2D grism stamp image file `example_1_MOT_slitless_2.STP.fits`
- the Model Images `template_images.fits`

The Model Images `template_images.fits` is a multi-extension fits image which contains all template images listed in `template_images.lis` and replicates user input. `axEsim` needs the template images in this compact form and generates this file, hence it is considered as part of the result.

Figure 3.3 shows a comparison of the input spectrum and the default extracted spectrum in the left panel and the image template in the right panel. The input spectral energy distribution is defined by the AB-magnitudes and a linear interpolation in between the values.

Some annotations

- The simulated object and brightness was taken from the NICMOS UDF catalogue published in [7]. The UDF was observed in the ACS filters

F425W, F606W, F775W, F814W, F850LP and the NICMOS filters F110W and F160W. However in the column names we have used the pivot wavelength of the filter (always in $[nm]$) and not the less accurate filter name to give the characteristic wavelength of the filter.

- The total passband (see Sect. 5.3 for a definition) of the direct image was given as the ASCII table `wfc3_ir_f105w_tpass_m01.dat` and transformed to the equivalent fits file `wfc3_ir_f105w_tpass_m01.fits` in `aXeSIM`.
- The default extraction of `aXeSIM` uses, like `aXe`, the columns `A_IMAGE` and `B_IMAGE` in the object table to determine the extraction width. For the definitely extended object in the image template (see Fig. 3.3) the tentative value 8.0 was given. An independent `SExtractor` run on the direct image would deliver the correct value. The Gauss parameters (`A_IMAGE`, `B_IMAGE` and `THETA_IMAGE`) are not used when making the simulated images, since the object shape is defined with a template image.
- There are a number of reasons why the default extracted spectrum does not completely coincide with the input spectrum in Fig. 3.3. As in the first example, there is a wiggle on the long wavelength end due to the object extent and the drop in sensitivity. Also when using image templates, care must be taken to put the sources into the centre of the template. `aXeSIM` puts the centre of the template image at the nominal position given in the Model Object Table, and the default extraction also uses this position for the extraction. An off-centre template source results in a wavelength shift between the simulation and the extraction, which can translate into a flux difference between the simulated and the extracted spectrum. Running `SExtractor` on the direct image and using the catalogue in an `aXe` spectral extraction would avoid this problem

3.3 Example 3

Simulation details

instrument:	WFC3/IR G102
# of objects:	6
object spectrum:	high resolution spectra at different redshift
object brightness:	defined at $\lambda = 11550\text{\AA}$
flux normalization:	in $[10400, 10600]\text{\AA}$
object shape	template image
default extraction	YES
direct image	YES, filter: <code>wfc3_ir_f110w_tpass_m01.dat</code>

Directories and files before the simulation

```
example_3> ls -R
.:
CONF DATA OUTPUT OUTSIM SIMDATA template_images.lis template_spectra.lis
```

```

./CONF:
wfc3_abcscal_IRg102_0th_sens.fits  WFC3_IR_G102_TV2_sim.conf
wfc3_abcscal_IRg102_1st_sens.fits
wfc3_abcscal_IRg102_2nd_sens.fits

./DATA:
example_3_MOT.dat

./OUTPUT:

./OUTSIM:

./SIMDATA:
s0_template.fits      WFC3_IR_1400nm_big_psf.fits  wfc3_ir_f105w_tpass_m01.dat
starb1_template.fits  WFC3_IR_1400nm_psf.fits          wfc3_ir_f110w_tpass_m01.dat
starb2_template.fits  WFC3_IR_950nm_big_psf.fits      WFC3_UVIS_350nm_psf.fits
starb3_template.fits  WFC3_IR_950nm_psf.fits

example_3>setenv AXE_IMAGE_PATH ./DATA
example_3>setenv AXE_OUTPUT_PATH ./OUTPUT
example_3>setenv AXE_CONFIG_PATH ./CONF
example_3>setenv AXE_OUTSIM_PATH ./OUTSIM
example_3>setenv AXE_SIMDATA_PATH ./SIMDATA

```

The Model Object Table example_3_MOT.dat

```

# 1 NUMBER
# 2 X_IMAGE
# 3 Y_IMAGE
# 4 A_IMAGE
# 5 B_IMAGE
# 6 THETA_IMAGE
# 7 MAG_F1155W
# 8 SPECTEMP
# 9 Z
# 10 MODIMAGE
  1 500.0 100.0 1.5 1.5 90.0 20.0 3 0.5 3
  2 500.0 120.0 1.5 1.5 90.0 20.0 3 0.7 3
  3 500.0 140.0 1.5 1.5 90.0 20.0 3 0.9 3
  4 500.0 160.0 1.5 1.5 90.0 20.0 3 1.1 3
  5 500.0 180.0 1.5 1.5 90.0 20.0 3 1.3 3
  6 500.0 200.0 1.5 1.5 90.0 20.0 3 1.5 3

```

The Image Template List template_images.lis

```

-> more template_images.lis
WFC3_IR_1400nm_big_psf.fits
WFC3_IR_1400nm_psf.fits
WFC3_IR_950nm_big_psf.fits
WFC3_IR_950nm_psf.fits
WFC3_UVIS_350nm_psf.fits

```

The Spectrum Template List template_spectra.lis

```

-> more template_spectra.lis
s0_template.fits
starb1_template.fits
starb2_template.fits
starb3_template.fits

```


Flux normalization of template spectra

The flux normalization of the template spectra is done by comparing the desired object flux values given in the Model Object Table (MOT) with the average flux values of the redshifted template spectra in the passband specified in the parameter `tpass_flux`. For the objects in the example in Sect. 3.3, the targeted flux value is $mag_{AB} = 20.0$ at 1155 nm (given in the wavelength column name) for all objects, which is $f_{\lambda} = 8.1593710^{-18}\text{ erg/cm}^2/\text{s}/\text{\AA}$. Thus, all redshifted template spectra were normalized to have this flux value in the interval $tpass_flux = [1040, 1060]\text{ nm}$. There is **no** correction applied if the wavelength given in the MOT and the passband specified in `tpass_flux` are dissimilar, as in this example. In case that the MOT contains several magnitude columns, the wavelength closest to the pivot wavelength of the passband given in `tpass_flux` is selected to compute the target flux level.

During the simulation phase `aXeSIM` extrapolates the object spectra or energy distributions as outlined in the box on page 19. However **no** extrapolation is attempted for the flux normalization. In the case that one of the redshifted templates does **not** completely cover the wavelength range of the normalization passband, `simdata` just stops at an early processing stage with an according error message.

The simulation command

```
--> simdata incat='example_3_MOT.dat' config='WFC3_IR.G102.TV2_sim.conf'
      utput_root='StarBurst' inlist_spec='template_spectra.lis' tpass_flux='1040,1060'
      inlist_ima='template_images.lis' exptime_disp=2000.0 bck_flux_disp=1.0
      tpass_direct='wfc3_ir_f110w_tpass_m01.dat' exptime_dir=100.0 bck_flux_dir=0.5
```

Directories and files after the simulation

```
> ls -R
.:
CONF DATA OUTPUT OUTSIM SIMDATA template_images.lis template_spectra.lis

./CONF:
wfc3_abscal_IRg102_0th_sens.fits  WFC3_IR.G102.TV2_sim.conf
wfc3_abscal_IRg102_1st_sens.fits  WFC3_IR.G102.TV2_sim.conf.simul
wfc3_abscal_IRg102_2nd_sens.fits

./DATA:
example_3_MOT.dat  StarBurst_images.fits  StarBurst_spectra.fits

./OUTPUT:

./OUTSIM:
StarBurst_direct.fits  StarBurst_slitless_2.SPC.fits  StarBurst_slitless.fits
StarBurst_images.fits  StarBurst_slitless_2.STP.fits  StarBurst_spectra.fits

./SIMDATA:
s0_template.fits      WFC3_IR_1400nm_big_psf.fits  wfc3_ir_f105w_tpass_m01.dat
starb1_template.fits  WFC3_IR_1400nm_psf.fits     wfc3_ir_f110w_tpass_m01.dat
starb2_template.fits  WFC3_IR_950nm_big_psf.fits  wfc3_ir_f110w_tpass_m01.fits
starb3_template.fits  WFC3_IR_950nm_psf.fits     wfc3_UVIS_350nm_psf.fits
```

The simulation results

The six objects simulated in this example all use the same model image, which is a WFC/IR PSF at 950 nm . All objects are based on the identical high resolution

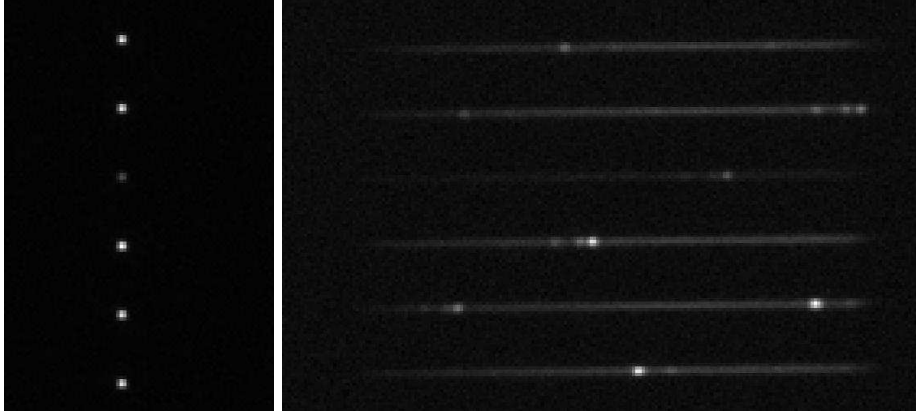


Figure 3.4: The results of the third simulation: the direct image (left) and the slitless image (right). Only a cutout image of 80×130 pix and 200×130 pix around the simulated objects is shown for the direct and slitless image, respectively.

spectral template `starb2_template.fits`, however the template is shifted to different redshift values prior to simulating the images.

Figure 3.4 shows the direct image and the slitless image in the left and right panels, respectively. As the redshift increases from bottom to top, the emission lines are shifted to longer wavelengths (to the right).

In total, the results comprise:

- the direct image `StarBurst_direct.fits`
- the slitless image `StarBurst_slitless.fits`
- the Model Images `StarBurst_images.fits`
- the Model Spectra `StarBurst_spectra.fits`
- the default extracted 1D spectra `StarBurst_slitless_2.SPC.fits`
- the 2D grism stamp image file `StarBurst_slitless_2.STP.fits`

All results start with the string given in the `simdata` parameter `output_root`. Before simulating the direct and slitless images, the template spectra are shifted in redshift and in flux. These spectra are stored in the Model Spectra file `StarBurst_spectra.fits`. The column `MODSPEC`, which `simdata` appends to the Model Object Table, contains for each object the corresponding extension in the Model Spectra file.

Figure 3.5 shows a comparison between the default extracted spectra from `StarBurst_slitless_2.SPC.fits` and their corresponding extension in `StarBurst_spectra.fits` for all objects in this example. The blue, dashed line marks the template, and

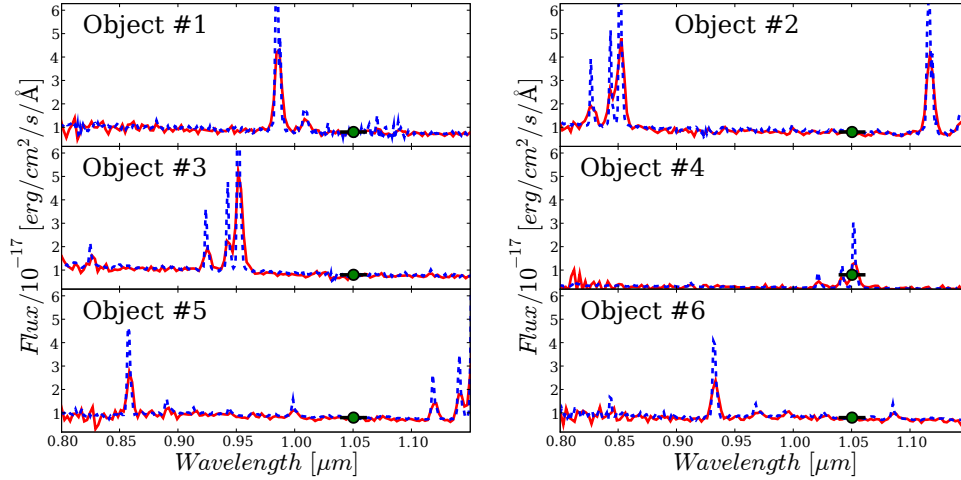


Figure 3.5: Comparison between the default extracted spectrum (red, solid line) and the simulated spectrum (dashed, blue line). The green point marks the flux normalization value, and the black bar the wavelength interval for the flux normalization.

the red, solid line the extracted spectrum. The green point marks the wavelength and normalization flux value, the black bar indicates the wavelength interval used for the flux normalization.

Some annotations

- Since exposure time values were specified for the slitless and the direct image, these images contain noise according to a simple noise model which takes into account photon noise (from sources and sky background) and readout noise (given in the configuration file). In addition, this noise model is used to generate the error extension of the slitless and direct images. The noise is clearly visible in the background of Fig. 3.4 and the continuum of the extracted spectra in Fig. 3.5.
- Background levels specified in the `simdata` parameters (`bck_flux_disp` and `bck_flux_disp`) are added to the simulated images. However the default spectral extraction is done on a temporary slitless image that had the background removed. Hence the default extraction in `aXeSIM` is always perfect concerning the sky background removal.
- When executing the task `simdata` with a long list of parameters as in this example, it is good practice **not** enter the command line directly but to work with a small PyRAF or cl script that contains the entire command.
- Due to the narrow passband chosen for flux normalization, the continuum

levels of the simulated objects vary significantly. At the redshift of object #4, the average flux in normalization passband is very high due to the strong emission line (see Fig. 3.5). This results in a large normalization factor and a lower continuum level of this source (see object traces in Fig. 3.4) as compared to the other objects.

3.4 Example 4

Simulation details

instrument:	WFC3/IR G141
# of objects:	14
object spectrum:	high resolution spectra and spectral energy distributions
object brightness:	defined at $\lambda = 11550\text{\AA}$
flux normalization:	filter: wfc3_ir_f105w_tpass_m01.dat
object shape	template images and Gaussian shapes
default extraction	YES
direct image	YES, filter: wfc3_ir_f105w_tpass_m01.dat

Directories and files before the simulation

```
example_4>ls -R
.:
CONF DATA OUTPUT OUTSIM SIMDATA template_images.lis template_spectra.lis

./CONF:
wfc3_abscal_IRg141_0th_sens.fits wfc3_abscal_IRg141_3rd_sens.fits
wfc3_abscal_IRg141_1st_sens.fits WFC3_IR_G141_TV2_sim.conf
wfc3_abscal_IRg141_2nd_sens.fits

./DATA:
example_4.MOT.dat

./OUTPUT:

./OUTSIM:
ObjMix_direct.fits ObjMix_slitless_2.SPC.fits ObjMix_slitless.fits
ObjMix_images.fits ObjMix_slitless_2.STP.fits ObjMix_spectra.fits

./SIMDATA:
Mann_E.dat          starb3_template.fits      WFC3_IR_950nm_psf.fits
Mann_S0.dat         WFC3_IR_1400nm_big_psf.fits wfc3_ir_f105w_tpass_m01.dat
starb1_template.fits WFC3_IR_1400nm_psf.fits  wfc3_ir_f110w_tpass_m01.dat
starb2_template.fits WFC3_IR_950nm_big_psf.fits WFC3_UVIS_350nm_psf.fits

example_4>setenv AXE_IMAGE_PATH ./DATA
example_4>setenv AXE_OUTPUT_PATH ./OUTPUT
example_4>setenv AXE_CONFIG_PATH ./CONF
example_4>setenv AXE_OUTSIM_PATH ./OUTSIM
example_4>setenv AXE_SIMDATA_PATH ./SIMDATA
```

The Model Object Table example_2.MOT.dat after a simdata run

Unlike the previous examples, we have listed here the Model Object Table as it looks **after** the execution of `simdata`. During the simulations, the columns #11-#16 were appended. No original table entry in #1-#10 was modified.

```
# 1 NUMBER
# 2 X_IMAGE
# 3 Y_IMAGE
# 4 A_IMAGE
# 5 B_IMAGE
# 6 THETA_IMAGE
# 7 MAG_F1155W
# 8 SPECTEMP
# 9 Z
# 10 MODIMAGE
```

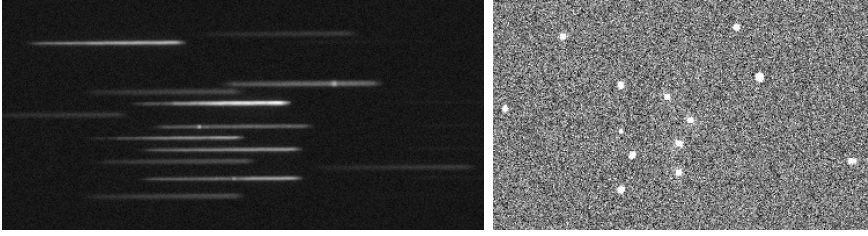


Figure 3.6: The image results of the fourth simulation: the slitless image (left) and the direct image. Only coutouts of 400×200 pix and 320×200 pix centered on the objects are shown.

```
# 11 MODSPEX
# 12 A_WORLD
# 13 B_WORLD
# 14 THETA_WORLD
# 15 X_WORLD
# 16 Y_WORLD
1 450.0 170.0 1.6 1.3 270.0 21.0 0 0.0 0 0 1.600000e+00 1.300000e+00 2.700000e+02 5.320378e+01 -2.778795e+01
2 460.0 110.0 1.5 1.2 45.0 21.0 0 0.0 0 0 1.500000e+00 1.200000e+00 4.500000e+01 5.320448e+01 -2.778852e+01
3 450.0 130.0 1.9 1.5 -70.0 21.0 5 2.0 3 1 1.900000e+00 1.500000e+00 -7.000000e+01 5.320431e+01 -2.778826e+01
4 450.0 80.0 1.5 1.5 90.0 21.0 0 0.0 0 0 1.500000e+00 1.500000e+00 9.000000e+01 5.320497e+01 -2.778865e+01
5 500.0 95.0 1.5 1.5 90.0 20.0 3 0.5 1 2 1.500000e+00 1.500000e+00 9.000000e+01 5.320429e+01 -2.778908e+01
6 500.0 120.0 1.5 1.5 90.0 20.0 0 0.7 1 0 1.500000e+00 1.500000e+00 9.000000e+01 5.320396e+01 -2.778888e+01
7 650.0 105.0 2.5 1.5 0 21.5 0 0.0 0 0 2.500000e+00 1.500000e+00 0.000000e+00 5.320272e+01 -2.779064e+01
8 350.0 150.0 1.5 1.0 90.0 21.5 0 0.0 0 0 1.500000e+00 1.000000e+00 9.000000e+01 5.320500e+01 -2.778701e+01
9 550.0 220.0 1.5 1.5 10.0 21.5 0 0.0 0 0 1.500000e+00 1.500000e+00 1.000000e+01 5.320217e+01 -2.778866e+01
10 50.0 105.0 1.5 1.5 135.0 21.5 0 0.0 0 0 1.500000e+00 1.500000e+00 1.350000e+02 5.320847e+01 -2.778407e+01
11 510.0 140.0 1.5 1.5 90.0 20.0 2 0.9 1 3 1.500000e+00 1.500000e+00 9.000000e+01 5.320360e+01 -2.778884e+01
12 490.0 160.0 1.5 1.5 90.0 20.0 5 2.1 1 4 1.500000e+00 1.500000e+00 9.000000e+01 5.320353e+01 -2.778846e+01
13 570.0 177.0 1.5 1.5 90.0 20.0 4 1.3 0 5 1.500000e+00 1.500000e+00 9.000000e+01 5.320254e+01 -2.778921e+01
14 400.0 212.0 1.5 1.5 90.0 20.0 1 1.5 1 6 1.500000e+00 1.500000e+00 9.000000e+01 5.320371e+01 -2.778707e+01
```

The Image Template List `template_images.lis`

```
-> more template_images.lis
WFC3_IR_1400nm_big_psf.fits
WFC3_IR_1400nm_psf.fits
WFC3_IR_950nm_big_psf.fits
WFC3_IR_950nm_psf.fits
WFC3_UVIS_350nm_psf.fits
```

The Spectrum Template List `template_spectra.lis`

```
-> more template_spectra.lis
Mann_S0.dat
starb1_template.fits
starb2_template.fits
starb3_template.fits
Mann_E.dat
```

The simulation command

```
-> simdata incat='example_4_MOT.dat' config='WFC3_IR.G141.TV2_sim.conf' output_root='ObjMix'
inlist_spec='template_spectra.lis' tpass_flux='wfc3_ir_f110w_tpass_m01.dat'
inlist_ima='template_images.lis' exptime_disp=2000.0 bck_flux_disp=1.4
tpass_direct='wfc3_ir_f105w_tpass_m01.dat' exptime_dir=100.0 bck_flux_dir=0.7
```

Directories and files after the simulation

```
example_4>ls -R
.:
CONF DATA OUTPUT OUTSIM SIMDATA template_images.lis template_spectra.lis
./CONF:
wfc3_abcscal_IRg141_0th_sens.fits wfc3_abcscal_IRg141_3rd_sens.fits
```

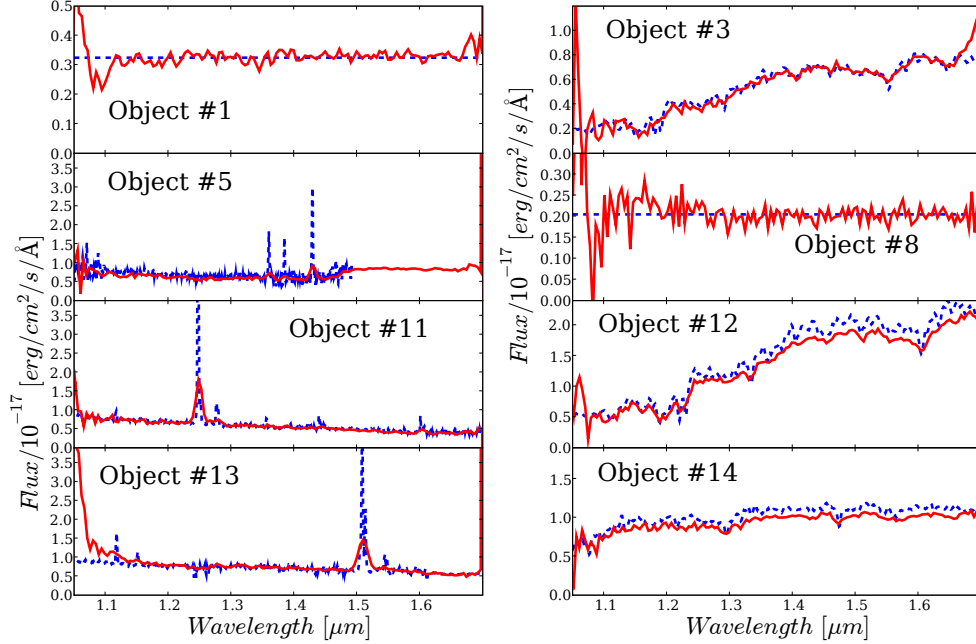


Figure 3.7: Comparison between the default extracted spectra (red, solid line) and the simulated spectrum (dashed, blue line) for selected objects.

```
wfc3_abecal_IRg141_1st_sens.fits WFC3_IR.G141.TV2_sim.conf
wfc3_abecal_IRg141_2nd_sens.fits WFC3_IR.G141.TV2_sim.conf.simul

./DATA:
example_4_MOT.dat ObjMix_images.fits ObjMix_spectra.fits

./OUTPUT:

./OUTSIM:
ObjMix_direct.fits ObjMix_slitless_2.SPC.fits ObjMix_slitless.fits
ObjMix_images.fits ObjMix_slitless_2.STP.fits ObjMix_spectra.fits

./SIMDATA:
Mann_E.dat WFC3_IR_1400nm_big_psf.fits wfc3_ir_f105w_tpass_m01.fits
Mann_S0.dat WFC3_IR_1400nm_psf.fits wfc3_ir_f110w_tpass_m01.dat
starb1_template.fits WFC3_IR_950nm_big_psf.fits WFC3_UVIS_350nm_psf.fits
starb2_template.fits WFC3_IR_950nm_psf.fits
starb3_template.fits wfc3_ir_f105w_tpass_m01.dat
```

The simulation results

Technically, this example only shows cases which had been already presented in the previous examples. The main issue here is to illustrate a mix of object definitions with various complexity.

The simplest object characterization using Gaussian shapes and a spectral energy distribution using at least one AB-magnitude value is **always** possible (see objects #1, #2, #7-10) and achieved by entering “0” in the columns

SPECTEMP and MODIMAGE. Beyond that it is possible to mix these simple objects with objects that have a template image (#6) with (e.g. #3, #5, #11) or without (#13) a high resolution template spectrum.

A simulation as shown here represents a typical mix of some primary targets together with many other, rather poorly defined surrounding objects.

In total, the results comprise:

- the direct image `ObjMix_direct.fits`
- the slitless image `ObjMix_slitless.fits`
- the Model Images `ObjMix_images.fits`
- the Model Spectra `ObjMix_spectra.fits`
- the default extracted 1D spectra `ObjMix_slitless_2.SPC.fits`
- the 2D grism stamp image file `ObjMix_slitless_2.STP.fits`

Some annotations

- The spectral templates can be given as fits tables and ASCII lists. Even a mix of these two formats as in this example is accepted.
- To see which extension in the Model Spectra `ObjMix_spectra.fits` corresponds to which simulated and extracted object, check the Model Object Table (here `example_2.MOT.dat`) **after** the simulations. The column MODSPEC (here #11) contains the corresponding extension number of the object in `ObjMix_spectra.fits`. When e.g. simulating object #3, the already redshifted and flux scaled extension 1 of `example_2.MOT.dat` was used as input spectrum. The panel for Object #5 in Fig. 3.7 was done plotting `ObjMix_slitless_2.SPC.fits[beam_5A]` over `ObjMix_spectra.fits[2]`. 0 in this column means **no** spectral template, hence a spectral energy distribution is used as spectral input.
- Fig. 3.7, object #5 is an example of the extrapolation (constant in f_λ) of a spectrum beyond the defined range shown with the blue, dotted line.
- For some objects in Fig. 3.7 the extracted flux seems to exceed the input spectra. As Fig. 3.6 shows, contamination (the mutual overlap of spectra) could be an issue here.
- Using the passband of an existing filter is natural if observational data in this filter already exists and the simulation is perhaps even based on an according SExtractor catalogue. Defining a boxcar passband as in Example 3.3 is more appropriate for purely synthetic data, or in cases where the spectral template does not completely cover the intended normalization passband (see box on page 25).

Chapter 4

The aXeSIM tasks

4.1 High Level Tasks – Low Level Tasks

Following the tradition of the aXe software package, aXeSIM contains two levels of tasks, the so called *Low Level Tasks* and the so called *High Level Tasks* (currently there exists only one of the latter, `simdata`).

While the Low Level Tasks perform one special step each, the High Level Tasks successively do a series of processing steps to generate complete simulated images as output from the basic input. The High Level Tasks use the Low Level Tasks for the individual steps.

The 'normal' user who simulates only one object will likely just use the High Level Task. For specific applications and large series of simulations, e.g. to determine limiting depths in deep fields, it might have advantages to use the Low Level Tasks saving time if successive steps need to be repeated.

4.2 High Level Tasks

4.2.1 SIMDATA

Description

This task generates the simulated data, which is the simulated dispersed image and, if requested, the simulated direct image. `simdata` comprises all the simulation possibilities available in aXeSIM, hence this is the only task most users will ever use.

Internally, `simdata` executes from several to all Low Level Tasks (depending on the parameters).

Usage

```
simdata incat config [...]
```

Parameters

- incat**, mandatory, [string]
The filename of the Model Object Table with the source list to be simulated. There is a minimal list of columns and additional columns which have to be present for certain direct image models.
- config**, mandatory [string]
The name of the aXe configuration file to be used in the simulations.
- output_root**, optional [string]
Root name for all output products. If not given, the root name is derived from the name of the Model Object Table
- silent**, optional [boolean, default='yes']
The silent mode reduces the output written to the screen.
- inlist_spec**, optional, [string]
The name of the Spectrum Template List. The template spectra therein can have ASCII or fits format.
- tpass_flux**, optional [string]
Total passband to be used for the scaling of the template spectra in the flux space. This is either an ASCII- or fits-file name, or an expression of the form ‘‘lambda_min, lambda_max’’ [nm]. In the latter case the passband has a boxcar shape with lambda_min and lambda_max as boundaries.
- inlist_ima**, optional, [string]
The Image Template List. The template images listed therein must have fits format.
- lambda_psf**, optional [float]
The wavelength where the object extent A_IMAGE and B_IMAGE was determined.
- nx_disp**, optional, [int]
The dimension of the simulated dispersed image in x
- ny_disp**, optional, [int]
The dimension of the simulated dispersed image in y
- exptime_disp**, optional [float]
The exposure time. If given, noise will be added to the simulated dispersed image.
- bck_flux_disp**, optional [float/string]

The background flux rate or a background image, both in [e/s], for the dispersed image.

extraction, optional [boolean, default='yes']

Flag to initiate a basic spectral extraction on the simulated dispersed image.

extrfwhm, optional [float, default=3.0]

Factor to set the extraction width.

orient, optional [boolean, default='yes']

Whether to use tilted extraction. This is the default. When set to "no" only vertical extraction (along columns) is performed.

slitless_geom [boolean, default='yes']

Whether to use an extraction orientation which is optimized for slitless spectroscopy.

adj_sens [boolean, default='yes']

Adjust sensitivity for extended objects.

tpass_direct, optional [string]

Total passband of the optical system to be simulated.

The can be in fits- or ASCII format. In case this parameter is not given, no direct image will be produced.

nx_dir, optional, [int]

The dimension of the simulated direct image in x

ny_dir, optional, [int]

The dimension of the simulated direct image in y

exptime_dir, optional [float]

The exposure time. If given, noise will be added to the simulated direct image.

bck_flux_dir, optional [float/string]

The background flux rate or a background image, both in [e/s], for the direct image.

version, optional [string]

Parameter to always have the version number accessible. No need to ever change/use or even touch it.

Input and Output

Input:

- `$AXE_IMAGE_PATH`: `incat`
- `$AXE_CONFIG_PATH`: `config`, `bck_flux_dir` and `bck_flux_disp` if image
- `$AXE_SIMDATA_PATH`: content of `inlist_spec`, content of `inlist_ima`, `tpass_direct`, `tpass_flux`
- `./`: `inlist_spec`, `inlist_spec`

Output:

- `$AXE_OUTSIM_PATH`: all output

Usage scenario for Low Level Tasks

A user measured in slitless data the equivalent width of a line. The user would like to know the error in the equivalent width measurement. One possibility would be to create an image with a larger number of simulated object spectra at the same brightness level and measure the equivalent width there. A convenient way to simulate a large number of the same object class is:

- Create a reduced Model Object Table with only the columns `NUMBER`, `MAG_?<number>*`, `Z` and `SPECTEMP` and fill in the objects (redshift, brightness and spectral template) you want to simulate
- Create a Model Spectrum Table using this reduced Table and the task `prepspectra`.
- Create a Model Image using the task `prepimages`
- Create a Model Object Table with positional information and specify in the columns `MODIMAGE` and `MODSPEC` which extension of the Model Image and Model Spectrum to use (see 3.4).
- Now simulate the slitless image using the task `simdispim`

4.3 Low Level Tasks

As already mentioned in previous chapters, few users will benefit from using the Low Level Tasks for their simulations. The box on page 36 describes a usage scenario based on Low Level Tasks. Most users will exclusively work with the High Level Task presented in Sect. 4.2.

4.3.1 PREPIMAGES

Description

This task generates the multi-extension fits image with normalized image templates to be used in `simdata`, `simdispim` or `simdirim`. The input is a list of template images in fits format. These templates are normalized to an integrated intensity of 1.0 and stored as a multi-extension fits image.

Usage

```
prepimages inlist [...]
```

Parameters

`inlist`, mandatory, [string]

The Image Template List. The template images listed therein must have fits format.

`model_images`, optional, [string]

Name of the multi-extension fits images containing the normalized template images.

Input an Output

Input:

- `$AXE_SIMDATA_PATH`: content of `inlist`
- `./`: `inlist`

Output:

- `$AXE_OUTSIM_PATH`: all output

4.3.2 PREPSPECTRA

Description

This task generates the multi-extension fits table with the spectra to be used by the tasks `simdata`, `simdispim` or `simdirim`. The task uses as input the template spectra (assumed restframe) in a two column ASCII- or fits-format plus a Model Object Table. From the Model Object Table, the task uses only the redshift and a magnitude column to shift the template spectra in redshift and flux. The output is a multi-dimension fits table. Every extension represents a template spectrum shifted to a certain redshift and scaled to some flux value specified in the corresponding row of the input table.

Usage

```
prepspectra inlist incat tpass_flux [...]
```

Parameters

- `inlist`, mandatory, [string]
 The name of the Spectrum Template List. The template spectra therein can have ASCII or fits format.
- `incat`, mandatory, [string]
 The Model Object Table with the magnitude values and redshift values to shift the template spectra in flux and redshift.
- `tpass_flux`, mandatory [string]
 Total passband to be used for the scaling of the template spectra in the flux space. This is either an ASCII- or fits-file name, or an expression of the form ‘‘lambda_min, lambda_max’’ [nm]. In the latter case the passband has a boxcar shape with lambda_min and lambda_max as boundaries.
- `model_spectra`, optional, [string]
 Name of the output multi-extension fits table to store the shifted template spectra.

Input and Output

Input:

- `$AXE_IMAGE_PATH`: `incat`
- `$AXE_SIMDATA_PATH`: content of `inlist`, `tpass_flux`
- `./`: `inlist`

Output:

- `$AXE_OUTSIM_PATH`: all output

4.3.3 SIMDIRIM**Description**

This task generates a simulated direct image. The objects to be simulated are given in a Model Object Table. The total passband (parameter `tpass_direct`) and the object spectra determine the simulated object flux in [e/s]. It is possible to add Poisson noise (from sources, background and readout). The final output image will be in units of [e/s].

Usage

`simdirim incat config [...]`

Parameters

incat, mandatory, [string]

The filename of the Model Object Table with the source list to be simulated.

config, mandatory [string]

The name of the aXe configuration file to be used in the simulations.

tpass_direct, mandatory [string]

Total passband of the optical system to be simulated.
This can be in fits- or ASCII format.

dirima_name, optional [string]

The name of the simulated image created by this task. If not given, a filename is derived from the name of the Model Object Table.

model_spectra, optional, [string]

The multi-extension fits table with the input spectra to be used in the simulations.

model_images, optional, [string]

The multi-extension fits image with the image templates to be used in the simulations.

nx, optional, [int]

The dimension of the simulated direct image in x

ny, optional, [int]

The dimension of the simulated direct image in y

exptime, optional [float]

The exposure time. If given, noise will be added to the simulated direct image.

bck_flux, optional [float]

The background flux rate or a background image, both in [e/s].

silent, optional [boolean, default='yes']

The silent mode reduces the output written to the screen.

Input an Output

Input:

- \$AXE_IMAGE_PATH: incat, model_spectra, model_images

- \$AXE_CONFIG_PATH: config, bck_flux if image
- \$AXE_SIMDATA_PATH: tpass_direct

Output:

- \$AXE_OUTSIM_PATH: all output

4.3.4 SIMDISPIM

Description

The task generates a simulated dispersed image. The objects to be simulated are given in a Model Object Table. The Model Object Table may contain references to extensions in a Model Spectra file and Model Image file to be used for specific objects. It is possible to add Poisson noise from sources, background and readout. The final output image will be in units of [e/s].

Usage

`simdispim incat config [...]`

Parameters

`incat`, mandatory, [string]

The filename of the Model Object Table with the source list to be simulated.

`config`, mandatory [string]

The name of the aXe configuration file to be used in the simulations.

`dispim_name`, optional [string]

The name of the simulated dispersed image created by this task. If not given, a filename is derived from the name of the Model Object Table.

`lambda_psf`, optional [float]

The wavelength where the object extent A_IMAGE and B_IMAGE was determined.

`model_spectra`, optional, [string]

The multi-extension fits table with the input spectra to be used in the simulations.

`model_images`, optional, [string]

The multi-extension fits image with the image templates to be used in the simulations.

`nx`, optional, [int]

The dimension of the simulated dispersed image in x

`ny`, optional, [int]
The dimension of the simulated dispersed image in y

`exptime`, optional [float]
The exposure time. If given, noise will be added to the simulated dispersed image.

`bck_flux`, optional [float]
The background flux rate or a background image, both in [e/s].

`extraction`, optional [boolean, default='yes']
Flag to initiate a basic spectral extraction on the simulated dispersed image.

`extrfwhm`, optional [float, default=3.0]
Factor to set the extraction width.

`orient`, optional [boolean, default='yes']
Whether to use tilted extraction. This is the default. When set to "no" only vertical extraction (along columns) is performed.

`slitless_geom` [boolean, default='yes']
Whether to use an extraction orientation which is optimized for slitless spectroscopy.

`adj_sens` [boolean, default='yes']
Adjust sensitivity for extended objects.

`silent`, optional [boolean, default='yes']
The silent mode reduces the output written to the screen.

Input and Output

Input:

- `$AXE_IMAGE_PATH`: `incat`, `model_spectra`, `model_images`
- `$AXE_CONFIG_PATH`: `config`, `bck_flux` if image

Output:

- `$AXE_OUTSIM_PATH`: all output

Chapter 5

File Formats

5.1 Image Template List

Used in:

- `simdata`, parameter: `inlist_ima`
- `prepimages`, parameter: `inlist`

An ASCII list loadable with the `AsciiData` module. Lines starting with `'#'` are not loaded. The list must contain at least one column, and the first column has to be of type `'string'`. NULL entries are not allowed in the first column. Data in possible further columns is not used. The entries must point to existing fits images which are located in the directory `$AXE_SIMDATA_PATH`.

5.2 Spectrum Template List

Used in:

- `simdata`, parameter: `inlist_spec`
- `prepspectra`, parameter: `inlist`

Similar to [5.1](#), but the entries must point to existing fits binary tables or ASCII tables in the directory `$AXE_SIMDATA_PATH`.

5.3 Total Passband File

Used in:

- `simdata`, parameter: `tput_direct`
- `prepspectra`, parameter: `tpass_flux`

A binary fits table with one extension or an ASCII table with two rows. The fits table must contain the two columns "wavelength" and "throughput". The column "wavelength" contains the wavelength positions in Angstrom in ascending order. The column "throughput" contains the system throughput for the wavelength (range [0.0, 1.0]). An ASCII table must contain two columns, the first is the wavelength in Angstrom and the second the total throughput. The total throughput is the fraction of photons that is recorded on the detector at a given wavelength (in [%]). All optical components (mirrors, filters, entrance windows) and the detector efficiency contribute as factors to the total throughput.

5.4 Template Spectrum

Files listed in the Spectrum Template List (`simdata`, parameter: `inlist_spec`) are all Template Spectra

An ASCII or binary fits table. The table contains exactly two columns with float values. The first column contains the ascending wavelength positions in [\AA]. The second column contains the associated flux values in [$\text{erg/s/cm}^2/\text{\AA}$]. NULL elements are not allowed.

5.5 Template Image

Files listed in the Image Template List (`simdata`, parameter: `inlist_ima`) are all Template Images

A binary fits image with at most two extensions. In case of two extensions the first extension must be empty and the second contain the data. In case of one extension only this extension must contain data.

5.6 Model Images

Used in:

- output of `simdata` and `prepimages`
- `simdispim`, parameter: `model_images`
- `simdirim`, parameter: `model_images`

A multi-dimension binary fits image. Each extension contains one image with the integrated intensity 1.0.

5.7 Model Spectra

Used in:

- output of `simdata` and `prepspectra`
- `simdispim`, parameter: `model_spectra`
- `simdirim`, parameter: `model_spectra`

A multi-dimension binary fits table. Each extension contains a table with the two columns "wav_nm" [*nm*] and "flux" [*erg/cm²/s/Å*].

5.8 Model Object Table

Used in:

- `simdata`, parameter: `incat`
- `simdispim`, parameter: `incat`
- `simdirim`, parameter: `incat`

An ASCII table loadable with AsciiData module with a table header in the SExtractor format . The table has the mandatory columns:

- NUMBER the object identifier; int
- X_IMAGE the x-position of the object [pix]; float
- Y_IMAGE the y-position of the object [pix]; float
- A_IMAGE the major axis rms [pix]; float
- B_IMAGE the minor axis rms [pix]; float
- THETA_IMAGE the position angle of the major axis [deg]; float
- MAG_?<number>* the AB-magnitude at the wavelength <number> [*nm*]; float

There might be several columns with AB-magnitudes. If the task SIMDATA is run with non-empty parameter "spec_models", the table must contain the columns:

- SPECTEMP which model spectrum to use; int
- Z redshift of object; float

If the task is run with the parameter "object_models", the table must contain the column:

- MODIMAGE which image template to use; int

NULL entries are not allowed in the mandatory columns. Any further columns and table entries are not used.

Chapter 6

Environmental variables

`aXeSIM` uses some of the 'traditional' aXe environmental variables and introduces two new, specific ones. The variables and their usage are:

- `AXE_SIMDATA_PATH`: the path where the template spectra, template images and total passband files are located.
- `AXE_OUTSIM_PATH`: all output of `aXeSIM` is written to this directory path
- `AXE_IMAGE_PATH`: the path where the model object table is located (as input to `simdispim`, `simdirim` and `simdata`).
- `AXE_CONFIG_PATH`: the path where the aXe configuration file and the sensitivity file(s) are located
- `AXE_OUTPUT_PATH`: the directory path to which all embedded aXe tasks write their output to. This is only intermediate data and can be deleted once the corresponding simulation has finished.

Bibliography

- [1] Kümmel, M., Walsh, J.R.; Pirzkal, N., Kuntschner, H. & Pasquali, A.J., 2009, PASP **121**
- [2] The aXe extraction software: http://www.stecf.org/software/slitless_software/axe
- [3] aXe configuration and calibration files:
<http://www.stecf.org/instruments/ACSgrism/calibration>,
<http://www.stecf.org/instruments/WFC3grism>
- [4] The aXe User Manual: <http://www.stecf.org/instruments/ACSgrism/axe/manual/html/index.html>
- [5] Kuemmel, M., Larsen , S.S. & Walsh, J.R., 2005, New Developments in
aXe, ST-ECF Newsletter 38, 8
(http://www.spacetelescope.org/about/further_information/newsletters/html/newsletter_38.html)
- [6] The stsci_python package: http://www.stsci.edu/resources/software_hardware/pyraf/stsci_python
- [7] Thompson, R.I., Illingworth, G., Bouwens, R., et al. 2005, ApJ **130**, 1

Index

- ACS, [7](#), [10](#)
- AXE_CONFIG_PATH, [47](#)
- AXE_IMAGE_PATH, [47](#)
- AXE_OUTPUT_PATH, [47](#)
- AXE_OUTSIM_PATH, [47](#)
- AXE_SIMDATA_PATH, [47](#)
- aXeSIM, [8](#)

- background, [10](#), [27](#)
- blooming, [10](#)

- calibration file, [8](#)
- CFITSIO, [15](#)
- compilation, [13](#)
- configuration file, [8](#), [19](#)

- dark current, [10](#)
- direct image, [7](#), [9](#), [19](#)

- environmental variables, [47](#)
- equivalent width, [36](#)
- ETC, [8](#)

- flatfield, [10](#)
- flux normalization, [25](#)
- format, [9](#), [45](#)

- GSL, [15](#)

- High Level Task, [33](#)
- HST, [7](#)

- image template list, [43](#)
- installation, [13](#)
- IRAF, [8](#)

- Linux, [14](#)
- login file, [14](#)

- Low Level Task, [36](#)

- model images, [44](#)
- Model Object Table, [8](#), [45](#)
- model spectra, [44](#)

- NICMOS, [10](#)
- NIMCOS, [7](#)
- noise, [10](#)

- online parameter, [15](#)

- preamble, [5](#)
- PREPIMAGES, [37](#)
- PREPSPECTRA, [37](#)
- PyRAF, [8](#), [13](#), [27](#)
- python, [13](#)

- saturation, [10](#)
- sensitivity file, [8](#)
- sensitivity variations, [10](#)
- SExtractor, [9](#), [45](#)
- SExtractor format, [45](#)
- SIMDATA, [33](#)
- SIMDIRIM, [38](#)
- SIMDISPIM, [40](#)
- spectral extrapolation, [19](#)
- spectral interpolation, [19](#)
- spectrum template list, [43](#)
- ST-ECF, [8](#)
- STScI, [8](#)
- STSDAS, [13](#)
- support, [5](#), [15](#)

- template image, [44](#)
- template spectrum, [44](#)
- total passband, [43](#)

WCS, [15](#)

WFC3, [7](#), [10](#)