Simulating Systematic Errors in Exoplanetary Transits for the James Webb Space Telescope [*]

University of Central Florida Honors Undergraduate Thesis

David Wright

April 9, 2021

Overview

1. Motivation
2. Literature Review and Relevant Work
3. Methodology & Results - Adding JWST Features to ExoSim, Validation, and Comparisons
4. Future Work and Closing Remarks

Motivation

Background Information

The upcoming™ James Webb Space Telescope (JWST) offers multiple observation modes tailored specifically for transit observation

- Exoplanet transits are prime observational targets for characterizing exoplanet atmospheres
- So far, we've only studied a small fraction of the more than 4,000 discovered exoplanets using transits

Need

The community needs simulation tools that create realistic simulations including instrument systematic errors and noise

- Why is this needed?
  - Plan observations
  - Create and test data reduction and analysis pipelines
  - Characterize instrument performance for specific observations
- Currently available JWST simulation tools do **not** satisfy these requirements
- In this thesis, I present an open-source time-domain simulator of exoplanet transits based on ExoSim **SP20** for JWST

[*] This PDF document is an inferior version of an OER HTML page; free/libre Org mode source repository.

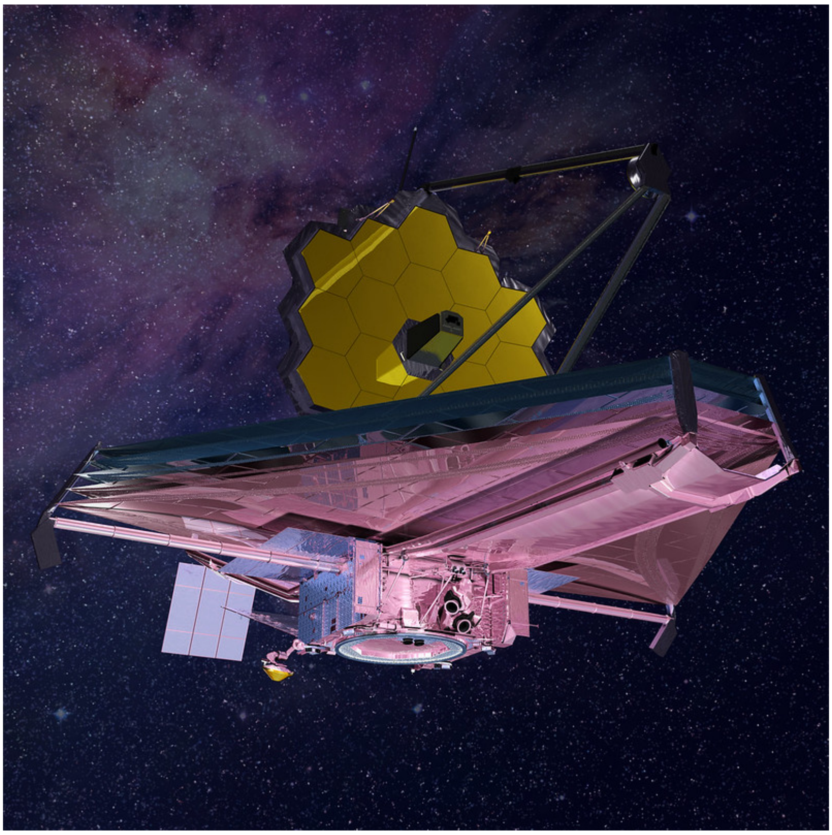


Figure 1: Artist render of JWST

Literature Review and Relevant Work

Exoplanet Transits

- In 1995, Mayor and Queloz discovered the first exoplanet orbiting a sun-like star [**MA95**]
  - 51 Pegasi b, detected via doppler spectroscopy of the host star
  - As the planet orbits its host star, it shifts the radial velocity of the star and induces doppler shifts in the star's spectra

In 1999, HD 209458 b was discovered via photometric transit observations [**CH00**]

Figure 2: Hot Jupiters. Image Credit: ESA

Exoplanet Transits

figures/dave/transitgif.gif

- Total flux of the star+planet system has a fractional decrease of $\approx \left(\frac{R_P}{R_S}\right)^2$, often called the transit **depth**

Exoplanet Transits

figures/dave/transitgif.gif

- The planet's radius is $R_P = R_S\sqrt{\text{Depth}}$
- This is a small ratio, usually ~1-2% for a typical hot Jupiter

Exoplanet Transits

- In practice, this ratio is found by fitting model light curves to the data, such as the model described by Mandel and Agol [**MA02**]

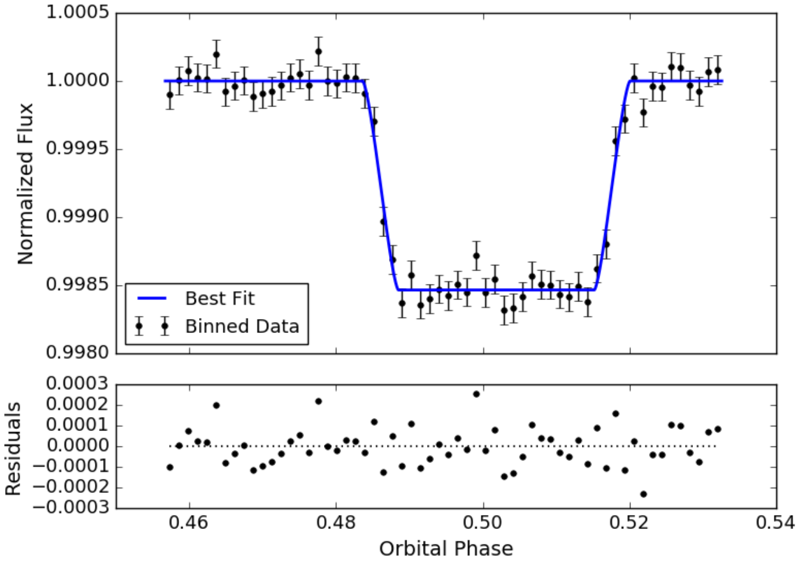


Figure 3: HD 209458 b light curve fit with a Mandel and Agol model. Image Credit: Kathleen McIntyre

- A photometric observation of a transit light curve (such as this one) allows for some characterization of the planet, such as surface gravity, average density, and orbital period
- Note how small the fractional decrease is, just 0.15%
  - Keep in mind this is for a **Hot Jupiter**

Exoplanet Transits

Spectroscopic observations yield more information on exoplanet atmospheres

Exoplanet Transits

The $\left(\frac{R_P}{R_S}\right)^2$ ratio at different wavelengths, $\left(\frac{R_P(\lambda)}{R_S}\right)^2$, is coined the "transmission spectrum"

- If the exoplanet has an atmosphere, the radius where it becomes optically opaque is wavelength-dependent
- Transmission spectra therefore yield information on the atmosphere's structure and composition

Exoplanet Transits

Transiting planets on circular orbits also pass behind their host star, known as an eclipse or secondary transit

Eclipse observations yield information on the thermal emission of a planet and reflected starlight

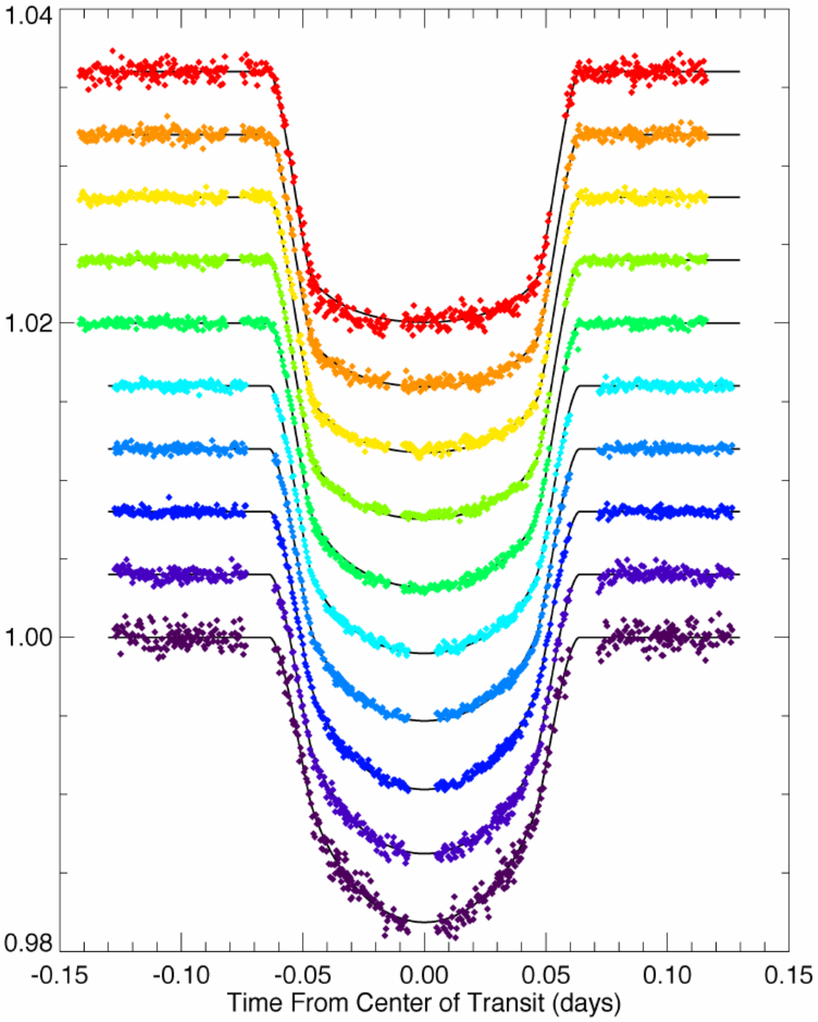


Figure 4: HD 209458 b light curves from $0.29 - 1.04\mu$m. Image Credit: Heather Knutson

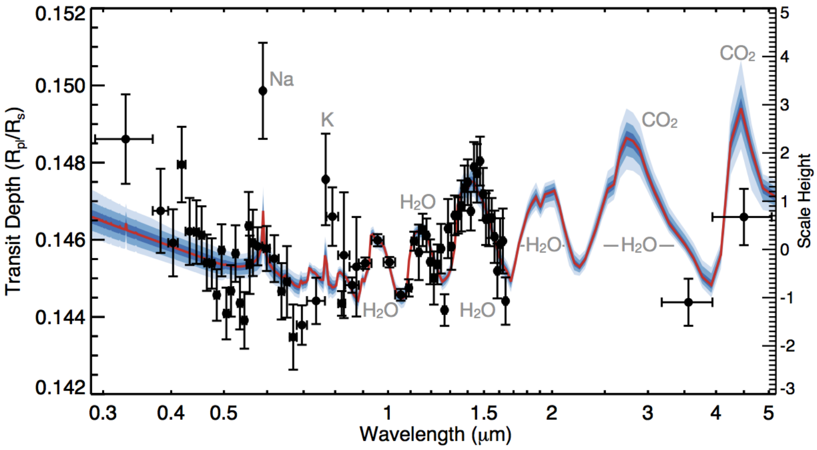


Figure 5: WASP-39b light curves from $0.3 - 5.0\mu$m. Image Credit: Wakeford et al. 2017 [**WA17**]

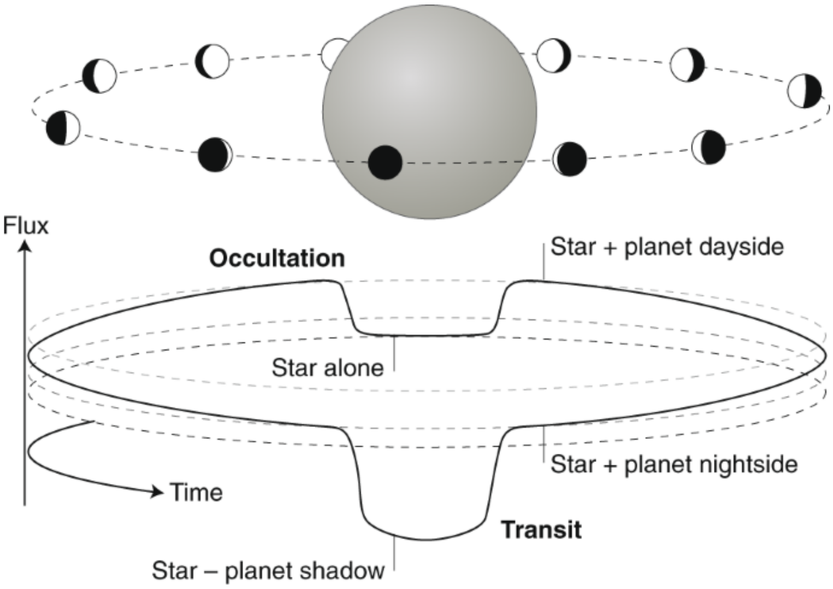


Figure 6: Exoplanet phase curve. Image Credit: Deming & Knutson 2020 [**DK20**]

Exoplanet Transits

As the planet passes behind its star, the observed flux decreased by $\left(\frac{R_P}{R_S}\right)^2 \frac{F_{\lambda,P}}{F_{\lambda,S}}$ with F_λ being the emergent flux density of the planet or star

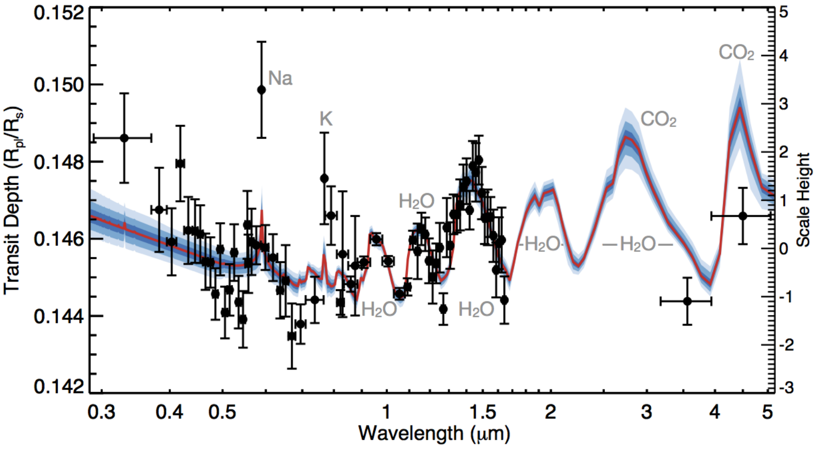


Figure 7: WASP-39b light curves from $0.3 - 5.0\mu$m. Image Credit: Wakeford et al. 2017 [**WA17**]

The James Webb Space Telescope (JWST)

The James Webb Space Telescope (JWST)

- JWST has exoplanet science as one of its main goals
- JWST will observe the 0.6–28μm range with detectors that are capable of greater than 100 parts per million (ppm) precision over time periods from hours to days [**BE14**]

- For the last two decades, the two most prominent telescopes used for exoplanet atmosphere characterizations were the Hubble and Spitzer Space Telescopes
  - At launch, neither of these telescopes had exoplanet science in their mission plans
- High sensitivity and broad wavelength coverage enables accurate measurement of transit and orbital parameters with a (comparatively) high signal-to-noise ratio (S/N)

The James Webb Space Telescope (JWST)

- JWST has a 25m^2 primary mirror, the largest of any space telescope to date
- It will orbit the Earth-Sun L2 point, providing extremely stable observations over long time periods

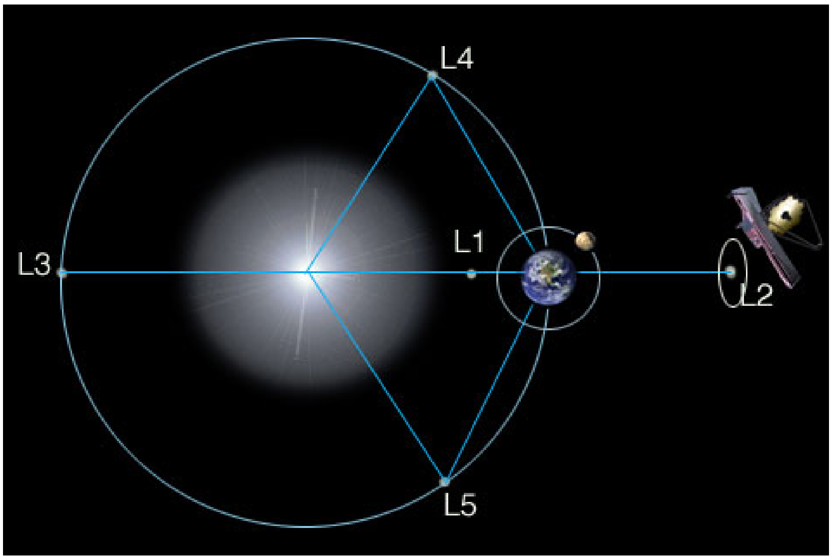
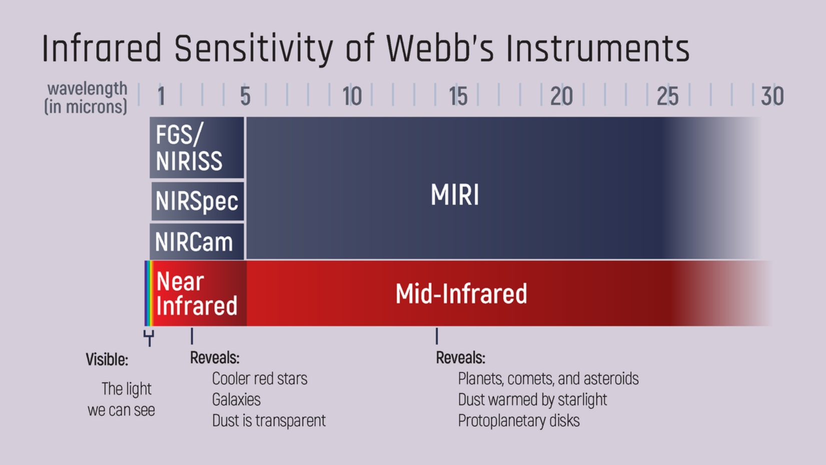


Figure 10: JWST orbit at Earth-Sun L2 point. Image credit: NASA

The James Webb Space Telescope (JWST)

- Webb carries four instruments on board:
  - The Near-InfraRed Camera (NIRCam, [**BE12**])
  - The Near-InfraRed Spectrograph (NIRSpec, [**FE12**])
  - The Near Infrared Imager and Slitless Spectrograph (NIRISS, [**DO12**])
  - The Mid-Infrared Instrument (MIRI, [**KE15**])



JWST Simulation Tools

Instrument Specific

- NIRSpec - N. Batalha [**BA15**]

- NIRISS - D. Louie [**LO18**]
- MIRI - P. Klaasen [**KL20**]
- NIRCam - J. Leisenring [**LEPr**]

- NIRSpec simulator is written in IDL and is not open source
- NIRISS simulator doesn't simulate curved or multiple-ordered traces
- MIRI simulator doesn't model time domain, in fact, none of these do
- NIRCam simulator is in preparation, not well documented
- Not designed for exoplanet transit simulations, save for NIRSpec and NIRISS

General JWST

- JexoSim - S. Sarkar [**SA20**]
- PandExo - N. Batalha [**BA17**]

- JexoSim is based on ExoSim
    - Does simulate in the time domain
    - Doesn't model multiple ordered traces or JWST readout patterns
    - Not open-source
- PandExo is built on top of Pandeia [**PO16**]
    - Doesn't simulate the time domain
    - Uses box model of transit

Methodology & Results

A Brief Overview of ExoSim

- ExoSim is a modular, time-domain simulator of exoplanet transits
- ExoSim simulates the full light curve, including instrument noise and systematic errors
- An ExoSim simulation is configured using a user-supplied XML file containing the parameters for the instrument(s) and observation and paths to relevant reference data (see reference files on diagram).
- Users can specify parameters such as which noise sources to include, whether to simulate a transit or eclipse, phase curve effects, and the planet to observe
- The Channel object is instantiate to hold channel specific parameters
- Astroscene
    - After reading in the XML configuration file, the ExoSim algorithm moves on to the Astroscene module
    - In this module, the Star and Planet object classes are instantiated
        - PHOENIX stellar model or black body
        - Open Exoplanet Catalogue (OEC) for planet and star parameters (distance, temperatures, orbital inclination, etc.)
    - The flux density at the telescope aperture is calculated using the generated stellar model and the parameters taken from the OEC
- The background flux density at the telescope is calculated immediately following the astroscene module
- Instrument

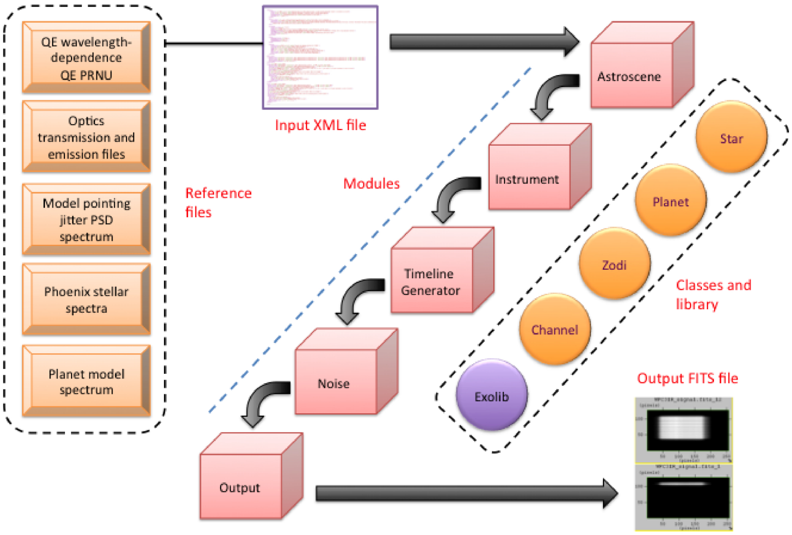


Figure 11: Overview of ExoSim Algorithm. Image credit: Enzo Pascale

- – The power per unit wavelength at the telescope aperture is calculated
- – The incoming signal is
  - ∗ modulated by the optical elements of the telescope
  - ∗ placed on an oversampled focal plane using the given wavelength-pixel relation
  - ∗ convolved with a point-spread function
  - ∗ converted to photoelectrons using the given quantum efficiency of the detector
  - ∗ convolved with an intra-pixel response function to give a final, digitized focal plane

- Timeline
  - Using the input timing parameters, the timeline module sets up the 3D data cube that contains the time series data of the observation
  - The full light curve is computed
  - For computation reasons, the light curve is actually applied in the Noise module
- Noise
  - Read noise, shot noise, dark current, jitter noise, and inter-pixel variations
  - Read and shot noise are generated by respectively sampling a normal and poisson distribution
  - Dark current value is given in the configuration file, and value*time is added to each pixel in a frame
  - Jitter noise is modeled using a user-supplied power spectral density (PSD) file
  - ExoSim uses this file to generate random shifts in the x and y directions over small time scales (less than a frame time)
    - ∗ Each small time scale focal plane is coadded to result in the final frame
- Output
  - ExoSim outputs FITS files, much like a real observation

  - Extra FITS HDUs with useful information, like the detector dispersion relation, are appended after the simulated image HDUs
  - The FITS headers contain other information, like the target star and planet and the detector readout noise
  - No post processing is done on these images, so they are very close to the results of a real observation

Adding New Features to ExoSim

- In order to simulate JWST, ExoSim requires the following additional capabilities

  1. Simulation of curved spectral traces
  2. Simulation of multiple-ordered spectral traces
  3. Use instrument specific PSFs
  4. Support JWST detector readout patterns
  5. Parallelization

Curved Spectral Traces

Allow a 2D wavelength-pixel relation, $\lambda(x, y)$

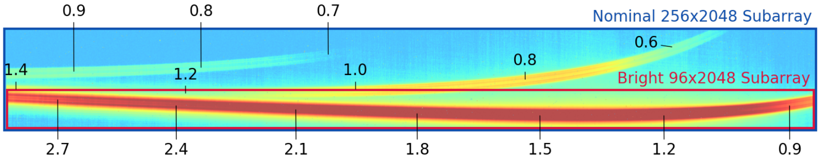


Figure 12: NIRISS SOSS. Three orders cover a wavelength range of $0.6 - 2.8\mu$m. Image Credit: STScI

- ExoSim originally assumed that the spectral trace fell exactly along the midpoint of the detector
- I modified ExoSim to allow for a 2D wavelength-pixel relation
  - Users can supply this in the configuration file
  - I wrote simple python scripts to convert relevant Pandeia reference data to ExoSim's format
- All aspects of ExoSim that expected a 1D wavelength-pixel relation were updated to work with a 2D relation
- PSFs can now fall anywhere in the y direction of the pixel, subsample the focal plane to place the psfs appropriately

Multiple-Ordered Spectral Traces

Simulate a focal plane for each spectral order and coadd

- 2 Issues
  - Order dependent configuration options
  - Interpolation over a function which is not one-to-one
- Add configuration options for order dependent dispersion relations and throughputs

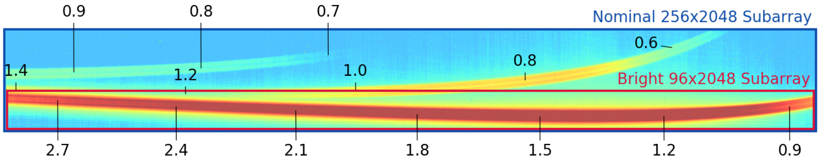


Figure 13: NIRISS SOSS. Three orders cover a wavelength range of $0.6 - 2.8\mu$m. Image Credit: STScI

- ExoSim relies heavily on interpolation and rebinning over a wavelength range
- As seen in NIRISS SOSS, these orders can overlap in wavelength
  - How to interpolate when you have multiple f(x) for the same x?
- I simulate a separate focal plane for each order and then coadd when appropriate, taking care not to double count any noise sources

JWST Readout Patterns

Readout	N_{frames}	N_{skip}
NISRAPID	1	0
NIS	4	0
NRSRAPIDD1	2	1
SHALLOW4	5	1

- Description of JWST readout pattern scheme
  - The basic unit of a JWST readout pattern is a **frame**
  - A **group** consists of a series of non-destructively read frames and possibly skipped frames at the end of the group which are not saved.
  - The frames in a group are averaged by the on-board electronics
  - A destructive read followed by collection of groups is an **integration**
  - An **exposure** is a set of integrations
- Basic problem: JWST's **group** does not exist in ExoSim
- I added in the ability to specify the number of frames and number of skipped frames in the configuration file (see table)
- The frames are averaged together appropriately before output

Instrument Specific PSFs

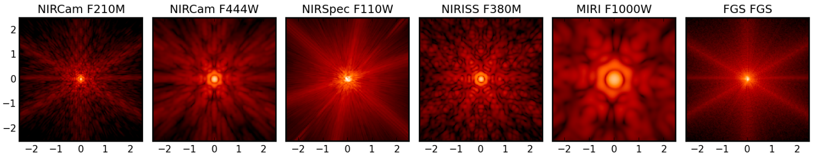


Figure 15: Example PSFs from each JWST instrument and the fine guidance sensor.

- ExoSim supports external PSFs, but expects them in a certain format
- I modified the existing exosim PSF functions to accept PSFs produced by WebbPSF directly

  - In practice, just use Pandeia data
  - I wrote a wrapper around WebbPSF to generate monochromatic psfs over a given wavelength grid in parallel

Parallelization

Parallelization gives nearly an exact proportional decrease in run time for number of parallel processes

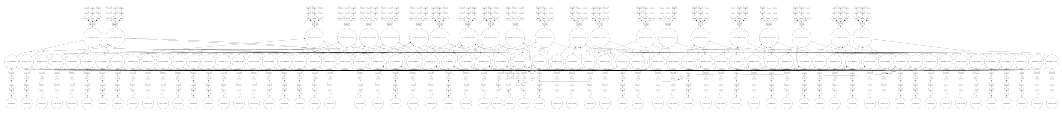


Figure 16: Dask parallel computing library task graph built during ExoSim execution.

- With the combination of large detectors, oversampling of the focal plane, long observation times, and the changes made to accommodate multiple ordered traces, run time drastically increased for certain configurations
- NIRISS SOSS took nearly two hours when simulating all three orders
- I used the Dask Python parallel computing library to parallelize sections of ExoSim
- I was able to drop NIRISS SOSS run time from two hours to 7 minutes with 20 threads
- Dask basics
  - It has two main parts:
    - Distributed data structures that are similar to NumPy and Pandas
    - A task scheduler
  - Dask, for most things, is a drop in replacement for scipy, numpy, and pandas that is parallelized
  - Users can specify how many threads they'd like to use, and how much memory is allowed per thread
    - Dask calls these units "workers"
  - Dask builds up a "task graph" whenever its functions are called, but it doesn't actually evaluate anything
  - When the user asks for the results, Dask determines what can be run in parallel and schedules tasks to the available workers
  - The end result is a normal numpy array
  - An interesting thing about Dask is that it can work nicely with other job schedulers, such as SLURM or PBS
  - In the code I wrote, Dask is configured to see the local computer as a "cluster" of workers
  - By changing this configuration, which is only 5 lines of code, one could actually run ExoSim in parallel on a normal university cluster
  - This could be extremely useful if one were to simulate a very large detector
  - It can also interface with AWS and Kubernetes for cloud computing

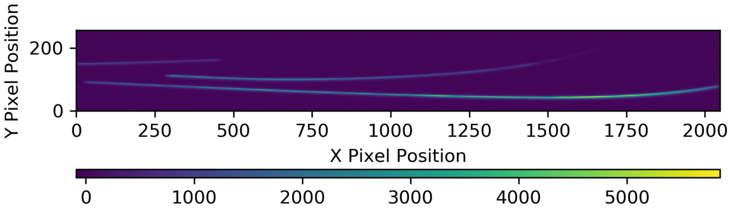


Figure 17: NIRISS SOSS simulation from ExoSim.

Example NIRISS SOSS Simulation

Validation

Validating the Focal Plane Signal

A simulated black body was compared to an analytic expression

- A simulation without noise was performed for an out-of-transit observation of the star 55 Cancri.
- The star was modeled as a black body with an effective temperature of 5172K.
- In this test, the PSFs were represented as a series of delta functions, i.e., all of the signal fell directly on a pixel and there was no convolution with a PSF.
- The focal plane was saved to disk in the **Instrument** module.
- The columns of the focal plane were summed over, and the resulting signal was compared to the predicted value given by

$$Q(x) = \pi B_{\lambda(x)}(5192K)\left(\frac{R_S}{D}\right)^2 A_{Tel}\eta_{Ch}(\lambda(x))QE(x)\Delta\lambda(x)\frac{\lambda(x)}{hc}.$$

- In order to validate the implementation of multiple order traces, a noiseless simulation of two orders was performed separately and together for a total of three images.
- The out-of-transit, black body system described previously was used in these simulations.
- The two individual simulated images were summed and compared to the simulation of the orders on the same detector.

Comparison to Original ExoSim

To ensure no unforeseen changes were made to the noise models, I ran identical simulations in my modified version of ExoSim and the original version.

- I ran a simulation without noise, then a simulation for each noise source, then a simulation with all noise sources turned on
- I used the FITSDIFF tool provided by astropy to compare the output FITS files
- I found that both versions of ExoSim produce exactly the same FITS files

Comparison to PandExo

- JexoSim and PandExo are the two tools mentioned which are comparable to my modified version of ExoSim.
- JexoSim is not publicly available, so comparisons between simulations are only possible with PandExo.
- Simulations of all four instruments configured for exoplanet transit observations were compared.

Table 1: Configuration for each instrument simulated (cont. next slide)

Parameter	NIRSpec BOTS	NIRCam F444W/Grism-R
Subarray	SUB512	SUBGRISM64
Size	32 × 512	64 × 2048
W_{slit}(pix)(pix)	16	N/A
Δ_{pix}(μm)	18	18
PS($^\circ \times 10^{-5}/\Delta_{pix}$)	2.78	1.75
T(K)	40	40

- In these simulations, all noise sources were disabled except for shot noise.
- Reference data from Pandeia was used for each simulation
- NIRSpec was set to observe 55 Cancri out-of-transit, and the star was modeled as a black body.
- The next instrument to be compared was NIRISS in its SOSS mode.
- 55 Cancri proved to be too bright of a target for NIRISS SOSS, so GJ 1214 was used instead.
- This observation was also out of transit, and the star was modeled as a black body.
- The same observation scene was used for MIRI in its low-resolution spectroscopy (LRS) mode and NIRCam in its grism time-series mode.
- The measured signal was calculated for each simulation using the last-minus-first procedure
  - Also known as correlated double sampling
  - Final readout in integration minus first
- Extracted spectra using JOSE (uses Horne's algorithm) except for NIRISS (had issues with curved trace)
- I used PandExo's binning tools to bin the results spectra down to a common wavelength grid

Comparison to PandExo

Table 2: Configuration for each instrument simulated

Parameter	NIRISS SOSS	MIRI LRS
Subarray	SUBSTRIP256	SLITLESSPRISM
Size	256 × 2048	72 × 416
W_{slit}(pix)	N/A	N/A
Δ_{pix}(μm)	18	25
PS($^\circ \times 10^{-5}/\Delta_{pix}$)	1.81	3.06
T(K)	40	7

- JexoSim and PandExo are the two tools mentioned which are comparable to my modified version of ExoSim.
- JexoSim is not publicly available, so comparisons between simulations are only possible with PandExo.
- Simulations of all four instruments configured for exoplanet transit observations were compared.
- In these simulations, all noise sources were disabled except for shot noise.
- Reference data from Pandeia was used for each simulation

- NIRSpec was set to observe 55 Cancri out-of-transit, and the star was modeled as a black body.
- The next instrument to be compared was NIRISS in its SOSS mode.
- 55 Cancri proved to be too bright of a target for NIRISS SOSS, so GJ 1214 was used instead.
- This observation was also out of transit, and the star was modeled as a black body.
- The same observation scene was used for MIRI in its low-resolution spectroscopy (LRS) mode and NIRCam in its grism time-series mode.
- The measured signal was calculated for each simulation using the last-minus-first procedure
  - Also known as correlated double sampling
  - Final readout in integration minus first
- Extracted spectra using JOSE (uses Horne's algorithm) except for NIRISS (had issues with curved trace)
- I used PandExo's binning tools to bin the results spectra down to a common wavelength grid

Comparisons to PandExo - NIRSpec

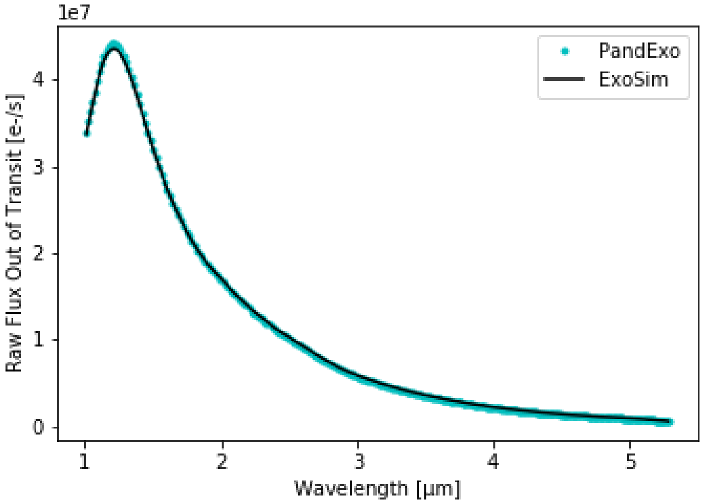


Figure 20: Comparison of PandExo and ExoSim simulation of NIRSpec. Average percent difference of $+1.53 \pm 1.63\%$.

- The reported JexoSim average percent differences are higher except for NIRISS
  - Each instrument has a larger standard deviation for the percent error as well
- Given the excellent results of the focal plane validation tests, I am left to conclude that the differences are due to
  - the internal models of each simulator (radiometric vs dynamic, I have to process the output data)
  - consequence of digitizing the focal plane
  - the configuration of each instrument, i.e., differences with the configuration files
- More time is needed to fully understand the differences

Comparisons to PandExo - NIRCam

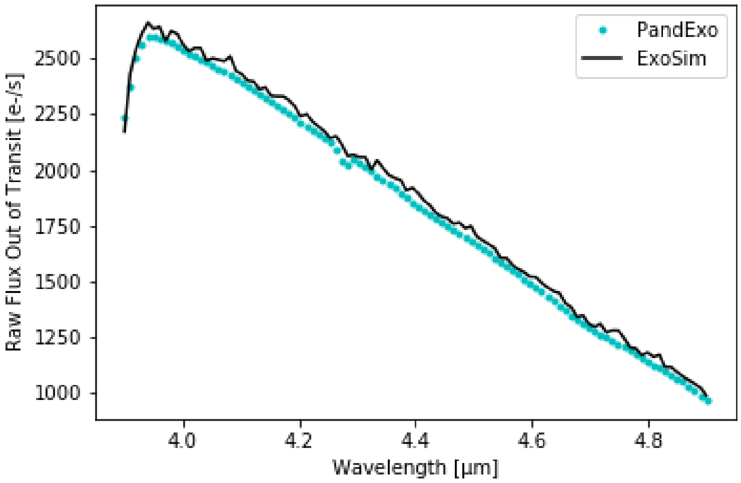


Figure 21: Comparison of PandExo and ExoSim simulation of NIRCam. Average percent difference of $+2.18 \pm 1.08\%$.

- The reported JexoSim average percent differences are higher except for NIRISS
  - Each instrument has a larger standard deviation for the percent error as well
- Given the excellent results of the focal plane validation tests, I am left to conclude that the differences are due to
  - the internal models of each simulator (radiometric vs dynamic, I have to process the output data)
  - consequence of digitizing the focal plane
  - the configuration of each instrument, i.e., differences with the configuration files
- More time is needed to fully understand the differences

Comparisons to PandExo - MIRI

- The reported JexoSim average percent differences are higher except for NIRISS
  - Each instrument has a larger standard deviation for the percent error as well
- Given the excellent results of the focal plane validation tests, I am left to conclude that the differences are due to
  - the internal models of each simulator (radiometric vs dynamic, I have to process the output data)
  - consequence of digitizing the focal plane
  - the configuration of each instrument, i.e., differences with the configuration files
- More time is needed to fully understand the differences

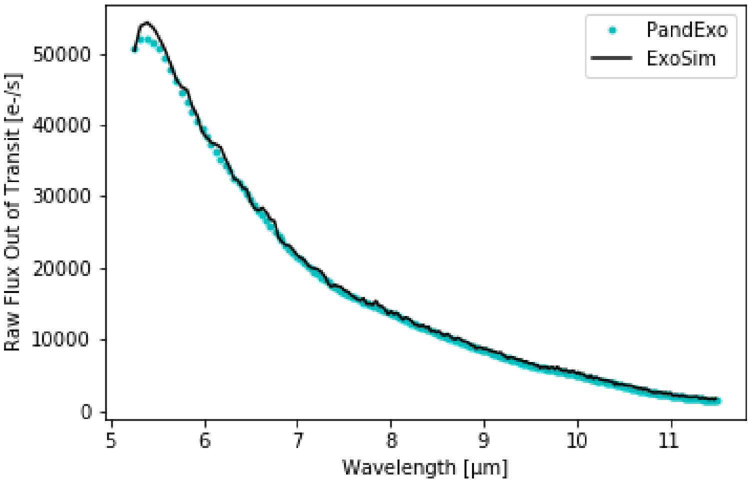


Figure 22: Comparison of PandExo and ExoSim simulation of MIRI. Average percent difference of $+3.0 \pm 2.2\%$.

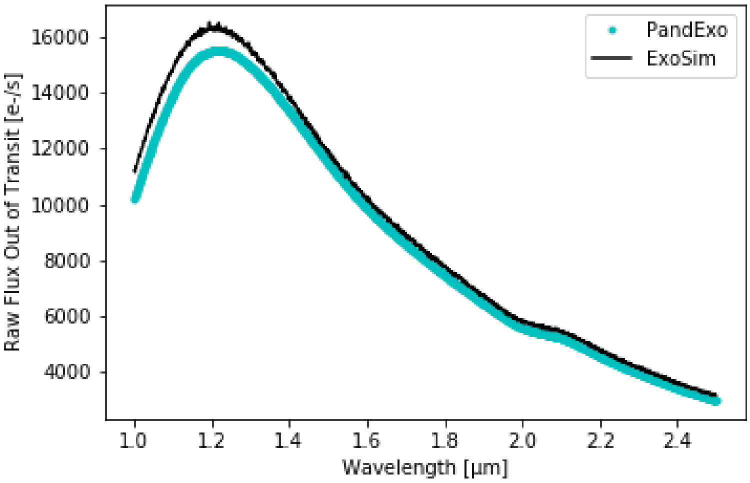


Figure 23: Comparison of PandExo and ExoSim simulation of NIRISS. Average percent difference of $+4.54 \pm 1.37\%$.

Comparisons to PandExo - NIRISS

- The reported JexoSim average percent differences are higher except for NIRISS
  - Each instrument has a larger standard deviation for the percent error as well
- Given the excellent results of the focal plane validation tests, I am left to conclude that the differences are due to
  - the internal models of each simulator (radiometric vs dynamic, I have to process the output data)
  - consequence of digitizing the focal plane
  - the configuration of each instrument, i.e., differences with the configuration files
- More time is needed to fully understand the differences

Future Work and Closing Remarks

Future Work

- Write bridge between ExoSim and Pandeia
- Find cause of discrepancy between simulations
- Combine ExoSim, Py-JOSE, and BART for an end-to-end simulation and retrieval

- Two of the most time consuming tasks when working with ExoSim are sourcing reference files and configuring the instruments.
- It would be beneficial to write code to bridge ExoSim and the Pandeia reference data directly.
- Users would be able to simply specify the name of the filter, disperser, etc. and ExoSim would find and use the correct reference data.
- An in depth study of PandExo and Pandeia is needed to find the cause of the differences observed in this thesis. This study could also be done in tandem with the previous future work mentioned.
- Outside of future work to be done to ExoSim, using ExoSim to simulate known exoplanet spectra and then running atmospheric retrieval code on the simulated spectra is an interesting prospect.
- This end-to-end simulation and retrieval would allow astronomers to benchmark the performance of instruments in regards to specific exoplanet observations or specific observational goals, such as detecting C/O ratios in an exoplanet's atmosphere.

Closing Remarks

In summary, I've described the development and validation of a time-domain simulator of exoplanet transits and systematic errors JWST.

This tool is open-source and available to the community with the hope that it is used to further our understanding of the effects of systematic noise on exoplanet transits and to prepare for observations of exoplanet transits with the James Webb Space Telescope.

Acknowledgements

Thanks to Dr. Joseph Harrington, Dr. Enzo Pascale, Dr. Ryan Challener, and Kathleen McIntyre for their support and guidance

Links

- https://github.com/davecwright3/ExoSimPublic
- https://jwst-docs.stsci.edu/
- https://dask.org/

Bibliography