

MIRISim



MIRISim: The Simulator for the MIRI Instrument

Pamela Klaassen¹
on behalf of the MIRISim team:
 Vincent Geers¹, Steven Beard¹, Alan O'Brien¹, Martin Topinka²,
 Rene Gastaud³, Alain Coulais³, Juergen Schreiber⁴ & Patrick Kavanagh²

¹UK ATC, ²DIAS, ³CEA, ⁴MPIA



ESAC 2016 JWST workshop

MIRISim was designed to simulate data from the MIRI instrument on the JWST. Instrumental effects from photon conversion efficiencies to diffractions through the spectrometers to read/shot noise (etc) of the sensor chip assembly are reproduced. This simulator gives the user a sense of what the instrument is capable of and returns data with the same format as the uncalibrated data coming from the telescope.

The user specifies a target scene and the required observing setup (e.g. imaging/spectroscopy, filter/sub-band, number of integrations, dithers, etc). The end result is a set of detector images representative of the outputs of the IFU or imager (including LRS spectrometer). These detector images are then ready for processing by the JWST pipeline.

MIRISim: The Simulator for the MIRI Instrument



- **Brief description of MIRISim's:**
 - Internals
 - Usage
 - Products
- **Overview of Data Pack**
 - What was simulated
 - Which files are which



ESAC 2016 JWST workshop

MIRISim is a simulator package designed to simulate how photons propagate through, and are processed by the MIRI instrument. It has been written in python and delivers detector images which are consistent with the expected on orbit performance of the instrument. The outputs of MIRISim have the same data structure as observations coming from the telescope, are consistent with the MIRI sensitivity model, and are delivered in FITS file format. All relevant meta data are included to be able to ingest the outputs of MIRISim into the JWST calibration pipeline.

The contents of the data pack are described in greater detail in the README associated with the data.

Structure



Intent:

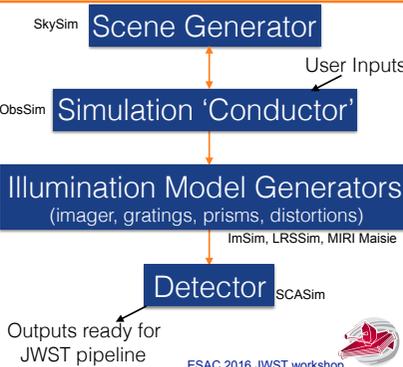
- simple to use
- robust

Usage:

- observation preparation
- pipeline testing
- familiarisation with MIRI data

User interface:

- command line based
- scriptable
- python based



```

graph TD
    UserInputs[User Inputs] --> SceneGenerator[Scene Generator]
    SceneGenerator -- SkySim --> SimulationConductor[Simulation 'Conductor']
    SimulationConductor -- ObsSim --> IMG[Imager, gratings, prisms, distortions]
    IMG -- ImSim, LRSSim, MIRI Maisie --> Detector[Detector]
    Detector -- SCASim --> Outputs[Outputs ready for JWST pipeline]
  
```



ESAC 2016 JWST workshop

Effort has been made to make MIRISim user friendly. A working knowledge of python is therefore not required to use MIRISim, and to avoid having to learn a number of different file formats, the required input text files for MIRISim are designed to have the same formatting/structure as those to be used for the JWST Calibration pipeline.

For MIRISim, we have followed the JWST nomenclature when defining terms used for variables for ease of use.

Full MIRISim simulations are intended for those who want to comprehensively simulate data coming from the MIRI instrument on the JWST. This includes:

- observers planning observations
- the instrument team helping to refine the data reduction pipeline
- scientists understanding the data products
- theorists who want to draw comparisons between their numerical simulations and what an observer would detect on the sky
- ..and many more.

Usage

Two primary user input files:

- setup of the observing 'scene' (incl. FITS ingest)
- setup of the instrument

Two (Three) output files:

- (sky cube - MRS only)
- illumination model
- detector image (Level 1B)
- Public release imminent

```

# Default simulation.in file generated by MIRISim
name = "Default Simulation"

[Scene]
Filename = scene.in # name of scene file to be used to generate sky cube

[Observation]
rel_obsdate = 0.0 # relative observation date (0 = launch, 1 = end of 5 yr

[Pointing_and_Optical_Path]

[[Primary_Optical_Path]]
pop = MRS # Component on which to centre, chose from MRS, IMA
ConfigPath = MRS_SHORT # Configure the opticalpath (MRS sub-band or Imager mode

[Integration_and_patterns]

[[Dither_Patterns]]
Dither = True # Include Dithering (True/False)
StartInd = 1 # Index of first position in dither pattern (lowest pos
N_Dither = 2 # Number of Dither Positions
DitherPat = mrs_recommended_dither.dat # Name of input dither pattern file
    
```

MIRISim does not produce final (calibrated) datasets, but has the same outputs as MIRI itself

There are two required input files for MIRISim: one defining an astronomical scene, the other defining the instrument configuration.

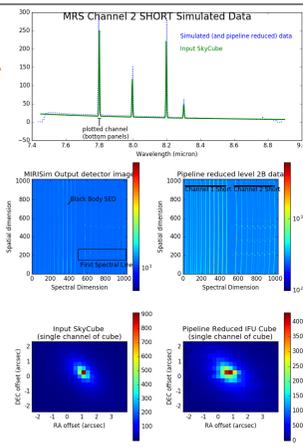
The first, the definition of an astronomical scene, points to an input FITS file, or can be used with the internal scene generator (called SkySim) to place point and/or extended objects throughout the field of view of the detector. These objects can be assigned SEDs, narrow and or broad spectral lines and/or can call pre-defined libraries of SEDs. Additionally, these spectral features can be modified by line-of-sight velocity distributions as defined by the user. Details of how to setup such scenes are provided in the user documentation provided with MIRISim.

The screen capture here shows part of the (MRS Default) simulator configuration file used as input to MIRISim. The formatting is the same as that for JWST pipeline input files, with nested (and indented) square brackets indicating different levels of substructure. In this example, the top level groupings are 'Scene', 'Observation', 'Pointing_and_Optical_Path', 'Integration_and_patterns', the last of which is shown to have two groupings of sub-structure shown. Note that because this configuration is for the MRS, imager configurations (those appearing in [[IMG_configuration]]) are ignored by MIRISim.

Data Pack

- Shown here are example images of what has been provided in the data pack on the conference USB stick

- MIRISim_input_skycube.fits
 - a FITS representation of what MIRISim is using as its input sky scene to observe (fluxes in microJy, wavelengths in microns)
 - in this case: a single extended blackbody emitter (10⁴ K) with four spectral lines in Channel 2Short (7.5-8.9 micron)



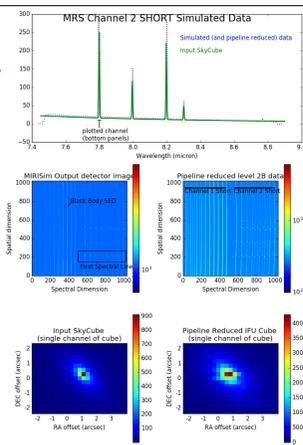
The input astronomical scene consisted of an extended 'disk' near the pointing centre with a radius of 1 arcsecond, an axial ratio of 0.5 and a position angle of 35deg. It has a 10⁴K blackbody SED, and four spectral lines were inserted at 7.8, 8.0, 8.2 and 8.3 microns (with varying intensities)

The outputs of MIRISim depend on which optical path (Imager/LRS/MRS) is being simulated. By default, two types of files are created: illumination models, and detector images. The illumination models show the illumination that reaches the detector, while the detector images are the final output which have detector effects and noises incorporated. It is this second type of file, the detector image, that is analogous to the outputs from MIRI. In the case of MRS observations, a third type of FITS file is output, a FITS representation of the input sky model (known as a sky cube). For each exposure and dither position, individual illumination models and detector images are created and placed in a sub-directory of the 'current working directory'.

Data Pack

- MIRISim_output_det_image.fits
 - a FITS file with the same formatting (and headers) as those to be produced by MIRI itself

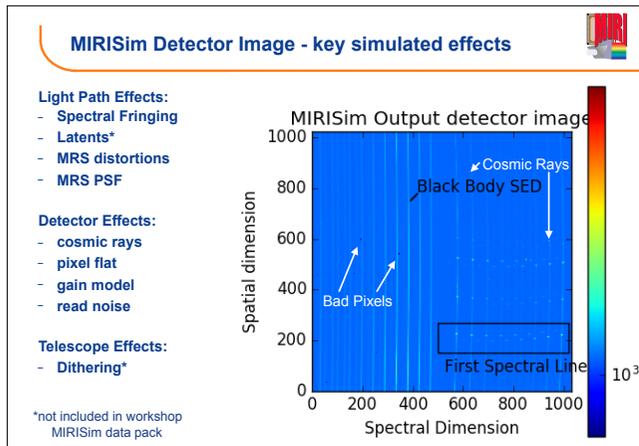
- MIRISim_jwst_pipeline_level2B.fits
 - JWST pipeline* processed data at the last step before cube building (for comparison to MIRISim_output_det_image.fits)
- MIRISim_jwst_pipeline_IFUCube.fits
 - JWST pipeline* processed final output cube (for comparison to MIRISim_input_skycube.fits), in (alpha,beta) coordinates



Note that MIRISim_output_det_image.fits is the only file in the data pack that is the end result of a MIRISim Simulation. The others are given for reference (the 'input' sky cube, and the pipeline products) in order to help the reader understand the information in the detector image. The bottom two panels are single channels from 3D (RA,DEC,Wavelength) FITS cubes of the MIRISim SkyCube (left) and final Pipeline reduced IFU Cube (right)

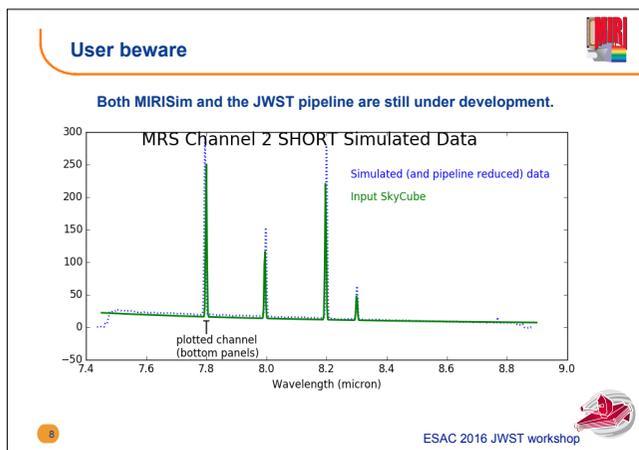
The MIRISim_output_det_image.fits file (a single frame of which is shown here in the centre left image) was produced using MIRISim to simulate MRS observations of Channel 2 short (7.5-8.9 micron) in a single exposure with three integrations and five frames per integration. Although SLOW read mode (27 s) is generally preferred for MRS observations, this simulation was run in FAST read mode (2.7 s).

The centre right and bottom right images here show JWST pipeline outputs at Level 2B (calibrated and co-added but not cube built) and the final IFU Cube. There are some slight differences between the two bottom data cubes. There are a number of reasons for this, including slightly different sampling of the data through MIRISim and the pipeline. Because this step of the pipeline is still under development, flux scaling as yet to be incorporated, and the intensity values given in the cube are count



To the left is an incomplete but generalised listing of the key distortions, effects and noise sources applied in MIRISim. Not all are applicable to the provided data pack (e.g. dithering and latent images are not possible to simulate in a single exposure), but are included in the simulator. A number of the detector effects are highlighted in the picture (right) in white, while the emission from the astronomical targets (a single point source with a blackbody SED and four spectra lines) are highlighted in black.

Most of the distortions, transmission efficiencies, PSFs (etc) applied to the input data are taken from the MIRI Calibration Data Products (CDPs) delivered by the MIRI team to STScI. Eventually these effects will be applied using the finalised STScI Calibration Reference Data System (CRDS), however, while the CRDS is still under development, the CDPs (from which the CRDS is drawing) are being used.



The JWST pipeline under development at STScI is still under active development. As such, some of the data reduction steps may not be in their final form. These results are still subject to change, and are given as representative for the time being.

Similarly, MIRISim is not yet ready for public release. It is undergoing testing and refinement within the MIRI team, with a first public release expected soon. Subsequent releases will then build on the first release set of simulated effects.

MIRISim: The Simulator for the MIRI Instrument

- **Brief description of MIRISim's:**
 - Internals
 - Usage
 - Products
- **Overview of Data Pack**
 - What was simulated
 - Which files are which

MIRISim isn't publicly available yet.

It's currently undergoing final testing, so watch this space.

ESAC 2016 JWST workshop




ESAC 2016 JWST workshop

```

tau_telescope = 0.00 # Overall telescopic transmission efficiency at sunspot
tau_eol = 0.8 # Reduction in overall telescope/MIRI transmission efficiency

[[Telescope]]
telescope_area = 25.032 # Telescope collecting area (m^2)
telescope_pupil_diam = 6.6852 # Telescope pupil diameter (m)

[SkySin]
# Default simulator.ini file generator
# It is unlikely that variables in this file need to be changed

[MIRISin]
[[Dither]]
max_fsm = 0.050
max_dither = 20.0
mrs_ref_channel = 1
mrs_ref_band = SHORT

[[Efficiencies]]
tau_telescope = 0.88
tau_eol = 0.8

[[Telescope]]
telescope_area = 25.032
telescope_pupil_diam = 6.6852

[SpecSin]

[ImSin]

[SCASin]
include_prefix = T # Include reference pixels in data.
include_poisson = T # Apply poisson noise.
include_readnoise = T # Apply read noise noise.
include_badpix = T # Apply bad pixels.
include_dark = T # Apply dark current / hot pixels.
include_flat = T # Apply flat field.
include_gain = T # Apply gain.
include_nonlinearity = T # Apply non-linearity.
include_drifts = T # Include detector drifts.
include_latency = T # Include detector latency.

[CDP]
tau_distortion_version = 06.03.00 # CDP version for IMA Distortion file
lrs_distortion_version = 06.02.00 # CDP version for ILS Distortion file
mrs_distortion_version = 06.04.00 # CDP version for MRS Distortion files
tau_pixmap_version = 06.00.00 # CDP version for IMA Pixel Area files
tau_psf_version = 05.02.00 # CDP version for IMA PSF files
tau_psf_F2550MR_version = 03.01.00 # CDP version for IMA F2550MR Filter
mrs_psf_version = 05.02.00 # CDP version for MRS PSF files

[SCASin]
include_prefix = T # Include reference pixels in data.
include_poisson = T # Apply poisson noise.
include_readnoise = T # Apply read noise noise.
include_badpix = T # Apply bad pixels.
include_dark = T # Apply dark current / hot pixels.
include_flat = T # Apply flat field.
include_gain = T # Apply gain.

```