

Simulating direct exoplanet detection with the James-Webb-Space Telescope

Niklas Viebig, e-mail: nviebig@student.ethz.ch

Institute for Particle and Astrophysics, ETH Zurich, 8093 Zurich, Switzerland

Abstract

Context. With the James-Webb Space Telescope (JWST), astronomy has gained the largest and most powerful space telescope in history. With it, it will be possible to explore not only our solar system, but the entire universe in a way that has never been possible before. Especially the field of exoplanetary science benefits from the possibilities of the JWST, as it gives researchers the possibility to directly detect exoplanets. So far, only about 0.01 % of exoplanets have been detected directly.

Aims. In this project, my goal was to create a python code to simulate direct, coronagraphic exoplanet detection with MIRI (the Mid-InfraRed Instrument) onboard the JWST. So far, there is no public available code to do so. To start, I will first introduce MIRI on board the JWST. Then I will explain how coronagraphs work, using MIRI's four coronagraphs as an example. Finally, I will simulate a direct exoplanet detection, and discuss the result.

Method. I wrote a Python script to combine PanCAKE, a simulator which can simulate coronagraphic observations of the MIRI instrument, with MIRIsim, the MIRI simulation engine. With this, I can simulate a coronagraphic observation with MIRI for a given target scene. To create such an observation, the program first creates a coronagraphic observation with PanCAKE, stores it as a file, and gives this file to MIRIsim. MIRIsim then takes this scene and simulates an imager observation.

Conclusion. In this project, I build a tool, which gives a user the ability to simulate a coronagraphic simulation with the JWST. The returned data are in the format of Level 1b data, which can be used further along the JWST pipeline.

Keywords: James-Webb-Space Telescope, MIRI, Exoplanet, Coronagraph

1. Introduction

With its launch in December 2021, the James Webb Space Telescope ushered in a new era of space exploration. With the telescope, astronomers hope to obtain clues to the most fundamental questions about the universe. The scientific goals of the JWST, as described by Gardner et al. [1], fall into four categories. First is the study of the early universe, where researchers suspect to detect the first supernovas and mini-quasars, which formed around 180 million years after the big bang. These findings could help us understand how reionization occurred, the first step in the formation of the first galaxies.

Second, there is the topic of the structure of galaxies. Here, the goal is to provide clues and evidence to the question of when and how the first galaxies formed. Third is the birth of stars and protoplanetary systems. Because there are many unanswered

questions about the details of how stars and planets form, astronomers hope to detect young exoplanets in their protoplanetary disk to study their formation.

Lastly, there is the topic of planetary systems and the origins of life. Here, scientists want to analyze the atmosphere of exoplanets to search for the building blocks of life on planets outside our solar system [2]. In all these topics, JWST gives astronomers the tools to revolutionize their respective fields and to make contributions to science that will revolutionize the way we think about the universe. The telescope is a collaboration between NASA, ESA, and the CSA. With a five-layer solar shield the size of a tennis court and a 6.5-meter diameter primary mirror composed of 18 gold-plated hexagonal mirrors, it is the largest telescope ever sent into space. It orbits L2, a Lagrange point 1.5 million kilometers from

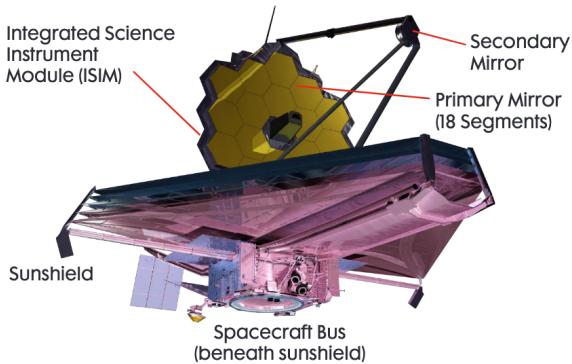


Figure 1. An illustration of the JWST, showing its main components[3].

Earth, and has enough fuel for an over ten-year mission.[6]

On board the JWST, the Integrated Science Instrument Module (ISIM) contains four instruments:

- the near-infrared camera (NIRCam), which covers the infrared wavelength range from 0.6 to 5 microns
 - the near-infrared Spectrograph (NIRSpec), which also covers a wavelength range from 0.6 to 5 microns
 - the near-infrared Imager and slitless Spectrograph (NIRISS), again covering the range from 0.6 to 5 microns
 - the Mid-Infrared Instrument (MIRI), which covers the wavelength range from 5 to 28 microns.
- This instrument not only has an imager, but also a spectrograph.

MIRI and NIRCam are also equipped with different coronagraphs. A coronagraph is a tool which can block out the light from a star. A coronagraph is especially important when trying to directly detect an exoplanet. Because an exoplanet's star is usually thousands to millions of times brighter than its planet, the star dominates the image, and you cannot detect the planet directly. With a coronagraph, it is possible to filter out the light of the star to detect the planet directly. Direct detection of an exoplanet is to this day rare. Of the 5000 exoplanets which we found, since the first detection in 1995, only around 50 have been observed directly.[7] For the MIRI instrument, there exists a simulator

called PanCAKE. PanCAKE is the coronagraphic simulation tool from the JWST exposure time calculator, Pandeia. In this semester project, I worked with a simulator for the MIRI instrument called MIRIsim. This tool allows the user to simulate an MIRI observation for a specific target scene. Simulating an observation is important for two reasons. First, it allows any researcher writing a proposal to test and reevaluate their observational strategies. Second, it gives researchers an opportunity to better understand the equipment and its capabilities. Unfortunately, MIRIsim does not allow the user to directly incorporate coronagraphic observations into the simulation, and no other such simulator exists publicly available. In this project, I will present my results of combining PanCAKE with MIRIsim to get a tool to more accurately simulate a direct exoplanet detection.

2. The Instruments

2.1 MIRI

As already mentioned, MIRI is one of the four instruments on board the JWST. It covers the mid-infrared wavelengths from 5 to 28 microns. Figure 2 shows the MIRI array. It is 113"×113" in size, of which 74"×113" is the imager. The four different coronagraphs cover the rest of the area. Beside the imager, the instrument contains four different coronagraphs, and a spectrometer, which can do low and medium resolution spectroscopy. The detectors used in the imager and the spectrometer^a are arsenic-doped silicon impurity band conductors (Si:As IBC)[8]. A table with the properties of such a detector can be found in the appendix (Table 4). When observing a scene with MIRI, four parameters must be specified: the filter, the dither pattern, the subarray, and the detector read-out mode, which determines the exposure time [4]. The choice of filter depends on the scene and therefore at which wavelength the observation will be made. The central wavelength and bandwidth of the nine broadband filters, which cover most of the range, can be seen in Table 2.

In astronomy, dithering is the practice of shift-

^aFrom now on, we will neglect the spectrometer, as it is not relevant to the project.

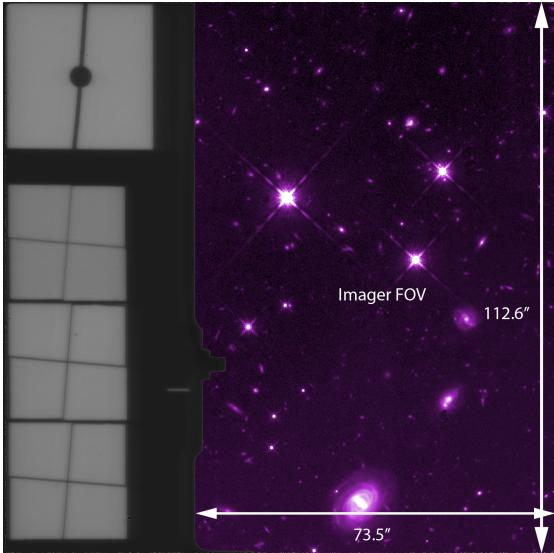


Figure 2: The MIRI-imager. On the right side is the main field of view (Fov) of the instrument. The four squares on the left side of the image are the coronagraphs of the instrument. The Lyot coronagraph is the easiest to recognize because it is distinguished by the large spot in the center of the Fov [4].

ing the target of the telescope after each exposure by a few pixels. This provides better sampling, removes the bad pixel effect, and optimizes self-calibration[9]. MIRI offers several types of dither patterns that can be picked, depending on the problem at hand.

MIRI offers nine types of subarrays to chose from, depending on the needs of the planned observation. Of these subarrays, four are the coronagraphs. The other five are imaging subarrays. They can be chosen depending on the brightness of the target. Figure 4 shows the MIRI instrument divided into different subarrays. [10].

To make every exposure interface similar, all detector readout patterns of the instruments onboard JWST follow the up-the-ramp (MULTIACCUM) readout pattern. This means that instead of just deciding on an exposure time, we divide our entire exposure into integrations. An integration is a number of groups, and a group is the average of a number of frames. We then have to specify a read-out pattern, the number of groups per integration N_{group} and the number of integrations per exposure N_{int} . This is visualized in Figure 5. With

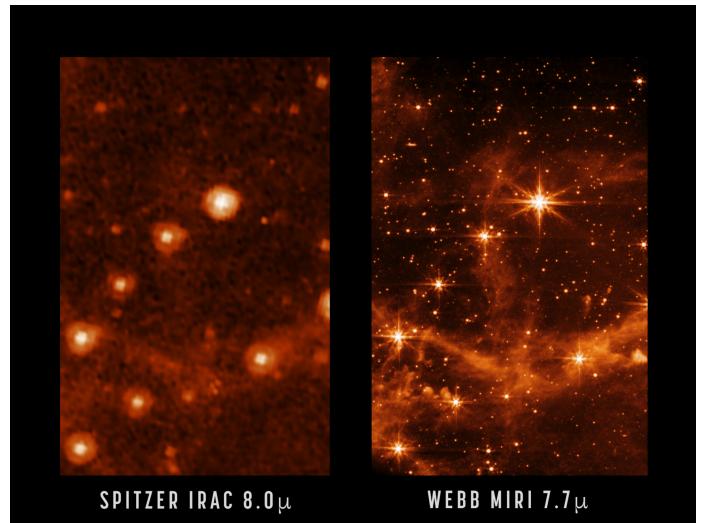


Figure 3 : Two Images of the Large Magellanic Cloud. The left was taken by the SPITZER infrared telescope, the right with the MIRI instrument [5].

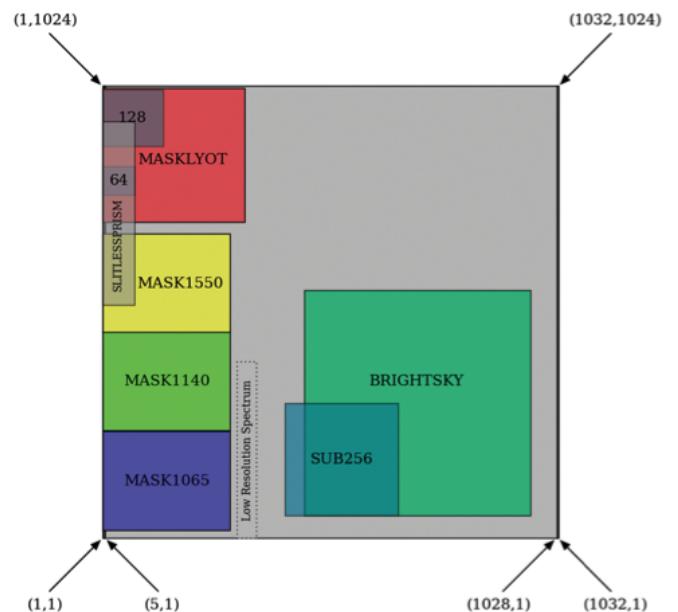


Figure 4. The MIRI focal plane, divided into all subarrays [10].

MIRI, by selecting a read-out pattern, we chose the number of samples per pixel per frame $N_{samples}$ and the frame time t_1 . Thus, the exposure time t_{exp} is calculated with the formula:

$$t_{exp} = (N_{int} \times t_{int}) + (N_{int} - 1) \times t_1 \quad (1)$$

where $t_{int} = N_{group} \times t_1$.

MIRI offers the user two read-out patterns:

- SLOWR1 ($N_{samples} = 9, t_1 = 23.890$ s)
- FASTR1 ($N_{samples} = 1, t_1 = 2.775$ s)

While FASTR1 is the default readout pattern, SLOWR1

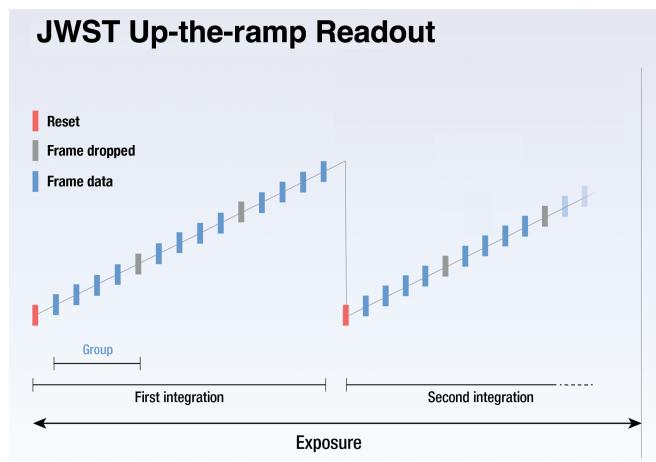


Figure 5. Diagram showing the up-the-ramp readout technique [11].

offers a lower detector noise. The up-the-ramp readout pattern offers several benefits. One, for example, is that by using this system, the volume of data that need to be sent to Earth is significantly smaller. Also, as we read the same pixels several times, we can detect cosmic rays that tamper with the image, as they induce spikes in the signal level. [11][12][13][14]

On Earth, observing the mid-infrared spectrum is difficult, as the thermal radiation in our atmosphere interferes with the observation of this spectrum. Here, space-based telescopes offer an advantage. Furthermore, MIRI is equipped with a cryo-cooler, a device that cools the instrument to 7 K. By comparison, the other instruments on-board JWST operate at about 40 K. The difference between MIRI and any other infrared telescope is astounding. MIRI's image sensitivity is

3000 times higher than that of ground-based infrared telescopes. Compared to other space-based telescopes, such as SPITZER, MIRI's image sensitivity is about 50 times better[15][16]. An example of the difference can be seen in Figure 3. While the stars in the SPITZER image appear rather blurry, the MIRI image even shows a clear background.

2.2 Coronagraphs

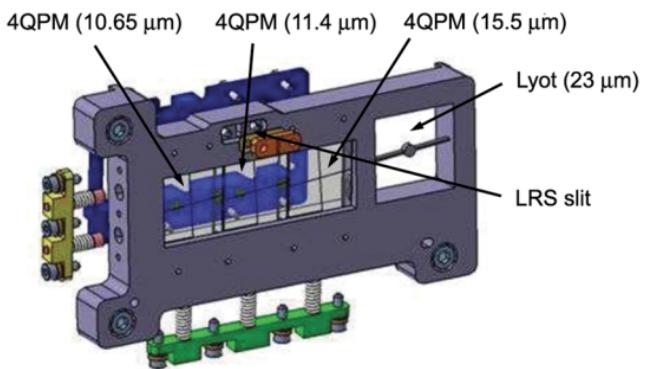


Figure 6. A closer look at the mounting bracket on which the coronagraphs are fixed[17].

A coronagraph is an instrument that blocks light from a bright source at which it is directed, and reduce the intensity of the surrounding light. It was originally used to block out light from the inner part of our sun, so that researchers could study the outer layer, the corona. Since then, many other applications have been found, such as in the discovery of exoplanets. There, its job is to block out the exoplanet's sun, since it is usually a thousand to a million times brighter than the planet. An important feature of a coronagraph is its inner working angle (IWA). This value indicates how close a planet can be to its star without being covered by the coronagraph.

MIRI has four coronagraphs on board. One of them is a Lyot coronagraph, the other three are so-called 4-quadrant phase mask coronagraphs (4QPM). Figure 6 shows the mounting bracket on which they are fixed.

2.2.1 Lyot-Coronagraph

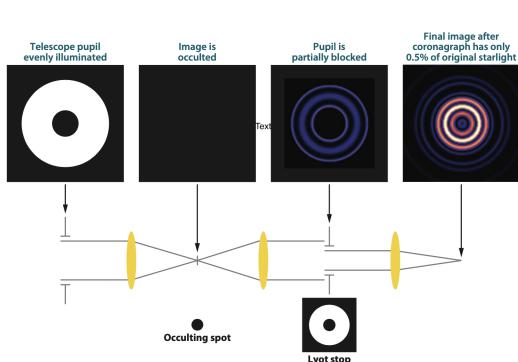


Figure 7. Sketch of a Lyot Coronagraph[18].

The Lyot Coronagraph (called the F2300C Filter) was chosen because it is capable of covering a broader range of wavelengths than the 4QPM Coronagraphs. The one installed on MIRI has a size of $30'' \times 30''$ and is placed in the top left corner of the imager, as can be seen in Figure 2 [17]. It is designed for a wavelength of 22.75 micrometers with a bandwidth of 5.5 micrometers (Table 3 shows all properties of the coronagraphs). This makes it useful for investigating bright sources, such as, for example, the outer regions of protoplanetary disks[19]. A diagram explaining how the Lyot coronagraph works can be seen in Figure 7. There, light from the Sun enters the instrument and is focused through a high-quality lens to avoid scattering effects on the occultation spot, which blocks the bright center and some of the first bright interference rings. The size of the occulting spot should be about the size of the sun in the first focal plane. The Lyot point in MIRI has a radius of $2.16''$, which is $3.3 \lambda/D$ in radius at 23 microns [19]. The remaining light is then scattered through a second lens and hits the Lyot stop, which prevents most of the sun's diffraction light from hitting the detector. As we directed the coronagraph directly onto the sun, the light of the planet is incident at an angle. So it misses the occulting spot and can reach the detector [20].

2.2.2 4QPM-Coronagraph

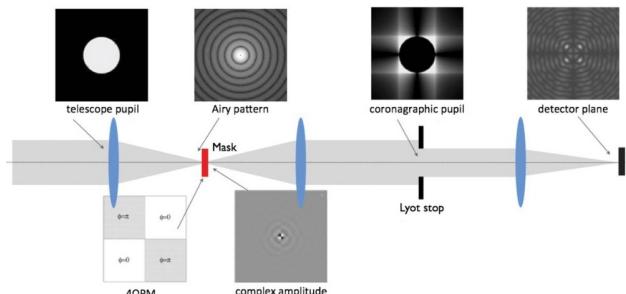


Figure 8. Sketch of a 4QPM Coronagraph [17].

The three 4-quadrant phase mask coronagraphs F1065C, F1140C, and F1550C operate at a central wavelength of 10.575, 11.30 and 15.5 micrometers with different bandwidths, all smaller than 1 micrometer (Table 3). They are placed beneath the Lyot coronagraph (Figure 2) and have a size of $24'' \times 24''$ [17]. The 4QPMs offer the advantage of having a very narrow IWA, which means that they will be able to detect signals much closer to a star. As can be seen by comparing Figure 7 and Figure 8, a 4QPM works differently than a Lyot coronagraph. Instead of hitting an occultation spot after the first lens, the light hits a phase mask. As is visible in Figure 8, the phase mask consists of four squares. Of these four squares, two diagonally opposed shift the phase of the light by 180 degrees. This creates a complex amplitude pattern. Through a lens, this pattern is then mapped onto a Lyot stop, which blocks some of the remaining diffraction light from the telescope. When the complex amplitude pattern finally hits the detector, after passing through the last lens, it interferes destructively. Because we can assume that the light of a star is emitted rather uniformly, the residual pattern on the detector should be rather dim. This makes it possible to detect the faint anomalous signal of an exoplanet [16].

3. Simulating JWST observations

As it is not possible to directly simulate a coronagraphic simulation with MIRIsim, we must use a trick. The idea is to first simulate the impact of a coronagraphic observation with PanCAKE, store this scene as a file, read the observation from the file

into a scene with MIRIsim, and then run the simulation of an imager detection of this scene. This should result in an observation in which both the coronagraph and the imager have been simulated.

3.0.1 MIRIsim

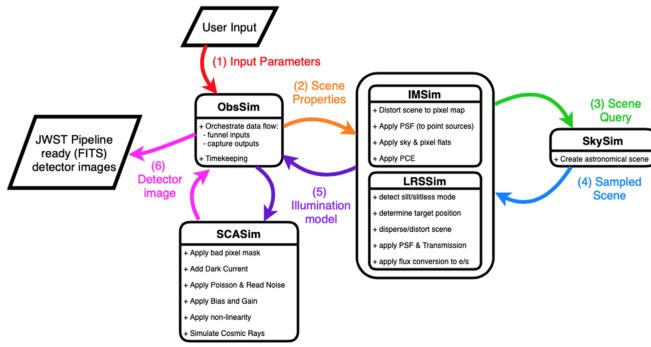


Figure 9. Diagram of how the simulation of the imager works[21].

MIRIsim is a python package, which was created by the MIRI European Consortium. It can simulate both the imager and the spectrometer within 10–20% of the nominal sensitivity baseline [21]. To accurately simulate the MIRI instrument, including the noise that the signal would experience in the telescope, the simulator requires the Calibration Data Products (CDPs) of MIRI. These CDPs describe the instrument. MIRIsim consists of different packages. For the sake of simplicity, MIRIsim is an anaconda environment. That means that by downloading MIRIsim all required packages are included. Figure 9 depicts the path the user input takes for a simulation of the imager. The input consists of an astronomical scene, which either can be given as its individual components, or as a premade scene in FITS format, and the parameters of the observation, which have been explained in the MIRI section. This is then passed on to the orchestration module (OBSSim). OBSSim forwards the user input to ImSim, which creates/reads the astronomical scene with skysim and builds the detector's illumination model according to the parameters. This is then sent through OBSSim to SCASim which simulates the detector image. SCASim takes detector effects into account. A list of all of these effects can be found in Table 1. SCASim then samples the illumination

Table 1. All detector effects SCASim takes into account.

Quantum efficiency
Poisson noise
Read noise
Reference pixels
Bad pixels
Dark current
Flat-field
Amplifier bias and gain
Detector non-linearity effects
Detector drift effects
Detector latency effects

model and returns detector images according to the group in which it was made. Finally, OBSSim takes these images, transforms them into FITS files, adjusts their header according to the requirements of the JWST calibration pipeline, and returns the resulting collection of detector images. [22] [21] The installation instructions can be found on the MIRICAL^b website. There, a bash script can be downloaded. After it is executed according to the instructions, the anaconda environment will build itself. For MIRIsim to function, additional data, such as CDP and PySynPhot files, have to be downloaded, and the environment variables that point to the additional data have to be set. To build the simulation in Python, the *MiriSimulation()* function can be called. It takes three inputs: the simulation configuration, the scene configuration, and the simulator configuration.

— With *SimConfig.makeSim* the simulation configuration can be built. The function takes all the observation parameters (the parameters are explained in Table 5 in the appendix).

— The scene configuration can be built by the function *SceneConfig.makeScene* which takes three inputs: the loglevel, the scene background and the actual scene. There, the scene can be either loaded from an external source or built with the skycube module.

— To build the default simulator configuration, the function *SimulatorConfig.from_default()* can be called. To execute the simulation, the function *.run()* has to

^b<https://wiki.miricle.org/Public/MirisimInstallation>

be called on the built simulation. After the simulation is run, MIRIsim will create a folder in which the detector image and the illumination model is stored. There, the detector image are Level 1b data.

3.0.2 PanCAKE

PanCAKE is the Coronagraphy Advanced Kit for Extractions. It is an improvement to the Exposure Time Calculator (ECT) of JWST, called Pandeia. Pandeia can simulate a range of the instruments onboard JWST, but lacks at the coronagraphic simulation. There, PanCAKE offers a user-friendly alternative.

As MIRIsim, for simplicity PanCAKE is also an anaconda environment. It can be downloaded from the GitHub page of Aaryan Carter^c with instructions. To function, it needs some additional packages and files, such as Pandeia, WebbPSF, and synphot data. The download link can also be found in the instructions.

To create a coronagraphic observation, one has to first create a scene. This can be archived with the command `pancake.scene.Scene("Scene Name")`. A scene in PanCAKE consists of a number of stellar objects. These can be assumed as either point sources or extended with a spatial brightness distribution. They also need the spectral energy distribution of the objects. Both can either be given in extra files or can be directly accessed from the database SIMBAD. To initialize a stellar object, the function `target.add_source()` can be called. It takes the name of the object, the radial distance from the center of the scene, where the center of the coordinate system lies, the angular position of the object, the filename, where the data of the spectral energy distribution are stored, and the units in which the wavelength and the flux data in the file are given. To initialize the observational sequence, the `pancake.sequence.Sequence()` function must be called. Various observations can be added with the function `add_observation()`. This function takes a pre-determined scene, exposure parameters, such as which filter and which read out mode is to be used and the Number of Groups and Interactions. To

run an observation, all that needs to be done is call `function.run()` on the sequence.[23]

PanCAKE will return a FITS file, which contains the simulated observation.

4. Method

My code consists of three Python files and one configuration file. In the configuration file, all parameters for PanCAKE and MIRIsim as well as all file paths are stored. This makes the whole code clearer. Of the three Python files, one contains the PanCAKE code, one contains the MIRIsim code, and the last one controls the order in which the code is executed and changes the environment when switching from PanCAKE to MIRIsim. This makes the procedures more efficient since we don't have to manually execute each file in each environment.

In the PanCAKE code, I began by defining the target scene with `pancake.scene.Scene("Scene Name")`. Then I constructed the scene. For my simulation, I added the star and planet to the scene with `target.add_source()`. To start an observation sequence, I called `pancake.sequence.Sequence()` and added the observation with `target.add_observation()`. Then I was able to run the simulation with the `.run()` command and save it as a FITS.

To create a MIRIsim simulation using `MiriSimulation()`, I had to initiate the scene, the simulation parameters and the simulator. I started by reading the FITS file from the PanCAKE code. As this file contains not only the simulated version of the scene, but also an unobstructed, not centered version, I had to edit the file to only contain the simulated image. To create a Skycube scene from the FITS file, I had to modify the header of the file because PanCAKE did not generate a sufficient version for MIRISim. Then I used `SceneConfig.makeScene()` to create the MIRIsim scene. There, the background and the log level had to be chosen the same way as in the PanCAKE code file. With `SimConfig.makeSim()` I then defined the simulation parameters. Finally, I used `SimulatorConfig.from_default()` to start the default MIRIsim simulator. With the command `.run()` the simulation could then be executed.

^c<https://aarynnccarter.com/PanCAKE/>

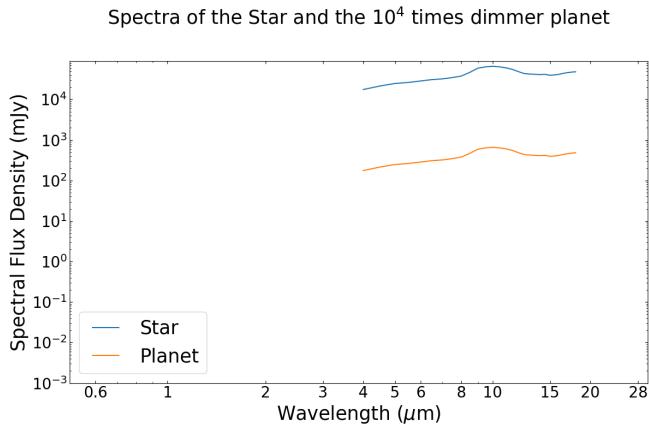


Figure 10. Diagram of the spectrum of the star and the exo-planet. For simplicity and clarity, I choose the planet to be much fainter than the sun. Therefore, it is easier to recognize.

5. Simulations Results

As an example, I simulated the detection of an exo-planet, which is a thousand times fainter than its star. Usually a planet is way more faint than this. The planet in this simulation is two arc seconds away of the star. The spectrum of the planet and the star are shown in Figure 10, and the detection parameters are listed in Table 6. I choose the F1140C

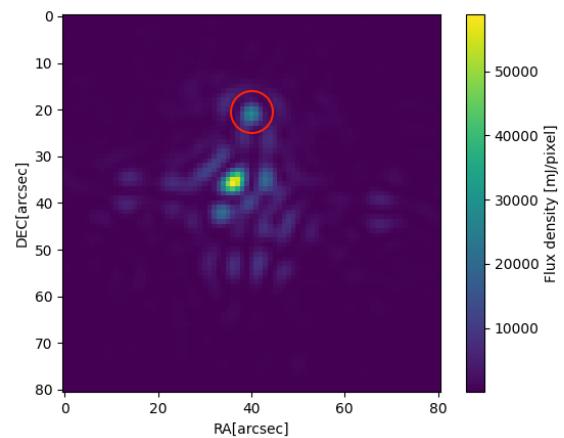


Figure 12. Here, the coronagraph F 1140C has been used to observe the scene. It can be seen that the light of the sun has been mostly filtered out. This scene is before it is given over to MIRIsim. Therefore, the MIRIs influence on the result has not been taken into account.

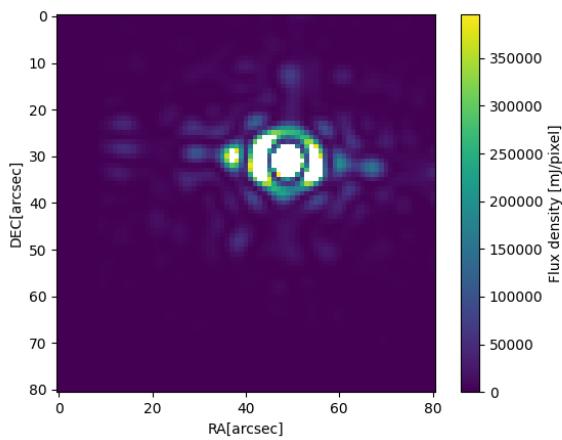


Figure 11. The Figure shows the astronomical scene created with PanCAKE. Here, no coronagraph has been used. Therefore, the planet is hidden inside the star's diffraction pattern. This image is not centered, as it is the off-axis image. PanCAKE uses it to compute the contrast curve with.

coronagraphic filter for this simulation, as its central wavelength of $11.3 \mu\text{m}$ coincides with the peaks of the spectra of the planet and the star. Also, I choose a very simple scene to illustrate the steps, processes,

and interim results of the code. Also, as my equipment is not strong enough to simulate more than 20 integrations with 200 groups, I choose this upper limit. As this example is for illustrative purposes only, this will not harm the qualitative nature of the plot. The output of the PanCAKE part of the program can be seen in Figure 11 and Figure 12. Figure 11 shows the astronomical scene created with PanCAKE. There, the interference patterns of the star on the detector, which hide the planet in orbit, can be seen. The image is not centered in the image, as PanCAKE uses the unobstructed image to calculate the contrast curve with. Figure 12 shows the same scene, after using the coronagraphic filter. There, the planet is visible. The scene of Figure 12 is then given to MIRIsim, to include the noise of the detector. This results in Figure 13. It shows the full imager field of view of MIRI. As described earlier, the result is not in the corresponding coronagraphic mask, but in the center FOV of the imager. Because the coronagraphic mask is already simulated in the PanCAKE output, it is not missing in the result. The image is the resulting FITS file summed up over all groups and integrations. Of this whole image, only the stellar neighborhood is important, as the rest is just empty detector noise. Figure 14 shows this section for our example. There, the planet is clearly separated from the sun.

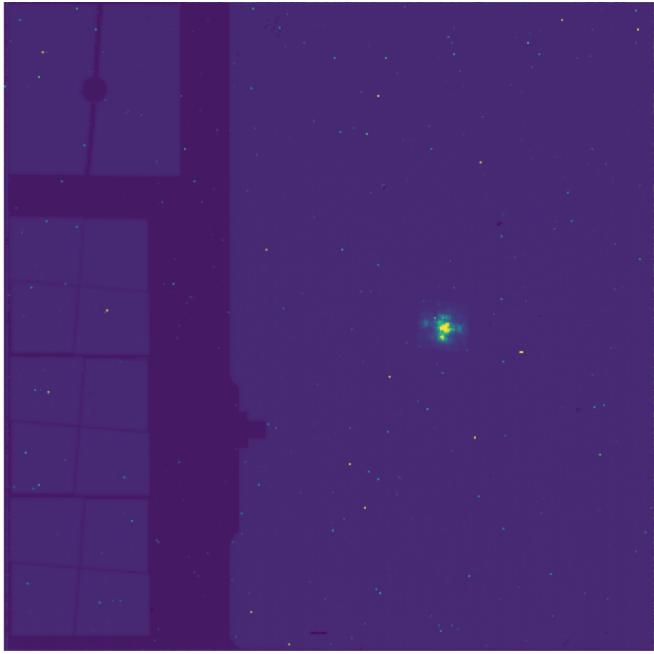


Figure 13. The finished MIRIsim simulation. Here, one can clearly see the outlines of the imager and the four coronagraphs.

6. Discussion

For me, a major challenge in this project was to find accurate resources to learn about the two simulators, as several of the online resources were outdated and obsolete. That's due to the fact that the two simulators have evolved drastically over the last year. For PanCAKE I would suggest the GitHub page of Aaryan Carter^d as it offers good download instructions and a basic example.

For MIRIsim, the miricle wiki^e, maintained by the MIRI European Consortium, also offers very clear download instructions with many helpful links and a troubleshooting section. Next to that, it also offers a lot of example code for the several simulations possibilities of MIRIsim.

Another problem for me was to set up the two simulators accurately, as both rely on various external data. Here, one has to take care of setting the environmental variables for both environments correctly with the accurate file paths. I would suggest to download all additional files in advance to better keep an overview. Also, to make sure to have all

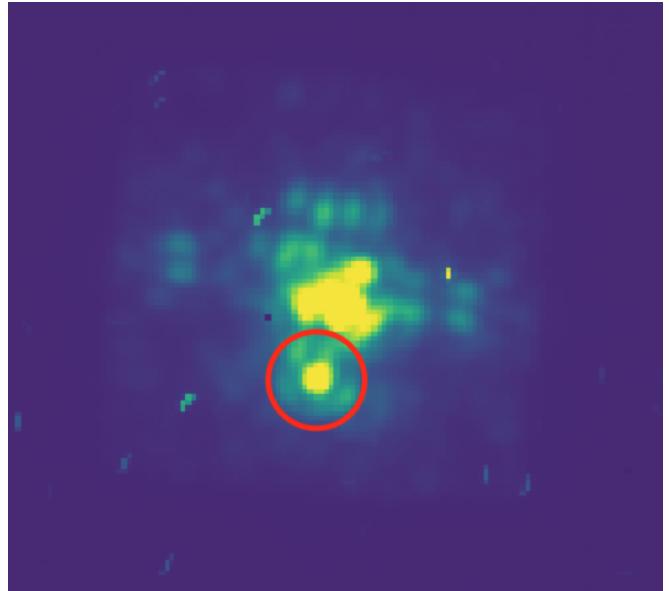


Figure 14. A close up of Figure 13. It shows the stellar environment

environmental variables set, I began both the PanCAKE and the MIRIsim python code by resetting the environmental variable with `os.environ["variable name"]`.

As can be seen in Section 5, the code creates an output that resembles a coronagraphic observation. But in comparison to a real observation, there are still several differences.

So far, my code does not support extended emission objects like circumstellar disks. In the future, I would definitely improve this by adjusting how I create a scene for this project and making it much more general. My code can be modified to support an observation with two roll angles. To create such an observation, the user must first define the two roll angles and give them as parameters to PanCAKE. Then the user must create two separate FITS files, which contain the two observations at the different roll angles. Then one has to only create two independent simulations with the two FITS files.

As can be seen in Figure 13, the scene created with Pandeia is rather small. Although this does not affect the quality of the coronagraphic observation, it can be a problem, especially if the observed scene is larger.

During the simulation, I faced the problem that the calculations with MIRIsim are computationally very

^d<https://aarynnbarton.com/PanCAKE/>

^ehttps://wiki.miricle.org/Public/MIRISim_Public

heavy. With my laptop, I choose 20 integrations per exposure, since Python crashed when I tried to do more. Therefore, a user must pay attention to adjust his scene, and especially the simulation, to their setup.

Because the JWST is a highly complex instrument,

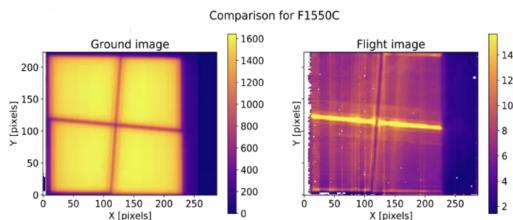


Figure 15. An example of the Glow-stick effect with the F1550C Filter. [24]

there are a few effects which have not been foreseen. One is the so called Glow-stick effect^f. Figure 15 illustrates this effect, which occurs when observing a scene with a coronagraph. It is caused by heat, which radiates from the sun shield and reflects over the secondary mirror and its support structure onto the MIRI instrument. As it the effect only depends on the exposure time and not on the target, it can be resolved by observing the empty sky with the same filter and the same exposure time and then subtract this from the original observation. In future versions, this effect could be simulated to further increase the accuracy of the simulated Level 1b data from the telescope.[25]

7. Conclusion

In this project, I succeeded in building a tool, which can simulate coronagraphic observations with the JWST. Although it has some flaws, as described in Section 5, it is so far the only public way to simulate these kinds of observations. In the process of this project, I learned a lot about JWST, MIRI and astronomy:

— I learned what the goals of JWST are and how the MIRI instrument works.

- How the MULTIACCUM read out pattern works, and how observations with MIRI are structured.
- How coronagraphs work and how, especially the Lyot and the 4QPM coronagraphs function.
- How MIRISim and PanCAKE work and how I can use them in Python.

— I worked with Level 1b data from the JWST and understood how image data is structured in astronomy.

— I gained in insight in the field of exoplanet detection and understood the importance of the JWST in its future.

With the JWST exceeding all expectations, it has already provided astronomers with breathtaking data, such as the deepest infrared image of our universe or high-resolution observations of Jupiter. In the future, the telescope will revolutionize the study of the early universe, the structure of galaxies, and will answer questions about planetary systems and the origins of life.

Acknowledgments

I would like to thank Prof. Dr. Sascha Quanz for giving me a chance to get my first insights into the field of astrophysics. I also want to thank Gabriele Cugno for his constant and kind support during this project.

^fhttps://jwst-docs.stsci.edu/jwst-mid-infrared-instrument/miri-features-and-caveats#MIRIFeaturesandCaveats-glow_sticksGlowsticksintheMIRI4QPMcoronagraphs

8. Appendix

Table 2. MIRI's imagine filters [4]

Filter name	Central wavelength (micrometers)	Bandwidth (micrometers)	FWHM (arcsec)	Comment
F560W	5.6	1.2	0.22	Broadband Imaging
F770W	7.7	2.2	0.25	PAH, broadband imaging
F1000W	10.0	2.0	0.32	Silicate, broadband imaging
F1130W	11.3	0.7	0.36	PAH, broadband imaging
F1280W	12.8	2.4	0.41	Broadband imaging
F1500W	15.0	3.0	0.48	Broadband imaging
F1800W	18.0	3.0	0.58	Silicate, broadband imaging
F2100W	21.0	5.0	0.67	Broadband imaging
F2550W	25.5	4.0	0.82	Broadband imaging

Table 3. MIRI's coronagraphic filters [19]

Filter	Coronagraph	Pupil mask transmission (%)	Central wavelength (micrometers)	Bandwidth (micrometers)	IWA5 (arcsec)	Rejection (on axis)
F1065C	4QPM1	62	10.575	0.75	0.33	260
F1140C	4QPM2	62	11.30	0.8	0.36	285
F1550C	4QPM3	62	15.50	0.9	0.49	310
F2300C	Lyot	72	22.75	5.5	2.16	850

Table 4. Detector properties[8]

Property	Value
Components	Si:As IBC devices manufactured by Raytheon Vision Systems (RVS)
Wavelength range	5 to 28 microns
Pixel format	1024 × 1024
Pixel size	25 micrometer
Plate scale	0.11"/pixel
Nominal operating temperature	≤6.7 K
Dark current	<0.2 e ⁻ /s/pix (median)
Read noise	14 e ⁻ (CDS)
Full well	250,000 e ⁻
Conversion gain	5.5 e ⁻ /DN

Table 5. All parameters the function SimConfig.makeSim takes

name	Name given to simulation
scene	Name of scene file to input
rel_obsdate	Relative observation date (0 = launch, 1 = end of 5 yrs)
POP	Primary Optical Path, Component on which to center (Imager or MRS)
ConfigPath	Configure the Optical path (MRS sub-band)
Dither	True/False depending on whether dithering should occur
StartInd	Start index for dither pattern
NDither	Number of dither positions
DitherPat	Dither pattern to use
disperser	Specify grating position (SHORT, MEDIUM, LONG) (only for spectrograph)
detector	Specify Channel (SW, LW or BOTH) (only for spectrograph)
mrs_mode	Detector read-out mode. Options are 'FAST' or 'SLOW' (only for spectrograph)
mrs_exposures	Number of Exposures (only for spectrograph)
mrs_integrations	Number of Integrations (per exposure) (only for spectrograph)
mrs_frames	Number of frames (or groups) per integration (only for spectrograph)
ima_exposures	Number of exposures
ima_integrations	Number of integrations !change in test case! 272
ima_frames	number of groups (for MIRI, # Groups = # Frames)
ima_mode	Imager read mode (FAST (2.775 s) or SLOW (27.75s) mode)
filter	Imager Filter to use
readDetect	Subarray of detector to read out

Table 6. Parameters of the example exposures

Coron.Mask/Filter	Readout Pattern	Group/Int	Integraions/Exp	Exposures/Dith	Total Dithers	Total Integrations
4QPM/F1140C	FASTR1	200	43	4	1	172

References

- [1] Jonathan P. Gardner et al. "The James Webb Space Telescope". In: *Space Science Reviews* 123.4 (Apr. 2006), pp. 485–606. doi: [10.1007/s11214-006-8315-7](https://doi.org/10.1007/s11214-006-8315-7). URL: <https://doi.org/10.1007%5C%2Fs11214-006-8315-7>.
- [2] NASA. *Webb's Science Themes*. URL: <https://webb.nasa.gov/content/science/>. (accessed: 07.09.2021).
- [3] Northrop Grumman. *James Webb Space Telescope Observing Cosmic History*. URL: <https://www.northropgrumman.com/wp-content/uploads/JWST-Datasheet.pdf>. (accessed: 08.09.2021).
- [4] Space Telescope Science Institute. *MIRI Imaging*. URL: <https://jwst-docs.stsci.edu/jwst-mid-infrared-instrument/miri-observing-modes/miri-imaging>. (accessed: 09.09.2021).
- [5] ESA. *MIRI and Spitzer comparison image*. URL: https://www.esa.int/ESA_Multimedia/Images/2022/05/MIRI_and_Spitzer_comparison_image. (accessed: 08.09.2021).
- [6] NASA. *Webb Key Facts*. URL: <https://webb.nasa.gov/content/about/faqs/facts.html>. (accessed: 06.09.2021).
- [7] NASA. *Exoplanet exploration*. URL: <https://exoplanets.nasa.gov>. (accessed: 07.09.2021).
- [8] Space Telescope Science Institute. *MIRI Detector Overview*. URL: <https://jwst-docs.stsci.edu/jwst-mid-infrared-instrument/miri-instrumentation/miri-detector-overview>. (accessed: 09.09.2021).
- [9] Space Telescope Science Institute. *MIRI Imaging Dithering*. URL: <https://jwst-docs.stsci.edu/jwst-mid-infrared-instrument/miri-operations/miri-dithering/miri-imaging-dithering>. (accessed: 09.09.2021).
- [10] Space Telescope Science Institute. *MIRI Detector Subarrays*. URL: <https://jwst-docs.stsci.edu/jwst-mid-infrared-instrument/miri-instrumentation/miri-detector-overview/miri-detector-subarrays>. (accessed: 09.09.2021).
- [11] Space Telescope Science Institute. *Understanding Exposure Times*. URL: <https://jwst-docs.stsci.edu/understanding-exposure-times>. (accessed: 09.09.2021).
- [12] Space Telescope Science Institute. *MIRI Detector Readout Overview*. URL: <https://jwst-docs.stsci.edu/jwst-mid-infrared-instrument/miri-instrumentation/miri-detector-overview/miri-detector-readout-overview>. (accessed: 09.09.2021).
- [13] Space Telescope Science Institute. *MIRI Detector Readout SLOWR1*. URL: [https://jwst-docs.stsci.edu/jwst-mid-infrared-instrument/miri-instrumentation/miri-detector-overview/miri-detector-readout-slowr1](https://jwst-docs.stsci.edu/jwst-mid-infrared-instrument/miri-instrumentation/miri-detector-overview/miri-detector-readout-overview/miri-detector-readout-slowr1). (accessed: 09.09.2021).
- [14] ESA. *Understanding JWST detectors: A brief guide for the ESA JWST Masterclass*. URL: https://www.cosmos.esa.int/documents/739790/3315704/ESA_JWST_Master_Class_Detectors_Assignment.pdf?7e7aca6-a944-1687-d12d-3d8ea0a6d36a?t=1581523521765. (accessed: 09.09.2021).
- [15] G. H. Rieke et al. "The Mid-Infrared Instrument for the James Webb Space Telescope/i, I: Introduction". In: *Publications of the Astronomical Society of the Pacific* 127.953 (July 2015), pp. 584–594. doi: [10.1086/682252](https://doi.org/10.1086/682252). URL: <https://doi.org/10.1086%5C%2F682252>.
- [16] Alistair Glasse et al. "The Mid-Infrared Instrument for the James Webb Space Telescope/i, IX: Predicted Sensitivity". In: *Publications of the Astronomical Society of the Pacific* 127.953 (July 2015), pp. 686–695. doi: [10.1086%5C%2F682259](https://doi.org/10.1086%5C%2F682259). URL: <https://doi.org/10.1086%5C%2F682259>.
- [17] A. Boccaletti et al. "The Mid-Infrared Instrument for the James Webb Space Telescope/i, V: Predicted Performance of the MIRI Coronagraphs". In: *Publications of the Astronomical Society of the Pacific* 127.953 (July 2015), pp. 633–645. doi: [10.1086/682256](https://doi.org/10.1086/682256). URL: <https://doi.org/10.1086%5C%2F682256>.
- [18] Space Telescope Science Institute. *Optical path for a typical Lyot coronagraph*. URL: <https://jwst-docs.stsci.edu/jwst-mid-infrared-instrument/miri-instrumentation/miri-coronagraphs#MIRICoronagraphs-ref>. (accessed: 08.09.2021).
- [19] Space Telescope Science Institute. *MIRI Coronagraphic Imaging*. URL: <https://jwst-docs.stsci.edu/jwst-mid-infrared-instrument/miri-observing-modes/miri-coronagraphic-imaging>. (accessed: 08.09.2021).
- [20] Mathias Scholz. "Nachweismethoden von Exoplaneten". In: *Planetologie extrasolarer Planeten*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 21–222. ISBN: 978-3-642-41749-8. doi: [10.1007/978-3-642-41749-8_3](https://doi.org/10.1007/978-3-642-41749-8_3). URL: https://doi.org/10.1007/978-3-642-41749-8_3.
- [21] P D Klaassen et al. "mirisim: a simulator for the Mid-Infrared Instrument on JWST". In: *Monthly Notices of the Royal Astronomical Society* 500.3 (Nov. 2020), pp. 2813–2821. ISSN: 0035-8711. doi: [10.1093/mnras/staa3416](https://doi.org/10.1093/mnras/staa3416). eprint: <https://academic.oup.com/mnras/article-pdf/500/3/2813/34561734/staa3416.pdf>. URL: <https://doi.org/10.1093/mnras/staa3416>.
- [22] Vincent Geers. *MIRISim*. URL: https://wiki.miricle.org/Public/MIRISim_Public#Example_Setups. (accessed: 09.09.2021).

- [23] Aarynn L. Carter et al. “Simulating JWST high contrast observations with PanCAKE”. In: *Techniques and Instrumentation for Detection of Exoplanets X*. Ed. by Stuart B. Shaklan and Garreth J. Ruane. Vol. 11823. International Society for Optics and Photonics. SPIE, 2021, 118230H. doi: [10.1117/12.2594501](https://doi.org/10.1117/12.2594501). url: <https://doi.org/10.1117/12.2594501>.
- [24] Space Telescope Science Institute. *MIRI Features and Caveats*. url: https://jwst-docs.stsci.edu/jwst-mid-infrared-instrument/miri-features-and-caveats#MIRIFeaturesandCaveats-glow_sticksGlowsticksintheMIRI4QPMcoronagraphs. (accessed: 13.09.2021).
- [25] Anthony Boccaletti et al. “JWST/MIRI coronagraphic performances as measured on-sky”. In: (July 2022). doi: [10.48550/arXiv.2207.11080](https://arxiv.org/abs/2207.11080).

Eigenständigkeitserklärung

Die unterzeichnete Eigenständigkeitserklärung ist Bestandteil jeder während des Studiums verfassten Semester-, Bachelor- und Master-Arbeit oder anderen Abschlussarbeit (auch der jeweils elektronischen Version).

Die Dozentinnen und Dozenten können auch für andere bei ihnen verfasste schriftliche Arbeiten eine Eigenständigkeitserklärung verlangen.

Ich bestätige, die vorliegende Arbeit selbständig und in eigenen Worten verfasst zu haben. Davon ausgenommen sind sprachliche und inhaltliche Korrekturvorschläge durch die Betreuer und Betreuerinnen der Arbeit.

Titel der Arbeit (in Druckschrift):

Simulating direct exoplanet detection with the James-Webb-Space Telescope

Verfasst von (in Druckschrift):

Bei Gruppenarbeiten sind die Namen aller Verfasserinnen und Verfasser erforderlich.

Name(n):

Viebig

Vorname(n):

Niklas

Ich bestätige mit meiner Unterschrift:

- Ich habe keine im Merkblatt „[Zitier-Knigge](#)“ beschriebene Form des Plagiats begangen.
- Ich habe alle Methoden, Daten und Arbeitsabläufe wahrheitsgetreu dokumentiert.
- Ich habe keine Daten manipuliert.
- Ich habe alle Personen erwähnt, welche die Arbeit wesentlich unterstützt haben.

Ich nehme zur Kenntnis, dass die Arbeit mit elektronischen Hilfsmitteln auf Plagiate überprüft werden kann.

Ort, Datum

Zürich, den 4 Dezember 2022

Unterschrift(en)

Niklas Viebig

Bei Gruppenarbeiten sind die Namen aller Verfasserinnen und Verfasser erforderlich. Durch die Unterschriften bürgen sie gemeinsam für den gesamten Inhalt dieser schriftlichen Arbeit.