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

Summary

Introduction

When astronomers capture images of the night sky, several factors – ranging from diffraction and optical aberrations to atmospheric turbulence and telescope jitter – affect the incoming light. The resulting distortions 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. The PSF can obscure or even mimic the astronomical signal of interest, making its accurate characterization essential. By effectively modeling the PSF, we can predict image distortions at any location and proceed to deconvolve the PSF, ultimately reconstructing distortion-free images.

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 optics, 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. 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, tailored to James Webb Space Telescope (JWST) imaging, for empirical PSF characterization across the field of view. ShOpt has two modes of operation: approximating stars with analytic profiles, and a more realistic pixel-level representation. Both modes take as input a catalog with image cutouts – or "vignettes" – of the stars targeted for analysis.

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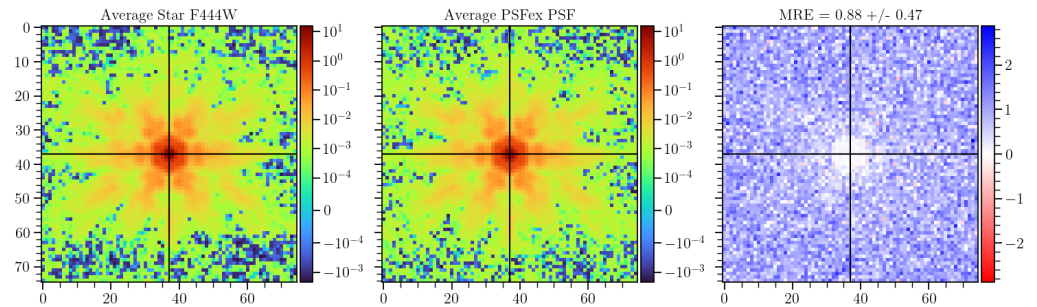
Figure 1: The plot on the left shows the average cutout of all stars in a supplied catalog. The plot in the middle shows the average point spread function model for each star. The plot on the right shows the average normalized error between the observed star cutouts and the point spread function model.

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

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

40 . Given a point $[u, v]$, we obtain coordinates $[u', v']$ by applying a shear and then a scaling
41 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
42 sum to unity over the cutout of our star. After we fit this function to our stars with `Optim.jl`
43 (Mogensen & Riseth, 2018) and `ForwardDiff.jl` (Revels et al., 2016), we interpolate the
44 parameters across the field of view according to position. Essentially, each star is a datapoint,
45 and the three variables are treated as polynomials in the focal plane. We express positions
46 in sky (astrometric) coordinates (u, v), as opposed to pixel coordinates (x, y) measured in
47 detector pixels.

48 **Notation**

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

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

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

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

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

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

52 **Analytic methods**

53 `ShOpt.jl`'s analytic profile fitting takes inspiration from a number of algorithms outside
54 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). `Julia` offers support for working with manifolds through the Manopt framework, which we may leverage in future releases (Bergmann, 2022).
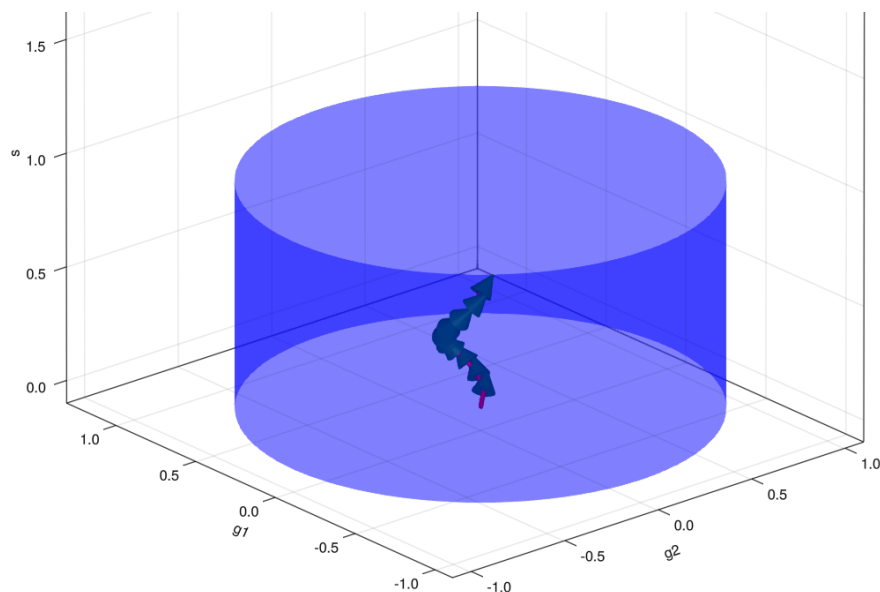

Figure 2: LFBGS algorithm used to find parameters subject to the cylindrical constraint. s is arbitrarily capped at 1 as a data cleaning method.

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. As in the analytic profile case, star positions are expressed directly in sky (astrometric) coordinates (u, v) rather than pixel coordinates (x, y). In the pixel grid modes, we also model each pixel in the star stamp as polynomial to be interpolated 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.

Pixel grid methods

PCA mode

```julia
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

```julia
# 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. The James Webb Space Telescope's (JWST) Near Infrared Camera (NIRCam) offers vast scientific opportunities (e.g., (Casey et al., 2023)); at the same time, this unprecedented data brings new challenges for PSF modeling:

(1) Analytic functions like Gaussians are incomplete descriptions of the NIRCam PSF, as evident from Figure 1. This calls for well-thought-out, non-parametric modeling and diagnostic tools that can capture the full dynamic range of the NIRCam PSF. ShOpt provides these models and diagnostics out of the box.

(2) The NIRCam detectors have pixel scales of 0.03 (short wavelength channel) and 0.06

98 (long wavelength channel) arcseconds per pixel (Beichman et al., 2012a; Rieke et al.,
99 2003, 2005). At these pixel scales, star vignettes need to be at least 131 by 131 pixels
100 across to fully capture the wings of the PSFs (4-5 arcseconds). These vignette sizes are
101 3-5 times larger than the ones used in surveys such as DES (Jarvis et al., 2020) and
102 SuperBIT (McCleary et al., 2023) and force us to evaluate how well existing PSF fitters
103 scale to this size. `ShOpt` has been designed for computational efficiency and aims to
104 meet the requirements of detectors like NIRCam.

State of the Field

106 There are several existing empirical PSF fitters, in addition to a forward model of the JWST
107 PSFs developed by STScI (Jarvis et al., 2020 ; Bertin, 2011; Perrin et al., 2014 ; Perrin et
108 al., 2012). We describe them here and draw attention to their strengths and weaknesses to
109 motivate the development of `ShOpt.jl`. As described in the statement of need, `PSFex` was one
110 of the first precise and general purpose tools used for empirical PSF fitting. However, the Dark
111 Energy Survey collaboration reported small but noticeable discrepancies between the sizes of
112 `PSFex` models and the sizes of observed stars. They also reported ripple-like patterns in the
113 spatial variation of star-PSF residuals across the field of view (Jarvis et al., 2020), which they
114 attributed to the astrometric distortion solutions for the Dark Energy Camera.

115 These findings motivated the Dark Energy Survey's development of `PIFF` (Point Spread
116 Functions in the Full Field of View). `PIFF` works in sky coordinates on the focal plane, as
117 opposed to image pixel coordinates used in `PSFex`, which minimized the ripple patterns in
118 the star-PSF residuals and the PSF model size bias. (Based on the DES findings, `ShOpt`
119 also works directly in sky coordinates.) `PIFF` is written in Python, a language with a large
120 infrastructure for astronomical data analysis, for example Astropy (Astropy Collaboration et
121 al., 2022) and Galsim (Rowe et al., 2015). The choice of language makes `PIFF` software
122 more accessible to programmers in the astrophysics community than `PSFex`, which was first
123 written in `C` twenty-five years ago and much less approachable for a community of open source
124 developers. As an aside, one of the motivations of `ShOpt` was to write astrophysics specific
125 software in `Julia`, because `Julia` provides a nice balance of readability and speed with its
126 high-level functional paradigm and just-in-time compiler.

127 While WebbPSF provides highly precise forward models of the JWST PSF, these models
128 are defined for single-epoch exposures (Perrin et al., 2014, 2012). Much of the NIRCam
129 science is accomplished with image mosaics – essentially, the combination of single exposure
130 detector images into a larger, deeper image. The rotation of the camera between exposures,
131 the astrometric transformations and resampling of images before their combination into a
132 mosaic, and the mosaic's large area all make the application of WebbPSF models to mosaics
133 a non-trivial procedure. Additionally, some recent work being done to generate hybrid PSF
134 models, which add an empirical correction to forward-model PSFs, for single-epoch exposures
135 (Lin et al., 2023). At the time of writing, there is no widely available software to do this.

Future Work

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

143 We speculate that petal diagrams may be able to approximate the spikey natures of JWST
144 PSFS. Consider $r = A \cos(k\theta + \gamma)$, shown below in figure 3 for different $[A, k]$ values where
145 $\gamma = 0$. 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.
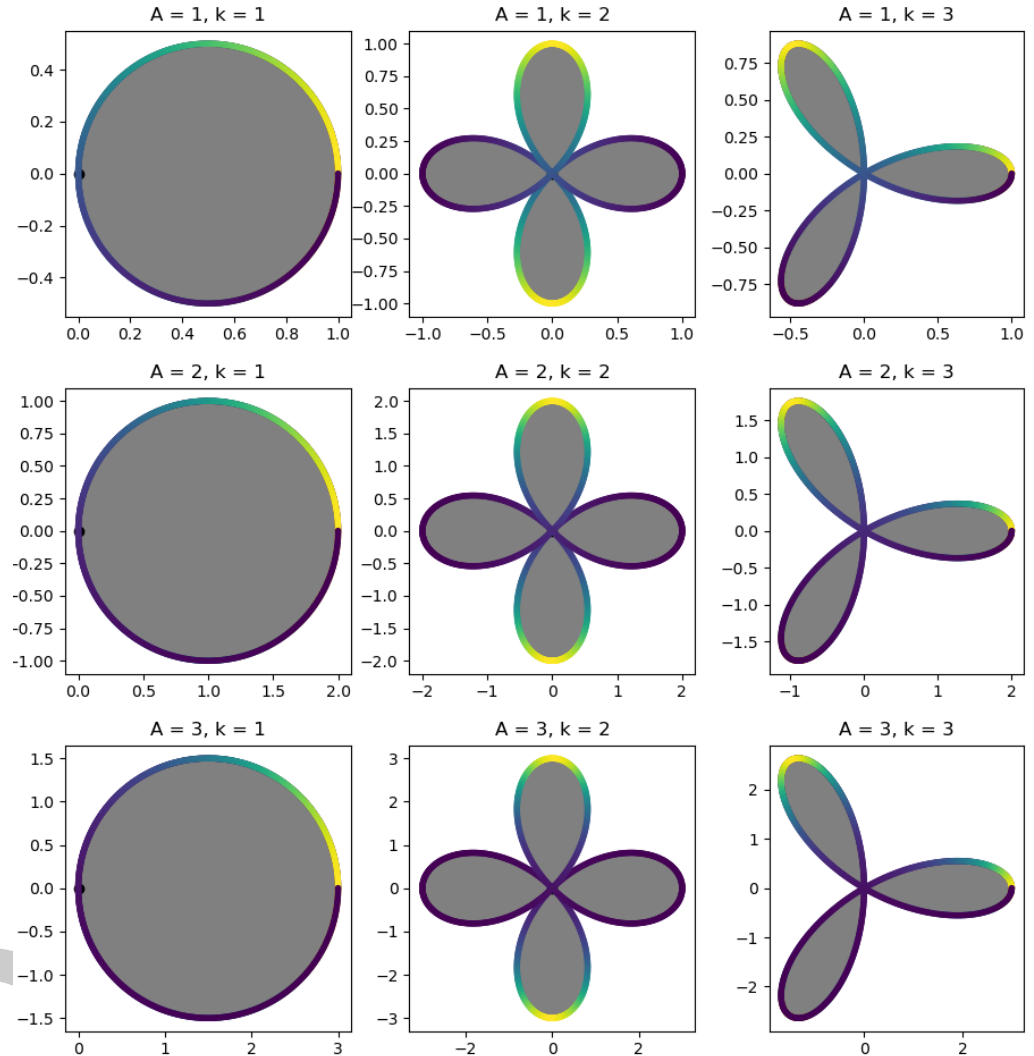

Figure 3: Petal Diagram

Acknowledgements

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

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

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

217 McCleary, J. E., Everett, S. W., Shaaban, M. M., Gill, A. S., Vassilakis, G. N., Huff, E. M.,
218 Massey, R. J., Benton, S. J., Brown, A. M., Clark, P., & others. (2023). Lensing in the
219 blue II: Estimating the sensitivity of stratospheric balloons to weak gravitational lensing.
220 *arXiv Preprint arXiv:2307.03295*.

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

223 Perrin, M. D., Sivaramakrishnan, A., Lajoie, C.-P., Elliott, E., Pueyo, L., Ravindranath, S., &
224 Albert, Loic. (2014). Updated point spread function simulations for JWST with WebbPSF.
225 In Jr. Oschmann Jacobus M., M. Clampin, G. G. Fazio, & H. A. MacEwen (Eds.), *Space*
226 *telescopes and instrumentation 2014: Optical, infrared, and millimeter wave* (Vol. 9143, p.
227 91433X). https://doi.org/10.1117/12.2056689

228 Perrin, M. D., Soummer, R., Elliott, E. M., Lallo, M. D., & Sivaramakrishnan, A. (2012).
229 Simulating point spread functions for the James Webb Space Telescope with WebbPSF. In
230 M. C. Clampin, G. G. Fazio, H. A. MacEwen, & Jr. Oschmann Jacobus M. (Eds.), *Space*
231 *telescopes and instrumentation 2012: Optical, infrared, and millimeter wave* (Vol. 8442, p.
232 84423D). https://doi.org/10.1117/12.925230

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

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

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

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

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

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