Monthly Notices of the ROYAL ASTRONOMICAL SOCIETY

MNRAS **508,** 433–452 (2021)
Advance Access publication 2021 September 9

https://doi.org/10.1093/mnras/stab2472

JexoSim 2.0: end-to-end JWST simulator for exoplanet spectroscopy – implementation and case studies

Subhajit Sarkar[1★] and Nikku Madhusudhan[2]
[1] *School of Physics and Astronomy, Cardiff University, The Parade, Cardiff CF24 3AA, UK*
[2] *Institute of Astronomy, University of Cambridge, Madingley Road, Cambridge CB3 0HA, UK*

Accepted 2021 August 17. Received 2021 July 30; in original form 2021 January 26

ABSTRACT

The recently developed JWST Exoplanet Observation Simulator (JexoSim) simulates transit spectroscopic observations of exoplanets by JWST with each of its four instruments using a time-domain approach. Previously, we reported the validation of JexoSim against Pandexo and instrument team simulators. In the present study, we report a substantially enhanced version, JexoSim 2.0, which improves on the original version through incorporation of new noise sources, enhanced treatment of stellar and planetary signals and instrumental effects, as well as improved user-operability and optimizations for increased speed and efficiency. A near complete set of instrument modes for exoplanet time-series observations is now included. In this paper, we report the implementation of JexoSim 2.0 and assess performance metrics for JWST in end-member scenarios using the hot Jupiter HD 209458 b and the mini-Neptune K2-18 b. We show how JexoSim can be used to compare performance across the different JWST instruments, selecting an optimal combination of instrument and subarray modes, producing synthetic transmission spectra for each planet. These studies indicate that the 1.4 μm water feature detected in the atmosphere of K2-18 b using the Hubble WFC3 might be observable in just one transit observation with JWST with either NIRISS or NIRSpec. JexoSim 2.0 can be used to investigate the impact of complex noise and systematic effects on the final spectrum, plan observations and test the feasibility of novel science cases for JWST. It can also be customized for other astrophysical applications beyond exoplanet spectroscopy. JexoSim 2.0 is now available for use by the scientific community.

Key words: techniques: spectroscopic – planets and satellites: atmospheres.

1 INTRODUCTION

Since the discovery of the first extra-solar planets a quarter of a century ago (Wolszczan & Frail 1992; Mayor & Queloz 1995), our understanding of planets around other stars has expanded greatly. We know of the existence of over 4000 exoplanets[1] covering a diverse range of sizes, masses, temperatures, and orbital configurations. We find that our Solar system is far from typical in the Galaxy. The discovery of diverse planets in wide-ranging system architectures has challenged planet formation and evolution theories, with a need to understand the factors and conditions that bring about the diversity seen. To this end, atmospheric spectroscopy characterizes exoplanets by obtaining compositional, structural, and dynamic information of their atmospheres (Madhusudhan 2019). The atmospheric compositions can in turn provide important insights into planetary formation and evolution mechanisms. Spectroscopy is also key in the search for biosignature gases indicative of life (Seager 2014; Schwieterman et al. 2018). Atmospheric spectroscopy of exoplanets has been achieved through the transit-based techniques of transmission and eclipse spectroscopy (Seager & Sasselov 2000; Charbonneau et al. 2002; Grillmair et al. 2008; Deming et al. 2013), as well as direct imaging (Janson et al. 2010; Konopacky et al. 2013). The latter, while not constrained by the need for transits, is currently limited to studying young gas-giant planets at wide separations from their host stars.

Transmission spectroscopy depends on wavelength-dependent absorption of stellar light by atmospheric atomic and molecular species, as the planet transits in front of the star (e.g. Charbonneau et al. 2002; Sing et al. 2016; Diamond-Lowe et al. 2020). This technique probes the high altitude atmosphere of the planet day–night terminator, returning a spectrum consisting of $(R_\mathrm{p}/R_\mathrm{s})^2$ with wavelength, where R_p is the apparent planet radius and R_s is the star radius. It has been applied in the visible and near-infrared wavelength ranges where the star is brighter, allowing higher signal-to-noise, thereby making detection of the spectral features of key chemical species more accessible at these wavelengths. Eclipse spectroscopy depends on the occultation of the planet's reflection and emission flux as it passes behind the star (secondary eclipse). The eclipse depth variations with wavelength give the spectrum of $F_\mathrm{p}/(F_\mathrm{s} + F_\mathrm{p})$, where F_p is the planet flux and F_s is the stellar flux. Eclipse spectroscopy provides constraints on the temperature structure and composition of the dayside atmosphere (e.g. Line et al. 2013; Stevenson et al. 2014; Madhusudhan 2018).

The *Hubble Space Telescope* has been used to great effect in obtaining transmission spectra of exoplanets using instruments such as STIS, NICMOS, ACS, and WFC3 (Charbonneau et al. 2002; Pont et al. 2008; Swain, Vasisht & Tinetti 2008; Sing et al. 2011; Kreidberg et al. 2014); however, the passbands of the spectra obtained are

[★] E-mail: subhajit.sarkar@astro.cf.ac.uk
[1] https://exoplanetarchive.ipac.caltech.edu/

Published by Oxford University Press on behalf of Royal Astronomical Society

narrow (e.g. 1.1–1.7 µm for the Hubble WFC3) so that only a small fraction of the spectrum is revealed. The *Spitzer Space Telescope* has also been used to obtain transmission and emission spectra, but these consisted of just a few photometric bands (Agol et al. 2010; Knutson et al. 2011; Deming & Knutson 2020). Ground-based facilities are hampered by the infrared absorption windows of the Earth's atmosphere, as well as having to contend with atmospheric scintillation and turbulence effects. Wider wavelength coverage is desirable to reduce degeneracies and overcome impact of clouds (Kreidberg et al. 2014; Barstow et al. 2015; Sing et al. 2016; Madhusudhan 2019; Welbanks & Madhusudhan 2019). The next generation of transmission and emission spectra will be obtained from space-based observatories with wide wavelength coverage such as the James Webb Space Telescope (JWST) (Gardner et al. 2006) and Ariel (Tinetti et al. 2018), greatly improving the quality and quantity of exoplanet spectra obtained.

JWST is planned for launch in 2021. All four instruments onboard are able to perform transit spectroscopy of exoplanets. These are the Near Infrared Imager and Slitless Spectrograph (NIRISS) (Doyon et al. 2012), the Near Infrared Camera (NIRCam) (Beichman et al. 2012), the Near Infrared Spectrograph (NIRSpec) (Ferruit et al. 2014), and the Mid-Infrared Instrument (MIRI) (Rieke et al. 2015). Combined, these instruments provide a wavelength coverage from 0.5 to 28.3 µm. JWST's 6.5-m primary mirror will be largest to date in space, reducing photon noise to unprecedented levels. Reduced photon noise, however, could mean that other noise sources and systematics become more important and impact the data. For JWST therefore, it is particularly important to consider additional sources of noise or systematic trends that could possibly confound the detection of small signals such as exoplanet atmospheric signatures afforded by its low photon noise. Proper accounting of all errors is also crucial when establishing the final experimental uncertainties of a study including the impact of correlated noise. The size of the final error bars on the planet spectrum impacts the uncertainties in spectral retrievals (e.g. Madhusudhan & Seager 2009; Line et al. 2013; Waldmann et al. 2015; Madhusudhan 2018) and the scientific conclusions of a study. Simulators that model the instrumental and astrophysical noise, as well as systematics, during an observation can help to assess the performance of JWST. They can be used in the planning and optimization of observational and data reduction strategies, and for testing the feasibility of specific science cases.

Since transit spectroscopy is a time-domain technique it is particularly vulnerable to time-dependent systematic trends from both the instrument, such as detector persistence (Berta et al. 2012), or from astrophysical sources, such as spot occultations and stellar variability (Pont, Zucker & Queloz 2006; Rackham, Apai & Giampapa 2018). In addition, time-correlated noise, e.g. from pointing jitter or stellar granulation or pulsation (Sarkar et al. 2018), can inflate the final uncertainties and needs to be properly accounted for (Pont et al. 2006). Most simulators for JWST do not model the time-domain directly, but estimate the final noise on the transit depth through a static calculation. This approach leads to rapid computations which is ideal for assessing large sets of planets; however, they lack the functionality to model time-correlated noise or time-dependent systematic trends. Pandexo (Batalha et al. 2017) simulates all four JWST instruments giving signal-to-noise ratio (SNR) estimates based on set of complex computations using the Pandeia engine (Pontoppidan et al. 2016). While Pandexo generates 2D detector images allowing it to model spatially correlated noise sources, it does not produce a frame-by-frame time-domain simulation which limits its ability to model complex time-dependent processes. Simulators that have been produced by individual JWST instrument teams include the NIRISS SOSS 1D[2] (Beichman et al. 2014) and 2D simulators[3], a 1D simulator developed for NIRCam and MIRI (Greene et al. 2016), PyNRC for NIRCam[4] (Leisenring 2018), the NIRSpec instrument performance simulator (Piquéras et al. 2010) and the NIRSpec Exoplanet Exposure Time Calculator (NEETC) (Nielsen et al. 2016). To our knowledge these do not produce frame-by-frame time-domain simulations.

We developed the first version of JexoSim (the JWST Exoplanet Observation Simulator) (Sarkar, Madhusudhan & Papageorgiou 2019) to fill this gap in the range of simulators available for JWST. Uniquely, JexoSim can simulate exoplanet transit observations using all four JWST instruments using a frame-by-frame time-domain approach, generating 2D image time-series akin to an actual observation. It can model and capture the effects of time-correlated noise and systematic trends. JexoSim can also capture 2D spatial- and wavelength-dependent systematics, and the potentially complex interaction between spatial and temporal processes, allowing measurement of their ensemble impact directly on the planet spectrum.

In the previous work (Sarkar et al. 2019), we described the JexoSim algorithm in detail and validated the simulator against Pandexo and instrument team simulators. In this work, we report an enhanced version, JexoSim 2.0, with new noise sources, added instrumental effects including quantum yield and inter-pixel capacitance (IPC) and enhanced fidelity in the stellar and planetary signal processing algorithm. We have made it faster, more versatile and easier to use. User-operability has been greatly improved with several architectural changes including a simplified user interface and installation procedure, together with more up-to-date and reduced number of dependent packages. In addition, a number of optimizations have been implemented to improve speed, reduce memory pressure, and enable the simulation of observations with very large numbers of integrations. We now include a near complete set of JWST instrument modes designated for exoplanet time-series observations. With these enhancements JexoSim 2.0 is now available to the community[5].

In this paper, we describe the upgrades made in JexoSim 2.0 and provide a guide to its implementation. We then use the simulator to assess JWST performance metrics for specific case studies. We focus on exoplanet transmission spectroscopy using two end-member scenarios: the hot Jupiter HD 209458 b and the mini-Neptune K2-18b. We perform simulated observations using all four JWST instruments and demonstrate different simulation recipes. These include using Allan deviation analysis and Monte Carlo simulations to obtain precision on transit and eclipse depths and a noise budget analysis. We use JexoSim to assess which instrumental configurations are optimal for these end-member scenarios.

2 JEXOSIM 2.0

JexoSim 2.0 is a time-domain simulator of exoplanet transit spectroscopy using JWST and its four instruments, designed to test and validate specific and novel science cases for JWST observation. Both primary and secondary transits can be simulated, as well as out-of-transit (OOT) observations. Time-series images are generated representing the sequence of non-destructive reads (NDRs) per integration. The images can be processed using an included pipeline for immediate results, or stored in FITS format for later analysis. Simulated planet spectra with error bars can be generated, which can

[2] http://maestria.astro.umontreal.ca/niriss/simu1D/simu1D.php
[3] https://github.com/jasonfrowe/JWSTNIRISS
[4] https://github.com/JarronL/pynrc
[5] https://github.com/subisarkar/JexoSim

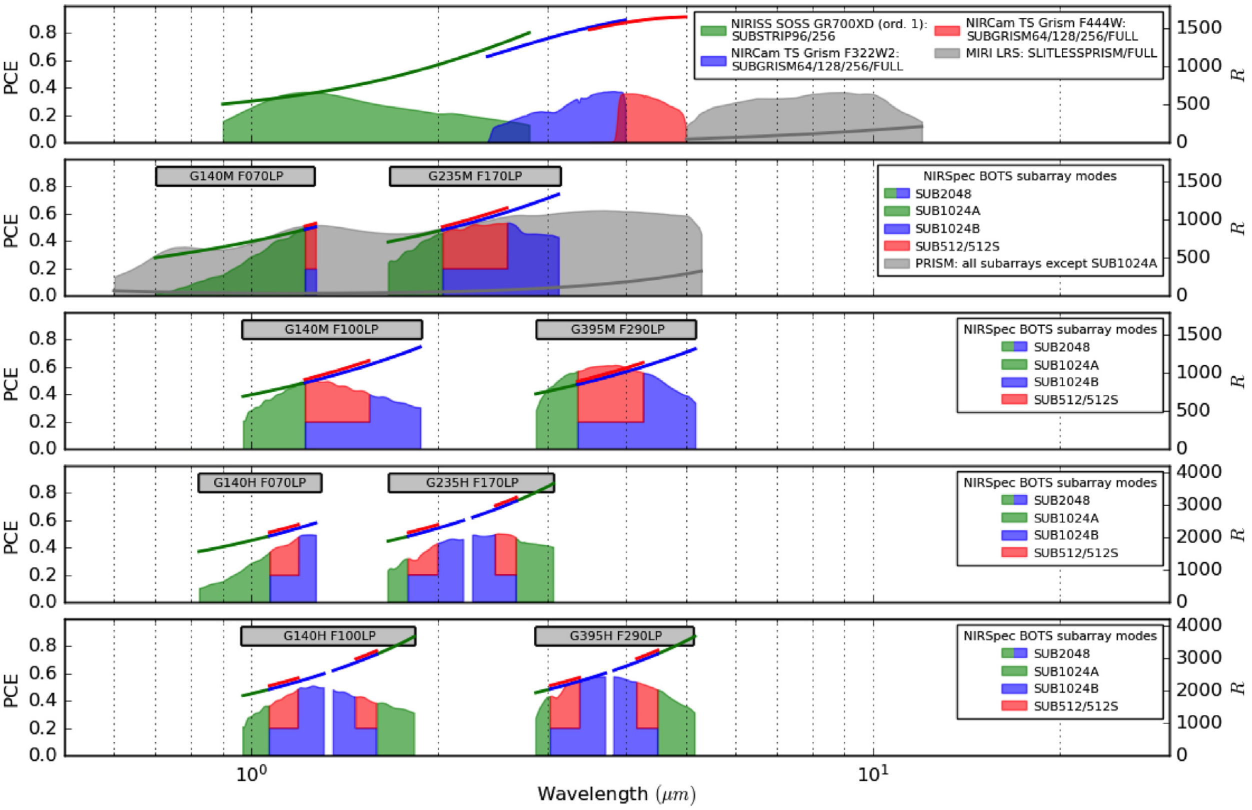


Figure 1. JexoSim 2.0 available instrument configurations shown with their photon-conversion efficiency (PCE) (upper bounds of shaded areas) and spectral resolving power (*R*) (floating lines) versus wavelength. Nominal wavelength ranges for each mode are shown. In NIRSpec, the SUB512 and SUB512S modes have wavelength ranges that are subsets of the SUB1024B mode, hence where they overlap in range the colours are split between the modes (SUB512/SUB512S in red, SUB1024B in blue). SUB2048 encompasses the full wavelength range of each NIRSpec mode, i.e. the combination of green and blue areas. JexoSim 2.0 uses instrument-specific transmissions and quantum efficiency files from the Pandeia data base (Pontoppidan et al. 2016), producing the final PCEs shown. The *R* power information is also derived from Pandeia. Where modes overlap, the *R* powers are slightly offset for clarity. SOSS = Single Object Slitless Spectroscopy. BOTS = Bright Object Time Series. TS Grism = Time Series Grism. LRS = Low Resolution Spectrometer.

be used to test the detectability of atmospheric features, e.g. through spectral retrieval studies. Alternately signal and noise information from the OOT stellar signal can be extracted. Nearly, all JWST instrument configurations dedicated to time-series exoplanet spectroscopy are now included in JexoSim 2.0, with their available subarray modes (Fig. 1), although the MIRI Medium Resolution Spectrograph (MRS) mode is currently not simulated. In NIRISS only the first-order spectrum from the GR700XD grism is currently simulated.

Next, we describe the upgrades made in JexoSim 2.0 in architecture, algorithm, dependences, and optimizations.

2.1 Architecture

The architecture of JexoSim 2.0 is outlined in Fig. 2. A core sequence of modules is supported with data bases and reference files, with various inputs and outputs to the system including an optional pipeline. Here, we briefly discuss these components.

2.1.1 Modules

JexoSim 2.0 retains from the original version the sequence of modules that executes the processing algorithm (Sarkar et al. 2019). The modules simulate the flow and modulation of the signal from the star to the detector followed by the generation of a time-series image stack of NDRs. The modules, however, now refer to dedicated libraries that hold associated functions, while the modules themselves control the calling of these functions. This organization will aid future augmentation allowing for easier upgrades and addition of new functions.

2.1.2 Inputs

To improve user operability, full control over the simulation is now achieved through a new text-based ‘input parameters file’ (Fig. 2). This file gives the user wide control over the simulation parameters and can be edited or duplicated to run particular simulation scenarios. A new ‘paths file’ (also a text file) allows the user to specify the path of the output folder for results. As in the previous version, default simulation and instrument parameters are held in XML-based configuration files which link to specific reference files (e.g. transmissions, wavelength solutions, etc.). A new `params` class mediates the interaction of the input parameters file with the rest of the code modifying default values held in the XML files. A new set of XML files has been added for extra-Solar system (‘exosystem’) parameters. When an exoplanet is first referenced from the planet data base (see Section 2.3), an XML file containing the system parameters is generated and then re-used in subsequent simulations of the same system. A new set of files called ‘recipe’ files have been added that control patterns of simulation, types of results and outputs. The code

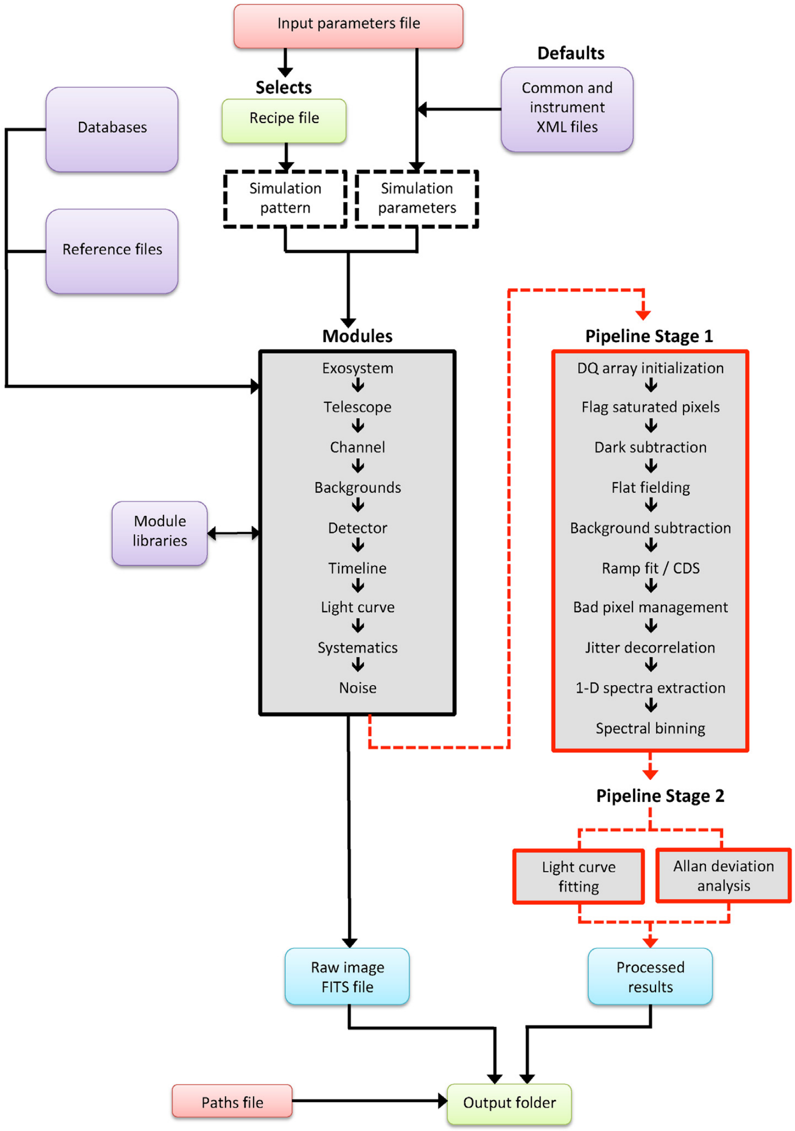


Figure 2. Overview of JexoSim 2.0 architecture. A new 'input parameters file' provides the user-interface and gives control over the simulation. Recipe files direct the pattern of the simulation, while XML files hold instrumental parameters, links to reference files and a default set of simulation parameters. The simulation proceeds through a sequence of 'modules'. These are described in Sarkar et al. (2019), together with the processing algorithm. Modifications to the algorithm in JexoSim 2.0 are illustrated in Fig. 3. The modules now link to dedicated libraries. The final image stack is stored as a FITS file or processed via the JexoSim pipeline.

will select a particular recipe depending on choices made in the input parameters file.

2.1.3 *Outputs*

The time-series of NDR images is the final output of the signal processing algorithm. This can be packaged into FITS format for processing by an external pipeline. Alternately, the user can choose to process the images through the included JexoSim pipeline for immediate results. The main steps in this pipeline are shown in Fig. 2 and described in detail in Sarkar et al. (2019). If processed through the pipeline, the output files are produced in.pickle format (containing processed data) and .txt format (containing parameters from the input parameters file or derived from the simulation itself). All output files are dumped to the specified output folder.

2.2 Algorithm

The original signal processing algorithm (the sequence of calculations executed in the modules) was described in detail in Sarkar et al. (2019) and illustrated in Fig. 1 of that paper. In the new version, the algorithm code has been refactored for efficiency and the entire JexoSim code updated from PYTHON 2 to PYTHON 3. Modifications to the algorithm are described below. A summary of the algorithm and changes made in JexoSim 2.0 is illustrated in Fig. 3.

2.2.1 *Stellar and planetary spectra*

To account for resolution-linked bias (RLB; Deming & Sheppard 2017), JexoSim 2.0 has a modified treatment of the star and planet spectra. The bias refers to a mismatch between the real transit depth

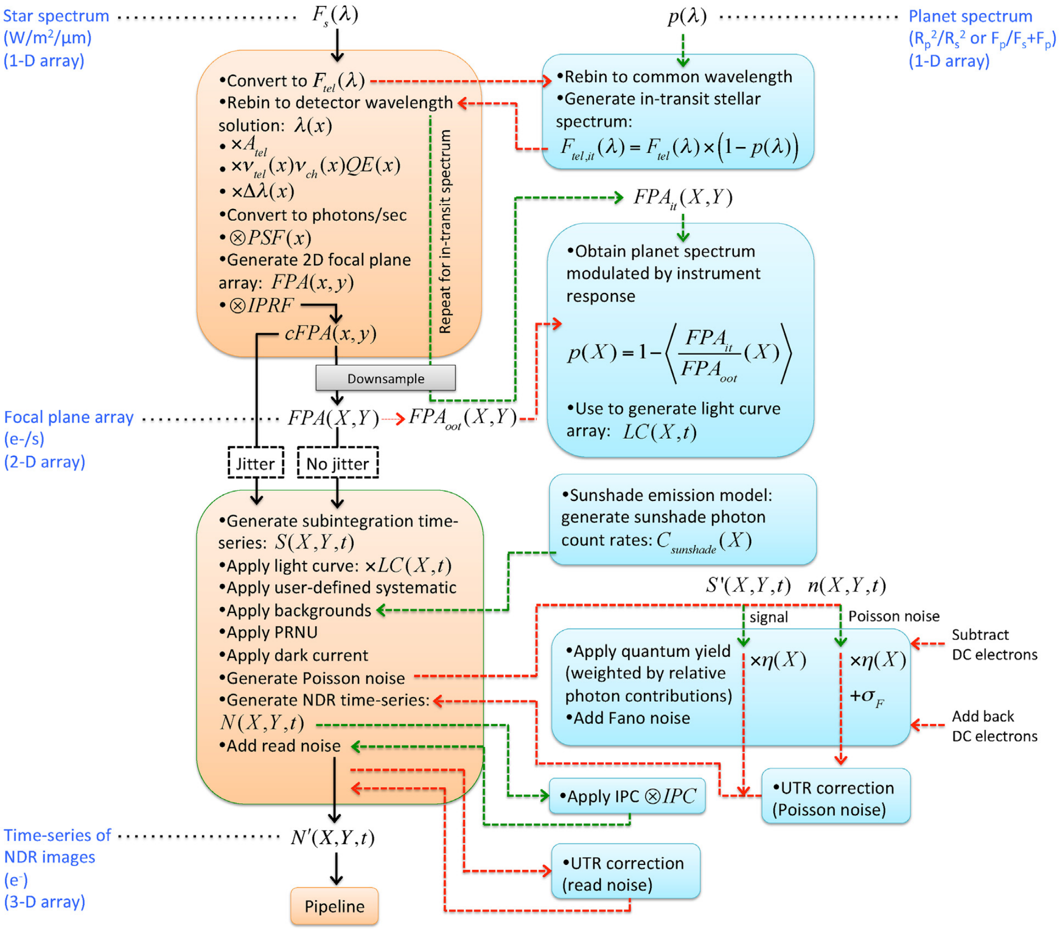

Star spectrum
(W/m^2/µm)
(1-D array)
$F_s(\lambda)$
$p(\lambda)$
Planet spectrum
(R$_p^2$/R$_s^2$ or F$_p$/F$_s$+F$_p$)
(1-D array)
•Convert to $F_{tel}(\lambda)$
•Rebin to detector wavelength solution: $\lambda(x)$
• $\times A_{tel}$
• $\times \nu_{tel}(x)\nu_{ch}(x)QE(x)$
• $\times \Delta\lambda(x)$
•Convert to photons/sec
• $\otimes PSF(x)$
•Generate 2D focal plane array: $FPA(x,y)$
• $\otimes IPRF$
$cFPA(x,y)$
Repeat for in-transit spectrum
•Rebin to common wavelength
•Generate in-transit stellar spectrum:
$F_{tel,it}(\lambda) = F_{tel}(\lambda) \times (1 - p(\lambda))$
$FPA_{it}(X,Y)$
•Obtain planet spectrum modulated by instrument response
$p(X) = 1 - \left\langle \frac{FPA_{it}}{FPA_{oot}}(X) \right\rangle$
•Use to generate light curve array: $LC(X,t)$
Downsample
Focal plane array
(e-/s)
(2-D array)
$FPA(X,Y)$
$FPA_{oot}(X,Y)$
Jitter
No jitter
•Sunshade emission model: generate sunshade photon count rates: $C_{sunshade}(X)$
•Generate subintegration time-series: $S(X,Y,t)$
•Apply light curve: $\times LC(X,t)$
•Apply user-defined systematic
•Apply backgrounds
•Apply PRNU
•Apply dark current
•Generate Poisson noise
•Generate NDR time-series: $N(X,Y,t)$
•Add read noise
$S'(X,Y,t)$
$n(X,Y,t)$
signal
Poisson noise
•Apply quantum yield (weighted by relative photon contributions)
•Add Fano noise
$\times\eta(X)$
$\times\eta(X)$
$+\sigma_F$
Subtract DC electrons
Add back DC electrons
•Apply IPC $\otimes IPC$
•UTR correction (Poisson noise)
•UTR correction (read noise)
Time-series of NDR images
(e$^-$)
(3-D array)
$N'(X,Y,t)$
Pipeline

Figure 3. Simplified representation of the updated JexoSim algorithm. Right-hand side boxes (blue) show main changes in the algorithm introduced in JexoSim 2.0. Left-hand side boxes (orange) mostly follow the original algorithm, the details of which are completely described in Sarkar et al. (2019). *X* and *Y* are the spectral and spatial coordinates of whole pixels on the focal plane detector array, *FPA*(*X*, *Y*). An oversampled grid, *FPA*(*x*, *y*), is initially generated (for Nyquist-sampling of the PSF and jitter simulation), which has subpixel coordinates *x* and *y*. $A_{\rm tel}$ = telescope aperture area. $\nu_{\rm tel}$ = telescope transmission. $\nu_{\rm ch}$ = instrument channel transmission. $\Delta\lambda$ = wavelength span across subpixel in spectral direction. *IPRF* = 2D intra-pixel response function. *cFPA*(*x*, *y*) = oversampled focal plane array convolved with intra-pixel response function. Other abbreviations are described in the figure or text.

in a spectral bin and that predicted using a transit spectrum model if the model utilizes insufficient spectral resolution. If the planet spectrum, $p(\lambda)$, is binned (or convolved with the PSF) *prior* to it modulating the out-of-transit stellar spectrum, $F_{\rm s}(\lambda)$, this bias is likely to result, and the fidelity of the simulation reduced. The bias increases when absorption lines from the star and planet overlap. To maximize fidelity by minimizing the RLB effect, JexoSim now multiplies the planet spectrum into the stellar spectrum early in the algorithm while both are at their highest resolution (Fig. 3). The 'in-transit' stellar spectrum is next obtained: $F_{\rm s,it}(\lambda) = F_{\rm s}(\lambda)[1 - p(\lambda)]$. The in-transit and out-of-transit stellar spectra are then treated in exactly the same way through a sequence of steps ending with the production of a 2D focal plane array for the out-of-transit signal, $FPA_{\rm oot}(X, Y)$, and the in-transit signal, $FPA_{\rm it}(X, Y)$, where *X* and *Y* are pixel coordinates. Fig. 3 summarizes these steps and a full description is given in Sarkar et al. (2019). We obtain the ratio of these 2D arrays, $FPA_{\rm it}/FPA_{\rm oot}$, and then find the average ratio per pixel column, $<FPA_{\rm it}(X)/FPA_{\rm oot}(X)>$. This returns a 1D array. Subtracting this 1D array from 1, gives $p(X)$, the planet spectrum correctly modulated by the instrument response (Fig. 3). $p(X)$ is used as the input to generating the light curve 2D array, *LC*(*X*, *Y*, *t*), where *t* corresponds to the time of a subintegration[6]. $FPA_{\rm oot}$ is used to generate the subintegration time-series, *S*(*X*, *Y*, *t*), which is modulated by the light curves, systematics, backgrounds, and noise (Fig. 3).

To minimize RLB, the user can supply both the planet spectrum and the star spectrum at very high resolution. JexoSim allows for the user to supply their own stellar and/or planet spectra through the options in the input parameters file (see Section 3.3) (select `file` for options `planet_spectrum_model` and `star_spectrum_model`). As mentioned in Section 2.3, JexoSim uses a default stellar Phoenix BT-Settl spectrum data base. For star parameters of temperature = 3000 K, log *g* = 5.0 and Fe/H = 0, this gives a spectrum with average *R*-power of ∼ 390 000 between 0.05 and 5.5 µm. For the same star and wavelength range,

[6] We use the term 'subintegration' to define as the count accumulated between two successive NDRs (i.e. the difference between them). Co-adding the subintegrations of an integration subsequently generates the ramped NDRs for that integration.

high-resolution Phoenix spectrum by Husser et al. (2013)[7] give an average R-power of ∼ 610 000, and can be used in JexoSim 2.0 by selecting for a user-supplied stellar file as described above[8]. After selecting the star and planet spectra, the code must rebin these to a common wavelength grid (that of the spectrum with the lower resolution) to allow for array multiplication (Fig. 3). Thus, if the initial planet spectrum is provided at much lower R than the stellar spectrum, there will be some inevitable loss of fidelity (re: the RLB effect) due to rebinning at this stage. This can be minimized by providing both planet and star spectra at high resolution.

As an alternative to the user providing a specific planet spectrum as input, in JexoSim 2.0 we have added the capability to use the fully scalable ATMO model grid of transmission spectra produced by Goyal et al. (2018)[9] for gaseous planets, as an option for the input planet spectrum. Alternately the user can select a 'simple' flat planet transmission spectrum based on $(R_\mathrm{p}/R_\mathrm{s})^2$ or a blackbody-based emission spectrum.

2.2.2 Backgrounds

JexoSim 2.0 uses the same zodiacal light and optical surfaces emission models developed for the previous version (Sarkar et al. 2019). In addition to these, JexoSim 2.0 now includes the thermal emission from the sunshield as a background source (with an associated Poisson noise contribution). We utilize the publicly available plot of equivalent in-field sky irradiance versus wavelength for the sunshield from the JWST User Documentation Background Model webpage[10]. This is sampled using WebPlotDigitizer (Rohatgi 2020) and converted from MJy/sr to W/m^2/µm/sr. A seventh-order polynomial is fitted to the sampled points. This polynomial is used to obtain the spectral flux density at the wavelength solution of the instrument above 7 µm. Below 7 µm, the flux is assumed to be negligible and set to zero. It is then processed in the same way as the zodi background, described in Sarkar et al. (2019). This results in a 1D array of photon count-rate per X pixel, $C_\mathrm{sunshade}(X)$, which is applied to the image time-series as a background and contributes to the total Poisson noise (Fig. 3). The list of current backgrounds is given in Table 1.

2.2.3 Noise and quantum yield

Noise sources in the original version of JexoSim (Sarkar et al. 2019) included: star (source photon noise), dark current, read-out noise, zodiacal reflected light and thermal emission, optical surfaces emission and pointing jitter. In JexoSim 2.0, we have added Poisson noise from the sunshade background and Fano noise (Fig. 3). Table 1 lists the current noise sources simulated.

Table 1. Summary of noise sources, backgrounds, systematics and instrumental effects included in JexoSim 2.0.

Effect	Type	New in JexoSim 2.0
Noise sources	Star (source photon noise)	
	Dark current	
	Read-out noise	
	Zodiacal light	
	Optical surfaces emission	
	Sunshade emission	✓
	Fano noise	✓
	Pointing jitter	
Background sources	Dark current	
	Zodiacal light	
	Optical surfaces emission	
	Sunshade emission	✓
Systematics / instrumental effects	Optical transmissions	
	Wavelength solutions and spectral traces	
	PSF	
	PRNU	
	Intra-pixel sensitivity	
	Inter-pixel capacitance	✓
	Wavelength-dependent QE	
	Quantum yield	✓
	Pointing jitter	
	User-defined systematic	✓

Fano noise is a type of photon-transfer noise (Janesick 2007), which results from variation in the number of electron-hole pairs in the detector semiconductor material per absorbed photon. The standard deviation is given by

$$\sigma_f(\lambda) = \sqrt{F(\lambda)\eta(\lambda)}, \tag{1}$$

where F is the Fano factor, and η is the quantum yield. The Fano factor is the ratio of the variance in the number of electrons generated to the average number of electrons generated per photon. We use the two-level approximation (valid for $\eta \sim 1$ to 2) for the Fano factor from McCullough et al. (2008), where $F = [(3\eta - 2) - \eta^2]/\eta$. Since both η and F are wavelength-dependent, the Fano noise is also. For $\eta = 1$, we obtain a Fano noise of zero e$^-$. In the algorithm (Fig. 3), both stellar and background photons are added and Poisson variations to the photon numbers generated before the Fano noise is added. The wavelength-dependent quantum yield, $\eta(\lambda)$, is obtained for each instrument channel from the Pandeia data base. For the point source, we simply rebin this to the wavelength solution for each X pixel column, to give $\eta_*(X)$ for the stellar photons. For diffuse backgrounds, since a broad range of wavelengths will fall on each pixel (depending on the slit size used), a weighted quantum yield for each X pixel, $\eta_w(X)$ is calculated separately for each diffuse source. The quantum yield for each pixel on each subintegration, $\eta(X, Y, t)$, is found from the weighted average of $\eta_*(X)$ and $\eta_w(X)$ for each diffuse source, where the weights are the relative photon contributions on the pixel. For computational efficiency this weighted quantum yield is multiplied into all the signal and noise photons at a single step in the algorithm (Fig. 3). The Fano noise is then simulated. For each pixel on each subintegration, $\sigma_F(X, Y, t) = \sqrt{SF\eta(X, Y, t)}$, where S is the number of photons on a pixel. This is used to apply a random Gaussian variation in the electron count. For computation efficiency, the dark current is added prior to the generation of Poisson noise (so that Poisson noise generation is performed once), and thus its contribution to the signal and Poisson noise is subtracted prior to applying quantum yield and Fano noise (since dark current does not

[7] https://phoenix.astro.physik.uni-goettingen.de/?page_id=15

[8] These need to be first repackaged as text files with the first column giving the wavelength in µm, and the second column the stellar flux density in W/m^2/µm. The wavelength grid of these files extends from 0.5 to 5.5 µm. JexoSim will return zero value for the stellar flux density outside this range, and thus these spectra are not suitable for MIRI LRS.

[9] https://drive.google.com/drive/folders/1ZFbkPdqg37_Om7ECSspSpEp5QrUMfA9J

[10] Fig. 2 from https://jwst-docs.stsci.edu/jwst-observatory-functionality/jwst-background-model.

result from interacting photons), with the contributions added back after these steps.

2.2.4 Systematics and instrumental effects

The list of systematics and instrumental effects currently included are given in Table 1. As in the previous version, photo-response non-uniformity (PRNU) is included as a grid of pixel gain variations. Intra-pixel sensitivity variations can also be modelled using an intra-pixel response function described in Sarkar et al. (2019). In the new version, we have added IPC (Giardino et al. 2012) for NIRISS, NIRCam, and NIRSpec detectors. This is applied as a 2D convolution in the algorithm (Fig. 3). JexoSim 2.0 allows for the application of an externally generated user-defined systematic grid, as a 2D array that modulates the signal in time and wavelength dimensions. To use this function, inside the `JexoSim/data/systematic_models/` folder must be placed a folder named for the specific systematic that contains text files for the systematic grid (containing the proportional change in signal), wavelength, and time. The input parameters file option (Table 2) `sim_use_systematic_model` must be set to 1, and the name of the systematic model folder entered under `sim_systematic_model_name`.

There may be additional noise sources and systematics not yet represented in JexoSim. Notably we have not yet added the spatially correlated read noise effect for NIR detectors (Rauscher et al. 2017). However, JexoSim has a versatile framework where noise sources and systematics can be added in the future, especially as our understanding of the instruments improve during operations. Astrophysical effects such as the impact of star-spot effects, stellar pulsation and granulation, transits of exomoons or ring systems, and phase-dependent modulations can also be investigated using JexoSim by interfacing with external models directly or through the generation of time- and wavelength-dependent flux variation grids from such models which can then be used to modulate JexoSim light curves.

2.3 Dependences

To reduce issues during installation and allow for better future upkeep, we have changed and reduced the dependences in JexoSim 2.0. The main dependent packages are: PyTransit (Parviainen 2015), Astropy (Astropy Collaboration 2013, 2018), Scipy (Virtanen et al. 2020), Numpy (Harris et al. 2020), Pandas (Wes McKinney 2010), Numba (Lam, Pitrou & Seibert 2015), and Matplotlib (Hunter 2007). A dedicated set-up process has been developed which imports each of these together with any sub-dependences. A number of publicly available data bases are used with JexoSim 2.0. As in the previous version, the Pandeia data base (Pontoppidan et al. 2016) is used to obtain transmissions and wavelength solutions for each instrument, and a Phoenix BT-Settl data base is used for model star spectra (Allard, Homeier & Freytag 2012)[11]. We also use values from Pandeia for the instrument dark current, read noise, pixel full-well capacity values, IPC kernels, and quantum yield. Other instrument parameter values are obtained from Ressler et al. (2015) and Space Telescope Science Institute (2018a, b, c, d). As previously, WebbPSF (Perrin et al. 2012) is used to generate the point spread functions (PSFs) for each instrument, and limb-darkening coefficients (LDCs) are generated using ExoTETHyS (Morello et al. 2020). Both the PSFs and LDCs are pre-calculated. New in JexoSim 2.0 is the use of the NASA Exoplanet Archive[1] 'Planetary Systems' table as the data base for extra-Solar system parameters. Since this table provides results from multiple studies for a given planet, JexoSim first chooses values from the study which has the most complete set of parameter data. Remaining missing parameters are then selected from studies in order of most recent publication date. In rare cases where a parameter is completely missing, the user will be prompted to enter this manually (this only needs to be done once since the resulting exosystem XML file will be stored for later use).

2.4 Optimization

The simulation of pointing jitter described in Sarkar et al. (2019) represented one of the slowest sections of the JexoSim algorithm, requiring repetitive sequential shifting of the image array at a subpixel level. In the previous version, we used the `scipy.weave` function to interface with a section of code written in C that performed these shifts and improved speed. However, with the change to PYTHON 3, this function can no longer be used. We use instead the PYTHON optimization package `Numba`, combined with its own parallelization function for this part of the algorithm. This combination reduces jitter code processing times by an order of magnitude compared to standard PYTHON. In addition, we implement splitting the full image array into longitudinal sections, jittering each section (with the same jitter sequence) and then recombining the sections. This leads to an additional speed improvement by reducing the array size used in the jitter function.

In some instrument modes, it is possible to 'crop' the image array in the 'Noise' module (Fig. 2) to a smaller size, reducing the overall size of the final time-series array while preserving the full spectrum. The backgrounds are sampled at the edges of the image prior to cropping and later used for background subtraction in the data pipeline. Reducing the array size improves speed of the simulation and pipeline processing. Some simulations may produce very large numbers of NDRs (e.g. many thousands) reducing speed of the simulation and potentially overwhelming computer memory. This is particularly an issue for bright targets that have short integration times, targets with long overall observing times, and instrument modes with short frame times.

JexoSim 2.0 currently simulates only 'fast' read out modes where the number of frames per group (m) equals 1: MIRI (FAST), NIRSpec (NRSRAPID), NIRISS (NISRAPID), and NIRCam (RAPID). For NIRCam, 1-output and 4-output readout patterns are offered, each of which has a different frame time. JexoSim can operate using either correlated-double-sampling (CDS) or up-the-ramp (UTR) modes. In the latter, a slope is fitted in the pipeline to the NDR ramp in each integration cycle. This reduces read noise compared to CDS, but slightly increases Poisson noise (Rauscher et al. 2007). The user can choose to simulate the ramps by generating every NDR directly. However, to increase speed and reduce memory pressure, an alternative is offered where only the first and last NDRs are actually generated per integration cycle but the noise is adjusted so that when CDS (last-minus-first subtraction) is performed, the resulting Poisson noise and read noise are at the level expected for a ramp with n NDRs, where n is the number of groups per integration. This is achieved by adjusting the total Poisson noise (and Fano noise) on each NDR by a factor of $\sqrt{6(n^2+1)/5n(n+1)}$, and the read noise by a factor of $\sqrt{6(n-1)/n(n+1)}$ (Fig. 3). We use this alternative mode in all the simulation examples given in this paper.

For full transit simulations where the number of NDRs is very large (many thousands), potential exists to overwhelm computer

[11] https://Phoenix.ens-lyon.fr/Grids/BT-Settl/CIFIST2011_2015/FITS/

Table 2. Input parameters file options.

Group	Entry	Description	Options
Simulation	`sim_diagnostics`	Display of diagnostic data	0 or 1
	`sim_mode`	Type of simulation recipe	1: Out-of-transit with Allan analysis. 2: Full transit. 3: Noise budget. 4: Out-of-transit without Allan analysis.
	`sim_output_type`	Format of the output	1: Pipeline processed. 2: FITS file only. 3: Pipeline processed with intermediate products.
	`sim_noise_source`	Choice of noise source	Enter number from provided list corresponding to noise source (e.g. 0 activates all noise sources).
	`sim_realisations`	Number of realizations	Enter integer number of realizations.
	`sim_full_ramps`	Choice of CDS or UTR	0: CDS, 1: UTR.
	`sim_use_UTR_noise_correction`	Apply noise correction for UTR	If `sim_full_ramps` = 0, choose 0 or 1.
	`sim_pointing_psd`	Type of pointing jitter PSD	'flat': white noise PSD, 'non-flat': Herschel-derived PSD[a].
	`sim_prnu_rms`	rms of PRNU grid (per cent)	Enter number. Typical value is 3 per cent.
	`sim_flat_field_uncert`	Uncertainty on flat-field (per cent)	Enter number. Typical value is 0.5 per cent.
	`sim_use_ipc`	Apply IPC	0 or 1. Not applied in MIRI.
	`sim_use_systematic_model`	Apply systematic model	0 or 1.
	`sim_systematic_model_name`	Name of the systematic model	Enter the name of the folder containing the model files
Observation	`obs_type`	Observation type	1: Primary transit. 2: Secondary eclipse.
	`obs_frac_T14_pre_transit`	Pre-transit T_{14} fraction	Enter fraction of transit duration pre-transit (e.g. 0.5).
	`obs_frac_T14_post_transit`	Post-transit T_{14} fraction	Enter fraction of transit duration post-transit (e.g. 0.5).
	`obs_use_sat`	Use saturation time to set n	0 or 1.
	`obs_fw_percent`	Saturation limit	Enter per cent of full well (e.g. 80 or 100) as the saturation limit.
	`obs_n_groups`	Set number of groups (n)	If `obs_use_sat` = 0, enter integer number of groups.
	`obs_n_reset_groups`	Set number of reset groups	Enter 'default' for default value or integer number of groups[b].
	`obs_inst_config`	Set instrument configuration	Choose from list of configurations with subarray modes.
Pipeline	`pipeline_binning`	Type of spectral binning	'R-bin': uses R. 'fixed-bin': uses number of pixel columns.
	`pipeline_R`	R-power for binning	If 'R-bin' used enter R-power (e.g. 100).
	`pipeline_bin_size`	Pixel columns per bin	If 'fixed-bin' used enter number of pixel columns (e.g. 5 or 10).
	`pipeline_ap_shape`	Extraction aperture shape	'rect': rectangular. 'wav': wavelength-dependent.
	`pipeline_apply_mask`	Apply extraction aperture	0 or 1.
	`pipeline_auto_ap`	Autoselect width of aperture	0 or 1.
	`pipeline_ap_factor`	Aperture width	Enter number α where width $= 2\alpha F\lambda$ [c].
	`pipeline_bad_corr`	Manage bad pixels	0 or 1.
Exosystem	`planet_name`	Planet name	Enter planet to search for in planet data base.
	`planet_use_database`	Use planet data base	1: Planet data base gives all exosystem parameters. 0: Exosystem parameters entered manually[d].
	`planet_spectrum_model`	Type of planet spectrum	'simple': uses flat transmission spectrum based on $(R_{\rm p}/R_{\rm s})^2$ or blackbody-based emission spectrum. 'complex': uses ATMO grid[e]. 'file': uses a specific spectrum from a file.
	`planet_spectrum_file`	Path to planet spectrum file	If `planet_spectrum_model` = 'file' provide path to file.
	`star_spectrum_model`	Type of stellar spectrum	'simple': uses blackbody. 'complex': uses Phoenix data base.
	`star_spectrum_file`	Path to star spectrum file	If `star_spectrum_model` = 'file' provide path to file.
	`star_spectrum_mag_norm`	Normalize magnitude	0: None. 1 normalises to J and/or K mag. 2: uses $(R_{\rm s}/d)^2$

[a] PSD = power spectral density profile. Both PSDs generate jitter timelines with an rms of 6.7 mas per 1D axis. The Herschel-derived PSD is scaled version of the PSD obtained using Herschel pointing information described in Sarkar et al. (2019).

[b] Default values are 0 for MIRI (assuming a read-reset method), and 1 for all NIR channels.

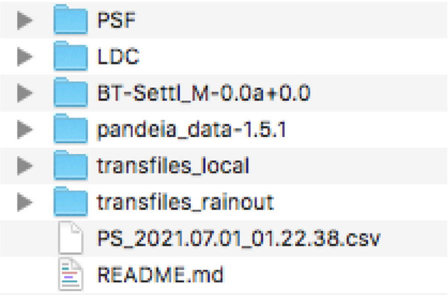
[c] F is the F-number and λ is the wavelength. For rectangular apertures λ is the longest wavelength in the band.

[d] There is a section to enter these system parameters manually.

[e] If a 'complex' planet spectrum is chosen, a section exists to fill in required parameters (e.g. C/O ratio, haze, cloud, etc.).

memory and slow simulation and pipeline running time due to the very large array size produced. To mitigate this, when the number of NDRs becomes very large, the timeline of images is split into consecutive segments. Each segment is processed through stage 1 of the pipeline (Fig. 2) producing spectrally binned light-curve segments that are later joined together to produce the final light curves. These are then processed in stage 2 of the pipeline where light-curve model fitting takes place to extract the planet spectrum.

To exclude background pixels which may contribute to noise, an extraction aperture (aperture mask) is applied over the integration image in the pipeline. However, the aperture may also reduce signal and increase spatial jitter noise if too narrow. We have added a new function that autoselects an optimal aperture width which maximizes the stellar SNR. If this option is selected, the code will rapidly trial different aperture widths on a sample segment of the image time-series, and then select the width that returns the highest average SNR over all wavelengths.



PSF
LDC
BT-Settl_M-0.0a+0.0
pandeia_data-1.5.1
transfiles_local
transfiles_rainout
PS_2021.07.01_01.22.38.csv
README.md

Figure 4. JexoSim `archive/` folder displaying the expected contents.

3 IMPLEMENTATION

3.1 Installation

JexoSim 2.0 is hosted on Github and is installed as follows. First, a dedicated virtual environment is recommended to prevent package conflicts. For stability, a number of dependent packages are installed during the set-up of the virtual environment. For `conda` users the following commands set-up the environment (with the initial set of dependences) and activate it.

```
   $ conda create -n jexosim python=3.8.5
matplotlib=3.3.1-0 setuptools=49.6.0
numpy=1.19.1 numba=0.50.1
   $ source activate jexosim
```

The activation command might vary with the system. Next, the Github repository must be cloned.

```
   $ git clone https://github.com/subisarkar/
JexoSim.git
```

Next, the data bases described in Section 2.3 need to be downloaded. These are Pandeia,[12] the Phoenix BT-Settl model grid[11], and the ATMO transmission spectra grid[13,14] (the latter is optional). The pre-calculated PSF and LDC folders must also be downloaded and links are provided on the Github 'README' page. The above folders need to be unzipped and moved to inside the `JexoSim/archive/` directory. Next, from the NASA Exoplanet Archive[1], the 'Planetery Systems' table (all rows, all columns) should be downloaded in .csv format, and moved into the `JexoSim/archive` directory. The final contents of the `JexoSim/archive/` folder should look similar to Fig. 4[15]. Next, from within the `JexoSim/` folder the user should run the following commands to install JexoSim (and remaining dependences) followed by the generation of JexoSim-compatible transmission files, wavelength solutions, IPC kernels and quantum yields from the Pandeia data base.

```
   $ cd JexoSim
   $ python setup.py install
   $ python generate.py
```

Finally, the user must decide on a directory location for the final results. By default these will be placed in the `JexoSim/output/` folder; however, the user may choose a different location by editing the 'paths file', `jexosim_paths.txt`, which is located in the `JexoSim/jexosim/input_files/` folder. The complete path to the output folder can be entered next to the `output_directory` option.

3.2 Running a simulation

To run a simulation, the user should navigate to within the `JexoSim/` folder and run the `run_jexosim.py` file with an input parameters file name (e.g. `jexosim_input_params_ex1.txt` as shown below) as the argument. If the user has chosen to apply the pipeline and obtain processed results, as mentioned previously these are packaged into a dictionary and dumped in .pickle format in the output directory (Fig. 2) with an associated text file. The files are identified with the type of simulation, the instrument configuration, the planet, and a time stamp. To display a set of plots showing results from the given .pickle file, the `results.py` file is run from within the `JexoSim/` folder with the filename as an argument. For example:

```
   $ cd JexoSim
   $ python results.py Noise_budget_MIRI_LRS_
slitless_SLITLESSPRISM_FAST_HD_209458_b_2021_
07_01_1005_28.pickle
```

If the user chooses to simply store the image stack for later analysis without running the pipeline, a FITS file is generated and stored in the output directory (Fig. 2).

3.3 Input parameters file options

In Table 2, we list the main options in the input parameters file, which provides the interface between the user and the code. These are divided into four categories: 'Simulation' (parameters that control the functioning of the simulation, but would not be options for a real observation), 'Observation' (parameters that control the simulated observation and could also be options for actual observations), 'Pipeline' (parameters controlling the pipeline), and 'Exosystem' (parameters for defining the extra-Solar system). As JexoSim evolves, these options may be modified in future.

4 CASE STUDIES

We now describe five case studies using JexoSim 2.0. These demonstrate the range of instrument models (NIRISS, NIRCam, NIRSpec, and MIRI), different simulation modes (out-of-transit versus full transit) and different methods to obtain the transit depth precision (Allan deviation analysis versus Monte Carlo). We use two end-member targets: the hot Jupiter, HD 209458 b, orbiting a bright (J mag 6.59) Sun-like star, and the mini-Neptune, K2-18 b, orbiting an M-dwarf of intermediate brightness (J mag 9.76). We simulate mostly primary transits, but also consider a secondary eclipse case. Finally, a 'noise budget' simulation is performed for MIRI LRS, to demonstrate how the total noise can be broken down to show the contribution of each noise source.

Table 3 summarizes the major and minor points illustrated by each case study. Each case study can be run by the user by implementing

[12] https://stsci.app.box.com/v/pandeia-refdata-v1p5p1/

[13] https://drive.google.com/file/d/1Kvfi7FTBqM1MfnkTnHsJJqI7EFwnuaIG/view

[14] https://drive.google.com/file/d/1LnpL_kpbZGPEN0G6QhHrL4DcD4hwvaia/view

[15] The exact name of the .csv file will vary based on the download date and time but the code will still recognize this as the planetary data base. Similarly, Pandeia will be recognized even if another version number is used.

Table 3. Summary of major and minor points of each case study.

Case	File	Major points	Minor points
1	*ex1.txt	Produces NIR transmission spectrum of K2-18b Uses OOT simulation mode Demonstrates NIRISS instrument model Uses Allan deviation method to obtain error bars All noise and background sources applied Post-processed results via JexoSim pipeline	NIRISS subarray mode selection Simulation/pipeline settings, e.g.: autocalculation of *n* groups, PRNU level, type of jitter PSD, *R*-binning, autosizing of extraction aperture, etc.
2	*ex2.txt	Produces NIR/MIR transmission spectrum of K2-18b Uses full-transit simulation mode Demonstrates NIRCam and MIRI instrument models Uses Monte Carlo method to obtain error bars All noise and background sources applied Post-processed results via JexoSim pipeline	NIRCam and MIRI LRS subarray mode selection Simulation/pipeline settings, e.g.: pre- and post-transit observation time Combines spectra from several instrument modes
3	*ex3.txt	Produces NIR transmission spectrum of K2-18b Produces NIR transmission spectrum of HD 209458 b Uses full-transit simulation mode Demonstrates NIRSpec instrument model Uses Monte Carlo method to obtain error bars All noise and background sources applied Post-processed results via JexoSim pipeline Saturation affecting subarray choices Comparison of hot Jupiter / sub-Neptune spectrum SNR	NIRSpec subarray mode selection Simulation/pipeline settings, e.g.: number of realizations set to 50 Combines spectra from several subarray modes
4	*ex4.txt	Produces NIR/MIR emission spectrum of HD 209458 b Uses full-eclipse simulation mode Demonstrates NIRCam and MIRI instrument models Uses Monte Carlo method to obtain error bars All noise and background sources applied Post-processed results via JexoSim pipeline	Uses same subarray modes as in case 2 Simulation/pipeline settings, e.g. selection of eclipse over transit observation Combines spectra from several instrument modes
5	*ex5.txt	Produces noise budget for MIRI K2-18b observation Uses Allan deviation method to find fractional noise Significant background and read noise at longer λ	Uses LRS SLITLESSPRISM subarray mode Simulation of multiple noise sources in isolation Simulation/pipeline settings, e.g. user-defined aperture size (autosizing disabled)

*Prefix: `jexosim_input_params_`

the input parameters files named in Table 3 and provided with JexoSim 2.0[16] with adjustments described in the text below.

Table 4 lists the simulation settings for each case study referring to the input parameters file entries given in Table 2.

In all cases, except Case 5 (noise budget), the full suite of noise sources, backgrounds, and systematics listed in Table 1 are used (excluding the 'user-defined systematic'). For the noise budget case, noise sources are implemented individually as explained further below. The JexoSim pipeline (Fig. 2) is used in all cases to obtain the final results. The results are therefore post-processed, including noise mitigation methods (e.g. aperture masking during 1D spectra extraction), subtraction of backgrounds, and decorrelation of correlated noise sources (e.g. pointing jitter noise).

4.1 Case 1: transmission spectrum of K2-18 b with NIRISS using Allan deviation method

4.1.1 Overview

For this case study, we simulate NIRISS SOSS (single object slitless spectroscopy) with the GR700XD grism and the SUBSTRIP96 subarray in NISRAPID read out mode. K2-18 b, is selected as the target, with all parameters obtained via the planet data base. The goal is to obtain the precision on the transit depth at a binned *R*-power of 100 and produce example transmission spectra with error bars. We utilize an out-of-transit simulation approach (i.e. no transit light curves implemented) and Allan deviation analysis of the results to obtain the precision. The advantages and disadvantages of this approach are discussed below.

4.1.2 Allan deviation analysis

Allan deviation analysis (Allan 1966) accounts for the effect of correlated noise on the standard error of a mean signal. This is achieved by taking a time-dependent signal and binning it into sequential segments of duration τ, then taking the mean in each segment, and measuring the standard error of the mean values. For uncorrelated ('white') noise, the fractional noise (the standard error of the mean/mean signal), falls predictably with τ with a power-law exponent of –0.5. This is the basis of 'integrating down' photon noise. However, for correlated noise, the exponent may be different and itself change with τ. At large enough values of τ the correlation time-scale is exceeded and the noise integrates down like 'white' noise. In Sarkar et al. (2019), we described in detail how JexoSim performs this kind of analysis using simulated out-of-transit spectrally binned timelines of integration signals. The outcome of this analysis is that the fractional noise, σ_s, can be predicted for any value of τ, taking into account any correlated noise. If τ is set to T_{14}, the transit duration, then assuming a simple 'box-car' model of a transit, with an equal

[16] These input parameters files are located in the folder `JexoSim/jexosim/input_files/`.

Table 4. Simulation settings per case study. See Table 2 for description of each entry. Some entries are left blank indicating they do not need to be filled in for these simulations.

Group	Entry	Case 1	Case 2	Case 3	Case 4	Case 5
Simulation	sim_diagnostics	0	0	0	0	0
	sim_mode	1	2	2	2	3
	sim_output_type	1	1	1	1	1
	sim_noise_source	0	0	0	0	
	sim_realisations	5	50	50	50	10
	sim_full_ramps	0	0	0	0	0
	sim_use_UTR_noise_correction	1	1	1	1	1
	sim_pointing_psd	'flat'	'flat'	'flat'	'flat'	'flat'
	sim_prnu_rms	3	3	3	3	3
	sim_flat_field_uncert	0.5	0.5	0.5	0.5	0.5
	sim_use_ipc	1	1	1	1	1
	sim_use_systematic_model	0	0	0	0	0
	sim_systematic_model_name					
Observation	obs_type	1	1	1	2	1
	obs_frac_T14_pre_transit		0.5	0.5	0.5	
	obs_frac_T14_post_transit		0.5	0.5	0.5	
	obs_use_sat	1	1	1	1	1
	obs_fw_percent	80	80	80	80	80
	obs_n_groups					
	obs_n_reset_groups	'default'	'default'	'default'	'default'	'default'
	obs_inst_config[a]	'NIRISS + SOSS_GR700XD + SUBSTRIP96 + NISRAPID'	'NIRCam + TSGRISM_F322W2 + SUBGRISM64_4_output + RAPID'	'NIRSpec BOTS_G395H_F290LP + SUB2048 + NRSRAPID'	'MIRI + LRS_slitless + SLITLESSPRISM + FAST'	'MIRI + LRS_slitless + SLITLESSPRISM + FAST'
Pipeline	pipeline_binning	'R-bin'	'R-bin'	'R-bin'	'R-bin'	'R-bin'
	pipeline_R	100	100	100	100	100
	pipeline_bin_size					
	pipeline_ap_shape	'rect'	'rect'	'rect'	'rect'	'rect'
	pipeline_apply_mask	1	1	1	1	1
	pipeline_auto_ap	1	1	1	1	0
	pipeline_ap_factor					1
	pipeline_bad_corr	1	1	1	1	1
Exosystem	planet_name[b]	'K2-18 b'	'K2-18 b'	'HD 209458 b'	'HD 209458 b'	'K2-18 b'
	planet_use_database	1	1	1	1	1
	planet_spectrum_model	'file'	'file'	'file'	'file'	'file'
	planet_spectrum_file	path given	path given	path given	path given	path given
	star_spectrum_model	'complex'	'complex'	'complex'	'complex'	'complex'
	star_spectrum_file					
	star_spectrum_mag_norm	1	1	1	1	1

[a] For cases 2–4, this entry is changed as required for the instrument configuration being simulated.

[b] In case 3 'HD 209458 b' is also used.

amount of out-of-transit and in-transit time, the noise on the transit depth can be approximated by $\sqrt{2}\sigma_s$.

This approach partially accounts for correlated noise (by finding the standard error on the mean in- and out-of-transit signals per spectral bin), but assumes the in- and out-of-transit light-curve sections are uncorrelated to each other. It also does not account for limb-darkening on the light curve, and cannot be used to assess systematic biases distorting the light curve, such as spot occultations or persistence. Full transit simulations using the Monte Carlo approach described in the next case study can capture these effects. However, for the assessment of transit depth precision in the absence of such light-curve distortions, this approach is adequate and delivers equivalent results to the Monte Carlo approach.

4.1.3 Simulation settings

Key settings in the input parameters file for this case are shown in Table 4. The option sim_mode is set to 1 choosing an OOT simulation with Allan analysis and sim_output_type set to 1, selecting pipeline processed results. The jitter option sim_pointing_psd is set to 'flat' selecting a white noise power spectrum generating a jitter timeline with rms 6.7 mas. The PRNU is set to 3 per cent rms with a flat-field uncertainty of 0.5 per cent. The number of groups per integration is set by the saturation time which is the time taken for the first pixel on the detector to reach the saturation limit. This time is then divided by the group time (same as frame time when $m = 1$) to obtain the integer, n, the number of groups per integration. 80 per cent of the pixel full-well is chosen as the saturation limit. In this case, n was calculated to be 13. For speed we implement CDS rather than full ramps but apply the UTR noise correction that adjusts the Poisson and read noise to the levels expected for the number of groups per integration.

The option planet_spectrum_model is set to 'file' so that the transmission spectrum is obtained from a specific file generated beforehand for this planet using a radiative transfer code. In our examples, we use models based on published works for each planet considered. We use a model transmission spectrum of HD 209458 b based on Welbanks & Madhusudhan (2019) and that of K2-18 b based on Madhusudhan et al. (2020). The pipeline option, `pipeline_binning` is set to 'R-bin' for spectral binning based on spectral resolving power, with the R-power set to 100. `pipeline_ap_shape` is set to 'rect' choosing a rectangular extraction aperture. Since in NIRISS SOSS the spectral trace is curved, choosing this latter option does not actually produce a rectangular-shaped aperture. Rather the aperture is curved to match the shape of the spectrum itself but the width of the aperture remains constant with wavelength. By setting `pipeline_auto_ap` to 1, the width of the aperture is autoselected by the code to optimize SNR. We use the average of five realizations to obtain the final noise results, by setting `sim_realisations` to 5[17].

4.1.4 Results

Fig. 5 shows the results from this case study. Fig. 5(a) shows an example integration image taken just before the jitter decorrelation step in the pipeline (Fig. 2). We see the curved spectral trace in this channel and the unusual PSF, elongated in the cross-dispersion direction by a cylindrical lens incorporated into the GR700XD optical element. Fig. 5(b) shows the fractional noise at T_{14} as a function of wavelength derived from the Allan deviation analysis. The average result is given by the large grey dots, with the small dots showing the individual realizations. The average value is used to then calculate the noise on the transit depth per spectral bin (Fig. 5c, blue dots) by multiplying by $\sqrt{2}$. We fit a fourth-order polynomial to these points (Fig. 5c, blue line) to estimate the expected noise (1σ error) per spectral bin.

We find that at $R = 100$, the 1σ error given by the polynomial fit ranges from a minimum of 31 ppm (at 1.41 μm) to maximum of 83 ppm (at 2.79 μm), the noise rising at both the blue and red ends of the wavelength range. We then generate a transmission spectrum in Fig. 5(d). In each spectral bin, the 1σ error from 5(b) line fit is used to randomize the value of $(R_p/R_s)^2$ using a Gaussian probability distribution centred on the input spectrum binned down to $R = 100$. The input spectrum is the planet spectrum modulated by the instrument response: $p(X)$ in Fig. 3. The size of the error bars represents the 1σ margin of error. The upper plot in Fig. 5(d) gives a randomized spectrum for a single transit observation. The spectral features are noisy but large-scale variations are discernible over the noise, including the 1.4 μm water featured discovered using the Hubble WFC3 (Benneke et al. 2019; Tsiaras et al. 2019). A future spectral retrieval study can quantify observability more exactly. The lower plot shows a spectrum for 10 co-added transits where the 1σ error has been reduced by a factor of $1/\sqrt{10}$. We can see that with 10 transits the spectral features are well recovered with very little noise at the shorter wavelengths.

4.2 Case 2: transmission spectrum of K2-18 b with NIRCam and MIRI using Monte Carlo method

4.2.1 Overview

In this case study, we simulate the NIRCam Time-series Grism F444W and F322W2 modes observing K2-18 b in primary transit. In both cases, the SUBGRISM64 subarray is used with the four-output readout pattern option. We also simulate MIRI LRS with the SLITLESSPRISM subarray. For each instrument mode, full primary transit observations are modelled in a 50-realization Monte Carlo simulation to obtain the precision on the transit depth. The pipeline runs through stage 1 and stage 2 (light-curve fitting) (Fig. 2). This combination of instrument modes allows us to generate a simulated transmission spectrum from 2.4 to 12 μm.

4.2.2 Monte Carlo method

This method generates full primary transit or secondary eclipse simulations with implementation of a transit or eclipse light curve. In each realization, the image time-series is processed through stage 1 of the pipeline generating synthetic data light curves spectrally binned to R of 100. In stage 2 of the pipeline, these are fitted with model light curves and the transit depth, $(R_p/R_s)^2$ or eclipse depth, $F_p/(F_s + F_p)$, recovered for each spectral bin. The fitting uses a downhill simplex algorithm minimizing chi-squared. The planet–star radius ratio (or square root of the eclipse depth in secondary eclipse simulations) is the only free-parameter in this fit. After multiple realizations, a distribution of transit depths is obtained for each spectral bin. We take the standard deviation of this distribution to be the 1σ confidence interval margin-of-error on the transit depth, i.e. the size of the error bar on the final spectrum. The Monte Carlo approach gives robust results since the uncertainty on the transit depth is directly measured, accounting for all correlated and uncorrelated noise effects. Also, any residual systematic wavelength-dependent biases on the spectrum can be detected through deviation of the mean value in a systematic manner relative to the input spectrum. The latter, for example, may occur due to the presence of stars spots and faculae.

4.2.3 Simulation settings

Table 4 gives the key settings used in the input parameters file for this case study. These are mostly the same as used for case 1, with the following exceptions. `sim_mode` is set to 2, choosing a full transit observation, and `sim_realisations` is set to 50. Under `obs_inst_config`, the option for NIRCam F332W2 SUBGRISM64 with four-output readout pattern is shown; however, this is substituted when simulating the F444W and MIRI LRS modes. The fraction of T_{14} that is observed before transit (`obs_frac_T14_pre_transit`) and the fraction after transit (`obs_frac_T14_post_transit`) are both set to 0.5.

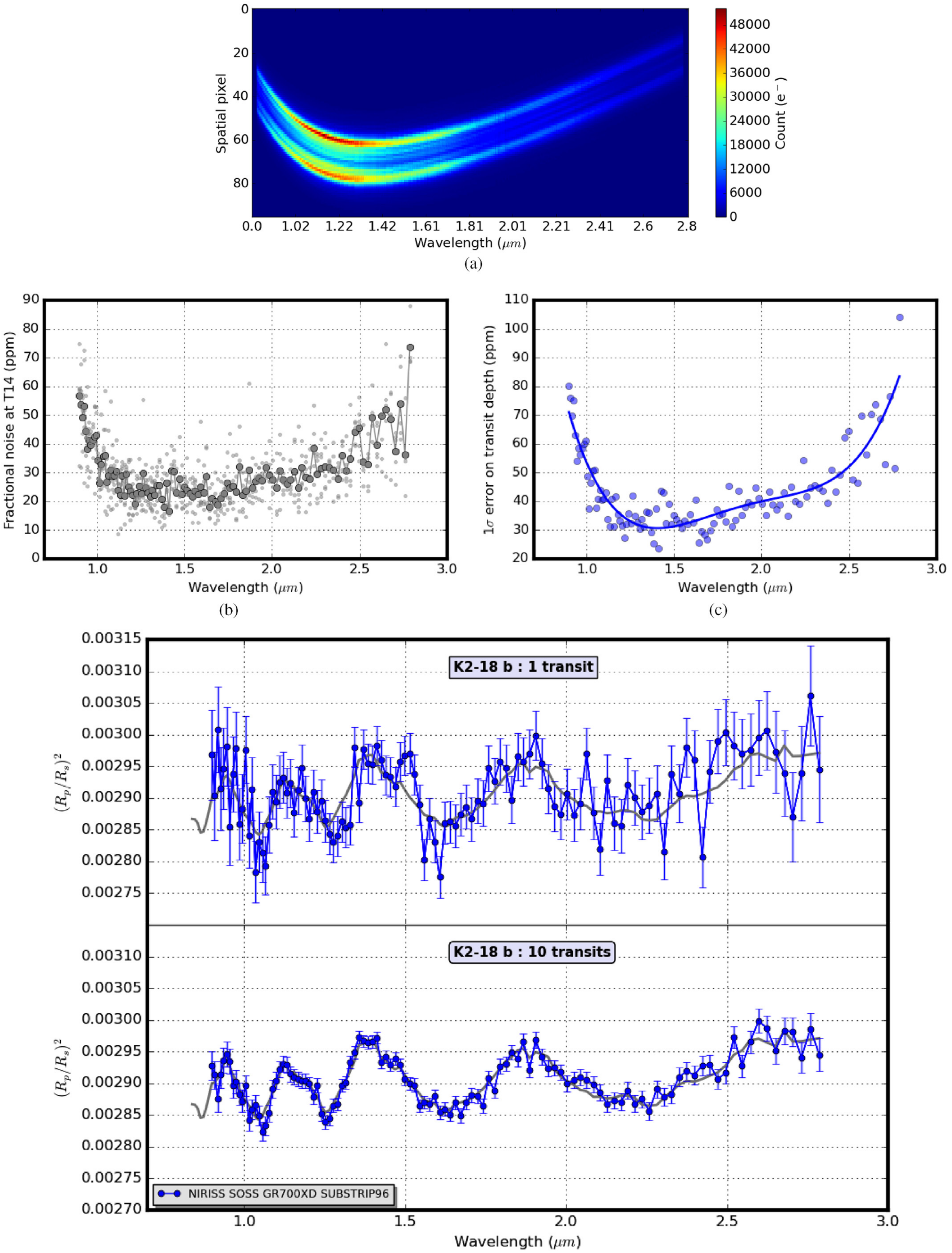
4.2.4 Results

Results from this case study are shown in Fig. 6. Example integration images are shown for each instrument configuration in Fig. 6(a). Due to cropping of the image in the 'Noise' module to improve simulation speed, the size of these images may not match the original subarray sizes. Fig. 6(b) shows the transit depths recovered from each of the 50 realizations (smaller dots). In each bin, a distribution of transit depth instances occurs. The mean value for the distribution in each spectral bin is shown (large dots). The input spectrum binned to $R = 100$ is shown as the grey line.

No discernible systematic bias is seen in the mean values. Fig. 6(c) shows the 1σ error versus wavelength, where each point is the standard deviation of transit depths from Fig. 6(b) for the given spectral bin. Polynomial fits to these points (second order for NIRCam and fourth order for MIRI) are shown. These line fits are used to predict the 1σ error for each wavelength bin for a single transit observation per instrument mode.

Based on these polynomial fits, the 1σ error in NIRCam ranges from 47 ppm (at 3.61 μm) to 65 ppm (at 2.45 μm) for the F332W2 mode, and from 49 ppm (at 3.94 μm) to 75 ppm (at 4.95 μm) for the

[17] One realization would suffice here but obtaining the average of many gives a less noisy final result.



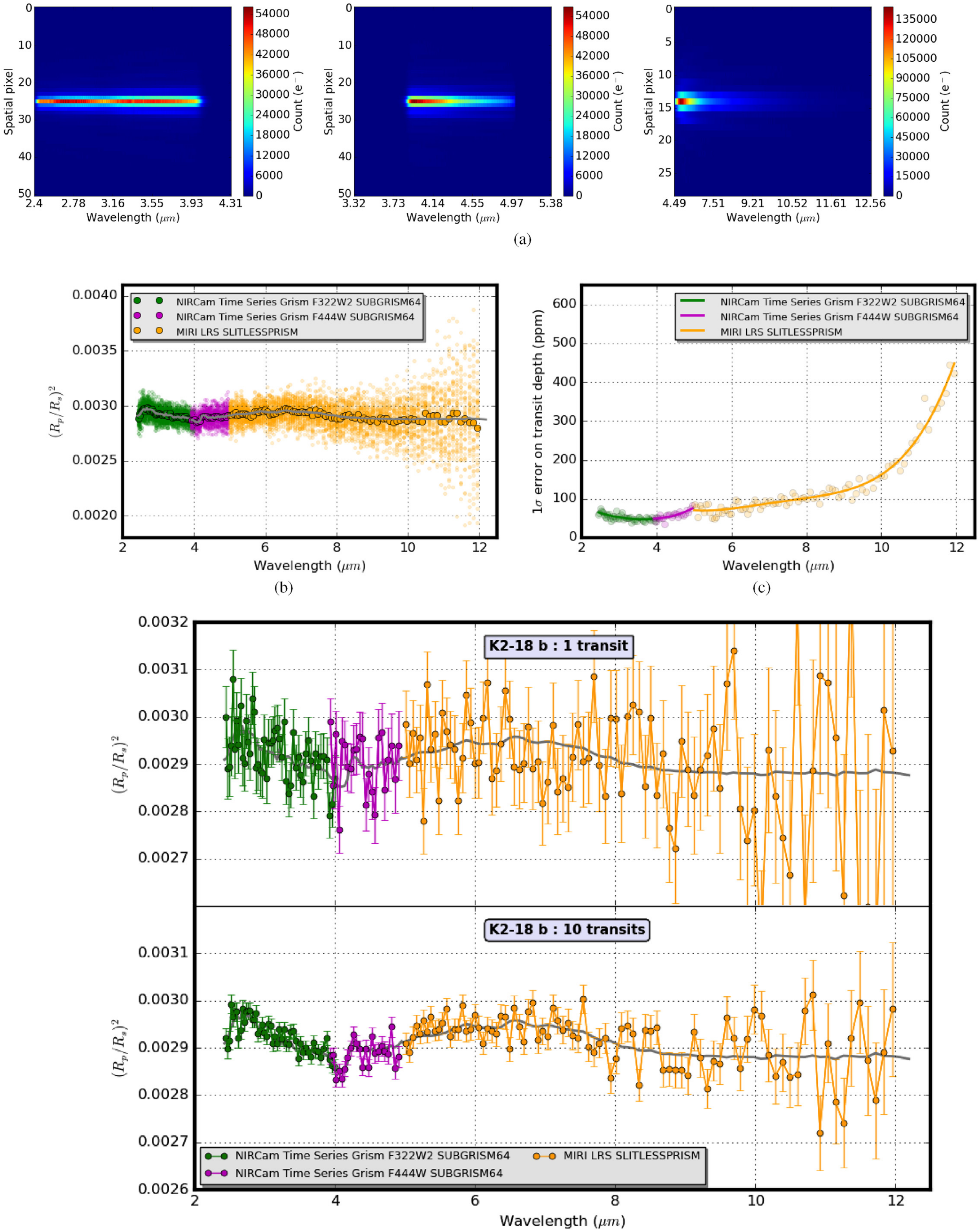
(a)

(b)

(c)

(d)

Figure 5. Case 1: simulation of transmission spectrum for K2-18 b with NIRISS SOSS GR700XD SUBSTRIP96 using the Allan deviation method. (a) Integration image from pipeline. (b) Fractional noise at T_{14} versus wavelength (small dots: individual realizations, large dots: average of 5 realizations). (c) 1σ error on transit depth versus wavelength with fourth-order polynomial line fit. (d) Synthetic post-processed transmission spectra at $R = 100$ for 1 transit and 10 co-added transits (grey line: input spectrum binned to $R = 100$).



(a)

(b)

(c)

(d)

Figure 6. Case 2: simulation of transmission spectrum for K2-18 b with NIRCam Time Series Grism F322W2 and F444W modes (both with SUBGRISM64) and MIRI LRS slitless mode using the Monte Carlo method. (a) Integration images from pipeline – left: NIRCam Grism F322W2, middle: NIRCam Grism F444W, right: MIRI LRS. (b) Monte Carlo simulation recovered transit depths versus wavelength (small dots: individual realizations, large dots: average of 50 realizations, grey line: input spectrum binned to $R = 100$). (c) 1σ error on transit depth versus wavelength with second and fourth-order polynomial line fits for NIRCam and MIRI respectively. (d) Synthetic post-processed transmission spectra for K2-18b for 1 transit and 10 co-added transits (per instrument mode) at $R = 100$ (grey line: input spectrum binned to $R = 100$).

F444W mode. Similarly, for MIRI LRS, the 1σ error ranges from a minimum of 70 ppm (at 5.26 μm) increasing substantially with wavelength beyond 10 μm to a maximum of 448 ppm (at 11.95 μm).

As in the previous case study, we generate example transmission spectra using these 1σ error values. Fig. 6(d) shows randomized transmission spectra for 1 and 10 transits. With one transit the spectrum is very noisy with features not readily discernible over noise. With 10 transits, the spectral features in both NIRCam and MIRI become discernible above noise upto about 8 μm.

4.3 Case 3: transmission spectra of HD 209458 b and K2-18 b with NIRSpec using Monte Carlo method

4.3.1 Overview

This case study simulates NIRSpec with a variety of subarray modes, using both medium- and high-resolution gratings. Simulated primary transit observations are performed on both HD 209458 b and K2-18 b. This example also illustrates the potential issues of detector saturation for bright targets with JWST requiring careful subarray selection, with different configurations chosen for each planet. We obtain the transit depth precision at $R = 100$ for each configuration and planet using the Monte Carlo method described in Section 4.2. Using a combination of instrument configurations, we generate post-processed transmission spectra with error bars for each planet from ~1–5 μm. These results highlight the comparative performance of JWST NIRSpec between a hot Jupiter with a bright host star and a mini-Neptune with a dim host star.

4.3.2 Simulation settings

Table 4 lists the input parameters file settings for this example. These mostly follow those of case 2. The entry for `obs_inst_config` is shown for the G395H grating with the F290LP filter and the SUB2048 subarray, but this is changed for other configurations as needed. The `planet_name` entry is changed between ‘HD 209458 b’ and ‘K2-18 b’.

4.3.3 NIRSpec configurations

NIRSpec is a complex instrument with several different dispersion elements, filter options, and subarray modes (Fig. 1). The choice of instrument configuration (grating, filter, and subarray) may be driven by the desired wavelength coverage and the wish to minimize the number of separate transit observations needed to cover this wavelength range. Each configuration requires an independent observation.

However, configuration choice must also take into account the feasibility of using each subarray with regards to pixel saturation. In particular ‘hard’ saturation must be avoided. This occurs when the saturation time (the time for the first pixel to reach the chosen saturation limit) is shorter than 2 frame times. 2 frames is the minimum needed to mitigate reset noise. Subarrays with shorter frame times are thus more likely to avoid hard saturation. As with the previous simulations, we use a saturation limit of 80 per cent full well (to minimize potential for non-linearity) and assume the NRSRAPID readout mode.

For both HD 209458 b and K2-18 b, JexoSim predicts NIRSpec PRISM will result in saturation with all subarray modes and is thus excluded in these cases. This is partly attributable to its very short spectral trace, concentrating spectral power on a small region of the detector, leading to high detector count rates and short saturation times.

The SUB512 and SUB512S subarrays have the shortest frame times and are thus least prone to saturation. However, as can be seen in Fig. 1, they cover only a fraction of the available wavelength range for a given dispersion element and so would need to be combined with other subarrays for wide wavelength coverage.

The high resolution gratings, which tend to disperse the spectral power over a greater physical distance compared to the medium resolution gratings, are better for avoiding saturation. For the brighter target, HD 209458 b, we found we could apply G395H using the SUB2048 subarray (thus capturing its full wavelength range) without saturation. For G295H, we found that SUB2048 leads to hard saturation, but subarrays SUB1024A and SUB1024B (which have shorter frame times) avoid this, and thus a combination of these two subarray modes is used. Finally for G140H, we found that subarrays SUB1024A and SUB1024B saturate, but subarray SUB512 is feasible with the F100LP filter giving coverage in two wavelength patches (1.07–1.19, 1.47–1.59 μm) (Fig. 1).

For K2-18 b, we found that all three medium resolution gratings (G140M, G235M, and G395M) could be used with the SUB2048 subarray (capturing the full wavelength range of each grating) without saturating. We chose the F100LP filter for use with G140M over the alternate F070LP filter as it avoids a gap in the wavelength coverage when combined with the other gratings (Fig. 1).

4.3.4 Results

Fig. 7 shows the results for this case study. The distributions and means of transit depth instances recovered from the Monte Carlo simulation are shown in Fig. 7(a). No discernible biases are seen when comparing the mean transit depths to the input spectrum. The standard deviations of the transit depths in each spectral bin are plotted in Fig. 7(b) with second-order polynomials fitted to the values.

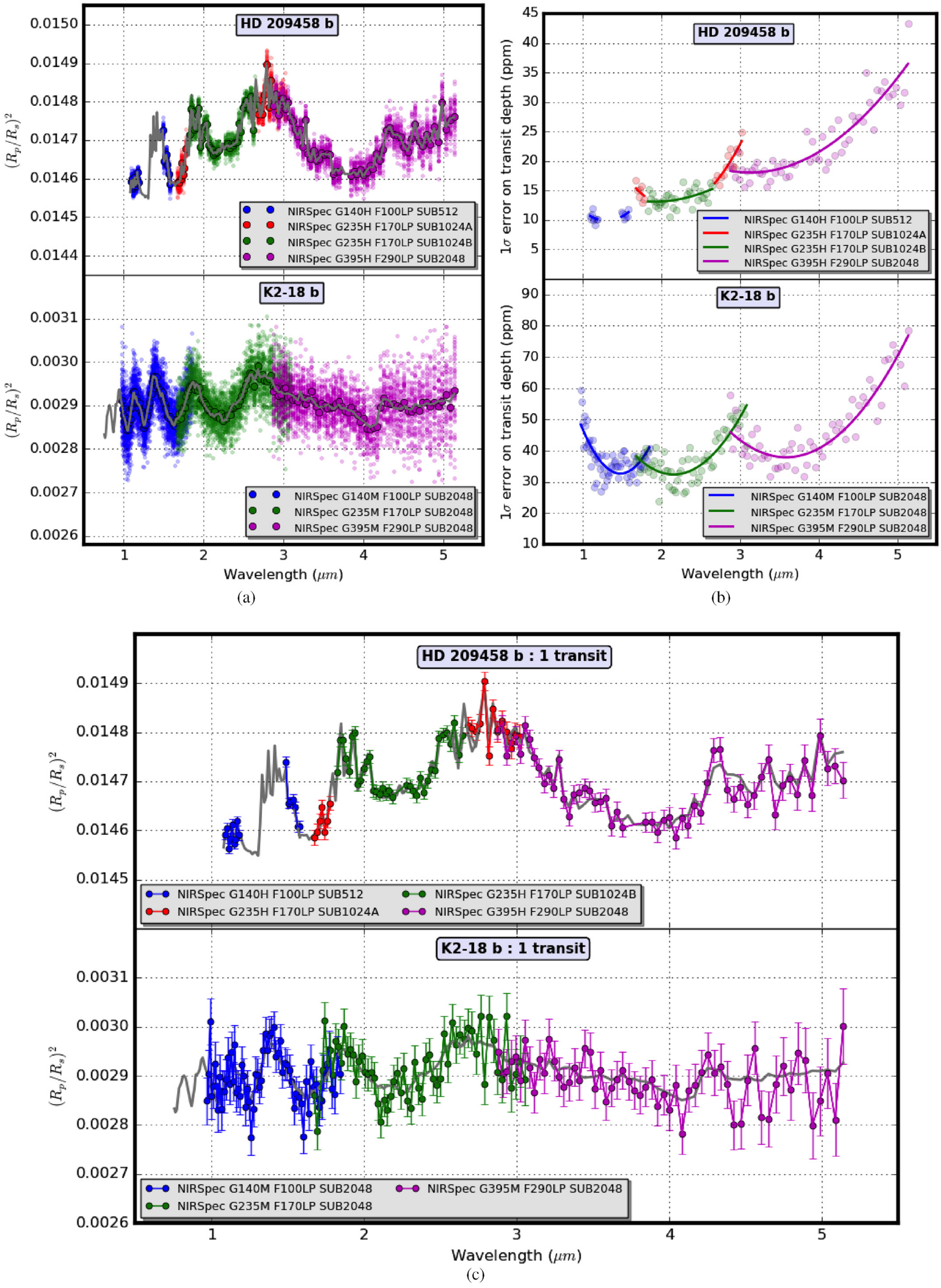
Using the polynomial values, for HD 209458 b the 1σ error on the transit depth across all the NIRSpec modes used ranges from 10 ppm (at 1.18 μm) to 36 ppm (at 5.14 μm). For K2-18 b, the 1σ error across the NIRSpec modes used ranges from 32 ppm (at 2.15 μm) to 77 ppm (at 5.14 μm). The noise on the transit depth is thus about 2–3 times higher for K2-18 b compared to HD 209458 b.

In additional to experiencing higher noise, the spectral amplitudes for the low-mass K2-18 b are much smaller than for the hot Jupiter HD 209458 b (Fig. 7c) that further reduces the final signal-to-noise on the spectrum (where the ‘signal’ is the spectral amplitude) compared to HD 209458 b. We can readily see from Fig. 7(c) that the transmission spectrum for HD 209458 b recovers planetary spectral features to a high degree of precision and accuracy. For K2-18 b, large NIR spectral features (such as the 1.4 μm water feature) are discernible above noise. The final spectrum for HD209458b contains a gap in wavelength coverage between 1.19 and 1.47 μm due to using the SUB512 subarray for the G140H grating.

4.4 Case 4: day-side emission spectrum of HD 209458 b with NIRCam and MIRI using Monte Carlo method

4.4.1 Overview

This case illustrates a simulated observation of the secondary eclipse of HD 209458 b using NIRCam and MIRI LRS. The same instrument configurations used in case 2 are used here (NIRCam Time-series Grism F444W and F332W2 with four-output readout pattern and SUBGRISM64 subarray and MIRI LRS SLITLESSPRISM). The simulations use eclipse light curves to modulate the stellar signal. These are generated in the same way as primary transit light curves but LDCs are set to zero and the minimum of the light curve is

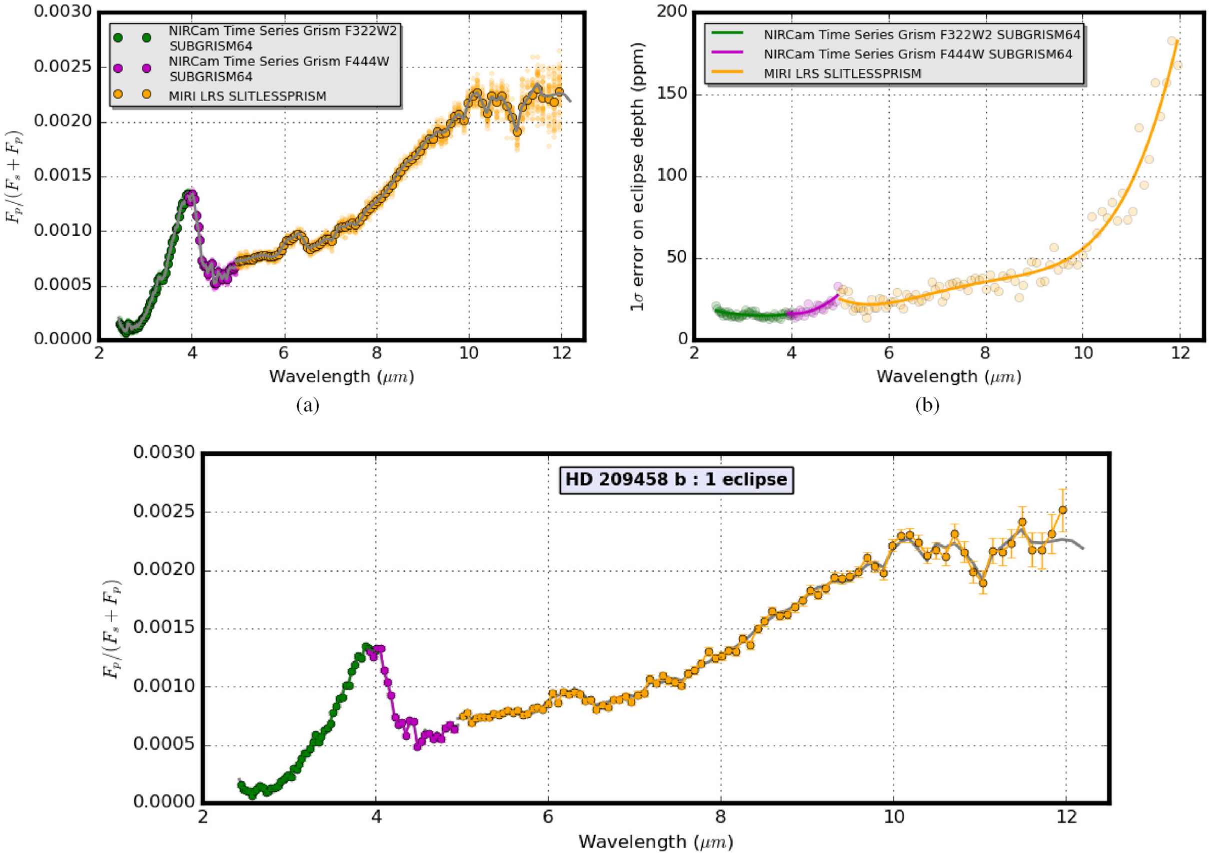


(a)

(b)

(c)

Figure 7. Case 3: simulation of transmission spectra for HD209458b and K2-18b using multiple NIRSpec instrument subarray modes using the Monte Carlo method. (a) Monte Carlo simulation recovered transit depths versus wavelength (small dots: individual realizations, large dots: average of 50 realizations, grey line: input spectrum binned to $R = 100$). (b) 1σ error on transit depth versus wavelength with second-order polynomial line fits. (c) Synthetic post-processed transmission spectra for each planet for 1 transit (per instrument mode) at $R = 100$ (grey line: input spectrum binned to $R = 100$).



(a)

(b)

(c)

Figure 8. Case 4: simulation of emission spectrum for HD 209458 b with NIRCam Time Series Grism F322W2 and F444W modes (both with SUBGRISM64) and MIRI LRS slitless mode, using the Monte Carlo method. (a) Monte Carlo simulation recovered eclipse depths versus wavelength (small dots: individual realizations, large dots: average of 50 realizations, grey line: input spectrum binned to $R = 100$). (b) 1σ error on eclipse depth versus wavelength with second and fourth-order polynomial line fits for NIRCam and MIRI respectively. (c) Synthetic post-processed dayside-emission spectrum for 1 eclipse (per instrument mode) at $R\ = 100$ (grey line: input spectrum binned to $R = 100$).

normalized to unity. The fractional eclipse depth of each light curve gives the planet–star flux ratio expressed as $F_{\rm p}/(F_{\rm s} + F_{\rm p})(\lambda)$. The input planet spectrum for HD 209458 b was obtained from a self-consistent model reported in Gandhi & Madhusudhan (2017). As in the transmission cases, all noise sources, background sources, and instrumental effects in Table 1 (excluding the user-defined systematic) are included, and the final results are processed through the pipeline in the same way as for cases 2 and 3. The Monte Carlo approach is used with 50 realizations for each instrumental configuration. We find the precision on the eclipse depth for an R of 100 and produce a post-processed day-side emission spectrum extending from 2.4 to 12 µm.

4.4.2 Simulation settings

The settings for this simulation are given in Table 4 and are mostly the same as used in case 2. The key change is for `obs_type` where 2 is selected to generate a secondary eclipse observation. 'HD 209458 b' is entered for `planet_name`. The path to the secondary eclipse input planet spectrum file is entered under `planet_spectrum_file`.

4.4.3 Results

Fig. 8 shows the results from this case study. We find that a very high precision result is obtained from just one eclipse per instrument mode given the brightness of the host star. Transit depths recovered from the Monte Carlo simulation are shown in Fig. 8(a). Fig. 8(b) shows the standard deviations for each bin and as previously polynomials are fitted to these (second order for NIRCam, fourth order for MIRI).

Using the polynomial fitted values, for NIRCam the eclipse depth precision ranges from 15 ppm (at 3.41 µm) to 18 ppm (at 2.45 µm) for the F332W2 mode, and from 16 ppm (at 4.06 µm) to 27 ppm (at 4.95 µm) for the F444W mode. For MIRI LRS, the precision ranges from a minimum of 22 ppm (at 5.64 µm) to a maximum of 183 ppm (at 11.95 µm). We can see that the randomized post-processed spectrum has very small error bars and traces the input spectrum to high degree of precision and accuracy with just a single observation (per instrument mode). These precision results are comparable to those obtained in transmission for the same target.

4.5 Case 5: noise budget analysis of K2-18 b observation with MIRI using Allan deviation method

4.5.1 Overview

A noise budget analysis breaks down the total noise into its constituent elements, helping us to understand which noise sources are most impactful and thus require the greatest attention for mitigation in observational and data reduction strategies. Here, we perform a noise budget for K2-18 b observed with MIRI LRS slitless mode. We use out-of-transit simulations with Allan deviation analysis, looping through different noise sources and recovering the fractional noise at

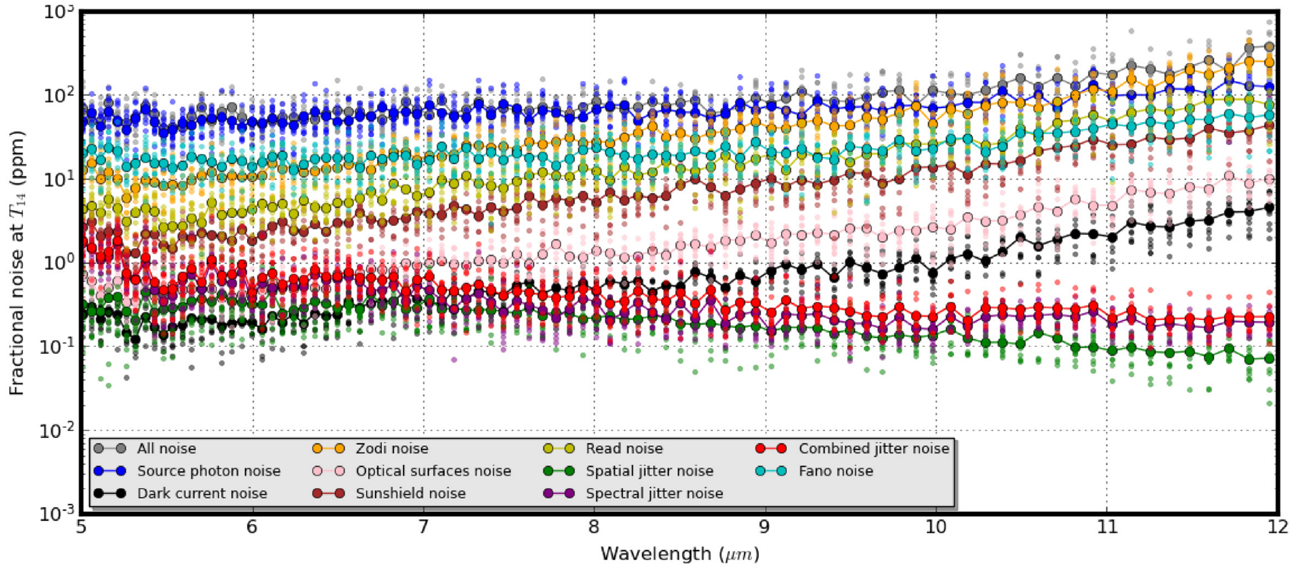


Figure 9. Case 5: noise budget for MIRI LRS slitless mode observing K2-18 b at $R = 100$ (small dots: individual realizations, large dots: average of 10 realizations).

T_{14} versus wavelength for each source. The noise sources are those listed in Table 1. Each is simulated with the other noise sources switched off and a simulation is also produced with all noise sources on. The instrumental effects listed in Table 1 (excluding the user-defined systematic) are applied in all cases with the exception of pointing jitter which is only applied in the pointing jitter noise case and in the all noise case. Background sources listed in Table 1 are applied only if the corresponding noise is being simulated, e.g. the zodical light background is not applied unless the zodi noise is being simulated, so only in the zodi noise and all noise cases. The results are processed through stage 1 and stage 2 (Allan deviation analysis) of the pipeline (Fig. 2) with some variations. The jitter decorrelation step is only applied in the pointing jitter and all noise cases. Background subtraction is only applied if a diffuse background (zodi, optical surfaces emission or sunshade) is included. Dark current subtraction is only applied for dark current noise and all noise cases. We obtain the average result of 10 realizations in each case.

4.5.2 Simulation settings

The simulation settings for the noise budget simulation are given in Table 4 and mostly follow those of case 1 with the following exceptions. The parameter `sim_mode` is set to 3, to select the noise budget option. When this is done `sim_noise_source` can be left blank as the code will loop through all noise sources. `obs_inst_config` is set for MIRI LRS SLITLESSPRISM. Also the aperture autosizing function is disabled (`pipeline_auto_ap` is set to 0) so that an extraction aperture size must be applied manually. This ensures that variation of the extraction aperture size is not a factor when comparing the different noise contributions. We set `pipeline_ap_factor` to 1 (based on the value obtained by the aperture autosizing function for a simulation with all noise activated). The width of the mask is $2 \times \alpha F\lambda$, where F is the F-number, λ is the wavelength, and α is the value of `pipeline_ap_factor`. For a rectangular aperture mask (as selected here) λ is the longest wavelength in the wavelength range. If a wavelength-dependent aperture mask is chosen, λ is the wavelength of each pixel column.

4.5.3 Results

Results from this example are shown in Fig. 9. The smaller dots show results from individual realizations with the larger dots giving the average per spectral bin. We find that below about 10 µm the spectra can be considered photon-noise-limited (the stellar 'source photon noise' being the largest noise contributor); however, at longer wavelengths zodiacal light noise becomes significant and exceeds source photon noise above ∼11 µm. Read noise is also significant at the long wavelength end but is mitigated by reading up-the-ramp and the large number of groups per integration (in this case $n = 35$). Fano noise, which is dependent on the photons from both star and backgrounds, also becomes a significant noise source at the red end, bolstered by the high number of zodi photons, but does not exceed the source photon noise. Noise from the sunshade emission increases with wavelength and is more significant than noise emission from optical surfaces; however, it too does not exceed the source photon noise up to 12 µm. Other noise sources such as pointing jitter, optical surfaces emission noise, and dark current noise are well-controlled and orders of magnitude smaller than the source photon noise. The contribution of some noise sources is dependent on the extraction aperture size. For example, if we use a larger aperture size, the impact of zodi noise increases overtaking photon noise at shorter wavelengths. This kind of analysis can also be used to assess the impact of changing observational parameters such as saturation limit (and thus integration time and total number of integrations) on the overall noise balance.

5 CONCLUSIONS

JexoSim 2.0 is a time-domain simulator of JWST exoplanet spectroscopy designed for the planning and validation of science cases for JWST. JexoSim generates synthetic spectra that capture the full impact of complex noise sources and systematic trends, allowing for assessment of both accuracy and precision in the final spectrum. In this paper, we described the upgraded simulator, which has new noise sources and instrumental effects and has been designed to improve operability, speed, efficiency, and versatility. JexoSim 2.0 is now freely available to the community. We have described here how to implement the code, with specific case studies that can be replicated by the user.

The case studies demonstrate a range of capabilities with simulated results from all four instruments and different subarray configurations. We examined how JWST can be used to observe the hot Jupiter HD 209458 b around a Sun-like star, and the low-mass planet K2-18 b around an M-dwarf. Observing bright targets, such as HD 209458 b, may present challenges in finding subarray modes that will not saturate, with possible gaps in wavelength coverage when combining certain NIRSpec grating configurations. K2-18 b is of great current interest given its location in the habitable zone with similar insolation to the Earth and the discovery of water vapour and clouds in its atmospheric spectrum using the Hubble WFC3 IR instrument (Benneke et al. 2019; Tsiaras et al. 2019) using nine transits. However, whereas the Hubble spectrum covered only 1.1 to 1.7 μm, the JWST spectra simulated here range from ∼0.9 to 12 μm. While we have not performed a spectral retrieval study in this paper, spectral features in the WFC3 range are clearly visible over noise with just one transit observation using NIRSpec or NIRISS. Similar levels of noise are obtained with each. The 1.4 μm water feature (observed with the Hubble WFC3) has an amplitude of ∼115 ppm in the spectrum from our radiative transfer model. JexoSim predicts a transit depth noise of 33 ppm with NIRSpec (G140M) and 31 ppm with NIRISS. If we divide the amplitude by the transit depth noise we obtain a signal-to-noise of about 3.5–3.7 for one transit, which compares well with the multiple transits needed to achieve a similar precision with Hubble. When observing K2-18 b, NIRCam achieves similar noise performance to NIRSpec and NIRISS, though its coverage does not reach down to the WFC3 wavelength range. While NIRSpec offers the widest total wavelength coverage of the three NIR instruments, when using the gratings, this must be achieved through applying multiple subarray modes, each requiring its own observation. The PRISM mode can cover the complete range in one pass; however, this will work only with dim targets that will not saturate the detector. For K2-18 b, the noise at $R = 100$ with MIRI LRS ranges from 70 ppm to 448 ppm. The noise increases with wavelength, so that particularly at the long wavelength end of MIRI, these results may not be useful. However, binning to lower R will improve precision. Future studies using spectral retrievals can incorporate JexoSim's ability to add confounding factors such as spot effects or additional systematics to see if spectral features including simulated biomarkers can be successfully recovered from low-mass planets around M-dwarf stars, targeted for observation by JWST.

Although there may be systematics not currently known or included in JexoSim 2.0, the code can be easily upgraded or modified in future to accommodate these. JexoSim 2.0 can be used to cross-validate initial results from other simulation tools in cases where time-domain effects are critical to consider, as well as to compare performance across the suite of JWST instruments and modes as performed here. JexoSim 2.0 provides a framework for investigating in detail the feasibility of specific and novel scientific cases for JWST where detectability may be vulnerable to astrophysical or instrumental confounding factors. It is also well suited to investigating the impact of star-spots and faculae as well as stellar pulsation and granulation noise. With simple modifications it can be used to investigate detection of time-dependent scientific signals such as exomoon transits or phase-related modulations. JexoSim 2.0 can also be customized for other astrophysical applications beyond exoplanet spectroscopy, such as brown dwarf spectroscopy.

With JWST, exoplanet science will be transformed. JexoSim 2.0 will assist in maximizing and optimizing the scientific possibilities of this ground-breaking observatory.

ACKNOWLEDGEMENTS

Thanks to Sam Rowe (Cardiff University) for alpha testing and help with implementing Numba. Thanks to Andreas Papageorgiou (Cardiff University) for alpha testing and to Tyler Gordon (University of Washington) for trialling the code and providing valuable feedback. Thanks to Jayesh Goyal (Cornell University) for help with using the ATMO scaleable grid. We use a modified version of his publicly available code snippet for rescaling the spectra from this grid for use in JexoSim. Our thanks to the referee for their helpful comments and suggestions for improvement. SS was supported by United Kingdom Space Agency grant ST/S002456/1.

DATA AVAILABILITY

The data underlying this article will be shared on reasonable request to the corresponding author.

REFERENCES

Agol E., Cowan N. B., Knutson H. A., Deming D., Steffen J. H., Henry G. W., Charbonneau D., 2010, ApJ, 721, 1861
Allan D., 1966, Proc. IEEE, 54, 221
Allard F., Homeier D., Freytag B., 2012, Phil. Trans. R. Soc. A, 370, 2765
Astropy Collaboration, 2013, A&A, 558, A33
Astropy Collaboration, 2018, AJ, 156, 123
Barstow J. K., Aigrain S., Irwin P. G. J., Kendrew S., Fletcher L. N., 2015, MNRAS, 448, 2546
Batalha N. E. et al., 2017, PASP, 129, 064501
Beichman C. A., Rieke M., Eisenstein D., Greene T. P., Krist J., McCarthy D., Meyer M., Stansberry J., 2012, in Clampin M. C., Fazio G. G., MacEwen H. A., Oschmann Jacobus M. J., eds, Proc. SPIE Conf. Ser. Vol. 8442, Space Telescopes and Instrumentation 2012: Optical, Infrared, and Millimeter Wave. SPIE, Bellingham, p. 84422N
Beichman C. et al., 2014, PASP, 126, 1134
Benneke B. et al., 2019, ApJ, 887, L14
Berta Z. K. et al., 2012, ApJ, 747, 35
Charbonneau D., Brown T. M., Noyes R. W., Gilliland R. L., 2002, ApJ, 568, 377
Deming D., Knutson H. A., 2020, Nat. Astron., 4, 453
Deming D., Sheppard K., 2017, ApJ, 841, L3
Deming D. et al., 2013, ApJ, 774, 95
Diamond-Lowe H., Charbonneau D., Malik M., Kempton E. M. R., Beletsky Y., 2020, AJ, 160, 188
Doyon R. et al., 2012, in Clampin M. C., Fazio G. G., MacEwen H. A., Oschmann Jacobus M. J., eds, Proc. SPIE Conf. Ser. Vol. 8442, Space Telescopes and Instrumentation 2012: Optical, Infrared, and Millimeter Wave. SPIE, Bellingham, p. 84422R
Ferruit P., Birkmann S., Böker T., Sirianni M., Giardino G., de Marchi G., Alves de Oliveira C., Dorner B., 2014, in Oschmann Jacobus M. J., Clampin M., Fazio G. G., MacEwen H. A., eds, Proc. SPIE Conf. Ser. Vol. 9143, Space Telescopes and Instrumentation 2014: Optical, Infrared, and Millimeter Wave. SPIE, Bellingham, p. 91430A
Gandhi S., Madhusudhan N., 2017, MNRAS, 472, 2334
Gardner J. P. et al., 2006, Space Sci. Rev., 123, 485
Giardino G. et al., 2012, in Holland A. D., Beletic J. W., eds, Proc. SPIE Conf. Ser. Vol. 8453, High Energy, Optical, and Infrared Detectors for Astronomy V. SPIE, Bellingham, p. 84531T
Goyal J. M., Wakeford H. R., Mayne N. J., Lewis N. K., Drummond B., Sing D. K., 2018, MNRAS, 482, 4503

Greene T. P., Line M. R., Montero C., Fortney J. J., Lustig-Yaeger J., Luther K., 2016, ApJ, 817, 17
Grillmair C. J. et al., 2008, Nature, 456, 767
Harris C. R. et al., 2020, Nature, 585, 357
Hunter J. D., 2007, Comput. Sci. Eng., 9, 90
Husser T. O., Wende-von Berg S., Dreizler S., Homeier D., Reiners A., Barman T., Hauschildt P. H., 2013, A&A, 553, A6
Janesick J. R., 2007, Photon Transfer DN → λ. SPIE, Bellingham
Janson M., Bergfors C., Goto M., Brandner W., Lafrenière D., 2010, ApJ, 710, L35
Knutson H. A. et al., 2011, ApJ, 735, 27
Konopacky Q. M., Barman T. S., Macintosh B. A., Marois C., 2013, Science, 339, 1398
Kreidberg L. et al., 2014, Nature, 505, 69
Lam S. K., Pitrou A., Seibert S., 2015, in Finkel H., ed., Proc. 2nd Workshop on the LLVM Compiler Infrastructure in HPC, LLVM '15, SC15: The International Conference for High Performance Computing, Networking, Storage and Analysis. Association for Computing Machinery, New York
Leisenring J., 2018, Pynrc, 0.8.0dev, https://pynrc.readthedocs.io/en/latest/
Line M. R. et al., 2013, ApJ, 775, 137
McCullough P. R., Regan M., Bergeron L., Lindsay K., 2008, PASP, 120, 759
Madhusudhan N., 2018, in Deeg H. J., Belmonte J. A., eds, Atmospheric Retrieval of Exoplanets. Springer International Publishing, Cham, p. 1
Madhusudhan N., 2019, ARA&A, 57, 617
Madhusudhan N., Seager S., 2009, ApJ, 707, 24
Madhusudhan N., Nixon M. C., Welbanks L., Piette A. A. A., Booth R. A., 2020, ApJ, 891, L7
Mayor M., Queloz D., 1995, Nature, 378, 355
Morello G., Claret A., Martin-Lagarde M., Cossou C., Tsiaras A., Lagage P.-O., 2020, AJ, 159, 75
Nielsen L. D. et al., 2016, in MacEwen H. A., Fazio G. G., Lystrup M., Batalha N., Siegler N., Tong E. C., eds, Proc. SPIE Conf. Ser. Vol. 9904, Space Telescopes and Instrumentation 2016: Optical, Infrared, and Millimeter Wave. SPIE, Bellingham, p. 99043O
Parviainen H., 2015, MNRAS, 450, 3233
Perrin M. D., Soummer R., Elliott E. M., Lallo M. D., Sivaramakrishnan A., 2012, in Clampin M. C., Fazio G. G., MacEwen H. A., Oschmann Jacobus M. J., eds, Proc. SPIE Conf. Ser. Vol. 8442, Space Telescopes and Instrumentation 2012: Optical, Infrared, and Millimeter Wave. SPIE, Bellingham, p. 84423D
Piquéras L. et al., 2010, in Angeli G. Z., Dierickx P., eds, Proc. SPIE Conf. Ser. Vol. 7738, Modeling, Systems Engineering, and Project Management for Astronomy IV. SPIE, Bellingham, p. 773812
Pont F., Zucker S., Queloz D., 2006, MNRAS, 373, 231
Pont F., Knutson H., Gilliland R. L., Moutou C., Charbonneau D., 2008, MNRAS, 385, 109
Pontoppidan K. M. et al., 2016, in Peck A. B., Seaman R. L., Benn C. R., eds, Proc. SPIE Conf. Ser. Vol. 9910, Observatory Operations: Strategies, Processes, and Systems VI. SPIE, Bellingham, p. 991016
Rackham B. V., Apai D., Giampapa M. S., 2018, ApJ, 853, 122
Rauscher B. J. et al., 2007, PASP, 119, 768
Rauscher B. J. et al., 2017, PASP, 129, 105003
Ressler M. E. et al., 2015, PASP, 127, 675
Rieke G. H. et al., 2015, PASP, 127, 584
Rohatgi A., 2020, Webplotdigitizer: Version 4.4, https://automeris.io/WebPlotDigitizer
Sarkar S., Argyriou I., Vandenbussche B., Papageorgiou A., Pascale E., 2018, MNRAS, 481, 2871
Sarkar S., Madhusudhan N., Papageorgiou A., 2019, MNRAS, 491, 378
Schwieterman E. W. et al., 2018, Astrobiology, 18, 663
Seager S., 2014, Proc. Natl. Acad. Sci., 111, 12634
Seager S., Sasselov D. D., 2000, ApJ, 537, 916
Sing D. K. et al., 2011, MNRAS, 416, 1443
Sing D. K. et al., 2016, Nature, 529, 59
Space Telescope Science Institute, 2018a, Technical Report, User Documentation for Cycle 1: NIRISS. Space Telescope Science Institute/NASA/ESA/CSA, Baltimore, USA, https://jwst-docs.stsci.edu/download/attachments/16744575/jwst_niriss_documentation_20180212.pdf?version=20&modificationDate=1556155233201&api=v2
Space Telescope Science Institute, 2018b, Technical Report, User Documentation for Cycle 1: NIRCam. Space Telescope Science Institute/NASA/ESA/CSA, Baltimore, USA, https://stsci.app.box.com/s/wdu8467g3yp7rjj8z6rs82nwu8a87wuy
Space Telescope Science Institute, 2018c, Technical Report, User Documentation for Cycle 1: NIRSpec. Space Telescope Science Institute/NASA/ESA/CSA, Baltimore, USA, https://jwst-docs.stsci.edu/download/attachments/16744575/jwst_niriss_documentation_20180212.pdf?version=20&modificationDate=1556155233201&api=v2
Space Telescope Science Institute, 2018d, Technical Report, User Documentation for Cycle 1: MIRI. Space Telescope Science Institute/NASA/ESA/CSA, Baltimore, USA, https://jwst-docs.stsci.edu/download/attachments/16744575/jwst_miri_documentation_20180212.pdf.
Stevenson K. B. et al., 2014, Science, 346, 838
Swain M. R., Vasisht G., Tinetti G., 2008, Nature, 452, 329
Tinetti G. et al., 2018, Exp. Astron., 46, 135
Tsiaras A., Waldmann I. P., Tinetti G., Tennyson J., Yurchenko S. N., 2019, Nat. Astron., 3, 1086
Virtanen P. et al., 2020, Nat. Methods, 17, 261
Waldmann I. P., Tinetti G., Rocchetto M., Barton E. J., Yurchenko S. N., Tennyson J., 2015, ApJ, 802, 107
Welbanks L., Madhusudhan N., 2019, AJ, 157, 206
Wes M., 2010, in van der Walt S., Millman J., eds, Proc. 9th Python in Science Conference, Data Structures for Statistical Computing in Python. SciPy, p. 56
Wolszczan A., Frail D. A., 1992, Nature, 355, 145

This paper has been typeset from a TEX/LATEX file prepared by the author.