The Journal of Open Source Software

ShOpt.jl | A Julia Library for Empirical Point Spread Function Characterization of JWST NIRCam Data

Edward Berman [1¶] **and Jacqueline McCleary** [1¶]

1 Northeastern University, USA ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

In partnership with

AMERICAN ASTRONOMICAL SOCIETY

This article and software are linked with research article DOI 10.3847/xxxxx, published in the Astronomical Journal.

Summary

Introduction

When astronomers image the night sky, the path of the incoming light has been altered by diffraction, optical aberrations, atmospheric turbulence, and telescope jitter. These are summarized in the image's point spread function (PSF), a mathematical model that describes the response of an optical system to an idealized point of light. Because the PSF can resemble or obscure the signal of interest, it must be carefully modeled to extract the maximum amount of information from an observation. Failing to do so can lead to inaccuracies in positions, sizes, and shapes of targets like galaxies.

The goal of PSF characterization is to be able to point to any position on your camera and predict what the distortion looks like. Once we have a model that can do this well, we can deconvolve the PSF to produce images free of distortion.

The PSF characterization methods used by astronomers fall into two main classes: forward-modeling approaches, which use physical optics propagation based on models of the optical array, and empirical approaches, which use stars as fixed points to model and interpolate the PSF across the rest of the image. (Stars are essentially point sources before their light passes through the atmosphere and telescope, so the shape and size of their surface brightness profiles define the PSF at that location.) Empirical PSF characterization proceeds by first cataloging the observed stars, separating the catalog into validation and training samples, and interpolating the training stars across the field of view of the camera. We use the term "vignette" to describe the image stamps containing stars that make up the catalog. After training, the PSF model can be validated by comparing the reserved stars to the PSF model's prediction.

Shear Optimization with `ShOpt.jl` introduces modern techniques for empirical PSF characterization across the field of view that are tailored to James Webb Space Telescope (JWST) imaging. ShOpt has two modes of operation: approximating stars with analytic profiles, and a more realistic pixel-level representation.

Analytic profile mode

A rough idea of the size and shape of the PSF can be obtained by fitting stars with analytic profiles. We adopt a multivariate Gaussian profile, as it is computationally cheap to fit one to an image. That is, Gaussian profiles are easy to differentiate and don't involve any numeric integration or other costly steps to calculate. Fitting other common models, such as a Kolmogorov profile, involves numeric integration and thus take much longer to fit. Moreover, the JWST point spread function is very "spikey" (cf. Figure 1). As a result, analytic profiles are limited in their ability to model the point spread function anyway, making the usual advantages of a more expensive analytic profile moot.

The Journal of Open Source Software

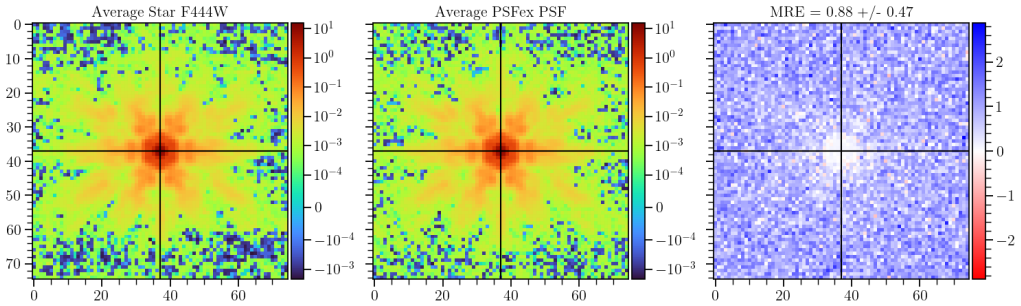

Figure 1: The plot on the left shows the average cutout of all of the stars in a supplied catalog. Likewise, plot in the middle shows the average point spread function model. The plot on the right shows the average normalized error between the observed star cutouts and the point spread function model.

Our multivariate gaussian is parameterized by three variables, $[s, g_1, g_2]$, where s corresponds to size and g_1, g_2 correspond to shear. A shear matrix has the form

$$\begin{pmatrix} 1+g_1 & g_2 \\ g_2 & 1-g_1 \end{pmatrix}$$

. Given a point $[u, v]$, we obtain coordinates $[u', v']$ by applying a shear and then a scaling by $\frac{s}{\sqrt{1-g_1^2-g_2^2}}$. Then, we choose $f(r) := Ae^{-r^2}$ to complete our fit, where A makes the fit sum to unity over the cutout of our star. After we fit this function to our stars with `Optim.jl` (Mogensen & Riseth, 2018) and `ForwardDiff.jl` (Revels et al., 2016), we interpolate the parameters across the field of view according to position. Essentially, each star is a datapoint, and the three variables are treated as polynomials in focal plane coordinates (u, v) of degree n, where n is supplied by the user. The focal plain refers to the set of points where an image appears to be in perfect focus. The units of focal plane coordinates are arcseconds. This is instead of pixel coordinates, where one just uses (x,y) as measured on an image. For a more precise model, we also give each pixel in our star stamp a polynomial and interpolate it across the field of view. That is, each pixel in position (i, j) of a star cutout gets its own polynomial, interpolated over k different star cutouts at different locations in the focal plane. This is referred to in the literature as a pixel basis (Jarvis et al., 2020).

Notation

1. For the set $B_2(r)$, we have:

$$B_2(r) \equiv \{[x, y] : x^2 + y^2 < 1\} \subset \mathbb{R}^2$$

2. For the set $\mathbb{R}_+$, we have:

$$\mathbb{R}_+ \equiv \{x : x > 0\} \subset \mathbb{R}$$

3. For the Cartesian product of sets A and B, we have:

$$A \times B \equiv \{(a, b) : a \in A, b \in B\}$$

Analytic methods

ShOpt.jl's analytic profile fitting takes inspiration from a number of algorithms outside of astronomy, notably SE-Sync (Rosen et al., 2019), an algorithm that solves the robotic mapping problem by considering the manifold properties of the data. With sufficiently clean data, the SE-Sync algorithm will descend to a global minimum constrained to the manifold $SE(d)^n/SE(d)$. Following suit, we are able to put a constraint on the solutions we obtain to $[s, g_1, g_2]$ to a manifold. The solution space to $[s, g_1, g_2]$ is constrained to the manifold $B_2(r) \times \mathbb{R}_+$ (Bernstein & Jarvis, 2002). The existence of the constraint on shear is well known; nevertheless, the parameter estimation task is usually framed as an unconstrained problem (Jarvis et al., 2020). For a more rigorous treatment of optimization on manifolds see (Absil et al., 2008) and (Boumal, 2023). XXX `Julia` has lots of support for working with manifolds with `Manopt`, which we may leverage in future releases (Bergmann, 2022). XXX

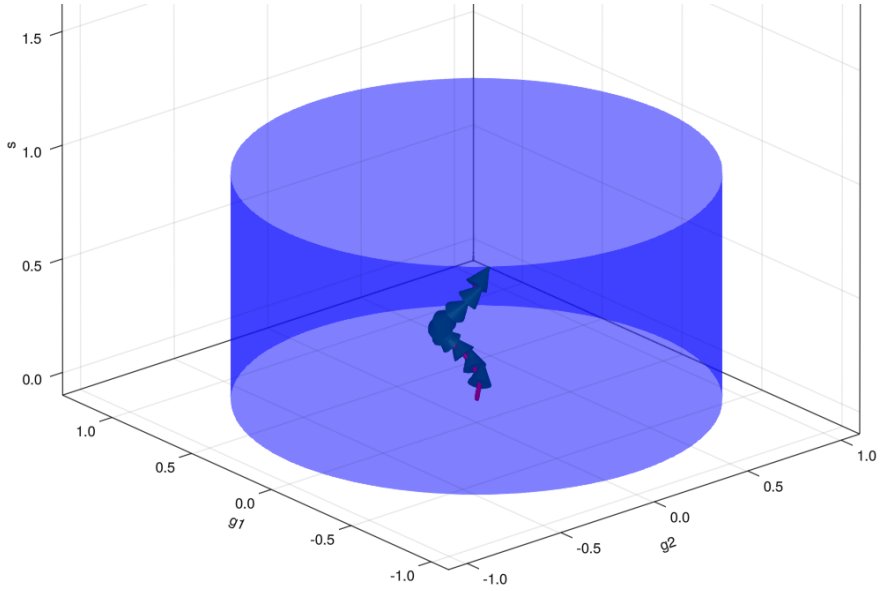

Figure 2: LFBGS algorithm used to find parameters subject to the cylindrical constraint. s is arbitrarily capped at 1 as a data cleaning method.

Pixel grid mode

A more complete description of the PSF can be obtained using the image pixels themselves as a basis, with an interpolation function to model PSF variation across the field of view. `ShOpt.jl` provides two modes for these pixel grid fits: `PCA mode` and `Autoencoder mode`. `PCA mode`, outlined below, reconstructs its images using the first n principal components. `Autoencoder mode` uses a neural network to reconstruct the image from a lower dimensional latent space. The network code written with `Flux.jl` is also outlined below (Innes, 2018). Both modes provide the end user with tunable parameters that allow for both perfect reconstruction of star cutouts and low dimensional representations. The advantage of these modes is that they provide good reconstructions of the distorted images that can learn the key features of the point spread function without overfitting the background noise. In this way it generates a datapoint for our algorithm to train on and denoises the image in one step. In both cases, the input star data is cleaned by first fitting an analytic (Gaussian) PSF profile and rejecting size outliers.

Pixel grid methods

`PCA mode`

```
function pca_image(image, ncomponents)
  #Load img Matrix
  img_matrix = image

  # Perform PCA
  M = fit(PCA, img_matrix; maxoutdim=ncomponents)

  # Transform the image into the PCA space
  transformed = MultivariateStats.transform(M, img_matrix)

  # Reconstruct the image
  reconstructed = reconstruct(M, transformed)

  # Reshape the image back to its original shape
  reconstructed_image = reshape(reconstructed, size(img_matrix)...)
end
```

```
Autoencoder mode

# Encoder
encoder = Chain(
                Dense(r*c, 128, leakyrelu),
                Dense(128, 64, leakyrelu),
                Dense(64, 32, leakyrelu),
                )
#Decoder
decoder = Chain(
                Dense(32, 64, leakyrelu),
                Dense(64, 128, leakyrelu),
                Dense(128, r*c, tanh),
                )
#Full autoencoder
autoencoder = Chain(encoder, decoder)

#x_hat = autoencoder(x)
loss(x) = mse(autoencoder(x), x)

# Define the optimizer
optimizer = ADAM()
```

Statement of need

Empirical PSF characterization tools like PSFEx (Bertin, 2011) and PIFF (Jarvis et al., 2020) are widely popular in astrophysics. However, the quality of PIFF and PSFEx models tends to be quite sensitive to the parameter values used to run the software, with optimization sometimes relying on brute-force guess-and-check runs. PIFF is also notably inefficient for large, well-sampled images, taking hours in the worst cases. Imaging from the James Webb Space Telescope's (JWST) Near Infrared Camera (NIRCam) offers vast scientific opportunities, but also presents new challenges for PSF fitting.

(1) NIRCam PSFs are not well approximated by analytic profiles as seen in Figure 1. This calls for well-thought-out, non-parametric modeling tools and PSF diagnostics that can capture the full dynamic range of the PSF. `ShOpt` provides models and diagnostics that meet the challenges of JWST imaging out of the box.

(2) The pixel scale of the NIRCam detectors are 0.03 and 0.06 arcseconds per pixel. (Be-

ichman et al., 2012a; Rieke et al., 2003, 2005). To fully capture the wings of the PSF at this pixel scale requires vignettes anywhere from 131 by 131 to 301 by 301 pixels across. These vignette sizes are much larger than the sizes used in previous large surveys such as DES (Jarvis et al., 2020) and SuperBIT (McCleary et al., 2023) and forces us to evaluate how well existing PSF fitters scale to this size. The DES and SuperBIT surveys needed PSF sizes of 17 by 17 and 48 by 48, an order of magnitude less than the JWST PSF sizes.

State of the Field

There are several existing empirical PSF fitters in addition to a forward model of the JWST PSFs developed by STScI (Jarvis et al., 2020 ; Bertin, 2011; Perrin et al., 2014 ; Perrin et al., 2012). We describe them here and draw attention to their strengths and weaknesses to motivate the development of `ShOpt.jl`. As described in the statement of need, `PSFex` was one of the first precise and general purpose tools used for empirical PSF fitting. However, `PSFex` produced a systematic size bias of the point spread function with how it calculated spatial variation across the field of view (Jarvis et al., 2020). It was discovered via the analytic profile fits that the size of the point spread function, governed by the variable $[s]$, was underestimated.

`PIFF` (Point Spread Functions in the Full Field of View) followed `PSFex` in the effort to correct this issue. The DES camera was 2.2 degrees across, which was large enough for the size bias to become noticable for their efforts. `PIFF` works in focal plane coordinates as opposed to pixel coordinates which fixes the systematic size bias. Jarvis and DES also used the `Python` libraries of astropy (Astropy Collaboration et al., 2022) and Galsim (Rowe et al., 2015) to make the software more accessible than PSFex to programmers in the astrophysics community. PSFex was written in `C` and has been active for more than 20 years. Due to being so old and written in a low level language it is much less approachable for a community of open source developers. One of the motivations of `ShOpt` was to write astrophysics specific software in `Julia`, because `Julia` provides a nice balance of readability and speed with it's high level functional paradigm and just in time compiler. `ShOpt` works directly in sky coordinates, which minimizes any bias that might be introduced by transformation from pixel to sky coordinates afterwords.

While we do have forwards models of the JWST PSF, these models are for single exposure images. The JWST images are either single exposure or mosaics (Perrin et al., 2014, 2012). Mosaiced images are essentially single exposure detector images averaged together. To account for the rotation of the camera between the capture of images and the wide field of view, there are a number of steps that make applying these forward models to mosaics a non trivial procedure. There is also some recent work being done to generate hybrid models for single exposure data (Lin et al., 2023). Hybrid models take these forward models and add an empirical correction. At present, there is yet to be any widely available software to do this.

The COMOS-Web survey is the largest JWST extragalactic survey according to area and prime time allocation (Casey et al., 2023), and takes up $0.54\ deg^2$ (Beichman et al., 2012b; Rieke et al., 2023). This is a large enough portion of the sky that we should prepare to see a lot of variation across the field of view. This gives `ShOpt` the oppurtunity to validate PIFF's correction for handling PSF variations and test how impactful (or not impactful) PSFex's size bias is.

Future Work

We speculate that petal diagrams may be able to approximate the spikey natures of JWST PSFS. Consider $r = A\cos(k\theta + \gamma)$, shown below in figure 3 for different $[A, k]$ values where $\gamma = 0$. In practice, $[A, k, \gamma]$ could be learnable parameters. Moreover, we could do this for a series of trigonmetric functions to get petals of different sizes. We could then choose some

$f(r) \propto \frac{1}{r}$ such that the gray fades from black to white. We would define $f(r)$ piece wise such that it is 0 outside of the petal and decreases radially with r inside the petal.

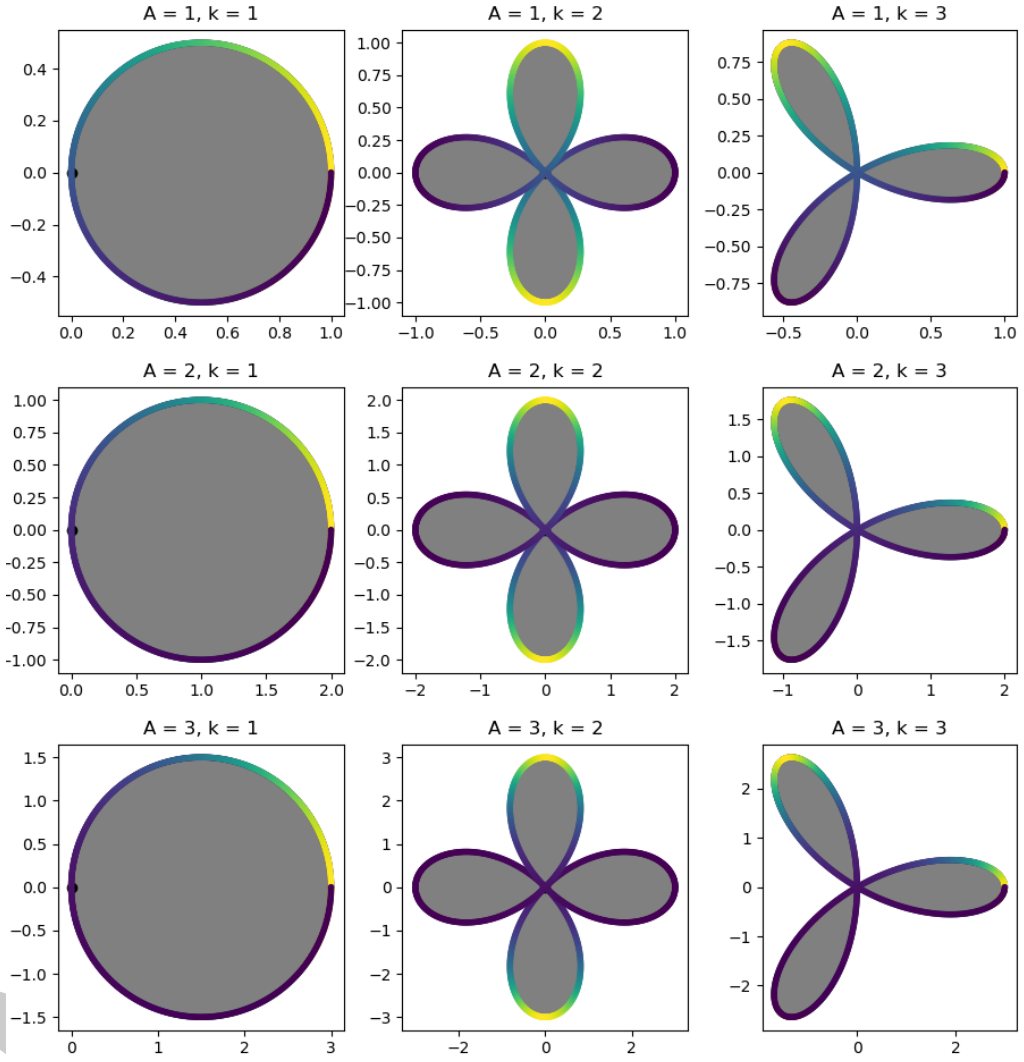

Figure 3: Petal Diagram

Acknowledgements

This material is based upon work supported by a Northeastern University Undergraduate Research and Fellowships PEAK Experiences Award. We would also like to thank the Northeastern Physics Department for making this project possible through the Physics Co-op Research Fellowship. Support for COSMOS-Web was provided by NASA through grant JWST-GO-01727 and HST-AR-15802 awarded by the Space Telescope Science Institute, which is operated by the Association of Universities for Research in Astronomy, Inc., under NASA contract NAS 5-26555. This work was made possible by utilizing the CANDIDE cluster at the Institut d'Astrophysique de Paris. Finally, we would like to thank Northeastern Research Computing for access to their servers. Additionally, we'd like to extend a thank you to Professor David Rosen for giving some valuable insights during the early stages of this work.

References

Absil, P.-A., Mahony, R., & Sepulchre, R. (2008). *Optimization algorithms on matrix manifolds* (p. xvi+224). Princeton University Press. ISBN: 978-0-691-13298-3

Astropy Collaboration, Price-Whelan, A. M., Lim, P. L., Earl, N., Starkman, N., Bradley, L., Shupe, D. L., Patil, A. A., Corrales, L., Brasseur, C. E., Nöthe, M., Donath, A., Tollerud, E., Morris, B. M., Ginsburg, A., Vaher, E., Weaver, B. A., Tocknell, J., Jamieson, W., … Astropy Project Contributors. (2022). The Astropy Project: Sustaining and Growing a Community-oriented Open-source Project and the Latest Major Release (v5.0) of the Core Package. *935*(2), 167. https://doi.org/10.3847/1538-4357/ac7c74

Beichman, C. A., Rieke, M., Eisenstein, D., Greene, T. P., Krist, J., McCarthy, D., Meyer, M., & Stansberry, J. (2012b). Science opportunities with the near-IR camera (NIRCam) on the James Webb Space Telescope (JWST). In M. C. Clampin, G. G. Fazio, H. A. MacEwen, & J. M. O. Jr. (Eds.), *Space telescopes and instrumentation 2012: Optical, infrared, and millimeter wave* (Vol. 8442, p. 84422N). International Society for Optics; Photonics; SPIE. https://doi.org/10.1117/12.925447

Beichman, C. A., Rieke, M., Eisenstein, D., Greene, T. P., Krist, J., McCarthy, D., Meyer, M., & Stansberry, J. (2012a). Science opportunities with the near-IR camera (NIRCam) on the James Webb Space Telescope (JWST). In M. C. Clampin, G. G. Fazio, H. A. MacEwen, & Jr. Oschmann Jacobus M. (Eds.), *Space telescopes and instrumentation 2012: Optical, infrared, and millimeter wave* (Vol. 8442, p. 84422N). https://doi.org/10.1117/12.925447

Bergmann, R. (2022). Manopt.jl: Optimization on manifolds in Julia. *Journal of Open Source Software*, *7*(70), 3866. https://doi.org/10.21105/joss.03866

Bernstein, G. M., & Jarvis, M. (2002). Shapes and shears, stars and smears: Optimal measurements for weak lensing. *The Astronomical Journal*, *123*(2), 583. https://doi.org/10.1086/338085

Bertin, E. (2011). Automated Morphometry with SExtractor and PSFEx. In I. N. Evans, A. Accomazzi, D. J. Mink, & A. H. Rots (Eds.), *Astronomical data analysis software and systems XX* (Vol. 442, p. 435).

Boumal, N. (2023). *An introduction to optimization on smooth manifolds*. Cambridge University Press. https://doi.org/10.1017/9781009166164

Casey, C. M., Kartaltepe, J. S., Drakos, N. E., Franco, M., Harish, S., Paquereau, L., Ilbert, O., Rose, C., Cox, I. G., Nightingale, J. W., Robertson, B. E., Silverman, J. D., Koekemoer, A. M., Massey, R., McCracken, H. J., Rhodes, J., Akins, H. B., Amvrosiadis, A., Arango-Toro, R. C., … Zavala, J. A. (2023). *COSMOS-web: An overview of the JWST cosmic origins survey*. https://arxiv.org/abs/2211.07865

Innes, M. (2018). Flux: Elegant machine learning with julia. *Journal of Open Source Software*. https://doi.org/10.21105/joss.00602

Jarvis, M., Bernstein, G. M., Amon, A., Davis, C., Lé get, P. F., Bechtol, K., Harrison, I., Gatti, M., Roodman, A., Chang, C., Chen, R., Choi, A., Desai, S., Drlica-Wagner, A., Gruen, D., Gruendl, R. A., Hernandez, A., MacCrann, N., Meyers, J., … and, R. D. W. (2020). Dark energy survey year 3 results: Point spread function modelling. *Monthly Notices of the Royal Astronomical Society*, *501*(1), 1282–1299. https://doi.org/10.1093/mnras/staa3679

Lin, Nie, Huanyuan, Shan, Guoliang, Li, Lei, Wang, Charling, Tao, Qifan, Cui, Yushan, Xie, Dezi, Liu, Zekang, & Zhang. (2023). *HybPSF: Hybrid PSF reconstruction for the observed JWST NIRCam image*. https://arxiv.org/abs/2308.14065

McCleary, J. E., Everett, S. W., Shaaban, M. M., Gill, A. S., Vassilakis, G. N., Huff, E. M., Massey, R. J., Benton, S. J., Brown, A. M., Clark, P., & others. (2023). Lensing in the

blue II: Estimating the sensitivity of stratospheric balloons to weak gravitational lensing. *arXiv Preprint arXiv:2307.03295*.

Mogensen, P. K., & Riseth, A. N. (2018). Optim: A mathematical optimization package for julia. *Journal of Open Source Software*, *3*(24), 615. https://doi.org/10.21105/joss.00615

Perrin, M. D., Sivaramakrishnan, A., Lajoie, C.-P., Elliott, E., Pueyo, L., Ravindranath, S., & Albert, Loic. (2014). Updated point spread function simulations for JWST with WebbPSF. In Jr. Oschmann Jacobus M., M. Clampin, G. G. Fazio, & H. A. MacEwen (Eds.), *Space telescopes and instrumentation 2014: Optical, infrared, and millimeter wave* (Vol. 9143, p. 91433X). https://doi.org/10.1117/12.2056689

Perrin, M. D., Soummer, R., Elliott, E. M., Lallo, M. D., & Sivaramakrishnan, A. (2012). Simulating point spread functions for the James Webb Space Telescope with WebbPSF. In M. C. Clampin, G. G. Fazio, H. A. MacEwen, & Jr. Oschmann Jacobus M. (Eds.), *Space telescopes and instrumentation 2012: Optical, infrared, and millimeter wave* (Vol. 8442, p. 84423D). https://doi.org/10.1117/12.925230

Revels, J., Lubin, M., & Papamarkou, T. (2016). Forward-mode automatic differentiation in Julia. *arXiv:1607.07892 [Cs.MS]*. https://arxiv.org/abs/1607.07892

Rieke, M. J., Baum, S. A., Beichman, C. A., Crampton, D., Doyon, R., Eisenstein, D., Greene, T. P., Hodapp, K.-W., Horner, S. D., Johnstone, D., Lesyna, L., Lilly, S., Meyer, M., Martin, P., Jr., D. W. M., Rieke, G. H., Roellig, T. L., Stauffer, J., Trauger, J. T., & Young, E. T. (2003). NGST NIRCam scientific program and design concept. In J. C. Mather (Ed.), *IR space telescopes and instruments* (Vol. 4850, pp. 478–485). International Society for Optics; Photonics; SPIE. https://doi.org/10.1117/12.489103

Rieke, M. J., Kelly, D. M., Misselt, K., Stansberry, J., Boyer, M., Beatty, T., Egami, E., Florian, M., Greene, T. P., Hainline, K., Leisenring, J., Roellig, T., Schlawin, E., Sun, F., Tinnin, L., Williams, C. C., Willmer, C. N. A., Wilson, D., Clark, C. R., … Young, E. T. (2023). Performance of NIRCam on JWST in flight. *Publications of the Astronomical Society of the Pacific*, *135*(1044), 028001. https://doi.org/10.1088/1538-3873/acac53

Rieke, M. J., Kelly, D., & Horner, S. (2005). Overview of James Webb Space Telescope and NIRCam's Role. In J. B. Heaney & L. G. Burriesci (Eds.), *Cryogenic optical systems and instruments XI* (Vol. 5904, pp. 1–8). https://doi.org/10.1117/12.615554

Rosen, D. M., Carlone, L., Bandeira, A. S., & Leonard, J. J. (2019). SE-sync: A certifiably correct algorithm for synchronization over the special euclidean group. *The International Journal of Robotics Research*, *38*(2-3), 95–125. https://doi.org/10.1177/0278364918784361

Rowe, B., Jarvis, M., Mandelbaum, R., Bernstein, G. M., Bosch, J., Simet, M., Meyers, J. E., Kacprzak, T., Nakajima, R., Zuntz, J., Miyatake, H., Dietrich, J. P., Armstrong, R., Melchior, P., & Gill, M. S. S. (2015). *GalSim: The modular galaxy image simulation toolkit*. https://arxiv.org/abs/1407.7676