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Published: unpublished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 take pictures of space, optical properties of the camera and atmospheric affects distort the incoming light. Stars are examples of what astronomers call point sources. Before any distortion is applied to an image, point sources of light can be thought of as delta functions. The aptly named point spread function (PSF) is a mathematical model that quantifies exactly how the light is being distorted. The point spread function takes as input a delta function and a position and outputs a distorted image. It can be though of as the impulse response of an optical system to incoming light. The goal of point spread function 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 our images with the point spread function to obtain what the image would look like in the absense of distortion. The empirical way to do this is to take our images of distorted stars and separate them into training and validation sets. Our point spread function will be found by interpolating the training stars across the field of view of the camera and validated by comparing the reserved stars to the point spread function models prediction.

Shear Optimization with `ShOpt.jl` introduces modern techniques for empirical point spread function characterization across the field of view tailored to the data captured by the James Webb Space Telescope (JWST). We can approximate our stars with analytic profiles. This gives us a rough idea of the size and shape of the point spread function and doubles as a mechanism to clean data. We adopt a multivariate gaussian profile because it is computationally cheap to fit to an image. That is, it is easy to differentiate and doesn't involve any numeric integration or other costly steps to calculate. Other common fits such as Kolmogorov fits involve numeric integration and thus take much longer to fit. The JWST point spread functions are very "spikey" as seen below, and as a result, analytic profiles are limited in their ability to model the point spread function. Thus, the usual advantages of a more expensive analytic profile are mute.

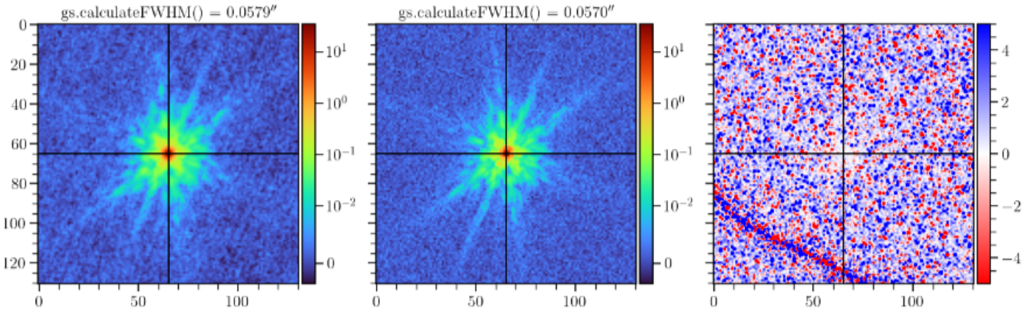

Figure 1: The plot on the left represents the average cutout of all of the stars in a supplied catalog. Likewise, plot in the middle represents the average point spread function model. The plot on the right represents the normalized error between the observed star cutouts and the point spread function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

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

Methods

`ShOpt.jl` takes inspiration from a number of algorithms outside of astronomy. Mainly, SE-Sync (Rosen et al., 2019), an algorithm that provides a certifiably correct solution to a robotic mapping problem by considering the manifold properites of the data. SE-Sync proves that with sufficiently clean data, their algorithm will descend to a global minimum constrained to the manifold $SE(d)^n/SE(d)$. Likewise, we are able to put a constraint on the solutions we obtain to $[s, g_1, g_2]$ to a manifold. (Bernstein & Jarvis, 2002) outlined that the solution to $[s, g_1, g_2]$ is constrained to the manifold $B_2(r) \times \mathbb{R}_+$. While it was known that this constrain existed in the literature, the parameter estimation task is generally 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` has lots of support for working with manifolds with `Manopt`, which we may leverage in future releases (Bergmann, 2022).

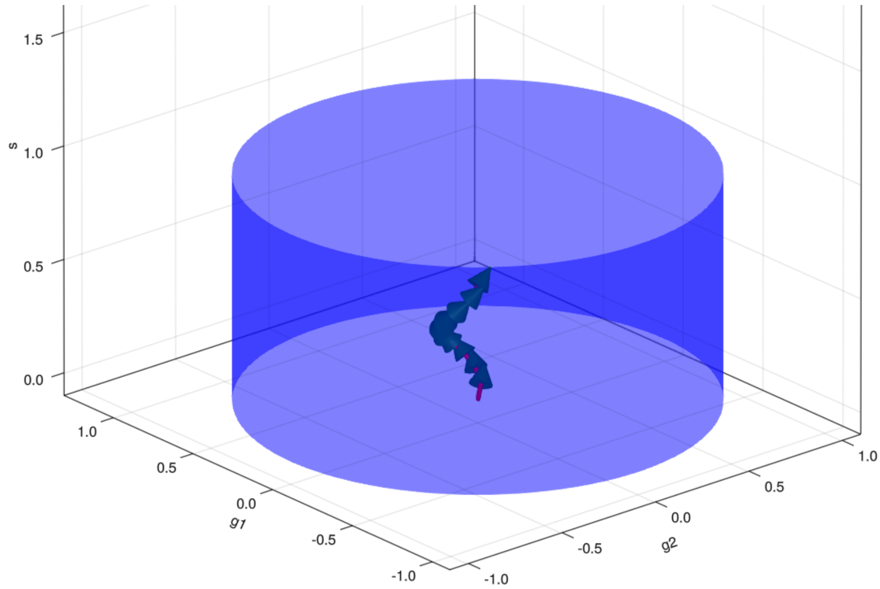

Figure 2: LFBGS algorithm used to find parameters subject to the cylindrical constraint. s is arbitrarily capped at 1 as a data cleaning method.

`ShOpt.jl` provides two modes for 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.

`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

```
# 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

We can trace the first empirical PSF fitters back to DAOPHOT (Stetson, 1987). PSFex made major advancements in precise PSF modeling. With PSFex, you could interpolate several different bases, including a basis of pixels, instead of relying on simple parametric functions. PSFex was built as a general purpose tool and to this day is widely used. Newer empirical PSF fitters are geared toward large scale surveys and the difficulties that arise specific to those datasets. As an example, The Dark Energy Survey and DESCam (Flaugher et al., 2015; Jarvis et al., 2020) sparked the creation of PIFF. The recent data from the James Webb Space Telescope poses new challenges.

(1) The JWST PSFs are not well approximated by analytic profiles as seen in Figure 1. This calls for well thought out parametric free models that can capture the full dynamic range of the Point Spread Function without fixating on the noise in the background. Previously, Rowe statistics and other parametic equations were used to diagnose PSF accurarcy (Rowe, 2010). ShOpt provides a suite of parametric free summary statistics out of the box.

(2) The NIRCam detectors are 0.03"/pix or 0.06" /pix (Gardner et al., 2006). To capture an accurate description of the point spread function at this scale we need images that are 131 by 131 to 261 by 261 pixels across. These vignet sizes are much larger in comparison to the sizes needed for 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 sky 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.

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.

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., 2012; 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. 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.

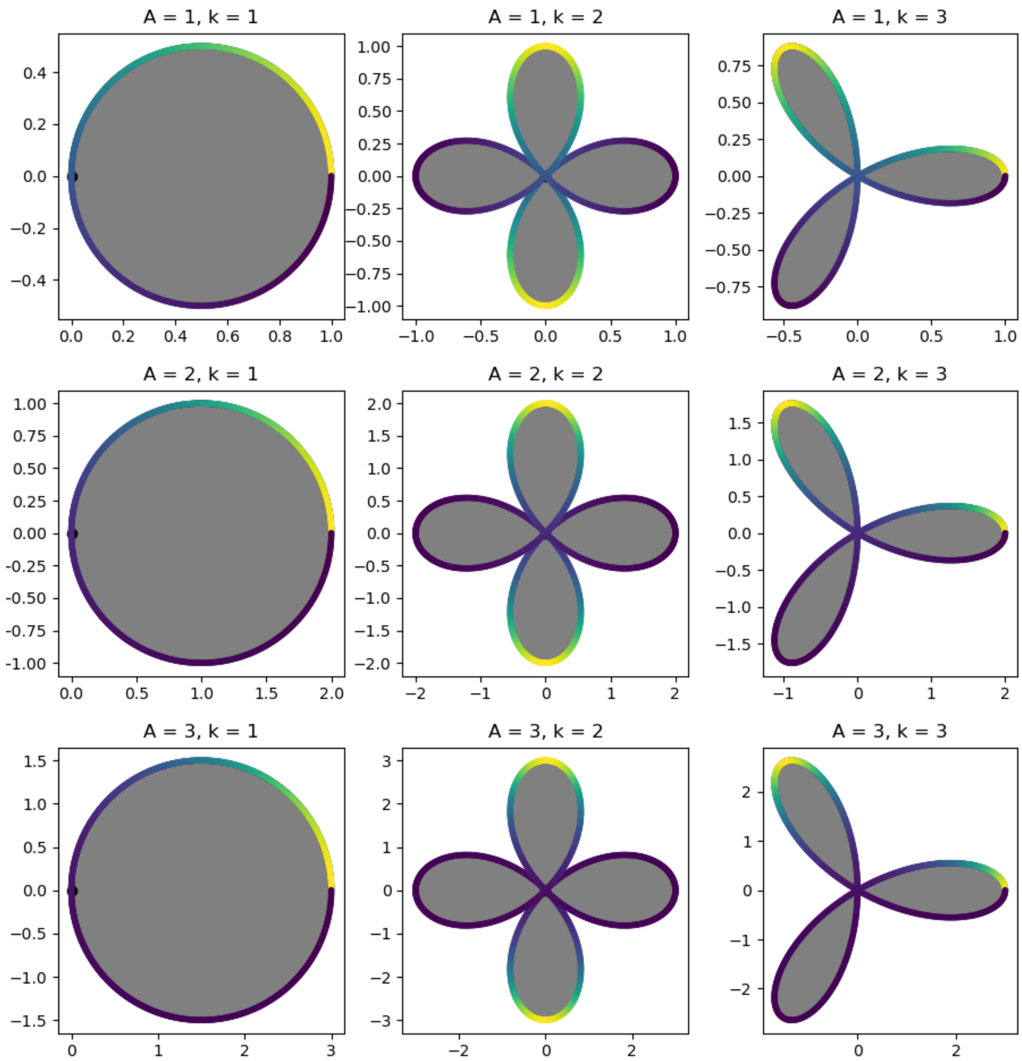

Figure 3: Petal Diagram

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 this work 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. (2012). 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

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

Flaugher, B., Diehl, H. T., Honscheid, K., Abbott, T. M. C., & others. (2015). The dark energy camera. *AJ*, *150*, 150. https://doi.org/10.1088/0004-6256/150/5/150

Gardner, J. P., Mather, J. C., Clampin, M., Doyon, R., Greenhouse, M. A., Hammel, H. B., Hutchings, J. B., Jakobsen, P., Lilly, S. J., Long, K. S., Lunine, J. I., Mccaughrean, M. J., Mountain, M., Nella, J., Rieke, G. H., Rieke, M. J., Rix, H.-W., Smith, E. P., Sonneborn, G., … Wright, G. S. (2006). The james webb space telescope. *Space Science Reviews*, *123*(4), 485–606. https://doi.org/10.1007/s11214-006-8315-7

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

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., 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

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. (2010). Improving PSF modelling for weak gravitational lensing using new methods in model selection. *404*(1), 350–366. https://doi.org/10.1111/j.1365-2966.2010.16277.x

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

Stetson, P. B. (1987). DAOPHOT: A COMPUTER PROGRAM FOR CROWDED-FIELD STELLAR PHOTOMETRY. *Publications of the Astronomical Society of the Pacific*, *99*(613), 191. https://doi.org/10.1086/131977