HSIM Manual

v210

14 January, 2019

Contents

Revision History

Version	Date	Comments
1.0	2018-09-27	HSIM v204 release
1.1	—	HSIM v210 release

Authors of this document

Miguel Pereira Santaella (University of Oxford)

1. Quick Start Guide

1.1 Introduction and Installation

HSIM is a dedicated pipeline for simulating observations with HARMONI on the Extremely Large Telescope. HSIM takes high spectral and spatial resolution input data cubes, encoding physical descriptions of astrophysical sources, and generates mock observed data cubes. The simulations incorporate detailed models of the sky, telescope, instrument, and detectors to produce realistic mock data (Zieleniewski et al., 2015).

HSIM is programmed in Python and the source code can be found at `https://github.com/HARMONI-ELT/HSIM`. It does not require any installation, just download it from .... and unzip. HSIM depends on the following Python packages to work:

- astropy 2.0.4
- numpy 1.14.1
- scipy 1.0.0
- matplotlib 2.1.2
- wxPython > 3.0

The code has been tested with the indicated package version, although more recent releases of these packages are likely to work as well.

1.2 Running Simulations

1.2.1 Preparing the Input Datacube

Before running HSIM, you will need an input model of your astronomical object stored as a datacube in a FITS file. The recommended sizes for the spatial pixels (spaxels) are 1, 2, or 3 mas depending on the selected spaxel scale for HARMONI (see Table 1.1). If your datacube has a different spaxel scale, HSIM will automatically interpolate/rebin the input datacube to the recommended values.

Similarly, it is recommended to oversample the spectral dimension of the input cube by a factor of 4 with respect to the nominal resolving power (R) of the selected grating (e.g., the spectral sampling for the K-grating R=7100 at 2.2μm should be 0.078 nm). If your input cube has a different spectral sampling, HSIM will interpolate/rebin it. If possible, it is recommend to use input cubes with a spectral resolution a factor of 2 better than the R of the selected grating.

The information on the spatial and spectral sampling of the input cube are passed through the FITS file header to HSIM (Table 1.2 shows a summary of all the header keywords used by HSIM). In particular, the `CDELT1` and `CDELT2` header values are used to

Table 1.1: Recommended input pixel size

HARMONI Spaxel scale (mas)	Recommended input scale (mas)
4×4	1
10×10	2
20×20	3
30×60	3

get the spatial scale, and `CDELT3` to obtain the spectral sampling. The spectral resolution of the input cube is indicated by `SPECRES` in wavelength units similar to `CDELT3`. The units of these values are those indicated by the `CUNIT1`, `CUNIT2`, and `CUNIT3` header values. Currently, for the spatial units (`CUNIT1` and `CUNIT2`), the accepted values are `mas` and `arcsec`. For the spectral units, `CUNIT3` and `SPECRES`, HSIM recognizes the following values: `angstrom`, `nanometers`, `nm`, `micron`, and `meter`. The size of the input cube is obtained from `NAXIS1`, `NAXIS2`, and `NAXIS3`.

The wavelength of each slice of the data cube are also calculated from the FITS header using the following relation:

$$\lambda_i = \texttt{CRVAL3} + \texttt{CDELT3} \times (i - \texttt{CRPIX3}) \tag{1.1}$$

where i is the slice number from 1 to `NAXIS3`. The wavelength range of the input cube and the selected grating should, at least, partially overlap.

Finally, the input data cube should be in surface brightness units indicated by `BUNIT`. The accepted units are the following: `erg/s/cm2/um/arcsec2`, `erg/s/cm2/A/arcsec2`, `J/s/m2/um/arcsec2`, and `J/s/m2/A/arcsec2`.

Table 1.2: FITS header keywords used by HSIM

Keyword	Description	Accepted values
	Spatial information	
NAXIS{1,2}	Number of pixels along x-axis and y-axis	
CTYPE1	Type of the spatial x-axis	x, ra
CTYPE2	Type of the spatial y-axis	y, dec
CDELT{1,2}	X and Y pixel size	
CUNIT{1,2}	Units of CDELT{1,2}	mas, arcsec
	Spectral information	
NAXIS3	Number of spectral pixels	
CTYPE3	Type of the spectral axis	wavelength
CDELT3	Size of the spectral pixel	
SPECRES	Spectral resolution in wavelength units	
CRPIX3	Reference pixel for the spectral axis (see Equation 1.1)	
CRVAL3	Wavelength of the reference pixel	
CUNIT3	Units of CDELT3, CRVAL3, and SPECRES	angstrom, nanometers, nm, micron, meter
	Flux units	
BUNIT	Flux units of the cube	erg/s/cm2/um/arcsec2, erg/s/cm2/A/arcsec2, J/s/m2/um/arcsec2, J/s/m2/A/arcsec2

1.2.2 Running HSIM

Once the input data cube has been prepared, the next step is to run HSIM. HSIM comes with a graphical user interface (GUI) which can be used to define the input parameters of the simulations. Alternatively, the same options can be directly accessed from the command line.

GUI

The GUI provides a relatively easy way to list the available options. To launch HSIM in the GUI mode, type the following command

```
$ python hsim2.py
```

Then, a window similar to that shown in Figure 1.1 should appear. In that window, it is possible to select the input cube and the output directory and define all the parameters needed by HSIM to run the simulation (these parameters are explained in Section 1.2.3). Clicking on the `Commence simulation` button will start the simulation. You can see how the simulation advances in the terminal window from where HSIM was called.

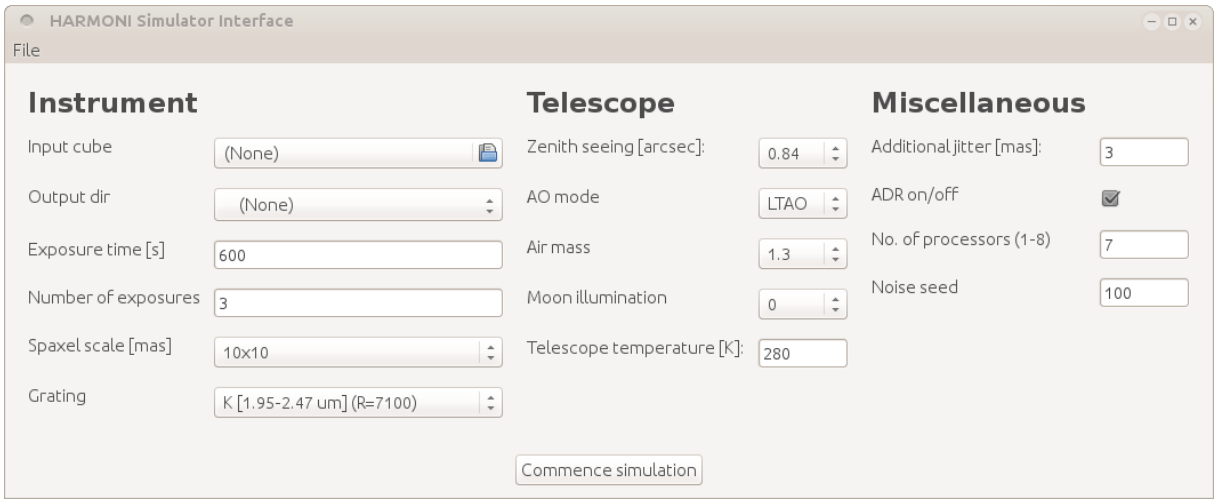

Figure 1.1: HSIM GUI.

Command-line

All the GUI options can be directly accessed from the command line mode using the following command

```
$ python hsim2.py -c arg1 arg2 ...
```

where `arg*` are the parameters of the simulation (see Section 1.2.3). All the parameters are needed to run simulation. After the `-c`, you can add `-p number`, where `number` is the number of processors that will be used by HSIM. You can also add `-o directory` where `directory` is the output directory. If no output directory is specified, the default `output_cubes` will be used and results from previous simulations might be overwritten. Finally, using `-h`, will show a short help message.

1.2.3 List of Options

The available HSIM options are listed below. The number before them indicates the parameter order when used in the command line mode.

1. Input cube: Path of the input datacube (e.g., `datacube.fits`).
2. Exposure time: Integration time of each exposure in seconds (e.g., `600`).
3. Number of exposures (e.g., `3`).
4. Grating: Indicates the HARMONI grating. The available options are the following: `V+R, Iz+J, H+K, Iz, J, H, K, z-high, J-high, H-high, K-short, K-long`.
5. Spaxel Scale: HARMONI spatial scale in mas. The available values are the following: `4x4, 10x10, 20x20, 30+x60`.
6. Zenith seeing: Atmospheric seeing FWHM in arcsec. The available values are the following: `0.43, 0.57, 0.64, 0.72, 1.04`.
7. Air mass: Average air mass during the HARMONI observation. The available values are the following: `1.1, 1.3, 1.5, 2.0`, which correspond to zenith angles of 25°, 40°, 48°, and 60°, respectively.
8. Moon illumination: Fraction of the Moon that is illuminated at the time of the observation. A 30° separation between target and Moon is assumed. The available values are the following: `0, 0.5, 1.0`.
9. Jitter: Additional telescope PSF blur in mas. Typical values are `2` and `3` for SCAO and LTAO, respectively.
10. Telescope temperature: Temperature of the site/telescope in K. This is used to calculate the telescope and part of the instrument background emission. Typical value is `280`.
11. ADR on/off: Indicates if the atmospheric differential refraction is simulated. When simulated, the differential refraction is corrected in the output cubes in a way similar to what is expected from the pipieline. Available values are `True, False`.
12. Noise seed: Seed used for the random number generator during the simulation. It only affects the observed outputs (see Section 1.2.4).
13. AO mode: Adaptive optics mode of the observations. The available values are the following: `LTAO, SCAO, noAO`. `LTAO` corresponds to the Laser Tomography Adaptive Optics mode, `SCAO` to the Single Conjugate Adaptive Optics mode and `noAO` indicates that no adaptive optics corrections are applied.

1.2.4 Output Files

HSIM stores the results of the simulation in the specified output directory (the default output directory is `output_cubes`). The name of the output files begins with the name of the input data cube and it is followed by a suffix indicating the content of the file. The input parameters of the simulation are stored in the header of the FITS files and also in the log file.

1. `_noiseless_obj_plus_back.fits`. Data cube containing the expected number of electrons per pixel detected by HARMONI due to the target and the background (sky, telescope, and instrument) emissions. **Cross-talk is not applied to the noiseless outputs. Therefore, their LSF is slightly narrower than that of the observed and reduced outputs.**

2. _noiseless_obj.fits. Expected number of electrons per pixel solely due to the target.
3. _noiseless_back.fits. Expected number of electrons per pixel from the background sources (sky, telescope, and instrument).
4. _observed_obj_plus_back.fits. Simulated observed data cube taking into account the photon noise, read noise, dark current, and cross-talk.
5. _observed_back.fits. Simulated observed sky background taking into account the photon noise, read noise, dark current, and cross-talk.
6. _reduced.fits. Simulated reduced data cube created by subtracting a simulated sky observation to the _observed_obj_plus_back.fits data cube.
7. _reduced_SNR.fits. Signal to noise ratio of the reduced data cube per pixel.
8. _std.fits. Noise standard deviation. Includes object and sky noise (Poisson) and dark current and read noise (Gaussian).
9. _PSF.fits. Point spread function at the mean wavelength of the input data cube. The PSF is oversampled by a factor of 4 to 10 with respect to the other output FITS data cubes.
10. _total_em.pdf. Plot of the background emission broken up into the individual modeled components. The average HARMONI to telescope+sky background ratio is indicated.
11. _total_tr.pdf. Plot of system transmission broken up into the individual modeled components.
12. _total_tr.txt. Total transmission stored as a text file.
13. .log. Detailed log of the simulation. The last line of this file shows if any problem was found during the simulation.

Bibliography

Zieleniewski S., Thatte N., Kendrew S., Houghton R. C. W., Swinbank A. M., Tecza M., Clarke F., Fusco T., 2015, MNRAS, 453, 3754