JWST Ramp Simulator

Hilbert,Canipe

March 2, 2017

Abstract

Handbook for ramp simulator This document serves to explain the use and functionality of the JWST ramp simulator software.

1 Introduction

The goal of the JWST ramp simulator software is to create high fidelity simulations of multiple non-destructive readout (MULTIACCUM) observations from JWST instruments. The simulated data are in a state which mimics JWST observations prior to processing through the level 2A DMS pipeline. For convenience, we refer to this state as "raw". This means that no calibrations (e.g reference pixel subtraction, dark current subtraction) have been applied to the data.

The output data files are in the proper format such that they can be run through the DMS pipeline. Currently, each simulated observation contains a single integration. An integration is composed of multiple "groups", where a group can be an image from a single non-destructive detector frame, or the average of multiple non-destructive frames.

2 Basic Flow

The basic method used by the script to create the simulated data is to create simulated images of astronomical sources, and to add these images to a real dark current integration. By basing the simulation on real dark current data, we are guaranteed to include many detector-specific effects, such as 1/f noise, hot pixels, dead pixels, and read noise, in the simulated data. The dark current image is assumed to be a raw integration; that is, an observation to which no calibrations have been applied. The simulated astronomical sources are also created as if they have not had calibrations applied.

The script begins by creating a countrate image of all of the astronomical sources to be simulated. These sources are gathered from input lists provided by the user. For point sources, this list includes the locations of the sources in RA and Dec (or x,y pixel locations on the detector) as well as the magnitudes.

For each listed point source, an appropriate PSF stamp image is selected from a PSF library. The program was designed around a PSF library generated by the WebbPSF utility (REFERENCE). The PSF library contains PSF images in all filters for sources centered at various sub-pixel locations. There are also files created using several values of wavefront error. EXPLAIN MORE HERE...

Once the appropriate PSF file is chosen for a given point source in the input list, the PSF is read in and scaled to match the requested magnitude of the source. It is then placed onto the output image in the requested location. If the location is provided as an x,y pixel location on the detector, the PSF is placed there. The script then loads from the input astrometric file a model capable of transforming the source's x,y location to RA and Dec. If the point source's location is given in RA and Dec, then the inverse transform is used to calculate the corresponding x,y location on the detector, and the source is placed there. In both cases, the final x,y pixel value, as well as associated RA and Dec, and the calculated count rate in electrons per second as well as electrons per readout frame are saved in a text file called "pointsources.list".

The code can also simulate small galaxies and add these to the countrate image. Similar to the input point source list, a list of galaxies to add can be provided. For galaxies, the input list includes the location and half-light radius, ellipticity, position angle, sersic index, and manitude for each source. Just as with the point sources, the locations of the galaxies can be given in detector x,y coordinates, or as an RA,Dec list. Once the location of a simulated galaxy is known, a stamp image is created using numpy's Sersec2D model. The stamp image is scaled to the requested magnitude, and placed into the output countrate image.

Another possible input is an "extended" image. In this case, the extended image is read in from a fits file and assumed to be in units of electrons per second. This image is then scaled by the appropriate exposure times and added to the integration.

Similar to these three types of target input lists, the simulator also accepts three target input lists for simulating moving targets. These are targets which move through the field of view during the course of the integration. Examples include asteroids, Kuiper belt objects, and other solar system objects.

Another accepted input is a non-sidereal target which JWST will then "track" when creating the simulated observation. In practice, this means that all targets other than the non-sidereal target, such as those listed in the point source and galaxy input files, will then trail through the field of view during the observation. This mode is meant to simulate observations of solar system targets with non-sidereal velocities.

Finally, the simulator can also create wide field slitless spectroscopy (WFSS) data. These integrations are dispersed, as if created using a grism.

3 Installation

Currently, the ramp simulator is available via github. The ramp simulator excluding the module for creating dispersed data can be downloaded from:

If you do not plan to simulated dispersed data, the simulator will produce simulated imaging integrations using only the code in this repository.

To obtain the files necessary for the creation of dispersed data, the following repositories are also needed:

4 Input Parameter File

The sole input to the simulator call is a yaml file (REFERENCE) which contains a list of parameters and input files to be used for the data creation. Here, we list the fields present in the input file and describe in more detail the parameters which dictate the contents of the simulated data. An example of an input file is shown in Appendix A.

For entries where an input filename is expected, placing 'None' will cause that step to be skipped. For example, to simulate a integration containing only stars, the input files for all other targets can be set to None.

5 Instrument Information

instrument:

Name of the JWST instrument to be simulated. The simulator was developed using NIRCam as an example, and hence works best for NIRCam. Other instruments' imaging and WFSS modes should also be easily simulated.

mode: e.g. imaging or moving_target. Use moving_target when you want to simulate non-sidereal tracking.

nresetlines

6 Readout Details

readout pattern:

This is the name of the timing pattern used when reading out the detector. For example, NIRCam data can be collected using the RAPID, BRIGHT1, BRIGHT2, DEEP8 readout patterns, among others. The list of possible readout patterns and their definitions is provided by an ascii file specified in the readpattdefs parameter in the Reffiles section of the input file.

nframe:

This parameter gives the number of non-destructive detector readouts which are averaged together to produce each group in the final output data integration. If the requested readout pattern is present in the readpattdefs definition file, then the corresponding nframe value in the definition will override the value manually entered by the user. If the requested readout pattern is not present in the readpattdefs file, then the user-entered nframe value will be used.

nskip:

This parameter gives the number of non-destructive detector readouts which are ignored between groups of the output integration. If the requested readout pattern is present in the readpattdefs definition file, then the corresponding nskip value in the definition will override the value manually entered by the user. If the requested readout pattern is not present in the readpattdefs file, then the user-entered nskip value will be used.

ngroup:

This parameter lists the number of groups to create for the output integration. The maximum number of groups for a given readout pattern is controlled by the readpattdefs file. If the user enters a value greater than the allowed maximum for a given readout pattern, then ngroup is automatically reset to the maximum value.

namp:

The parameter lists the number of amplifiers used when reading out the detector in the requested configuration. This parameter is also controlled by the definitions in the readpattdefs file unless an unrecognized readout pattern is requested. This parameter is used only to calculate the detector readout time. For NIRCam, 4 amplifiers are used for all full frame readouts, while the majority of subarrays are read out through a single amplifier.

array_name:

This is the name of the aperture used for the simulated data. Generally, this is composed of the name of the detector combined with the name of the subarray used. For example, a full frame observation using NIRCam's A1 detector has an array_name of 'NRCA1_FULL'. The list of possible array_name values are given in the subarray_defs input file described below.

subarray bounds:

This input is a list of 4 comma-separated coordinates which describe the lower left and upper right corners of the subarray used for the simulation. In general, these values are overridden by the values in the subarray_defs input file. The only time that the user-specified values are used is when the array_name does not match any of the entries in the subarray_defs file.

filter:

Name of the filter to use for the simulated data. (e.g. F090W). The filter is used when scaling astronomical sources from the requested brightness in magnitudes to counts on the detector. For NIRCam, the filter name is also used to determine whether the simulated data are to be produced using a shortwave or longwave detector, and to select the appropriate pixel scale. (SEEMS LIKE THIS SHOULD CHANGE. SW/LW and PIXSCALE SHOULD COME FROM THE DETECTOR NAME)

pupil:

Name of the pupil wheel element to use for the simulated data. Currently this information is not used in the actual simulation, and is only saved to the output file.

7 Reference Files

dark:

The dark current integration file that will be used as the basis for adding many of the instrumental effects to the simulated astronomical sources. The dark current integration must have a readout pattern of either RAPID or the same value as the integration to be simulated. RAPID data keep every readout of the detector without any averaging. From this, any other readout pattern can be simulated by averaging and ignoring the appropriate frames. Other readout patterns cannot be translated in this way as their data are already averaged or missing some frames. However, if simulating, for example, a BRIGHT2 integration, then the input dark current integration can be a BRIGHT2 integration, as no translation is necessary in this case.

If a translation between RAPID and another readout pattern is necessary, then frames will be averaged/ignored as necessary. If the input dark current integration does not contain enough frames to be translated into the requested number of output groups, then the script creates enough additional dark current frames to make up the difference. These additional frames are created by making a copy of an appropriate number of existing initial dark current frames, and adding their signals to that in the final dark current frame. Note that this can lead to apparent double cosmic rays in pixels where a cosmic ray appeared in the dark current integration.

hotpixmask:

(UNTESTED) List of hot pixels to be combined when using the "PROPER" method.

superbias:

Superbias reference file for the detector of the simulation. This file is used only when combining the simulated signals and the dark current integration using the "PROPER" method. In this case, the dark current integration has the reference pixel correction, superbias subtraction, and linearity correction applied before adding its signals to the linear simulated signals. The combined signals are then "non-linearized" and the superbias and reference pixel corrections are removed. As the superbias subtraction is performed by the DMS pipeline, the superbias file must be in CRDS format. If the superbias file is not specified but the "PROPER" method is used, the script will fall back to using the current superbias reference file in CRDS.

subarray_defs:

A whitespace-delimited ascii file that lists all of the possible supported subarray apertures for the instrument used in the simulation. For each subarray, the file must list the aperture name and OPGS name(define). Also, the detector on which that subarray is used (remove), and the necessary filter for the subarray (for the special GRISM entries).

In addition, the file must list the x,y values of the lower left and upper right corners of the subarray, as well as the number of amplifiers used in the readout. Finally, the x,y and corresponding v2,v3 coordinates of the reference location within the subarray are needed. These values are used when translating locations between RA, Dec and detector x,y. An example of several lines of the NIRCam subarray definition file are shown below.

```
# Definitions of available NIRCam subarrays.
# x and y starting and ending coordinates, and reference pixel coordinates
# are 0-indexed.
#
AperName Name Detector Filter xstart ystart xend yend num_amps refpix_x refpix_y refpix_v2 refpix_v3
NRCA1_FULL FULL A1 ANY 0 0 2047 2047 4 1023.5 1023.5 120.6714 -527.3877
NRCA2_FULL FULL A2 ANY 0 0 2047 2047 4 1023.5 1023.5 120.1121 -459.6806
NRCA3_FULL FULL A3 ANY 0 0 2047 2047 4 1023.5 1023.5 51.9345 -527.8034
NRCA4_FULL FULL A4 ANY 0 0 2047 2047 4 1023.5 1023.5 52.2768 -459.8097
NRCA5_FULL FULL A5 ANY 0 0 2047 2047 4 1023.5 1023.5 86.1035 -493.2275
NRCB1_SUB160 SUB160 B1 ANY 0 1 159 160 1 79.5 79.5 -92.0533 -487.085
NRCB2_SUB160 SUB160 B2 ANY 1 1887 160 2046 1 79.5 79.5 -91.5481 -496.3272
NRCB3_SUB160 SUB160 B3 ANY 1887 1 2046 160 1 79.5 79.5 -82.0967 -487.1041
NRCB4_SUB160 SUB160 B4 ANY 1888 1887 2047 2046 1 79.5 79.5 -82.2869 -496.2056
NRCB5_SUB160 SUB160 B5 ANY 944 944 1103 1103 1 79.5 79.5 -86.8726 -491.7008
NRCA5_GRISM_F277W FULL B5 F277W 0 0 2047 2047 4 1581.0 484.0 50.8765 -527.4932
NRCA5_GRISM_F322W2 FULL B5 F322W2 0 0 2047 2047 4 1581.0 484.0 50.8765 -527.4932
```

readpattdefs: Text file which gives the definitions of the possible readout patterns for the instrument. For each readout pattern, the nframe and nskip values must be defined, as well as the maximum number of allowed groups. The file used for NIRCam is shown below. This will be the primary source of information for ¡i¿nframe¡/i¿ and ¡i¿nskip¡/i¿. For a given ¡i¿readpatt¡/i¿ the simulator will first search the entries in this file in order to determine the proper ¡i¿nframe¡/i¿ and ¡i¿nskip¡/i¿ values to use. In the case where no entry exists, the simulator will attempt to move on, and simulate the data with the given ¡i¿nframe¡/i¿ and ¡i¿nskip¡/i¿ values.

```
name nframe nskip maxgroups
RAPID 1 0 10
BRIGHT1 1 1 10
```

BRIGHT2 2 0 10
SHALLOW2 2 3 10
SHALLOW4 4 1 10
MEDIUM2 2 8 10
MEDIUM8 8 2 10
DEEP2 2 18 20

DEEP8 8 12 20

linearity:

Name of the reference file containing the linearity correction coefficients. If the "PROPER" method of combining simulated and dark current signals is used, the reference file is used by the DMS pipeline to linearize the signal in the dark current integration, as described in the superbias entry above. Regardless of the combination method, this file is used to linearize the values in the saturation reference file, such that saturated signals in the linear simulated signal integration can be found. This file must be in CRDS format.

saturation:

Name of the reference file listing the saturation signal level for all pixels. If the "PROPER" method of combining the simulated and dark current signals is used, the reference file is used by the DMS pipeline to flag saturated pixels in the dark current integration prior to linearizing. If the saturation reference file is not given but the "PROPER" combination method is used, then the script falls back to using the appropriate reference file within CRDS. In this case, the saturation reference file must be in CRDS format.

If the "STANDARD" combination method is used, the saturation reference file is used only to look for saturated signals in the combined dark current and simulated signal integration. In this case, the saturation file does not necessarily have to be in CRDS format. The only assumption is that the saturation map is located in the 1st (not 0th) extension of the fits file.

gain:

Name of the file containing the gain map appropriate for the detector being used. This file is used to translate the simulated signals, which are in units of electrons or electrons per second, to units of DN. The gain map is assumed to be in the 1st (not 0th) extension of the fits file. The file need not be in CRDS format.

phot:

The phot reference file allows the input simulated astronomical sources to be translated from units of magnitudes to electrons per second. Listed in the phot reference file is the countrate in electrons per second, which corresponds to a vega magnitude of 15. An additional column lists the quantum yield corresponding to each filter. Example lines from the NIRCam phot file are shown below:

filter countrate_for_vegamag15 quantum_yield
F070W 49281.1447294 1.0
F090W 57680.7834349 1.0
F115W 51327.5199917 1.0
F140M 21316.1696660 1.0
F150W2 149792.82264 1.0

illumflat:

Name of the illumination flat to use. (UNTESTED) Once the simulated integration is created, the result is divided/multiplied by the illumination flat.

pixelflat:

Name of the pixel flat file to use. Once the simulated integration is created, the result is multiplied/divided by the pixel flat. This is done to un-flatten the image.

astrometric:

Name of the CRDS-formatted astrometric distortion reference file to use to introduce distortion to the simulated data. This file is used to translate input source locations between RA and Dec coordinates and pixel x and y coordinates. In this way, the locations of the simualted sources include the effects of distortion.

distortion_coeffs:

Name of a csv file that contains the full set of coefficients for translating between pixel x, y coordinates and RA, Dec. This purpose of this file is that the translation from RA,Dec to x,y as provided by the astrometric file is only approximate. For example, in the corners of the NIRCam detectors, the x,y location of a given source can be translated into RA and Dec. When that RA and Dec is translated back into x,y, the result can differ from the original x,y by up to about 20 pixels.

The coefficients within this distortion_coeffs file can produce a translation which is exact. Like the astrometric file, this file is used for translating source locations between RA, Dec and x,y.

The ramp simulator software was designed around the distortion coefficient file provided by Colin Cox of the Telescopes Group.

ipc:

File containing the IPC kernel to apply to the simulated data. After all simulated objects have been added to a countrate image, that image is convolved with the IPC kernel. The file must be a fits file with the IPC kernel located in the first (rather than 0th) extension. Typical JWST IPC reference file kernels are a 3x3 array, but the ramp simulator supports kernels of any odd-numbered size.

invertIPC:

If set to True, the IPC kernel is inverted before convolving with the signal rate image. JWST IPC kernel reference files contain the kernel necessary to remove IPC from the data. Therefore these kernels must be inverted before they can add IPC to the data.

crosstalk:

ASCII file containing crosstalk coefficients. Crosstalk is only simulated in data read out through more than one amplifer. The file is assumed to have one row for each detector. The row must contain entries for TONS AND TONS OF COEFFICIENTS.

occult:

Untested. File containing an image of an occulter to be added to the data.

filtpupilcombo:

Name of an ascii file containing a list of the filter and pupil wheel elements in place when requesting simulated data for a given filter. An example of the file, convering some of the shortwave filters in NIRCAm, is shown below.

filter filter_wheel pupil_wheel
F070W F070W CLEAR
F090W F090W CLEAR
F115W F115W CLEAR
F140M F140M CLEAR

F150W2 F150W2 CLEAR
F150W F150W CLEAR
F162M F150W2 F162M
F164N F150W2 F164N
F182M F182M CLEAR

pixelAreaMap:

Fits file containing the pixel area map for the detector to be simulated. This is used to introduce distortion into the simulated data prior to combining it with the dark current integration. We assume the file is in the format of the JWST pixel area map reference file. The pixel area map is multiplied (????) into the data in order to introduce the distortion.

8 nonlin

limit:

Signal limit, in units of DN, above which the linearity correction is not applied. Pixels with signals above this limit are considered saturated. This single value across the entire detector is only used if a saturation reference file is not provided in the reference file section of the parameter file.

accuracy:

When introducing non-linearity back into the linear data, the Newton-Raphson method is used. The value in this accuracy parameter is the threshold below which the solution is considered to have converged. Units? (DN or fractional signal?)

maxiter:

The maximum number of iterations of the Newton-Raphson method to use when introducint non-linearity back into the data before declaring failure.

robberto:

If set to True, the script assumes a Massimo Robberto-style of linearity coefficients. (REFERENCE TRs). For traditional-style coefficients, set to False.

9 cosmicRay

path:

Path of the location of the cosmic ray library to use. .

library:

Specification of which cosmic ray library to choose cosmic rays from. Options are SUNMIN, SUNMAX, FLARE

scale:

Scaling factor to apply to the cosmic ray rate. For example, to simulate cosmic rays at a rate twice as high as that in SUNMIN, set scale to 2.0.

suffix: Filename suffix of the cosmic ray library files.

seed:

Random number generator seed to use when selecting cosmic rays to add.

10 simSignals

pointsource:

Name of an ascii file listing point sources to add to the simulated image. The file has three columns. The first two give the positions of the sources. Positions can be in either detector x,y coordinates, or in RA and Dec. RA and Dec entries can be either strings (e.g. 23:59:59.8) or decimal degrees (e.g. 18.99857). If the positions are given in units of pixels, then the top line of the file must contain: # position_pixels

The third column lists the vega magnitudes of the sources.

psfpath:

Path name to the PSF library to be used for adding point sources to the data.

psfbasename:

Basename of the files in the PSF library.

psfpixfrac:

It is assumed that the PSF library contains PSFs for sources centered at various sub-pixel locations. This parameter specifies the fraction of a pixel between entries in the library. For example, if the library contains PSFs centered at every 0.1 pixels across a pixel, then enter 0.1.

psfwfe:

PSF wavefront error value to assume. Use this parameter if your PSF library contains PSFs for multiple wavefront error values. For NIRCam, we produced a library for the following WFE values: XXXXXX

psfwfegroup:

WebbPSF contains 10 different realizations for each filter/wavefront error-specified PSF. Here, place the realization number to use. XXXXINSERT WEBBPSF documentation link hereXXXX

galaxyListFile:

Similar to the pointsource entry, this is an ascii file containing a list of the galaxies to simulate in the data. The file contains columns for position (x and y, or RA and Dec) half-light radius in pixels or arcseconds, ellipticity, position angle (degrees), sersic index, and magnitude. Positions can be given in x and y pixels on the detector, RA and Dec strings, or RA and Dec decimal degrees. Similarly, the half-light radius can be given in units of pixels or arcseconds. If the positions are given as x and y pixel values, you must include '# position_pixels' as the top line of the file. If the half-light radii are given in pixels, then the second line of the file must be '#radius_pixels'.

Here is an example of the file when positions and half-light radii are given in pixels:

```
# position_pixels
# radius_pixels
#
#Columns 1 and 2 can be either x,y positions on the detector aperture (e.g.
#0,0 is lower left corner of the full frame of the subarray used for the
#output) or RA,Dec location of the center of the source. If they are x,y
#positions, make the top line of the file '# pixels'
#
#radius is the half-light radius in pixels or arcseconds. If in pixels
#make the second line of the file '# radius_pixels
#
#pos_angle is the position angle of the semimajor axis, in degrees
#0 causes the semi-major axis to be horizontal.
```

x_or_RA y_or_Dec radius ellipticity pos_angle sersic_index magnitude

1000 1000 10 0.25 20 2.0 20

1000 1015 15 0.5 79 1.2 19

And an example when the locations are given in RA and Dec, and the radii in units of arcseconds:

#

x and y can be pixel values, or RA and Dec strings or floats. To differentiate, put 'pixels' in the top line if the inputs are pixel values.

radius can also be in units of pixels or arcseconds. Put 'radius_pixels' at top of file to specify radii in pixels.

position angle is given in degrees counterclockwise. A value of 0 will align the semimajor axis with the x axis of the detector.

x_or_RA y_or_Dec radius ellipticity pos_angle sersic_index magnitude

23.97834 0.006157822 3.07 0.51 183.2 1.55 25.86

0.001729 -0.01565028 1.55 0.62 61.1 1.58 19.11

extended:

Name of a fits file containing an extended image that you wish to add to the simulation. This is a way to add any type of signal you wish to the data. It is assumed that the fits file contains an array in the 1st (rather than 0th) extension. The array is assumed to be in units of electrons per second. The array can be any size. If it is larger than the field of view of the simualted data, then it is cropped. If it is smaller, then it is centered at the location specified in the extendedCenter parameter.

extendedScale:

Factor by which to scale the data in the extended image file before adding to the simulated data. The extended image is multiplied by this factor.

extendedCenter:

x,y location (in pixels) at which to center the array contained in the extended image.

PSFConvolveExtended:

True/False, whether to convolve the extended image with the appropriate PSF prior to adding to the output image.

movingTargetList: Similar to the point source list file, this is a file containing a list of targets to treat as moving (non-sidereal) targets which will move through the background of the field of view. This is the list to use if you wish to insert an asteroid or KBO that is moving through the field of view of your observation. As shown below, the columns in the moving target list file are similar to those in the point source list file.

The same logic applies for the target positions in this file. Positions can be in units of pixels on the detector, RA,Dec in decimal degrees, or RA,Dec strings. To specify that the positions have units of pixels, "position_pixels" must be present in the top line of the file. In the example below the units of the position are decimal degrees.

Similarly, if the objects' velocities must be given in arcsec per hour or pixels per house. If they are given in units of pixels per hour, then velocity_pixels must be present in the second line of the file.

#

#

#

#list of point sources to create as moving targets (KBOs, asteroids, etc)

#position can be x,y or RA,Dec. If x,y, put the phrase 'position_pixels' in the top

#line of the file. Velocity can be in units of pix/hour or arcsec/hour.

#If using pix/hour, place 'velocity_pixels' in the second line of the file.

#Note that if using velocities of pix/hour, the results will not be

#strictly correct because in reality distortion will cause object's

#velocities to vary in pixels/hour. Velocities in arcsec/hour will be

#constant.

x_or_RA y_or_Dec magnitude x_or_RA_velocity y_or_Dec_velocity

12.0 12.0 19.0 -0.5 -0.02

0.007 0.003 18.0 -0.5 -0.02

359.998 0.010 19.5 0.05 -0.65

movingTargetSersic: Similar to the galaxy target list file, this file contains a list of galaxies (2D sersic profiles) to be used as moving targets. These sources will move through the background of the simulated data. This may be useful for inserting a resolved moon/asteroid into the scene. An example file is shown below. The 'position_pixels' and 'radius_pixels' options discussed in other input files are used here as well.

#

#

#

#Columns 1 and 2 can be either x,y positions on the detector aperture (e.g.

#0,0 is lower left corner of the full frame of the subarray used for the

#output) or RA,Dec location of the center of the source. If they are x,y

#positions, make the top line of the file '# position_pixels'

#

#radius is the half-light radius in pixels or arcseconds. If in pixels

#make the second line of the file '# radius_pixels

#

#pos_angle is the position angle of the semimajor axis, in degrees.

#0 causes the semi-major axis to be horizontal.

x_or_RA y_or_Dec radius ellipticity pos_angle sersic_index magnitude x_or_RA_velocity y_or_Dec_velocity

23:59:59.9986 +00:00:00.0047 1.0 0.25 20 2.0 16.000 -0.5 -0.02

00:00:00.0003 +00:00:00.0216 1.5 0.5 79 1.2 16.000 -0.5 -0.02

23:59:58.6185 +00:00:32.1260 1.5 0.25 0 2.0 18.000 -0.5 -0.02

00:00:02.1327 +00:00:20.7322 1.5 0.25 70 2.0 17.000 -0.5 -0.02

movingTargetExtended:

This is a text file listing extended targets to move through the background of the image. The first column of the file lists the (fits) filenames from which the background images are collected. An example is shown below. Inserting "velocity_pixels" in the second line of the file will specify velocities given in pixels per second rather than the default arcseconds per second.

#

#

#List of stamp image files to read in and use to create moving targets.

#This is the method to use in order to create moving targets of

#extended sources, like planets, moons, etc.

#position can be x,y or RA,Dec. If x,y, put the word 'pixels' in the top

#line of the file. Velocity can be in units of pix/sec or arcsec/sec.

#If using pix/sec, place 'velocity_pixels' in the second line of the file.
#Note that if using velocities of pix/sec, the results will not be
#strictly correct because in reality distortion will cause object's
#velocities to vary in pixels/sec. Velocities in arcsec/sec will be
#constant.
filename x_or_RA y_or_Dec magnitude pos_angle x_or_RA_velocity y_or_Dec_velocity
ring_nebula.fits 0.007 0.003 12.0 0.0 -0.5 -0.02

movingTargetConvolveExtended: Set this input to True if you wish to convolve the scenes listed in the files listed in movingTargetExtended with the instrument PSF prior to adding them to the simulated data.

movingTargetToTrack: This text file is used for what are traditionally called 'moving targets'. Targets listed in this file are treated as non-sidereal targets which JWST will track during the simulated observation. In this case, the target listed in this file will appear static in the output data, but all other sources (e.g. those listed in pointSource, galaxyListFile, and extended) will all appear trailed through the data. An example of the file is shown below. In the first column of the file, the type of object is specified. Objects listed as 'point' are considered point sources. Objects labeled 'sersic' will be treated as 2d sersic profiles. Any other labels will cause an object to be treated as an extended target. XXcheck thisXXXXXX.

#

#

x and y can be pixel values, or RA and Dec strings or floats. To differentiate, put 'pixels' in the top line if the inputs are pixel values.

radius can also be in units of pixels or arcseconds. Put 'radius_pixels' at top of file to specify radii in pixels.

position angle is given in degrees counterclockwise. A value of 0 will align the semimajor axis with the x axis of the detector.

#An object value containing 'point' will be interpreted as a point source.
#Anything containing 'sersic' will create a 2D sersic profile.
Any other value will be interpreted as an extended source.
#x_or_RA_velocity is the proper motion of the target in units of arcsec (or pixels) per year
#Y_or_Dec_velocity is the proper motion of the target in units of arcsec (or pixels) per year
#if the units are pixels per year, include 'velocity pixels' in line 2 above.
object x_or_RA y_or_Dec x_or_RA_velocity y_or_Dec_velocity magnitude
pointSource 359.9984688844023 -0.0016999909612954896 87660. 0.0 20.794845635303744

zodiacal:

Zodiacal light countrate image file.

zodiscale:

Scaling factor to multiply the zodiacal light countrate image by prior to adding to the output data.

scattered:

Scattered light countrate image file.

scatteredscale:

Scaling factor to multiply the scattered light countrate image by prior to adding to the output data.

bkgdrate:

Constant (across all pixels) background countrate to add to the output data, in units of electrons/second.

poissonseed:

Random number generator seed used for Poisson simulation

photonyield:

True/False. Include the effects of photon yield in the simulated data.

pymethod:

True/False. Use double Poisson simulation for photon yield.

11 Telescope

ra:

Right ascension of the observation. This will be the RA at the location of the reference location o the detector. The reference location varies with the requested subarray, but is generally in the center of the field of view. This input can be a string "HH:MM:SS.sss", or a float in decimal degrees.

dec:

Declination of the observation. This will be the Dec at the location of the reference location o the detector. The reference location varies with the requested subarray, but is generally in the center of the field of view. This input can be a string "DD:MM:SS.sss" or a float in decimal degrees.

textbfrotation:

Rotation of the scene. Currently this rotation is defined around the reference location of the chosen subarray?? (or maybe it's simply around the center of the field of view). Units are degrees.

12 newRamp

combine_method:

****NOTE: as of the addition of moving targets, only 'PROPER' is supported.******

Possible values are "STANDARD", "PROPER", or "HIGHSIG". This value controls the method with which the simulated signal integration is combined with the dark current integration. In the STANDARD method, the linear simulated signals are added to the dark current integration. The resulting integration then has non-linearity applied to it. This method is slightly faster, but not strictly correct, as any pixels with significant signal in the dark current integration (hot pixels, cosmic ray hits), are affected by non-linearity and should be linearized before adding to the linear simulated signal.

In the PROPER method, the dark current integration is first superbias subtracted, reference pixel corrected, and then linearized, using the appropriate steps of the JWST SSB pipeline. The linear dark current is then added to the linear simualted signals. The resulting ramp then has non-linearity re-added, along with the reference pixel values and superbias. This method is mathematically the correct way to add the two integrations together.

If HIGHSIG is specified, then pixels which have significant signal in the dark current integration are combined using the PROPER method, and all other pixels are combined with with STANDARD method. This choice involves essentially running both the STANDARD and PROPER methods, and is therefore the slowest.

proper_combine:

NOTE: as of the addition of moving targets, only 'BOTH' is supported.

Possible inputs: "BOTH", "HOTPIX", or "COSMICRAYS" If the "HIGHSIG" method for combining the simualted signals and dark current integration is specified, this input controls which pixels are combined using the PROPER method. If HOTPIX is specified, then only hot pixels are combined with the PROPER method. In this case, the script will either identify hot pixels on the fly as any pixel with a signal at the end of the ramp greater than the value in the **proper_signal_limit** entry, or it will use the hot pixel mask listed in the **hotpixmask** entry in the **Reffiles** section of the parameter file.

linearized_darkfile:

If provided, this file will be read in and used as the linearized dark current integration for the PROPER or HIGHSIG combination methods. If this file is provided, the user still must provide the name of the raw dark current ramp in the **Reffiles:dark** entry. This is because the difference between the raw and linearized dark ramps is used as a measure of the superbias and reference pixel signals which must be added back into the dark current file after adding the simulated data and non-linearizing. Again, using this option is not strictly mathematically correct for pixels with high signal levels in the raw dark current file.

dq_configfile:

Name of configuration file to be used in the dq_init step of the SSB pipeline when it is run on the raw dark current integration when using the PROPER or HIGHSIG combination methods.

sat_configfile:

Name of configuration file to be used in the saturation step of the SSB pipeline when it is run on the raw dark current integration when using the PROPER or HIGHSIG combination methods.

superbias_configfile:

Name of configuration file to be used in the superbias step of the SSB pipeline when it is run on the raw dark current integration when using the PROPER or HIGHSIG combination methods.

refpix_configfile:

Name of configuration file to be used in the reference pixel subtraction step of the SSB pipeline when it is run on the raw dark current integration when using the PROPER or HIGHSIG combination methods.

linear_configfile:

Name of configuration file to be used in the linearity correction step of the SSB pipeline when it is run on the raw dark current integration when using the PROPER or HIGHSIG combination methods.

13 Outputs

file:

Filename of the output simulated integration fits file.

format:

Input options: DMS

DMS - save the output fits file in DMS format using JWST data models. This option will produce an output file which can immediately be run through the SSB pipeline.

textbfsave_intermediates:

True/False. If true intermediate products are saved to disk, including the simulated-only integration, the countrate image, etc.

textbfgrism_source_image:

True/False. If True, an additional output image is produced. This output image is larger than the requested array size, and can be used as a direct image in a grism simulator.

textbfgrism_input_only:

True/False. If True, the simulator will exit after producing the grism source image.

textbfunsigned:

True/False. If True, output the simulated integration in unsigned integers. This will produce output which matches real raw data.

textbfdmsOrient:

True/False. If True, data will be output in DMS orientation, as opposed to FITSWriter orientation. Use True to simulate raw data as an observer will see it.