# pysersic: A Python package for determining galaxy structural properties via Bayesian inference, accelerated with jax

Imad Pasha [1,2*] **and Tim B. Miller** [1*]

1 Department of Astronomy, Yale University, USA **2** National Science Foundation Graduate Research Fellow ***** These authors contributed equally.

Summary

The modern standard for measuring structural parameters of galaxies involves a forward-modeling procedure in which parametric models are fit directly to images while accounting for the effect of the point-spread function (PSF). This is an integral step in many extragalactic studies. The most common parametric form is a Sérsic profile (Sersic, 1968) which is described by a radial surface brightness profiles following,

$$I(R) \propto F_{\text{total}} \exp\left[\left(\frac{R}{R_e}\right)^{1/n} - 1\right],$$

where the total flux, F_{total}, half-light radius, R_e and Sérsic index, n are the parameters of interest to be fit and subsequently used to characterize a galaxy's morphology.

Here we present pysersic, a Bayesian framework created to facilitate the inference of structural parameters from galaxy images. It is written in pure Python, and built using the jax framework (Bradbury et al., 2018) allowing for just-in-time (JIT) compilation, auto-differentiation and seamless execution on CPUs, GPUs or TPUs. Inference is performed with the numpyro (Bingham et al., 2019; Phan et al., 2019) package utilizing gradient based methods, e.g., No U-Turn Sampling (NUTS) (Hoffman et al., 2014), for efficient and robust posterior estimation. pysersic was designed to have a user-friendly interface, allowing users to fit single or multiple sources in a few lines of code. It was also designed to scale to many images, such that it can be seamlessly integrated into current and future analysis pipelines.

Statement of need

Parametric profile fitting has become a ubiquitous and essential tool for numerous applications including measuring the photometry — or total flux — of galaxies, as well as the investigation of the structural evolution of galaxies over cosmic time (Kawinwanichakij et al., 2021; Lange et al., 2015; Mowla et al., 2019). This approach allows one to both extrapolate galaxy surface brightness profiles beyond the noise limit of images, as well as account for the PSF to accurately measure the structure of galaxies near the resolution limit of those images. The empirically derived Sérsic profile is the most common parametric form for the surface-brightness profile as it provides a reasonable approximation to nearly all galaxies, given the additional freedom of the Sérsic index, n, over fixed-index profiles.

Given the long history of Sérsic fitting codes with many available tools, the development of pysersic was largely motivated by two related factors, first and foremost of which was the

desire to implement Sérsic fitting in a fully Bayesian context *at speed*. The *ability* to place the typical Sérsic fitting problem into a Bayesian context with runtimes that are not prohibitive (the traditional drawback of MCMC methods) has recently been unlocked by the second motivation: to leverage the `jax` library. `jax` utilizes JIT compilation to decrease computational runtimes, provides seamless integration with hardware accelerators such as GPUs and TPUs for further improvements in performance, and enables automatic differentiation, facilitating gradient based optimization and sampling methods. Together, these features greatly increase speed and efficiency, especially when sampling or optimizing a large number of parameters.

Inference in `pysersic` is implemented using the `numpyro` probabilistic programming language (PPL). This allows for total control over the priors and methods used for inference. The `numpyro` package utilizes `jax`'s auto-differentiation capabilities for gradient based samplers such as Hamiltonian Monte Carlo (HMC) and No-U-Turn-Sampling (NUTS). In addition, there are recently-developed techniques for posterior estimation, including variational inference ([Ranganath et al., 2014](#)) utilizing normalizing flows ([De Cao et al., 2020](#)). These techniques dramatically reduce the number of likelihood calls required to provide accurate estimates of the posterior relative to gradient-free methods. Combined with the `jax`'s JIT compilation, posteriors can now be generated in a few minutes or less on modern laptops.

Code Description

`pysersic` was designed to have a user-friendly API with sensible defaults. Tools are provided to automatically generate priors for all free parameters based on an initial characterization of a given image — but can also easily be set manually. We provide default inference routines for NUTS MCMC and variational inference using neural flows. Users can access the underlying `numpyro` model if desired, to perform inference using any tools available within the `numpyro` ecosystem. The goal for `pysersic` is to provide reasonable defaults for new users interested in a handful of galaxies, yet maintain the ability for advanced users to tweak options as necessary to perform inference for entire surveys.

A crucial component of any Sérsic fitting code is an efficient and accurate rendering algorithm. Sérsic profiles with high index, $n \gtrsim 3$ are notoriously difficult to render accurately given the steep increase in brightness as $r \rightarrow 0$. In `pysersic`, the `rendering` module is kept separate from the frontend API and inference modules, such that different algorithms can be interchanged and therefore easily tested (and hopefully encourage innovation as well). In this initial release, we provide three algorithms. The first is a traditional rendering algorithm in which the intrinsic profile is rendered in real space, with oversampling in the center to ensure accurate results for high index profiles. The second and third methods render the profiles in Fourier space, providing accurate results even for strongly peaked profiles and avoiding artifacts due to pixelization. In `pysersic`, this is achieved by representing the profiles using a series of Gaussian following the algorithm presented in Shajib ([2019](#)). We include one algorithm that is fully based in Fourier space, along with a version of the hybrid real-Fourier algorithm introduced in Lang ([2020](#)) which helps avoid some of the aliasing present when rendering solely in Fourier space.

Related Software

There is a long history and many software tools designed for Sérsic profile fitting. Some of the most popular libraries are listed below.

- `galfit` ([Peng et al., 2002](#))
- `imfit` ([Erwin, 2015](#))
- `profit` ([Robotham et al., 2017](#))
- `galight` ([Ding et al., 2021](#)), which is built on top of `lenstronomy` ([Birrer et al., 2021](#))
- `PetroFit` ([Geda et al., 2022](#))

- PyAutoGalaxy ([Nightingale et al., 2023](#))

Software Citations

`pysersic` makes use of the following packages:

- arviz ([Kumar et al., 2019](#))
- asdf ([D'Avella et al., 2023](#))
- astropy ([Astropy Collaboration et al., 2013, 2018, 2022](#))
- corner ([Foreman-Mackey, 2016](#))
- jax ([Bradbury et al., 2018](#))
- matplotlib ([Hunter, 2007](#))
- numpy ([Harris et al., 2020](#))
- numpyro ([Bingham et al., 2019](#); [Phan et al., 2019](#))
- pandas ([team, 2020](#))
- photutils ([Bradley et al., 2022](#))
- pytest ([Krekel et al., 2004](#))
- scipy ([Virtanen et al., 2020](#))
- tqdm ([Costa-Luis et al., 2023](#))

Acknowledgements

References

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., N"othe, 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. *Astrophysical Journal*, *935*(2), 167. https://doi.org/10.3847/1538-4357/ac7c74

Astropy Collaboration, Price-Whelan, A. M., Sipőcz, B. M., Günther, H. M., Lim, P. L., Crawford, S. M., Conseil, S., Shupe, D. L., Craig, M. W., Dencheva, N., Ginsburg, A., Vand erPlas, J. T., Bradley, L. D., Pérez-Suárez, D., de Val-Borro, M., Aldcroft, T. L., Cruz, K. L., Robitaille, T. P., Tollerud, E. J., … Astropy Contributors. (2018). The Astropy Project: Building an Open-science Project and Status of the v2.0 Core Package. *Astronomical Journal*, *156*(3), 123. https://doi.org/10.3847/1538-3881/aabc4f

Astropy Collaboration, Robitaille, T. P., Tollerud, E. J., Greenfield, P., Droettboom, M., Bray, E., Aldcroft, T., Davis, M., Ginsburg, A., Price-Whelan, A. M., Kerzendorf, W. E., Conley, A., Crighton, N., Barbary, K., Muna, D., Ferguson, H., Grollier, F., Parikh, M. M., Nair, P. H., … Streicher, O. (2013). Astropy: A community Python package for astronomy. *Astronomy and Astrophysics*, *558*, A33. https://doi.org/10.1051/0004-6361/201322068

Bingham, E., Chen, J. P., Jankowiak, M., Obermeyer, F., Pradhan, N., Karaletsos, T., Singh, R., Szerlip, P. A., Horsfall, P., & Goodman, N. D. (2019). Pyro: Deep universal probabilistic programming. *J. Mach. Learn. Res.*, *20*, 28:1–28:6. http://jmlr.org/papers/v20/18-403.html

Birrer, S., Shajib, A. J., Gilman, D., Galan, A., Aalbers, J., Millon, M., Morgan, R., Pagano, G., Park, J. W., Teodori, L., Tessore, N., Ueland, M., Vyvere, L. V. de, Wagner-Carena, S., Wempe, E., Yang, L., Ding, X., Schmidt, T., Sluse, D., … Amara, A. (2021). Lenstronomy

II: A gravitational lensingW software ecosystem. *Journal of Open Source Software*, *6*(62), 3283. https://doi.org/10.21105/joss.03283

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., & Zhang, Q. (2018). *JAX: Composable transformations of Python+NumPy programs* (Version 0.3.13). http://github.com/google/jax

Bradley, L., Sipőcz, B., Robitaille, T., Tollerud, E., Vinícius, Z., Deil, C., Barbary, K., Wilson, T. J., Busko, I., Donath, A., Günther, H. M., Cara, M., Lim, P. L., Meßlinger, S., Conseil, S., Bostroem, A., Droettboom, M., Bray, E. M., Bratholm, L. A., … Souchereau, H. (2022). *Astropy/photutils: 1.5.0* (Version 1.5.0). Zenodo. https://doi.org/10.5281/zenodo.6825092

Costa-Luis, C. da, Larroque, S. K., Altendorf, K., Mary, H., richardsheridan, Korobov, M., Raphael, N., Ivanov, I., Bargull, M., Rodrigues, N., Chen, G., Lee, A., Newey, C., CrazyPython, JC, Zugnoni, M., Pagel, M. D., mjstevens777, Dektyarev, M., … Nordlund, M. (2023). *tqdm: A fast, Extensible Progress Bar for Python and CLI* (Version v4.65.0). Zenodo. https://doi.org/10.5281/zenodo.7697295

D'Avella, D., Jamieson, W., Droettboom, M., Slavich, E., Graham, B., Robitaille, T., Dencheva, N., perrygreenfield, Simon, B., MacDonald, K., Bray, E. M., Burnett, Z., Davies, J., Mumford, S., Markovtsev, V., Tollerud, E., Sipőcz, B., Bradley, L., Fabry, Ç., … Ginsburg, A. (2023). *Asdf-format/asdf: 2.15.0* (Version 2.15.0). Zenodo. https://doi.org/10.5281/zenodo.7799772

De Cao, N., Aziz, W., & Titov, I. (2020). Block neural autoregressive flow. *Uncertainty in Artificial Intelligence*, 1263–1273.

Ding, X., Birrer, S., Treu, T., & Silverman, J. D. (2021). Galaxy shapes of Light (GaLight): a 2D modeling of galaxy images. *arXiv e-Prints*, arXiv:2111.08721. https://doi.org/10.48550/arXiv.2111.08721

Erwin, P. (2015). IMFIT: A Fast, Flexible New Program for Astronomical Image Fitting. *Astrophysical Journal*, *799*(2), 226. https://doi.org/10.1088/0004-637X/799/2/226

Foreman-Mackey, D. (2016). Corner.py: Scatterplot matrices in python. *The Journal of Open Source Software*, *1*(2), 24. https://doi.org/10.21105/joss.00024

Geda, R., Crawford, S. M., Hunt, L., Bershady, M., Tollerud, E., & Randriamampandry, S. (2022). PetroFit: A Python Package for Computing Petrosian Radii and Fitting Galaxy Light Profiles. *Astronomical Journal*, *163*(5), 202. https://doi.org/10.3847/1538-3881/ac5908

Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. van, Brett, M., Haldane, A., Río, J. F. del, Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hoffman, M. D., Gelman, A., & others. (2014). The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *J. Mach. Learn. Res.*, *15*(1), 1593–1623.

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Kawinwanichakij, L., Silverman, J. D., Ding, X., George, A., Damjanov, I., Sawicki, M., Tanaka, M., Taranu, D. S., Birrer, S., Huang, S., Li, J., Onodera, M., Shibuya, T., & Yasuda, N. (2021). Hyper Suprime-Cam Subaru Strategic Program: A Mass-dependent Slope of the Galaxy Size-Mass Relation at z < 1. *Astrophysical Journal*, *921*(1), 38. https://doi.org/10.3847/1538-4357/ac1f21

Krekel, H., Oliveira, B., Pfannschmidt, R., Bruynooghe, F., Laugher, B., & Bruhin, F. (2004). *Pytest*. https://doi.org/10.1007/978-1-4842-2677-3_5

Kumar, R., Carroll, C., Hartikainen, A., & Martin, O. (2019). ArviZ a unified library for exploratory analysis of Bayesian models in Python. *Journal of Open Source Software*, *4*(33), 1143. https://doi.org/10.21105/joss.01143

Lang, D. (2020). A hybrid Fourier–Real Gaussian Mixture method for fast galaxy–PSF convolution. *arXiv e-Prints*, arXiv:2012.15797. https://doi.org/10.48550/arXiv.2012.15797

Lange, R., Driver, S. P., Robotham, A. S. G., Kelvin, L. S., Graham, A. W., Alpaslan, M., Andrews, S. K., Baldry, I. K., Bamford, S., Bland-Hawthorn, J., Brough, S., Cluver, M. E., Conselice, C. J., Davies, L. J. M., Haeussler, B., Konstantopoulos, I. S., Loveday, J., Moffett, A. J., Norberg, P., … Wilkins, S. M. (2015). Galaxy And Mass Assembly (GAMA): mass-size relations of z < 0.1 galaxies subdivided by Sérsic index, colour and morphology. *Monthly Notices of the RAS*, *447*(3), 2603–2630. https://doi.org/10.1093/mnras/stu2467

Mowla, L. A., van Dokkum, P., Brammer, G. B., Momcheva, I., van der Wel, A., Whitaker, K., Nelson, E., Bezanson, R., Muzzin, A., Franx, M., MacKenty, J., Leja, J., Kriek, M., & Marchesini, D. (2019). COSMOS-DASH: The Evolution of the Galaxy Size-Mass Relation since z ∼ 3 from New Wide-field WFC3 Imaging Combined with CANDELS/3D-HST. *Astrophysical Journal*, *880*(1), 57. https://doi.org/10.3847/1538-4357/ab290a

Nightingale, James. W., Amvrosiadis, A., Hayes, R. G., He, Q., Etherington, A., Cao, X., Cole, S., Frawley, J., Frenk, C. S., Lange, S., Li, R., Massey, R. J., Negrello, M., & Robertson, A. (2023). PyAutoGalaxy: Open-source multiwavelength galaxy structure & morphology. *Journal of Open Source Software*, *8*(81), 4475. https://doi.org/10.21105/joss.04475

Peng, C. Y., Ho, L. C., Impey, C. D., & Rix, H.-W. (2002). Detailed Structural Decomposition of Galaxy Images. *Astronomical Journal*, *124*(1), 266–293. https://doi.org/10.1086/340952

Phan, D., Pradhan, N., & Jankowiak, M. (2019). Composable effects for flexible and accelerated probabilistic programming in NumPyro. *arXiv Preprint arXiv:1912.11554*.

Ranganath, R., Gerrish, S., & Blei, D. (2014). Black box variational inference. *Artificial Intelligence and Statistics*, 814–822.

Robotham, A. S. G., Taranu, D. S., Tobar, R., Moffett, A., & Driver, S. P. (2017). PROFIT: Bayesian profile fitting of galaxy images. *Monthly Notices of the RAS*, *466*(2), 1513–1541. https://doi.org/10.1093/mnras/stw3039

Sersic, J. L. (1968). *Atlas de Galaxias Australes*.

Shajib, A. J. (2019). Unified lensing and kinematic analysis for any elliptical mass profile. *Monthly Notices of the RAS*, *488*(1), 1387–1400. https://doi.org/10.1093/mnras/stz1796

team, T. pandas development. (2020). *Pandas-dev/pandas: pandas* (latest). Zenodo. https://doi.org/10.5281/zenodo.3509134

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, *17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2